GMM Priors for Image Correlations

A

Thesis submitted

for the award of the degree of

DOCTOR OF PHILOSOPHY

by

Sandeep P

DEPARTMENT OF ELECTRONICS AND ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI

GUWAHATI - 781039, ASSAM, INDIA

June 2018

Dedicated to my
Parents, Teachers, Family and Friends

Certificate

This is to certify that the thesis entitled **"GMM Priors for Image Correlations"** submitted by **"Sandeep P"** (Roll No:11610203), a research scholar in the *Department of Electronics and Electrical Engineering, Indian Institute of Technology, Guwahati*, for the award of the degree of Doctor of Philosophy, is a record of an original research carried out by him under my supervision and guidance. The thesis has fulfilled all the requirements as per the regulations of the institute and in my opinion has reached the standard needed for submission. The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

Dr. Tony Jacob

Assistant Professor
Department of Electronics and Electrical Engineering
Indian Institute of Technology Guwahati
Guwahati, Assam - 781039, India

Date:
Guwahati

Acknowledgements

I would like to gratefully and sincerely thank my thesis supervisor Dr. Tony Jacob for his guidance, understanding, patience, and most importantly, his friendship during my doctoral studies at IIT Guwahati. His mentorship was of paramount importance in building a clear perspective to look at the research problems. The long discussions with him has always revealed the beauty and philosophy of various aspects of the research problems that we have worked on.

I would also like to thank my doctoral committee members, Prof. Prabin Kumar Bora, Prof. Rohit Sinha and Dr. Benny George for their valuable inputs, guidance and discussions. I would like to extend my sincere gratitude to Prof. Roy P Paily, who gave me an opportunity to work on some of the interesting projects. I sincerely thank Dr A. Rajesh, Dr. Ganesh Natarajan, Dr. Vibin Ramakrishnan, for their friendship, suggestions, and motivations. I would like to extend my thanks to Mr. Sanjib Das for providing high performance computational facilities. I thank 'Param Ishan', the high performance computing server, without which I would have been still continuing my experiments.

I would like to acknowledge my deeptest thanks to my parents for their constant support and motivations. I would like to extend my thanks to my family, friends, relatives and collegues who provided me with unending encouragement and support. It was under their watchful eye that I gained so much drive and an ability to tackle challenges head on. I thank all of them for their faith in me and allowing me to be as ambitious as I wanted. Especially, I would like to acknowledge the constant support provided by Jiss Joseph, Vasudevan, Arjun, Haris, Vijith, Rahul Nair and many others. Finally and most importantly, I would like to acknowledge my deepest thanks to my wife Roopa. I believe that her constant support, motivation and patience during the preparation of this thesis cannot be expressed in words.

Abstract

Patch based image restoration algorithms split a given degraded and/or noisy input image into the set of all possible overlapping patches, separately restore each of the patches, and finally compute the restored image by aggregating all the restored overlapping patches. Patch based algorithms exploit various types of patch priors to regularize the ill-posed inverse problems arising in commonly encountered image restoration problems. Patch priors encapsulate the essential characteristics typically exhibited by natural image patches. Recently, patch based image restoration algorithms exploiting Gaussian Mixture Models (GMMs) as a patch prior have been shown to produce impressive results in various image restoration problems. Recent works have also established the close connections between image restoration algorithms exploiting GMM priors, and the widely studied sparse representation or sparsity priors characterizing natural image patches. These works have proposed GMM analogues of several image restoration algorithms exploiting the sparsity prior. The GMM analogues generally offer superior performance, and are computationally faster when compared with the sparsity based algorithms.

In previous works, sparsity priors have also been shown to be good at jointly characterizing highly correlated natural image patches arising in certain image restoration problems such as Single Image Super Resolution (SISR) and color image restoration. In the case of SISR problem, a High Resolution (HR) patch and the corresponding Low Resolution (LR) patch are highly correlated as they differ only in scale. Similarly, the corresponding patches from different color channels, i.e., the patches around a common patch center, but from different color channels of a color image are generally observed to be highly correlated. In the literature, SISR algorithms exploiting sparsity prior for jointly characterizing the correlated HR-LR patch pairs, and color image restoration algorithms exploiting sparsity prior for jointly characterizing the corresponding patches from different color channels, have been demonstrated to achieve state of the art performance. However, joint characterization of correlated natural image patches using GMMs has not been previously studied in the literature. In this thesis, we investigate the potential of GMMs in jointly characterizing highly correlated natural image patches.

We propose GMM analogues of some of the previously studied image restoration algorithms exploiting sparsity prior for jointly characterizing natural image patches in SISR and color image restoration problems. We first propose a fast, example based SISR algorithm by exploiting a Joint GMM prior for characterizing the concatenated HR-LR patches. The

HR-LR patch correlations captured by the Joint GMM prior are exploited to estimate the unknown HR patch corresponding to an input LR patch. The proposed Joint GMM method can be interpreted as the GMM analogue of the sparsity based ScSR algorithm previously studied in literature. Next, we propose a SISR algorithm by using a GMM based regression method which can be seen as the GMM analogue of the recently proposed A+ algorithm delivering state of the art performance in patch based SISR. The GMM based regression method addresses the computational bottleneck of the learning method used for estimating the Joint GMM parameters from a given database of HR-LR patch pairs. The GMM based regression method significantly reduces the learning effort when compared with the Joint GMM method, and achieves superior performance in the case of large magnification factors. Finally, we propose color image restoration methods by exploiting a Joint Color Space GMM (JCS-GMM) prior that jointly characterizes the corresponding patches from different color channels of a color image. The JCS-GMM prior characterizes monochrome as well as color features, and captures the strong inter channel correlations typically observed in color images. We demonstrate the potential of the JCS-GMM prior by proposing JCS-GMM denoising and JCS-GMM demosaicking algorithms for addressing color image denoising and demosaicking. The proposed JCS-GMM method can be interpreted as the GMM analogue of the sparsity based Color-KSVD algorithm previously studied in literature. Our experiments demonstrate that the proposed algorithms achieve superior or competitive performance when compared with various state of the art SISR and color image restoration algorithms.

Contents

List of Figures

List of Tables

List of Abbreviations

ADMM	**A**lternating **D**irection **M**ethod of **M**ultipliers
A-EPLL	**A**daptive **E**xpected **P**atch **L**og **L**ikelihood
ANR	**A**nchored **N**eighborhood **R**egression
AP	**A**lternating **P**rojection
APCA	**A**daptive **P**rincipal **C**omponent **A**nalysis
ARI	**A**daptive **R**esidual **I**nterpolation
AWGN	**A**dditive **W**hite **G**aussian **N**oise
BPDN	**B**asis **P**ursuit **D**e**N**oising
BPJDL	**B**eta **P**rocess **J**oint **D**ictionary **L**earning
C-BM3D	**C**olor-**BM3D**
CFA	**C**olor **F**ilter **A**rray
CNN	**C**onvolutional **N**eural **N**etwork
CPSNR	**C**olor **P**eak **S**ignal to **N**oise **R**atio
CSSIM	**C**olor **S**tructural **S**imilarity **I**ndex **M**easure
DDR	**D**irectional **D**ifference **R**egression
DLIS	**D**iscriminatively **L**earned **I**terative **S**hrinkage
DLMMSE	**D**irectional **L**inear **M**inimum **M**ean **S**quare **E**rror
EM	**E**xpectation **M**aximization
E-PLE	**E**nhanced **PLE**
EPLL	**E**xpected **P**atch **L**og **L**ikelihood
ERP	**E**fficient **R**egression **P**riors
FR	**F**used **R**egression
FoE	**F**ield **o**f **E**xperts

GBTF	**G**radient **B**ased **T**hreshold **F**ree
GFM	**G**aussian **F**actor **M**odel
GMM	**G**aussian **M**ixture **M**odel
HBE	**H**yper prior **B**ayesian **E**stimator
HQS	**H**alf **Q**uadratic **S**plitting
HR	**H**igh **R**esolution
HSV	**H**ue **S**aturation **V**alue
IRI	**I**terative **R**esidual **I**nterpolation
IRVS	**I**mage **R**estoration algorithm based on **V**ariable **S**plitting
ISPA	**I**mage **S**pecific **P**rior **A**daptation
JCS-GMM	**J**oint **C**olor **S**pace **GMM**
KL divergence	**K**ullback-**L**eibler divergence
KRR	**K**ernel **R**idge **R**egression
KSVD	**K**-**S**ingular **V**alue **D**ecomposition
LLE	**L**ocally **L**inear **E**mbedding
LR	**L**ow **R**esolution
MAP	**M**aximum **A** **P**osteriori
ML	**M**aximum **L**ikelihood
MLRI	**M**inimized **L**aplacian **R**esidual **I**nterpolation
MP	**M**atching **P**ursuit
MRF	**M**arkov **R**andom **F**ield
MSG	**M**ultiscale **G**radients
MS-KSVD	**M**ulti **S**cale **KSVD**
NE	**N**eighborhood **E**mbedding
NL Bayes	**N**on **L**ocal **B**ayesian
NLM	**N**on **L**ocal **M**eans
OMP	**O**rthogonal **M**atching **P**ursuit
PCSD	**P**rimary **C**onsistent **S**oft **D**ecision
pdf	**p**robability **d**ensity **f**unction
PDS	**P**rimary **D**ifference **S**ignals
PLE	**P**iecewise **L**inear **E**stimator

POCS	**P**rojection **O**nto **C**onvex **S**ets
PSNR	**P**eak **S**ignal to **N**oise **R**atio
Q-KSVD	**Q**uaternion–**KSVD**
RBM	**R**estricted **B**oltzmann **M**achine
RI	**R**esidual **I**nterpolation
ScSR	**S**parse **c**oding based **S**uper **R**esolution
SCDL	**S**emi **C**oupled **D**ictionary **L**earning
SF	**S**imple **F**unctions
SIDI	**S**ingle frame **I**mage **D**enoising and **I**npainting
SISR	**S**ingle **I**mage **S**uper **R**esolution
SPM	**S**tatistical **P**rediction **M**odel
SR	**S**uper **R**esolution
SRCNN	**S**uper **R**esolution using deep **C**onvolutional **N**eural **N**etworks
SRF	**R**andom **F**orests
SSIM	**S**tructural **S**imilarity **I**ndex **M**easure
SURE	**S**tein's **U**nbiased **R**isk **E**stimate
TV	**T**otal **V**ariation
UBM	**U**niversal **B**ackground **M**odel

List of Symbols

$\mathbf{x}$	Vector representation of an unknown original monochrome or color image
N_x	Number of pixels in the original image
$\mathbf{y}$	Vector representation of degraded and/or noisy observed image
N_y	Number of pixels in the observed image
$\mathbf{A}$	Linear degradation operator which corrupts a clean image
$\mathbf{n}$	AWGN vector added to an observed noisy image
σ^2	Variance of AWGN
$\mathbf{H}$	Matrix operator corresponding to a blur kernel
$\mathbf{S}$	Matrix representing the subsampling operation applied on an image
$\|.\|_p$	l_p norm
$\mathbf{p}_i$	i-th unknown monochrome or concatenated color patch vector
$\mathbf{q}_i$	i-th degraded and/or noisy patch vector
L	Total number of image patches
$\mathbf{w}_i$	AWGN vector added to the i-th patch
τ_x	size of the unknown original patch
τ_y	size of the observed degraded and/or noisy patch
$\mathbf{B}_i$	Degradation operator restricted to the i-th patch
$\mathbf{\Theta}$	Set of GMM parameters
ω_k	Mixing weight of k-th Gaussian model from the GMM
$\boldsymbol{\mu}_k$	Mean vector of k-th Gaussian model from the GMM
$\mathbf{\Sigma}_k$	Covariance matrix of k-th Gaussian model from the GMM
K	Total number of Gaussian models that constitute the GMM
$\Phi\left(\right)$	Multivariate Gaussian pdf

k_i	Index of Gaussian model responsible for the i-th patch
$\mathcal{C}_k$	Cluster of patches associated to the k-th Gaussian model
$\mathbf{D}$	Overcomplete dictionary matrix
$\boldsymbol{\alpha}$	Sparse representation vector
T	Sparsity threshold
$\mathbf{D}_H$	Dictionary characterizing high resolution patches
$\mathbf{D}_L$	Dictionary characterizing low resolution patches
$\mathbf{p}_{iR}$	i-th patch from the Red channel of a color image
$\mathbf{p}_{iG}$	i-th patch from the Green channel of a color image
$\mathbf{p}_{iB}$	i-th patch from the Blue channel of a color image
$\mathbf{R}_i$	The matrix that extracts the i-th patch from an image
λ, β	Parameters of the EPLL algorithm
γ_{ik}, η_{ik}	Posterior responsibility of the k-th Gaussian model on the i-th patch
q	Magnification factor in super resolution
$\mathbf{W}_k$	Transformation matrix of the k-th cluster
$\mathcal{I}_a$	Set of indices of available pixels
$\mathcal{I}_m$	Set of indices of masked pixels
$\boldsymbol{\mu}_{H_k}$	Mean vector characterizing HR patches
$\boldsymbol{\mu}_{L_k}$	Mean vector characterizing LR patches
$\boldsymbol{\Sigma}_{H_k}$	Covariance matrix characterizing HR patches
$\boldsymbol{\Sigma}_{L_k}$	Covariance matrix characterizing LR patches
$\boldsymbol{\Sigma}_{HL_k}$	Cross covariance matrix characterizing HR-LR correlations
$\boldsymbol{\mu}_{R_k}$	Mean vector characterizing the patches from Red channel
$\boldsymbol{\mu}_{G_k}$	Mean vector characterizing the patches from Green channel
$\boldsymbol{\mu}_{B_k}$	Mean vector characterizing the patches from Blue channel
$\boldsymbol{\Sigma}_{R_k}$	Covariance matrix characterizing the patches from Red channel
$\boldsymbol{\Sigma}_{G_k}$	Covariance matrix characterizing the patches from Green channel
$\boldsymbol{\Sigma}_{B_k}$	Covariance matrix characterizing the patches from Blue channel
$\boldsymbol{\Sigma}_{RG_k}$	Cross covariance between patches from Red and Green channels
$\boldsymbol{\Sigma}_{RB_k}$	Cross covariance between patches from Red and Blue channels
$\boldsymbol{\Sigma}_{GB_k}$	Cross covariance between patches from Green and Blue channels

Chapter 1

Introduction

Natural image restoration algorithms address the problem of computing a high quality estimate of an image from one or several degraded observations. We focus on the image restoration algorithms which compute the estimate of an unknown natural image from a single degraded and/or noisy observation. In most of the commonly encountered natural image restoration problems, the degraded observations can be modeled as resulting from the combined effect of a linear operator and additive white Gaussian noise acting on the latent original image. Using vector representation obtained through lexicographically ordering the image pixels, a degraded image can be written as,

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n} \tag{1.1}$$

where, $\mathbf{y} \in \mathbb{R}^{N_y}$ denotes the degraded observation, $\mathbf{x} \in \mathbb{R}^{N_x}$ denotes the unknown original image, $\mathbf{A} \in \mathbb{R}^{N_y \times N_x}$ denotes the linear degradation operator, $\mathbf{n} \in \mathbb{R}^{N_y}$ is the additive white Gaussian noise, and, N_x and N_y respectively denote the number of pixels in the original and degraded images. The degradation of natural images in different image restoration problems can be modeled by appropriately choosing the degradation operator $\mathbf{A}$. In the case of Additive White Gaussian Noise (AWGN) denoising problem, $\mathbf{A} = \mathbf{I}$, where, $\mathbf{I}$ denotes the identity matrix, and, $\mathbf{n}$ denotes the AWGN noise vector assumed to be distributed as i.i.d Gaussian with noise variance σ^2. The image de-blurring problem corresponds to $\mathbf{A} = \mathbf{H}$, where, $\mathbf{H}$ denotes the blur kernel. In the case of Single Image

Super Resolution (SISR) problem which tries to estimate an HR image from a single LR observation, the degradation operator $\mathbf{A}$ can be modeled as $\mathbf{A} = \mathbf{SH}$, where, $\mathbf{H}$ denotes the blur kernel, and, $\mathbf{S}$ denotes the subsampling operator. The degraded input images in the case of denoising, de-blurring and SISR problems are shown in Figure 1.1.

FIGURE 1.1: Degraded images arising in different image restoration problems. (A) Original Image, (B) Noisy Image (denoising), (C) Blurred Image (de-blurring), (D) Low Resolution Image (super resolution)

The non-invertible nature of the degradation operator $\mathbf{A}$, and/or the presence of AWGN, makes single image restoration problems severely ill-posed with infinitely many candidate solutions. As a consequence of the assumption that the noise is i.i.d and Gaussian distributed, the intuitive approach in computing the estimate $\hat{\mathbf{x}}$ of the unknown original image $\mathbf{x}$ is to obtain the least square solution which targets to minimize (assuming that

the degradation operator $\mathbf{A}$ is known) $\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2$ with respect to $\mathbf{x}$. However, the large solution space arising from the ill-posedness of the problem may cause the estimated image obtained as the least square solution to contain a lot of spurious artifacts unlikely to be observed in natural images. The traditional approach in selecting an appropriate solution which resembles a natural image is to constrain the solution space by using various types of regularizers. The constrained least square solution can be obtained as,

$$\hat{\mathbf{x}} = \arg\min_{\tilde{\mathbf{x}}} \|\mathbf{y} - \mathbf{A}\tilde{\mathbf{x}}\|_2^2 + \lambda R(\tilde{\mathbf{x}}), \tag{1.2}$$

where, $R(.)$ is the regularization function, and, λ is the weight parameter deciding the relative importance of the data fidelity term $\|\mathbf{y} - \mathbf{A}\tilde{\mathbf{x}}\|_2^2$ and the regularizer $R(.)$. The fidelity term penalizes the candidate solutions which are inconsistent with the degraded observation $\mathbf{y}$. The regularization function models the basic characteristics typically observed in natural images and favors solutions which resemble natural images when compared with solutions containing spurious artifacts. A widely studied example of constrained least squares method is Total Variation (TV) minimization [1, 2, 3] which utilizes the sum of l_2 norm of gradient vectors at every pixel of the image as a regularization function. TV minimization method favors those solutions with smaller gradients, thereby exploiting the observation that natural image contents are piecewise smooth with sparsely located edges and corners leading to small gradient values.

In the probabilistic framework, the constrained least square solution $\hat{\mathbf{x}}$ obtained by using (1.2) can be interpreted as the Maximum A Posteriori (MAP) estimate of the unknown original image $\mathbf{x}$, computed from the observation corrupted by the known degradation operator $\mathbf{A}$ and AWGN. The regularization function $R(.)$ can bee seen as the prior model characterizing natural images. The performance of an image restoration algorithm depends heavily on the choice of the prior. A prior which is good at capturing the essential characteristics of natural images often leads to superior restoration performance. However, the MAP restoration framework given in (1.2) suffers from a serious drawback. The large dimensionality of natural images captured by modern high resolution cameras makes it difficult choose an appropriate prior model which globally characterizes the entire image. Moreover, the large dimensionality of natural images also makes the optimization problem

associated with MAP restoration framework given in (1.2) computationally intractable. On the other hand, natural images are easier to describe and to process, when restricted to smaller sized windows known as image patches. Consequently, most of the recent algorithms giving state of the art performance in various natural image restoration problems operate on natural image patches rather than globally operating on the entire image.

The patch based image restoration algorithms split a given degraded and/or noisy input image into several smaller sized (typically 8×8 or 9×9) patches, separately restore each of these patches, and finally aggregate all the restored patches by arranging them at their respective locations to produce the restored image. In order to avoid the blocky artifacts arising from the independent restoration of image patches, patch based algorithms generally operate on all possible overlapping patches of a given input image. Note that, each of the pixels (except the pixels on the image boundaries) appears in multiple patches while considering all possible overlapping patches. The restored estimate corresponding to a given pixel is computed by fusing the multiple estimates available from all the restored patches containing the given pixel. A commonly used method for fusing multiple estimates corresponding to a pixel is to average these estimates by using overlap-add method.

The image degradation model given in (1.1) can be equivalently written in the case of image patches as,

$$\mathbf{q}_i = \mathbf{B}_i \mathbf{p}_i + \mathbf{w}_i,\ i = 1, 2, ..., L, \tag{1.3}$$

where, $\mathbf{q}_i \in \mathbb{R}^{\tau_y^2}$ denotes the vector representation of the i-th degraded and/or noisy patch of size $\tau_y \times \tau_y$ pixels, $\mathbf{p}_i \in \mathbb{R}^{\tau_x^2}$ denotes the corresponding unknown original image patch of size $\tau_x \times \tau_x$ pixels, $\mathbf{B}_i \in \mathbb{R}^{\tau_y^2 \times \tau_x^2}$ denotes the linear degradation operator restricted to the i-th patch, $\mathbf{w}_i \in \mathbb{R}^{\tau_y^2}$ denotes the AWGN added to the i-th patch, and L denotes the total number of patches in the image. In the case of patch based algorithms, the MAP restoration framework given in (1.2) can be written as,

$$\hat{\mathbf{p}}_i = \underset{\tilde{\mathbf{p}}_i}{\arg\min}\ \|\mathbf{q}_i - \mathbf{B}_i \tilde{\mathbf{p}}_i\|_2^2 + \lambda_p R_p\left(\tilde{\mathbf{p}}_i\right),\ i = 1, 2, ..., L, \tag{1.4}$$

where, $\hat{\mathbf{p}}_i$ denotes the restored estimate of the i-th patch, $R_p\left(.\right)$ denotes the patch prior, and λ_p denotes the weight parameter deciding the relative weight assigned for the prior $R_p\left(.\right)$ when compared with the data fidelity term. As the minimization in (1.4) operates

on smaller sized patches instead of the whole image containing a large number of pixels, the optimization problem involved in the estimation of a single restored patch is computationally much simpler and numerically more stable. Note that the patch estimation given in (1.4) has to be repeated for all the patches $i = 1, 2, ..., L$. However, in many cases, the recent progress in parallel processing techniques such as efficient GPUs/multi-core processors enables parallel computation of a group of large number of restored patches (possibly, even the entire set of L patches), instead of sequentially iterating the patch estimation procedure on each of the patches.

The key factor deciding the performance of a patch based image restoration algorithm is the patch prior $R_p(.)$ which captures the essential characteristics of natural image patches. The patch based image restoration algorithms shift the requirement of good prior models for globally characterizing natural images, to the corresponding requirement of appropriate prior models for characterizing natural image patches. Though the patch based MAP restoration framework given in (1.4) requires that the degradation operator $\mathbf{A}$ (and hence the degradation operators $\{\mathbf{B}_i\}_{i=1}^{L}$ restricted to the patches) to be known, note that a good patch prior characterizing natural image patches might be beneficial even in the case of image restoration problems with unknown degradation operator $\mathbf{A}$. As an example, consider the SISR problem which tries to estimate the unknown HR image $\mathbf{x}$ from a single LR observation $\mathbf{y}$. Consider the situation that the degradation operator $\mathbf{A}$ producing the LR image is unknown, but we have a large database of HR and the corresponding LR images related through the same unknown degradation operator $\mathbf{A}$. In this scenario, a SISR algorithm can utilize the given database to learn the mapping from the space of LR images to the space of HR images by applying statistical learning techniques. Assuming that the learnt mapping generalizes also to the other images from the LR space which are not included in the database, the HR image corresponding to a given input LR image can be predicted by applying the learnt mapping on the given input LR image. A patch based single image super resolution algorithm would learn the mapping from the space of LR patches to the space of HR patches. Instead of directly learning the mapping between LR and HR patch spaces, previous works [4, 5, 6, 7] have demonstrated that the mapping between parameters of the priors characterizing LR and HR patches can be learnt to produce better results in single image super resolution. These works have effectively

utilized natural image patch priors either through constructing HR and LR patch priors [4, 5] under the assumption that the parameters of the priors follow a fixed mapping, or by explicitly learning [6, 7] the mapping between the parameters of the priors characterizing HR and LR patches.

Over the years, several patch priors, each having its own merits and demerits have been proposed in the literature. The central idea of many of the recently proposed [8, 9, 10, 11, 12] patch priors offering competitive results in various image restoration problems is to exploit the non-local self similarity of natural image patches. Among various patch priors previously proposed in the literature, the works reported in this thesis focus on two closely related patch priors, namely, the *Sparsity* priors and the *Gaussian Mixture Model (GMM)* priors. The sparsity prior exploits the observation that natural image patches can often be approximated as sparse linear combinations of a few prototype patches called *atoms* selected from a redundant collection of such atoms called the *dictionary*. Image restoration algorithms exploiting sparsity priors have been extensively studied [10, 13, 14, 15, 4, 5, 16, 17] in the literature, and these algorithms have proven to achieve impressive results in different image restoration problems such as image denoising [10, 13], inpainting [14, 15], super resolution [4, 5], de-blurring [16, 17] etc. The GMM prior characterizes natural image patches by approximating the statistical distribution of patch vectors using a mixture of multivariate Gaussian distributions. Each patch is assumed to be generated by a Gaussian model randomly selected from the mixture. Recent studies [11, 18, 19, 20] have demonstrated that image restoration algorithms exploiting GMM prior produce superior restoration performance with much faster speed of operation when compared with sparsity based algorithms. These studies have also revealed the close relations between the restored patch estimates computed by GMM and sparsity based algorithms. Specifically, the work reported in [11] has shown that the patch estimates computed by GMM based methods can be interpreted to be equivalent to the patch estimates obtained by sparsity based algorithms which exploit a structured variant of sparsity prior known as block sparsity [21, 22, 23]. Moreover, it was also demonstrated that there exist GMM analogues corresponding to many of the sparsity based image restoration algorithms. The GMM analogues offer superior restoration performance and faster speed of operation when compared with sparsity based algorithms. The close relations between GMM and sparsity

based algorithms has motivated the previous works [11, 18] to propose GMM analogues corresponding to several sparsity based algorithms.

In this thesis, we further extend the study on GMM based algorithms by focusing on image restoration problems in which several *highly correlated* natural image patches are available. In some of the image restoration problems such as example based SISR and color image restoration, we can observe the availability of a set of highly correlated natural image patches. In the case of example based (training based) SISR problem with a given training database containing several example HR and the corresponding LR images, the contents of HR and the corresponding LR patches extracted from a given pair of HR-LR images can be observed to be highly correlated, as these patches vary only in scale. Similarly, in the case of color image restoration problems, the contents of corresponding patches from Red, Green and Blue channels of a color image are generally observed to be highly correlated. Though these correlated patches can be independently characterized by any of the existing patch priors, note that the conventional patch priors cannot characterize the correlated structures available in these patches. In the case of image restoration problems providing access to such highly correlated natural image patches, exploiting the correlated structures can significantly improve the performance of image restoration algorithms utilizing patch priors which independently characterize these patches. Specifically, image restoration algorithms can improve the restoration performance by using a patch prior which captures the correlated structures by jointly characterizing the highly correlated patches. Image restoration algorithms exploiting sparsity prior for characterizing the correlated natural image patches have already been studied in the literature [4, 24, 25, 13]. These algorithms have demonstrated to produce state of the art results in the case of image restoration problems such as SISR [4, 24] and color image restoration [25, 13]. Motivated by the close relations between image restoration algorithms based on sparsity and GMM priors, it is natural to investigate about the GMM analogues corresponding to the sparsity based algorithms exploiting patch correlations in different image restoration problems. In this thesis, we study the GMM analogues corresponding to the previously proposed sparsity based algorithms [4, 24, 25] exploiting the strong correlations between natural image patches in SISR and color image restoration problems.

1.1 Thesis Contributions

The main contributions of this thesis can be summarized as follows. We propose the idea of exploiting GMM priors for jointly characterizing correlated natural image patches arising in some of the image restoration problems. Based on these GMM priors, we propose different patch based image restoration algorithms addressing example based SISR and color image restoration problems.

In the case of SISR, we address the problem of estimating an unknown HR image from a single LR observation by exploiting a given training database containing several examples of HR and corresponding LR images. We propose two different SISR algorithms which predict the unknown HR patches corresponding to the LR patches from a given input image, by using the HR-LR patch correlations learnt from the given training database.

- **Single Image Super Resolution using Joint GMM Method** - We propose a Joint GMM method for SISR by jointly characterizing HR and the corresponding LR patches using a GMM prior. The Joint GMM prior characterizes the patch vectors obtained by concatenating HR and the corresponding LR patches. The parameters of the Joint GMM prior are learnt from a set of concatenated HR-LR patches sampled from the given training database. The Joint GMM prior consists of several Gaussian models whose covariance matrices capture the inherent correlations between HR and LR patches. These correlations captured by the covariance matrices are utilized for estimating the HR patches corresponding to the LR patches from a given input image to be super resolved. The proposed Joint GMM method can be interpreted as the GMM analogue of the ScSR algorithm [4] exploiting sparsity prior for jointly characterizing the HR-LR patch pairs.

- **Single Image Super Resolution using GMM based Regression Method** - We propose a SISR algorithm using a GMM based regression method which can be seen as the GMM analogue of recently proposed A+ algorithm [24] delivering state of the art performance in patch based SISR. The proposed GMM based regression method exploits a generic GMM prior to partition the entire set of LR training patches into as many clusters as the number of Gaussian models from the prior. Assuming that

all the LR training patches from a given cluster are mapped to the corresponding HR training patches through a simple linear transformation, the proposed method estimates the transformation matrix corresponding to each of the clusters by using regularized least squares regression. Given an LR patch from the input image to be magnified, the GMM based regression method estimates the corresponding HR patch by first selecting the best cluster matched to the given LR patch, and then applying the linear transformation corresponding to the selected cluster.

In the case of color image restoration, we address the problem of recovering a high quality estimate of a color image from the degraded and/or noisy observation. We propose a Joint Color Space GMM (JCS-GMM) method for addressing color image restoration problems by learning a GMM prior for jointly characterizing the corresponding patches from Red, Green and Blue channels of a color image. The JCS-GMM prior characterizes the vectors obtained by concatenating the corresponding patch vectors from Red, Green and Blue channels of a color image. The parameters of the JCS-GMM prior are learnt form a large external database of natural color image patches. The covariance matrices of the Gaussian models capture the strong inter channel correlations typically observed between different spectral components of a color image. We propose color image denoising and demosaicking algorithms to demonstrate the potential of JCS-GMM prior in exploiting inter channel correlations. The proposed JCS-GMM method can be interpreted as the GMM analogue of Color-KSVD algorithm [25].

- **JCS-GMM Denoising** - We propose JCS-GMM denoising algorithm to address the problem of estimating a noise free version of a color image from a noisy observation corrupted by AWGN. The proposed JCS-GMM denoising algorithm computes the denoised estimate of a noisy color image by using the GMM based EPLL denoising framework [18]. In contrast with the fixed, externally learnt GMM parameters used by the EPLL framework, we propose a controlled GMM adaptation scheme to adapt the externally learnt JCS-GMM parameters to the specific image being denoised.

- **JCS-GMM Demosaicking** - We propose JCS-GMM Demosaicking algorithm to address the problem of estimating the full color version of a mosaicked color image produced by Color Filter Arrays (CFAs) commonly employed in acquiring color

images using digital cameras. The proposed JCS-GMM demosaicking algorithm computes the demosaicked estimate of a mosaicked color image by using an iterative method which alternates several times between Patch Estimation and GMM Adaptation steps. The Patch Estimation step computes demosaicked estimates of the color image patches by using the JCS-GMM parameters obtained from the previous iteration (or initialization). The GMM Adaptation step updates the JCS-GMM parameters by using the estimates of the color image patches obtained from the Patch Estimation step.

1.2 Thesis Organization

The organization of this thesis is as follows:

Chapter 2 contains a detailed review of previous works on image restoration algorithms exploiting GMM prior for characterizing natural image patches. This chapter also contains a brief review of image restoration algorithms exploiting sparsity prior, and the connections between sparsity and GMM based algorithms. A brief review of image restoration algorithms exploiting sparsity prior for jointly characterizing the correlated natural image patches in SISR and color image restoration problems is also given in this chapter.

Chapter 3 describes the proposed SISR algorithm based on the Joint GMM method. This chapter includes a review of literature on patch based SISR algorithms, detailed description of the proposed method, and the details of experiments carried out for comparing the performance of the proposed algorithm with that of several state of the art SR algorithms.

Chapter 4 is devoted to the proposed SISR algorithm based on the GMM based regression method. This chapter motivates the proposed GMM based regression method by discussing some of the drawbacks of Joint GMM method proposed in chapter 3, and gives a detailed description of the proposed alorithm and the results of SR experiments carried out for evaluating the proposed method.

Chapter 5 describes the JCS-GMM method proposed for addressing color image denoising and demosaicking problems. This chapter includes a review of literature on color image denoising and demosaickig algorithms, detailed description of the proposed JCS-GMM denoising and JCS-GMM demosaicking algorithms, and the details of experiments carried out for evaluating the performance of the proposed algorithms.

Chapter 6 concludes the thesis with a brief overview of future research related to the work reported in this thesis.

Chapter 2

Literature Review

In this chapter, we review the literature on image restoration algorithms exploiting GMM and sparsity priors for characterizing natural image patches. We first review the literature on GMM based image restoration algorithms in sections 2.1 - 2.4 of this chapter, and discuss the close connections between image restoration algorithms exploiting GMM and sparsity priors. Section 2.5 of this chapter describes the previous works on sparsity based image restoration algorithms which exploit the structures available in highly correlated natural image patches. The single image super resolution and color image restoration algorithms proposed in this thesis can be interpreted as GMM analogues of the sparsity based algorithms reviewed in section 2.5.

2.1 GMM Prior for characterizing Natural Image Patches

The vectorized representations of natural image patches are often considered as realizations of a vector valued random variable. The aim of any statistical prior characterizing natural image patches is to model the probability density function (pdf) of the random variable generating the image patches. The non Gaussian nature of complex structures typically observed in natural image patches prevents the possibility of modeling natural image patches as realizations of a single Gaussian distribution. However, being universal approximators of probability density functions, Gaussian Mixture Models(GMMs) can

reasonably approximate the pdf of natural image patches when sufficient number of Gaussian models are available in the mixture. The GMM prior characterizes natural image patches by approximating the statistical distribution of patch vectors using a mixture of multivariate Gaussian distributions.

The GMM prior characterizing natural image patches of size $\tau \times \tau$ pixels (corresponding to the patch vectors belonging to $\mathbb{R}^{\tau^2}$) is specified by parameters $\boldsymbol{\Theta} = \{\omega_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{K}$, where, K denotes the total number of Gaussian models in the mixture, and, $\omega_k \in [0, 1]$, $\boldsymbol{\mu}_k \in \mathbb{R}^{\tau^2}$ and $\boldsymbol{\Sigma}_k \in \mathbb{R}^{\tau^2 \times \tau^2}$ respectively denote the mixing weight, mean vector and covariance matrix of the Gaussian model with index k for $k = 1, 2, ..., K$. Each natural image patch $\mathbf{p} \in \mathbb{R}^{\tau^2}$ is assumed to be generated by one of the K Gaussian models randomly selected from the mixture. The mixing weight ω_k associated with the k-th Gaussian model signifies the a priori probability that the k-th Gaussian model is responsible for generation of a randomly selected patch. Hence, the mixing weights satisfy the condition $\sum_{k=1}^{K} \omega_k = 1$. According to the GMM prior, the pdf of a randomly selected patch $\mathbf{p}$ can be written as,

$$f(\mathbf{p}) = \sum_{k=1}^{K} \omega_k \Phi(\mathbf{p}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{2.1}$$

where, $f(.)$ denotes the pdf corresponding to the GMM prior, and, $\Phi(.; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes the multivariate Gaussian pdf with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, i.e.,

$$\Phi(\mathbf{p}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{\frac{\tau^2}{2}} |\boldsymbol{\Sigma}|^{\frac{1}{2}}} \exp\left[-\frac{1}{2}(\mathbf{p} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{p} - \boldsymbol{\mu})\right]. \tag{2.2}$$

In practice, the parameters $\boldsymbol{\Theta} = \{\omega_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{K}$ of the GMM prior are not known, and hence need to be estimated. Image restoration algorithms exploiting GMM prior utilize different techniques to obtain the estimates $\hat{\boldsymbol{\Theta}} = \left\{\hat{\omega}_k, \hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\Sigma}}_k\right\}_{k=1}^{K}$ of the GMM parameters. Some of the algorithms [11, 20] estimate the GMM parameters solely by using the given degraded input image to be restored, while some other algorithms [18] estimates these parameters from a large corpus of natural image patches sampled from an external database of clean natural images. We denote the GMM prior obtained from the former method by the name internal GMM prior as it does not require any external database for estimating the GMM parameters. As such, the GMM prior obtained from the later method is denoted

by the name external GMM prior.

One of the initial attempts in exploiting GMM prior for characterizing natural image patches might be the image denoising algorithm proposed in [26] by using Adaptive Principal Component Analysis (APCA). The APCA based algorithm computes a PCA basis for each of the image patches by using a set of spatially neighboring noisy patches. The LMMSE estimates of the PCA coefficients of original unknown patches are computed from the PCA coefficients of the noisy patches, and, the patch estimates are obtained by projecting back the estimated PCA coefficients to the patch domain. Though APCA based image denoising algorithm does not explicitly use a GMM prior, patch characterization using local PCAs is equivalent to assuming that a given patch and its spatial neighbors are realizations of a multivariate Gaussian random variable. The entire set of patches in a given image can be interpreted as characterized by a mixture of Gaussian distributions with as many Gaussian models as the number of patches in the image. An early attempt in explicitly utilizing the GMM prior for characterizing natural image patches might be the work proposed in [27]. In the image denoising algorithm proposed in [27], image patches are characterized using a GMM prior whose parameters are learnt from the noisy input patches by utilizing the Expectation Maximization (EM) algorithm. Given the parameters of the GMM prior, the denoised estimates of the patches are obtained as the MMSE estimates under GMM prior. In the recent literature, mainly there are three variants of image restoration algorithms exploiting GMM prior in different image restoration problems – the PLE algorithm [11], the EPLL algorithm [18] and the NL-Bayes algorithm [20]. In the following sections, we discuss PLE, EPLL and NL-Bayes algorithms, and briefly review some of the algorithms closely related to them.

2.2 The PLE Algorithm

The PLE algorithm was proposed in [11] as a general framework for addressing the commonly encountered image inverse problems such as image interpolation, super resolution, and de-blurring. The PLE algorithm characterizes natural image patches by using an internal GMM prior with parameters $\{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{K}$. The mixing weight ω_k is assumed to be same and is equal to $\frac{1}{K}$ for all the Gaussian models. Given a degraded image assumed to be

generated as in 1.1, the PLE algorithm extracts all the degraded patches $\mathbf{q}_i$, $i = 1, 2, ..., L$, corresponding to the unknown original image patches $\{\mathbf{p}_i\}_{i=1}^{L}$, and computes the restored estimates $\{\hat{\mathbf{p}}_i\}_{i=1}^{L}$ of all the unknown patches. The i-th degraded patch $\mathbf{q}_i$ is assumed to be generated from the corresponding unknown original image patch $\mathbf{p}_i$ as,

$$\mathbf{q}_i = \mathbf{B}_i\mathbf{p}_i + \mathbf{w}_i, \tag{2.3}$$

for $i = 1, 2, ..., L$, where, $\mathbf{B}_i$ denotes the linear degradation operator restricted to the i-th patch, and $\mathbf{w}_i$ denotes noise vector assumed to be distributed as $\mathbf{w}_i \sim \mathcal{N}\left(\mathbf{0}, \sigma^2\mathbf{I}\right)$, where, σ^2 is the noise variance. The restored estimates $\{\hat{\mathbf{p}}_i\}_{i=1}^{L}$ are computed by using an iterative MAP-EM algorithm [11]. In every iteration, MAP-EM algorithm alternates between the Patch Estimation step (denoted as E-Step in [11]) and the GMM update step (denoted as M-Step). The Patch Estimation step computes the estimates $\{\tilde{\mathbf{p}}_i\}_{i=1}^{L}$ of the unknown patches by using the GMM parameters obtained from the previous iteration, and, the GMM update step updates the parameters of the GMM prior by using the estimates $\{\tilde{\mathbf{p}}_i\}_{i=1}^{L}$ obtained from the Patch Estimation step. The Patch Estimation and GMM update steps are alternatively iterated several times (typically 4-5 iterations), and the patch estimates $\{\tilde{\mathbf{p}}_i\}_{i=1}^{L}$ obtained in the final iteration are chosen as the restored patches $\{\hat{\mathbf{p}}_i\}_{i=1}^{L}$. The MAP-EM algorithm requires an initial GMM prior to start Patch Estimation step of the very first iteration. In the following, we describe the details of Patch Estimation and GMM update step by assuming that the GMM parameters are initialized by using some method. The exact procedure used for initializing the GMM parameters is discussed at the end of this section.

2.2.1 Patch Estimation Step

The patch estimation step computes the estimates $\{\tilde{\mathbf{p}}_i\}_{i=1}^{L}$ of the unknown original image patches by using the GMM parameters $\hat{\mathbf{\Theta}} = \left\{\hat{\boldsymbol{\mu}}_k, \hat{\mathbf{\Sigma}}_k\right\}_{k=1}^{K}$ obtained from the previous iteration (or initialization). According to the GMM prior, each unknown image patch $\mathbf{p}_i$ is assumed to be distributed as $\mathbf{p}_i \sim \mathcal{N}\left(\hat{\boldsymbol{\mu}}_{k_i}, \hat{\mathbf{\Sigma}}_{k_i}\right)$, where, k_i is the mixture index of the Gaussian model responsible for the i-th patch $\mathbf{p}_i$. If the index k_i of the Gaussian model responsible for the i-th patch is known a priori, the estimate $\tilde{\mathbf{p}}_i$ can be computed as the

argument $\mathbf{g}$ maximizing the a posteriori probability $p\left(\mathbf{g}\left|\mathbf{q}_i;\hat{\boldsymbol{\mu}}_{k_i},\hat{\boldsymbol{\Sigma}}_{k_i}\right.\right)$. However, as k_i is unknown for all the patches, the patch estimation step first computes the estimate $\tilde{\mathbf{p}}_i^k$ defined as,

$$\begin{aligned}\tilde{\mathbf{p}}_i^k &= \arg\max_{\mathbf{g}} \log p\left(\mathbf{g}\left|\mathbf{q}_i;\hat{\boldsymbol{\mu}}_k,\hat{\boldsymbol{\Sigma}}_k\right.\right)\\ &= \arg\max_{\mathbf{g}} \log p\left(\mathbf{q}_i\left|\mathbf{g};\hat{\boldsymbol{\mu}}_k,\hat{\boldsymbol{\Sigma}}_k\right.\right)+\log p\left(\mathbf{g}\left|\hat{\boldsymbol{\mu}}_k,\hat{\boldsymbol{\Sigma}}_k\right.\right)\\ &= \arg\min_{\mathbf{g}} \frac{1}{\sigma^2}\left\|\mathbf{B}_i\mathbf{g}-\mathbf{q}_i\right\|_2^2+\left(\mathbf{g}-\hat{\boldsymbol{\mu}}_k\right)^T\hat{\boldsymbol{\Sigma}}_k^{-1}\left(\mathbf{g}-\hat{\boldsymbol{\mu}}_k\right) \qquad (2.4)\end{aligned}$$

with all the Gaussian models $k=1,2,...,K$, where, the second equality follows from the Bayes rule and the third line follows from the assumption $\mathbf{w}_i\sim\mathcal{N}\left(\mathbf{0},\sigma^2\mathbf{I}\right)$ and $\mathbf{p}_i\sim\mathcal{N}\left(\hat{\boldsymbol{\mu}}_k,\hat{\boldsymbol{\Sigma}}_k\right)$. The solution of (2.4) which minimizes the risk $\mathbb{E}\left[\left\|\tilde{\mathbf{p}}_i^k-\mathbf{p}_i\right\|_2^2\right]$ can be obtained in closed form as,

$$\begin{aligned}\tilde{\mathbf{p}}_i^k &= \hat{\boldsymbol{\mu}}_k+\hat{\boldsymbol{\Sigma}}_k\mathbf{B}_i^T\left(\mathbf{B}_i\hat{\boldsymbol{\Sigma}}_k\mathbf{B}_i^T+\sigma^2\mathbf{I}\right)^{-1}\left(\mathbf{q}_i-\hat{\boldsymbol{\mu}}_k\right)\\ &= \hat{\boldsymbol{\mu}}_k+\mathbf{W}_{ki}\left(\mathbf{q}_i-\hat{\boldsymbol{\mu}}_k\right), \qquad (2.5)\end{aligned}$$

where,

$$\mathbf{W}_{ki}=\hat{\boldsymbol{\Sigma}}_k\mathbf{B}_i^T\left(\mathbf{B}_i\hat{\boldsymbol{\Sigma}}_k\mathbf{B}_i^T+\sigma^2\mathbf{I}\right)^{-1} \qquad (2.6)$$

denotes the Weiner filter derived from the covariance matrix of the k-th Gaussian model. The index k_i of the Gaussian model responsible for the i-th patch is then estimated by using $\left\{\tilde{\mathbf{p}}_i^k\right\}_{k=1}^K$ as,

$$\tilde{k}_i=\arg\min_k\left(\frac{1}{\sigma^2}\left\|\mathbf{B}_i\tilde{\mathbf{p}}_i^k-\mathbf{q}_i\right\|_2^2+\left(\tilde{\mathbf{p}}_i^k-\hat{\boldsymbol{\mu}}_k\right)^T\hat{\boldsymbol{\Sigma}}_k^{-1}\left(\tilde{\mathbf{p}}_i^k-\hat{\boldsymbol{\mu}}_k\right)+\log\left|\hat{\boldsymbol{\Sigma}}_k\right|\right) \qquad (2.7)$$

As the index k_i of the Gaussian model is now estimated as $\tilde{k}_i$, the required estimate $\tilde{\mathbf{p}}_i$ of the i-th patch is obtained as,

$$\tilde{\mathbf{p}}_i=\tilde{\mathbf{p}}_i^{\tilde{k}_i}. \qquad (2.8)$$

The Patch Estimation step uses (2.5), (2.7) and (2.8) to compute the estimate $\tilde{\mathbf{p}}_i$ for all the patches $i=1,2,...,L$. The overall Patch Estimation step implements a Piecewise Linear Estimator (PLE) involving K linear filters and a non-linear model selection.

2.2.2 GMM Update Step

The GMM update step utilizes the estimates $\{\tilde{\mathbf{p}}_i\}_{i=1}^{L}$ obtained from the Patch Estimation Step to update the parameters $\left(\hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\Sigma}}_k\right)$ of all the Gaussian models $k = 1, 2, ..., K$. Note that the Patch Estimation step selects the index $\tilde{k}_i$ of the Gaussian model responsible for each patch $i = 1, 2, ..., L$. The Gaussian model selection procedure introduces a clustering on the entire set of patches as it assigns each patch to one of the Gaussian models. Let $\mathcal{C}_k$ be the cluster of patch estimates $\{\tilde{\mathbf{p}}_i\}$ associated to the k-th Gaussian model, i.e.,

$$\mathcal{C}_k = \left\{ \tilde{\mathbf{p}}_i \,\middle|\, i : \tilde{k}_i = k \right\}, \tag{2.9}$$

for $k = 1, 2, ..., K$. The GMM update step estimates the parameters of the k-th Gaussian model by using the patches from the cluster $\mathcal{C}_k$ for $k = 1, 2, ..., K$. Given $\mathcal{C}_k$, the parameters of the k-th Gaussian model are updated by using the ML estimates,

$$\hat{\boldsymbol{\mu}}_k = \frac{1}{|\mathcal{C}_k|} \sum_{\tilde{\mathbf{p}}_i \in \mathcal{C}_k} \tilde{\mathbf{p}}_i, \;\; \hat{\boldsymbol{\Sigma}}_k = \frac{1}{|\mathcal{C}_k|} \sum_{\tilde{\mathbf{p}}_i \in \mathcal{C}_k} \left(\tilde{\mathbf{p}}_i - \hat{\boldsymbol{\mu}}_k\right) \left(\tilde{\mathbf{p}}_i - \hat{\boldsymbol{\mu}}_k\right)^T \tag{2.10}$$

The Patch Estimation and GMM update steps are iterated several times (typically 4 to 5 iterations) and the patch estimates $\{\tilde{\mathbf{p}}_i\}_{i=1}^{L}$ obtained in the final iteration are chosen as the restored patches $\{\hat{\mathbf{p}}_i\}_{i=1}^{L}$. The overlapping restored patches $\{\hat{\mathbf{p}}_i\}_{i=1}^{L}$ are arranged at their respective locations and averaged by using overlap-add method to compute the estimate $\hat{\mathbf{x}}$ of the unknown original image $\mathbf{x}$.

2.2.3 Initialization of GMM Prior

The MAP-EM algorithm requires the parameters of the GMM prior to be initialized. Instead of the conventional method of initializing the GMM prior with random parameter values, the PLE algorithm adopts a carefully designed initialization strategy based on the connections between GMM and sparsity priors. The PLE algorithm observes that the eigenvectors of covariance matrices of Gaussian models from the GMM prior can be interpreted as atoms of an overcomplete dictionary used by the sparsity based algorithms exploiting block sparsity structure. Previous works on sparsity based image restoration

algorithms [10] have observed that the prominent dictionary atoms learnt from an example database of natural image patches are similar to the local edges [11]. Inspired from the connections between GMM and sparsity priors, the PLE algorithm initializes the GMM prior in such a way that the covariance matrices of Gaussian models capture the directional edge regularities. The possible edge directions between 0 and π are sampled at K angles, where, K is the total number of Gaussian models, and one covariance matrix is computed for each direction. The covariance matrix corresponding to a given edge direction is initialized by using the patches extracted from a synthetic black and white image containing an edge feature with the given direction. The estimate of the covariance matrix is computed as the sample covariance matrix of the local patches touching the edge contour at different locations.

It has been observed that the average PSNR performance achieved by the PLE algorithm in the case of image inverse problems increases as the number of Gaussian models K increases, and gets saturated when $K = 36$. As a compromise between performance and complexity, the number of Gaussian models in the mixture is chosen as $K = 19$. Apart from the 18 Gaussian models characterizing directional edge features, a 19-th Gaussian model whose covariance matrix is initialized by choosing DCT basis as the eigen vectors has also been included in the mixture for characterizing textured and flat patches. The mean vectors of all the Gaussian models are initialized to $\mathbf{0}$.

2.2.4 Connections between GMM and Sparsity Priors

The sparsity prior assumes that natural image patches can be often approximated as linear combinations of a few prototype patches called *atoms*, selected from a possibly redundant/overcomplete collection of such atoms, called the *dictionary*. Arranging the (vectorized) prototype patches $\left\{\mathbf{d}_j \in \mathbb{R}^{\tau^2}\right\}_{j=1}^{J}$ as the columns of a dictionary matrix $\mathbf{D} \in \mathbb{R}^{\tau^2 \times J}$, where, J is the total number atoms, a clean natural image patch $\mathbf{p}_i$ is approximated as

$$\mathbf{p}_i \approx \mathbf{D}\boldsymbol{\alpha}_i \text{ s.t } \|\boldsymbol{\alpha}_i\|_0 \leq T, \tag{2.11}$$

where, $\boldsymbol{\alpha}_i \in \mathbb{R}^J$ denotes the sparse representation vector, and $\|\boldsymbol{\alpha}\|_0$ denotes the l_0 pseudo norm which is the number of non-zero entries of the vector $\boldsymbol{\alpha}$. The constraint on l_0

pseudo norm makes the vector $\boldsymbol{\alpha}$ sparse, with number of non-zero entries not exceeding T. The sparsity based image restoration techniques exploit sparse coding algorithms to first compute the estimate $\hat{\boldsymbol{\alpha}}_i$ of the sparse representation vector $\boldsymbol{\alpha}_i$, and estimates an unknown patch $\mathbf{p}_i$ as,

$$\hat{\mathbf{p}}_i = \mathbf{D}\hat{\boldsymbol{\alpha}}_i. \tag{2.12}$$

A conventional sparse coding algorithm has $\binom{J}{T}$ degrees of freedom in choosing the dictionary atoms contributing to the linear approximation of a given patch. In a typical example with $J = 256$ and $T = 8$, the sparse coding algorithm has $\sim 10^{14}$ degrees of freedom in choosing the dictionary atoms. This large degrees of freedom makes the computation of sparse representations unstable and imprecise [11, 28]. The instabilities arising from the large degrees of freedom can be reduced by using structured sparsity priors. The structured sparsity priors regularize the sparse approximation problem by imposing further structures on the sparsity pattern, i.e., the pattern of non-zero coefficients of the sparse representation vector. The block sparsity structure [21, 22, 23], group sparsity [29, 30] etc. are some of the examples of structured sparsity models.

The block sparsity model [21, 22, 23] assumes that the J dictionary atoms, and the corresponding sparse representations coefficients, can be grouped into K non overlapping blocks. The block sparsity model approximates an image patch as linear combination of atoms from a *few blocks* of the dictionary. A block sparse coding algorithm selects the best T_B (out of K) blocks of dictionary atoms to approximate a given image patch. Thus, the block sparse coding algorithms have $\binom{K}{T_B}$ degrees of freedom in choosing the non-zero blocks. As the number of blocks K is usually much smaller than J, the $\binom{K}{T_B}$ degrees of freedom available for block sparse coding algorithms is much smaller than $\binom{J}{T}$, and the sparse approximations produced by the block sparse coding algorithms are more stable [11] than the sparse approximations obtained by using conventional sparse coding algorithms. In the PLE algorithm [11], the authors have observed that the patch estimates computed by exploiting a GMM prior can be interpreted to be equivalent to the patch estimates obtained by exploiting block sparsity models. The union of K PCA bases corresponding to the K covariance matrices of Gaussian models can be interpreted as a block dictionary with K blocks. The GMM based algorithms thus enjoy the stability advantages of block

sparsity models without explicitly using the computationally heavy block sparse coding algorithms.

2.2.5 Related Algorithms

The S-PLE algorithm proposed in [31] for addressing the natural image denoising problem observes that covariance matrices of most of the Gaussian models from the GMM prior used by the PLE algorithm are practically singular with a rank smaller than the patch dimension. This observation has motivated the S-PLE algorithm to characterize natural image patches using a mixture of K Gaussian Factor Models (GFMs) instead of GMMs used by the PLE algorithm. The factor model [32] is a linear generative model good at probabilistically characterizing data samples from manifolds with an ambient dimension smaller than the dimension of observed samples. In contrast with the method of assigning equal weights for all the mixture components as practiced in PLE, the S-PLE algorithm exploits the conventional mixture which assigns appropriate mixing weights for different factor models. The S-PLE algorithm utilizes a mixture of 20 Gaussian factor models, in which, 18 factor models are chosen to capture directional features as in the case of PLE, and the remaining two factor models characterize multi-oriented features and flat patches.

Instead of the synthetic patch based initialization scheme used by the PLE algorithm, S-PLE initializes the factor models by using a database of natural image patches containing different directional features. The patches from the input image to be denoised are utilized to update the parameters of the mixture model by using EM algorithm with a Stein's Unbiased Risk Estimate (SURE) based stopping criterion. The S-PLE algorithm selects the best Gaussian factor model responsible for each of the noisy patches and computes the denoised estimate by using a Weiner filter derived from the selected factor model. Apart from using factor models in the place of full covariance Gaussian models used by the PLE algorithm, the S-PLE algorithm also introduces several numerical tweaks on the original PLE algorithm, enabling S-PLE to achieve superior performance in image denoising. The E-PLE (Enhanced PLE) algorithm proposed in [33] extended the GFM based S-PLE denoising algorithm into the case of image inpainting (interpolation).

The Single frame Image Denoising and inpainting (SIDI) algorithm proposed in [34] can be observed to be closely related to the PLE algorithm. In contrast with the PLE algorithm, the SIDI algorithm characterizes natural image patches by using the traditional GMM prior which includes a mixing weight ω_k for each of the Gaussian models $k = 1, 2, ..., K$. The PLE algorithm performs a hard clustering on the entire set of patches from an image by assigning each given patch to one of the Gaussian models which is most likely to be responsible for the given patch. The restored patch corresponding to a given degraded patch is computed by using a Wiener filter derived from the covariance matrix of the Gaussian model to which the given patch is assigned. On the other hand, the SIDI algorithm computes the restored patch as the MMSE estimate obtained under GMM prior. The MMSE estimate corresponds to the weighted linear combination of hard clustering based patch estimates produced by different Gaussian models. Both PLE and SIDI algorithms characterize natural image patches by using an internal GMM prior learnt from the degraded input image itself. However, the PLE algorithm computes the parameters of different Gaussian models by using the cluster of patches associated to the respective Gaussian models, whereas, the SIDI algorithm estimates the parameters of the GMM prior by using EM algorithm.

The GMM based PLE algorithm has also been extended to different inverse problems other than the natural image restoration problems studied in [11]. In [35], the PLE algorithm was extended to address the problem of high dynamic range imaging from a single shot acquired with spatially varying exposures. The work reported in [36] exploits MAP-EM based PLE algorithm for solving matrix completion problems in different applications such as prediction of movie rating, personalized item recommendation system etc. Signal recovery algorithms closely related to the PLE algorithm have also been proposed for statistical compressive sensing [37, 38, 39].

2.3 The EPLL Algorithm

The EPLL algorithm [18] characterizes natural image patches by using an external GMM prior specified by the parameters $\boldsymbol{\Theta} = \{\omega_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{K}$. The parameters of the GMM prior containing a total of $K = 200$ Gaussian models are learnt from a large database of

10^6 clean natural image patches. In contrast with the PLE algorithm which assigns equal weights for different Gaussian models from the GMM prior, the EPLL algorithm exploits a traditional GMM prior which includes a mixing weight ω_k for each Gaussian model. The EPLL algorithm computes the estimate of an unknown natural image $\mathbf{x}$, from the degraded and/or noisy observation $\mathbf{y}$ assumed to be generated as,

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n}, \tag{2.13}$$

where, $\mathbf{A}$ denotes the degradation operator, and, $\mathbf{n}$ denotes AWGN with variance σ^2. The restored estimate $\hat{\mathbf{x}}$ of the unknown image $\mathbf{x}$ is computed by solving the MAP estimation problem,

$$\hat{\mathbf{x}} = \arg\min_{\tilde{\mathbf{g}}} \frac{\lambda}{2} \|\mathbf{A}\tilde{\mathbf{g}} - \mathbf{y}\|_2^2 - \sum_{i=1}^{L} \log f\left(\mathbf{R}_i\tilde{\mathbf{g}}\right), \tag{2.14}$$

where, $f(.)$ denotes the patch prior characterizing natural image patches, and $\mathbf{R}_i$ denotes the operator which extracts the i-th patch from the image $\tilde{\mathbf{g}}$. The first term in (2.14) corresponds to the likelihood function which constrains the restored estimate $\hat{\mathbf{x}}$ to be consistent with the observation $\mathbf{y}$, the second term corresponds to the Expected Patch Log Likelihood(EPLL) under the prior $f(.)$, and λ decides the relative importance of the patch prior when compared with the likelihood function. Though EPLL algorithm achieves impressive results in different image restoration problems by choosing the patch prior $f(.)$ as GMM, note that the EPLL algorithm is a generalized patch based image restoration framework which can be used in conjunction with any patch prior described by the pdf $f(.)$.

