

Prior-Guided Deep Neural Networks for Image Restoration Tasks

PRIOR-GUIDED DEEP NEURAL NETWORKS FOR IMAGE
RESTORATION TASKS

By Seyed Mehdi AYYOUBZADEH,

*A Thesis Submitted to the School of Graduate Studies in the Partial Fulfillment
of the Requirements for the Degree Doctor of Philosophy*

McMaster University © Copyright by Seyed Mehdi AYYOUBZADEH April 28,
2023

McMaster University

Doctor of Philosophy (2023)

Hamilton, Ontario (Department of Electrical and Computer Engineering)

TITLE: Prior-Guided Deep Neural Networks for Image Restoration Tasks

AUTHOR: Seyed Mehdi AYYOUBZADEH (McMaster University)

SUPERVISOR: Dr. Xiaolin WU

NUMBER OF PAGES: xvii, 145

This thesis work is dedicated to my beloved wife, Pegah, who has been a constant source of support and encouragement during the challenges of graduate school and life. I am genuinely thankful for having you in my life. This work is also dedicated to my parents, Ali and Farkhondeh, who have always loved me unconditionally and whose good examples have taught me to work hard for the things I aspire to achieve.

Abstract

In recent years, deep learning-based image restoration neural networks have become the methodology of choice, outperforming their traditional counterparts and being gradually adopted in all systems that process, store, or display images. Despite the apparent successes of these networks, rooms for improvements still exist, as demonstrated by this thesis; particularly in terms of restoring sharp and clean high frequency details and textures, which still remains a challenge for existing deep learning-based image restoration methods.

To overcome the said weaknesses of existing methods, we study and try to identify the common root causes for the lack of desired sharpness and clarity in the output images of those learned deep restoration models against various types of degradations. Our observations point to the necessity of incorporating into neural restoration models the priors on viewer desired high frequency constructs.

In our study, we introduce several novel techniques to investigate and utilize informative high-frequency priors. These techniques include: (i) inducing convolutional neural networks' filters to extract valuable frequency information from images via a pre-designed filter bank, (ii) modifying the loss function of the restoration model during training to prioritize high-frequency textures, (iii) incorporating an auxiliary loss function on the metadata to shape the neural network outputs according to the prior knowledge of the input images, and (iv) integrating the desired priors within the model architecture.

In our first work, we propose to put additional hand-crafted constraints on the filters in convolutional neural networks to train them in faster convergence and better performance. We encourage the convolutional neural network kernels to conform to common spatial structures and features of natural images. The proposed regularization technique aims to include structural image priors of traditional filter banks to improve the robustness and generality of convolutional neural network solutions. The usefulness of this approach is not limited to image restoration; it can also be applied to other image processing and computer vision tasks.

In our second work, we design a new training strategy to adjust the loss function of image restoration neural networks adaptively. By formulating a classical optimization problem, we are able to pinpoint the complex textures for the image restoration neural network to recover. The resulting textures can be used in the loss function of the neural network during training, which leads to better estimation of the high-frequency textures and details.

We have also researched on the problem of improving optical flow estimation. Specifically, we investigate how to increase the accuracy of deep learning based optical flow estimators. We develop a test time adaptive method that efficiently uses motion vector maps provided in H.264 encoders to alleviate the domain shift problem at the inference time. It is critical to handle the out-of-domain inferences as deep learning-based optical flow estimators are mostly trained on synthetic datasets.

Finally, we propose a novel asymmetric image compression system with a high throughput real-time encoder and a heavy-duty neural network decoder that is responsible for high rate-distortion performance. The key technical development of the above asymmetric coding system is a special image restoration network that can remove compression artifacts due to the aggressively streamlined encoder.

Acknowledgements

To begin, I'd like to thank my supervisor Dr.Xiaolin Wu for his constant guidance and patience throughout the course of this investigation. I have been fortunate to receive his valuable assistance, comments, suggestions, and support. Without him, none of the thesis work would have materialized. I would also like to show my gratitude to my committee members Dr.Shahram Shirani and Dr.Jun Chen, for their careful and thorough review of my thesis draft and valuable comments in the meetings. My deepest gratitude is to my wife, Pegah, for her great help and support during all these years. Many thanks to my parents and my parents-in-law for all their support in my journey.

Contents

Abstract	ii
Acknowledgements	iv
1 Introduction	1
1.1 Introduction	1
1.1.1 Image Restoration	1
1.1.2 Deep Learning	3
1.1.3 Challenges of Deep Learning for Image Restoration Tasks	5
1.1.4 Priors in Restoration CNNs	6
1.1.5 Contributions and Thesis Organization	8
2 Filter Bank Regularization of Convolutional Neural Networks	12
2.1 Abstract	12
2.2 Introduction	13
2.2.1 Regularization	13
2.2.2 Related Work	16
2.3 Proposed Method	17
2.3.1 Filter banks	18
2.3.2 Filter bank regularization as a Maximum A Posteriori Estimation (MAP) Problem	19
2.3.3 Kernel regularization using a filter bank	21
2.3.4 Adding orthogonality regularization	23
2.4 Experiments and Discussions	24

2.4.1	Results on MNIST benchmark	24
2.4.2	Performance evaluation on CIFAR-10 benchmark	25
2.4.3	Effect of kernel size	27
2.4.4	Results on Caltech-101	28
2.5	Conclusion	30
3	High Frequency Detail Accentuation in CNN Image Restoration	32
3.1	Abstract	32
3.2	Introduction	33
3.3	Related Work	36
3.4	FAS Properties	38
3.5	Construction of FAS	39
3.6	FANet Construction	43
3.7	Experiments and Evaluations	45
3.7.1	Super Resolution	46
3.7.2	Denoising	60
3.8	Conclusion	64
4	Test-Time Adaptation for Optical Flow Estimation Using Motion Vectors	66
4.1	Abstract	66
4.2	Introduction	67
4.3	Related Work	71
4.3.1	Optical flow estimation	71
4.3.2	Computer vision with compressed videos	72
4.3.3	Test-time adaption	73
4.4	Proposed Framework	74
4.4.1	Overview of the proposed framework	74
4.4.2	Flow2MV module	76
4.4.3	Test-time adaptation	78

4.5	Experiments	80
4.5.1	Evaluation datasets and optical flow estimators	80
4.5.2	Implementation details	80
4.5.3	Ablation study	83
4.5.4	Main results	88
4.6	Conclusion	90
5	Asymmetric Coding for Ultrahigh Throughput Encoding (ACUTE)	93
5.1	Abstract	93
5.2	Introduction	94
5.3	Deep Learning based ACUTE Paradigm	96
5.4	Compression by Downsampling and Lattice Quantization	98
5.5	SR-LQ ⁻¹ Decoder	103
5.5.1	Super-resolution Module	104
5.5.2	Soft Lattice Dequantization	105
5.5.3	Solving Linear Equations	107
5.6	Experiments and Evaluations	108
5.6.1	Experiment Setting	109
5.6.2	Results	110
5.6.3	Ablation Study	111
5.7	Conclusion	118
6	Conclusion	119
A	Chapter 2 Supplement	120
A0.1	Results on image super resolution	120
A0.2	Large Size Image Classification	120
A0.3	Results of using a VGG-derived filter bank as the regularizer	121
	Bibliography	130

List of Figures

1.1	Common degradation types (A, B, C) and the desired clean image after image restoration (D)	2
1.2	Comparison of deep learning and traditional machine learning generalization performance	4
1.3	Sequential Structure of Deep Neural Networks	4
2.1	96 convolutional kernels of size $11 \times 11 \times 3$ learned by the first convolutional layer (AlexNet) Krizhevsky et al. 2017	15
2.2	LM filter bank	19
2.3	Finding best matches in \mathcal{F} for kernels in layer l	22
2.4	CNN baseline model	23
2.5	Gabor filter bank (7 orientations and 10 frequencies)	24
2.6	Cross entropy (CE) loss on test dataset (CIFAR-10)	26
2.7	Effects of γ on the learned convolutional kernel weights (the second layer of the DCNN.)	27
2.8	Accuracy on test dataset (Caltech-101)	30
3.1	Schematic description of the FANet construction process.	44
3.2	The changes of FAS member filters during the FANet training process.	45
3.3	Objective vs. subjective performance of different methods (lower NIQE values indicate more natural hence better perceptual quality).	51
3.4	Comparison between EDSR networks trained with different metrics	52
3.5	Visual comparison of EDSR vs. FANet for $\times 4$ super resolution	53
3.6	Visual comparison of GAN vs. FANet for $\times 4$ super resolution	55

3.7	Comparison of different methods for reduced networks for $\times 4$ super resolution.	57
3.8	Results of different methods on a structured dataset (FFHQ) for $\times 4$ super resolution.	60
3.9	Samples of denoising results	63
4.1	This figure illustrates an overview of the proposed method. Our TTV-MV framework uses the MV map extracted from the compressed video, to adjust the optical flow estimator at test time. The dashed line outlines the classic way for optical flow estimation. The red block depicts the adjustment process	69
4.2	We illustrate the overlay of two consecutive frames (left), the corresponding MV map (middle), and the optical flow (right). The MV map provides a rough and sparse estimation of the optical flow	70
4.3	This figure illustrates a schematic block diagram of TTA-MV. The Flow2MV module links the optical flow prediction to the MV map. An MV loss is thus defined as the error of MV prediction. In addition, a warping loss is defined based on the optical flow prediction. We perform gradient alignment to combine gradients from the two losses into one, which is then used to update the parameters of the optical flow estimator. Green arrows represent feed-forward inference, while red arrows indicate gradient back-propagation	75
4.4	Flow2MV architecture. Flow2MV module is a simplified version of the U-Net (Ronneberger et al. 2015b) architecture. All convolutional layers except the last one have the filter size of 3×3 and followed by ReLU activation function. The last convolution layer filters are 1×1	77

4.5	Visualization of MV maps, MV validity masks and optical flow maps. We show (a) the MV maps, (b) their corresponding validity masks, and (c) the optical flows of four samples from the Sintel Final dataset. Each row represents one sample. Despite the apparent similarity between the MV and the optical flow maps, motion vectors may exist only on a portion of pixels in the whole image field, indicating the necessity of including the MV validity mask in the calculation of the MV loss.	86
4.6	Optical flow prediction vs K. We show the adaptation process over iterations K . For each iteration, we indicate the current average end point error (AEPE). The bottom right panel depicts the ground truth optical flow.	87
4.7	TTA-MV improvement in optical flow estimation over different CRFs. We can see the optical flow prediction is better for the TTA-MV FlowNet compared with the FlowNet for all the CRF levels. The improvement is particularly significant for higher CRF levels. Note that the result for CRF=10 is better than CRF=0. The reason lies in the provided information by the motion vector map. The motion vector map for CRF=10 is richer compared to CRF=0. When CRF=0, most of the pixels are encoded in intra-prediction mode. Thus the motion vector map has less information.	88
4.8	Visual comparison between the results from the same optical flow estimator with and without the TTA-MV framework (first column: result without TTA-MV, second column: result with TTA-MV, third column: ground truth)	89
5.1	Schematic diagram of the ACUTE system. The encoder is exceptionally lightweight to achieve very high throughput. The compressed data consists of a downsampled image I_{\downarrow} and a lattice quantized gradient image I_{∇} . Decompression is done by a dual task CNN model (SR-LQ ⁻¹) that performs soft gradient dequantization and superresolution.	95

5.2	Statistics of g and h. Images in the DIV2K dataset (Agustsson and Timofte 2017b; Timofte et al. 2018) are sampled. The plots are all in log scale.	99
5.3	A-Law companding functions for $\alpha = 17.62$. The input x is unnormalized. Expansion function (left) versus compression function (right).	100
5.4	Voronoi diagram of the quantizer. Shown are the boundaries of the A_2 lattice quantizer and the expanded samples of g and h	102
5.5	Schematic diagram of the ACUTE encoder. The input image is partitioned into 2×2 blocks. The gradient vector (g, h) goes through the expanding function F and then is coded by the A_2 lattice quantizer into 8 bits (the 2D lattice codebook has 256 codewords). These 8 bits are concatenated to the quantized 2×2 block average value z to form the code stream.	103
5.6	Schematic diagram of the SR-LQ⁻¹ decoder. The downsampled image I_{\downarrow} is fed to a super-resolution subnetwork to make an estimate of the original resolution. In parallel, the lattice dequantization subnetwork LQ ⁻¹ embarks on removing quantization errors in $Q(I_{\nabla})$ to restore the gradient image I_{∇} . The results of these two subnetworks are combined to refine the final reconstruction image.	104
5.7	Structural correlation between I_{\downarrow} and I_{∇}. The gradient image has both angle and magnitude which is color coded by value and hue in HSV space respectively. The presence of specific structures and shapes in I_{∇} exhibits the local correlation in the gradient image. Moreover, the similarity between I_{\downarrow} and I_{∇} shows the cross-correlation between these two images.	105

5.8	The architecture of soft lattice dequantization LQ^{-1}. LQ^{-1} extracts features from quantized downsampled image $Q(I_{\downarrow})$ and lattice quantized gradient image $Q(I_{\nabla})$, separately. These two sets of features are then fused by convolution and upsampling layers to estimate the original gradient vector (r, θ) in polar coordinates.	106
5.9	Left: lattice quantizer cells; Right: the inscribed circle for one of the cells. In the right image, P is the actual expanded gradient vector $(\mathcal{G}, \mathcal{H})$ in terms of r and θ ; the cell center P_Q is the quantized P ($\hat{\mathcal{G}}, \hat{\mathcal{H}}$).	108
5.10	Visual comparison of different compression methods for scenes of very fast motions (PSNR/SSIM).	112
5.11	Visualizing the improved precision of quantized gradient images I_{∇} by soft lattice dequantization (LQ^{-1}) module for some test samples. From left to right: the original images, quantized gradient images, improved gradient images after soft dequantization LQ^{-1} , the true gradient images.	115
5.12	PSNR of different quantizers. * means A-law companding is performed before quantization.	117
5.13	PSNR and SSIM trends vs the number of training epochs for DIV2K validation data. The red graphs are for scalar quantizer without companding. The green graphs demonstrate the higher performance of SR- LQ^{-1}	118
A1.1	121
A1.2	256 sampled filters (3×3) from pretrained VGG16	122
A1.3	123
A1.4	Example images	124
A1.5	Feature maps for the first test image (Very similar feature maps are marked with red and blue colors)	125
A1.6	Feature maps for the second test image	126
A1.7	Example images	127
A1.8	Feature maps for the first test sample (FBR in comparison with ℓ_2)	128

A1.9 Feature maps for the first test sample (FBR in comparison with ℓ_2) 129

List of Tables

2.1	Error percentage on test dataset (MNIST)	25
2.2	The effect of the kernel size in the first DCNN layer on the performance.	27
2.3	Accuracy and cross entropy loss on test dataset (CIFAR-10)	28
2.4	Accuracy and cross entropy loss on test dataset (Caltech-101)	29
3.1	Super-resolution performance results on various datasets for different accentuation level α 's ($\alpha = 0$ corresponds to the MSE loss function).	48
3.2	Comparison of various loss functions and methods (Perceptual Loss (\mathcal{L}_{VGG} (johnson2016perceptual)), SSIM Loss (\mathcal{L}_{SSIM}) Zhao et al. 2017, MS-SSIM Loss ($\mathcal{L}_{MS-SSIM}$) (Zhao et al. 2017), Adversarial Loss (\mathcal{L}_{adv}) (Sajjadi et al. 2017), Texture Loss ($\mathcal{L}_{texture}$) (Sajjadi et al. 2017)))	50
3.3	Performance numbers for reduced networks for $\times 4$ super-resolution. The numbers in brackets are changes due to network reduction.	58
3.4	Denoising performance results on various noise levels for different accentuation levels α 's ($\alpha = 0$ corresponds to the MSE loss function).	62
4.1	Specifications of employed optical flow estimation benchmark datasets	80
4.2	Ablation study on α (left) and β (right). Performances are evaluated on the Sintel Final dataset with the FlowNet	83
4.3	We report the reduction of AEPE (\downarrow) and the relative improvement compared to the baseline (\uparrow) under different settings of the proposed TTA-MV framework on the Sintel Final dataset with FlowNet	84
4.4	AEPE performances of three baseline optical flow estimators with or without TTA-MV on benchmark datasets. The relative improvement for each case is measured in percentages and shown in green	90

5.1	Comparison of different compression methods in quantitative image quality for common test image sets.	113
5.2	Comparison of various super resolution networks as the upsampler module in SR-LQ ⁻¹	116
A1.1	PSNR and SSIM on DIV2K validation dataset	120

Acronyms

<i>DCNN</i>	Deep Convolutional Neural Network
<i>CNN</i>	Convolutional Neural Network
<i>GCN</i>	Gabor Convolutional Network
<i>MLP</i>	Multi-Layer Perceptrons
<i>FBR</i>	Filter Bank Regularization
<i>MAP</i>	Maximum A Posteriori Estimation
<i>DNN</i>	Deep Neural Network
<i>DSO</i>	Double Soft Orthogonality
<i>GAN</i>	Generative Adversary Neural Network
<i>FAS</i>	Feature Accentuation Space
<i>FANet</i>	Feature Accentuation Network
<i>SSIM</i>	Structural Similarity Index
<i>MSSSIM</i>	Multi-scale Structural Similarity Index
<i>HVS</i>	Human Visual System
<i>PSNR</i>	Peak Signal-to-Noise Ratio
<i>PSD</i>	Positive Semi Definite
<i>QCQP</i>	Quadratically Constrained Quadratic Programming
<i>SDR</i>	Semidefinite Relaxation
<i>MSE</i>	Mean Squared Error
<i>UQI</i>	Universal Quality Image Index
<i>NIQE</i>	Naturalness Image Quality Evaluator
<i>TTA</i>	Test-Time Adaptation
<i>MV</i>	Motion Vector
<i>DNN</i>	Deep Neural Network
<i>CV</i>	Computer Vision
<i>AEPE</i>	Average Endpoint Error
<i>GA</i>	Gradient Alignment

<i>CRF</i>	Constant Rate Factor
<i>ACUTE</i>	Asymmetric Coding for Ultrahigh Throughput Encoding
<i>GPU</i>	Graphics Processing Unit
<i>CS</i>	Compressive Sensing
<i>GPU</i>	Graphics Processing Unit
<i>DCT</i>	Discrete Cosine Transform
<i>DVC</i>	Distributed Video Coding
<i>SR</i>	Super Resolution

Chapter 1

Introduction

1.1 Introduction

1.1.1 Image Restoration

In engineering practice, images acquired by all light-sensing modalities are far from being ideal for most of downstream applications. Through all steps of the image acquisition and representation process, including sensing, sampling, analog-to-digital conversion, compression and transmission, the original image signal is contaminated and distorted successively, causing image quality degradation. There are many degradation causes. For examples, light scattering due to the presence of aerosols in the air, sensor noises, poor illuminations, motions between the object and camera, dirty lenses, improper use of imaging equipment, etc. Some common types of image degradation are illustrated in Figure 1.1. To human viewers the most noticeable visual quality deteriorations are compression artifacts, insufficient resolution, and the effects of various noises. Image restoration is the endeavor of recovering the clean latent image from its degraded version. Image restoration algorithms are indispensable in all imaging systems and applications, playing vital roles in various imaging modalities and tasks of both human and computer vision, ranging from the professional fields of medicine, sciences, astronomy, precision engineering to the gamut of consumer applications, such as Internet, social media, entertainment, etc.



FIGURE 1.1: Common degradation types (A, B, C) and the desired clean image after image restoration (D)

The image degradation process can be modeled as a mapping that transforms a clean image into a degraded counterpart. The mapping is, in general, not bijective, i.e., the degradation process is not invertible. Consequently, this multidimensional inverse problem has an infinite number of solutions, complicating the task of image restoration. The design objective of image restoration algorithms is to find a recovered image as close as possible to the original clean image. For most applications the restored images are presented to human viewers, thus the meaningful image quality metric should be based on psychovisual perception. In other words, optimal image restoration is to find the closest point to the clean image in human perception space.

Traditionally, most degradation processes are considered linear systems followed by additive noise. Therefore the degraded image can be written as $\hat{I} = I * B + N$ where $*$ represents convolution operation, N is the additive noise, B is the convolution kernel for the degradation process, and I and \hat{I} are clean and observed degraded images respectively. To overcome the difficulty of under-determinism in solving the above inverse problem of recovering I , it is necessary to constrain the solution space using prior knowledge about clean images.

Before deep learning, many conventional machine learning methods were proposed for image restoration. They attempt to recover the clean image using the iterative maximum likelihood estimation or Bayesian approaches Hong et al. 2016; Boudjelal et al. 2018. In such processes, the priors are incorporated into the solution by a regularization

function. Different priors may be chosen for countering different types of degradation. For instance, the regularization can put some statistical constraints on the probability distribution of the latent clean image. On the other hand, prior-driven regularization can also be applied in some transform space (e.g., sparsity space).

However, employing traditional machine learning methods to solve image restoration problems has following disadvantages. First, not all degradation processes are amenable to mathematical modeling, such as inpainting. In such cases, constructing a suitable regularization (prior) is not always possible. Compared with modern deep learning methods, a more serious limitation of traditional counterparts is that the latter cannot build very large non-linear models for image restoration by taking advantage of the hardware advances in computing power and storage capacity. These powerful non-linear deep neural network models can be learnt using large training data to draw statistical inferences based on underlying distributions of the images. In contrast, due to their much limited model size compared to deep neural networks, the traditional machine learning models will prematurely saturate as the amount of training data increases, and thus incapable of fully benefiting from the big data.

1.1.2 Deep Learning

As pointed out above, a hallmark of Deep Neural Networks (DNN) is their large model size, having millions or even billions of parameters. The vast parameter space of DNNs and the freedom of network architectures allow DNNs to learn highly non-linear complex mappings, as the Vapnik-Chervonenkis dimensions for such models are very high. Therefore, the performance curve of DNNs versus the number of training data does not saturate quickly, as demonstrated in Figure 1.2.

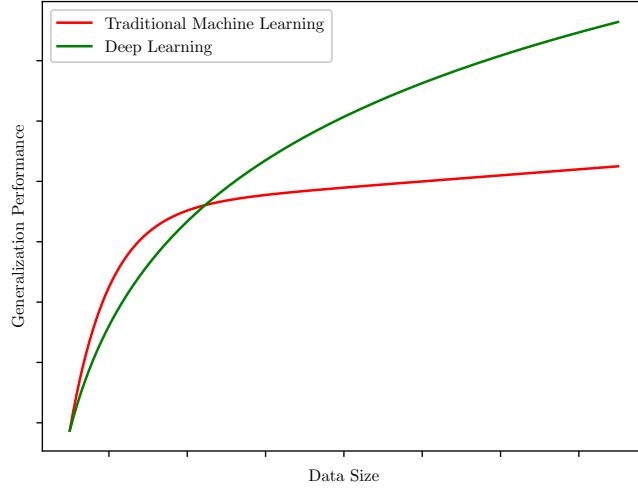


FIGURE 1.2: Comparison of deep learning and traditional machine learning generalization performance

Like in the early neural network architecture of multi-layer perceptrons (MLP), DNNs are comprised of many sequential layers, each of which transforms the previous layer's output to another representation of the input image. In a properly trained DNN, the representation of a given layer has more relevant information on the intended task than the previous layer representation (Figure 1.3).

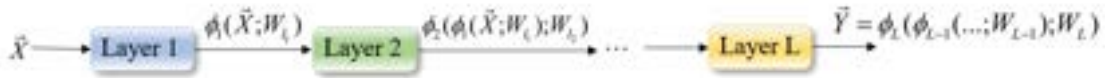


FIGURE 1.3: Sequential Structure of Deep Neural Networks

In the literature, the term "Deep" originally refers to the large number of layers of such neural networks, i.e., the network depth, and subsequently also to corresponding learning strategies and techniques for successfully training these deep neural networks.

One of the most popular DNNs is Convolutional¹ Neural Networks (CNN). In CNNs, each layer consists of some learnable filters (kernels) that extract features from the

¹the notion of convolution here is slightly different from mathematical convolution operation

previous layer output (feature map). Finally, a non-linear activation function is applied to the output of each layer to make it non-linear and increase the model complexity.

CNNs have demonstrated their superiority over MLPs in almost all computer vision and image restoration tasks. The primary difference between these two types of neural networks is in their structural organization. CNNs have neurons connected to a specific local region from the preceding layers, and the network’s weights are shared across multiple regions. Convolution layers in CNNs are designed based on two fundamental assumptions about their input. Firstly, important visual patterns should be identified regardless of their spatial location. Secondly, visual patterns tend to appear in local regions of the input. These assumptions are reasonable for image inputs since detecting edges or textures are generally useful for various computer vision and image processing tasks, irrespective of their occurrence position.

The inductive bias in the CNN architecture results in convolution layers having significantly fewer parameters than fully connected layers. As a result, deep CNNs can be trained without overfitting, leading to their superior performance compared to MLPs. The parameters of the CNNs, which are the weights of the convolution filters, are learned via backpropagation algorithm Rumelhart et al. 1986 that minimizes a loss function. For image restoration tasks, the typical loss functions are Mean Squared Error (MSE), Mean Absolute Error (MAE), SSIM, and MS-SSIM loss (Zhao et al. 2015). In other words, the CNN filters are fine tuned for the specific task and the training data, instead of being prefixed as in traditional methods of image processing and computer vision. This partially explains the superior performances of the former to the latter, particularly on image restoration and computer vision tasks (Lim et al. 2017a; Yu et al. 2018b; Tian et al. 2019; Liu et al. 2018), in the interest of this thesis.

1.1.3 Challenges of Deep Learning for Image Restoration Tasks

Despite their current successes in image restoration, CNN methods still face some technical challenges that hinder further progress of the field if not overcome.

- First, although minimizing common losses such as MSE and MAE usually results in a faster and more smooth convergence for most of the first-order optimizers, they are not well-aligned metrics for perceptual quality of the human visual system (Ayyoubzadeh and Royat 2021).
- Second, from the statistical point of view, MSE and MAE attempt to approximate the mean and median of the output image distribution given a certain input. The solution usually results in a blurry output image when the latent image has a multimodal distribution. To address this issue, the training framework of Generative Adversarial Networks (GANs) Goodfellow et al. 2014a can be employed to tune the parameters of neural networks so that the restored image agrees with the latent image in distribution space rather than in signal space. By adding the probability diversity loss of GAN in the objective function, deep learning based image restoration methods can regenerate much sharper details. However, GANs have the following weaknesses of its own. (i) GANs are notoriously difficult to train and converge. They are overly sensitive to parameters and hyperparameters. (ii) although GAN-produced results are sharp, they usually contain artifacts that are absent in natural scenes. (iii) since GANs have a mode-seeking tendency, they may get trapped into one mode of the multimodal distribution (mode collapse).
- Third, a very large amount of data is required for CNNs to prevent overfitting and have good generalization performance, which may not be always attainable for all image restoration tasks.

1.1.4 Priors in Restoration CNNs

Just like for traditional image restoration methods, effective use of priors is critical for successful recovery of the latent image. Using priors of the latent image can reduce the risk of CNN overfitting without materially limiting the CNN learning capacity. Properly chosen priors can improve the generalization performance of the CNN and make the convergence process faster and smooth by "regularizing" the hypothesis space. A common type of regularization methods assume the well-known signal sparsity prior to constrain

the network weights or outputs. For example, using l_q norm penalty favors models whose weights have small magnitude. Krogh and Hertz 1992 proposed a regularization term that penalizes large weights via the ℓ_2 norm minimization to improve generalization capability of neural networks.

Most of the regularization methods do not reduce the number of model parameters but rather control the model complexity by adjusting the variance of the model and its parameters. This is because simply reducing the number of parameters in a network risks removing the true hypothesis from the hypothesis space. Safely simplifying hypothesis space is challenging. In this regard, prior knowledge on the relationship between model parameters is particularly valuable as it can be used to reduce the number of model parameters.

Some of the regularization methods are procedural. For example, Ioffe and Szegedy 2015 proposed to perform batch normalization in each layer to reduce the internal covariate shift in the network and improve the generalization and performance of the network. Srivastava et al. 2014 used a dropout technique to regularize the network stochastically; they showed that the dropout works like an ensemble of simpler models. In Chapter 3, we have proposed a procedural regularization method that iteratively optimizes the choice of priors to prioritize the reconstruction of sharp details and textures in image restoration CNNs.

Some priors can be implicitly induced in the network architecture. For example, convolution kernels are subsets of fully connected dense layers. Two assumptions are necessary to convert fully connected dense layers to convolution kernels (which have far fewer parameters): (i) relevant input information can be extracted from pixel correlations in spatial locality; (ii) extracting the same pattern is valuable irrespective of its location in the input, which results in using the same set of weights at various locations. Many of the research works attempt to incorporate meaningful priors into image restoration networks, for example, using wavelets in the network architecture Liu et al. 2018 for image restoration as they can capture both spatial and frequency information of the

input efficiently. In Chapter 2, we have proposed a novel regularization technique for CNNs. It uses, as a prior, a set of filters known for their capability of extracting image features to regulate the shape of the learnable CNN kernels. This improves the learning efficiency by guiding weight optimization with statistically significant 2D structures of natural images.

Some priors can be represented in the form of auxiliary data or metadata. We have also investigated ways of using this type of priors in learning of image restoration. The auxiliary data can be either provided by another algorithm for another purpose or can be generated specifically to help boost the performance of the CNN. For the former, in Chapter 4, we have used the motion vectors to improve the optical flow estimation of a CNN optical flow estimator. The H.264 encoder already provides the motion vector information for compressed videos. The motion vector map contains information that can guide optical flow estimators and reduce the impact of domain shifting (Sun et al. 2020) at the inference time. For the latter, in Chapter 5, we have proposed a method to generate efficient metadata to compress images. These metadata can help the decoder to recover details and delicate textures of the original image.

1.1.5 Contributions and Thesis Organization

The thesis is in a sandwich thesis format following the terms and regulations of McMaster University. It consists of four published/submitted journal articles; we studied techniques of incorporating prior knowledge into the design and training of CNNs for various image restoration tasks to improve the performance, efficiency and robustness of these network models. Articles are listed in the preface of Chapter 2, Chapter 3, Chapter 4, and Chapter 5. Here is the reference information for the four articles:

- Seyed Mehdi Ayyoubzadeh, Xiaolin Wu, "Filter Bank Regularization of Convolutional Neural Networks".
arxiv:1907.11110. 2019 November 26.
- Seyed Mehdi Ayyoubzadeh and Xiaolin Wu, "High Frequency Detail Accentuation

in CNN Image Restoration".

IEEE Transactions on Image Processing. 10.1109/TIP.2021.3120678. 21 October 2021.

- Seyed Mehdi Ayyoubzadeh, Wentao Liu, Irina Kezele, Yuanhao Yu, Xiaolin Wu, Yang Wang and Tang Jin, "Test-Time Adaptation for Optical Flow Estimation Using Motion Vectors".

Submitted to IEEE Transactions on Image Processing.

- Seyed Mehdi Ayyoubzadeh, Xiaolin Wu and Xi Zhang, "Asymmetric Coding for Ultra-high Throughput Encoding (ACUTE)".

Submitted to IEEE Transactions on Image Processing.

In Chapter 2, we propose a novel approach to regularize DCNN convolutional kernels by utilizing a predetermined filter bank, which differs from existing methods that do not consider spatial correlations between samples in a kernel. This technique allows for molding of DCNN kernels to the spatial structures and features found in natural images. Unlike other methods, the proposed approach still permits DCNN weights to be optimized through backpropagation. This strategy combines traditional structural image priors with modern deep learning capabilities to enhance the robustness and generalization of DCNN solutions.

In Chapter 3, we propose a new framework to restore complex textures and details in image restoration tasks. Existing CNN image restoration methods suffer from blurred details, so we suggest a new training methodology to sensitize CNNs to desired events, even if they are atypical. We introduce a high-frequency feature accentuation space to promote image sharpness and clarity, and use an auxiliary loss term in training to ensure agreement between the ground truth image and the CNN-restored image in this feature accentuation space. Our proposed approach aims to penalize image blurs and has been implemented and tested for image super-resolution and denoising tasks, with experimental results demonstrating success in achieving my design objective.

In Chapter 4, we propose a method to address the domain shift issue for optical flow estimators. We suggest a self-supervised learning approach to adjust the optical flow estimation model during testing. We make use of the fact that most videos are stored in compressed formats, which provide compact information on motion in the form of motion vectors and residuals. We use the motion vector prediction as a self-supervised task and connect it to optical flow estimation. The proposed Test-Time Adaption guided with Motion Vectors (TTA-MV) is the first attempt, to our knowledge, to perform such adaptation for optical flow. The experimental results indicate that TTA-MV can enhance the generalization capability of several popular deep learning methods for optical flow estimation.

In Chapter 5, we introduce an Asymmetric Coding scheme for Ultrahigh Throughput Encoding (ACUTE), which includes a simple and fast encoder capable of compressing raw sensor data as fast as it is read out. In contrast, the ACUTE decoder is a deep decompression CNN model that can achieve good rate-distortion performance. The key innovation of this approach is gradient coding using fast 2D lattice vector quantization at the encoder, and optimized deep dequantization and super-resolution at the decoder.

Preface

The following chapter is reproduced from a paper on arXiv: Seyed Mehdi Ayyoubzadeh, Xiaolin Wu. "Filter Bank Regularization of Convolutional Neural Networks". arXiv preprint arXiv:1907.11110. 2019 November 26.

Contribution Declaration: Seyed Mehdi Ayyoubzadeh (the author of this thesis) is the first author and main contributor of this article. He proposed the method, conducted experiments and composed the article. Prof. Xiaolin Wu is the supervisor of Seyed Mehdi Ayyoubzadeh.

Chapter 2

Filter Bank Regularization of Convolutional Neural Networks

2.1 Abstract

Regularization techniques are widely used to improve the generality, robustness, and efficiency of deep convolutional neural networks (DCNNs). In this Chapter, we propose a novel approach of regulating DCNN convolutional kernels by a structured filter bank. Comparing with the existing regularization methods, such as ℓ_1 or ℓ_2 minimization of DCNN kernel weights and the kernel orthogonality, which ignore sample correlations within a kernel, the use of filter bank in regularization of DCNNs can mold the DCNN kernels to common spatial structures and features (e.g., edges or textures of various orientations and frequencies) of natural images. On the other hand, unlike directly making DCNN kernels fixed filters, the filter bank regularization still allows the freedom of optimizing DCNN weights via deep learning. This new DCNN design strategy aims to combine the best of two worlds: the inclusion of structural image priors of traditional filter banks to improve the robustness and generality of DCNN solutions and the capability of modern deep learning to model complex non-linear functions hidden in training data. Experimental results on object recognition tasks show that the proposed regularization approach guides DCNNs to faster convergence and better generalization than existing regularization methods of weight decay and kernel orthogonality.

