Towards Learning Representations in Visual Computing Tasks

by

Parag Shridhar Chandakkar

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved October 2017 by the
Graduate Supervisory Committee:

Baoxin Li, Chair
Pavan Turaga
Hasan Davulcu
Yezhou Yang

ARIZONA STATE UNIVERSITY

December 2017

ABSTRACT

The performance of most of the visual computing tasks depends on the quality of the features extracted from the raw data. Insightful feature representation increases the performance of many learning algorithms by exposing the underlying explanatory factors of the output for the unobserved input (Bengio *et al.*, 2013). A good representation should also handle anomalies in the data such as missing samples and noisy input caused by the undesired, external factors of variation. It should also reduce the data redundancy. Over the years, many feature extraction processes have been invented to produce good representations of raw images and videos.

The feature extraction processes can be categorized into three groups. The first group contains processes that are hand-crafted for a specific task. Hand-engineering features requires the knowledge of domain experts and manual labor. However, the feature extraction process is interpretable and explainable. Next group contains the latent-feature extraction processes. While the original feature lies in a high-dimensional space, the relevant factors for a task often lie on a lower dimensional manifold. The latent-feature extraction employs hidden variables to expose the underlying data properties that cannot be directly measured from the input. Latent features seek a specific structure such as sparsity or low-rank into the derived representation through sophisticated optimization techniques. The last category is that of deep features. These are obtained by passing raw input data with minimal pre-processing through a deep network. Its parameters are computed by iteratively minimizing a task-based loss.

In this dissertation, I present four pieces of work where I create and learn suitable data representations. The first task employs hand-crafted features to perform clinically-relevant retrieval of diabetic retinopathy images. The second task uses latent features to perform content-adaptive image enhancement. The third task ranks a pair of images based on their aestheticism. The goal of the last task is to capture localized image artifacts in small datasets

with patch-level labels. For both these tasks, I propose novel deep architectures and show significant improvement over the previous state-of-art approaches. A suitable combination of feature representations augmented with an appropriate learning approach can increase performance for most visual computing tasks.

हा शोध प्रबंध माझे आई-बाबा, माझा भाऊ, योगेश, आणि माझी बायको, जिज्ञासा, यांना समर्पित.
(This dissertation is dedicated to my parents, my brother, Yogesh, and my wife, Jidnyasa.)

ACKNOWLEDGMENTS

I am grateful to my adviser, professor Baoxin Li, whose guidance, understanding and expertise made my Ph.D. an enjoyable experience. I got opportunities to work across a broad spectrum of technologies while working with him. I have always been amazed at how well he picks up a topic and can offer his valuable insight. My association with him dates back to 2011 when I joined his lab as a Masters Thesis student. I remember his patient attitude while explaining me the nuances of research and listen to my (sometimes misinformed) opinions. I have always admired his ability to provide equal attention to all his students. He assigned me challenging tasks which need familiarity to a broad set of technologies. This helped me build my confidence and it elevated my profile. Till date, I have learned something new every time I have had a discussion with him. It was a pleasure working under his supervision and I hope we can keep our association intact.

I am thankful to my family and my wife, Jidnyasa, who showed immense understanding and supported me at every stage during my Ph.D. I would like to mention my friends, Aditi, Charan, Himabindu, Tejas, Madhurima, Yash, Vimala, Vijetha, Jashmi, Qiongjie, Archana, Neel and Aditi. Without them, I could have finished my Ph.D. in four years, but those years would not be worth remembering. I would also like to thank my labmates Qiang, Ragav, Lin, Yilin, Yuzhen, Yikang, Xu, Peng, Kevin. It was a pleasure working with them.

I also take this opportunity to thank the rest of my committee members, Dr. Pavan Turaga, Dr. Yezhou Yang and Dr. Hasan Davulcu for being a part of the dissertation committee and providing constructive remarks that helped me improve my dissertation.

TABLE OF CONTENTS

CHAPTER Page

CHAPTER Page

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

Figure Page

Chapter 1

INTRODUCTION

The success of many machine learning algorithms depends on having better input representations that expose the underlying explanatory factors of the output for the observed input (Bengio *et al.*, 2013). An effective data representation should reduce the data redundancy and adapt to the undesired, external factors of variation introduced by sensor noise, labeling errors, missing samples, etc. All these properties help reduce the complexity of the real-world data which is often high-dimensional. However, according to the manifold hypothesis, the real-world data are expected to lie in a lower-dimensional manifold that embeds the high-dimensional real-world input (Bengio *et al.*, 2013). This assumption serves us well in case of images. Real-world images lie in an extremely high-dimensional space. For example, a two-dimensional space of size 32×32 in which each point is either 0 or 1, can produce 2^{1024} images. However, the number of images recognizable to humans will only be a small fraction of this. There must be an underlying low-dimensional manifold that embeds the space of the original 1024D manifold. The challenge for representation learning technique is to find this manifold while achieving high performance on the desired task. In the following sections, I explain the various feature representation hierarchies broadly categorized into three groups.

1.1 Hand-crafted features

In general, hand-crafted features refer to fundamental features such as image gradients as well as sophisticated, computationally non-trivial features such as the histogram of oriented gradients (Dalal and Triggs, 2005). These are designed by domain experts who have prior knowledge about the data properties and the underlying data distribution. Hand-engineering

features for each task requires a lot of manual labor. However, it is easy to integrate the human knowledge of the real-world and of that specific task into the feature design process (Paladugu *et al.*, 2013; Chandakkar *et al.*, 2014, 2015c), making it possible to obtain good, interpretable results for the said task along with an explainable feature extraction process. These properties are desirable when the feature extraction process is used for high-risk tasks such as computer-aided diagnosis, automated trading, etc. However, note that it is not entirely correct to call all traditional features as being hand-crafted since some of them are general-purpose features with little task-specific tuning (such as outputs of simple gradient filters).

1.2 Latent-feature representation

The raw data, especially images, lie in a very high-dimensional space. Most times, the relevant factors for a task are contained in a lower-dimensional space that is hidden (Bengio *et al.*, 2013). Latent-feature extraction processes discover these low-dimensional spaces by employing hidden variables. These representations measure the underlying properties of the data that cannot be readily measured. These processes usually seek a specific structure into the features such as sparsity, decorrelation of reduced dimensions, low-rank, etc. The sparsity and the low-dimensionality are often encouraged as many real-world signals naturally have sparse-representations in some fixed, appropriate bases (e.g., Fourier). These signals may be embedded in a low-dimensional manifold (Wright *et al.*, 2010). However, discovering these latent representations is a complicated optimization process often requiring extensive reformulation of the original task objective and advanced optimization techniques such as alternating minimization.

1.3 Deep representations

Deep representations are obtained by passing raw input data with minimal pre-processing through a neural network consisting of a stack of convolutional layers that function as a feature extractor and fully-connected layers that work as a classifier. As we traverse through all the network layers, we obtain a different data representation that abstracts a specific semantic concept at that layer. The captured concepts become progressively complex and of semantically higher-level as we move deeper into the stack of network layers. For example, earlier layers may encode simple concepts such as image edges, color differences, etc. The higher layers may capture properties specific to each object such as object contour and shape. The networks are trained iteratively by minimizing a task-specific loss that alters the parameters/weights for all of those layers. Recently, deep features have been found highly effective in many visual computing tasks. Their most attractive property is their ability to learn from a raw input with minimal pre-processing. Moreover, representations obtained from generic feature extractors provide a reasonable performance on many tasks, alleviating the need for domain experts at every stage. However, learning deep representations needs substantial computational resources and data collections. They also require extensive storage making the processing suitable only on computing clusters.

This dissertation focuses on creating and learning different data representations for a set of visual computing tasks. I present four pieces of work that employ feature representations at all the three hierarchies.

Clinically-relevant Diabetic Retinopathy Image Retrieval: Diabetic retinopathy (DR) is a consequence of diabetes and is the leading cause of blindness among working adults (Centers for Disease Control and prevention and others, 2011). Regular screening is critical to early detection and treatment of DR. Computer-aided diagnosis has the potential of improving the practice of DR screening or diagnosis (Quellec *et al.*, 2011). To this end,

there is a need for an automated and unsupervised approach to retrieving clinically-relevant images from a set of previously-diagnosed fundus camera images. Such computer-aided procedures will improve the efficiency of screening and diagnosis of DR. Considering the unique visual properties of DR images; I developed a feature space consisting of a modified color correlogram appended with statistics of steerable Gaussian filter responses selected by the fast radial symmetric transform points. Considering that many DR lesions are often localized, I propose a multi-class multiple-instance retrieval framework. Extensive experiments with real DR images collected from five different data-sets demonstrate that the proposed approach outperforms existing methods (Venkatesan *et al.*, 2012; Chandakkar *et al.*, 2013, 2017b).

Content-adaptive Image Enhancement: Social networking on mobile devices has become a commonplace of everyday life. Also, the photo-capturing process has become trivial due to the advances in mobile imaging. People are taking a lot of photos everyday and they want them to be visually-attractive. This has given rise to automated, one-touch enhancement tools. However, the inability of those devices to provide personalized and content-adaptive enhancement has paved way for machine-learned methods to do the same (Bychkovsky *et al.*, 2011; Yan *et al.*, 2014a; Chandakkar *et al.*, 2015a; Kapoor *et al.*, 2014; Hwang *et al.*, 2012; Kang *et al.*, 2010; Kaufman *et al.*, 2012; Joshi *et al.*, 2010; Yan *et al.*, 2014c). The existing typical machine-learned methods heuristically predict the enhancement parameters of a new image from a set of neighboring image parameters. Performing k-nearest neighbor search on the training set makes the parameter prediction sub-optimal and computationally expensive at test time. The cardinality of the set of possible enhancement parameters is enormous, but only a fraction of those parameters produce "enhanced" images. This suggests there lies a low-dimensional latent space for enhancement parameters. I present a novel approach to predicting the enhancement parameters given a new image using only its features, without using any training images. I propose to model the interaction between

the image features and its corresponding enhancement parameters using latent variables. This approach outperforms heuristic approaches as well as recent approaches in structured prediction on synthetic and on real-world data of image enhancement (Chandakkar and Li, 2016). Motivated by this work, I propose an extension that uses the Gaussian-process (GP) based joint regression and ranking for a unified image enhancement pipeline (Chandakkar and Li, 2017b). Unlike the earlier approach, this GP-based approach traverses parameter space, performs regression to find a set of possible enhancement parameters and ranks them using a unified pipeline. Comparative evaluation using the ground-truth based on the MIT-Adobe FiveK dataset (Bychkovsky *et al.*, 2011) and subjective tests on an additional data-set were used to demonstrate the effectiveness of the proposed approach.

Relative aesthetics estimation using deep features: Computational visual aesthetics has recently become an active research area. Existing state-of-art methods formulate this as a binary classification task where a given image is predicted to be aesthetic or not (Dhar *et al.*, 2011; Nishiyama *et al.*, 2011; Lu *et al.*, 2014, 2015). In many applications such as visual search and image enhancement, it is more important to rank images based on their aesthetic quality instead of merely categorizing them into two classes. Furthermore, in such applications, it may be possible that all images belong to the same category. Hence determining the aesthetic ranking of the images is more appropriate. To this end, I formulate a novel problem of ranking images based on their aesthetic quality. I construct a new dataset of image pairs with relative labels by carefully selecting images from the popular AVA dataset. Unlike in aesthetics classification, there is no single threshold which would determine the ranking order of the images across our entire dataset. I propose a deep neural network based approach that is trained on image pairs by incorporating principles from relative learning (Chandakkar *et al.*, 2016). Results show that such relative training procedure allows our network to rank the images with a higher accuracy than a state-of-art network trained on the same set of images using binary labels.

No-reference image quality estimation using hyper-image representation: Images get distorted due to defects in acquisition devices, transmission-based errors, etc. The task of image quality assessment (IQA) requires an automated method to estimate the visual quality of an image. Conventional and simple error metrics such as RMSE/PSNR cannot capture the correlation between image appearance and human-perception of the image quality (Wang *et al.*, 2004). Two variants of this problem exist - full-reference IQA and no-reference IQA (NR-IQA). Full-reference IQA task gives access to an original image and its distorted counterpart. The distorted image is assigned a quality score by considering the original image as the reference. Few representative approaches that try to solve this problem are SSIM (Wang *et al.*, 2004), MSSIM (Wang *et al.*, 2003), FSIM (Zhang *et al.*, 2011a), VSI (Zhang *et al.*, 2014) etc. However, in real-world scenarios, one may not have a perfect, non-distorted image available for comparison. Thus NR-IQA variant was proposed. In NR-IQA, a single image needs to be assigned a quality score with respect to a non-distorted, *unobserved* version of that image. This score must correlate well with human perception of image quality. While creating ground-truth for this problem, a constant value is associated with a non-distorted image. This value serves as a reference on the quality score scale.

This problem involves developing a discriminative feature space to different kinds and degrees of distortions. Such setting is more suitable for learning schemes, which is reflected by the fact that most of the approaches tackling this problem belong to the learning paradigm. Few of the representative approaches include BRISQUE (Mittal *et al.*, 2012), CORNIA (Ye *et al.*, 2012, 2013), DIIVINE (Moorthy and Bovik, 2011), BLIINDS (Saad *et al.*, 2012), CBIQ (Mittal *et al.*, 2013), LBIQ (Tang *et al.*, 2011) and the current convolutional neural network (CNN)-based state-of-art (Kang *et al.*, 2014).

The distortions could have a non-uniform distribution over the entire image. Also, the observed effect of many distortions depends on the image texture and the saliency of the distorted region. Learning-based approaches that utilize hand-crafted features fail to account

for all such scenarios. It results in a reduced correlation between the predicted quality scores and the ground-truth scores. To combat this, I propose a novel CNN-based approach containing two network stages. The first stage is trained on image patches. The second stage is trained on hyper-image representations that are derived from the last layer features of the first stage. The proposed approach works well in case of non-uniform noise distributions. It relaxes the requirement that all image regions should equally contribute to the quality score prediction.

In the upcoming chapters, I will dive into the details of each task, the related literature, the proposed approach and the experiments.

Chapter 2

CLINICALLY-RELEVANT RETRIEVAL OF DIABETIC RETINOPATHY IMAGES

2.1 Introduction

Diabetic retinopathy (DR) is a consequence of diabetes, and it is one manifestation of a systemic disease which can affect a myriad of organ systems. It can cause vision loss if not treated early (Centers for Disease Control and prevention and others, 2011). Studies showed that 381.8 million patients worldwide were diagnosed with diabetes in 2013 (Guariguata *et al.*, 2014). In 2010, 10.9 million US residents, aged 65 or older were suffering from diabetes (Centers for Disease Control and prevention and others, 2011). According to the recent estimates, the number of diabetic patients will rise to 591.9 million by 2035 which indicates a 55% increase in the number of adults with diabetes (Guariguata *et al.*, 2014). Recent reports show that close to 25,000 people who have diabetes turn blind every year in the US due to DR (Abràmoff *et al.*, 2008). It is estimated that DR is the cause for 5% of the world's blind population (Salomão *et al.*, 2009). The risk of vision loss due to DR can be significantly reduced by early screening and treatment (Garg and Davis, 2009; Zhang *et al.*, 2010). Increasing population, cost, limitations of health-care facilities, lack of enough providers in densely populated areas, lack of awareness on patients' side and other factors are constraints for regular screening of every diabetic patient. Thus to identify and treat DR in its early stages requires a new perspective on the problem.

Two important stages of DR are non-proliferative diabetic retinopathy (NPDR) and proliferative diabetic retinopathy (PDR). Microaneurysm (MA) is a sign of DR. MAs rarely cause notable harm, but their presence indicates an underlying systematic disorder. Detailed eye examinations can usually detect MAs. Neovascularization (NV) is a proliferation of

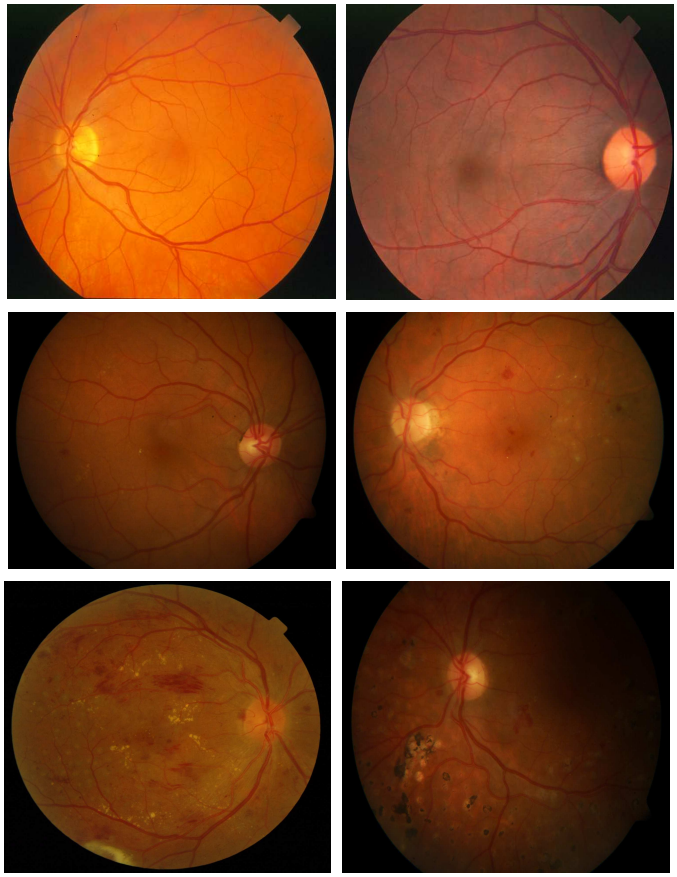


Figure 2.1. Fundus Image of Eyes: Normal (Top Row), NPDR (Middle Row) and PDR (Bottom Row)

functional blood vessels and is a symptom of PDR. Other complications such as vitreous hemorrhages (extravasation of blood around the vitreous body) and/or tractional retinal detachment follow. Timely and effective DR treatment to patients demands reliable detection. Assistance in the form of effective computer-aided technologies may help improve the sensitivity, consistency, and efficiency of DR severity detection (Li and Li, 2013). Recent advancements in computer-aided diagnosis and medical imaging have propelled the development of automated image analysis techniques to solve the DR problem (Li and Li, 2013). It was observed that a content-based image retrieval (CBIR) approach could help new and experienced ophthalmologists (Quellec *et al.*, 2011) in a diagnosis. For example, a CBIR

system can provide them with both standard reference images and previously-diagnosed images containing similar lesions, so that a more precise grading may be achieved.

Conventional CBIR techniques may not be helpful when directly applied on DR images. Fig. 2.1 shows typical fundus images diagnosed with PDR and NPDR against normal ones. Symptoms of NPDR include yellow, waxy exudates. It is noteworthy that very little difference can be observed between the normal images and the NPDR images. Most parts of an affected image would appear to be no different from a normal images as the lesions are localized. A global representation of the image, which is often used in conventional CBIR, may not capture its discriminative characteristics. Properly-designed feature extractors evaluated on small regions would be needed to capture lesion-defining information, although such regions need to be determined. Therefore, to achieve the goal of retrieving DR images that are relevant to a given image, we would need a novel retrieval framework that can take into consideration unknown, localized pathologies. Further, new features need to be designed so that lesion-specific characteristics of an image may be captured.

In this chapter, I propose a multi-class multiple-instance clinically-relevant retrieval framework called *MIRank-KNN* that addresses the problem of localized pathologies in unknown regions (Chandakkar *et al.*, 2013, 2017b). The proposed approach is based on the observation that both color and gradient features occurring in small, localized regions characterize the lesions of interest. Thus for feature extraction, I propose to use spectrally-tuned color-correlogram (CC) (Venkatesan *et al.*, 2012) and statistics of steerable Gaussian filter (SGF) response (Freeman and Adelson, 1991) of the points selected by fast radial symmetric transform (FRST) (Loy and Zelinsky, 2003). This approach gets its clinical relevance from the multi-instance retrieval framework - MIRank-KNN (Chandakkar *et al.*, 2013, 2017b) - and the feature design. For example, MIRank-KNN allows the retrieval of images with similar localization and number of lesions. The features determine the type of lesions to be retrieved. To facilitate the evaluation of the proposed method, the database and

the source code used to carry out most of the experiments in this chapter are posted on the author's web-page [1] .

2.2 Related Work

Several CBIR systems were previously used in the areas of dermatology, radiology, pathology, and ophthalmology. Research groups have used CBIR for computed tomography, magnetic resonance imaging, positron emission tomography, and retinal images (Quellec *et al.*, 2011; Cai *et al.*, 2000; Chaum *et al.*, 2008; Gupta *et al.*, 1996; Chandakkar *et al.*, 2013, 2017b; Chen *et al.*, 2008; Deepak *et al.*, 2010; Quellec *et al.*, 2012b, 2010; Chu *et al.*, 1994; Kelly *et al.*, 1995; Korn *et al.*, 1998; Lamard *et al.*, 2007). In the field of positron emission tomography, physiological kinetic features were used to build a CBIR-based system (Cai *et al.*, 2000). The hierarchical, spatiotemporal and evolutionary semantics of neural images were captured by Chu et al. to develop a semantic model for CBIR purpose (Chu *et al.*, 1994). Textures and histograms of pathologies were combined in a signature, which was formed on a per-image basis, in a system developed by Kelly et al. (Kelly *et al.*, 1995). This system made use of query-by-example techniques to retrieve images. Korn et al. used fast query using nearest neighbor search to retrieve medical tumor images (Korn *et al.*, 1998).

Two of the significant works in previous CBIR investigations on ophthalmology include the Structured Analysis of the Retina (STARE) project and the CBIR system developed in 1996 (Gupta *et al.*, 1996; Goldbaum *et al.*, 1989). STARE was developed for performing automatic diagnosis of images, annotation of image contents and searching for similar images. STARE used basic image features to define the similarity metric. Recent work by Quellec et al. illustrates the importance of CBIR in DR diagnosis (Quellec *et al.*, 2011). It provides statistical support for the claim that CBIR systems will assist experienced as well as relatively inexperienced ophthalmologists in DR diagnosis. A supervised learning approach

[1] `www.public.asu.edu/~bli24/DR-System-and-Data.html`

that employed features extracted on segmented lesions and macula was used for CBIR of DR images (Chaum *et al.*, 2008). A classifier is trained for detecting lesions with the help of ground-truth data of manually segmented lesions. Similarly, some supervised learning approaches use adaptive variants of wavelet transform (Quellec *et al.*, 2011, 2010, 2012b). The DR image retrieval problem was broken into two steps, namely, image background learning and feature extraction (Deepak *et al.*, 2010). Background of a normal (unaffected) DR image was first learned, and then nearest-neighbor retrieval was performed with the help of intensity and texture features. Most of the above state-of-art DR CBIR frameworks rely on a training stage followed by a nearest-neighbor retrieval technique to give them accurate results.

Lesion detection is handled as a separate problem by many since its an essential step in DR image classification and retrieval. Many solutions have emerged for detecting retinal landmarks such as MAs, NV, and hemorrhages from developments in the area of image processing. A bag-of-words scheme was first employed to train individual lesion detectors and then a high-level classifier was used to determine if patients met the criteria for referral-warranted diabetic retinopathy (Pires *et al.*, 2013). Interest point detection and the visual dictionary was also used (Rocha *et al.*, 2012) for individual DR lesion detection. Multiple-instance-learning was used to find relevant patterns in a DR image after being trained on a database consisting of relevant and irrelevant images (Quellec *et al.*, 2012a). Active learning has been employed to select the most informative examples for labeling, thus reducing the load on human experts and obtaining the optimal classification accuracy at the same time (Sánchez *et al.*, 2010). Recent machine learning frameworks, such as multi-class multiple instance learning frameworks has shown promising results with images without lesion-level labels (Xu and Li, 2008; Chandakkar, 2012; Chandakkar *et al.*, 2012; Venkatesan *et al.*, 2012; Chandakkar *et al.*, 2013, 2017b; Venkatesan *et al.*, 2015). An encouraging fact about all these pioneering works is that all of them suggest that the clinicians will be benefited

from CBIR systems or the introduction of advanced vision-based approaches has improved the underlying systems.

Color and textures are proven to be competitive features in spite of their simplicity (Deselaers *et al.*, 2008). Color correlograms tried to model the color correlation in an image (Huang *et al.*, 1997; Li, 2007). Correlograms are considered better than color histograms and most other color features (Huang *et al.*, 1997). They incorporate spatial correlation of image pixels that gives them an edge over histogram features. They describe the global correlation of local spatial correlation of colors. Correlograms are a strong representation of texture with considerably small dimensionality. Gabor features (Manjunath and Ma, 1996), and the histogram of neighborhood mean moments (HNM) (Chen *et al.*, 2008) are popular alternatives for describing texture and color of an image respectively. HNM and Gabor features are widely used approaches in medical image retrieval, but both of them may not produce discriminative features for DR images due to their unique color spectrum. SIFT keypoint detection and description have been shown to be effective for retrieval of radiograph images (Deselaers *et al.*, 2008).

Though there is a significant amount of progress in the area of CBIR of ophthalmological images, the lack of a widely accepted ophthalmological system remains. It shows that the central problems are still unsolved. Some of the shortcomings of the mentioned DR-CBIR retrieval systems are as follows. Most of the systems require a training stage and thus labeled data. For example, manually segmenting lesions involves expertise, a lot of time and labor. Parameters tuning may not be consistent across different types of images and data-sets. These systems either use local features or extract features only on the detected lesions (usually through another classifier) to characterize localized lesions. Though local features are good at characterizing small regions, features of a small but relevant region may get suppressed when features from all the regions are bundled into a long one-dimensional vector. Such a vector is necessary if a nearest-neighbor retrieval scheme is to be used.

Another option is to use multiple-instance learning, but that requires a training stage too. The proposed approach is motivated by the need to overcome these shortcomings. To this end, my contributions are as follows.

1. I develop a completely unsupervised DR image retrieval system. I use the labeling information from the public DR data-sets only to calculate retrieval metrics.

2. I develop a new indexing and retrieval framework, which considers multiple regions of an image simultaneously. It outperforms the nearest-neighbor and the Citation-KNN retrieval scheme by a large margin, even when state-of-art local features are used.

3. I develop a spectrally-tuned color quantization scheme aimed towards DR images. I show its superiority over the original correlogram feature through entropy analysis and experiments.

4. The proposed approach works across five data-sets, and its performance varies only slightly over a wide range of parameters.

5. Through my study on DR images and extensive experiments, I propose a unique combination of color and texture features. The proposed features outperform several other state-of-art features used in natural and medical image retrieval.

A detailed study of emerging trends in automated analysis of DR images and a discussion on the need for better clinically-relevant CBIR systems by Li et al. shows the space and scope for this research effort (Li and Li, 2013).

2.3 Proposed Approach

The proposed retrieval approach can be broken up and studied in two parts.

1. The feature extraction process consisting of color and texture features.

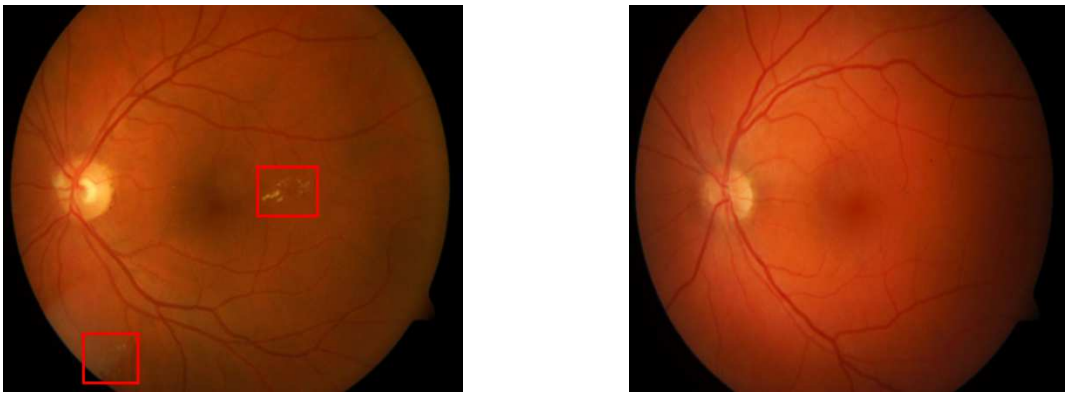


Figure 2.2. Visualizing the Necessity of Multiple-instance Framework. NPDR Image (on Left), Instances Marked in Red Are the Lesions. Normal Image (on Right).

2. MIRank-KNN - The multiple-instance retrieval framework.

2.3.1 Feature Extraction

I consider a holistic representation of a DR image as opposed to a cascade representation that usually first identifies individual lesions and then forms the entire feature (Pires *et al.*, 2013; Rocha *et al.*, 2012). Clinicians also work with a global representation of an image and perform a diagnosis only after considering all the lesions. I use color and texture properties of an image which will be detailed in this section.

Color is a distinguishing characteristic for DR images. However, accurate modeling of the color spectrum of DR images can be challenging due to the variation in lighting among the images of the same class. Fig. 2.1 shows some images with varying lighting conditions. In spite of belonging to the same class, the images in the first and last row have a significant difference in their color spectrum. On the other hand, images belonging to different classes may have subtle differences in their color spectrum, and often those differences are localized. Fig. 2.2 illustrates this perfectly.

The color feature should be invariant to light color change and shift. It should encode

local correlation of colors to model small and localized differences in the color spectrum of two images. Formally, light color change and shift is represented as,

$$\begin{pmatrix} R' \\ G' \\ B' \end{pmatrix} = \begin{pmatrix} cR & 0 & 0 \\ 0 & cG & 0 \\ 0 & 0 & cB \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} + \begin{pmatrix} sR \\ sG \\ sB \end{pmatrix} \tag{2.1}$$

where R, G, B and R', G', B' are the current and modified pixel value respectively. cR, cG, cB and sR, sG, sB are the factors contributing to the lighting change and shift respectively. The color feature is called invariant if it produces identical feature vectors for both the pixel sets - (R, G, B) and (R', G', B').

I use color correlogram to represent DR images as it encodes global distribution of local spatial correlation of colors. I modify the correlogram feature to be invariant to the lighting changes, shifts and most importantly, the unique color spectrum of DR images. They have an almost always saturated red channel as observed in Fig. 2.1. Additionally, DR images of different classes may have similar histograms, making it difficult for traditional color-based features to produce discriminative feature vectors. Effective retrieval of DR images, therefore, demands spectrally-tuned color features. The following subsection describes the details.

Spectrally-tuned Color Correlogram

Color correlogram (CC) was proposed as an effective color feature which encodes global distribution of local spatial correlation of colors (Huang *et al.*, 1997). The CC of an image is a table indexed by color pairs such that the k^{th} entry for the color pair (i, j) gives the probability of finding a pixel of color j at a distance k from color i. I quantize the image into $(m =)$ 16 colors and use $k = 1$. I calculate the color correlation of the center pixel with all other pixels in a 3×3 block. A 2D histogram is then defined over entire image for

$i, j \in [m]$ as,

$$h_{c_i,c_j}(I) = \Pr_{p_1 \in c_i(p_1), p_2 \in I} [p_2 \in c_j(p_2) \mid d(p_1, p_2) = 1]. \tag{2.2}$$

This gives the probability that a pixel belonging to quantized color c_1 has another pixel of color c_2 at a unit distance, where $c_1, c_2 \in [m]$. This models the global distribution of local correlation of colors. The dimensionality of color correlogram is $O(m^2k)$. I quantize the image into 16 colors which gives us a feature vector of 256 dimensions.

The choice of the quantization scheme is crucial to the performance of the CC features. A popular quantization scheme proposed by Li et al. was designed based on human vision and the stimulus reactions of human vision to various colors (Li, 2007). From Fig. 2.1, it can be easily observed that the spectrum of DR images is entirely different than that of natural images. Therefore, a new quantization scheme has to be designed which is spectrally-tuned towards DR images.

I proposed a spectrally-tuned quantization scheme for the CC features tuned towards DR images (Chandakkar *et al.*, 2017b) that is robust to light color change and shift. I create a transformed color space where all three channels have zero mean and unit variance. Let the new color space be represented by (R^T,G^T,B^T).

$$\begin{pmatrix} R^T \\ G^T \\ B^T \end{pmatrix} = \begin{pmatrix} \frac{R-\mu_R}{\sigma_R} \\ \frac{G-\mu_G}{\sigma_G} \\ \frac{B-\mu_B}{\sigma_B} \end{pmatrix} \tag{2.3}$$

The transformed color space is invariant against arbitrary light color changes and shifts due to normalization of all channels. Its invariance properties have also been analyzed (Van De Sande *et al.*, 2010). This transformed color space will henceforth be used for all operations. The unique shades (i.e., $< R^T, G^T, B^T >$ triplets) in all the images are now extracted and arranged in $M \times 3$ matrix. K-means clustering is performed on the matrix to

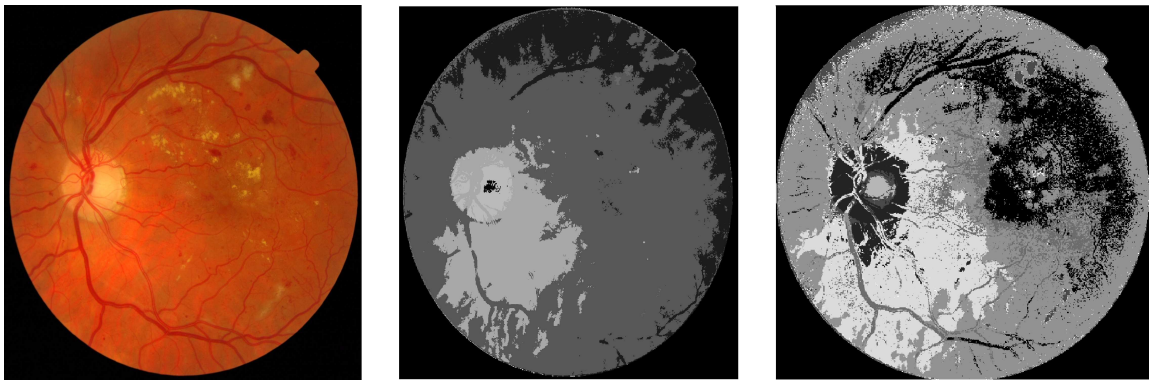


Figure 2.3. Quantization of a DR Image Using AutoCC (Li, 2007) (Middle) and the Proposed Approach (Right).

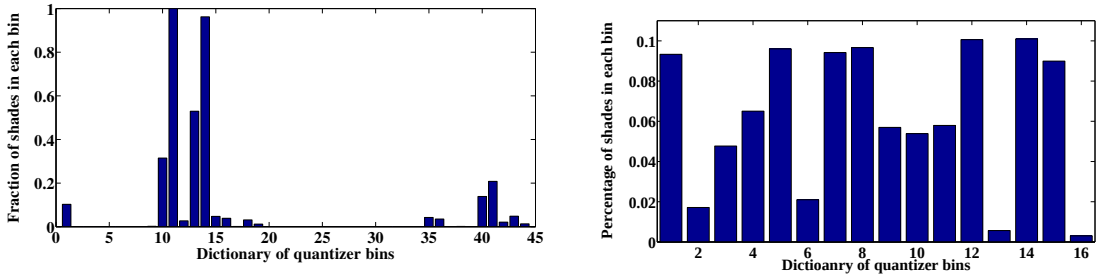


Figure 2.4. Histogram of Quantized Shades for Li's and the Proposed Quantizers Respectively.

generate the spectrally-tuned quantization bins (i.e., centroids). Once the bins are obtained, the CC features are calculated for all possible color pairs (i, j) where $i, j \in [1, \ldots, 16]$. It is empirically observed that modeling the correlation of all the possible pairs gives better retrieval results. However, changing the number of bins affects the performance only by a small amount. The evaluation of the spectrally-tuned CC, as well as its comparison with original CC, is given in section 2.5.

While the spectral-tuning of the quantizer attempts to maximize the entropy by keeping

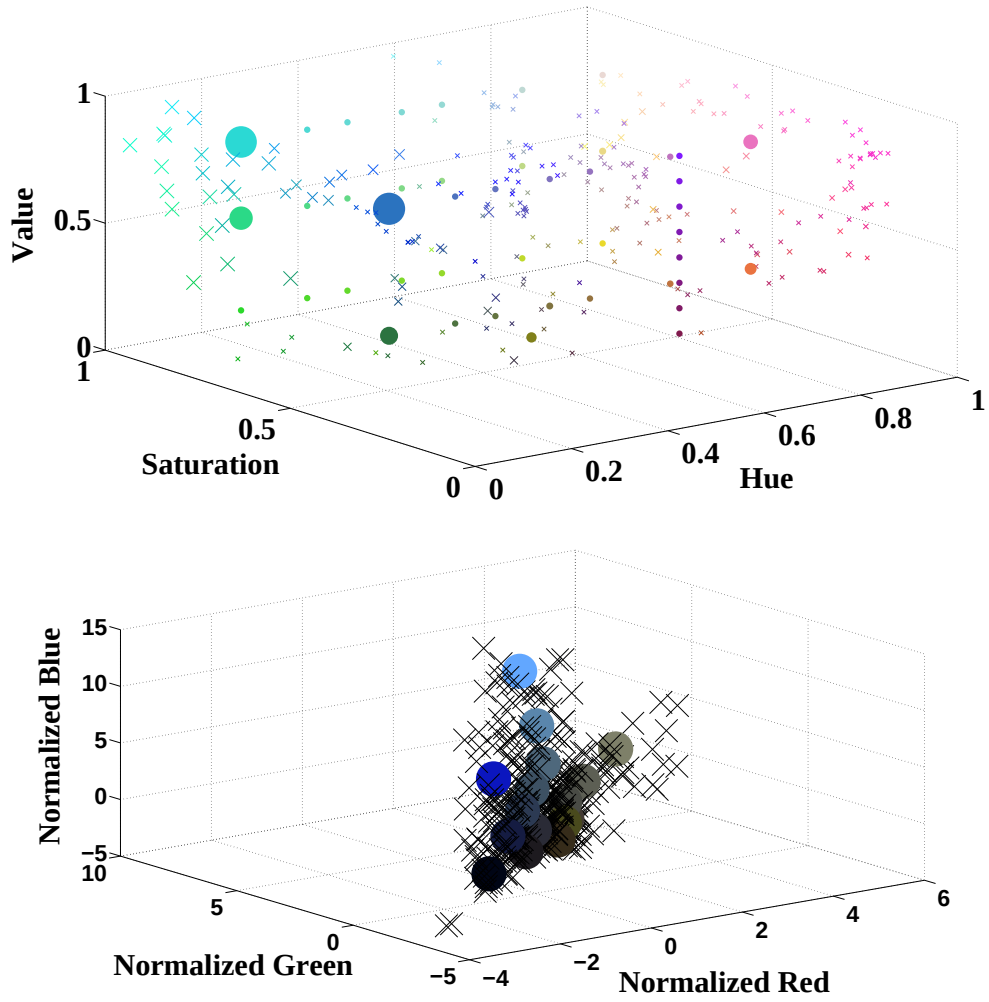


Figure 2.5. 3-D Visualization of Li's Quantization and the Spectrally-tuned Quantization Schemes for DR Image Color Space. The Centroids Are Indicated by Spheres and Points Associated with Each Centroid Are Shown with Cross Marks in Appropriate Colors (Please Zoom in for Better Viewing).

the density of points (shades) associated with each centroid (quantized color bin) uniform, the one proposed by Li et al., has a highly varying density of points associated with each centroid (Li, 2007). A 3D visualization of both the quantization scheme is shown in Fig. 2.5. While the points get non-uniformly distributed by using Li's quantization approach, I obtain a reasonably well-distributed set of points from my approach. The effect clustering has on

this can be noticed better in Fig. 2.4. The spectrally-tuned CC makes better utilization of bandwidth provided by the 16 color bins than the other quantization scheme where most of the shades fall under 4 out of 45 bins, thereby heavily affecting the ability of the CC feature to encode the spatial and global correlation of colors in DR images.

Entropy Analysis: I provide a measure of the amount of information encoded in both approaches using the given number of bins. I further show the superiority of the proposed approach using the developed entropy measure. The entropy is a measure of randomness in the data which in turn corresponds to the amount of information present in it. Entropy of a random variable X consisting of values $(x_1, x_2, \ldots, x_n)$ is given by

$$H(X) = -\sum_i P(x_i) log_b P(x_i) \tag{2.4}$$

$$\text{where } P(x_i) = \frac{\text{\# shades falling under bin } i}{\text{Total \# of shades in the database}}.$$

I use logarithm base 2 and define $0 * log0 = 0$. In Fig. 2.4, the distribution on the left is denoted by D_1 and the one on the right is denoted by D_2. The entropies of both distributions are $H(D_1) = 2.9201$ and $H(D_2) = 3.7222$ bits. Though the proposed quantization scheme provides an increase of 27.47%, both the values cannot be compared due to different-sized distributions (16 vs. 44 bins). Various methods such as scaled entropy, normalized entropy, sliding window approach have been proposed to allow a fair comparison between different-sized distributions (Liu *et al.*, 2008; Heikinheimo *et al.*, 2007). Instead of using a normalizing/scaling approach, I compare the two distributions by using an indirect method which uses a reference distribution possessing maximum entropy with the given number of bins. Uniform distribution is chosen as reference since it provides the maximum entropy among all discrete distributions supported on a finite set $(x_1, x_2, \ldots, x_n)$ (Park and Bera, 2009). I measure the percentage difference in the entropy values of the original and its corresponding uniform distribution. I define dE_1 as the percentage difference between D_1 and its corresponding uniform distribution, denoted by UD_1. dE_2 and UD_2 is similarly

defined. By performing entropy calculations, values of UD_1 and UD_2 are found to be 5.4594 and 4 bits respectively. By comparing dE_1 and dE_2, it is possible to evaluate the similarity of two different-sized distributions. Since this approach initially compares a distribution with its corresponding uniform distribution, the comparison between two different-sized distributions always happens on a standard scale as desired. The values of dE_1 and dE_2 indicate 46.51% and 7.46% of decrease in entropy respectively. Thus the right-hand side distribution (proposed) packs in much more information with less number of bins and thus provides a better feature representation of a DR image as shown in Fig. 2.3. Though the image on the right does not explicitly capture lesions, it better represents the DR image by packing in more information with the help of spectrally-tuned bins and in turn, produces a more discriminative feature space.

