ABSTRACT

Title of dissertation: DOCUMENT AND NATURAL IMAGE APPLICATIONS OF DEEP LEARNING

Le Kang, Doctor of Philosophy, 2015

Dissertation directed by: Dr. David Doermann
University of Maryland Institute for Advanced Computer Studies

Professor Rama Chellappa
Department of Electrical and Computer Engineering

A tremendous amount of digital visual data is being collected every day, and we need efficient and effective algorithms to extract useful information from that data. Considering the complexity of visual data and the expense of human labor, we expect algorithms to have enhanced generalization capability and depend less on domain knowledge. While many topics in computer vision have benefited from machine learning, some document analysis and image quality assessment problems still have not found the best way to utilize it. In the context of document images, a compelling need exists for reliable methods to categorize and extract key information from captured images. In natural image content analysis, accurate quality assessment has become a critical component for many applications. Most current approaches, however, rely on the heuristics designed by human observations on severely limited data. These approaches typically work only on specific types of images and are hard to generalize on complex data from real applications.

This dissertation looks to address the challenges of processing heterogeneous visual data by applying effective learning methods that directly model the data with minimal preprocessing and feature engineering. We focus on three important problems - text line detection, document image categorization, and image quality assessment. The data we work on typically contains unconstrained layouts, styles, or noise, which resemble the real data from applications. First, we present a graph-based method, learning the line structure from training data for text line segmentation in handwritten document images, and a general framework to detect multi-oriented scene text lines using Higher-Order Correlation Clustering. Our method depends less on domain knowledge and is robust to variations in fonts or languages. Second, we introduce a general approach for document image genre classification using Convolutional Neural Networks (CNN). The introduction of CNNs for document image genre classification largely reduces the needs of hand-crafted features or domain knowledge. Third, we present our CNN based methods to general-purpose No-Reference Image Quality Assessment (NR-IQA). Our methods bridge the gap between NR-IQA and CNN and opens the door to a broad range of deep learning methods. With excellent local quality estimation ability, our methods demonstrate the state of art performance on both distortion identification and quality estimation.

DOCUMENT AND NATURAL IMAGE APPLICATIONS OF DEEP LEARNING

by

Le Kang

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2015

Advisory Committee:
Professor Rama Chellappa, Chair
Dr. David Doermann, Co-Advisor
Professor Min Wu
Professor Ramani Duraiswami
Professor Piya Pal

Acknowledgments

On the journey to my Ph.D. degree, I have received support and encouragement from many individuals. First and foremost, I would like to thank my advisor Dr. David Doermann. I am grateful for his patient guidance on choosing research topics, timely supervision on project progress, late night and early morning editing on papers, and running the comfortable lab, LAMP, for us over the years. Without him, I would never have been able to finish the dissertation. I also would like to thank Professor Rama Chellappa, for showing me the big picture of computer vision and machine learning. Sincere thanks goes to other committee members, including Professor Min Wu, Professor Ramani Duraiswami, and Professor Piya Pal. Thanks also goes to Professor Prakash Narayan, who served on my Ph.D. proposal committee and provided invaluable suggestions.

My colleagues and friends were extremely supportive. Peng Ye and Jayant Kumar worked closely with me in the lab, and I cherish the time we together discussed, experimented, ate, and laughed. Yi Li collaborated with me remotely, yet he provided amazingly effective help in shaping research ideas and executing them. Thanks also goes to Wael Abd-Almageed, Elena Zotkina, Rajiv Jain, Xianzhi Du, Sungmin Eum, Varun Manjunatha, Qixiang Ye, Jingtao Xu, Zhuolin Jiang, Fang Wang, Jin Chen, Huaigu Cao, Yan Zhang and Xiaofan Lin.