2.2 Introduction

2.2.1 Regularization

Deep convolutional neural networks (DCNNs) have rapidly matured as an effective tool for almost all computer vision tasks (Zoph et al. 2017; Szegedy et al. 2014; Howard et al. 2017; Huang et al. 2016; He et al. 2015; Simonyan and Zisserman 2014), including object recognition, classification, segmentation, superresolution, etc. Compared with traditional vision methods based on analytical models, DCNNs are able to learn far more complex, non-linear functions hidden in the training images. However, DCNNs are also known for their high model redundancy and susceptibility to data overfitting. When having a very large number of parameters, DCNNs have high Vapnic-Chervonenkis (VC) dimension. If trained on limited amount of samples from the data generating distribution, DCNNs are less likely to choose the correct hypothesis from the large hypothesis space (Caruana et al. 2000). In other words, there should be a balance between information in the training examples and the complexity of the network (Krogh and Hertz 1992). The simplest model that could perform the task and generalize well on the real world data is the best one. But choosing the simplest model is not an easy task; simply reducing the number of parameters in a network runs the risk of removing the true hypothesis from the hypothesis space. To prevent overfitting and improve the generalization capability, a common strategy is to use a complex model but put some constraints on the model to make it overlook noise samples. In this way reducing the model complexity is not achieved by reducing the number of free parameters in the network, but rather by controlling the variance of the model and its parameters. This strategy is known as regularization (Khan et al. 2018).

Regularization methods for DCNNs fall into two categories. The regularization methods of the first category are procedural. For example, Ioffe and Szegedy proposed to perform batch normalization in each layer to reduce the internal covariate shift in the network and improve the generalization and performance of the network (Ioffe and Szegedy 2015).

Srivastava et al. 2014 used a dropout technique to stochastically regularize the network; they showed that the dropout works like an ensemble of simpler models. Khan et al. 2018 proposed a so-called Bridgeout stochastic regularization technique, and they proved that their method is equivalent to the L_q norm penalty on the weights for a generalized linear model, where norm q is a learnt hyperparameter.

All regularization methods of the first category are implicit and quite weak in the sense that they do not directly act on the CNN loss function, nor they require the convolutional kernels to have any spatial structures.

The methods in the second category explicitly add a regularization term in the loss function to penalize the CNN weights. One example is the weight decay method by Krogh and Hertz 1992 that penalizes large weights via the ℓ_2 norm minimization. Among the published methods weight decay is the most common one to regularize CNNs. For simple linear models, it can be shown, using Bayesian inference, that weight decay statistically means that the weights obey a multivariate normal distribution with diagonal covariance prior. In this case, maximum a posterior probability (MAP) estimation with Gaussian prior on the weights is equivalent to the maximum likelihood estimation with the weight decay term (Goodfellow et al. 2016). However, weight decay regularization can be justified only if the weights within a CNN kernel have no correlation with each other. This assumption is obviously false, as it is well known that the CNN kernels, upon convergence, typically exhibit strong spatial structures. To illustrate our point, the kernels of the Alexnet after training are shown in Figure 2.1, where the weights in a kernel are clearly correlated.

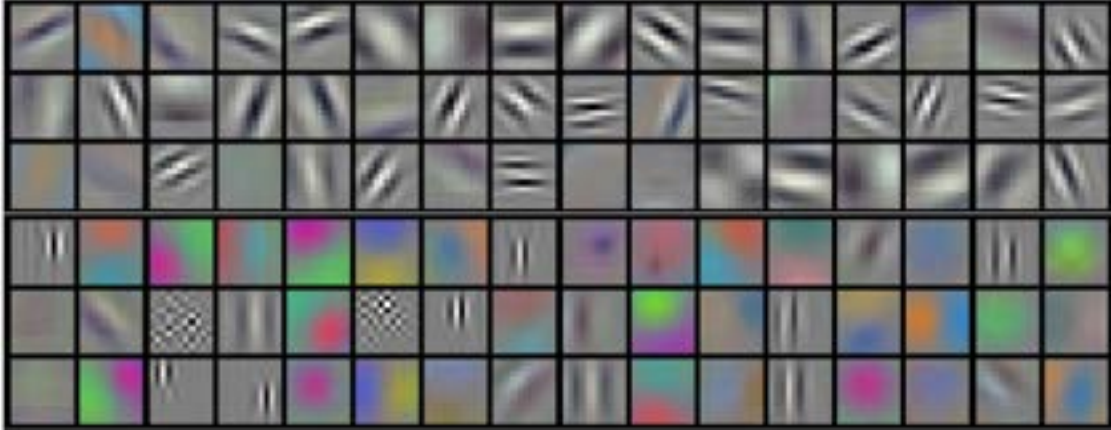


FIGURE 2.1: 96 convolutional kernels of size $11 \times 11 \times 3$ learned by the first convolutional layer (AlexNet) Krizhevsky et al. 2017

Another form of penalty term in the cost function for regularizing CNN weights is the orthogonality of the kernels (Xie et al. 2017). But the requirement of mutually orthogonal kernels also ignores the spatial structures.

In summary, all existing CNN regularization methods overlook spatial structures of images. This research sets out to rectify the above common problem. Our solution to it is a novel approach of regularizing CNN convolutional kernels by a structured filter bank. The idea is to encourage the CNN kernels to conform to common spatial structures and features (e.g., edges or textures of various orientations and frequencies) of natural images. But this is different from simply making the CNN kernels fixed structured filters; the filter bank regularization still allows the CNN filters to be fine-tuned based on input data. This new CNN design strategy aims to combine the best of two worlds: the inclusion of structural image priors of traditional filter banks to improve the robustness and generality of CNN solutions and the capability of modern deep learning to model complex non-linear functions hidden in training data. More specifically, our technical innovations are

- Considering a convolutional kernel as a set of correlated weights and penalize them based on their structural difference from adaptively chosen reference 2D filters.

- Using filter banks as guidance for the convolutional kernels of DCNNs but at the same time allowing the kernel weights to deviate from the reference filters, if so required by data.

The remainder of the Chapter is structured as follows. The next section briefly review related works. Section 3 is the main technical body of this Chapter, in which we present the details and justifications of the proposed new regularization method. In Section 4, we report experimental results on object recognition tasks. The proposed regularization approach is shown to lead to faster convergence and better generalization than existing regularization methods of weight decay and kernel orthogonality.

2.2.2 Related Work

Xie et al. [2017](#) proposed an orthonormal regularizer for each layer of the CNNs, as a means to improve the accuracy and convergence of the network. Except being free of redundancy, orthogonal kernels do not consider spatial correlation between the weights within a given kernel, and hence irrespective of spatial structures. Some attempts were made to reduce the complexity of model by including priors in the kernels. Bruna and Mallat [2012](#) proposed a method called convolutional scattering networks in which they used fixed cascaded wavelets to decompose images . Although the method had good performance on specific datasets, it reduces the capability of CNNs. The wavelet prior is too rigid to effectively characterize a great variety of unknown image structures. Similarly, Chan et al. [2014](#) proposed a network architecture called PCANet in order to create filter banks in the layers based on a PCA decomposition of input images. This method can learn convolutional kernels from the inputs, but the output cannot affect the filter bank design. This is in conflict with the design objective of DCNNs, which is to learn convolutional kernels with respect to outputs not just inputs. For instance, for classification tasks, the goal is to learn conditional probabilities of output data in relation to the input. To gain flexibility over the scattering network and also to use the wavelet features, Jacobsen et al. [2016](#) proposed a method called structured receptive fields. They make every kernel a weighted sum of filters in a fixed filter bank that

consists of Gaussian derivative basis functions. However, this method works under the assumption that every kernel filter is smooth and can be decomposed into a compact filter basis (Jacobsen et al. 2016), which may not hold in all layers. Keshari et al. 2018 published a method to learn a dictionary of filters in an unsupervised manner. Then they made DCNN convolutional kernels linear combinations of the dictionary words with weights optimized by training data. Although this method reduces number of parameters of the network significantly, it limits the network performance, because the dictionary is not fine tuned by the training dataset. The implementation of this network is not an easy task because it needs a customized backpropagation for updating the weights. By combining pre-determined filters to form DCNN kernels, both of the mentioned methods severely limit the solution space of convolutional kernels when optimizing the DCNN for the given task. Sarwar et al. 2017 proposed a combination of Gabor filters and learnable filters when choosing convolutional kernels of DCNN . For each layer they fixed some filters to be Gabor filters and allowed others to be trained. Luan et al. 2017 proposed a so-called Gabor convolutional network (GCN). The Gabor filters are used to modulate convolutional kernels of the DCNN. The modulated filter kernels are optimized via back propagation. These two methods try to take advantage of the spatial structures of Gabor filters, but they are not used in regularization as we do in this Chapter.

2.3 Proposed Method

In this section, we explain our new filter bank regularization (FBR) technique for DCNNs in detail. In the FBR method, we include in the DCNN objective function a penalty term that encourages the convolutional kernels of the DCNN to approach some member filters of a filter bank. In addition to controlling the model complexity of DCNNs to prevent overfitting and expedite convergence, the FBR strategy has a multitude of other advantages. 1. It is a way of incorporating into the DCNN design priors of spatial structures that are interpretable and effective; 2. The filter bank approach allows the DCNN kernels to be chosen from a large pool of candidate 2D filters suitable for a given computer vision application; 3. It is a general regularization mechanism that can be

applied to any DCNN architecture without any modification.

2.3.1 Filter banks

Filter banks have proven their effectiveness for extracting useful features to facilitate many computer vision tasks. Being a set of different filters a filter bank can be used as bases (often overcomplete) to decompose images into meaningful construction elements. Arguably the best known filter bank used in computer vision is Gabor filter bank, as the family of Gabor filters are noted for their power to characterize and discriminate texture and edges, thanks to their parameterization in orientation and frequency. This is why the Gabor filter bank is a main construct used in the development of our FBR method. In addition to their mathematical properties, Gabor filters can also, in view of many vision researchers, model simple cells in the visual cortex of mammalian brains (Marçelja 1980).

The generic formula of Gabor filter is:

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \exp\left(i\left(2\pi \frac{x'}{\lambda} + \psi\right)\right) \quad (2.1)$$

where:

$$\begin{aligned} x' &= x \cos(\theta) + y \sin(\theta) \\ y' &= -x \sin(\theta) + y \cos(\theta) \end{aligned}$$

Typically, the real part of this filter is used for filtering images. The Gabor filter enables us to extract orientation-dependent frequency contents of the image (Luan et al. 2017). Transforming an image with Gabor filter bank decomposes it in a way that can enhance the separation capability of the machine learning model between different classes. Also, using the Gabor filter bank may be justified cognitively as some researchers showed that simple visual cortex cells of mammals could be modeled by Gabor filters (Daugman 1985).

To further enrich the DCNN model, we augment the Gabor filters in the regularization filter bank by adding the Leung-Malik (LM) filter bank (Leung and Malik 2001) that has shown potential for extracting textures. LM filter bank consists of first and second derivatives of Gaussians at 6 orientations and 3 scales, 8 Laplacian of Gaussian (LoG) filters, and 4 Gaussians filters. This filter bank is shown in Figure 2.2.

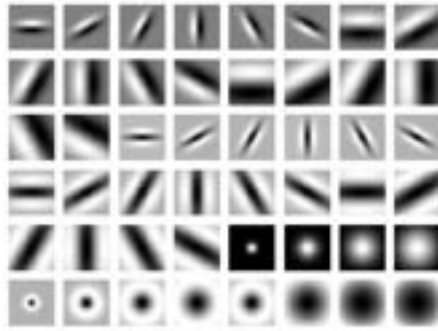


FIGURE 2.2: LM filter bank

Using filter banks can increase robustness of the model for small variations. In the context of deep learning it has been shown that first layer kernels in the DCNNs that are trained on relatively large datasets such as VGGNet, ResNet and AlexNet are very similar to Gabor filters. Scale-space theory (Witkin 1987) gives a method for convolving an image with filters that have different scales, this method can be used to extract useful descriptors from general signals. Similarly, we try to use filter banks as a guidance for CNN filters, but as previously mentioned, the filter banks are suitable for general signals, so we guide DCNN kernels to be close to the filter bank. In what follows, we give detail about the implementation of this regularization.

2.3.2 Filter bank regularization as a Maximum A Posteriori Estimation (MAP) Problem

Using Bayesian statistics is a common approach to derive the regularized loss functions. We use MAP estimation to make the Bayesian posterior tractable. Consider the simple

case of regression ($\mathbb{R}^n \rightarrow \mathbb{R}$). The model parameter is \mathbf{w} ($\mathbf{w} \in \mathbb{R}^n$). The dataset contains N pairs of datapoints denoted by $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)$, in which \mathbf{x}_i s are 1-D vectors and y_i s are scalars. In the presense of Gaussian noise we could write $y(\mathbf{x}) = \hat{y}(\mathbf{x}) + \epsilon$, where ϵ is the Gaussian noise, and \hat{y} is the model output. The conditional distribution of y can be written as follows:

$$P(y|\mathbf{x}) = \mathcal{N}(\hat{y}, I_N) \quad (2.2)$$

where I_N is the $N \times N$ identity matrix. The MAP estimation for the model parameters \mathbf{w} is defined by:

$$\begin{aligned} \mathbf{w}^* &= \underset{\mathbf{w}}{\operatorname{argmax}} P(\mathbf{w}|\mathbf{x}, y) \\ &= \underset{\mathbf{w}}{\operatorname{argmax}} \log(P(y|\mathbf{x}, \mathbf{w})) + \log(P(\mathbf{w})) \end{aligned} \quad (2.3)$$

where $P(\mathbf{w})$ is the prior distribution of the model parameters. Substituting (2) into (3) gives us:

$$\begin{aligned} \mathbf{w}^* &= \underset{\mathbf{w}}{\operatorname{argmax}} -(y - \hat{y})^T (y - \hat{y}) + \log(P(\mathbf{w})) \\ &= \underset{\mathbf{w}}{\operatorname{argmin}} \|y - \hat{y}\|_2^2 - \log(P(\mathbf{w})) \end{aligned} \quad (2.4)$$

Therefore, the cost function can be derived as follows:

$$E(\mathbf{w}) = \|y - \hat{y}\|_2^2 - \log(P(\mathbf{w})) \quad (2.5)$$

This result can be easily extended for 2-D datasets and parameters. As we discussed earlier in this paper, many researchers use the Gabor filters to model simple cells in the visual cortex of mammalian brains. We Can use this information to presume a reasonable prior distribution for the model parameters. We assume that the model parameters have a Gaussian distribution around a vectorized filter \mathbf{f} in the Gabor filter bank. In other

words, we could write the prior distribution of the parameters as follows:

$$P(\mathbf{w}) = \mathcal{N}(\mathbf{f}, \frac{1}{\lambda} I_n) \quad (2.6)$$

λ determines the deviation of the model kernel from \mathbf{f} . So, we can write the loss function as:

$$E(\mathbf{w}) = \|y - \hat{y}\|_2^2 + \lambda \|\mathbf{w} - \mathbf{f}\|_2^2 \quad (2.7)$$

2.3.3 Kernel regularization using a filter bank

Denote by $\mathcal{F} = \{\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_N\}$ a 2D filter bank of dimension $W \times H$. For a DCNN of L convolution layers, let M_l be the number of kernels in layer l , $1 \leq l \leq L$. Denote the M_l convolutional kernels of layer l by $\mathbf{k}_{l,1}, \mathbf{k}_{l,2}, \dots, \mathbf{k}_{l,M_l}$. Each kernel in the DCNN is a three-dimensional tensor of dimension $W \times H \times D_l$, where D_l is the number of channels in layer l . In each iteration of the learning process, for layer l of the DCNN we find the filter in the filter bank \mathcal{F} that best matches kernel m in channel d , namely,

$$\mathbf{f}_{l,m,d}^* = \operatorname{argmin}_{\mathbf{f} \in \mathcal{F}} \|\mathbf{f} - \mathbf{k}_{l,m,d}\|_2 \quad (2.8)$$

where $\mathbf{k}_{l,m,d}$ is the 2D cross section of the 3D kernel $\mathbf{k}_{l,m}$ in channel d . Accordingly, the FBR regularizer produces the penalty term

$$\Omega_{l,m,d} = \|\mathbf{k}_{l,m,d} - \mathbf{f}_{l,m,d}^*\|_2^2 \quad (2.9)$$

Therefore, the total loss function of the DCNN is

$$E(\mathbf{w}) = \frac{1}{N} \sum_{n=1}^N L(x^{(n)}, y^{(n)}, \mathbf{w}) + \lambda \sum_{l=1}^L \sum_{m=1}^{M_l} \sum_{d=1}^{D_l} \Omega_{l,m,d} \quad (2.10)$$

where \mathbf{w} is the total weights of the DCNN, $L(x^{(n)}, y^{(n)}, w)$ is the per sample classification loss for the input $x^{(n)}$ and corresponding output $y^{(n)}$. The interactions between the

DCNN convolutional kernels and the regularization filter bank are depicted in Figure 2.3.

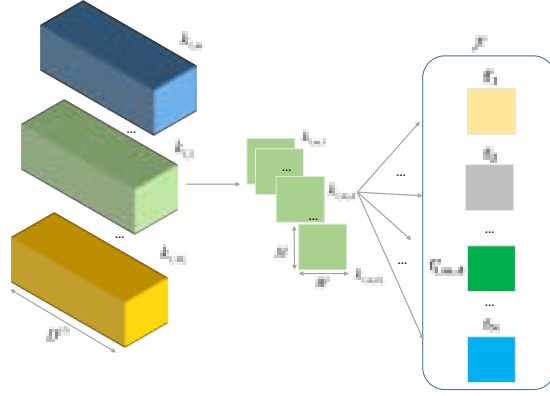


FIGURE 2.3: Finding best matches in \mathcal{F} for kernels in layer l

As one could see, in FBR, kernels choose the reference regularizer adaptively, moreover, the reference filters are well structured in the spatial domain. The proposed algorithm is shown in Algorithm 1.

Algorithm 1 DCNN regularization using FBR

```

for each iteration do
   $reg \leftarrow 0$ 
  for all layers in the DCNN do
     $l \leftarrow \text{layer index}$ 
    for all filters in  $l$ th layer do
       $m \leftarrow \text{kernel index}$ 
      for all channels in  $m$ th kernel do
         $d \leftarrow \text{channel index}$ 
         $i \leftarrow \text{argmin}_i \|k[l][m][d] - f[i]\|_2$ 
         $reg += \|k[l][m][d] - f[i]\|_2^2$ 
      end for
    end for
  end for
   $(X, Y) \leftarrow \text{Select } N \text{ random samples from the dataset}$ 
   $L_N(X, Y, \mathbf{w}) \leftarrow \text{Calculate average classification loss on } (X, Y)$ 
   $E(\mathbf{w}) \leftarrow \text{Add } reg \text{ to } L_N(X, Y, \mathbf{w})$ 
   $\mathbf{w} \leftarrow \text{Update } \mathbf{w} \text{ via backpropagation}$ 
end for

```

2.3.4 Adding orthogonality regularization

Due to the random initialization of the DCNN, it is likely that some of the kernels tend to select the same reference filter from the filter bank \mathcal{F} . This can create redundant or correlated kernels after DCNN training stage. To resolve this issue, we introduce an orthogonality regularization term (Huang et al. 2017) to encourage uncorrelated kernels. Adding the orthogonality term can change the reference regularizer filters and as a result, enables the DCNN to learn a richer set of kernels. Letting \mathbf{w}_l be the kernel weight matrix of DCNN layer l in which each column is a vectorized kernel, the orthogonality regularization term for this layer can be written as

$$\psi_l = \|\mathbf{w}_l^T \mathbf{w}_l - \mathbf{I}\|_F \quad (2.11)$$

where \mathbf{I} is the identity matrix and F denotes Frobenius norm. By adding the orthogonality regularization, we can rewrite the final loss function as follows:

$$E(\mathbf{w}) = \frac{1}{N} \sum_{n=1}^N L(x^{(n)}, y^{(n)}, \mathbf{w}) + \lambda \sum_{l=1}^L \sum_{m=1}^{M_l} \sum_{d=1}^{D_l} \Omega_{l,m,d} + \gamma \sum_{l=1}^L \psi_l \quad (2.12)$$

where ψ_l is the orthogonality regularization for layer l of the DCNN.

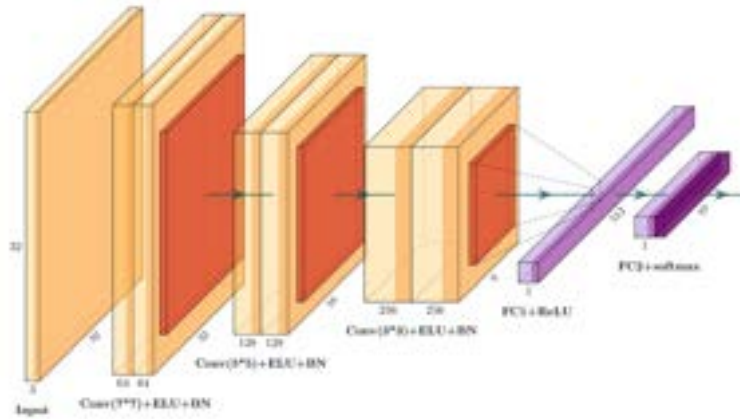


FIGURE 2.4: CNN baseline model

2.4 Experiments and Discussions

We implemented the proposed FBR method with classification DCNNs and evaluated its performances in comparison with existing regularization methods, including ℓ_1 , ℓ_2 penalty norm on the weights and pure orthogonality regularization. Two commonly used benchmark datasets CIFAR 10 (Krizhevsky et al. [n.d.](#)) and Caltech-101 (Li et al. [2003](#)) are used in our evaluations. In our experiment setup, the filter bank is the union of the Gabor and LM filter banks. These filter banks are shown in Figure [2.2](#) and [2.5](#). We designed the Gabor filter bank using 10 different orientations and 7 frequencies with $\sigma = \frac{1}{f}$ resulting 70 Gabor member filters.

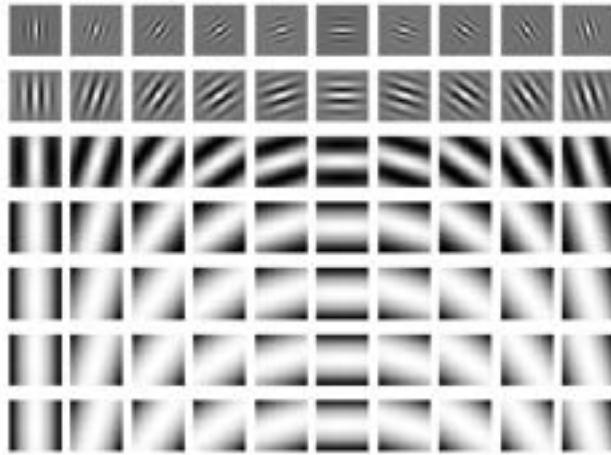


FIGURE 2.5: Gabor filter bank (7 orientations and 10 frequencies)

2.4.1 Results on MNIST benchmark

First, we report our experimental results on the benchmark dataset MNIST (LeCun and Cortes [2010](#)). We apply Double Soft Orthogonality (DSO) regularization (Bansal et al. [2018](#)) to the DCNN model. The model consists of 5 convolution layers. The first 3 convolutional layers are regularized. The learning rate of 0.001 is used, and we make it half every 10 epoch. We compare our method with Gabor Convolutional Networks

(GCNs). As one could see in table 2.1, FBR has the best accuracy with much fewer parameters in comparison with GCNs.

Reg. Type	Err. (%)	#Params (M)
Baseline	0.52	0.61
Ortho. ($\gamma = 10^{-5}$)	0.49	0.61
Ortho. ($\gamma = 0.01$)	0.53	0.61
GCN4 (with 3×3)	0.56	0.78
GCN4 (with 5×5)	0.48	1.86
GCN4 (with 7×7)	0.42	3.17
FBR ($\lambda = 0.0001, \gamma = 0.0001$)	0.40	0.61
FBR ($\lambda = 0.0001, \gamma = 0.0$)	0.34	0.61

TABLE 2.1: Error percentage on test dataset (MNIST)

2.4.2 Performance evaluation on CIFAR-10 benchmark

we report our experimental results on the benchmark dataset CIFAR-10. CIFAR-10 contains 50000 training images and 10000 testing images from 10 different categories. The images dimensions are 32×32 . The architecture that we used for DCNN is shown in Figure 2.4. We applied regularization on the 7×7 and 5×5 convolution kernels and trained the model for 300 epochs, using the RMSProp optimizer with learning rate 10^{-3} and decay of 10^{-6} . The batch size was set to 128, and data augmentation was used. We also used step decay to half the learning rate after every 25 epochs. It is worth mentioning that we employed regularization only for 4 layers of the DCNN when dimension of kernels were larger than 3×3 , in order to have an effective filter bank with reasonable representational capability. To make the comparisons fair, we used the same weight initialization for all experiments.

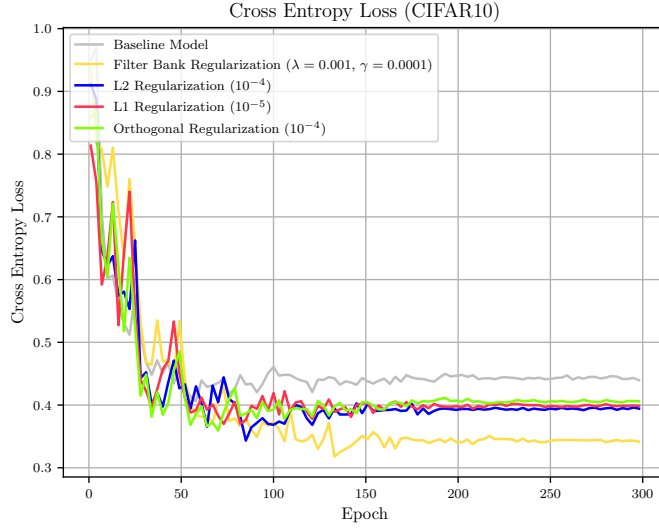


FIGURE 2.6: Cross entropy (CE) loss on test dataset (CIFAR-10)

Discussions

The cross entropy loss results of different regularization methods on the test dataset are plotted in Figure 2.6. In the figure, the baseline curve is for the classification DCNN of Figure 4 without any regularization. As shown, the FBR method has the lowest cross entropy loss (0.341) among all tested methods. An interesting observation is that, the reduction in cross entropy with respect to the baseline is almost the same for the orthogonal regularization, ℓ_1 and ℓ_2 regularization. And the FBR method can reduce the cross entropy further from the above three methods by approximately same margin. Additionally in Table 2.3, we tabulate the experiments results in more details, including both the classification accuracy and cross entropy numbers in relationship to hyperparameters λ and γ . The table demonstrates that the FBR method outperforms all other regularization methods, in both the classification accuracy and cross entropy loss for suitable λ and γ .

Also, one can see the effects of γ on the spatial structures of DCNN kernels in Figure 2.7. Emphasizing the orthogonality of the DCNN kernels can reduce the degree of kernel redundancy, i.e., preventing similar kernels from being chosen.

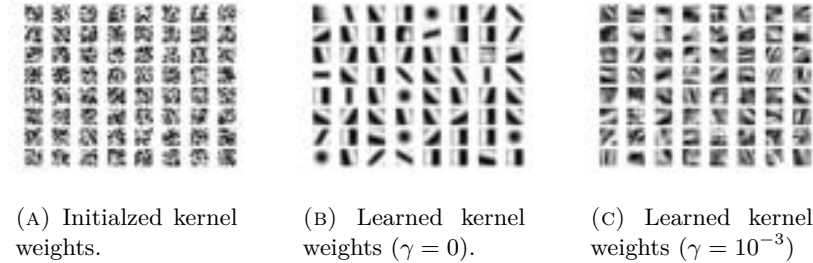


FIGURE 2.7: Effects of γ on the learned convolutional kernel weights (the second layer of the DCNN.)

2.4.3 Effect of kernel size

In the FBR method, as opposed to other regularization techniques, the DCNN kernel size affects both the DCNN architecture and the regularization filter bank \mathcal{F} . Increasing the kernel size improves the spatial resolution of the filter bank while sacrificing the locality of the feature maps. In practice, we need to trade off between the spatial resolution and the locality by varying the kernel size. Decreasing the kernel size can reduce the representational power of the filter bank, but on the other hand, improve the locality. To examine how much the kernel size can affect the DCNN model, we trained the DCNN baseline with different kernel sizes. The results of these experiments are shown in Table 2.2.

Kernel Size	Accuracy (%)	CE Loss
5	90.39	0.350
7	90.90	0.346
9	90.24	0.384
11	88.41	0.426

TABLE 2.2: The effect of the kernel size in the first DCNN layer on the performance.

As one can see, the aforementioned trade-off creates an optimal kernel size for the DCNN. In fact, this can be achieved by optimizing a hyperparameter in a cross validation approach.

Reg. Type	γ	λ	L_1 or L_2 Coeff.	Acc. (%)	CE Loss
Baseline	0	0	0	89.43	0.439
L_1	0	0	10^{-6}	89.93	0.417
L_1	0	0	10^{-5}	89.89	0.398
L_1	0	0	10^{-4}	87.87	0.432
L_2	0	0	10^{-5}	89.32	0.465
L_2	0	0	10^{-4}	90.75	0.394
L_2	0	0	10^{-3}	90.51	0.347
Ortho.	10^{-3}	0	0	90.55	0.393
Ortho.	10^{-4}	0	0	90.23	0.405
FBR	10^{-2}	10^{-5}	0	91.06	0.358
FBR	10^{-4}	10^{-5}	0	89.68	0.472
FBR	0	10^{-5}	0	89.14	0.435
FBR	10^{-2}	10^{-4}	0	90.89	0.374
FBR	10^{-4}	10^{-3}	0	90.7	0.341

TABLE 2.3: Accuracy and cross entropy loss on test dataset (CIFAR-10)

2.4.4 Results on Caltech-101

As discussed above, applying a large kernel to very small images like CIFAR-10 (32×32), can lead to poor locality. To avoid the problem and evaluate the performance of the FBR method with larger kernel sizes, we conduct the above experiments using the Caltech-101 dataset (Li et al. 2003) and compare different regularization methods. Caltech-101 has 101 categories and each class contains 40 to 800 images. We resized all of the images to 128×128 and used the DCNN baseline architecture with two extra max pooling at the first and third convolutional layers to control the number of DCNN parameters.

Reg. Type	γ	λ	L_1, L_2 Coeff.	Acc. (%)	CE Loss
Baseline	0	0	0	72.35	1.578
L_1	0	0	10^{-6}	74.27	1.561
L_1	0	0	10^{-5}	70.81	1.660
L_1	0	0	10^{-4}	70.62	1.505
L_1	0	0	10^{-3}	56.45	1.951
L_2	0	0	10^{-6}	71.69	1.670
L_2	0	0	10^{-5}	73.11	1.659
L_2	0	0	10^{-4}	72.84	1.597
L_2	0	0	10^{-3}	71.62	1.469
Ortho.	10^{-4}	0	0	73.54	1.567
Ortho.	10^{-3}	0	0	73.65	1.654
Ortho.	10^{-2}	0	0	75.65	1.453
Ortho.	10^{-1}	0	0	75.34	1.437
FBR	10^{-1}	10^{-5}	0	75.84	1.448
FBR	10^{-2}	10^{-5}	0	74.15	1.619
FBR	10^{-3}	10^{-4}	0	76.65	1.556
FBR	10^{-3}	$5 * 10^{-5}$	0	75.72	1.410
FBR	10^{-2}	10^{-4}	0	75.84	1.480

TABLE 2.4: Accuracy and cross entropy loss on test dataset (Caltech-101)

The experimental results with the Caltech-101 dataset are presented in Table 2.4. By comparing Table 2.4 with Table 2 (CIFAR-10), we can see that not only the FBR method achieves the best performance in the comparison group, but also its performance gain over others increases by a significant margin with larger kernel sizes and higher resolution images. In other words, the FBR method is more advantageous on high resolution images of greater variations, because it can adapt the kernel size.

The classification accuracy results on the test dataset for different methods are plotted in Figure 2.8. As one can observe in Figure 2.8, the ℓ_2 regularization method improves the generalization of the model over the baseline by a very small amount, the ℓ_1 regularization performs much better than the ℓ_2 regularization. The orthogonal regularization outperforms both ℓ_1 and ℓ_2 , because the orthogonality prevents the choice of highly

correlated kernels and promotes more diverse kernels to extract more novel features.

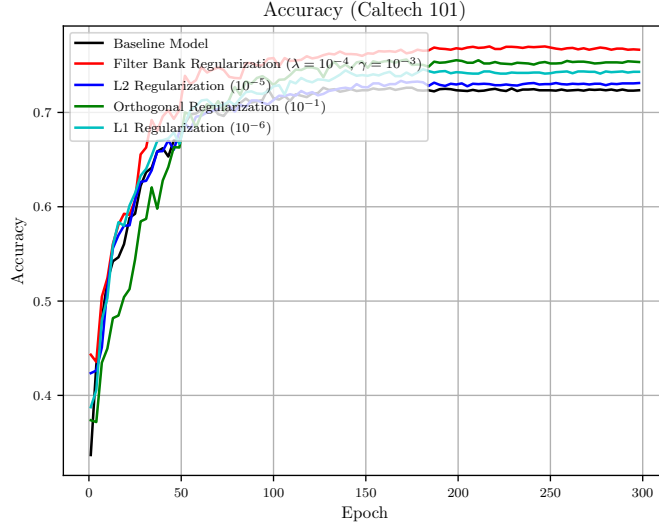


FIGURE 2.8: Accuracy on test dataset (Caltech-101)

2.5 Conclusion

Regularization techniques are widely used to prevent DCNNs from overfitting. While the importance of regularization is generally accepted, no previously existing explicit regularization techniques take into account the spatial correlation of the weights of a convolution kernel in DCNNs. This oversight has been addressed and it is corrected by our novel approach of filter bank regularization of DCNNs. This regularization approach allows us to incorporate into the network training process interpretable feature extractors such as Gabor filters to improve the convergence, robustness and generality of DCNNs.

Preface

The following chapter is a reproduction of an Institute of Electrical and Electronics Engineers (IEEE) copyrighted, published paper:

Seyed Mehdi Ayyoubzadeh and Xiaolin Wu. "High Frequency Detail Accentuation in CNN Image Restoration". IEEE Transactions on Image Processing, 21 October 2021, 10.1109/TIP.2021.3120678.

In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of McMaster University's products or services. Internal or personal use of this material is permitted. If interested in reprinting republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to <https://www.ieee.org/publications/rights> to learn how to obtain a License from RightsLink.

Contribution Declaration: Seyed Mehdi Ayyoubzadeh (the author of this thesis) is the first author and main contributor of this article. He proposed the method, conducted experiments and composed the article. Prof. Xiaolin Wu is the supervisor of Seyed Mehdi Ayyoubzadeh.

Chapter 3

High Frequency Detail Accentuation in CNN Image Restoration

3.1 Abstract

Given its nature of statistical inference, machine learning methods incline to downplay relatively rare events. But in many applications statistical outliers carry disproportional significance; they can, if being left without special treatment as of now, cause CNNs to perform unsatisfactorily on instances of interests. This is the reason why existing CNN image restoration methods all suffer from the problem of blurred details. To overcome this weakness, we advocate a new training methodology to sensitize the CNNs to desired events even they are atypical. Specifically for image restoration, we propose a so-called high frequency feature accentuation space that promotes image sharpness and clarity by maximally discriminating the ground truth image and the CNN-restored image in atypical but semantically important features. Then we force the restored image to agree with the ground truth image in the feature accentuation space by including an auxiliary loss term in the training process. This aims at a high degree of agreement of the two images on high frequency constructs such as sharp edges and fine textures, i.e., penalizes image blurs. The new CNN design method is implemented and tested for tasks of image

super-resolution and denoising. Experimental results demonstrate the achievement of our design objective.