Steerable Gaussian Filter

To retrieve clinically relevant images for all the three classes, color features alone are not sufficient. Since the shape and texture of DR lesions are significant factors in deciding the severity level of the disease, I make use of the steerable Gaussian filter (SGF) coupled with fast radial symmetric transform (FRST).

Steerable Gaussian filters are oriented filters and are used in many computer vision and image processing tasks such as edge detection, classification of lines, edges and contours. They are filters of desired orientation obtained from the linear combination of basis filters (Freeman and Adelson, 1991). This process is called "steering". The ability of SGF to model edges and contours makes it useful for DR images since edges and contours at various orientations are distinguishing features for different severity levels of DR. The filters shown in Fig. 2.6 are separated by 45°. These filters are used as kernels to calculate directional derivatives of an image which help to model contours at different orientations.

In case of DR images, it means blood vessels, lesions, optic disk, etc. will have different

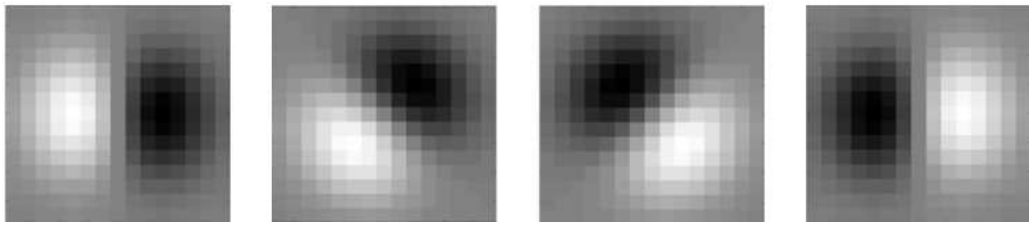


Figure 2.6. Some of the Bases of Steerable Gaussians Filters.

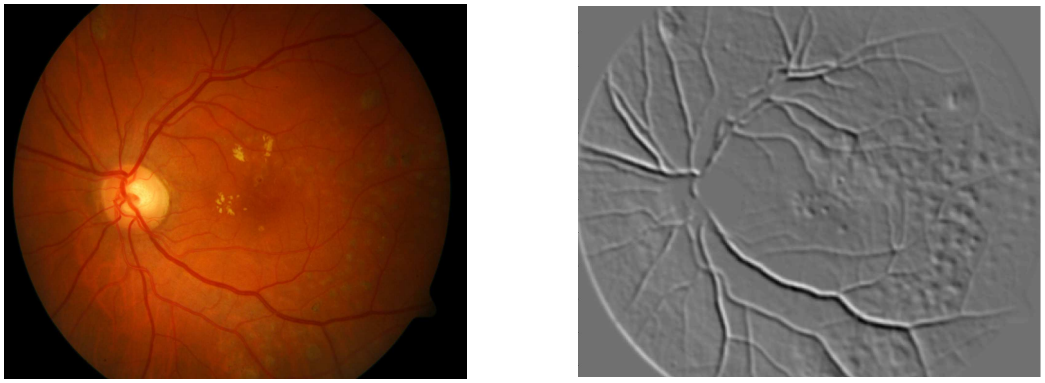


Figure 2.7. SGF Filter Response to an NPDR Image. Input NPDR Image (Left) and Filter Response on the Right.

filter response since each of them has a different structure. Fig. 2.7 shows the filter response of a typical DR image to a filter orientated at $225°$. The peaks in the SGF response can be attributed to the textural discontinuity at those particular locations. The statistical variation in its values can be modeled by taking the standard deviation, skewness, and kurtosis. I form a 24D feature vector for a total of eight angles by including the first three moments of the SGF response for each angle. Fig. 2.7 shows the SGF response rightly peaking at locations containing the DR lesions, but many blood vessels are also unnecessarily highlighted. Due to such false-positives in the feature space, the SGF response alone cannot be used as features. To filter out unwanted high SGF response regions, a second complementary stage is needed which can select regions of interest. To detect such regions, I use FRST (Loy and Zelinsky, 2003).

Fast Radial Symmetric Transform

Numerous context-free point-of-interest operators have been proposed in the literature. These work on the principle that points which possess local radial symmetry are the ones which draw human attention. The principle is based on the psychological findings proposed by researchers in the past (Locher and Nodine, 1987; Richards and Kaufman, 1969; Kaufman and Richards, 1969). Use of FRST is preferred over other point-of-interest operators because of its superior performance and its speed. It works in linear time. The comparison of FRST with other state-of-art interest point detectors is given in section 2.5.

FRST [2] uses local radial symmetry to highlight interest points. It accounts the contribution of radial symmetry of gradients that are at a distance d from the point under consideration. The pixel at a distance d to which the gradient points is called as the positive pixel and the pixel from which the gradient points away is called as the negative pixel. The coordinates of these positive and negative pixels are obtained as follows.

$$p_{\substack{+\\(-)}ve}(p) = p_{\substack{+\\(-)}} round\left(\frac{g(p)}{||g(p)||}n\right), \tag{2.5}$$

where $g(p)$ is the gradient of pixel p and n is the radius. Therefore $p_{+ve}(p)$ ($p_{-ve}(p)$) gives coordinates of a positive (negative) pixel at a distance n from pixel p. The transform estimates the contribution of each pixel to the symmetry in its neighborhood by using the concept of positive and negative pixels instead of calculating the contribution of the local neighborhood to a central pixel. I have used a set of radii - $\{1, 3, 5\}$, to detect interest points robustly and to negate the presence of nerves and other spurious components that may lead to noisy feature space. Quantitative analysis shows that examining a small subset of radii gives a good approximation to the output (Loy and Zelinsky, 2003). The values of radii should be low enough to capture smallest of lesions. The retrieval performance worsens as

[2] FRST was implemented using publicly available code (Kovesi, 2000)

Figure 2.8. Left Three Images: Interest Point Detection Using FRST on Normal, NPDR and PDR Images. The Extreme Right Image: It Shows the FRST Interest Points Superimposed on the SGF Response Shown in Fig. 2.7 (Please Zoom in for Better Viewing)

the values of radii increase. Effect of parameters of FRST on retrieval performance has been analyzed in section 7.1.4.

In Fig. 2.8, though most of the detected points are lesions in images of affected eyes, false detections do exist. FRST works on local maxima detection. The interest points are then generated by performing non-maxima thresholding on the image created by FRST. The detected interest points are superimposed on the original image for visualization purpose (See Fig. 2.8). Fig. 2.8 illustrates that the FRST points (shown in red marks) are complementary to the SGF response, thereby forming an effective combination. The procedure to couple the SGF response and FRST is described below.

Once the interest points have been obtained through FRST, the image is divided into 64 blocks by an 8×8 grid. Blocks containing a majority of the black background are removed by thresholding. Every image block is considered independently, and a region of 15×15 is selected around each interest point. A 24D feature vector is calculated for the corresponding 15×15 region in the filtered image (Fig. 2.7). Consider that there is n number of interest points detected on a given image block, they would correspond to n feature vectors. I take mean over each dimension to reduce the n feature vectors to a single 24D vector. The same action is carried out when there are multiple interest points lying in the 15×15 region. If

there are no interest points detected in the region, then the feature vector is all zeros. For PDR images, more interest points are likely to be found in the small region as compared to NPDR and normal images. By taking the average of the features of all the interest points in an image block, the NPDR image features obtain fewer values while normal image features get even lesser values. Therefore instead of removing a region containing zero interest points, replacing its feature vector with zeros produces a more discriminative feature space. The feature space is now suitable for multiple-instance retrieval framework.

2.3.2 MIRank-KNN

The retrieval scheme I propose here is called *MIRank-KNN* (multiple-instance-rank KNN), and it produces a ranked list of all the images in an archived data-set by considering all the blocks in a given image simultaneously. Thus the proposed scheme emphasizes even on a small block of an image containing a lesion. To the best of my knowledge, this is the first attempt at developing an unsupervised approach that retrieves images based on multiple blocks (instances) of the query image. Supervised multiple-instance retrieval methods have been developed that borrow concepts from multiple-instance learning (MIL) (Zhang *et al.*, 2002, 2005; Yang and Lozano-Perez, 2000; Rahmani *et al.*, 2005).

MIL was first introduced in Dietterich et al. (Dietterich *et al.*, 1997). It is a form of supervised learning where the data is in the form of labeled bags. Each bag contains a variable number of instances. The labels are provided on a bag-level. A positive bag has at least one positive instance. In a negative bag, all the instances are negative. The main goal is to classify each bag as positive or negative and if possible, infer instance-level labels. This is a difficult problem due to lack of knowledge of instance labels. There have been a plethora of approaches developed to solve MIL problems. I list only those approaches which laid the foundation and are relevant in this context.

Diverse density (DD) was proposed as a general purpose solution for the MIL problems

(Maron and Lozano-Pérez, 1998). It attempts to find a concept point in the feature space which is close to at least one instance from a positive bag and is away from all the instances in a negative bag. It is also called as instance prototype and has maximum diverse density. With the knowledge of these instance prototypes, each bag can now be classified based on its distance from the prototype. The expectation-maximization procedure was coupled with DD (EM-DD) to solve MIL problems (Zhang and Goldman, 2001). Support vector machines were also employed for MIL (Andrews *et al.*, 2002). KNN-based MIL method called Citation-KNN was developed (Wang and Zucker, 2000). It uses the concept of references and citers of a bag for its classification. References of a bag (say bag B) are its nearest-neighbors, and citer is a bag which considers bag B among one of its neighbors. Table 2.1 illustrates the concept of references and citers of a bag. Two nearest references of bag B_1 are B_2, B_4 while its two nearest citers are B_4, B_2. Assume that the query image is B_2, then the nearest-neighbor rank of B_1 is 2 (since B_1 is the second nearest neighbor of B_2) and the citer-rank of B_1 is 1 (since B_2 is the nearest neighbor of B_1). For a more formal definition of references and citers, readers are pointed to Wang and Zucker (Wang and Zucker, 2000). Multiple instance learning techniques like diverse density (DD) and expectation-maximization DD (EM-DD) are used for multiple-instance CBIR (Yang and Lozano-Perez, 2000; Zhang *et al.*, 2002). EM-DD was used in Rahmani et al. (Rahmani

Table 2.1. Nearest References and Citers of Four Bags.

	$N = 1$	$N = 2$	$N = 3$
B_1	B_2	B_4	B_3
B_2	B_3	$\mathbf{B_1}$	B_4
B_3	B_2	B_1	B_4
B_4	$\mathbf{B_1}$	B_2	B_3

et al., 2005). One-class support vector machine (SVM) was also used (Zhang *et al.*, 2005). All the methods mentioned above use a learning stage before starting the process of retrieval. I describe the proposed unsupervised retrieval framework - MI-RankKNN - in detail.

Each image is divided into 64 blocks on a 8×8 grid. CC features are extracted for each block separately. The SGF and FRST features are also extracted for these blocks. This gives us a 280-D (256-CC features + 24-statistics of SGF features) feature vector for each block. It should be noted that the number of blocks per image may vary as the number depends on the thresholding scheme. Recall that citation-KNN (Wang and Zucker, 2000) is one of the most popular KNN-based-method used to solve multiple-instance classification problems. It uses modified Hausdorff distance. Hausdorff distance is used to calculate the distance between two subsets of metric space. In short, it gives a measure of dissimilarity between the two metric spaces. Since the Hausdorff distance is very sensitive to outliers, minimal Hausdorff distance was proposed (Wang and Zucker, 2000) which is defined as,

$$H(A, B) = \min_{b \epsilon B} \min_{a \epsilon A} \|a - b\|, \tag{2.6}$$

where $H(A, B)$ denotes the Hausdorff distance between two bags (non-empty subsets of a metric space) A and B whereas a and b are instances in bags A and B respectively. When applied to DR images, this may not give desired results. For example, in Fig. 2.2, only two instances (marked by red borders) are different from the instances in the normal image. Thus the minimal Hausdorff distance between an NPDR and a normal image will almost be equal to the distance between two normal images. This calls for a new way of measuring the distance between images which will effectively capture small, localized lesions.

Consider two images X and Y with m and n blocks each. The minimum distance between features of i^{th} block of X and all blocks of Y is given by,

$$D_{(i)}(X, Y) = \min_{y \epsilon Y} d(x_i, y_j) \ \forall \ j \epsilon \{1, 2, \ldots, n\}. \tag{2.7}$$

Here, $d(\cdot, \cdot)$ represents Euclidean distance between features of two blocks . D is an m-

dimensional vector which records the distance between each block of X and its closest match in Y. The distances used in this chapter are Euclidean unless otherwise specified. I use the above equation to calculate the D vector for each block in the query image with every image in the database. I match each block of the query image to its closest image in the database. A simple sorting of all the matches for each block in the query image gives the list of best-matched blocks for each query block in the data-set and is called the similarity ranked-list. An aggregated ranked-list is created for each image in the database by averaging the similarity rank for each block in every image in the database. The sorted list of aggregated ranks of images gives the m-rank list which is to be treated as the final ranking. The algorithm to obtain the m-rank is as follows.

---

m-Rank algorithm:

---

1. Calculate D' between Q (query image) and every image in the database.

2. Sort D' along columns and its indices give similarity list. $SL(x, y) = z$ represents the y^{th} best match for the x^{th} block of Q and the best match is a block belonging to image z.

3. Calculate aggregated ranked-list of all images in the database by averaging the similarity rank for each block.

4. Sort aggregated ranked-list to obtain the *m-Rank* list.

The nearest-neighbor metric may not be sufficient always to generate optimum retrieval results for DR images (Wang and Zucker, 2000). This problem is avoided by incorporating the citer-rank to the framework of *m-Rank*. The citer-rank is calculated as mentioned in section 2.3.2. This inclusion of citer-rank is accomplished by obtaining a final *meanRank* which is the average of *m-Rank* and citer-rank of Q. The top-k retrieved images are then based on the *meanRank* list.

Since the distances in MIRank-KNN are calculated on the block-level, it is more sensitive to the presence of localized lesions. For example, the distance between an affected image that contains a small block of lesions, and a normal image will produce a vector D which has one large value. That lone high value can improve the overall retrieval ranking. By storing the distances between blocks of all the images in the databases (and thereby continuously updating that distance matrix), the process of retrieval can be made significantly faster.

2.4 Experimental Setup

The data-set used to evaluate the proposed set of features consists of 493 images, assembled from four well-known and publicly available databases. Those include DIARETDB0 (Kauppi *et al.*, 2006), DIARETDB1 (Kauppi *et al.*, 2007), STARE (McCormick and Goldbaum, 1975) and Messidor [3] and an annotated data-set of 84 PDR images from Jaeb Center for Health Research. For reviewing purpose, the data-set and the source code used in this study is publicly available [4] . There are 164 normal images, 161 NPDR images, and 168 PDR images. Labeling of the first three data-sets was unambiguous since the labels were well-defined and had three categories as desired. The Messidor data-set has three categories apart from the normal condition. They are defined by the amount of presence of microaneurysms, hemorrhage, and neovascularization. The images from the Messidor data-set were classified as NPDR if the annotations indicated absence of neovascularization in them, and accordingly, I also classified PDR images if their annotations indicated a strong presence of microaneurysms, hemorrhages as well as neovascularization. It should be noted that the database used consists of DR images from five different sources. The retrieval results show that the results are consistent with all images. Therefore the proposed approach

[3] Kindly provided by the Messidor program partners, see http://messidor.crihan.fr

[4] The implementation, and the database is available at `www.public.asu.edu/~bli24/DR-System-and-Data.html`

is robust in handling variations in color and brightness of images as well as other conditions while capturing them. Due to the adaptive nature of the approach, it can be applied in various places to get effective results.

The proposed approach was evaluated against prior state-of-art vision methods which use Gabor features (Manjunath and Ma, 1996) and semantic of neighborhood color moment histogram features (Chen *et al.*, 2008). A short description of each approach follows.

1. Gabor Feature-based image retrieval: Gabor features have been used for textured image retrieval (Manjunath and Ma, 1996). Gabor wavelet transform of an image is calculated over four scales and six orientations. Mean, and standard deviation of the transform is calculated over all orientations and scales. I Combine the statistics of all the orientations and scales to form the final feature vector.

2. Histogram of Neighborhood Mean moments (HNM): HNM has been used for retrieval of gastroscopic images which are predominantly red (Chen *et al.*, 2008). The image is first transformed into HSV color space and then quantized. Low-order moments are known to express the color distribution well. Therefore, first three central moments for each pixel in its 3×3 neighborhood are calculated. After operating on each pixel, three distinct histograms are calculated from the matrices of central moments. These histograms are concatenated to form the final feature vector.

These two approaches use k-nearest neighbor based retrieval systems. Distances providing best retrieval results are selected. All 493 images were queried one-by-one and the top ($k =$) 5 images were retrieved using the approach. Popular evaluation metrics were adopted and used (Sigurbjörnsson and Van Zwol, 2008) and (Liu *et al.*, 2007).

2.5 Results and Analysis

This section presents the result of the proposed multiple-instance retrieval approach on the DR image data-set. The approach produces state-of-art results even when there is a wide variation in the lighting of images. There are two main parameters in this approach: number of bins in CC features, FRST radii. I show the effect of parameter tuning on the results. I compare the proposed approach with two state-of-art image retrieval systems: textured image retrieval using Gabor features (Manjunath and Ma, 1996), medical image retrieval using HNM (Chen *et al.*, 2008). The proposed approach consists of several important components; namely, CC features, FRST, SGF and the multiple-instance retrieval framework - MIRank-KNN. I analyze the effect of individual components by comparing them with other widely used features. I compare FRST with SIFT (Lowe, 2004a), FAST (Rosten and Drummond, 2005, 2006) and Harris corner with color saliency boosting (Van De Weijer *et al.*, 2006) and show that FRST produces better results. I also compare CC features with transformed color histogram (Van De Sande *et al.*, 2010), color moments (Van De Sande *et al.*, 2010) and original AutoCC features (Li, 2007). The MIRank-KNN framework considers image blocks (instances) while retrieving images. To show its superiority, I replace it with the Citation-KNN retrieval framework (explained later) which works on multiple-instance feature space but does not concentrate on all the blocks of an image simultaneously. Finally, to justify the choice of multiple-instance-based-retrieval framework, I compute the proposed features as well as a set of state-of-art local features on the entire image and compare the results.

2.5.1 Results of the proposed approach

In this section, I present statistical analysis of the proposed retrieval approach and compare it with three aforementioned retrieval approaches. I present four evaluation metrics: 1. $\geq k$ hit-rate 2. mean accuracy at k^{th} rank 3. precision at k^{th} rank and 4. mean average precision (MAP). $P@k$ and MAP metrics were proposed for document retrieval (Liu

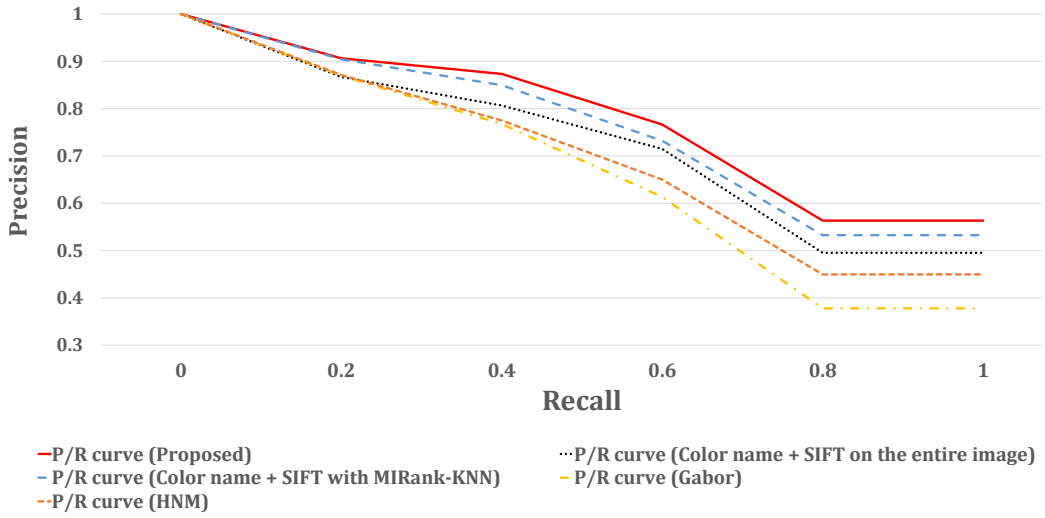


Figure 2.9. Precision-recall Curves for Five Methods When Five Images Are Retrieved.

et al., 2007) but were also used effectively for image retrieval in (Faria *et al.*, 2010). The $\geq k$ hit-rate (HR) is defined as the percentage of images for which at least k relevant images were retrieved. Mean accuracy at k^{th} rank denotes the percentage of relevant images retrieved at that particular rank. Precision at k^{th} rank ($P@k$) measures the relevance of top k images in the ranking result with respect to the query image. It is also the probability of finding a relevant image in the top k images, given by,

$$P@k = \frac{\#\text{ relevant images in top k images}}{k}. \tag{2.8}$$

I average the $P@k$ values for all the queries to get a single $P@k$ value. Average precision (AP) is defined as average of $P@k$ values for all relevant queries.

$$AP = \frac{\sum_{k=1}^{N}(P@k * rel(k))}{\#\text{ total relevant images for the query}}, \tag{2.9}$$

where N is the number of retrieved images, and $rel(k)$ is an indicator function on the relevance of the n^{th} image given by:

Table 2.2. Mean Accuracy and Precision at k^{th} Rank (in %). Best Results Are in Bold.

	$Acc@1/P@1$	$Acc@2/P@2$	$Acc@3/P@3$	$Acc@4/P@4$	$Acc@5/P@5$
Gabor	73.43/73.43	72.62/73.02	68.76/71.60	67.75/70.64	70.39/70.59
HNM	77.69/77.69	74.04/75.86	72.01/74.58	73.83/74.39	69.57/73.43
Proposed	**84.38/84.38**	**81.54/82.96**	**83.37/83.10**	**79.51/82.20**	**80.93/81.95**

Table 2.3. $\geq k$ Hit-rate (in %). Best Results Are in Bold.

	≥ 1 HR	≥ 2 HR	≥ 3 HR	≥ 4 HR	≥ 5 HR	Mean Acc.	MAP
Gabor	93.91	86.61	76.06	60.24	36.10	70.59	79.68
HNM	94.52	87.22	77.28	64.50	43.61	73.43	82.49
Proposed	**96.55**	**91.48**	**87.42**	**76.88**	**57.40**	**81.95**	**87.6**

$$rel(k) = \begin{cases} 1, & \text{if the } k^{th} \text{ image is relevant} \\ 0, & \text{otherwise.} \end{cases} \tag{2.10}$$

MAP is obtained by averaging AP values for all the queries. It is clear that $P@n$ and MAP tends to emphasize the quality at higher ranks. But MAP is less affected by this since it depends on the entire list of retrieved images.

Five ROC curves are shown in Fig. 2.9, each one corresponding to one of the five retrieval approaches, namely, color name and SIFT on the entire image as well as blocks, Gabor features, HNM and the proposed approach. Each curve was calculated as the average of precision-recall curves for normal, NPDR and PDR images as follows. I retrieve five images and calculate the precision and recall values in response to each query image. To get precision and recall for the entire category, I average over all images of that category. To

calculate precision at all recall values, I use the concept of *interpolated precision* (IP). IP p_{interp} at a recall level r is defined as the maximum precision obtained for any recall level $r' \geq r$. Mathematically,

$$p_{interp}(r) = \max_{r' \geq r} p(r'). \tag{2.11}$$

I get IP at recall values ranging from 0 to 1 in the steps of 0.2. For more details about precision-recall curve calculation, I refer the reader to Manning et al. (Manning *et al.*, 2008).

Mean accuracy at k^{th} rank and precision at k^{th} rank are given in table 2.2. Table 2.3 provides values of MAP, overall retrieval accuracy and the hit-rates at all ranks. Mean accuracy produces consistently higher values which suggest that the proposed approach retrieves clinically relevant images at all ranks. The ratio of relevant to irrelevant retrieved images is high in the proposed approach as compared to the other methods. The proposed approach produces higher precision at every rank and higher MAP value than other approaches. Therefore the proposed approach has produced better results by retrieving clinically relevant images at all ranks instead of just at top ranks.

I conduct a visual inspection of the results produced by HNM. It shows that HNM did not quite retrieve images with clinically similar lesions, particularly images containing hemorrhages. HNM retrieved only those images which have similar brightness conditions as the query image. Thus HNM reduced the clinical relevance and the generalization ability in this case. Since HNM models the global distribution of patch-level statistics, it produces higher performance than Gabor features. When compared to the results of the proposed approach, the reduction in performance can be attributed to the following factors:

1. HNM quantization scheme is not tuned to DR images.

2. HNM fails to model the global distribution of local spatial correlation of colors, unlike CC features.

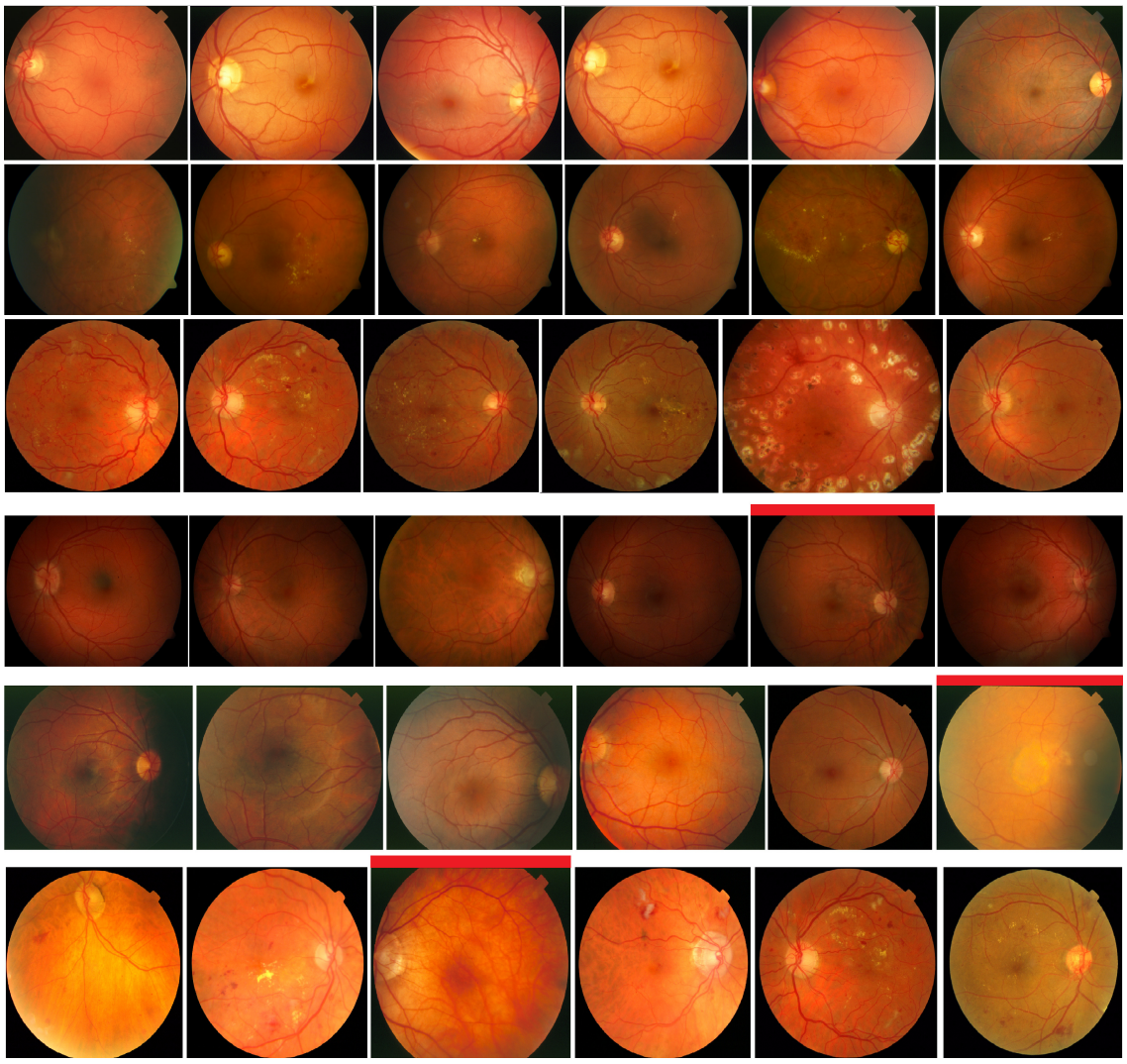


Figure 2.10. Retrieved Images Using the Proposed Approach. Each Row Contains a Query Image (Leftmost) and Five Retrieved Images. A Retrieved Image Belongs to the Same Category as the Query Image Unless the Retrieved Image Has a Red Bar over It. Query and Its Corresponding Retrieved Images in Top Three Rows Belong to Normal, NPDR and PDR Category, Respectively. In the Fourth Row, the Query Image Is NPDR and the Fourth Retrieved Image Is Normal. The Fifth Row Contains a Normal Query and the Fifth Retrieved Image Is PDR. In the Sixth Row, the Query Has PDR, and the Second Retrieved Image Is Normal. Please View in Color.

Table 2.4. Mean Confusion Matrix (in %)

	Normal	NPDR	PDR
Normal	81.59	13.78	4.63
NPDR	9.81	80.50	9.69
PDR	8.45	7.86	83.69

3. HNM feature is calculated on an image level, whereas CC features in the proposed approach are calculated on a block level. This makes CC features more *descriptive*.

I perform two experiments to show: 1. the compatibility and reproducibility of the approach and 2. adaptability to various databases with minor updates. In the first experiment, I perform 20 iterations and on each iteration, a random subset of 95% of total images is selected and all the images are queried one-by-one. In the second experiment, the quantization scheme is designed by using a randomly selected database of 95% of the images, which ensures that algorithm performance does not heavily degrade by adding a few images. In both experiments, accuracy and hit-rates remained consistent as desired.

A mean confusion matrix is created to better understand the results of the proposed approach. It is shown in Table 2.4. The following example illustrates the process of constructing a confusion matrix. Suppose all images are queried one-by-one resulting in n retrieved images. The first row of the confusion matrix shows that 81.59% of the n images were normal when a normal image was queried. Similarly, 13.78% images were NPDR and 4.63% images obtained were PDR, while querying a normal image. The second and third row can be similarly explained. The rate of retrieving a normal image in response to a queried PDR image (and vice versa) is less. This is following the requirements of the algorithm. Retrieving a normal image for a PDR image can be quite harmful. Similarly, a retrieved PDR image in response to a normal query image might mislead the ophthalmologist.

Images retrieved by the proposed approach are shown in Fig. 2.10. Top three rows contain successful retrievals for normal, NPDR and PDR images respectively. In spite of varying illumination and exposure, my approach yields perfect results. However, there are some cases when the approach fails. In the fourth row of Fig. 2.10, the query image is NPDR where the lesions span only a few pixels. My approach retrieves a normal image on the fourth rank. There are subtle differences in the retinal blood vessels of the query NPDR and the retrieved normal image, which the proposed method fails to capture. In the fifth row, a PDR image is retrieved in response to a normal image. In that PDR image, no lesions are visible, and even optic disc is hardly visible. That makes the retrieval difficult. In the last row, a normal image is retrieved when a PDR image is queried. The normal image is extremely red and the retinal blood vessels have a different appearance as compared to the other normal images. The goal of perfect retrieval on a variety of fundus images is still far from achieved but I think the presented results are a promising step in that direction.

My approach is robust to exposure and lighting changes in an image to some extent. Today, cameras are high-quality and usually do not produce noisy or poorly exposed images. In a rare case, clinicians prefer to retake the picture so that their diagnosis is not affected. However, I acknowledge that small changes in exposure can happen and may be acceptable to an ophthalmologist as long as a correct diagnosis can be made. In the literature, many approaches perform histogram equalization to try to correct effects of bad exposure. The proposed spectral tuning method is better since it adapts to the color spectrum of retinal images. The robustness of this system can be assessed by the fact that I get 87.6% MAP on a data-set of varying illumination and exposure retinal images.

2.5.2 Effect of varying illumination

Through my interactions with ophthalmologists, I have observed that if the quality of the retinal image captured is unacceptable, then another photo is captured. Thus it is reasonable

Table 2.5. $\geq k$ Hit-rate (in %) with Images of Varying Intensity.

	≥ 1 HR	≥ 2 HR	≥ 3 HR	≥ 4 HR	≥ 5 HR	Mean Acc.	MAP
No variation	96.55	91.48	87.42	76.88	57.40	81.95	87.6
Linear variation	97.36	94.73	90.47	82.76	64.71	86.00	90.85
Non-linear variation	99.19	95.74	90.67	82.56	63.69	86.37	92.32

to assume that the photos will always be of high-quality. However, photos captured in different labs or settings can have different contrasts and illuminations. I show that the proposed algorithm can handle these changes well through an experiment described as follows. I change (i.e., increase or decrease, which is randomly chosen) the brightness of each image by a random amount, ranging anywhere from 15% to 25% of the original brightness. I change the brightness by mapping the values of the "V" channel of an image to a new intensity curve. I also introduce a nonlinear effect by controlling the shape of the new intensity curve using γ. The value of γ is also randomly chosen from the interval, $[0.6, 1.4]$. I pick an image from the original set of images and retrieve from the other set, having modified values of brightness. I repeat this for each image and calculate the same metrics as done previously. Note that I do not spectrally-tune the proposed color correlogram feature using the modified images, which is needed for a fair assessment of the effect of varying brightness on the performance. Table 2.5 shows the results of both the experiments. It is interesting to see that the images with modified intensity values get better results than the original images. Since the original database has a lot of intensity variation, it is possible that the images which were too dark got benefited from this experiment and hence the performance improvement. The results of both the experiments show that the proposed approach can handle intensity variations with a reasonable tolerance.

Table 2.6. Effect of Parameter Tuning on Retrieval Accuracy

Parameters		Overall Accuracy	MAP
FRST radii	$\{1, 3, 5\}$	**81.95**	**87.60**
	$\{5, 7, 9\}$	80.57	87.17
	$\{10, 15, 20\}$	78.34	86.33
Number of bins in CC	8	79.95	86.08
	32	**80.16**	87.78
	64	80.12	**88.02**

2.5.3 Effect of different feature combinations in the proposed approach

I analyze the contribution of color features and texture features in the proposed approach. Images are retrieved using two separate sets of features: 1. CC features and 2. FRST + SGF features. By using texture features in set 2 alone, I get 55.25% accuracy whereas by using only color features of set 1, 77.32% can be obtained. The features are indeed complementary since the results improve considerably after using both feature sets.

2.5.4 Effect of parameter tuning

The proposed approach has two main parameters: 1. The number of bins in CC features 2. Set of radii in FRST. I analyze the effect of those parameters on the retrieval accuracy.

1. Number of bins in CC features: The dimensionality of CC features is a function of the number of bins - $O(m^2)$. I choose to quantize each image into 16 bins. However, I show that the retrieval performance does not vary by a large amount with respect to the number of bins.

2. FRST radii: The values of FRST radii should be small enough to capture smallest of

Table 2.7. Analysis and Comparison Between the Proposed and the Other State-of-art Approaches

Method	Parameters	Overall Accuracy	MAP
Proposed approach	**FRST radii=$\{1, 3, 5\}$ and # bins=16**	**81.95**	**87.6**
FRST replaced with other interest point detectors			
SIFT	Peak Threshold=1	79.07	86.27
Harris Corner with Boosted Color Saliency	Top 150 interest points	77.04	84.29
FAST	Threshold=7.5	80.08	86.79
CC replaced with other color features			
Transformed color histogram	# bins=32	74.85	83.15
	# bins=64	73.51	80.98
	# bins=128	68.36	78.00
Color moments	# bins=16	75.74	83.31
	# bins=32	76.51	83.15
Original AutoCC features	# bins=64	53.79	68.99
MIRank-KNN replaced with Citation-KNN retrieval framework			
Citation-KNN retrieval	FRST radii=$\{1, 3, 5\}$ and # bins=16	65.60	74.38
	FRST radii=$\{5, 7, 9\}$ and # bins=16	65.23	73.81
	FRST radii=$\{1, 3, 5\}$ and # bins=8	67.99	76.00
	FRST radii=$\{1, 3, 5\}$ and # bins=32	69.70	79.27
Results without the use of multiple-instance framework			
Whole image features	FRST radii=$\{1, 3, 5\}$ and # bins=16	58.83	72.02

lesions. A set containing small, arbitrary values of radii produces good results. As the values of radii increase, the retrieval performance drops. If the lesion size is known beforehand, values of radii can be set accordingly. I use three radii - $\{1, 3, 5\}$ in my implementation.

The results of parameter tuning are given in Table 2.6.

2.5.5 Comparison with other state-of-art interest point detectors

FRST is a crucial component of this system. Its main advantages are speed and quality of detected interest points. By adjusting its parameters, it is possible to detect even smallest of lesions. Interest point detectors are bound to produce false detections. The SGF stage acts as a complementary stage to FRST. I compare three other state-of-art interest point detectors, namely, SIFT (Lowe, 2004a), FAST (Rosten and Drummond, 2005, 2006) and Harris corner with color saliency boosting [5] (Van De Weijer *et al.*, 2006). Results in Table 2.7 show the superiority of FRST.

2.5.6 Comparison with other state-of-art color features

Color is a distinguishing characteristic for DR images and hence spectrally-tuned CC features play a major role in the retrieval process. The CC features are invariant to lighting color changes. I compare them with three other color features which are also invariant to lighting color changes. The three color features are: 1. transformed color histogram (Van De Sande *et al.*, 2010) 2. color moments (Chen *et al.*, 2008) 3. original AutoCC features (Li, 2007).

[5] SIFT, FAST and Harris corner with color saliency boosting have been implemented from publicly available code at (Vedaldi and Fulkerson, 2008), (Rosten and Drummond, 2006) and `http://lear.inrialpes.fr/people/vandeweijer/code/ColorConstancy.zip`

Transformed color histogram

A transformed color space is created as shown in equation 2.1. Histograms of individual color channels are then concatenated to obtain the final feature vector. This feature, though invariant to light color changes, does not encode local spatial correlation of colors.

Color moments

The image is quantized by using the proposed quantization approach. Then the color moment feature vector is calculated identically to HNM. Since the proposed quantization scheme is used, the pros and cons of color moments can be analyzed.

Original AutoCC features

Original AutoCC features were used for content-based image retrieval. I use their human-vision based quantization scheme and calculate the AutoCC features for all possible color pairs (i, i) where $i \in [1, \ldots, 64]$. I get a 64D feature vector for each image. From the results in Table 2.7, the ineffectiveness of this quantization scheme in case of DR images is observed.

2.5.7 Effect of MIRank-KNN on retrieval performance

MIRank-KNN considers features of all the image blocks simultaneously in the retrieval process. The existing multiple-instance-retrieval algorithms (Yang and Lozano-Perez, 2000; Zhang *et al.*, 2002; Rahmani *et al.*, 2005; Zhang *et al.*, 2005) for natural images and DR-CBIR algorithms (Quellec *et al.*, 2011, 2012b, 2010) require training and some of them even require user to specify the region of interest. On the other hand, the proposed algorithm does not require labeled data for training and the entire approach is automated. The labeled data is only used for calculating retrieval accuracy and other metrics. I show the superiority

Table 2.8. Performance of Local Features

Method	Overall Accuracy	MAP
Color Name + SIFT + KNN on the entire image	75.29	83.06
Color Name + SIFT + MIRank-KNN on the image blocks	**78.86**	**85.87**

of MIRank-KNN through a three-fold experiment.

Firstly, I replace MIRank-KNN with a conventional k-nearest neighbor approach while keeping the same feature space. I calculate the CC features for the entire image instead for individual blocks. Averaging of the feature vectors of the interest points given by FRST is done for the entire image instead on a block-level to produce the final features. As expected, this fails to encode characteristics of small lesions and produces poor results. The results are given in Table 2.7 (last row).