As the direct solution of the minimization in (2.14) is computationally intractable, the EPLL algorithm introduces a set of auxiliary variables $\{\mathbf{z}_i\}$, and computes the required estimate $\hat{\mathbf{x}}$ by exploiting the Half Quadratic Splitting (HQS) method as,

$$\hat{\mathbf{x}} = \arg\min_{\tilde{\mathbf{g}},\{\mathbf{z}_i\}} \frac{\lambda}{2} \|\mathbf{A}\tilde{\mathbf{g}} - \mathbf{y}\|_2^2 + \left(\sum_{i=1}^{L} \frac{\beta}{2} \|\mathbf{R}_i\tilde{\mathbf{g}} - \mathbf{z}_i\|_2^2 - \log f\left(\mathbf{z}_i\right) \right). \tag{2.15}$$

As the solutions of (2.14) and (2.15) are identical when $\beta \to \infty$, the EPLL algorithm iteratively minimizes (2.15) for a set of increasing values of β. In a single iteration corresponding to a fixed value of β, the objective function given in (2.15) is alternatively

minimized with respect to $\{\mathbf{z}_i\}$ and $\tilde{\mathbf{g}}$. The EPLL algorithm thus consists of two nested levels of iterations - the inner loop of iterations corresponds to the alternating minimization with respect to $\{\mathbf{z}_i\}$ and $\tilde{\mathbf{g}}$, whereas, the outer loop iterates over a set of increasing values of β. We use the name Patch Estimation step to denote the inner loop minimization of (2.15) with respect to $\{\mathbf{z}_i\}$, as the auxiliary variables $\{\mathbf{z}_i\}$ are identical to the restored patches when $\beta \to \infty$. As such, the minimization with respect to $\tilde{\mathbf{g}}$ is denoted as the Image Estimation.

2.3.1 Patch Estimation Step

The Patch Estimation step assumes that the unknown image $\tilde{\mathbf{g}}$ in (2.15) is fixed to the estimate $\hat{\mathbf{g}}$ obtained from the previous iteration (or initialization $\hat{\mathbf{g}} = \mathbf{y}$), and computes the estimates $\{\hat{\mathbf{z}}_i\}$ as,

$$\hat{\mathbf{z}}_i = \arg\min_{\mathbf{z}_i} \frac{\beta}{2} \|\mathbf{R}_i\tilde{\mathbf{g}} - \mathbf{z}_i\|_2^2 - \log f(\mathbf{z}_i) \tag{2.16}$$

for all the patches $i = 1, 2, ..., L$. Note that, (2.16) can be interpreted as the MAP denoising problem which targets to compute the denoised estimate $\hat{\mathbf{z}}_i$ of an unknown patch distributed according to GMM prior $f(.)$, from the noisy observation $\mathbf{R}_i\tilde{\mathbf{g}}$ contaminated by AWGN distributed as $\mathcal{N}\left(\mathbf{0}, \frac{1}{\beta}\mathbf{I}\right)$. As the solution of (2.16) cannot be obtained in closed form when $f(.)$ is chosen as a GMM prior, the EPLL algorithm approximately computes the patch estimate $\hat{\mathbf{z}}_i$ as follows. For each patch $\mathbf{z}_i$, the index k_i of the Gaussian model responsible for the patch is estimated by maximizing the posterior responsibilities as,

$$\begin{aligned} \hat{k}_i &= \arg\max_k \log \gamma_{ik} \\ &= \arg\max_k \log \Phi\left(\mathbf{R}_i\tilde{\mathbf{g}}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k + \frac{1}{\beta}\mathbf{I}\right) \end{aligned} \tag{2.17}$$

where, γik is the posterior responsibility that the kth Gaussian model is responsible for the i-th patch. The patch estimate $\hat{\mathbf{z}}_i$ is then computed by using the Wiener filter derived from the covariance matrix of the selected Gaussian model with index $\hat{k}_i$ as,

$$\hat{\mathbf{z}}_i = \boldsymbol{\mu}_{\hat{k}_i} + \boldsymbol{\Sigma}_{\hat{k}_i}\left(\boldsymbol{\Sigma}_{\hat{k}_i} + \frac{1}{\beta}\mathbf{I}\right)^{-1}\left(\mathbf{R}_i\tilde{\mathbf{g}} - \boldsymbol{\mu}_{\hat{k}_i}\right) \tag{2.18}$$

The patch estimation step computes the estimate $\hat{\mathbf{z}}_i$ for all the patches $i = 1, 2, ..., L$, by using (2.18).

2.3.2 Image Estimation Step

The image estimation step assumes that auxiliary variables $\{\mathbf{z}_i\}$ in (2.15) are fixed to the estimates $\hat{\mathbf{z}}_i$ obtained from the Patch Estimation step, and computes the estimate $\hat{\mathbf{g}}$ of the unknown image $\tilde{\mathbf{g}}$ as,

$$\hat{\mathbf{g}} = \arg\min_{\tilde{\mathbf{g}}} \frac{\lambda}{2} \|\mathbf{A}\tilde{\mathbf{g}} - \mathbf{y}\|_2^2 + \frac{\beta}{2} \sum_{i=1}^{L} \|\mathbf{z}_i - \mathbf{R}_i\tilde{\mathbf{g}}\|_2^2 . \tag{2.19}$$

The solution of (2.19) can be obtained in closed form as,

$$\hat{\mathbf{g}} = \left(\lambda \mathbf{A}^T\mathbf{A} + \beta \sum_{i=1}^{L} \mathbf{R}_i^T\mathbf{R}_i\right)^{-1} \left(\lambda \mathbf{A}^T\mathbf{y} + \beta \sum_{i=1}^{L} \mathbf{R}_i^T\mathbf{z}_i\right), \tag{2.20}$$

which corresponds to linearly combining the image obtained by averaging the overlapping patches $\{\hat{\mathbf{z}}_i\}_{i=1}^{L}$ and a weighted version of the degraded input image $\mathbf{y}$.

The Patch Estimation and Image Estimation steps are iterated several times for a fixed value of β, and the whole process is repeated for a set of increasing values of β. The estimate $\hat{\mathbf{g}}$ of the unknown image computed in (2.20) for a sufficiently large value of β (ideally, $\beta \to \infty$) is chosen as the required estimate $\hat{\mathbf{x}}$ of the unknown original image $\mathbf{x}$. The weight parameter λ which decides the relative importance of the patch prior when compared with the likelihood term is empirically chosen as $\lambda = \frac{C}{\sigma^2}$, where, σ^2 is the noise variance. It can be noted from the Patch Estimation step given in (2.16) that the value $\frac{1}{\beta}$ can be interpreted as variance of the residual noise contained in the estimated image $\hat{\mathbf{g}}$ obtained from the previous iteration. The EPLL algorithm chooses the estimate of residual noise variance computed by using [40] as the value of $\frac{1}{\beta}$ to be used in every iteration of the outer loop. The EPLL algorithm exploits a fixed external GMM prior which is not adapted to the specific image being restored, whereas, the internal GMM prior used by the PLE algorithm is adapted to the input image being restored. However, the rich external

GMM prior containing 200 Gaussian models enables EPLL algorithm to achieve superior restoration performance when compared with the PLE algorithm.

2.3.3 Related Algorithms

The image denoising algorithm proposed in [41] partitions the entire set of noisy patches from a given input image into as many clusters as the number of Gaussian models of an external GMM prior learnt from a large database of clean natural image patches. As each of the noisy patches is associated to the best Gaussian model responsible for the patch, the cluster of patches corresponding to a given Gaussian model contains all the noisy patches assigned to the given model. Assuming that the patch matrix whose columns are the patches from a given cluster is low rank, each noisy patch from the cluster is denoised by applying nuclear norm minimization.

The adaptive image denoising algorithm proposed in [42, 43] takes a generic external GMM prior, and adapts the parameters of the GMM prior to the specific image being denoised. The GMM parameters are adapted by using the patches extracted from a pre-filtered pilot image obtained by applying any of the existing denoising algorithms on the given input noisy image. In contrast with this adaptation procedure, the Adaptive EPLL (A-EPLL) algorithm proposed in [44] adapts the parameters of a generic external GMM prior by using the uncertainty based EM algorithm [45].

The Image Restoration algorithm based on Variable Splitting (IRVS) method proposed in [46] targets to compute the restored estimate $\hat{\mathbf{x}}$ of an unknown clean image $\mathbf{x}$ by minimizing the cost function,

$$\hat{\mathbf{x}} = \arg\min_{\tilde{\mathbf{x}}} \frac{1}{2}\|\mathbf{A}\tilde{\mathbf{x}} - \mathbf{y}\|_2^2 + \alpha\Psi\left(\tilde{\mathbf{x}}\right), \tag{2.21}$$

where, $\Psi\left(.\right)$ is a convex regularizer characterizing natural images. The IRVS algorithm minimizes the cost function in (2.21) by using Alternating Direction Method of Multipliers (ADMM). The ADMM approach iteratively alternates between two steps corresponding to the two decoupled minimization problems obtained by splitting the cost function in (2.21). The first step corresponds to a quadratic minimization whose solution can be obtained in closed form, whereas, the second step corresponds to solving a global image denoising

problem which constrains the solution space by using the regularization function $\Psi(.)$. The IRVS algorithm approximates the global image denoising step by using a patch based image denoising algorithm exploiting GMM prior. As in the case of EPLL algorithm, the parameters of the GMM prior are learnt from a large external database of clean natural image patches. In the case of image restoration problems such as de-blurring of text and face images, the IRVS algorithm demonstrates that the GMM prior learnt from targeted external databases containing a specific class of images such as face images, text images etc. can significantly improve the restoration performance when compared with the performance achieved by using a GMM prior learnt from a generic database.

The Image Restoration algorithm based on Locally Selected Class Adapted Models (IRLSCAM) proposed in [47] is an extension of IRVS algorithm into the case of images containing different regions belonging to different classes of images. As an example of such a case, consider the restoration of an image containing both text and facial images. In general, an image may contain different regions belonging to C different classes of images. The IRLSCAM algorithm restores a degraded image containing different regions belonging to C different classes of images, by exploiting C different GMM priors learnt from targeted databases containing images from the respective classes. A given degraded patch from the input image is classified into one of the C classes by maximizing the a posteriori probability under C different GMM priors, and the restored patch estimate is computed by plugging the GMM prior corresponding to the selected class.

The natural image denoising algorithm proposed in [48] exploits an external GMM prior for characterizing patch groups containing several non local self similar patches from a natural image. The parameters of the GMM prior are learnt from several self similar patch groups extracted from a large database of clean natural images. Given a noisy input patch, the group of non local noisy patches similar to the given patch are extracted from the noisy image, and the best Gaussian model describing the patch group is selected from the mixture. The denoised estimates corresponding to the group of nonlocal noisy patches are computed by applying a weighted sparse coding over the eigenvectors of the covariance matrix.

In [49], the external GMM prior characterizing non local self similar patch groups is used as

a guide to learn several internal orthogonal dictionaries adapted to the noisy patches from the given input noisy image. The patch groups containing non local self similar patches corresponding to the noisy patches are extracted, and each patch group is assigned to the best Gaussian model. Corresponding to each of the Gaussian models from the GMM prior, an internal orthogonal dictionary is learnt by using all the patch groups assigned to the Gaussian model. A given noisy patch is denoised by applying a weighted sparse coding over the internal orthogonal dictionary corresponding to the selected Gaussian model.

The Image Specific Prior Adaptation (ISPA) method proposed in [50] address natural image denoising problem by characterizing image patches using an adapted GMM prior obtained by combining a generic external GMM prior and the image specific internal GMM prior learnt from the input noisy image. The ISPA algorithm is motivated from the observation that, a generic external GMM prior fails to characterize some of the image specific outlier patches from a noisy input image, but an internal GMM prior learnt directly from the input image provides better characterization for these outlier patches. Given a noisy input image, the ISPA algorithm learns an internal GMM prior with as many Gaussian models as the number of Gaussian models in the generic external GMM prior, and for each Gaussian model from the internal GMM prior, the closest Gaussian model with minimum Kullback-Leibler divergence (KL divergence) is identified from the generic GMM prior. If the KL divergence between a given Gaussian model from the internal GMM prior and the closest Gaussian model from the generic GMM prior is greater than a certain threshold, the given Gaussian model is added to the generic prior. The GMM prior obtained by merging external and internal GMM priors is further adapted to the input image by using the GMM adaptation method proposed in [51]. A given noisy patch is denoised by first selecting the best Gaussian model (from the adapted GMM prior) responsible for the patch, and then computing the denoised estimate by using the wiener filter derived from the covariance matrix of the selected Gaussian model.

The multi scale EPLL algorithm proposed in [52] exploits a GMM prior for characterizing natural image patches of different scales, i.e., different patch sizes. The multi-scale EPLL algorithm aims to characterize natural image patches of larger scales, i.e., patches with dimensions chosen as integer multiples of dimensions of base scale patches with size $\tau \times \tau$

pixels. The larger scale patches of dimensions $2\tau \times 2\tau, 3\tau \times 3\tau...$ are filtered and downsampled to the base scale patch dimensions $\tau \times \tau$, and characterized by the same GMM prior characterizing the base scale patches. The filters are chosen in such a way that the filtered and downsampled versions of larger scale patches are consistent with the same GMM prior characterizing base scale patches. In addition to the expected log likelihood of base scale patches, the multi scale EPLL algorithm also incorporates the expected log likelihood of larger scale patches into the objective function used by the original EPLL algorithm. As in the case of EPLL algorithm, the modified objective function used by the multi-scale EPLL algorithm is iteratively minimized by using half quadratic splitting method.

The EPLL algorithm was originally proposed as a general image restoration framework which can exploit any patch prior characterizing natural image patches. The work proposed in [53] addresses natural image restoration problems such as image denoising and inpainting by incorporating a sparsity prior in the EPLL image restoration framework.

2.4 The NL-Bayes Algorithm

The Nonlocal Bayesian (NL-Bayes) image denoising algorithm [20] and some of the related algorithms [54, 55] exploit a different approach in characterizing natural image patches using Gaussian mixture models. The NL-Bayes algorithm characterizes each natural image patch from an image by using a multivariate Gaussian distribution. The parameters of the Gaussian models characterizing each of the patches can be different, i.e., corresponding to each patch, there exists a Gaussian model from which the patch is drawn. Though this characterization does not explicitly invoke GMM priors, it is equivalent to the characterization by using a GMM prior containing as many Gaussian models as the number of patches in the image. However, there are two important differences between the GMM priors used by NL-Bayes and PLE/EPLL algorithms. The PLE and EPLL algorithms assumes that the Gaussian model responsible for a given i-th patch is not known a priori, but chosen randomly from the mixture, whereas, NL-Bayes algorithm assumes that the Gaussian model responsible for the i-th patch is known a priori, given the mixture of Gaussian models. The GMM prior used by PLE and EPLL algorithms consists of a fixed number of Gaussian models regardless of the size of the image, whereas, the number of

Gaussian models in GMM prior used by NL-Bayes algorithm depends on the number of patches in the image. Interestingly, one of the initial attempts in exploiting GMM prior for characterizing natural image patches - the APCA based image denoising algorithm described in section 2.1 - utilizes a similar method for characterizing image patches.

The NL-Bayes algorithm estimates the parameters of the Gaussian models by using the noisy patches from the given input image to be denoised. The parameters of the Gaussian model characterizing the i-th patch $\mathbf{p}_i$ are estimated from the corresponding noisy patch $\mathbf{q}_i$, and a group of non local (noisy) patches similar to $\mathbf{q}_i$. The similarity between noisy patches is measured by using euclidean norm. As the noisy patches are corrupted by zero mean AWGN with noise variance σ^2, the mean vector $\tilde{\boldsymbol{\mu}}_i$of the Gaussian model characterizing the i-th patch is estimated as the sample mean of the set of nonlocal noisy patches similar to $\mathbf{q}_i$, and the covariance matrix $\tilde{\boldsymbol{\Sigma}}_i$ is estimated as $\tilde{\boldsymbol{\Sigma}}_i = \mathbf{S} - \sigma^2\mathbf{I}$, where, $\mathbf{S}$ is the sample covariance matrix. Given $\tilde{\boldsymbol{\mu}}_i$ and $\tilde{\boldsymbol{\Sigma}}_i$, the initial coarse estimate $\hat{\tilde{\mathbf{p}}}_i$ of the unknown patch $\mathbf{p}_i$ is computed by solving the Bayesean MAP estimation problem,

$$\begin{aligned} \hat{\tilde{\mathbf{p}}}_i &= \arg\max_{\tilde{\mathbf{p}}_i} p\left(\tilde{\mathbf{p}}_i \,|\mathbf{q}_i\right) \\ &= p\left(\mathbf{q}_i \,|\tilde{\mathbf{p}}_i\right) p\left(\tilde{\mathbf{q}}_i\right), \end{aligned} \tag{2.22}$$

whose solution corresponds to the conditional expectation,

$$\begin{aligned} \hat{\tilde{\mathbf{p}}}_i &= \mathbb{E}\left[\tilde{\mathbf{p}}_i \,|\mathbf{q}_i\right] \\ &= \tilde{\boldsymbol{\mu}}_i + \tilde{\boldsymbol{\Sigma}}_i \left(\tilde{\boldsymbol{\Sigma}}_i + \sigma^2\mathbf{I}\right)^{-1} \left(\mathbf{q}_i - \tilde{\boldsymbol{\mu}}_i\right), \end{aligned} \tag{2.23}$$

for $i = 1, 2, ..., L$. Note that the initial coarse estimates $\left\{\hat{\tilde{\mathbf{p}}}_i\right\}$ of the unknown patches are computed by using the Gaussian model parameters estimated directly from the noisy patches. These patch estimates can be seen as good proxy for the unknown clean patches, and a more robust estimate of the Gaussian model parameters can be obtained by computing these parameters from the initial patch estimates $\left\{\hat{\tilde{\mathbf{p}}}_i\right\}$. The NL-Bayes algorithm updates the parameters of the Gaussian model characterizing the i-th patch, $i = 1, 2, ..., L$, by repeating the process of non local patch grouping on the initial patch estimates $\left\{\hat{\tilde{\mathbf{p}}}_i\right\}$, and estimating the mean vector and covariance matrix of the Gaussian model as the sample mean and covariance matrix of the set of non local self similar patches. The denoised

patch estimate $\hat{\mathbf{p}}_i$ is then computed for each patch $i = 1, 2, ..., L$, by substituting the updated Gaussian model parameters in (2.23).

The image restoration algorithm based on Gaussian mixture models with spatially constrained patch clustering proposed in [54] shows that Gaussian models learnt from the non local patches restricted to an $N \times N$ sized window around each exemplar patch better fit the observed patches when compared with the Gaussian models learnt from unrestricted non local patches spread over the entire image. In principle, the spatially constrained GMM based algorithm is closely similar to the NL-Bayes algorithm except for the difference in the patch averaging step. The NL-Bayes algorithm aggregates overlapping denoised patches through a plain averaging, whereas, the spatially constrained GMM based algorithm aggregates overlapping patches by using a weighted averaging method. The weights corresponding to the overlapping patches contributing to the i-th patch depend on how well these overlapping patches fit to the Gaussian model responsible for the i-th patch.

The joint image denoising and interpolation framework based on hyper prior Bayesian Estimator (HBE) proposed in [55] exploits a multivariate Gaussian distribution for characterizing each natural image patch from a given image. The unknown parameters $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$ of the Gaussian model characterizing the i-th patch are assumed to be distributed according to the Normal-Wishart hyper prior for all the patches $i = 1, 2, ..., L$. Given a noisy image, possibly containing several missing pixels, the HBE algorithm computes the restored estimate by iteratively alternating several times between the patch estimation step and hyper prior parameters update step. The patch estimation step assumes that the hyper prior parameters obtained from the previous iteration (or initialization) are fixed, and jointly estimates the parameters of the Gaussian model characterizing the i-th patch, and the restored estimates of a group of nonlocal patches similar to (including) the i-th patch for $i = 1, 2, ..., L$. The hyper prior parameter update step updates the parameters of the hyper prior by using the image estimate obtained by averaging the overlapping restored patches.

2.5 Sparsity based Algorithms Exploiting Highly Correlated Natural Image Patches

The close relations between GMM and sparsity based algorithms has motivated previous works to propose GMM analogues [11, 18] of the sparsity based algorithms for different image restoration problems. In this section, we review some of the sparsity based algorithms exploiting highly correlated natural image patches in image restoration problems such as single image super resolution and color image denoising. The image restoration algorithms proposed in this thesis can be interpreted as GMM analogues of the sparsity based algorithms reviewed in this section.

2.5.1 Sparse coding based Super Resolution (ScSR)

The ScSR algorithm [4] for single image super resolution exploits a sparsity prior for jointly characterizing HR and the corresponding LR patches. The ScSR algorithm assumes that the patch vectors obtained by concatenating HR and the corresponding LR patches can be approximated as linear combinations of a few atoms from a dictionary. Let $\mathbf{q}$ denote the LR patch, $\mathbf{p}$ denote the corresponding HR patch, and $\mathbf{v}$ denote the concatenated patch vector constructed as,

$$\mathbf{v} = \begin{bmatrix} \mathbf{p} \\ \mathbf{q} \end{bmatrix}. \tag{2.24}$$

The ScSR algorithm assumes that the concatenated patch vector $\mathbf{v}$ can be approximated as,

$$\mathbf{v} \approx \mathbf{D}\boldsymbol{\alpha} \text{ s.t } \|\boldsymbol{\alpha}\|_0 \leq T \tag{2.25}$$

The overcomplete dictionary $\mathbf{D}$ jointly characterizing HR and LR patches can be interpreted as,

$$\mathbf{D} = \begin{bmatrix} \mathbf{D}_H \\ \mathbf{D}_L \end{bmatrix}, \tag{2.26}$$

where the subdictionary $\mathbf{D}_H$ characterizes the HR patches, and $\mathbf{D}_L$ characterizes the LR patches. The ScSR algorithm assumes that the sparse representation of an HR patch $\mathbf{p}$ over the dictionary $\mathbf{D}_H$, and the corresponding LR patch $\mathbf{q}$ over the dictionary $\mathbf{D}_L$, are

same and equal to $\boldsymbol{\alpha}$. Specifically, the sparse approximation given in (2.25) can be written as,

$$\mathbf{p} \approx \mathbf{D}_H \boldsymbol{\alpha},\ \mathbf{q} \approx \mathbf{D}_L \boldsymbol{\alpha},\ \text{s.t}\ \|\alpha\|_0 \leq T \tag{2.27}$$

The assumption of invariance of sparse representations of HR and LR patches over their respective dictionaries implies that the estimate $\hat{\mathbf{p}}_i$ of an unknown HR patch $\mathbf{p}_i$ can be computed from the sparse representation $\boldsymbol{\alpha}_i$ of the corresponding LR patch $\mathbf{q}_i$ as,

$$\hat{\mathbf{p}}_i = \mathbf{D}_H \boldsymbol{\alpha}_i \tag{2.28}$$

Thus, the ScSR algorithm estimates the sparse representation vector $\boldsymbol{\alpha}_i$ of a given input LR patch $\mathbf{q}_i$ by solving the sparse coding problem,

$$\hat{\boldsymbol{\alpha}}_i = \arg\min_{\tilde{\boldsymbol{\alpha}}_i} \|\mathbf{D}_L \tilde{\boldsymbol{\alpha}}_i - \mathbf{q}_i\|_2^2 + \lambda \|\tilde{\boldsymbol{\alpha}}_i\|_1 , \tag{2.29}$$

and computes the estimate $\hat{\mathbf{p}}_i$ of the unknown patch $\mathbf{p}_i$ as,

$$\hat{\mathbf{p}}_i = \mathbf{D}_H \hat{\boldsymbol{\alpha}}_i, \tag{2.30}$$

for all the patches from the given input LR image. The estimated HR patches are tiled at their respective locations in the HR image, and all the overlapping patches are averaged to compute the HR image. The dictionary $\mathbf{D} = \left[\mathbf{D}_H^T, \mathbf{D}_L^T\right]^T$ jointly characterizing HR and LR patches is learnt from a large database of concatenated HR-LR patches.

In chapter 3 of this thesis, we propose a single image SR algorithm based on joint GMM method which can be seen as the GMM analogue of the ScSR algorithm. The review of a series of works [5, 6, 56, 7, 24] proposed for addressing some of the drawbacks of the original ScSR algorithm is included in chapter 3.

2.5.2 The A+ Algorithm for Single Image Super Resolution

The Adjusted Anchored Neighborhood Regression (A+) algorithm proposed in [24] exploits a regression based approach for addressing single image super resolution problem. The A+ algorithms assumes that the mapping from an LR patch to the corresponding HR

patch is locally linear, i.e., the HR patches corresponding to all the LR patches within a small neighborhood of manifold of LR patches can be obtained by applying a linear transformation on the LR patches from the neighborhood. Based on this assumption, the A+ algorithm partitions the manifold of LR patches into several clusters, and for each cluster, the linear transformation that maps the LR patches from the cluster to the corresponding HR patches is learnt from a training database containing HR-LR patch pairs. The A+ algorithm partitions the set of LR patches from a large database containing LR and the corresponding HR patches into several clusters by utilizing an over-complete dictionary characterizing LR patches. The entire set of LR training patches is partitioned into as many clusters as the number of atoms in the dictionary. A cluster of LR patches is associated to each of the atoms from the dictionary. A given LR patch from the training databases is assigned to the cluster corresponding to the dictionary atom to which the given patch is mostly correlated under normalized inner product metric.

Assuming that all the LR patches from a given cluster are mapped to the corresponding HR patches through a common linear transformation, A+ algorithms learns the transformation matrix for all the clusters. The transformation matrix corresponding to a given cluster is learnt by applying ridge regression on the set of LR training patches from the given cluster and the corresponding HR training patches. Given an input LR patch to be super resolved, the A+ algorithm assigns the given patch to the cluster associated with dictionary atom mostly correlated to the given patch. The HR estimate corresponding to the given LR patch is computed by applying the linear transformation corresponding to the cluster to which the given patch is assigned. The HR patch estimates corresponding to each of the LR patches from a given input image are computed, and the overlapping HR patch estimates averaged to compute the estimated HR image. The single image super resolution algorithm based on GMM based regression method proposed in chapter 4 of this thesis can be interpreted as the GMM analogue of the A+ algorithm.

2.5.3 The Color-KSVD Algorithm for Color Image Restoration

The widely studied KSVD image denoising algorithm [10] was later extended to a color image restoration framework [25] addressing restoration problems such as color image denoising, interpolation and demosaicking. In this thesis, we use the name Color-KSVD to denote this color image restoration framework. The Color-KSVD algorithm exploits sparsity prior for jointly characterizing the patches from Red, Green and Blue channels of a color image. Let $\mathbf{p}_{iR}$, $\mathbf{p}_{iG}$ and $\mathbf{p}_{iB}$ be the corresponding patches from Red, Green and Blue channels, i.e., coordinates of center pixels of these patches are same in different color channels. The Color-KSVD algorithm exploits sparsity prior for characterizing concatenated color patches $\mathbf{v}_i \in \mathbb{R}^{3\tau^2}$ constructed as,

$$\mathbf{v}_i = \begin{bmatrix} \mathbf{p}_{iR} \\ \mathbf{p}_{iG} \\ \mathbf{p}_{iB} \end{bmatrix}, \tag{2.31}$$

where $\mathbf{p}_{ic} \in \mathbb{R}^{\tau^2}$ for $c = R, G, B$, and τ denotes the patch dimension.

The Color-KSVD algorithm assumes that the concatenated color patches $\{\mathbf{v}_i\}$ can be approximated as,

$$\mathbf{v}_i \approx \mathbf{D}\boldsymbol{\alpha}_i \text{ s.t } \|\boldsymbol{\alpha}_i\|_0 \leq T, \tag{2.32}$$

where, $\mathbf{D} \in \mathbb{R}^{3\tau^2 \times M}$ denotes the dictionary containing M atoms characterizing concatenated color patches, and, $\boldsymbol{\alpha}_i \in \mathbb{R}^M$ denotes the sparse representation of $\mathbf{v}_i$ over the dictionary $\mathbf{D}$. The dictionary atoms $\{\mathbf{d}_j\}_{j=1}^{M}$, where, $\mathbf{d}_j$ is the j-th column of the dictionary matrix $\mathbf{D}$, capture the strong inter channel correlations observed in color image patches.

Given a degraded and/or noisy color image, the Color-KSVD algorithm computes the restored estimate by iteratively alternating several times between sparse coding and dictionary update steps. The sparse coding step assumes that the dictionary obtained from the previous iteration (or initialization) is fixed, and computes the sparse representation $\hat{\boldsymbol{\alpha}}_i$ of the concatenated color patch $\mathbf{v}_i$, for all the patches $i = 1, 2, ..., L$. The sparse coding step exploits Orthogonal Matching Pursuit (OMP) algorithm for computing the sparse

representations. The dictionary update step assumes that the sparse representations obtained from the sparse coding step are fixed, and updates the dictionary atoms by utilizing KSVD dictionary learning algorithm. The initial dictionary is learnt from a large database of clean concatenated color patches. The sparse coding and dictionary update steps are alternatively iterated several times, and the sparse representations $\{\hat{\boldsymbol{\alpha}}_i\}$ obtained in the final iteration are used to compute the restored estimates $\{\hat{\mathbf{v}}_i\}$ of the concatenated color patches as,

$$\hat{\mathbf{v}}_i = \mathbf{D}\hat{\boldsymbol{\alpha}}_i,\ i = 1, 2, ..., L \tag{2.33}$$

The estimates $\hat{\mathbf{p}}_{iR}$, $\hat{\mathbf{p}}_{iG}$ and $\hat{\mathbf{p}}_{iB}$ of the unknown patches from Red, Green and Blue channel are extracted from $\hat{\mathbf{v}}_i$, and the overlapping patches from respective color channels are averaged to compute the restored estimates of the color channels.

The dictionary atoms $\{\mathbf{d}_j\}$ characterizing concatenated color patches can be seen as prototype color patches capturing the basic features of color image patches. In the Color– KSVD algorithm, the authors have observed that the dictionary atoms produced by KSVD dictionary learning algorithm are more biased to capture monochrome image features such as edges of different orientations, corners etc., leaving the color features with less importance. This bias is expected to be produced by the abundance of structures such as edges and corners observed in monochrome as well as color image patches. The bias on the dictionary atoms produces a color wash out effect [25] in the restored images. In the Color-KSVD algorithm, the color wash out effect is reduced by modifying the OMP algorithm used in the sparse coding step. The OMP algorithm selects a sparse subset of dictionary atoms to be used in the linear approximation of a concatenated color patch. The Color-KSVD algorithm modifies the selection of dictionary atoms in such a way that the selected atoms preserve the average color information in the patch.

The Color-KSVD algorithm, and also the original KSVD image denoising algorithm, were later improved in [13] by exploiting multiscale sparse representations. The Quaternion–KSVD (Q-KSVD) algorithm [57] proposes a color image restoration framework by representing the three values of a color image pixel using a quaternion number. The Q-SVD algorithm is closely similar to the original KSVSD image denoising algorithm except that the Q-KSVD algorithm operates on quaternion vectors representing color image patches.

The color image restoration method based on Joint Color Space GMMs (JCS-GMMs) proposed in chapter 5 of this thesis can be seen as the GMM analogue of the Color-KSVD algorithm [25].

Chapter 3

Single Image Super Resolution Using Joint GMM Method

In this chapter, we propose a Single Image Super Resolution (SISR) algorithm using a Joint GMM prior. The proposed Joint GMM method exploits a GMM prior for jointly characterizing HR and the corresponding LR patch vectors. The parameters of the Joint GMM prior are learnt from concatenated HR-LR patches sampled from a large database of pairs of HR and the corresponding LR images. The covariance matrices of Gaussian models from the Joint GMM prior capture the inherent correlations between high and low resolution patches, and are utilized for estimating the unknown HR patches from the given input LR patches. We study the performance of the proposed method by comparing with various competing algorithms for SISR. Our experiments on various natural images demonstrate the competitive performance obtained by the proposed method at low computational cost. The proposed Joint GMM method can be interpreted as a GMM analogue of the ScSR algorithm [4] exploiting sparse representation of concatenated HR-LR patches over joint dictionaries.

Super Resolution (SR) is a process which artificially produces a high resolution (HR) image from one or more low resolution (LR) images. Image SR is utilized in various applications involving low cost, low resolution camera sensors, to improve the resolution for better image analysis or to make the images visually appealing for the end user. Examples of

such applications include medical imaging, satellite imaging, cell phone cameras, digital photography etc. Image processing and computer vision literature contains a wide variety of SR algorithms which can be broadly classified into multi image SR and single image SR. Multi image SR [58, 59, 60, 61, 62, 63, 64, 65] targets to estimate a HR image from several LR snapshots with sub-pixel shifts. Various studies [61, 66] on the fundamental limits of SR have revealed that the underlying problem in multi image SR is highly ill-conditioned due to insufficient number of observations and unknown registration parameters [7]. These studies have also observed that multi image SR method tends to overly smooth the estimated HR image when the desired magnification factor increases, and is not suggested for a magnification factor greater than 1.6.

In this work, we focus on Single Image SR (SISR) that addresses the problem of computing the estimate of an unknown HR image from a single degraded LR observation. The Observed LR image is assumed to be generated by first convolving the original HR image with a blur kernel, and subsequently decimating uniformly in the horizontal and vertical directions by some integer factor q. Using vector representation obtained through lexicographic ordering of image pixels, the observed LR image can be written as,

$$\mathbf{y} = \mathbf{SHx} + \mathbf{n} \tag{3.1}$$

where, $\mathbf{x} \in \mathbb{R}^{N_x}$ and $\mathbf{y} \in \mathbb{R}^{N_y}$ respectively denote the vector representations of unknown HR image and the observed LR image, N_x and N_y denote the number of pixels in the unknown HR and the observed LR images respectively, $\mathbf{H} \in \mathbb{R}^{N_x \times N_x}$ denotes the blur kernel, $\mathbf{S} \in \mathbb{R}^{N_y \times N_x}$ denotes the subsampling operator, and, $\mathbf{n} \in \mathbb{R}^{N_y}$ denotes the additive white Gaussian noise added to the LR image.

Interpolation based methods and learning based methods are two broad categories of SISR algorithms previously proposed in the literature. Interpolation based methods [67, 68, 69, 70, 71] estimate the unknown HR image by utilizing various linear and non linear interpolators. Adaptive interpolators [72, 73, 74, 75, 76, 77], and the interpolation algorithms exploiting more sophisticated image models [78, 79, 80], attempt

to preserve the high frequency details in the estimated HR image by adapting the interpolation process to various local structures such as edges and gradients of an image. Recently, state of the art results in SISR were achieved by learning based methods [81, 82, 83, 84, 85, 86, 87, 88, 4, 6, 56] that utilize various statistical learning techniques to learn different types of HR-LR co-occurrence priors. The example based learning methods estimate the HR image corresponding to a given input LR image by exploiting the HR-LR co-occurrence learnt from a training database containing several example HR and the corresponding LR images. Another recent approach in learning based SISR is self example based methods that exploit self recurring nature of image patches within and across different scales [89, 90, 91, 92]. In the present work, we focus on the former method, i.e., example based SISR algorithms that utilize an external training database containing HR-LR pairs of natural images.

In our work, the blur kernel $\mathbf{H}$ is assumed to be unknown and shift invariant. As we have only a single LR observation $\mathbf{y}$ of the unknown HR image $\mathbf{x}$, and the blur kernel $\mathbf{H}$ is assumed to be unknown, estimating $\mathbf{x}$ from $\mathbf{y}$ is a severely ill-posed inverse problem. But, we assume that we have a large database of example HR and LR images, in which, each LR image is produced from the corresponding HR image by applying the unknown degradation model (blur and decimation) that we are trying to invert. Given this database, our proposed SISR algorithm targets to learn a joint model for the co-occurrence of HR and LR patch pairs, and use this learnt model to compute the estimate of an unknown HR patch corresponding to a given input LR patch.

An important line of research in the field of SISR was initiated by the ScSR algorithm proposed in [4]. The ScSR algorithm exploits sparse representation of HR and LR patches over redundant dictionaries jointly characterizing HR-LR patch pairs. The ScSR algorithm was later modified in a series works, and resulted in a class of joint dictionary based SISR algorithms [4, 5, 6, 56, 93, 24], including the present state of the art patch based SISR algorithm [24]. However, the simple sparsity model utilized by sparse representation based SISR algorithms [4, 6, 56] makes the computation of sparse representations of natural image patches unstable due to the high degrees of freedom in choosing a sparse solution [94, 28]. The problems associated with such instabilities can be reduced up to a certain extent by imposing structured sparsity models such as block sparsity [21, 22, 23], group sparsity

[29, 30] etc. In spite of the impressive results achieved by sparsity based algorithms, most of these algorithms suffer from high computational load even for images of moderate size.

Recently, Gaussian Mixture Models (GMMs) have been successfully utilized as a prior characterizing natural image patches [94, 18, 19, 95, 96], and its connections with block sparsity models have been explored [94, 28]. In the image restoration framework proposed in [94], it is observed that the restored patch estimates computed by exploiting GMM prior containing K Gaussian models is equivalent to the patch estimates computed by sparsity based algorithms exploiting blocks sparsity structure with a block dictionary containing K blocks. Each block from the dictionary represents the PCA basis corresponding to a Gaussian model from the GMM prior. GMM based image restoration methods can thus be interpreted as sparsity based methods exploiting block sparsity structure. However, GMM based methods enjoy the advantage of having low computational load when compared with sparsity based methods. Moreover, it was demonstrated in [94], that corresponding to most of the sparsity based image restoration methods, there exist GMM analogues delivering superior performance with much faster speed of operation. The above observations lead us to the following questions:

1. Is there a natural GMM analogue of the joint dictionary based methods for single image SR ?
2. How does such a GMM based method compare to the ScSR algorithm and its improved versions ?
3. How does it perform when compared with other learning based methods for single image SR ?

In this work, we attempt to answer these questions by proposing a Joint GMM method for SISR. We demonstrate that our computationally efficient Joint GMM method delivers a performance comparable to the competing methods while having very good speed.

The rest of this chapter is organized as follows. Section 3.1 briefly reviews the literature on SISR with special emphasis on joint dictionary based approaches. In Section 3.2, we describe the proposed Joint GMM method for SISR. Section 3.3 discusses the details of

super resolution experiments conducted on a set of natural images, and the results of performance comparisons with competing methods. Section 3.4 concludes the chapter.

3.1 Review of Prior Art

Super resolution of a natural image from a single low resolution observation is a well studied topic in image processing literature. Earlier approaches on single image SR were based on various linear and non-linear interpolators [67, 68, 69, 70, 71] such as bicubic and cubic spline interpolators. Adaptive interpolators [72, 73, 74, 75], and interpolation algorithms that incorporate various natural image models [78, 79, 80], were proposed later to reduce the blurring effect inherent to the simple interpolation methods. Since interpolation methods perform poorly as the desired magnification factor increases, the idea of 'learning from data' was introduced from machine learning literature to further push the performance of SR algorithms. Learning based methods [81, 82, 83, 84, 85, 86, 87, 88, 4, 6, 56], also known as example based methods, apply various statistical learning methods on a large set of examples of either HR or HR-LR pairs of natural images to learn to infer the missing high frequency details from a given LR image. Learning based SR algorithms [83, 97, 98] that utilize a database of only HR images learn the characteristics of high frequency details missing in the LR images, whereas, the algorithms [81, 82, 84, 85, 86, 87, 88, 4, 6, 56] that utilize a collection of pairs of HR and the corresponding LR images learn the essential characteristics of LR to HR mapping. In the following, we focus only on learning based methods that utilize a database of HR-LR pairs as our proposed method belongs to this category.

SR algorithms proposed in [81, 82] using a Markov network model for characterizing natural image patches can be considered as the introductory work in the area of learning based methods for SISR. Baker and Kanade proposed an image hallucination algorithm [61] which recognizes the local features of a LR image whose resolution is then enhanced by using high frequency details of similar features selected from a training database. The work proposed in [87] constructs primal sketch priors such as edges, ridges and corners, and exploits these priors to enhance the quality of the estimated HR image. The Neighborhood Embedding (NE) algorithm proposed in [84], and several of its variants

[99, 100, 101, 102, 103, 24], utilize the idea of Locally Linear Embedding (LLE) adapted from the area of manifold learning. These algorithms assume that LR and HR image patches form manifolds with similar local geometry in their respective patch spaces. In [104], Kernel Ridge Regression (KRR) is utilized to learn the mapping from LR to HR image by using a database of HR-LR pairs of natural images. In [83], the parametric gradient profile model learnt from a large collection of natural images is used to constrain the gradient fields of the estimated HR image.

A different route that produced several closely related algorithms giving state of the art results in example based SISR was introduced in the ScSR algorithm [4] using sparse representation of HR and LR image patches over joint dictionaries characterizing HR-LR patch pairs. The ScSR algorithm jointly learns a pair of dictionaries $\{\mathbf{D}_H, \mathbf{D}_L\}$ from a large collection of HR-LR patch pairs in such a way that the learnt dictionaries provide same sparse representations for a given pair of HR and the corresponding LR patches. The over-complete dictionary $\mathbf{D}_H$ characterizes HR patches, whereas, $\mathbf{D}_L$ characterizes LR patches. Given an input LR patch $\mathbf{q}_i$, the estimate $\hat{\boldsymbol{\alpha}}_i$ of the sparse representation of $\mathbf{q}_i$ is computed by using LR dictionary $\mathbf{D}_L$, and the unknown HR patch $\mathbf{x}_i$ is estimated as $\hat{\mathbf{x}}_i = \mathbf{D}_H \hat{\boldsymbol{\alpha}}_i$. As a consequence of joint learning, the learnt HR and LR dictionaries $\mathbf{D}_H$ and $\mathbf{D}_L$ used by the ScSR algorithm are not well adapted to their respective patch spaces. The coupled dictionary learning algorithm proposed in [5] addresses this drawback by learning HR and LR dictionaries using an objective function minimizing the sparse approximation errors of the HR training patches, and a regularization constraint minimizing sparse approximation errors of LR patches. The SISR algorithm proposed in [88] learns the LR dictionary using K-SVD dictionary learning algorithm, and computes the HR dictionary using least square regression exploiting the assumption of invariance of sparse representations of HR and LR patches as introduced in ScSR.

The Semi Coupled Dictionary Learning (SCDL) algorithm proposed in [6] relaxes the strict assumption of invariance of sparse representations introduced in ScSR. The SCDL algorithm assumes that the sparse representations of HR and LR patches are related through a linear transformation. The SCDL algorithm partitions the entire set of training patches into several smaller clusters containing patches of similar structures, and individually learns HR-LR dictionary pairs and linear transformation matrix mapping the sparse

representations of LR patches to the sparse representations of HR patches for each of these clusters. The Beta Process Joint Dictionary Learning (BPJDL) proposed in [56] exploits a Bayesian approach for joint dictionary learning. The BPJDL algorithm utilizes beta process prior for characterizing sparse representation vectors by assuming that the support of the sparse representations of HR and LR features to be same, while the sparse representation coefficients are related through a linear transformation. This assumption ensures the coupling between HR and LR feature spaces through the invariance of support of sparse representations, and at the same time, allows the individual dictionaries to be well adapted to their respective feature spaces. A general image style transformation algorithm similar to SCDL was proposed in [93] using coupled dictionaries learnt by assuming a common sparse representation support for the corresponding patches from images of different styles. As the intrinsic dimension of the LR patch space is much smaller when compared to the HR patch space, it is reasonable to assume a lesser number of atoms in the LR dictionary when compared with the HR dictionary. The SISR algorithm proposed in [7] uses an undercomplete orthogonal dictionary to represent the LR features, and a traditional overcomplete dictionary to represent the HR features. This method uses a statistical prediction model based on Restricted Boltzmann Machines (RBMs) to infer the sparse representations of HR features from the sparse representations of the corresponding LR features.

The SISR algorithm based on Anchored Neighborhood Regression (ANR) method proposed in [103] fused the idea of sparse representations and neighborhood embedding. The ANR algorithm represents a given LR patch by using least square approximation computed with a small neighborhood of atoms selected from the LR dictionary. The neighborhood of atoms corresponds to a small subset of similar dictionary atoms, where, similarity is measured by using normalized inner product. Assuming that the geometry of neighborhoods of atoms from the LR and HR dictionaries is same, the HR patch estimate corresponding to the LR patch is computed by projecting the least square approximation coefficients onto the neighborhood of HR dictionary atoms corresponding to the neighborhood of LR dictionary atoms. If the neighborhood of LR dictionary atoms contributing to the least square approximation of a given LR patch is known a priori, the estimate of the unknown HR patch can be obtained by multiplying the given LR patch with a projection matrix

computed from the neighborhood of HR and LR dictionary atoms. Corresponding to each of the atoms from the LR dictionary, the ANR algorithm identifies a neighborhood of atoms, and computes the projection matrix for each of the neighborhoods. Given an input LR patch, the ANR algorithm estimates the unknown HR patch by first choosing the neighborhood corresponding to the dictionary atom most correlated to the given LR patch, and then multiplying the given LR patch with the projection matrix corresponding to the chosen neighborhood. The ANR algorithm is much faster than the sparse representation based SISR algorithms as it does not use computationally demanding sparse coding algorithms based on l_0 or l_1 norms. Instead of relying on the neighborhood structure of dictionary atoms, the Adjusted ANR (A+) algorithm [24] associates a neighborhood of LR and HR training patches to each of the atoms from an LR dictionary. Corresponding to each dictionary atom, the A+ algorithm computes the projection matrix using l_2 regression carried out on the neighborhood of HR-LR patch pairs associated to the dictionary atom. Given an input LR patch, the corresponding HR patch estimate is computed by selecting the dictionary atom most correlated to the given patch, and then applying the projection matrix corresponding to the chosen dictionary atom. The A+ algorithm achieves superior performance when compared with ANR, and does not significantly compromise on speed of operation. The SRCNN (Super Resolution using deep Convolutional Neural Networks) algorithm proposed in [105] achieves competing performance when compared to ANR and A+ algorithms by exploiting Convolutional Neural Network (CNN) to learn the end-to-end mapping between HR and LR images. The SRCNN algorithm further showed that the conventional sparse representation based methods for SISR can be interpreted as a deep convolutional neural network. The SISR algorithm proposed in [106] employs Random Forests to learn a set of data dependent and locally linear mappings which maps a given input LR patch to the corresponding HR patch.

3.2 Proposed Method

In this section, we describe the details of our proposed Joint GMM method for SISR. Let us consider the desired magnification factor to be an integer q. The general degradation model in (3.1) assumes that the input LR image is degraded by additive white Gaussian

noise. As considered in most of the previous works on single image SR, we assume that the observed image is noiseless and focus only on recovering the high frequency details lost due to the blur and decimation operators. Following the recent trends in patch based image processing, we first divide the input LR image into a set of overlapping patches of size $\tau \times \tau$ pixels with maximum overlap, i.e., adjacent patches in the horizontal or vertical directions overlap for $\tau \times (\tau - 1)$ pixels. Using vector representation of patches, we have the set of LR patch vectors $\{\mathbf{q}_i\}_{i=1}^{L}$, where, $\mathbf{q}_i \in \mathbb{R}^{\tau^2}$ denotes the i-th patch vector, and L denotes the total number of patches. Corresponding to every $\tau \times \tau$ patch from the LR image, we have a $\tau q \times \tau q$ patch in the HR image. The set of L HR patches corresponding to the set of LR patches is denoted as $\{\mathbf{p}_i\}_{i=1}^{L}$, where, $\mathbf{p}_i \in \mathbb{R}^{\tau^2 q^2}$.

We now define the vector $\mathbf{v}_i \in \mathbb{R}^{\tau^2(q^2+1)}$ by concatenating the HR and the LR patches as,

$$\mathbf{v}_i = \begin{bmatrix} \mathbf{p}_i \\ \mathbf{q}_i \end{bmatrix}. \tag{3.2}$$

Corresponding to the set of LR and HR patches, we have the set of L concatenated vectors $\{\mathbf{v}_i\}_{i=1}^{L}$. The basic idea behind our Joint GMM method is to exploit GMM prior for characterizing concatenated vectors $\{\mathbf{v}_i\}$ containing HR and LR patches. The GMM prior is a mixture of K Gaussian models with parameters $\{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{K}$, where, $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ respectively denote the mean vector and covariance matrix of the k-th Gaussian model. The probability density function (pdf) of a randomly chosen concatenated vector $\mathbf{z}$ can be written as,

$$f(\mathbf{z}) = \sum_{k=1}^{K} \omega_k \Phi(\mathbf{z}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{3.3}$$

where, $\omega_k \in [0, 1]$ denotes the mixing weight of the k-th Gaussian model, and $\Phi(\mathbf{z}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ is the multivariate Gaussian pdf,

$$\Phi(\mathbf{z}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} \exp\left[-\frac{1}{2}(\mathbf{z} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{z} - \boldsymbol{\mu})\right]$$

for $\mathbf{z}, \boldsymbol{\mu} \in \mathbb{R}^n$ and $\boldsymbol{\Sigma} \in \mathbb{R}^{n \times n}$. The GMM prior is completely specified by the parameter set $\boldsymbol{\Theta} = \{\omega_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{K}$.

Each concatenated vector $\mathbf{v}_i$ is assumed to be generated by a Gaussian model selected

from the mixture. The mixing weight ω_k, $k = 1, 2, ..., K$, represents the a priori probability that the k-th Gaussian model is responsible for a randomly selected concatenated vector. Consequently, we have, $\sum_{k=1}^{K} \omega_k = 1$. The index $k_i \in [1, 2, ..., K]$ of the Gaussian model responsible for the i-th concatenated vector $\mathbf{v}_i$ is assumed to be unknown for all $i = 1, 2, ..., L$. In practice, the index k_i of the Gaussian model responsible for $\mathbf{v}_i$ can be estimated as follows. Let I_i denotes the random variable which takes the value $k \in \{1, 2, ..., K\}$, if the i-th concatenated vector $\mathbf{v}_i$ is generated by the k-th Gaussian model. The posterior probability γ_{ik} that the k-th Gaussian model is responsible for a given $\mathbf{v}_i$ can be written as,

$$\begin{aligned} \gamma_{ik} &= p\left(I_i = k \,|\mathbf{v}_i\right) \\ &\propto p\left(\mathbf{v}_i \,|I_i = k\right) p\left(I_i = k\right) \\ &= p\left(\mathbf{v}_i \,|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right) p\left(I_i = k\right) \\ &= \Phi\left(\mathbf{v}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right) \omega_k. \end{aligned} \tag{3.4}$$

The index k_i of the Gaussian model responsible for $\mathbf{v}_i$ can be estimated by maximizing the posterior responsibilities over all Gaussian models as,

$$\begin{aligned} \hat{k}_i &= \arg\max_{k \in [1,..,K]} \gamma_{ik} \\ &= \arg\max_{k \in [1,..,K]} \Phi\left(\mathbf{v}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right) \omega_k, \end{aligned} \tag{3.5}$$

where, $\hat{k}_i$ denotes the estimate of the index k_i. As each concatenated vector $\mathbf{v}_i$ is assumed to be generated by a Gaussian model with index $\hat{k}_i$, the entire set of concatenated vectors can be partitioned into K clusters, where, k-th cluster contains all the concatenated vectors generated by the k-th Gaussian model. As the vectors $\{\mathbf{v}_i\}$ are defined by concatenating HR and LR patches, clusters corresponding to the K Gaussian models can be interpreted as K pairs of clusters. The HR part of each concatenated vector from a cluster forms a cluster in the HR patch space, and the LR part forms a cluster in the LR patch space.

We can observe that the GMM prior characterizing concatenated vectors consists of Gaussian models whose parameters capture the correlations between HR and the corresponding

LR patches. The parameters of the k-th Gaussian model can be interpreted as

$$\boldsymbol{\mu}_k = \begin{bmatrix} \boldsymbol{\mu}_{H_k} \\ \boldsymbol{\mu}_{L_k} \end{bmatrix}, \tag{3.6}$$

$$\boldsymbol{\Sigma}_k = \begin{bmatrix} \boldsymbol{\Sigma}_{H_k} & \boldsymbol{\Sigma}_{HL_k} \\ \boldsymbol{\Sigma}_{HL_k}^T & \boldsymbol{\Sigma}_{L_k} \end{bmatrix}, \tag{3.7}$$

where, $\boldsymbol{\mu}_{H_k}$ and $\boldsymbol{\Sigma}_{H_k}$ respectively denote the mean vector and covariance matrix corresponding to the HR part of the k-th Gaussian model, and, $\boldsymbol{\mu}_{L_k}$ and $\boldsymbol{\Sigma}_{L_k}$ denote the mean and covariance matrix corresponding to the LR part. The cross covariance matrix $\boldsymbol{\Sigma}_{HL_k}$ characterizes the correlations between HR-LR patch pairs. We call this mixture of Gaussian models as Joint GMM, as each Gaussian model characterizes the individual HR and LR patches, and also the correlations between them. In the case of image super resolution, we are specifically interested in the cross covariance matrices $\boldsymbol{\Sigma}_{HL_k}$, as the HR-LR patch correlations captured by these matrices can be utilized to make predictions about an unknown HR patch corresponding to a given LR patch. We construct a large database of concatenated vectors $\{\mathbf{v}_i\}$ by extracting HR-LR patch pairs from a collection of natural images, and learn the Joint GMM parameters using Expectation Maximization (EM) algorithm. The learnt parameters are used to estimate the unknown HR patches corresponding to the LR patches extracted from a given input LR image. Next, we describe the details of GMM learning procedure and the SR reconstruction algorithm that estimates the HR patches corresponding to the LR patches.

3.2.1 Joint GMM Learning

We use a large database of HR and corresponding LR patches of natural images to learn the Joint GMM parameters $\{\boldsymbol{\omega}_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K$. Let us consider a set of M HR patches $\{\mathbf{p}_i\}_{i=1}^M$ and the corresponding LR patches $\{\mathbf{q}_i\}_{i=1}^M$ sampled from a collection of HR-LR pairs of natural images. We have $\mathbf{p}_i \in \mathbb{R}^{\tau^2 q^2}$, and $\mathbf{q}_i \in \mathbb{R}^{\tau^2}$, where, the integer q is the desired magnification factor for which the GMM parameters are to be learnt. The training database $\{\mathbf{v}_i\}_{i=1}^M$ is obtained by concatenating each $\mathbf{p}_i$ and $\mathbf{q}_i$ as $\mathbf{v}_i = \left[\mathbf{p}_i^T, \mathbf{q}_i^T\right]^T$. Note that we construct the training database by directly concatenating HR and LR patches

without applying any pre-processing. Most of the previous works on SISR using joint dictionaries have used pre-processing, mainly in the form of high pass filtering, to extract the high frequency details which are more important in the context of super resolution.

Given the training vectors $\{\mathbf{v}_i\}_{i=1}^{M}$, the maximum Likelihood (ML) estimates $\mathbf{\Theta}^{ML} = \left\{\hat{\omega}_k^{ML}, \hat{\boldsymbol{\mu}}_k^{ML}, \hat{\mathbf{\Sigma}}_k^{ML}\right\}_{k=1}^{K}$ of the unknown Joint GMM parameters $\mathbf{\Theta} = \{\omega_k, \boldsymbol{\mu}_k, \mathbf{\Sigma}_k\}_{k=1}^{K}$ can be obtained by maximizing the likelihood of training vectors as,

$$\begin{aligned}
\mathbf{\Theta}^{ML} &= \arg\max_{\tilde{\mathbf{\Theta}}} p\left(\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_M \,\middle|\, \tilde{\mathbf{\Theta}}\right) \\
&= \arg\max_{\tilde{\mathbf{\Theta}}} \prod_{i=1}^{M} p\left(\mathbf{v}_i \,\middle|\, \tilde{\mathbf{\Theta}}\right) \\
&= \arg\max_{\tilde{\mathbf{\Theta}}} \prod_{i=1}^{M} \sum_{k=1}^{K} \tilde{\omega}_k \Phi\left(\mathbf{v}_i; \tilde{\boldsymbol{\mu}}_k, \tilde{\mathbf{\Sigma}}_k\right) \\
&= \arg\min_{\tilde{\mathbf{\Theta}}} -\sum_{i=1}^{M} \log \sum_{k=1}^{K} \tilde{\omega}_k \Phi\left(\mathbf{v}_i; \tilde{\boldsymbol{\mu}}_k, \tilde{\mathbf{\Sigma}}_k\right),
\end{aligned} \tag{3.8}$$

where, $\tilde{\mathbf{\Theta}} = \left\{\tilde{\omega}_k, \tilde{\boldsymbol{\mu}}_k, \tilde{\mathbf{\Sigma}}_k\right\}_{k=1}^{K}$. As the solution of the ML estimation problem given in (3.8) cannot be obtained closed form, a commonly used method for estimating GMM parameters is to compute an approximate solution $\hat{\mathbf{\Theta}} = \left\{\hat{\omega}_k, \hat{\boldsymbol{\mu}}_k, \hat{\mathbf{\Sigma}}_k\right\}_{k=1}^{K}$ by using the Expectation-Maximization (EM) algorithm. The EM algorithm iteratively computes the parameter estimates corresponding to a local maximum of the likelihood function. The EM algorithm is widely popular in the machine learning and statistics literature, and is summarized in Appendix A.