3.2 Introduction

Thanks to rapid advances of deep learning research, convolutional neural networks (CNN) have become a ubiquitous method for image restoration and enhancement tasks, including super resolution, denoising, deblurring, etc (Yu et al. 2018b; Lim et al. 2017c; Tian et al. 2019; Nah et al. 2017). However, the existing CNN image restoration methods all have a common weakness: the restored images have blurred details or low contrast compared with the latent pristine images.

There are two reasons for the lack of fidelity in high frequency features of CNN restored images. Foremost, deep learning is an approach of statistical inference; hence CNNs, by nature, favor statistically dominant features. As low-frequency patterns have much higher probabilities of occurrence in natural images, they set a bias of smoothness. The second reason is the use of differentiable norms of error vectors in the objective functions in the CNN training. Minimizing error norms tends to average out similar image waveforms and hence smooth sharp details.

But the occurrence probability is not necessarily proportional to the level of significance in terms of semantics or subjective perception. Neuroscience studies indicate that human vision is built upon fundamental components of the scene encoded by edges (region boundaries), similar to a quick sketch drawn by an artist as an impression (Weale 1983; Zhaoping 2014). In other words, high frequency features, although having lower probability of occurrence, are nonproportionally important to perception and cognition; therefore, they should be emphasized in image restoration. The standard counter measure to mitigate the over-smoothing artifacts of the CNN restoration methods is to argument the error norm by a probabilistic divergency loss term. The latter is computed by a generative adversary neural network (GAN) (Goodfellow et al. 2014a) to penalize deviations of the signal distribution of the reconstructed image from that of the ground

truth image. But GAN introduces two problems of its own. First, it makes the training process difficult to converge (Kodali et al. 2017); second, it tends to fabricate unnatural image structures (Zhang et al. 2019). Troubled by the above weaknesses of the existing GAN methods for image restoration tasks, we set out to find a more effective technique to boost high frequency features in the CNN-restored images without introducing objectionable artifacts. We share the core idea of GANs and search for a space in which the discrimination of the output image of the CNN and the ground truth image is maximized. But unlike GANs, we do not discriminate the two images in the probability distribution space. Instead, we want to find a space in which the CNN restored image and the ground truth image exhibit the maximum discrimination with respect to desired features (e.g., high frequency spatial structures) in the pixel domain. Therefore, successfully passing the discrimination test in this space means a high degree of agreement of the two images in targeted features such as sharp edges and fine textures. The above idea leads to the main innovation of this work: the use of a so-called feature accentuation space (FAS), which is spanned by a set of spatially adaptive filters, to promote image sharpness and clarity. The member filters of FAS are designed to maximally discriminate the ground truth image and the CNN-restored image in atypical but semantically important features. These filters are optimized by sample data of the desired features, instead of being manually crafted. Also, the FAS is made to have certain properties so that it is suited for an auxiliary loss function to be combined with the main CNN objective for whatever the restoration task. The novel FAS-guided CNN restoration system is called feature accentuation network (FANet). The FAS construction is formulated as an optimization problem. This optimization problem needs to be solved multiple times during the training of the image restoration FANet. Thus, we have to have a fast solution of the underlying optimization problem to facilitate the FAS construction. One of technical contributions of this work is to convert the original optimization problem to an equivalent one that can be solved efficiently by semi-definite relaxation (Luo et al. 2010). In the design of CNNs for image restoration tasks, adding to the objective function an auxiliary loss term defined in the proposed FAS has the following advantages:

- More faithful recovery of sharp edges and fine textures in the restored images without fabricated features.
- A flexible mechanism of incorporating explicit constraints (prior knowledge) into the CNN design, by designing filters to emphasize on the high-frequency structures that are important for given tasks.
- As opposed to GANs, the training process is stable and does not depend on the architecture of the restoration CNN.

More significantly, the way we use an auxiliary loss term in a carefully chosen accentuation space suggests a new training methodology to sensitize the CNN methods to desired events even they have low probability. As all machine learning methods perform statistical inferences using large data, they tend to devote modeling resources mostly to dominant trends in the data at the expense of atypical events. For example, in natural image statistics, smooth transitions are much more common than abrupt discontinuities in the 2D image signal waveform. But in many applications of image processing and computer vision, statistical outliers in the form of rare and unique discontinuous pixel patterns often carry disproportionately important information; they warrant special attention. This research introduces a mechanism to force the CNN methods not to overlook atypical cases that are nevertheless crucial to the intended tasks. In this initial study in the above line of investigation, we focus on image super resolution and denoising tasks; however, our FANet methodology can be easily applied to other image restoration tasks such as deblurring and demoireing. The proposed strategy of sensitizing CNNs for targeted features is general and it may be explored further to boost CNN performances in solving other problems of much biased statistics. Our feature accentuation method is independent of the network architecture and can be coupled with any architecture of CNN.

3.3 Related Work

Training CNNs with imbalanced datasets (skewed data distributions) is a well-known issue for classification tasks. Masko and Hensman 2015, Mako and Henseman proposed over-sampling of the under-represented classes to mitigate this issue. In Lin et al. 2017, Lin et. al altered cross-entropy loss to derive a cost function called Focal loss to sensitize the CNN for hard examples. They have used Focal loss for object detection and shown it enforces the CNN to focus more on objects rather than the background while the background is the majority class. However, for image restoration tasks the subject of skewed datasets is quite underdeveloped. Most of the existing works add a fixed term to the loss function that does not depend on the statistical characteristics of the training data. In Johnson et al. 2016, Johnson et. al suggested an auxiliary loss term based on MSE in the high level feature representation space of the images derived by pretrained VGG network Simonyan and Zisserman 2014. They called it perceptual loss. There are three concerns about perceptual loss for image restoration tasks: (i) if the distribution of the training images are different from the distribution of the pretrained VGG, then employing this loss is illogical. (ii) One of the incentives about using CNNs is that they are shift invariant to some degree. Therefore, the alignment in the high level feature representation space does not guarantee the alignment in the high frequency components of the signals. (iii) This high level feature representation is fixed and it does not depend on the training data. In Liu et al. 2021, Liu et al. show that the perceptual loss function can be computed using the networks without any training. However, their so-called Generic Perceptual Loss still suffers from the same weakness as perceptual loss for image restoration tasks.

In Pandey et al. 2018, Krishna et al. have proposed an auxiliary loss function in the edge space for single image super resolution task. They have applied Canny operator to derive the edge map of high-resolution images and ground truths, then computed MSE on these maps. This loss function can recover more details in comparison with MSE. However, this high frequency domain is not optimized based on the outputs of the CNN. Particularly, different frequencies and textures are not considered in designing this loss.

Some researchers have used the loss function of Generative Adversarial Networks (GANs) (GAN loss) Goodfellow et al. 2014b for super-resolution task Ledig et al. 2016 as the auxiliary loss function. Besides the disadvantages of GAN loss for image restoration tasks previously mentioned in 5.2, the GAN discriminator typically has a complex architecture. The training time increases considerably since the only practical approaches to train the CNNs are the first-order optimization methods. In Nazeri et al. 2019 Nazeri et al. used two stages of adversarial networks for single image super resolution task. One adversarial stage is used for edge enhancement and the other for image completion. The edge enhancement stage tries to match the distribution of the outputs edges to the distribution of training data edges. In Yang et al. 2021, Yang et al. pretrained a GAN network and then tried to embed it into another CNN design for face restoration task. In their proposed loss function, one goal is to minimize the difference between the outputs of the discriminator for authentic and restored images (L_F). In fact, L_F is similar to the perceptual loss, but the distance is measured in a discriminator space rather than by a pre-trained network. Both above methods also suffer from the same issues of GANs including unintended artifacts and training complexity. In Zhao et al. 2015, Zhao et al. used structural similarity index (SSIM) and multi-scale structural similarity index (MS-SSIM) as the loss function of image restoration CNNs. SSIM and MS-SSIM are designed to be more aligned with the sensitivity of Human Visual System (HVS). Using SSIM and MS-SSIM as the loss function of the image restoration CNN can lead to more pleasing results for HVS, but still the loss function fails to stress high-frequency details.

Before the deep learning counterpart, there were traditional methods that tried to preserve sharpness of high-frequency features. Banham and Katsaggelos 1996 proposed filtering of the images in the 2-D wavelet domain. They adjusted the parameters of the filters based on the local information to restore sharp edges. In Naik and Patel 2013, Naik and Patel proposed a method to promote image sharpness for single image super-resolution and denoising. Their algorithm iteratively used wavelet and spatial domains to minimize the reconstruction error of the back-projected image. In Li et al. 2014, Li et al. proposed a method to enhance an image based on a dictionary learning. Their

method learnt a dictionary for each block of the image separately. Finally, they try to reconstruct the enhanced image by using adjusted dictionaries for each block. These methods were typically tailored for some niche applications. Very recently, Liu et al. developed a hybrid method that combines traditional and CNN approaches. The idea was to use a Wiener-type filter to produce a cartoon-like clean-and-sharp version of the latent image to replace the ground truth in CNN training (Liu et al. 2020). Some authors studied the effects of different error norms on perceptual image quality, which is related to the sharpness of details (Zhao et al. 2017; Seif and Androutsos 2018). For ℓ_p error norm, a larger value of p exerts a heavier penalty on large errors, regardless of the structure of the image. Which p is most suited for visual quality depends on image structures and on the space in which the error is calculated. This is one of the reasons for us to advocate FAS. Other papers were published to discuss the high-frequency representation learning for image restoration, such as MWCNN (Liu et al. 2018) and ENet (Sajjadi et al. 2017). In terms of basic mechanism, the proposed FAS method, which will be detailed in the next section, is similar to Liu et al.’s Orthogonal Network (Liu et al. 2019a); however, their focus is on network pruning/acceleration in image classification.

3.4 FAS Properties

The FAS is the central piece of the proposed CNN image restoration system, because minimizing a loss function in this space promotes the sharpness and clarity of high-frequency details in the restored images. To this end we need to construct the FAS in a way such that it manifests even small differences between the CNN recovered image and the ground truth image in high frequency domain. The FAS is represented by a set of basis filters ($\mathcal{F} = \{\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_M\}$). The filters in this filter bank have the following three key properties:

- Each filter should have band pass or high pass frequency characteristic. This is a necessary feature of the filters in order to extract or emphasize fine details and

textures of the images. In absence of the DC component, these filters should satisfy

$$\sum_j \mathbf{f}_{m,j} = 0 \quad \forall m \quad (3.1)$$

where $\mathbf{f}_{m,j}$ is the j th element of \mathbf{f}_m .

- In order to minimize the redundancy between the member filters, or require each member filter to carry new information, we would like to make the set of filters $\{\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_M\}$ a basis that is as orthogonal as possible, namely,

$$|\mathbf{f}_i^T \mathbf{f}_j| \leq \epsilon \quad \forall i, j, i \neq j \quad (3.2)$$

With this property, vectors f_1, f_2, \dots, f_M will span the high frequency domain efficiently, and thus they can represent a rich set of sharp spatial patterns that the existing CNN methods are somewhat inept.

- The filters are learnt from the training data rather than predetermined by an artificial design.

the goal is to learn/discover high frequency structures in natural images in general, or a targeted class of images in particular.

- Finally, we want to make FAS unitary so it is invariant to energy level of member filters. In other words, all basis filters need to be of unit norm:

$$\|\mathbf{f}_m\|_2^2 = 1 \quad \forall m \quad (3.3)$$

In this way the FAS filters preserve the energy of the signals.

3.5 Construction of FAS

In this section, we formulate the construction of the FAS as an optimal filter bank design problem. The design objective is to make the estimated and ground truth images

maximally differ from each other in the high frequency domain. Let \mathbf{y} and $\hat{\mathbf{y}}$ be the ground truth and the output of the CNN in pixel patch of size $W \times H$, for a fixed filter bank size ($|\mathcal{F}| = M$), the filter bank \mathcal{F}^* constituting the FAS is determined by

$$\mathcal{F}^* = \underset{\mathcal{F}}{\operatorname{argmax}} \sum_{m=1}^M \sum_{n=1}^{N_s} \|\mathbf{f}_m * (\mathbf{y}_n - \hat{\mathbf{y}}_n(\mathbf{w}))\|_2^2$$

subject to

$$\begin{aligned} \sum_j \mathbf{f}_{m,j} &= 0 \quad \forall m \\ |\mathbf{f}_i^T \mathbf{f}_j| &\leq \epsilon \quad \forall i, j, i \neq j \\ \|\mathbf{f}_m\|_2^2 &= 1 \quad \forall m \end{aligned} \tag{3.4}$$

where \mathbf{w} is the parameters of the CNN and N_s is a small fraction of the total number of the training data (N). The size of each member of the filter bank is k^2 . As the FAS is much smaller than CNN in terms of the number of parameters, the optimization of the constraining FAS is less prone to overfitting than the optimization of the network itself; a much smaller amount of training data is sufficient to design the FAS. A standard way of solving the non-convex optimization problem Eq(3.4) is the interior-point (IP) method. However, the IP method for non-convex problems is inefficient and time consuming.

One of our main contributions in this paper is to convert the optimization problem Eq(3.4) to a form that can be solved more efficiently using mathematical manipulation. This step is necessary since the optimization problem for determining the FAS is required to be solved repeatedly. We transform and simplify Eq(3.4) in a way so that the FAS construction problem can be solved efficiently by the Semi-Definite Relaxation (SDR) method (Luo et al. 2010).

In the following, we outline the required steps for this transformation. We start by simplifying the objective function. To write the objective function of Eq(3.4) in a more

compact form, let Y_n denotes $\mathbf{y}_n - \hat{\mathbf{y}}_n$. Therefore, the objective function is:

$$\sum_{m=1}^M \sum_{n=1}^{N_s} \|\mathbf{f}_m * Y_n\|_2^2, \mathbf{f}_m \in \mathbb{R}^{k^2}, Y_n \in \mathbb{R}^{W \times H} \quad (3.5)$$

For further simplification of Eq(3.5), it is necessary to write the convolution in the matrix multiplication form. Let D_{Y_n} denotes the doubly block circulant matrix of Y_n , the objective function can be rewritten as:

$$\sum_{m=1}^M \sum_{n=1}^{N_s} \|D_{Y_n} \mathbf{f}_m\|_2^2, \mathbf{f}_m \in \mathbb{R}^{k^2}, D_{Y_n} \in \mathbb{R}^{l \times k^2} \quad (3.6)$$

$$l = (W + k - 1) \times (H + k - 1)$$

We can simply write Eq(3.6) in the quadratic form as follows:

$$\sum_{m=1}^M \sum_{n=1}^{N_s} \mathbf{f}_m^T D_{Y_n}^T D_{Y_n} \mathbf{f}_m \quad (3.7)$$

Since $D_{Y_n}^T D_{Y_n}$ is a Positive Semi Definite (PSD) matrix, the objective function in Eq(3.7) is convex with respects to the parameters of the filters. Next, we need to simplify the orthogonality constraint in order to be able to convert the problem to the standard form. To handle this constraint, we design the FAS filters one by one and then add the orthogonality constraint. In other words, at the time when we want to design \mathbf{f}_m , $\{\mathbf{f}_1, \dots, \mathbf{f}_{m-1}\}$ are determined. In this case, the optimization problem is a nonconvex quadratically constrained quadratic programming (QCQP). To develop m th filter, we use the method described in Luo et al. 2010 to convert the inhomogeneous QCQP to the

homogeneous form.

$$\begin{aligned}
 & \underset{\mathbf{f}_m}{\text{minimize}} && - \sum_{n=1}^{N_s} \mathbf{f}_m^T D_{Y_n}^T D_{Y_n} \mathbf{f}_m \\
 & \text{subject to} && \begin{pmatrix} \mathbf{f}_m^T & t_m \end{pmatrix} \begin{pmatrix} \mathbf{f}_m \\ t_m \end{pmatrix} = 2 \quad \forall m \\
 & && t_m^2 = 1 \\
 & && \begin{pmatrix} \mathbf{f}_m^T & t_m \end{pmatrix} \begin{pmatrix} 0 & \frac{1}{2} \\ \frac{1}{2} & 0 \end{pmatrix} \begin{pmatrix} \mathbf{f}_m \\ t_m \end{pmatrix} = 0 \quad \forall m \\
 & && \begin{pmatrix} \mathbf{f}_m^T & t_m \end{pmatrix} \begin{pmatrix} 0 & \frac{\mathbf{f}_i}{2} \\ \frac{\mathbf{f}_i^T}{2} & 0 \end{pmatrix} \begin{pmatrix} \mathbf{f}_m \\ t_m \end{pmatrix} \leq \epsilon, i < m \\
 & && \begin{pmatrix} \mathbf{f}_m^T & t_m \end{pmatrix} \begin{pmatrix} 0 & \frac{\mathbf{f}_i}{2} \\ \frac{\mathbf{f}_i^T}{2} & 0 \end{pmatrix} \begin{pmatrix} \mathbf{f}_m \\ t_m \end{pmatrix} \geq -\epsilon, i < m
 \end{aligned} \tag{3.8}$$

Let $C_n = -D_n^T D_n$, $\mathbf{x}_m = \begin{pmatrix} \mathbf{f}_m \\ t_m \end{pmatrix}$ and $X_m = \mathbf{x}_m \mathbf{x}_m^T$, we can rewrite the problem as follows:

$$\begin{aligned}
 & \underset{X_m}{\text{minimize}} && \sum_{n=1}^{N_s} \text{Tr}(C_n X_m) \\
 & \text{subject to} && \text{Tr}(X_m) = 2 \\
 & && \text{Rank}(X_m) = 1 \\
 & && X_m \succeq 0 \\
 & && \text{Tr}\left(\begin{pmatrix} \mathbf{0} & \frac{1}{2} \\ \frac{1}{2} & 0 \end{pmatrix} X_m\right) = 0 \\
 & && \text{Tr}\left(\begin{pmatrix} \mathbf{0} & \frac{\mathbf{f}_i}{2} \\ \frac{\mathbf{f}_i^T}{2} & 0 \end{pmatrix} X_m\right) \leq \epsilon, i < m \\
 & && \text{Tr}\left(\begin{pmatrix} \mathbf{0} & \frac{\mathbf{f}_i}{2} \\ \frac{\mathbf{f}_i^T}{2} & 0 \end{pmatrix} X_m\right) \geq -\epsilon, i < m
 \end{aligned} \tag{3.9}$$

where $\text{Tr}()$ represents the trace operator. The problem Eq(3.9) can be solved efficiently using well-known convex optimization techniques SDR or Convex Concave Programming (Shen et al. 2016). In the worst case scenario, SDR complexity is $\mathcal{O}(\max\{m, n\}^4 n^{\frac{1}{2}} \log(\frac{1}{\epsilon}))$, where m is the number of constraints, n is the dimension of the problem and ϵ is the given solution accuracy (Luo et al. 2010).

3.6 FANet Construction

Having the FAS basis filters, we are now ready to describe how to train the feature accentuation network FANet for image restoration. In order to train FANet to learn a restoration mapping that avoids blurred high-frequency details, we add an accentuation penalty term to the objective function that is the discrepancy between the output and ground truth in FAS. As the disagreement level in FAS drops in the iterative training process, the reconstruction fidelity of the desired high-frequency patterns increases. The above FAS loss function for the CNN restoration task is:

$$\mathcal{L}_{\text{FAS}}(\mathbf{w}) = \frac{1}{MN} \sum_{n=1}^N \sum_{m=1}^M \|\mathbf{f}_m * \mathbf{y}_n - \mathbf{f}_m * \hat{\mathbf{y}}_n(\mathbf{w})\|^2 \tag{3.10}$$

where F_m is the m th filter of the FA filter bank. The final loss function of the CNN is a convex combination of the main loss \mathcal{L}_{MSE} (e.g., the ubiquitous Euclidean norm) and the FA auxiliary loss term \mathcal{L}_{FAS} :

$$L(\mathbf{w}) = (1 - \alpha)\mathcal{L}_{\text{MSE}} + \alpha\mathcal{L}_{\text{FAS}} \quad (3.11)$$

The CNN is trained by minimizing Eq(3.11) concerning its parameters via backpropagation.

The entire procedure to design the FAS and train the FANet is summarized in Figure 3.1. As shown, it is a two-stage iterative training process. In the first stage, we design the accentuation control module by solving Eq(3.9) and adjust the loss function \mathcal{L}_{FAS} of the FANet; afterwards, we sensitize the FANet to chosen textures and patterns by training it with the adjusted loss function described in Eq(3.11). We repeat this procedure to continue the training process. Note that the architecture of the restoration CNN can be any type of neural networks, as the application sees fit.

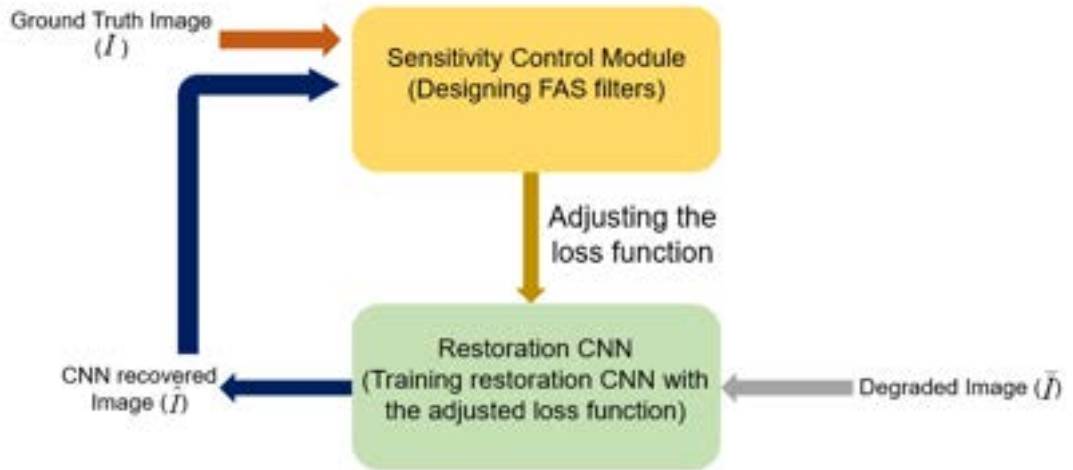


FIGURE 3.1: Schematic description of the FANet construction process.

Since EDSR (Lim et al. 2017a) is one of the best CNN architectures for image super-resolution, we adopt it in our experiments. It is helpful to appreciate the advantage of feature accentuation by observing the changes of the FAS filters during the training

process, as shown in Figure 3.2. In the initial stages of the training, outputs of the CNN lacks the capability to recover complex types of textures and details; accordingly the beginning states of the FAS filters are random bandpass and highpass. As the FANet learns to restore high-frequency details with increasing sharpness and clarity, the FAS member filters gradually adjust themselves to fit target textures of certain frequencies and orientations, which the existing CNN methods fail to recover properly.

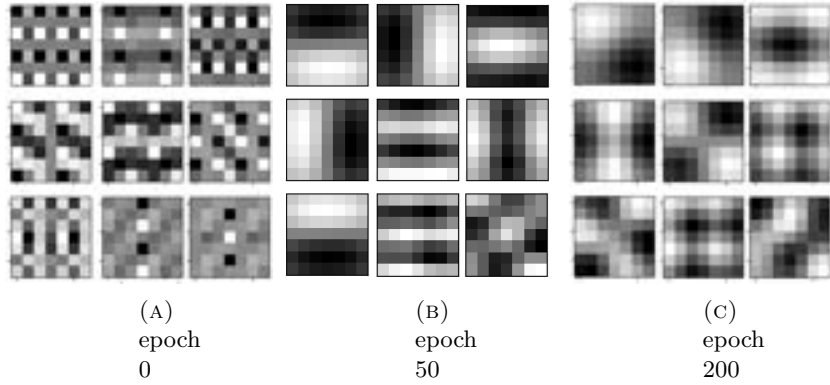


FIGURE 3.2: The changes of FAS member filters during the FANet training process.

3.7 Experiments and Evaluations

In this section, we present empirical evidences to establish the validity of our feature accentuation method and the practical value of FANet. The proposed FANet is tested and evaluated on two of the most investigated image restoration tasks: super resolution and denoising. For both tasks, we use the DIV2K dataset (Agustsson and Timofte 2017a; Timofte et al. 2017) to train the FANet. In addition to the common PSNR and SSIM image quality metrics, we introduce two other high-pass metrics to quantify the clarity or the detail sharpness of the restored images. The first metric is the so-called high frequency error E_h that is defined as below:

$$E_h(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{W \times H} \|((1 - G_\sigma) * \mathbf{y} - (1 - G_\sigma) * \hat{\mathbf{y}})\|^2 \quad (3.12)$$

where \mathbf{y} and $\hat{\mathbf{y}}$ are the ground truth and output images of the CNN respectively, G_σ is the Gaussian low-pass filter of standard deviation σ and W and H are the width and height of the image. E_h is a measure for the fidelity of restored high frequency features, such as very sharp and ultra fine details and textures.

By varying the parameter σ , we can choose the width of the high frequency subband to emphasize. For instance, increasing σ will force the resorted image to match the ground truth image on higher frequency features. The second quality metric is for the overall sharpness of restored images. It is defined to be the absolute energy level of the restored high frequency components:

$$\Delta = 20 \log_{10}(\|(1 - G_\sigma) * \hat{\mathbf{y}}\|) \quad (3.13)$$

3.7.1 Super Resolution

Experiment Setting

For the superresolution task, we accentuate the EDSR model in FAS. Adam (Kingma and Ba 2014b) optimizer is used to train the FANet of 16 residual blocks, with learning rate 10^{-4} . Specifically in our experiments, FANet for supersolution of scaling factor 4 is implemented; the paired training data are generated by bicubic downsampling process. To focus on local textures, we train FANet with relatively small square patches of width 48. This has the side benefit of faster convergence. Only a subset (100 samples) of the training set are used to design the FAS ($N_s = 100$). The training process is carried out for 200 epochs, with batch size 8. Orthogonality error (ϵ in Eq(3.4)) is set to 0.1. For solving the optimization probm 3.9, we use the CVXPY framework (Diamond and Boyd 2016; Agrawal et al. 2018). The CNN tool Keras (Chollet et al. 2015) is used in our implementation. The filter support for the FAS bases (k) is set to 7×7 and the number of filters in FAS (M) is 9. In the interest of statistical significance, we have tested the proposed FANet on as many as five different datasets: DIV2K, Set5 (Bevilacqua et al. 2012a), Set14 (Zeyde et al. 2012a), Urban100 (Martin et al. 2001a)

and BSD100 (Martin et al. 2001c). The last four datasets are unseen by the FANet at the training stage at all. The test results are tabulated for different datasets and different levels of accentuation in Table 3.1. The level of high-frequency accentuation is quantified by parameter value α in Eq 3.11. For each dataset the FANet is tested at four levels of accentuation, $\alpha = 0, 0.01, 0.1, 0.5$; for $\alpha = 0$ the FANet reduces to the original EDSR model of no accentuation. Four image quality metrics are used to evaluate the test results, the two ubiquitous metrics PSNR and SSIM, plus the two just introduced high-frequency focused metrics E_h and Δ .

Effect of accentuation coefficient (α)

As shown in the Table 3.1, the energy of the high frequency components in the restored images is higher by 0.61dB on average if feature accentuation is applied when training the restoration network than if only the MSE loss function is used. We can see that for some values of α , the FAS loss function not only improves the HFA, but also it increases PSNR value. In fact, the accentuation term acts as the regularizer of the CNN in such cases.

Dataset	α	PSNR(<i>db</i>)	SSIM	E_h	Δ (<i>db</i>)
DIV2K (x4)	0.0	27.21	0.79	88.10	39.71
	0.01	27.26	0.79	86.96	39.91
	0.1	27.22	0.79	87.44	40.12
	0.5	27.11	0.78	87.20	40.22
Urban100 (x4)	0.0	22.82	0.72	226.03	38.81
	0.01	22.83	0.73	223.14	39.11
	0.1	22.79	0.72	225.46	39.31
	0.5	22.68	0.71	227.95	39.18
BSD100 (x4)	0.0	24.87	0.68	142.07	29.72
	0.01	24.85	0.69	141.36	29.99
	0.1	24.87	0.69	140.92	30.09
	0.5	24.82	0.68	141.09	30.11
Set14 (x4)	0.0	24.48	0.71	130.22	31.72
	0.01	24.46	0.71	128.85	32.23
	0.1	24.51	0.71	128.61	32.14
	0.5	24.33	0.70	129.04	32.63
Set5 (x4)	0.0	27.54	0.83	59.28	29.19
	0.01	27.53	0.83	58.94	29.51
	0.1	27.49	0.83	58.12	29.95
	0.5	27.36	0.82	59.34	29.57

TABLE 3.1: Super-resolution performance results on various datasets for different accentuation level α 's ($\alpha = 0$ corresponds to the MSE loss function).

Evaluation of other networks and by other quality metrics

We further compare FANet with three other networks for high-frequency representation learning, ENet-PAT (Sajjadi et al. 2017), MWCNN (Liu et al. 2018) and SRGAN (Ledig et al. 2017a). We also add the EDSR network trained under various perceptual loss functions into the comparison group. In addition to PSNR and SSIM, we include the

following image quality metrics: NIQE (Mittal et al. 2013), Multi-scale Structural Similarity Index (MS-SSIM) (Wang et al. n.d.), LPIPS (Zhang et al. 2018a), and Universal Quality Image Index (UQI) (Wang and Bovik 2002). Note that, except for the ENet-PAT (Sajjadi et al. 2017) and MWCNN (Liu et al. 2018) in which we adopt the original architectures proposed by the authors, the architectures for all EDSR variants are the same as FANet. The results are shown in Table 3.2 for different datasets.

TABLE 3.2: Comparison of various loss functions and methods (Perceptual Loss (\mathcal{L}_{VGG} (**johnson2016perceptual**)), SSIM Loss (\mathcal{L}_{SSIM}) Zhao et al. 2017, MS-SSIM Loss ($\mathcal{L}_{MS-SSIM}$) (Zhao et al. 2017), Adversarial Loss (\mathcal{L}_{adv}) (Sajjadi et al. 2017), Texture Loss ($\mathcal{L}_{texture}$) (Sajjadi et al. 2017))

Network	EDSR					ENet-PAT	MWCNN	SRGAN
	\mathcal{L}_{MSE}	$\mathcal{L}_{MSE} + \mathcal{L}_{SSIM}$	$\mathcal{L}_{MSE} + \mathcal{L}_{MS-SSIM}$	$\mathcal{L}_{MSE} + \mathcal{L}_{VGG}$	$\mathcal{L}_{MSE} + \mathcal{L}_{FAS}$	$\mathcal{L}_{VGG} + \mathcal{L}_{adv} + \mathcal{L}_{texture}$	\mathcal{L}_{MSE}	$\mathcal{L}_{VGG} + \mathcal{L}_{adv}$
DIV2K								
PSNR	27.19	26.82	26.98	24.67	27.14	27.13	24.37	18.70
SSIM	0.81	0.83	0.82	0.75	0.82	0.82	0.78	0.69
MS-SSIM	0.93	0.94	0.93	0.91	0.93	0.94	0.91	0.86
Δ	39.85	40.25	39.69	41.45	40.51	39.79	41.08	39.23
UQI	0.96	0.96	0.96	0.93	0.96	0.97	0.95	0.84
LPIPS (Lower is better)	0.26	0.26	0.24	0.19	0.26	0.12	0.13	0.23
NIQE (Lower is better)	4.38	4.66	4.66	7.64	4.13	4.47	4.88	3.41
Urban100								
PSNR	22.74	22.55	22.56	21.22	22.73	22.33	19.37	17.52
SSIM	0.77	0.78	0.77	0.71	0.77	0.73	0.69	0.64
MS-SSIM	0.91	0.91	0.91	0.89	0.91	0.91	0.85	0.83
Δ	38.85	39.25	38.73	39.30	39.59	37.89	40.06	37.50
UQI	0.95	0.96	0.95	0.92	0.95	0.95	0.93	0.86
LPIPS (Lower is better)	0.24	0.25	0.24	0.20	0.25	0.20	0.22	0.20
NIQE (Lower is better)	4.18	4.41	4.45	8.33	4.22	4.63	4.98	3.57
BSD100								
PSNR	24.84	24.53	24.72	23.10	24.84	24.78	22.43	19.72
SSIM	0.73	0.75	0.74	0.67	0.73	0.71	0.69	0.64
MS-SSIM	0.90	0.90	0.90	0.87	0.90	0.90	0.86	0.86
Δ	29.94	30.48	29.81	31.89	30.56	30.41	31.29	30.65
UQI	0.97	0.97	0.97	0.95	0.98	0.97	0.96	0.90
LPIPS (Lower is better)	0.34	0.33	0.32	0.28	0.35	0.17	0.17	0.23
NIQE (Lower is better)	6.54	6.96	7.06	10.23	5.91	5.21	6.47	4.42
Set14								
PSNR	24.34	24.25	24.33	22.73	24.39	24.40	22.00	19.30
SSIM	0.76	0.78	0.78	0.70	0.77	0.75	0.73	0.68
MS-SSIM	0.91	0.92	0.92	0.89	0.91	0.91	0.88	0.87
Δ	32.16	32.33	32.17	33.48	32.89	32.25	33.14	31.71
UQI	0.96	0.97	0.97	0.95	0.97	0.97	0.95	0.89
LPIPS (Lower is better)	0.28	0.27	0.25	0.22	0.28	0.17	0.16	0.19
NIQE (Lower is better)	6.02	6.47	6.57	9.92	5.46	5.25	6.66	3.88
Set5								
PSNR	27.52	27.13	27.39	24.77	27.59	26.57	24.59	21.41
SSIM	0.88	0.89	0.88	0.81	0.88	0.85	0.83	0.77
MS-SSIM	0.96	0.96	0.96	0.95	0.96	0.95	0.93	0.92
Δ	29.26	28.88	28.55	30.03	30.18	30.42	30.04	27.32
UQI	0.97	0.97	0.97	0.90	0.97	0.97	0.96	0.84
LPIPS (Lower is better)	0.18	0.18	0.16	0.12	0.18	0.13	0.11	0.12
NIQE (Lower is better)	6.61	6.85	7.11	11.47	6.44	6.87	7.51	3.88

As we can see, FANet outperforms other methods in most of the performance metrics.

NIQE and our sharpness metric Δ are two non-reference image quality metrics, and they can be used for assessing subjective image quality. On the other hand, PSNR and SSIM are widely used objective image quality metrics. Therefore, (NIQE,PSNR) and (Δ ,SSIM) can be used as subjective vs. objective quality metric pairs to evaluate different restoration methods. Figure 3.3 compares the performances of the evaluated methods in the subjective-objective quality plane (averaged over all datasets).