Secondly, I introduce another baseline framework - *Citation-KNN retrieval* - based on the lazy-learning framework (Wang and Zucker, 2000). In the Citation-KNN retrieval framework, the image is first divided into 64 blocks as done previously. The same feature set is calculated for each block. Consider two images I_1 and I_2 consisting of n_1 and n_2 blocks respectively. I measure the similarity between these two images by calculating the minimal Hausdorff distance between them as shown in equation 2.6. This distance measure was also used in Zucker and Wang (Wang and Zucker, 2000). Though this distance can work with multiple blocks, it cannot capture the appearances of localized lesions in its distance computation. For example, the minimal Hausdorff distance between a normal image and an MA image, shown in Fig. 2.2, will be very small. Using maximal Hausdorff distance is a bad option too since it is sensitive to outliers. On the other hand, MIRank-KNN will

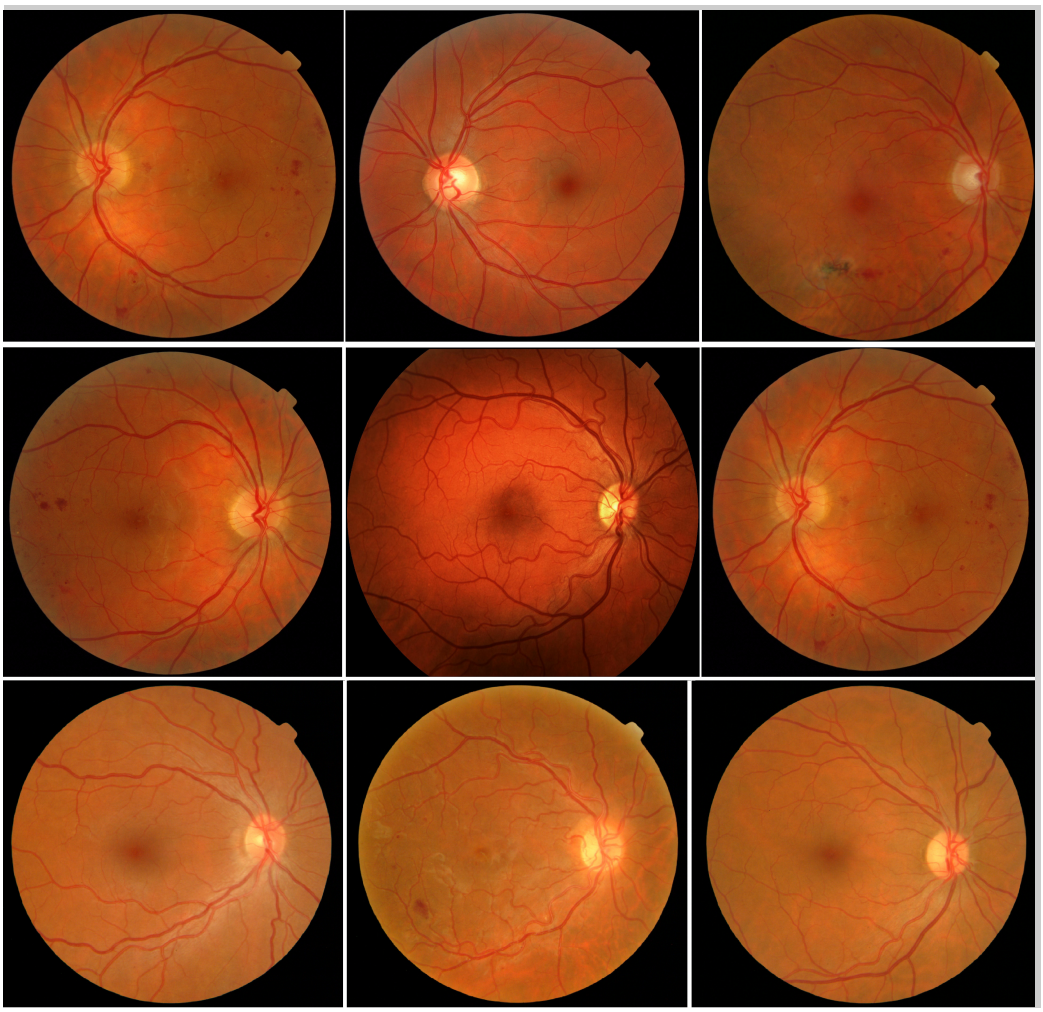


Figure 2.11. Left Column Shows a Query Image. Middle and Right Columns Show Retrieved Images by Using Local Features with k-nearest Neighbor Retrieval and Local Features with MIRank-kNN Retrieval Respectively. In the Top Two Rows, Left and Right Images Are PDR Whereas the Middle Image Is Normal. In the Last Row, Left and Right Images Are Normal and the Middle One Is PDR.

output large distance for at least two blocks in its distance vector, thus not recognizing the other image as a clinically-relevant image. Table 2.7 contains the results of Citation-KNN retrieval framework.

Finally, I show the importance of multiple-instance framework. I show that merely using

state-of-art local features is not enough to characterize localized lesions. I do so in two parts as follows.

1. I assess the performance of local features alone, without involving multiple-instance framework. I use SIFT and color name descriptors which are excellent local shape and color features respectively (Deselaers *et al.*, 2008; Shahbaz Khan *et al.*, 2012). Color name descriptor has been recently shown to be effective for image retrieval (Zheng *et al.*, 2014). I follow a similar procedure for feature extraction. I extract SIFT keypoints and their descriptors to characterize the shape of lesions. Color name descriptors are extracted around SIFT keypoints and are appended to the SIFT descriptors. I then use nearest-neighbor retrieval.

2. I show the effectiveness of multiple-instance framework. I divide the image into 64 patches (8×8 grid). I extract the same features as described above for each patch. MIRank-KNN is used for retrieval.

Results in Table 2.8 show that the combination of local features and MIRank-KNN produces better performance than local features alone. I also show three visual examples in Fig. 2.11. It shows that using local features on the entire image can yield wrong results when the lesions are small and localized. Thus normal images may be retrieved in response to affected (NPDR or PDR) images and vice versa.

2.6 Discussion

This chapter presents a novel unsupervised approach for retrieving clinically-relevant DR images [6] . The approach consists of a feature space which is spectrally-tuned to the DR spectrum. Feature space makes near-optimal utilization of the quantization scheme and

[6]Most of the material in this chapter has appered in (Chandakkar *et al.*, 2017b). See the full credit statement in appendix A.

thus produces a better representation of the image even with less number of bins. It makes sure that shades in DR images are almost uniformly spread across all the quantization bins, thereby creating a feature space with much higher entropy. The proposed approach also consists of a multi-class multiple-instance retrieval framework called MIRank-KNN that uses minimal Hausdorff distance and considers multiple regions of an image simultaneously. The results using the proposed approach are reported and compared with other state-of-art retrieval frameworks in the literature. The ability of multiple-instance framework to capture localized lesions was also analyzed over state-of-art local features. It was found that in all of the cases, the proposed multiple-instance retrieval framework provides a boost to the results. The results based on the DR image data set suggest that the proposed method can give good performance on a different data-set and is robust against varying illuminations and exposures. It is also invariant to small additions or removal of data.

Chapter 3

STRUCTURED PREDICTION OF IMAGE ENHANCEMENT PARAMETERS

3.1 Introduction

The growth of social networking websites such as Facebook, Google+, Instagram etc. along with the ubiquitous mobile devices has enabled people to generate multimedia content at an exponentially increasing rate. Due to the easy-to-use photo-capturing process of mobile devices, people are sharing close to two billion photos per day on the social networking sites [1] . and they want their photos to be visually-attractive. This has given rise to the automated, one-touch enhancement tools. However, most of these tools are pre-defined image filters which lack the ability of doing content-adaptive or personalized enhancement. This has fueled the development of machine-learning based image enhancement algorithms.

Many of the existing machine-learned image enhancement approaches first learn a model to predict a score quantifying the aesthetics of an image. Then given a new low-quality image [2] , a widely-followed strategy to generate its enhanced *version* is as follows:

- Generate a large number of candidate enhancement parameters [3] by densely sampling the entire range of image parameters. Computational complexity may be reduced by applying heuristic criteria such as, densely sampling only near the parameter space of most similar training images.

[1] `http://www.kpcb.com/internet-trends`

[2] I call the images before enhancement as low-quality and those after enhancement as high-quality in the rest of this chapter. The process of enhancing a new image is called "the testing stage".

[3] The brightness, saturation and contrast are referred to as "parameters" of an image in this chapter.

- Apply these candidate parameters to the original low-quality image to create a set of candidate images.
- Perform feature extraction on every candidate image and then compute its aesthetic score by using the learned model.
- Present the highest-scoring image to the user.

There are two obvious drawbacks for the above strategy. First, generating and applying a large number of candidate parameters to create candidate images may be computationally prohibitive even for low-dimensional parameter space. For example, a space of three parameters where each parameter $\in \{0, ..., 9\}$ produces 10^3 combinations. Second, even if creating candidate images is efficient, extracting features from them is always computationally intensive and is the bottleneck. Also, such heuristic methods need constant interaction with the training database (which might be stored on a server) that makes the parameter prediction sub-optimal. All these factors contribute to making the testing stage inefficient.

My approach assumes that a model quantifying image aesthetics has already been learned and instead focuses on finding a structured approach to enhancement parameter prediction. During training, the model learns the inter-relationship between the low-quality images, its features, its parameters and the high-quality enhancement parameters. During the testing stage, the model only has access to a new low-quality image, its features, parameters and the learned model and it have to predict the enhancement parameters. Using these enhancement parameters, the model can generate the candidate images and select the best one using the learned model. The stringent requirement of not accessing the training images arises from real-world requirements. For example, to enhance a single image, it would be inefficient to establish a connection with the training database, generate hundreds of candidate images, perform feature extraction on them and then find the best image.

The search space spanned by the parameters is huge. However, the enhancement

parameters are not randomly scattered. Instead they depend on the parameters and features of the original low-quality image. Thus I hypothesize that the enhancement parameters should have a low-dimensional structure in another latent space. I employ an MF-based approach because it allows expressing the enhancement parameters in terms of three latent variables, which model the interaction across: 1. the low-quality images 2. their corresponding enhancement parameters 3. the low-quality parameters. The latent factors are learned during inference by Gibbs sampling. Additionally, I need to incorporate the low-quality image features since the enhancement parameters also depend on the color composition of the image, which can be characterized by the features. The feature incorporation in this framework is achieved by representing the latent variable which models the interaction across these images as a linear combination of their features, by solving a convex $\ell_{2,1}$-norm problem. I show that the proposed approach outperforms the heuristic approaches as well as the recent approaches in MF and structured prediction on synthetic as well as on the real-world data of image enhancement. I review the related work on MF as well as image enhancement in the following section.

3.2 Related Work

Automated image enhancement has recently been an active research area. Various solutions have been proposed for this task. I review those works which aim to improve the visual appeal of an image using automated techniques. A novel tone-operator was proposed to solve the tone reproduction problem (Reinhard *et al.*, 2002). A database named MIT-Adobe FiveK of corresponding low and high-quality images was published in (Bychkovsky *et al.*, 2011). They also proposed algorithm to solve the problem of global tonal adjustment. The tone adjustment problem only manipulates the luminance channel. In (Joshi *et al.*, 2010), an approach was presented, focusing on correcting images containing faces. They built a system to align faces between a "good" and a "bad" photo and then use the good

faces to correct the bad ones.

Content-aware enhancement approaches have been developed which aim to improve a specific image region. Some examples of such approaches are (Berthouzoz *et al.*, 2011; Kaufman *et al.*, 2012). A drawback of these is the reliance on obtaining segmented regions that are to be enhanced, which itself may prove difficult. Pixel-level enhancement was performed by using local scene descriptors. First, images similar to the input are retrieved from the training set. Then for each pixel in the input, a set of pixels was retrieved from the training set and they were used to improve the input pixel. Finally, Gaussian random fields are used to maintain the spatial smoothness in the enhanced image. This approach does not take the global information of an image into account and hence the local adjustments may not look right when viewed globally. A deep-learning based approach was presented in (Yan *et al.*, 2014c). In (Kang *et al.*, 2010), users were required to enhance a small amount of images to augment the current training data.

Two closely related and recent works involve training a ranking model from low and high-quality image pairs (Yan *et al.*, 2014a; Chandakkar *et al.*, 2015a). In a recent state-of-art method (Yan *et al.*, 2014a), a dataset of 1300 corresponding image pairs was reported, where even the intermediate enhancement steps are recorded. A ranking model trained with this information can quantify the (enhancement) quality of an image. In (Chandakkar *et al.*, 2015a), non-corresponding low and high-quality image pairs were used to train a ranking model. Both the approaches use kNN search at the test time to create a pool of candidate images first. After extracting features and ranking all of them, the best image is presented to the user.

The task of enhancement parameter prediction could be related to the attribute prediction (Parikh and Grauman, 2011a; Parikh *et al.*, 2012; Li *et al.*, 2013; Chen *et al.*, 2014). However, the goal of the work on attribute prediction has been to predict relative strength of an attribute in the data sample (or image). I am not aware of any work of 2015 that predicts parameters

of an enhanced version of a low-quality image given only the parameters and features of that image. Since my approach is based on MF principles, I review the recent related work on MF.

MF (Rennie and Srebro, 2005; Mnih and Salakhutdinov, 2007; Salakhutdinov and Mnih, 2008; Lawrence and Urtasun, 2009; Xiong *et al.*, 2010) is extensively used in recommender systems (Ma *et al.*, 2008; Baltrunas *et al.*, 2011; Ma *et al.*, 2011; Wang *et al.*, 2015; Marlin *et al.*, 2012; Song *et al.*, 2015; Shi *et al.*, 2014). These systems predict the rating of an item for a user given his/her existing ratings for other items. For example, in Netflix problem, the task is to predict favorite movies based on user's existing ratings. MF-based solutions exploit following two key properties of such user-item rating matrix data. First, the preferred items by a user have some similarity to the other items preferred by that user (or by other similar users, if we have sufficient knowledge to build a similarity list of users). Second, though this matrix is very high-dimensional, the patterns in that that matrix are structured and hence they must lie on a low-dimensional manifold. For example, there are $17,770$ movies in Netflix data and ratings range from $1-5$. Thus, there are 5^{17770} rating combinations possible per user and there are $480,189$ users. Therefore, the number of actual variations in the rating matrix should be a lot smaller than the number of all possible rating combinations. These variations could be modeled by latent variables lying near a low-dimensional manifold. This principle is formalized in (Mnih and Salakhutdinov, 2007) with probabilistic matrix factorization (PMF). It hypothesizes that the rating matrix can be decomposed into two latent matrices corresponding to user and movies. Their dot product should give the user-ratings. This works fairly well on a large-scale data-set such as Netflix. However, a lot of parameters have to be tuned. This requirement is alleviated in (Salakhutdinov and Mnih, 2008) by developing a Bayesian approach to MF (BPMF). BPMF has been extended for temporal data (BPTF) in (Xiong *et al.*, 2010). MF is used in other domains such as computer vision to predict feature vectors of another viewpoint of a person given a feature for one

viewpoint (Chen and Grauman, 2014). I adopt and modify BPTF since it allows us to model joint interaction across low-quality images, corresponding enhancement parameters and the low-quality parameters.

3.3 Problem Formulation

We have a training set consisting of N images $\{\boldsymbol{S}_1, \ldots, \boldsymbol{S}_N\}$ [4] . Parameters of all images are represented as $\boldsymbol{A} = \{A_1, \ldots, A_N\}$ where $A_i \in \mathbb{R}^{K \times 1} \; \forall \; i \in \{1, \ldots, N\}$. Each image has M enhanced versions and each version has the same size as that of its corresponding low-quality image. All versions corresponding to the i^{th} image are represented as $\{\boldsymbol{W}_i^1, \ldots, \boldsymbol{W}_i^M\}$. All versions are of higher quality as compared to its corresponding image. Parameters of all M versions of the i^{th} image (also called as candidate parameters) are represented as $\boldsymbol{A'} = \{A_i'^1, \ldots, A_i'^M\}$, where $A_i'^j \in \mathbb{R}^{K \times 1} \; \forall \; i, j$. Features of all low-quality images are represented as $\boldsymbol{F} = \{F_1, \ldots, F_N\}$ where $F_i \in \mathbb{R}^{L \times 1} \; \forall \; i$. In practice, I observe that $M \ll N, K < M$. The goal of this work is to predict the candidate parameters for all the versions of the i^{th} image by only using the information provided by A_i and F_i. To the best of my knowledge, this is a novel problem of real significance that has not been addressed in the literature.

3.4 Proposed Approach

The task is to predict the candidate parameters for all the enhanced versions of a low-quality image with the help of its parameters and features. The values for all the K parameters corresponding to N images and their $N \cdot M$ versions (total $N + N \cdot M$) can be stored in three-dimensional matrix $\mathbf{R} \in \mathbb{R}^{N \times (M+1) \times K}$. We need to predict $\hat{R}_{ij}^k = R_i^k + \Delta R_{ij}^k$

[4] In this chapter, I use bold letters to denote matrices. Non-bold letters denote scalars/vectors which will either be clear from the context or will be mentioned. $X^i, X_i, \boldsymbol{X}^T, X_{ij}$ and $||\boldsymbol{X}||_p$ denote row, column, transpose, entry at row i and column j of a matrix $\boldsymbol{X}$ and p^{th} norm of matrix $\boldsymbol{X}$ respectively.

or in turn just ΔR_{ij}^k. R_i^k denotes the k^{th} parameter value ($k \in \{1, \ldots, K\}$) of the i^{th} low-quality image and $\hat{R}_{ij}^k$ is the k^{th} parameter value of j^{th} version of the i^{th} image. Given a new n^{th} low-quality image, we only need to predict $\Delta R_{nj}^k \; \forall \; j = \{1, \ldots, M\}, \forall \; k$.

During training, one can compute ΔR_{ij}^k from available R_{ij}^k and $\hat{R}_{ij}^k$. Following MF principles, I express $\mathbf{\Delta R}$ as an inner product of three latent factors, $\mathbf{U} \in \mathbb{R}^{D \times N}$, $\mathbf{V} \in \mathbb{R}^{D \times M}$ and $\mathbf{T} \in \mathbb{R}^{D \times K}$ (Salakhutdinov and Mnih, 2008; Xiong *et al.*, 2010). D is the latent factor dimension. These latent factors should presumably model the underlying low-dimensional subspace corresponding to the low-quality images, its enhanced versions and its parameters. This can be formulated as:

$$\Delta R_{ij}^k =< U_i, V_j, T_k > \equiv \sum_{d=1}^{D} U_{di} V_{dj} T_{dk}, \tag{3.1}$$

where U_{di} denotes the d^{th} feature of the i^{th} column of $\mathbf{U}$. Presumably, as one increases D, the approximation error $\Delta R_{ij}^k - < U_i, V_j, T_k >$ should decrease (or stay constant) if the prior distributions for latent factors $\mathbf{U}, \mathbf{V}$ and $\mathbf{T}$ are chosen correctly. The following paragraph provides some details on how to choose the proper prior distributions for the parameters.

Prior Distributions: The prior distributions on $\mathbf{U}, \mathbf{V}$ and $\mathbf{T}$ are chosen as normal distributions. I also consider a normal distribution to model the randomness in the attribute difference values $\Delta \mathbf{R}$. The details are as follows:

$$\begin{aligned}
&p(\Delta \mathbf{R} | \mathbf{U}, \mathbf{V}, \mathbf{T}, \alpha) = \mathcal{N}_1(< U_i, V_j, T_k >, \alpha^{-1}) \\
&U_i \sim \mathcal{N}_D(0, \sigma_U^2 \mathbf{I_D}), \; \forall \; i = \{1, \ldots, N\} \\
&V_j \sim \mathcal{N}_D(0, \sigma_V^2 \mathbf{I_D}), \; \forall \; j = \{1, \ldots, M\} \\
&T_k \sim \mathcal{N}_D(0, \sigma_T^2 \mathbf{I_D}), \; \forall \; k = \{1, \ldots, K\},
\end{aligned} \tag{3.2}$$

where α is precision, $\mathbf{I_D}$ is a $D \times D$ identity matrix, $\mathcal{N}_Z(\mu, \mathbf{\Lambda})$ is a Z-dimensional multivariate Gaussian distribution with Z-dimensional mean vector μ and a $Z \times Z$ covariance

matrix $\mathbf{\Lambda}$. For both simulation and enhancement experiment, I use $\alpha = 2$, $\sigma_U^2 = \sigma_V^2 = \sigma_T^2 = 0.01$.

I choose prior distributions for the hyper-priors.

$$
\begin{aligned}
p(\alpha) &= \mathcal{W}(\alpha|\tilde{W}_0, \tilde{\nu}_0), \\
p(\Theta_U) &= p(\mu_U|\mathbf{\Lambda}_U) \cdot p(\mathbf{\Lambda}_U) \cdot \mathcal{N}(\mu_0, (\beta_0\mathbf{\Lambda}_U)^{-1}) \cdot \\
&\quad \mathcal{W}(\mathbf{\Lambda}_U|\mathbf{W}_0, \nu_0), \\
p(\Theta_V) &= p(\mu_V|\mathbf{\Lambda}_V) \cdot p(\mathbf{\Lambda}_V) \cdot \mathcal{N}(\mu_0, (\beta_0\mathbf{\Lambda}_V)^{-1}) \cdot \\
&\quad \mathcal{W}(\mathbf{\Lambda}_V|\mathbf{W}_0, \nu_0), \\
p(\Theta_T) &= p(\mu_T|\mathbf{\Lambda}_T) \cdot p(\mathbf{\Lambda}_T) \cdot \mathcal{N}(\mu_0, (\beta_0\mathbf{\Lambda}_T)^{-1}) \cdot \\
&\quad \mathcal{W}(\mathbf{\Lambda}_T|\mathbf{W}_0, \nu_0).
\end{aligned}
\tag{3.3}
$$

Here, $\mathcal{W}$ is the Wishart distribution of a $D \times D$ random matrix $\mathbf{\Lambda}$ with ν_0 degrees of freedom and a $D \times D$ scale matrix $\mathbf{W}_0$. Wishart distribution is chosen since it is a conjugate prior for multivariate normal distribution (with precision matrix). The parameters in the hyper-priors: $\mu_0, \beta_0, \mathbf{W}_0, \nu_0, \tilde{W}_0$ and $\tilde{\nu_0}$ are treated as constants during training. They are set using prior knowledge of the application. For both experiments, I use: $\mu_0 = 0, \beta_0 = 1, \mathbf{W}_0 = \mathbf{I}_D, \nu_0 = D, \tilde{W}_0 = 1, \tilde{\nu}_0 = 1$. The Bayesian formulation of the factorization adjusts the parameters within a reasonable range.

The latent factors $\mathbf{U}, \mathbf{V}$ and $\mathbf{T}$ are found by doing inference through Gibbs sampling. It will sample each latent variable from its distribution, conditional on the values of other variables. The predictive distribution for ΔR_{ij}^k is found by using Monte-Carlo approximation (explained later). However, it is important to note the following major differences in the proposed task when compared with the previous work on MF (Salakhutdinov and Mnih, 2008; Xiong *et al.*, 2010). In product or movie rating prediction problems, an average (non-personalized) recommendation may be provided to a user who has not provided any

preferences (not necessarily constant for all users). For the proposed task, each image may require a different kind of parameter adjustment to create its enhanced version and thus no "average" adjustment exists. The adjustment should depend on the image's features that characterize the image (e.g. bright vs. dull, muted vs. vibrant). In this task, it is particularly difficult to get a good generalizing performance on the testing set as illustrated later. The loss in performance of existing approaches on the testing set can be attributed to the different requirements for parameter adjustments for each image. Thus it becomes necessary to include the information obtained from image features into the formulation. I show that simply concatenating the parameters and features and applying MF techniques presented in (Salakhutdinov and Mnih, 2008; Xiong *et al.*, 2010) does not provide good performance, possibly because they lie in different regions of the feature space.

To overcome this problem, I observe that the conditional distribution of each U_i factorizes with respect to the individual samples. I propose to express $\mathbf{U}$ as a linear function of $\mathbf{F}$ by using a convex optimization scheme. I integrate it into the inference algorithm to find out the latent factors. The linear transformation can be expressed as,

$$U_i = F_i^T \mathbf{P} + Q, \; \forall \, i \in \{1, \ldots, N\}, \tag{3.4}$$

where $F_i \in \mathbb{R}^{L \times 1}, U_i \in \mathbb{R}^{D \times 1}, \mathbf{P} \in \mathbb{R}^{D \times D}$ and $Q \in \mathbb{R}^{1 \times D}$. Note that to carry out this decomposition, I have to set $D = L$. This is not a severe limitation since L is usually large (~ 1000) and as I have mentioned before, increasing D should decrease the approximation error at the cost of increased computation. Henceforth I assume that the feature extraction process generates $F_i \in \mathbb{R}^{D \times 1}$. Also, note that large L does not mean that the latent space is no longer low-dimensional, because L is still smaller as compared to all the possible combinations of parameters (e.g. 5^{17770}).

I propose an iterative convex optimization process to determine coefficients $\mathbf{P}$ and Q of Equation 3.4. I propose the following objective function to determine them:

$$\min_{\mathbf{P},Q} \sum_{i=1}^{N} ||F_i^T \mathbf{P} + Q - U_i^T||_2 + \beta||\mathbf{P}||_{2,1} + \gamma||Q||_2 \tag{3.5}$$

The objective function tries to reconstruct $\mathbf{U}$ using $\mathbf{P}, Q$ and F while controlling the complexity of coefficients. Let us concentrate on the structure of $\mathbf{P}$ (by neglecting the effect of Q momentarily). The columns of $\mathbf{P}$ act as coefficients for F_i. Ideally, we would want the elements of U_i to be determined by a sparse set of features, which implies sparsity in the columns of $\mathbf{P}$. To this end, I impose $\ell_{2,1}$-norm on $\mathbf{P}$, which gives a block-row structure for $\mathbf{P}$.

Let us consider the structure of Q along with $\mathbf{P}$. Equation 3.4 shows that different columns of U_i depend on different image features F_i. Also, a different set of columns of $\mathbf{P}$ should get activated (take on large values) for different F_i. I add an offset $Q \in \mathbb{R}^{1\times D}$ for regularization. Thus the offset introduced by Q remains constant across all the images but changes for each $F_{i,j}$. Making Q to be a row vector also forces $\mathbf{P}$ to play a major role in Equation 3.5. This in turn increases the dependence of U_i on F_i. If I were to define Q as the same size of $\mathbf{U}$ (which would mean different offsets for each image as well as its features), it would pose two potential disadvantages. Firstly, optimal $\mathbf{P}$ and Q could be (trivially) obtained by just setting each entry of $\mathbf{P}$ to a very small value and letting a column of $Q \approx U_i$ (which makes F_i redundant). Secondly, while testing for a new image, I would have to devise a strategy to determine the suitable value for Q. For example, I could take the column of Q that corresponds to the nearest training image. This adds unnecessary complexity and reduces generalization. By making Q a row vector, I consider that it may be possible to arrive to the space of enhancement parameters by linearly transforming the low-quality image features with a constant offset. In other words, I want $\mathbf{P}$ to transform the features into a region in the latent space where all the other high-quality images lie and Q provides an offset to avoid over-fitting. This is a joint $\ell_{2,1}$-norm problem which can be solved efficiently by reformulating it as convex. I reformulate Equation 3.5 as follows,

inspired by (Nie *et al.*, 2010):

$$\min_{\mathbf{P},Q} \frac{1}{\beta}\sum_{i=1}^{N} ||F_i^T\mathbf{P} + Q - U_i^T||_2 + ||\mathbf{P}||_{2,1} + \frac{\gamma}{\beta}||Q||_2. \tag{3.6}$$

The $\ell_{2,1}$-Norm of a matrix $\mathbf{X} \in \mathbb{R}^{M\times N}$ is defined as, $\ell_{2,1}(\mathbf{X}) = \sum_{i=1}^{M} ||\mathbf{X}^i||_2$. Also, for a row vector Q, I have $||Q||_2 = ||Q||_{2,1}$. Thus Equation 3.6 can be further written as:

$$\min_{\mathbf{P},Q} \frac{1}{\beta}||\mathbf{F}^T\mathbf{P} + 1^N Q - \mathbf{U}^T||_{2,1} + ||\mathbf{P}||_{2,1} + \delta||Q||_{2,1}, \tag{3.7}$$

where $\delta = \frac{\gamma}{\beta}$ and 1^N is a column vector of ones $\in \mathbb{R}^N$. Now, put $\mathbf{F}^T\mathbf{P} + 1^N Q - \beta\mathbf{E} = \mathbf{U}^T$. Thus Equation 3.7 becomes:

$$\begin{aligned} &\min_{\mathbf{E},\mathbf{P},Q} ||\mathbf{E}||_{2,1} + ||\mathbf{P}||_{2,1} + \delta||Q||_{2,1}, \\ &\text{s.t. } \mathbf{F}^T\mathbf{P} + 1^N Q - \beta\mathbf{E} = \mathbf{U}^T, \\ &\min_{\mathbf{E},\mathbf{P},Q} \left|\left|\left|\begin{bmatrix}\mathbf{E}\\ \mathbf{P}\\ \delta Q\end{bmatrix}\right|\right|\right|_{2,1} \text{ s.t. } \begin{bmatrix}-\beta\mathbf{I}_N & \mathbf{F}^T & \delta^{-1}1^N\end{bmatrix}\begin{bmatrix}\mathbf{E}\\ \mathbf{P}\\ \delta Q\end{bmatrix} = \mathbf{U}^T \end{aligned} \tag{3.8}$$

Equation 3.8 is now in the form of: $\min_{\mathbf{X}} ||\mathbf{X}||_{2,1}$ s.t. $\mathbf{ZX} = \mathbf{B}$ and is convex. It can be iteratively solved by an efficient algorithm mentioned in (Nie *et al.*, 2010). I set $\beta = 0.1$ and $\delta = 3$. Once $\mathbf{U}$ has been expressed as a function of $\mathbf{F}$, I use Gibbs Sampling to determine the latent factors $\mathbf{P}, Q, \mathbf{V}$ and $\mathbf{T}$ (Salakhutdinov and Mnih, 2008). The predictive distribution for a new parameter value $\Delta\hat{R}_{ij}^k$ is given by a multidimensional integral as:

$$\begin{aligned} p(\Delta\hat{R}_{ij}^k|\Delta\mathbf{R}) = \int & p(\Delta\hat{R}_{ij}^k|U_i, V_j, T_k, \alpha)\cdot \\ & p(\mathbf{U},\mathbf{V},\mathbf{T},\alpha,\Theta_U,\Theta_V,\Theta_T|\Delta\mathbf{R})\cdot \\ & d(\mathbf{U},\mathbf{V},\mathbf{T},\alpha,\Theta_U,\Theta_V,\Theta_T). \end{aligned} \tag{3.9}$$

I resort to numerical approximation techniques to solve the above integral. To sample from the posterior, I use Markov Chain Monte Carlo (MCMC) sampling with Gibbs sampling as the algorithm. The integral can be approximated as,

$$p(\Delta\hat{R}_{ij}^{k}|\Delta\mathbf{R}) \approx \sum_{y=1}^{Y} p\left(\Delta\hat{R}_{ij}^{k}|U_i^{(y)}, V_j^{(y)}, T_k^{(y)}, \alpha^{(y)}\right). \tag{3.10}$$

Here I draw Y samples and the value of Y is set by observing the validation error. The following paragraph explains the sampling process for all the hyper-parameters in detail.

Conditional distributions in Gibbs Sampling: The joint posterior distribution can be factorized as:

$$\begin{aligned} p(\mathbf{U}, \mathbf{V}, \mathbf{T}, \alpha, \Theta_U, \Theta_V, \Theta_T|\Delta\mathbf{R}) \propto\ & p(\Delta\mathbf{R}|\mathbf{U}, \mathbf{V}, \mathbf{T}, \alpha)\cdot \\ & p(\mathbf{U}|\Theta_U) \cdot p(\mathbf{V}|\Theta_V)\cdot \\ & p(\mathbf{T}|\Theta_T) \cdot p(\Theta_U)\cdot \\ & p(\Theta_V) \cdot p(\Theta_T) \cdot p(\alpha). \end{aligned} \tag{3.11}$$

I derive the desired conditional distribution by substituting all the model components previously described.

Hyper-parameters: I use the conjugate prior for the parameter value precision α, I have that the conditional distribution of α given $\Delta\mathbf{R}, \mathbf{U}, \mathbf{V}$ and $\mathbf{T}$ follows the Wishart distribution:

$$\begin{aligned} p(\alpha|\Delta\mathbf{R}, \mathbf{U}, \mathbf{V}, \mathbf{T}) &= \mathcal{W}(\alpha|W_0^*, \nu_0^*), \\ \nu_0^* &= \tilde{\nu}_0 + \sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{k=1}^{K} I_{ij}^{k}, \\ (\tilde{W}_0^*)^{-1} &= \tilde{W}_0^{-1} + \\ & \sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{k=1}^{K}(\Delta R_{ij}^{k} - < F_i^T\mathbf{P}^* + Q^*, V_j^*, T_k^* >)^2, \end{aligned} \tag{3.12}$$

where $I_{ij}^k = 1$ if an attribute value ΔR_{ij}^k is present (not missing), otherwise $I_{ij}^k = 0$. Also, $\mathbf{U}^* = F_i^T \mathbf{P}^* + Q^*$. For $\Theta_U = \{\mu_U, \mathbf{\Lambda_U}\}$, I can integrate out all the random variables given in Equation 3.11 except $\mathbf{U}$ and obtain the Gaussian-Wishart distribution:

$$
\begin{aligned}
& p(\Theta_U|\mathbf{U}) = \mathcal{N}(\mu_U|\mu_0^*, (\beta_0^*\mathbf{\Lambda}_U)^{-1}) \cdot \mathcal{W}(\mathbf{\Lambda}_U|\mathbf{W}_0^*, \nu_0^*), \\
& \mu_0^* = \frac{\beta_0\mu_0 + N\bar{U}}{\beta_0 + N}, \beta_0^* = \beta_0 + N, \nu_0^* = \nu_0 + N; \\
& (\mathbf{W}_0^*)^{-1} = \mathbf{W}_0^{-1} + N\bar{\mathbf{S}} + \frac{\beta_0 N}{\beta_0 + N} \cdot (\mu_0 - \bar{U})(\mu_0 - \bar{U})^T, \\
& \text{where, } \bar{U} = \frac{1}{N}\sum_{i=1}^{N} U_i, \ \bar{\mathbf{S}} = \frac{1}{N}\sum_{i=1}^{N}(U_i - \bar{U})(U_i - \bar{U})^T.
\end{aligned}
\tag{3.13}
$$

Similarly, $\Theta_V = \{\mu_V, \mathbf{\Lambda}_V\}$ is conditionally independent of all other parameters given $\mathbf{V}$, and its conditional distribution has the form:

$$
\begin{aligned}
& p(\Theta_V|\mathbf{V}) = \mathcal{N}(\mu_V|\mu_0^*, (\beta_0^*\mathbf{\Lambda}_V)^{-1}) \cdot \mathcal{W}(\mathbf{\Lambda}_V|\mathbf{W}_0^*, \nu_0^*), \\
& \mu_0^* = \frac{\beta_0\mu_0 + N\bar{V}}{\beta_0 + N}, \beta_0^* = \beta_0 + N, \nu_0^* = \nu_0 + N; \\
& (\mathbf{W}_0^*)^{-1} = \mathbf{W}_0^{-1} + N\bar{\mathbf{S}} + \frac{\beta_0 N}{\beta_0 + N} \cdot (\mu_0 - \bar{V})(\mu_0 - \bar{V})^T, \\
& \bar{V} = \frac{1}{N}\sum_{i=1}^{N} V_i, \ \bar{\mathbf{S}} = \frac{1}{N}\sum_{i=1}^{N}(V_i - \bar{V})(V_i - \bar{V})^T.
\end{aligned}
\tag{3.14}
$$

Similarly, $\Theta_T = \{\mu_T, \mathbf{\Lambda}_T\}$ is conditionally independent of all other parameters given $\mathbf{T}$, and its conditional distribution has the form:

$$
\begin{aligned}
& p(\Theta_T|\mathbf{T}) = \mathcal{N}(\mu_T|\mu_0^*, (\beta_0^*\mathbf{\Lambda}_T)^{-1}) \cdot \mathcal{W}(\mathbf{\Lambda}_T|\mathbf{W}_0^*, \nu_0^*), \\
& \mu_0^* = \frac{\beta_0\mu_0 + N\bar{T}}{\beta_0 + N}, \beta_0^* = \beta_0 + N, \nu_0^* = \nu_0 + N; \\
& (\mathbf{W}_0^*)^{-1} = \mathbf{W}_0^{-1} + N\bar{\mathbf{S}} + \frac{\beta_0 N}{\beta_0 + N} \cdot (\mu_0 - \bar{T})(\mu_0 - \bar{T})^T, \\
& \bar{T} = \frac{1}{N}\sum_{i=1}^{N} T_i, \ \bar{\mathbf{S}} = \frac{1}{N}\sum_{i=1}^{N}(T_i - \bar{T})(T_i - \bar{T})^T.
\end{aligned}
\tag{3.15}
$$

Model Parameters: Firstly, I consider the latent example (data sample) features $\mathbf{U}$. Since its columns affect the example features independently, its conditional distribution factorizes w.r.t. individual U_i.

$$p(\mathbf{U}|\Delta\mathbf{R}, \mathbf{V}, \mathbf{T}, \alpha, \Theta) = \prod_{i=1}^{N} p(U_i|\Delta\mathbf{R}, \mathbf{V}, \mathbf{T}, \alpha, \Theta_U). \tag{3.16}$$

Then for each latent example feature vector $\mathbf{U}_i$,

$$\begin{aligned} &p(U_i|\Delta\mathbf{R}, \mathbf{V}, \mathbf{T}, \alpha, \Theta_U) = \mathcal{N}(U_i|\mu_i^*, (\mathbf{\Lambda}_i^*)^{-1}), \\ &\mu_i^* \equiv (\mathbf{\Lambda}_i^*)^{-1}(\mathbf{\Lambda}_U\mu_U + \alpha\sum_{j=1}^{M}\sum_{k=1}^{K} I_{ij}^k R_{ij}^k Y_{jk}) \\ &\mathbf{\Lambda}_i^* \equiv \mathbf{\Lambda}_U + \alpha\sum_{k=1}^{K}\sum_{j=1}^{M} I_{ij}^k Y_{jk} Y_{jk}^T, \end{aligned} \tag{3.17}$$

where $Y_{jk} \equiv V_j \cdot T_k$, which represents element-wise product between V_j and T_k.

Similarly, for each latent modified version feature V_j, I have:

$$\begin{aligned} &p(V_j|\Delta\mathbf{R}, \mathbf{U}, \mathbf{T}, \alpha, \Theta_V) = \mathcal{N}(V_j|\mu_j^*, (\mathbf{\Lambda}_j^*)^{-1}), \\ &\mu_j^* \equiv (\mathbf{\Lambda}_j^*)^{-1}(\mathbf{\Lambda}_V\mu_V + \alpha\sum_{i=1}^{N}\sum_{k=1}^{K} I_{ij}^k R_{ij}^k Y_{ik}) \\ &\mathbf{\Lambda}_j^* \equiv \mathbf{\Lambda}_V + \alpha\sum_{k=1}^{K}\sum_{j=1}^{M} I_{ij}^k Y_{ik} Y_{ik}^T, \end{aligned} \tag{3.18}$$

where $Y_{ik} \equiv (F_i^T\mathbf{P} + Q) \cdot T_k$

For each latent attribute feature T_k, I have:

$$
\begin{aligned}
&p(T_k|\Delta\mathbf{R}, \mathbf{U}, \mathbf{V}, \alpha, \Theta_T) = \mathcal{N}(T_k|\mu_k^*, (\mathbf{\Lambda}_k^*)^{-1}), \\
&\mu_k^* \equiv (\mathbf{\Lambda}_k^*)^{-1}(\mathbf{\Lambda}_T\mu_T + \alpha \sum_{i=1}^{N}\sum_{j=1}^{M} I_{ij}^k R_{ij}^k Y_{ij}) \\
&\mathbf{\Lambda}_k^* \equiv \Lambda_T + \alpha \sum_{k=1}^{K}\sum_{j=1}^{M} I_{ij}^k Y_{ij} Y_{ij}^T,
\end{aligned}
\tag{3.19}
$$

where $Y_{ij} \equiv (F_i^T\mathbf{P} + Q) \cdot V_j$

This illustrates how to sample all the hyper-parameters used in this chapter. This sampling process is repeatedly used by the Gibbs Sampling method presented in Algorithm 1.

Note that it is required in the algorithm to reconstruct $\mathbf{U}^{(y+1)}$ at every iteration since there will always be a small reconstruction error $||\hat{\mathbf{U}}^{(y+1)} - \mathbf{U}^{(y+1)}||$. The error occurs because Q is forced to be a row vector, which makes the exact recovery of $\mathbf{U}^{(y+1)}$ difficult. The reconstructed error causes adjustment of $\mathbf{V}$ and $\mathbf{T}$. Once the four latent factors are obtained, the next task is to predict the parameter values for M enhanced versions having K parameters each. Suppose F_t is the feature vector of a new image, then the parameter values $\Delta\hat{R}_{tj}^k$ can be simply obtained by computing, $\Delta\hat{R}_{tj}^k =< F_t^T\mathbf{P} + Q, V_j, T_k > \ \forall\, j \in \{1, \ldots, M\}$ and $k \in \{1, \ldots, K\}$. If the parameter value predictions lie beyond a certain range then a thresholding scheme can be used based on the prior knowledge. For example, to constrain the predictions between $[0, 1]$, a logistic function may be used.

3.5 Experiments and Results

I conduct two experiments to show the effectiveness of the proposed approach. I performed the first one on a synthetic data and compared it with: 1. BPMF 2. a discrete version of BPTF, called D-BPTF. 3. multivariate linear regression (MLR) 4. twin Gaussian processes (TGP) (Bo and Sminchisescu, 2010) 5. Weighted kNN regression (WKNN).