In contrast with the MAP-EM algorithm [94] utilized by one of the previous works (PLE) on image restoration using GMM prior, our proposed method utilizes the traditional EM algorithm commonly used in learning the GMM parameters. The MAP-EM algorithm [94] performs a hard clustering on the training database by assigning each training vector $\mathbf{v}_i$ to one of the Gaussian models which maximizes the likelihood of observation of $\mathbf{v}_i$, and estimates the parameters of a given Gaussian model by computing the empirical mean and covariance matrices of those training vectors assigned to the given Gaussian model. On the other hand, the traditional EM algorithm used in our proposed method performs a soft clustering on the training database, and each training vector contributes to the estimation

of parameters of any given Gaussian model by appropriately weighting the contributions with respective likelihood values.

3.2.2 SR Reconstruction

In this section, we describe the SR reconstruction algorithm which computes the estimate $\hat{\mathbf{x}}$ of the unknown HR image $\mathbf{x}$ from the corresponding LR observation $\mathbf{y}$ by exploiting the Joint GMM parameters learnt from a training database as discussed in section 3.2.1. A given input LR image $\mathbf{y}$ to be super resolved is first divided into the set of all possible overlapping patches $\{\mathbf{q}_i\}_{i=1}^{L}$, where, L denotes the total number of LR patches. The proposed SR reconstruction algorithm computes the estimates $\{\hat{\mathbf{p}}_i\}_{i=1}^{L}$ of the unknown HR patches $\{\mathbf{p}_i\}_{i=1}^{L}$, from the LR patches $\{\mathbf{q}_i\}_{i=1}^{L}$, and aggregates the overlapping HR patch estimates to compute the estimate $\hat{\mathbf{x}}$ of the unknown HR image. The concatenated vector $\mathbf{v}_i$ corresponding to the input LR patch $\mathbf{q}_i$ can be written as $\mathbf{v}_i = \left[\mathbf{p}_i^T, \mathbf{q}_i^T\right]^T$, where, the $\mathbf{p}_i$ is the unknown HR patch to be estimated. The Joint GMM prior assumes that the concatenated vector $\mathbf{v}_i$ is generated by a Gaussian model with unknown mixture index k_i. As $\mathbf{v}_i$ is assumed to be jointly Gaussian, the unknown HR patch $\mathbf{p}_i$ is also Gaussian. If the index k_i of the Gaussian model responsible for $\mathbf{v}_i$ was known a priori, the MMSE estimate of the unknown HR patch corresponds to the conditional expectation $\mathbb{E}\left[\mathbf{p}_i \,|\mathbf{q}_i\right]$, where, $\left[\mathbf{p}_i^T, \mathbf{q}_i^T\right]^T \sim \mathcal{N}\left(\hat{\boldsymbol{\mu}}_{k_i}, \hat{\boldsymbol{\Sigma}}_{k_i}\right)$. The conditional expectation can be obtained in closed form (a brief outline of the derivation is given in Appendix B) as,

$$\mathbb{E}\left[\mathbf{p}_i \,|\mathbf{q}_i\right] = \hat{\boldsymbol{\mu}}_{H_{k_i}} + \hat{\boldsymbol{\Sigma}}_{HL_{k_i}} \hat{\boldsymbol{\Sigma}}_{L_{k_i}}^{-1} \left(\mathbf{q}_i - \hat{\boldsymbol{\mu}}_{L_{k_i}}\right) \tag{3.9}$$

where, $\hat{\boldsymbol{\mu}}_{H_{k_i}}$, $\hat{\boldsymbol{\mu}}_{L_{k_i}}$, $\hat{\boldsymbol{\Sigma}}_{HL_{k_i}}$ and $\hat{\boldsymbol{\Sigma}}_{L_{k_i}}$ are as shown in 3.6 and 3.7. Note that, $\hat{\boldsymbol{\mu}}_{H_{k_i}}$ and $\hat{\boldsymbol{\mu}}_{L_{k_i}}$ respectively denote the HR and LR parts of the mean vector $\hat{\boldsymbol{\mu}}_{k_i}$, $\hat{\boldsymbol{\Sigma}}_{L_{k_i}}$ denotes the LR part of the covariance matrix $\hat{\boldsymbol{\Sigma}}_{k_i}$, and, $\hat{\boldsymbol{\Sigma}}_{HL_{k_i}}$ denotes the cross covariance matrix capturing HR-LR patch correlations. However, the index k_i of the Gaussian model responsible for $\mathbf{v}_i = \left[\mathbf{p}_i^T, \mathbf{q}_i^T\right]^T$ is not known in practice, and thus needs to be estimated. As only the LR part $\mathbf{q}_i$ of the concatenated vector $\mathbf{v}_i$ is available, the index k_i is estimated by maximizing

the posterior probability $\gamma_{L_{ik}}$ of $\mathbf{q}_i$ defined as,

$$\begin{aligned}\gamma_{L_{ik}} &\propto p\left(\mathbf{q}_i \left| \hat{\boldsymbol{\mu}}_{L_k}, \hat{\boldsymbol{\Sigma}}_{L_k}\right.\right) \hat{\omega}_k \\ &= \Phi\left(\mathbf{q}_i; \hat{\boldsymbol{\mu}}_{L_k}, \hat{\boldsymbol{\Sigma}}_{L_k}\right) \hat{\omega}_k, \end{aligned} \tag{3.10}$$

over all Gaussian models $k = [1, 2, ..., K]$. The estimate $\hat{k}_i$ of the unknown index k_i is computed as,

$$\begin{aligned}\hat{k}_i &= \underset{k\in[1,..,K]}{\arg\max}\, \gamma_{L_{ik}} \\ &= \underset{k\in[1,..,K]}{\arg\max}\, \Phi\left(\mathbf{q}_i; \hat{\boldsymbol{\mu}}_{L_k}, \hat{\boldsymbol{\Sigma}}_{L_k}\right) \hat{\omega}_k. \end{aligned} \tag{3.11}$$

Given the estimate $\hat{k}_i$ of the index of the Gaussian model responsible for $\mathbf{v}_i$, the required estimate $\hat{\mathbf{p}}_i$ of the unknown HR patch $\mathbf{p}_i$ is computed as,

$$\begin{aligned}\hat{\mathbf{p}}_i &= \boldsymbol{\mu}_{H_{\hat{k}_i}} + \boldsymbol{\Sigma}_{HL_{\hat{k}_i}} \boldsymbol{\Sigma}^{-1}_{L_{\hat{k}_i}} \left(\mathbf{q}_i - \boldsymbol{\mu}_{L_{\hat{k}_i}}\right) \\ &= \boldsymbol{\mu}_{H_{\hat{k}_i}} + \mathbf{W}_{\hat{k}_i} \left(\mathbf{q}_i - \boldsymbol{\mu}_{L_{\hat{k}_i}}\right) \end{aligned} \tag{3.12}$$

where, $\mathbf{W}_{\hat{k}_i} = \boldsymbol{\Sigma}_{HL_{\hat{k}_i}} \boldsymbol{\Sigma}^{-1}_{L_{\hat{k}_i}}$ is the Wiener filter derived from the covariance matrix of the selected Gaussian model indexed by $\hat{k}_i$.

Note that the matrix inversion in (3.12) is well defined due to the eigenvalue regularization used while learning the covariance matrices using EM algorithm. The proposed SR reconstruction method computes the HR patch estimates $\{\hat{\mathbf{p}}_i\}_{i=1}^{L}$ corresponding to all the overlapping LR patches $\{\mathbf{q}_i\}_{i=1}^{L}$ using (3.12). The HR patch estimates computed in (3.12) corresponds to a set of overlapping patches in the desired HR image. If LR patches are of size $\tau \times \tau$ pixels, the estimated HR patches are of size $\tau q \times \tau q$ pixels, and for each pixel in the desired HR image (except for those pixels on the image boundaries), there will be τ^2 estimates available from different overlapping LR patches. There are different ways to combine multiple estimates of a pixel available from the overlapping patches. The simplest method is to tile up all the overlapping patches at their respective positions and compute the average of the multiple estimates of a pixel available from different overlapping patches. Simple averaging results in blurring of the estimated HR image, when the

multiple estimates corresponding to a pixel are not consistent. In our proposed method, we use a weighted averaging of overlapping patches. The highest weight is assigned to center pixel of a patch, and the weight decreases towards the patch boundaries. We use a Gaussian window of same size as our patch size to weight the estimated overlapping patches before averaging.

Our proposed SR reconstruction algorithm involves a hard clustering step and a linear filtering step to compute the estimate of unknown HR patch corresponding to a given LR patch. The hard clustering step associates the given LR patch to the best Gaussian model responsible for the patch. The linear filtering step estimates the unknown HR patch using the Wiener filter derived from the covariance matrix of the Gaussian model to which the LR patch is associated. The overall algorithm is piecewise linear with a non linear Gaussian model selection and a set of K Wiener filters, where, K is the total number of Gaussian models. The piecewise linear nature makes the proposed SR reconstruction algorithm computationally much simpler compared to sparse representation based methods. Sparse coding algorithms are typically iterative and computationally demanding, whereas, the proposed method summarized in**Algorithm 1** involves only a one pass filtering on the LR patches and is not iterative.

Algorithm 1 : SR Reconstruction Algorithm

1: **Input:** LR patches $\{\mathbf{q}_i\}_{i=1}^{L}$, Joint GMM parameters $\hat{\mathbf{\Theta}} = \left\{\hat{\omega}_k, \hat{\boldsymbol{\mu}}_k, \hat{\mathbf{\Sigma}}_k\right\}_{k=1}^{K}$

2: **Hard Clustering:** For each LR patch $\mathbf{q}_i$, select the index $\hat{k}_i$ of the Gaussian model responsible for the patch as,

$$\hat{k}_i = \underset{k\in[1,..,K]}{\arg\max}\, \gamma_{L_{ik}}$$

3: **Wiener Filtering:** For each LR patch $\mathbf{q}_i$, compute the estimate $\hat{\mathbf{p}}_i$ of the HR patch as,

$$\hat{\mathbf{p}}_i = \hat{\boldsymbol{\mu}}_{H_{\hat{k}_i}} + \hat{\mathbf{\Sigma}}_{HL_{\hat{k}_i}} \hat{\mathbf{\Sigma}}_{L_{\hat{k}_i}}^{-1} \left(\mathbf{q}_i - \hat{\boldsymbol{\mu}}_{L_{\hat{k}_i}}\right)$$

4: **Patch Averaging:** Compute the weighted average of overlapping HR patch estimates by arranging the patches at their respective locations and averaging the overlapping patches weighed by a Gaussian window.

5: **Return:** Estimated HR image $\hat{\mathbf{x}}$.

3.3 Experiments and Results

In this section, we describe the experimental set up used to evaluate our proposed algorithm and present the results obtained by conducting SR experiments on a set of natural images. As our proposed SISR algorithm based on Joint GMM method can be interpreted as a GMM analogue of the joint dictionary based approach introduced in ScSR [4], the proposed algorithm is mainly compared with ScSR and various joint dictionary based methods for SISR. Apart from ScSR, the proposed algorithm is also compared with its modified versions Semi Coupled Dictionary Learning algorithm (SCDL) [6], Beta Process Joint Dictionary Learning algorithm (BPJDL) [56], Statistical Prediction Model (SPM) based method [7] and the recent state of the art SISR algorithm - Adjusted Anchored Neighborhood Regression (A+)[24]. We have also compared the performance of our proposed method with a recent patch based algorithm [106] using Random Forests (SRF), as it offers competing performance when compared with the state of art A+ algorithm. As a baseline reference, the simple bicubic interpolation method is also included in the performance comparisons.

As considered in most of the recent works [4, 56, 7, 24, 106] in SISR, the blur kernel $\mathbf{H}$ in (3.1) is assumed to be the kernel corresponding to bicubic interpolation, and SR experiments are conducted for two different magnification factors $q = 2$ and $q = 3$. Note that our proposed method does not explicitly use the degradation kernel in the SR algorithm. The blur kernel is used only for synthesizing LR training images from the corresponding HR images. The synthesized HR-LR pairs of training images are utilized to learn the Joint GMM parameters. The basic intention of our performance comparison studies is to analyze the capabilities of various algorithms in learning the co-occurrence relations between HR-LR patch pairs using a given training database containing several example HR-LR patch pairs. The transformation between HR and the corresponding LR images are assumed to be same but unknown (except for the downsampling factor q) for all the images in the training database. As the transformation from HR to LR images is assumed to be unknown, the back projection step used by ScSR and BPJDL algorithms is turned off in all the experiments, as back projection explicitly uses the knowledge of blur kernel. The SCDL algorithm [6] was originally proposed and used only for image zooming,

which assumes that the input LR image is generated through the direct decimation of an unknown HR image without blurring (or equivalently, the blur kernel is a dirac delta function). As our aim is to study how well a joint learning strategy performs in a general SR problem regardless of the particular degradation model that produces LR images from the corresponding HR images, we have learnt coupled dictionaries using SCDL method for the case of general SR scenario, and tested the same in our experiments.

3.3.1 Joint GMM Learning

The Joint GMM parameters are learnt from a database containing 103 natural images obtained by merging the publicly available training databases used by various joint dictionary based methods [4, 56, 24] previously proposed in the literature. All the color images from the database are converted into monochrome images. Each HR image from the database is degraded as in (3.1) to synthesize the corresponding set of LR training images. A training database of HR-LR pairs is generated for each of the cases corresponding to the two magnification factors $q = 2$ and $q = 3$. Separate Joint GMMs are learnt for each of these cases by using a set of $M = 10^6$ HR-LR patch pairs randomly sampled from the respective databases. The HR and LR patch dimensions are respectively chosen as 8×8 and 4×4 for $q = 2$, and, 9×9 and 3×3 for $q = 3$. The HR and LR patches are vectorized and concatenated to construct the training database $\{\mathbf{v}_i\}_{i=1}^{M}$ of concatenated vectors. The DC component is removed from each of the LR patches before they are used in training. The estimates $\hat{\mathbf{\Theta}} = \left\{\hat{\omega}_k, \hat{\boldsymbol{\mu}}_k, \hat{\mathbf{\Sigma}}_k\right\}_{k=1}^{K}$ of the Joint GMM parameters are computed from the training database of concatenated vectors by using the traditional Expectation-Maximization algorithm. We used the publicly available[1] MATLAB code for learning the GMM parameters using EM algorithm.

The Initialization step of the EM algorithm shown in Appendix A requires the training database of concatenated vectors to be partitioned into K clusters $\{\mathcal{C}_k\}_{k=1}^{K}$, where K is the total number of Gaussian models. The training database is partitioned into K clusters by randomly sampling K training vectors as cluster centers, and assigning each training vector to one of the K clusters by using nearest neighbor classification under Euclidean

[1] http://in.mathworks.com/matlabcentral/fileexchange/26184-em-algorithm-for-gaussian-mixture-model

distance measure. Each training vector is associated to the cluster whose cluster center is the nearest when compared to others. Given the K clusters of training vectors, initial values of the joint GMM parameters are obtained by computing the empirical weights, means and covariance matrices of each of these clusters.

In order to choose an appropriate number of Gaussian models K, we conducted a series of SR experiments on a set of natural images for magnification factor $q = 2$, by using Joint GMMs learnt independently with different number of Gaussian models. The number of Gaussian models K is varied from 25 to 325 with a step size of 25. The Peak Signal to Noise Ratio (PSNR) and Structural Similarity Index (SSIM) values of estimated HR images are computed for each choice of K. Table 3.1 shows the average PSNR and SSIM values (averaged over all the test images) obtained for different number of Gaussian models K. The test Images used to conduct this experiment are selected from the B100 test data set (test images from the Berkeley segmentation data set) used in the A+ [24] algorithm. These images have similar PSNR(SSIM) performance in SR so that averaging the PSNR(SSIM) values is more sensible. As can be observed from Table 3.1, average PSNR increases by 0.13 dB as the number of Gaussian models K increases from $K = 25$ to 225. The PSNR and SSIM performance improves as the number of Gaussian models K increases from 25 to 225, saturates thereafter, and even drops slightly when the number of Gaussian models is increased to very large values such as 300 and 325. Table 3.1 also shows the average run time taken by our SR algorithm in seconds per 10^4 input LR pixels, for different number of Gaussian models. As expected, average run time increases with number of Gaussian models, due to the increase in the complexity of model selection step used by the SR reconstruction algorithm. It can be noted from Table 3.1 that the best performance in terms of PSNR and SSIM measures is obtained in two different cases corresponding to $K = 225$ and $K = 250$. We have chosen the number of Gaussian models to be $K = 225$ as it gives the best performance with a smaller run time.

The estimates of Joint GMM parameters obtained from the EM algorithm depends on the initialization of GMM parameters. The initialization in turn depends on the random clustering carried out on the training database as described above. In order to study the sensitivity of estimated Joint GMM parameters to the initialization, we independently

K	Avg PSNR	Avg SSIM	Avg Run Time
25	30.34	0.8404	0.147
50	30.39	0.8415	0.188
75	30.41	0.8418	0.218
100	30.42	0.8420	0.256
125	30.44	0.8423	0.281
150	30.45	0.8424	0.317
175	30.46	0.8424	0.360
200	30.47	0.8426	0.378
225	30.47	0.8427	0.397
250	30.47	0.8427	0.442
275	30.47	0.8425	0.483
300	30.47	0.8426	0.517
325	30.46	0.8424	0.548

TABLE 3.1: Performance Vs No. of Gaussian Models (K)

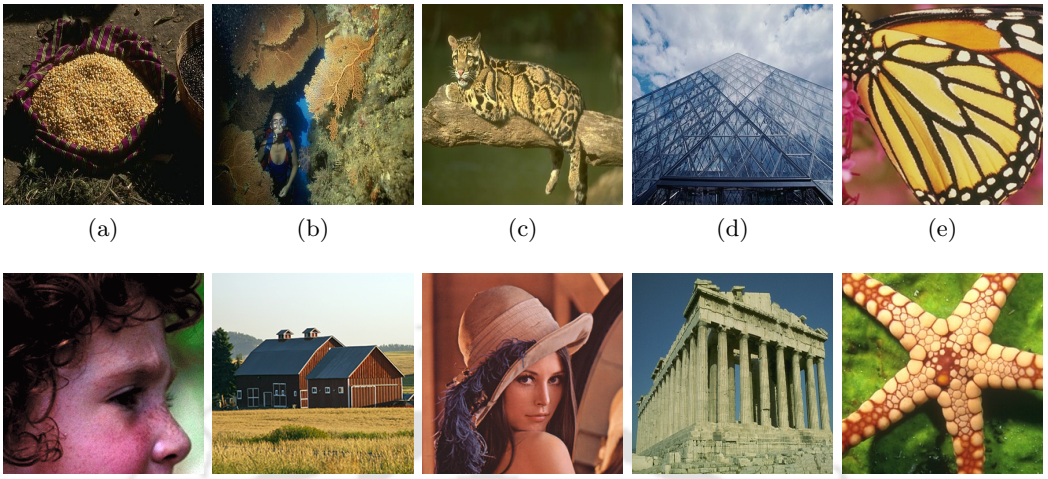
Instance	Avg PSNR	Avg SSIM
1	30.43	0.8422
2	30.43	0.8421
3	30.42	0.8421
4	30.43	0.8422
5	30.43	0.8421
6	30.43	0.8422
7	30.43	0.8422
8	30.42	0.8420
9	30.43	0.8422
10	30.43	0.8422
std.dev	0.0042	7.07×10^{-5}

TABLE 3.2: Sensitivity to Initialization

learnt 10 instances of Joint GMM parameters with $K = 100$ Gaussian models for magnification factor 2. Table 3.2 shows the average PSNR and SSIM values obtained by independently conducting SR experiments using each of the 10 instances of the learnt Joint GMM parameters. As can be seen from the last column of Table 3.2, the PSNR and SSIM values have negligible standard deviations of 0.0042 dB and 7.07×10^{-5} respectively. Hence, we conclude that the Joint GMM parameters learnt by using EM algorithm are relatively insensitive to the randomness in the initialization.

3.3.2 Performance Comparison

The proposed Joint GMM method for SISR is evaluated by comparing the SR performance with ScSR, SCDL, BPJDL, SPM, A+ and SRF algorithms on a set of 10 natural images shown in Figure 3.1. These images are selected from the test dataset used by various SISR algorithms [4, 6, 56, 24] previously proposed in the literature. The color images from the test dataset are converted from RGB color space to YCbCr, and the proposed SR algorithm is applied only on the luminance channel Y. The chrominance channels Cb and Cr are magnified by using simple bicubic interpolation. The SR performance of various



(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

FIGURE 3.1: Test images used in SR experiments: (a) Grains, (b) Coral, (c) Leopard, (d) Louvre, (e) Butterfly, (f) Child, (g) House, (h) Lena, (i) Parthenon, (j) Starfish

algorithms is measured by using PSNR and SSIM values. In all the experiments, our proposed algorithm uses a Gaussian weighting function with $\sigma = 1.5$ for patch averaging.

Figure 3.2 shows the estimated HR images produced by various SR algorithms in the case of SR experiment conducted on Lena image for magnification factor 2. A small portion of the image is magnified and shown as inset in all the images for a better subjective analysis of capabilities of various algorithms in preserving the fine details of the image. We can observe that the fine details of the estimated HR image are better preserved by the proposed algorithm when compared with the images produced by ScSR and its modified versions SCDL and BPJDL. The HR estimate produced by our proposed method appears to be much sharper than the one obtained from the SCDL algorithm. The visual quality of the super-resolved image produced by the proposed method is closely similar with no noticeable differences when compared to the HR estimates produced by state of the art SR algorithms such as A+ and SRF. The visual comparison between HR estimates produced by the proposed Joint GMM method and various SR algorithms in the case of different magnification factors and several images is given in Appendix C.

Table 3.3 shows the SR performance of various algorithms for magnification factor 2. In the case of our proposed Joint GMM method, the last two columns show the SR performance obtained with two different choice of number of Gaussian models, i.e., $K = 100$ and

FIGURE 3.2: Results of SR experiment conducted on Lena image for magnification factor 2. (a) Bicubic, (b) ScSR, (c) SCDL, (d) BPJDL, (e) SPM, (f) A+, (g) SRF, (h) Prop (K=100), (i) Prop (K=225), (j) Original

$K = 225$. Each cell of the table is divided into two parts, with the upper and lower parts respectively showing the PSNR and SSIM values obtained by a given algorithm on a given image. The PSNR and SSIM values within a gap of 0.1 dB and 0.0050 respectively when compared with the corresponding best (highest) values are shown in boldface. As can be observed from the PSNR and SSIM values shown in Table 3.3, the proposed method with $K = 225$ Gaussian models significantly outperforms the ScSR algorithm and its modified versions SCDL and BPJDL in all the images. The proposed method with 225 Gaussian models achieves an average PSNR improvement of 0.35 dBs when compared with ScSR algorithm. The performance improvement of the proposed method over SCDL and BPJDL algorithms in terms of average PSNR is 2.34 dBs and 0.81 dBsrespectively. In

	Bicubic	ScSR	SCDL	BPJDL	SPM	A+	SRF	Prop (K=100)	Prop (K=225)
Grains	23.11	23.70	22.83	23.42	**24.02**	**24.07**	**24.07**	**23.97**	**23.99**
	0.7441	0.7836	0.7275	0.7681	**0.8090**	**0.8119**	**0.8089**	0.8046	0.8056
Coral	26.48	26.92	25.83	26.61	**27.12**	**27.15**	**27.12**	27.03	**27.05**
	0.7242	0.7582	0.6862	0.7379	**0.7811**	**0.7813**	**0.7774**	0.7714	0.7722
Leopard	29.78	30.74	28.73	30.18	**31.21**	**31.09**	**31.15**	30.93	30.97
	0.9082	0.9255	0.8921	0.9170	**0.9360**	**0.9345**	**0.9344**	0.9308	**0.9311**
Louvre	26.29	27.14	26.02	26.97	27.40	**27.57**	27.41	**27.49**	**27.53**
	0.7860	0.8284	0.7748	0.8196	0.8474	**0.8529**	0.8457	0.8464	**0.8479**
Butterfly	27.41	30.85	26.94	29.64	30.97	32.02	**32.26**	30.68	31.35
	0.9155	0.9561	0.9193	0.9421	0.9564	**0.9647**	**0.9645**	0.9572	**0.9617**
Child	34.71	35.38	33.30	34.91	**35.68**	**35.68**	**35.64**	35.53	35.54
	0.8648	0.8824	0.8380	0.8718	**0.8897**	**0.8898**	**0.8879**	**0.8850**	**0.8853**
House	26.25	27.17	26.04	26.98	**27.61**	**27.70**	**27.62**	27.37	27.57
	0.8039	0.8365	0.7956	0.8242	**0.8553**	**0.8578**	**0.8546**	0.8477	0.8520
Lena	32.58	34.45	31.49	33.93	34.94	**35.26**	35.01	34.81	35.03
	0.9071	0.9303	0.8915	0.9229	**0.9376**	**0.9410**	**0.9378**	0.9354	**0.9371**
Parthenon	28.01	29.00	27.55	28.65	29.18	**29.42**	**29.38**	29.07	29.18
	0.8045	0.8362	0.7907	0.8239	**0.8483**	**0.8517**	**0.8498**	0.8431	0.8452
Starfish	30.09	32.20	28.96	31.67	**32.78**	**32.88**	**32.82**	32.54	**32.83**
	0.9068	0.9362	0.8867	0.9269	**0.9430**	**0.9450**	**0.9439**	**0.9412**	**0.9432**
Average	28.47	29.75	27.76	29.29	30.09	**30.28**	**30.24**	29.94	30.10
	0.8365	0.8673	0.8202	0.8554	**0.8803**	**0.8830**	**0.8804**	0.8762	**0.8781**

TABLE 3.3: PSNR and SSIM performance in the case of SR by a factor of 2

	Bicubic	ScSR	SCDL	BPJDL	SPM	A+	SRF	Proposed (K=100)	Proposed (K=225)
Grains	21.24	21.40	20.97	21.40	**21.63**	**21.68**	**21.68**	21.57	21.56
	0.6001	0.6179	0.5782	0.6207	0.6562	**0.6610**	**0.6567**	0.6437	0.6439
Coral	24.94	25.11	24.28	25.10	**25.31**	**25.35**	**25.33**	25.18	25.18
	0.5910	0.6066	0.5492	0.6064	0.6330	**0.6397**	**0.6369**	0.6211	0.6212
Leopard	27.00	27.38	26.11	27.52	**27.94**	**27.95**	**27.96**	27.64	27.61
	0.8312	0.8427	0.8113	0.8467	**0.8643**	**0.8653**	**0.8645**	0.8556	0.8551
Louvre	24.54	24.81	24.12	24.89	**25.12**	**25.25**	25.13	24.97	24.96
	0.6615	0.6818	0.6365	0.6886	**0.7213**	**0.7244**	0.7166	0.7040	0.7043
Butterfly	24.06	26.02	23.87	26.58	26.84	**27.33**	27.16	26.25	26.47
	0.8230	0.8870	0.8498	0.8937	0.9018	**0.9111**	0.9035	0.8942	0.9004
Child	32.59	33.01	30.94	33.13	**33.47**	**33.57**	**33.51**	33.18	33.24
	0.7996	0.8109	0.7629	0.8127	0.8233	**0.8287**	**0.8264**	0.8172	0.8178
House	24.26	24.65	23.86	24.70	**25.00**	**25.03**	**24.96**	24.75	24.84
	0.6930	0.7134	0.6793	0.7167	0.7357	**0.7411**	**0.7373**	0.7250	0.7286
Lena	29.77	30.87	28.35	31.02	31.50	**31.74**	31.59	31.16	31.22
	0.8318	0.8542	0.8080	0.8583	0.8692	**0.8768**	**0.8725**	0.8647	0.8663
Parthenon	25.84	26.37	25.17	26.44	26.66	**26.84**	**26.79**	26.42	26.43
	0.6879	0.7107	0.6672	0.7134	0.7283	**0.7362**	**0.7324**	0.7185	0.7192
Starfish	26.75	27.46	25.48	27.71	**28.50**	28.31	28.31	27.85	27.82
	0.8080	0.8316	0.7794	0.8376	**0.8618**	**0.8600**	**0.8591**	0.8476	0.8472
Average	26.09	26.70	25.31	26.84	**27.20**	**27.30**	**27.24**	26.89	26.93
	0.7327	0.7556	0.7121	0.7594	**0.7794**	**0.7844**	**0.7805**	0.7691	0.7704

TABLE 3.4: PSNR and SSIM performance in the case of SR by a factor of 3

	Grains	Coral	Leopard	Louvre	Butterfly	Child	House	Lena	Parthenon	Starfish	Average
Prop	23.99	27.05	30.97	27.53	31.35	35.54	27.57	35.03	29.18	32.83	30.10
	0.8056	0.7722	0.9311	0.8479	0.9617	0.8853	0.8520	0.9371	0.8452	0.9432	0.8781
SRCNN	24.29	27.20	31.30	27.65	32.80	35.61	28.09	35.36	29.55	33.69	30.55
	0.8209	0.7832	0.9365	0.8524	0.9649	0.8889	0.8644	0.9401	0.8528	0.9496	0.8853

TABLE 3.5: Comparison with SRCNN (magnification factor 2)

	Grains	Coral	Leopard	Louvre	Butterfly	Child	House	Lena	Parthenon	Starfish	Average
Prop	21.56	25.18	27.61	24.96	26.47	33.24	24.84	31.22	26.43	27.82	26.93
	0.6439	0.6212	0.8551	0.7043	0.9004	0.8178	0.7286	0.8663	0.7192	0.8472	0.7704
SRCNN	21.76	25.38	28.13	25.20	28.06	33.50	25.32	31.84	27.02	28.94	27.51
	0.6696	0.6418	0.8683	0.7206	0.9114	0.8285	0.7488	0.8772	0.7408	0.8723	0.7879

TABLE 3.6: Comparison with SRCNN (magnification factor 3)

terms of PSNR measure, the proposed method outperforms the SPM algorithm in 5 out of 10 images, and outperforms the SRF algorithm in 3 out of 10 images. The proposed method outperforms SPM with a slight improvement of 0.01 dB in average PSNR whereas the SRF algorithm shows a slight improvement of 0.14 dB over the proposed method. The A+ algorithm, which can be considered as the present state of the art algorithm in patch based single image SR, achieves slightly better performance when compared with the proposed method in all the images. However, the performance gap between A+ and our proposed method is negligibly small in most of the images. This clearly demonstrates the superior performance achieved by the proposed Joint GMM method when compared with the sparsity based ScSR algorithm and its modified versions, and the competing performance when compared with the state of art algorithms such as A+ and SRF. In the case of proposed method with $K = 100$ Gaussian models, A+ and SRF algorithms have a performance improvement of 0.31 and 0.23 dBs respectively in terms of average PSNR.

Table 3.4 similarly shows the PSNR and SSIM values obtained by various algorithms for magnification factor 3. As can be observed from Table 3.4, the proposed method with $K = 225$ Gaussian models consistently outperforms ScSR, SCDL and BPJDL algorithms in terms of average PSNR, with an improvement of 0.23, 1.62 and 0.01 dBs respectively. Though the performance of SPM, SRF and A+ algorithms in this case is consistently better than the performance of the proposed method, the PSNR and SSIM values obtained by the proposed method is still quite close to the ones obtained by these algorithms. In terms of average PSNR, we can see that the performance improvement of SPM over the proposed method is 0.27 dBs, whereas, SRF and A+ algorithms have a performance improvement of 0.41 and 0.35 dBs respectively. When compared with our proposed method with $K = 100$ Gaussian models, A+ and SRF algorithms have a performance improvement of 0.35 and 0.29 dBs respectively in terms of average PSNR.

Note that the SRCNN algorithm [105] for SISR exploits a deep convolutional neural network whose parameters are learnt by minimizing an objective function involving the global errors between HR training images and the corresponding super-resolved images. Neural networks learnt by minimizing such objective functions are capable of capturing some of the global image structures, apart from the much simpler local structures captured by the patch based learning algorithms. As SRCNN algorithm cannot be considered as a

patch based algorithm in the strict sense, performance comparisons between SRCNN and pure patch based algorithms considered in Table 3.3 and Table 3.4 do not appear to be fair. Hence, we have separately shown the comparisons between SRCNN and our proposed method in Tables 3.5 and 3.6. It can be noted that the performance of SRCNN algorithm is consistently better than the proposed method in all the images, and also better than the state of the art patch based algorithms such as A+ and SRF in most of the images. This result seems to be obvious, as the neural network exploited by SRCNN algorithm is capable of capturing more complex image structures when compared with simple patch based algorithms.

3.3.3 Comparison of Average Run time

We now compare the speed of our proposed SR reconstruction algorithm with that of various SISR algorithms included in our performance comparisons. The comparison of complexity of different SR algorithms in terms of exact number of computations required for super resolving a given LR image is practically infeasible due to the iterative nature of most of these algorithms. Hence, as it was done in [7], we experimentally compare the speed of different SR algorithms in terms of average run time taken by these algorithms when they are run under similar conditions on a PC. We used MATLAB R2012b to run different SR algorithms on a PC with a RAM of 8GB and Intel i5-3470 quad core processor running at 3.20 GHz. Table 3.7 shows the average run time of various SR algorithms in seconds per 10^4 pixels of the input LR image. As can be observed from Table 3.7, the bicubic algorithm, being a simple interpolation method without utilizing image models, has the highest speed as expected. Among the learning based algorithms, our proposed Joint GMM method with $K = 100$ Gaussian models has the least average run time for both the magnification factors $q = 2$ and 3. Figure 3.3 shows a bidimensional scatter plot visualizing the compromise between speed and SR performance. The Horizontal axis in

	Bicu-bic	ScSR	SCDL	BP JDL	SPM	SR CNN	SRF	A+	Prop K=100	Prop K=225
x2	0.001	51.74	145.81	104.49	2.57	1.95	0.385	0.356	0.252	0.397
x3	0.002	114.22	314.12	220.96	11.97	3.61	0.711	0.459	0.214	0.321

TABLE 3.7: Average Run Time in seconds per 10^4 LR pixels.

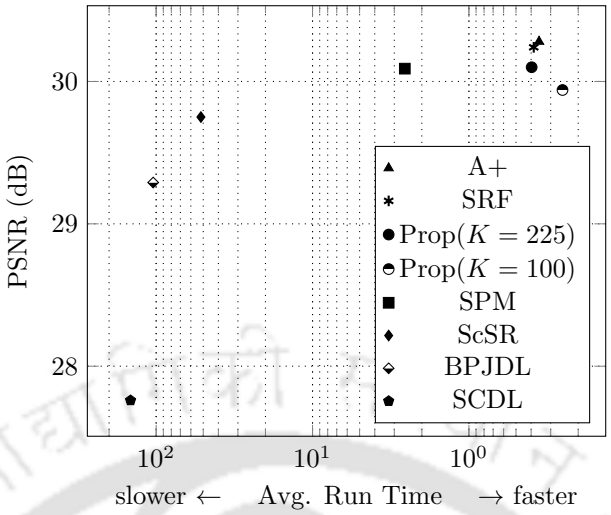


FIGURE 3.3: Comparison of speed and super resolution performance

Figure 3.3 shows the average run time computed as mentioned above, and the vertical axis shows the SR performance in terms of average PSNR value obtained for a magnification factor $q = 2$. It can be observed that the proposed Joint GMM method is more than 100 times faster than the joint dictionary based ScSR algorithm, and achieves superior performance even with $K = 100$ Gaussian models. In the case of magnification by a factor of 2, the performance of our proposed method with $K = 225$ Gaussian models is comparable to that of state of the art SR algorithms A+ and SRF. The A+ and SRF algorithms are slightly faster than the proposed method with a negligible difference of 0.041 and 0.012 seconds per 10^4 LR pixels. In the case of magnification by a factor of 3, the proposed method shows the least average run time when compared to all other methods, even with $K = 225$ Gaussian models . The proposed method with $K = 225$ Gaussian models runs approximately 6.5 times faster than SPM, and 5 times faster than SRCNN, for magnification factor 2. We can observe from Figure 3.3 that our proposed method with $K = 100$ Gaussian models is faster than A+ and SRF algorithms, but at the same time compromises the SR performance.

3.4 Conclusions

In this chapter, we have presented a new algorithm for SISR using the Joint GMM method. The Joint GMM method can be seen as a GMM analogue of the sparsity based ScSR

algorithm exploiting sparse representation of HR-LR patch pairs over joint dictionaries. We used Joint GMMs for characterizing concatenated pairs of HR and LR patches and its parameters are learnt from a large database of HR-LR patch pairs by using Expectation-Maximization (EM) algorithm. The proposed SR reconstruction algorithm is a piecewise linear filter involving K Wiener filters, where, K is the total number of Gaussian models included in the Joint GMM. Due to the piecewise linear nature, the proposed method is much faster when compared to the sparse representation based methods exploiting joint dictionaries. The SR experiments conducted on a set of natural images demonstrate that the performance our proposed method is comparable to the state of the art patch based SISR algorithms while being competitive in speed of operation.

Chapter 4

Single Image Super Resolution using GMM Based Regression Method

In this chapter, we propose a GMM based regression method for SISR. The proposed method addresses some of the drawbacks of the SISR algorithm based on the Joint GMM method proposed in chapter 3. In the Joint GMM method, the parameters of the GMM prior are learnt from concatenated HR-LR patch vectors by using the computationally heavy EM algorithm. This computational bottleneck restricts the dimension of LR patches used by the joint GMM method to a few pixels, which in turn deteriorates the SR performance, especially for large magnification factors. The proposed GMM based regression method exploits computationally cheaper regression techniques to learn the HR-LR patch correlations, and achieves superior SR performance in the case of large magnification factors. The proposed method considerably reduces the learning effort when compared with the Joint GMM method. The GMM based regression method can be interpreted as the GMM analogue of the A+ algorithm which uses a dictionary based regression method for learning the mapping from LR to HR patches.

The SISR algorithm based on the Joint GMM method proposed in chapter 3 exploits a computationally fast, piecewise linear filtering approach for estimating the HR patches

corresponding to a given set of input LR patches. However, the learning time required for estimating the Joint GMM parameters from a training database containing the long vectors obtained by concatenating HR-LR patches is very large. The EM algorithm used for estimating the Joint GMM parameters is computationally challenging when the dimension of the concatenated vectors is large. Assuming LR patches of size $\tau \times \tau$ pixels, the corresponding HR patches are of size $\tau q \times \tau q$ pixels, and the dimensions of LR, HR and the concatenated patch vectors are τ^2, $\tau^2 q^2$ and $\tau^2 \left(q^2 + 1\right)$ respectively, where, q denotes the desired magnification factor. In a typical example of SR by a magnification factor $q = 2$, the Joint GMM method used an LR patch size $\tau = 4$, resulting in concatenated vectors of dimension 80. In the case of magnification factor $q = 3$, the same LR patch size leads to concatenated vectors of dimension 160. In our experiments, we have observed that the EM algorithm (run on a Desktop PC with 8GB RAM and Intel i5 quad core processor operating at 3.20 GHz) takes approximately $48 - 50$ hours to estimate the Joint GMM parameters when $q = 3$ and $\tau = 4$.

In order to reduce the learning time required by the EM algorithm, the Joint GMM method restricted the LR patch size to $\tau = 3$, in the case of SR by a magnification factor $q = 3$. However, restricting the LR patch size to a few pixels deteriorates the SR performance. Note that the Joint GMM method estimates the HR patch corresponding to a given input LR patch by exploiting HR-LR patch correlations captured by the covariance matrices of Gaussian models from the Joint GMM prior. The smaller the size of LR patches, the lesser will be the information available in these patches, and thus the covariance matrices cannot capture the necessary HR-LR patch correlations. This deteriorates the SR performance of Joint GMM method in the case of large magnification factors such as $q = 3$. Though Joint GMM method achieves competing performance in the case of SR by a factor of 2, the results of SR experiments described in section 3.3 of chapter 3 show that the performance achieved by the Joint GMM method is inferior to that of state of the art SISR algorithms in the case of magnification factor 3. The proposed GMM based regression method adresses the computational bottleneck associated with the EM algorithm, and targets to improve the SR performance in the case of large magnification factors.

The GMM based regression method is motivated from the observation that the SR reconstruction algorithm used by the Joint GMM method can be interpreted as a two step

procedure involving a clustering step and a linear filtering step. The clustering step associates each LR patch to a Gaussian model from the Joint GMM prior. A cluster of LR patches is associated to each of the Gaussian models. The linear filtering step computes the HR patch estimates corresponding to the LR patches from a given cluster by linearly projecting the LR patches using a Wiener filter matrix. The Wiener filter is derived from the covariance matrix of the Gaussian model to which the patches from the given cluster are associated. The Wiener filters exploit HR-LR patch correlations captured by the covariance matrices to infer the HR patches from the corresponding LR patches. The linear filtering step thus assumes that all the LR patches from a given cluster are mapped to the corresponding HR patches through a common linear transformation, where, the transformation matrix is the Wiener filter associated with the cluster. Based on this observation, the proposed GMM based regression method precomputes the set of Wiener filters from a training database of HR-LR patch pairs by avoiding the computationally heavy Joint GMM learning.

The GMM based regression method partitions the entire set of LR patches from a given training database into as many clusters as the number of Gaussian models from a generic GMM prior characterizing natural image patches. Each LR patch from the training database is associated to a Gaussian model selected from the generic GMM prior, and this results in a cluster of patches associated to each of the Gaussian models. The linear transformation matrix that maps the LR patches from a cluster to the corresponding HR patches are learnt for each of the clusters by using regularized least squares regression. Instead of computing the Wiener filters by using covariance matrices of Gaussian models from the Joint GMM prior, note that the proposed GMM based regression method estimates the filters by exploiting computationally cheaper regression method.

The Clustering-Regression framework described above assumes that the mapping from LR patches to the corresponding HR patches is locally linear in the manifold of LR patches. Interestingly, many of the recently proposed patch based SISR algorithms can be interpreted to be exploiting the Clustering-Regression framework in some form. Some of the algorithms (including the proposed GMM based regression method) [107, 24, 106] explicitly utilize the Clustering-Regression framework. On the other hand, the SR algorithms such as [4, 6, 88] can be interpreted to be exploiting a similar framework, through these

algorithms do not explicitly use clustering and regression. The specific method used for clustering and regression varies among different algorithms. The simple functions (SF) method for SISR proposed in [107] exploits K-means clustering for partitioning the LR training patches into K clusters. The mapping from LR patches contained in a given cluster to the corresponding HR patches are learnt by using least squares regression or support vector regression. The SISR algorithm based on Super Resolution Forests (SRF) [106] exploits random forests for clustering, and least squares regression for learning the mapping from LR to HR patches. The A+ algorithm [24] exploits an overcomplete dictionary for partitioning the set of LR training patches into as many clusters as the total number of atoms from the dictionary, and computes the mapping from LR to HR patches by using ridge regression. The sparsity based ScSR algorithm [4], and several of its variants [6, 56, 88, 7] can also be interpreted to be exploiting the Clustering-Regression framework [106].

Though the clustering techniques used by different SISR algorithms exploiting Clustering-Regression framework are totally different, the regression method used by most of these algorithms are closely similar, and is based on least squares regression. The performance of a SISR algorithm exploiting Clustering-Regression framework thus depends heavily on the specific method used for clustering. In spite of the closely similar regression methods used by the A+ [24] and SF [107] algorithms, note that the difference in the clustering method used by these algorithms produces a significant difference in the SR performance. The GMM based regression method proposed in this work can be seen as GMM analogue of the A+ algorithm which exploits an overcomplete dictionary for partitioning the LR training patches into several clusters. Instead of the dictionary based clustering method, our proposed algorithm exploits a clustering method based on a generic GMM prior characterizing natural image patches.

The rest of this chapter is organized as follows: Section 4.1 describes the A+ algorithm to which the proposed GMM based regression method is closely related. Section 4.2 describes the proposed SR algorithm, and section 4.3 discusses the SR experiments carried out for evaluating the proposed algorithm. Section 4.4 concludes the chapter.

4.1 Related Works

In this section, we briefly review the A+ algorithm to which the proposed GMM based regression method is closely related. A detailed review of literature on SISR algorithms is given in chapter 3. The A+ algorithm exploits an overcomplete dictionary characterizing LR patches to partition the entire set of LR training patches into several clusters. The set of LR patches from a given training database is partitioned into as many clusters as the total number of atoms contained in the dictionary. A cluster of T LR training patches is associated to each of the dictionary atoms. The cluster of LR patches associated to a given dictionary atom corresponds to the set of T LR patches most correlated (under normalized inner product metric) to the given atom. The A+ algorithm assumes that all the LR patches from a given cluster are mapped to the corresponding HR patches through a common linear transformation. The transformation matrix that maps the LR patches from a given cluster to the corresponding HR patches is learnt by applying ridge regression on the set of LR training patches from the given cluster and the corresponding set of HR training patches. A given input LR patch to be magnified is first associated to the dictionary atom to which the given patch is most correlated. The estimate of the unknown HR patch is then computed by multiplying the given LR patch with the transformation matrix corresponding to the dictionary atom to which the given patch is associated.

4.2 Proposed Method

In this section, we describe the proposed GMM basbed regression method for SISR. As in the case Joint GMM method, the observed LR image $\mathbf{y}$ corresponding to an unknown HR image $\mathbf{x}$ is assumed to be obtained as,

$$\mathbf{y} = \mathbf{SHx}, \tag{4.1}$$

where, $\mathbf{S}$ denotes the matrix corresponding to the subsampling operation, and $\mathbf{H}$ denotes the blur kernel. The blur kernel $\mathbf{H}$ is assumed to be unknown and shift invariant. However,

we assume that a large database of LR and the corresponding HR images are available, in which, each LR image is obtained by applying the degradation model (4.1) on the corresponding HR image. Given the database of HR-LR pairs of images, the training database of HR-LR patch vectors is constructed by randomly sampling LR patches from the LR images, and the corresponding HR patches from the HR images. Let $\left\{\mathbf{t}_i \in \mathbb{R}^{\tau_l^2}\right\}_{i=1}^{M}$ denotes the set of M LR training vectors, where τ_l denotes the dimension of LR patches. In the Joint GMM method proposed in chapter 3, the parameters of the Joint GMM prior are learnt from the training vectors obtained by concatenating each LR patch vector $\mathbf{t}_i \in \mathbb{R}^{\tau_l^2}$ with the corresponding HR patch vector of dimension $\tau_l^2 q^2$, where, q is the desired magnification factor. Thus, the SR reconstruction algorithm of the Joint GMM method targets to predict HR patches of size $\tau_l q \times \tau_l q$ pixels from a set of input LR patches of size $\tau_l \times \tau_l$ pixels. However, predicting a large HR patch of size $\tau_l q \times \tau_l q$ pixels from a small LR patch of size $\tau_l \times \tau_l$ pixels may cause the boundary pixels of the predicted HR patch to be poorly estimated, especially for large magnification factors. Instead of predicting a large HR patch of size $\tau_l q \times \tau_l q$ pixels from an input LR patch of size $\tau_l \times \tau_l$ pixels, the proposed GMM based regression method targets to predict a smaller HR patch of size $\tau_h \times \tau_h$ pixels extracted from the center of the actual HR patch of size $\tau_l q \times \tau_l q$ pixels. The set of HR training vectors $\left\{\mathbf{r}_i \in \mathbb{R}^{\tau_h^2}\right\}_{i=1}^{M}$ corresponding to the set of M LR training vectors $\{\mathbf{t}_i\}_{i=1}^{M}$ is obtained by vectorizing the center portion of size $\tau_h \times \tau_h$ pixels of the actual HR patches of size $\tau_l q \times \tau_l q$ pixels as shown in Figure 4.1. In the following, HR patch refers to the smaller patch of size $\tau_h \times \tau_h$ pixels extracted from the center portion of a large HR patch, and HR patch vector refers to the corresponding vector representation.

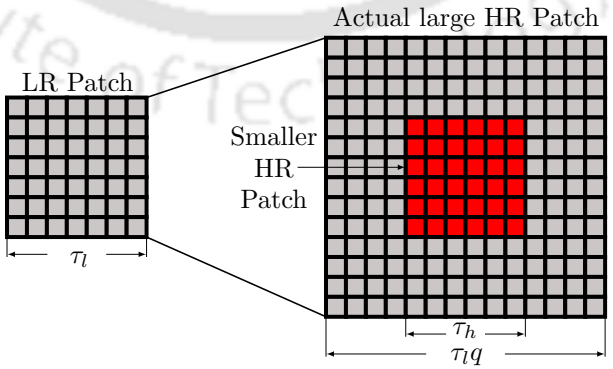


FIGURE 4.1: Extraction of a small HR patch from a larger patch

The proposed GMM based regression method exploits a generic GMM prior to partition the set of LR training patches $\{\mathbf{t}_i\}_{i=1}^{M}$ into K clusters, where, K is the total number of Gaussian models from the GMM prior. The proposed method assumes that all the LR patches from a given cluster are mapped to the corresponding HR patches through a common linear transformation, and computes the transformation matrices corresponding to each of the clusters. The GMM prior used for clustering the LR training patches is specified by the parameter set $\mathbf{\Theta} = \{\omega_k, \boldsymbol{\mu}_k, \mathbf{\Sigma}_k\}_{k=1}^{K}$, where, $\omega_k \in [0,1]$, $\boldsymbol{\mu}_k \in \mathbb{R}^{\tau_l^2}$, and $\mathbf{\Sigma}_k \in \mathbb{R}^{\tau_l^2 \times \tau_l^2}$ respectively denote the mixing weight, mean vector and covariance matrix of the k-th Gaussian model for $k = 1, 2, ..., K$. As considered in the previous works on GMM based image restoration algorithms such as PLE [11] and EPLL [18], the generic GMM prior is assumed to be capable of characterizing natural image patches. Each LR patch is assumed to be generated by a Gaussian model selected from the generic GMM prior. Though the proposed GMM based regression method exploits GMM prior for clustering LR training patches, the generic GMM prior is assumed to be capable of characterizing natural image patches regardless of the scale of the patches. In order to capture as much image details as possible, the generic GMM prior is learnt from a large database of HR image patches. Note that the collection of HR patches used for learning the GMM parameters is not the set of HR patches $\{\mathbf{r}_i\}_{i=1}^{M}$ used for computing the transformation matrices corresponding to different clusters. The estimates $\hat{\mathbf{\Theta}} = \left\{\hat{\omega}, \hat{\boldsymbol{\mu}}_k, \hat{\mathbf{\Sigma}}_k\right\}_{k=1}^{K}$ of the parameters of the generic GMM prior are learnt by using the EM algorithm [108, 109]. The EM algorithm is widely popular in machine learning and statistics literature, and is summarized in Appendix A.

The proposed GMM based regression method divides the entire set of LR training patches $\{\mathbf{t}_i\}_{i=1}^{M}$ into K clusters by associating each LR patch to a Gaussian model selected from the generic GMM prior. Each LR training patch is associated to the Gaussian model responsible for the generation of the patch. As the mixing index k_i of the Gaussian model responsible for a given LR training patch $\mathbf{t}_i$ is unknown, the estimate $\hat{k}_i$ of k_i is computed as follows. Let $I_i \in \{1, 2, ..., K\}$ denotes the random variable which takes the value k, if the i-th LR patch $\mathbf{t}_i$ is generated by the k-th Gaussian model. The unknown index k_i is estimated by maximizing the posterior probability γ_{ik} that the k-th Gaussian model is responsible for the i-th patch $\mathbf{t}_i$, over all the Gaussian models $k = 1, 2, ..., K$. The estimate

$\hat{k}_i$ of k_i is thus computed as,

$$\begin{aligned}
\hat{k}_i &= \arg\max_k \gamma_{ik} \\
&= \arg\max_k p\left(I_i = k \,|\mathbf{t}_i\right) \\
&= \arg\max_k p\left(\mathbf{t}_i \,|I_i = k\right)\, p\left(I_i = k\right) \\
&= \arg\max_k p\left(\mathbf{t}_i \,\middle|\hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\Sigma}}_k\right) \omega_k \\
&= \arg\max_k \Phi\left(\mathbf{t}_i; \hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\Sigma}}_k\right) \omega_k
\end{aligned} \tag{4.2}$$

Given $\hat{k}_i$, the i-th patch $\mathbf{t}_i$ is associated to the cluster corresponding to the Gaussian model indexed by $\hat{k}_i$ for $i = 1, 2, ..., M$. Each Gaussian model has a cluster of LR training patches associated to it. Let $\mathcal{C}_k$ be the cluster of LR patches associated to the k-th Gaussian model, i.e,

$$\mathcal{C}_k = \left\{\mathbf{t}_i \,\middle|\, \hat{k}_i = k\right\} \tag{4.3}$$

The proposed GMM based regression method assumes that all the LR patches from the cluster $\mathcal{C}_k$, $k = 1, 2, ..., K$, are mapped to the corresponding HR patches through a common linear transformation. The HR patch $\mathbf{r}_i$ corresponding to the LR patch $\mathbf{t}_i$ from a cluster $\mathcal{C}_k$ can be written as,

$$\mathbf{r}_i = \mathbf{W}_k \mathbf{t}_i,\ \forall \mathbf{t}_i \in \mathcal{C}_k,\ k = 1, 2, ..., K, \tag{4.4}$$

where, $\mathbf{W}_k$ denotes the transformation matrix corresponding to the k-th cluster (or equivalently, the transformation matrix corresponding to the k-th Gaussian model from the GMM prior). The proposed method computes the transformation matrix $\mathbf{W}_k$ corresponding to the k-th Gaussian model by applying ridge regression on the set of LR patches from $\mathcal{C}_k$, and the corresponding HR patches. The transformation matrix $\mathbf{W}_k$ is computed as,

$$\begin{aligned}
\mathbf{W}_k &= \arg\min_{\mathbf{W}} \sum_{i|\mathbf{t}_i \in \mathcal{C}_k} \|\mathbf{r}_i - \mathbf{W}\mathbf{t}_i\|_2^2 + \lambda \|\mathbf{W}\|_F^2 , \\
&= \arg\min_{\mathbf{W}} \|\mathbf{R}_k - \mathbf{W}\mathbf{T}_k\|_F^2 + \lambda \|\mathbf{W}\|_F^2 ,
\end{aligned} \tag{4.5}$$

where, $\mathbf{T}_k$ denotes the matrix obtained by stacking the LR patches from $\mathcal{C}_k$ as columns, and $\mathbf{R}_k$ denotes the matrix obtained by stacking the corresponding HR patches. The

solution of (4.5) can be obtained in closed form (detailed derivation is given in Appendix D) as,

$$\mathbf{W}_k = \mathbf{R}_k \mathbf{T}_k^T \left(\mathbf{T}_k \mathbf{T}_k^T + \lambda \mathbf{I} \right)^{-1} \tag{4.6}$$

The penalty on the Frobenius norm of the transformation matrix ensures that the matrix inversion in (4.6) is well behaved. The regression method used by the proposed algorithm is exactly similar to the regression method used by the A+ algorithm [24]. In the learning phase, the proposed method computes the transformation matrix $\mathbf{W}_k$ for $k = 1, 2, ..., K$, by using the training database of LR and HR patches.

4.2.1 SR Reconstruction

The SR reconstruction algorithm of the proposed GMM based regression method splits a given input LR image into the set of all possible overlapping LR patches, and computes the estimates of unknown HR patches corresponding to the set of LR patches. The estimated HR patches are brought back to their respective positions in the HR image, and the overlapping patches are averaged to compute the estimate of the unknown HR image. Let $\left\{ \mathbf{q}_i \in \mathbb{R}^{\tau_l^2} \right\}_{i=1}^{L}$ denotes the set of input LR patches, where, L denotes the total number of patches from the image. The proposed SR reconstruction algorithm computes the estimates $\left\{ \hat{\mathbf{p}}_i \in \mathbb{R}^{\tau_h^2} \right\}_{i=1}^{L}$ of the unknown HR patches $\left\{ \mathbf{p}_i \in \mathbb{R}^{\tau_h^2} \right\}_{i=1}^{L}$ from the set of input LR patches $\{\mathbf{q}_i\}_{i=1}^{L}$. Given the parameters $\hat{\mathbf{\Theta}} = \left\{ \hat{\omega}_k, \hat{\boldsymbol{\mu}}_k, \hat{\mathbf{\Sigma}}_k \right\}_{k=1}^{K}$ of the generic GMM prior, and the set of K transformation matrices $\{\mathbf{W}_k\}_{k=1}^{K}$, the SR reconstruction algorithm associates each LR patch to a Gaussian model selected from the GMM prior, and computes the estimate of the unknown HR patch by applying the linear transformation corresponding to the selected Gaussian model. The i-th LR patch $\mathbf{q}_i$ is associated to the Gaussian model with index $\hat{k}_i \in \{1, 2, ..., K\}$ computed as,

$$\begin{aligned} \hat{k}_i &= \arg\max_k \gamma_{ik} \\ &= \arg\max_k \Phi \left(\mathbf{q}_i; \hat{\boldsymbol{\mu}}_k, \hat{\mathbf{\Sigma}}_k \right) \hat{\omega}_k \end{aligned} \tag{4.7}$$

for all $i = 1, 2, ..., L$. The estimate $\hat{\mathbf{p}}_i$ of the unknown HR patch is computed as,

$$\hat{\mathbf{p}}_i = \mathbf{W}_{\hat{k}_i}\mathbf{q}_i,\ i = 1, 2, ..., L. \tag{4.8}$$

The estimated HR patch vectors are reshaped into patches of size $\tau_h \times \tau_h$ pixels, and the unknown HR image is estimated by averaging the overlapping patches arranged at their respective positions in the desired HR image. As in the case of Joint GMM method, the proposed GMM based regression method exploits a weighted averaging method for patch averaging. The overlapping patches are weighted by a Gaussian window of the same size as the estimated HR patches. The Gaussian window assigns highest weight to the pixel at the center of the patch, and the weight decreases towards patch boundaries.

Similar to the covariance matrices of Gaussian models used by the Joint GMM method, the transformation matrices $\{\mathbf{W}_k\}_{k=1}^{K}$ used by the proposed GMM based regression method capture the necessary HR-LR patch correlations to be exploited for estimating the unknown HR patch from a given LR patch. In contrast with the Joint GMM method which learns the GMM parameters from a training database of long concatenated HR-LR patch vectors, the proposed method exploits computationally cheaper regression techniques to learn the transformation matrices. Moreover, the proposed method works on smaller HR patches extracted from the center portion of the actual large HR patches. The smaller size of HR patches enables the proposed method to use larger size of LR patches when compared with the LR patch size used by the Joint GMM method. The large LR patches help the transformation matrices to better capture the HR-LR patch correlations, thereby improving the SR performance. In the case of Joint GMM method, the computational bottleneck associated with the EM algorithm restricts the size of LR patches to a few pixels, which deteriorates the SR perfomance, especially in the case of large magnification factors.