Finally, I owe my deepest thanks to my family - my parents, grandparents, aunt, and uncle. They have always stood by me and encouraged me ever since I was born.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

UMIACS	University of Maryland Institute for Advanced Computer Studies
NSF	National Science Foundation
DARPA	Defense Advanced Research Projects Agency
NIST	National Institute of Standards and Technology
CC	Connected Components
SGS	Sequential Gap Significance
SVM	Support Vector Machine
HOCC	Higher Order Correlation Clustering
SDP	Semi-Definite Programming
MSER	Maximally Stable Extremal Region
ANN	Artificial Neural Network
CNN	Convolutional Neural Network
IQA	Image Quality Assessment
NR-IQA/NRIQA	No-Reference Image Quality Assessment
FR-IQA	Full-Reference Image Quality Assessment
LCC	Linear Correlation Coefficient
SROCC	Spearman Rank Order Correlation
OCR	Optical Character Recognition

Chapter 1: Introduction

Digital image content on the internet has been increasing rapidly, and we are in need of effective methods to evaluate and understand visual data. The complexity of visual data requires algorithms with enhanced learning capability and less dependence on prior knowledge which can be difficult to obtain and is often inaccurate. While many topics in computer vision have benefited from machine learning, some document analysis and image quality assessment problems have not yet found the best way to utilize machine learning. In the context of document images, a compelling need exists for reliable methods to categorize and extract key information from captured images. In natural image analysis, accurate quality assessment has become a critical component for many applications. Most current approaches, however, rely on the heuristics designed by human observations on severely limited data. These approaches typically work only on specific types of images, and are hard to generalize on complex data from real applications.

This dissertation aims to address the challenges of processing heterogeneous visual data by applying effective learning methods that directly model the data, with minimal preprocessing and feature engineering. We focus on open problems including text line detection, document image categorization, and image quality as-

sessment. The data we work on typically contains unconstrained layouts, styles, or noise, which resemble the real data from applications. We first define the problems in various application areas, then formulate them as machine learning tasks. We provide solutions that learn from the data with minimal hand-designed rules, achieving state of the art performance.

1.1 Handwritten and scene text line detection

Text can provide crucial information about the image where it resides. Sometimes, we are interested only in the information carried by text that occupies the majority of the image, such as scanned documents. In other cases, text may not be dominant but provide context to understand the image, such as a traffic sign in a street view image. Extracting text from images has been extensively studied by the research community. Despite the successes obtained on well-constrained conditions, such as clean machine printed documents, more problems remain to solve. We address text line detection on unconstrained handwritten documents and natural images as part of the MADCAT program.

Text line extraction on scanned documents presents an important step for many document processing tasks such as word/character recognition [1], layout-analysis [2] and skew estimation [3]. Unlike printed documents, the lines in handwritten documents are often skewed and curved. Moreover, overlapping spatial envelopes of text lines, touching of characters across lines, and irregularity of layout and character shapes originated from the variability of writing styles make the prob-

lem more challenging. Recent work has focused on addressing each of these issues individually, but a unified framework of all the challenges associated with handwriting is still necessary. The past few years have seen tremendous growth and success of learning based methods for object recognition and image segmentation, but existing methods for text line segmentation in handwritten documents tend to use unsupervised approaches. Because training data is difficult to obtain, most systems use only a small validation set for tuning parameters. Earlier methods primarily targeted printed and a limited class of handwritten documents where encoding heuristics in an unsupervised setting produced acceptable results. For unconstrained handwritten documents, it has been difficult to encode all of the knowledge, and the unsupervised methods have not performed satisfactoryly for many datasets. When the images are degraded, many existing methods fail even for printed and less complex handwritten documents. Although many tools for efficient groundtruthing [4] are available and it is less expensive to obtain labels for text line data, most existing approaches have hand-coded the knowledge obtained from inspecting the training data.

Text in natural images carries important semantic information. Localizing text aids scene understanding and is also relevant to a number of computer vision applications, such as internet image indexing, mobile vision, and low vision aids [5]. Detecting text lines in natural images differs from the detection in handwritten documents, because typically non-text content prevails in a natural image. The major challenge is to locate text accurately from numerous distracting patterns. Generally, text lines in natural images are curvilinear and diversified with different

orientations, fonts, sizes, and scripts, designed to attract attention. However, most current methods focus on building models for a certain range of fonts and scripts, such as detection-by-recognition approaches. The bounding boxes of the areas for potential character regions are detected and classified, and text line structures are enforced to link bounding boxes heuristically. These kinds of approaches may not be easily adapted to multi-orientation cases. We hypothesize that text can be better identified by properties of a group rather than by individual characters. Individual image elements vary greatly and tend to cause false alarms for those methods explicitly using character models. A group of similar elements can provide more robust statistics for discriminating text from noise. Therefore, it is natural to group image elements based on pairwise and groupwise similarity, then classify them as text or non-text regions. This can be regarded as a trade-off between top down detections and bottom up heuristic rules.