Algorithm 1 Gibbs Sampling for Latent Factor Estimation

Initialize model parameters $\{\mathbf{P}^{(1)}, Q^{(1)}, \mathbf{V}^{(1)}, \mathbf{T}^{(1)}\}$. Obtain $\left(\mathbf{U}^{(1)}\right)^T = \mathbf{F}^T\mathbf{P}^{(1)} + Q^{(1)}$

For $y = 1, 2, \ldots, Y$

- Sample the hyper-parameters according to the derivations [5]:

$$\alpha^{(y)} \sim p(\alpha^{(y)}|\mathbf{U}^{(y)}, \mathbf{V}^{(y)}, \mathbf{T}^{(y)}, \Delta\mathbf{R}),$$
$$\Theta_U^{(y)} \sim p(\Theta_U^{(y)}|\mathbf{U}^{(y)}), \;\; \Theta_V^{(y)} \sim p(\Theta_V^{(y)}|\mathbf{V}^{(y)}), \Theta_T^{(y)} \sim p(\Theta_T^{(y)}|\mathbf{T}^{(y)})$$

- For $i = 1, ..., N$, sample the latent features of an image (in parallel):

$$U_i^{(y+1)} \sim p(U_i|\mathbf{V}^{(y)}, \mathbf{T}^{(y)}, \Theta_U^{(y)}, \alpha^{(y)}, \Delta\mathbf{R})$$

Determine $\mathbf{P}^{(y+1)}$ and $Q^{(y+1)}$ using the iterative optimization by substituting $\mathbf{B} = \left(\mathbf{U}^{(y+1)}\right)^T$.

Reconstruct $\mathbf{U}^{(y+1)}$: $\left(\hat{\mathbf{U}}^{(y+1)}\right)^T = \mathbf{F}^T\mathbf{P}^{(y+1)} + Q^{(y+1)}$

- For $j = 1, ..., M$, sample the latent features of the enhanced versions (in parallel):

$$V_j^{(y+1)} \sim p(V_j|\hat{\mathbf{U}}^{(y+1)}, \mathbf{T}^{(y)}, \Theta_V^{(y)}, \alpha^{(y)}, \Delta\mathbf{R})$$

- For $k = 1, ..., K$, sample the latent features of parameter (in parallel):

$$T_k^{(y+1)} \sim p(T_k|\hat{\mathbf{U}}^{(y+1)}, \mathbf{V}^{(y+1)}, \Theta_T^{(y)}, \alpha^{(y)}, \Delta\mathbf{R})$$

To develop D-BPTF, I make minor modifications in the original BPTF approach (Xiong *et al.*, 2010) by removing the temporal constraints on their temporal variable, since there are no temporal constraints in this case. The inference for their temporal variable is then done in the exactly same manner as the other non-temporal variables. This gave a marginal boost in the performance. For MLR, I use a standard multivariate regression by maximum likelihood estimation method. Specifically, I use MATLAB's `mvregress` command. TGP is a generic structured prediction method. It accounts correlation between both input and output resulting in improved performance as compared to MLR or WKNN. The WKNN approach predicts the test sample as a weighted combination of the k-nearest inputs. The first

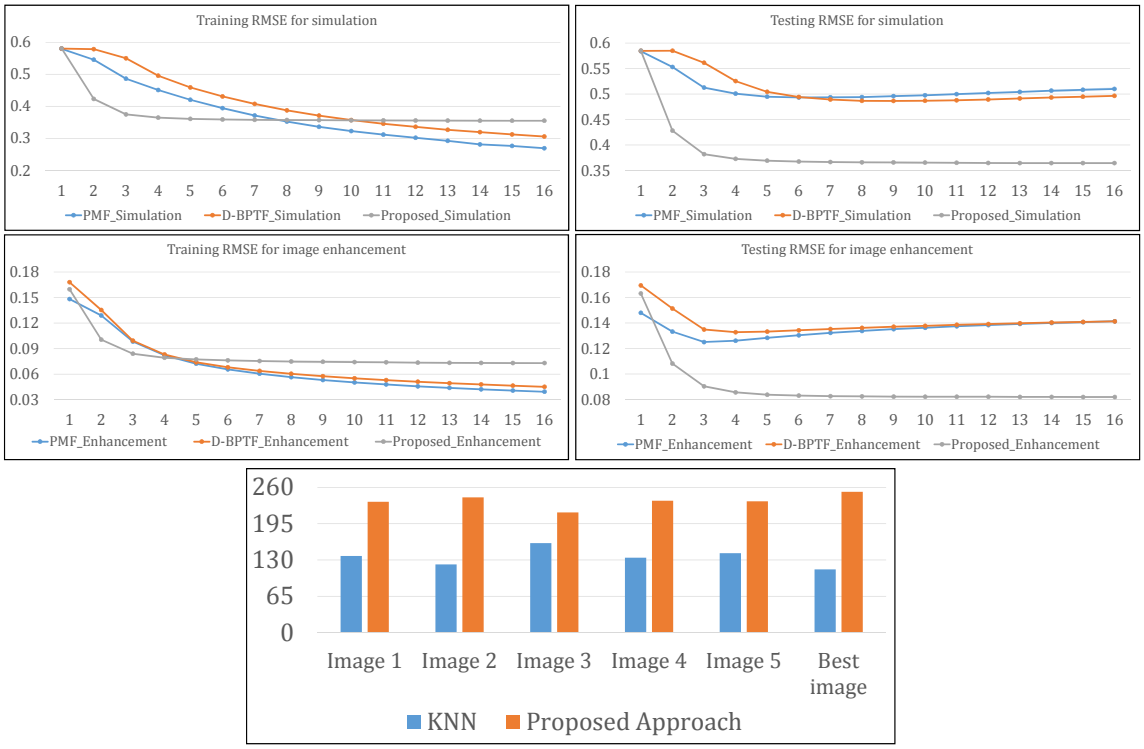


Figure 3.1: Top Plots: Train and Test RMSEs for Both the Experiments. Bottom Plot: First 5 Sets of Bars Show Votes for Version 1 to 5 of $k$nn Versus the Best Image of the Proposed Approach. The Last Set of Bars Shows Votes for the Best Image of Both Approaches. Please Zoom in for Better Viewing. See in Color

two algorithms do not allow features inclusion. For MLR, TGP and WKNN, I concatenate A_i and F_i, and use it to predict $A_i'^j$. Even for the proposed approach, I concatenate A_i and sample feature to form F_i. The intuition behind this concatenation is that the enhancement parameters should be a function of input parameters as well along with the features. I did observe performance boost after concatenating the features and parameters.

The second experiment demonstrates the usefulness of this approach in a real-world setting where one has to predict parameters of the enhanced versions of an image (then generate those versions by applying predicted parameters to the input low-quality image) without using any information about the versions. I compare the propsoed approach with the

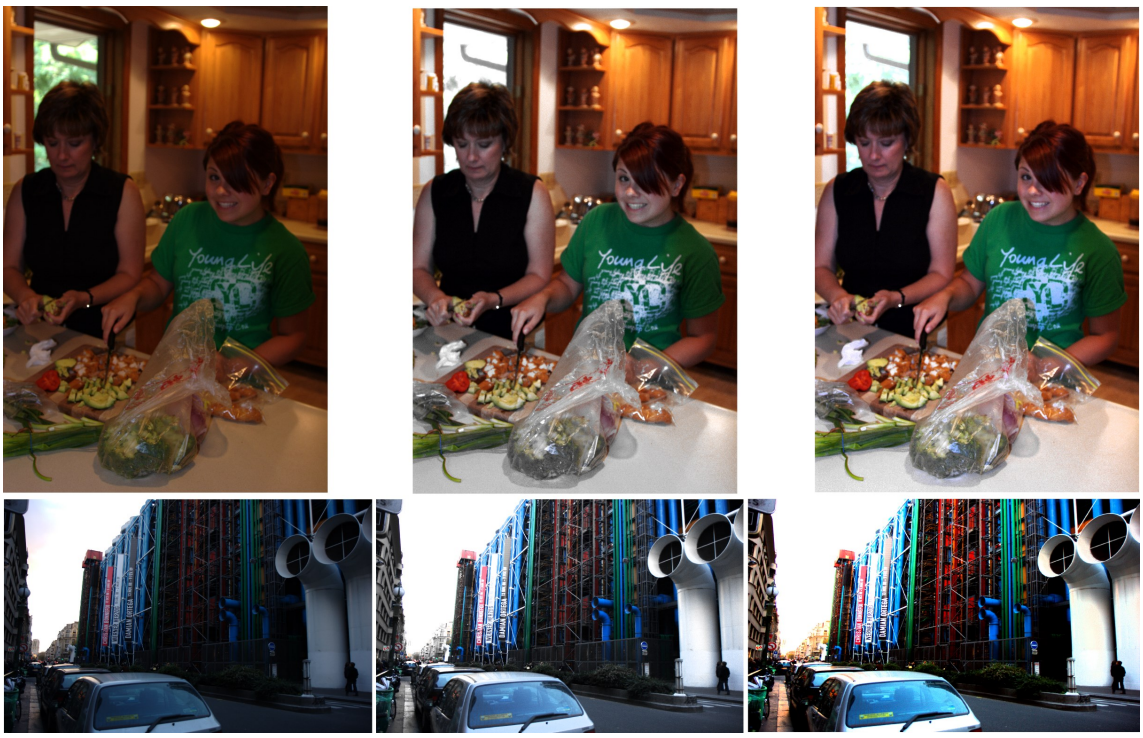


Figure 3.2: Left: Original Image, Middle: Enhanced Image by $k$nn and Right: Proposed Approach [6]. View in Color ©2016 IEEE.

competing 5 algorithms in addition to kNN-search as it is also used in (Yan *et al.*, 2014b; Chandakkar *et al.*, 2015b). I also analyzed the effect of Q in the proposed solution by: removing Q i.e. $\mathbf{U} = \mathbf{F}^T\mathbf{P}$.

3.5.1 Data set description and experiment protocol

The synthetic data is carefully constructed by keeping the following task in mind. I am given a training set consisting of: 1. $\mathbf{F} \in \mathbb{R}^{D\times N}$; 2. $\mathbf{A} \in \mathbb{R}^{K\times N}$; and 3. *only* parameters of M versions for each input sample - $\mathbf{A}' \in \mathbb{R}^{K\times N\times M}$. The goal of the task is to predict parameters for a set of M versions given a new F_i and A_i. In real-world problems, $\mathbf{A}$ and $\mathbf{F}$ are interdependent. The parameters of M versions are dependent on both $\mathbf{A}, \mathbf{F}$. Hence I construct the synthetic data as follows.

Firstly, I generate a set of 3-D input parameters - $\mathbf{A}$ - drawn from a uniform distribution $[0, 1]$. Then I generate a 50-D feature set $\mathbf{F}$, where each element of F_i is related to all $A_{k,i}\ \forall\ i = \{1, \ldots, 10^3\}, k = \{1, 2, 3\}$ by a nonlinear function. For example, $F_{j,i} = r_1^{A_{1,i}} + \frac{1}{1+e^{-r_2 A_{2,i}}} + A_{3,i}^{r_3}, \forall\, j \in \{1, \ldots, 50\}$ and r_1, r_2, r_3 are random numbers. The parameters of enhanced versions, $A'_{k,i,m}$, are also non-linearly related to $A_{k,i}\ \forall\ k, \forall\ m \in \{1, \ldots, 4\}$ and F_i. For example, $A'_{k,i,m} = \eta\left(r_1^{A_{1,i}} + \frac{1}{1+e^{-r_2 A_{2,i}}} + A_{3,i}^{r_3}\right) + (1-\eta)\cdot||F_i||_2$. The contribution of F_i is decided by η. I perform a 3-fold cross-validation. The values of $\mathbf{A}'$ are predicted for the test set (disjoint from training) using corresponding $\mathbf{A}$ and $\mathbf{F}$. RMSE is computed between the predicted and actual $\mathbf{A}'$.

The MIT-Adobe FiveK data-set contains 5000 high-quality photographs taken with SLR cameras. Each photo is then enhanced by five experts to produce 5 enhanced versions. I extract average saturation, brightness and contrast for every image, which are parameters $\in \mathbf{A}$. I also extract 1274-D color histogram with 26 bins for hue, 7 bins each for saturation and value. I calculate localized features of 144-D each for contrast, brightness and saturation. Finally, the average saturation, brightness and contrast of the input low-quality image are appended. These are also called as the parameters of an image. Thus I get a 1709-D $(= 1274 + 3 \times 144 + 3)$ representation for every image $\in \mathbf{F}$. I train using 4000 images and use 500 images each for validation and testing. The parameters are predicted for 5 versions in a 3×5 matrix for each image in the testing set. An entry $A'_{i,j}$ denotes the value for i^{th} parameter of j^{th} enhanced version. To enable comparison with the expert-enhanced images of the data-set, the parameters for 5 enhanced versions for each image are also computed, which I treat as ground-truth. I evaluate this experiment in two ways. Firstly, I calculate RMSE between the parameters of 5 expert-enhanced photos and the parameters of the predicted versions using five aforementioned algorithms. Secondly, I conduct a subjective test under standard test settings (constant lighting, position, distance from the screen). In this case, I compare my approach with the popular kNN-search-based approach. It first finds

the nearest original image in the training set to the testing image - $\mathbf{im}$ - and then applies the same parameter transformation to $\mathbf{im}$ to generate 5 version. In the proposed approach, the parameters for enhanced versions are predicted using the proposed formulation. I threshold the parameter values as:

$$\begin{aligned} A'_{k,i,m} &= \min(A'_{k,i,m}, A_{k,i} + \lambda_k A_{k,i}), \\ A'_{k,i,m} &= \max(A'_{k,i,m}, A_{k,i} - \zeta_k A_{k,i}), \end{aligned} \tag{3.20}$$

where λ and ζ are multipliers for the k^{th} parameter. In this case, the multipliers for saturation, brightness and contrast are: $\lambda = \{0.4, 0.4, 0.05\}, \zeta = \{0.3, 0.3, 0.01\}$. As mentioned before, the clipping scheme in the proposed formulation should be set using prior knowledge. Here, I know that the enhanced images usually have a larger increase (as compared to decrease) associated with their parameters. Also, changing contrast by a very small amount affects the image greatly.

The predicted parameters are applied to the input image to obtain its enhanced versions. The procedure is the same for both the approaches and is as follows. First I change contrast till the difference between the updated and the predicted contrast is marginal. I update contrast first since changing it updates both brightness and saturation. I then update brightness and saturation till they come significantly closer to their corresponding predicted values. This provides 5 versions for both approaches. To allow comparisons within a reasonable amount of time, I use a pre-trained ranking weight vector w (from (Chandakkar *et al.*, 2015b)) to select the best image of my approach (*im-proposed*) and kNN-approach (*im-kNN*). For the subjective test, people are told to compare *im-proposed* with the 5 enhanced versions of kNN-approach as well as with *im-kNN*. Thus for every input image, people perform 6 comparisons. The image order was randomized. I conducted the test with 11 people and 35 input images. Thus every person compared 210 pairs of images. They were told to choose a visually-appealing image. The third option of simultaneously preferring

both images was also provided. This option has no effect on cumulative votes.

3.5.2 Results

The parameters for the synthetic data were more accurately predicted by the proposed approach than BPMF, D-BPTF, MLR, TGP and WKNN. It is worth noting that though the training error continues to decrease for the proposed approach, BPMF and D-BPTF, the testing error starts increasing after only 5 and 8 iterations for BPMF and D-BPTF, respectively. However, testing error in the proposed approach decreases rapidly for 4 iterations and then it decreases very slowly for the next 12, as shown in Fig. 3.1. The RMSE on test set for BPMF, D-BPTF, MLR, TGP, WKNN and the proposed approach is $0.4933, 0.4865, 0.6293, 0.4947, 0.8014$ and 0.3644. The numbers show that my approach is able to effectively use the additional information provided by features and the interaction between $\mathbf{A}, \mathbf{F}$ and all versions to provide better prediction. On the other hand, BPMF and D-BPTF start over-fitting quickly due to lack of sample feature information while MLR and WKNN fail to model the complex interaction between variables. TGP performs better because of its ability to capture correlations between input and output. However, TGP still treats each version independently and thus its performance still falls short of the proposed approach.

In the second experiment, the RMSE for BPMF, D-BPTF, MLR, TGP, WKNN and the proposed approach is $0.1251, 0.1328, 1.2420, 0.1268, 0.1518$ and 0.0820 respectively. The testing error starts increasing after only 3 and 5 iterations for BPMF and D-BPTF, respectively. It is important to note that I do *not* use the clipping scheme mentioned in Equation 3.20 in order to do a fair comparison of RMSEs between all the five approaches and the proposed appraoch. For the subjective evaluation, Fig. 3.1 shows cumulative votes obtained for ours and the kNN-based approach for comparison between 5 images chosen by kNN and the best image chosen by the proposed approach. Fig. 3.1 also shows votes

Table 3.1: Effect of Varying β and δ ©2016 IEEE

Parameter setting	RMSE (lower the better)
$\beta = 0.001, \gamma = 6$	0.3162
$\beta = 0.01, \gamma = 6$	0.0962
$\beta = 0.02, \gamma = 0.1$	0.0907
$\beta = 0.2, \gamma = 0.05$	0.0930
$\beta = 0.8, \gamma = 0.05$	0.0872
$\beta = 0.1, \gamma = 0.3$	**0.0820**
$\beta = 0.1, \gamma = 0.8$	0.0821
$\beta = 0.1, \gamma = 2$	**0.0820**

obtained for the best images chosen by both approaches. Fig. 3.2 shows two input images enhanced by both the approaches. The top row of Fig. 3.2 shows that kNN reduces the saturation while increasing the brightness. The proposed approach balances both of them to obtain a more appealing image. In the bottom row, however, both approaches fail to produce aesthetic images as images become too bright. It is probably due to the portion of the sky in the input image. For both the images, most people prefer images enhanced by the proposed approach. Computationally, the proposed approach is superior than kNN. Complexity of my approach is independent of data-set size at testing time whereas kNN searches the entire data-set for the closet image and then applies its parameters.

I reconstructed $\mathbf{U} = \mathbf{F}^T\mathbf{P}$ and observed performance drop as it overfits. I get RMSE of 0.9305 and 0.3762 on enhancement and simulation data, respectively. I believe the real-world enhancement data has correlations naturally embedded in it unlike in synthetic data. Thus the performance drop is drastic in case of enhancement since the problem of

recovering $\mathbf{P}$ only from $\mathbf{U}$ and $\mathbf{F}$ is ill-posed.

I also analyzed the effect of varying β and δ. Since the proposed approach uses Bayesian probabilistic inference, small variations in β and δ do not significantly affect the performance. Table 3.1 lists the various parameter settings and its effect on the performance of the second experiment (i.e. image enhancement):

3.6 Discussion

In this chapter [7] , I introduced a novel problem of predicting parameters of enhanced versions for a low-quality image by using its parameters and features. I developed an MF-inspired approach to solve this problem. I showed that by modeling the interactions across low-quality images, its parameters and its versions, one can outperform five state-of-art models in structured prediction and MF. I proposed inclusion of feature information into the formulation through a convex $\ell_{2,1}$-norm minimization, which works in an iterative fashion and is efficient. Thus the proposed approach utilizes information which helps characterize input image. This leads to better generalization and prediction performance. Since other approaches do not model interdependence between image features and parameters of their corresponding enhanced versions, they start over-fitting quickly and produce an inferior prediction performance on the test set. Experiments on synthetic and real data demonstrated superiority of the proposed approach over other state-of-art methods.

The matrix-factorization based approaches are used for personalization purposes. However, the current image enhancement datasets are not suitable for personalized image enhancement. To that end, one would need access to the favorite images of a specific person. In other words, rows should correspond to the various people and the columns should contain the images. Each entry is the rating that a person provided for that image. This is

[7] Most of the material in this chapter has appered in (Chandakkar and Li, 2016). See the full credit statement in appendix.

similar to the Netflix challenge dataset (Bennett *et al.*, 2007) where rows contain users and the columns contain movies, and each entry is an integer rating. Current datasets have a variable number of anonymized ratings for each image. Thus favorite images of a person cannot be inferred from the available information. However, a structured dataset from the social media websites such as Facebook, Instagram and Flickr can propel the development of personalized image enhancement methods.

Chapter 4

TOWARDS UNIFIED, CONTENT-ADAPTIVE IMAGE ENHANCEMENT

4.1 Introduction

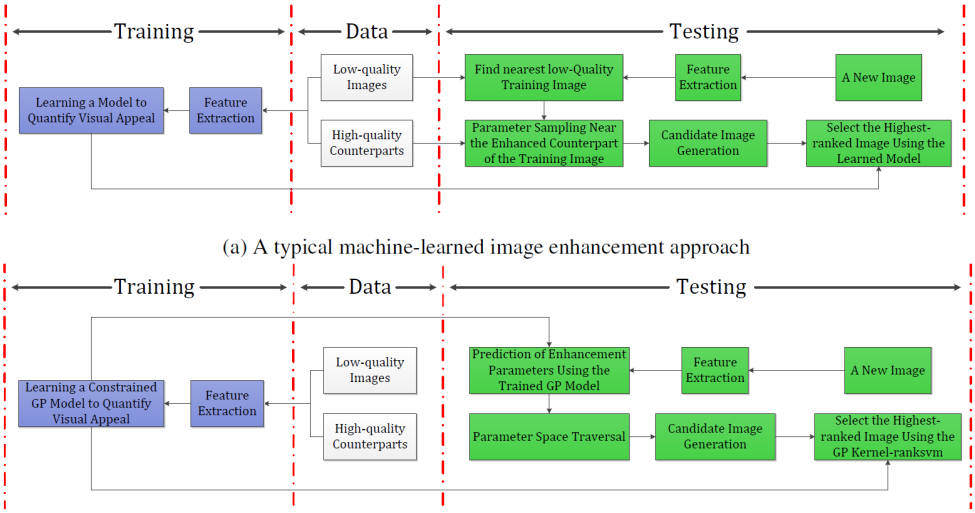
The previous chapter describes the motivation and the need for content-adaptive image enhancement methods. The size of the corpus of images on the Web is increasing at an exponential rate due to multiple people sharing their pictures on social media. Easy-to-use smart-phone cameras have played an equally significant role in this explosion of multimedia data. As a result, enhancing the captured images has become an essential feature for many social media websites and for smart-phones to have. Therefore, there is a need to modify the current content-adaptive image enhancement pipeline and make it fast and adaptable to users' needs.

The previous chapter describes a structured prediction technique for image enhancement parameters. It significantly reduces the amount of time spent in creating the enhanced versions of an image. It may also find better-enhanced versions due to the structured exploration of the parameter space. Though the approach alleviates the need to interact with the training set constantly, the enhancement process is still split into two stages:

1. Train a model to rank low and high-quality images.

2. Given a new image, predict the enhancement parameters using the image parameters and features.

I propose a Gaussian process (GP) based joint regression and ranking methodology that unifies these two pipelines. The comparison between the current approaches and the proposed approach is illustrated in Fig. 4.1.



(a) A typical machine-learned image enhancement approach

(b) Proposed approach. Note that our approach has no interaction with the training data during the testing stage

Figure 4.1: Pipelines of Image Enhancement Approaches ©2017 IEEE.

First, I model the problem as a joint regression and ranking problem. Given an image, multiple sets of enhancement parameters need to be predicted, which can be modeled as a regression problem. After obtaining the parameters, numerous enhanced versions need to be generated, and the highest-ranked image needs to be shown to the user. The proposed approach unifies the regression and the ranking using GPs. GP has widely been used as a regressor. The parameters of a GP kernel are determined from the training data. The proposed approach employs GP to predict the mean and the variance of the parameters of the enhanced versions. During training, it takes the feature vector of the original (low-quality) image, its parameters as the input. The parameters of the enhanced image are the target variables for the GP regressor.

The parameters predicted also need to be ranked. With the availability of a suitable ranking system, the top-ranked parameter set can be used to generate a single enhanced version which can be directly shown to the user. This saves a lot of time since multiple enhanced versions need not be generated.

To achieve this, I train a ranking model on the GP-covariance-kernel-induced feature

space. I develop a dual form of ranking SVM (Joachims, 2002) and replace that kernel with the GP kernel. Thus the same GP kernel regresses to the target enhancement parameters as well as ranks them. The GP kernel builds a mapping between the image feature space and the enhanced parameter space. It automatically learns a weighting strategy for image features so that more weight can be assigned to the image features that are crucial for making a higher-quality image. I put an additional constraint that all the low-quality images should be clustered together in the GP-kernel-induced feature space. Similar constraints are placed on high-quality images. This facilitates a structured exploration of the parameter space in the GP-kernel feature space.

The process to enhance a new image is as follows:

1. The GP model predicts the target values of the enhancement parameters and sorts them by their ranks. It also provides their mean and variance. This significantly reduces the computation since the model does not interact with the training set at all.

2. By using the mean and variance values, I generate some enhanced versions by applying parameters that lie k standard deviations away from the mean value. Here, k is a user-defined parameter and increasing it will result in more images being shown to the user. It can also be changed on-the-fly so that user can override the model ranking and choose a lower ranked image as the best-enhanced version. In the future, this feedback can be used to improve the overall pipeline.

I perform extensive experiments to illustrate the benefits of the proposed approach. It is computationally efficient during the testing phase. GP model provides a high-quality prediction of the target parameters. It also correctly predicts the ranking order between the new images and its enhanced counterparts. I perform subjective tests and quantitative analyses to show the effectiveness of the proposed approach.

4.2 Related Work

Content-adaptive image enhancement has become a topic of active research in the past few years. The previous chapter covered the literature on enhancement techniques. In this chapter, we cover techniques related to Gaussian processes that are relevant in this context.

The strength of a GP lies in learning in complex mappings between several variables using a small amount of data (in the order of several hundred) (Urtasun and Darrell, 2007). GP-based view-invariant face recognition was presented in (Eleftheriadis *et al.*, 2015). A GP latent-variable model was used to learn a discriminative feature space using LDA prior. In that feature space, examples from similar classes form clusters. In (Rudovic *et al.*, 2010), GP regression builds a mapping between the non-frontal facial points and the frontal view. These projected frontal view-points can be used by the facial expression methods. Facial expression recognition performance can be further improved by employing coupled GPs to capture dependencies between the mappings learned between non-frontal and frontal poses (Rudovic *et al.*, 2013).

As in the previous chapter, the considered parameters for enhancement are brightness, saturation, and contrast of an image. I describe the proposed approach in the following section.

4.3 Proposed Approach

The task involves prediction of the set of image parameters that enhances a given image. The proposed approach should simultaneously achieve the following two objectives: 1. Probabilistic estimation of the parameters from a given low-quality image feature: These predicted parameters should generate the enhanced counterpart. 2. The predicted parameters should be ranked in the GP-kernel-induced feature space: This would allow structured exploration in the parameter space and thereby discover the features essential for making an

enhanced, higher-quality image.

The training data contains pairs of low and high-quality images along with their parameters. Features of N low-quality images are represented by $\boldsymbol{F} = \{\boldsymbol{f}_1, \boldsymbol{f}_2, \ldots, \boldsymbol{f}_N\}$ [1] . There exist p high-quality versions for a given low-quality image in the database. Its features are represented by $\boldsymbol{F}^+ = \{\boldsymbol{F}_1^+, \ldots, \boldsymbol{F}_N^+\}$, where $\boldsymbol{F}_i^+ = \{\boldsymbol{f}_{i1}^+, \ldots, \boldsymbol{f}_{ip}^+\}$, and $\boldsymbol{f}_i, \boldsymbol{f}_{ij}^+ \in \mathbb{R}^{D\times 1}\ \forall\ i, j$. There also exist p sets of high-quality parameters for a given low-quality image. For simplicity of illustration, I predict parameters only for the first set. It should be noted that all the p sets of high-quality images are used to train a ranking model. The parameter sets for low and high-quality images are represented by $\boldsymbol{Y} = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N\}$ and $\boldsymbol{Y}^+ = \{\boldsymbol{y}_1^+, \ldots, \boldsymbol{y}_N^+\}$ respectively. Three image parameters were used to enhance an image, namely, brightness, contrast and saturation, hence $\boldsymbol{y}_i, \boldsymbol{y}_i^+ \in \mathbb{R}^{3\times 1}\ \forall\ i$. The task is to predict $\boldsymbol{y}_i^+$ from $\boldsymbol{f}_i$ and $\boldsymbol{y}_i$. I predict each parameter using a separate GP. I concatenate the m^{th} parameter of all low and high-quality images to form $\bar{\boldsymbol{y}}_m = (y_{1m}, \ldots, y_{Nm})^T$ and $\bar{\boldsymbol{y}}_m^+ = (y_{1m}^+, \ldots, y_{Nm}^+)^T$, respectively and train a separate GP model that predicts a single parameter.

4.3.1 GP Regression

GPs define a prior distribution over functions that becomes a posterior over functions after observing the data. GPs assume that this distribution over functions is jointly Gaussian that has a positive definite covariance kernel. GPs provide well-calibrated, probabilistic outputs (Murphy, 2012). This property plays a vital role in our application. The prior on the regression function is a GP, and it is represented as: $GP(m(\boldsymbol{f}), \kappa(\boldsymbol{f}, \boldsymbol{f}'))$ where $\boldsymbol{f}$ and $\boldsymbol{f'}$ are image features $\in \mathbb{R}^{D\times 1}$, $m(\boldsymbol{f})$ is a mean function and $\kappa(\boldsymbol{f}, \boldsymbol{f}')$ is a covariance function. The posterior predictive density for a single test input can be written as:

[1] In this chapter, I represent vectors by lower-case bold letters. Matrices are represented by upper-case bold letters. Scalars are denoted by non-bold letters.

$$p(\bar{y}^{+}_{*m}|\boldsymbol{f}_*, \boldsymbol{F}, \boldsymbol{Y}) = \mathcal{N}(\bar{y}^{+}_{*m}|\boldsymbol{k}_*^T \boldsymbol{K}_y^{-1} \bar{\boldsymbol{y}}^{+}_m, k_{**} - \boldsymbol{k}_*^T \boldsymbol{K}_y^{-1} \boldsymbol{k}_*) \tag{4.1}$$

where $\boldsymbol{k}_* = [\kappa(\boldsymbol{f}_*, \boldsymbol{f}_1), \ldots, \kappa(\boldsymbol{f}_*, \boldsymbol{f}_N)]$, N is the number of samples, $k_{**} = \kappa(\boldsymbol{f}_*, \boldsymbol{f}_*)$ and $\boldsymbol{K}_y = \boldsymbol{K} + \sigma_y^2 \boldsymbol{I}_N$. $\boldsymbol{K}$ is a kernel function between all training inputs $\boldsymbol{f}$, and $(\cdot)_*$ denotes a new data point. The noise variance σ_y^2 accommodates the real-world uncertainty.

The log-likelihood function of a GP regression model can be derived by using a standard multivariate Gaussian distribution. It is as follows:

$$\begin{aligned} \log p(\bar{\boldsymbol{y}}^{+}_m|\boldsymbol{F}) = -0.5 \left(\bar{\boldsymbol{y}}^{+}_m\right)^T \boldsymbol{K}_y^{-1} \bar{\boldsymbol{y}}^{+}_m - 0.5 log|\boldsymbol{K}_y| - \\ 0.5 N log(2\pi) \end{aligned} \tag{4.2}$$

I choose a standard squared exponential kernel for this task. It is as follows:

$$\kappa(\boldsymbol{f}_i, \boldsymbol{f}_j) = \sigma_f^2 \exp(-\frac{1}{2}(\boldsymbol{f}_i - \boldsymbol{f}_j)^T \cdot \boldsymbol{\Lambda} \cdot (\boldsymbol{f}_i - \boldsymbol{f}_j)) + \sigma_y^2 \delta_{ij} \tag{4.3}$$

The parameter σ_f^2 controls the vertical scale of the regression function, σ_y^2 models uncertainty, $\boldsymbol{\Lambda}$ is a diagonal matrix with entries $\{\theta_1, \ldots, \theta_D\}$ and δ_{pq} is a Kronecker delta function that takes the value 1 if $p = q$ and zero elsewhere. The $\boldsymbol{h} = \{\sigma_f^2, \boldsymbol{\Lambda}, \sigma_y^2\}$ are hyper-parameters. The prediction in Equation 4.1 is dependent on the kernel and in turn on the hyper-parameters: σ_f, $\boldsymbol{\Lambda}$ and σ_y^2. The procedure of obtaining optimal hyper-parameters is described later.

4.3.2 GP Ranking

By building a ranking relation in the GP-kernel-induced feature space, the GP kernel can determine the subset of features responsible for enhancing an image. It assigns higher weight to such features by adjusting the hyper-parameters. The primal form of rank SVM (Joachims, 2002) is given by:

$$\min_{w, \xi_{ij}} \frac{1}{2} \boldsymbol{w}^T \boldsymbol{w} + C \sum_{i,j} \xi_{ij}, \text{ subject to: } \boldsymbol{u}_i \succ \boldsymbol{u}_j \; \forall \; (i, j) \tag{4.4}$$

where $\boldsymbol{u}_i \succ \boldsymbol{u}_j$ indicates that $\boldsymbol{u}_i$ is ranked higher than $\boldsymbol{u}_j$.

In one of my previous papers, I have observed that a learned mapping between low and high-quality images alone does not ensure high ranking accuracy on new images. The enhanced images are often characterized by high saturation, brightness or contrast. Training only on the pairs of low and high-quality images biases the ranking model in a way that it sometimes assigns a higher score to over-saturated and over-exposed images. This could have been avoided by having intermediate information about the enhancement steps. This information was available in the database created by the authors of (Yan *et al.*, 2014a). However, creating such database requires the availability of experts. As a workaround, I deteriorate the original low-quality images by introducing an additional shift to the image parameters. To determine the amount of parameter shift for each image, I initially deteriorate 20 images in a photo-manipulation software such as Adobe Photoshop. It provides me with heuristics that help define a relation between existing image parameters and the amount of parameter shift required for a significant deterioration of an image. I call these deteriorated images as *poor-quality* images. I create p poor-quality images for every low-quality image. This provides features for poor, low and high-quality images, denoted by $\boldsymbol{F}^-$, $\boldsymbol{F}$ and $\boldsymbol{F}^+$ respectively. Primal form for the proposed ranking model can be written as follows:

$$\begin{aligned}
&\min_{w,\xi_{ij}} \ \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C_1\sum_{i,j}\xi_{ij} + C_2\sum_{i,k}\xi'_{ik}, \\
&\text{subject to: } \boldsymbol{w}^T\boldsymbol{f}_{ij}^+ \geq \boldsymbol{w}^T\boldsymbol{f}_i + 1 - \xi_{ij}, \\
&\text{subject to: } \boldsymbol{w}^T\boldsymbol{f}_i \geq \boldsymbol{w}^T\boldsymbol{f}_{ik}^- + 1 - \xi'_{ik}, \\
&\text{subject to: } \boldsymbol{w}^T\boldsymbol{f}_{ij}^+ \geq \boldsymbol{w}^T\boldsymbol{f}_{ik}^- + 1 - \xi''_{ik}, \xi_{ij},\ \xi'_{ik},\ \xi''_{ik} \geq 0 \\
&\forall\, i = \{1,\ldots,N\}, \forall j = \{1,\cdots,p\}, \forall k = \{1,\cdots,p\}.
\end{aligned} \tag{4.5}$$

To incorporate the GP kernel κ, a dual form of the ranking SVM is needed. The dual form of the Equation 4.5 would be cumbersome to derive unless its representation can be slightly

altered. To this end, I define a new set of data $\boldsymbol{D}$ consisting of $\boldsymbol{f}_i - \boldsymbol{f}_{ij}^+$, $\boldsymbol{f}_{ik}^- - \boldsymbol{f}_i$ and $\boldsymbol{f}_{ik}^- - \boldsymbol{f}_{ij}^+ \; \forall \; i, j, k$. The data $\boldsymbol{D}$ has $N' = N(2p + p^2)$ elements. The primal form can be written as follows:

$$\begin{gathered} \min_{w,\xi_i} \; \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C\sum_i \xi_i, \\ \text{subject to: } \boldsymbol{w}^T\boldsymbol{D}_i + 1 - \xi_i \leq 0, \xi_i \geq 0, \forall i = \{1, \ldots, N'\}. \end{gathered} \tag{4.6}$$

Lagrangian multipliers are used to convert the above equation into an unconstrained optimization problem.

$$\begin{aligned} L(\boldsymbol{w}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C\sum_i \xi_i + \\ \sum_i \alpha_i(\boldsymbol{w}^T\boldsymbol{D}_i + 1 - \xi_i) - \sum_i \beta_i\xi_i \end{aligned} \tag{4.7}$$

Differentiating with respect to $\boldsymbol{w}$ and ξ and equating them to zero, I get,

$$\begin{aligned} \nabla_w L(\boldsymbol{w}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = 0 \Rightarrow \boldsymbol{w} = -\sum_i \alpha_i \boldsymbol{D}_i \\ \nabla_\xi L(\boldsymbol{w}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = C - \alpha_i - \beta_i = 0 \Rightarrow \alpha_i \leq C. \end{aligned} \tag{4.8}$$

Substituting $\boldsymbol{w}$ back into Equation 4.6 and doing some algebraic manipulation, I get a dual maximization problem as follows:

$$\max_{\boldsymbol{\alpha}} \sum_i \alpha_i - \frac{1}{2}\sum_i\sum_j \alpha_i\alpha_j \boldsymbol{D}_i^T\boldsymbol{D}_j, \text{ subj. to: } 0 \leq \alpha_i \leq C. \tag{4.9}$$

Following the kernel trick, I replace the inner product in the above equation with the GP kernel. Now, the final optimization problem to determine $\boldsymbol{\alpha}$ becomes,

$$\max_{\boldsymbol{\alpha}} \; \mathbf{1}^T\boldsymbol{\alpha} - \frac{1}{2}\boldsymbol{\alpha}^T\boldsymbol{K}_y\boldsymbol{\alpha}. \tag{4.10}$$

Here, $\mathbf{1}$ is a column vector of ones. The length of both $\boldsymbol{\alpha}$ and $\mathbf{1}$ is $N(2p + p^2)$. The dimensions of $\boldsymbol{K}_y$ are $N(2p + p^2) \times N(2p + p^2)$. The $(i, j)^{th}$ element of $\boldsymbol{K}_y$ is $\kappa(\boldsymbol{D}_i, \boldsymbol{D}_j)$.

4.3.3 Clustering high-quality images together

In this subsection, I introduce the third constraint. For a low-quality image: 1. it tries to cluster all its high-quality counterparts and 2. it attempts to pull apart the poor-quality and the high-quality images in the GP-kernel-induced feature space. The intuitive reasoning in introducing this constraint is as follows: For an unseen query image, multiple parameter sets corresponding to its enhanced counterparts should be predicted. If the enhanced counterparts are clustered in the GP-feature-space, then the parameter sets obtained in a single traversal can be high. Multiple parameter sets need to be predicted since image enhancement is a subjective task and a single set of parameters would not do justice. This constraint minimizes distance between $\boldsymbol{f}_i$ and $\boldsymbol{f}_{ij}^{+}\ \forall j$. Therefore, by definition of GP, the corresponding output parameters, $\boldsymbol{y}_{ij}^{+}\ \forall j$, will be clustered, that achieves the said traversal. The rest of the constraint pulls apart the predicted parameters and the low-quality image parameters. The traversal of the parameter space post GP predictions is detailed later. I formulated the above constraints as follows:

$$\min_{h} \left(\sum_{i} ||\boldsymbol{K}_{y}^{\boldsymbol{F}_i^{+}}||_F^2 - ||\boldsymbol{K}_{y}^{\boldsymbol{F}_i^{+},\boldsymbol{F}_i^{-}}||_F^2 \right), \tag{4.11}$$

where $||\cdot||_F^2$ indicates squared Frobenius norm. The term $\boldsymbol{K}_{y}^{\boldsymbol{F}_i^{+},\boldsymbol{F}_i^{-}}$ is a $p \times p$ matrix defined as follows:

$$\boldsymbol{K}_{y}^{\boldsymbol{F}_i^{+},\boldsymbol{F}_i^{-}} = \begin{bmatrix} \kappa(\boldsymbol{f}_{i1}^{+}, \boldsymbol{f}_{i1}^{-}) & \cdots & \kappa(\boldsymbol{f}_{i1}^{+}, \boldsymbol{f}_{ip}^{-}) \\ \kappa(\boldsymbol{f}_{i2}^{+}, \boldsymbol{f}_{i1}^{-}) & \cdots & \kappa(\boldsymbol{f}_{i2}^{+}, \boldsymbol{f}_{ip}^{-}) \\ \vdots & \ddots & \vdots \\ \kappa(\boldsymbol{f}_{ip}^{+}, \boldsymbol{f}_{i1}^{-}) & \cdots & \kappa(\boldsymbol{f}_{ip}^{+}, \boldsymbol{f}_{ip}^{-}) \end{bmatrix} \tag{4.12}$$

The term $\boldsymbol{K}_{y}^{\boldsymbol{F}_i^{+}}$ is equal to $\boldsymbol{K}_{y}^{\boldsymbol{F}_i^{+},\boldsymbol{F}_i^{+}}$.