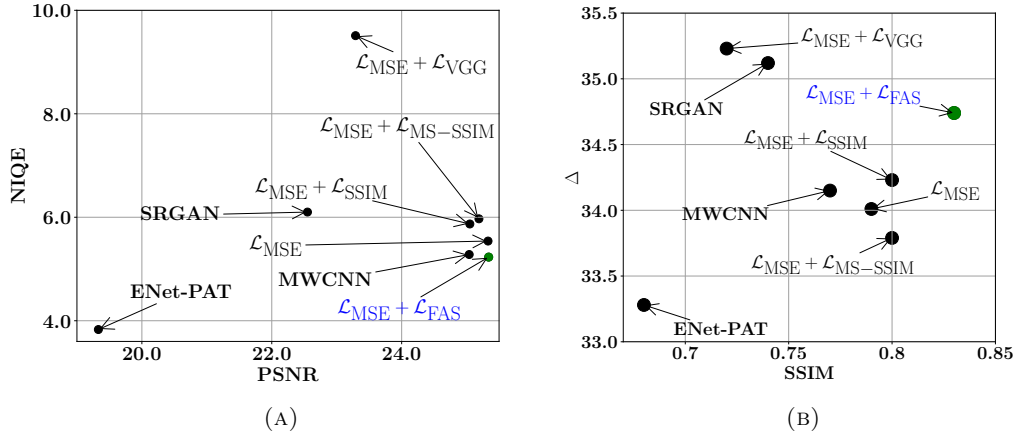


FIGURE 3.3: Objective vs. subjective performance of different methods (lower NIQE values indicate more natural hence better perceptual quality).

As illustrated, FANet strikes a better balance between the subjective and objective image quality than other methods, which typically sacrifice one to improve the other. In Figure 3.4, one can see clear advantage of using FAS loss over other loss functions; FANet can recover sharp details with a negligible amount of artifacts compared to other methods.

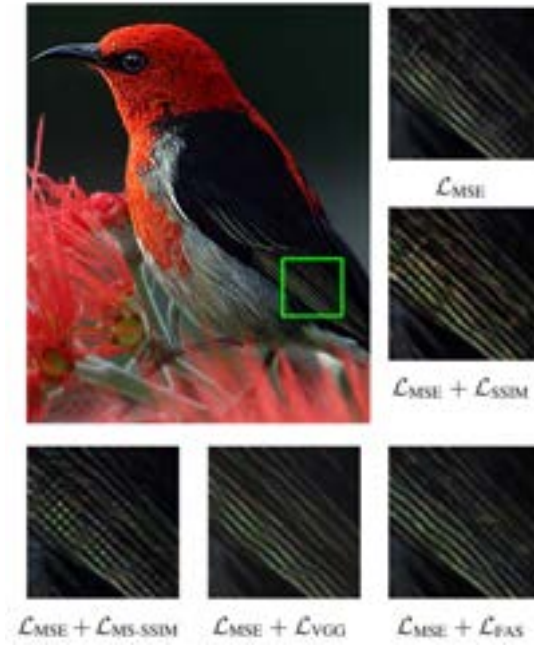


FIGURE 3.4: Comparison between EDSR networks trained with different metrics

Comparison of perceptual quality

The provided performance metrics are not always the best indicator of the visual quality of the images.

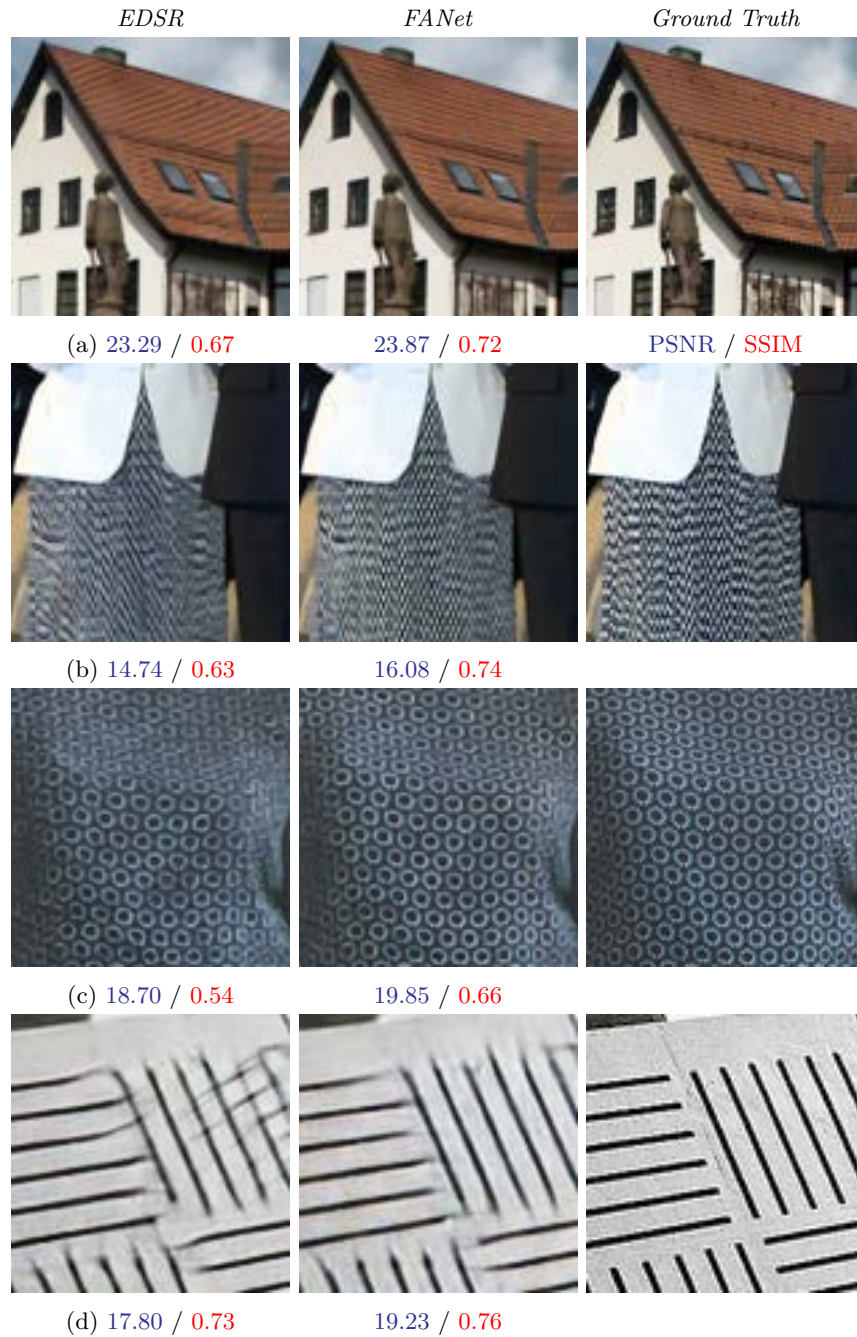


FIGURE 3.5: Visual comparison of EDSR vs. FANet for $\times 4$ super resolution

Therefore, in addition to the quantitative results, let us visually compare the results of the restoration CNN coupled with and without high-frequency feature accentuation.

We present some sample output images in Figure 3.5. As can be seen, when the images contain rich and sharp edges and textures, the CNN trained without accentuation fails to recover them, whereas FANet restores such details successfully. In Figure 3.5 (a), the EDSR trained with the MSE loss fails to recover the true slope of the lines on the roof as opposed to FANet. In fact, the plain EDSR generates false structures that do not exist at all in the original scene. In Figures 3.5 (b) and 3.5 (c), one can see that for more complex textures that are not simple lines, the plain EDSR produces blurry and alias patterns, whereas the corresponding reconstruction of FANet is far superior. Also in Figure 3.5 (d), the plain EDSR has produced much more artifacts in comparison with FANet. In all of these examples, the FAS accentuation forces the network to recover high-frequency details in order to minimize the FAS loss.

Removal of GAN artifacts by FAS

When motivating this research in the introduction, we criticised the common practice of using GAN to generate high-frequency features in image restoration CNNs. Although GAN can alleviate the problem of oversmoothing in CNN-superresolved images by implanting some details, it tends to fabricate false non-existing structures. The FANet method is proposed to fix the above problem, as a new way of restoring sharp details faithfully without the artifacts of GAN. To verify the advantage of FANet over GAN, we need to compare the super-resolution results of FANet and GANs in terms of perceptual quality. To this end, we train a GAN network, in which the generator architecture is the same as the FANet of the previous section, and the discriminator architecture is borrowed from SRGAN (Ledig et al. 2017a). In terms of network size FANet is far more compact than GAN, because the GAN discriminator is an extra part that FANet does not need. Also, the number of parameters in FAS filter bank is far fewer than the number of GAN discriminator parameters. The superresolution output images of GAN and FANet are presented in Figure 3.6. As evidently in these figures, the FANet results are visually superior to those of GAN; in particular, FANet is free of the false, objectionable structures that are fabricated by GAN. No users will accept semantically

erroneous features in the output image solely for the illusion of more details.

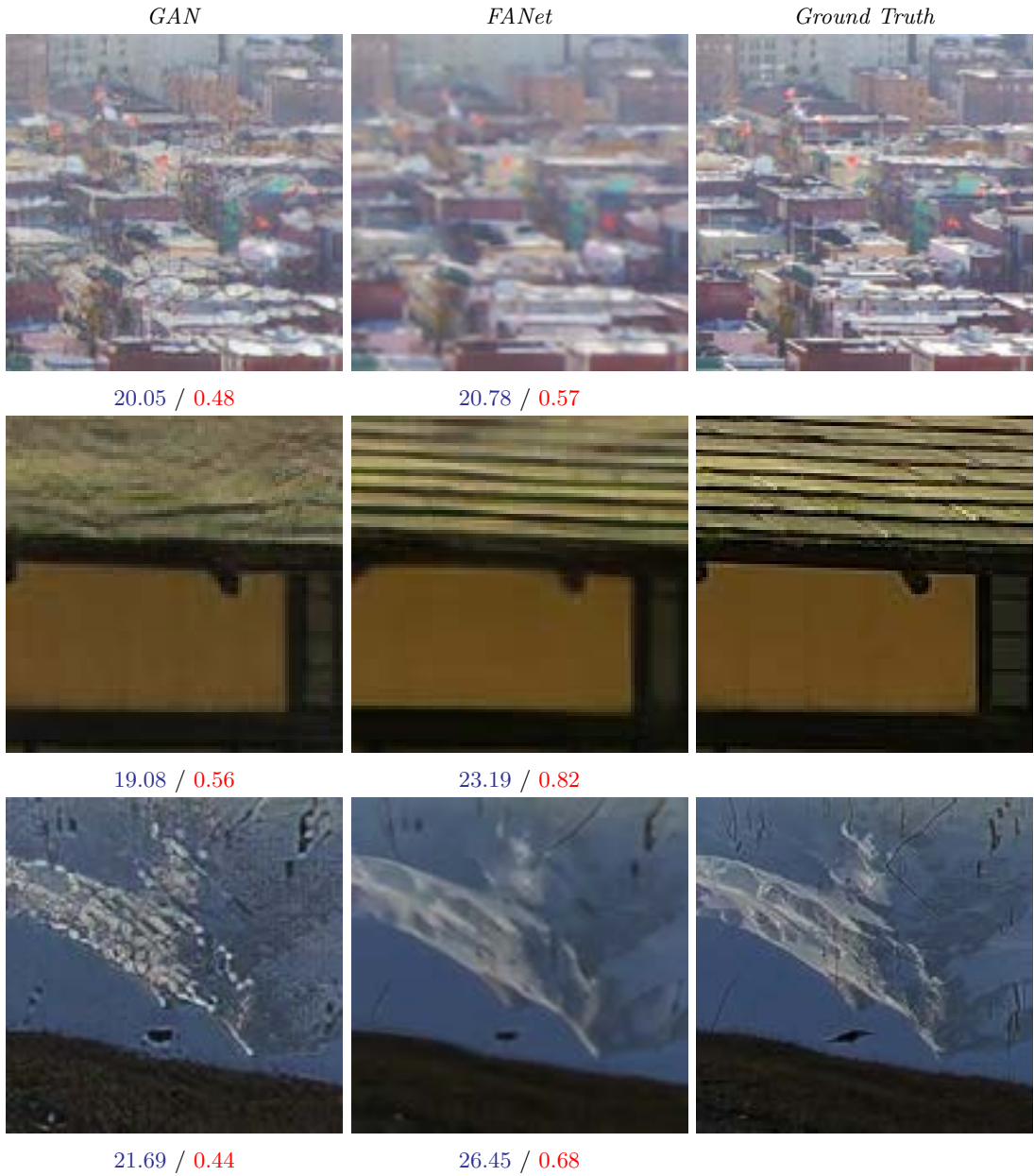


FIGURE 3.6: Visual comparison of GAN vs. FANet for $\times 4$ super resolution

Performance for lighter network architecture

To demonstrate that the successful learning of high-frequency details is primarily credited to the adoption of FAS loss function rather than a large network size, we test a lighter version of EDSR that has only 4 residual blocks and 32 filters as opposed to 16 residual blocks and 64 filters in the original model. This reduced EDSR is trained on DIV2K dataset for the $\times 4$ super-resolution task. To compare with GAN, we additionally train the GAN counterpart of the reduced network with the same generator architecture and the discriminator of SRGAN. In Figure 3.7, one can visually compare some results of this experiment. As shown, using FAS loss to train the reduced network also improves the network’s ability to recover fine details and textures. The objective quality metric values of different methods are reported in Table 3.3, in which they are compared with the counterpart numbers before network simplification. One can see that network size reduction does cause performance numbers to drop, but it does not change the relative ranking of different methods. This agrees with the visual comparison in Figure 3.7. The FAS criterion still delivers higher contrast (Δ) and lower high-frequency error (E_h) than MSE.

Note that GAN scores higher in sharpness metric Δ but has a significantly lower PSNR and SSIM. Although GAN generates high-frequency textures, it performs too poorly in objective quality metrics to keep image semantics intact.

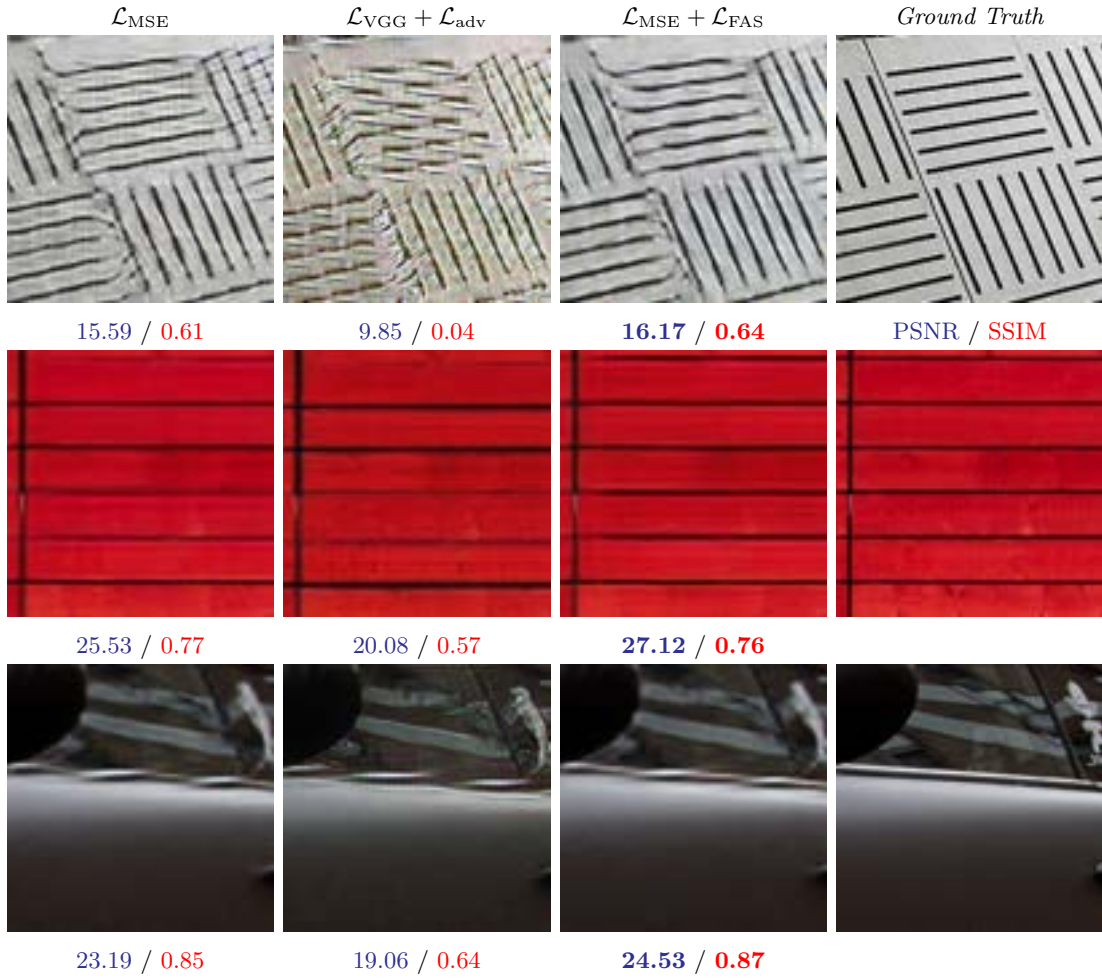


FIGURE 3.7: Comparison of different methods for reduced networks for $\times 4$ super resolution.

TABLE 3.3: Performance numbers for reduced networks for $\times 4$ super-resolution. The numbers in brackets are changes due to network reduction.

Network	EDSR		GAN
Metric	\mathcal{L}_{MSE}	$\mathcal{L}_{\text{MSE}} + \mathcal{L}_{\text{FAS}}$	$\mathcal{L}_{\text{VGG}} + \mathcal{L}_{\text{adv}}$
DIV2K			
PSNR	26.36 [-0.83]	26.35 [-0.79]	22.05
SSIM	0.77 [-0.04]	0.76 [-0.06]	0.60
E_h	92.23 [4.13]	91.76 [4.8]	98.74
Δ	37.65 [-2.2]	37.88 [-2.53]	39.77
Urban100			
PSNR	21.56 [-1.18]	21.57 [-1.16]	17.98
SSIM	0.68 [-0.09]	0.68 [-0.09]	0.47
E_h	246.14 [20.11]	245.35 [22.21]	270.4
Δ	36.69 [-2.16]	36.95 [-3.56]	38.03
BSD100			
PSNR	24.06 [-0.78]	24.06 [-0.78]	20.56
SSIM	0.66 [-0.07]	0.66 [-0.07]	0.48
E_h	142.04 [-0.03]	142.03 [1.11]	141.03
Δ	27.79 [-2.15]	27.98 [-2.58]	30.99
Set14			
PSNR	23.77 [-0.57]	23.76 [-0.63]	20.06
SSIM	0.68 [-0.08]	0.68 [-0.09]	0.5
E_h	129.10 [-1.12]	129.31 [0.7]	130.06
Δ	30.28 [-1.88]	30.34 [-2.55]	31.24
Set5			
PSNR	26.49 [-1.03]	26.45 [-1.14]	21.48
SSIM	0.80 [-0.08]	0.80 [-0.08]	0.60
E_h	66.88 [7.6]	66.83 [8.71]	91.95
Δ	27.68 [-1.58]	28.00 [-2.18]	28.13

Results on structured datasets

The gains made by the proposed FAS criterion over GAN discriminator in perceptual quality become more pronounced, if the images have some known priors. For example, when superresolving face images, the training process can make use of prior knowledge on the structure, shape and textures of the object in question. We use the Flickr Faces HQ dataset (FFHQ) consisting of 70,000 human face images to evaluate the performances of FAS and GAN. For the $\times 4$ super-resolution task on this dataset, we have used 4 residual blocks for FANet and the generator network. The discriminator has the same architecture as SRGAN. The results are presented in Figure 3.8. As illustrated, the GAN-based results appear unnatural and suffer from severe distortions (note the reconstructed noses, mouths and teeth). On the other hand, using EDSR with the MSE loss function alone cannot reconstruct sharp details (see the areas around eyes and mouths). In contrast, FANet successfully recovers these details with clarity and largely free of objectionable artifacts.

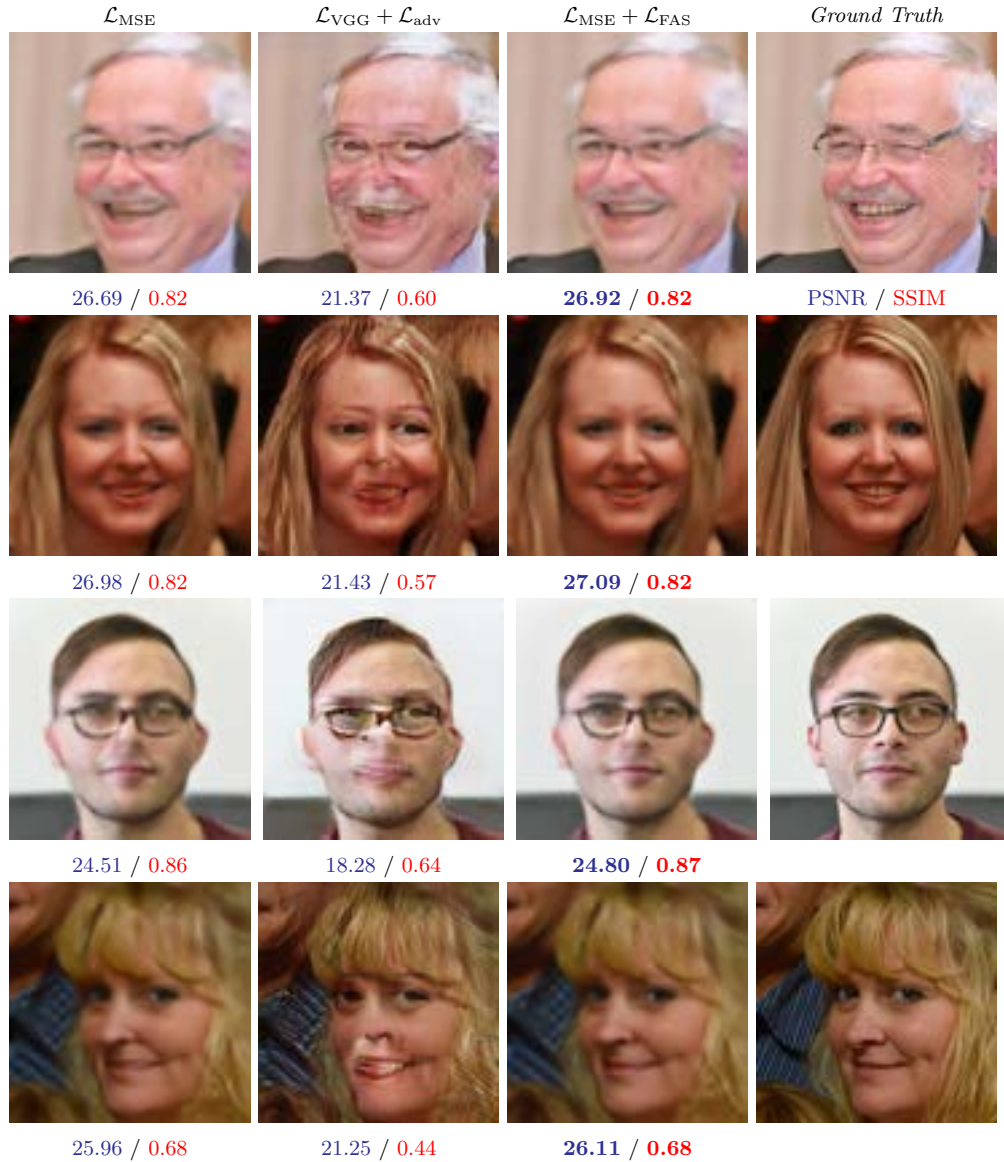


FIGURE 3.8: Results of different methods on a structured dataset (FFHQ) for $\times 4$ super resolution.

3.7.2 Denoising

Experiment Setting

Another intensively researched image restoration task is denoising. For image denoising methods, including those of deep learning, blurring artifacts are inevitable; they or lack

of them largely determine the quality of denoised images. We let our FANet method take up the challenge of preserving the sharpness and clarity of high-frequency details in the CNN denoising process. Specifically, to build the FANet for image denoising, we accentuate the EDSR model without upsampling layers (modified EDSR). The same hyperparameters in the super-resolution experiments are used to train the denoising FANet. The training data are generated by adding zero-mean Gaussian noise with variance (σ^2) 0.1 to the images. The trained denoising FANet is tested and evaluated below. These experimental results validate the effectiveness of the FAS accentuation method when being applied to CNNs for other image restoration tasks besides super-resolution.

Quantitative Results

To evaluate the efficacy and robustness of the denoising FANet, we add Gaussian noises of different variances to the validation images. This allows us to check how well the denoising FANet, which is designed for a fixed noise level, performs against different noise levels. The results are presented in Table 3.4. As shown in the table, when the CNN is integrated with FAS accentuation, all quality metrics improve for image denoising. This is consistent with our observations in the super-resolution case.

Noise level (σ^2)	α	PSNR(<i>db</i>)	SSIM	E_h	Δ (<i>db</i>)
0.01	0.0	24.27	0.66	24.40	38.40
	0.01	24.25	0.67	24.31	38.79
	0.07	24.03	0.66	24.30	38.75
	0.1	24.45	0.67	24.10	38.51
0.02	0.0	24.71	0.68	24.16	38.66
	0.01	24.64	0.68	24.08	38.90
	0.07	24.45	0.68	24.08	38.84
	0.1	24.84	0.69	23.85	38.68
0.05	0.0	26.38	0.73	23.79	38.66
	0.01	26.27	0.73	23.69	39.10
	0.07	26.08	0.72	23.74	38.95
	0.1	26.38	0.73	23.44	38.97
0.1	0.0	27.68	0.78	23.09	39.70
	0.01	27.66	0.78	23.01	39.85
	0.07	27.66	0.78	23.17	39.85
	0.1	27.68	0.78	22.85	39.89

TABLE 3.4: Denoising performance results on various noise levels for different accentuation levels α 's ($\alpha = 0$ corresponds to the MSE loss function).

Qualitative Results

We illustrate samples of the denoised images in Figure 3.9.

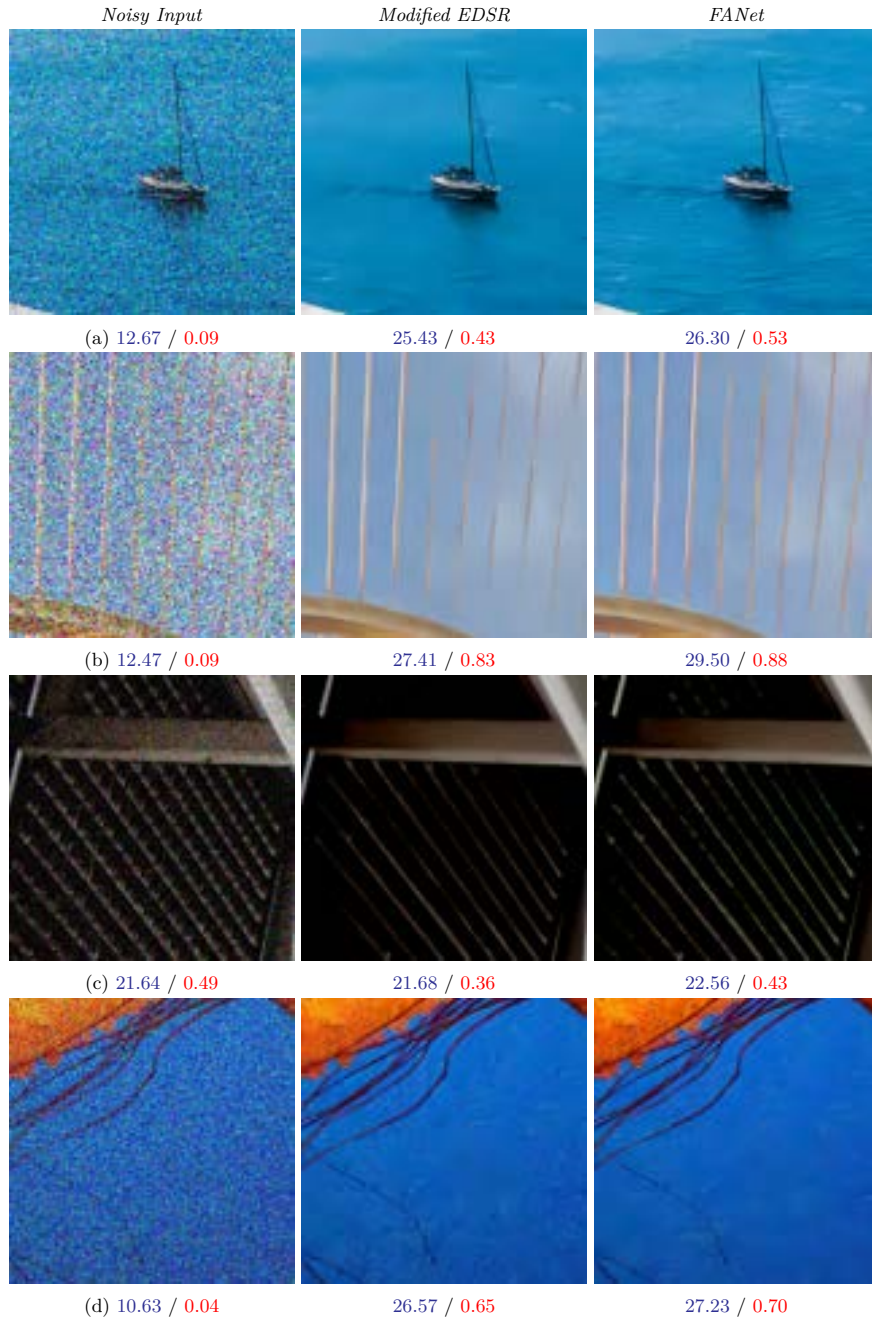


FIGURE 3.9: Samples of denoising results

As can be seen, the denoising FANet can effectively remove noises and at the same time it also keeps edges and high-frequency textures sharp and clean. In perceptual quality FANet is clearly superior to the denoising CNN without FAS accentuation (the

modified EDSR model of $\alpha = 0$). For example, in Figure 3.9 (a), the modified EDSR model fails to recover the sea wave texture and flattens the water surface, while FANet has much less over smoothing artifacts and recovers the wave structure approximately. In Figures 3.9 (b) and (c), the modified EDSR model is not able to recover the thin lines, which do not trouble FANet nearly as much. Similarly, in Figure 3.9 (d), FANet works equally well in restoring both low-frequency and high-frequency regions; the FANet recovered image is visually much more pleasant than that of the modified EDSR. Although the above comparison studies between with and without feature accentuation are carried out only on the EDSR architecture, the same conclusions should hold for other network architectures, simply because the FAS affects the optimization criterion that is independent of CNN architectures.

3.8 Conclusion

In this chapter, we propose a novel design method for image restoration CNNs to achieve sharpness and clarity of high-frequency details. The key innovation is to construct a feature accentuation space that defines desired features and sensitizes reconstruction errors in these features. The FAS construction is done by efficient optimization techniques. As opposed to GANs, which is commonly used to generate high-frequency details in recovered images, the proposed FAS method has lower computational complexity, and more importantly it does not generate nonexistent features as GANs are prone to. Experiments show that our method can improve visual quality of restored images, especially on edges and high textures. The new method is general and it can be applied to many different restoration tasks, including super-resolution, denoising, deblurring, and etc.

Preface

The following chapter is a reproduction of an Institute of Electrical and Electronics Engineers (IEEE) copyrighted, Submitted paper:

Seyed Mehdi Ayyoubzadeh, Wentao Liu, Irina Kezele, Yuanhao Yu, Xiaolin Wu, Yang Wang and Tang Jin. "Test-Time Adaptation for Optical Flow Estimation Using Motion Vectors". IEEE Transactions on Image Processing. In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of McMaster University's products or services. Internal or personal use of this material is permitted. If interested in reprinting republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to <https://www.ieee.org/publications/rights> to learn how to obtain a License from RightsLink.

Contribution Declaration: Seyed Mehdi Ayyoubzadeh (the author of this thesis) is the first author and main contributor of this article. He proposed the method, conducted experiments and composed the article. Prof. Xiaolin Wu is the supervisor of Seyed Mehdi Ayyoubzadeh. Wentao Liu, Irina Kezele, Yuanhao Yu, Yang Wang, and Tang Jin were the members of the research group at Huawei who helped with some of the ideas and polishing the manuscript.

Chapter 4

Test-Time Adaptation for Optical Flow Estimation Using Motion Vectors

4.1 Abstract

Due to the prohibitive cost as well as technical challenges in annotating ground-truth optical flow for large-scale realistic video datasets, the existing deep learning models for optical flow estimation mostly rely on synthetic data for training, which in turn may lead to significant performance degradation under test-data distribution shift in real-world environments. In this work, we propose the methodology to tackle this important problem. We design a self-supervised learning task for adjusting the optical flow estimation model at test time. We exploit the fact that most videos are stored in compressed formats, from which compact information on motion, in the form of motion vectors and residuals, can be made readily available. We formulate the self-supervised task as motion vector prediction, and link this task to optical flow estimation. To the best of our knowledge, our Test-Time Adaptation guided with Motion Vectors (TTA-MV), is the first work to perform such adaptation for optical flow. The experimental results demonstrate that TTA-MV can improve the generalization capability of various well-known deep learning methods for optical flow estimation, such as FlowNet, PWCNet, and RAFT.

4.2 Introduction

Optical flow estimation refers to the problem of estimating apparent motion velocities of brightness patterns between two images (most of the time, two consecutive frames of a video) at the pixel level, which plays a critical role in a wide range of computer vision applications, such as action recognition (Feichtenhofer et al. 2016), video denoising (Xue et al. 2019), frame interpolation (Niklaus and Liu 2018), etc. As a fundamental vision task, optical flow estimation has been heavily studied in the past several decades and tackled mainly in two ways. Traditional approaches, such as the Lucas-Kanade (Lucas and Kanade 1981) and Gunner-Farneback methods (Farneback 2003), model optical flow estimation as an optimization problem for a pair of images, generating a sparse/dense displacement map that best matches similar visual patterns between the two images. Approaches of this type rely on hand-crafted priors, which are often challenged in real-world applications, leading to mediocre performance. On the other hand, deep neural networks (DNN) have demonstrated outstanding competence in learning many pixel-level computer tasks, including super-resolution (Ledig et al. 2017b; Lim et al. 2017b), semantic segmentation (Long et al. 2015), image deblurring (Kupyn et al. 2018), image generation (Kataoka et al. 2016) and stylization (Gatys et al. 2016). Recently, DNNs have been exploited to estimate the optical flow from two consecutive frames (Ilg et al. 2017; Ranjan and Black 2017; Sun et al. 2018; Teed and Deng 2020) and achieved the state of the art performances on benchmark datasets such as KITTI 2015 (Menze and Geiger 2015), and MPI Sintel (Butler et al. 2012). One main issue for DNN-based methods is that the models trained on data from one distribution often exhibit a significant performance drop on some other distribution. This distribution shift issue is particularly relevant for DNN-based optical flow estimation models at the test time. We, humans, are good at perceiving movement but less successful in estimating the magnitude or form of the underlying motion vector field. To obtain accurate ground-truth optical flow for natural videos, 3D motion trajectories of each pixel need to be captured and projected to the camera plane, which is practically impossible for in-the-wild videos. Therefore, a common practice is to use synthetic datasets such as FlyingChairs (Dosovitskiy et al.