4.3 Experiments and Results

In this section, we describe the experimental set up used to evaluate the performance of the proposed GMM based regression method, and present the results of SR experiments

conducted on a set of natural images. The SR performance of the proposed GMM based regression method is evaluated by reusing the same experimental set up used for evaluating the performance of the Joint GMM method proposed in chapter 3. The SR experiments are conducted for two different magnification factors, $q = 2$ and 3. In all the experiments, the blur kernel $\mathbf{H}$ is assumed to be the kernel corresponding to bicubic interpolation. As in the case of Joint GMM method, the proposed GMM based regression method does not explicitly use the blur kernel in the SR reconstruction algorithm. The blur kernel is required only for synthesizing HR-LR pairs of training images used for learning the transformation matrices $\{\mathbf{W}_k\}_{k=1}^{K}$. As the the back projection step used by ScSR and BPJDL algorithms explicitly utilizes the knowledge of blur kernel, the back projection step is intentionally turned off in all the experiments.

4.3.1 Learning the Generic GMM prior and Transformation Matrices

The generic GMM prior and the set of transformation matrices $\{\mathbf{W}_k\}_{k=1}^{K}$ are learnt from the same database of 103 training images used for learning the GMM parameters in the Joint GMM method described in chapter 3. Each HR image from the database is blurred and downsampled as in (4.1) to synthesize the database of HR-LR image pairs corresponding to two different magnification factors $q = 2$ and 3. The parameters of the generic GMM prior are learnt from 10^6 HR patches randomly sampled from the set of HR images. The dimension of the HR patches used for learning the GMM prior is chosen to be 7 for both the magnification factors $q = 2$ and 3. As in the case of Joint GMM method, the number of Gaussian models K is set to 225. The estimates $\hat{\mathbf{\Theta}} = \left\{\hat{\omega}, \hat{\boldsymbol{\mu}}_k, \hat{\mathbf{\Sigma}}_k\right\}_{k=1}^{K}$ of the parameters of the generic GMM prior are learnt by using EM algorithm conventionally used in learning the GMM parameters. The EM algorithm is summarized in Appendix A.

The set of transformation matrices $\{\mathbf{W}_k\}_{k=1}^{K}$ is separately learnt for magnification factors 2 and 3 by using the training database of HR-LR patch pairs sampled from the respective database of HR-LR image pairs. The transformation matrices corresponding to a given magnification factor are learnt from 2×10^6 HR-LR patch pairs. The generic GMM prior is used to partition the set of LR training patches into K clusters, and the transformation matrix corresponding to each of the clusters is learnt by applying the regression method

given in (4.6). The value of parameter λ used for regularizing the matrix inversion in (4.6) is set to 0.001 in all experiments. The patch dimension τ_l of the LR patches is set to 7 for both the magnification factors $q = 2$ and 3. Hence, the same generic GMM prior learnt from the HR patches of dimension 7 can be used for both the magnification factors. Though the actual dimension of HR patches corresponding to the LR patches of dimension $\tau_l = 7$ is 14 and 21 respectively for magnification factors 2 and 3, our proposed GMM based regression method targets to estimate smaller HR patches extracted from center portion of the actual large HR patches. The dimension τ_h of the smaller HR patches is chosen to be 6 and 9 respectively for magnification factors 2 and 3. Learning the parameters of the proposed SR algorithm for different magnification factors corresponds to leaning different sets of transformation matrices by exploiting the same generic GMM prior. This considerably reduces the learning effort when compared with the Joint GMM method, as the GMM parameters used by the Joint GMM method must be learnt separately for different magnification factors by using the computationally heavy EM algorithm.

4.3.2 Performance Comparison

The proposed GMM based regression method is evaluated by comparing the SR performance with that of ScSR [4], SCDL [6], BPJDL [56], SPM [7], A+ [24], SRF [106], and the Joint GMM method proposed in chapter 3. As in the case of the Joint GMM method, SR experiments are conducted for two different magnification factors $q = 2$ and 3 on the set of 10 natural images shown in Figure 3.1. Each color image from the test dataset is converted from RGB color space to YCbCr, and the proposed SR algorithm is applied only on the Y channel. The chrominance channels Cb and Cr are magnified by using simple bicubic interpolation. The SR performance of various algorithms is measured in terms of PSNR and SSIM values of the Y channel. In all the experiments, the proposed GMM based regression method utilizes a Gaussian weighting function with $\sigma = 5$ for patch averaging.

Figure 4.2 shows the estimated HR images produced by various algorithms in the case of SR experiment conducted on Leopard image for magnification factor 3. A small portion of the image is magnified and shown as inset for a better visual comparison of the capabilities of various algorithms in preserving fine details of the image. As can be observed from

subfigures (c), (d) and (e) of Figure 4.2, the proposed GMM based regression method is better at preserving the fine details of the image when compared with ScSR, SCDL and BPJDL algorithms. The visual quality of the HR estimate produced by the proposed method is similar to that of state of the art patch based SISR algorithms such as A+ and SRF. As can be observed from the left portion of the black curve shown in the inset, visual quality of the textures preserved by the proposed method is slightly better when compared to that of SPM and Joint GMM method. The visual comparison between HR estimates produced by the proposed GMM based regression method and various SR algorithms in the case of different magnification factors and several images is given in Appendix E.

Table 4.1 shows the SR performance obtained by various algorithms in the case of magnification by a factor of 2. In Table 4.1, each cell corresponding to a given image and a given algorithm is vertically divided into two parts, with the upper half showing the PSNR value and lower half showing the SSIM value. In each row, the PSNR/SSIM values within a gap of 0.1 dB/0.0050 when compared with the corresponding best (highest) values are shown in boldface. As in the case of Joint GMM method, the proposed GMM based regression method significantly outperforms the sparsity based ScSR algorithm, and its modified versions SCDL and BPJDL. The proposed method achieves an average PSNR improvement of 0.24, 2.23 and 0.7 dBs respectively when compared with ScSR, SCDL and BPJDL algorithms. In terms of averages SSIM values, the proposed method shows an improvement of 0.0122, 0.0593, and 0.0241 respectively when compared with ScSR, SCDL and BPJDL algorithms. In most of the images, the performance of the proposed algorithm is comparable to that of SPM. In terms of PSNR and SSIM values, the proposed method achieves a slight improvement over SPM in the case of Grains, Louvre and Lena images, whereas, the SPM algorithm slightly outperforms the proposed method in the case of other images. The performance of the proposed GMM based regression method is highly competitive when compared with state of the art patch based SISR algorithms A+ and SRF. In terms of PSNR values, the proposed method achieves a slight improvement over the A+ algorithm in the case of Grains and Leopard images. As such, the proposed method outperforms the SRF algorithm in the case of Grains and Louvre images. In all other images, the A+ and SRF algorithms achieve slight improvement over the proposed method.

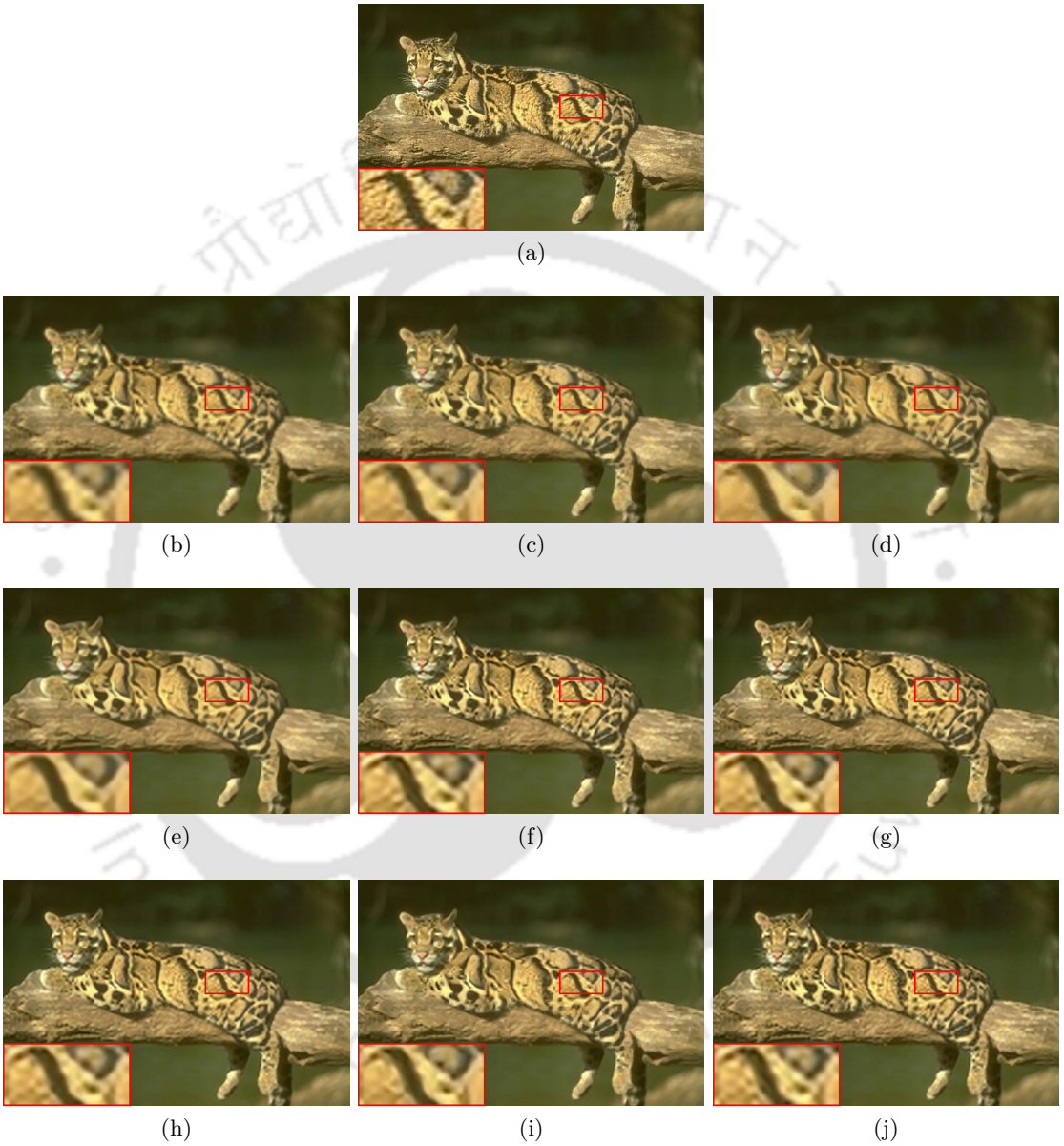


FIGURE 4.2: Results of SR experiment conducted on Leopard image for magnification factor 3. (a) Ground Truth, (b) Bicubic, (c) ScSR, (d) SCDL, (e) BPJDL, (f) SPM, (g) A+, (h) SRF, (i) Joint GMM, (j) Prop

In most of the images, the performance of GMM based regression method is superior to that of the Joint GMM method proposed in chapter 3. However, when compared with the Joint GMM method, we note that the GMM based regression method shows a significant performance drop in two isolated cases corresponding to Butterfly and Starfish images. In terms of PSNR and SSIM values, the performance drop is 0.84 dBs and 0.0082 respectively in the case of Butterfly image, and, 0.61 dBs and 0.0046 respectively in the case of Starfish image. The reason for this performance drop observed in two isolated cases is yet to be understood. Though the proposed GMM based regression method achieves superior or competitive performance in most of the images when compared with the Joint GMM method, the performance drop in the case of Butterfly and Starfish images makes the average PSNR performance of the proposed method slightly inferior to that of the Joint GMM method. In terms of average PSNR, the proposed method shows a performance drop of 0.11 dBs when compared with the Joint GMM method. However, the proposed method outperforms Joint GMM method in terms of average SSIM. The average PSNR improvement achieved by SPM, A+ and SRF algorithms is 0.1, 0.29, and 0.25 dBs respectively when compared with the proposed method. Table 4.2 shows the PSNR and SSIM values obtained by various algorithms in the case of magnification by a factor of 3. As can be observed, the proposed GMM based regression method achieves superior PSNR and SSIM performance in all the images when compared with ScSR, SCDL and BPJDL algorithms. In the case of magnification factor 3, note that the proposed method consistently outperforms the Joint GMM method in all the images. The computational bottleneck associated with the EM algorithm restricts the size of LR patches used by the Joint GMM method to a few pixels. The small size of LR patches results in deteriorated SR performance, especially in the case of large magnification factors such as $q = 3$. On the other hand, the learning phase of the proposed algorithm exploits computationally cheaper regression methods which enables the proposed algorithm to use LR patches of larger size. As can be observed from Table 4.2, the larger LR patches lead to superior SR performance when compared to that of the Joint GMM method. In terms of average PSNR and SSIM values, the proposed GMM based regression method achieves an improvement of 0.13 dBs and 0.0107 respectively when compared with the Joint GMM method. The performance of the proposed method can be observed to be highly competitive to that of state of the art SR algorithms SPM, A+ and SRF. As in the case of magnification factor 2, we note that

	Bicubic	ScSR	SCDL	BPJDL	SPM	A+	SRF	Joint GMM	Prop
Grains	23.11	23.70	22.83	23.42	**24.02**	**24.07**	**24.07**	**23.99**	**24.09**
	0.7441	0.7836	0.7275	0.7681	**0.8090**	**0.8119**	**0.8089**	0.8056	**0.8116**
Coral	26.48	26.92	25.83	26.61	**27.12**	**27.15**	**27.12**	**27.05**	**27.11**
	0.7242	0.7582	0.6862	0.7379	**0.7811**	**0.7813**	**0.7774**	0.7722	**0.7803**
Leopard	29.78	30.74	28.73	30.18	**31.21**	**31.09**	**31.15**	30.97	**31.13**
	0.9082	0.9255	0.8921	0.9170	**0.9360**	**0.9345**	**0.9344**	**0.9311**	**0.9339**
Louvre	26.29	27.14	26.02	26.97	27.40	**27.57**	27.41	**27.53**	**27.51**
	0.7860	0.8284	0.7748	0.8196	0.8474	**0.8529**	0.8457	**0.8479**	**0.8501**
Butterfly	27.41	30.85	26.94	29.64	30.97	32.02	**32.26**	31.35	30.51
	0.9155	0.9561	0.9193	0.9421	0.9564	**0.9647**	**0.9645**	**0.9617**	0.9535
Child	34.71	35.38	33.30	34.91	**35.68**	**35.68**	**35.64**	35.54	**35.64**
	0.8648	0.8824	0.8380	0.8718	**0.8897**	**0.8898**	**0.8879**	**0.8853**	**0.8888**
House	26.25	27.17	26.04	26.98	**27.61**	**27.70**	**27.62**	27.57	27.57
	0.8039	0.8365	0.7956	0.8242	**0.8553**	**0.8578**	**0.8546**	0.8520	**0.8536**
Lena	32.58	34.45	31.49	33.93	34.94	**35.26**	35.01	35.03	35.10
	0.9071	0.9303	0.8915	0.9229	**0.9376**	**0.9410**	**0.9378**	**0.9371**	**0.9385**
Parthenon	28.01	29.00	27.55	28.65	29.18	**29.42**	**29.38**	29.18	29.07
	0.8045	0.8362	0.7907	0.8239	**0.8483**	**0.8517**	**0.8498**	0.8452	0.8464
Starfish	30.09	32.20	28.96	31.67	**32.78**	**32.88**	**32.82**	**32.83**	32.22
	0.9068	0.9362	0.8867	0.9269	**0.9430**	**0.9450**	**0.9439**	**0.9432**	0.9386
Average	28.47	29.75	27.76	29.29	30.09	**30.28**	**30.24**	30.10	29.99
	0.8365	0.8673	0.8202	0.8554	**0.8803**	**0.8830**	**0.8804**	**0.8781**	**0.8795**

TABLE 4.1: PSNR and SSIM performance in the case of SR by a factor of 2

	Bicubic	ScSR	SCDL	BPJDL	SPM	A+	SRF	Joint GMM	Prop
Grains	21.24	21.40	20.97	21.40	**21.63**	**21.68**	**21.68**	21.56	**21.67**
	0.6001	0.6179	0.5782	0.6207	0.6562	**0.6610**	**0.6567**	0.6439	**0.6584**
Coral	24.94	25.11	24.28	25.10	**25.31**	**25.35**	**25.33**	25.18	**25.30**
	0.5910	0.6066	0.5492	0.6064	0.6330	**0.6397**	**0.6369**	0.6212	**0.6376**
Leopard	27.00	27.38	26.11	27.52	**27.94**	**27.95**	**27.96**	27.61	**27.92**
	0.8312	0.8427	0.8113	0.8467	**0.8643**	**0.8653**	**0.8645**	0.8551	**0.8650**
Louvre	24.54	24.81	24.12	24.89	**25.12**	**25.25**	25.13	24.96	25.11
	0.6615	0.6818	0.6365	0.6886	**0.7213**	**0.7244**	0.7166	0.7043	0.7174
Butterfly	24.06	26.02	23.87	26.58	26.84	**27.33**	27.16	26.47	25.93
	0.8230	0.8870	0.8498	0.8937	0.9018	**0.9111**	0.9035	0.9004	0.8724
Child	32.59	33.01	30.94	33.13	**33.47**	**33.57**	**33.51**	33.24	**33.52**
	0.7996	0.8109	0.7629	0.8127	0.8233	**0.8287**	**0.8264**	0.8178	**0.8278**
House	24.26	24.65	23.86	24.70	**25.00**	**25.03**	**24.96**	24.84	24.91
	0.6930	0.7134	0.6793	0.7167	0.7357	**0.7411**	**0.7373**	0.7286	0.7345
Lena	29.77	30.87	28.35	31.02	31.50	**31.74**	31.59	31.22	31.44
	0.8318	0.8542	0.8080	0.8583	0.8692	**0.8768**	**0.8725**	0.8663	**0.8721**
Parthenon	25.84	26.37	25.17	26.44	26.66	**26.84**	**26.79**	26.43	26.54
	0.6879	0.7107	0.6672	0.7134	0.7283	**0.7362**	**0.7324**	0.7192	0.7250
Starfish	26.75	27.46	25.48	27.71	**28.50**	28.31	28.31	27.82	27.91
	0.8080	0.8316	0.7794	0.8376	**0.8618**	**0.8600**	**0.8591**	0.8472	0.8485
Average	26.09	26.70	25.31	26.84	**27.20**	**27.30**	**27.24**	26.93	27.02
	0.7327	0.7556	0.7121	0.7594	**0.7794**	**0.7844**	**0.7805**	0.7704	0.7758

TABLE 4.2: PSNR and SSIM performance in the case of SR by a factor of 3

	Grains	Coral	Leopard	Louvre	Butterfly	Child	House	Lena	Parthenon	Starfish	Average
Prop	24.09	27.11	31.13	27.57	30.51	35.64	27.57	35.10	29.07	32.22	29.99
	0.8116	0.7803	0.9339	0.8501	0.9535	0.8888	0.8536	0.9385	0.8464	0.9386	0.8795
SRCNN	24.29	27.20	31.30	27.65	32.80	35.61	28.09	35.36	29.55	33.69	30.55
	0.8209	0.7832	0.9365	0.8524	0.9649	0.8889	0.8644	0.9401	0.8528	0.9496	0.8853

TABLE 4.3: Comparison with SRCNN (magnification factor 2)

	Grains	Coral	Leopard	Louvre	Butterfly	Child	House	Lena	Parthenon	Starfish	Average
Prop	21.67	25.30	27.92	25.11	25.93	33.52	24.91	31.44	26.54	27.91	27.02
	0.6584	0.6376	0.8650	0.7174	0.8724	0.8278	0.7345	0.8721	0.7250	0.8485	0.7758
SRCNN	21.76	25.38	28.13	25.20	28.06	33.50	25.32	31.84	27.02	28.94	27.51
	0.6696	0.6418	0.8683	0.7206	0.9114	0.8285	0.7488	0.8772	0.7408	0.8723	0.7879

TABLE 4.4: Comparison with SRCNN (magnification factor 3)

the proposed method shows a significant performance drop in the case of Butterfly and Starfish images when compared with SPM, A+ and SRF algorithms. In all other images, the PSNR and SSIM performance of the proposed method is very close to that of SPM, A+ and SRF algorithms. The performance gain of SPM, A+ and SRF algorithms over the proposed method is 0.18, 0.28 and 0.22 dBs respectively in terms of average PSNR, and, 0.0036, 0.0086 and 0.0047 respectively in terms of average SSIM.

As mentioned in chapter 3, the Convolutional Neural Network (CNN) utilized by the SRCNN algorithm is capable of capturing the local as well as global features of correlations between HR and LR images. In a strict sense, the SRCNN algorithm cannot be considered to be a patch based SISR algorithm such as A+, SRF or the proposed GMM based regression method. Hence, the performance comparison between SRCNN and the proposed method is separately shown in Tables 4.3 and 4.4. Table 4.3 corresponds to $q = 2$, and Table 4.4 corresponds to $q = 3$. As expected, the SRCNN algorithm achieves superior PSNR and SSIM performance in most of the images when compared with that of proposed method, as well as the state of the art patch based algorithms SPM, A+ and SRF. However, when compared with the Joint GMM method, the performance of the proposed GMM based regression method is closer to that of SRCNN algorithm.

4.3.3 Comparison of Average Run Time

The comparison of computational complexity of different SR algorithms in terms of exact number of computations required for super resolving a given LR image is practically infeasible due to the iterative nature of most of these algorithms. As in the case of Joint GMM method, we compare the speed of operation of the proposed GMM based regression method with that of various SR algorithms in terms of average run time of these algorithms, when they are run under similar conditions on a Desktop PC. We used MATLAB R2012b to run different SR algorithms on a PC with a RAM of 8GB and Intel i5-3470 quad core processor operating at 3.20 GHz. Table 4.5 shows the average run time of various SR algorithms in seconds per 10^4 pixels of the input LR image. In the case of magnification by a factor of 2, it can be observed that the average run time of the proposed algorithm is much smaller when compared to that of ScSR, SCDL and BPJDL algorithms. The

proposed algorithm is approximately 80 times faster than ScSR, and more than 100 times faster than SCDL and BPJDL algorithms. The proposed algorithm is approximately 4 times faster than SPM and 3 times faster than SRCNN. The Joint GMM method, SRF, and A+ algorithms are slightly faster when compared with the proposed GMM based regression method. However, note that the difference between the average run time of the proposed method and these algorithms are much small, and is approximately in the range of 0.25 seconds. The proposed GMM based regression method is slightly slower when compared with Joint GMM method, which is expected as the proposed method uses larger LR patches when compared with that of Joint GMM method.

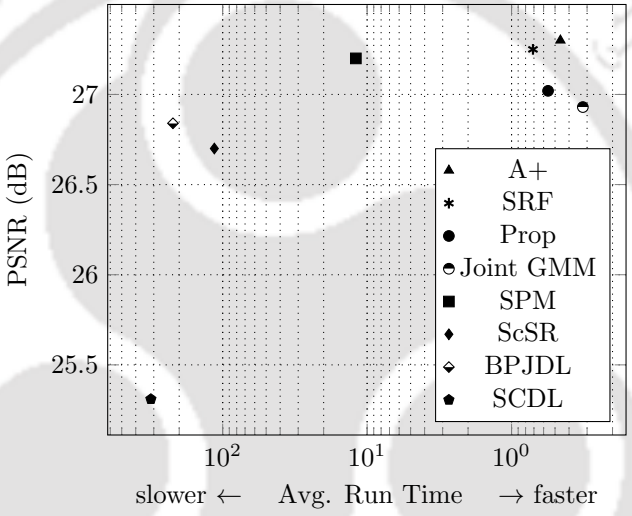


FIGURE 4.3: Comparison of Speed and Super Resolution Performance

In the case of magnification by a factor of 3, the proposed method is much faster when compared with ScSR, SCDL and BPJDL algorithms. The proposed method is more than 100 times faster than all of these algorithms. The average run time of the proposed method can be observed to be smaller than that of SPM, SRCNN and SRF algorithms, and is comparable to that of A+ algorithm. As expected, the GMM based regression method is slightly slower when compared with Joint GMM method proposed chapter 3.

	Bicu-bic	ScSR	SCDL	BP JDL	SPM	SR CNN	SRF	A+	Joint GMM	Prop
x2	0.001	51.74	145.81	104.49	2.57	1.95	0.385	0.356	0.397	0.645
x3	0.002	114.22	314.12	220.96	11.97	3.61	0.711	0.459	0.321	0.559

TABLE 4.5: Average Run Time in seconds per 10^4 LR pixels.

However, note that the learning phase of the GMM based regression method is much faster when compared with that of Joint GMM method. Moreover, the proposed GMM based regression method achieves superior performance when compared with Joint GMM method, especially for large magnification factors. The bidimensional scatter plot shown in Figure 4.3 depicts the compromise between speed and SR performance. The Horizontal axis in Figure 3.3 shows the average run time computed as mentioned above, and the vertical axis shows the SR performance in terms of average PSNR values corresponding to the case of magnification factor 3.

4.4 Conclusions

In this chapter, we proposed a GMM based regression method for SISR. The proposed algorithm addresses the computational bottleneck of the Joint GMM method introduced in chapter 3. The proposed method assumes that the manifold of LR patches can be partitioned into several clusters, and all the LR patches from a given cluster are mapped to the corresponding HR patches through a linear transformation. The GMM based regression method exploits a generic GMM prior characterizing natural image patches to partition a given set of LR training patches into as many clusters as the number of Gaussian models from the GMM prior. The transformation matrix that maps the LR patches from a given cluster to the corresponding HR patches are learnt for each of the clusters by using a computationally cheap regression method. The proposed method considerably reduces the learning effort and achieves superior performance when compared with Joint GMM method, especially in the case of large magnification factors. The proposed method achieves competitive performance when compared with state of the art SISR algorithms such as A+ and SRF.

Chapter 5

Joint Color Space GMMs for Color Image Restoration

In this chapter, we propose a Joint Color Space GMM (JCS-GMM) method for addressing color image restoration problems. The proposed JCS-GMM method exploits a GMM prior for jointly characterizing the patches from different color channels of a color image. In the case of color images, a given patch from one of the color channels has the corresponding patches in all the other color channels. The corresponding patches from different color channels of a color image are typically observed to be highly correlated. The JCS-GMM prior characterizes the vectors obtained by concatenating the corresponding patch vectors from Red, Green and Blue channels of a color image. The parameters of the JCS-GMM prior are learnt from a large database of concatenated color patches. The covariance matrices of Gaussian models from the JCS-GMM prior capture the strong inter channel correlations typically observed in color image patches. Though the naive and intuitive approach in color image restoration problems might be to interpret different color channels of a color image as independent monochrome images and separately restore them by applying monochrome image restoration techniques, such a method does not exploit the strong inter channel correlations observed in color images. The proposed JCS-GMM method targets to exploit these inter channel correlations to improve the color image restoration performance when compared with the performance obtained by independently restoring different color

channels of a degraded color image. We propose JCS-GMM denoising and JCS-GMM demosaicking algorithms for addressing color image denoising and demosaicking problems by using JCS-GMM prior. The color image denoising experiments conducted on a set of natural color images demonstrate that the proposed JCS-GMM denoising algorithm achieves superior denoising performance when compared with state of the art color image denoising algorithms. The performance of the proposed JCS-GMM demosaicking algorithm is evaluated on Kodak and IMAX datasets which are commonly used for evaluating the performance of demosaicking algorithms. The proposed JCS-GMM demosaicking algorithm achieves superior performance in the case of Kodak dataset, and competing performance in the case IMAX dataset, when compared with state of the art demosaicking algorithms.

Color images are the most common examples of a much broader class of images known as multispectral images. The pixels of a natural color image encode the response of image sensors exposed to different spectral components of visible light. The spectral components corresponding to the primary colors of vision, i.e., the Red, Green and Blue wavelengths of visible spectrum are commonly used for acquiring color images. A color image can be interpreted as a 3D stack of 2D monochrome images, in which, each 2D monochrome image corresponds to a given spectral component. The spectral components of a color image are also known as color channels. The Red, Green and Blue channels of a natural color image are typically observed to be highly correlated. Given a degraded color image, a naive and intuitive strategy to estimate the corresponding unknown clean color image is to restore each of the color channels of the given degraded image by using a monochrome image restoration algorithm. However, such a method does not exploit the strong inter channel correlations commonly observed in color images. In color image restoration problems, exploiting inter channel correlations can significantly improve the restoration performance when compared with the performance obtained by independently restoring different color channels. Many of the image restoration algorithms [10, 110, 111], originally proposed for monochrome image restoration problems, were later extended [25, 112, 113] to the case of color images by exploiting inter channel correlations in some form.

Patch based image restoration algorithms exploiting sparse representation of natural image patches over redundant dictionaries is a well studied topic in the image processing literature [10, 114, 4, 17]. Sparsity based algorithms have shown to achieve impressive

results in various image restoration problems such as image denoising [10], super resolution [4] and de-blurring [17]. The KSVD denoising algorithm [10] proposed for addressing monochrome image denoising problem is one of the widely studied sparsity based algorithms. The Color-KSVD algorithm proposed in [25] has extended the original KSVD image denoising algorithm into a color image restoration framework. The Color-KSVD algorithm exploits inter channel correlations in color image restoration problems by jointly characterizing the corresponding patches from different color channels of a color image. In color image restoration problems such as color image denoising and demosaicking, the Color-KSVD algorithm achieves state of the art results by exploiting sparse representation of concatenated color patches over redundant dictionaries. The recent works [11, 18] on image restoration algorithms exploiting GMM priors for characterizing natural image patches have close connections [11, 28] with the sparsity based image restoration algorithms. The work proposed in [11] has revealed that there exist GMM analogues of many of the sparsity based image restoration algorithms. The GMM analogues are computationally simpler when compared with sparsity based algorithms, yet offer superior restoration performance. Though previous works have proposed sparsity based techniques for addressing color image restoration problems, the previous works on GMM based image restoration algorithms were mainly focused on monochrome image restoration problems. Motivated by the close relations between GMM and sparsity based algorithms, it is natural to ask the following questions:

- What will be the GMM analogue of the sparsity based color image restoration techniques proposed in the Color-KSVD algorithm ?
- How does the performance of such a GMM analogue compare with that of sparsity based techniques and other state of the art color image restoration algorithms ?

In this work, we attempt to answer these questions by proposing a color image restoration method based on Joint Color Space-GMM (JCS-GMM) prior. Analogous to the sparsity prior used by the Color-KSVD algorithm, the JCS-GMM prior exploits inter channel correlations in color image restoration problems by jointly characterizing the corresponding patches from Red, Green and Blue channels of a color image.

The rest of this chapter is organized as follows: Section 5.1 briefly reviews the previous work on color image restoration algorithms, and Section 5.2 describes the JCS-GMM prior used for characterizing concatenated color patches. Section 5.3 gives a detailed description of the proposed JCS-GMM denoising and JCS-GMM demosaicking algorithms, and, Section 5.4 describes the experiments carried out for evaluating the performance of the proposed algorithms. Section 5.5 concludes the chapter.

5.1 Review of Prior Art

In this section, we briefly review the previous works on different color image restoration algorithms. As our proposed JCS-GMM method address color image denoising and demosaicking problems, this review is focused mainly towards the color image restoration algorithms addressing denoising and demosaicking problems. The most intuitive strategy that can be used by a color image restoration algorithm might be to independently restore each of the color channels of a given degraded color image by using an appropriate monochrome image restoration algorithm. However, such a strategy is certainly not the best choice, as it is possible to significantly improve the restoration performance using carefully designed techniques for exploiting the strong inter channel correlations observed in color images. Different approaches for exploiting inter channel correlations in color image restoration problems can be broadly classified into two [113].

In the first approach, prior knowledge about the inter channel dependencies is exploited to define an appropriate color space in such a way that the new color space decorrelates the correlated components of an RGB color image. A given degraded image is restored by first transforming to the newly selected color space and then applying a monochrome image restoration algorithm on each of the color channels of the transformed image. The color image denoising algorithm proposed in [115] first transforms a given image into chromaticity-brightness space and then process each of these components using diffusion techniques based on partial differential equations or diffusion flows adapted from the theory of harmonic maps in liquid crystals. In [116], chromaticity-brightness space and Hue-Saturation-Value (HSV) space have been investigated for color image denoising using image restoration techniques based on Total Variation (TV) minimization. HSV space was opted

in [117] to address color image denoising problem by utilizing a geometrical continuity assumption on the hue component, while using independent anisotropic diffusion methods for filtering the noise in the other two components, i.e., saturation and value. An optimal luminance/color-difference space projection was used in [118] for color image denoising based on wavelet shrinkage.

In the literature, several demosaicking algorithms [119, 120, 121, 122, 123] have been proposed to operate on the color space of *Primary Difference Signals (PDS)* [119], also known as color difference components. These algorithms are based on the observation that, strong inter channel correlations make the color difference components much smoother and easier to interpolate when compared with the original mosaicked color channels. Color difference based algorithms first compute the estimates of color difference components $\Delta_{g,r} = G - R$ and $\Delta_{g,b} = G - B$, and add them with the available pixel values (R or B) to interpolate the Green channel. The Red and Blue channels are interpolated by estimating the missing values of color difference components using simple interpolation techniques such as bilinear interpolation, and subtracting them from the interpolated Green channel. The Directional LMMSE (DLMMSE) algorithm proposed in [119] first interpolates the Green channel by fusing LMMSE estimates of color difference components computed along horizontal and vertical directions and adding with the available pixel values. The Red and Blue channels are then estimated by interpolating the respective color difference components using bilinear interpolation and subtracting from the Green channel. The Gradient Based Threshold Free (GBTF) algorithm [120] exploits color difference gradients to compute the weights used for fusing the color difference components estimated along horizontal and vertical directions. Multiscale Gradients (MSG) based algorithm proposed in [121] used the color difference gradients computed at different scales for adaptively fusing the color difference components along horizontal and vertical directions. The Red and Blue channels are estimated by applying bilinear interpolation algorithm on the color difference components.

A slightly different approach which exploits the interpolation of residual between a color channel and its tentative estimate has produced several important demosaicking algorithms [124, 125, 126, 127, 128] offering state of the art performance. The residual interpolation based methods exploit the observation that the residual between a color channel and its tentative estimate is a much smoother signal when compared with the color difference

components, if the tentative estimate is close enough to the ground truth. The Residual Interpolation (RI) algorithm proposed in [124] utilizes GBTF algorithm [120] for estimating the Green channel which is used as a guide image for computing the tentative estimates of Red and Blue channels using a guided filter [129]. The residual values between Red/Blue channels and their tentative estimates at the available pixel locations of Red/Blue channels are interpolated by using bilinear interpolation, and added with tentative estimates to compute the demosaicked Red and Blue channels. The Minimized Laplacian Residual Interpolation (MLRI) algorithm, [126] and its improved version MLRI+wei proposed in [127], compute the tentative estimates of Red/Blue channels by using a guided filter which minimizes the Laplacian energies of the residuals. The Iterative Residual Interpolation (IRI) algorithm [128] improved the estimation of Green channel by iteratively applying RI on each of the color channels. The estimate of Green channel obtained in the final iteration is used as a guide image for computing the tentative estimates of Red and Blue channels, which are subsequently interpolated through a single iteration of the original RI algorithm. The Adaptive Residual Interpolation (ARI) algorithm proposed in [125] adaptively selects the number of iterations to be performed on each of the pixels, and fuses the pixel estimates obtained by RI and MLRI algorithms.

The second approach for exploiting inter channel correlations in color image restoration problems is to explicitly incorporate the correlations in the restoration framework. In the following part of this review, we focus only on this approach as our proposed JCS-GMM method belongs to this category. Based on the Field of Experts (FoE) method [130] for monochrome image denoising, a color image denoising algorithm was proposed in [131] by using a Markov Random Field (MRF) model for color images. The MRF model is learnt from a large database of noiseless color images. The celebrated BM3D algorithm [110] for monochrome image denoising was extended in [112] to the corresponding color version named Color BM3D (C-BM3D). The C-BM3D algorithm exploits a grouping constraint for selecting similar image blocks from different color channels, which are subsequently denoised through collaborative filtering. In C-BM3D algorithm which operates on YU-V/opponent color spaces, patch grouping is carried out only on the Y channel, and the same grouping is reused in the chrominance channels by assuming invariance of patch grouping due to the correlations between luminance and chrominance channels. In the

Non-Local Bayesian (NL-Bayes) image denoising algorithm [20] which operates on the $Y_0U_0V_0$ space introduced in [132], patch grouping carried out on the Y_0 channel is reused on U_0 and V_0 channels. In [133], an alternating projection method based on POCS (Projection Onto Convex Sets) was proposed for color image denoising by exploiting inter channel dependencies through imposing a spatially adaptive smoothness constraint on the color differences of each of the pixels from different color channels. The SURE-LET algorithm [111] for monochrome image denoising using wavelet shrinkage was extended to a color image denoising algorithm in [113] by utilizing a linear combination of pointwise thresholding functions optimized using Stein's Unbiased Risk Estimator (SURE).

A unified framework for color image denoising, deblurring and upsampling was proposed in [134] by imposing a color constraint based on local color statistics derived from the given degraded color image. In the case of widely studied Non Local Means (NLM) image denoising algorithm [8, 135], the computation of weights used for the non local averaging of pixels from respective color channels exploits inter channel correlations through treating color image pixels as three dimensional vectors. Color image denoising algorithms proposed in [136] and [137] by extending the NLM algorithm exploit inter channel correlations as additional prior constraints in the penalty function corresponding to the NLM algorithm. The HOSVD algorithm [138] computes the denoised color image patches by applying higher order singular value decomposition on 4D stacks of color image patches. Similar to the well studied wavelet shrinkage algorithm for natural image denoising, Discriminatively Learned Iterative Shrinkage (DLIS) [139] algorithm for color image denoising computes the denoised image by alternately iterating between shrinkage of patch groups and aggregation of denoised patches. The transform basis and shrinkage functions are learnt discriminatively from a collection of pairs of noiseless and the corresponding noisy images. An improved geometric diffusion scheme for color image denoising was proposed in [140] by incorporating the edges detected from different color channels into the diffusion process, aiming to prevent color smearing artifacts.

In the case of demosaicking algorithms, the Primary Consistent Soft Decision (PCSD) method proposed in [141] exploits inter channel correlations by constraining to use same direction of interpolation for the corresponding pixels from Red, Green and Blue channels. The SSCAD algorithm [142] exploits non local self similarity of image patches for filtering

the initial demosaicked estimate obtained by fusing the images interpolated along four different directions. The interpolation of an unknown pixel is adapted to the degree of inter channel correlations by regularizing the interpolation process using chromaticity gradients computed on a local neighborhood. The Directional Difference Regression (DDR) and Fused Regression (FR) methods [143] exploit the Efficient Regression Priors (ERP) originally introduced in a state of the art single image super resolution algorithm named A+ [24]. The AVSC algorithm proposed in [144] adaptively combines the intraband estimates computed without considering inter channel correlations, and the estimate obtained by applying any demosaicking algorithm which exploits spectral correlations. The Alternating Projection (AP) method proposed in [145] computes the demosaicked image by iteratively projecting the intermediate estimates onto the constraint sets defined using observed color samples and prior knowledge about the spectral correlations.

The color image restoration method proposed in our work can be seen as the GMM analogue of sparsity based color image restoration framework (Color-KSVD) [25] derived from the widely studied KSVD denoising algorithm [10]. In color image restoration problems such as denoising and demosaicking, Color-KSVD algorithm [25] exploits inter channel correlations by using sparse representations for characterizing concatenated color patches from Red, Green and Blue channels of a color image. The original KSVD denoising algorithm [10], and also the Color-KSVD algorithm [25], were improved later in [13] by proposing multi scale versions of these algorithms. The Quaternion KSVD (Q-KSVD) algorithm proposed in [57] extends KSVD denoising algorithm into color image restoration problems by representing the three values of a color image pixel using a quaternion. In contrast with sparsity based algorithms, the potential of GMM prior in exploiting inter channel correlations in color image denoising has not been previously studied in the literature. In the PLE algorithm, the authors have mentioned to use GMM priors for concatenated color image patches as a general framework for color image restoration problems, but have not studied the potential of such a GMM prior in exploiting inter channel correlations in color image denoising. The performance of the PLE algorithm in the case of monochrome or color image denoising problems have not been reported in [11]. Morever, the PLE algorithm suffers from the drawback that it does not exploit the power of external databases.

5.2 Joint Color Space GMMs (JCS-GMM)

In this section, we describe the Joint Color Space GMM (JCS-GMM) prior used for characterizing color image patches. We denote a clean color image by a vector $\mathbf{x} = \left[\mathbf{x}_R^T, \mathbf{x}_G^T, \mathbf{x}_B^T\right]^T \in \mathbb{R}^{3N}$, where, N is the total number of pixels in any given color channel, and, $\mathbf{x}_R, \mathbf{x}_G, \mathbf{x}_B \in \mathbb{R}^N$ respectively denote the vector representations obtained by lexicographically ordering the Red, Green and Blue channels. The vector representations of the i-th patch from Red, Green and Blue channels of the color image $\mathbf{x}$ are respectively denoted as $\mathbf{p}_{iR}, \mathbf{p}_{iG}, \mathbf{p}_{iB} \in \mathbb{R}^{\tau^2}$, where, each patch is of size $\tau \times \tau$ pixels. Note that, each of these patches is centered at the same pixel location in different color channels. Given $\mathbf{p}_{iR}, \mathbf{p}_{iG}$ and $\mathbf{p}_{iB}$, the i-th concatenated color patch $\mathbf{p}_i \in \mathbb{R}^{3\tau^2}$ is defined as,

$$\mathbf{p}_i = \begin{bmatrix} \mathbf{p}_{iR} \\ \mathbf{p}_{iG} \\ \mathbf{p}_{iB} \end{bmatrix}. \tag{5.1}$$

The basic idea behind the JCS-GMM prior is to characterize the concatenated color patches $\{\mathbf{p}_i\}$, by using a multivariate Gaussian Mixture Model (GMM) with parameter set $\mathbf{\Theta} = \{\omega_k, \boldsymbol{\mu}_k, \mathbf{\Sigma}_k\}_{k=1}^K$, where, ω_k denotes the mixture weight, $\boldsymbol{\mu}_k \in \mathbb{R}^{3\tau^2}$ denotes the mean vector, and, $\mathbf{\Sigma}_k \in \mathbb{R}^{3\tau^2 \times 3\tau^2}$ denotes the covariance matrix, of the k-th Gaussian model, and, K denotes the total number of Gaussian models. Each concatenated color patch $\mathbf{p}_i$ is assumed to be distributed according to a Gaussian model with parameters $\left(\boldsymbol{\mu}_{k_i}, \mathbf{\Sigma}_{k_i}\right)$ selected from the mixture. The index $k_i \in [1, 2, \ldots, K]$ of the Gaussian model responsible for the i-th concatenated color patch $\mathbf{p}_i$ is assumed to be unknown for all the patches. The mixture weight ω_k associated with the k-th Gaussian model represents the a priori probability that the Gaussian model is responsible for a randomly selected concatenated patch vector. The probability density function (pdf) of a randomly selected concatenated color patch $\mathbf{w}$ can be written as,

$$f\left(\mathbf{w}\right) = \sum_{k=1}^{K} \omega_k \Phi\left(\mathbf{w}; \boldsymbol{\mu}_k, \mathbf{\Sigma}_k\right), \tag{5.2}$$

where, $\Phi\left(., \boldsymbol{\mu}, \mathbf{\Sigma}\right)$ denotes the Gaussian pdf with mean $\boldsymbol{\mu}$ and covariance matrix $\mathbf{\Sigma}$.

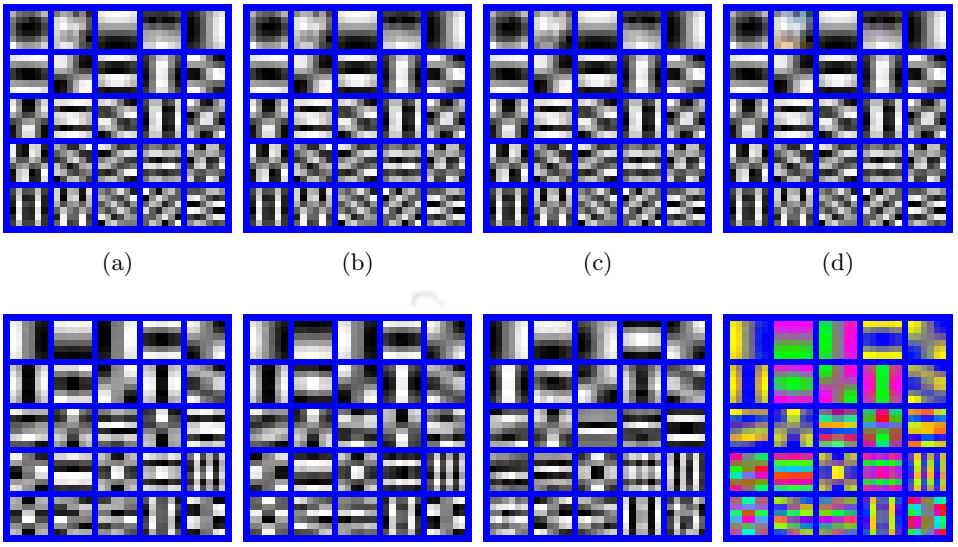


FIGURE 5.1: Eigen patches of a Gaussian model selected from the JCS-GMM prior. Top row - first (principal) 25 eigen patches, Bottom row - eigen patches with index $40-64$, (a)-(c) and (e)-(g) - Red, Green and Blue components, (d) and (h) - color versions of eigen patches.

The JCS-GMM prior jointly characterizes the corresponding patches from different color channels. The covariance matrices of Gaussian models from the JCS-GMM prior capture the inter channel correlations between these patches. Specifically, mean and covariance matrix of the k-th Gaussian model selected from the mixture can be interpreted as,

$$\boldsymbol{\mu}_k = \begin{bmatrix} \boldsymbol{\mu}_{R_k} \\ \boldsymbol{\mu}_{G_k} \\ \boldsymbol{\mu}_{B_k} \end{bmatrix}, \quad \boldsymbol{\Sigma}_k = \begin{bmatrix} \boldsymbol{\Sigma}_{R_k} & \boldsymbol{\Sigma}_{RG_k} & \boldsymbol{\Sigma}_{RB_k} \\ \boldsymbol{\Sigma}_{RG_k}^T & \boldsymbol{\Sigma}_{G_k} & \boldsymbol{\Sigma}_{GB_k} \\ \boldsymbol{\Sigma}_{RB_k}^T & \boldsymbol{\Sigma}_{GB_k}^T & \boldsymbol{\Sigma}_{B_k} \end{bmatrix}, \tag{5.3}$$

where, $\boldsymbol{\mu}_{R_k}, \boldsymbol{\mu}_{G_k}, \boldsymbol{\mu}_{B_k}$ and $\boldsymbol{\Sigma}_{R_k}, \boldsymbol{\Sigma}_{G_k}, \boldsymbol{\Sigma}_{B_k}$ respectively represent the mean vectors and covariance matrices characterizing the individual patches from Red, Green and Blue channels of a color image, and, the cross covariance matrices $\boldsymbol{\Sigma}_{RG_k}, \boldsymbol{\Sigma}_{RB_k}$ and $\boldsymbol{\Sigma}_{GB_k}$ characterize the inter channel correlations between the patches from Red-Green, Red-Blue, and Blue-Green pairs of color channels. The parameters $\boldsymbol{\Theta} = \{\omega_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K$ of the JCS-GMM prior are learnt from a large database of concatenated color patches randomly sampled from a set of clean natural color images. In our work, JCS-GMM parameters are learnt by using the Expectation Maximization (EM) algorithm [108, 109] conventionally used in

learning the GMM parameters. The EM algorithm exploits an iterative method to compute an approximate solution $\hat{\boldsymbol{\Theta}} = \left\{\hat{\omega}_k, \hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\Sigma}}_k\right\}_{k=1}^{K}$ of the ML estimation problem which estimates the unknown GMM parameters by maximizing the likelihood of concatenated color patches from the training database. The EM Algorithm is summarized in Appendix A.

Let us try to interpret the image structures captured by the covariance matrices of Gaussian models learnt by applying EM algorithm on a training database of clean concatenated color patches. We learnt a JCS-GMM prior containing $K = 150$ Gaussian models by using a training database of 10^6 concatenated color patches sampled from color images contained in the Berkeley segmentation database. The patch size τ is selected as 6, resulting in covariance matrices of dimension 108×108 ($3\tau^2$). Figure 5.1 depicts the graphical representation of a subset of eigenvectors of the covariance matrix corresponding to a Gaussian model selected from the JCS-GMM prior. The eigenvectors of dimension 108×1 are split into three 36×1 vectors corresponding to the Red, Green and Blue components, and reshaped into patches of size 6×6. The top row in Figure 5.1 corresponds to the first (principal) 25 eigenvectors and the bottom row corresponds to the eigenvectors with index 40-64. The subfigures (a)-(c) in the top row respectively show the Red, Green and Blue components of the first set of 25 eigenvectors, and subfigures (e)-(g) similarly correspond to the second set of eigenvectors (with index 40-64). In each subfigure, the respective components (Red/Green/Blue) of 25 eigenvectors are shown as a 5×5 array. We use the name *eigen patches* to denote the patches extracted from the eigenvectors. The subfigures (d) and (g) show the color versions of eigen patches obtained by stacking the Red, Green and Blue eigen patches as different color channels.

The following observations help us to interpret the image structures captured by the eigen patches shown in Figure 5.1.

- Eigen patches capture different low level image features such as edges of different orientations, corners, textures etc. A similar observation has been made by the sparsity based algorithms [10, 25] while exploring the image structures captured by the dictionary atoms learnt from natural image patches. The principal eigen patches capture the important and basic structures such as simple edges, whereas,

eigen patches corresponding to smaller eigenvalues (of the covariance matrix) capture more complex structures or color information.

- Eigen patches capture inter channel correlations - A visual comparison of the structures captured by the corresponding eigen patches in subfigures (a), (b) and (c) or (e), (f) and (g) reveals that the eigen patches capture the strong inter channel correlations observed in color images.

- Eigen patches capture monochrome as well as color information. It can be observed that the leading eigen patches shown in subfigure (d) appear as monochrome image patches. On the other hand, eigen patches associated with smaller eigenvalues (shown in subfigure (h)) capture more color information. The basic reason behind this observation is the abundance of monochrome image features such as luminance edges and textures which bias the learning algorithm towards these features. In the color-KSVD algorithm [25] exploiting sparse representation of concatenated color patches, the dictionary atoms learnt from a database of concatenated natural color image patches were observed to exhibit a similar bias producing a color wash out effect in the restored color images. This color wash out effect was resolved by modifying the inner product metric used by the OMP (Orthogonal Matching Pursuit) algorithm employed in the sparse coding step of the color-KSVD algorithm. In the case of JCS-GMM prior, eigen patches capture monochrome as well as color information, and their relative weights are embedded in the eigenvalues. This enables the proposed JCS-GMM method to achieve superior performance in color image restoration problems when compared with the sparsity based Color-KSVD algorithm.

The potential in capturing monochrome as well as color features of natural images makes JCS-GMM prior an appropriate model for characterizing color image patches. In color image restoration problems, inter channel correlations captured by the components $\mathbf{\Sigma}_{RG_k}$, $\mathbf{\Sigma}_{RB_k}$ and $\mathbf{\Sigma}_{GB_k}$ of the covariance matrix $\mathbf{\Sigma}_k$ are particularly important as these correlations can be exploited to improve the restoration performance. We use the name Joint Color Space GMM (JCS-GMM) to denote the GMM prior described above, as it jointly models the individual patches from different color channels as well as the correlations between these patches. In the single image super resolution algorithm proposed in [146], the authors have

demonstrated the potential of such a joint GMM approach in capturing the correlations between pairs of high resolution and low resolution image patches.

5.3 Color Image Restoration using JCS-GMMs

In this section, we use JCS-GMM prior to propose JCS-GMM denoising and JCS-GMM demosaicking algorithms for color image denoising and demosaicking problems. Consider a degraded and/or noisy color image $\mathbf{y} = \left[\mathbf{y}_R^T, \mathbf{y}_G^T, \mathbf{y}_B^T\right]^T \in \mathbb{R}^{3N}$ assumed to be generated from an unknown clean color image $\mathbf{x}$ as,

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n}, \tag{5.4}$$

where, $\mathbf{A} \in \mathbb{R}^{3N \times 3N}$ is the degradation operator, and $\mathbf{n} \in \mathbb{R}^{3N}$ is the additive white Gaussian noise vector. The noise vector $\mathbf{n}$ is assumed to be distributed as $\mathcal{N}\left(\mathbf{0}, \sigma^2\mathbf{I}\right)$, where, σ^2 is the noise variance. The degraded and/or noisy concatenated color patches corresponding to the clean unknown concatenated patches $\mathbf{p}_i, i = 1, 2, ..., L$ are denoted as $\mathbf{q}_i, i = 1, 2, ..., L$, where, L is the total number of patches in the image. Given $\mathbf{y}, \mathbf{A}$ and σ^2, the color image restoration problem targets to compute the estimate $\hat{\mathbf{x}} = \left[\hat{\mathbf{x}}_R^T, \hat{\mathbf{x}}_G^T, \hat{\mathbf{x}}_B^T\right]^T$ of the unknown clean color image $\mathbf{x}$. Patch based algorithms compute the restored estimates $\hat{\mathbf{p}}_i = \left[\hat{\mathbf{p}}_{iR}^T, \hat{\mathbf{p}}_{iG}^T, \hat{\mathbf{p}}_{iB}^T\right]^T$ of all the unknown concatenated color patches $\mathbf{p}_i, i = 1, 2, ...L$, extract different color components $\hat{\mathbf{p}}_{iR}, \hat{\mathbf{p}}_{iG}$ and $\hat{\mathbf{p}}_{iB}$ from all the restored concatenated patches, and average all the overlapping patches from respective color channels to compute the restored color channels $\hat{\mathbf{x}}_c$, $c = R, G, B$.

In our proposed JCS-GMM method, the estimates $\{\hat{\mathbf{p}}_i\}_{i=1}^L$ are computed by using a MAP restoration framework which utilizes JCS-GMMs as a prior model for characterizing concatenated color patches. The JCS-GMM method combines the advantages of previously studied EPLL [18] and PLE [11] algorithms exploiting GMM prior in different image restoration problems. The EPLL algorithm [18] exploits the power of external databases, but it does not adapt the externally learnt GMM prior to the image being denoised. In natural image restoration problems, adapting a patch prior to the specific image being restored has often produced impressive results [10, 25]. On the other hand, PLE algorithm

[11] adapts the GMM prior to the specific image being restored, but it does not utilize the power of external databases. Starting from a synthetic patch based initialization scheme [11], PLE algorithm iteratively learns the GMM prior from the degraded input image itself. A naive strategy for combining the advantages of external GMM prior and GMM parameter adaptation is to utilize the JCS-GMM prior learnt from an external database as the initial GMM prior used by the PLE algorithm. However, the iterative GMM parameter update mechanism used by the PLE algorithm completely destroys the externally learnt GMM prior immediately after the first iteration, and thus results only in a negligible improvement in the restoration performance. The proposed JCS-GMM method exploits an iterative MAP restoration framework which starts with the externally learnt JCS-GMM prior, and iteratively adapts the GMM parameters to the specific image being restored. In every iteration, GMM parameters are updated through a controlled GMM adaptation scheme.

5.3.1 JCS-GMM Denoising

In the case of denoising, the degradation operator $\mathbf{A} = \mathbf{I}$, and the observed color image $\mathbf{y}$ corresponding to the unknown clean image $\mathbf{x}$ is assumed to be corrupted by AWGN with noise variance σ^2. Given a noisy color image $\mathbf{y}$, the JCS-GMM denoising algorithm follows the EPLL denoising framework [18] to compute the denoised estimate $\hat{\mathbf{x}}$ as,

$$\hat{\mathbf{x}} = \underset{\mathbf{u}}{\operatorname{argmin}} \frac{\lambda}{2} \|\mathbf{u} - \mathbf{y}\|_2^2 - \sum_{i=1}^{L} \log f\left(\mathbf{R}_i \mathbf{u}\right), \tag{5.5}$$

where, $\mathbf{R}_i$ denotes the matrix which extracts the i-th concatenated color patch from the color image $\mathbf{u}$, $f(.)$ denotes the pdf corresponding to the JCS-GMM prior as given in (5.2), and, L denotes the total number of patches. The first term in (5.5) corresponds to the likelihood function with λ being the associated weight parameter, and, the second term corresponds to the expected log likelihood of concatenated color patches. As in the case of EPLL algorithm, minimization of (5.5) is carried out by using the Half Quadratic

Splitting method which targets to minimize the cost function,

$$c_\beta\left(\mathbf{u},\{\mathbf{z}_i\}\right)=\frac{\lambda}{2}\left\|\mathbf{u}-\mathbf{y}\right\|_2^2+\sum_{i=1}^{L}\frac{\beta}{2}\left\|\mathbf{R}_i\mathbf{u}-\mathbf{z}_i\right\|_2^2-\log f\left(\mathbf{z}_i\right), \tag{5.6}$$

for $\beta>0$, where, $\{\mathbf{z}_i\}_{i=1}^{L}$ is a set of auxiliary variables. As $\beta\to\infty$, minimization of c_β is identical to the minimization of the cost function in (5.5), and the auxiliary variables $\{\mathbf{z}_i\}$ converge to the concatenated color patches of the unknown clean image. Following this observation, c_β is iteratively minimized for a set of increasing values of β (ideally $\beta\to\infty$).