For text line detection in unconstrained handwritten documents, we develop a dictionary-based method that learns the similarity from training data. We use image-patches in the training data to obtain the contextual evidence needed for detecting text lines in a new document images. We construct a graph based on a dictionary of context patches, then partition the graph using Normalized Cuts guided by a novel method that predicts the number of clusters. For the text detection from natural images, we propose a higher-order correlation clustering based framework to detect multi-oriented scene text lines with less dependency on font or language. We group similar elements first, then identify each group as text or non-text. Both of our methods achieve state of the art performance.

1.2 Document image categorization

Classifying and grouping large collections of document images into known categories is often a prerequisite step toward document understanding tasks, such as text recognition, document retrieval, and information extraction [6]. These tasks can be greatly simplified if we know a priori the genre or the layout-type of documents. Existing approaches in the literature differ mainly in their choices of local features, global representations, and learning mechanisms [7]. Various structure or layout-based features have been introduced [8–12] and are shown to be effective for document image classification and retrieval. These approaches, however, are limited to a particular class of documents, such as forms, memos, contracts, and orders. To apply existing classification systems to other types of documents, we need to reconsider spatial features and tune them manually. Moreover, when the content and structure in documents are unconstrained, as in handwritten documents, pre-defined features may not be able to capture all variations of a particular class.

A more general approach needs to be developed, which automatically learns different abstractions of structure hierarchy and spatial relationship among document elements. Document images usually have a hierarchical structure, such as cells in rows and columns of tables, words in sentences, and sentences in paragraphs. These hierarchical patterns often repeat throughout a document. These properties imply the possibility of learning the layout as a combination of small group of middle or lower level features.

We present a general approach for document image classification using Con-

volutional Neural Networks (CNNs). CNN offers a kind of neural networks that shares weights among neurons in the same layer. CNN discovers spatially local correlation by enforcing a local connectivity pattern between neurons of adjacent layers [13]. With multiple layers and pooling between layers, CNN automatically learns the hierarchical layout features with tolerance to spatial translation, and by sharing weights, it captures repeating patterns efficiently.

1.3 Image quality assessment

Visual quality is a complex and inherent characteristic of images. In principle, it offers the measure of the distortions when compared with ideal imaging models or perfect reference images. When reference images are available, Full Reference Image Quality Assessment (FR-IQA) methods [14–18] can be directly applied to quantify the differences between distorted images and their corresponding ideal versions. State of the art FR measures, such as VIF [14] and FSIM [15], achieve a high correlation with human perception. However, in many practical computer vision tasks, no perfect versions exist of the distorted images, thus these tasks require No-Reference Image Quality Assessment (NR-IQA). NR-IQA measures can directly quantify image degradations by exploiting features that are discriminant for image degradations.

Most successful approaches toward this challenge use the Natural Scene Statistics (NSS) based features. Typically, NSS-based features characterize the distributions of certain filter responses, and they are extracted in image transforma-

tion domains, such as the wavelet transform [19] and DCT transform [20]. These methods usually operate slowly due to the use of computationally expensive image transformations. NSS-based methods heavily rely on domain knowledge. Different NSS-based methods demonstrate large variations in performance given different design choices. Traditional NSS-based methods achieve much lower performance when compared to the state of the art. Recent development in NR-IQA methods, CORNIA [21, 22] and BRISQUE [23], promote extracting NSS features from the image spatial domain, which leads to a significant reduction in computational time. CORNIA shows that it is possible to learn discriminant image features directly from the raw image pixels, instead of using handcrafted features. Although CORNIA and BRISQUE have performed promisingly, a performance gap still exists between machine and human. The state of the art methods typically focus on 5 types of distortions on the relatively easy LIVE dataset. Many other types of distortions still exist and need to be assessed, for example, the performance on the 17 types of distortions on TID2008 dataset remains far from perfect. Challenges, such as assessment of local image distortions and multi-distortions, are rarely attempted, let alone solved. Therefore, NR-IQA still exists as an open problem.