2015) or FlyingThings3D (Mayer et al. 2016) to train DNN-based optical flow estimators, with an expectation that those would generalize well on real-world videos. One procedure to bridge the discrepancy between the training on synthetic and testing on real-world data is to finetune the pre-trained model on real-world annotated test data (Dosovitskiy et al. 2015; Sun et al. 2018; Teed and Deng 2020). However, this approach may not be practical in the real-world deployment. First, it is difficult to obtain the ground-truth optical flow in an uncontrolled environment, even for a small dataset. Second, it is hard to verify that the small dataset for finetuning is representative of the test distribution. Third, the world we are faced with is constantly evolving, and the visual distribution drifts with time. One common approach to combat the distribution shift is the test-time adaptation (TTA) (Sun et al. 2020), a method that can adjust a model based on the test data it encounters. This approach has been shown effective in improving the robustness of DNN models to distribution shifts in some computer vision (CV) tasks, such as image classification (Sun et al. 2020; Wang et al. 2021) and image deblurring (Chi et al. 2021). In this work, we propose a novel TTA method for optical flow estimation based on compressed-domain information readily available in encoded video streams. Thanks to the similarity between neighboring frames of a video, future picture frames can be effectively predicted by motion-compensating previously coded frames with motion vector (MV) maps. By encoding the MV maps and prediction residuals only, modern video codecs, including H.264 (Wiegand et al. 2003), H.265 (Sullivan et al. 2012), etc., can compress video data at very high compression ratios.

Conceptually, these MV maps resemble the optical flow fields (Young et al. 2020) and have been found to act similarly to optical flow in many CV tasks such as action recognition (Shou et al. 2019; Wu et al. 2018) or semantic segmentation (Chen et al. 2021b; Huynh et al. 2021). However, the potential value of the MV map has been largely ignored in previous studies of optical flow estimation. In particular, we create a self-supervised learning task, i.e., motion vector prediction, to adapt the pre-trained DNN optical flow estimation model to better fit the characteristics of the input test video at the test time. Intuitively, if the model can adapt to predict the MV map drawn from the test

video distribution competently at the inference time, the fine-tuned model parameters will adapt to the new data distribution, leading to an improved optical-flow estimation task. The schematic overview of our proposed approach is illustrated in Figure 4.1.

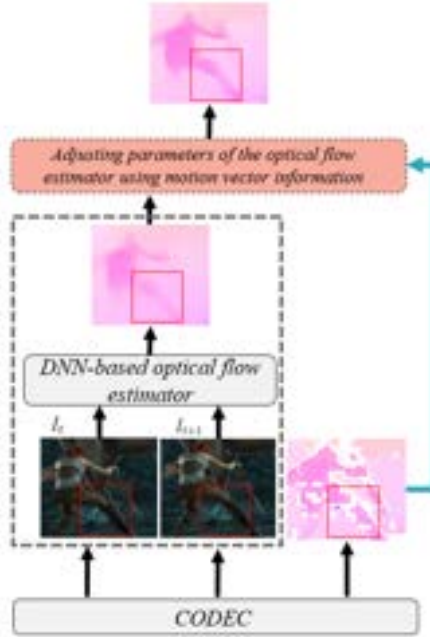


FIGURE 4.1: This figure illustrates an overview of the proposed method. Our TTV-MV framework uses the MV map extracted from the compressed video, to adjust the optical flow estimator at test time. The dashed line outlines the classic way for optical flow estimation. The red block depicts the adjustment process

As manifested, the motion vector information is exploited to adjust the parameters of the optical flow estimator to the test data distribution. However, the MV maps from video codecs cannot be directly used to supervise the flow. Since the goal of video coding is data compression, a single MV may be assigned to all pixels in a block for efficient storage, and incorrect MVs may be deliberately used as long as it improves the overall coding efficiency. Therefore, the MV map from video coding can be viewed as a rough and sparse estimation of the optical flow as exhibited in Figure 4.2. Nonetheless, the motion characteristics related to the specific test data distribution are encoded in MV maps, despite them following the distinctive point-block pattern compared to dense

optical flow.

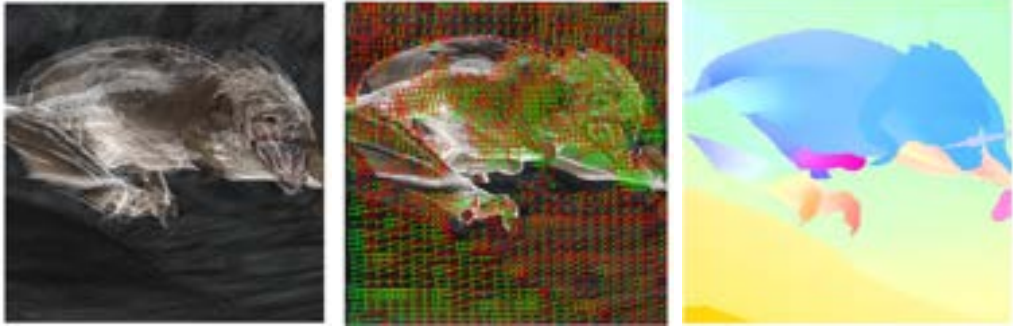


FIGURE 4.2: We illustrate the overlay of two consecutive frames (left), the corresponding MV map (middle), and the optical flow (right). The MV map provides a rough and sparse estimation of the optical flow

We argue that these data-related motion characteristics encoded in MV maps can be successfully exploited towards adjusting the parameters of the optical flow estimator, assuming we are able to model the functional relationship between the flow and MV fields (Young et al. 2020). To make use of the MV maps, an assisting CNN module dubbed Flow2MV is designed. This module is appended to the end of the optical flow estimator, and aims to transform the estimated optical flow map to its MV counterpart. The MV loss for self-supervision is defined as the error between the predicted MV map and the ground truth from the compressed video. Since both the optical flow estimator and the Flow2MV module are end-to-end trainable, the weights of the optical flow estimator can be effectively adjusted by the gradient flow from the MV prediction task. Occasionally, the motion vector map can be very sparse compared to the optical flow, meaning that we could only have supervision for a tiny portion of pixels. For those reasons, in addition to the MV loss, we propose to use another auxiliary loss, *e.g.* the photometric loss (Liu et al. 2019b), to further stabilize the process of test-time adaptation. This auxiliary loss is also beneficial for gradient alignment of the MV loss. The contributions of this chapter are as follows:

- Based on the MV prediction task, we propose the first test time adaptation framework, dubbed as *TTA-MV*, to combat the distribution shift issue in optical flow

estimation. To the best of our knowledge, this is also the first work that exploits compressed-domain information in the test-time adaptation setting.

- The proposed framework is not restricted to specific network architecture, and can be applied to any DNN-based optical flow estimation models, such as FlowNet (Ilg et al. 2017), PWCNet (Sun et al. 2018), and RAFT (Teed and Deng 2020).

Experimental results show that our method can consistently improve the performance of several popular optical flow estimators under the distribution shift.

4.3 Related Work

4.3.1 Optical flow estimation

Traditional methods often treat optical flow estimation in a variational framework and solve an energy minimization problem to encourage brightness pattern alignment and extra flow field regularities. Since the seminal work of Lucas *et al.* (Lucas and Kanade 1981), this computational framework has achieved remarkable success and is further strengthened by techniques like coarse-to-fine refinement and descriptor matching in a series of follow-up works (Brox et al. 2004; Brox and Malik 2010; Revaud et al. 2015; Sun et al. 2014). Nevertheless, these methods are typically designed for short-range motion estimation. They are vulnerable to incomplete correspondences, making them unsuitable for real-world examples with large motions and naturally occurring occlusions.

Another line of optical flow estimation research resorts to data-driven approaches and has flourished with the development of deep learning techniques. In the ground-breaking work Dosovitskiy et al. 2015, Dosovitskiy *et al.* proposed the first DNN-based models, known as FlowNet, that can directly predict a dense optical flow map from a pair of images in a feed-forward manner. Without complex variational optimization steps, the FlowNet models achieve on-par or even superior performance than many traditional optical flow estimation methods. The promising result of FlowNet thus triggered a huge wave of research in the field of DNN-based optical flow estimation (Ilg et al. 2017;

Ranjan and Black 2017; Sun et al. 2018; Yang and Ramanan 2019; Zhao et al. 2020), and some recent networks, such as RAFT (Teed and Deng 2020), are able to outperform the best traditional method (Xu et al. 2017) by a sizeable margin on standard benchmark datasets (Butler et al. 2012; Menze and Geiger 2015). Despite their different network architectures, all these models are trained with supervised learning and follow a similar learning schedule due to the lack of sufficient realistic training data. Specifically, the model is first trained on a large-scale synthetic dataset before being finetuned to a target small-scale dataset of realistic videos. This experimental setting may not be practical in real-world deployment as it can be extremely difficult to obtain ground-truth optical flow even for a small number of natural videos. Without finetuning on the target distribution, these supervised-learning methods are susceptible to the distribution shift issue in the test phase.

Closely related to our approach is the unsupervised-learning method for optical flow estimation, such as SelFlow (Liu et al. 2019b). This method does not require annotations of optical flow for training but still assumes that training videos are sampled from the same distribution of test data. However, this assumption may not hold as test data distribution often evolves with time. In contrast, our method only utilizes the information from the test data to adjust the model and is more practical in real applications. Moreover, SelFlow employs unsupervised learning in the training process, while our method adopts self-supervised learning only at test time. Lastly, note that supervised training of optical flow estimators, even with the synthetic data, leads to better performance than training optical flow estimators in a fully unsupervised manner (Liu et al. 2019b; Teed and Deng 2020).

4.3.2 Computer vision with compressed videos

Videos are normally compressed by video codecs for efficient storage and thus represented by compressed domain information rather than raw pixels in standard video files. More compact, in terms of information density and more representative for temporal variations, the compressed domain information, such as motion vectors and/or the residual

maps, find themselves useful in a variety of video CV tasks. For instance, many studies (Wu et al. 2018; Shou et al. 2019; Hu et al. 2021; Zhang et al. 2016; Cao et al. 2019) find that compressed domain information may stand in for the costly optical flow in action recognition tasks and can accelerate two-stream models by several hundred folds. Similar acceleration potential of compressed domain information is also observed in semantic segmentation (Feng et al. 2020), human pose estimation (Fan et al. 2021), and video super-resolution (Chen et al. 2020). Besides, compressed domain information is also employed in designing self-supervised pretext tasks for video representation learning (Huang et al. 2021; Yu et al. 2020b), while their target downstream tasks are action recognition or video retrieval. Motion vector maps contain rich motion information of a video and are readily available when decoding a video, before applying a CV model to the decoded picture frames. The authors of Young et al. 2020 prove that motion vector maps may be regarded as a blocky version of optical flow and propose a fast method for optical flow estimation by filtering the motion vector map in a traditional framework (Revaud et al. 2015). Our proposed method substantially differs from the approach in Young et al. 2020, in our work we use the motion vectors to improve the robustness of data-driven optical flow estimation models in an unknown test distribution and in the context of DNN-based models.

4.3.3 Test-time adaption

Test-time adaptation (TTA), or test-time training, is a technique that adjusts a model based only on the unlabeled test sample presented at test time in order to improve the out-of-distribution performance of the model. A key step of TTA is designing a meaningful self-supervised learning task based on the test sample. Shocher et al. 2018 propose to learn a sample-specialized image super-resolution model by enforcing the model to upsample a down-scaled version of the test image back to the original. Similarly, Chi et al. 2021 propose an auxiliary task of reconstructing the blurry input image from deep features to adapt the model towards each test sample. Sun et al. 2020 improve the robustness of image classification models by predicting the rotation angle of test images, while

Wang et al. 2021 approach a similar goal via directly minimizing prediction entropy. In this chapter, we propose the first TTA framework for optical flow estimation models and design the self-supervised learning task using representations from compressed videos.

4.4 Proposed Framework

In this section, we present our proposed test time adaptation framework that aims at improving the performance of pretrained optical flow estimators on out-of-distribution test samples.

4.4.1 Overview of the proposed framework

An overview of our proposed TTA-MV framework for optical flow estimation is shown in Fig. 4.1. The framework is built upon the normal optical flow inference pipeline for compressed videos, as illustrated in the dashed black box at the left bottom corner of Fig. 4.1. Specifically, the normal inference pipeline first decodes two consecutive frames, denoted by I_t and I_{t+1} , from the compressed video and then passes them to a pretrained DNN-based optical flow estimator to obtain the optical flow map between the two frames. In contrast, the proposed TTA-MV approach further decodes the motion vector map M between the two frames from the compressed video and uses it to adjust the parameters of the optical flow estimator at test time, with which an improved optical flow prediction may be achieved. Fig. 4.3 depicts the detailed pipeline of the proposed TTA-MV approach.

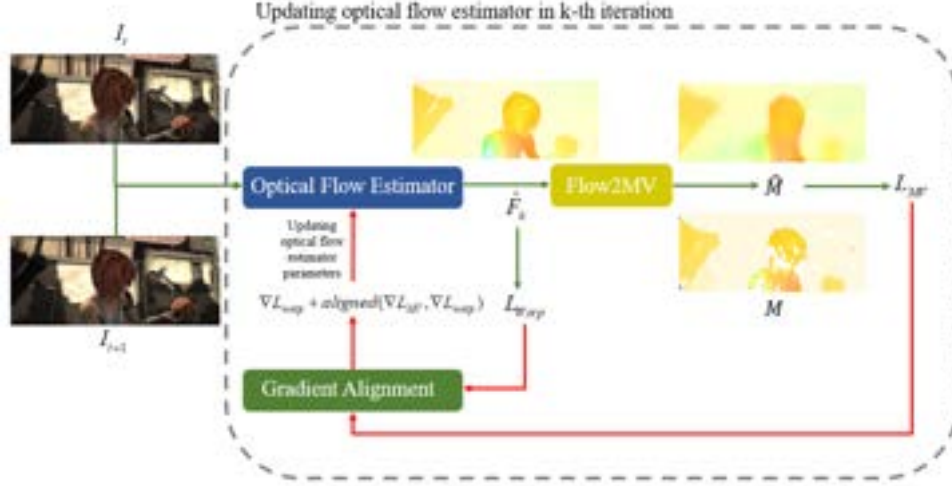


FIGURE 4.3: This figure illustrates a schematic block diagram of TTA-MV. The Flow2MV module links the optical flow prediction to the MV map. An MV loss is thus defined as the error of MV prediction. In addition, a warping loss is defined based on the optical flow prediction. We perform gradient alignment to combine gradients from the two losses into one, which is then used to update the parameters of the optical flow estimator. Green arrows represent feed-forward inference, while red arrows indicate gradient back-propagation

The decoded image pair is fed into a DNN-based Optical Flow Estimator, denoted by $f(I_t, I_{t+1}; \theta)$, to generate an optical flow prediction \hat{F} , where θ denotes the network weights of the optical flow estimator collectively.

Next, the predicted optical flow map is transformed to a motion vector prediction \hat{M} by the proposed Flow2MV module $g(F; \phi)$ with ϕ denoting its weights collectively.

The predicted motion vector map is then compared with the previously decoded, ground-truth motion vector map M , and a motion vector prediction loss is calculated. We implement the Flow2MV module as a neural network, so the gradient of the motion vector loss with respect to θ can be obtained by back-propagating the gradient flow through the Flow2MV module. In addition, we design a warping loss based on the input image pair and the predicted optical flow, whose gradient with respect to the weights θ of the optical flow estimator is also calculated. The gradients from both losses are then combined to update the optical flow model, where a gradient alignment technique (Yu

et al. 2020a) is adopted to solve potential conflicts between the two descent directions in some cases.

4.4.2 Flow2MV module

In the following, we introduce the architecture of the Flow2MV module and how we train it for test time adaptation.

Network architecture. The Flow2MV module takes an optical flow map $F \in \mathbb{R}^{H \times W \times 2}$ as its input and produces a corresponding motion vector map $\widehat{M} \in \mathbb{R}^{H \times W \times 2}$, where $H \times W$ is the spatial resolution. Since the motion vector map may be regarded as a degraded version of the optical flow map, with a comparably lower and spatially variable resolution related to the underlying scene content (Young et al. 2020), we let the lightweight Flow2MV module learn this spatially and scene dependent degradation process. Therefore, we adopt a lightweight UNet-like (Ronneberger et al. 2015a) architecture with an explicit bottleneck to mimic the information loss. The detailed network architecture is shown in Figure 4.4.

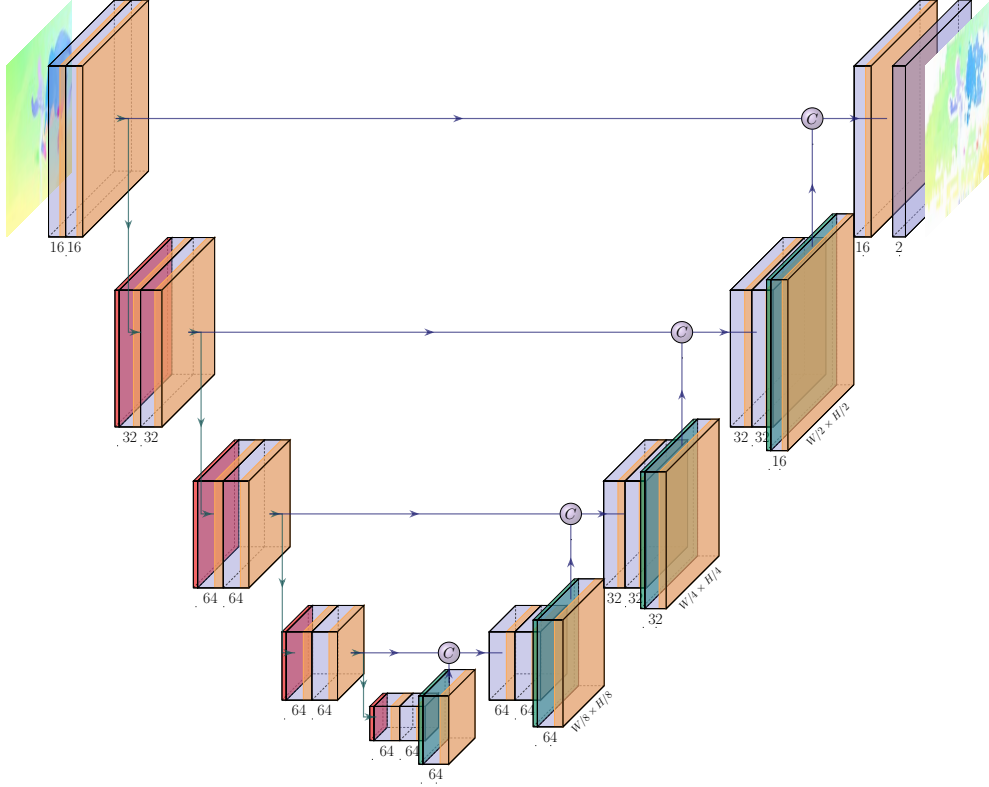


FIGURE 4.4: **Flow2MV architecture.** Flow2MV module is a simplified version of the U-Net (Ronneberger et al. 2015b) architecture. All convolutional layers except the last one have the filter size of 3×3 and followed by ReLU activation function. The last convolution layer filters are 1×1 .

Motion vector loss. We compare the predicted motion vector map \widehat{M} with the ground-truth map M from the decoder by a masked ℓ_1 loss:

$$\mathcal{L}_{MV} = \left\| B \odot (\widehat{M} - M) \right\|_1, \quad (4.1)$$

Where B is a binary mask defined as below, and \odot represents element-wise multiplication. Since video codecs may encode a pixel in either inter- or intra-mode (Wiegand et al. 2003; Sullivan et al. 2012; Mukherjee et al. 2013), not all pixels have associated motion vectors even in a P- or B-frame. Therefore, the entries of M are actually undefined for such pixels. By assigning zero weights in the mask B to these positions and unit weights to others, we avoid the bias that may be introduced by any preset default values in M .

Two-stage training. The Flow2MV module is trained in a two-stage manner before being used for test-time adaptation. We first use ground-truth optical flow F as the input and pre-train the module with the motion vector loss $\mathcal{L}_{MV}(F, M; \phi)$. In the second stage, we initialize the module with pre-trained weights and attach it to a baseline optical flow estimator. Assuming that the baseline optical flow estimator was trained with a loss function $\mathcal{L}_{flow}(I_t, I_{t+1}, F; \theta)$, we jointly finetune both modules with a combined loss:

$$\mathcal{L}_{total} = \alpha \mathcal{L}_{flow}(I_t, I_{t+1}, F; \theta) + \mathcal{L}_{MV}(f(I_t, I_{t+1}; \theta), M; \phi), \quad (4.2)$$

With α being the trade-off weight between the two losses. The joint training process serves two purposes. First, it encourages the Flow2MV module to better predict motion vector maps from the optical flows generated by the specific optical flow estimator. Second, it mitigates the potential risk of the contrapositive of the self-defeating effect (Wu et al. 2021): "optimizing the downstream task loss may not necessarily improve the performance of the upstream task", by adapting the Flow2MV module to the actual output distribution of the baseline optical flow estimator.

4.4.3 Test-time adaptation

With the help of the pre-trained Flow2MV module, we are now able to make use of the motion vector information from compressed videos to perform test-time adaptation for optical flow estimation. A naïve way of doing this is directly updating the weights θ of the optical flow estimator based on the motion vector loss:

$$\theta_{k+1} \leftarrow \theta_k - \gamma \nabla_{\theta_k} \mathcal{L}_{MV}(f(I_t, I_{t+1}; \theta_k), M), \quad (4.3)$$

Where γ is the step size of gradient descent and k indicates the iteration index during test-time adaptation. Note that the parameters ϕ are omitted from \mathcal{L}_{MV} in Eq. (4.3) since we freeze the Flow2MV module in the inference phase. However, for a minor portion of data, we have observed the “catastrophic forgetting” phenomenon (McCloskey and Cohen 1989), where our self-supervision may degrade the main task performance, as

modern video encoders may occasionally generate sparse MV data to achieve a high compression ratio. To hand those marginal cases, we propose to regularize the gradient descent direction with a *warping loss*. Given the test image pair I_t and I_{t+1} and the optical flow prediction $\hat{F}_k = f(I_t, I_{t+1}; \theta_k)$, the warping loss is defined as

$$\mathcal{L}_{warp} = \left\| \mathcal{T}(I_t) - \mathcal{W}(\mathcal{T}(I_{t+1}), \hat{F}_k) \right\|_1, \quad (4.4)$$

Where \mathcal{T} and \mathcal{W} represent an arbitrary image transformation and the backward warping operation. In our experiments, we have tried both identity mapping and the feature extractor of the RAFT encoder as \mathcal{T} . Note that, in the case of identity transformation, the warping loss is equal to the photometric loss (Liu et al. 2019b). The gradient of the warping loss is thus obtained by $\nabla_{\theta_k} \mathcal{L}_{warp}(I_t, I_{t+1}; \theta_k)$. However, simply combining the two gradients may not always be constructive for optical flow estimation as they may be conflicting with each other (Yu et al. 2020a). Therefore, we perform gradient alignment following the method in (Yu et al. 2020a) to obtain

$$\overline{\nabla_{\theta_k} \mathcal{L}_{MV}} = \begin{cases} \nabla_{\theta_k} \mathcal{L}_{MV} & \text{if } \nabla_{\theta_k} \mathcal{L}_{MV} \cdot \nabla_{\theta_k} \mathcal{L}_{warp} \geq 0 \\ \nabla_{\theta_k} \mathcal{L}_{MV} - \frac{\nabla_{\theta_k} \mathcal{L}_{MV} \cdot \nabla_{\theta_k} \mathcal{L}_{warp}}{\|\nabla_{\theta_k} \mathcal{L}_{warp}\|_2^2} \nabla_{\theta_k} \mathcal{L}_{warp} & \text{otherwise} \end{cases} \quad (4.5)$$

and update the parameters of the optical flow estimator by

$$\theta_{k+1} \leftarrow \theta_k - \gamma \left(\nabla_{\theta_k} \mathcal{L}_{warp} + \beta \overline{\nabla_{\theta_k} \mathcal{L}_{MV}} \right), \quad (4.6)$$

Where β is a weighting hyper-parameter. We repeat this process to update the optical flow estimator parameters K times before performing optical flow estimation.

4.5 Experiments

4.5.1 Evaluation datasets and optical flow estimators

In order to simulate the distribution shift between training and test phases, we deliberately design the experiment protocol so that the training and test data are sampled from significantly different distributions. Specifically, we train baseline optical flow estimators on the FlyingChairs (Dosovitskiy et al. 2015) dataset, and compare the performances of the baseline estimators with and without the MV-TTA on three other widely-used benchmark datasets, i.e. MPI Sintel (Butler et al. 2012) (both final and clean passes), KITTI 2012 (Geiger et al. 2012), and KITTI 2015 (Menze and Geiger 2015). Specifications of the four employed datasets are summarized in Table 4.1, from which we can see that the data distribution of the FlyingChairs dataset is completely different from those of the other three. In addition, the average endpoint error (AEPE) is used to quantify the performance of each model on each dataset.

TABLE 4.1: Specifications of employed optical flow estimation benchmark datasets

Name	FlyingChairs	SintelClean	SintelFinal	KITTI2012	KITTI2015
Split	Train	Test	Test	Test	Test
Type	Synthesized	Animation	Animation	Realistic	Realistic
Sample #	22872	1041	1041	194	200

4.5.2 Implementation details

We conduct all the experiments with Tensorflow 2.3 on an NVIDIA TITAN X GPU.

Data preprocessing. We train the Flow2MV module on the FlyingChairs (Dosovitskiy et al. 2015) dataset, and the trained module is then used for adapting the baseline model at the test time. Both training the Flow2MV module and test-time adaptation require motion vector maps as supervision signals. Therefore, we need to generate motion vector maps for the four involved datasets, FlyingChairs, MPI Sintel, KITTI 2012,

and KITTI 2015. Training samples of FlyingChairs, KITTI2012, and KITTI 2015 are given as independent image pairs, denoted by I_t and I_{t+1} .

We then encode each image pair into a 2-frame video and then extract the motion vector map between them by decoding the encoded video stream. For MPI Sintel, we encode a sequence of images belonging to the same clip (*e.g.* alley_1) into a single video and extract motion vectors for all frames except the first one. Since the encoded videos in MPI Sintel contain more than two frames, the extracted motion vectors of the same frame may refer to different reference frames, a feature supported by many modern video codecs. Therefore, an extra normalization step is taken to re-scale motion vectors for the MPI Sintel dataset as if the motion vectors still refer to the immediately previous frame as in the other three datasets.

Another issue is that the extracted motion vector map after re-scaling always points from I_{t+1} to I_t , while the provided optical flow map points to the opposite direction. We hence propose to reverse the direction of motion vector maps before using them for training and test-time adaptation. Denote the raw motion vector map by $M_{t+1 \rightarrow t}^r(\mathbf{x})$, $\mathbf{x} \in V$, where V represents the set of pixels in the second image that have associated motion vectors. We thus obtain the reversed motion vector map $M_{t \rightarrow t+1}$ and the corresponding binary mask B by

$$M(\mathbf{x}) = \begin{cases} -M^r(\mathbf{x}') & \text{if } \exists \mathbf{x}' \in V, \text{ s.t. } \mathbf{x} = \lfloor \mathbf{x}' + M^r(\mathbf{x}') \rfloor \\ \text{none} & \text{otherwise} \end{cases} \quad (4.7)$$

$$B(\mathbf{x}) = \begin{cases} 1 & M(\mathbf{x}) \text{ exists} \\ 0 & \text{otherwise} \end{cases} \quad (4.8)$$

where we drop the subscripts of $M_{t \rightarrow t+1}$ and $M_{t+1 \rightarrow t}^r$ for simplicity, and where $\lfloor \cdot \rfloor$ is an operator rounding every entry of the input vector to the nearest integer. The reversed motion vector map $M_{t \rightarrow t+1}$ and associated binary mask are then used to calculate the

motion vector loss in (4.1) for both training the Flow2MV module and performing the test-time adaptation for baseline optical flow estimation models.

In the experiments, we choose to encode videos with H.264 (Wiegand et al. 2003), since it is by far the most popular video codec in the world. However, the data pre-processing pipeline applies to most video codecs, such as H.264 (Wiegand et al. 2003), HEVC (Sullivan et al. 2012), MPEG2 (Tudor 1995), VP9 (Mukherjee et al. 2013), etc.

Training procedure. We first pre-train the Flow2MV module using the ground-truth optical flow map as input and the processed motion vector map as supervision on the FlyingChairs dataset with Adam optimizer (Kingma and Ba 2014a). The loss function used for pre-training is as defined in (4.1). We set the mini-batch size to 16 and the learning rate to 10^{-4} for the whole pre-training process. In the end, the total pre-training process takes $600k$ iterations to converge. The pre-trained Flow2MV module is then attached to a baseline optical flow estimator, and the two parts are finetuned together in an end-to-end manner. Note that the finetuning is also performed on the FlyingChairs dataset, so both the Flow2MV module and the optical flow estimator are still blind to the test distribution. Adam optimizer (Kingma and Ba 2014a) is used with the learning rate of 10^{-5} and batch size is set to 8 for finetuning the modules. The finetuning process ends after 15000 iterations. We continue to use the motion vector loss for finetuning, but regularize it with the loss function that was used to train each baseline optical flow estimator in the original papers (Ilg et al. 2017; Sun et al. 2018; Teed and Deng 2020), resulting a total loss as shown in (4.2).

Test-time adaptation. The weights of the Flow2MV module are fixed at the inference time. Adam optimizer is used to perform test-time adaptation on the optical flow estimator with a small learning rate of 5×10^{-6} .

The hyper-parameter β in Eq. (4.6) is determined experimentally as shown in next section.

4.5.3 Ablation study

To determine the contribution of each component in improving the generalization of the optical flow estimator, we have conducted several experiments on the Sintel Final dataset with FlowNet (Ilg et al. 2017) being the baseline model. **Training strategy.** In this experiment, we show that the two-stage training strategy is crucial for the Flow2MV module to be able to guide TTA of the optical flow estimator effectively. We compare three different training strategies, including the two-stage training strategy we currently take. First, we drop the first stage, and train both modules jointly from scratch. We find it difficult for the two modules to converge within a reasonable period of time, indicating the necessity of the pre-training stage. Then we pre-train both the FlowNet and the Flow2MV module, and evaluate the performances of MV-TTA before and after the second-stage finetuning, respectively. We observe that, adding the second-stage finetuning reduces the AEPE of MV-TTA from 5.51 to 5.32, accounting for 3.44% relative improvement.

TABLE 4.2: Ablation study on α (left) and β (right). Performances are evaluated on the Sintel Final dataset with the FlowNet

α	0.1	1	5	10	20	∞
AEPE	5.35	5.37	5.32	5.32	5.36	5.41
β	0	0.01	0.1	1	10	100
AEPE	5.62	5.32	5.31	5.32	5.32	5.32

Ablation study on α . The purpose of the second-stage finetuning is to couple the Flow2MV with the optical flow estimator to be adapted. However, without carefully balancing the importance of the regularization term, the fine-tuning may be pulled away more strongly from the main task, leading to an impaired performance. In order to investigate the influence of the hyper-parameter α , we evaluate the AEPE performance of TTA-MV with different α values on the Sintel Final dataset as listed in Table 4.2 (left). The column $\alpha = \infty$ corresponds to the case where the optical flow estimator is kept frozen during finetuning. From Table 4.2 (left), we can see that the AEPE changes

with α gracefully and reaches the minimum when α is set to 5 or 10. We then set $\alpha = 5$ for the rest of the experiments unless otherwise stated. Note that no matter which value α is set to, a smaller AEPE is achieved than that without finetuning (5.51), reconfirming the necessity of the two-stage training strategy.

Ablation study on β . We also evaluate the performance of MV-TTA with different β values, and summarize the results in Table 4.2 (right). From the results, we can see that including the gradient from the MV loss for TTA significantly improves the performance regardless the exact value of β . A possible reason for this phenomenon is that erroneous optical flow prediction may lead to zero warping loss, but should lead to inaccurate MV prediction. Therefore, combining the two losses, which turn out to be complementary to each other, leads to more stable TTA results. While different β values scale the gradients differently, we adjust the number of iterations K until the process of TTA converges, so similar AEPE performances are achieved across a relatively wide value range of β . Nevertheless, we choose $\beta = 0.1$ (where $K = 10$) as it achieves a slightly better performance.

TABLE 4.3: We report the reduction of AEPE (\downarrow) and the relative improvement compared to the baseline (\uparrow) under different settings of the proposed TTA-MV framework on the Sintel Final dataset with FlowNet

Exp ID	\mathcal{L}_{MV}	\mathcal{L}_{warp}	Grad. Alignment	AEPE	Rel. improvement
0	-	-	-	5.68	0%
1	✗	✓	-	5.62	1.06%
2	✓	✗	-	5.49	3.34%
3	✓	✓	✗	5.36	5.63%
4	✓	✓	✓	5.31	6.51%

Contributions of each component in MV-TTA. In this experiment, we investigate the contributions of each individual component in the proposed TTA-MV framework as shown in Table 4.3, from which we have several observations. First, compared to the baseline without any adaptation (Exp 0), both the warping loss (Exp 1) and the MV loss

(Exp 2) can help adapt the optical flow estimator to the test distribution. Second, using the MV loss (Exp 2) alone for TTA yields a better result than the warping loss (Exp 1), implying that the MVs from the compressed domain encode rich motion characteristics which are necessary for adapting the optical flow estimator to the test video. Third, we observe that simply combining both losses without gradient alignment (Exp 3) provides better results than using only one loss (Exp 1 and 2), indicating the complementary properties of the two losses as we discussed in the previous paragraph. Last, with the gradient alignment module (Exp 4), the prediction AEPE is further reduced to 5.31, translating to 6.51% improvement relative to the baseline (Exp 0).

Examples of MV validity mask ($B(\mathbf{x})$). The validation mask for the motion vector (MV) map can be easily determined from the information provided by the encoder. The mask value is 1 for the places where the motion vector exists and 0 otherwise. Some examples of the motion vector maps and their corresponding masks from the Sintel Final dataset are shown in Figure 4.5.

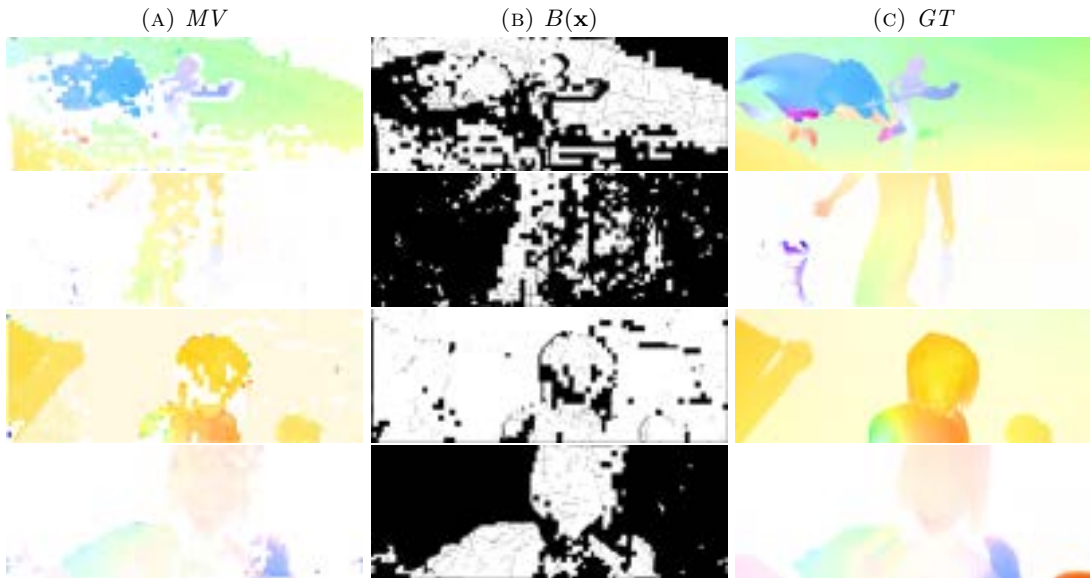


FIGURE 4.5: **Visualization of MV maps, MV validity masks and optical flow maps.** We show (a) the MV maps, (b) their corresponding validity masks, and (c) the optical flows of four samples from the Sintel Final dataset. Each row represents one sample. Despite the apparent similarity between the MV and the optical flow maps, motion vectors may exist only on a portion of pixels in the whole image field, indicating the necessity of including the MV validity mask in the calculation of the MV loss.