Given the JCS-GMM parameters and a fixed value of β, c_β can be minimized by iteratively alternating several times between Patch Estimation and Image Estimation steps. The Patch Estimation step assumes that the image $\mathbf{u}$ in (5.6) is fixed to the estimate $\hat{\mathbf{u}}$ obtained from the previous iteration, and computes the estimates $\{\hat{\mathbf{z}}_i\}$ of the concatenated color patches. The Image Estimation step assumes that the concatenated color patches $\{\mathbf{z}_i\}$ in (5.6) are fixed to the estimates $\{\hat{\mathbf{z}}_i\}$ obtained from the Patch Estimation step, and computes the estimate $\hat{\mathbf{u}}$ of the unknown color image. Note that the Patch Estimation and Image Estimation steps assume that the JCS-GMM parameters are fixed. As our proposed JCS-GMM algorithm targets to adapt the GMM prior to the image being denoised, in every iteration, a GMM Adaptation step has also been incorporated to update GMM parameters by using the estimates $\{\hat{\mathbf{z}}_i\}$ of the concatenated color patches obtained from the Patch Estimation step. Details of Patch Estimation, Image Estimation and GMM Adaptation steps are given next.

5.3.1.1 Patch Estimation Step

The Patch Estimation step assumes that the image $\mathbf{u}$ in (5.6) is fixed to the estimate $\hat{\mathbf{u}}$ obtained from the previous iteration (or initialization $\mathbf{u}=\mathbf{y}$) and computes the estimates $\{\hat{\mathbf{z}}_i\}$ of the concatenated color patches. If $\mathbf{u}$ is fixed, the minimization of (5.6) over $\{\mathbf{z}_i\}$ can be computed independently for each of the concatenated patches as,

$$\hat{\mathbf{z}}_i=\arg\min_{\mathbf{v}}\frac{\beta}{2}\left\|\mathbf{R}_i\mathbf{u}-\mathbf{v}\right\|_2^2-\log\sum_{k=1}^{K}\hat{\omega}_k\Phi\left(\mathbf{v};\hat{\boldsymbol{\mu}}_k,\hat{\boldsymbol{\Sigma}}_k\right), \tag{5.7}$$

for $i = 1, 2, \ldots, L$, where, $\left\{\hat{\omega}_k, \hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\Sigma}}_k\right\}_{k=1}^{K}$ denotes the estimates of JCS-GMM parameters obtained from the GMM adaptation step of the previous iteration (or the initial estimates learnt from an external database). Note that, (5.7) can be interpreted as the MAP denoising (under JCS-GMM prior) of concatenated color patches contaminated by additive white Gaussian noise with variance $\frac{1}{\beta}$. The noise variance $\frac{1}{\beta}$ can be interpreted as the variance of the residual noise contained in the estimate of the unknown color image obtained from the previous iteration. As the solution in (5.7) cannot be obtained in closed form, an approximate solution is computed in two steps. First, the index k_i of the Gaussian model most likely to be responsible for the i-th concatenated color patch is estimated as,

$$\begin{aligned} \hat{k}_i &= \arg\max_k \gamma_{ik} \\ &= \arg\max_k \frac{\hat{\omega}_k \Phi\left(\mathbf{R}_i\mathbf{u}; \hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\Sigma}}_k + \frac{1}{\beta}\mathbf{I}\right)}{\sum_{k=1}^{K} \hat{\omega}_k \Phi\left(\mathbf{R}_i\mathbf{u}; \hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\Sigma}}_k + \frac{1}{\beta}\mathbf{I}\right)}, \end{aligned} \tag{5.8}$$

where, γ_{ik} is the posterior responsibility of the k-th Gaussian model in the generation of the i-th noisy concatenated color patch. Assuming that the i-th concatenated color patch is generated by the Gaussian model indexed by $\hat{k}_i$, the estimate $\hat{\mathbf{z}}_i$ is computed by solving,

$$\hat{\mathbf{z}}_i = \arg\min_{\mathbf{v}} \frac{\beta}{2} \|\mathbf{R}_i\mathbf{u} - \mathbf{v}\|_2^2 + \frac{1}{2}\left(\mathbf{v} - \hat{\boldsymbol{\mu}}_{\hat{k}_i}\right)^T \hat{\boldsymbol{\Sigma}}_{\hat{k}_i}^{-1}\left(\mathbf{v} - \hat{\boldsymbol{\mu}}_{\hat{k}_i}\right), \tag{5.9}$$

whose solution can be obtained in closed form (derivation is given in Appendix F) as,

$$\begin{aligned} \hat{\mathbf{z}}_i &= \hat{\boldsymbol{\Sigma}}_{\hat{k}_i}\left(\hat{\boldsymbol{\Sigma}}_{\hat{k}_i} + \frac{1}{\beta}\mathbf{I}\right)^{-1}\left(\mathbf{R}_i\mathbf{u} - \hat{\boldsymbol{\mu}}_{\hat{k}_i}\right) + \hat{\boldsymbol{\mu}}_{\hat{k}_i} \\ &= \mathbf{W}_{\hat{k}_i}\left(\mathbf{R}_i\mathbf{u} - \hat{\boldsymbol{\mu}}_{\hat{k}_i}\right) + \hat{\boldsymbol{\mu}}_{\hat{k}_i}, \end{aligned} \tag{5.10}$$

where, $\mathbf{W}_{\hat{k}_i} = \hat{\boldsymbol{\Sigma}}_{\hat{k}_i}\left(\hat{\boldsymbol{\Sigma}}_{\hat{k}_i} + \frac{1}{\beta}\mathbf{I}\right)^{-1}$ is the Wiener filter derived from the covariance matrix of the selected Gaussian model. The Wiener filter $\mathbf{W}_{\hat{k}_i}$ exploits inter channel correlations captured by the covariance matrix $\hat{\boldsymbol{\Sigma}}_{\hat{k}_i}$ for computing the denoised estimate of a concatenated color patch. The patch estimation step computes the estimate $\hat{\mathbf{z}}_i$ for each $i = 1, 2..., L$.

5.3.1.2 Image Estimation Step

The Image Estimation step assumes that the concatenated color patches $\{\mathbf{z}_i\}$ in (5.6) are fixed to the estimates $\{\hat{\mathbf{z}}_i\}$ obtained from the Patch Estimation step, and computes the estimate $\hat{\mathbf{u}}$ of the unknown color image as,

$$\hat{\mathbf{u}} = \arg\min_{\mathbf{u}} \frac{\lambda}{2} \|\mathbf{u} - \mathbf{y}\|_2^2 + \sum_{i=1}^{L} \frac{\beta}{2} \|\mathbf{R}_i\mathbf{u} - \mathbf{z}_i\|_2^2 \tag{5.11}$$

Solution of (5.11) can be obtained in closed form (derivation is given in Appendix G) as,

$$\hat{\mathbf{u}} = \left(\lambda\mathbf{I} + \beta \sum_{i=1}^{L} \mathbf{R}_i^T\mathbf{R}_i\right)^{-1} \left(\lambda\mathbf{y} + \beta \sum_{i=1}^{L} \mathbf{R}_i^T\mathbf{z}_i\right), \tag{5.12}$$

which corresponds to separately averaging the overlapping patches from respective color channels and linearly combining with the noisy input image weighted by λ.

5.3.1.3 GMM Adaptation Step

In every iteration, the GMM adaptation step adapts the estimates $\hat{\boldsymbol{\Theta}} = \left\{\hat{\omega}_k, \hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\Sigma}}_k\right\}_{k=1}^{K}$ of the JCS-GMM parameters to the concatenated color patches from the input image. The JCS-GMM parameters are adapted by using the estimates $\{\hat{\mathbf{z}}_i\}$ of concatenated color patches obtained from the Patch Estimation step. Let $\tilde{\boldsymbol{\Theta}} = \left\{\tilde{\omega}_k, \tilde{\boldsymbol{\mu}}_k, \tilde{\boldsymbol{\Sigma}}_k\right\}_{k=1}^{K}$ be the JCS-GMM parameters available from the previous iteration (or the initial estimates in the case of first iteration). A naive strategy for adapting the GMM prior is to update the GMM parameters through a single iteration of the EM algorithm [108, 109]. This results in the update equations given as,

$$\bar{\omega}_k = \frac{m_k}{L} \tag{5.13}$$

$$\bar{\boldsymbol{\mu}}_k = \frac{1}{m_k} \sum_{i=1}^{L} \eta_{ik}\hat{\mathbf{z}}_i \tag{5.14}$$

$$\bar{\boldsymbol{\Sigma}}_k = \frac{1}{m_k} \sum_{i=1}^{L} \eta_{ik}\hat{\mathbf{z}}_i\hat{\mathbf{z}}_i^T - \bar{\boldsymbol{\mu}}_k\bar{\boldsymbol{\mu}}_k^T \tag{5.15}$$

where, $\bar{\omega}_k$, $\bar{\boldsymbol{\mu}}_k$ and $\bar{\boldsymbol{\Sigma}}_k$ denote the updated GMM parameters, η_{ik} denotes the posterior responsibility of the k-th Gaussian model in the generation of the i-th concatenated color patch $\hat{\mathbf{z}}_i$, i.e.,

$$\eta_{ik} = \frac{\tilde{\omega}_k \Phi\left(\hat{\mathbf{z}}_i; \tilde{\boldsymbol{\mu}}_k, \tilde{\boldsymbol{\Sigma}}_k\right)}{\sum_{j=1}^{K} \tilde{\omega}_j \Phi\left(\hat{\mathbf{z}}_i; \tilde{\boldsymbol{\mu}}_j, \tilde{\boldsymbol{\Sigma}}_j\right)}, \tag{5.16}$$

and,

$$m_k = \sum_{i=1}^{L} \eta_{ik}. \tag{5.17}$$

The PLE algorithm [11] iteratively updates the GMM parameters by using a method closely similar to (5.13), (5.14) and (5.15). However, updating the GMM parameters as shown above has two major drawbacks. First, directly updating the GMM parameters using the update equations given above completely destroys the external GMM prior learnt from a rich database. Moreover, the parameter estimates computed above by using the denoised versions $\{\hat{\mathbf{z}}_i\}$ of the concatenated color patches obtained in every iteration will be slightly noisy as $\{\hat{\mathbf{z}}_i\}$ still contain some residual noise. Second, if the posterior responsibility η_{ik} corresponding to the k-th Gaussian model is too small (close to zero) for most of the patches $i = 1, 2, \ldots, L$, covariance matrix estimation in (5.15) leads to numerically unstable and near singular covariance matrices. In order to tackle these issues, we propose a controlled GMM adaptation scheme. In every iteration, the proposed GMM adaptation method preserves a history of the GMM parameter values obtained from the previous iteration.

The proposed JCS-GMM denoising algorithm adapts the parameters of the k-th Gaussian model as,

$$\hat{\omega}_k = \alpha_k \bar{\omega}_k + (1 - \alpha_k)\, \tilde{\omega}_k \tag{5.18}$$

$$\hat{\boldsymbol{\mu}}_k = \alpha_k \bar{\boldsymbol{\mu}}_k + (1 - \alpha_k)\, \tilde{\boldsymbol{\mu}}_k \tag{5.19}$$

$$\hat{\boldsymbol{\Sigma}}_k = \alpha_k \left(\bar{\boldsymbol{\Sigma}}_k + \bar{\boldsymbol{\mu}}_k \bar{\boldsymbol{\mu}}_k^T\right) + (1 - \alpha_k) \left(\tilde{\boldsymbol{\Sigma}}_k + \tilde{\boldsymbol{\mu}}_k \tilde{\boldsymbol{\mu}}^T\right) - \hat{\boldsymbol{\mu}}_k \hat{\boldsymbol{\mu}}_k^T \tag{5.20}$$

for $k = 1, 2, \ldots, K$, where, $\left\{\bar{\omega}_k, \bar{\boldsymbol{\mu}}_k, \bar{\boldsymbol{\Sigma}}_k\right\}$ denote the parameter values computed using (5.13), (5.14) and (5.15), $\left\{\tilde{\omega}_k, \tilde{\boldsymbol{\mu}}_k, \tilde{\boldsymbol{\Sigma}}_k\right\}$ denote the parameter values obtained from the previous iteration, and, $\alpha_k \in [0, 1]$ denotes the adaptation weight. The adaptation weight

α_k controls the adaptation of parameters of the k-th Gaussian model in such a way that, a higher value of α_k corresponds to relying more on the estimates computed in (5.13), (5.14) and (5.15), whereas, a smaller value corresponds to inheriting the parameter values $\left\{\tilde{\omega}_k, \tilde{\boldsymbol{\mu}}_k, \tilde{\boldsymbol{\Sigma}}_k\right\}$ from the previous iteration.

We propose a simple and intuitive strategy for computing the adaptation weight α_k corresponding to the k-th Gaussian model for $k = 1, 2, \ldots, K$. Each of the estimates $\hat{\mathbf{z}}_i$ of the concatenated color patches is associated to one of the Gaussian models with index $\hat{k}_i$ selected as,

$$\hat{k}_i = \arg\max_k \eta_{ik}. \tag{5.21}$$

Let $\mathcal{C}_k$ be the set of concatenated color patches associated to the k-th Gaussian model, i.e.,

$$\mathcal{C}_k = \left\{ \hat{\mathbf{z}}_i \,\middle|\, \hat{k}_i = k \right\}. \tag{5.22}$$

As the number of patches $|\mathcal{C}_k|$ associated to a given k-th Gaussian model is more, the corresponding parameter estimates computed in (5.13), (5.14) and (5.15) are assumed to be more reliable. If a large number of patches is associated to a given Gaussian model, note that the posterior responsibility values $\{\eta_{ik}\}$ corresponding to the given model are likely to be high for many of the patches. Based on the above assumption, the adaptation weight α_k corresponding to the k-th Gaussian model is computed as,

$$\alpha_k = 1 - \frac{\sum_{i=1}^{N} \eta_{ik}}{\sum_{i=1}^{N} \eta_{ik} + |\mathcal{C}_k|} \tag{5.23}$$

As the number of patches $|\mathcal{C}_k|$ associated to a Gaussian model becomes smaller, the corresponding adaptation weight α_k becomes closer to zero, and the parameter adaptations in (5.18), (5.19) and (5.20) mostly inherits the parameter values from the previous iteration.

The proposed GMM adaptation method is inspired from the GMM adaptation technique used by the speaker recognition algorithm [51] based on GMM-UBM (Universal Background Model). We note that our proposed GMM adaptation method bears some similarities with the GMM adaptation scheme used by the monochrome image denoising algorithm proposed in [43], but our proposed method differs from [43] in the following aspects. In [43], an externally learnt GMM prior is adapted to the input image by using the pilot denoised

patches obtained by applying any existing denoising algorithm on the input noisy image. The adapted GMM prior is then used in the EPLL denoising framework to compute the denoised estimate of a given input noisy image. On the other hand, our proposed method starts with an externally learnt GMM prior, and iteratively adapts the GMM parameters by using the intermediate denoised patches obtained in every iteration. Consequently, our proposed method does not require any other external denoising algorithm. Moreover, the proposed GMM adaptation method computes the adaptation weights by using a simple heuristic based on the number of patches associated to the Gaussian models, whereas, GMM adaptation procedure used in [43] computes the adaptation weights by imposing a hyper prior on the GMM parameters.

The Patch Estimation, Image Estimation and GMM Adaptation steps of the proposed JCS-GMM denoising algorithm are repeated several times for a set of increasing values of β, and the image estimate $\hat{\mathbf{u}}$ obtained for a sufficiently large value of β is chosen as the denoised estimate $\hat{\mathbf{x}}$ of the unknown clean color image $\mathbf{x}$. The proposed JCS-GMM denoising algorithm can be seen as GMM analogue of the sparsity based color image denoising method introduced in the Color-KSVD [25] algorithm. The proposed GMM adaptation scheme is analogous to the dictionary learning used by the Color-KSVD algorithm for adapting a global dictionary to the specific image being denoised.

5.3.2 JCS-GMM Demosaicking

The most commonly utilized method for capturing color images using digital cameras is to record the response of CCD sensors exposed to the spectral components corresponding to Red, Green and Blue wavelengths of the incoming light. Ideally, the response of each of the CCD sensors has to be measured by using three different spectral filters corresponding to the Red, Green and Blue wavelengths. However, as it is difficult to implement three different spectral filters at all the pixel locations, color images are generally captured with the help of Color Filter Arrays (CFAs). A color filter array is a mosaic of three spectral filters arranged in such a way that only one filter is available at any given pixel location. The color images captured by using CFAs have only one of the color components among R, G and B available at any given pixel location, while the other two components are masked

by the CFA. The most commonly used pattern in CFAs is the Bayer pattern. The image captured by using CFAs is called the *mosaicked image* as it contains a spatial mosaic of three spectral responses recorded at different pixel locations. The color image restoration problem corresponding to the estimation of a full color image by interpolating the two missing color components at every pixel location is called *demosaicking*.

In the case of demosaicking problem, the degradation operator $\mathbf{A}$ in (5.4) is a mask matrix which selectively admits one of the three color components among R, G and B at every pixel location while masking the other two components. As considered in most of the previous works on color image demosaicking, we assume a noiseless scenario and focus only on interpolating the missing color components. The mosaicked concatenated color patch $\mathbf{q}_i$ corresponding to an unknown full color concatenated patch $\mathbf{p}_i$ can be written as,

$$\mathbf{q}_i = \mathbf{B}_i \mathbf{p}_i, \tag{5.24}$$

where, $\mathbf{B}_i \in \mathbb{R}^{3\tau^2 \times 3\tau^2}$ is the mask operator restricted to the i-th patch. Given the masked observations $\{\mathbf{q}_i\}_{i=1}^{L}$, the proposed JCS-GMM demosaicking algorithm computes the estimate $\hat{\mathbf{p}}_i = \left[\hat{\mathbf{p}}_{iR}^T, \hat{\mathbf{p}}_{iG}^T, \hat{\mathbf{p}}_{iB}^T\right]^T$ of the unknown full color concatenated patch for all the patches $i = 1, 2, ..., L$, and average the overlapping patches $\{\hat{\mathbf{p}}_{ic}\}_{i=1}^{N}$ from respective color channels $c = R, G, B$ to compute the interpolated color channels $\hat{\mathbf{x}}_R, \hat{\mathbf{x}}_G, \hat{\mathbf{x}}_B$.

The proposed JCS-GMM demosaicking algorithm computes the interpolated estimates $\{\hat{\mathbf{p}}_i\}_{i=1}^{L}$ of the unknown full color concatenated patches $\{\mathbf{p}_i\}_{i=1}^{L}$ by iteratively alternating several times between the Patch Estimation and GMM Adaptation steps. In every iteration, the Patch Estimation step assumes that the JCS-GMM parameters obtained from the previous iteration (or initialization) are fixed, and computes the interpolated estimates $\{\hat{\mathbf{p}}_i\}$. The GMM Adaptation step updates JCS-GMM parameters by using the estimates $\{\hat{\mathbf{p}}_i\}$ of the concatenated color patches obtained in the Patch Estimation step.

5.3.2.1 Patch Estimation Step

The patch estimation step assumes that the estimates $\hat{\mathbf{\Theta}} = \left\{\hat{\omega}_k, \hat{\boldsymbol{\mu}}_k, \hat{\mathbf{\Sigma}}_k\right\}_{k=1}^{K}$ of the JCS-GMM parameters available from the previous iteration (or the initial estimates learnt

from an external database) are fixed, and computes the interpolated estimate $\hat{\mathbf{p}}_i$ of the unknown full color concatenated patch $\mathbf{p}_i$ for all $i = 1, 2, ..., L$. The Patch Estimation step selects the index $\hat{k}_i$ of the Gaussian model responsible for each of the concatenated color patches, and interpolates the missing pixel values by using the selected Gaussian models. Let $\mathcal{I}_m \subset \left\{1, 2, ..., 3\tau^2\right\}$ denotes the set of indices of masked pixels of a mosaicked concatenated patch vector $\mathbf{q}_i$, and $\mathcal{I}_a = \left\{1, 2, ..., 3\tau^2\right\} \setminus \mathcal{I}_m$ denotes the set of indices of available pixels. As the pixel values at the indices contained in $\mathcal{I}_a$ is available in a given mosaicked concatenated patch vector $\mathbf{q}_i$, note that, $\mathbf{p}_i^{\mathcal{I}_a} = \mathbf{q}_i^{\mathcal{I}_a}$, where, $\mathbf{p}_i^{\mathcal{I}_a}$ and $\mathbf{q}_i^{\mathcal{I}_a}$ respectively denote the subvectors of $\mathbf{p}_i$ and $\mathbf{q}_i$ restricted to the index set $\mathcal{I}_a$. Our task is to estimate the missing pixels $\mathbf{p}_i^{\mathcal{I}_m}$ of the concatenated color patch $\mathbf{p}_i$ for all $i = 1, 2, ..., L$.

The index $\hat{k}_i$ of the Gaussian model responsible for a given mosaicked concatenated patch $\mathbf{q}_i$ is selected as,

$$\hat{k}_i = \arg\max_k \frac{\hat{\omega}_k \Phi\left(\mathbf{q}_i^{\mathcal{I}_a}; \hat{\boldsymbol{\mu}}_k^{\mathcal{I}_a}, \hat{\boldsymbol{\Sigma}}_k^{\mathcal{I}_a\mathcal{I}_a}\right)}{\sum_{k=1}^{K} \hat{\omega}_k \Phi\left(\mathbf{q}_i^{\mathcal{I}_a}; \hat{\boldsymbol{\mu}}_k^{\mathcal{I}_a}, \hat{\boldsymbol{\Sigma}}_k^{\mathcal{I}_a\mathcal{I}_a}\right)}, \tag{5.25}$$

where, $\hat{\boldsymbol{\mu}}_k^{\mathcal{I}_a}$ denotes the mean vector $\hat{\boldsymbol{\mu}}_k$ restricted to the index set $\mathcal{I}_a$, and $\hat{\boldsymbol{\Sigma}}_k^{\mathcal{I}_a\mathcal{I}_a}$ denotes the submatrix obtained by restricting the rows and columns of the covariance matrix $\hat{\boldsymbol{\Sigma}}_k$ to the index set $\mathcal{I}_a$. Assuming that the i-th concatenated color patch is distributed according the selected Gaussian model with index $\hat{k}_i$, it can be observed that the subvectors $\mathbf{p}_i^{\mathcal{I}_a}$ and $\mathbf{p}_i^{\mathcal{I}_m}$ are also Gaussian with parameters $\left(\hat{\boldsymbol{\mu}}_{\hat{k}_i}^{\mathcal{I}_a}, \hat{\boldsymbol{\Sigma}}_{\hat{k}_i}^{\mathcal{I}_a\mathcal{I}_a}\right)$ and $\left(\hat{\boldsymbol{\mu}}_{\hat{k}_i}^{\mathcal{I}_m}, \hat{\boldsymbol{\Sigma}}_{\hat{k}_i}^{\mathcal{I}_m\mathcal{I}_m}\right)$ respectively. Given $\hat{k}_i$, the MMSE estimate $\hat{\mathbf{p}}_i^{\mathcal{I}_m}$ of the missing pixels $\mathbf{p}_i^{\mathcal{I}_m}$ can be obtained as the conditional expectation (a brief outline of the derivation is given in Appendix B),

$$\begin{aligned}
\hat{\mathbf{p}}_i^{\mathcal{I}_m} &= \mathbb{E}\left[\mathbf{p}_i^{\mathcal{I}_m} \middle| \mathbf{p}_i^{\mathcal{I}_a}\right] \\
&= \hat{\boldsymbol{\mu}}_{\hat{k}_i}^{\mathcal{I}_m} + \hat{\boldsymbol{\Sigma}}_{\hat{k}_i}^{\mathcal{I}_m\mathcal{I}_a} \left(\hat{\boldsymbol{\Sigma}}_{\hat{k}_i}^{\mathcal{I}_a\mathcal{I}_a}\right)^{-1} \left(\mathbf{p}_i^{\mathcal{I}_a} - \hat{\boldsymbol{\mu}}_{\hat{k}_i}^{\mathcal{I}_a}\right) \\
&= \hat{\boldsymbol{\mu}}_{\hat{k}_i}^{\mathcal{I}_m} + \hat{\boldsymbol{\Sigma}}_{\hat{k}_i}^{\mathcal{I}_m\mathcal{I}_a} \left(\hat{\boldsymbol{\Sigma}}_{\hat{k}_i}^{\mathcal{I}_a\mathcal{I}_a}\right)^{-1} \left(\mathbf{q}_i^{\mathcal{I}_a} - \hat{\boldsymbol{\mu}}_{\hat{k}_i}^{\mathcal{I}_a}\right),
\end{aligned} \tag{5.26}$$

where, $\hat{\boldsymbol{\Sigma}}_{\hat{k}_i}^{\mathcal{I}_m\mathcal{I}_a}$ denotes the submatrix obtained by restricting the rows and columns of the covariance matrix $\hat{\boldsymbol{\Sigma}}_{\hat{k}_i}$ to the index sets $\mathcal{I}_m$ and $\mathcal{I}_a$ respectively. The Patch estimation step computes the estimate $\hat{\mathbf{p}}_i^{\mathcal{I}_m}$ of the missing pixels, and copies the available pixels as $\hat{\mathbf{p}}_i^{\mathcal{I}_a} = \mathbf{q}_i^{\mathcal{I}_a}$, to compute the interpolated estimate $\hat{\mathbf{p}}_i$ for all the concatenated patches $i = 1, 2, ..., L$.

5.3.2.2 GMM Adaptation Step

In every iteration, the GMM Adaptation step updates the estimates $\hat{\boldsymbol{\Theta}} = \left\{\hat{\omega}_k, \hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\Sigma}}_k\right\}_{k=1}^{K}$ of the JCS-GMM parameters by using the interpolated estimates $\hat{\mathbf{p}}_i$ of the concatenated color patches obtained from the Patch Estimation step. Given the interpolated estimates $\{\hat{\mathbf{p}}_i\}_{i=1}^{L}$, the GMM Adaptation step exactly follows the same procedure used for updating the GMM parameters in the JCS-GMM denoising algorithm.

The Patch Estimation and GMM Adaptation steps are iterated several times, and the estimates $\{\hat{\mathbf{p}}_i\}_{i=1}^{L}$ obtained in the final iteration are chosen as the interpolated full color concatenated patches. The interpolated patches $\hat{\mathbf{p}}_{iR}$, $\hat{\mathbf{p}}_{iG}$ and $\hat{\mathbf{p}}_{iB}$ from different color channels are extracted from each of the concatenated vectors $\hat{\mathbf{p}}_i$, and the overlapping patches from respective color channels are averaged to compute the interpolated estimates $\hat{\mathbf{x}}_R$, $\hat{\mathbf{x}}_G$ and $\hat{\mathbf{x}}_B$ of different color channels.

5.4 Experiments and Results

In this section, we describe the color image denoising and demosaicking experiments carried out for comparing the performance of our proposed JCS-GMM method with that of various competing state of the art denoising and demosaicking algorithms, and discuss the corresponding results. The selection of appropriate values for different parameters of our proposed algorithms is also discussed.

5.4.1 Selection of Parameter Values

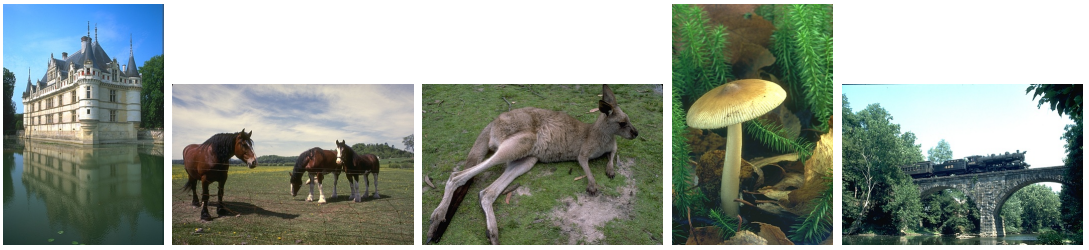
In all our denoising and demosaicking experiments, the initial GMM prior used by the proposed JCS-GMM method is learnt from a set of 10^6 noiseless color image patches randomly sampled from 495 color images contained in the Berkeley segmentation database (excluding the test images shown in Figure 5.2). Each color image patch of size $\tau \times \tau \times 3$ pixels is reshaped as in (5.1) to construct the concatenated color patches, and the JCS-GMM parameters are estimated from the database of concatenated color patches by using the EM algorithm [108, 109] summarized in Appendix A. In order to choose an appropriate

patch dimension τ, and the number of Gaussian models K included in the GMM, a series of denoising experiments have been conducted by using the JCS-GMM prior learnt with different patch sizes $\tau = 6, 7$ and 8, and with different number of Gaussian models $K = 75, 100, 150$ and 200. It has been observed that the denoising performance increases as the number of Gaussian models K increases, whereas, different patch dimensions have little or negligible effect in the performance. On the other hand, increasing the number of Gaussian models K, and/or the patch dimension τ, significantly increases the run time of the EM algorithm used for learning the GMM parameters. As a compromise between the learning effort and denoising performance, the patch dimension and the number of Gaussian models are selected as $\tau = 6$ and $K = 150$ respectively.

In all our denoising experiments, the Patch Estimation, Image Estimation, and GMM Adaptation steps are repeated for a fixed set of increasing values of β chosen as $\beta = 1, 4, 8, 16, 32, 64$, and the weight parameter λ in (5.5) is set to $\frac{\tau^2}{\sigma^2}$. The number of iterations of Patch Estimation, Image Estimation, and GMM Adaptation steps for a fixed value of β is set to 1 in all the experiments. In the case of demosaicking experiments, the GRBG structure of the widely used Bayer pattern is used for simulating the CFA. Note that the proposed demosaicking algorithm can be trivially extended to any other CFA. The number of iterations of Patch Estimation and GMM Adaptation steps are fixed to 5 in all the demosaicking experiments.

5.4.2 JCS-GMM Denoising

The proposed JCS-GMM denoising algorithm is compared with PLE [11], Color-KSVD [25], Quaternion KSVD (Q-KSVD) [57], NL-Bayes [20], HOSVD [138] and, color version of the widely studied BM3D algorithm (C-BM3D) [112]. Though, the PLE algorithm [11] was originally proposed as a generalized framework for various monochrome/color image restoration problems, the results on image denoising were not reported in [11]. However, we have implemented a color image denoising version of the PLE algorithm and included in our comparison. Color image denoising experiments are conducted on a set of five natural color images shown in Figure 5.2. These images are selected from the Berkeley segmentation database, and they were used for evaluating the denoising



(a) (b) (c) (d) (e)

FIGURE 5.2: Test images used in color image denoising experiments: (a) castle, (b) horses, (c) kangaroo, (d) mushroom, (e) train

performance in some of the previous works [131, 25, 13, 139] on color image denoising. Denoising experiments are conducted for five different noise levels $\sigma = 5, 10, 15, 25$ and 35, and the denoising performance is measured in terms of Color Peak Signal to Noise Ratio (CPSNR) and Color Structural Similarity Index Measure (CSSIM). The CPSNR values are computed with the MSE averaged over different color channels, and CSSIM values are computed by averaging the SSIM values corresponding to different color channels of a given color image. Each test image shown in Figure 5.2 is corrupted by adding white Gaussian noise with different noise variance σ^2, and is denoised independently by using different color image denoising algorithms included in the comparison. All CPSNR/CSSIM values reported in the following are computed by averaging over five independent trials to reduce the effect of random noise in performance evaluation.

In the case of PLE [11] algorithm, a blockwise denoising approach as suggested in [11] has been followed by splitting a given noisy color image into several smaller blocks of size 128×128 pixels with an overlap of 64 pixels, and by independently denoising each of such blocks by using the PLE restoration framework. Denoised blocks are then aggregated by arranging each of the blocks at their respective positions and averaging the overlapping blocks. The parameters of the initial GMM prior used by the PLE algorithm are computed by the synthetic patch based initialization scheme proposed in [11]. The MAP-EM algorithm [11] used by the PLE restoration framework has been set to iterate 4 times between the Patch Estimation and Parameter Update steps. The values of different parameters used by Q-KSVD, Color-KSVD and C-BM3D algorithms are as configured in the publicly available source codes. The initial dictionaries used by Q-KSVD and Color-KSVD algorithms are same as the ones provided in the publicly available source codes. Among

several variants of the HOSVD algorithm, the 4DHOSVD2 algorithm [138] producing the best results in color image denoising has been adopted in our experiments.

Figure 5.3 shows the denoised results obtained by different color image denoising algorithms in the case of 'Horses' image for noise level $\sigma = 25$. Each subfigure shows a cropped region of size 128×128 pixels selected from the actual denoised images of size 321×481 pixels produced by different denoising algorithms. The selected region of the image contains both textural contents (mane of the horse and flowers on the ground) as well as smooth regions (sky) which are helpful in the subjective analysis of capabilities of different denoising algorithms in preserving fine structures of the image, and control over spurious artifacts introduced in the smooth regions. It can be observed from Figure 5.3 that our proposed JCS-GMM algorithm preserves the fine structural details of the image and achieves the best performance in terms of CPSNR when compared with all other algorithms. As can be observed from subfigures (f) and (i), our proposed algorithm is better at preserving the fine details such as flowers on the ground and mane of the horse when compared with C-BM3D which is the second best performing algorithm. This demonstrates the improvement in the visual quality of the denoised image produced by our proposed method when compared with that of C-BM3D, which can be considered as a state of the art color image denoising algorithm. Both C-BM3D and our proposed JCS-GMM algorithm introduce negligible artifacts in the smooth regions of the image, such as the sky appearing on the top part of the image. On the other hand, the Color-KSVD algorithm preserves the fine details of the image upto some extent, but introduces spurious artifacts in the smooth regions, which is clearly visible on the sky above horse's head in subfigure (e). As can be verified from subfigures (g) and (h), the NL-Bayes algorithm slightly destroys the textured region of flowers on the ground, whereas, HOSVD algorithm produces ringing artifacts in these regions. The subfigures (c) and (d) show that the denoised image produced by the PLE algorithm contains a lot of spurious artifacts and the denoised image obtained from the Q-KSVD algorithm is overly smooth. The visual comparison between denoised images produced by the proposed JCS-GMM method and various color image denoising algorithms in the case of different noise levels and several images is shown in Appendix H.

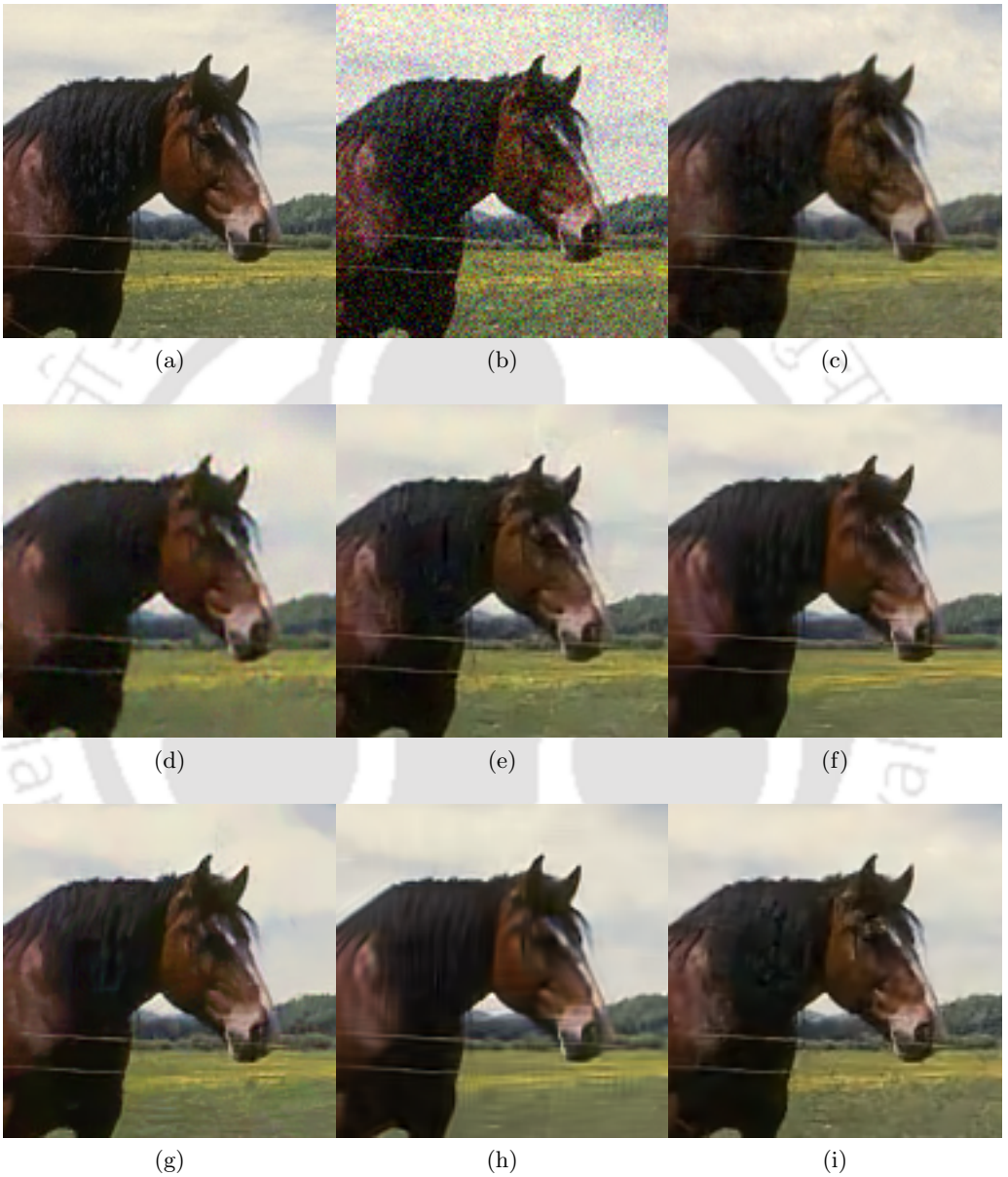


FIGURE 5.3: Denoised results (cropped to a size 128×128 pixels) in the case of Horses image and noise level $\sigma = 25$. (a) Ground Truth, (b) Noisy Image (20.17 dB) (c) PLE (29.62 dB), (d) Q-KSVD (28.35 dB), (e) Color-KSVD (29.96 dB), (f) C-BM3D (29.97 dB), (g) NL-Bayes (29.89 dB), (h) HOSVD (29.40 dB), (i) Proposed (30.28 dB)

σ	PLE	Q-KSVD	Color-KSVD	C-BM3D	NL-Bayes	HO SVD	Prop
5	40.45	35.22	40.47	**40.83**	40.55	40.28	**40.85**
	0.9717	0.9497	0.9737	**0.9745**	**0.9732**	**0.9725**	**0.9754**
10	36.07	32.77	36.43	**36.61**	36.34	36.19	**36.64**
	0.9401	0.9121	**0.9461**	**0.9448**	0.9428	0.9409	**0.9487**
15	33.64	31.25	34.15	**34.35**	34.01	34.04	**34.37**
	0.9107	0.8831	**0.9218**	**0.9227**	0.9151	0.9166	**0.9258**
25	30.71	29.08	31.40	**31.67**	31.46	31.54	**31.69**
	0.8640	0.8320	0.8824	**0.8885**	0.8797	0.8829	**0.8899**
35	28.91	27.56	29.61	29.85	29.68	29.92	**30.03**
	0.8215	0.7912	0.8488	0.8562	0.8467	0.8539	**0.8628**

TABLE 5.1: CPSNR and CSSIM performance on Castle Image

σ	PLE	Q-KSVD	Color-KSVD	C-BM3D	NL-Bayes	HO SVD	Prop
5	39.76	35.24	39.71	**40.04**	39.64	39.25	**40.13**
	0.9755	0.9447	**0.9761**	**0.9774**	**0.9748**	**0.9752**	**0.9778**
10	34.95	32.20	35.23	**35.43**	35.03	34.87	**35.51**
	0.9402	0.8953	**0.9443**	**0.9468**	0.9391	0.9410	**0.9484**
15	32.41	30.39	32.74	32.89	32.49	32.28	**33.02**
	0.9020	0.8466	0.9098	0.9099	0.8984	0.8936	**0.9168**
25	29.62	28.35	29.96	29.97	29.89	29.40	**30.28**
	0.8254	0.7544	0.8396	0.8323	0.8318	0.8012	**0.8533**
35	28.00	27.08	28.37	28.29	28.20	28.04	**28.68**
	0.7611	0.6813	0.7741	0.7631	0.7644	0.7406	**0.7917**

TABLE 5.2: CPSNR and CSSIM performance on Horses Image

σ	PLE	Q-KSVD	Color-KSVD	C-BM3D	NL-Bayes	HO SVD	Prop
5	38.44	34.27	38.38	**38.63**	38.24	37.76	**38.64**
	0.9781	0.9413	**0.9786**	**0.9790**	**0.9770**	**0.9755**	**0.9792**
10	33.47	30.98	**33.63**	**33.71**	33.33	33.20	**33.71**
	0.9314	0.8751	**0.9344**	**0.9339**	0.9265	0.9258	**0.9367**
15	30.95	29.16	**31.12**	30.75	31.09	30.42	**31.21**
	0.8767	0.8016	0.8816	0.8735	0.8630	0.8406	**0.8880**
25	28.27	27.06	**28.44**	28.28	28.21	27.71	**28.54**
	0.7677	0.6565	0.7748	0.7503	0.7549	0.6952	**0.7846**
35	26.77	25.79	**26.96**	26.72	26.63	26.46	**27.00**
	0.6711	0.5496	0.6755	0.6428	0.6511	0.6025	**0.6844**

TABLE 5.3: CPSNR and CSSIM performance on Kangaroo Image

σ	PLE	Q-KSVD	Color-KSVD	C-BM3D	NL-Bayes	HO SVD	Prop
5	39.44	36.82	39.36	**39.61**	39.23	38.84	**39.60**
	0.9720	0.9558	**0.9726**	**0.9733**	**0.9705**	**0.9699**	**0.9736**
10	35.24	33.42	35.33	**35.47**	35.09	35.02	**35.38**
	0.9368	0.9076	**0.9383**	**0.9388**	0.9317	0.9335	**0.9395**
15	32.88	31.40	33.05	**33.23**	32.77	32.92	33.12
	0.8988	0.8582	**0.9040**	**0.9053**	0.8918	0.8970	**0.9069**
25	30.08	28.80	30.40	**30.55**	30.38	30.29	**30.48**
	0.8252	0.7606	**0.8392**	**0.8419**	0.8328	0.8274	**0.8435**
35	28.22	27.19	28.71	28.73	28.62	28.66	**28.87**
	0.7505	0.6802	0.7737	0.7776	0.7681	0.7661	**0.7861**

TABLE 5.4: CPSNR and CSSIM performance on Mushroom Image

σ	PLE	Q-KSVD	Color-KSVD	C-BM3D	NL-Bayes	HO SVD	Prop
5	39.39	31.67	39.44	39.63	39.52	38.70	**39.75**
	0.9876	0.9604	**0.9883**	**0.9896**	**0.9893**	**0.9878**	**0.9900**
10	34.08	29.51	34.59	34.63	34.56	33.76	**34.75**
	0.9686	0.9321	**0.9717**	**0.9729**	**0.9726**	0.9684	**0.9751**
15	31.07	27.99	31.74	31.79	31.72	31.12	**31.94**
	0.9454	0.9035	0.9516	**0.9532**	0.9511	0.9451	**0.9568**
25	27.73	25.76	28.39	28.36	28.54	27.97	**28.66**
	0.8933	0.8400	0.9059	0.9051	0.9080	0.8888	**0.9154**
35	25.76	24.32	26.40	26.22	26.46	25.77	**26.69**
	0.8386	0.7762	0.8591	0.8507	0.8580	0.8224	**0.8726**

TABLE 5.5: CPSNR and CSSIM performance on Train Image

Tables 5.1 - 5.5 respectively show the comparison of CPSNR and CSSIM values obtained by different color image denoising algorithms in the case of Castle, Horses, Kangaroo, Mushroom and Train images shown in Figure 5.2. In each table, rows separated by double lines correspond to different noise levels, and columns correspond to different denoising algorithms. Each cell corresponding to a given noise level and a given denoising algorithm is vertically divided into two parts with the upper half showing the CPSNR value and the lower half showing the CSSIM value. Corresponding to each noise level, the best (highest) CPSNR and CSSIM values obtained among different denoising algorithms are shown in boldface. As such, the highly competitive CPSNR and CSSIM values within a gap of 0.1 dB and 0.0050 respectively from the corresponding best values are also shown in

boldface. As can be observed from Table 5.1, our proposed JCS-GMM algorithm achieves the best performance in terms of CPSNR and CSSIM values in the case of Castle image for all the noise levels. The proposed algorithm achieves a CPSNR improvement of 0.02, 0.03, 0.02, 0.02 and 0.18 dBs respectively for the noise levels $\sigma = 5, 10, 15, 25$ and 35, when compared with the state of the art C-BM3D algorithm. The proposed JCS-GMM algorithm consistently outperforms the Color-KSVD algorithm for all the noise levels, and maintains a CPSNR improvement of 0.38, 0.21, 0.22, 0.29 and 0.42 dBs respectively for $\sigma = 5, 10, 15, 25$ and 35. In terms of CSSIM values, the proposed algorithm achieves an average (averaged over all noise levels) CSSIM improvement of 0.0031 when compared with the C-BM3D algorithm, and an improvement of 0.0059 when compared with Color-KSVD algorithm. It can be verified that the proposed method achieves superior performance in terms of both CPSNR and CSSIM measures for all the noise levels when compared with PLE, Q-KSVD, NL-Bayes and HOSVD algorithms.

Table 5.2 shows the comparison of CPSNR and CSSIM performance of different denoising algorithms in the case of Horses image. It can be observed from Table 5.2 that the proposed JCS-GMM algorithm consistently achieves superior denoising performance for all the noise levels when compared with C-BM3D and Color-KSVD algorithms, especially in the high noise regime. The proposed algorithm maintains a CPSNR improvement of 0.31 and 0.39 dBs respectively for noise levels $\sigma = 25$ and 35 when compared with C-BM3D algorithm, and, 0.32 and 0.31 dBs when compared with Color-KSVD algorithm. In terms of CSSIM values, the proposed method achieves an improvement of 0.0210 and 0.0286 respectively for noise levels $\sigma = 25$ and 35 when compared with C-BM3D algorithm, and, 0.0137 and 0.0176 when compared with Color-KSVD algorithm. Table 5.3 shows the CPSNR and CSSIM performance obtained by various algorithms in the case of Kangaroo image. As in the case of Castle and Horses images, our proposed JCS-GMM algorithm outperforms C-BM3D and Color-KSVD algorithms in terms of CPSNR measure for all the noise levels except for $\sigma = 10$, in which case, both C-BM3D and our proposed algorithm achieves the same CPSNR performance. On the other hand, the proposed algorithm consistently achieves superior performance in terms of CSSIM values when compared with C-BM3D and Color-KSVD algorithms. The proposed algorithm shows significant improvement in

CPSNR and CSSIM values when compared with that of PLE, Q-KSVD, NL-Bayes and HOSVD algorithms for all the noise levels in the case of Horses and Kangaroo images.

In the case of Mushroom image, it can be observed from Table 5.4 that the CPSNR performance of our proposed algorithm is slightly inferior to that of C-BM3D algorithm, especially in the low noise regime. The C-BM3D algorithm achieves a CPSNR improvement of $0.01, 0.09, 0.12$ and 0.09 dBs when compared with the proposed algorithm in the case of noise levels $\sigma = 5, 10, 15$ and 25, whereas, the proposed algorithm outperforms C-BM3D in the case of $\sigma = 35$. The proposed JCS-GMM algorithm consistently achieves superior CSSIM performance for all the noise levels when compared with the C-BM3D algorithm. Note that the proposed algorithm achieves improved denoising performance for all the noise levels when compared with PLE, Q-KSVD, Color-KSVD, NL-Bayes and HOSVD algorithms in terms of both CPSNR and CSSIM measures. Table 5.5 corresponding to the case of Train image further demonstrates the improved denoising performance of our proposed JCS-GMM algorithm when compared with other denoising algorithms. In the case of Train image, the proposed algorithm achieves a CPSNR improvement of $0.12, 0.12$ and 0.15 dBs for noise levels $\sigma = 5, 10$ and 25 when compared with the C-BM3D algorithm. As observed in the case of Horses image, the proposed algorithm significantly outperforms the C-BM3D algorithm in the high noise regime, and maintains a CPSNR gap of 0.3 and 0.47 dBs respectively for noise levels $\sigma = 25$ and 35. The proposed algorithm achieves a CSSIM improvement of $0.0004, 0.0022, 0.0036, 0.0103$ and 0.0219 respectively for noise levels $\sigma = 5, 10, 15, 25$ and 35 when compared with C-BM3D algorithm, and shows significantly better performance when compared with other algorithms.

As the source codes corresponding to the color image denoising algorithms based on Multi Scale KSVD (MS-KSVD) [13] and Discriminatively Learned Iterative Shrinkage (DLIS) [139] are not publicly available, we could not compare the denoising performance of our proposed JCS-GMM denoising algorithm with that of MS-KSVD and DLIS algorithms. However, a comparison of denoising performance of our proposed JCS-GMM algorithm with that of MS-KSVD and DLIS algorithms can be carried out directly in terms of CPSNR values reported in [13] and [139]. As can be observed from the average CPSNR values reported in [13] (Table 4), the performance of MS-KSVD algorithm is very close to that of C-BM3D, with an average performance improvement of 0.03 and 0.04 dBs

respectively for noise levels $\sigma = 10$ and 20. Similarly, Table II in [139] shows that the DLIS algorithm achieves an average performance improvement of 0.07 and 0.15 dBs respectively for noise levels $\sigma = 10$ and 20 when compared with the C-BM3D algorithm. In the case of our proposed JCS-GMM algorithm, the average performance improvement over C-BM3D algorithm is 0.02 and 0.11 dBs respectively for noise levels $\sigma = 10$ and 20. Note that the above comparison roughly demonstrates that the denoising performance of our proposed JCS-GMM algorithm is superior to the MS-KSVD algorithm in the high noise regime and comparable to that of DLIS algorithm.

5.4.3 JCS-GMM Demosaicking

In the case of demosaicking experiments, the proposed JCS-GMM demosaicking algorithm is compared with Color-KSVD [25], MSG [121], RI [124], MLRI [126], SSCAD [142], ARI [125], MLRI+wei [127], DDR and FR [143] algorithms. Color image demosaicking algorithms are generally evaluated on two different databases [142], namely, the Kodak and IMAX databases. The Kodak database contains color images which are challenging in their Nyquist frequencies, whereas, IMAX database contains images with saturated color regions [142]. We evaluate the performance of our proposed JCS-GMM demosaicking algorithm on both Kodak and IMAX databases. The demosaicking performance of various algorithms is compared in terms of PSNR values of individual color channels, CPSNR, SSIM values of individual color channels, and CSSIM. The visual comparison between the demosaicked images produced by the proposed algorithm and various demosaicking algorithms in the case of Kodak and IMAX databases is shown in Appendix I.

Table 5.6 shows the comparison of PSNR and SSIM performance of our proposed JCS-GMM demosaicking algorithm with that of various demosaicking algorithms. Table 5.6(a) corresponds to the Kodak database, and Table 5.6(b) corresponds to the IMAX database. In Tables 5.6(a) and 5.6(b), each row corresponding to a given demosaicking algorithm is vertically divided into two parts, with the upper half showing the average PSNR value obtained over all the images in a database, and lower half showing the average SSIM value. The R, G and B columns of Table 5.6(a) and 5.6(b) show the average PSNR/SSIM values obtained by different algorithms in the case of the Red, Green and Blue channels

	R	G	B	C
Color-KSVD	39.54	43.20	39.72	40.46
	0.9832	**0.9910**	**0.9825**	**0.9856**
MSG	40.09	43.81	40.35	41.01
	0.9843	**0.9920**	**0.9837**	**0.9867**
RI	38.95	43.11	39.27	39.95
	0.9807	**0.9909**	0.9801	0.9839
MLRI	39.29	42.85	39.58	40.19
	0.9821	**0.9905**	**0.9814**	**0.9847**
SSCAD	39.66	43.31	40.02	40.63
	0.9829	**0.9909**	**0.9823**	**0.9854**
ARI	39.07	42.28	38.88	39.77
	0.9790	0.9878	0.9759	0.9809
MLRI+wei	39.38	42.96	39.63	40.28
	0.9826	**0.9907**	**0.9817**	**0.9850**
DDR	40.13	4384	40.33	41.04
	0.9845	**0.9920**	**0.9837**	**0.9868**
FR	40.13	43.77	40.29	41.01
	0.9844	**0.9919**	**0.9835**	**0.9866**
Prop	**41.38**	**45.29**	**41.28**	**42.21**
	0.9875	**0.9937**	**0.9858**	**0.9890**

(a) Kodak Database

	R	G	B	C
Color-KSVD	34.97	37.53	34.08	35.26
	0.9435	0.9621	0.9136	0.9397
MSG	34.42	36.67	33.39	34.75
	0.9397	0.9630	0.9022	0.9349
RI	36.12	39.96	35.37	36.51
	0.9594	**0.9794**	**0.9399**	**0.9596**
MLRI	36.59	39.99	35.40	36.77
	0.9624	**0.9802**	**0.9391**	**0.9606**
SSCAD	35.44	39.12	34.02	35.62
	0.9588	0.9763	0.9287	0.9546
ARI	**37.41**	40.69	**36.05**	**37.51**
	0.9665	**0.9826**	**0.9431**	**0.9641**
MLRI+wei	36.72	40.20	35.58	36.92
	0.9637	**0.9809**	**0.9415**	**0.9620**
DDR	37.09	40.30	35.61	37.14
	0.9633	**0.9812**	**0.9381**	**0.9609**
FR	**37.47**	**40.96**	35.80	**37.46**
	0.9656	**0.9828**	**0.9401**	**0.9629**
Prop	37.32	40.27	35.93	**37.42**
	0.9621	**0.9792**	0.9379	**0.9598**

(b) IMAX Database

TABLE 5.6: Results of Demosaicking Experiments

respectively. The column C corresponds to the average CPSNR/CSSIM values obtained by different algorithms.

In the case of Kodak database, Table 5.6(a) shows that the proposed JCS-GMM demosaicking algorithm achieves significantly better performance when compared with various state of the art demosaicking algorithms. Note that the proposed GMM based algorithm achieves an improvement of 1.95 dBs in CPSNR and 0.0034 in CSSIM values when compared with the sparsity based Color-KSVD algorithm. The proposed algorithm achieves a PSNR improvement of 1.25, 1.45 and 0.95 dBs on Red, Green and Blue channels when compared with second best performing algorithm, i.e., DDR. In terms of SSIM values, the proposed method achieves an improvement of 0.0030 0.0017 and 0.0021 respectively on Red, Green and Blue channels when compared with the DDR algorithm. The proposed algorithm achieves an improvement of 1.17 dBs in CPSNR and 0.0022 in CSSIM values when compared with DDR.

Table 5.6(b) shows that the proposed JCS-GMM demosaicking algorithm achieves superior performance in the case of IMAX database, when compared with various demosaicking algorithms except ARI and FR. In the case of IMAX database, the ARI algorithm achieves the best performance in terms of CPSNR and CSSIM values. The FR algorithm achieves the best performance on Red and Green channels in terms of PSNR measure, whereas, ARI algorithm achieves the best performance on the Blue channel. In terms of SSIM values, ARI algorithm achieves the best performance on Red and Blue channels, and FR algorithm achieves the best performance on Green channel. The ARI algorithm outperforms the proposed demosaicking algorithm by 0.09 dBs in terms of CPSNR, and 0.0043 in terms of CSSIM. The proposed method outperforms FR algorithm in terms of PSNR measure on Green channel, and is quite close to FR algorithm in terms of CPSNR. The performance of the proposed method is very close to that of the best performing ARI algorithm in terms of CPSNR and CSSIM measures. The IMAX database contains several images with saturated color regions [142], and is significantly different from the color image database used for learning the JCS-GMM parameters. This might be a possible reason for the slightly inferior performance of the proposed algorithm in the case of IMAX database.

5.5 Conclusions

In this chapter, the idea of a Joint Color Space GMM prior has been proposed for exploiting inter channel correlations in color image restoration problems. The JCS-GMM prior characterizes concatenated color patches from Red, Green and Blue channels of a color image. The parameters of the JCS-GMM prior are learnt from a large database of clean concatenated color patches. A JCS-GMM denoising algorithm and a JCS-GMM demosaicking algorithm have been proposed for addressing color image denoising and demosaicking problems. A controlled GMM adaptation scheme has also been proposed to adapt the parameters of the externally learnt JCS-GMM prior to the specific image being restored. It was demonstrated through experiments that the proposed algorithms achieve superior performance when compared with various state of the art color image restoration algorithms.

Chapter 6

Conclusions and Future Research

The work reported in this thesis is motivated by the close connections between image restoration algorithms exploiting GMM priors and the restoration algorithms exploiting sparsity priors for characterizing natural image patches. Several widely popular sparsity based algorithms have been shown to produce state of the art results in many natural image restoration problems. Recent works from the image processing literature have demonstrated that the image restoration algorithms exploiting Gaussian Mixture Models for characterizing natural image patches achieve superior restoration performance and faster speed of operation when compared with several sparsity based algorithms. These works have proposed GMM analogues corresponding to many of the sparsity based algorithms addressing different image restoration problems. The work reported in this thesis further extends the investigation on GMM analogues corresponding to the sparsity based algorithms, and presents several novel results on a class of image restoration problems featured by the abundance of highly correlated natural image patches. Specifically, we proposed the GMM analogues of previously studied sparsity based algorithms addressing single image super resolution and color image restoration problems. In the case of SISR, sparsity based algorithms exploit sparse representations for capturing the structures available in highly correlated pairs of HR and LR patches, whereas, in the case of color image restoration problems, sparsity based algorithms exploit sparse representations for capturing inter channel correlations through jointly characterizing the highly correlated patches

from different color channels of a color image. We proposed two different algorithms addressing SISR problem, and a color image restoration framework addressing color image denoising and demosaicking problems.