I form the objective function by combining Equations 4.2, 4.10 as follows:

$$\min_{h} \; Z = \frac{1}{2}\left(\bar{\boldsymbol{y}}_m^+\right)^T \boldsymbol{K}_y^{-1}\bar{\boldsymbol{y}}_m^+ + \frac{1}{2}\log|\boldsymbol{K}_y| - \mathbf{1}^T\boldsymbol{\alpha} + \frac{1}{2}\boldsymbol{\alpha}^T\boldsymbol{K}_y\boldsymbol{\alpha} \; + \sum_i \left(||\boldsymbol{K}_y^{\boldsymbol{F}_i^+}||_F^2 - ||\boldsymbol{K}_y^{\boldsymbol{F}_i^+,\boldsymbol{F}_i^-}||_F^2\right) \tag{4.13}$$

Note that the constant term has been removed. Equations 4.10 and 4.13 can now be solved to get $\boldsymbol{\alpha}$ and $\boldsymbol{h}$.

4.3.4 Optimization

The optimization problem is separable in $\boldsymbol{\alpha}$ and $\boldsymbol{h}$. I optimize $\boldsymbol{\alpha}$ by employing a standard rank-SVM solver. It could also be solved by using quadratic programming. However, that would be memory inefficient. In particular, I use a rank-SVM implementation which uses the LASVM algorithm proposed in (Bordes *et al.*, 2005). LASVM employs active example selection to significantly reduce the accuracy after just one pass over the training examples.

After optimizing $\boldsymbol{\alpha}$, I find the local minimizer of Equation 4.13, denoted by $\boldsymbol{h}^*$. I use scaled conjugate gradient descent (SCG) algorithm for the same. SCG is chosen due to its ability to handle tens of thousands of variables. SCG has also been widely used in previous approaches involving GPs (Rasmussen, 2006; Eleftheriadis *et al.*, 2015; Rudovic *et al.*, 2013). I use chain rule to compute $\frac{\partial Z}{\partial \boldsymbol{h}}$ by evaluating first $\frac{\partial Z}{\partial \boldsymbol{K}_y}$ and then $\frac{\partial \boldsymbol{K}_y}{\partial \boldsymbol{h}}$. The matrix calculus identities from (Petersen *et al.*, 2008) are used while computing the following expressions:

$$\frac{\partial Z}{\partial \boldsymbol{K}_y} = -\frac{1}{2}\boldsymbol{K}_y^{-1}\boldsymbol{y}_m^+ \left(\boldsymbol{y}_m^+\right)^T \boldsymbol{K}_y^{-1} + \frac{1}{2}\boldsymbol{K}_y^{-1} + \frac{1}{2}\boldsymbol{\alpha}\boldsymbol{\alpha}^T + 2\sum_i \left(\boldsymbol{K}_y^{\boldsymbol{F}_i^+} - \boldsymbol{K}_y^{\boldsymbol{F}_i^+,\boldsymbol{F}_i^-}\right),$$

$$\left[\frac{\partial \boldsymbol{K}_y}{\partial \theta_q}\right]_{ij} = -\frac{1}{2}\sigma_f^2 \exp\left(-\frac{1}{2}(\boldsymbol{f}_i - \boldsymbol{f}_j)^T \Lambda (\boldsymbol{f}_i - \boldsymbol{f}_j)\right) \cdot (\boldsymbol{f}_i^{(q)} - \boldsymbol{f}_j^{(q)})^2, \tag{4.14}$$

$$\frac{\partial \boldsymbol{K}_y}{\partial \sigma_f^2} = \sigma_f^2 \exp(-\frac{1}{2}(\boldsymbol{f}_i - \boldsymbol{f}_j)^T \cdot \Lambda \cdot (\boldsymbol{f}_i - \boldsymbol{f}_j)),$$

$$\left[\frac{\partial \boldsymbol{K}_y}{\partial \sigma_y^2}\right]_{ij} = \delta_{ij},$$

$$\frac{\partial Z}{\partial \theta_q} = \text{tr}\left[\left(\frac{\partial Z}{\partial \boldsymbol{K}_y}\right)^T \left(\frac{\partial \boldsymbol{K}_y}{\partial \theta_q}\right)\right] \quad \forall q \in \{1, \ldots, D\},$$

where tr denotes matrix trace. Similarly, $\frac{\partial Z}{\partial \sigma_f^2}$ and $\frac{\partial Z}{\partial \sigma_y^2}$ are computed to construct $\frac{\partial Z}{\partial \boldsymbol{h}} \in \mathbb{R}^{D+2}$. This derivative can be used to obtain the optimal set of hyper-parameters, $\boldsymbol{h}$. In practice, all the matrix inverses are implemented using Cholesky decomposition. I alternately optimize for $\boldsymbol{\alpha}$ and $\boldsymbol{h}$ till Equation 4.13 converges or the maximum cycles are reached. I set the convergence criterion to be 10^{-3} and the maximum cycles to 20.

4.3.5 Testing

After obtaining the optimal $\boldsymbol{\alpha}$ and $\boldsymbol{h}$, I predict the parameters, $\{\bar{y}_{*1}^+, \bar{y}_{*2}^+, \bar{y}_{*3}^+\}$, for the enhanced counterpart by using three trained GP models in Equation 4.1. Let us call the mean and variances of the predicted parameters as $\boldsymbol{m} = \{m_1, m_2, m_3\}$ and $\boldsymbol{s} = \{s_1, s_2, s_3\}$ respectively. With their availability, I explain the proposed parameter space traversal.

People's choices vary a lot in such applications. Thus, it is essential to explore the parameter space to generate additional enhancement parameters. The first advantage of the proposed approach is that it can generate such parameters without referring to the training set. Since it explores the parameter space in a structured manner (with a certain mean and

variance), it is plausible to generate only 32 parameters per image instead of hundreds as done in conventional kNN-based heuristic methods.

The First step in parameter space traversal is to determine lower and upper bounds. Those can be decided heuristically. For example, I decrease the saturation, brightness and contrast at most by an amount of $\{15\%, 15\%, 5\%\}$ and increase it at most by $\{35\%, 35\%, 20\%\}$ of the original image parameter values. I observed that these limits are not critical to the quality since the generated images will be ranked later using the learned $\boldsymbol{\alpha}$ and the images with extreme parameter settings will usually be filtered out.

Now, I change (increase and decrease) the mean value of the parameters by $\boldsymbol{\mu s}$ till it reaches the pre-specified thresholds. Intuitively, $\boldsymbol{s}$ should provide the direction of the stride in the parameter space and $\boldsymbol{\mu}$ provides the length of that stride. The value of $\boldsymbol{\mu}$ is determined by the number of enhanced counterparts the user wants to generate for each low-quality image. I set that value to be 30. This value could be decreased if the user is on a mobile device with a smaller screen and similarly increased when operating on a desktop. These settings can be changed on-the-fly.

4.3.6 Image feature representation

I extract 432-D color histogram with 12 bins for hue, six bins each for saturation and value, which acts as a global feature. The image is divided into a 12×12 grid. For each grid, I calculate its saturation, value by taking the mean values of those image blocks in the HSV color space. I also calculate RMS contrast on that grid. These act as localized features of 144-D each. I finally append the image parameters, which are average saturation, value and RMS contrast. Appending the image parameters allows GP to express the parameters of the enhanced counterparts as a function of both, the low-quality parameters and its feature vector. Finally, I get a 867-D $(= 432 + 3 \times 144 + 3)$ representation for every image.

4.3.7 *Implementation Details and Efficiency*

GPs are known to be computationally intensive. They take about $O(N^3)$ time for training, where N are the number of training examples. The matrix inversion of an $N \times N$ matrix and the computation of the derivative of the kernel are the bottlenecks in the GP training procedure. I train a GP model using about 1200 low-quality images and six counterparts per image in about 18 hours on an Intel Xeon @2.4 GHz $\times$ 16. The computational efficiency can be improved by using GP regression techniques proposed for large data (Hensman *et al.*, 2013; Ambikasaran *et al.*, 2014) or using efficient data-structures such as KD-trees (Shen *et al.*, 2006). During testing, the proposed approach executes fast. I tested it on two systems, Intel Xeon, and a modern desktop system with Intel i7 @3.7GHz. It can predict all the three parameters for 3150 and 1287 images per second using Intel Xeon and i7 systems respectively. A built-in kNN-search function processes only 224 images per second when asked to find one nearest-neighbor in 5000 image data-set on the Intel Xeon system. All the implementations are done in MATLAB. Since the proposed approach need not query the training database, it could be portable and potentially allow for enhancements being performed on mobile devices.

4.4 Data-sets and Experimental Setup

In this section, I describe the data and the experimental setup. Results of these experiments are presented in the Section 7.1.4. I perform four kinds of experiments. The first experiment provides a weak quantitative measure of the accuracy of the proposed approach. I use the MIT-Adobe FiveK (Bychkovsky *et al.*, 2011) data-set for this experiment. This data-set has 5000 low-quality images with 5 expert-enhanced counterparts for each image. This is the largest such data-set available. I use 1200 images and six counterparts (three each for poor and high-quality) per low-quality image to train the GP models. I use 1500 and

800 images for validation and testing respectively. I predict the parameters (i.e., brightness, contrast, and saturation) for the first enhanced counterpart of all the images in the test set. Then, I calculate the root mean square error (RMSE) and a more stringent criterion - Pearson's correlation - between the ground-truth parameters computed from the expert-enhanced image and the predicted parameters. I compare the obtained quantitative results against twin Gaussian processes (TGP) (Bo and Sminchisescu, 2010). TGP is a structured prediction method which considers the correlation between both input and output to produce predictions. Though a low RMSE between ground truth and predicted parameters does not guarantee that the enhancement will be visually appealing (unless the RMSE tends to zero), it confirms that the prediction is lying near the ground-truth in the parameter space. Also, this experiment validates the effectiveness of the GP regressor.

The second experiment is a qualitative measure of the image quality produced by the proposed and the competing algorithms, namely kNN, Picasa and that of (Yan *et al.*, 2014a). The metric of $L2$ error in the L*ab space was adopted in (Yan *et al.*, 2014a). I believe that it is a poor indicator of the enhancement quality and instead opt for Visual Information Fidelity (VIF) metric (Sheikh and Bovik, 2006). This metric can predict whether the visual quality of the other image has been enhanced in comparison with the reference image by producing a value greater than one. This is unlike other quality metrics such as SSIM (Wang *et al.*, 2004), FSIM (Zhang *et al.*, 2011b), VSI (Zhang *et al.*, 2014) etc. I use the publicly available implementation of VIF [2] . I calculate the VIF between the proposed enhancement and the enhancement by 1. kNN 2. Picasa and 3. the approach of (Yan *et al.*, 2014a). Thus VIF < 1 implies that the proposed enhancement is better than the one produced by the competing algorithm and vice-versa. This comparison is made for 60 pairs where 15 images each are enhanced using Picasa and (Yan *et al.*, 2014a), whereas the remaining 30 images are enhanced using the kNN approach.

[2]available at `live.ece.utexas.edu/research/quality/`

The third experiment is aimed towards evaluating the effectiveness of GP ranking. For each image, I generate only 32 enhanced versions. Our GP ranker selects the highest ranked image out of those 32 and presents it to the user. The top-ranked image is supposed to have the best quality. I compute the VIF metric between the best image selected by the ranker and the other 31 images. Ideally, for all these 31 images, the proposed approach should obtain values less than one indicating that GP ranker has indeed selected the best image.

I also carry out a subjective evaluation test to assess if people prefer the enhanced counterparts generated by our approach. I compare the proposed approach against three other methods. First one is the kNN-based approach. Given a low-quality image, I search for the nearest non-duplicate image from the 5000 images of MIT-Adobe dataset. The parameters of the expert-enhanced counterparts of the nearest image are applied to the given low-quality image. In this manner, I generate 5 enhanced counterparts per low-quality image. Note that, kNN utilizes all other 4999 images whereas I only use the model trained on 1200 images for prediction. Then I compare against Picasa's one-touch-enhance tool. The third approach is from (Yan *et al.*, 2014a), which also is a learning-to-rank based image enhancement approach that uses the pipeline shown at the top in Fig. 4.1.

I use 60 images for the subjective test which was performed by 15 people. Thirty images are selected from the testing set of the MIT-Adobe data-set. The rest of the images are from the data-set used in the paper (Yan *et al.*, 2014a). Since I only have access to their testing set, I use that data-set solely for subjective test purposes. It contains 124 images out of which I randomly select 30 images. I enhance all the 60 images using the proposed approach. The comparison against other methods is made as follows.

The first 30 images from the MIT-Adobe data is split into two halves. The first half is enhanced using the kNN approach and the second half is enhanced using Picasa. The remaining 30 images from (Yan *et al.*, 2014a) are split into two halves. The first half is enhanced using the kNN approach, and for the second half, I directly use the high-resolution

results of the test data-set of (Yan *et al.*, 2014a). Thus each person compares 60 image pairs. One of the image in that pair has been enhanced using the proposed approach, and the other image has been enhanced using either kNN approach, Picasa or the approach of (Yan *et al.*, 2014a). The subject has to choose the image which he/she finds "visually-appealing". If the subject feels that both images have almost the same visual appeal, a third option of preferring neither image is provided. The order in which the images appear in front of a subject is always randomized. The pairing order is also randomized. The subjects do the evaluation test in standard lighting conditions and at a comfortable and constant distance from the screen.

4.5 Results

I present results of the quantitative analysis first. I have trained three GP models to predict saturation, brightness, and contrast for 800 images from the test set of MIT-Adobe data-set. When compared with the parameters of expert-enhanced counterparts, the models achieve RMSEs of $0.0057, 0.0022, 0.0037$ and correlations of $0.5359, 0.5553, 0.8023$ respectively, for the above three parameters. TGP gets an average RMSE of 0.0022, but it suffers while producing an average correlation of only 0.3326. It is relatively easier for a GP to relate the contrast to the image quality, which is intuitive since contrast variation changes the image drastically and it also makes the image look vibrant or dull. This, in turn, contributes most to the visual appeal of an image.

The left bar chart in Fig. 4.3 shows the results of the second experiment. VIF between the proposed enhancement and competing enhancements produces values which are, in most cases, less than one. Thus according to VIF metric, the proposed approach produces better enhancements than Picasa, kNN-based heuristics, and (Yan *et al.*, 2014a). For the third experiment, I get 32 VIF values for each image, which correspond to 32 enhanced versions generated by the proposed approach. The GP ranker selects one, as mentioned

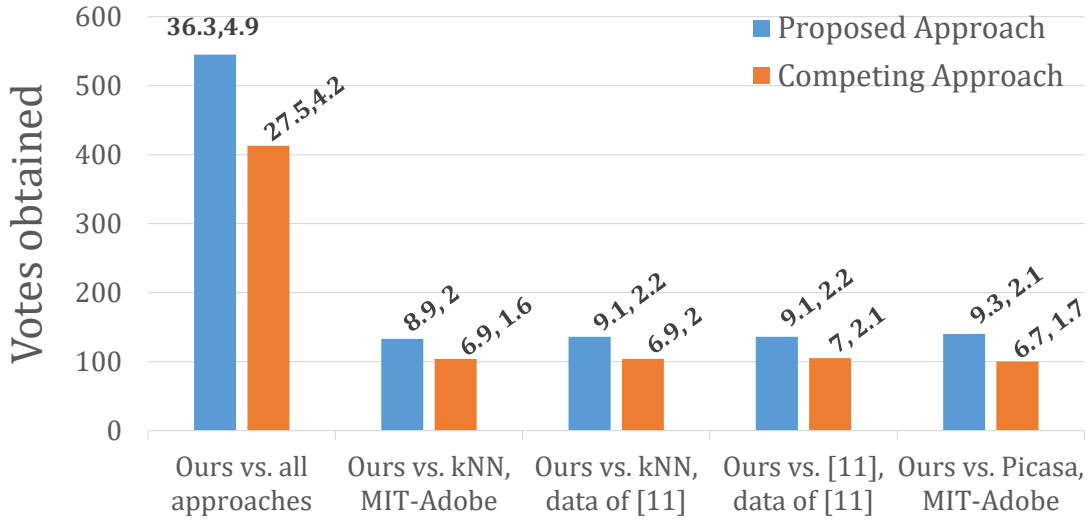


Figure 4.2: Subjective evaluation test metrics ©2017 IEEE.

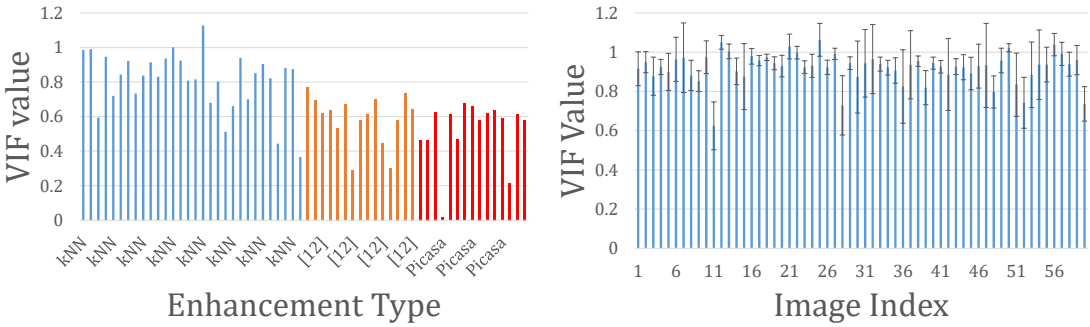


Figure 4.3: Left Plot Shows VIF Values Comparing Proposed Enhancement and Enhancements Produced by the Competing Algorithms. The Right Plot Shows the Mean and Standard Deviation of the VIF Values Between the Best Enhancements and 31 Other Enhancements "rejected" by Gp Ranker. VIF Values < 1 Are Desirable in Both the Cases ©2017 IEEE.

earlier. I compute the average VIF value and its standard deviation over 31 other images. This process is repeated for all the 60 images, and the VIF values are shown in the right bar chart of Fig. 4.3.

I now analyze the results of the subjective tests. I provide the following five metrics about the subjective test in Fig. 4.2. 1. I count votes gathered by the proposed approach and

Figure 4.4: The Left Column Always Contains an Original Low-quality Image. Row 1 and 3: Columns 2, 3 and 4 Contain Images Enhanced by $k$nn, Picasa, and GP. Row 2: The Right Three Columns Contain Enhanced Versions Generated by GP. Please Read Text for Details[3]

by all other competing approaches bundled into one. This is a coarse measure of how much preference people have towards enhancements generated by the proposed approach. 2. I count votes gathered by the proposed approach and by the kNN approach on the MIT-Adobe data-set. 3. comparison of votes gathered by the proposed approach and by the kNN approach on the data-set of (Yan *et al.*, 2014a). 4. comparing the proposed approach against the results of (Yan *et al.*, 2014a) on their data. 5. Lastly, I compare the proposed approach versus Picasa on the MIT-Adobe data. Fig. 4.2 shows all these metrics. On top of each bar, I indicate the mean and standard deviation for that particular approach and metric. For example, the second set of bars denote that for the MIT-Adobe data, the proposed approach gathered 133 votes against 104 votes gathered by the kNN approach. The average number of votes obtained per user for our and the kNN approach were 8.9 and 6.9 with the standard

deviations of 2 and 1.6, respectively. Fig. 4.2 shows that people consistently prefer the proposed approach over other state-of-art approaches.

Fig. 4.4 shows some of the results obtained by the proposed, the approach of (Yan *et al.*, 2014a), kNN and Picasa's auto-enhance tool. The first and the third row illustrate that the kNN approach is not always effective and sometimes may give over(under)-exposed results due to its dependence on the nearest training image parameters. The second row shows three representative versions generated by GP. We can see that the image in the fourth column is over-exposed. However, my ranking model successfully filters out that image and selects the one in the third column. In general, I observed that kNN could only get comparable results to Picasa and the proposed approach if it finds a good match in the training set. Thus kNN is unlikely to scale to large-scale enhancement tasks.

4.6 Discussion

GPs for image enhancement work well given that proper constraints are imposed over the covariance function. As mentioned before, the proposed approach learns a separate GP model for each image parameter. This makes the training computationally expensive. Testing is affected by a little amount given its current execution speed. A non-trivial extension of this approach would be to use multi-output (Alvarez *et al.*, 2010; Nguyen *et al.*, 2014; Alvarez and Lawrence, 2011) or multi-task (Bonilla *et al.*, 2007; Yu *et al.*, 2005) GPs. Multi-output GPs will be able to predict all parameters jointly whereas multi-task GPs can predict all enhancement versions (e.g., five in case of MIT-Adobe data-set) as well as all parameter outputs jointly.

The proposed approach, though computationally expensive while training, should not be highly affected by the interaction between the three image parameters. The reason lies in the fact that the GP models are trained on the images that have *corresponding* enhanced counterparts. If I denote the high-quality parameters by p_1^+, p_2^+, p_3^+ and the low-quality

image feature along with its parameters denoted by $\boldsymbol{u}$, then I can write,

$$\Pr(p_1^+, p_2^+, p_3^+|\boldsymbol{u}) = \Pr(p_1^+|p_2^+, p_3^+, \boldsymbol{u}) \cdot \Pr(p_2^+|p_3^+, \boldsymbol{u}) \cdot \Pr(p_3^+|\boldsymbol{u}). \tag{4.15}$$

Due to corresponding low and high-quality images, the changes in the image feature across the low and high-quality versions can be directly related to the parameter changes. Thus I believe that, it is possible for a GP to model $\Pr(p_1^+|p_2^+, p_3^+, \boldsymbol{u})$ without having access to p_2^+ and p_3^+, i.e., $\Pr(p_1^+|\boldsymbol{u})$. The multi-task GPs could be valuable while training them on data-sets which have non-corresponding low and high-quality images.

In this chapter [4] , I presented a novel approach to content-adaptive image enhancement using joint regression and ranking by employing GPs. I train the GP models on the pairs formed from poor, low and high-quality images. The learned GP models predict the desired parameters for a low-quality image from its features, which may produce its enhanced counterparts. I also described a strategy to traverse the parameter space without referring to the training images, which makes the proposed approach efficient during testing. The GP prediction is defined by the covariance kernel, on which two constraints are imposed. The first one enables the kernel to learn the feature dimensions responsible for making an image of higher-quality. The other constraint clusters all the enhancement parameters corresponding to a low-quality image, thereby allowing for effective parameter traversal. I perform quantitative and subjective evaluation experiments on two-data sets to assess the effectiveness of the proposed approach. The two data-sets used are the MIT-Adobe data (Bychkovsky *et al.*, 2011) and the one proposed in (Yan *et al.*, 2014a). Quantitative experiments show that the proposed predictions produce a low RMSE when compared with the ground-truth parameters of the MIT-Adobe data. The results show that people consistently prefer the enhancements produced by the proposed approach over the other

[4]Most of the material in this chapter has appered in (Chandakkar and Li, 2017b). See the full credit statement in appendix.

state-of-art approaches.

Chapter 5

A COMPUTATIONAL APPROACH TO RELATIVE AESTHETICS

5.1 Problem Introduction

This chapter introduces the topic of automatic assessment of image aesthetics which has recently become an active area of research due to its wide-spread applications. Most of the existing state-of-art methods treat this as a classification problem where an image is categorized as either beautiful (having high aestheticism) or non-beautiful (having low aestheticism) [1] . In (Datta *et al.*, 2006; Ke *et al.*, 2006), this problem has been formulated as a classification/regression problem by mapping an image to a rating value. Various approaches such as (Datta *et al.*, 2006; Ke *et al.*, 2006; Bhattacharya *et al.*, 2010; Luo and Tang, 2008; Dhar *et al.*, 2011; Luo *et al.*, 2011; Nishiyama *et al.*, 2011; O'Donovan *et al.*, 2011; Su *et al.*, 2011; Marchesotti *et al.*, 2011) have been proposed which either use photographic rules or hand-crafted features to assess the aesthetics of an image. Due to the recent success of deep convolutional networks, approaches such as (Lu *et al.*, 2014, 2015) claim to have learned the feature representations necessary to categorize the given image as either beautiful or non-beautiful.

The approaches based on photographic rules have certain limitations. For example, the implementations of these rules may be an approximation, thus affecting the accuracy of the aesthetic assessment. Also, the rules may not sufficiently govern the process of how the aesthetic quality of an image is decided. It is possible that some of the essential rules have been left out or some erroneous ones have been included. These rules are mostly accompanied by generic image descriptors or task-specific hand-crafted features. Such

[1] This terminology is used throughout the chapter.

approaches suffer from the disadvantages of generic/hand-crafted features that they may not be suited for a particular task such as aesthetic assessment or the feature space does not adequately represent the key characteristics which make an image aesthetic. The deep neural network based approaches may overcome these disadvantages by learning the feature representations.

While deep learning approaches have advanced the state-of-art for this task, I observe that classifying a given image as beautiful or non-beautiful may not always be the natural choice for some applications. It may also be more intuitive for humans to compare two images rather than giving an absolute rating to an image based on its aesthetic quality. Moreover, all images in a set could belong to the beautiful or non-beautiful category according to a classification model. In such cases, it may often be necessary to rank the images according to their aesthetic quality. For example, a machine-learned enhancement system (Yan *et al.*, 2014a) has to provide an enhanced version of the query image to the user. To do so, it needs to compare two images with respect to their aesthetics to determine which enhancement results in a more beautiful image. In an image retrieval engine, it would be desirable to have an option to retrieve images having low/similar/high aesthetic quality as compared to the query image.

Motivated by these observations, I introduce a novel problem of picking a more beautiful image from a pair. I term this problem as "Relative Aesthetics". I build a new dataset of image pairs for this task by carefully choosing images from the popular AVA dataset (Murray *et al.*, 2012) to satisfy certain constraints. For example, I observed that comparing images from unrelated categories (for example, a close-up of a car and a wedding scene) does not make sense and hence such pairs are avoided. There exists no single threshold which can binary-classify the pairs correctly across the entire dataset. In other words, if images were categorized into beautiful and non-beautiful, then some of the pairs in the data used could contain both beautiful or both non-beautiful images. The details of dataset creation and its

statistical analysis are provided in Section 5.4.

The proposed problem draws certain parallels with "relative attributes" (Parikh and Grauman, 2011b), where it was observed that training on relatively-labeled data leads to models that capture more general semantic relationships. They also mention that by using attributes as a semantic bridge, their model can relate to an unseen object category quite well. On the other hand, the proposed problem presents different challenges. In (Parikh and Grauman, 2011b), they compare two images with respect to attributes (for example, more natural, furrier, narrower, etc.), which are better defined than the aesthetics of two images. Thus even though it is trivial to use models trained on categorical data to solve these ranking tasks, I found that using relative learning principles allows us to outperform previous state-of-art classification models by gaining a more general and a semantic-level understanding of the proposed problem.

My contributions are as follows:

1. I propose a novel problem termed as "relative aesthetics", which involves picking a more beautiful image from a given pair of images. I create a new dataset which has such relative labels from the popular AVA dataset by careful and constrained selection of image pairs.

2. I build a deep network incorporating the relative learning paradigm and train it end-to-end. To the best of my knowledge, there is no prior work on studying aesthetics in a relative manner using deep neural networks.

3. I show that the proposed model trained on relatively-labeled data can outperform a recent state-of-art method (Lu *et al.*, 2014) trained on a similar sized, categorically labeled dataset for the proposed task.

5.2 Related Work

Computational aesthetics research in the earlier years was focused on employing photographic rules, hand-crafted features or generic image descriptors. Intuitive and common properties such as color (Datta *et al.*, 2006; Nishiyama *et al.*, 2011; O'Donovan *et al.*, 2011), texture (Datta *et al.*, 2006; Ke *et al.*, 2006), content (Luo *et al.*, 2011; Dhar *et al.*, 2011), combination of photographic rules, picture composition and hand-crafted features (Dhar *et al.*, 2011; Luo and Tang, 2008; Luo *et al.*, 2011) have been used. One of the most commonly used photographic rules is the *Rule of Thirds* used in (Dhar *et al.*, 2011; Luo and Tang, 2008; Datta *et al.*, 2006). Other compositional rules include low depth of field, opposing colors, etc. (Dhar *et al.*, 2011). Common color features such as lightness, color harmony, and distribution, colorfulness have been quantified for aesthetics assessment by computational models (Datta *et al.*, 2006; Nishiyama *et al.*, 2011; O'Donovan *et al.*, 2011). Texture features based on wavelets edge distribution, low depth of field, amount of blur have also been used (Ke *et al.*, 2006; Dhar *et al.*, 2011). Approaches specifically trying to model content in the image by detecting people (Luo *et al.*, 2011; Dhar *et al.*, 2011; Luo and Tang, 2008), generic image descriptors such as SIFT (Lowe, 2004b) have been proposed in (Dhar *et al.*, 2011). Inspired by the then success of deep neural network on various tasks such as image classification (Krizhevsky *et al.*, 2012; Ciresan *et al.*, 2012), object segmentation (Chen *et al.*, 2013), facial point detection (Sun *et al.*, 2013), Decaf features (Donahue *et al.*, 2013) for style classification (Karayev *et al.*, 2014) etc., (Lu *et al.*, 2014) proposed a deep-learning-based approach to aesthetics assessment. This approach classifies a given image as beautiful or non-beautiful depending on the entire image as well as its local patches. Another such approach was presented in (Lu *et al.*, 2015) where the authors aggregate the information from multiple patches in a multiple-instance-learning manner to improve the result of aesthetics assessment. Most of these approaches treat aesthetics

assessment as a binary classification task, which may not always be the best choice for many applications, as discussed before.

The concept of training on relatively-labeled data to improve model performance and provide it with a certain semantic understanding of the problem has been well-explored. The work on relative attributes (Parikh and Grauman, 2011b) predicts the relative strength of individual property in images. It allows for comparison with an unseen object category in the attribute space. Models learned in such a way enable richer text descriptions of images. Relative attribute feedback was used in conjunction with semantic language queries to improve the image search capability in (Kovashka *et al.*, 2012). There are many such applications where relative learning has explored a new dimension of the problem and improved the overall understanding of the model of a given task.

In this chapter, I propose to employ the relative learning principles for the task of image aesthetics assessment. This task is extremely subjective and has vaguely-defined properties than other general attributes like size, being more natural, etc. Various datasets have been proposed such as *Photo.net, DpChallenge.com, AVA* datasets to allow for learning using hand-crafted features. The first two datasets contain 20,278 and 16,509 images respectively [2] , whereas the AVA dataset (Murray *et al.*, 2012) contains 250,000 images. Thus I use AVA to form image pairs which in turn will facilitate the learning of the proposed approach. I propose a Siamese deep neural network architecture (Bromley *et al.*, 1993) with a relative ranking loss, which takes an image pair as input and ranks them with respect to their aesthetic quality. The back-propagation happens with the loss obtained from the ranking function, which, I believe, helps the network explore the attributes of certain images that make them more beautiful than others.

Table 5.1. The Architecture of a Column in the Proposed Network. Convolution Is Represented as (Padding, # Filters, Receptive Field, Stride).

layers	specifications
Padded Input	$3 \times 230 \times 230$
Conv	$2, 64, 11, 2$
Max-pooling	2×2
Conv	$1, 64, 5, 1$
Max-pooling	2×2
Conv	$1, 64, 3, 1$
Conv	$-, 64, 3, 1$
Dropout	0.5
Dense	1000
Dropout	0.5
Dense	256
Dropout	0.5

5.3 Proposed Approach

The comparison of the aesthetics of two images is dependent on many factors and people's visual preferences. Some of the factors include color harmony (Nishiyama *et al.*, 2011), colorfulness (Datta *et al.*, 2006), inclusion of opposing colors (Dhar *et al.*, 2011), composition (Litzel, 1975), visual balance (Niekamp, 1981) etc. They are also affected by the content in the image (Luo and Tang, 2008; Luo *et al.*, 2011). Though determination of aesthetics is a subjective process, there are some well-established rules in the photography

[2]Datasets hosted on `ritendra.weebly.com/aesthetics-datasets.html`

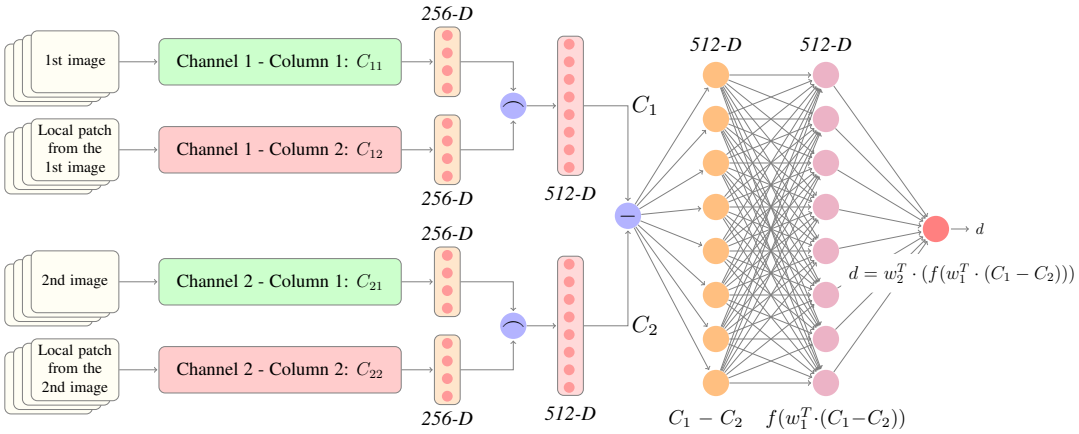


Figure 5.1: The Architecture of the Proposed Network. Weights are Shared Between the Columns C_{11} and C_{21} (Shown in Green), C_{12} and C_{22} (Shown in Red); The Features Obtained From C_{11} and C_{12} are Concatenated (Represented by $\frown$ Symbol) to Get C_1 and C_{21} and C_{22} are Concatenated to Get C_2; The Vector $C_1 - C_2$ is Passed Through Two Dense Layers to Obtain a Score d Comparing the Aesthetics of Two Images. $f(\cdot)$ Denotes an ReLU Non-linearity. Please Refer to the Text for Further Details ©2016 IEEE.

community such as low depth-of-field, the rule of thirds, golden ratio (Joshi *et al.*, 2011). However, making hand-crafted features for such rules is difficult and often will lead to approximation or misrepresentation of those rules. Therefore, I take a deep neural network based approach in which I incorporate relative ranking by designing a suitable loss function. Most of the rules or aesthetic criteria can be defined using either an entire image or a part of it. Therefore, for each image in the pair, the proposed network is trained on two *views* of an image as also done in (Lu *et al.*, 2014): the entire image and a local patch. This enables the network to see different aspects of the input. For example, a view of the entire image may provide the network with the knowledge of color composition while the local patch may help with resolution, depth-of-field, etc. I now describe the network architecture and its training procedure in detail.

5.3.1 Network Architecture

The proposed deep convolutional neural network (DCNN) architecture takes an image pair as input. For each image in the pair, it takes as input that image itself and its local patch. Since all images have to be of the same size, they are warped to be $224 \times 224 \times 3$. A same-sized local patch is cropped from the original image. I choose to warp the image based on the findings in (Lu *et al.*, 2014), which shows that local patches along with warped image give the best result. The proposed network has two "channels" as shown in Fig. 5.1, corresponding to the input pair of images. A channel is defined as the part of the proposed CNN which takes an image along with its local patch as input. Each channel has two "columns". One column takes the warped image, and the other one takes its local patch as input.

The proposed architecture is a Siamese network where each channel shares weights in a certain way, which is shown in Fig. 5.1 by means of color coding. The columns with the same color (i.e., either red or green) share the weights. This is because the ranking produced by the network should be invariant to the order of the images in the pair. Both channels have an identical architecture until they are merged at the final dense layer of $512 - D$. I now describe the architecture of the upper channel (channel 1). This channel has two columns which take the image and its local patch as input. Since these two inputs are on a different spatial scale and trying to convey different aesthetic properties as discussed earlier, I do not set constraints on the weights of both the columns in a channel. The upper column in channel 1 (C_{11}) takes the entire image as input which is of size $224 \times 224 \times 3$, zero-padded with 3 pixels on all sides. The column has five convolutional layers. The first convolutional layer has 64 filters each of size $11 \times 11 \times 3$ with stride 2. The second convolutional layer has 64 filters of size 5×5 with stride 1. Third and fourth layer have 64 filters of size 3×3 with stride 1. These are followed by two dense layers of size 1000 and 256 respectively. Then

50% Dropout at these two dense layers is applied. Max-pooling is applied after first two convolutional layers. Each max-pooling operation halves the input in both the directions. ReLU activation is used in the entire network. The architecture of C_{11} is also detailed in Table 5.1. The lower column of channel 1 (C_{12}) and both the columns of channel 2 (i.e. C_{21} and C_{22}) have the same architecture as C_{11} including dropout, max-pooling and zero-padding operations.

The key thing to note here is that the weights are shared for (i) the two columns which take the entire image as input i.e. C_{11} and C_{21}, and (ii) the remaining two columns which take the local patches as input i.e. C_{12} and C_{22}. C_{11} and C_{21} each generate a $256 - D$ representation (i.e. of the entire image). Similarly, C_{12} and C_{22} also generate $256 - D$ features (i.e. of the local patch). The $256 - D$ representations from (C_{11}, C_{12}) as well as from (C_{21}, C_{22}) are concatenated to form two $512 - D$ representations. Fig. 5.1 shows this architecture and the sharing of weights.

I explain the proposed ranking loss function which takes the above two $512 - D$ representations and gives a quantitative measure comparing the aesthetics of the two images in a pair.

5.3.2 Ranking Loss Layer

The proposed network aims at correctly ranking two input images based on their underlying aesthetic quality. Formally, given two input images I_1 and I_2, I_1 is more beautiful than I_2 (also denoted as $I_1 > I_2$ here onward) if a positive value is obtained for $d(I_1, I_2)$ and vice versa. In other words, $d(I_1, I_2)$ is a measure comparing aesthetics of two images.

$$d(I_1, I_2) = w^T \cdot (g(I_1) - g(I_2)) \tag{5.1}$$

Here, $g(I_1)$ and $g(I_2)$ are the CNN representations. In the proposed network, $g(I_1)$ and $g(I_2)$ are represented by C_1 and C_2 respectively, as shown in Fig. 5.1. To increase the

representational power, $(C_1 - C_2)$ is passed through two dense layers separated by a ReLU non-linearity. Thus for the proposed network, Equation 5.1 takes a slightly modified form as follows:

$$d(I_1, I_2) = w_2^T \cdot (f(w_1^T \cdot (C_1 - C_2)), \tag{5.2}$$

where $f(\cdot)$ denotes an ReLU non-linearity.

Keeping this in mind, I design the final loss function with the following properties:

1. It should propagate zero loss when all image pairs are ranked "correctly" (i.e., the representations of the images in these pairs are separated by a margin δ).

2. It should only be able to produce a non-negative loss.

Hence the loss function is designed as follows:

$$L = \max(0, \delta - y \cdot d(I_1, I_2)), \tag{5.3}$$

where y is a ground-truth label which takes value 1 if the first image in the pair is more beautiful than the second (i.e. $I_1 > I_2$) and it equals -1 if $I_1 < I_2$. The term $\max(0, \cdot)$ is necessary to ensure that only non-negative loss gets back-propagated. The δ is a user-defined parameter which serves two purposes. First, it defines a required separation to declare $I_1 > I_2$ (or $I_1 < I_2$). That means if $y \cdot d(I_1, I_2) > \delta$, then no loss should be back-propagated for such pairs. Secondly, and more importantly, $\delta > 0$ avoids a trivial solution to the optimization objective. To clarify further, if $\delta = 0$, then for $y = 1$ and $y = -1$, a common trivial solution exists which makes either $w_1 = 0$ or $w_2 = 0$. I set $\delta = 3$ as I do not find any performance boost by further increasing the separation between CNN feature representations of I_1 and I_2.

In the further subsections, I explain the training and testing procedures of the proposed network. Then I compare the aesthetic ranking results of the proposed network against a state-of-art network that is trained on a categorical data.

5.3.3 Training the Architecture

This architecture is trained using mini-batch SGD with a learning rate of 0.001, momentum = 0.9, weight decay of 10^{-6} and by employing Nesterov momentum. The learning rate is reduced by 15% after every ten epochs. The batch size is set to 50. Apart from warping and cropping out the local patch, only the mean RGB value computed on the training set is subtracted from each pixel of the image. During training, when the network makes a wrong decision, it is forced to learn by exploiting the difference between some other characteristics of the image in the next iteration. Over many epochs, it manages to discover the relevant image properties which better define image aesthetics.

The dataset has $23,000$ image pairs containing all unique images (i.e. total $46,000$ images). I use subsets of $20,000$ and $3,000$ pairs for training and validation respectively. I stop the training when the accuracy on the validation set does not show significant improvement for 10 consecutive epochs. I train using relative labels i.e. a pair is labeled as 1 if $r_1 - r_2 > 1$, otherwise it is labeled as -1. Here, r_i is the average rating of I_i in AVA dataset. More details on the data creation are given in Section 5.4.

5.3.4 Testing the Architecture

Given a new pair of images, initially, I subtract the mean of the training data from each pixel of both the images. I would like to point out that the test set does not share any pairs or any individual images with the training and validation set. I pass both the images and their patches into the network and get the value of $d(I_1, I_2)$ from Equation 5.2. I_1 is then predicted as a more beautiful image than I_2 if $d(I_1, I_2) > 0$ and vice versa. The test set

contains $20,000$ image pairs. I use the weights of the epoch where the highest ranking accuracy with the least amount of loss was achieved on the validation set.