Optical flow prediction vs. number of iterations (K) To understand how the iterative update of the optical flow module improves the optical flow prediction. We have visualized the refinement of the optical flow versus the number of update iterations (K) in Figure 4.6, for some instances of the iteration number K , on an example from the Sintel Final dataset, adapting the baseline FlowNet model.

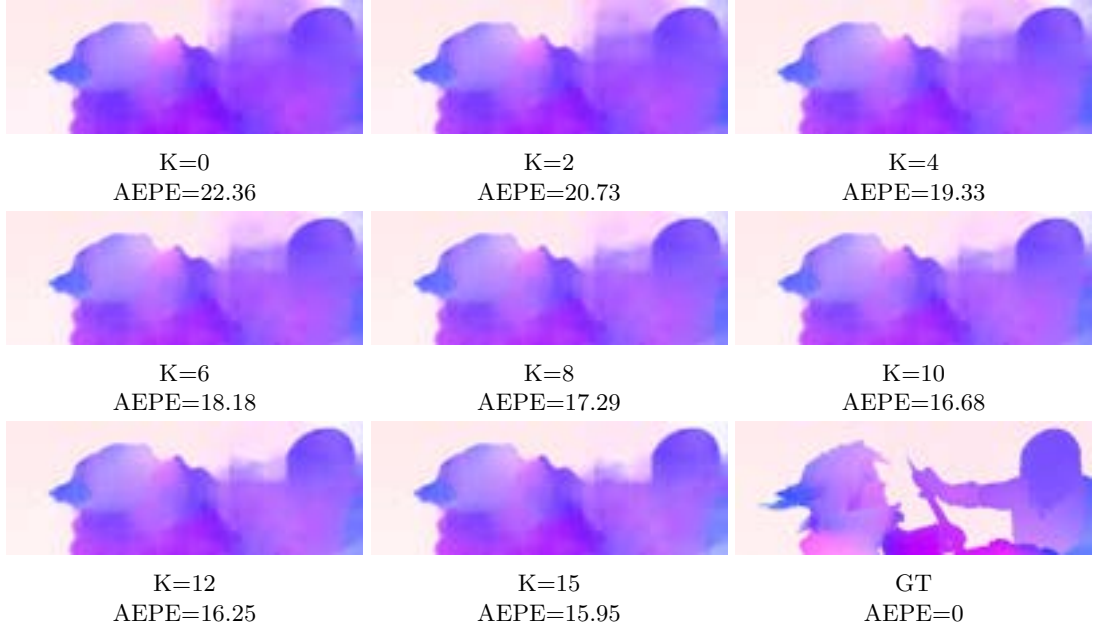


FIGURE 4.6: **Optical flow prediction vs K .** We show the adaptation process over iterations K . For each iteration, we indicate the current average end point error (AEPE). The bottom right panel depicts the ground truth optical flow.

Performance vs. Constant Rate Factor (CRF) The available videos for optical flow prediction are stored with various compression ratios. To determine how TTA-MV can perform on the compressed videos compared to the baseline approach, we have evaluated the videos in the Sintel Final dataset compressed with different Constant Rate Factors (CRF). With a higher CRF, the video encoder can encode a source video at a higher compression ratio, resulting in worse visual quality of the encoded video. We can see how different CRFs affect the frame quality in Figure 4.7, first column. Note that the CRF impacts not only the quality of decoded frames but also that of the motion vector maps, as shown in Figure 4.7, second column. We have visualized the output of the FlowNet and TTA-MV FlowNet in the last two columns of Figure 4.7. As demonstrated, when the FlowNet is equipped with the TTA-MV framework, the optical flow prediction is more robust for all tested CRFs.

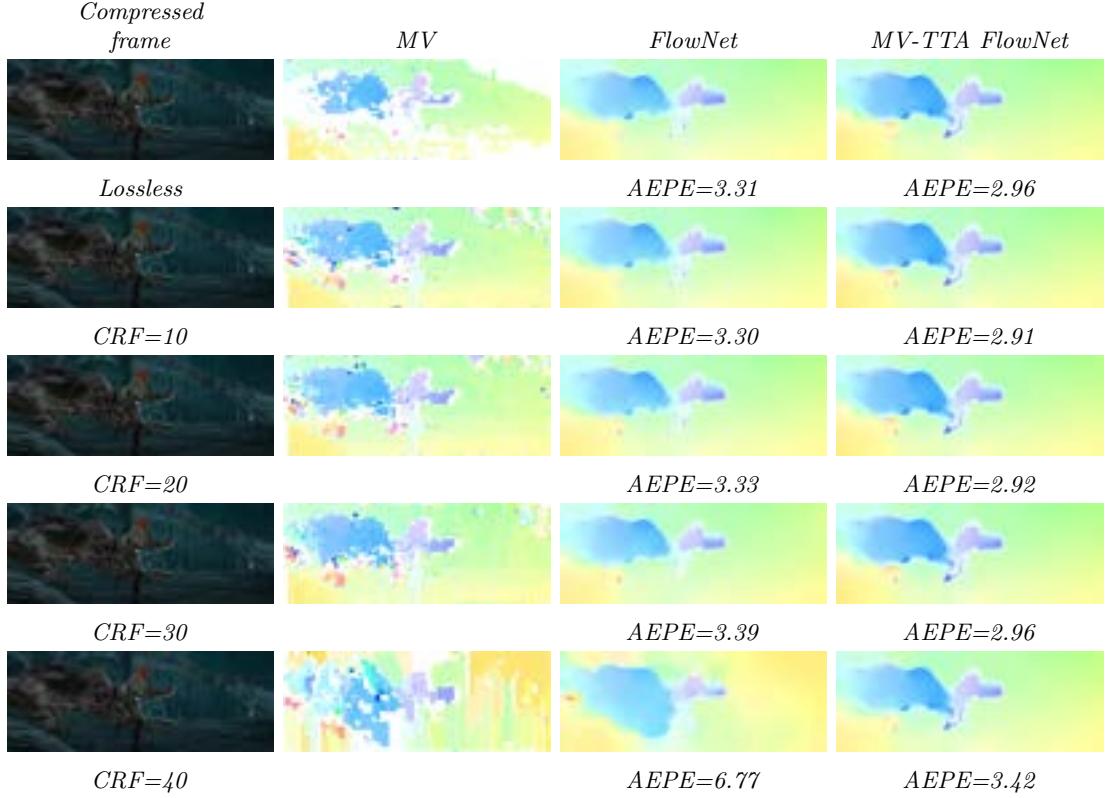


FIGURE 4.7: **TTA-MV improvement in optical flow estimation over different CRFs.** We can see the optical flow prediction is better for the TTA-MV FlowNet compared with the FlowNet for all the CRF levels. The improvement is particularly significant for higher CRF levels. Note that the result for CRF=10 is better than CRF=0. The reason lies in the provided information by the motion vector map. The motion vector map for CRF=10 is richer compared to CRF=0. When CRF=0, most of the pixels are encoded in intra-prediction mode. Thus the motion vector map has less information.

4.5.4 Main results

We evaluate our TTA-MV framework with three popular DNN-based optical flow estimators, *i.e.* FlowNet (Ilg et al. 2017), PWCNet (Sun et al. 2018), and RAFT (Teed and Deng 2020) on four standard benchmark datasets. We compare the prediction accuracy in terms of AEPE with or without TTA-MV for all the four datasets in Table 4.4, from which we have two observations. First, the performance is improved for all three baseline

models with TTA-MV, indicating that our proposed TTA-MV framework is compatible with any DNN-based optical flow estimator. Second, the estimator performance is also consistently improved for all four datasets, showing that the proposed TTA-MV framework may generalize well to any unknown test distribution.

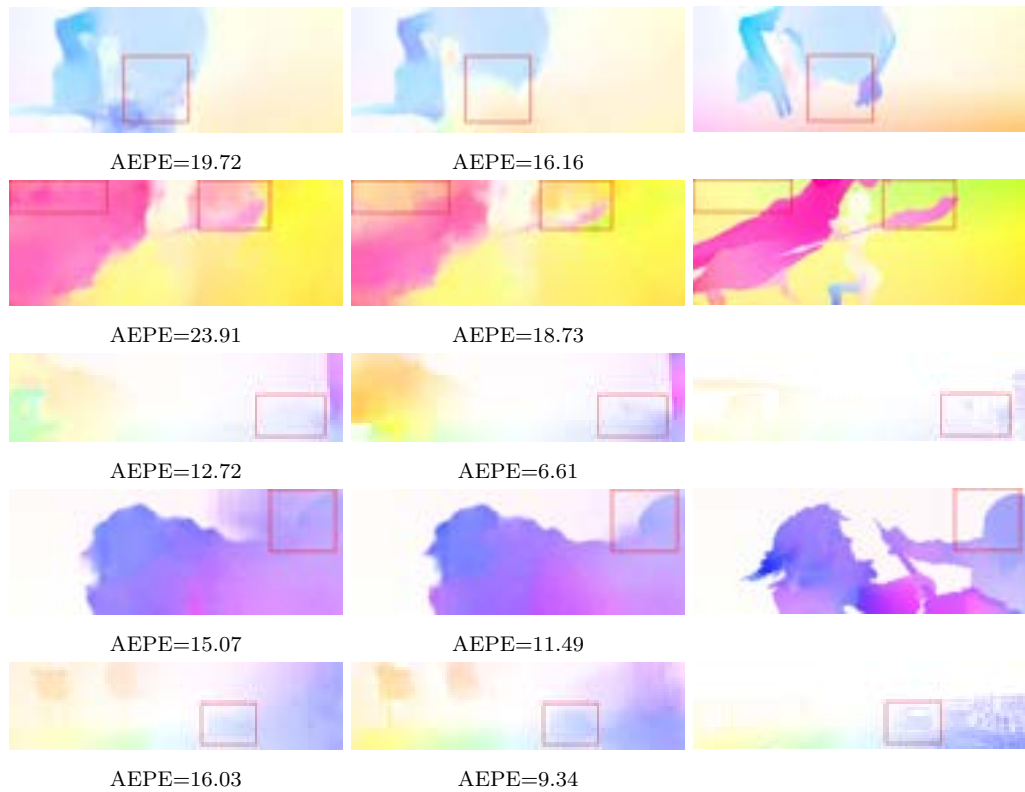


FIGURE 4.8: Visual comparison between the results from the same optical flow estimator with and without the TTA-MV framework (first column: result without TTA-MV, second column: result with TTA-MV, third column: ground truth)

TABLE 4.4: AEPE performances of three baseline optical flow estimators with or without TTA-MV on benchmark datasets. The relative improvement for each case is measured in percentages and shown in green

Method	Sintel	Sintel	KITTI	KITTI
	Clean	Final	2012	2015
FlowNet	5.29	5.68	9.51	14.82
with TTA-MV	4.64	5.31	6.79	12.48
	12.28%	6.51%	28.60%	15.78%
PWCNet	2.67	3.89	5.63	10.05
with TTA-MV	2.63	3.86	5.44	9.94
	1.49%	0.77%	3.37%	1.09%
RAFT	2.55	4.80	4.52	9.81
with TTA-MV	2.40	4.29	4.32	9.27
	5.88%	10.62%	4.42%	5.50%

In order to gain an intuitive impression of MV-TTA, we visually compare the optical flow predictions with or without the proposed framework in Figure 4.8, from which we can see that TTA-MV helps correct large prediction errors as highlighted by the red boxes.

4.6 Conclusion

In this Chapter, we have presented a novel test-time adaption framework to combat the test distribution shift issue of DNN-based optical flow estimation models. Leveraging the fact that most videos are stored in compressed format, we utilize the motion vector map as a hint to adjust the pre-trained optical flow estimator towards the direction that better fits the test data. Albeit at a lower resolution compared to the dense optical flow, the motion vector map encodes the motion characteristics, specific to the given test data distribution, which we capitalize on. Specifically, we propose a lightweight Flow2MV module to link the two motion representations, optical flow and motion vector. Through joint training with the baseline optical flow estimation model, the Flow2MV captures the data-specific motion characteristics of the predicted optical flow map and effectively

extracts related motion information to reconstruct a motion vector counterpart. At the test time, the Flow2MV is fixed, and adaptation is performed on the optical flow estimator only. Through the self-supervised task of motion vector map prediction, the optical flow estimator is encouraged to extract relevant motion features from the test data. The gradient alignment operation further facilitates the convergence of the test-time adaptation process and improves the performance of the optical flow estimator. Experiments on standard benchmark datasets show the effectiveness of the proposed framework.

Preface

The following chapter is a reproduction of an Institute of Electrical and Electronics Engineers (IEEE) copyrighted, Submitted paper:

Seyed Mehdi Ayyoubzadeh, Xiaolin Wu and Xi Zhang. "Asymmetric Coding for Ultra-high Throughput Encoding (ACUTE)". IEEE Transactions on Image Processing.

In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of McMaster University's products or services. Internal or personal use of this material is permitted. If interested in reprinting republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to <https://www.ieee.org/publications/rights> to learn how to obtain a License from RightsLink.

Contribution Declaration: Seyed Mehdi Ayyoubzadeh (the author of this thesis) is the first author and main contributor of this article. He proposed the method, conducted experiments and composed the article. Prof. Xiaolin Wu is the supervisor of Seyed Mehdi Ayyoubzadeh. Xi Zhang helped with some ideas in the paper.

Chapter 5

Asymmetric Coding for Ultrahigh Throughput Encoding (ACUTE)

5.1 Abstract

Thanks to steady advancement of the imaging sensor technology, modern cameras can achieve very high precision in space and time. However, increasing the video frame rate while maintaining high spatial resolution is an arduous task because of the limited memory bandwidth to handle the huge data flow from the sensor array. The compression algorithms for high speed cameras must have an encoder whose throughput at least matches camera’s data rate. Also, the encoder needs to be simple enough for implementation in camera’s hardware pipeline. On the other hand, the decoder is not constrained by the limited on-camera computational resources and battery life; it can work off line and use powerful computers, such as GPUs, to reconstruct high speed videos. In this research, we propose an Asymmetric Coding scheme for Ultrahigh Throughput Encoding (ACUTE). ACUTE consists of a simple and fast encoder that can compress raw sensor data as fast as they are read out. In contrast to the light-duty encoder, the ACUTE decoder is a heavy-duty deep decompression CNN model that can achieve good rate-distortion performance. The key technical innovation is gradient coding by fast 2D lattice vector quantization at the encoder and optimized deep dequantization and super-resolution at the decoder.

5.2 Introduction

Thanks to ever increasing sophistication and capabilities of imaging technologies, professionals of many technical fields can now acquire images of very high resolutions simultaneously in spatial, spectral and temporal domains. One example is the wide use of high-speed cameras that can reach peak frame rate 1000 Hz and above; they find applications in safety studies (e.g., crash tests of automobiles), studies of high speed phenomena, aerospace, manufacturing and production, entertainment, etc. High speed in vivo imaging and ultrafast functional medical imaging are other examples. In remote sensing, hyper-spectral images need to offer high resolutions in both spatial and spectral domains. The above high-end imaging processes all require an ultrahigh data throughput. The total generated data volume per unit time is the product of the three resolutions in space, time and wavelength. As a result, the limiting factor for the total achievable precision over all domains is the memory bandwidth of the imaging device. This is why high speed cameras are compelled to sacrifice spatial resolution for higher temporal resolution.

In order to push the envelop of achievable precision in all imaged dimensions, one has to overcome the memory bandwidth bottleneck and hence comes the necessity of image compression. But unlike conventional image compression tasks, multidimensional high-resolution image compression faces a special challenge: the encoder throughput has to be high enough to match or exceed the memory bandwidth; it is the lower of the two speed specifications that determines how high a data rate can be sustained for a long duration of continuous shooting of high fidelity multidimensional cameras. A case in point is compression for ultrahigh speed video cameras. Conventional image and video compression methods, such as H.264/H.265 (Wiegand et al. 2003; Sullivan et al. 2012), JPEG (Wallace 1992), JPEG 2000, are not suited for the task. Although offering high compression, their encoding throughput is not high enough to sustain continuous operations of high speed cameras.

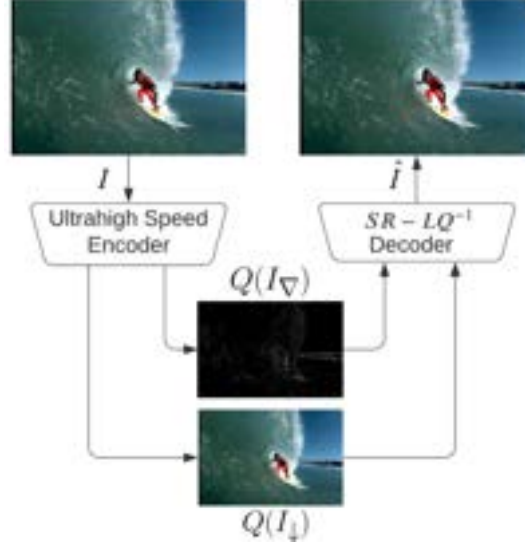


FIGURE 5.1: **Schematic diagram of the ACUTE system.** The encoder is exceptionally lightweight to achieve very high throughput. The compressed data consists of a downsampled image I_{\downarrow} and a lattice quantized gradient image I_{∇} . Decompression is done by a dual task CNN model (SR-LQ⁻¹) that performs soft gradient dequantization and superresolution.

The encoders of these compression methods all require a 2D signal transform (either DCT or wavelet), transform coefficient quantization, entropy coding (either Huffman or arithmetic coding), and in the case of H.264/H.265 motion compensation. Even if the 2D signal transformation can be implemented fast enough to keep up with the required throughput, it still must buffer many rows of the pixels in the image (8 in the case of DCT) in the readout process, generating a serious blockage of the data pipeline of high speed cameras. Apparently, none of existing image compression methods is designed to handle the ultrahigh data rate of video cameras running at 1000 Hz or above. In theory, compressive sensing (CS) is a possible solution to the problem. CS can greatly reduce the sampling rate and hence generate a much smaller amount of data in the first place. But none of CS-based compression techniques can match the rate-distortion performance of conventional image coding methods. In Wang et al. 2012, Wang et al. tried to mitigate the issue by a more efficient quantization technique; however, their encoder still cannot match the high throughput of high speed cameras. Another strategy to reduce the encoder complexity without sacrificing rate-distortion performance much is

distributed source coding (Dragotti and Gastpar 2009). The idea is not to aim at full removal of high-order statistical redundancies at the light encoder, but rather exploit such redundancies to make coding gains by the LZ-type algorithm (Ziv and Lempel 1977) at the heavy decoder. In fact, distributed video coding (DVC) has been thoroughly investigated. Compared with conventional hybrid video coding paradigm, DVC escapes the expensive motion estimation step and hence greatly reduces the encoder complexity (Girod et al. 2005). But the DVC encoder is still too expensive to meet the stringent throughput requirement of high speed cameras. To our best knowledge, the highest encoder throughput of digital cameras is achieved by what we propose in this paper, called the strategy of Asymmetric Coding for Ultrahigh Throughput Encoding (ACUTE). As outlined in Figure 5.1, the ACUTE image compression system employs, based on the principle of distributed source coding, a light-duty, hardware-friendly encoder and a coupled heavy-duty decoder. In an ACUTE system, the encoder complexity is reduced to minimum by forgoing prediction or transform, and by using fixed length code without entropy coding. The compression is performed by simple downsampling by a factor of 2 and storing a small piece of prior information on the original. The prior information is used to assist the decoder to solve an essentially image superresolution problem. Here we stress that the decoder of ACUTE should be expected to outperform SR methods, because the encoder knows the exact HR ground truth image, which is unknown for the conventional SR task, and sends the decoder the prior information on the original HR image. The rest of the paper is structured as follows. We propose the design procedure for a high-speed encoder. Then we introduce our deep learning-based decoder (SR-LQ⁻¹), followed by the experiments that show the effectiveness of ACUTE. Finally, we do the ablation study to demonstrate the effectiveness of each designed module.

5.3 Deep Learning based ACUTE Paradigm

In the ACUTE system design, the computation burdens of achieving high rate-distortion performance are shifted from encoder to decoder. No variable-length entropy coding greatly simplifies the code stream organization and parsing at the hardware level. Such

a maximally streamlined encoder can be integrated into an image sensor, and so it can achieve a coding speed as fast as sensor array readout speed, while still achieving compression ratio from 2:1 to 3:1. If the ACUTE decoder can recover the original image in perceptually lossless quality, then the real-time ACUTE encoder effectively doubles the camera’s raw data throughput without loss of image quality. The key issue when image compression is done by downsampling is how to recover high frequency features. An obvious treatment of the problem is to make downsampling scheme adaptive to local gradients and signal waveform, as suggested by Wu *et al.* (Wu et al. 2009). This will facilitate the recovery of details at the decoder, however, also inevitably increase the encoder complexity and hence defeat our original purpose of making the complexity of the ACUTE encoder as low as possible. Therefore, we deliberately leave the technical challenge to the decoder of the ACUTE image compression system and meet it like in distributed source coding. As such, all technical innovations and developments of this work are with respect to the ACUTE decoder. The task of the ACUTE decoder appears to be one of image superresolution (SR); but unlike in common SR setting, not only the downsampling kernel is exactly known to the ACUTE decoder, more importantly the ACUTE encoder also transmits some prior information (say local gradient, as to be elaborated in Section 5.4) on the very latent image. Now with the extra information and ample computation resources available to the decoder, we can strive for much higher reconstruction quality than the current state of the art methods of image superresolution. In Shu and Wu 2018, Shu and Wu proposed an ACUTE image compression method for ultrahigh speed cameras. They treated ACUTE decoding as a classical inverse problem and solved it via convex programming with the constraints of the prior information provided by the encoder.

In this work, encouraged by recent successes of deep learning in image restoration, we tap the power of CNN in nonlinear mapping to squeeze out extra coding gains from the ACUTE image compression paradigm. Our main innovation is to incorporate into the CNN a hexagonal A_2 lattice quantizer to code the 2D gradient prior information. Thanks to optimal space packing property and high regularity of A_2 lattice (Conway

and Sloane 1998), it improves quantization precision without materially increasing the ACUTE encoder complexity. More importantly, we design a soft lattice dequantization module to enforce a feasible region of all local gradients, and in this way aid the CNN superresolution module at the decoder. The two modules are end-to-end optimized in the training process. We denote by SR-LQ⁻¹, the above novel CNN architecture consisting of a superresolution (SR) subnetwork and a lattice dequantization (LQ⁻¹) subnetwork. To further enhance the performance of the SR-LQ⁻¹ network, we couple the A_2 lattice quantizer with a companding mapping (Ogunfunmi and Narasimha 2010; Sonawane and Khobragade 2013) to compensate for biases in source distributions. Although the proposed SR-LQ⁻¹ CNN has a training phase that involves a computationally expensive deep learning process, at the inference stage, the decoding speed is substantially higher than the classical convex programming solver (Andersen et al. 2011). Unlike the pure CNN end-to-end compression methods, the proposed SR-LQ⁻¹ method combines deep learning and hand-crafted modeling. To appreciate the advantages of our algorithm design, it should be noted that in foreseeable future, due to its sheer size, the CNN architecture for pure end-to-end compression (Nakanishi et al. 2019; Toderici et al. 2017; Chen et al. 2021a) cannot achieve the ultra high encoder throughput required by high frame rate cameras.

5.4 Compression by Downsampling and Lattice Quantization

In the proposed ACUTE image compression system, the encoder uses the basic idea presented in Shu and Wu 2018. In each 2×2 non-overlapping window of the image, the average value and diagonal gradients are computed. Let the i -th block in the image be

$$x_i = \begin{pmatrix} x_{i,1} & x_{i,3} \\ x_{i,2} & x_{i,4} \end{pmatrix} \quad (5.1)$$

The average value and diagonal gradients are denoted by

$$z_i = \frac{x_{i,1} + x_{i,2} + x_{i,3} + x_{i,4}}{4} \quad (5.2)$$

$$g_i = x_{i,1} - x_{i,4} \quad (5.3)$$

$$h_i = x_{i,2} - x_{i,3} \quad (5.4)$$

To minimize the encoder complexity, we quantize z uniformly into 8 bits for 8-bit pixel values, keeping the quantization distortion of z below 0.5.

Figure 5.1 presents the downsampled image I_{\downarrow} and the corresponding gradient image I_{∇} . In Shu and Wu 2018 the gradient components g_i and h_i are each uniformly quantized into 4 bits. In this work we replace the old scheme by a two-dimensional hexagonal A_2 lattice quantizer to code the gradient vector $\nabla_i = (g_i, h_i)$ as a whole. The regular structure of the A_2 vector lattice (shown in Figure 5.4) still retains low encoder complexity, while reducing quantization distortion by more efficient hexagonal space packing (Conway and Sloane 1998). However, one problem remains, that is the mismatch between the uniform cell size of the A_2 lattice and the highly skewed distribution of ∇_i , as shown in Figure 5.2.

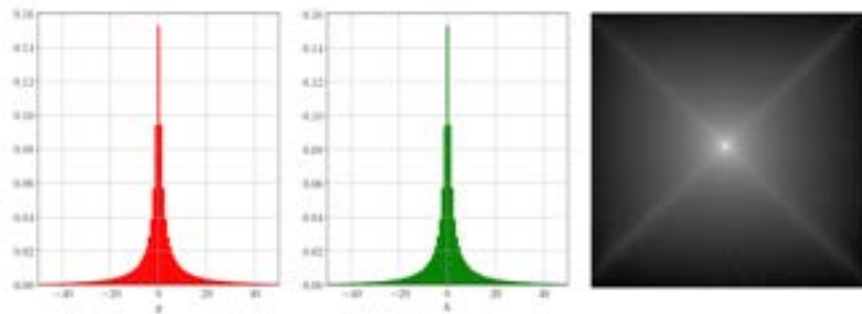


FIGURE 5.2: **Statistics of g and h .** Images in the DIV2K dataset (Agustsson and Timofte 2017b; Timofte et al. 2018) are sampled. The plots are all in log scale.

Compensating for distribution biases before quantization is one way of mitigating the coding loss of nonadaptive quantization. A simple and yet effective technique to even out

biased sample distribution is companding, such as the *mu*-law and A-law companding algorithms. They are used to improve the dynamic range of highly biased data (e.g., speech). For the task of ACUTE image compression, we choose the A-law algorithm to expand the gradient values near the origin, because it increases the quantization precision for values of high probability. This not only contributes to the ℓ_2 objective image quality but also has the perceptual benefit of making quantization distortions in smooth areas less noticeable.

Specifically, we apply the following A-law companding transformation to g and h before quantization:

$$F(x) = 255 \times \text{sgn}(x) \begin{cases} \frac{\alpha|x|}{1 + \ln(\alpha)} & |x| \leq \frac{1}{\alpha} \\ \frac{1 + \ln(\alpha|x|)}{1 + \ln(\alpha)} & \frac{1}{\alpha} \leq |x| \leq 1 \end{cases} \quad (5.5)$$

where x is the normalized g and h between -1 and 1 . $F(x)$ is illustrated for the unnormalized input in Figure 5.3. To reverse the expanding effect of the A-law transformation done by the encoder, the decoder uses inverse function F^{-1} to compress the received signal again and recover the actual values of g and h . To simplify notations, hereafter, the curly letters $\mathcal{G} = F(g)$ and $\mathcal{H} = F(h)$ are used to represent the expanded signals of the corresponding originals.

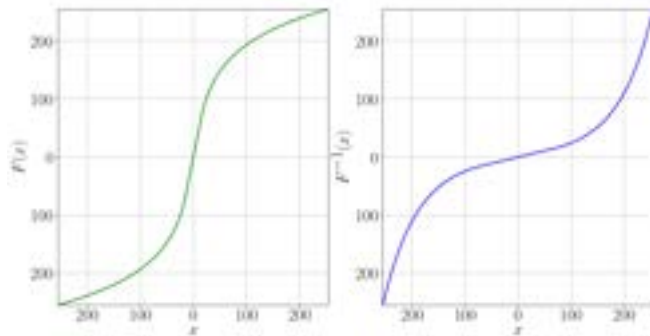


FIGURE 5.3: **A-Law companding functions for $\alpha = 17.62$.** The input x is unnormalized. Expansion function (left) versus compression function (right).

After the A-law mapping the next step is to quantize transformed gradient vector ∇_i in A_2 lattice. To achieve 2:1 compression ratio we encode each vector ∇_i with 8 bits, i.e., adopting a codebook of 256 lattice codewords. Unlike other vector quantization methods, the nearest neighbor encoding of A_2 lattice can be carried out in constant time independent of the codebook size. This is because the A_2 codewords are the centers of regularly structured hexagonal lattice cells. The generator matrix of these A_2 codewords is:

$$G_{A_2} = s \begin{pmatrix} 1 & \frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} \end{pmatrix} \quad (5.6)$$

In which s is a hyperparameter and can scale the quantization cells (hexagons) in the 2D space. A_2 lattice encoding process can be simply implemented by using the uniform quantizer. The lattice centroids are the union of two cosets linked by a global translation vector. The centroids in each coset are evenly distributed in the 2D space. Therefore, to find the corresponding centroid for an arbitrary point in the lattice quantizer, one can use two scalar quantizers to find two candidates in each coset and then pick one with the better distortion. The proposed encoding procedure for A_2 lattice ensures that time and space complexities are nearly as low as the uniform quantizer, while making significant performance gains as we will report and discuss in Figure 5.12.

Aiming to achieve the best possible performance under the stringent real-time throughput constraint, we jointly optimize the scaling factor s and the companding parameter α , by solving the following optimization problems in an off-line design step.

$$\begin{pmatrix} s^* \\ \alpha^* \end{pmatrix} = \operatorname{argmin}_{s, \alpha} \sum_{i=1}^N \left\| \begin{pmatrix} F^{-1}(Q(\mathcal{G}_i; s)) \\ F^{-1}(Q(\mathcal{H}_i; s)) \end{pmatrix} - \nabla_i \right\|_2^2 \quad (5.7)$$

Note that both \mathcal{G} and \mathcal{H} depends on the parameter α . The optimized values for s and α are 31.38 and 17.62 respectively. The Voronoi diagram of the designed quantizer are illustrated in Figure 5.4. The corresponding quantizer codebook consists of the 256 closest centroids to the origin (in ℓ_1 norm) produced by the generator matrix G . In effect, each of g and h is encoded into 8 bits by this lattice quantizer.

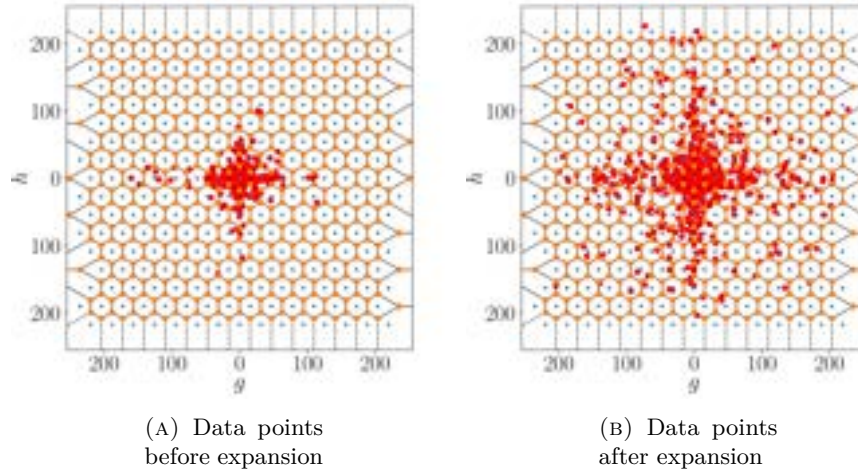


FIGURE 5.4: **Voronoi diagram of the quantizer.** Shown are the boundaries of the A_2 lattice quantizer and the expanded samples of g and h .

Finally, the encoder sends, for each 2×2 block i , the quantized block average z_i in 8 bits, and the 8-bits codeword index of the corresponding gradient ∇_i . The design and algorithmic flow of the ACUTE encoder are schematically depicted in Figure 5.5.

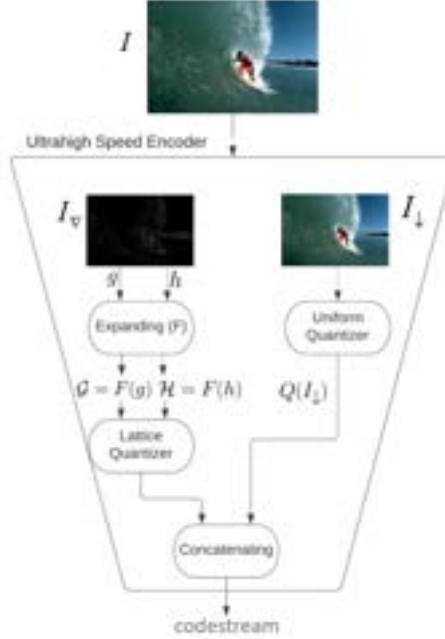


FIGURE 5.5: **Schematic diagram of the ACUTE encoder.** The input image is partitioned into 2×2 blocks. The gradient vector (g, h) goes through the expanding function F and then is coded by the A_2 lattice quantizer into 8 bits (the 2D lattice codebook has 256 codewords). These 8 bits are concatenated to the quantized 2×2 block average value z to form the code stream.

5.5 SR-LQ⁻¹ Decoder

Fortunately, for many applications of image/video compression, the decoder or receiver side is not limited by computational resources. Therefore, we can exploit the power of Deep Convolutional Neural Networks to remove or alleviate compression errors and hence achieve significantly higher reconstruction fidelity than conventional decoding. Specifically in the ACUTE compression diagram, we develop a CNN decoder to reverse two encoding operations, down sampling and quantization. This novel decoder is called hereafter the SR-LQ⁻¹ method or network for it solves the two underlying inverse problems of superresolution (SR) and lattice dequantization (LQ⁻¹) and optimize the solutions jointly. Unlike conventional decoders, the SR-LQ⁻¹ method can be a highly complex nonlinear mapping, if so required to recover the original image best possible. Also, it

should be noted that the LQ^{-1} lattice dequantization is a more sophisticated soft decoding operation than the conventional hard decoding of simply choosing the center of lattice cell.