We explore using a Convolutional Neural Network (CNN) to learn discriminant features for the NR-IQA task. In one of CNN's advantages, it can take raw images as input and incorporate feature learning into the training process. With a deep structure, the CNN can effectively learn complicated mappings while requiring minimal domain knowledge. To the best of our knowledge, CNNs have not been applied to general-purpose NR-IQA. The original CNN is not designed for capturing

image quality features. In the object recognition domain good features generally encode local invariant parts, however, for the NR-IQA task, good features should be able to capture NSS-like properties. The difference between NR-IQA and object recognition makes the application of CNN nonintuitive. In one of our contributions, we modified the network structure so that it can learn image quality features more effectively and estimate the image quality more accurately. We extend our CNN to a multi-task architecture, which can simultaneously identify the distortion and estimate the quality.

1.4 Deep learning and its applications

Recently deep learning has gained researchers' attention with great success on various machine learning tasks. Common public sources, such as Wikipedia, characterize deep learning as those machine learning algorithms that use architecture consisting of multiple non-linear transformations to model high-level abstractions in data. Many successful deep learning methods are based on artificial neural networks (ANNs). ANNs had been less employed compared to other shallow models, such as support vector machines (SVM), because it is difficult find optimal solutions with ANNs. Also, their training processes require a tremendous amount of computation. The research community recently rediscovered the value of ANNs, and can obtain excellent performance by applying a number of modern techniques to overcome the traditional flaws of ANNs and achieving their potential efficacy.

Deep learning has its root in human information processing mechanisms (e.g.,

vision and speech), which suggests the need of deep architecture for extracting complex structure and building internal representation from input signals. Typical deep learning systems are built with deep architectures consisting of many layers of nonlinear processing stages, where each lower layer's outputs are fed to its immediate higher layer as the input. Compared to shallow architectures, deep learning methods usually have greater modeling and representational power, which enables us to deal with more complicated real-world applications, such as understanding unconstrained natural images and visual scenes.

Among several streams of deep learning methods, CNNs are the most widely used as the top performing methods for object recognition tasks. CNNs represent a particular kind of multi-layer neural networks, being designed to exploit how objects usually undergo shifts or translational variations in images. Feature detectors that work well on one part of the image are likely to be succeed across the entire image. With the guidance of this knowledge, each neuron in a convolutional layer is forced to receive input from only a small set of neighboring neurons in the previous layer, called the receptive field. In addition, the neurons located at different places (with different receptive fields) are forced to have identical weights. Each convolutional layer is typically followed by a spatial pooling operation, which enhances tolerance to the variation of features. The output of one convolutional layer is partitioned into small sets, and one value is sampled (mean or the max) from each set as the input for the next layer. With local receptive fields, neurons in the first convolutional layer extract low level visual features, such as edges, dots and corners. These subsequent layers then combine these features to detect higher-order features.

CNNs integrate feature learning and classification/regression model learning into one process, and can recognize visual patterns directly from pixel images with minimal preprocessing. With the wide application of CNNs in recent years, handcrafted features are losing focus in the object recognition community. Domain knowledge from experts is becoming less important for designing visual recognition systems. CNNs have shown superior performance on many standard object recognition benchmarks [24–26], with a large margin compared to other kinds of methods.

We would like to address several problems in image understanding and processing with the help of deep learning techniques, including document image categorization and blind image quality assessment. In the past, all of these problems were handled using various handcrafted features and shallow-structure learning methods. Although some previous methods demonstrated reasonable performance, these problems remain mostly unsolved with room for improvement. We approach these problems from the perspective of deep learning. Instead of relying heavily on domain knowledge or heuristics, our methods directly work on the image domain, learning the complicated features and classifiers/regressors in one network.

In our problems, the typical structure of CNN may not be the best choice because it is designed for object recognition. For example, blind image quality assessment has different properties from the typical object recognition, and we need to design a customized network structure to adapt to the problem domain for better performance. We demonstrate through experiments that our methods achieve state-of-the-art performance on all the problems we work on.

1.5 Outline

In the following chapters we present the details of our methods. Chapter 2 addresses text line detection in handwritten documents and natural images. Chapter 3 describes our work on document image categorization. Chapter 4 presents solutions to No-Reference image quality assessment problems. In Chapter 5, we summarize our work.