5.3.5 Ranking using a Network Trained on Categorical Labels

I train a network on categorically-labeled data using our implementation of the RAPID approach (Lu *et al.*, 2014), which is a recent state-of-art method for aesthetics assessment. It is trained on the same set of 40,000 images that is used to train the proposed network. However, in this case, these images have been categorized as either beautiful or non-beautiful depending on the average ratings obtained directly from the AVA dataset. The ratings in the AVA dataset range from 1-10. I set the threshold to 5.5, and that determines the class of an image. This network consists of stacks of convolutional layers, followed by dense layers and finally a sigmoid to convert the raw scores into a probability measure, $p(y = 1|I)$, i.e., the probability of an image I belonging to the beautiful class. I point the reader to (Lu *et al.*, 2014) for more details about the RAPID network architecture. While testing for a pair of input images, the first image is passed through the network, and the probability measure - $p(y = 1|I_1)$ - is obtained. Passing the second image provides $p(y = 1|I_2)$. The first image is judged to be more beautiful than the second one if $p(y = 1|I_1) > p(y = 1|I_2)$. This test set contains $20,000$ image pairs and is identical to the test set used for the proposed approach as mentioned in Section 5.3.4. Despite RAPID network being similar in size to the proposed network, it gets a significantly lower accuracy on this relative ranking problem, which suggests that a network trained on categorically-labeled data fails to learn the complex, relative ranking order in the data.

5.4 Dataset

The task is to determine the more beautiful image in a pair. To the best of my knowledge, there exists no such dataset containing relatively-labeled pairs with respect to their aesthetic

rating. I created a dataset containing $43,000$ image pairs. The individual images in these pairs belong to the AVA dataset (Murray *et al.*, 2012). I use $20,000$ pairs for training, $3,000$ for validation and the rest for testing. I now describe the protocol used to form the pairs out of the images from the AVA dataset. The protocol can be defined by these three constraints:
1. The difference between the average ratings of images in a pair should be ≥ 1. Constraining this difference ensures that the training/test pairs are more likely to be aesthetically different.
2. Each image in the AVA dataset has 210 ratings on an average. I computed the variance of all the ratings for each image. I observed that the distribution of all these variances over the entire the AVA dataset takes the form of a Gaussian with a mean of 2.08 and a standard deviation of 0.6. The minimum and maximum variance in the image ratings are 0.8 and 4.5 respectively. As mentioned in (Murray *et al.*, 2012), high variances among the image ratings are a result of the collective disagreement between the raters, which suggests that such images may have certain abstract/novel content or photographic style, preferred only by a certain group of people. I avoid the images which cause such significant disagreements among the raters by only considering the images having rating-variance less than 2.6.
3. I avoid including pairs from different categories since the characteristics which make an image aesthetic may vary with the category. For example, a beautiful picture of a car may have bright colors whereas a beautiful picture of a human face may have low-depth of field and better details. Additionally, since the ratings in the AVA dataset are crowd-sourced ratings, the opinions may exhibit a preference towards some category. The effect of these two factors can be mitigated by using pictures from the same category to form pairs.

After such selection of pairs, the relative labels can be formed. A pair is labeled as 1 if the average rating of the first image is greater than that of the second image and -1 otherwise. The majority of the pairs in the dataset have the rating-difference ≈ 1. To quantify, the rating-difference for about 85% of the training and test data is between 1 and 1.5. As the rating difference between the images of a pair decreases, choosing the more

Figure 5.2: Rankings Produced by the Proposed Network Are Shown Above. Top and Bottom Rows Show Correct and Wrong Predictions Respectively for a Total of Four Pairs. Each of Them Is Enclosed in Either Red/Green Boxes. For Every Pair, the Network Ranks the Right Image Higher than the Left Image. Please View in Color ©2016 IEEE.

beautiful image in that pair gets difficult. Also, to ensure that the proposed network is not biased towards this dataset, I replicate the experiments on another reference test-set provided by the creators of the AVA dataset (Murray *et al.*, 2012). This reference test-set contains $20,000$ images and has also been used by (Lu *et al.*, 2014). By following the aforementioned protocol, these $20,000$ images yield us $7,670$ pairs. I call these set of pairs as the standard test set. I now describe the experiments and give an analysis of results.

5.5 Experiments and Results

I run the proposed network on the test set and the standard test set containing $20,000$ and $7,670$ image pairs respectively. I achieve a ranking accuracy of 70.51% and 76.77% on the test-set and the standard test-set respectively. Here, ranking accuracy is defined as the fraction of pairs for which the model correctly picks the more beautiful image according

Table 5.2: Results for Ranking and Binary Classification ©2016 IEEE

	Ranking on the test-set	Ranking on the pairs from standard test-set	Classification on the test-set	Classification on standard test-set
RAPID (Lu *et al.*, 2014)	62.21	65.87	**59.92**	69.18
Proposed	**70.51**	**76.77**	59.41	**71.60**

to the ground-truth labels. I compare the proposed approach with a state-of-art aesthetics classification network called RAPID (Lu *et al.*, 2014), trained as described in Section 5.3.5: both the images are passed one-by-one to the RAPID network, and the more beautiful image is chosen. RAPID produces a ranking accuracy of 62.21% and 65.87% on the test set and the standard test-set respectively. Since each channel of the proposed architecture is a replica of (Lu *et al.*, 2014) with the modified ranking loss, I compare the proposed architecture only with (Lu *et al.*, 2014). However, I believe that similar performance improvements can be obtained if a different state-of-art model (e.g., (Lu *et al.*, 2015)) was used for each of the channels.

5.5.1 Performing Binary Classification using the Proposed Network

Due to the proposed relative-learning-based approach, I believe that the network has gained a semantic-level understanding of the properties which make an image highly aesthetic. To verify this, I attempted binary classification on the test set as well as the standard test-set. For this purpose, I extracted the top channel of the network i.e. C_{11} and C_{12} (see Fig. 5.1). I use the best weights learned from the ranking task for this channel. After the last node, I append a sigmoid layer to convert the values into decision values.

The input image is passed through the network to obtain the probability of that image being beautiful. I compute the results on a subset of $10,000$ images taken from the test set and the entire standard test set (Murray *et al.*, 2012). On the test set, proposed approach obtains 59.41% classification accuracy as compared to 59.92% obtained by RAPID. On the standard test set, the proposed approach obtain an accuracy of 71.60% as compared to 69.18% obtained by RAPID. *Note that no additional training has been performed to adopt the network for classification*, which shows that the learned features may be capturing the characteristics that are responsible for making an image aesthetic. The proposed network outperforms RAPID on the ranking task and produces a competitive performance on the classification task without any additional training. Note that the performance of both the networks is significantly lower on the test-set as compared to that of on the standard test-set. This performance difference could be attributed to the fact that all images in the standard test-set are distributed only over eight categories, whereas the images in the test-set are distributed over all 65 categories. The results of all the experiments are summarized in Table 5.2

Fig. 5.2 illustrates some ranking results obtained by the proposed network. The wrong predictions in the bottom row show that the network lacks semantic knowledge about objects and natural phenomena. For example, even though the picture containing two birds has better color harmony/contrast, the lightning phenomena is a rare capture, making it more picturesque.

5.6 Discussion

In this chapter [3] , I introduced a novel problem of relative aesthetics which could have widespread applications in image search, enhancement, retrieval, etc. I created a dataset with

[3]Most of the material in this chapter has appeared in the paper (Chandakkar *et al.*, 2016). See the full credit statement in the appendix. I and my co-author equally contributed to this paper.

a careful and constrained selection of $43,000$ pairs of images from the AVA dataset where one image is always more beautiful than the other. I showed that a deep neural network trained with an appropriate loss function which accounts for such relatively-labeled data significantly outperforms a state-of-art network trained on same data with categorical labels. The proposed network is also able to achieve a competitive performance on an aesthetics classification problem with trivial modifications to its architecture and no fine-tuning at all. This shows that it has gained a certain semantic-level understanding of the factors involved in making an image aesthetic.

We will now discuss a case where the true labels of an image are reversed i.e. for any pair of Image A and B, if the true ranking is "A better than B", we change that to "B better than A"). Now, the question is will the model still learn (i.e. after re-training) a consistent ranking function except that the ranking is reversed? That is, can the model now tell (with high probability of being correct) an image C is better than an image D when the true label is "D better than C"?

The loss function is,

$$L = \max(0, \delta - y \cdot (w \cdot (C_1 - C_2))), \tag{5.4}$$

The reversed labels are represented by $\hat{y} = -y$. The loss function (with the reversed labels) becomes,

$$L = \max(0, \delta - \hat{y} \cdot (w \cdot (C_1 - C_2))), \tag{5.5}$$

Consider a case where the true label is 1 indicating "image A better than image B". However, after reversal the label becomes -1 indicating "image B better than image A". The loss function when the reversed label is -1 is $L = \max(0, \delta + (w \cdot (C_1 - C_2)))$. To back-propagate a non-negative loss, it is now necessary that $w^T C_1 < w^T C_2 - \delta$. After sufficient

learning iterations the model would learn to predict -1 since $sign(w \cdot (C_1 - C_2)) = -1$. Thus the model will predict that "image B is better than image A" with a high confidence. Similar argument can be made if we start with a true label of -1.

Chapter 6

EMPLOYING DEEP FEATURES TO CAPTURE LOCALIZED IMAGE ARTIFACTS

6.1 Problem Introduction

In this chapter, I will employ deep networks to capture localized image artifacts. Before we dive in and see the particular challenges posed by this task, I will briefly describe the areas that deep networks have excelled in. CNNs have surpassed the performance of previous state-of-art approaches by significant margins in various fields. For example, CNNs have shown considerable superiority in object recognition (Simonyan and Zisserman, 2014; He *et al.*, 2015), face recognition (Taigman *et al.*, 2014), semantic segmentation (Dai *et al.*, 2016) etc. The previous chapter showed that CNNs could assess aesthetics value of images, which is a subjective task and is quite different from tasks such as object detection. CNNs have also shown good results in some understudied but long-standing and difficult problems such as gaze-following (Recasens *et al.*, 2015). Since the advent of (Krizhevsky *et al.*, 2012), deep network architectures have also been continuously evolving for tasks such as segmentation (Zheng *et al.*, 2015), object detection and image classification (He *et al.*, 2015; Goodfellow *et al.*, 2014; Salimans *et al.*, 2016).

However, training CNNs to characterize localized image artifacts and label the image accordingly on a relatively small dataset remains a challenging task. With large amounts of data, deep CNNs may be able to learn a good representation for localized artifacts using a conventional pipeline (i.e., end-to-end training on images). Unfortunately, there are many applications where the labeled data is scarce, and the only way to obtain more data is by employing human experts, which is expensive and subject to their availability. This real-world constraint hinders the widespread use of advanced CNN architectures in

(a)

(b)

(c)

Figure 6.1. (a) and (b) Clean (Left) and Distorted (Right) Image Pairs in the TID 2013 Dataset. The Images on the Right in (a) and (b) Are Distorted by Non-uniform and Uniform Noise Respectively. (c) Authetic and Forged Image Pair from CASIA v2.0 Dataset. Red Overlay Shows the Distorted/Forged Regions in an Image. Please Zoom in to See Details and View the Online Version.

such problems. On the other hand, the nature of some of these problems may exhibit properties that can be leveraged to increase the localization power as well as the volume of useful training data. For example, the images can be divided into smaller patches, and the labels of these patches could be derived from the original image, the number of labeled training samples could be increased potentially by a factor of, say 10-100, depending on how the patches are formed. Then CNNs could be trained on the augmented data, and image-level results could be derived by averaging patch-level results. Such patch-level training followed by averaging is the current state-of-art for the problem of no-reference image quality estimation (NR-IQA) (Kang *et al.*, 2014) (see Section 6.4 for details on NR-IQA).

The effectiveness of this patch-level training technique is only observed if one would assume that the artifacts are uniformly spread over the entire image, which is unfortunately too strong a constraint in practice. Certain real-world problems such as NR-IQA and image forgery classification are good examples where these strong constraints are often violated. Fig. 6.1a and 6.1b shows NR-IQA examples containing original (left) and distorted (right) images. The red overlay shows the distorted region in both the images. The distortions are localized, and thus only few image patches are responsible for degrading its quality. Note that in the bottom image, the flower in the salient central region is distorted whereas, in the upper image, some parts towards the lower non-salient region are distorted, preserving the quality of salient face. This affects the perceived image quality as the top image (score = 5.56) has been judged to be of higher quality than the bottom one (score = 3.53) based on the extensive subjective tests conducted in (Ponomarenko *et al.*, 2015). *Interestingly, the type and the severity of the distortion added are identical for both images*. Thus the image quality is dependent on various factors such as distortion type, severity, affected patch locations and their texture, etc. This cannot be handled by the aforementioned patch-level training technique. Similar observations can be made about Fig. 6.1c. The image needs

to be categorized into authentic or forged. The forgery is localized (shown by the red bounding box) and may not be effectively captured by a conventional deep CNN-pipeline or independent patch-level training, especially when the training data is only in the order of thousands. Patch-level training is only effective in case of uniform distortion as shown in Fig. 6.1b.

To combat these scenarios, I present a novel CNN-based approach focused towards such type of data (Chandakkar and Li, 2017a). The proposed approach does not require the image and its patches to share a similar distribution. It works well even if there is only one patch per image which plays an important role in determining the image label (relaxing the requirement that all patches from an image should each contribute to the decision, e.g., employing patch-result averaging). I evaluate the approach on one synthetic data and two real-world, challenging vision tasks - NR-IQA and image forgery classification. I will demonstrate that the proposed method produces a state-of-art performance for both the applications.

6.2 Problem Setup

I consider a problem where the data has some localized information embedded in it that is crucial to getting the desired output. Additionally, the quantity of available data is limited, and its artificial generation is non-trivial or even impossible. I assume that inferring important patches is possible or this information is provided as ground-truth.

Consider a database of images $\mathcal{X}$ containing N images, and their labels, denoted by $(X_i, Y_i) \; \forall \; i$. An image has m patches, $x_i^1, ..., x_i^m$. Here, x_i^j denotes j^{th} patch in the i^{th} image. I denote the patch-level labels by $y_i^1, \ldots, y_i^m$. Patch-level labels can be inferred from the image, or they can be a part of ground-truth labels. Thus training on patches and then averaging all the patch scores to obtain the image label is a naïve way which, actually, works reasonably well in practice. However, the learner has not been fully-equipped to understand

the relation between patch scores and the image score. In other words, the network cannot learn an optimal weighing strategy for all image regions, especially when only a few regions contain the important localized artifacts. Training on the entire image with a deep stack of convolutional filters may achieve the desired result, but then limited amounts of data prevent CNN from generalizing well. Thus the problem becomes: given an image X_i and its patches $x_i^1, \ldots, x_i^m$, first obtain the patch-level labels - $y_i^1, \ldots, y_i^m$. Subsequently, develop a CNN framework which aggregates the information from the collection $(x_i^j, y_i^j)\ \forall i, \forall j$ and forms a mapping function $f : X_i \rightarrow Y_i,\ \ \forall\, i$.

6.3 Proposed Approach

The proposed approach has two CNN stages out of which the first one is trained on a collection of labeled patches. Given a database of labeled images - $\mathcal{X}, \mathcal{Y}$ - I extract all the patches and derive their corresponding labels. As mentioned earlier, the patch-level labels are either inferred from the image label or are provided as ground-truth, if available.

6.3.1 Training the First Stage

The first stage CNN follows a conventional training pipeline and is trained on a collection of labeled patches. I detail the CNN architectures, preprocessing techniques used and the specifics of the training procedure in Section 6.4 as they vary by application.

It is well-known and empirically verified that deeper layers encode increasingly powerful features (Zeiler and Fergus, 2014; Yosinski *et al.*, 2015) that are semantically more meaningful (Zhou *et al.*, 2014). Thus after training the first stage, I extract the ReLU-activated responses of the last but one layer for all the patches in the train and validation set. During my experiments, I observed that ReLU-activated sparse response provides better validation loss than the pre-ReLU responses. This will be used as the input representation for the second stage described as follows.

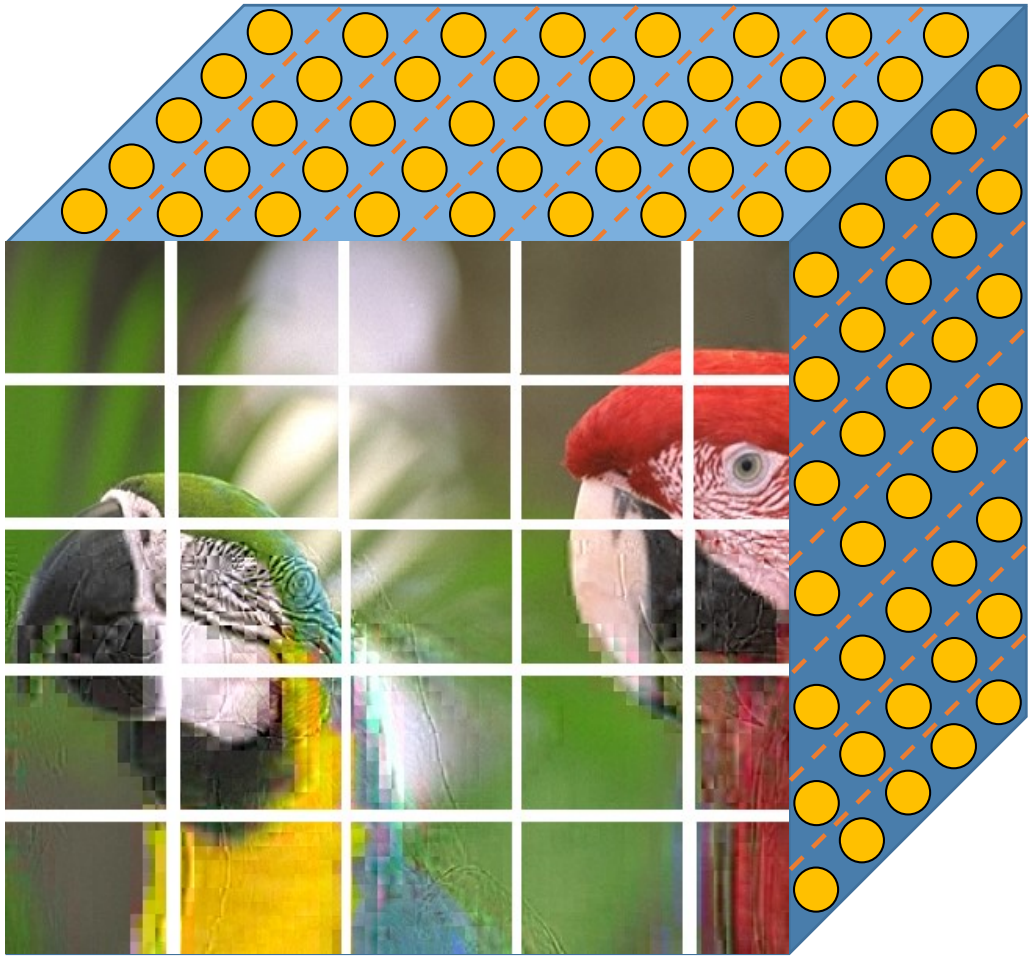


Figure 6.2: Illustration of a Hyper-image. The Yellow Circles along the Depth Axis Denote the D-dimensional Representation for That Patch.

6.3.2 Training the Second Stage

A trained first stage CNN provides a D-dimensional representation obtained from the last but one layer for any given image patch. Let us denote the m overlapped patches for the i^{th} image by $x_i^1, \ldots, x_i^m$. I extract the D-dimensional representations from these m patches and arrange them in a $U \times V \times D$ hyper-image denoted by H_i. The arrangement is done such that the (u, v) element of the hyper-image - H_i^{uv} - corresponds to the representation of (u, v) patch in the original image. In other words, each patch in the original image now

corresponds to a point in D-dimensions in the hyper-image. Note that U and V are smaller as compared with the image height and width respectively, by a factor proportional to the patch size and overlap factor. An illustration of the hyper-image can be seen in Fig. 6.2.

Now, a second stage CNN can be trained on the hyper-images and their labels. This CNN shares architectural similarities with its counterpart - the first stage CNN. I do not perform mean-centering or any other pre-processing since these representations are obtained from normalized images.

Each pixel in the hyper-image being inputted to the second stage CNN is of the form $H_i^{uv} = f_1(f_2(\ldots(f_n(x_i^{uv}))\ldots))\ \forall\ u, v$, where $f_1, \ldots, f_n$ represent the non-linear operations (i.e. max-pooling, convolutions, ReLUs etc.) on an image as it makes a forward pass through the first stage. Then in the second stage, the label is inferred as, $y_i = g_1(g_2(\ldots(g_n(H_i))\ldots))$, where H_i denotes the hyper-image being inputted to the second stage corresponding to the i^{th} image and $g_1, \ldots, g_n$ denote the non-linear operations in the second stage. The following equation expresses y_i as a highly nonlinear function of $x_i^{uv}\ \forall\ u, v$.

$$\begin{aligned} y_i &= g_1(g_2(\ldots(g_n(\{H_i^{11}, \ldots, H_i^{uv}\}))\ldots)), \\ \text{where, } H_i^{uv} &= f_1(f_2(\ldots(f_n(x_i^{uv}))\ldots))\ \forall\ u, v \end{aligned} \tag{6.1}$$

This allows the multi-stage CNN to take decisions based on context and by jointly considering all the image patches. Both the stages combined provide higher representational capacity than had a single stage been trained merely with patches followed by averaging.

Note that the convolutional filters of the second stage CNN only operate on a small neighborhood at a time (usually 3×3), where each point in this neighborhood corresponds to the D-dimensional representation of a patch. So if the receptive field of a neuron in the first layer is 3×3, it is looking at nine patches arranged in a square grid. Thus filters in the early layers learn simple relations between adjacent patches whereas the deeper ones will start pooling in patch statistics from all over the image to build a complex model which

relates the image label and all patches in an image.

This two-stage training scheme raises an issue that it needs the first stage to produce a good representation of the data since the second stage is solely dependent on it. Any errors in the initial stage may be propagated. However, in the experiments, I find that the accuracy of the entire architecture is always more than the accuracy obtained by:

1. Training end-to-end with a deep CNN.
2. Training on patches and averaging patch scores over the entire image.
3. End-to-end fine-tuning of a pre-trained network.

This points to two possibilities. Firstly, the second stage is powerful enough to handle slight irregularities in the first stage representation. This is expected since CNNs are resilient even in the presence of noise or with jittered/occluded images to some extent (Dodge and Karam, 2016). Secondly, inputting a hyper-image containing D-dimensional points to a CNN results in a performance boost. For example, predicting a quality score for the distorted images shown in Fig. 6.1a and Fig. 6.1b can be viewed as a regression task with multiple instances. A single patch having a lower quality will not necessarily cause the entire image to have a lower quality score. Similarly, an image with multiple mildly distorted patches at corners may have higher quality score than an image with a single severely distorted patch placed in a salient position. Thus quality score is related to the appearance/texture of all patches, their locations, distortion severity, etc. The proposed network acquires location sensitivity to a certain extent (shown in experiments on synthetic data) as it learns on a $U \times V \times D$ grid. Distortion strengths of the individual patches are encoded in the D-dimensions, which are unified by the second stage to get desired output for a given image. On the other hand, an end-to-end conventional network pipeline will need to learn both - 1. the discriminative feature space and 2. the patches which contribute to the image label. The patch-averaging technique will fail as it will assign equal weight to all the individual patches potentially

containing distortions of different strengths. To summarize, the proposed architecture attempts to learn the optimal mapping between image label and spatially correlated patches of that image.

6.3.3 Testing

Initially, a fixed number of patches from an image are extracted. Their representations are computed using the first stage and arranged in a $U \times V \times D$ grid to form a hyper-image. The proposed approach does not require resizing of input images. Instead, the strides between patches are changed at run-time to obtain the desired shape. In applications where resizing images could change the underlying meaning of images or introduce additional artifacts (e.g., image quality, image forgery classification), this approach could be useful. The procedure to compute the strides at run-time is as follows. I first compute (or assume) the maximum height and width of all the images in the dataset. If a test image exceeds those maximum dimensions, I scale it isotropically so that both its dimensions meet the size constraints. Let the maximum height and width be denoted by M and N respectively. Thus the number of patches required in the x and y direction of the grid are $n_{p_x} = \lceil \frac{N}{sz_x} \rceil, n_{p_y} = \lceil \frac{M}{sz_y} \rceil$. Here, $sz_y \times sz_x$ is the patch size that is used in the first stage. For any new $\hat{M} \times \hat{N}$ image, the required strides (denoted by s_x and s_y) to obtain a fixed number of patches in the grid are as follows:

$$s_x = rnd\left(\frac{\hat{N} - sz_x}{n_{p_x} - 1}\right), s_y = rnd\left(\frac{\hat{M} - sz_y}{n_{p_y} - 1}\right) \tag{6.2}$$

After obtaining the hyper-image $\in \mathbb{R}^{U \times V \times D}$ from the first stage, it is forward propagated through the second stage to obtain the desired output. In the upcoming sections, I apply the proposed approach to a synthetic problem and two real-world challenging vision tasks, review relevant literature and discuss other aspects of the proposed approach.

Table 6.1: Results of Experiments on Synthetic Data

Experiment 1 on synthetic data		
Approach	SROCC	PLCC
Patch-averaging	0.9132	0.8982
Proposed	$\mathbf{0.9611}$	$\mathbf{0.9586}$

Experiment 2 on synthetic data	
Approach	Mean of SROCC and PLCC
Patch-averaging	$0.665, 0.7, 0.544, 0.5, 0.439$
Proposed	$\mathbf{0.886}$,$\mathbf{0.811}$,$\mathbf{0.738}$,$\mathbf{0.744}$,$\mathbf{0.718}$

6.4 Experiments and Results

I evaluate my approach on a synthetic task and two challenging real-world problems - 1. no-reference image quality assessment (NR-IQA) and 2. image forgery classification. Apart from the dependence on localized image artifacts, an additional common factor which aggravates the difficulty level of both these problems is that the amount of data available is scarce. Manual labeling is expensive as subject experts need to be appointed to evaluate the image quality or to detect forgery. Additionally, artificial generation of data samples is non-trivial in both these cases. I will begin by describing the setup for the synthetic task.

Synthetic task: While this task is constructed to be simple; I have included certain features that will examine the effectiveness of the proposed approach. The task is to map an image to a real-valued score that quantifies the artifacts introduced in that image. The dataset contains 128×128 gray-scale images that have a constant colored background. The color is chosen randomly from $[0, 1]$. I overlay between one to five patches on that image. Each patch contains only two shades - a dark one $\in [0, 0.5)$ and a bright one $\in [0.5, 1]$. A random percentage of pixels in a patch is made bright, and the others are set to dark. Finally,

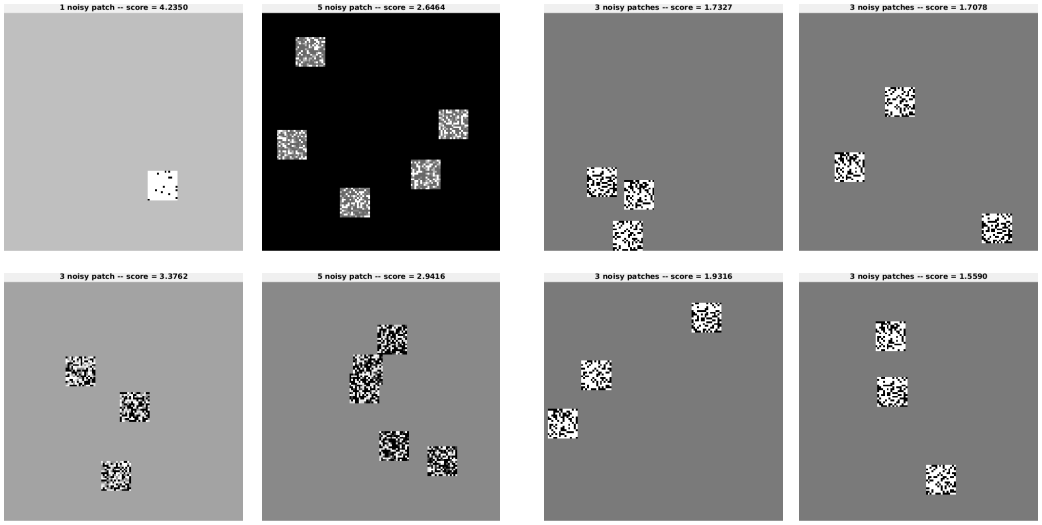


Figure 6.3: Images Used in Both the Synthetic Tasks. Left 2×2 Grid Shows the Images Used in the First Task and the Other Grid Shows Images Used in the Second Task.

all the pixels are scrambled. Size of each patch is 16×16. Some of the images belonging to the first synthetic task are shown in Fig. 6.3.

Let the synthetic image be denoted by S and the i^{th} patch by p_i. The number of patches overlaid on S is η (≤ 5). Let s_0 denote the constant background value of S. p_i^{jk} denotes the (j, k) pixel of p_i. The center of the patch p_i is denoted by c_i. The image center is denoted by $\hat{c}$. The score of this image can now be defined as $5 - \sqrt{\sum_{i=1}^{\eta} \alpha_i^2}$, where α_i is:

$$\alpha_i = \left(\sum_{j=1}^{16} \sum_{k=1}^{16} \left(\frac{|s_0 - p_i^{jk}|}{16 \times 16} \right) \right) + \left(1 - \frac{||c_i - \hat{c}||_2}{\text{dist}_N} \right) \tag{6.3}$$

The first term computes the Manhattan distance between a background pixel and all the pixels of a patch. This can be viewed as a dissimilarity metric between the background and a patch. The second term imposes a penalty if the patch is too close to the image center. The term dist_N is a normalization factor. If the patch lies at any of the four corners, then $||c_i - \hat{c}||_2 = \text{dist}_N$ and the penalty reduces to zero. A Higher score indicates the presence of lower artifacts in an image.

I perform another experiment to test the sensitivity of the proposed approach to the patch locations. To this end, I created 1K images that all had an identical background. The number of patches and their content (i.e., two shades and pixel scrambling) was also fixed across all 1K images. The only variable was their positions in the image. I compared the two-stage approach and patch-averaging. Patch-averaging assigns nearly identical scores to all 1K images whereas the proposed approach gives higher correlation with the ground truth. When everything except patch positions was fixed, the proposed approach had higher correlation and slow degradation with increasing number of patches. See Table 6.1 for results. See Fig. 6.3 for example images belonging to the second task.

No-reference image quality assessment (NR-IQA): Images may get distorted due to defects in acquisition devices, transmission-based errors, etc. The task of IQA requires us to build an automated method to assess the visual quality of an image. Conventional error metrics such as RMSE/PSNR cannot capture the correlation between the image appearance and human-perception of the image quality. Two variants of this problem exist - full-reference IQA and no-reference IQA (NR-IQA). In the former task, an original image and its distorted counterpart are given. A quality score needs to be assigned to the distorted one with respect to the original image. Few representative approaches that try to solve this problem are SSIM (Wang *et al.*, 2004), MSSIM (Wang *et al.*, 2003), FSIM (Zhang *et al.*, 2011a), VSI (Zhang *et al.*, 2014) etc. However, in real-world scenarios, one may not have a perfect, non-distorted image available for comparison. Thus NR-IQA variant has been proposed. In NR-IQA, we are given a single image that needs to be assigned a quality score with respect to a non-distorted, *unobserved* version of that image. This score must correlate well with human perception of image quality. While creating ground-truth for this problem, a constant value is associated with a non-distorted image. This serves as a reference on the quality score scale. This problem involves developing a discriminative feature space to different kinds and degrees of distortions. Such setting is more suitable for learning schemes,

which is reflected by the fact that most approaches used to tackle this problem belong to the learning paradigm. Few of the representative approaches include BRISQUE (Mittal *et al.*, 2012), CORNIA (Ye *et al.*, 2012, 2013), DIIVINE (Moorthy and Bovik, 2011), BLIINDS (Saad *et al.*, 2012), CBIQ (Mittal *et al.*, 2013), LBIQ (Tang *et al.*, 2011) and the current state-of-art, a CNN-based approach (Kang *et al.*, 2014).

I perform all the NR-IQA experiments on two widely used datasets - 1. LIVE (Sheikh *et al.*, 2005) containing 29 reference images, 779 distorted images, 5 distortion types and 5-7 distortion levels. The images are gray-scale. Through subjective evaluation tests, each user has assigned a score to an image according to its perceptual quality. A metric named difference of mean opinion scores (DMOS) $\in [0, 100]$ was then developed, where 0 indicates highest quality image. 2. TID 2013 (Ponomarenko *et al.*, 2015) has 25 reference RGB images, 3000 distorted images, 24 distortion types and 5 distortion levels. In subjective tests, each user was told to assign a score between 0-9 where 9 indicates best visual quality. Mean opinion scores (MOS) were calculated and were provided as ground truth scores/labels for each image. Four of the 24 distortions are common to LIVE and TID datasets. LIVE has all uniform distortions whereas 12 distortions out of total 24 in TID 2013 are non-uniform. See Fig. 1 for an example of uniform/non-uniform distortion. The aim is to learn a mapping between the images and the scores which maximizes Spearman's (SROCC), Pearson's correlation coefficient (PLCC) and Kendall's correlation as those are the standard metrics used for IQA. Since the number of images is so small, I run the proposed and the competing algorithms for 100 splits to remove any data-split bias in all four experiments. As a widely followed convention in IQA, I use 60% of reference and distorted images for training, 20% each for validation and testing everywhere.

The subjective tests conducted for these datasets are extensive. Extreme care was taken to make them statistically unbiased, and it is non-trivial to reproduce them. LIVE data creation needed more than 25K human comparisons for a total of 779 images. TID 2013

data creation had over $1M$ visual quality evaluations and $524,340$ comparisons. To avoid geographical bias, the participants came from five countries. In summary, the constraint of small data in such applications comes from the real-world, and it is difficult to generate new data or conduct additional tests.

In all the experiments, before I feed training images to the first stage CNN, I preprocess it following the approach of (Mittal *et al.*, 2012) that performs local contrast normalization as follows.

$$\begin{aligned} \hat{I}(i,j) = \frac{I(i,j) - \mu(i,j)}{\sigma(i,j) + \epsilon},\ \mu(i,j) = \sum_{k,l=-3}^{3} w_{k,l} I_{k,l}(i,j), \\ \text{and } \sigma(i,j) = \sqrt{\sum_{k,l=-3}^{3} w_{k,l} \left(I_{k,l}(i,j) - \mu(i,j)\right)^2}. \end{aligned} \tag{6.4}$$

Here, w is a 2-D circular symmetric Gaussian with three standard deviations and normalized to unit volume.

Data generation: The data preparation method of (Kang *et al.*, 2014) is common to both datasets. It extracts overlapping 32×32 patches and then assigns them equal score as that of the image. This strategy works well for (Kang *et al.*, 2014) as they only handle LIVE and specific TID distortions that are shared by LIVE. However, to handle non-uniform distortions, I make a slight but important modification to their method. I compare the *corresponding* patches of the original image and its distorted counterpart with the help of SSIM (Wang *et al.*, 2004). SSIM is used to measure the structural similarity between two images and is robust to slight alterations, unlike RMSE. I keep all the patches belonging to a distorted image that has low SSIM scores with their original counterparts. This indicates low similarity between patches which could only point to distortion. For patches belonging to reference (or clean) images, I select them all. Finally, I make the number of reference and distorted patches equal. I call this method as *selective patch training* (SPT). I now describe certain protocols common to all the NR-IQA experiments.

Table 6.2: Architectures of the Deep Networks Used. The Term $C(n)$ Denotes 3×3 "Same" Convolutions With Stride 1. MP(N) Is a Max-pooling That Reduces the Image Size by a Factor of n. FC(n) and Drop(n) Denote a Dense Layer with n Neurons and a Dropout Rate of n Respectively.

Architecture	Layer descriptions
Synthetic stage 1	Input$(128, 128)$ - C(16) - MP(2) - C(32) – MP(2) - 2 × C(48) - MP(2) - 2 × C(64) - MP(2) -2 × C(128) - MP(2) - FC(400) - Drop(0.5) - FC(400) - Drop(0.5) - FC(1,'linear')
Synthetic stage 2	Input(10,10,400) - 2 × C(16) - MP(2) - 2 × C(32) - MP(2) - 2 × C(64) - MP(2) - FC(400) - Drop(0.5) - FC(400,'tanh') - Drop(0.5) - FC(1,'linear')
LIVE/TID stage 1	Please refer to (Kang *et al.*, 2014).
LIVE stage 2	Input(24,23,800) - 2 × C(32) - MP(2) - 2 × C(48) - MP(2) - 2 × C(64) - MP(2) - 2 × C(128) - MP(2) - FC(500) - Drop(0.5) - FC(500,'tanh') - Drop(0.5) - FC(1,'linear')
TID stage 2	Input(23,31,800) - 2 × C(64) - MP(2) - 2 × C(64) - MP(2) - 2 × C(128) - MP(2) - 2 × C(128) - MP(2) - FC(500) - Drop(0.5) - FC(500,'tanh') - Drop(0.5) - FC(1,'linear')
Forgery channel	Input(64,64,3) - 2 × C(64) - MP(2) - 2 × C(128) - MP(2) - 2 × C(128) - MP(2) - 2 × C(256) - MP(2) - FC(500) - Drop(0.5) - FC(500) - Drop(0.5)
Forgery stage 2	Input(15,15,500) - 3 × C(64) - MP(2) - 3 × C(128) - MP(2) - 3 × C(256) - MP(2) - FC(800) - Drop(0.5) - FC(800) - Drop(0.5) - FC(1,'sigmoid')
Forgery end-to-end CNN	Input(256,384,3) - C(32) - MP(2) - C(64) - MP(2) - 2 × C(64) - MP(2) - 2 × C(128) - MP(2) - 2 × C(128) - MP(2) - 2 × C(256) - MP(2) -FC(500) - Drop(0.5) - FC(500) - Drop(0.5) - FC(1,'sigmoid')

Training/testing pipeline: The method of (Kang *et al.*, 2014) trains on 32×32 patches and averages patch-level scores to get a score for an entire image. I train the first stage on 32×32 patches obtained by my selection method. The second stage is then trained by hyper-images that are formed using the first-stage patch representations. In the first three NR-IQA experiments, I use the same first stage as that of (Kang *et al.*, 2014) to be able to assess the impact of adding a second stage. Addition of a second stage entails little overhead as on LIVE (TID) data; one epoch requires 23 (106) and 3 (43) seconds for both stages respectively. Both the stages as well as (Kang *et al.*, 2014) uses mean absolute error as the loss function.

All the networks are trained using SGD with initial learning rate 0.005 and decaying at a rate of 10^{-5} with each update. Nesterov momentum of 0.9 was also used. The learning rate was reduced by 10 when the validation error reached a plateau. The first stage was trained for 80 epochs, and training was terminated if no improvement in validation error was seen for 20 epochs. The second stage was trained for 200 epochs with the termination criterion set at 40 epochs. Implementations were done in Keras (Chollet *et al.*, 2015) (with Theano (Theano Development Team, 2016) as a backend) on an Nvidia Tesla K40. I now describe individual experiments.

Experiment 1: I evaluate the proposed and the competing approaches on the LIVE data. The architecture used is given in Table 6.2. The input sizes for both the stages are 32×32 and $24 \times 23 \times 800$ respectively. Even though LIVE contains only uniform distortions, the proposed approach marginally improves over (Kang *et al.*, 2014) over 100 splits. This could be due to the better representational capacity of the two-stage network as all the image patches contain the same kind of distortion. The results obtained for all the approaches are given in Table 6.3.

Experiment 2: Intuitively, the SPT should give us a significant boost in case of TID 2013 data. Since only a few patches are noisy, assigning the same score to all patches will

corrupt the feature space during training. To verify this, I train on TID 2013 data using the approach of (Kang *et al.*, 2014) - with and without the selection strategy. Input sizes for both stages are $32 \times 32 \times 3$ and $23 \times 31 \times 800$. I also evaluate the two-stage network to show its superiority over both these approaches. I find that SPT boosts Spearman (SROCC) by 0.0992 and Pearson correlation coefficient (PLCC) by 0.072. Two-stage training further improves SROCC by 0.0265 and PLCC by 0.0111. The architecture and results are in Table 6.2 and 6.3 respectively. *From now on, I compare with (Kang* et al.*, 2014) assisted with the SPT to understand the benefits of the second stage.*

Experiment 3: First, I take the four distortions from TID that are shared with LIVE (thus these are uniform). I observe a marginal improvement here for similar reasons as the first experiment. The second part is designed to show the adverse effects of non-uniform, localized distortions on the correlations. Out of 24 distortions, there are four common ones. I add just two most non-uniform distortions - 1. Non-eccentricity pattern noise and 2. Local block-wise distortions. On these six distortions, the proposed approach significantly outperforms that of (Kang *et al.*, 2014) with patch selection. Thus to characterize non-uniform distortions, one needs to weight every patch differently, which is exactly what the second stage achieves. Finally, in the third part, I test on the entire TID 2013 data. To the best of my knowledge, no other learning-based approach has attempted the entire data. The only approach I am aware of that tested on TID is CORNIA (Ye *et al.*, 2012, 2013). However, even they skip two kinds of block distortions. The reasons could be lack of training samples or the severe degradation in performance as observed here. I compare the proposed approach with the approach of (Kang *et al.*, 2014) augmented with SPT. The detailed results are listed in Table 6.3. The architecture used was identical to that used in the second experiment.