In chapter 3, we proposed an SISR algorithm based on the Joint GMM method which exploits a GMM prior for jointly characterizing HR and the corresponding LR patches. The Joint GMM method can be interpreted as the GMM analogue of sparsity based ScSR algorithm which exploits sparse representation of HR-LR patch pairs over joint dictionaries. The proposed method predicts the HR patches corresponding to the input LR patches by exploiting the HR-LR patch correlations captured by the Joint GMM parameters learnt from a large training database of HR-LR patch pairs. Given an input LR patch, the Joint Gaussian model whose LR part best describes the given patch is selected from the mixture, and the HR patch estimate is computed by using a Wiener filter derived from the covariance matrix of the selected Gaussian model. The proposed SR reconstruction algorithm is a fast, piecewise linear filter involving a non linear Gaussian model selection and K Wiener filters, where, K is the total number of Gaussian models that constitute the mixture. The proposed Joint GMM method achieves superior performance, and is much faster, when compared with the sparsity based ScSR algorithm. The proposed method achieves competitive performance and speed of operation when compared with state of the art patch based SISR algorithms.

The major drawback of the Joint GMM method is the computationally heavy EM algorithm used for learning the Joint GMM parameters. The learning time required by the iterative EM algorithm is significantly large, especially in the case of large magnification factors. The Joint GMM method makes the EM algorithm computationally tractable by restricting the LR patch size to a few pixels. However, small LR patches restricts the Joint GMM parameters from capturing the necessary HR-LR correlations, which in turn produces a performance drop, especially in the case of large magnification factors. The SISR algorithm based on GMM based regression method proposed in chapter 4 addresses the computational bottleneck associated with the Joint GMM method. The proposed GMM based regression method assumes that the space of LR patches can be partitioned into several clusters, and all the LR patches from a given cluster are mapped to the corresponding HR patches through a common linear transformation. Based on this assumption,

the GMM based regression method exploits a generic GMM prior characterizing natural image patches to partition a given set of LR training patches into K clusters, where, K is the total number of Gaussian models from the GMM. The transformation matrix that maps the LR patches from a given cluster to the corresponding HR patches is learnt for each of the clusters by using a computationally cheap regression method. The SR reconstruction algorithm of the proposed method associates a given input LR patch to the cluster corresponding to the Gaussian model that best describes the given patch, and computes the HR patch estimate by applying the linear transformation corresponding to the selected cluster. The computationally cheaper regression method used for estimating the transformation matrices considerably reduces the learning effort when compared with that of EM algorithm used by the Joint GMM method. The proposed GMM based regression method can be seen as the GMM analogue of A+ algorithm which exploits a similar regression method based on an overcomplete dictionary providing sparse representations for the LR patches. The proposed method achieves superior performance in the case of large magnification factors when compared with the Joint GMM method, and achieves competitive performance in the case of smaller magnification factors.

In chapter 5, we proposed a Joint Color Space GMM (JCS-GMM) method for addressing color image restoration problems. The JCS-GMM method exploits a GMM prior for jointly characterizing the highly correlated patches from different color channels of a color image. The parameters of the JCS-GMM prior learnt from a large database of natural color image patches capture the strong inter channel correlations typically observed between the patches from Red, Green and Blue channels of a color image. We proposed the JCS-GMM denoising and the JCS-GMM demosaicking algorithms for addressing color image denoising and demosaicking problems. The proposed JCS-GMM denoising algorithm computes the denoised estimate of a color image by using an iterative method which alternates several times between Patch Estimation, Image Estimation and GMM Adaptation steps. The JCS-GMM denoising algorithm starts with an initial JCS-GMM prior learnt from an external database, and in every iteration, the parameters of the GMM prior are adapted to the specific image being denoised. The proposed JCS-GMM denoising algorithm is demonstrated to achieve superior denoising performance when compared with state of the art color image denoising algorithms. Similar to the JCS-GMM denoising algorithm, the

proposed JCS-GMM demosaicking algorithm exploits an iterative method to compute the demosaicked estimate of a color image from the mosaicked input image. In every iteration, the JCS-GMM demosaicking algorithm alternates between Patch Estimation and GMM Adaptation steps. The demosaicking performance of the proposed algorithm is evaluated on Kodak and IMAX databases commonly used for evaluating demosaicking algorithms. The proposed JCS-GMM demosaicking algorithm achieves superior performance in the case of Kodak database, and competitive performance in the case of IMAX database, when compared with state of the art demosaicking algorithms.

6.1 Scope of Future Research

A brief outline of the scope for future research related to the work reported in this thesis is given below.

- **Future Work on SISR:** The Joint GMM method proposed as a GMM analogue of the sparsity based ScSR algorithm achieves superior performance and speed of operation when compared with ScSR. The proposed method is already competitive with state of the art patch based SISR algorithms. The original ScSR algorithm was later improved in a series of works [5, 56, 88, 7] by introducing several modifications aiming to improve the SR performance. The future work on SISR targets to investigate the feasibility of introducing similar modifications in the proposed Joint GMM method, leading to further improvements in SR performance and speed of operation. The GMM based regression method proposed in chapter 4 is an initial attempt in this line of research, as it addresses the computational bottleneck associated with the EM algorithm used by the Joint GMM method, and achieves superior performance in the case of magnification by a factor of 3.

- **Future Work on Color Image Restoration:** We have demonstrated that the JCS-GMM method proposed in chapter 5 achieves impressive results in color image denoising and demosaicking problems by exploiting inter channel correlations typically observed in natural color images. The super resolution of color images is another interesting venue of research in the area of color image restoration. Though

the performance of most of the previously proposed SISR algorithms is demonstrated by conducting SR experiments on color images, these algorithms (except a few such as [105]) operate only on the luminance channel of a color image, while the chrominance channels are generally magnified by using simple techniques such as bicubic interpolation. This approach is motivated by the fact that the Human Visual System is more sensitive to luminance variations than the variations in color information. However, in the case of color images, exploiting inter channel correlations can certainly improve the overall SR performance. A recent work [147] has extended the sparsity based ScSR algorithm into a color image SR algorithm by explicitly incorporating inter channel correlations as a constraint in the objective functions used for learning the joint dictionary and computing the sparse representations. Potential future work on color image restoration problems can focus on extending the JCS-GMM method into the case of color image super resolution.

- **Future Work on Computational Imaging:** The recently emerging area of computational photography has several interesting problems such as High Dynamic Range (HDR) hallucination and light field reconstruction/restoration, in which, the potential of GMM priors in characterizing correlated natural image patches can be explored. The HDR hallucination problem of estimating an HDR image from one or more Low Dynamic Range (LDR) images [148, 149, 150] has much in common with the natural image SR problem. The SR problem is to hallucinate the high frequency details missing from the observed LR images. Though learning based SR algorithms is an active research topic in the image processing literature, HDR hallucination algorithms which learn to predict HDR images by exploiting a database of several HDR-LDR pairs have not been extensively studied in the literature. Some of the recent attempts [151, 152] have exploited convolutional and deep neural networks for learning the mapping from LDR to HDR images. Motivated by the similarities between HDR hallucination and SR problems, our future research targets to address the HDR hallucination problem by using techniques similar to the GMM based SR algorithms proposed in chapter 3 and 4.

  The discretized *light field* [153, 154] of a scene can be interpreted as a stack of several snapshots of the scene captured from different viewpoints. The stack of images

obviously contains several highly correlated patches as these images are the snapshots of the same scene. Light field reconstruction and restoration algorithms exploiting sparsity priors for characterizing light field patches have been previously studied [155] in the literature. As a future direction of research, we plan to investigate the potential of GMMs in characterizing light field patches in light field reconstruction and restoration problems.

Appendix A

EM Algorithm for Learning the GMM Parameters

The Expectation Maximization (EM) algorithm [108, 109] is widely popular in statistics and machine learning literature. It is used to compute approximate solutions to several Maximum Likelihood (ML) estimation problems whose solutions cannot be obtained in closed form. In our work, the EM algorithm is used to compute the estimate $\hat{\boldsymbol{\Theta}} = \left\{\hat{\omega}_k, \hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\Sigma}}_k\right\}_{k=1}^{K}$ of the set of parameters of a Gaussian Mixture Model (GMM) by using a given set of training vectors $\{\mathbf{t}_i \in \mathbb{R}^n\}_{i=1}^{M}$ assumed to be drawn from the GMM. The EM algorithm iteratively alternates between the Expectation (E) and Maximization (M) steps to compute the parameter estimates corresponding to a local maximum of the function that maximizes the likelihood of the given set of training vectors. The construction of training database varies according to the specific image restoration problem being addressed. In the case of SISR algorithm based on Joint GMM method proposed in chapter 3, the training database corresponds to the set of vectors obtained by concatenating HR and LR patch vectors. The GMM based regression method proposed in chapter 4 exploits a GMM prior whose parameters are estimated from a set of vectorized natural image patches. In the case of color image restoration framework proposed in chapter 5, the parameters of the Joint Color Space GMM (JCS-GMM) prior are estimated from the training vectors obtained by concatenating the corresponding patch vectors from Red,

Green, and Blue channels of color images. The EM algorithm used for learning the GMM parameters is given below.

EM Algorithm for Learning GMM Parameters

1: **Input:** Training database $\{\mathbf{t}_i\}_{i=1}^{M}$, number of Gaussian models K, stopping threshold δ, maximum number of iterations L.

2: **Initialization:** Set the iteration counter to $l = 1$. The overall log likelihood value $R^{'}$ of the set of training vectors is initialized to a large negative value. The initial estimate $\hat{\mathbf{\Theta}} = \left\{\hat{\omega}_k, \hat{\boldsymbol{\mu}}_k, \hat{\mathbf{\Sigma}}_k\right\}_{k=1}^{K}$ of the GMM parameters are obtained by randomly partitioning the training vectors into K disjoint clusters $\{\mathcal{C}_k\}_{k=1}^{K}$ and computing,

$$\hat{\omega}_k = \frac{|\mathcal{C}_k|}{M}, \;\; \hat{\boldsymbol{\mu}}_k = \frac{1}{|\mathcal{C}_k|}\sum_{j\in\mathcal{C}_k}\mathbf{t}_j,$$

$$\hat{\mathbf{\Sigma}}_k = \frac{1}{|\mathcal{C}_k|}\sum_{j\in\mathcal{C}_k}\left(\mathbf{t}_j - \hat{\boldsymbol{\mu}}_k\right)\left(\mathbf{t}_j - \hat{\boldsymbol{\mu}}_k\right)^T.$$

3: **E-Step:** For $k = 1, 2, .., K$, compute the likelihood values $\{\gamma_{ik}\}_{i=1}^{M}$ as,

$$\gamma_{ik} = \frac{\hat{\omega}_k \Phi\left(\mathbf{t}_i \left| \hat{\boldsymbol{\mu}}_k, \hat{\mathbf{\Sigma}}_k\right.\right)}{\sum_{j=1}^{K}\hat{\omega}_j \Phi\left(\mathbf{t}_i \left| \hat{\boldsymbol{\mu}}_j, \hat{\mathbf{\Sigma}}_j\right.\right)}, \;\; i = 1, ..., M,$$

and compute,

$$n_k = \sum_{i=1}^{M}\gamma_{ik}$$

4: **M-Step:** For $k = 1, 2, ..., K$, update the estimates $\hat{\mathbf{\Theta}} = \left\{\hat{\omega}_k, \hat{\boldsymbol{\mu}}_k, \hat{\mathbf{\Sigma}}_k\right\}_{k=1}^{K}$ of the GMM parameters as,

$$\hat{\omega}_k = \frac{n_k}{M}, \;\; \hat{\boldsymbol{\mu}}_k = \frac{1}{n_k}\sum_{i=1}^{M}\gamma_{ik}\mathbf{t}_i,$$

$$\hat{\mathbf{\Sigma}}_k = \frac{1}{n_k}\sum_{i=1}^{M}\gamma_{ik}\left(\mathbf{t}_i - \hat{\boldsymbol{\mu}}_k\right)\left(\mathbf{t}_i - \hat{\boldsymbol{\mu}}_k\right)^T$$

5: **Covariance Regularization:** For $k = 1, 2, ..., K$, apply a simple eigenvalue regularization to prevent the singularity of covariance matrices

$$\hat{\mathbf{\Sigma}}_k = \hat{\mathbf{\Sigma}}_k + \epsilon\mathbf{I}, \text{ where, } \epsilon = 10^{-6}$$

6: **Stopping Criterion:** Compute the overall log likelihood value R of the set of training vectors as,

$$R = \frac{1}{M}\sum_{i=1}^{M}\log\left(\sum_{k=1}^{K}\hat{\omega}_k \Phi\left(\mathbf{t}_i \left| \hat{\boldsymbol{\mu}}_k, \hat{\mathbf{\Sigma}}_k\right.\right)\right)$$

If $|R' - R| < \delta$ or $l > L$, goto step 7. Else, set $R' = R$, $l = l + 1$ and goto step 3.

7: **Return:** The estimate $\hat{\mathbf{\Theta}} = \left\{\hat{\omega}_k, \hat{\boldsymbol{\mu}}_k, \hat{\mathbf{\Sigma}}_k\right\}_{k=1}^{K}$ of the set of GMM parameters.

Appendix B

Conditional Expectation of a Jointly Gaussian Random Variable

Let the vector $\mathbf{v} = \left[\mathbf{p}^T, \mathbf{q}^T\right]^T \in \mathbb{R}^n$ be jointly Gaussian with mean $\boldsymbol{\mu} \in \mathbb{R}^n$ and covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{n \times n}$. We can use block matrix notations to rewrite $\boldsymbol{\mu} \in \mathbb{R}^n$ and $\boldsymbol{\Sigma} \in \mathbb{R}^{n \times n}$ as,

$$\boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_p \\ \boldsymbol{\mu}_q \end{bmatrix} \tag{B.1}$$

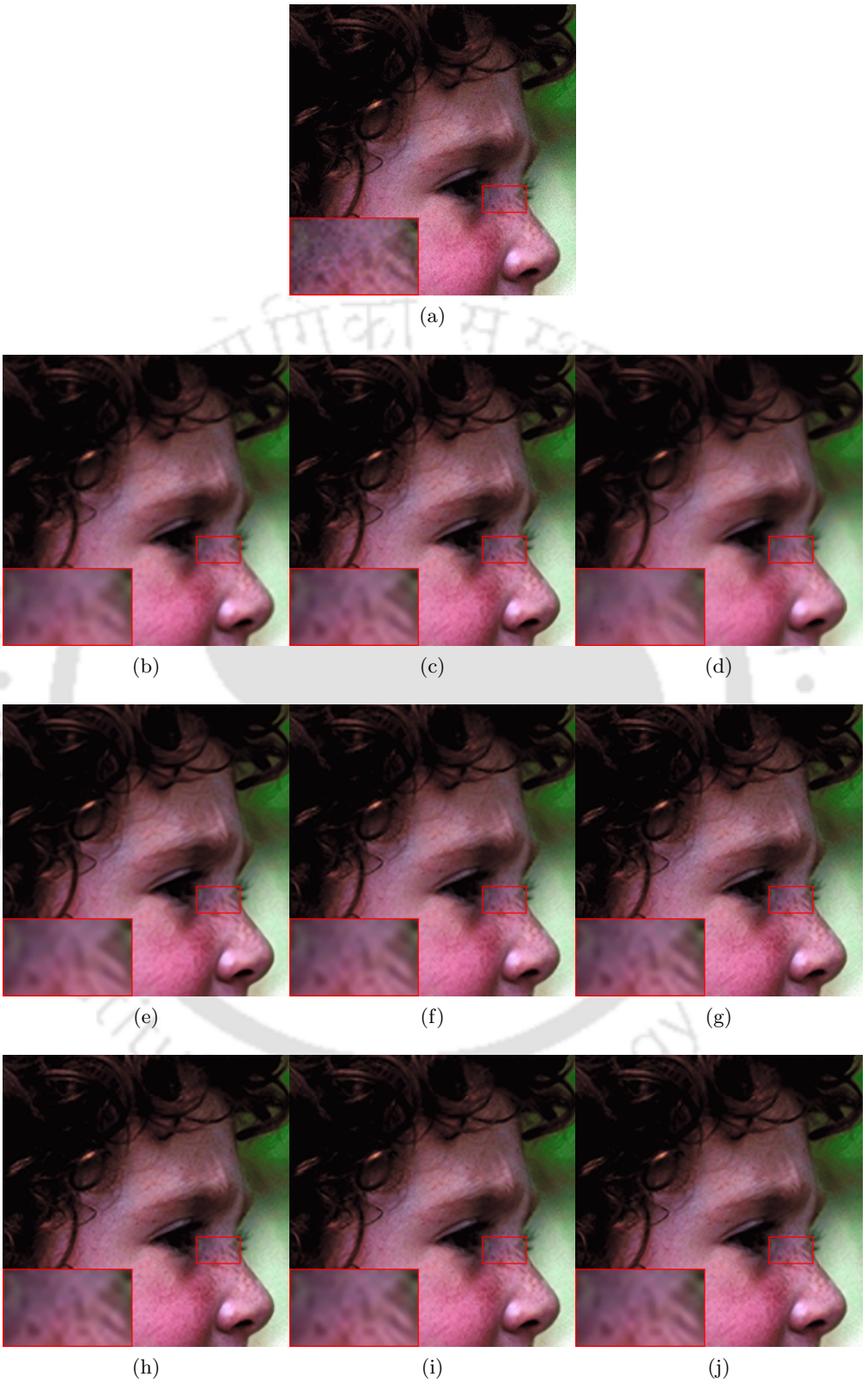
$$\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_p & \boldsymbol{\Sigma}_{pq} \\ \boldsymbol{\Sigma}_{pq}^T & \boldsymbol{\Sigma}_q \end{bmatrix} \tag{B.2}$$

where, $\boldsymbol{\mu}_p \in \mathbb{R}^{n_1}$, $\boldsymbol{\mu}_q \in \mathbb{R}^{n_2}$, $\boldsymbol{\Sigma}_p \in \mathbb{R}^{n_1 \times n_1}$, $\boldsymbol{\Sigma}_q \in \mathbb{R}^{n_2 \times n_2}$, $\boldsymbol{\Sigma}_{pq} \in \mathbb{R}^{n_1 \times n_2}$, and, $n_1 + n_2 = n$. Our aim is to obtain the conditional expectation $\mathbb{E}_{\mathbf{p}|\mathbf{q}}$. It is well known [156, 157] that the subvectors $\mathbf{p}$ and $\mathbf{q}$ of $\mathbf{v}$ are Gaussian distributed with parameters $\left(\boldsymbol{\mu}_p, \boldsymbol{\Sigma}_p\right)$ and $\left(\boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q\right)$ respectively, and the conditional distribution $f_{\mathbf{p}|\mathbf{q}}$ is also Gaussian with mean $\boldsymbol{\mu}_p + \boldsymbol{\Sigma}_{pq}\boldsymbol{\Sigma}_q^{-1}\left(\mathbf{q} - \boldsymbol{\mu}_q\right)$ and covariance matrix $\boldsymbol{\Sigma}_p - \boldsymbol{\Sigma}_{pq}\boldsymbol{\Sigma}_q^{-1}\boldsymbol{\Sigma}_{pq}^T$. Thus, the required conditional expectation $\mathbb{E}_{\mathbf{p}|\mathbf{q}}$ is the mean of the conditional distribution $f_{\mathbf{p}|\mathbf{q}}$, i.e.,

$$\mathbb{E}_{\mathbf{p}|\mathbf{q}} = \boldsymbol{\mu}_p + \boldsymbol{\Sigma}_{pq}\boldsymbol{\Sigma}_q^{-1}\left(\mathbf{q} - \boldsymbol{\mu}_q\right) \tag{B.3}$$

Appendix C

Visual Comparison of HR Images Produced by Joint GMM Method



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j)

FIGURE C.1: Results of SR experiment conducted on Child image for magnification factor 2. (a) Original
(b) Bicubic, (c) ScSR, (d) SCDL, (e) BPJDL, (f) SPM, (g) A+, (h) SRF, (i) Prop (K=100), (j) Prop (K=225)

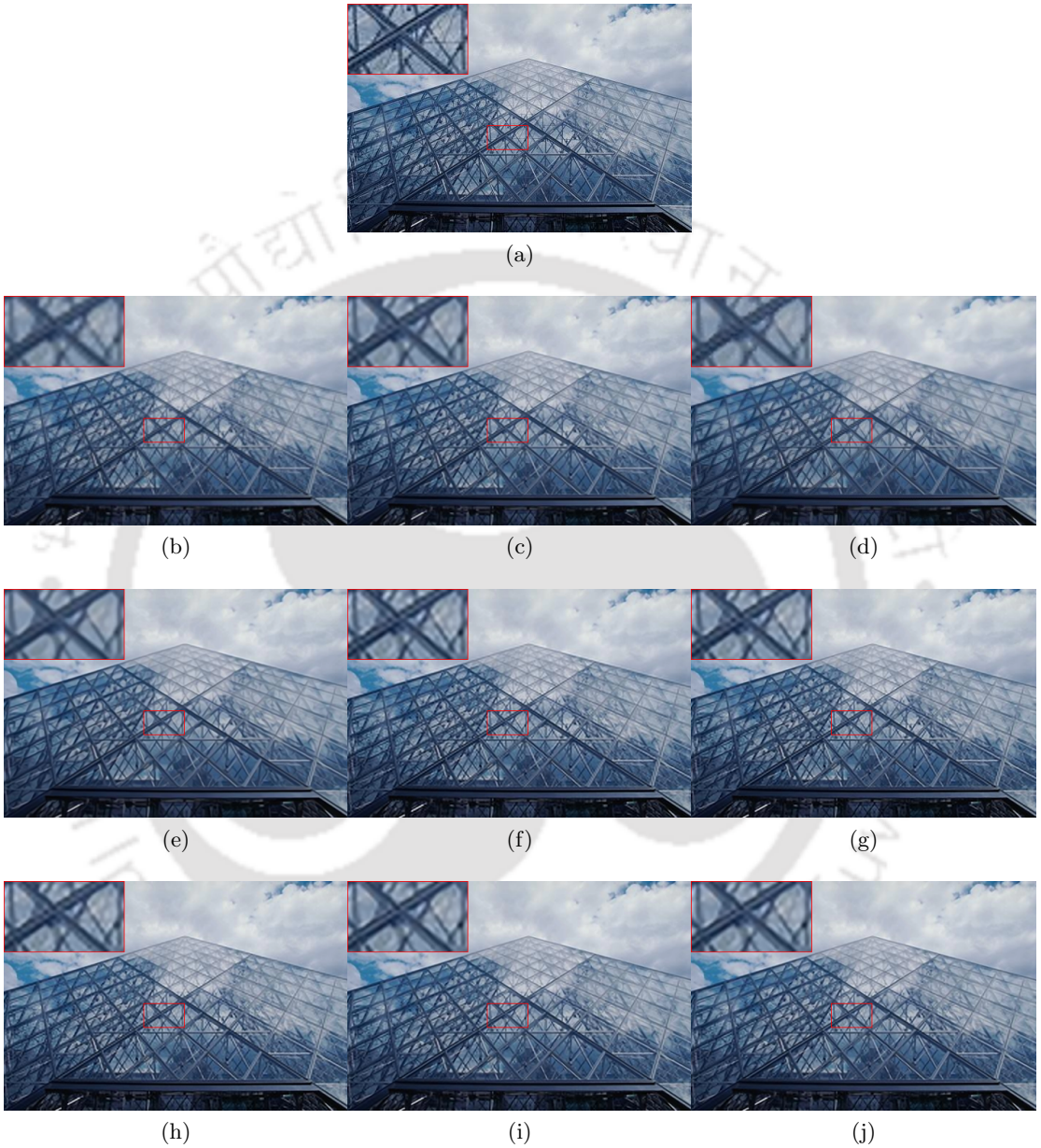
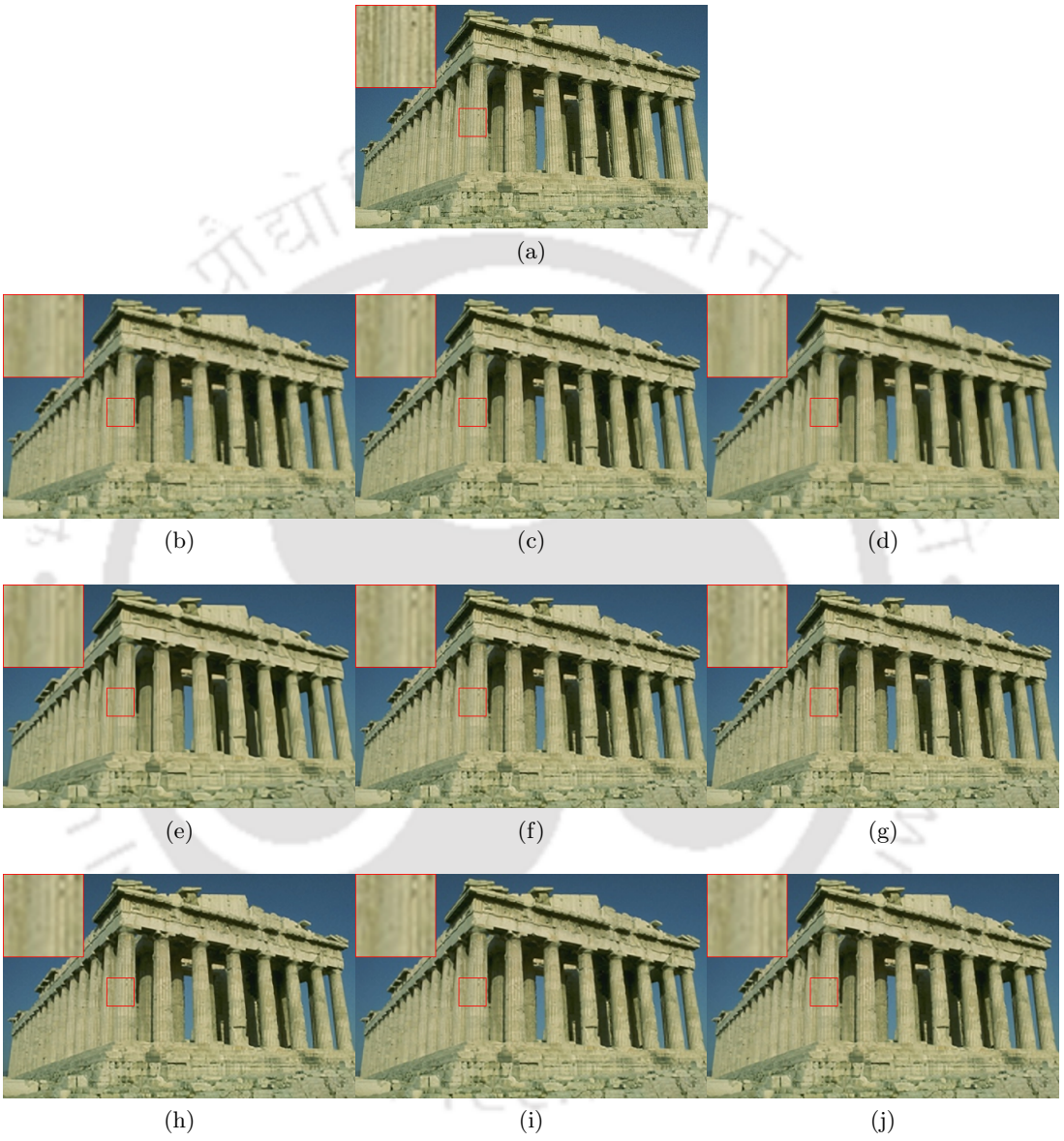
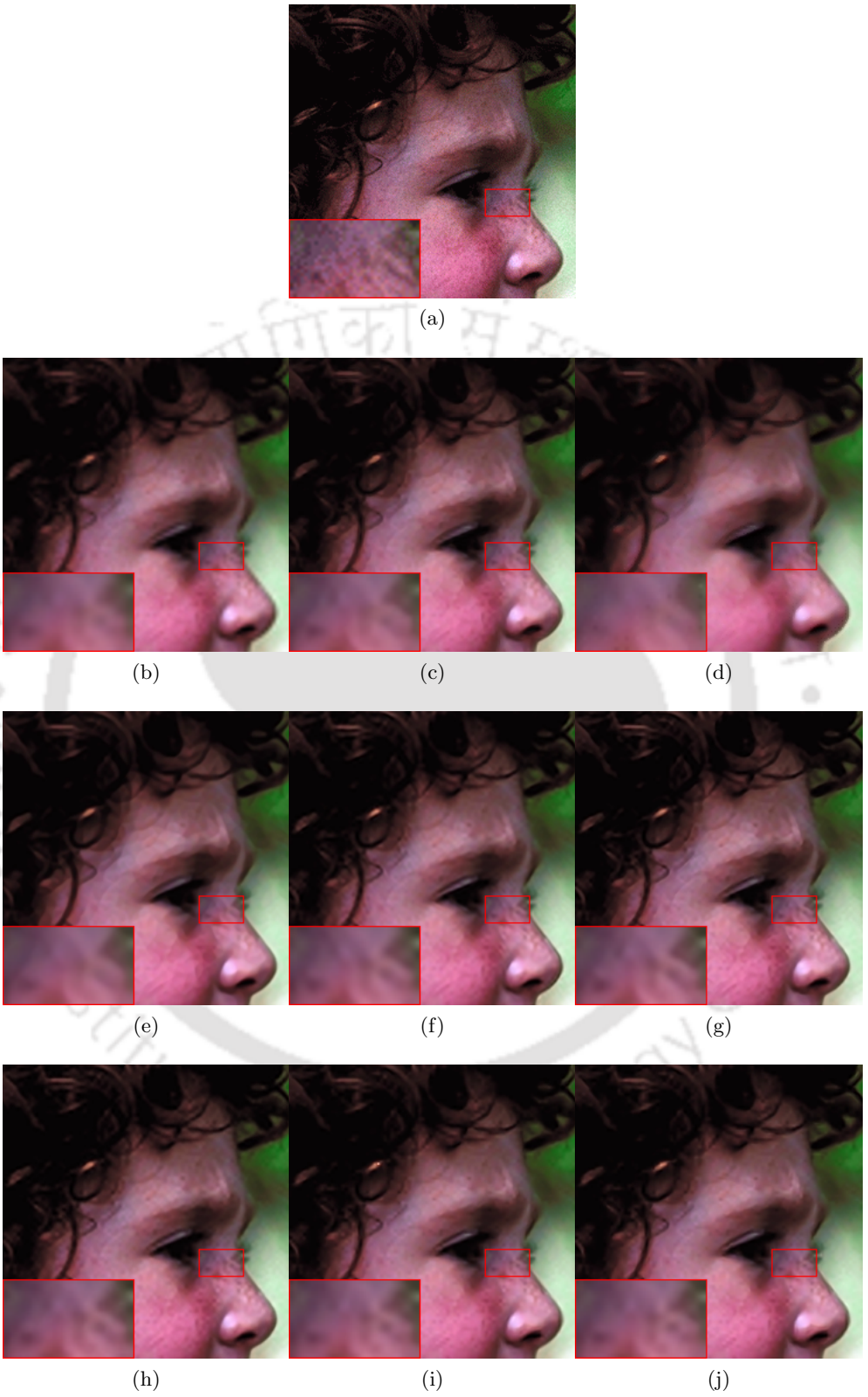


FIGURE C.2: Results of SR experiment conducted on Louvre image for magnification factor 2. (a) Original
(b) Bicubic, (c) ScSR, (d) SCDL, (e) BPJDL, (f) SPM, (g) A+, (h) SRF, (i) Prop (K=100), (j) Prop (K=225)



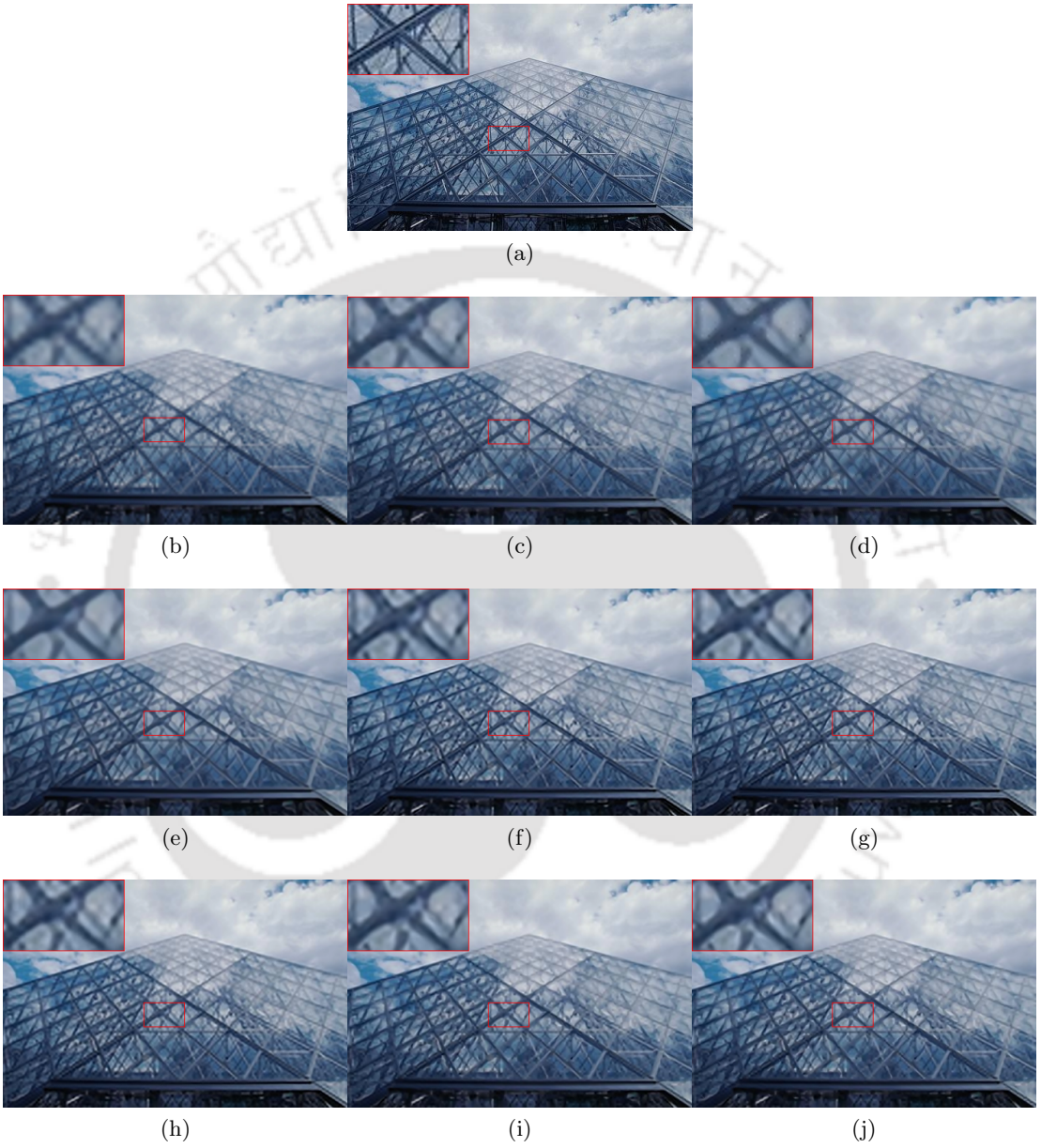
(a) (b) (c) (d) (e) (f) (g) (h) (i) (j)

FIGURE C.3: Results of SR experiment conducted on Parthenon image for magnification factor 2. (a) Original
(b) Bicubic, (c) ScSR, (d) SCDL, (e) BPJDL, (f) SPM, (g) A+, (h) SRF, (i) Prop (K=100), (j) Prop (K=225)



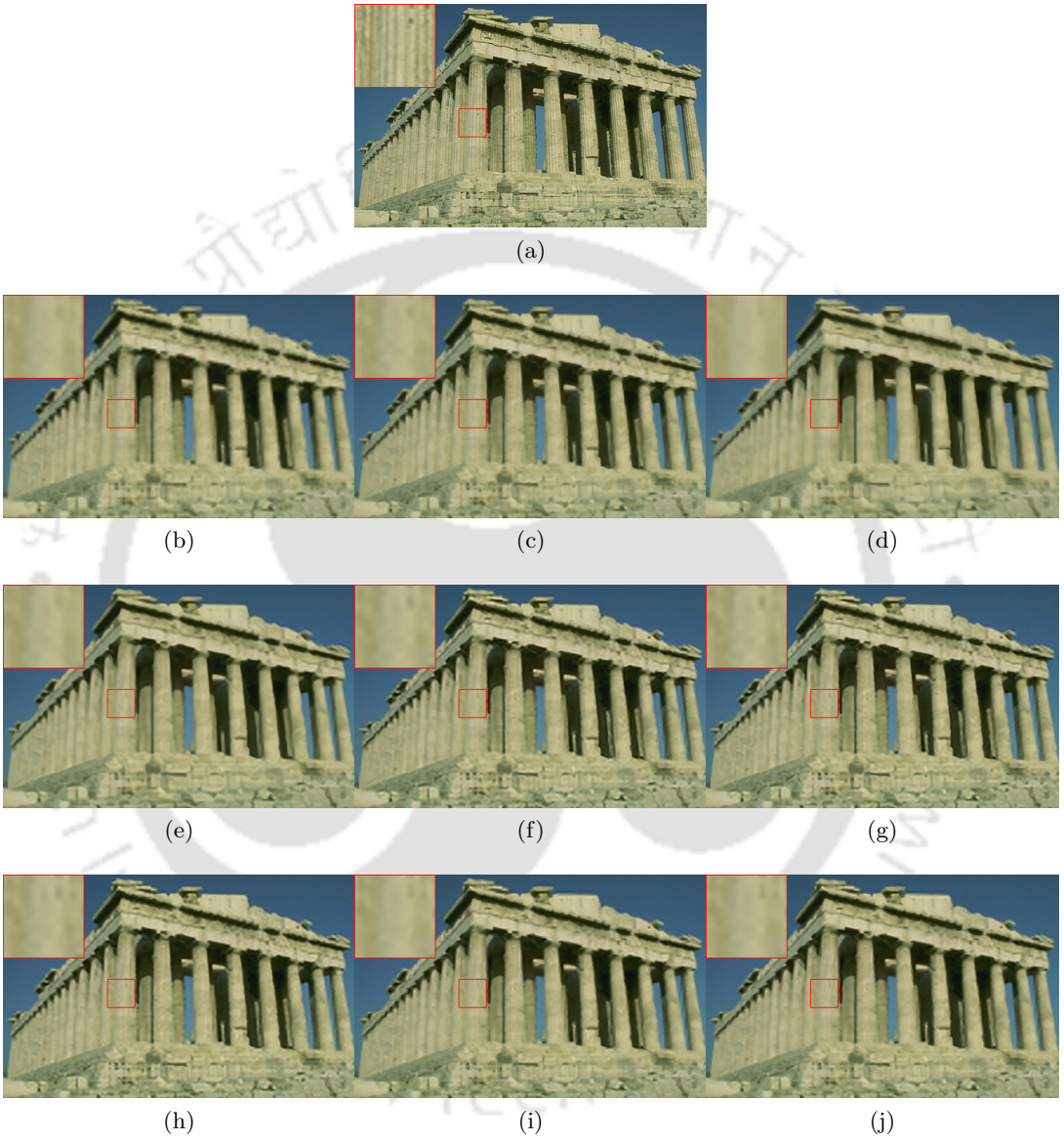
(a) (b) (c) (d) (e) (f) (g) (h) (i) (j)

FIGURE C.4: Results of SR experiment conducted on Child image for magnification factor 3. (a) Original
(b) Bicubic, (c) ScSR, (d) SCDL, (e) BPJDL, (f) SPM, (g) A+, (h) SRF, (i) Prop (K=100), (j) Prop (K=225)



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j)

FIGURE C.5: Results of SR experiment conducted on Louvre image for magnification factor 3. (a) Original
(b) Bicubic, (c) ScSR, (d) SCDL, (e) BPJDL, (f) SPM, (g) A+, (h) SRF, (i) Prop (K=100), (j) Prop (K=225)



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j)

FIGURE C.6: Results of SR experiment conducted on Parthenon image for magnification factor 3. (a) Original
(b) Bicubic, (c) ScSR, (d) SCDL, (e) BPJDL, (f) SPM, (g) A+, (h) SRF, (i) Prop (K=100), (j) Prop (K=225)

Appendix D

Transformation Matrix in GMM Based Regression Method

The transformation matrix $\mathbf{W}_k$ is computed by solving,

$$\mathbf{W}_k = \arg\min_{\mathbf{W}} \|\mathbf{R}_k - \mathbf{W}\mathbf{T}_k\|_F^2 + \lambda \|\mathbf{W}\|_F^2 . \tag{D.1}$$

The minimum is computed differentiating $f(\mathbf{W})$ with respect to $\mathbf{W}$ and setting the result to zero. Thus,

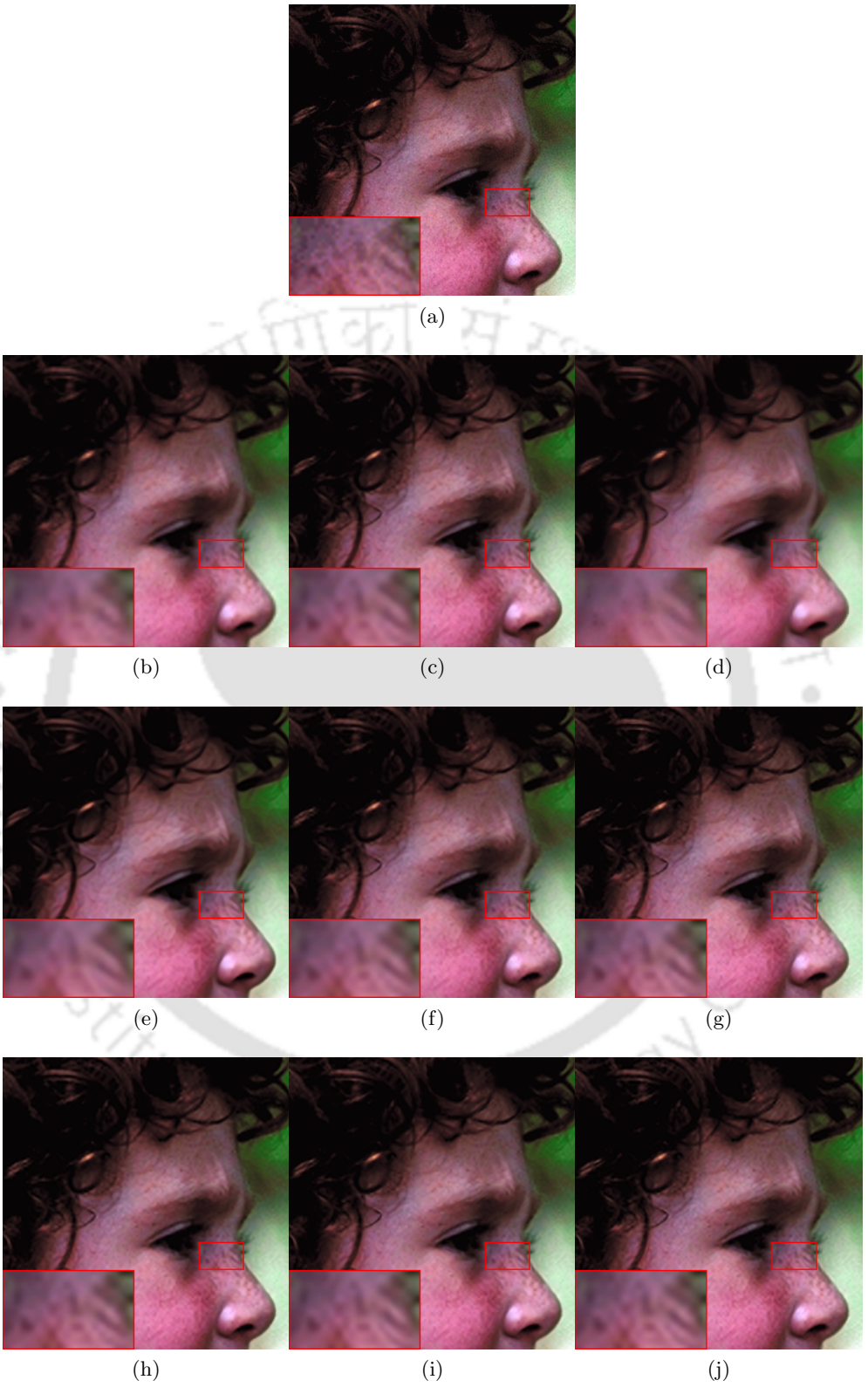
$$\begin{aligned} \frac{\partial}{\partial \mathbf{W}} \|\mathbf{R}_k - \mathbf{W}\mathbf{T}_k\|_F^2 + \lambda \|\mathbf{W}\|_F^2 &= \mathbf{0} \\ \Rightarrow \frac{\partial}{\partial \mathbf{W}} \mathrm{Tr}\left[(\mathbf{R}_k - \mathbf{W}\mathbf{T}_k)(\mathbf{R}_k - \mathbf{W}\mathbf{T}_k)^T\right] + \lambda \frac{\partial}{\partial \mathbf{W}} \mathrm{Tr}\left(\mathbf{W}\mathbf{W}^T\right) &= \mathbf{0} \\ \Rightarrow -\mathbf{R}_k\mathbf{T}_k^T - \mathbf{R}_k\mathbf{T}_k^T + 2\mathbf{W}\mathbf{T}_k\mathbf{T}_k^T + 2\lambda\mathbf{W} &= \mathbf{0} \end{aligned}$$

Thus, we have,

$$\mathbf{W}_k = \mathbf{R}_k\mathbf{T}_k^T \left(\mathbf{T}_k\mathbf{T}_k^T + \lambda\mathbf{I}\right)^{-1} \tag{D.2}$$

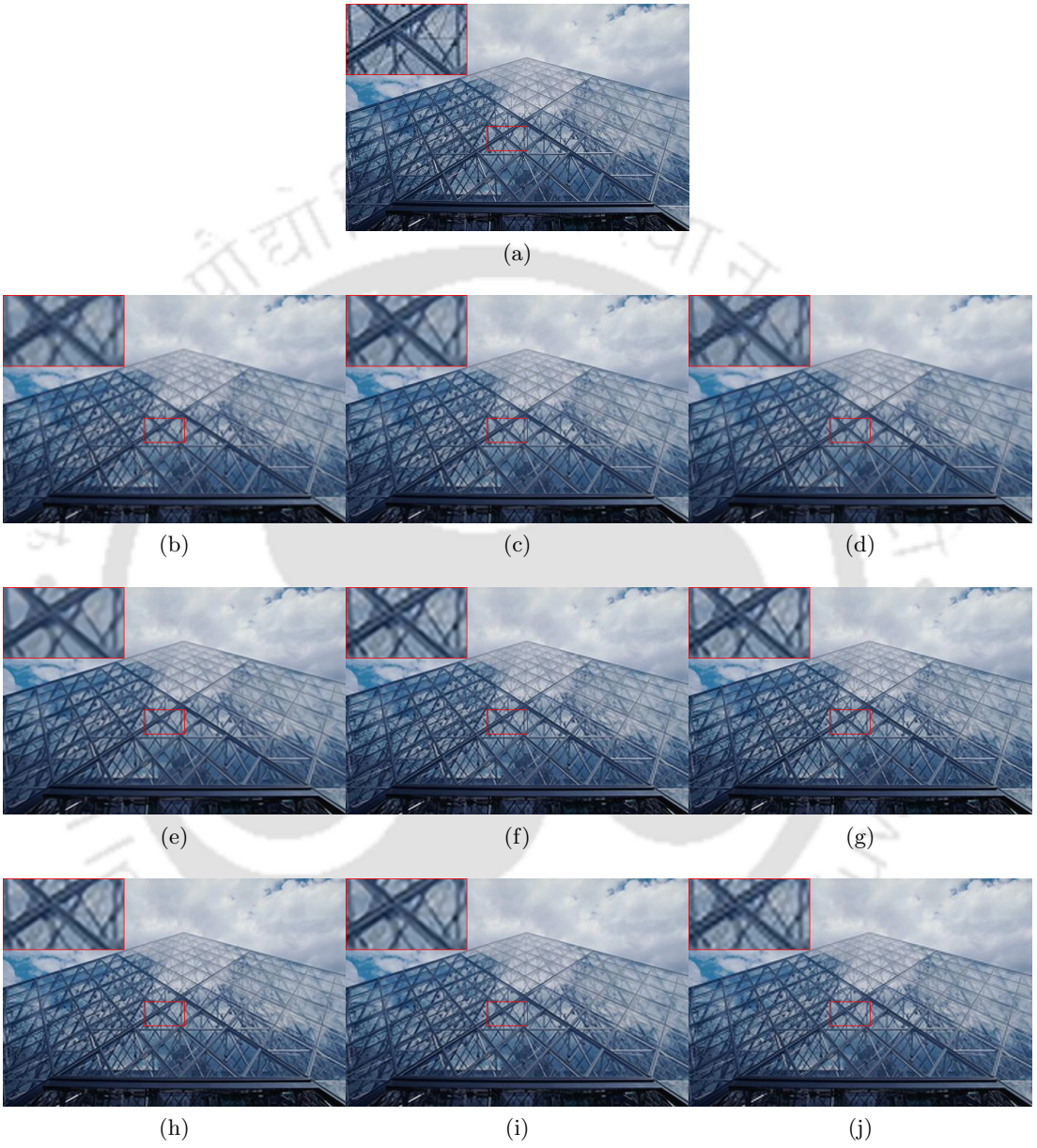
Appendix E

Visual Comparison of HR Images Produced by GMM Based Regression Method



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j)

FIGURE E.1: Results of SR experiment conducted on Child image for magnification factor 2. (a) Original (b) Bicubic, (c) ScSR, (d) SCDL, (e) BPJDL, (f) SPM, (g) A+, (h) SRF, (i) Joint GMM, (j) Prop



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j)

FIGURE E.2: Results of SR experiment conducted on Louvre image for magnification factor 2. (a) Original
(b) Bicubic, (c) ScSR, (d) SCDL, (e) BPJDL, (f) SPM, (g) A+, (h) SRF, (i) Joint GMM, (j) Prop

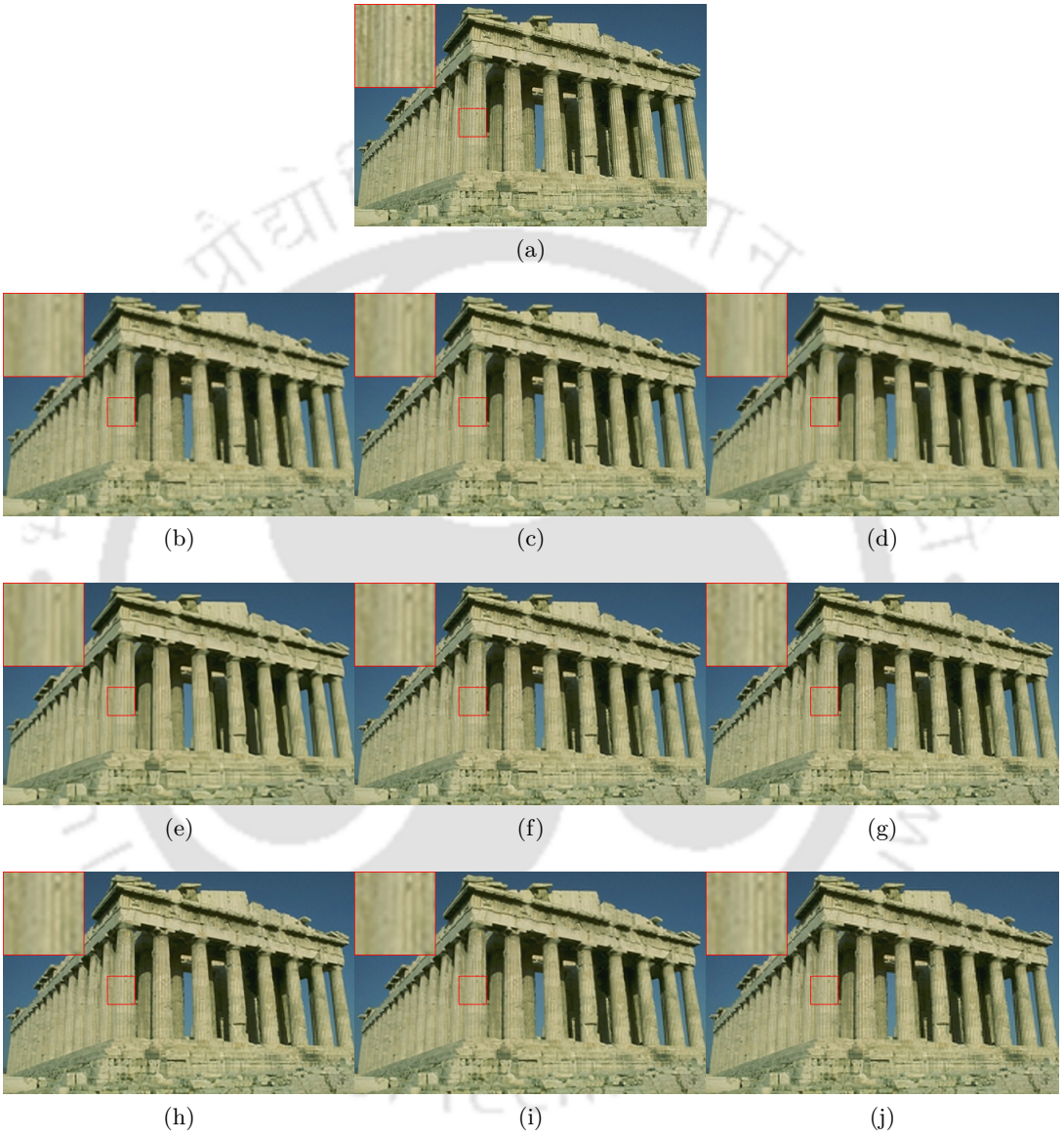
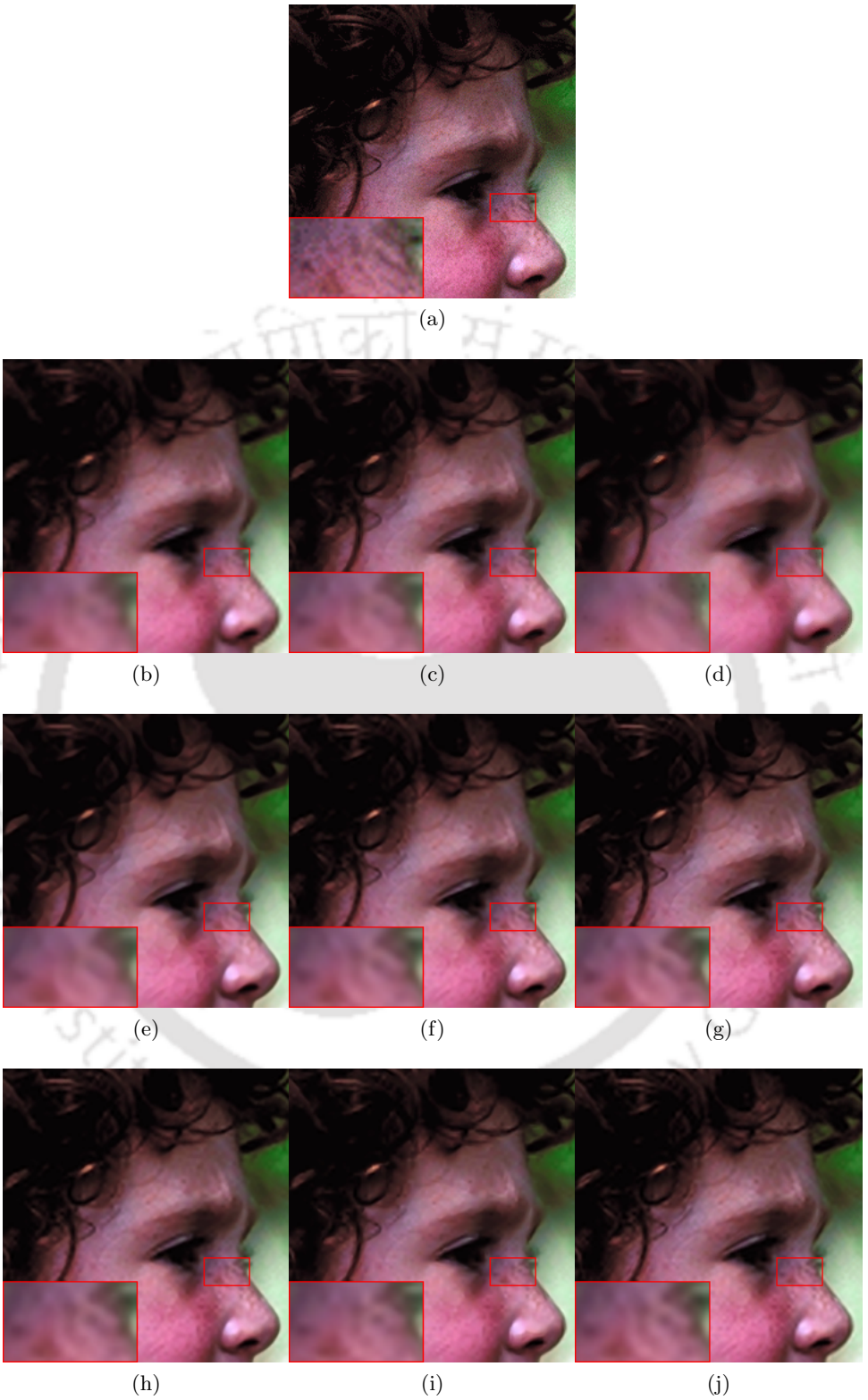
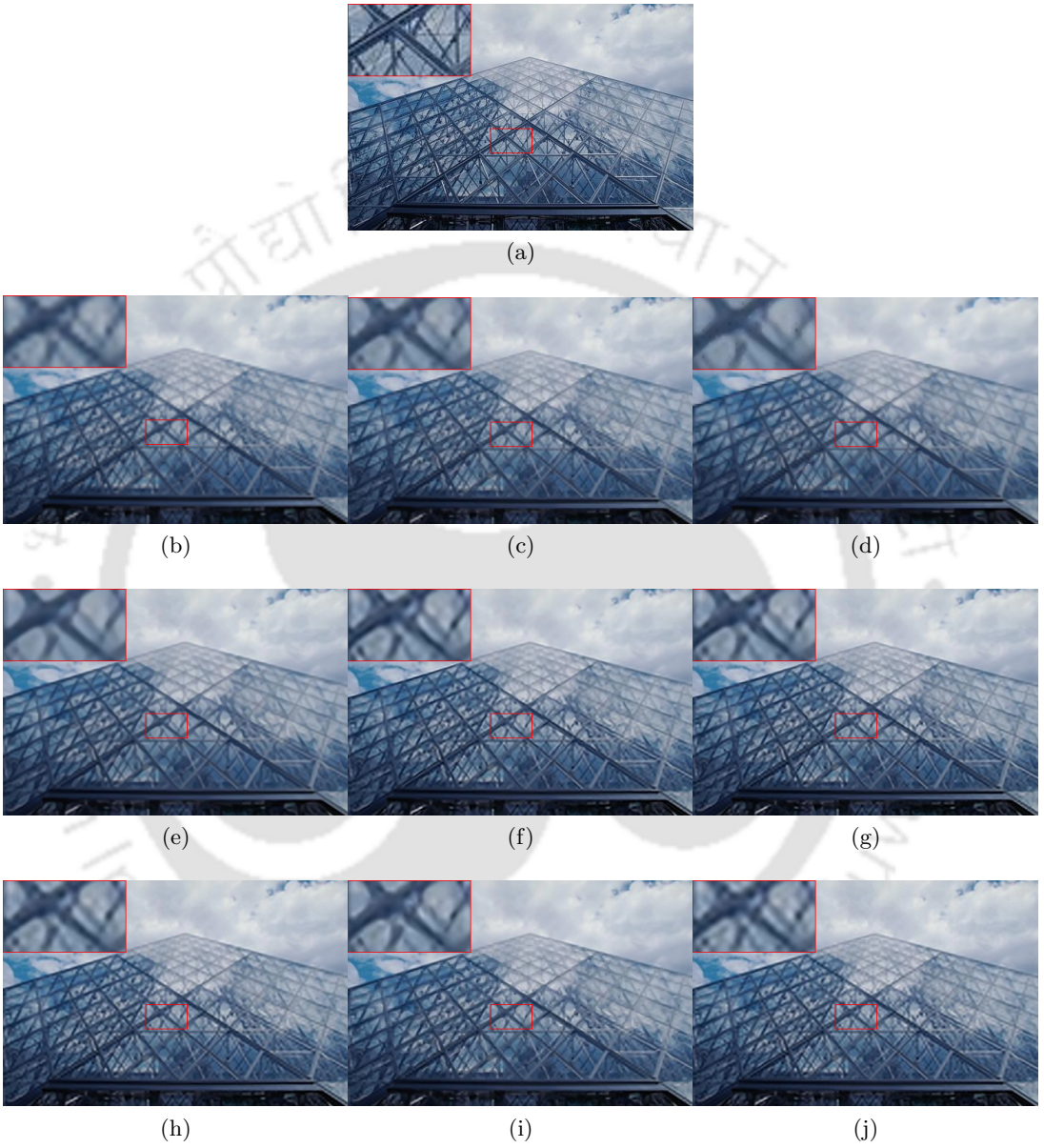