Chapter 2: Handwritten and Scene Text Line Detection

2.1 Text line detection in handwritten documents

Text line extraction remains an important step for many document processing tasks such as word/character recognition [1], layout-analysis [2], and skew estimation [3]. Unlike printed documents, the lines in handwritten documents are often non-uniformly skewed and curved. Moreover, overlapping spatial envelopes of text lines, touching of characters across lines, and irregularity of layout and character shapes from the variability of writing styles produce a more challenging problem. Recent work has focused on addressing each of these issues individually, no one has developed a unified framework to account for all the challenges associated with handwriting. For example, methods based on level-sets [27] are effective but computationally slow, and methods based on connected-components (CCs) are fast but challenged by touching components and overlapping lines [28]. Similarly, projection-based methods [29] cannot handle overlapping lines or touching, and perform poorly when given a large variation in character or word dimensions. In [30], the authors

report that the method, which gave almost 100 percent accuracy on ICDAR 2009 competition data set [31], performed poorly on field data of degraded handwritten Arabic documents, and an ensemble of multiple methods was required to obtain a reliable segmentation/recognition accuracy.

In the past few years tremendous growth and success of learning based methods for object recognition and image segmentation [32] has occurred, but existing methods for text line segmentation still tend to use unsupervised approaches. Training data are difficult to obtain, so most systems use only a small validation set for tuning parameters. Earlier methods were targeted primarily at printed and a limited class of handwritten documents where heuristics produced acceptable results. For unconstrained handwritten documents, it has been difficult to encode all of the knowledge into the model, and the performance using unsupervised methods has not been satisfactory for many data sets. When the images are degraded, many existing methods fail even for printed and less complex handwritten documents. Although many tools for efficient groundtruthing exist [4], and it is less expensive to obtain labels for text line data, most existing approaches hand-coded the knowledge obtained from "inspecting" the training data.

We have developed a graph-based method for text line segmentation that uses image-patches in the training data to obtain the contextual evidence needed for detecting text lines in a new document images. Training images are transformed into summary maps, and patches with local groundtruth masks are randomly sampled from the summary maps. Using k-medoids clustering, representative patches are selected to construct a codebook. A new document image is also transformed into a

summary map, and each non-zero bin is considered as a node in the graph. Context patches are sampled at each node. For each context patch, the best match is found from the codebook, and its associated groundtruth mask assists in updating the similarities among the nodes in the patch. After constructing the similarity graph, normalized cut [33] is employed to partition the graph. A novel Sequential Gap Significance feature combined with a support vector machine predicts the number of clusters. The partitioning continues recursively until no further split occurs. Decisions on the graph are mapped back to the original image to provide the text line segmentation. A postprocessing is employed to resolve fragmentation errors.

Our contributions includes learning a graph based similarity among large variations of text lines patterns, and solving the graph partitioning with accurate prediction of the graph structure.

2.1.1 Related work

Existing methods in an unsupervised setting for text line segmentation can be broadly categorized into three classes: top-down projection based methods [29], bottom-up component grouping based methods [28, 34], and hybrid methods [30]. While top-down methods partition the document image recursively to obtain text lines, bottom-up methods group small units of the document image (pixels, CCs, characters, words) into text lines. Bottom-up grouping can be implemented through clustering, which aggregates image components according to similarity and does not rely on the assumption of straight lines. Kumar et al. [34] proposed a local orienta-

tion detection based similarity for clustering primary CCs using Affinity propagation. In a post-processing step, errors in the text line are corrected iteratively using Expectation-Maximization (EM) [28]. Their method achieves high accuracy on a set of Arabic documents, but given the method is based on grouping CCs, it is less likely to work on degraded document images where CCs are broken. Another method that achieves high accuracy on many data sets is based on steerable directional filters [35]. It determines the local orientation of a text line by scanning in multiple directions for maximum response of a convolution of the filter with the image. In the final step, touching components are split at the contour level and the character images are reconstructed. This method also fails to group components when the image is degraded and spacing between character/words in text lines varies greatly.

Some recent work has incorporated supervised learning at different levels. Yin and Liu [36] learn a distance metric for pairs of CCs using a labeled dataset. The CCs are then grouped into a tree structure, where text lines are extracted by dynamically cutting the edges using a hyper-volume reduction criterion. By learning the distance metric, this algorithm performs robustly with multi-skewed and curved text lines. The method works only if spatial envelopes of text lines do not overlap, which limits the application to unconstrained handwritten lines of different scripts. Manohar et al. [30] proposed an ensemble system as a formulation of graph-clustering problem and applied it to combine outputs of multiple text line segmentation methods. They construct a co-occurrence graph with nodes corresponding to CCs and edges connecting pairs of CCs with an associate cost of having the pair same label (line). The edge cost is determined by the cost of a false split and merge, and

the likelihood that a pair of CCs has the same label conditioned on the ensemble output. These likelihoods are learned during training. Text line segmentation is then formulated as the problem of minimum cost partitioning of the nodes in the graph. As expected, the method performs better than individual methods but the scaling and time performance of method is always lower-bounded by performance of individual methods.