Experiment 4: I verify that training networks end-to-end from scratch gives a poor performance with such low amounts of training data. I define a shallow and a deep CNN

of 8 and 14 layers respectively and train them end-to-end on 384×512 images from TID 2013. Out of all the experiments, this produces the worst performance, making it clear that end-to-end training on such small data is not an option. See Table 6.3 for results. I provide these CNN architectures in the supplementary material for conciseness.

Experiment 5: A popular alternative when the training data is scarce is to fine-tune a pre-trained network. I took VGG-16, pre-trained on ILSVRC 2014. I used it as the first stage to get patch representations. VGG-16 takes 224×224 RGB images whereas I have 32×32 RGB patches. Thus I only consider layers till "conv4_3" and get its ReLU-activated responses. All the layers till "conv4_3" reduce a patch to a size of $2 \times 2 \times 512$. I append two dense layers of 800 neurons each and train them from scratch. Rest of the layers are frozen. Please refer to the Caffe VGG prototxt for further architectural details. To train this network, I use a batch size of 256 and a learning rate of 0.01. I average the patch scores obtained from fine-tuned VGG and compute the correlations over 5 splits. In principle, I should get a performance boost by appending the second stage after VGG, since it would pool in VGG features for all patches and regress them jointly. I use a second stage CNN identical to the one used in experiment 2. I observe that SROCC and PLCC improve by 0.06 and 0.0287 respectively. For detailed results, see Table 6.3. On the other hand, I see a sharp drop in performance for VGG despite it being deep and pre-trained on ImageNet. The reasons for this could be two-fold. As also observed in (Kang *et al.*, 2014), the filters learned on NR-IQA datasets turn out to be quite different than those learned on ImageNet. Thus the semantic concepts represented by the deeper convolutional layers of pre-trained VGG may not be relevant for NR-IQA. Secondly, VGG performs a simple mean subtraction on input images versus the pre-processing for this task involves local contrast normalization (LCN). The latter helps in enhancing the discontinuities (e.g., edges, noise, etc.) and suppresses smooth regions, making LCN more suitable for NR-IQA.

The extensive evaluations on NR-IQA show that the proposed approach is better at

Table 6.3: Results of the NR-IQA Experiments

Experiment 1 on LIVE data - 100 Splits		
Approach	SROCC	PLCC
DIIVINE (Moorthy and Bovik, 2011)	0.916	0.917
BLIINDS-II (Saad *et al.*, 2012)	0.9306	0.9302
BRISQUE (Mittal *et al.*, 2012)	0.9395	0.9424
CORNIA (Ye *et al.*, 2012)	0.942	0.935
CNN (Kang *et al.*, 2014) + SPT	0.956	0.953
Proposed	**0.9581**	**0.9585**

Experiment 2 on TID data - 100 Splits		
CNN	0.6618	0.6907
CNN + SPT	0.761	0.7627
CNN + SPT + Stage 2 CNN (proposed)	**0.7875**	**0.7738**

Experiment 3 on select distortions of TID - 100 splits		
# distortions	CNN + SPT (SROCC, PLCC)	**Proposed (SROCC,PLCC)**
Four	0.92,0.921	**0.932,0.932**
Six	0.625,0.653	**0.76,0.755**
All (24)	0.761,0.763	**0.788,0.774**

Experiment 4 on TID data - 10 splits		
Approach	SROCC	PLCC
Shallow end-to-end CNN	0.2392	0.4082
Deep end-to-end CNN	**0.3952**	**0.52**

Experiment 5 on TID using pre-trained VGG - 10 splits		
VGG + patch-averaging	0.6236	0.6843
VGG + second stage CNN	**0.6878**	**0.713**

characterizing local distortions present in an image. It improves on the current state-of-art (Kang *et al.*, 2014) and various other approaches, such as training a shallow/deep network from scratch or fine-tuning a pre-trained network.

Image forgery classification: In today's age of social media, fake multimedia has become an issue of extreme importance. To combat this, it is necessary to improve detection systems to categorize fake posts. Here, I focus on image forgery/tampering, which is defined as altering an image by various means and then applying post-processing (e.g., blurring) to conceal the effects of forging. Image tampering comes in various forms, for example, copy-move forgery (Bayram *et al.*, 2009; Sutthiwan *et al.*, 2011) and manipulating JPEG headers (Farid, 2009a; He *et al.*, 2006). Some other techniques have also been developed to detect forgeries from inconsistent lighting, shadows (Fan *et al.*, 2012; Kee and Farid, 2010) etc. See the surveys for more information (Bayram *et al.*, 2008; Farid, 2009b). However, the problem of tampering detection from a single image without any additional information is still eluding researchers. The current state-of-art uses a block-based approach (Sutthiwan *et al.*, 2011) which use block-DCT features. It forms a Markov transition matrix from these features and finally feeds them into a linear SVM. They carry out their experiments on the CASIA-2 tampered image detection database [1] . It contains 7491 authentic and 5123 tampered images of varying sizes as well as types. I have also done some studies in the past investigating effect of human factors in image forgery detection (Chandakkar and Li, 2014).

Data generation: Given a database of authentic and (corresponding) tampered images; I focus on getting the contour of the tampered region(s) by doing image subtraction followed by basic morphological operations. The resultant contour is shown in Fig. 6.4. I sample 15 pixels along this contour and crop 64×64 patches by keeping the sampled points as the patch-centroids. Similar to (Sutthiwan *et al.*, 2011), I train on $\frac{2}{3}^{rd}$ of the data and use $\frac{1}{6}^{th}$ data each for validation and testing. I subtract the mean of training patches from each

[1] `http://forensics.idealtest.org/casiav2/`

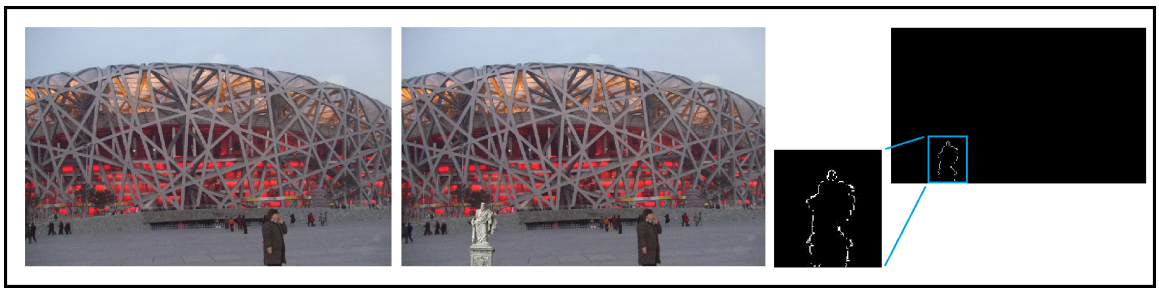


Figure 6.4: Authentic (Left) and Tampered (Middle) Image. The Resultant Contour of the Tampered Region (Right). Please Zoom-in and View in Color.

Table 6.4: Results of Image Forgery Classification

Approach	Classification accuracy
End-to-end CNN	75.22%
Current state-of-art (Sutthiwan *et al.*, 2011)	79.20%
Proposed two stage CNN	**83.11%**

patch and do on-the-fly data augmentation by horizontally-flipping the images. Instead of categorizing *patches* as authentic/tampered, I develop a ranking-based formulation, where the rank of an authentic patch is higher than its counterpart. Note that during testing, a single image is given to be classified as authentic or forged and thus a contour of the forged region cannot be found (or used).

Experiment 1: I train an end-to-end deep CNN that takes an entire image as input and categorizes it as authentic or tampered. The architecture used is shown in Table 6.2. It takes 256×384 RGB images as input. This size is chosen since it needs a minimum number of resizing operations over the dataset. The classification accuracy is shown in Table 6.4.

Experiment 2: The first stage CNN learns to rank authentic patches higher than tampered ones. I propose ranking because every patch may contain different amount of forged region or blur. This CNN has two identical channels that share weights. Its architecture

is shown in Table 6.2. Let the last dense layer features obtained from an authentic and a tampered patch be denoted by C_{Au} and C_{Tp} respectively. A weight vector needs to be learned such that $w^T C_{Au} - w^T C_{Tp} > 0$. However, the network trains poorly if I keep feeding authentic patches into the first channel and the tampered ones into the second channel. Shuffling of patches is necessary to achieve convergence. I assign a label of 1 if two channels get authentic and tampered patches (in that order), else -1. Thus I need $d(C_1, C_2) = w_2^T y \cdot (f(w_1^T \cdot (C_1 - C_2)) > 0$, where C_i is the feature from the i^{th} channel, whereas $y \in \{-1, 1\}$ denotes the label. The transformations achieved through two dense layers and an ReLU are denoted by $w_2(\cdot), w_1(\cdot)$ and $f(\cdot)$ respectively, as shown in Fig. 6.5. The loss function becomes, $L = \max(0, \delta - y \cdot d(C_1, C_2))$. The term $\max(0, \cdot)$ is necessary to ensure that only non-negative loss gets back-propagated. The $\delta(= 3)$ is a user-defined parameter that avoids trivial solutions and introduces class separation.

The first stage representation should discriminate between neighborhood patterns along a tampered and an authentic edge (since I chose patches centered on the contour of the tampered region). Given an image, I extract patches and form the hyper-image required to train the second stage. I use binary labels, where 1 denotes authentic image and vice-versa along with a binary cross-entropy loss. The architecture of the second stage is shown in Table 6.2. To overcome class imbalance, I weigh the samples accordingly. I compare the proposed approach with an end-to-end CNN network (experiment 1) and the current state-of-art in passive, generic image forgery classification (Sutthiwan *et al.*, 2011). CNN-baseline gives the worst performance followed by the current state-of-art. This is expected since the latter extracts *block-level* DCT features whereas the CNN-baseline tries to learn from an entire image - a considerably difficult task especially when the tampered/forged parts are localized and well camouflaged. My hybrid approach beats the CNN-baseline and the state-of-art by 8% and 4% respectively. All these experiments underline the importance of collectively learning from image patches when the data is scarce and shows the flexibility of

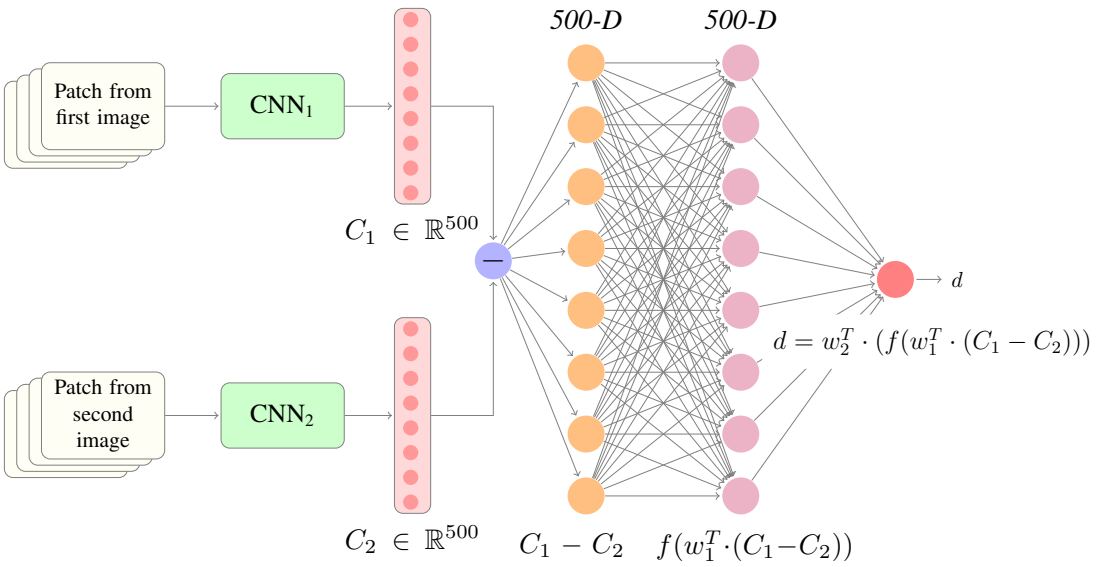


Figure 6.5: Proposed Channel Architecture. Weight Sharing Occurs Between Both Channels. Please Zoom in to See Details.

the proposed approach.

6.5 Discussion

The proposed approach shares certain conceptual similarities with the approach of (Hariharan *et al.*, 2015). They consider a vector of activations of all the CNN units "ahead" a pixel. They refer to this vector as a "hypercolumn" of activations. I consider all the responses of the last fully-connected layer belonging to an image patch. There are some major differences between these two approaches as follows. To make a location-aware classifier, they need to train a 10×10 interpolated, coarse grid of classifiers, whereas my approach needs only one coherent classifier (i.e., the second stage) irrespective of how fine the grid is. They note that training their coarse classifier grid is a hard optimization problem and they ignore this interpolation at training time. Using the notion of hyper-image, I can train the second stage and observe a considerable improvement over training with the whole image or simple patch averaging, feature-pooling, etc. A recent paper uses pre-trained object

detection models such as VGG-19 and simultaneously trains just the fully-connected layers and region-adaptation modules on multiple image regions (Gidaris and Komodakis, 2015). This multi-region protocol gives them a boost in the classification performance on PASCAL VOC 2007 and 2012. In comparison, the proposed approach can be trained on patches first (i.e., the object parts) and then those patches (parts) can be combined to learn the entire object representation. This may handle occlusions and deformations well since the trained first stage may have a better idea of which patches belong to either an object, context or irrelevant background. Also, even if one of the object parts have been deformed/occluded, the other parts can guide the classifier in the second stage in the right direction. Training the first stage, in this case, requires patch-level labels for objects. A promising future direction could be to apply the proposed approach on pixel-level labeled data such as MS-COCO.

I presented the notion of CNN-based hyper-image representations. The training scheme involving these hyper-images excels in scenarios where the label is dependent on the localized artifacts in an image. In these networks, the first stage is only responsible for learning the discriminative representations of small image patches. The second stage collectively considers all the patches of an image, unlike many other previous approaches. It optimally weighs and pools all the patches, and develops a mapping between them and the image label. The proposed approach enables training deep networks with greater representational capacity than their conventional counterparts in specific cases where patch-level labels are available. I observe in all the experiments that the second stage always provides a significant improvement. I apply the approach to a synthetic and two challenging vision tasks - NR-IQA and image forgery classification. The approach comfortably outperforms other CNN-baselines as well as the existing state-of-art approaches.

Chapter 7

FUTURE WORK AND CONCLUSION

7.1 Future Work

In chapter 5 and 6, I presented two visual computing tasks that employ deep features and obtain state-of-art performance. Given the recent success of deep features, it is possible that increasingly large number of applications hosted on smart devices will start delivering results that utilize deep neural networks (DNN). The sensing ability of smart devices and the network connectivity to other such devices makes them attractive for users. Sensors that continuously monitor users' activity coupled with sharing of information with other devices allows them to provide proactive, smart and most importantly personalized suggestions to their users. Their non-trivial computational ability coupled with clever algorithms is what makes them smart.

7.1.1 Problem Introduction

There is great potential in developing on-device, intelligent applications that analyze every aspect of our life and provide proactive suggestions. However, the ubiquity of smart devices combined with DNN-based intelligent applications means increased computational load on the servers. To reduce the ever-increasing load, the applications need to move to the edge of the cloud. This has given rise to a new field called "edge computing" (Shi *et al.*, 2016).

However, the downside of DNNs used in such tasks is that their training requires massive computational resources in order to achieve effective performance (Bhattacharjee *et al.*, 2017). DNNs also require a large storage space. As a result, DNN deployment has not

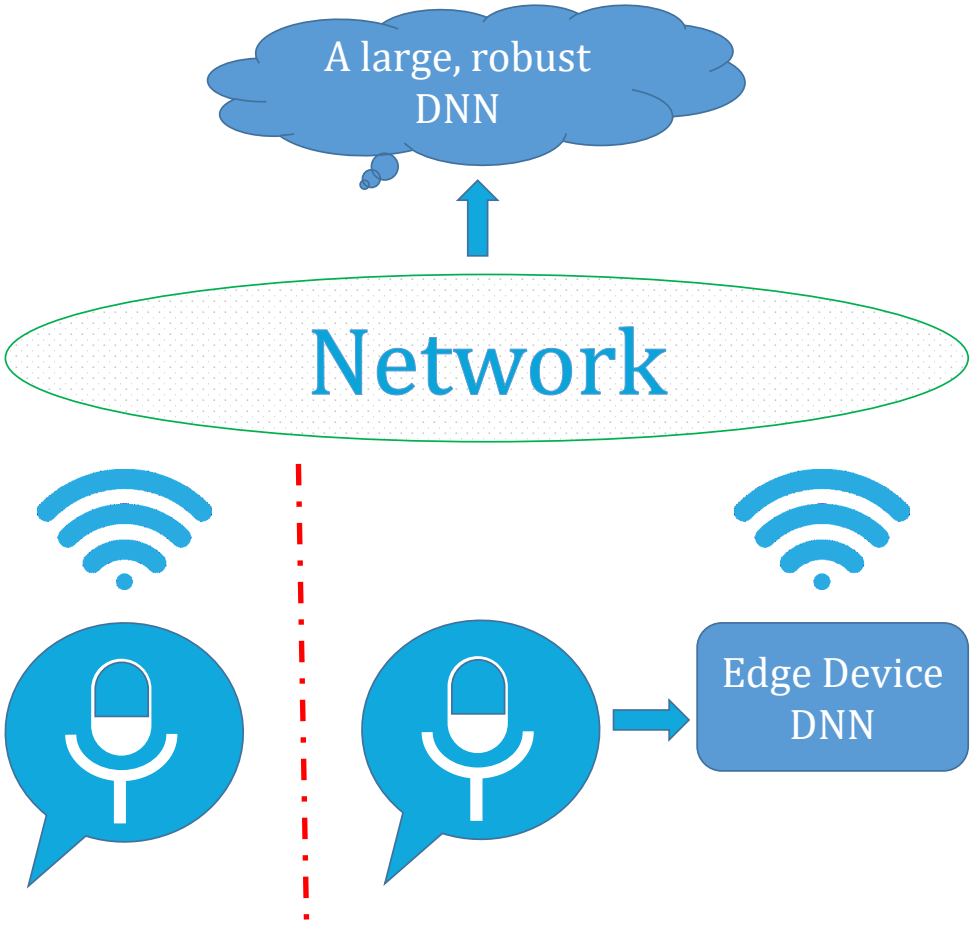


Figure 7.1: Role of DNN Hosted on an Edge Device in Case of Speech Recognition. Left of the Dotted Line Shows a Conventional Speech Recognition Pipeline. On the Right, an Edge Device Could Be Used to First Pre-process the Speech That Normalizes Different Accents and Sends It to the Cloud ©2017 IEEE.

yet moved towards the edge of the cloud. Operating on the edge of the cloud provides several advantages. Firstly, if we were able to train DNNs on edge devices (such as a smartphone), then DNNs could extract and store knowledge from the users' behavior on the source device. This would aid in personalization as DNN develops a tailor-made algorithm for each user depending on his/her needs. In overly complex tasks where a large DNN is unavoidable, this could provide a customized pre-processing step and aids the large DNN

to take a correct decision. For example, in speech recognition, it is unlikely that accents from different geographical parts could be understood equally well by a single DNN. Thus a network hosted on an edge device could extract the acoustic features from its user's speech, pre-process them to aid the large DNN identify the speech in a better manner. To summarize, an edge device DNN could be used to introduce a transformation that increase invariance of the features with respect to different accents. This is shown in Figure 7.1.

Reducing the storage footprint and thereby reducing the computational complexity is key to hosting a DNN on an edge device. Most of the memory in DNNs is consumed by the weight matrices. It is well-known that DNN are typically over-parameterized and thereby their weights have significant redundancy in them (Denil *et al.*, 2013). Storage footprint can thus be reduced by doing weight pruning. A typical pruning procedure removes weights with small magnitude. This has shown to reduce the DNN model size by an order of magnitude (Han *et al.*, 2015a,b). However, DNNs would need to continuously learn from the sensory data in order to provide personalization as discussed earlier. In this work, I study the re-training of the pruned networks, aiming at improving the overall performance of the retrained, pruned network (Chandakkar *et al.*, 2017a).

I show that modifying the pruning strategies before re-training helps the DNN to better generalize to new data while minimizing the performance reduction on the original data.

7.1.2 Related Work

Deployment of DNNs on embedded systems have attractive prospects (Han *et al.*, 2016, 2017). One of the early works applies singular value decomposition (SVD) to a pre-trained model to achieve weight compression (Denton *et al.*, 2014). Magnitude-based weight pruning was introduced in (Han *et al.*, 2015b,a). The authors observed that many weights have small values that produce negligible output response. Making these weights zero could remove connections between neurons which saves memory. Adaptive quantization and

weight sharing can also be applied to reduce the number of bits needed per weight. Huffman coding has also been explored to quantize the weights in (Han *et al.*, 2015a).

Modifying the original architecture of large DNNs is explored in (Iandola *et al.*, 2016). The modification is based on certain guidelines such as replacing most 3×3 filters with those of 1×1, thereby saving $9\times$ parameters. Delayed downsampling is employed to produce large activation maps early on in the network that helps maximize accuracy with a given number of parameters (Iandola *et al.*, 2016). Other line of research includes developing specialized hardware accelerators (Han *et al.*, 2016, 2017).

Binarized neural networks (BNN) that have binary weights and activations (1 or -1) are proposed in (Courbariaux *et al.*, 2016). Only real-valued quantities in these networks are the gradients that are obtained through standard DNN optimization algorithms such as stochastic gradient descent or Adam. BNN reduces time complexity by almost 60%.

All the network compression techniques mentioned above are useful for cases where the only purpose of DNN is to make inference. However, in the proposed problem, a DNN is deployed on an edge device that is supposed to constantly learn from a dynamic environment. In the upcoming section, I describe the proposed DNN pruning and re-training strategies and then compare them with some obvious baselines.

7.1.3 Proposed Approach

The focus of this task is on re-training a pruned DNN that will maximize performance on the new data while minimizing the performance reduction on the old data. I describe the weight pruning in detail.

There is a large redundancy in parameters of a DNN (Denil *et al.*, 2013). Thus a magnitude-based weight pruning method was proposed in (Han *et al.*, 2015b). As mentioned before, the method is only suitable if we were to just deploy (and not update) the pruned DNN on an edge device. Consider a simple three layer MLP that two weight matrices

$w(1,2)$ and $w(2,3)$. The response of layer l is denoted by $f(w(l-1,l)^T \cdot x)$, where $f(\cdot)$ is a non-linearity such as ReLU. A loss can be computed between the response of the last layer and the ground-truth labels. The weight matrices can then be updated layer by layer, starting from the last one, using backpropagation. The skeleton of the weight update rule is as follows:

$$\boldsymbol{w}(l-1,l) \coloneqq \boldsymbol{w}(l-1,l) + \alpha * \boldsymbol{\delta}(l) * \boldsymbol{a}(l-1), \tag{7.1}$$

where bold letters indicate matrices throughout this chapter. The gradient step is denoted by α, wheras $\delta(l)$ denotes the error at layer l. All the weights are updated in the above manner. Data for DNN training is usually fed in minibatches. Once all data has been covered, it is called an epoch. A DNN almost always needs multiple epochs depending on its complexity. I develop two different styles of pruning strategies as follows by adapting the weight-magnitude pruning strategy defined in (Han *et al.*, 2015b).

Weight Pruning

Global pruning: Once the network is trained, I set the (i,j) element of a weight matrix between layer $(l-1,l)$ - $\boldsymbol{w}_{i,j}(l-1,l)$ - to zero if it is less than ($T\times$ maximum weight in the entire network), where T is a user defined threshold. I repeat this for all layers. Since the threshold remains the same for all layers, I call this the global pruning method.

Layer-wise pruning: The only difference between this and global pruning is that there are different thresholds for different layers i.e. $\boldsymbol{w}_{i,j}(l-1,l) = 0$ if $\boldsymbol{w}_{i,j}(l-1,l) < (T_{l-1,l} \times \max(\boldsymbol{w}(l-1,l))$.

It was observed in (Han *et al.*, 2015b) that magnitude-based weight pruning methods achieve between 9-13 times compression on DNNs such as AlexNet and VGG-16. I now propose re-training of pruned networks and integrate the above two approaches.

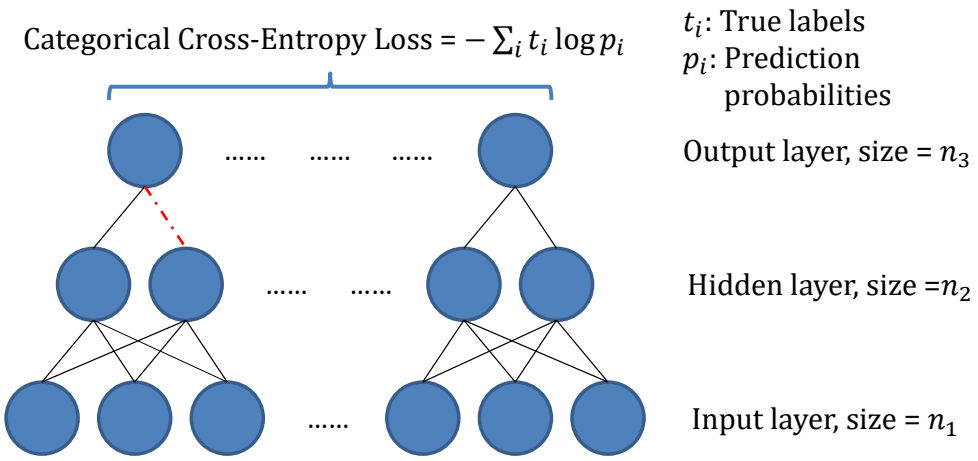


Figure 7.2: A Three-layer MLP for a 10-class Classification Task. ©2017 IEEE.

Re-training of Pruned Networks

Pruning a weight matrix element actually removes a connection in a DNN. For example, in Figure 7.2, pruning the element - $w_{2,1}(2,3)$ - will remove the connection denoted by red dotted line. It is necessary to keep track of such removed connections as not only those weights do not get updated in the forward propagation but no gradients can flow backwards through such connections. This will affect neurons in earlier layers as the gradients from the pruned neuron will not be added anymore into its update equation. As mentioned before, I re-train the network on a different data after *pruning is complete*. Since the data distribution is not identical (but similar), I need to come up with new pruning strategies if a high performance on both - old and new data has to be maintained.

A naïve way is to lower the pruning threshold T that will reduce the number of weights getting pruned. In other words, DNN has more feature dimensions available when it retrains on the new data, resulting in a richer representation and in turn increased accuracy on the new data. I present results for various values of T in Section 7.1.4. Based on this intuition, I propose following three re-training strategies.

Global re-training with global pruning: I train the DNN to convergence and apply

global pruning with a user-defined threshold T. Then I re-train the pruned network till convergence. During re-training, no additional modification/pruning to weights is performed i.e. the indices of the pruned weight elements stay constant during re-training. Thus this method is the simplest to implement.

Global re-training with layer-wise pruning: The only difference between this and the above technique is that I apply layer-wise pruning and then I re-train the pruned network till convergence.

Iterative re-training with layer-wise pruning: This technique performs the following steps in cyclic order.

1. DNN is trained on the original data for an epoch.
2. Then at the end of the epoch, layer-wise pruning is performed with threshold $T_{l-1,l} \times \max(\boldsymbol{w}(l-1,l)$.
3. The pruned indices are now collected and the weight values as well as the backpropagated gradients from the elements at that indices are cut off. While training the next epoch, these indices are used and continuously accumulated throughout the training.

Since the pruned indices are accumulated, I get a monotonic decrease in the number of DNN weights. After the DNN has finished the desired number of epochs or reached convergence, I re-train the DNN with new data by keeping in mind the pruned indices.

All the above re-training strategies only use variations of the weight-magnitude-based pruning methods and do not exploit the fact that the new data distribution is similar to that of the original training data. Below I present a technique that utilizes the back-propagated gradients of weight matrices.

Gradient-based pruning for re-training of DNNs: Due to the similarity in the distribution of old and new training data, I hypothesize that the good features for the old, original

training data work well even for the new data. Consider the face recognition example to elaborate this. For example, a person's eyes is his/her one of the most distinctive features. Changing the eyes on someone's face will cause a large amount of change in their facial features (that is why people put a black mask on eyes when they want to hide identity). If a feature is distinctive, a small change in its weight coefficient could introduce a large change in the loss. If such weight coefficients are identified, then they could be preserved under the hypothesis that the features they are acting on will still be relevant for the new data.

Identifying such weights from just magnitude turns out to be difficult. However, the back-propagated gradients over the entire data - $\frac{\partial L}{\partial w_{i,j}}$ - give us an estimate of how much the loss changes by introducing an infinitesimal change in the weight element (i, j). I compute the entire matrix - $\frac{\partial L}{\partial \boldsymbol{w}}$ - that quantifies the "importance" of the all the weight elements. I sort the elements of $\frac{\partial L}{\partial \boldsymbol{w}}$ in ascending order and then record the indices of the top T' elements in the sorted array. Ascending sort implies the index of weight causing the least (or most negative) change in loss will be one. Thus top T' indices give us most important weights. Here, T' is a user-defined parameter.

I outline the steps for the gradient-based pruning re-training below.

1. DNN is trained on the original data for an epoch.

2. With threshold $T_{l-1,l} \times \max(\boldsymbol{w}(l-1, l)$, only indices produced from layer-wise pruning are recorded and stored in an index array (i.e. weights are not yet pruned).

3. With gradient-based pruning and threshold of T', I obtain the indices of most important weights. These indices are now removed from the array produced above.

4. Actual layer-wise pruning is now performed with the filtered indices. Once the indices are obtained, this step is same as the iterative training explained before.

Table 7.1: Results of Pruning and Re-training Experiments. Bold Typeface Indicates Best Results among Pruned Networks ©2017 IEEE.

Global pruning						
T	T'	Post-prune pre-retrain Clean test	Post-prune Pre-retrain noisy test	Post-prune Post-retrain Clean test	Post-prune Post-retrain Noisy test	compression factor
0.08	-	0.9423	0.5564	0.8798	0.9235	$7.3\times$
0.12	-	0.8417	0.4479	0.8489	0.9022	$19.1\times$
Layer-wise pruning						
0.08	-	**0.9576**	0.5888	0.875	**0.9338**	$1.6\times$
0.12	-	0.9542	0.5651	0.8765	0.9317	$2.1\times$
0.15	-	0.9505	0.5347	0.8769	0.9301	$2.9\times$
0.3	-	0.8499	0.4039	0.8352	0.902	$17.2\times$
Iterative pruning						
0.3	-	0.938	0.5736	0.8492	0.9031	$\mathbf{21.8\times}$
Iterative gradient pruning						
0.3	0.1	0.9445	0.5736	0.8646	0.9056	$7.1\times$
0.3	0.2	0.9451	0.5618	0.8765	0.9134	$4.3\times$
0.3	0.3	0.9461	**0.6147**	**0.8831**	0.9147	$3\times$
Reference - No pruning						
T	T'	Clean test	Noisy test	Clean test	Noisy test	factor
-	-	0.9580	0.5957	0.862	0.9376	0

7.1.4 Results

I test the proposed pruning and re-training strategies on the popular MNIST data. In order to generate new data, I apply Gaussian blur to the original data with kernel size $= 9$ and sigma$= 3$. I build a simple three layer MLP following the standard LeNet. The first layer contains 784 neurons, the hidden and the output layer contains 50 and 10 neurons respectively. The network is trained till validation loss shows no improvement for 10 consecutive epochs. I use stochastic gradient descent (SGD) with 0.9 nesterov momentum and weight decay to train the network. All implementations were done in Theano. Even though I use a three-layer MLP, extending these results to a deep network is trivial in theory, since the pruning is always done one layer at a time.

The results are shown in Table 7.1. Gradient-based pruning strategy gives best results on a clean test set post-pruning. Layer-wise pruning gives best results for a noisy set post pruning and post re-training. With respect to compression, iterative pruning gives best results by reducing the parameters from $39,760$ to just $1,833$. Global pruning gives good compression but its results vary drastically depending on the threshold T unlike layer-wise pruning. The heavy dependence of global weight pruning on T is highly undesirable. Interestingly, all but two methods beat the uncompressed network on clean test data post-pruning and post training. The reference network performs best on the noisy test post training on noisy data. This is due to the catastrophical forgetting phenomenon observed in neural networks. Catastrophical forgetting is widely studied but only on the un-pruned networks. It would be an interesting future direction to integrate catastrophical forgetting techniques along with the pruning techniques to enable re-training on smart devices.

7.1.5 Discussion

In this chapter, I proposed and evaluated various strategies for re-training a pruned network. While this is extremely useful in today's edge computing paradigm, I found that the best algorithm is usually a trade-off between the compression factor and the accuracy. I observed that though naïve re-training gives the best performance on the new data (i.e. noisy), it suffers on the original data due to catastrophic forgetting. Most re-training approaches significantly boost the performance on the noisy data while minimizing the performance reduction on the old data. Further studies are needed to investigate catastrophic forgetting in pruned networks and to maximize the performance of DNNs on both original as well as new data.

7.2 Conclusion

I described three hierarchies of feature representations, namely, hand-crafted, latent and those obtained from deep neural networks. Every type of representation has its own pros and cons. Hand-crafted features are easy to interpret and they have explainable behavior, a characteristic that is crucial in high-risk tasks such as automated medical diagnosis or financial transactions. Hand-crafted also allow easy-inclusion of the prior knowledge, potentially allowing machines to handle real-world scenarios where ambiguity is present or a scenario that is absent from the training samples. In such cases, prior knowledge can help choose an appropriate action. Latent-features try to discover the underlying structure in the feature-space such as sparsity, decorrelation of reduced dimensions, low-rank, etc. The discovered structure helps in reducing feature redundancy, and the transformed feature space usually highlights explanatory variables that helps increase performance. Finally, I introduce deep features. Since the advent of AlexNet (Krizhevsky *et al.*, 2012) in 2012, deep features have proven most effective for many visual computing tasks. A desirable property of deep

networks is that they directly operate on the raw data with minimal pre-processinhg and discover task-specific data representations. Deep networks do so by iteratively minimizing a task-based loss and updating its parameters/weights in that process. To train deep networks from scratch, one needs large amounts of data as well as massive computing power. However, the learned deep features can be easily transferred to other tasks by means of fine-tuning. Fine-tuning requires small amount of data and training time (in comparison to the amount of data/time needed to train networks from scratch) that helps with small datasets.

I presented five visual computing tasks employing various representation hierarchies. The task of clinically-relevant retrieval of diabetic retinopathy fundus image data that has unique color spectrum and other unique attributes that decide the category of that image. Fundus data from hospitals is difficult to obtain. Thus DR image datasets at that time contained only thousands of images. The retrieval was unsupervised. As a result, hand-crafted feature representations were most suited for this task.

The next task of image enhancement has two parts: 1. structured prediction of image enhancement parameters, and 2. content-adaptive, unified image enhancement using GPs employed latent feature space. For the first part, I hypothesize that there exists a traversal pattern in some feature space that would lead to structured exploration and prediction of enhancement parameters. I construct the objective function such that predicted parameters will be an inner-produce of three latent factors. The three latent factors are derived using iterative optimization techniques. The second part unifies the image enhancement pipeline by employing GPs. In the GP-kernel-induced feature space, the parameters are ranked. Each predicted parameter also has a mean and a variance. This automatically paves way for structured exploration of the parameter space and allows for generation of small number of candidate enhancements. This makes the enhancement process efficient.

The next two tasks utilize deep features. First task is defined as ranking a pair of images with respect to their aestheticism. In the literature, aesthetic estimation has often been

approached as a classification task. However, I argue that in many applications such as image search, image enhancement, etc., ranking will be more intuitive and will produce better results than binary-categorization of the images. I propose a Siamese network that takes a pair of images with a label indicating the ranking order, and learns the parameters with a ranking-loss. The ranking-loss is non-negative if the network produces an incorrect ranking order. I also show that the proposed network can perform binary classification with almost no re-training and with minor structural modifications. The network is able to produce a state-of-art performance but still there are challenges on the semantic level. For example, the proposed network often labels an image as non-aesthetic if it captures some rare phenomenon but does not follow photographic rules (such as high contrast, rule-of-thirds, etc.). Humans will often overlook the photography style in such cases and consider the difficulty in capturing that brilliant phenomenon. The proposed network lacks in understanding such semantics.

Second task utilizing deep features involves capturing localized image artifacts. Training CNNs to capture localized artifacts and label the image accordingly on relatively small datasets is challenging. On the other hand, the nature of some of these datasets may exhibit properties that can be leveraged to increase the localization power as well as the volume of useful training data. For example, the images can be divided into smaller patches, and if the labels of these patches could be derived from the original image, then the training data could be augmented by upto a couple of orders of magnitude. However, to the best of my knowledge, there did not exist any approach that can collectively consider all patches to predict a label for the given image. To combat this problem, I propose a two-stage deep network architecture that utilizes *hyper-image* representation. I evaluate my approach on a syntetic and two real-world vision problems: 1. no-reference image quality estimation and 2. image forgery classification. I show that the two-stage CNN is able to beat the current state-of-art by pooling in statistics from all the patches and integrating them to arrive at a

prediction.

In this chapter, I describe some future directions to my work. Deep networks may be adapted into many fields, increasing the need of massive computational resources by a few orders of magnitude. Moreover, the world is becoming highly interconnected and there is a push for moving the computation away from the center of the cloud (i.e. servers) to the edge of the cloud (i.e. individual devices). This will reduce the computational load on servers as well as allow the devices to learn on-the-fly. The devices will also have the opportunity to observe human lives more closely, allowing them to learn finer nuances of personal life. In turn, this will result smarter devices that can predict many aspects of our lives with higher precision. The first obstacle in pushing the computation to the edge is the size of deep networks and their need for significant computational resoruces - even during inference. Reducing the size of deep networks without sacrificing performance will alleviate both these problems as reduction in size translates to reduced number of parameters. If the parameters are removed appropriately, then it can also speed up execution on deep networks. This process is called deep network pruning. However, in the literature, deep networks cannot be trained once the pruning is done. I propose a problem where a network should retain its ability of learning to perform different tasks with a pruned architecture. I conduct a pilot study where I prune the architecture and re-train it with a perturbed version of the same dataset used in the initial training. Results show that the pruned network behaves differently with different types of pruning techniques. The results suggest that the pruned networks may be forgetting the earlier task. This phenomenon is widely studied and is known as catestrophical forgetting. However, integrating it with pruning algorithms has never been done before. I believe that is an interesting future direction to take.

REFERENCES

Abràmoff, M. D., M. Niemeijer, M. S. Suttorp-Schulten, M. A. Viergever, S. R. Russell and B. van Ginneken, "Evaluation of a system for automatic detection of diabetic retinopathy from color fundus photographs in a large population of patients with diabetes", Diabetes care **31**, 2, 193–198 (2008).

Alvarez, M. A. and N. D. Lawrence, "Computationally efficient convolved multiple output gaussian processes", The Journal of Machine Learning Research **12**, 1459–1500 (2011).

Alvarez, M. A., D. Luengo, M. K. Titsias and N. D. Lawrence, "Efficient multioutput gaussian processes through variational inducing kernels", in "International Conference on Artificial Intelligence and Statistics", pp. 25–32 (2010).

Ambikasaran, S., D. Foreman-Mackey, L. Greengard, D. W. Hogg and M. O'Neil, "Fast direct methods for gaussian processes and the analysis of nasa kepler mission data", arXiv preprint arXiv:1403.6015 (2014).

Andrews, S., I. Tsochantaridis and T. Hofmann, "Support vector machines for multiple-instance learning", in "Advances in neural information processing systems", pp. 561–568 (2002).

Baltrunas, L., B. Ludwig and F. Ricci, "Matrix factorization techniques for context aware recommendation", in "Proceedings of the fifth ACM conference on Recommender systems", pp. 301–304 (ACM, 2011).

Bayram, S., H. T. Sencar and N. Memon, "A survey of copy-move forgery detection techniques", in "IEEE Western New York Image Processing Workshop", pp. 538–542 (Citeseer, 2008).

Bayram, S., H. T. Sencar and N. Memon, "An efficient and robust method for detecting copy-move forgery", in "2009 IEEE International Conference on Acoustics, Speech and Signal Processing", pp. 1053–1056 (IEEE, 2009).

Bengio, Y., A. Courville and P. Vincent, "Representation learning: A review and new perspectives", IEEE transactions on pattern analysis and machine intelligence **35**, 8, 1798–1828 (2013).

Bennett, J., S. Lanning *et al.*, "The netflix prize", in "Proceedings of KDD cup and workshop", vol. 2007, p. 35 (2007).

Berthouzoz, F., W. Li, M. Dontcheva and M. Agrawala, "A framework for content-adaptive photo manipulation macros: Application to face, landscape, and global manipulations.", ACM Trans. Graph. **30**, 5, 120 (2011).

Bhattacharjee, B., M. L. Hill, H. Wu, P. S. Chandakkar, J. R. Smith and M. N. Wegman, "Distributed learning of deep feature embeddings for visual recognition tasks", IBM Journal of Research and Development **61**, 4, 4:1–4:8 (2017).

Bhattacharya, S., R. Sukthankar and M. Shah, "A framework for photo-quality assessment and enhancement based on visual aesthetics", in "The 18th ACM international conference on Multimedia", pp. 271–280 (2010).

Bo, L. and C. Sminchisescu, "Twin gaussian processes for structured prediction", International Journal of Computer Vision **87**, 1-2, 28–52 (2010).