FIGURE E.3: Results of SR experiment conducted on Parthenon image for magnification factor 2. (a) Original
(b) Bicubic, (c) ScSR, (d) SCDL, (e) BPJDL, (f) SPM, (g) A+, (h) SRF, (i) Joint GMM, (j) Prop



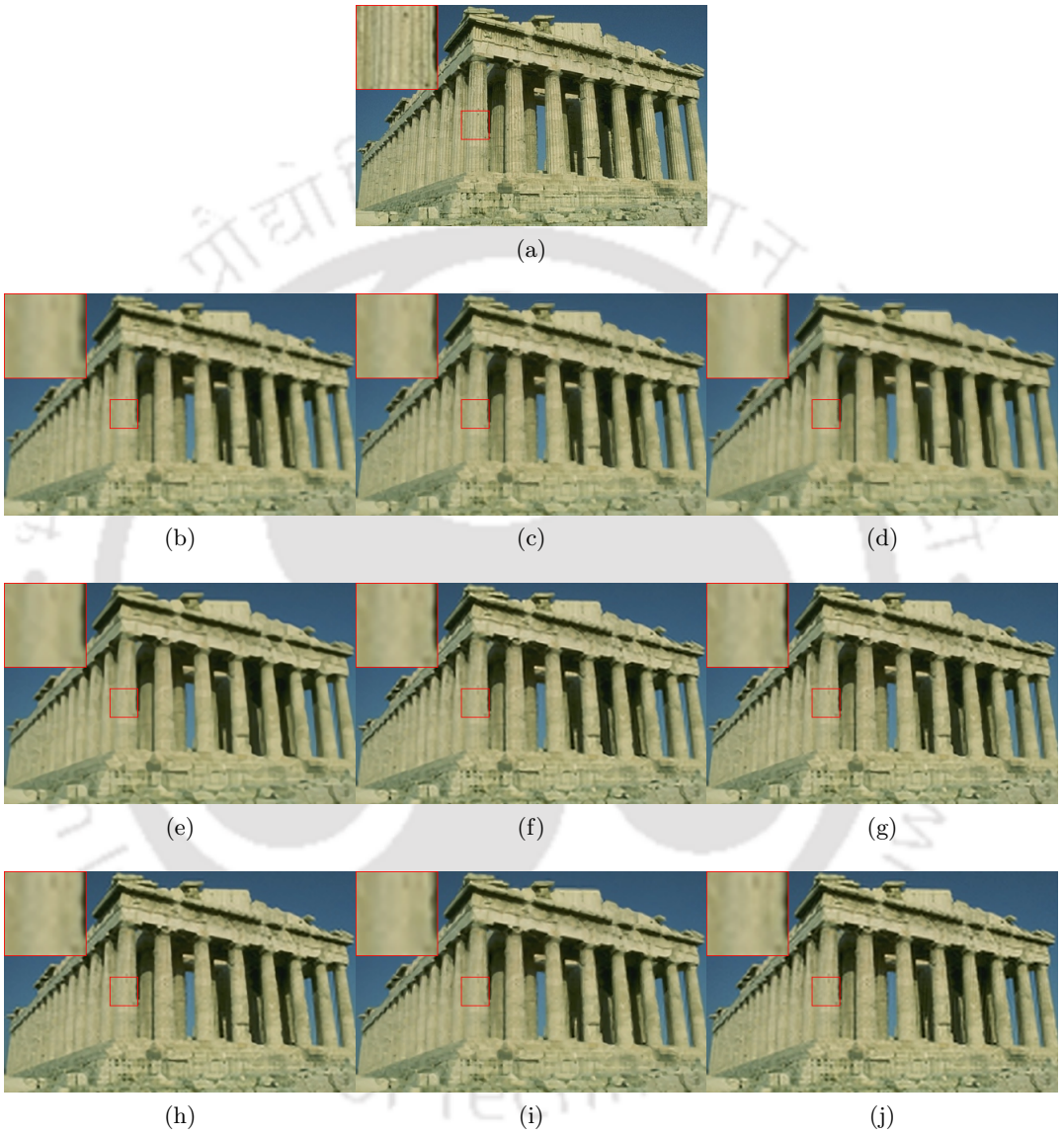
(a) (b) (c) (d) (e) (f) (g) (h) (i) (j)

FIGURE E.4: Results of SR experiment conducted on Child image for magnification factor 3. (a) Original (b) Bicubic, (c) ScSR, (d) SCDL, (e) BPJDL, (f) SPM, (g) A+, (h) SRF, (i) Joint GMM, (j) Prop



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j)

FIGURE E.5: Results of SR experiment conducted on Louvre image for magnification factor 3. (a) Original
(b) Bicubic, (c) ScSR, (d) SCDL, (e) BPJDL, (f) SPM, (g) A+, (h) SRF, (i) Joint GMM, (j) Prop



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j)

FIGURE E.6: Results of SR experiment conducted on Parthenon image for magnification factor 3. (a) Original
(b) Bicubic, (c) ScSR, (d) SCDL, (e) BPJDL, (f) SPM, (g) A+, (h) SRF, (i) Joint GMM, (j) Prop

Appendix F

Patch Estimation in JCS-GMM Denoising

The Patch Estimation step of the JCS-GMM Denoising algorithm computes the estimate $\hat{\mathbf{z}}_i$ of the i-th concatenated color patch as,

$$\hat{\mathbf{z}}_i = \arg\min_{\mathbf{v}} \frac{\beta}{2} \|\mathbf{R}_i\mathbf{u} - \mathbf{v}\|_2^2 + \frac{1}{2}\left[\left(\mathbf{v} - \hat{\boldsymbol{\mu}}_{\hat{k}_i}\right)^T \hat{\boldsymbol{\Sigma}}_{\hat{k}_i}^{-1} \left(\mathbf{v} - \hat{\boldsymbol{\mu}}_{\hat{k}_i}\right)\right], \tag{F.1}$$

where, $\mathbf{u}$ denotes the vector representation of the estimated color image obtained from the previous iteration, $\mathbf{R}_i$ denotes the matrix which extracts the i-th concatenated color patch from $\mathbf{u}$, and, $\hat{\boldsymbol{\mu}}_{\hat{k}_i}$ and $\hat{\boldsymbol{\Sigma}}_{\hat{k}_i}$ denote the mean and covariance matrix of the Gaussian model responsible for the i-th concatenated color patch. Substituting $\mathbf{v} - \hat{\boldsymbol{\mu}}_{\hat{k}_i} = \mathbf{w}$, we rewrite the minimization in (F.1) as,

$$\hat{\mathbf{z}}_i = \arg\min_{\mathbf{w}} \frac{\beta}{2} \left\|\mathbf{R}_i\mathbf{u} - \mathbf{w} - \hat{\boldsymbol{\mu}}_{\hat{k}_i}\right\|_2^2 + \frac{1}{2}\left[\mathbf{w}^T \hat{\boldsymbol{\Sigma}}_{\hat{k}_i}^{-1} \mathbf{w}\right]. \tag{F.2}$$

The minimum is computed by differentiating the RHS of (F.2) with respect to $\mathbf{w}$ and setting the result to zero as,

$$\begin{aligned}
\frac{\partial}{\partial \mathbf{w}}\left[\frac{\beta}{2}\left\|\mathbf{R}_i\mathbf{u}-\mathbf{w}-\hat{\boldsymbol{\mu}}_{\hat{k}_i}\right\|_2^2+\frac{1}{2}\mathbf{w}^T\hat{\boldsymbol{\Sigma}}_{\hat{k}_i}^{-1}\mathbf{w}\right]&=0\\
\Rightarrow \frac{\partial}{\partial \mathbf{w}}\left[\frac{\beta}{2}\left\|\left(\mathbf{R}_i\mathbf{u}-\hat{\boldsymbol{\mu}}_{\hat{k}_i}\right)-\mathbf{w}\right\|_2^2+\frac{1}{2}\mathbf{w}^T\hat{\boldsymbol{\Sigma}}_{\hat{k}_i}^{-1}\mathbf{w}\right]&=0\\
\Rightarrow \frac{\partial}{\partial \mathbf{w}}\left[\frac{\beta}{2}\left(-2\mathbf{w}^T\left(\mathbf{R}_i\mathbf{u}-\hat{\boldsymbol{\mu}}_{\hat{k}_i}\right)+\mathbf{w}^T\mathbf{w}\right)+\frac{1}{2}\mathbf{w}^T\hat{\boldsymbol{\Sigma}}_{\hat{k}_i}^{-1}\mathbf{w}\right]&=0\\
\Rightarrow -\beta\left(\mathbf{R}_i\mathbf{u}-\hat{\boldsymbol{\mu}}_{\hat{k}_i}\right)+\beta\mathbf{w}+\hat{\boldsymbol{\Sigma}}_{\hat{k}_i}^{-1}\mathbf{w}&=0\\
\Rightarrow \left(\mathbf{I}+\frac{1}{\beta}\hat{\boldsymbol{\Sigma}}_{\hat{k}_i}^{-1}\right)\mathbf{w}&=\left(\mathbf{R}_i\mathbf{u}-\hat{\boldsymbol{\mu}}_{\hat{k}_i}\right)\\
\Rightarrow \mathbf{w}=\left(\mathbf{I}+\frac{1}{\beta}\hat{\boldsymbol{\Sigma}}_{\hat{k}_i}^{-1}\right)^{-1}&\left(\mathbf{R}_i\mathbf{u}-\hat{\boldsymbol{\mu}}_{\hat{k}_i}\right)\\
=\beta\hat{\boldsymbol{\Sigma}}_{\hat{k}_i}\left(\beta\hat{\boldsymbol{\Sigma}}_{\hat{k}_i}+\mathbf{I}\right)^{-1}&\left(\mathbf{R}_i\mathbf{u}-\hat{\boldsymbol{\mu}}_{\hat{k}_i}\right) \qquad \text{(F.3)}
\end{aligned}$$

where, (F.3) follows from the matrix identity [158] given as,

$$\left(\mathbf{I}+\mathbf{A}^{-1}\right)^{-1}=\mathbf{A}\left(\mathbf{A}+\mathbf{I}\right)^{-1}.$$

As $\mathbf{w}=\mathbf{v}-\hat{\boldsymbol{\mu}}_{\hat{k}_i}$, the required estimate $\hat{\mathbf{z}}_i$ can be obtained as,

$$\hat{\mathbf{z}}_i=\hat{\boldsymbol{\Sigma}}_{\hat{k}_i}\left(\hat{\boldsymbol{\Sigma}}_{\hat{k}_i}+\frac{1}{\beta}\mathbf{I}\right)^{-1}\left(\mathbf{R}_i\mathbf{u}-\hat{\boldsymbol{\mu}}_{\hat{k}_i}\right)+\hat{\boldsymbol{\mu}}_{\hat{k}_i} \qquad \text{(F.4)}$$

Appendix G

Image Estimation in JCS-GMM Denoising

The Image Estimation step of JCS-GMM Denoising algorithm computes the estimate $\hat{\mathbf{u}}$ of the unknown color image as,

$$\hat{\mathbf{u}} = \arg\min_{\mathbf{u}} \frac{\lambda}{2} \|\mathbf{u} - \mathbf{y}\|_2^2 + \sum_{i=1}^{L} \frac{\beta}{2} \|\mathbf{R}_i \mathbf{u} - \mathbf{z}_i\|_2^2 , \tag{G.1}$$

where, $\mathbf{y}$ denotes the input noisy color image, $\mathbf{R}_i$ denotes the matrix which extracts the i-th concatenated color patch from the image $\mathbf{u}$, and, $\mathbf{z}_i$ denotes the estimate of i-th concatenated color patch obtained from the Patch Estimation step. We can rewrite the minimization in (G.1) as,

$$\begin{aligned}
\hat{\mathbf{u}} &= \arg\min_{\mathbf{u}} \frac{\lambda}{2} \|\mathbf{u} - \mathbf{y}\|_2^2 + \sum_{i=1}^{L} \frac{\beta}{2} \|\mathbf{R}_i \mathbf{u} - \mathbf{z}_i\|_2^2 \\
&= \arg\min_{\mathbf{u}} \frac{\lambda}{2} \left(\mathbf{u}^T \mathbf{u} - 2\mathbf{u}^T \mathbf{y}\right) + \sum_{i=1}^{L} \frac{\beta}{2} \left(\mathbf{u}^T \mathbf{R}_i^T \mathbf{R}_i \mathbf{u} - 2\mathbf{u}^T \mathbf{R}_i^T \mathbf{z}_i\right) \\
&= \arg\min_{\mathbf{u}} \mathbf{u}^T \left(\lambda \mathbf{I} + \sum_{i=1}^{L} \beta \mathbf{R}_i^T \mathbf{R}_i\right) \mathbf{u} - 2\mathbf{u}^T \left(\lambda \mathbf{y} + \sum_{i=1}^{L} \beta \mathbf{R}_i^T \mathbf{z}_i\right) .
\end{aligned} \tag{G.2}$$

The optimal $\mathbf{u}$ can be obtained by differentiating (G.2) with respect to $\mathbf{u}$ and setting the result to zero as,

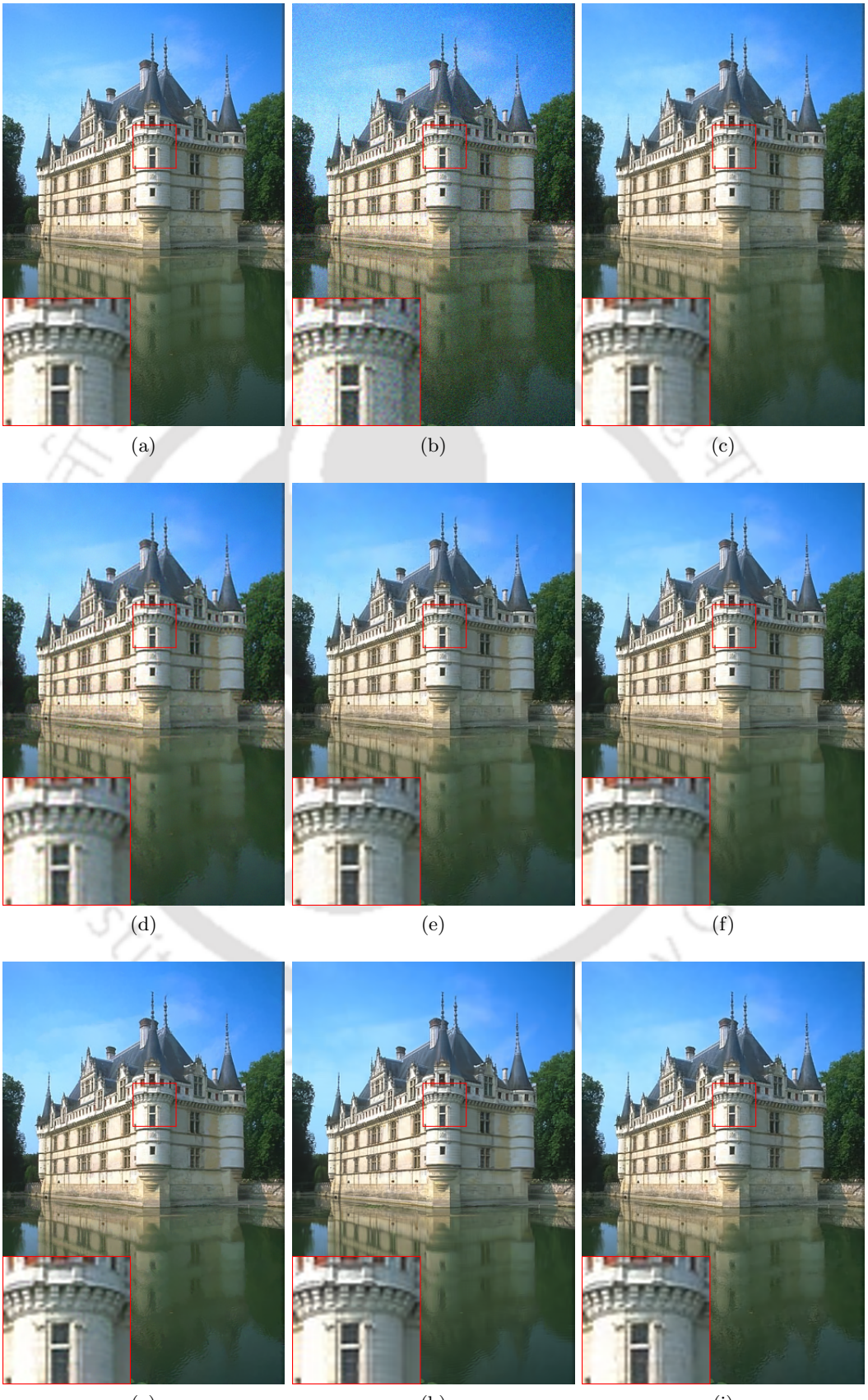
$$\begin{aligned}
&\frac{\partial}{\partial \mathbf{u}}\left[\mathbf{u}^T\left(\lambda\mathbf{I}+\sum_{i=1}^{L}\beta\mathbf{R}_i^T\mathbf{R}_i\right)\mathbf{u}-2\mathbf{u}^T\left(\lambda\mathbf{y}+\sum_{i=1}^{L}\beta\mathbf{R}_i^T\mathbf{z}_i\right)\right]=\mathbf{0}\\
&\Rightarrow 2\left(\lambda\mathbf{I}+\sum_{i=1}^{L}\beta\mathbf{R}_i^T\mathbf{R}_i\right)\mathbf{u}-2\left(\lambda\mathbf{y}+\sum_{i=1}^{L}\beta\mathbf{R}_i^T\mathbf{z}_i\right)=\mathbf{0}\\
&\Rightarrow \mathbf{u}=\left(\lambda\mathbf{I}+\sum_{i=1}^{L}\beta\mathbf{R}_i^T\mathbf{R}_i\right)^{-1}\left(\lambda\mathbf{y}+\sum_{i=1}^{L}\beta\mathbf{R}_i^T\mathbf{z}_i\right). \qquad \text{(G.3)}
\end{aligned}$$

Thus, the required estimate $\hat{\mathbf{u}}$ of the unknown color image can be written as,

$$\hat{\mathbf{u}}=\left(\lambda\mathbf{I}+\beta\sum_{i=1}^{L}\mathbf{R}_i^T\mathbf{R}_i\right)^{-1}\left(\lambda\mathbf{y}+\beta\sum_{i=1}^{L}\mathbf{R}_i^T\mathbf{z}_i\right) \qquad \text{(G.4)}$$

Appendix H

Visual Comparison of Denoised Images Produced by JCS-GMM Denoising Algorithm



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

FIGURE H.1: Denoised results in the case of Castle image and $\sigma = 10$. (a) Ground Truth, (b) Noisy Image, (c) PLE, (d) Q-KSVD, (e) Color-KSVD, (f) C-BM3D, (g) NL-Bayes, (h) HOSVD, (i) Prop

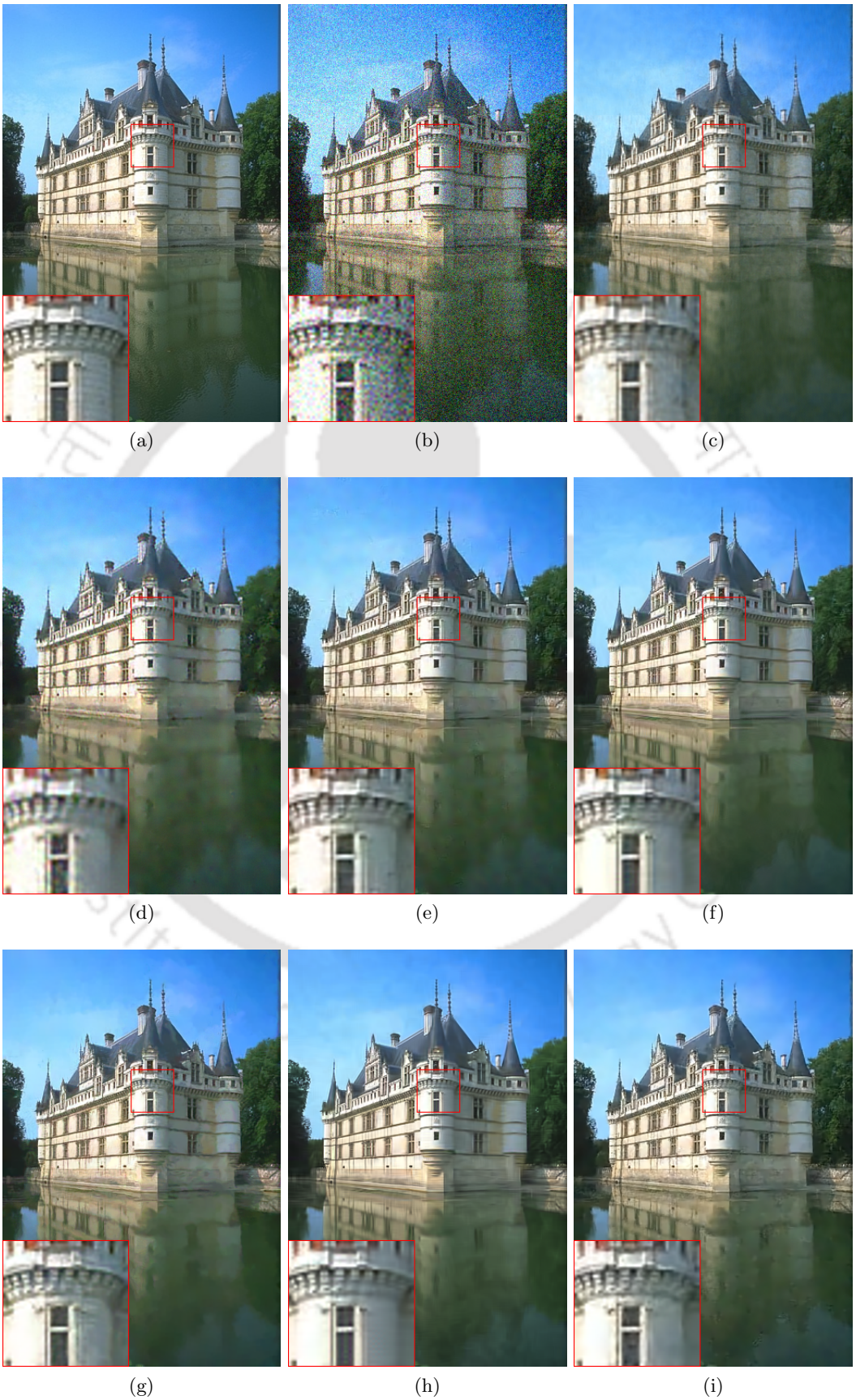


FIGURE H.2: Denoised results in the case of Castle image and $\sigma = 25$. (a) Ground Truth, (b) Noisy Image, (c) PLE, (d) Q-KSVD, (e) Color-KSVD, (f) C-BM3D, (g) NL-Bayes, (h) HOSVD, (i) Prop

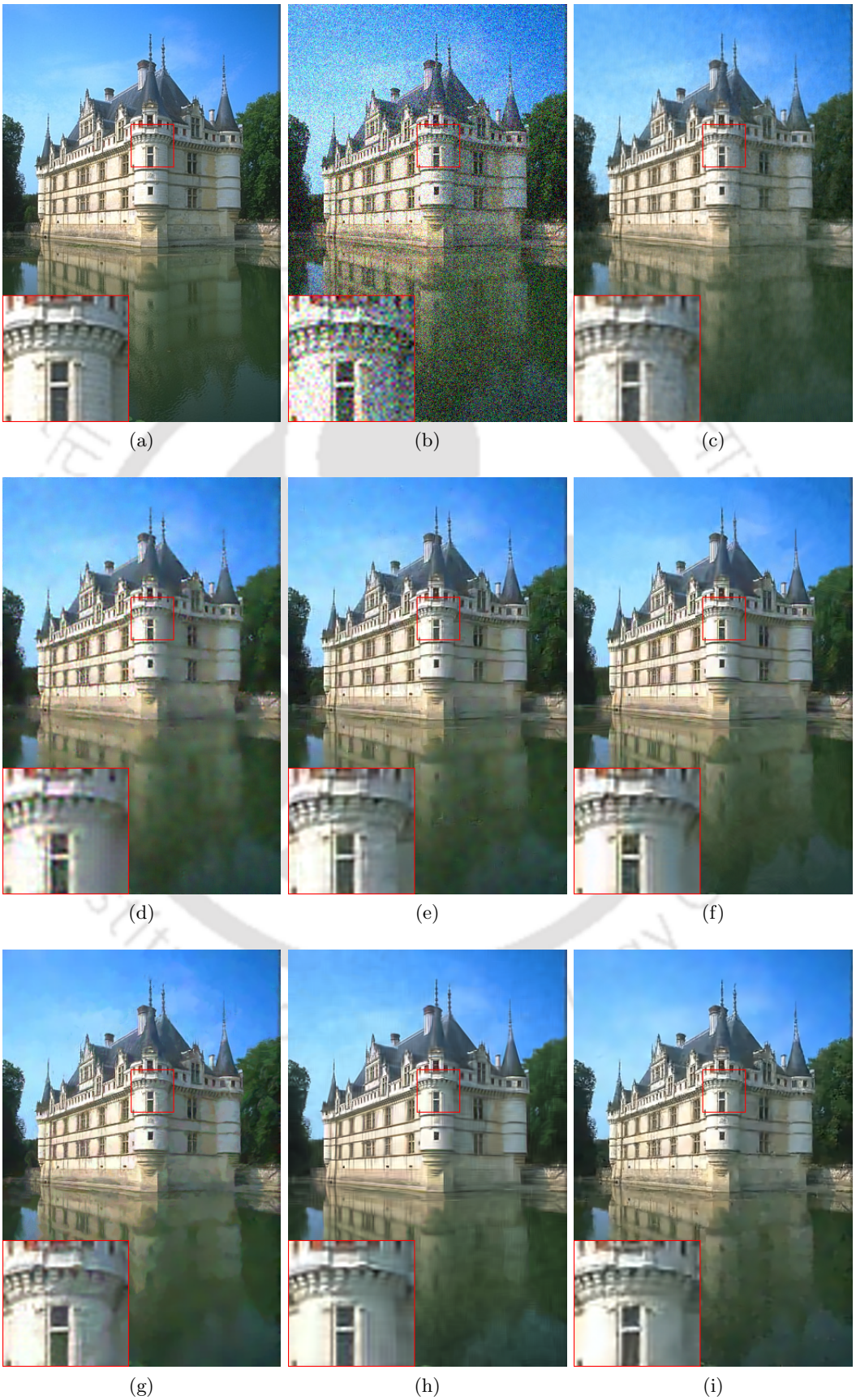
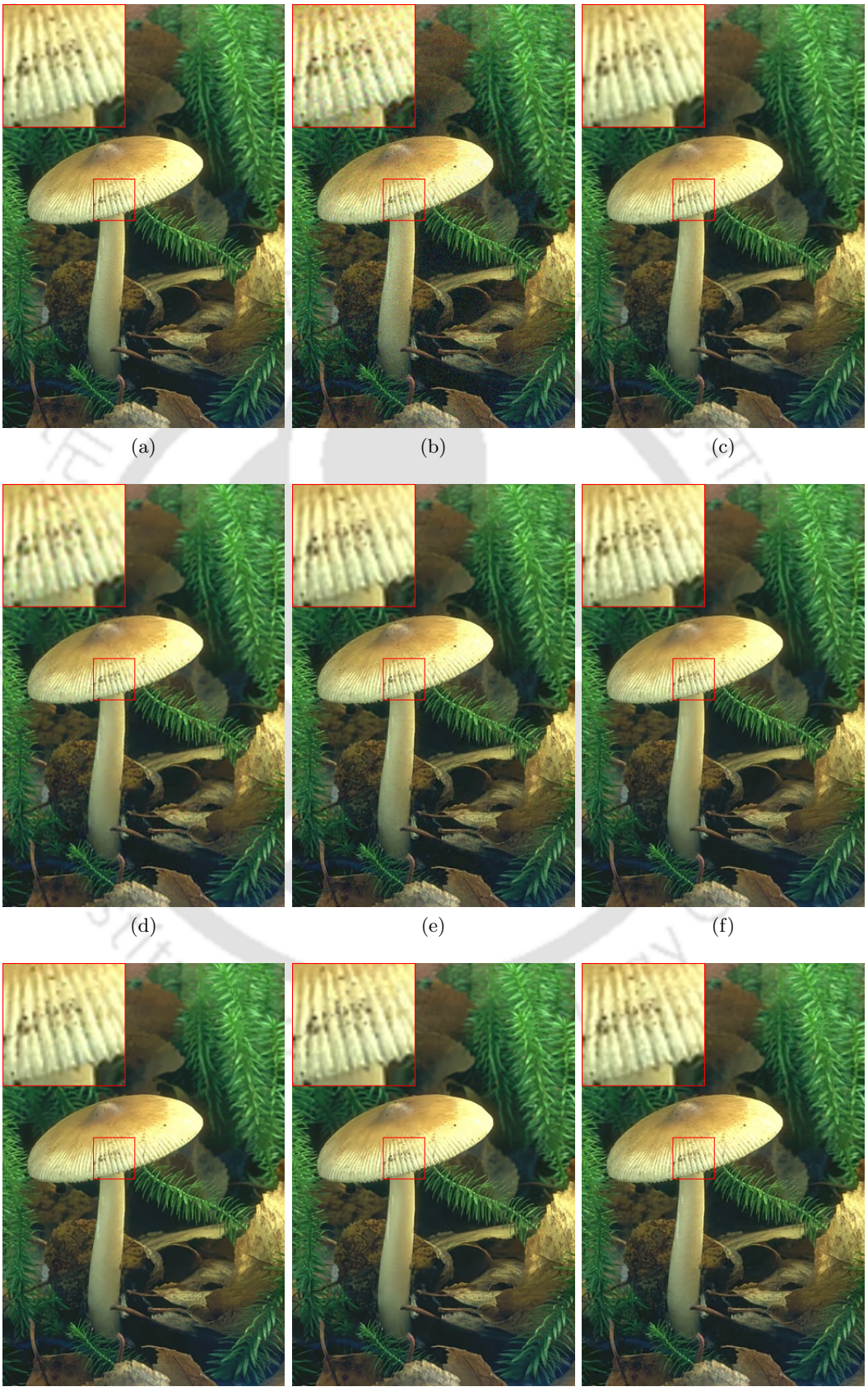


FIGURE H.3: Denoised results in the case of Castle image and $\sigma = 35$. (a) Ground Truth, (b) Noisy Image, (c) PLE, (d) Q-KSVD, (e) Color-KSVD, (f) C-BM3D, (g) NL-Bayes, (h) HOSVD, (i) Prop



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

FIGURE H.4: Denoised results in the case of Mushroom image and $\sigma = 10$. (a) Ground Truth, (b) Noisy Image, (c) PLE, (d) Q-KSVD, (e) Color-KSVD, (f) C-BM3D, (g) NL-Bayes, (h) HOSVD, (i) Prop

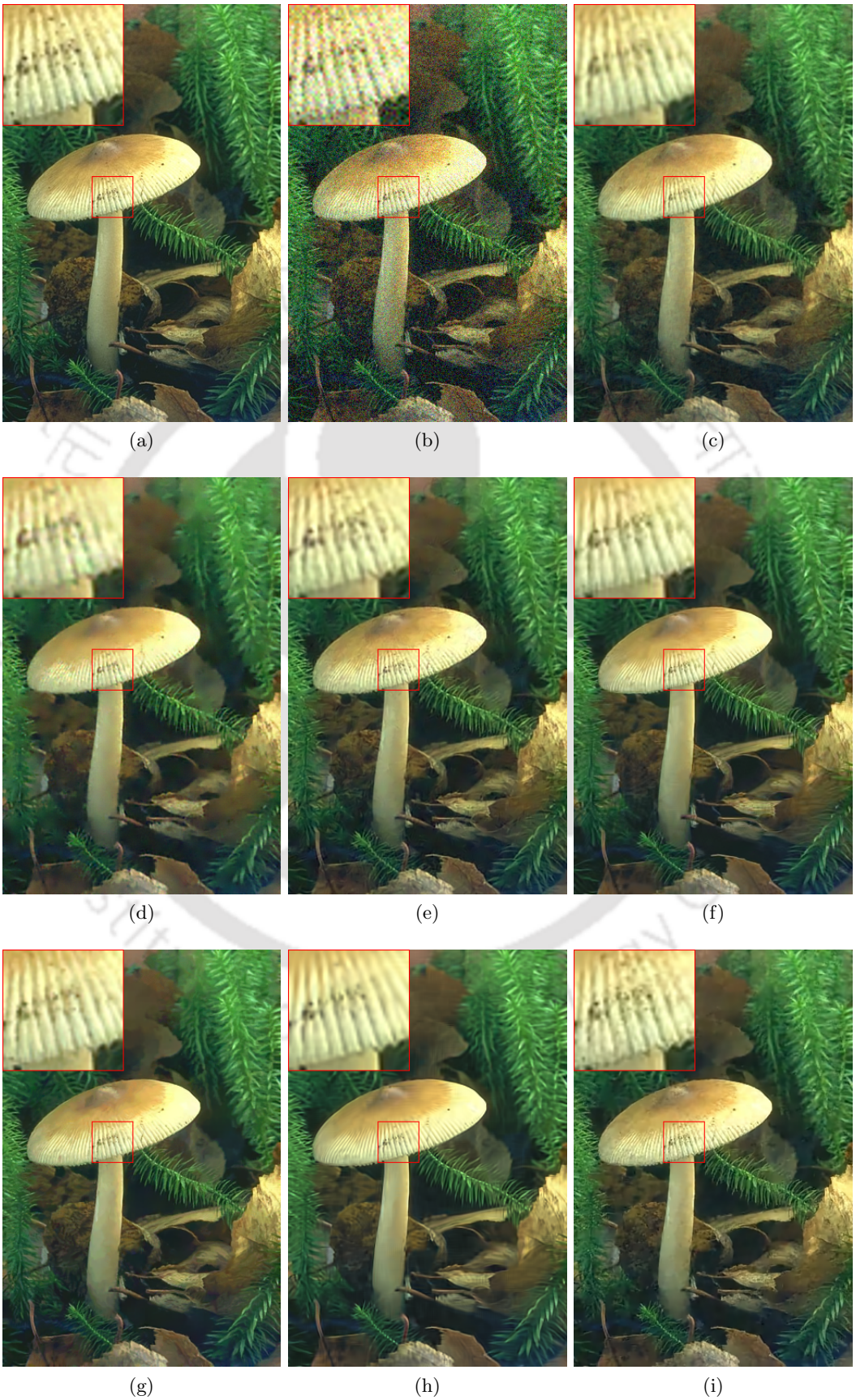
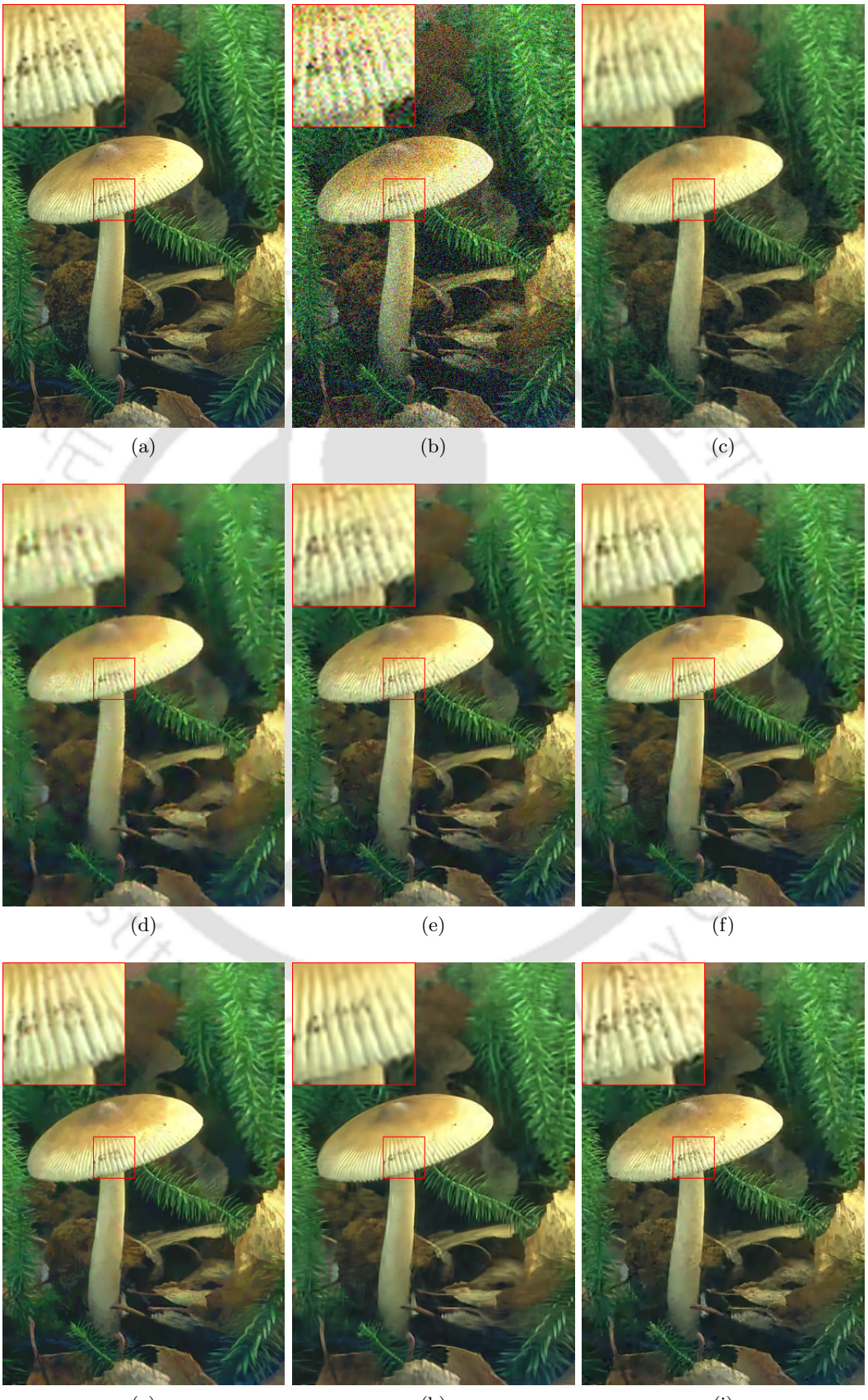


FIGURE H.5: Denoised results in the case of Mushroom image and $\sigma = 25$. (a) Ground Truth, (b) Noisy Image, (c) PLE, (d) Q-KSVD, (e) Color-KSVD, (f) C-BM3D, (g) NL-Bayes, (h) HOSVD, (i) Prop

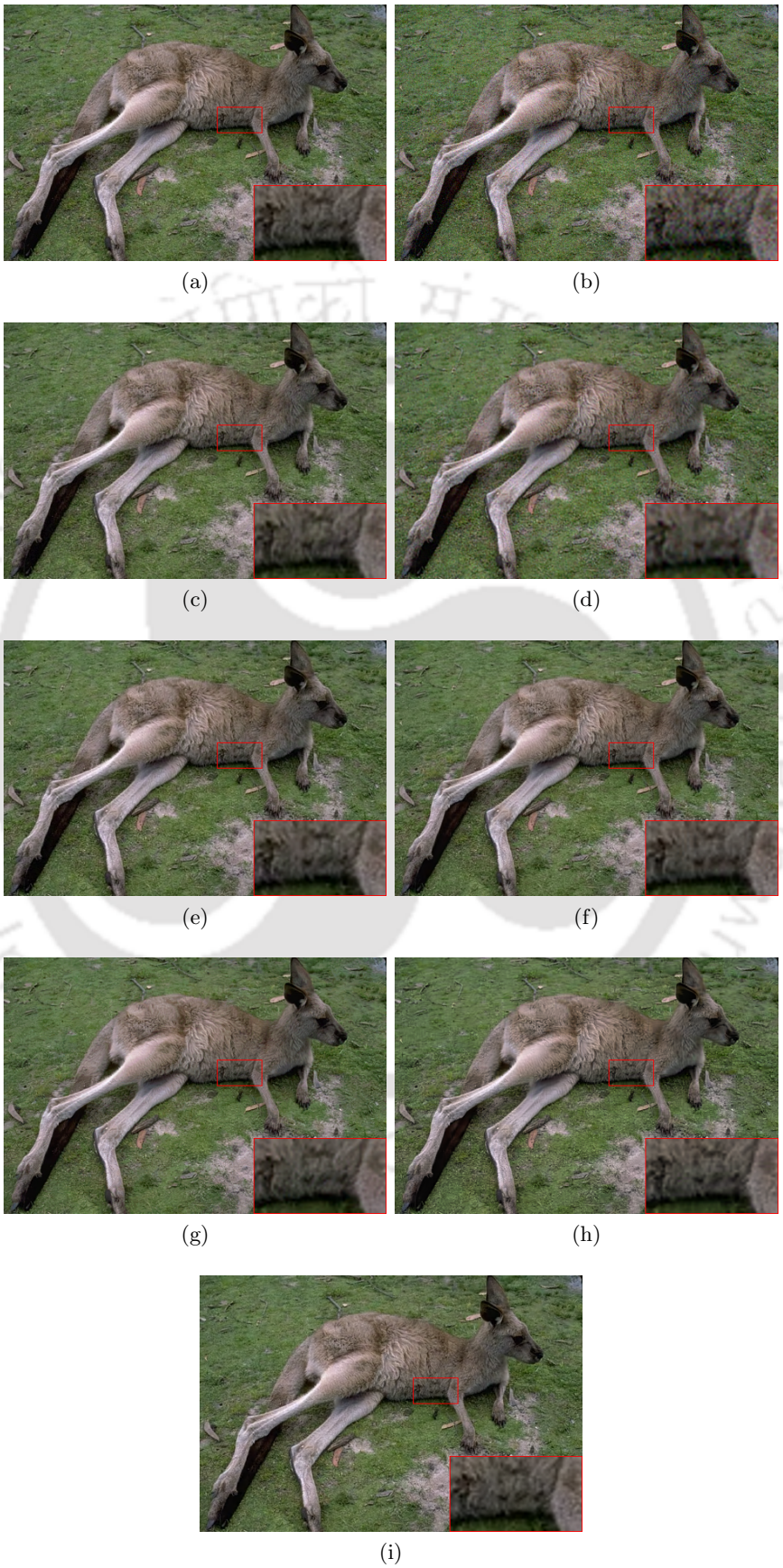


(a) (b) (c)

(d) (e) (f)

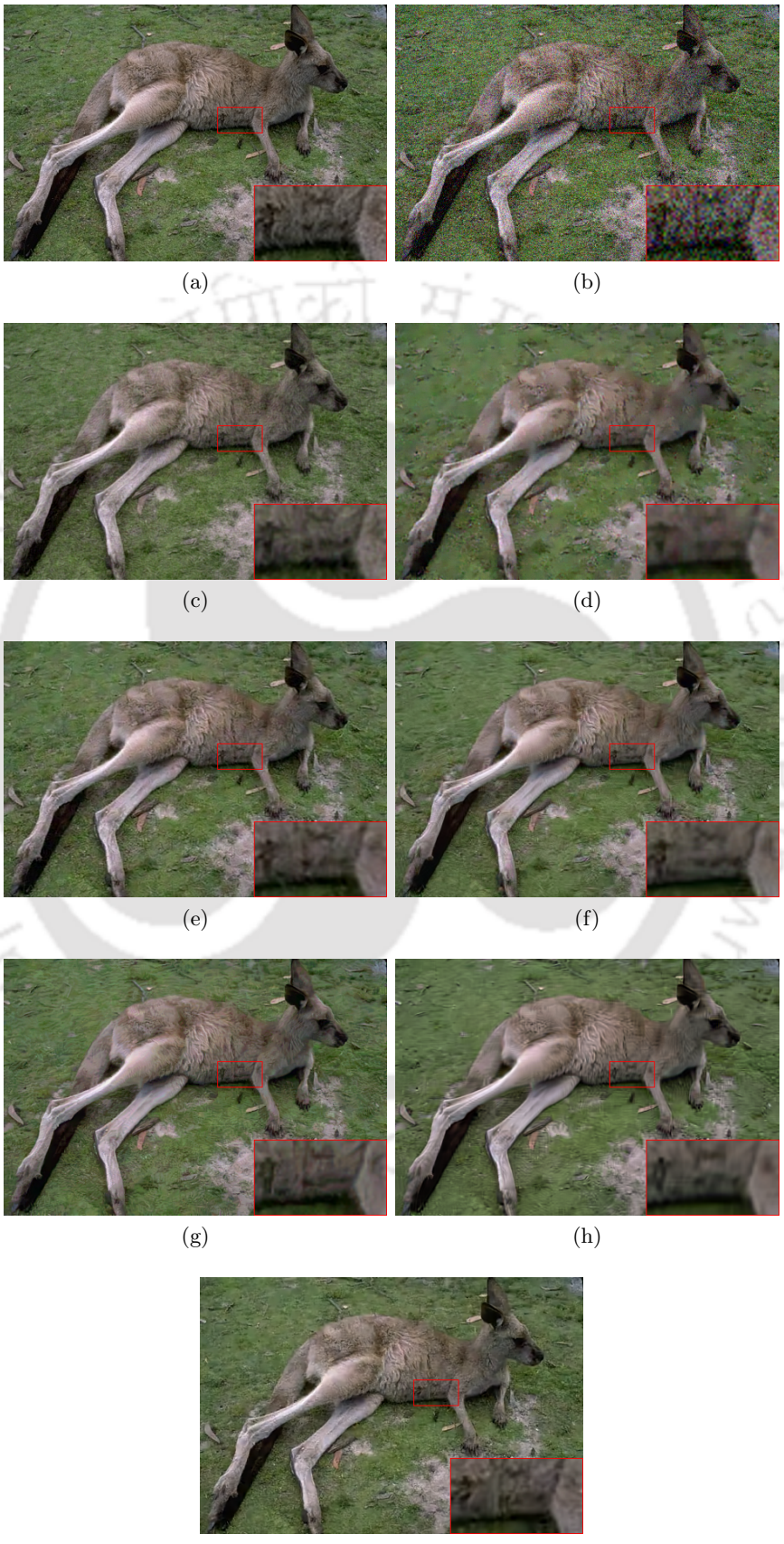
(g) (h) (i)

FIGURE H.6: Denoised results in the case of Mushroom image and $\sigma = 35$. (a) Ground Truth, (b) Noisy Image, (c) PLE, (d) Q-KSVD, (e) Color-KSVD, (f) C-BM3D, (g) NL-Bayes, (h) HOSVD, (i) Prop



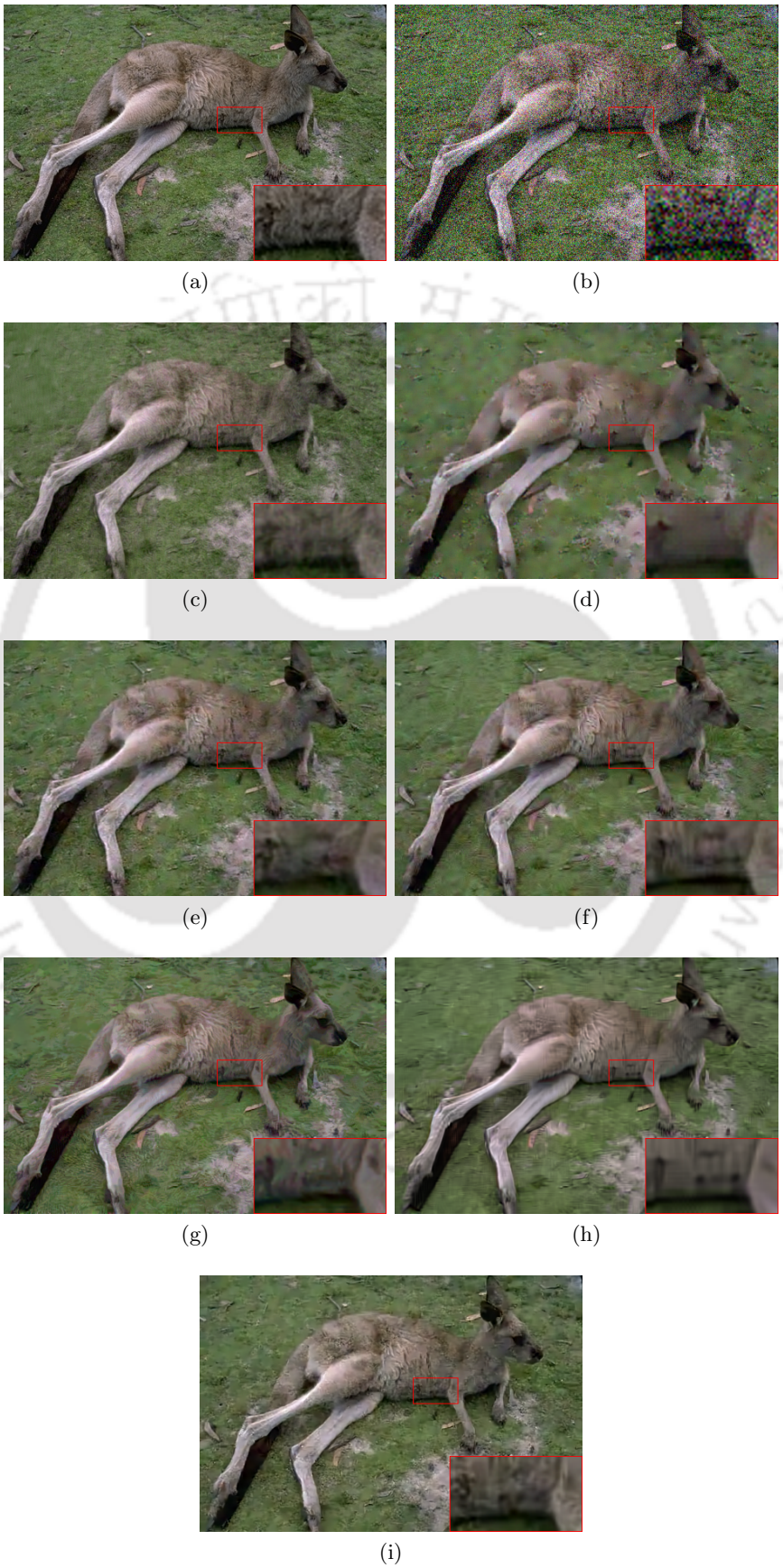
(a) (b) (c) (d) (e) (f) (g) (h) (i)

FIGURE H.7: Denoised results in the case of Kangaroo image and $\sigma = 10$. (a) Ground Truth, (b) Noisy Image, (c) PLE, (d) Q-KSVD, (e) Color-KSVD, (f) C-BM3D, (g) NL-Bayes, (h) HOSVD, (i) Prop



(a) (b) (c) (d) (e) (f) (g) (h) (i)

FIGURE H.8: Denoised results in the case of Kangaroo image and $\sigma = 25$. (a) Ground Truth, (b) Noisy Image, (c) PLE, (d) Q-KSVD, (e) Color-KSVD, (f) C-BM3D, (g) NL-Bayes, (h) HOSVD, (i) Prop

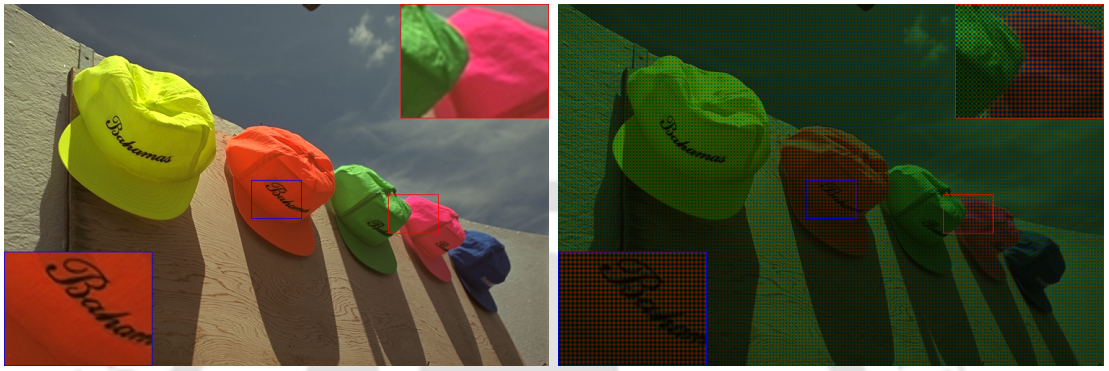


(a) (b) (c) (d) (e) (f) (g) (h) (i)

FIGURE H.9: Denoised results in the case of Kangaroo image and $\sigma = 35$. (a) Ground Truth, (b) Noisy Image, (c) PLE, (d) Q-KSVD, (e) Color-KSVD, (f) C-BM3D, (g) NL-Bayes, (h) HOSVD, (i) Prop

Appendix I

Visual Comparison of Demosaicked Images Produced by JCS-GMM Demosaicking Algorithm



(a) (b)

(c) (d)

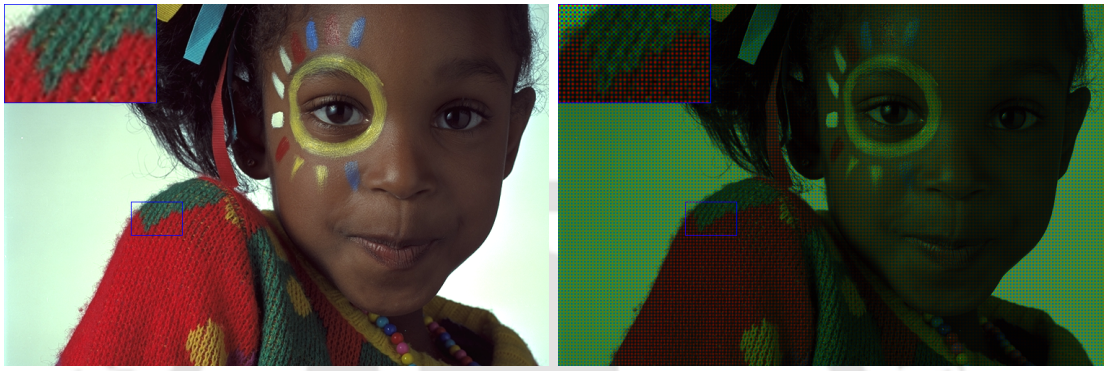
(e) (f)

(g) (h)

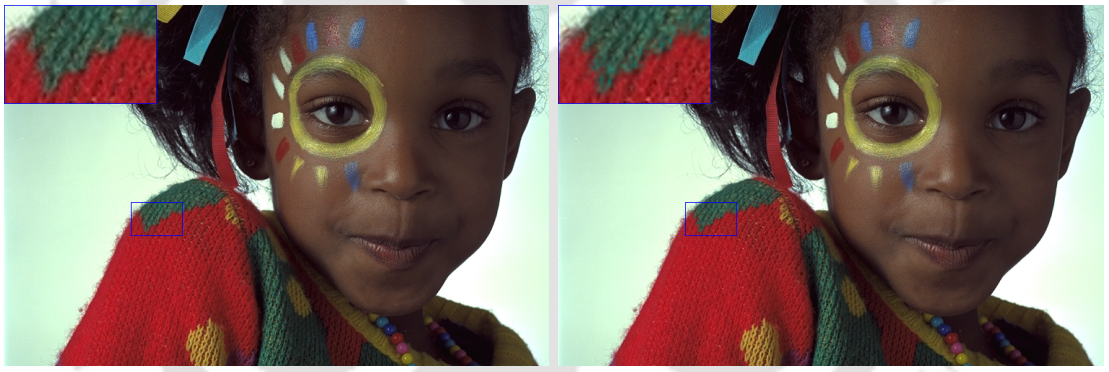
(i) (j)

(k) (l)