2.1.2 Approach

Our method learns the local spatial relationship between text lines and applies an effective strategy to predict the graph structure for partitioning.

Visual codebooks constructed from invariant descriptors extracted from local image patches have been widely used in texture analysis and visual recognition [37]. In [38], an effective method for image quality assessment used raw image patches for codebook construction and achieved state-of-the-art performance. Inspired by these methods, we use image patches to encode the local evidence of text lines.

2.1.2.1 Context patch extraction

An $M \times N$ binary document image is divided into square cells of size $p \times p$. A summary map of $(M/p) \times (N/p)$ bins is constructed, where each bin records the number of foreground pixels in the corresponding cell of the document image. Thus, the summary map resembles a downsampled version of the original image. In the summary map, we define the context patch of a bin as an $H \times W$ region

centered at this bin, where H and W are odd numbers so an exact geometric center exists. In practice, each context patch is constructed as a vector by concatenating its columns, and is normalized to unit norm. Figure 2.1(a) and (b) show part of a document image and its summary map with a sampled context patch.

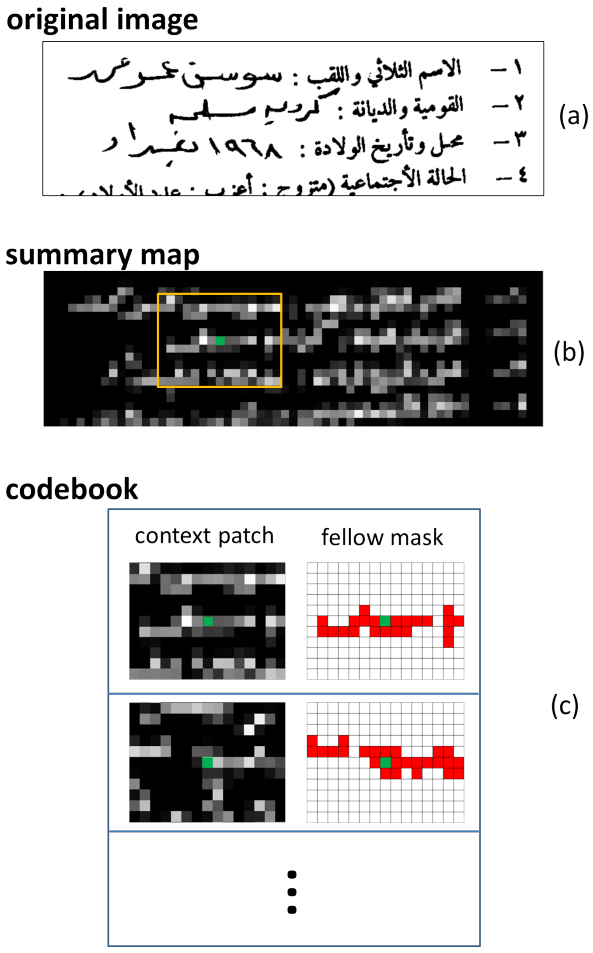


Figure 2.1: (a) Part of a sample image (b) Summary map and a context patch (in yellow box) centered at the bin marked green. (c) Codebook consisting of context patches and fellow masks. Fellows are marked in red.

2.1.2.2 Codebook construction

Each training image is transformed into a summary map, and context patches are randomly sampled at non-zero bins. The ground truth of each training image exists at pixel level, i.e., each foreground pixel has a label indicating to which text line it belongs. Thus we can obtain the bin level ground-truth by a majority voting among the pixels to which a bin corresponds. We define one bin as a "fellow" of another if the two bins have the same label, i.e. they are in the same text line. Each context patch is accompanied by a mask of the same size, termed the fellow mask, which records fellows of the center bin in this context patch.

Considering the storage of the codebook and the efficiency of comparison between a query patch and all codewords, it is better to use fewer patches to represent the space of training data. We simply use k-medoids to select representative patches. Figure 2.1(c) shows a codebook containing context patches and fellow masks.