Bonilla, E. V., K. M. Chai and C. Williams, "Multi-task gaussian process prediction", in "Advances in neural information processing systems", pp. 153–160 (2007).

Bordes, A., S. Ertekin, J. Weston and L. Bottou, "Fast kernel classifiers with online and active learning", The Journal of Machine Learning Research **6**, 1579–1619 (2005).

Bromley, J., J. W. Bentz, L. Bottou, I. Guyon, Y. LeCun, C. Moore, E. Säckinger and R. Shah, "Signature verification using a siamese time delay neural network", International Journal of Pattern Recognition and Artificial Intelligence **7**, 04, 669–688 (1993).

Bychkovsky, V., S. Paris, E. Chan and F. Durand, "Learning photographic global tonal adjustment with a database of input/output image pairs", in "Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on", pp. 97–104 (IEEE, 2011).

Cai, W., D. Feng and R. Fulton, "Content-based retrieval of dynamic PET functional images", Information Technology in Biomedicine, IEEE Transactions on **4**, 2, 152–158 (2000).

Centers for Disease Control and prevention and others, "National diabetes fact sheet: national estimates and general information on diabetes and prediabetes in the united states, 2011", Atlanta, GA: US Department of Health and Human Services, Centers for Disease Control and Prevention **201** (2011).

Chandakkar, P. S., *Clinically Relevant Classification and Retrieval of Diabetic Retinopathy Images*, Ph.D. thesis, URL `http://login.ezproxy1.lib.asu.edu/login?url=https://search.proquest.com/docview/1035337244?accountid=4485`, copyright - Database copyright ProQuest LLC; ProQuest does not claim copyright in the individual underlying works; Last updated - 2016-03-11 (2012).

Chandakkar, P. S., V. Gattupalli and B. Li, "A computational approach to relative aesthetics", in "23rd International Conference on Pattern Recognition (ICPR)", pp. 2446–2451 (2016).

Chandakkar, P. S. and B. Li, "Investigating human factors in image forgery detection", in "Proceedings of the 1st ACM International Workshop on Human Centered Event Understanding from Multimedia", HuEvent '14, pp. 41–44 (ACM, New York, NY, USA, 2014), URL `http://doi.acm.org/10.1145/2660505.2660510`.

Chandakkar, P. S. and B. Li, "A structured approach to predicting image enhancement parameters", in "Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on", pp. 1–9 (IEEE, 2016).

Chandakkar, P. S. and B. Li, "Capturing localized image artifacts through a cnn-based hyper-image representation", arXiv preprint arXiv:1711.04945 (2017a).

Chandakkar, P. S. and B. Li, "Joint regression and ranking for image enhancement", in "Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on", pp. 235–243 (IEEE, 2017b).

Chandakkar, P. S., Y. Li, P. L. K. Ding and B. Li, "Strategies for re-training a pruned neural network in an edge computing paradigm", in "2017 IEEE International Conference on Edge Computing (EDGE)", pp. 244–247 (2017a).

Chandakkar, P. S., Q. Tian and B. Li, "Relative learning from web images for content-adaptive enhancement", in "Multimedia and Expo (ICME), 2015 IEEE International Conference on", pp. 1–6 (IEEE, 2015a).

Chandakkar, P. S., Q. Tian and B. Li, "Relative learning from web images for content-adaptive enhancement", in "Multimedia and Expo (ICME), 2015 IEEE International Conference on", pp. 1–6 (IEEE, 2015b).

Chandakkar, P. S., R. Venkatesan and B. Li, *Video-Based Self-positioning for Intelligent Transportation Systems Applications*, pp. 718–729 (Springer International Publishing, Cham, 2014), URL https://doi.org/10.1007/978-3-319-14249-4_69.

Chandakkar, P. S., R. Venkatesan and B. Li, "Mirank-knn: multiple-instance retrieval of clinically relevant diabetic retinopathy images", Journal of Medical Imaging **4**, 3, 034003 (2017b).

Chandakkar, P. S., R. Venkatesan, B. Li and H. Li, "Retrieving clinically relevant diabetic retinopathy images using a multi-class multiple-instance framework", in "SPIE Medical Imaging", pp. 86700Q–86700Q (International Society for Optics and Photonics, 2013).

Chandakkar, P. S., R. Venkatesan, B. Li and H. K. Li, "A machine-learning approach to retrieving diabetic retinopathy images", in "Proceedings of the ACM Conference on Bioinformatics, Computational Biology and Biomedicine", BCB '12, pp. 588–589 (ACM, New York, NY, USA, 2012), URL http://doi.acm.org/10.1145/2382936.2383030.

Chandakkar, P. S., Y. Wang and B. Li, "Improving vision-based self-positioning in intelligent transportation systems via integrated lane and vehicle detection", in "2015 IEEE Winter Conference on Applications of Computer Vision", pp. 404–411 (2015c).

Chaum, E., T. P. Karnowski, V. P. Govindasamy, M. Abdelrahman and K. W. Tobin, "Automated diagnosis of retinopathy by content-based image retrieval", Retina **28**, 10, 1463–1477 (2008).

Chen, C.-Y. and K. Grauman, "Inferring unseen views of people", in "Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on", pp. 2011–2018 (IEEE, 2014).

Chen, F., H. Yu, R. Hu and X. Zeng, "Deep learning shape priors for object segmentation", in "IEEE CVPR", (2013).

Chen, L., Q. Zhang and B. Li, "Predicting multiple attributes via relative multi-task learning", in "Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on", pp. 1027–1034 (IEEE, 2014).

Chen, Q., X. Tai, Y. Dong, S. Pan, X. Wang and C. Yin, "Medical image retrieval based on semantic of neighborhood color moment histogram", in "Bioinformatics and Biomedical Engineering, 2008. ICBBE 2008. The 2nd International Conference on", pp. 2221–2224 (IEEE, 2008).

Chollet, F. *et al.*, "Keras", `https://github.com/fchollet/keras` (2015).

Chu, W. W., I. T. Ieong and R. K. Taira, "A semantic modeling approach for image retrieval by content", The International Journal on Very Large Data Bases **3**, 4, 445–477 (1994).

Ciresan, D., U. Meier and J. Schmidhuber, "Multi-column deep neural networks for image classification", in "IEEE CVPR", pp. 3642–3649 (2012).

Courbariaux, M., I. Hubara, D. Soudry, R. El-Yaniv and Y. Bengio, "Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1", arXiv preprint arXiv:1602.02830 (2016).

Dai, J., K. He, and J. Sun, "Instance-aware semantic segmentation via multi-task network cascades", (2016).

Dalal, N. and B. Triggs, "Histograms of oriented gradients for human detection", in "IEEE CVPR", vol. 1, pp. 886–893 (IEEE, 2005).

Datta, R., D. Joshi, J. Li and J. Z. Wang, "Studying aesthetics in photographic images using a computational approach", in "ECCV", pp. 288–301 (Springer, 2006).

Deepak, K. S., G. D. Joshi and J. Sivaswamy, "Content-based retrieval of retinal images for maculopathy", in "Proceedings of the 1st ACM International Health Informatics Symposium", pp. 135–143 (2010).

Denil, M., B. Shakibi, L. Dinh, N. de Freitas *et al.*, "Predicting parameters in deep learning", in "NIPS", (2013).

Denton, E. L., W. Zaremba, J. Bruna, Y. LeCun and R. Fergus, "Exploiting linear structure within convolutional networks for efficient evaluation", in "NIPS", (2014).

Deselaers, T., D. Keysers and H. Ney, "Features for image retrieval: an experimental comparison", Information Retrieval **11**, 2, 77–107 (2008).

Dhar, S., V. Ordonez and T. L. Berg, "High level describable attributes for predicting aesthetics and interestingness", in "IEEE CVPR", pp. 1657–1664 (2011).

Dietterich, T. G., R. H. Lathrop and T. Lozano-Pérez, "Solving the multiple instance problem with axis-parallel rectangles", Artificial intelligence **89**, 1, 31–71 (1997).

Dodge, S. and L. Karam, "Understanding how image quality affects deep neural networks", arXiv preprint arXiv:1604.04004 (2016).

Donahue, J., Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition", arXiv preprint arXiv:1310.1531 (2013).

Eleftheriadis, S., O. Rudovic and M. Pantic, "Discriminative shared gaussian processes for multiview and view-invariant facial expression recognition", Image Processing, IEEE Transactions on **24**, 1, 189–204 (2015).

Fan, W., K. Wang, F. Cayre and Z. Xiong, "3d lighting-based image forgery detection using shape-from-shading", in "Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European", pp. 1777–1781 (IEEE, 2012).

Faria, F. F., A. Veloso, H. M. Almeida, E. Valle, R. d. S. Torres, M. A. Gonçalves and W. Meira Jr, "Learning to rank for content-based image retrieval", in "Proceedings of the International Conference on Multimedia information retrieval", pp. 285–294 (ACM, 2010).

Farid, H., "Exposing digital forgeries from jpeg ghosts", IEEE Transactions on information forensics and security **4**, 1, 154–160 (2009a).

Farid, H., "Image forgery detection–a survey", (2009b).

Freeman, W. T. and E. H. Adelson, "The design and use of steerable filters", IEEE TPAMI **13**, 9, 891–906 (1991).

Garg, S. and R. M. Davis, "Diabetic retinopathy screening update", Clinical diabetes **27**, 4, 140–145 (2009).

Gidaris, S. and N. Komodakis, "Object detection via a multi-region and semantic segmentation-aware cnn model", in "Proceedings of the IEEE International Conference on Computer Vision", pp. 1134–1142 (2015).

Goldbaum, M. H., N. P. Katz, S. Chaudhuri and M. Nelson, "Image understanding for automated retinal diagnosis.", in "The Annual Symposium on Computer Application [sic] in Medical Care. Symposium on Computer Applications in Medical Care", pp. 756–760 (American Medical Informatics Association, 1989).

Goodfellow, I., J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio, "Generative adversarial nets", in "Advances in Neural Information Processing Systems", pp. 2672–2680 (2014).

Guariguata, L., D. Whiting, I. Hambleton, J. Beagley, U. Linnenkamp and J. Shaw, "Global estimates of diabetes prevalence for 2013 and projections for 2035", Diabetes research and clinical practice **103**, 2, 137–149 (2014).

Gupta, A., S. Moezzi, A. Taylor, S. Chatterjee, R. Jain, I. Goldbaum and S. Burgess, "Content-based retrieval of ophthalmological images", in "Image Processing, 1996. Proceedings., International Conference on", vol. 3, pp. 703–706 (IEEE, 1996).

Han, S., J. Kang, H. Mao, Y. Hu, X. Li, Y. Li, D. Xie, H. Luo, S. Yao, Y. Wang *et al.*, "ESE: Efficient speech recognition engine with sparse lstm on fpga", in "ISFPGA", (2017).

Han, S., X. Liu, H. Mao, J. Pu, A. Pedram, M. A. Horowitz and W. J. Dally, "EIE: efficient inference engine on compressed deep neural network", in "ISCA", (2016).

Han, S., H. Mao and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding", arXiv preprint arXiv:1510.00149 (2015a).

Han, S., J. Pool, J. Tran and W. Dally, "Learning both weights and connections for efficient neural network", in "NIPS", (2015b).

Hariharan, B., P. Arbeláez, R. Girshick and J. Malik, "Hypercolumns for object segmentation and fine-grained localization", in "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition", pp. 447–456 (2015).

He, J., Z. Lin, L. Wang and X. Tang, "Detecting doctored jpeg images via dct coefficient analysis", in "European conference on computer vision", pp. 423–435 (Springer, 2006).

He, K., X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition", arXiv preprint arXiv:1512.03385 (2015).

Heikinheimo, H., E. Hinkkanen, H. Mannila, T. Mielikäinen and J. K. Seppänen, "Finding low-entropy sets and trees from binary data", in "Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining", pp. 350–359 (ACM, 2007).

Hensman, J., N. Fusi and N. D. Lawrence, "Gaussian processes for big data", arXiv preprint arXiv:1309.6835 (2013).

Huang, J., S. R. Kumar, M. Mitra, W.-J. Zhu and R. Zabih, "Image indexing using color correlograms", in "IEEE CVPR", (1997).

Hwang, S. J., A. Kapoor and S. B. Kang, "Context-based automatic local image enhancement", in "Computer Vision–ECCV 2012", pp. 569–582 (Springer, 2012).

Iandola, F. N., S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size", arXiv preprint arXiv:1602.07360 (2016).

Joachims, T., "Optimizing search engines using clickthrough data", in "Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining", pp. 133–142 (ACM, 2002).

Joshi, D., R. Datta, E. Fedorovskaya, Q.-T. Luong, J. Z. Wang, J. Li and J. Luo, "Aesthetics and emotions in images", IEEE Signal Processing Magazine **28**, 5, 94–115 (2011).

Joshi, N., W. Matusik, E. H. Adelson and D. J. Kriegman, "Personal photo enhancement using example images", ACM Trans. Graph **29**, 2, 12 (2010).

Kang, L., P. Ye, Y. Li and D. Doermann, "Convolutional neural networks for no-reference image quality assessment", in "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition", pp. 1733–1740 (2014).

Kang, S. B., A. Kapoor and D. Lischinski, "Personalization of image enhancement", in "Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on", pp. 1799–1806 (IEEE, 2010).

Kapoor, A., J. C. Caicedo, D. Lischinski and S. B. Kang, "Collaborative personalization of image enhancement", International Journal of Computer Vision **108**, 1-2, 148–164 (2014).

Karayev, S., M. Trentacoste, H. Han, A. Agarwala, T. Darrell, A. Hertzmann and H. Winnemoeller, "Recognizing image style", in "Proceedings of the British Machine Vision Conference.", (2014).

Kaufman, L., D. Lischinski and M. Werman, "Content-aware automatic photo enhancement", in "Computer Graphics Forum", vol. 31, pp. 2528–2540 (Wiley Online Library, 2012).

Kaufman, L. and W. Richards, "Spontaneous fixation tendencies for visual forms", Perception & Psychophysics **5**, 2, 85–88 (1969).

Kauppi, T., V. Kalesnykiene, J.-K. Kamarainen, L. Lensu, I. Sorri, A. Raninen, R. Voutilainen, H. Uusitalo, H. Kälviäinen and J. Pietilä, "The diaretdb1 diabetic retinopathy database and evaluation protocol.", in "BMVC", pp. 1–10 (2007).

Kauppi, T., V. Kalesnykiene, J.-K. Kamarainen, L. Lensu, I. Sorri, H. Uusitalo, H. Kälviäinen and J. Pietilä, "Diaretdb0: Evaluation database and methodology for diabetic retinopathy algorithms", Machine Vision and Pattern Recognition Research Group, Lappeenranta University of Technology, Finland (2006).

Ke, Y., X. Tang and F. Jing, "The design of high-level features for photo quality assessment", in "IEEE CVPR", vol. 1, pp. 419–426 (2006).

Kee, E. and H. Farid, "Exposing digital forgeries from 3-d lighting environments", in "2010 IEEE International Workshop on Information Forensics and Security", pp. 1–6 (IEEE, 2010).

Kelly, P. M., T. M. Cannon and D. R. Hush, "Query by image example: the CANDID approach", SPIE Storage and Retrieval for Image and Video Databases III **2420**, 238–248 (1995).

Korn, P., N. Sidiropoulos, C. Faloutsos, E. Siegel and Z. Protopapas, "Fast and effective retrieval of medical tumor shapes", Knowledge and Data Engineering, IEEE Transactions on **10**, 6 (1998).

Kovashka, A., D. Parikh and K. Grauman, "Whittlesearch: Image search with relative attribute feedback", in "IEEE CVPR", pp. 2973–2980 (2012).

Kovesi, P. D., "MATLAB and Octave functions for computer vision and image processing", Centre for Exploration Targeting, School of Earth and Environment, The University of Western Australia, available from: <http://www.csse.uwa.edu.au/∼pk/research/matlabfns/> (2000).

Krizhevsky, A., I. Sutskever and G. E. Hinton, "Imagenet classification with deep convolutional neural networks", in "Advances in neural information processing systems", pp. 1097–1105 (2012).

Lamard, M., G. Cazuguel, G. Quellec, L. Bekri, C. Roux and B. Cochener, "Content based image retrieval based on wavelet transform coefficients distribution", in "Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE", pp. 4532–4535 (IEEE, 2007).

Lawrence, N. D. and R. Urtasun, "Non-linear matrix factorization with gaussian processes", in "Proceedings of the 26th Annual International Conference on Machine Learning", pp. 601–608 (ACM, 2009).

Li, B. and H. K. Li, "Automated analysis of diabetic retinopathy images: Principles, recent developments, and emerging trends", Current diabetes reports pp. 1–7 (2013).

Li, M., "Texture moment for content-based image retrieval", in "Multimedia and Expo, 2007 IEEE International Conference on", pp. 508–511 (IEEE, 2007).

Li, S., S. Shan and X. Chen, "Relative forest for attribute prediction", in "Computer Vision–ACCV 2012", pp. 316–327 (Springer, 2013).

Litzel, O., *On Photographic Composition* (Amphoto, 1975).

Liu, T.-Y., J. Xu, T. Qin, W. Xiong and H. Li, "Letor: Benchmark dataset for research on learning to rank for information retrieval", in "Proceedings of SIGIR 2007 workshop on learning to rank for information retrieval", pp. 3–10 (2007).

Liu, X., C.-T. Lu and F. Chen, "An entropy-based method for assessing the number of spatial outliers", in "IEEE International Conference on Information Reuse and Integration, 2008.", pp. 244–249 (IEEE, 2008).

Locher, P. J. and C. Nodine, "Symmetry catches the eye", Eye movements: From physiology to cognition pp. 353–361 (1987).

Lowe, D. G., "Distinctive image features from scale-invariant keypoints", International journal of computer vision **60**, 2, 91–110 (2004a).

Lowe, D. G., "Distinctive image features from scale-invariant keypoints", International journal of computer vision **60**, 2, 91–110 (2004b).

Loy, G. and A. Zelinsky, "Fast radial symmetry for detecting points of interest", IEEE TPAMI **25**, 8, 959–973 (2003).

Lu, X., Z. Lin, H. Jin, J. Yang and J. Z. Wang, "Rapid: Rating pictorial aesthetics using deep learning", in "The ACM International Conference on Multimedia", pp. 457–466 (2014).

Lu, X., Z. Lin, X. Shen, R. Mech and J. Z. Wang, "Deep multi-patch aggregation network for image style, aesthetics, and quality estimation", in "IEEE ICCV", pp. 990–998 (2015).

Luo, W., X. Wang and X. Tang, "Content-based photo quality assessment", in "IEEE ICCV", pp. 2206–2213 (2011).

Luo, Y. and X. Tang, "Photo and video quality evaluation: Focusing on the subject", in "ECCV", pp. 386–399 (Springer, 2008).

Ma, H., H. Yang, M. R. Lyu and I. King, "Sorec: social recommendation using probabilistic matrix factorization", in "Proceedings of the 17th ACM conference on Information and knowledge management", pp. 931–940 (ACM, 2008).

Ma, H., D. Zhou, C. Liu, M. R. Lyu and I. King, "Recommender systems with social regularization", in "Proceedings of the fourth ACM international conference on Web search and data mining", pp. 287–296 (ACM, 2011).

Manjunath, B. S. and W.-Y. Ma, "Texture features for browsing and retrieval of image data", IEEE TPAMI **18**, 8, 837–842 (1996).

Manning, C. D., P. Raghavan and H. Schütze, *Introduction to information retrieval*, vol. 1 (Cambridge university press Cambridge, 2008).

Marchesotti, L., F. Perronnin, D. Larlus and G. Csurka, "Assessing the aesthetic quality of photographs using generic image descriptors", in "IEEE ICCV", pp. 1784–1791 (2011).

Marlin, B., R. S. Zemel, S. Roweis and M. Slaney, "Collaborative filtering and the missing at random assumption", arXiv preprint arXiv:1206.5267 (2012).

Maron, O. and T. Lozano-Pérez, "A framework for multiple-instance learning", in "Proceedings of NIPS", pp. 570–576 (MIT Press, 1998).

McCormick, B. and M. Goldbaum, "STARE= structured analysis of the retina: Image processing of tv fundus image", in "Jet Propulsion Laboratory, Pasadena, CA: USA-Japan Workshop on Image Processing", (1975).

Mittal, A., A. K. Moorthy and A. C. Bovik, "No-reference image quality assessment in the spatial domain", Image Processing, IEEE Transactions on **21**, 12, 4695–4708 (2012).

Mittal, A., R. Soundararajan and A. C. Bovik, "Making a completely blind image quality analyzer", IEEE Signal Processing Letters **20**, 3, 209–212 (2013).

Mnih, A. and R. Salakhutdinov, "Probabilistic matrix factorization", in "Advances in neural information processing systems", pp. 1257–1264 (2007).

Moorthy, A. K. and A. C. Bovik, "Blind image quality assessment: From natural scene statistics to perceptual quality", IEEE Transactions on Image Processing **20**, 12, 3350–3364 (2011).

Murphy, K. P., *Machine learning: a probabilistic perspective* (MIT press, 2012).

Murray, N., L. Marchesotti and F. Perronnin, "AVA: A large-scale database for aesthetic visual analysis", in "IEEE CVPR", pp. 2408–2415 (2012).

Nguyen, V., E. Bonilla *et al.*, "Collaborative multi-output gaussian processes", (UAI, 2014).

Nie, F., H. Huang, X. Cai and C. H. Ding, "Efficient and robust feature selection via joint $\ell_{2,1}$-norms minimization", in "Advances in Neural Information Processing Systems 23", edited by J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel and A. Culotta, pp. 1813–1821 (Curran Associates, Inc., 2010).

Niekamp, W., "An exploratory investigation into factors affecting visual balance", ECTJ **29**, 1, 37–48 (1981).

Nishiyama, M., T. Okabe, I. Sato and Y. Sato, "Aesthetic quality classification of photographs based on color harmony", in "CVPR", pp. 33–40 (2011).

O'Donovan, P., A. Agarwala and A. Hertzmann, "Color compatibility from large datasets", in "ACM Transactions on Graphics (TOG)", vol. 30, p. 63 (2011).

Paladugu, A., P. S. Chandakkar, P. Zhang and B. Li, "Supporting navigation of outdoor shopping complexes for visuallyimpaired users through multi-modal data fusion", in "2013 IEEE International Conference on Multimedia and Expo (ICME)", pp. 1–7 (2013).

Parikh, D. and K. Grauman, "Relative attributes", in "Computer Vision (ICCV), 2011 IEEE International Conference on", pp. 503–510 (IEEE, 2011a).

Parikh, D. and K. Grauman, "Relative attributes", in "IEEE ICCV", pp. 503–510 (2011b).

Parikh, D., A. Kovashka, A. Parkash and K. Grauman, "Relative attributes for enhanced human-machine communication.", in "AAAI", (2012).

Park, S. Y. and A. K. Bera, "Maximum entropy autoregressive conditional heteroskedasticity model", Journal of Econometrics **150**, 2, 219–230 (2009).

Petersen, K. B., M. S. Pedersen *et al.*, "The matrix cookbook", Technical University of Denmark **7**, 15 (2008).

Pires, R., H. Jelinek, J. Wainer, S. Goldenstein, E. Valle and A. Rocha, "Assessing the need for referral in automatic diabetic retinopathy detection", Biomedical Engineering, IEEE Transactions on **60**, 12, 3391–3398 (2013).

Ponomarenko, N., L. Jin, O. Ieremeiev, V. Lukin, K. Egiazarian, J. Astola, B. Vozel, K. Chehdi, M. Carli, F. Battisti *et al.*, "Image database tid2013: Peculiarities, results and perspectives", Signal Processing: Image Communication **30**, 57–77 (2015).

Quellec, G., M. Lamard, M. D. Abràmoff, E. Decencière, B. Lay, A. Erginay, B. Cochener and G. Cazuguel, "A multiple-instance learning framework for diabetic retinopathy screening", Medical image analysis **16**, 6, 1228–1240 (2012a).

Quellec, G., M. Lamard, G. Cazuguel, L. Bekri, W. Daccache, C. Roux and B. Cochener, "Automated assessment of diabetic retinopathy severity using content-based image retrieval in multimodal fundus photographs", Investigative Ophthalmology & Visual Science **52**, 11, 8342–8348 (2011).

Quellec, G., M. Lamard, G. Cazuguel, B. Cochener and C. Roux, "Wavelet optimization for content-based image retrieval in medical databases", Medical image analysis **14**, 2, 227–241 (2010).

Quellec, G., M. Lamard, G. Cazuguel, B. Cochener and C. Roux, "Fast wavelet-based image characterization for highly adaptive image retrieval", Image Processing, IEEE Transactions on **21**, 4, 1613–1623 (2012b).

Rahmani, R., S. A. Goldman, H. Zhang, J. Krettek and J. E. Fritts, "Localized content based image retrieval", in "Proceedings of the 7th ACM SIGMM international workshop on Multimedia information retrieval", pp. 227–236 (ACM, 2005).

Rasmussen, C. E., "Gaussian processes for machine learning", (2006).

Recasens, A., A. Khosla, C. Vondrick and A. Torralba, "Where are they looking?", in "Advances in Neural Information Processing Systems", pp. 199–207 (2015).

Reinhard, E., M. Stark, P. Shirley and J. Ferwerda, "Photographic tone reproduction for digital images", in "ACM Transactions on Graphics (TOG)", vol. 21, pp. 267–276 (ACM, 2002).

Rennie, J. D. and N. Srebro, "Fast maximum margin matrix factorization for collaborative prediction", in "Proceedings of the 22nd international conference on Machine learning", pp. 713–719 (ACM, 2005).

Richards, W. and L. Kaufman, "center-of-gravity tendencies for fixations and flow patterns", Perception & Psychophysics **5**, 2, 81–84 (1969).

Rocha, A., T. Carvalho, H. Jelinek, S. Goldenstein and J. Wainer, "Points of interest and visual dictionaries for automatic retinal lesion detection", Biomedical Engineering, IEEE Transactions on **59**, 8, 2244–2253 (2012).

Rosten, E. and T. Drummond, "Fusing points and lines for high performance tracking.", in "IEEE International Conference on Computer Vision", vol. 2, pp. 1508–1511 (2005), URL `http://edwardrosten.com/work/rosten_2005_tracking.pdf`.

Rosten, E. and T. Drummond, "Machine learning for high-speed corner detection", in "ECCV", vol. 1, pp. 430–443 (2006), URL `http://edwardrosten.com/work/rosten_2006_machine.pdf`.

Rudovic, O., M. Pantic and I. Patras, "Coupled gaussian processes for pose-invariant facial expression recognition", Pattern Analysis and Machine Intelligence, IEEE Transactions on **35**, 6, 1357–1369 (2013).

Rudovic, O., I. Patras and M. Pantic, "Regression-based multi-view facial expression recognition", in "Pattern Recognition (ICPR), 2010 20th International Conference on", pp. 4121–4124 (IEEE, 2010).

Saad, M. A., A. C. Bovik and C. Charrier, "Blind image quality assessment: A natural scene statistics approach in the dct domain", Image Processing, IEEE Transactions on **21**, 8, 3339–3352 (2012).

Salakhutdinov, R. and A. Mnih, "Bayesian probabilistic matrix factorization using markov chain monte carlo", in "Proceedings of the 25th international conference on Machine learning", pp. 880–887 (ACM, 2008).

Salimans, T., I. Goodfellow, W. Zaremba, V. Cheung, A. Radford and X. Chen, "Improved techniques for training gans", arXiv preprint arXiv:1606.03498 (2016).

Salomão, S. R., M. R. Mitsuhiro and R. Belfort Jr, "Visual impairment and blindness: an overview of prevalence and causes in brazil", Anais da Academia Brasileira de Ciências **81**, 3, 539–549 (2009).

Sánchez, C. I., M. Niemeijer, M. D. Abràmoff and B. van Ginneken, "Active learning for an efficient training strategy of computer-aided diagnosis systems: application to diabetic retinopathy screening", in "International Conference on Medical Image Computing and Computer-Assisted Intervention", pp. 603–610 (Springer, 2010).

Shahbaz Khan, F., R. M. Anwer, J. van de Weijer, A. D. Bagdanov, M. Vanrell and A. M. Lopez, "Color attributes for object detection", in "IEEE CVPR", pp. 3306–3313 (IEEE, 2012).

Sheikh, H. R. and A. C. Bovik, "Image information and visual quality", Image Processing, IEEE Transactions on **15**, 2, 430–444 (2006).

Sheikh, H. R., Z. Wang, L. Cormack and A. C. Bovik, "Live image quality assessment database release 2", (2005).

Shen, Y., A. Ng and M. Seeger, "Fast gaussian process regression using kd-trees", in "Proceedings of the 19th Annual Conference on Neural Information Processing Systems", No. EPFL-CONF-161316 (2006).

Shi, W., J. Cao, Q. Zhang, Y. Li and L. Xu, "Edge computing: Vision and challenges", IEEE Internet of Things Journal **3**, 5, 637–646 (2016).

Shi, Y., M. Larson and A. Hanjalic, "Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges", ACM Computing Surveys (CSUR) **47**, 1, 3 (2014).

Sigurbjörnsson, B. and R. Van Zwol, "Flickr tag recommendation based on collective knowledge", in "Proceedings of the 17th international conference on World Wide Web", pp. 327–336 (ACM, 2008).

Simonyan, K. and A. Zisserman, "Very deep convolutional networks for large-scale image recognition", arXiv preprint arXiv:1409.1556 (2014).

Song, Q., J. Cheng and H. Lu, "Incremental matrix factorization via feature space re-learning for recommender system", in "Proceedings of the 9th ACM Conference on Recommender Systems", pp. 277–280 (ACM, 2015).

Su, H.-H., T.-W. Chen, C.-C. Kao, W. H. Hsu and S.-Y. Chien, "Scenic photo quality assessment with bag of aesthetics-preserving features", in "The 19th ACM international conference on Multimedia", pp. 1213–1216 (2011).

Sun, Y., X. Wang and X. Tang, "Deep convolutional network cascade for facial point detection", in "IEEE CVPR", (2013).

Sutthiwan, P., Y. Q. Shi, H. Zhao, T.-T. Ng and W. Su, "Markovian rake transform for digital image tampering detection", in "Transactions on data hiding and multimedia security VI", pp. 1–17 (Springer, 2011).

Taigman, Y., M. Yang, M. Ranzato and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification", in "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition", pp. 1701–1708 (2014).

Tang, H., N. Joshi and A. Kapoor, "Learning a blind measure of perceptual image quality", in "Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on", pp. 305–312 (IEEE, 2011).

Theano Development Team, "Theano: A Python framework for fast computation of mathematical expressions", arXiv e-prints **abs/1605.02688**, URL `http://arxiv.org/abs/1605.02688` (2016).

Urtasun, R. and T. Darrell, "Discriminative gaussian process latent variable model for classification", in "Proceedings of the 24th international conference on Machine learning", pp. 927–934 (ACM, 2007).

Van De Sande, K. E., T. Gevers and C. G. Snoek, "Evaluating color descriptors for object and scene recognition", IEEE TPAMI **32**, 9, 1582–1596 (2010).

Van De Weijer, J., T. Gevers and A. D. Bagdanov, "Boosting color saliency in image feature detection", IEEE TPAMI **28**, 1, 150–156 (2006).

Vedaldi, A. and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms", `http://www.vlfeat.org/`, (2008).

Venkatesan, R., P. Chandakkar and B. Li, "Simpler non-parametric methods provide as good or better results to multiple-instance learning", in "The IEEE International Conference on Computer Vision (ICCV)", (2015).

Venkatesan, R., P. Chandakkar, B. Li and H. K. Li, "Classification of diabetic retinopathy images using multi-class multiple-instance learning based on color correlogram features", in "Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE", pp. 1462–1465 (IEEE, 2012).

Wang, J. and J.-D. Zucker, "Solving multiple-instance problem: A lazy learning approach", (2000).

Wang, S., J. Tang, Y. Wang and H. Liu, "Exploring implicit hierarchical structures for recommender systems", in "International Joint Conference on Artificial Intelligence (IJCAI)", (IJCAI, 2015).

Wang, Z., A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity", Image Processing, IEEE Transactions on **13**, 4, 600–612 (2004).

Wang, Z., E. P. Simoncelli and A. C. Bovik, "Multiscale structural similarity for image quality assessment", in "Signals, Systems and Computers, 2004. Conference Record of the Thirty-Seventh Asilomar Conference on", vol. 2, pp. 1398–1402 (Ieee, 2003).

Wright, J., Y. Ma, J. Mairal, G. Sapiro, T. S. Huang and S. Yan, "Sparse representation for computer vision and pattern recognition", Proceedings of the IEEE **98**, 6, 1031–1044 (2010).

Xiong, L., X. Chen, T.-K. Huang, J. G. Schneider and J. G. Carbonell, "Temporal collaborative filtering with bayesian probabilistic tensor factorization.", in "SDM", vol. 10, pp. 211–222 (SIAM, 2010).

Xu, X. and B. Li, "Automatic classification and detection of clinically relevant images for diabetic retinopathy", in "Medical Imaging", pp. 69150Q–69150Q (International Society for Optics and Photonics, 2008).

Yan, J., S. Lin, S. B. Kang and X. Tang, "A learning-to-rank approach for image color enhancement", in "Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on", pp. 2987–2994 (IEEE, 2014a).

Yan, J., S. Lin, S. B. Kang and X. Tang, "A learning-to-rank approach for image color enhancement", in "Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on", pp. 2987–2994 (IEEE, 2014b).

Yan, Z., H. Zhang, B. Wang, S. Paris and Y. Yu, "Automatic photo adjustment using deep neural networks", arXiv preprint arXiv:1412.7725 (2014c).

Yang, C. and T. Lozano-Perez, "Image database retrieval with multiple-instance learning techniques", in "Data Engineering, 2000. Proceedings. 16th International Conference on", pp. 233–243 (IEEE, 2000).

Ye, P., J. Kumar, L. Kang and D. Doermann, "Unsupervised feature learning framework for no-reference image quality assessment", in "Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on", pp. 1098–1105 (IEEE, 2012).

Ye, P., J. Kumar, L. Kang and D. Doermann, "Real-time no-reference image quality assessment based on filter learning", in "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition", pp. 987–994 (2013).

Yosinski, J., J. Clune, A. Nguyen, T. Fuchs and H. Lipson, "Understanding neural networks through deep visualization", arXiv preprint arXiv:1506.06579 (2015).

Yu, K., V. Tresp and A. Schwaighofer, "Learning gaussian processes from multiple tasks", in "Proceedings of the 22nd international conference on Machine learning", pp. 1012–1019 (ACM, 2005).

Zeiler, M. D. and R. Fergus, "Visualizing and understanding convolutional networks", in "Computer vision–ECCV 2014", pp. 818–833 (Springer, 2014).

Zhang, C., X. Chen, M. Chen, S.-C. Chen and M.-L. Shyu, "A multiple instance learning approach for content based image retrieval using one-class support vector machine", in "IEEE International Conference on Multimedia and Expo", pp. 1142–1145 (2005).

Zhang, L., Y. Shen and H. Li, "Vsi: A visual saliency-induced index for perceptual image quality assessment", Image Processing, IEEE Transactions on **23**, 10, 4270–4281 (2014).

Zhang, L., L. Zhang, X. Mou and D. Zhang, "Fsim: A feature similarity index for image quality assessment", IEEE Transactions on Image Processing **20**, 8, 2378–2386 (2011a).

Zhang, L., L. Zhang, X. Mou and D. Zhang, "Fsim: a feature similarity index for image quality assessment", Image Processing, IEEE Transactions on **20**, 8, 2378–2386 (2011b).

Zhang, Q. and S. A. Goldman, "Em-dd: An improved multiple-instance learning technique", in "Advances in neural information processing systems", pp. 1073–1080 (2001).

Zhang, Q., S. A. Goldman, W. Yu and J. E. Fritts, "Content-based image retrieval using multiple-instance learning", in "ICML", vol. 2, pp. 682–689 (Citeseer, 2002).

Zhang, X., J. B. Saaddine, C.-F. Chou, M. F. Cotch, Y. J. Cheng, L. S. Geiss, E. W. Gregg, A. L. Albright, B. E. Klein and R. Klein, "Prevalence of diabetic retinopathy in the united states, 2005-2008", JAMA: the journal of the American Medical Association **304**, 6, 649–656 (2010).

Zheng, L., S. Wang, Z. Liu and Q. Tian, "Packing and padding: Coupled multi-index for accurate image retrieval", in "IEEE CVPR", pp. 1947–1954 (IEEE, 2014).

Zheng, S., S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang and P. H. Torr, "Conditional random fields as recurrent neural networks", in "Proceedings of the IEEE International Conference on Computer Vision", pp. 1529–1537 (2015).

Zhou, B., A. Khosla, A. Lapedriza, A. Oliva and A. Torralba, "Object detectors emerge in deep scene cnns", arXiv preprint arXiv:1412.6856 (2014).

APPENDIX A

PERMISSION STATEMENTS

Permission for including co-authored material in this dissertation was obtained through e-mail from *Vijetha Gattupalli* in one case where we both had equal contribution.

Regarding material used from the IEEE papers, the policy of IEEE is as follows:

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license. Authors need to follow certain requirements listed below (taken from IEEE website):

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line ©2011 IEEE. 2) In the case of illustrations or tabular material, we require that the copyright line ©[Year of original publication] IEEE appear prominently with each reprinted figure and/or table. 3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/credit notice should be placed prominently in the references: ©[year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication] 2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line. 3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'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 `http://www.ieee.org/publications_standards/publications/rights/rights_link.html` to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

Regarding material used from SPIE papers:

I obtained permission for reusing the material from SPIE papers under the following conditions:

- You obtain permission of one of the authors;
- The material to be used has appeared in our publication without credit or acknowledgment to another source; and
- You credit the original SPIE publication. Include the authors' names, title of paper, volume title, SPIE volume number, and year of publication in your credit statement.

APPENDIX B

RELATED PUBLICATIONS

- **Parag S. Chandakkar**, Ragav Venkatesan and Baoxin Li, "MIRank-KNN: multiple-instance retrieval of clinically relevant diabetic retinopathy images." *Journal of Medical Imaging*, Volume 4, Issue 3, 2017

- **Parag S. Chandakkar**, Yikang Li, Kevin Ding and Baoxin Li, "Strategies for Re-training a Pruned Network in an Edge Computing Paradigm", *IEEE EDGE*, 2017.

- Bishwaranjan Bhattacharjee, Matthew Hill, Hui Wu, **Parag S. Chandakkar**, John R Smith, Mark Wegman, "Distributed learning of deep feature embeddings for visual recognition tasks", *IBM Journal of Research and Development*, Volume 61, Issue 4, 2017.

- **Parag S. Chandakkar** and Baoxin Li, "Joint Regression and Ranking for Image Enhancement", in *IEEE WACV*, 2017

- **Parag S. Chandakkar** [1] , Vijetha Gattupalli[1] and Baoxin Li, "A Computational Approach to Relative Aesthetics", *IEEE ICPR*, 2016 [Oral]

- **Parag S. Chandakkar** and Baoxin Li, "A Structured Approach to Predicting Image Enhancement Parameters", in *IEEE WACV*, 2016 [Acceptance rate: 34%]

- Ragav Venkatesan, **Parag S. Chandakkar** and Baoxin Li, "Simpler non-parametric methods provide as good or better results to multiple-instance learning", in *IEEE ICCV*, 2015 [Acceptance rate: 30.92%]

- **Parag S. Chandakkar**, Qiongjie Tian and Baoxin Li, "Relative Learning from Web Images for Content-adaptive Enhancement", in *IEEE ICME*, 2015 [Acceptance rate: 30%]

- **Parag S. Chandakkar**, Yilin Wang and Baoxin Li, "Improving Vision-based Self-positioning in Intelligent Transportation Systems via Integrated Lane and Vehicle Detection", in *IEEE WACV*, 2015 [Acceptance rate: 36.7%]

- **Parag S. Chandakkar**, Ragav Venkatesan and Baoxin Li, "Video-Based Self-positioning for Intelligent Transportation Systems Applications", in *Advances in Visual Computing, Lecture Notes in Computer Science (LNCS), Springer*, 2014 [Oral].

- **Parag S. Chandakkar** and Baoxin Li, "Investigating Human Factors in Image Forgery Detection", *ACM MM Intl. Workshop on Human Centered Event Understanding*, 2014.

- **Parag S. Chandakkar**, Ragav Venkatesan and Baoxin Li, "Retrieving clinically relevant diabetic retinopathy images using a multi-class multiple-instance framework", in *Proceedings of SPIE conference on Medical Imaging*, 2013 [Oral].

- Archana Paladugu, **Parag S. Chandakkar**, Peng Zhang and Baoxin Li, "Supporting navigation of outdoor shopping complexes for visually-impaired users through multi-modal data fusion", in *IEEE ICME*, 2013.

[1] Authors contributed equally.

- Ragav Venkatesan, **Parag S. Chandakkar**, Baoxin Li, Helen K. Li, "Classification of Diabetic Retinopathy Images Using Multi-Class Multiple-Instance Learning Based on Color Correlogram Features", in *IEEE EMBS*, 2012.

APPENDIX C

RELATED PATENT

Baoxin Li, Parag Shridhar Chandakkar, and Qiongjie Tian. ”Systems and methods for a content-adaptive photo-enhancement recommender.” U.S. Patent No. 9,576,343. 21 Feb. 2017.