FIGURE I.1: Demosaicked Results in the case of Image No. 3 from Kodak Database. (a) Ground Truth, (b) Mosaicked Image, (c) MSG, (d) RI, (e) ARI, (f) MLRI, (g) MLRI+wei, (h) SSCAD, (i) Color-KSVD, (j) DDR, (k) FR, (l) Prop



(a) (b)



(c) (d)

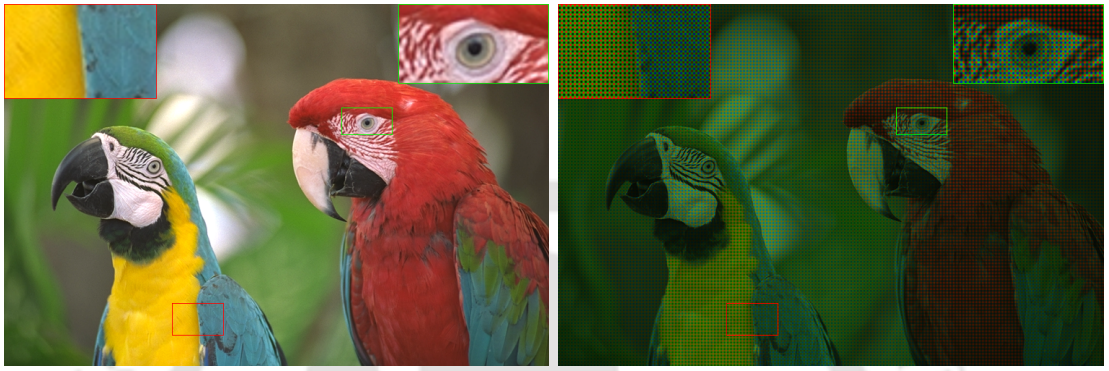
(e) (f)

(g) (h)

(i) (j)

(k) (l)

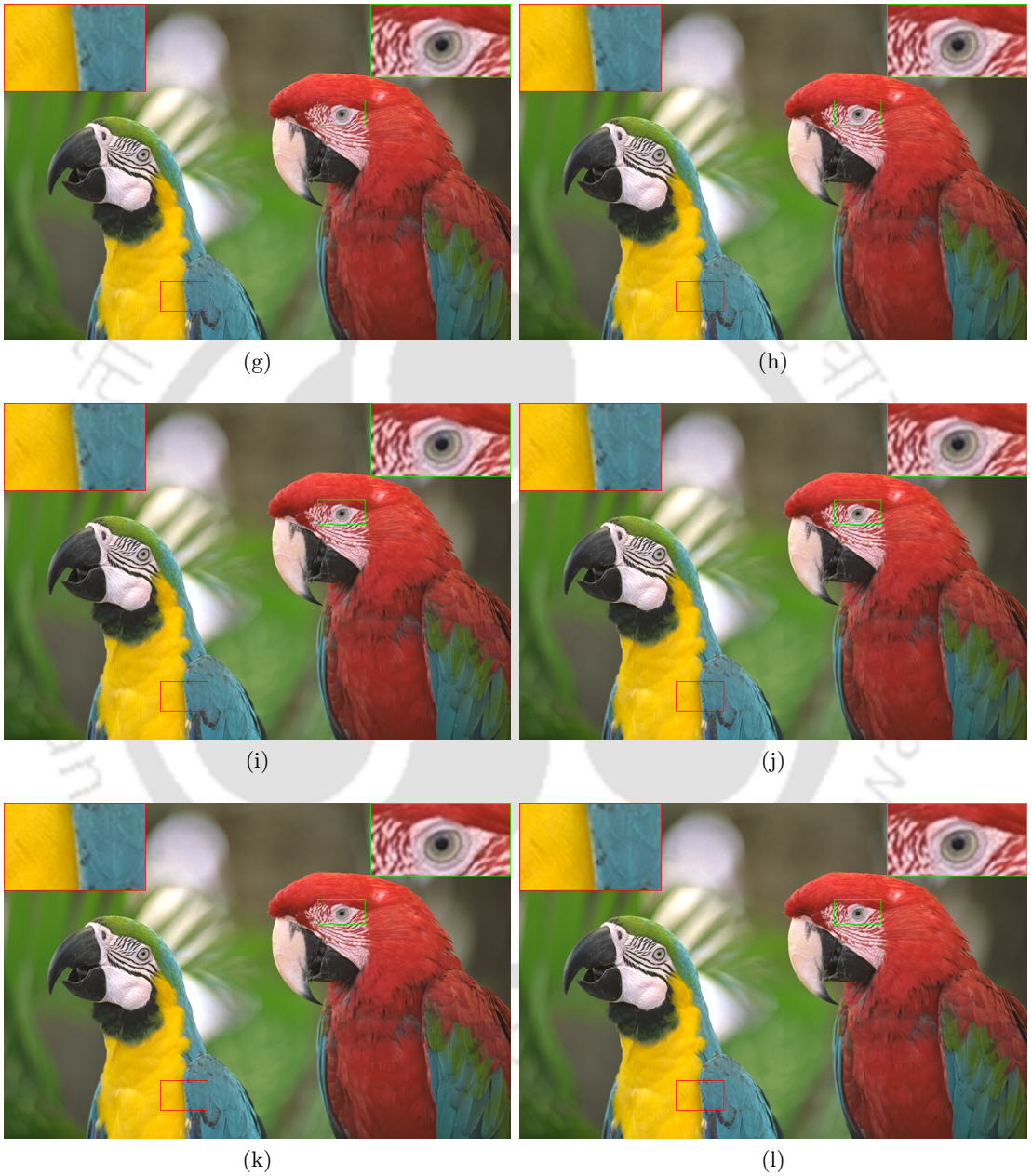
FIGURE I.2: Demosaicked Results in the case of Image No. 15 from Kodak Database. (a) Ground Truth, (b) Mosaicked Image, (c) MSG, (d) RI, (e) ARI, (f) MLRI, (g) MLRI+wei, (h) SSCAD, (i) Color-KSVD, (j) DDR, (k) FR, (l) Prop



(a) (b)

(c) (d)

(e) (f)

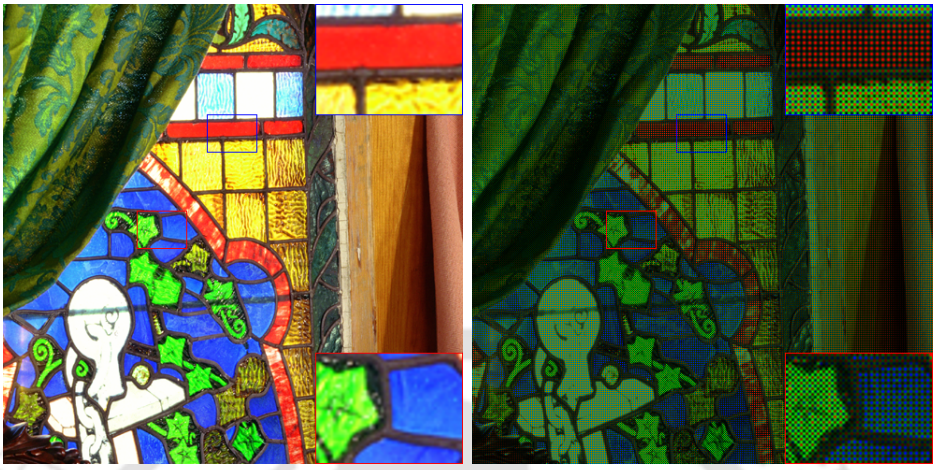


(g) (h)

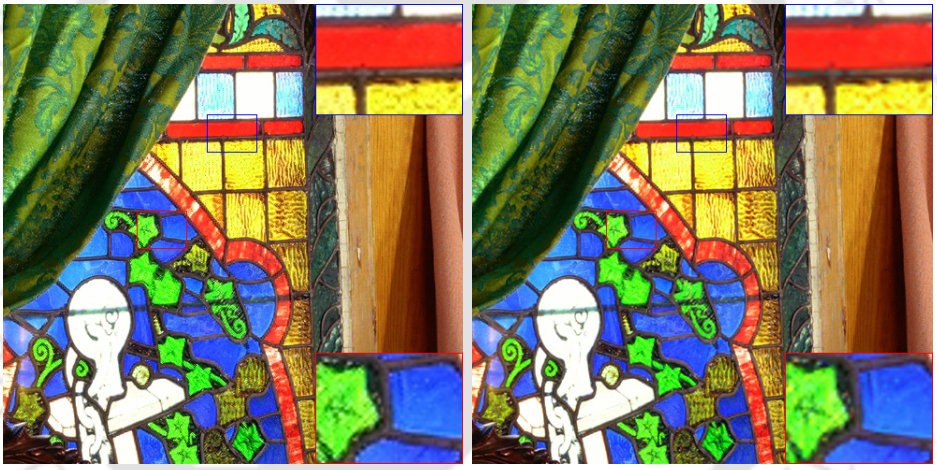
(i) (j)

(k) (l)

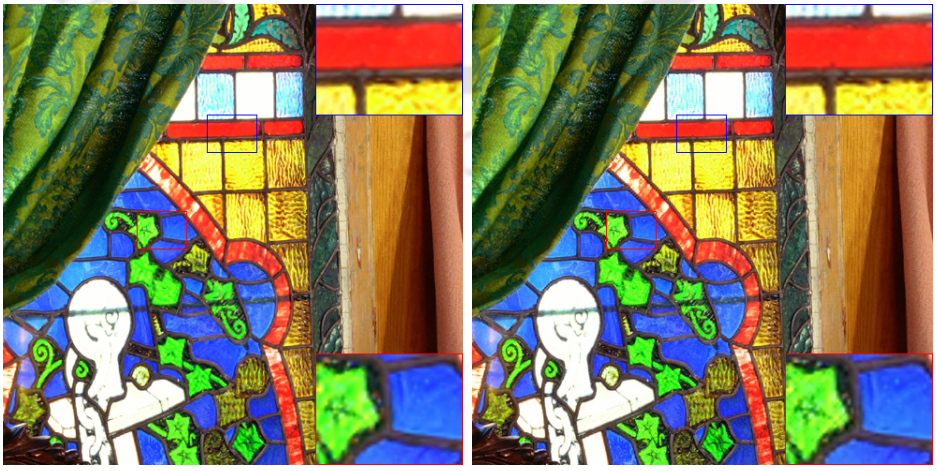
FIGURE I.3: Demosaicked Results in the case of Image No. 23 from Kodak Database. (a) Ground Truth, (b) Mosaicked Image, (c) MSG, (d) RI, (e) ARI, (f) MLRI, (g) MLRI+wei, (h) SSCAD, (i) Color-KSVD, (j) DDR, (k) FR, (l) Prop



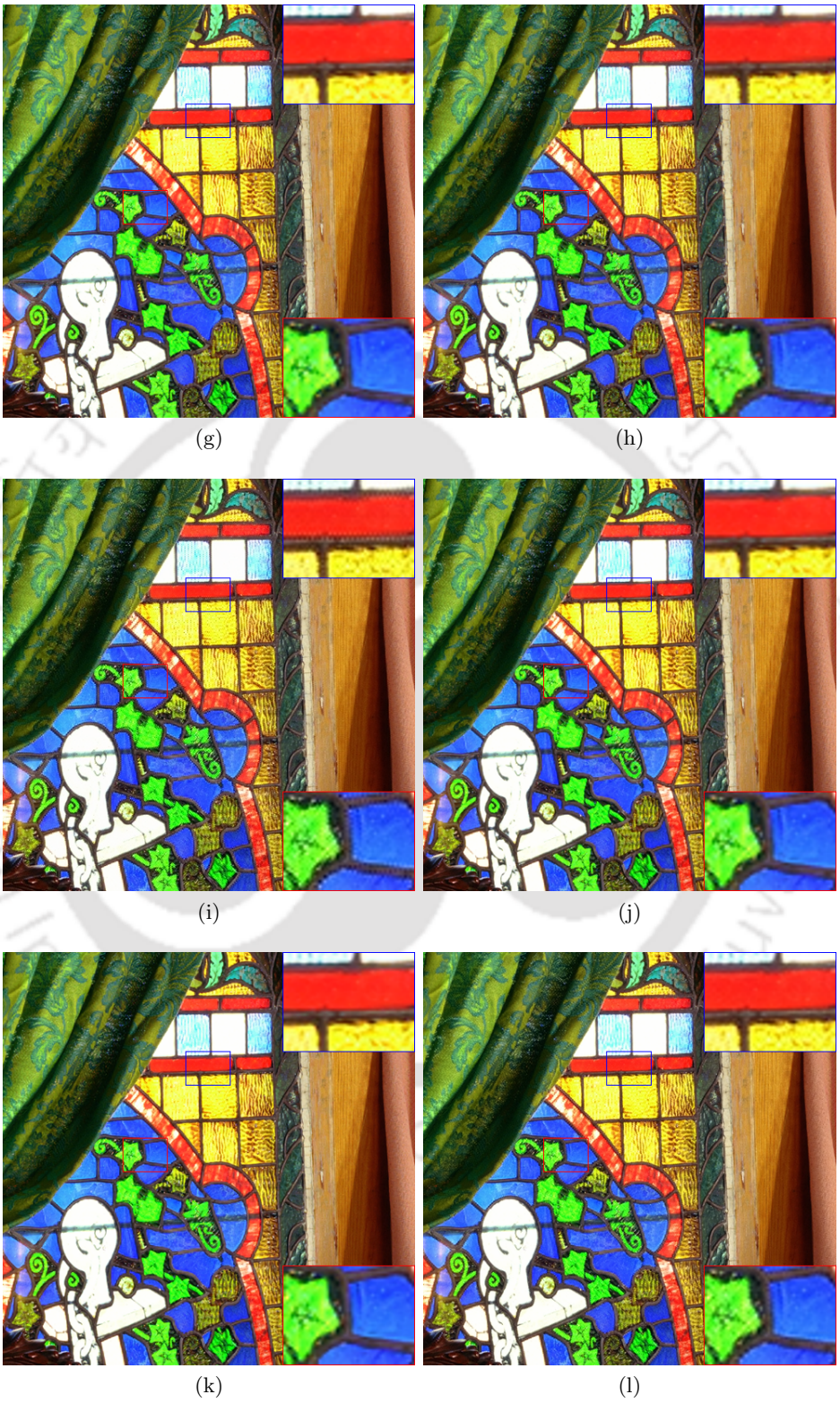
(a) (b)



(c) (d)



(e) (f)



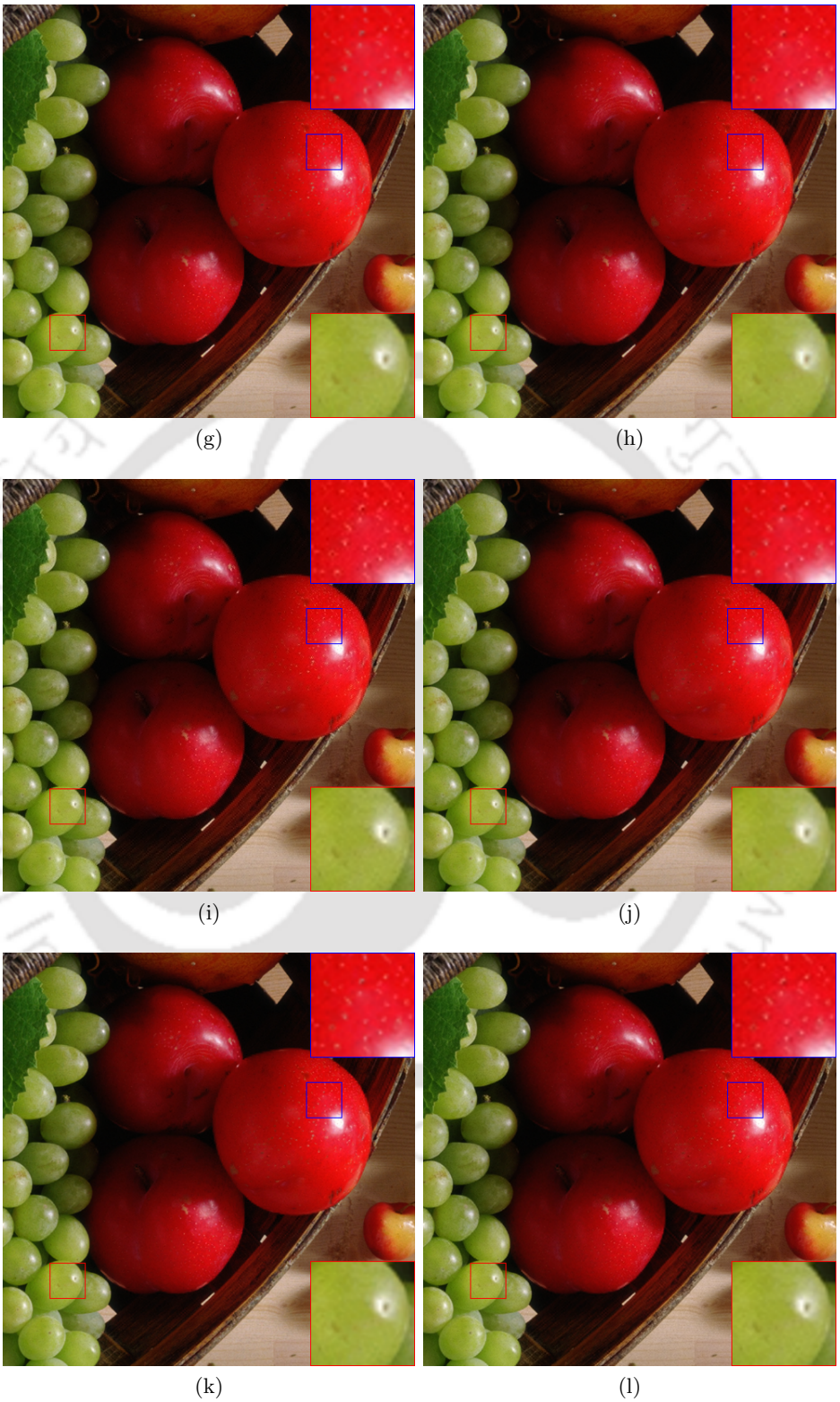
(g) (h) (i) (j) (k) (l)

FIGURE I.4: Demosaicked Results in the case of Image No. 01 from IMAX Database. (a) Ground Truth, (b) Mosaicked Image, (c) MSG, (d) RI, (e) ARI, (f) MLRI, (g) MLRI+wei, (h) SSCAD, (i) Color-KSVD, (j) DDR, (k) FR, (l) Prop

(a) (b)

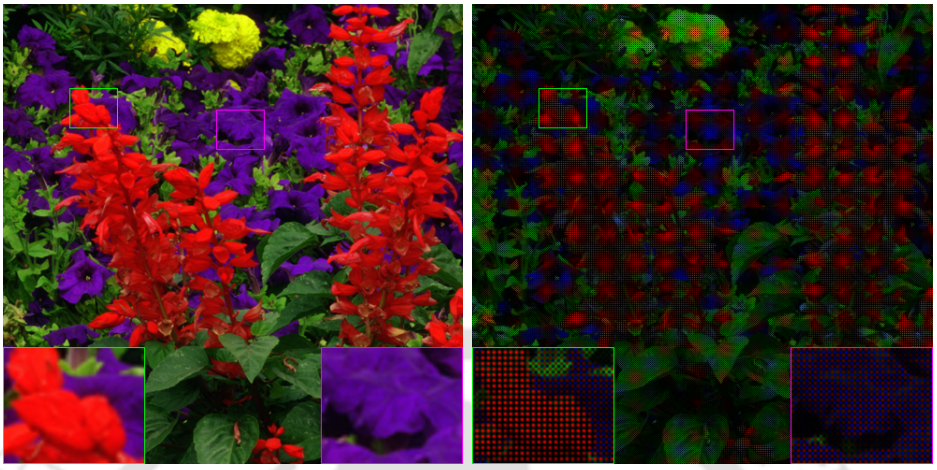
(c) (d)

(e) (f)

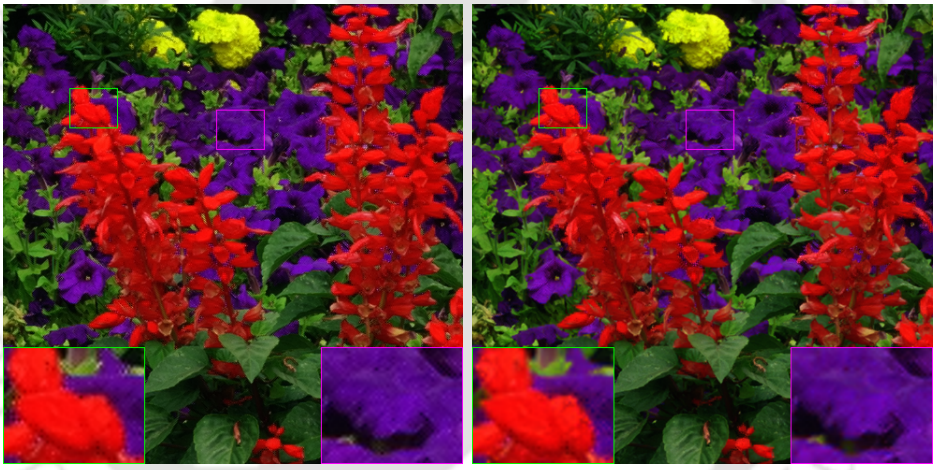


(g) (h) (i) (j) (k) (l)

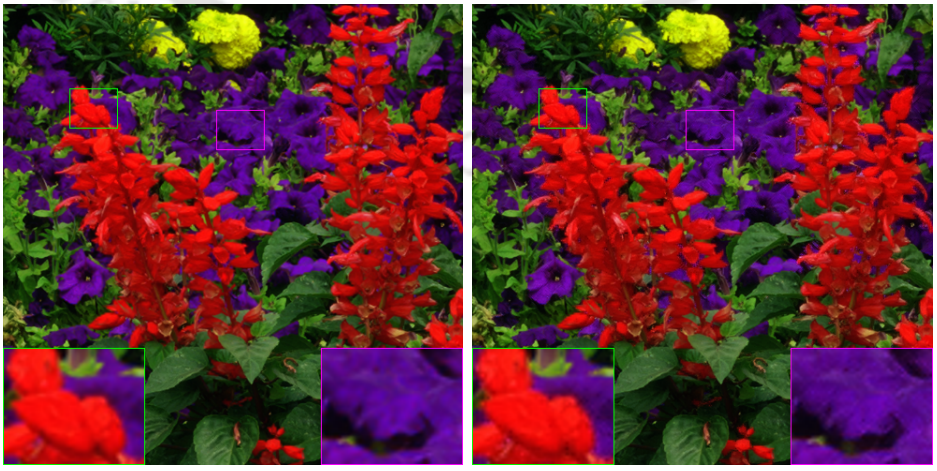
FIGURE I.5: Demosaicked Results in the case of Image No. 15 from IMAX Database. (a) Ground Truth, (b) Mosaicked Image, (c) MSG, (d) RI, (e) ARI, (f) MLRI, (g) MLRI+wei, (h) SSCAD, (i) Color-KSVD, (j) DDR, (k) FR, (l) Prop



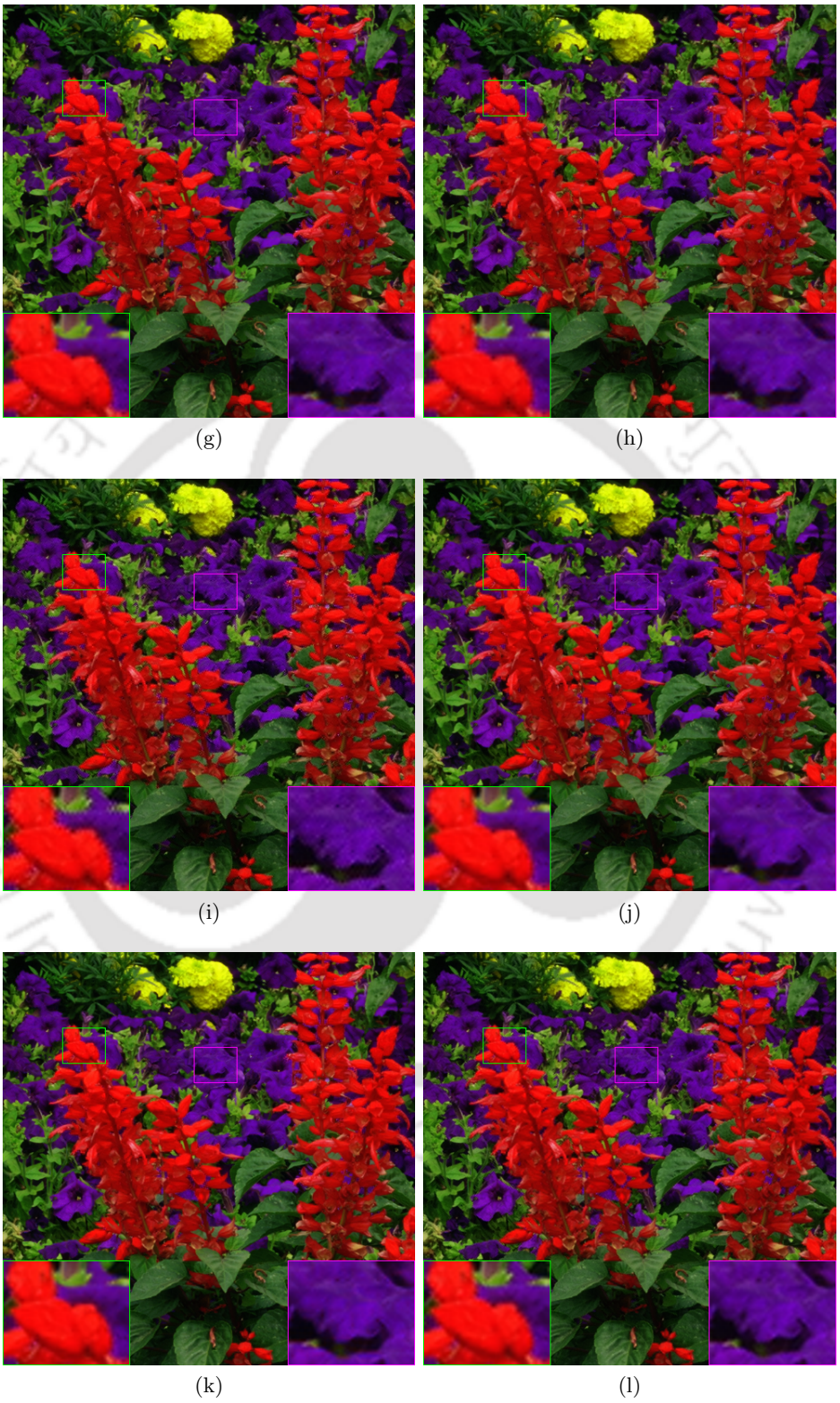
(a) (b)



(c) (d)



(e) (f)



(g) (h) (i) (j) (k) (l)

FIGURE I.6: Demosaicked Results in the case of Image No. 17 from IMAX Database. (a) Ground Truth, (b) Mosaicked Image, (c) MSG, (d) RI, (e) ARI, (f) MLRI, (g) MLRI+wei, (h) SSCAD, (i) Color-KSVD, (j) DDR, (k) FR, (l) Prop

Bibliography

[1] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: Nonlinear Phenomena*, vol. 60, no. 1, pp. 259–268, 1992.

[2] T. Chan, S. Esedoglu, F. Park, and A. Yip, *Total variation image restoration: overview and recent developments*. Boston, MA: Springer US, 2006, pp. 17–31.

[3] Y. W. Wen, M. K. Ng, and Y. M. Huang, "Efficient total variation minimization methods for color image restoration," *IEEE Transactions on Image Processing*, vol. 17, no. 11, pp. 2081–2088, Nov 2008.

[4] J. Yang, J. Wright, T. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE Transactions on Image Processing*, vol. 19, no. 11, pp. 2861–2873, Nov 2010.

[5] J. Yang, Z. Wang, Z. Lin, S. Cohen, and T. Huang, "Coupled dictionary training for image super-resolution," *IEEE Transactions on Image Processing*, vol. 21, no. 8, pp. 3467–3478, Aug 2012.

[6] S. Wang, D. Zhang, Y. Liang, and Q. Pan, "Semi-coupled dictionary learning with applications to image super-resolution and photo-sketch synthesis," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2012, pp. 2216–2223.

[7] T. Peleg and M. Elad, "A statistical prediction model based on sparse representations for single image super-resolution," *IEEE Transactions on Image Processing*, vol. 23, no. 6, pp. 2569–2582, June 2014.

[8] A. Buades, B. Coll, and J. M. Morel, "A review of image denoising algorithms, with a new one," *Multiscale Modeling & Simulation*, vol. 4, no. 2, pp. 490–530, 2005.

[9] M. Zontak and M. Irani, "Internal statistics of a single natural image," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2011, pp. 977–984.

[10] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Transactions on Image Processing*, vol. 15, no. 12, pp. 3736–3745, Dec 2006.

[11] G. Yu, G. Sapiro, and S. Mallat, "Solving inverse problems with piecewise linear estimators: From Gaussian Mixture Models to structured sparsity," *IEEE Transactions on Image Processing*, vol. 21, no. 5, pp. 2481–2499, May 2012.

[12] S. Gu, L. Zhang, W. Zuo, and X. Feng, "Weighted nuclear norm minimization with application to image denoising," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2014*, June 2014, pp. 2862–2869.

[13] J. Mairal, G. Sapiro, and M. Elad, "Learning multiscale sparse representations for image and video restoration," *Multiscale Modeling & Simulation*, vol. 7, no. 1, pp. 214–241, 2008.

[14] J. M. Fadili, J.-L. Starck, and F. Murtagh, "Inpainting and zooming using sparse representations," *The Computer Journal*, vol. 52, no. 1, pp. 64–79, 2007.

[15] Z. Xu and J. Sun, "Image inpainting by patch propagation using patch sparsity," *IEEE Transactions on Image Processing*, vol. 19, no. 5, pp. 1153–1165, May 2010.

[16] Y. Lou, A. L. Bertozzi, and S. Soatto, "Direct sparse deblurring," *Journal of Mathematical Imaging and Vision*, vol. 39, no. 1, pp. 1–12, Jan 2011.

[17] W. Dong, L. Zhang, G. Shi, and X. Wu, "Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization," *IEEE Transactions on Image Processing*, vol. 20, no. 7, pp. 1838–1857, July 2011.

[18] D. Zoran and Y. Weiss, "From learning models of natural image patches to whole image restoration," in *IEEE International Conference on Computer Vision (ICCV)*, Nov 2011, pp. 479–486.

[19] ——, "Natural images, gaussian mixtures and dead leaves," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1736–1744.

[20] M. Lebrun, A. Buades, and J. M. Morel, "A nonlocal Bayesian image denoising algorithm," *SIAM Journal on Imaging Sciences*, vol. 6, no. 3, pp. 1665–1688, 2013.

[21] Y. Eldar and M. Mishali, "Robust recovery of signals from a structured union of subspaces," *IEEE Transactions on Information Theory*, vol. 55, no. 11, pp. 5302–5316, Nov 2009.

[22] Y. Eldar, P. Kuppinger, and H. Bolcskei, "Block-sparse signals: uncertainty relations and efficient recovery," *IEEE Transactions on Signal Processing*, vol. 58, no. 6, pp. 3042–3054, June 2010.

[23] Y. Eldar and H. Bolcskei, "Block-sparsity: coherence and efficient recovery," in *IEEE International Conference on Acoustics, Speech and Signal Processing, 2009. ICASSP 2009.*, April 2009, pp. 2885–2888.

[24] R. Timofte, V. De, and L. V. Gool, "A+: Adjusted anchored neighborhood regression for fast super-resolution," in *12th Asian Conference on Computer Vision Computer Vision (ACCV)*, vol. 9006, November 2015, pp. 111–126.

[25] J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," *IEEE Transactions on Image Processing*, vol. 17, no. 1, pp. 53–69, Jan 2008.

[26] D. D. Muresan and T. W. Parks, "Adaptive principal components and image denoising," in *Proceedings of International Conference on Image Processing (ICIP)*, vol. 1, Sept 2003, pp. I–101–4.

[27] Y. Cao, Y. Luo, and S. Yang, "Image denoising with gaussian mixture model," in *Congress on Image and Signal Processing 2008*, vol. 3, May 2008, pp. 339–343.

[28] Y. Eldar and G. Kutyniok, *Compressed Sensing: Theory and Applications*. Cambridge University Press, 2012.

[29] J. Huang and T. Zhang, "The benefit of group sparsity," *Ann. Statist.*, vol. 38, no. 4, pp. 1978–2004, 08 2010.

[30] W. Deng, W. Yin, and Y. Zhang, "Group sparse optimization by alternating direction method," vol. 8858, 2013, pp. 88 580R–88 580R–15.

[31] Y.-Q. Wang and J.-M. Morel, "SURE guided Gaussian mixture image denoising," *SIAM Journal on Imaging Sciences*, vol. 6, no. 2, pp. 999–1034, 2013.

[32] M. E. Tipping and C. M. Bishop, "Mixtures of probabilistic principal component analyzers," *Neural Computing*, vol. 11, no. 2, pp. 443–482, Feb. 1999.

[33] Y.-Q. Wang, "E-PLE: an algorithm for image inpainting," *Image Processing On Line*, vol. 3, pp. 271–285, 2013.

[34] A. Teodoro, M. Almeida, and M. Figueiredo, "Single-frame image denoising and inpainting using gaussian mixtures," in *Proceedings of the International Conference on Pattern Recognition Applications and Methods (ICPRAM)*, vol. 2, 2015, pp. 283–288.

[35] C. Aguerrebere, A. Almansa, Y. Gousseau, J. Delon, and P. Musé, "Single shot high dynamic range imaging using piecewise linear estimators," in *IEEE International Conference on Computational Photography (ICCP)*, May 2014, pp. 1–10.

[36] F. Léger, G. Yu, and G. Sapiro, "Efficient matrix completion with gaussian models," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2011, pp. 1113–1116.

[37] J. Yang, X. Yuan, X. Liao, P. Llull, D. J. Brady, G. Sapiro, and L. Carin, "Video compressive sensing using gaussian mixture models," *IEEE Transactions on Image Processing*, vol. 23, no. 11, pp. 4863–4878, Nov 2014.

[38] G. Yu and G. Sapiro, "Statistical compressive sensing of gaussian mixture models," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2011, pp. 3728–3731.

[39] ——, "Statistical compressed sensing of gaussian mixture models," *IEEE Transactions on Signal Processing*, vol. 59, no. 12, pp. 5842–5858, Dec 2011.

[40] D. Zoran and Y. Weiss, "Scale invariance and noise in natural images," in *IEEE 12th International Conference on Computer Vision (ICCV)*, Sept 2009, pp. 2209–2216.

[41] F. Chen, L. Zhang, and H. Yu, "External patch prior guided internal clustering for image denoising," in *IEEE International Conference on Computer Vision (ICCV)*, Dec 2015, pp. 603–611.

[42] S. H. Chan, E. Luo, and T. Q. Nguyen, "Adaptive patch-based image denoising by EM-adaptation," in *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Dec 2015, pp. 810–814.

[43] E. Luo, S. H. Chan, and T. Q. Nguyen, "Adaptive image denoising by mixture adaptation," *IEEE Transactions on Image Processing*, vol. 25, no. 10, pp. 4489–4503, Oct 2016.

[44] J. Feng, L. Song, X. Huo, X. Yang, and W. Zhang, "Image restoration via efficient Gaussian mixture model learning," in *IEEE International Conference on Image Processing (ICIP)*, Sept 2013, pp. 1056–1060.

[45] A. Ozerov, M. Lagrange, and E. Vincent, "Uncertainty-based learning of gaussian mixture models from noisy data," Research Report RR-7862, Jan 2012.

[46] A. M. Teodoro, J. M. Bioucas-Dias, and M. A. T. Figueiredo, "Image restoration and reconstruction using variable splitting and class-adapted image priors," in *IEEE International Conference on Image Processing (ICIP)*, Sept 2016, pp. 3518–3522.

[47] ——, "Image restoration with locally selected class-adapted models," in *IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*, Sept 2016, pp. 1–6.

[48] J. Xu, L. Zhang, W. Zuo, D. Zhang, and X. Feng, "Patch group based nonlocal self-similarity prior learning for image denoising," in *IEEE International Conference on Computer Vision (ICCV)*, Dec 2015, pp. 244–252.

[49] J. Xu, L. Zhang, and D. Zhang, "External prior guided internal prior learning for real noisy image denoising," *https://arxiv.org/abs/1705.04505*, 2017.

[50] X. Lu, Z. Lin, H. Jin, J. Yang, and J. Z. Wang, "Image-specific prior adaptation for denoising," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5469–5478, Dec 2015.

[51] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted Gaussian Mixture Models," *Digital Signal Processing*, vol. 10, no. 1, pp. 19 – 41, 2000.

[52] V. Papyan and M. Elad, "Multi-scale patch-based image restoration," *IEEE Transactions on Image Processing*, vol. 25, no. 1, pp. 249–261, Jan 2016.

[53] J. Sulam and M. Elad, "Expected patch log likelihood with a sparse prior," *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pp. 99–111, 2015.

[54] M. Niknejad, H. Rabbani, and M. Babaie-Zadeh, "Image restoration using Gaussian mixture models with spatially constrained patch clustering," *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 3624–3636, Nov 2015.

[55] C. Aguerrebere, A. Almansa, J. Delon, Y. Gousseau, and P. Muse, "A bayesian hyperprior approach for joint image denoising and interpolation, with an application to hdr imaging," *IEEE Transactions on Computational Imaging*, vol. PP, no. 99, pp. 1–1, 2017.

[56] L. He, H. Qi, and R. Zaretzki, "Beta process joint dictionary learning for coupled feature spaces with application to single image super-resolution," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013, pp. 345–352.

[57] Y. Xu, L. Yu, H. Xu, H. Zhang, and T. Nguyen, "Vector sparse representation of color image using quaternion matrix analysis," *IEEE Transactions on Image Processing*, vol. 24, no. 4, pp. 1315–1329, April 2015.

[58] M. Irani and S. Peleg, "Super resolution from image sequences," in *Proceedings., 10th International Conference on Pattern Recognition*, vol. ii, Jun 1990, pp. 115–120 vol.2.

[59] R. Hardie, K. Barnard, and E. Armstrong, "Joint map registration and high-resolution image estimation using a sequence of undersampled images," *IEEE Transactions on Image Processing*, vol. 6, no. 12, pp. 1621–1633, Dec 1997.

[60] M. Elad and A. Feuer, "Super-resolution reconstruction of image sequences," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 9, pp. 817–834, Sep 1999.

[61] S. Baker and T. Kanade, "Limits on super-resolution and how to break them," in *Proceedings., IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, 2000, pp. 372–379 vol.2.

[62] M. E. Tipping and C. M. Bishop, "Bayesian image super-resolution," in *Advances in Neural Information Processing Systems*. MIT Press, 2003, pp. 1303–1310.

[63] S. Farsiu, D. Robinson, M. Elad, and P. Milanfar, "Advances and challenges in super-resolution," *International Journal of Imaging Systems and Technology*, vol. 14, no. 2, pp. 47–57, 2004.

[64] S. Farsiu, M. Robinson, M. Elad, and P. Milanfar, "Fast and robust multiframe super resolution," *IEEE Transactions on Image Processing*, vol. 13, no. 10, pp. 1327–1344, Oct 2004.

[65] M. Protter, M. Elad, H. Takeda, and P. Milanfar, "Generalizing the nonlocal-means to super-resolution reconstruction," *IEEE Transactions on Image Processing*, vol. 18, no. 1, pp. 36–51, Jan 2009.

[66] Z. Lin and H.-Y. Shum, "On the fundamental limits of reconstruction-based super-resolution algorithms," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2001, pp. I–1171–I–1176.

[67] H. Hou and H. Andrews, "Cubic splines for image interpolation and digital filtering," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 26, no. 6, pp. 508–517, Dec 1978.

[68] X. Li and M. Orchard, "New edge-directed interpolation," *IEEE Transactions on Image Processing*, vol. 10, no. 10, pp. 1521–1527, Oct 2001.

[69] W.-S. Tam, C.-W. Kok, and W.-C. Siu, "A modified edge directed interpolation for images," in *17th European Signal Processing Conference, 2009*, Aug 2009, pp. 283–287.

[70] R. Keys, "Cubic convolution interpolation for digital image processing," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 29, no. 6, pp. 1153–1160, Dec 1981.

[71] M. Unser, "Splines: a perfect fit for signal and image processing," *IEEE Signal Processing Magazine*, vol. 16, no. 6, pp. 22–38, Nov 1999.

[72] J. Allebach and P. Wong, "Edge-directed interpolation," in *Proceedings., International Conference on Image Processing, 1996.*, vol. 3, Sep 1996, pp. 707–710 vol.3.

[73] K. Jensen and D. Anastassiou, "Subpixel edge localization and the interpolation of still images," *IEEE Transactions on Image Processing*, vol. 4, no. 3, pp. 285–295, Mar 1995.

[74] H. Takeda, S. Farsiu, and P. Milanfar, "Kernel regression for image processing and reconstruction," *IEEE Transactions on Image Processing*, vol. 16, no. 2, pp. 349–366, Feb 2007.

[75] V. Algazi, G. Ford, and R. Potharlanka, "Directional interpolation of images based on visual properties and rank order filtering," in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Apr 1991, pp. 3005–3008 vol.4.

[76] Q. Wang and R. Ward, "A new orientation-adaptive interpolation method," *IEEE Transactions on Image Processing*, vol. 16, no. 4, pp. 889–900, April 2007.

[77] D. Zhang and X. Wu, "An edge-guided image interpolation algorithm via directional filtering and data fusion," *IEEE Transactions on Image Processing*, vol. 15, no. 8, pp. 2226–2238, Aug 2006.

[78] H. Aly and E. Dubois, "Image up-sampling using total-variation regularization with a new observation model," *IEEE Transactions on Image Processing*, vol. 14, no. 10, pp. 1647–1659, Oct 2005.

[79] F. Malgouyres and F. Guichard, "Edge direction preserving image zooming: A mathematical and numerical analysis," *SIAM Journal on Numerical Analysis*, vol. 39, no. 1, pp. 1–37, 2001.

[80] B. Morse and D. Schwartzwald, "Image magnification using level-set reconstruction," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2001, pp. I–333–I–340 vol.1.

[81] W. Freeman, E. Pasztor, and O. Carmichael, "Learning low-level vision," *International Journal of Computer Vision*, vol. 40, no. 1, pp. 25–47, 2000.

[82] W. Freeman, T. Jones, and E. Pasztor, "Example-based super-resolution," *IEEE Computer Graphics and Applications*, vol. 22, no. 2, pp. 56–65, Mar 2002.

[83] J. Sun, J. Sun, Z. Xu, and H.-Y. Shum, "Gradient profile prior and its applications in image super-resolution and enhancement," *IEEE Transactions on Image Processing*, vol. 20, no. 6, pp. 1529–1542, June 2011.

[84] H. Chang, D.-Y. Yeung, and Y. Xiong, "Super-resolution through neighbor embedding," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, June 2004, pp. I–I.

[85] Y. Zhu, Y. Zhang, and A. Yuille, "Single image super-resolution using deformable patches," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014, pp. 2917–2924.

[86] C. Kim, K. Choi, and J. B. Ra, "Example-based super-resolution via structure analysis of patches," *IEEE Signal Processing Letters*, vol. 20, no. 4, pp. 407–410, April 2013.

[87] J. Sun, N.-N. Zheng, H. Tao, and H.-Y. Shum, "Image hallucination with primal sketch priors," in *Proceedings. 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, June 2003, pp. II–729–36 vol.2.

[88] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in *Proceedings of the 7th International Conference on Curves and Surfaces*. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 711–730.

[89] D. Glasner, S. Bagon, and M. Irani, "Super-resolution from a single image," in *IEEE 12th International Conference on Computer Vision (ICCV)*, Sept 2009, pp. 349–356.

[90] G. Freedman and R. Fattal, "Image and video upscaling from local self-examples," *ACM Trans. Graph.*, vol. 30, no. 2, pp. 1–11, Apr. 2011.

[91] M.-C. Yang and Y.-C. Wang, "A self-learning approach to single image super-resolution," *IEEE Transactions on Multimedia*, vol. 15, no. 3, pp. 498–508, April 2013.

[92] X. Gao, K. Zhang, D. Tao, and X. Li, "Joint learning for single-image super-resolution via a coupled constraint," *IEEE Transactions on Image Processing*, vol. 21, no. 2, pp. 469–480, Feb 2012.

[93] K. Jia, X. Wang, and X. Tang, "Image transformation based on learning dictionaries across image spaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 2, pp. 367–380, Feb 2013.

[94] G. Yu, G. Sapiro, and S. Mallat, "Solving inverse problems with piecewise linear estimators: From gaussian mixture models to structured sparsity," *IEEE Transactions on Image Processing*, vol. 21, no. 5, pp. 2481–2499, May 2012.

[95] J. Feng, L. Song, X. Huo, X. Yang, and W. Zhang, "Image restoration via efficient gaussian mixture model learning," in *20th IEEE International Conference on Image Processing (ICIP)*, Sept 2013, pp. 1056–1060.

[96] M. Elad, "Sparse and redundant representation modeling: What next?" *IEEE Signal Processing Letters*, vol. 19, no. 12, pp. 922–928, Dec 2012.

[97] S. Yang, M. Wang, Y. Chen, and Y. Sun, "Single-image super-resolution reconstruction via learned geometric dictionaries and clustered sparse coding," *IEEE Transactions on Image Processing*, vol. 21, no. 9, pp. 4016–4028, Sept 2012.

[98] C. V. Jiji and S. Chaudhuri, "Single-frame image super-resolution through contourlet learning," *EURASIP J. Appl. Signal Process.*, vol. 2006, pp. 235–235, Jan. 2006.

[99] K. Zhang, X. Gao, X. Li, and D. Tao, "Partially supervised neighbor embedding for example-based image super- resolution," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 2, pp. 230–239, April 2011.

[100] T.-M. Chan, J. Zhang, J. Pu, and H. Huang, "Neighbor embedding based super-resolution algorithm through edge detection and feature selection," *Pattern Recognition Letters*, vol. 30, no. 5, pp. 494 – 502, 2009.

[101] X. Gao, K. Zhang, D. Tao, and X. Li, "Image super-resolution with sparse neighbor embedding," *IEEE Transactions on Image Processing*, vol. 21, no. 7, pp. 3194–3205, July 2012.

[102] ——, "Joint learning for single-image super-resolution via a coupled constraint," *IEEE Transactions on Image Processing*, vol. 21, no. 2, pp. 469–480, Feb 2012.

[103] R. Timofte, V. De, and L. V. Gool, "Anchored neighborhood regression for fast example-based super-resolution," in *IEEE International Conference on Computer Vision (ICCV)*, Dec 2013, pp. 1920–1927.

[104] K. I. Kim and Y. Kwon, "Single-image super-resolution using sparse regression and natural image prior," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 6, pp. 1127–1133, June 2010.

[105] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *13th European Conference on Computer Vision (ECCV)*, September 2014, pp. 184–199.

[106] S. Schulter, C. Leistner, and H. Bischof, "Fast and accurate image upscaling with super-resolution forests," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 3791–3799.

[107] C. Y. Yang and M. H. Yang, "Fast direct super-resolution by simple functions," in *IEEE International Conference on Computer Vision (ICCV)*, Dec 2013, pp. 561–568.

[108] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of The Royal Statistical Society, Series B*, vol. 39, no. 1, pp. 1–38, 1977.

[109] M. R. Gupta and Y. Chen, "Theory and use of the EM algorithm," *Foundations and Trends in Signal Processing*, vol. 4, no. 3, pp. 223–296, 2011.

[110] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2095, Aug 2007.

[111] F. Luisier, T. Blu, and M. Unser, "A new SURE approach to image denoising: Interscale orthonormal wavelet thresholding," *IEEE Transactions on Image Processing*, vol. 16, no. 3, pp. 593–606, March 2007.

[112] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Color image denoising via sparse 3D collaborative filtering with grouping constraint in luminance-chrominance space," in *IEEE International Conference on Image Processing (ICIP)*, vol. 1, Sept 2007, pp. 313–316.

[113] F. Luisier and T. Blu, "SURE-LET multichannel image denoising: Interscale orthonormal wavelet thresholding," *IEEE Transactions on Image Processing*, vol. 17, no. 4, pp. 482–492, April 2008.

[114] B. Shen, W. Hu, Y. Zhang, and Y. J. Zhang, "Image inpainting via sparse representation," in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, April 2009, pp. 697–700.

[115] B. Tang, G. Sapiro, and V. Caselles, "Color image enhancement via chromaticity diffusion," *IEEE Transactions on Image Processing*, vol. 10, no. 5, pp. 701–707, May 2001.

[116] T. F. Chan, S. H. Kang, and J. Shen, "Total variation denoising and enhancement of color images based on the CB and HSV color models," *Journal of Visual Communication and Image Representation*, vol. 12, no. 4, pp. 422–435, 2001.

[117] O. Ben-Shahar and S. W. Zucker, "Hue fields and color curvatures: a perceptual organization approach to color image denoising," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, June 2003, pp. 713–720.

[118] N.-X. Lian, V. Zagorodnov, and Y.-P. Tan, "Edge-preserving image denoising via optimal color space projection," *IEEE Transactions on Image Processing*, vol. 15, no. 9, pp. 2575–2587, Sept 2006.

[119] L. Zhang and X. Wu, "Color demosaicking via directional linear minimum mean square-error estimation," *IEEE Transactions on Image Processing*, vol. 14, no. 12, pp. 2167–2178, Dec 2005.

[120] I. Pekkucuksen and Y. Altunbasak, "Gradient based threshold free color filter array interpolation," in *IEEE International Conference on Image Processing (ICIP), 2010*, Sept 2010, pp. 137–140.

[121] ——, "Multiscale gradients-based color filter array interpolation," *IEEE Transactions on Image Processing*, vol. 22, no. 1, pp. 157–165, Jan 2013.

[122] J. S. J. Li and S. Randhawa, "Color filter array demosaicking using high-order interpolation techniques with a weighted median filter for sharp color edge preservation," *IEEE Transactions on Image Processing*, vol. 18, no. 9, pp. 1946–1957, Sept 2009.

[123] I. Pekkucuksen and Y. Altunbasak, "Edge strength filter based color filter array interpolation," *IEEE Transactions on Image Processing*, vol. 21, no. 1, pp. 393–397, Jan 2012.

[124] D. Kiku, Y. Monno, M. Tanaka, and M. Okutomi, "Residual interpolation for color image demosaicking," in *IEEE International Conference on Image Processing (ICIP), 2013*, Sept 2013, pp. 2304–2308.

[125] Y. Monno, D. Kiku, M. Tanaka, and M. Okutomi, "Adaptive residual interpolation for color image demosaicking," in *IEEE International Conference on Image Processing (ICIP), 2015*, Sept 2015, pp. 3861–3865.

[126] D. Kiku, Y. Monno, M. Tanaka, and M. Okutomi, "Minimized-laplacian residual interpolation for color image demosaicking," pp. 90 230L–90 230L–8, 2014.

[127] ——, "Beyond color difference: Residual interpolation for color image demosaicking," *IEEE Transactions on Image Processing*, vol. 25, no. 3, pp. 1288–1300, March 2016.

[128] W. Ye and K. K. Ma, "Color image demosaicing using iterative residual interpolation," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5879–5891, Dec 2015.

[129] K. He, J. Sun, and X. Tang, "Guided image filtering," in *Proceedings of the 11th European Conference on Computer Vision (ECCV)*, vol. 6311, 2010, pp. 1–14.

[130] S. Roth and M. J. Black, "On the spatial statistics of optical flow," in *Tenth IEEE International Conference on Computer Vision (ICCV)*, vol. 1, Oct 2005, pp. 42–49.

[131] J. J. Mcauley, T. S. Caetano, A. J. Smola, and M. O. Franz, "Learning high-order MRF priors of color images," in *Proceedings of the 23rd International Conference on Machine Learning*, 2006, pp. 617–624.

[132] Y.-I. Ohta, T. Kanade, and T. Sakai, "Color information for region segmentation," *Computer Graphics and Image Processing*, vol. 13, no. 3, pp. 222–241, 1980.

[133] X. Li, "On modeling interchannel dependency for color image denoising: Articles," *Int. J. Imaging Syst. Technol.*, vol. 17, no. 3, pp. 163–173, Oct. 2007.

[134] N. Joshi, C. L. Zitnick, R. Szeliski, and D. J. Kriegman, "Image deblurring and denoising using color priors," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2009, pp. 1550–1557.

[135] A. Buades, B. Coll, and J. M. Morel, "A non-local algorithm for image denoising," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 2, June 2005, pp. 60–65 vol. 2.

[136] J. Dai, O. C. Au, L. Fang, C. Pang, F. Zou, and J. Li, "Multichannel nonlocal means fusion for color image denoising," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 11, pp. 1873–1886, Nov 2013.

[137] J. Dai, O. C. Au, F. Zou, C. Pang, and L. Fang, "Color image denoising based on multichannel non-local means fusion," in *19th IEEE International Conference on Image Processing (ICIP)*, Sept 2012, pp. 1193–1196.

[138] A. Rajwade, A. Rangarajan, and A. Banerjee, "Image denoising using the higher order singular value decomposition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 4, pp. 849–862, April 2013.

[139] J. Sun, J. Sun, and Z. Xu, "Color image denoising via discriminatively learned iterative shrinkage," *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 4148–4159, Nov 2015.

[140] V. B. S. Prasath, J. C. Moreno, and K. Palaniappan, "Color image denoising by chromatic edges based vector valued diffusion," *CoRR*, vol. abs/1304.5587, 2013.

[141] X. Wu and N. Zhang, "Primary-consistent soft-decision color demosaicking for digital cameras (patent pending)," *IEEE Transactions on Image Processing*, vol. 13, no. 9, pp. 1263–1274, Sept 2004.

[142] J. Duran and A. Buades, "Self-similarity and spectral correlation adaptive algorithm for color demosaicking," *IEEE Transactions on Image Processing*, vol. 23, no. 9, pp. 4031–4040, Sept 2014.

[143] J. Wu, R. Timofte, and L. V. Gool, "Demosaicing based on directional difference regression and efficient regression priors," *IEEE Transactions on Image Processing*, vol. 25, no. 8, pp. 3862–3874, Aug 2016.

[144] F. Zhang, X. Wu, X. Yang, W. Zhang, and L. Zhang, "Robust color demosaicking with adaptation to varying spectral correlations," *IEEE Transactions on Image Processing*, vol. 18, no. 12, pp. 2706–2717, Dec 2009.

[145] B. K. Gunturk, Y. Altunbasak, and R. Mersereau, "Color plane interpolation using alternating projections," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 4, May 2002, pp. IV–3333–IV–3336.

[146] P. Sandeep and T. Jacob, "Single image super-resolution using a joint GMM method," *IEEE Transactions on Image Processing*, vol. 25, no. 9, pp. 4233–4244, Sept 2016.

[147] H. S. Mousavi and V. Monga, "Sparsity-based color image super resolution via exploiting cross channel constraints," *IEEE Transactions on Image Processing*, vol. 26, no. 11, pp. 5094–5106, Nov 2017.

[148] P. E. Debevec and J. Malik, "Recovering high dynamic range radiance maps from photographs," in *ACM SIGGRAPH 2008 Classes*, ser. SIGGRAPH '08. New York, NY, USA: ACM, 2008, pp. 31:1–31:10.

[149] M. A. Robertson, S. Borman, and R. L. Stevenson, "Dynamic range improvement through multiple exposures," in *Proceedings of International Conference on Image Processing (ICIP)*, vol. 3, 1999, pp. 159–163 vol.3.

[150] M. Song, D. Tao, C. Chen, J. Bu, J. Luo, and C. Zhang, "Probabilistic exposure fusion," *IEEE Transactions on Image Processing*, vol. 21, no. 1, pp. 341–357, Jan 2012.

[151] N. K. Kalantari and R. Ramamoorthi, "Deep high dynamic range imaging of dynamic scenes," *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2017)*, vol. 36, no. 4, 2017.

[152] R. Gomez-Ojeda, Z. Zhang, J. Gonzalez-Jimenez, and D. Scaramuzza, "Learning-based image enhancement for visual odometry in challenging HDR environments," *CoRR*, vol. abs/1707.01274, 2017.

[153] M. Levoy, "Light fields and computational imaging," *IEEE Computer*, vol. 39, no. 8, pp. 46–55, Aug 2006.

[154] M. Levoy and P. Hanrahan, "Light field rendering," in *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*. ACM, 1996, pp. 31–42.

[155] K. Marwah, G. Wetzstein, Y. Bando, and R. Raskar, "Compressive Light Field Photography using Overcomplete Dictionaries and Optimized Projections," *ACM Trans. Graph. (Proc. SIGGRAPH)*, vol. 32, no. 4, pp. 1–11, 2013.

[156] R. G. Gallager, *Stochastic Processes, Theory for Applications*. Cambridge University Press, Cambridge, UK, 2013.

[157] "Lecture notes on gaussian process regression," http://fourier.eng.hmc.edu/e161/lectures/gaussianprocess/node7.html, accessed: 2017-09-14.

[158] K. B. Petersen and M. S. Pedersen, "The matrix cookbook," nov 2012, version 20121115.

List of Publications

Journal Publications

1. **P. Sandeep** and T. Jacob, "Single Image Super-Resolution using a Joint GMM Method," in *IEEE Transactions on Image Processing*, vol. 25, no. 9, pp. 4233–4244, Sept 2016.

2. **P. Sandeep** and T. Jacob, "Joint Color Space GMMs for Color Image Denoising," **submitted** to *IEEE Transactions on Image Processing*.

3. **P. Sandeep** and T. Jacob, "Joint Color Space GMMs for CFA Demosaicking," **submitted** to *IEEE Signal Processing Letters*.

Conference Publications

1. **P. Sandeep** and T. Jacob, "Supervised Dictionary Learning for Signals from Union of Subspaces," in *International Conference on Signal Processing and Communications (SPCOM)*, July 2014, pp. 1–5.

2. **P. Sandeep** and T. Jacob, "Image Restoration from Multiple Copies: A GMM based Method," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2013, pp. 1593–1597.