In designing the $SR-LQ^{-1}$ decoder, the challenge is how to maximally profit from the prior knowledge on the local average z_i and gradient ∇_i of each pixel i . These priors tie the four samples x_1, x_2, x_3 and x_4 by three linear equations. Such strong constraints cannot be exploited by existing super-resolution CNN architectures when upsampling the local average value of the original image; even the best of SR CNNs do not respect, in general, the priors of z_i and ∇_i . In the next subsections, we present the $SR-LQ^{-1}$ CNN decoder in detail and justify our architecture design and optimization strategy. As schematically illustrated in Figure 5.6, the $SR-LQ^{-1}$ CNN decoder consists of two subnetworks: (i) a super-resolution CNN (SR) and (ii) a Soft Lattice Dequantization CNN (LQ^{-1}).

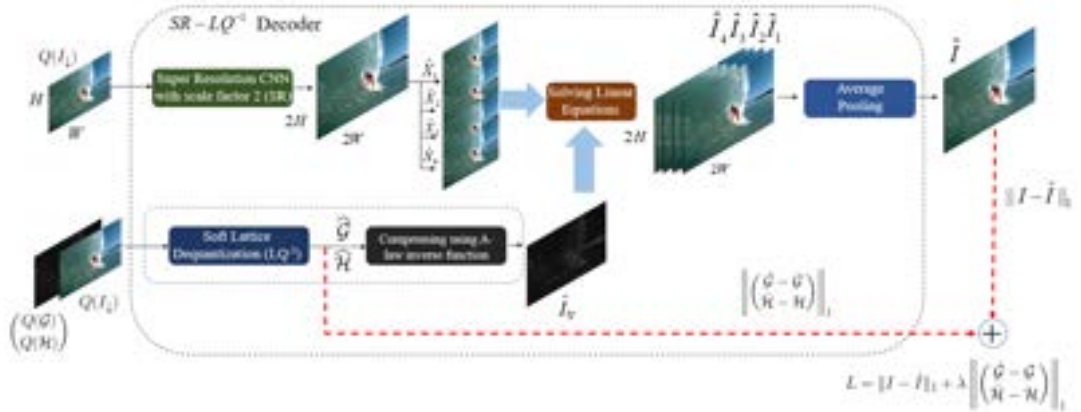


FIGURE 5.6: **Schematic diagram of the $SR-LQ^{-1}$ decoder.** The downsampled image I_d is fed to a super-resolution subnetwork to make an estimate of the original resolution. In parallel, the lattice dequantization subnetwork LQ^{-1} embarks on removing quantization errors in $Q(I_v)$ to restore the gradient image I_v . The results of these two subnetworks are combined to refine the final reconstruction image.

5.5.1 Super-resolution Module

In the $SR-LQ^{-1}$ method, any SR CNN can play the role of upsampling the 2×2 block average z to x_1, x_2, x_3 , and x_4 . We choose the EDSR CNN, one of the best known SR

architectures, in the place of SR module in our SR-LQ⁻¹ design. As well known, SR results, regardless produced by traditional or CNN methods, tend to be somewhat too smooth and less capable of recovering sharp details, representing a low-pass approximations of the original HR image. Precisely for overcoming this difficulty and recovering high-frequency features, the ACUTE encoder sends the quantized gradient image \hat{I}_∇ to aid the decoder. The SR-LQ⁻¹ decoder does not use the quantized gradient image $Q(I_\nabla)$ directly when refining high frequency details. Instead, it tries to reduce quantization errors and use an improved estimate \hat{I}_∇ of the gradient image I_∇ . To recover I_∇ , we design a soft lattice dequantization LQ⁻¹ subnetwork that is jointly end-to-end optimized with the SR subnetwork. This is the main technical innovation of SR-LQ⁻¹ to be detailed in the following subsection.

5.5.2 Soft Lattice Dequantization

One can easily observe from Figure 5.7 that the downsampled image I_\downarrow and the corresponding gradient image I_∇ have structural cross correlations, and also local correlations exist within I_∇ itself. Accordingly, we set out to restore the gradient image I_∇ from its lattice quantized version $Q(I_\nabla)$ via supervised deep learning, and use the restored gradient image \hat{I}_∇ to facilitate the final CNN reconstruction of the latent image I .



FIGURE 5.7: **Structural correlation between I_\downarrow and I_∇ .** The gradient image has both angle and magnitude which is color coded by value and hue in HSV space respectively. The presence of specific structures and shapes in I_∇ exhibits the local correlation in the gradient image. Moreover, the similarity between I_\downarrow and I_∇ shows the cross-correlation between these two images.

Arguably, even more beneficial is the lattice structure of the A_2 quantizer, which effectively confines the solution space for gradient vector ∇_i at each pixel i . As depicted in Figure 5.9 (left), after the expanding operation, the actual vector $(\mathcal{G}_i, \mathcal{H}_i)$ is bounded by

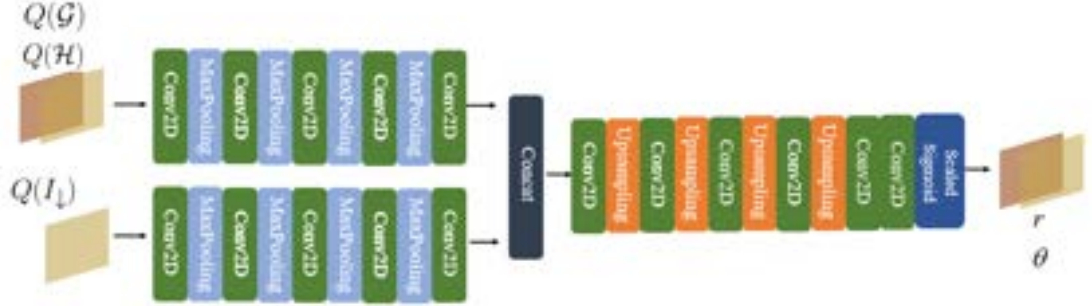


FIGURE 5.8: **The architecture of soft lattice dequantization LQ^{-1} .** LQ^{-1} extracts features from quantized downsampled image $Q(I_{\downarrow})$ and lattice quantized gradient image $Q(I_{\nabla})$, separately. These two sets of features are then fused by convolution and upsampling layers to estimate the original gradient vector (r, θ) in polar coordinates.

the hexagon quantizer cell. For algorithm amenability, we approximate this hexagonal domain by its inscribing circle with a negligible error near the vertices. In the training of the soft lattice dequantization LQ^{-1} CNN, we impose the above circular feasibility region. For easy implementation of the circular constraint we switch to polar coordinates of two parameters, radius r and angle θ , where the origin of the coordinate system is the center of the inscribed circle of each quantizer cell. This approximation can introduce small errors for points close to the hexagon vertices, and for the points laying outside of the maximum and minimum allowable values for the quantization (boundaries). However, the probabilities of these occurrences are negligible.

The proposed LQ^{-1} network module estimates the true gradient vector after expanding operation F in polar coordinates (r, θ) , namely,

$$\widehat{\mathcal{G}} = Q(\mathcal{G}) + r \cos(\theta) \quad (5.8)$$

$$\widehat{\mathcal{H}} = Q(\mathcal{H}) + r \sin(\theta) \quad (5.9)$$

and back to the original scale by applying F^{-1} :

$$\hat{g} = F^{-1}(\hat{\mathcal{G}}) \quad (5.10)$$

$$\hat{h} = F^{-1}(\hat{\mathcal{H}}) \quad (5.11)$$

The valid range for r is $[0, R_{max}]$, where R_{max} is the radius of the inscribed circle of each cell of the lattice quantizer (our optimal design of Eq(7) finds $R_{max} = 15.69$). The θ value is in the range $[0, 2\pi]$. The lower and upper bounds on r can be conveniently implemented in the LQ^{-1} CNN architecture by using the scaled sigmoid as the activation function in the subnetwork's last layer. The architecture of the LQ^{-1} module is illustrated in Figure 5.8.

5.5.3 Solving Linear Equations

For the i -th 2×2 pixel block, each of the values z_i , g_i and h_i defines a linear equation in terms of the four pixel values $x_{i,1}$, $x_{i,2}$, $x_{i,3}$ and $x_{i,4}$. We have three equations for four unknowns. If the value of one pixel in the block is fixed, the other pixels can be determined uniquely by solving the linear equations for that block. Therefore, there are four different solutions depending on which pixel value in block i is known:

$$\hat{I}_1 = \begin{pmatrix} x_{i,1} \\ 2z_i - x_{i,1} + \frac{g_i+h_i}{2} \\ 2z_i - x_{i,1} + \frac{g_i-h_i}{2} \\ x_{i,1} - g_i \end{pmatrix} \quad \hat{I}_2 = \begin{pmatrix} 2z_i - x_{i,2} + \frac{h_i+g_i}{2} \\ x_{i,2} \\ x_{i,2} - h_i \\ 2z_i - x_{i,2} + \frac{h_i-g_i}{2} \end{pmatrix} \quad (5.12)$$

$$\hat{I}_3 = \begin{pmatrix} 2z_i - x_{i,3} + \frac{g_i-h_i}{2} \\ x_{i,3} + h_i \\ x_{i,3} \\ 2z_i - x_{i,3} - \frac{g_i+h_i}{2} \end{pmatrix} \quad \hat{I}_4 = \begin{pmatrix} x_{i,4} + g_i \\ 2z_i - x_{i,4} + \frac{h_i-g_i}{2} \\ 2z_i - x_{i,4} - \frac{h_i+g_i}{2} \\ x_{i,4} \end{pmatrix} \quad (5.13)$$

\hat{I}_j , $j = 1, 2, 3, 4$, is an estimate of block i by setting the value of $x_{i,j}$ to the output

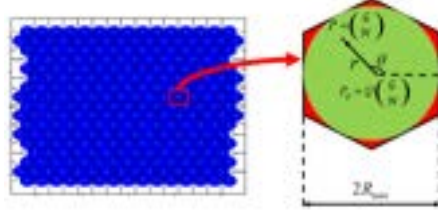


FIGURE 5.9: **Left: lattice quantizer cells; Right: the inscribed circle for one of the cells.** In the right image, P is the actual expanded gradient vector $(\mathcal{G}, \mathcal{H})$ in terms of r and θ ; the cell center P_Q is the quantized P $(\hat{\mathcal{G}}, \hat{\mathcal{H}})$.

value of the upsampling network at location j of block i and using the quantized block average z_i and the estimated gradient vector (\hat{g}_i, \hat{h}_i) . We average the above four estimated images to obtain the final soft decoded image. This fusion strategy is motivated by our observation that all the operations in the proposed soft decoder work flow are differentiable, thus we can train the decoder network end-to-end with an appropriate loss function (refer to Figure 5.6). In the training stage, we use both the original image I and the gradient image (g, h) to supervise the learning of the SR-LQ⁻¹ model. Specifically, the loss function per image in terms of \hat{I}_j and the restored expanded gradients $\hat{\mathcal{G}}$ and $\hat{\mathcal{H}}$ is

$$L(I, \hat{I}) = \|I - \frac{1}{4} \sum_{j=1}^4 \hat{I}_j\|_1 + \lambda \left\| \begin{pmatrix} \hat{\mathcal{G}} - \mathcal{G} \\ \hat{\mathcal{H}} - \mathcal{H} \end{pmatrix} \right\|_1$$

where I is the ground truth, \hat{I} is the output of the SR-LQ⁻¹ model and λ determines the importance of the recovery of true gradients.

5.6 Experiments and Evaluations

This section presents empirical evidences for the competitive rate-distortion performance of the proposed ACUTE CNN soft decoding system despite the extremely simplistic encoder. We compare ACUTE with three competing image compression methods of two categories. In the first category, the compression and decompression are simply downsampling and upsampling operations, like ACUTE. Two different methods of this

category are included in the comparison group. The first one is to replace the SR-LQ⁻¹ decoder by a deep learning-based upsampler, the EDSR network (Lim et al. 2017b), with the upsampling factor of 2. EDSR is among the best performing image super-resolution CNN models. The second method shares the same encoder as ACUTE, but it adopts a convex programming approach to solve the inverse problem of soft decoding (Shu and Wu 2018), instead of deep learning. The second category includes traditional image compression methods without machine learning in either the encoder or the decoder. JPEG is selected to represent this category because it is most widely used and it offers both good rate-distortion performance and high speed.

Various datasets are used to validate the generalization capability of the learnt SR-LQ⁻¹ soft decoder, which is trained by the DIV2K dataset, over different types of images. They are the DIV2K validation set (excluded from the training set), BSD100 (Martin et al. 2001b), Set 14 (Bevilacqua et al. 2012b), Set5 (Zeyde et al. 2012b), CLIC (Toderici et al. 2020) and KODAK (*Kodak Image Set n.d.*). Two well-known image quality metrics, PSNR and SSIM, are used to evaluate the quality of the decoded images.

5.6.1 Experiment Setting

The upsampler module in SR-LQ⁻¹ has the same architecture as EDSR with upscaling factor 2. Randomly selected 256 × 256 patches are used to train the SR-LQ⁻¹ network; the training is carried out by the Yogi optimizer (Zaheer et al. 2018), with an initial learning rate of 10⁻⁴. The training process takes 50000 iterations and sets the minibatch size to 8.

At the inference time, we use the self-ensembling technique similar to the proposed method in Lim et al. 2017b to increase the performance of ACUTE. Specifically, each test image goes through multiple geometric transforms (rotation by 0, 90, 180, 270 degrees, flipping vertically and horizontally) to have different versions. Each of these transformed versions is fed to the SR-LQ⁻¹ network and restored. Then the inverse geometrical transform is applied to bring each restored version back to the original geometry. The median over all of these restored results is taken as the final output.

When compressing the images with the JPEG algorithm, we run for three different quality factors, 85, 90, and 92. The quality factor of 85 is typically used as the default value in practice. The quality factors of 90 and 92 roughly give the average compression ratio of 2:1 for the natural images. It should be stressed that JPEG has much higher encoder complexity than the encoder of ACUTE. The former is not suited for extremely high throughput imaging devices, such as very high frame rate video cameras, while the latter is.

5.6.2 Results

The performance results of tested compression methods in quantitative quality metrics of PSNR and SSIM are tabulated in Table 5.1. As exhibited, our method outperforms EDSR, Shu and Wu 2018 and JPEG (QF=85) for all the datasets, which proves the generalization capability of ACUTE. ACUTE even surpasses JPEG of a high-quality factor (QF=90) for all of the datasets except Set14. For QF=92, as can be seen, the JPEG starts to beat ACUTE in reconstruction quality, while yielding compression ratio around 2:1. However, keep in mind that the JPEG algorithm is too complex and too slow to run in ultra-high speed camera data processing pipeline.

JPEG compression involves two computationally demanding steps: DCT and entropy encoding. DCT requires 64 multiplications and 56 additions for each 8×8 block, while entropy encoding (using either arithmetic or Huffman coding) requires expensive operations to estimate the probability distribution and encode the block. In contrast, the ACUTE encoder only requires 16 multiplications and 80 additions to encode each 8×8 block, and it does not require the entropy encoding step. This makes the ACUTE encoder much faster than JPEG in terms of processing time. In addition, JPEG compression requires buffering 8 lines of the image during the encoding process. On the other hand, the ACUTE encoder only requires 2 lines of the image to begin encoding, making it more memory-efficient than JPEG.

A comparison between the results of EDSR and ACUTE indicates that sending gradient information can significantly improve the quality of the decoded images. Figure

5.10 compares different compression methods in visual quality. We choose scenes of high speed movements considering that the ultra-high throughput ACUTE encoder is designed for sustained shooting of high frame rate videos. As illustrated, our method can preserve important details for fast-moving objects (e.g., edges and fine textures) better than other methods, with a greater degree of sharpness and clarity.

5.6.3 Ablation Study

In this section, we conduct various experiments to verify the role and evaluate the effectiveness of the individual components of the ACUTE system.





















Original Image	EDSR	Shu and Wu 2018	ACUTE (Ours)	JPG (QF=90)
	 27.37 / 0.94	 31.56 / 0.97	 37.56 / 0.99	 38.67 / 0.99
	 26.60 / 0.92	 28.32 / 0.96	 35.73 / 0.99	 38.25 / 0.98
	 31.77 / 0.93	 34.65 / 0.98	 41.20 / 0.99	 40.48 / 0.98
	 24.01 / 0.83	 28.21 / 0.94	 33.06 / 0.98	 37.13 / 0.98

FIGURE 5.10: Visual comparison of different compression methods for scenes of very fast motions (PSNR/SSIM).

TABLE 5.1: Comparison of different compression methods in quantitative image quality for common test image sets.

	EDSR	Shu and Wu 2018	JPEG			ACUTE
	Lim et al. 2017b		QF=85	QF=90	QF=92	
DIV2K						
PSNR	33.40	36.59	39.26	41.27	42.42	41.82
SSIM	0.93	0.97	0.97	0.98	0.98	0.99
Ratio	4	2	3.19	2.5	2.26	2
BSD100						
PSNR	29.96	33.96	37.33	38.81	42.04	39.08
SSIM	0.89	0.96	0.97	0.98	0.99	0.99
Ratio	4	2	2.76	2.05	1.85	2
Set14						
PSNR	31.32	34.22	38.54	41.25	42.66	39.68
SSIM	0.90	0.96	0.96	0.98	0.98	0.99
Ratio	4	2	2.73	2.17	1.98	2
Set5						
PSNR	35.66	37.55	40.49	42.38	44.0	42.97
SSIM	0.96	0.98	0.98	0.98	0.98	0.99
Ratio	4	2	2.92	2.37	2.19	2
CLIC (test)						
PSNR	35.61	38.13	40.61	42.35	43.33	42.87
SSIM	0.94	0.97	0.97	0.98	0.98	0.99
Ratio	4	2	3.77	2.87	2.57	2
CLIC (val)						
PSNR	35.06	38.50	40.13	41.91	42.91	43.16
SSIM	0.93	0.97	0.97	0.98	0.98	0.99
Ratio 4	2	2	3.54	2.72	2.44	2
KODAK						
PSNR	31.67	35.40	38.18	40.23	41.42	40.83
SSIM	0.90	0.97	0.96	0.97	0.98	0.99
Ratio	4	2	2.97	2.33	2.11	2
Average						
PSNR	33.24	36.33	39.22	41.17	42.68	41.48
SSIM	0.92	0.96	0.96	0.97	0.98	0.99
Ratio	4	2	3.12	2.43	2.2	2

Soft Lattice Dequantization Module (LQ^{-1})

To appreciate how effective the LQ^{-1} soft dequantization is to reduce the quantization error, we present, in Figure 5.11, four examples of the lattice quantized gradient image $(Q(g), Q(h))$, the LQ^{-1} restored gradient image (\hat{g}, \hat{h}) , and the original gradient image (g, h) , so one can visualize the accuracy improvements of LQ^{-1} over hard dequantization in both angle (coded by color) and amplitude (coded by intensity) of the local gradients. As an objective error metric, the mean absolute errors (MAE) for each sample image are also given in the Figure 5.11.

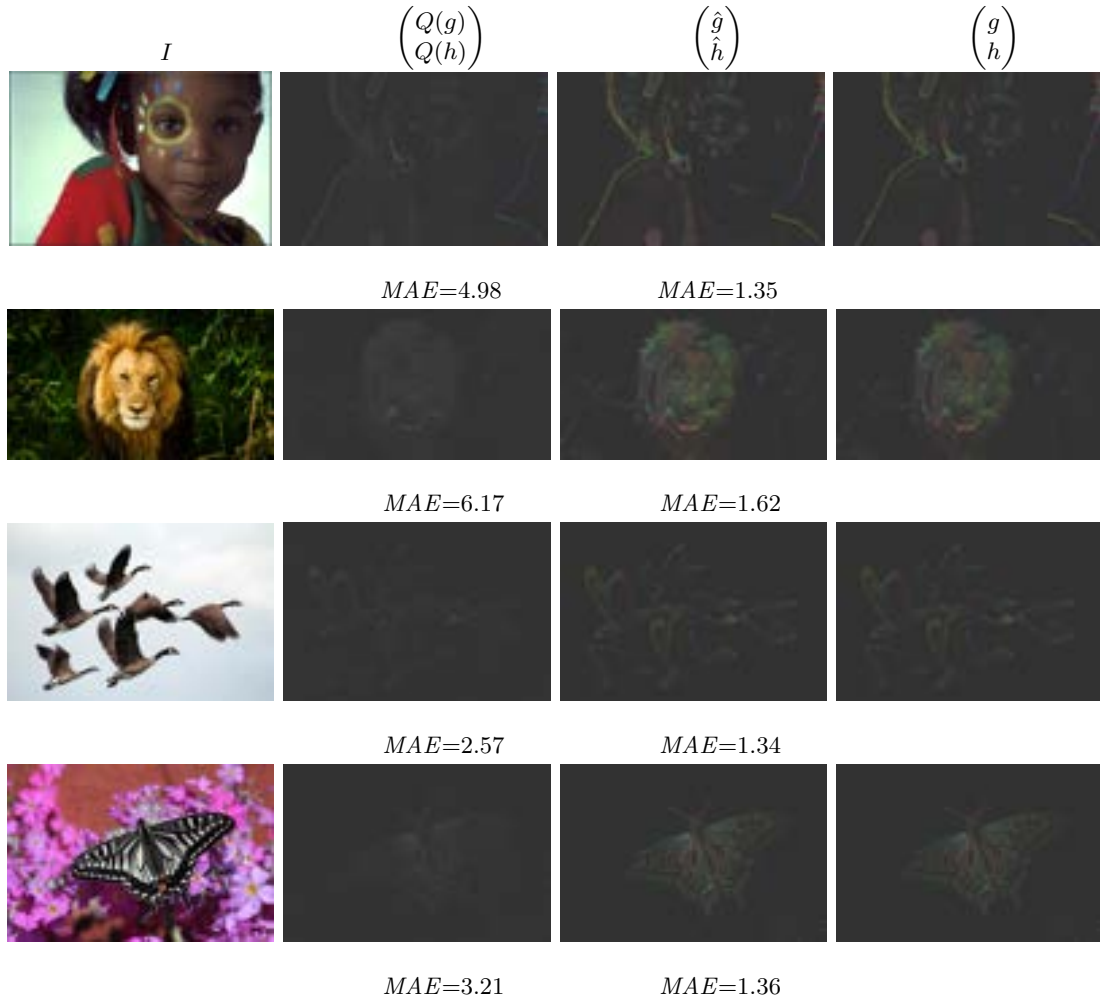


FIGURE 5.11: Visualizing the improved precision of quantized gradient images I_{∇} by soft lattice dequantization (LQ^{-1}) module for some test samples. From left to right: the original images, quantized gradient images, improved gradient images after soft dequantization LQ^{-1} , the true gradient images.

Different Upsampling Modules

The proposed method can use any available super-resolution network as the upsampler as long as they are differentiable. We have tested our method with three well-known super-resolution networks: RCAN (Zhang et al. 2018b), EDSR (Lim et al. 2017b), and WDSR (Yu et al. 2018a). The results are reported in Table 5.2.

TABLE 5.2: Comparison of various super resolution networks as the upsampler module in SR-LQ⁻¹.

	SR-LQ ⁻¹ (EDSR)	SR-LQ ⁻¹ (WDSR)	SR-LQ ⁻¹ (RCAN)
DIV2K			
PSNR	41.82	41.49	41.53
SSIM	0.99	0.99	0.99
BSD100			
PSNR	39.00	38.83	38.90
SSIM	0.99	0.99	0.99
Set14			
PSNR	39.61	39.43	39.57
SSIM	0.99	0.99	0.98
Set5			
PSNR	42.92	42.81	42.85
SSIM	0.99	0.98	0.99

Comparing various quantizers

To demonstrate the advantages of the A_2 lattice quantizer for encoding the gradient image I_{∇} in the proposed ACUTE compression system, we compare it, in Figure 5.12, against various other quantizers in quantization precision (PSNR), including 1D Lloyd-Max quantizer (1D K-means), 2D Lloyd-Max quantizer (2D K-means), and uniform quantizer. In the case of 1D K-means, we design two optimal scale quantizers for g and h separately with $K = 16$. In the case of 2D K-means, we design a 2D vector quantizer of $K = 256$ codewords by clustering g and h in the 2D space. As exhibited, the proposed lattice quantizer coupled with companding has the best performance in the comparison group.

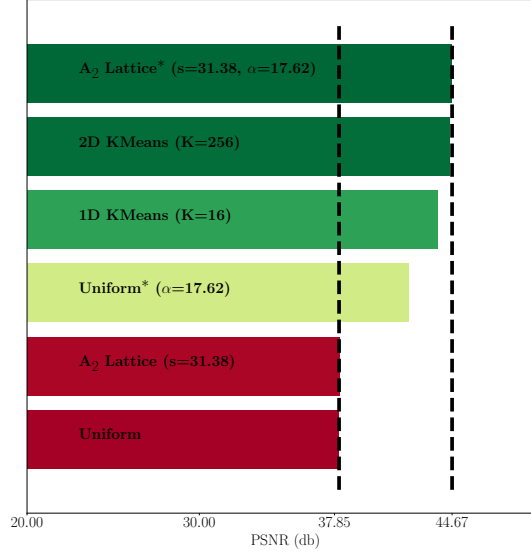


FIGURE 5.12: **PSNR of different quantizers.** * means A-law companding is performed before quantization.

Do companding and A_2 quantization matter?

One may question the necessity of employing the A_2 quantizer coupled with companding at the encoder, considering that a powerful deep learning-based decoder like SR-LQ⁻¹ can be made highly nonlinear and complex to compensate for the rigidity of uniform quantization. To clear any doubt, we train the deep decoder separately for the scheme of companding followed by A_2 2D lattice quantization and for simple uniform scalar quantization of z, g, h . PSNR and SSIM values of reconstructed images, averaged over the DIV2K validation data, are plotted and compared in Figure 5.13 for the two schemes. The same architecture for the SR module is used in the two cases to have a fair comparison. These results show that the latter is considerably inferior to the former in reconstruction quality.

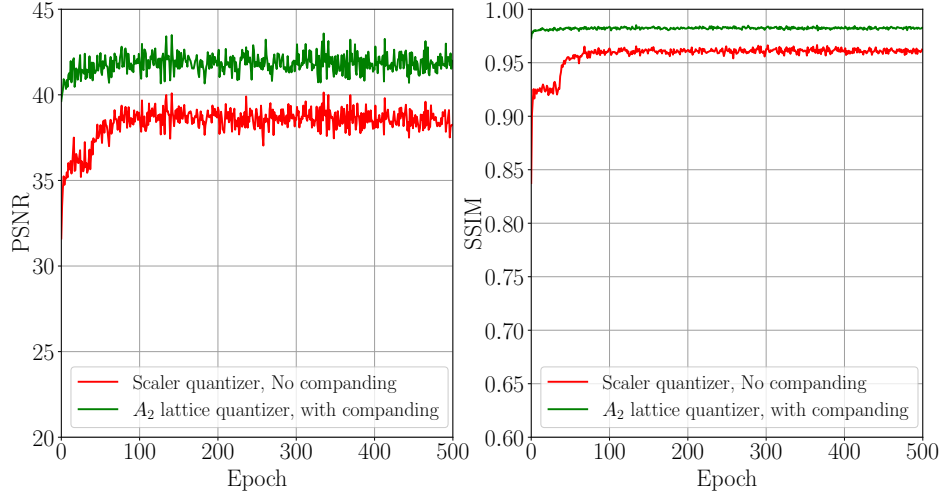


FIGURE 5.13: **PSNR and SSIM trends vs the number of training epochs for DIV2K validation data.** The red graphs are for scalar quantizer without companding. The green graphs demonstrate the higher performance of $SR-LQ^{-1}$.

5.7 Conclusion

We have developed an asymmetric image compression system (ACUTE) of an ultra-high throughput encoder coupled with a powerful deep soft decoder. The proposed ACUTE system can sustain long duration of real-time video compression at very high frame rates or similar scenarios, while offering competitive rate-distortion performance. The light, hardware-friendly but efficient encoder is made possible by simple down sampling and 2D lattice quantization of gradients with companding. The learnt $SR-LQ^{-1}$ decoder thoroughly exploits the gradient information transmitted by the encoder to superresolve the downsampled image in good precision.

Chapter 6

Conclusion

While deep learning-based image restoration neural networks have made remarkable progress in recent years, they still face challenges in restoring sharp, clean high-frequency details and textures. This thesis identifies the lack of suitable priors on high-frequency information as the leading cause of this deficiency.

The thesis introduces novel techniques for incorporating informative high-frequency priors into neural restoration models, including encouraging convolutional neural networks' filters to extract valuable frequency information from images using a pre-designed filter bank, modifying the loss function of the restoration model to emphasize the vital high-frequency details, incorporating an auxiliary loss function on the metadata to reduce the domain shift issue, and integrating the desired priors within the model architecture.

The proposed techniques enhance the frequency characteristics of neural networks' produced images. To support the effectiveness of our proposed approaches, we conduct extensive experiments for various image restoration tasks.

Appendix A

Chapter 2 Supplement

A0.1 Results on image super resolution

To show that FBR is an effective method to regularize the DCNN in image restoration tasks, we evaluated different regularization strategies for image super-resolution on EDSR Lim et al. 2017a architecture. We trained the models with random 64×64 patches of DIV2K dataset with the batch size of 8 for 200 epochs. The regularization is applied to the convolutional layers in the residual blocks of EDSR architecture. PSNR and SSIM for different strategies on the validation split of DIV2K dataset are shown in Table . As presented, FBR outperforms other regularization methods in image super-resolution.

Reg. Type	PSNR	SSIM
Baseline	24.8	0.68
Ortho. ($\gamma = 10^{-4}$)	24.95	0.68
ℓ_1 ($\gamma = 0.01$)	24.7	0.67
ℓ_2 ($\gamma = 10^{-6}$)	24.9	0.68
FBR ($\lambda = 0.0001, \gamma = 10^{-5}$)	25.29	0.69

TABLE A1.1: PSNR and SSIM on DIV2K validation dataset

A0.2 Large Size Image Classification

To show that FBR is an effective method to regularize the DCNN on the large scale images as well as the small scale images, we use ImageNet Russakovsky et al. 2015 dataset. It contains color images (224×224) from 1000 different objects. We use 100

classes from the objects to train the DCNN. We use ResNet-50 architecture He et al. 2015 as our baseline model. We could see the training loss and top-5 accuracy on the validation data in the Figure A1.1 (A) and (B) respectively.

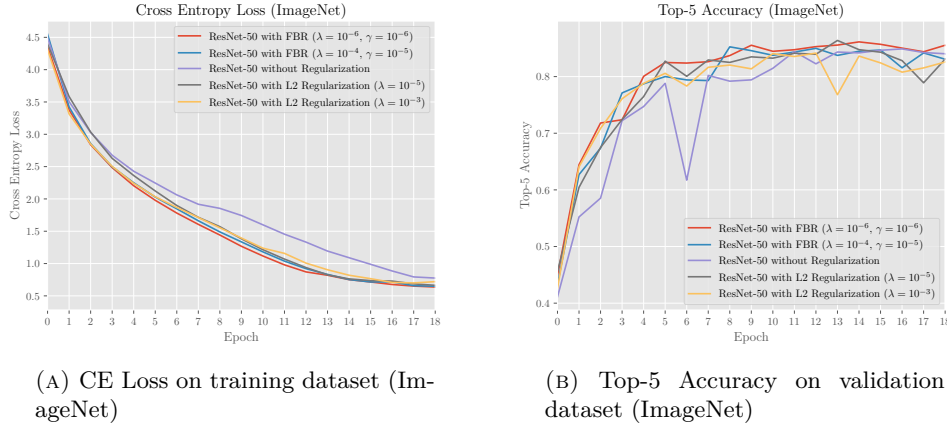


FIGURE A1.1

As shown in Figure A1.1 (A), both models that regularized with FBR have the lower training loss in comparison with $L2$ regularization or baseline model without regularization. For the validation, as one could see in Figure A1.1 (B), the model with FBR regularization has the best accuracy. In addition, the abrupt changes in validation accuracy for FBR model is less than other models.

A0.3 Results of using a VGG-derived filter bank as the regularizer

We can also construct the regularization filter bank using the lower layer convolutional kernels of some pretrained DCNNs, for example, those of VGG16. As mentioned previously, Gabor filters do not work effectively if the filter kernel is small. One way of creating a regularization filter bank of a small kernel size is to choose a subset of pretrained VGG convolutional kernels at first few front layers. Specifically, we randomly select 256 VGG16 kernels of the first two layers pre-trained on Imagenet to form the regularization filter bank, which is shown in Figure A1.2.

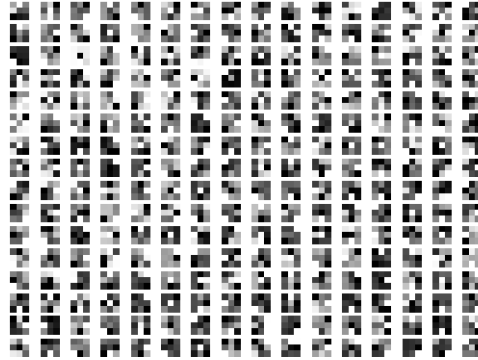


FIGURE A1.2: 256 sampled filters (3×3) from pretrained VGG16

For comparison purposes, we train the DCNN of Figure 2.4 to classify the Caltech 101 dataset, with the above VGG-derived filter bank regularization, the ℓ_2 regularization, and without any weight regularization at all (the baseline), and compare the performances of these methods. The classification accuracy and cross entropy results are displayed in Figures A1.3 (a) and (b), respectively. As shown, the DCNNs regularized by the VGG16-derived 3×3 filter bank outperform the ℓ_2 -regularized DCNN and the baseline model without regularization.

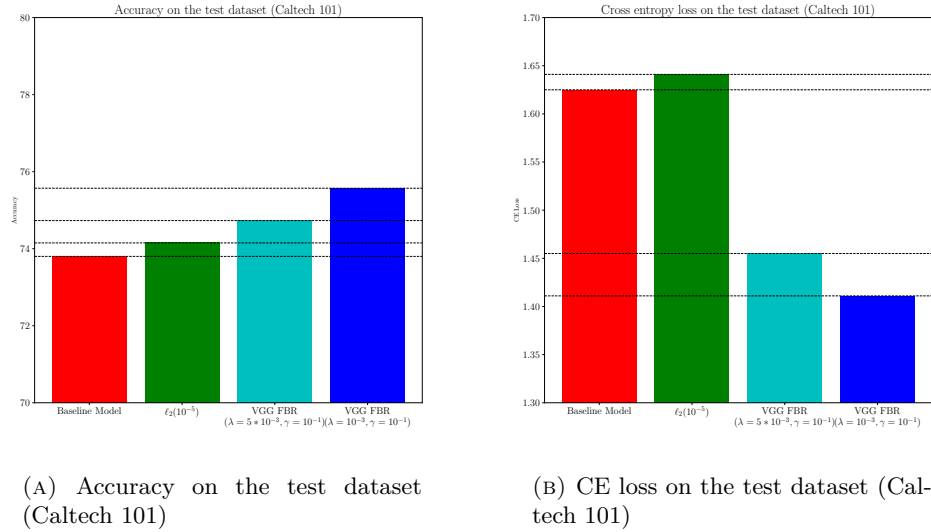
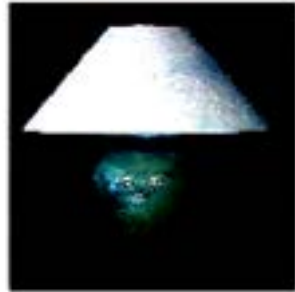


FIGURE A1.3

Comparison of different regularization methods based on feature maps

In order to understand how the proposed FBR method works and its advantages over the other methods, we examine the DCNN feature maps generated under different and without regularizations. First we compare the baseline model without regularization and the FBR regularization method. Let us examine two examples from the Caltech 101 test dataset on which the baseline model misclassifies whereas the FBR method correctly classifies. The two test images after normalization (mean subtracted and then divided by standard deviation) are shown in Figures A1.4.



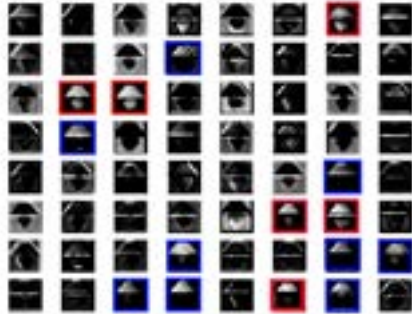
(A) FBR prediction:
"lamp", Baseline Pre-
diction: "nautilus"



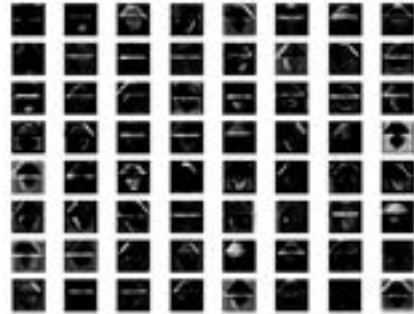
(B) FBR prediction:
"headphone", Baseline
Prediction: "scissors"

FIGURE A1.4: Example images

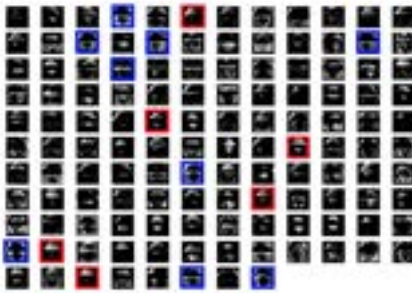
The feature maps of layer 1 and layer 3 for the baseline and the FBR method are displayed in Figure A1.5 and Figure A1.6 respectively. The similar feature maps of the baseline model are marked in these Figs/fbr



(A) First layer, baseline model



(B) First layer, CNN using FBR



(C) Third layer, baseline model

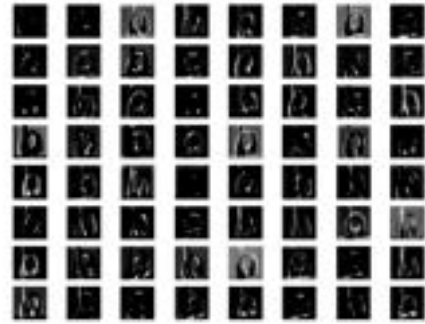


(D) Third layer, CNN using FBR

FIGURE A1.5: Feature maps for the first test image (Very similar feature maps are marked with red and blue colors)



(A) First layer, baseline model



(B) First layer, CNN using FBR



(C) Third layer, baseline model



(D) Third layer, CNN using FBR

FIGURE A1.6: Feature maps for the second test image

As can be easily observed in the figures, the feature maps of the FBR method are more sparse than those of the baseline model without regularization. This increased sparsity improves the robustness of the FBR method. Also, we bring the reader's attention to interpreting the feature maps of the FBR method in Figures A1.5 and A1.6. Thanks to the Gabor filters included in the regularization filter bank, the FBR method extracts features of strong directionality and high frequency that may explain the superior classification performance of the FBR method.



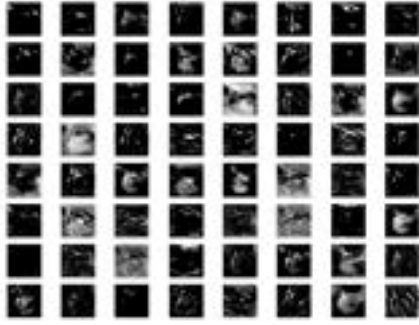
(A) FBR prediction: "flamingo", ℓ_2 Prediction: "ibis"



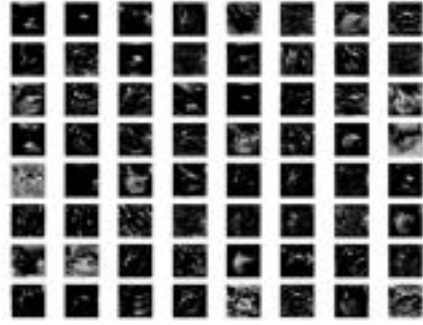
(B) FBR prediction: "pagoda", ℓ_2 Prediction: "accordion"

FIGURE A1.7: Example images

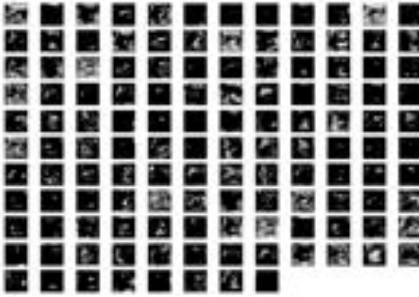
Next, we compare the FBR method and the ℓ_2 regularization. Again, two sample images of the CalTech 101 dataset are selected and shown in Figures A1.7. For these two images the FBR method correctly classifies, whereas the ℓ_2 regularization does not. The feature maps of layer 1 and layer 3 for the two methods are shown in Figures A1.8 and A1.9.



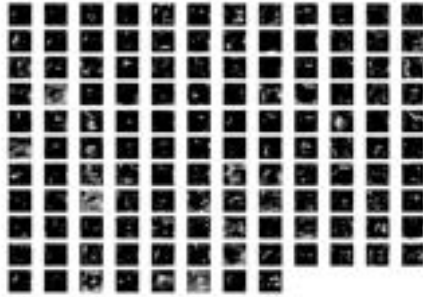
(A) First layer, baseline model with ℓ_2



(B) First layer, CNN using FBR

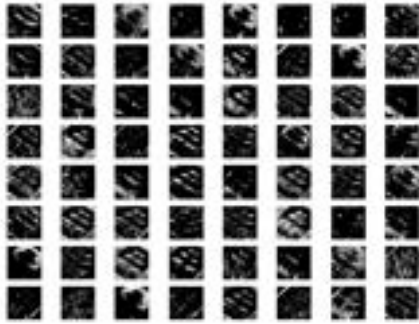


(C) Third layer, baseline model with ℓ_2

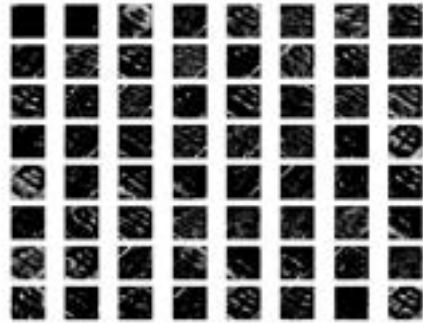


(D) Third layer, CNN using FBR

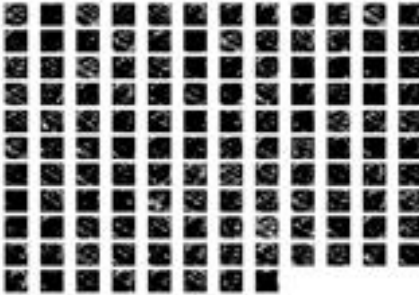
FIGURE A1.8: Feature maps for the first test sample (FBR in comparison with ℓ_2)



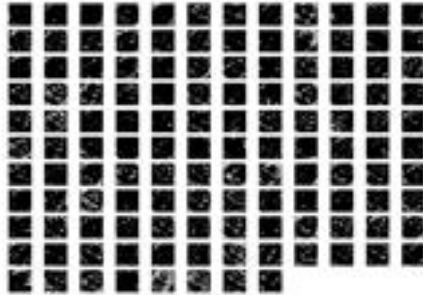
(A) First layer, baseline model with ℓ_2



(B) First layer, CNN using FBR



(C) Third layer, baseline model with ℓ_2



(D) Third layer, CNN using FBR

FIGURE A1.9: Feature maps for the first test sample (FBR in comparison with ℓ_2)

Here, the observations are very similar to what we discussed about Figures A1.5 and A1.6. The feature maps of the FBR method appear to be sparser and exhibit greater discriminating power in high frequency and directionality than the ℓ_2 regularization. This explains the superior performance of the former over the latter.

Bibliography

- Agrawal, A., Verschueren, R., Diamond, S., and Boyd, S. (2018). A Rewriting System for Convex Optimization Problems. *Journal of Control and Decision* 5(1), 42–60.
- Agustsson, E. and Timofte, R. (July 2017a). NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.
- Agustsson, E. and Timofte, R. (July 2017b). NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.
- Andersen, M., Dahl, J., Liu, Z., and Vandenberghe, L. (2011). Interior-point methods for large-scale cone programming. In: *Optimization for Machine Learning*. The MIT Press.
- Ayyoubzadeh, S. M. and Royat, A. (June 2021). (ASNA) An Attention-based Siamese-Difference Neural Network with Surrogate Ranking Loss function for Perceptual Image Quality Assessment. In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE.
- Banham, M. and Katsaggelos, A. (1996). Spatially adaptive wavelet-based multiscale image restoration. *IEEE Transactions on Image Processing* 5(4), 619–634.
- Bansal, N., Chen, X., and Wang, Z. (2018). *Can We Gain More from Orthogonality Regularizations in Training Deep CNNs?*
- Bevilacqua, M., Roumy, A., Guillemot, C., and Morel, M.-l. A. (2012a). Low-Complexity Single-Image Super-Resolution based on Nonnegative Neighbor Embedding. In: *Proceedings of the British Machine Vision Conference*. BMVA Press, 135.1–135.10. ISBN: 1-901725-46-4.

- Bevilacqua, M., Roumy, A., Guillemot, C., and Morel, M.-l. A. (2012b). Low-Complexity Single-Image Super-Resolution based on Nonnegative Neighbor Embedding. In: *Proceedings of the British Machine Vision Conference*. BMVA Press, 135.1–135.10. ISBN: 1-901725-46-4.
- Boudjelal, A., Messali, Z., and Attallah, B. (2018). PET image reconstruction based on Bayesian inference regularised maximum likelihood expectation maximisation (MLEM) method. *International Journal of Biomedical Engineering and Technology* 27(4), 337.
- Brox, T., Bruhn, A., Papenberg, N., and Weickert, J. (2004). High accuracy optical flow estimation based on a theory for warping. In: *Proc. of the European Conf. on Computer Vision*. Springer, 25–36.
- Brox, T. and Malik, J. (2010). Large displacement optical flow: descriptor matching in variational motion estimation. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)* 33(3), 500–513.
- Bruna, J. and Mallat, S. (2012). *Invariant Scattering Convolution Networks*.
- Butler, D. J., Wulff, J., Stanley, G. B., and Black, M. J. (Oct. 2012). A naturalistic open source movie for optical flow evaluation. In: *Proc. of the European Conf. on Computer Vision*, 611–625.
- Cao, H., Yu, S., and Feng, J. (2019). Compressed video action recognition with refined motion vector. *arXiv preprint arXiv:1910.02533*.
- Caruana, R., Lawrence, S., and Giles, L. (2000). Overfitting in Neural Nets: Backpropagation, Conjugate Gradient, and Early Stopping. In: *IN PROC. NEURAL INFORMATION PROCESSING SYSTEMS CONFERENCE*, 402–408.
- Chan, T.-H., Jia, K., Gao, S., Lu, J., Zeng, Z., and Ma, Y. (2014). PCANet: A Simple Deep Learning Baseline for Image Classification?
- Chen, P., Yang, W., Sun, L., and Wang, S. (2020). When Bitstream Prior Meets Deep Prior: Compressed Video Super-resolution with Learning from Decoding. In: *Proc. of the ACM International Conf. on Multimedia (ACM MM)*, 1000–1008.

- Chen, T., Liu, H., Ma, Z., Shen, Q., Cao, X., and Wang, Y. (2021a). End-to-End Learnt Image Compression via Non-Local Attention Optimization and Improved Context Modeling. *IEEE Transactions on Image Processing* 30, 3179–3191.
- Chen, X., Yuan, Y., Zeng, G., and Wang, J. (June 2021b). Semi-Supervised Semantic Segmentation With Cross Pseudo Supervision. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2613–2622.
- Chi, Z., Wang, Y., Yu, Y., and Tang, J. (2021). Test-Time Fast Adaptation for Dynamic Scene Deblurring via Meta-Auxiliary Learning. In: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 9137–9146.
- Chollet, F. et al. (2015). *Keras*. <https://github.com/fchollet/keras>.
- Conway, J. and Sloane, N. (1998). *Sphere Packings, Lattices and Groups*. Grundlehren der mathematischen Wissenschaften. Springer New York. ISBN: 9780387985855.
- Daugman, J. G. (July 1985). Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *Journal of the Optical Society of America A* 2(7), 1160.
- Diamond, S. and Boyd, S. (2016). CVXPY: A Python-Embedded Modeling Language for Convex Optimization. *Journal of Machine Learning Research* 17(83), 1–5.
- Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., Smagt, P. van der, Cremers, D., and Brox, T. (Dec. 2015). FlowNet: Learning Optical Flow With Convolutional Networks. In: *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*.
- Dragotti, P. L. and Gastpar, M. (2009). *Distributed Source Coding: Theory, Algorithms and Applications*. USA: Academic Press, Inc. ISBN: 0123744857.
- Fan, Z., Liu, J., and Wang, Y. (2021). Motion Adaptive Pose Estimation From Compressed Videos. In: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 11719–11728.
- Farneback, G. (2003). Two-Frame Motion Estimation Based on Polynomial Expansion. In: *Proc. of Image Analysis*. Ed. by J. Bigun and T. Gustavsson. Berlin, Heidelberg: Springer Berlin Heidelberg, 363–370.

- Feichtenhofer, C., Pinz, A., and Zisserman, A. (2016). Convolutional two-stream network fusion for video action recognition. In: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 1933–1941.
- Feng, J., Li, S., Li, X., Wu, F., Tian, Q., Yang, M.-H., and Ling, H. (Sept. 2020). TapLab: A Fast Framework for Semantic Video Segmentation Tapping into Compressed-Domain Knowledge. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*.
- Gatys, L. A., Ecker, A. S., and Bethge, M. (2016). Image Style Transfer Using Convolutional Neural Networks. In: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2414–2423.
- Geiger, A., Lenz, P., and Urtasun, R. (June 2012). Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.
- Girod, B., Aaron, A. M., Rane, S., and Rebollo-Monedero, D. (2005). Distributed video coding. *Proceedings of the IEEE* 93(1), 71–83.
- Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. <http://www.deeplearningbook.org>. MIT Press.
- Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014a). *Generative Adversarial Networks*.
- Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014b). *Generative Adversarial Networks*.
- He, K., Zhang, X., Ren, S., and Sun, J. (2015). *Deep Residual Learning for Image Recognition*.
- Hong, H., Hua, X., Zhang, X., and Shi, Y. (Aug. 2016). Multi-frame real image restoration based on double loops with alternative maximum likelihood estimation. *Signal, Image and Video Processing* 10(8), 1489–1495.
- Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. (2017). *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications*.

- Hu, H., Zhou, W., Li, X., Yan, N., and Li, H. (Jan. 2021). MV2Flow: Learning Motion Representation for Fast Compressed Video Action Recognition. *ACM Trans. on Multimedia Computing, Communications, and Applications (TOMM)* 16(3s). ISSN: 1551–6857.
- Huang, G., Liu, Z., Maaten, L. van der, and Weinberger, K. Q. (2016). *Densely Connected Convolutional Networks*.
- Huang, L., Liu, X., Lang, B., Yu, A. W., Wang, Y., and Li, B. (2017). *Orthogonal Weight Normalization: Solution to Optimization over Multiple Dependent Stiefel Manifolds in Deep Neural Networks*.
- Huang, L., Liu, Y., Wang, B., Pan, P., Xu, Y., and Jin, R. (2021). Self-supervised Video Representation Learning by Context and Motion Decoupling. In: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 13886–13895.
- Huynh, C., Tran, A. T., Luu, K., and Hoai, M. (June 2021). Progressive Semantic Segmentation. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 16755–16764.
- Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., and Brox, T. (July 2017). FlowNet 2.0: Evolution of Optical Flow Estimation With Deep Networks. In: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.
- Ioffe, S. and Szegedy, C. (2015). *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*.
- Jacobsen, J.-H., Gemert, J. van, Lou, Z., and Smeulders, A. W. M. (2016). *Structured Receptive Fields in CNNs*.
- Johnson, J., Alahi, A., and Fei-Fei, L. (2016). Perceptual Losses for Real-Time Style Transfer and Super-Resolution. In: *Computer Vision – ECCV 2016*. Springer International Publishing, 694–711.
- Kataoka, Y., Matsubara, T., and Uehara, K. (2016). Image generation using generative adversarial networks and attention mechanism. In: *Proc. of the IEEE/ACIS International Conf. on Computer and Information Science (ICIS)*, 1–6.

- Keshari, R., Vatsa, M., Singh, R., and Noore, A. (2018). *Learning Structure and Strength of CNN Filters for Small Sample Size Training*.
- Khan, N., Shah, J., and Stavness, I. (2018). *Bridgeout: stochastic bridge regularization for deep neural networks*.
- Kingma, D. and Ba, J. (Dec. 2014a). Adam: A Method for Stochastic Optimization. *Proc. of the International Conf. on Learning Representations (ICLR)*.
- Kingma, D. P. and Ba, J. (2014b). *Adam: A Method for Stochastic Optimization*.
- Kodak Image Set (n.d.). <http://www.cs.albany.edu/~xypan/research/snr/Kodak.html>.
- Kodali, N., Abernethy, J. D., Hays, J., and Kira, Z. (2017). How to Train Your DRAGAN. *CoRR* abs/1705.07215.
- Krizhevsky, A., Nair, V., and Hinton, G. (n.d.). CIFAR-10 (Canadian Institute for Advanced Research) ().
- Krizhevsky, A., Sutskever, I., and Hinton, G. E. (May 2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM* 60(6), 84–90.
- Krogh, A. and Hertz, J. A. (1992). A Simple Weight Decay Can Improve Generalization. In: *Advances in Neural Information Processing Systems 4*. Ed. by J. E. Moody, S. J. Hanson, and R. P. Lippmann. Morgan-Kaufmann, 950–957.
- Kupyn, O., Budzan, V., Mykhailych, M., Mishkin, D., and Matas, J. (June 2018). Deblurgan: Blind motion deblurring using conditional adversarial networks. In: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.
- LeCun, Y. and Cortes, C. (2010). MNIST handwritten digit database.
- Ledig, C., Theis, L., Huszar, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., and Shi, W. (2016). *Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network*.
- Ledig, C., Theis, L., Huszar, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., and Shi, W. (2017a). *Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network*.

- Ledig, C., Theis, L., Huszar, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., and Shi, W. (July 2017b). Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. In: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.
- Leung, T. and Malik, J. (2001). *International Journal of Computer Vision* 43(1), 29–44.
- Li, F.-F., Andreetto, M., and Ranzato, M. (2003). Caltech101 Image Dataset.
- Li, H., Wang, X., Liu, W., and Wang, Y. (Oct. 2014). Dictionary learning based image enhancement for rarity detection. In: *2014 12th International Conference on Signal Processing (ICSP)*. IEEE.
- Lim, B., Son, S., Kim, H., Nah, S., and Lee, K. M. (2017a). *Enhanced Deep Residual Networks for Single Image Super-Resolution*.
- Lim, B., Son, S., Kim, H., Nah, S., and Lee, K. M. (July 2017b). Enhanced Deep Residual Networks for Single Image Super-Resolution. In: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition Workshops (CVPRW)*.
- Lim, B., Son, S., Kim, H., Nah, S., and Lee, K. M. (2017c). Enhanced Deep Residual Networks for Single Image Super-Resolution. *CoRR* abs/1707.02921.
- Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. (2017). *Focal Loss for Dense Object Detection*.
- Liu, C., Gao, Q., and Wu, X. (2020). Exaggerated Learning For Clean-And-Sharp Image Restoration. In: *2020 IEEE International Conference on Image Processing (ICIP)*, 673–677.
- Liu, C., Wan, F., Ke, W., Xiao, Z., Yao, Y., Zhang, X., and Ye, Q. (2019a). Orthogonal Decomposition Network for Pixel-Wise Binary Classification. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 6057–6066.
- Liu, P., Zhang, H., Zhang, K., Lin, L., and Zuo, W. (2018). Multi-level Wavelet-CNN for Image Restoration. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 886–88609.

- Liu, P., Lyu, M., King, I., and Xu, J. (June 2019b). SelfFlow: Self-Supervised Learning of Optical Flow. In: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.
- Liu, Y., Yin, W., Chen, Y., Chen, H., and Shen, C. (2021). Generic Perceptual Loss for Modelling Structured Output Dependencies. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'21) (CVPR)*.
- Long, J., Shelhamer, E., and Darrell, T. (June 2015). Fully Convolutional Networks for Semantic Segmentation. In: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.
- Luan, S., Zhang, B., Chen, C., Cao, X., Han, J., and Liu, J. (2017). Gabor Convolutional Networks.
- Lucas, B. D. and Kanade, T. (1981). An Iterative Image Registration Technique with an Application to Stereo Vision. In: *Proc. of the International Joint Conf. on Artificial Intelligence (IJCAI)*. Vancouver, BC, Canada, 674–679.
- Luo, Z.-q., Ma, W.-k., So, A., Ye, Y., and Zhang, S. (May 2010). Semidefinite Relaxation of Quadratic Optimization Problems. *IEEE Signal Processing Magazine* 27(3), 20–34.
- Marçelja, S. (Nov. 1980). Mathematical description of the responses of simple cortical cells*. *J. Opt. Soc. Am.* 70(11), 1297–1300.
- Martin, D., Fowlkes, C., Tal, D., and Malik, J. (July 2001a). A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics. In: *Proc. 8th Int'l Conf. Computer Vision*. Vol. 2, 416–423.
- Martin, D., Fowlkes, C., Tal, D., and Malik, J. (July 2001b). A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics. In: *Proc. 8th Int'l Conf. Computer Vision*. Vol. 2, 416–423.

- Martin, D., Fowlkes, C., Tal, D., and Malik, J. (2001c). A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: *In ICCV*, 416–423.
- Masko, D. and Hensman, P. (2015). The Impact of Imbalanced Training Data for Convolutional Neural Networks. In:
- Mayer, N., Ilg, E., Hausser, P., Fischer, P., Cremers, D., Dosovitskiy, A., and Brox, T. (June 2016). A Large Dataset to Train Convolutional Networks for Disparity, Optical Flow, and Scene Flow Estimation. In: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.
- McCloskey, M. and Cohen, N. J. (1989). Catastrophic interference in connectionist networks: The sequential learning problem. In: *Psychology of learning and motivation*. Vol. 24. Elsevier, 109–165.
- Menze, M. and Geiger, A. (June 2015). Object Scene Flow for Autonomous Vehicles. In: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.
- Mittal, A., Soundararajan, R., and Bovik, A. C. (2013). Making a “Completely Blind” Image Quality Analyzer. *IEEE Signal Processing Letters* 20(3), 209–212.
- Mukherjee, D., Han, J., Bankoski, J., Bultje, R., Grange, A., Koleszar, J., Wilkins, P., and Xu, Y. (2013). A Technical Overview of VP9 – The Latest Open-Source Video Codec. In: *SMPTE 2013 Annual Technical Conference Exhibition*, 1–17.
- Nah, S., Hyun Kim, T., and Mu Lee, K. (July 2017). Deep Multi-Scale Convolutional Neural Network for Dynamic Scene Deblurring. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Naik, S. and Patel, N. (Aug. 2013). Single Image Super Resolution in Spatial and Wavelet Domain. *The International journal of Multimedia & Its Applications* 5(4), 23–32.
- Nakanishi, K. M., Maeda, S.-i., Miyato, T., and Okanohara, D. (2019). Neural Multi-scale Image Compression. In: *Computer Vision – ACCV 2018*. Springer International Publishing, 718–732.

- Nazeri, K., Thasarathan, H., and Ebrahimi, M. (Oct. 2019). Edge-Informed Single Image Super-Resolution. In: *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*. IEEE.
- Niklaus, S. and Liu, F. (June 2018). Context-aware synthesis for video frame interpolation. In: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 1701–1710.
- Ogunfunmi, T. and Narasimha, M. (Apr. 2010). *Principles of speech coding*. Boca Raton, FL: CRC Press.
- Pandey, R. K., Saha, N., Karmakar, S., and Ramakrishnan, A. G. (2018). *MSCE: An edge preserving robust loss function for improving super-resolution algorithms*.
- Ranjan, A. and Black, M. J. (2017). Optical Flow Estimation Using a Spatial Pyramid Network. In: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2720–2729.
- Revaud, J., Weinzaepfel, P., Harchaoui, Z., and Schmid, C. (2015). Epicflow: Edge-preserving interpolation of correspondences for optical flow. In: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 1164–1172.
- Ronneberger, O., Fischer, P., and Brox, T. (2015a). U-Net: Convolutional Networks for Biomedical Image Segmentation. In: *Proc. of International Conf. on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 234–241.
- Ronneberger, O., Fischer, P., and Brox, T. (2015b). U-Net: Convolutional Networks for Biomedical Image Segmentation. In: *Lecture Notes in Computer Science*. Lecture notes in computer science. Cham: Springer International Publishing, 234–241.
- Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning Representations by Back-propagating Errors. *Nature* 323(6088), 533–536.
- Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)* 115(3), 211–252.

Bibliography

- Sajjadi, M. S. M., Schölkopf, B., and Hirsch, M. (2017). EnhanceNet: Single Image Super-Resolution Through Automated Texture Synthesis. In: *2017 IEEE International Conference on Computer Vision (ICCV)*, 4501–4510.
- Sarwar, S. S., Panda, P., and Roy, K. (2017). Gabor Filter Assisted Energy Efficient Fast Learning Convolutional Neural Networks.
- Seif, G. and Androutsos, D. (2018). Edge-Based Loss Function for Single Image Super-Resolution. In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1468–1472.
- Shen, X., Diamond, S., Gu, Y., and Boyd, S. (2016). *Disciplined Convex-Concave Programming*.
- Shocher, A., Cohen, N., and Irani, M. (2018). “zero-shot” super-resolution using deep internal learning. In: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 3118–3126.
- Shou, Z., Lin, X., Kalantidis, Y., Sevilla-Lara, L., Rohrbach, M., Chang, S.-F., and Yan, Z. (June 2019). DMC-Net: Generating Discriminative Motion Cues for Fast Compressed Video Action Recognition. In: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.
- Shu, X. and Wu, X. (2018). Real-Time High-Fidelity Compression for Extremely High Frame Rate Video Cameras. *IEEE Transactions on Computational Imaging* 4(1), 172–180.
- Simonyan, K. and Zisserman, A. (2014). *Very Deep Convolutional Networks for Large-Scale Image Recognition*.
- Sonawane, V. N. and Khobragade, S. V. (2013). Comparative Analysis between A-law & μ -law Companding Technique for PAPR Reduction in OFDM. In:
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research* 15, 1929–1958.

- Sullivan, G. J., Ohm, J.-R., Han, W.-J., and Wiegand, T. (2012). Overview of the High Efficiency Video Coding (HEVC) Standard. *IEEE Trans. on Circuits and Systems for Video Technology (TCSVT)* 22(12), 1649–1668.
- Sun, D., Roth, S., and Black, M. J. (2014). A quantitative analysis of current practices in optical flow estimation and the principles behind them. *International Journal of Computer Vision (IJCV)* 106(2), 115–137.
- Sun, D., Yang, X., Liu, M.-Y., and Kautz, J. (June 2018). PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume. In: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.
- Sun, Y., Wang, X., Liu, Z., Miller, J., Efros, A., and Hardt, M. (2020). Test-time training with self-supervision for generalization under distribution shifts. In: *Proc. of International Conf. on Machine Learning (ICML)*, 9229–9248.
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2014). *Going Deeper with Convolutions*.
- Teed, Z. and Deng, J. (Nov. 2020). RAFT: Recurrent All-Pairs Field Transforms for Optical Flow. In: *Proc. of the European Conf. on Computer Vision*, 402–419.
- Tian, C., Fei, L., Zheng, W., Xu, Y., Zuo, W., and Lin, C.-W. (2019). *Deep Learning on Image Denoising: An overview*.
- Timofte, R., Agustsson, E., Van Gool, L., Yang, M.-H., Zhang, L., Lim, B., et al. (July 2017). NTIRE 2017 Challenge on Single Image Super-Resolution: Methods and Results. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.
- Timofte, R., Gu, S., Wu, J., Van Gool, L., Zhang, L., Yang, M.-H., Haris, M., et al. (June 2018). NTIRE 2018 Challenge on Single Image Super-Resolution: Methods and Results. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.
- Toderici, G., Shi, W., Timofte, R., Theis, L., Balle, J., Agustsson, E., Johnston, N., and Mentzer, F. (2020). *Workshop and Challenge on Learned Image Compression (CLIC 2020)*. CVPR.

- Toderici, G., Vincent, D., Johnston, N., Hwang, S. J., Minnen, D., Shor, J., and Covell, M. (2017). Full Resolution Image Compression with Recurrent Neural Networks. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 5435–5443.
- Tudor, P. (1995). MPEG-2 video compression. *Electronics & Communication Engineering Journal* 7(6), 257–264.
- Wallace, G. (1992). The JPEG still picture compression standard. *IEEE Transactions on Consumer Electronics* 38(1), xviii–xxxiv.
- Wang, D., Shelhamer, E., Liu, S., Olshausen, B., and Darrell, T. (2021). Tent: Fully test-time adaptation by entropy minimization. In: *Proc. of the International Conf. on Learning Representations (ICLR)*.
- Wang, L., Wu, X., and Shi, G. (2012). Binned Progressive Quantization for Compressive Sensing. *IEEE Transactions on Image Processing* 21(6), 2980–2990.
- Wang, Z., Simoncelli, E., and Bovik, A. (n.d.). Multiscale structural similarity for image quality assessment. In: *The Thirty-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*. IEEE.
- Wang, Z. and Bovik, A. (2002). A universal image quality index. *IEEE Signal Processing Letters* 9(3), 81–84.
- Weale, R. A. (June 1983). Vision. A Computational Investigation Into the Human Representation and Processing of Visual Information. David Marr. *The Quarterly Review of Biology* 58(2), 299–299.
- Wiegand, T., Sullivan, G. J., Bjontegaard, G., and Luthra, A. (2003). Overview of the H. 264/AVC video coding standard. *IEEE Trans. on Circuits and Systems for Video Technology (TCSVT)* 13(7), 560–576.
- Witkin, A. P. (1987). SCALE-SPACE FILTERING. In: *Readings in Computer Vision*. Elsevier, 329–332.
- Wu, C.-Y., Zaheer, M., Hu, H., Manmatha, R., Smola, A. J., and Krähenbühl, P. (June 2018). Compressed video action recognition. In: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 6026–6035.

- Wu, R., Guo, C., Hannun, A., and Maaten, L. van der (2021). Fixes That Fail: Self-Defeating Improvements in Machine-Learning Systems. In: *Advances in Neural Information Processing Systems (NeurIPS)*.
- Wu, X., Zhang, X., and Wang, X. (2009). Low Bit-Rate Image Compression via Adaptive Down-Sampling and Constrained Least Squares Upconversion. *IEEE Transactions on Image Processing* 18(3), 552–561.
- Xie, D., Xiong, J., and Pu, S. (2017). *All You Need is Beyond a Good Init: Exploring Better Solution for Training Extremely Deep Convolutional Neural Networks with Orthonormality and Modulation*.
- Xu, J., Ranftl, R., and Koltun, V. (2017). Accurate optical flow via direct cost volume processing. In: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 1289–1297.
- Xue, T., Chen, B., Wu, J., Wei, D., and Freeman, W. T. (2019). Video enhancement with task-oriented flow. *International Journal of Computer Vision (IJCV)* 127(8), 1106–1125.
- Yang, G. and Ramanan, D. (2019). Volumetric Correspondence Networks for Optical Flow. In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 32.
- Yang, T., Ren, P., Xie, X., and Zhang, L. (2021). *GAN Prior Embedded Network for Blind Face Restoration in the Wild*.
- Young, S. I., Girod, B., and Taubman, D. (2020). Fast Optical Flow Extraction From Compressed Video. *IEEE Trans. on Image Processing (TIP)* 29, 6409–6421.
- Yu, J., Fan, Y., Yang, J., Xu, N., Wang, Z., Wang, X., and Huang, T. (2018a). *Wide Activation for Efficient and Accurate Image Super-Resolution*.
- Yu, J., Fan, Y., Yang, J., Xu, N., Wang, Z., Wang, X., and Huang, T. S. (2018b). Wide Activation for Efficient and Accurate Image Super-Resolution. *CoRR* abs/1808.08718.
- Yu, T., Kumar, S., Gupta, A., Levine, S., Hausman, K., and Finn, C. (2020a). Gradient Surgery for Multi-Task Learning. In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 33, 5824–5836.

- Yu, Y., Lee, S., Kim, G., and Song, Y. (2020b). Self-Supervised Learning of Compressed Video Representations. In: *Proc. of the International Conf. on Learning Representations (ICLR)*.
- Zaheer, M., Reddi, S. J., Sachan, D., Kale, S., and Kumar, S. (2018). Adaptive Methods for Nonconvex Optimization. In: *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. NIPS'18. Montréal, Canada: Curran Associates Inc., 9815–9825.
- Zeyde, R., Elad, M., and Protter, M. (2012a). On Single Image Scale-Up Using Sparse Representations. In: *Curves and Surfaces*. Springer Berlin Heidelberg, 711–730.
- Zeyde, R., Elad, M., and Protter, M. (2012b). On Single Image Scale-Up Using Sparse Representations. In: *Curves and Surfaces*. Springer Berlin Heidelberg, 711–730.
- Zhang, B., Wang, L., Wang, Z., Qiao, Y., and Wang, H. (2016). Real-time Action Recognition with Enhanced Motion Vector CNNs. In: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2718–2726.
- Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. (2018a). The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In: *CVPR*.
- Zhang, X., Karaman, S., and Chang, S.-F. (2019). *Detecting and Simulating Artifacts in GAN Fake Images*.
- Zhang, Y., Li, K., Li, K., Wang, L., Zhong, B., and Fu, Y. (2018b). Image Super-Resolution Using Very Deep Residual Channel Attention Networks. In: *ECCV*.
- Zhao, H., Gallo, O., Frosio, I., and Kautz, J. (2015). *Loss Functions for Neural Networks for Image Processing*.
- Zhao, H., Gallo, O., Frosio, I., and Kautz, J. (2017). Loss Functions for Image Restoration With Neural Networks. *IEEE Transactions on Computational Imaging* 3(1), 47–57.
- Zhao, S., Sheng, Y., Dong, Y., Chang, E. I., Xu, Y., et al. (2020). Maskflownet: Asymmetric feature matching with learnable occlusion mask. In: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 6278–6287.
- Zhaoping, L. (May 2014). *Understanding Vision*. Oxford University Press.

Bibliography

- Ziv, J. and Lempel, A. (1977). A universal algorithm for sequential data compression. *IEEE Transactions on Information Theory* 23(3), 337–343.
- Zoph, B., Vasudevan, V., Shlens, J., and Le, Q. V. (2017). *Learning Transferable Architectures for Scalable Image Recognition*.