The best matching codeword for a query patch is obtained by finding the nearest neighbor in Euclidean space, which is equivalent to finding the maximum inner product, given that the points sit on the unit sphere. It is also possible to store all the sample patches and apply an efficient (approximate) nearest neighbor search.

2.1.2.3 Computing similarities in the graph

We formalize the core step of text line segmentation as a graph partitioning problem. Each non-zero bin corresponds to a node in the graph. The weight of an

edge, i.e., the similarity between two nodes, indicates how likely two nodes have the same label. Therefore, a graph that reasonably represents the structure of text lines should have high similarity between any two nodes in the same text line and low similarity where not.

To segment a document image, we first transform it into a summary map. For each bin in the summary map, we extract its context patch and find the best match in the codebook. We define a non-zero bin as a potential fellow of the center bin if its position is marked as fellow by the codeword's fellow mask. The similarities between any two nodes are initialized as zero, and they are updated as follows. Assuming the center bin of a context patch has f potential fellows, the similarity between the center bin and each potential fellow is increased by $1/f$. Using this strategy, a bin with fewer fellows distributes stronger similarity to each fellow to maintain their connections, while a bin densely surrounded by fellows will distribute smaller similarity to each of its fellows. Together, they will have sufficient connections. This similarity updating strategy achieves a balanced similarity distribution among dense and sparse text regions, which greatly facilitates the graph partitioning stage afterwards.

2.1.2.4 Estimating the number of clusters and graph partitioning

We employ normalized cut [33] to partition a similarity graph. Normalized cut provides a near optimal solution for partitioning a graph where only local similarities are obtained for data points, which complements our situation well.

However, it is necessary to determine the number of clusters, which remains a difficult problem. In [39], an eigengap heuristic is suggested - a relatively large eigengap usually indicates the number of clusters if the dataset contains well pronounced clusters. We tested this eigenvalue heuristic by choosing the number k, if the kth eigengap is the largest one. This strategy did not perform satisfactorily, with an accuracy below 20% on a semi-synthetic dataset explained in Section 4. We observed that the real kth gap was usually significant, but often larger gaps appeared at somewhere larger than the k. We conjectured that the number should be determined by both the sequential order and the significance of the eigengaps. We designed a feature called the Sequential Gap Significance (SGS). Assume we have m eigenvalues $\lambda_1, \lambda_2, \cdots, \lambda_m$ in ascending order and their eigengaps g_i computed as follows:

$$g_i = |\lambda_{i+1} - \lambda_i|, i = 1, 2, \ldots, m-1 \tag{2.1}$$

The SGS feature is expressed as follows:

$$SGS(i) = \begin{cases} \dfrac{g_i}{T}, & i = 1, \\ \dfrac{g_i}{max_{j=1,\ldots,i-1} g_j}, & i = 2, \ldots, m-1. \end{cases} \tag{2.2}$$

where T is a small positive offset. Essentially, SGS measures how "significant so far" each eigengap is. Figure 2.2 compares eigenvalues, eigengaps, and SGS features in characterizing the cluster structure. If we let k' be the number of clusters such that

$$k' = \arg \max_{i=1,\ldots,m-1} SGS(i) \tag{2.3}$$

then a reasonable estimation accuracy around 90% was obtained on the same semi-synthetic dataset mentioned earlier. This still does not perform satisfactorily. Training data is available, so we can solve this problem in a supervised learning fashion. We trained a multiclass SVM classifier that took the SGS features as input feature vectors and predicted the number of clusters, and achieved about 97% accuracy.

One remaining issue with the method described above is that it can only estimate the cluster number between 1 and $m - 1$. In practice, the real cluster number k is not guaranteed to be less than m. To address this issue, we iteratively partition the subgraphs until no further split is predicted.

2.1.2.5 Text line segmentation and postprocessing

After graph partitioning, each bin obtains a label indicating its cluster membership, and the mainframes of text lines are clear. Then each bin's label is mapped to the foreground pixels on the original image, and a coarse text line segmentation is obtained.

We observed that characters touching across text lines are reasonably segmented, while some non-touching strokes close to neighboring text lines were incorrectly segmented into two or more pieces. This fragmentation is understandable because the proposed approach is based on bins rather than connected components. We developed a defragmentation technique to repair the errors. For each fragment

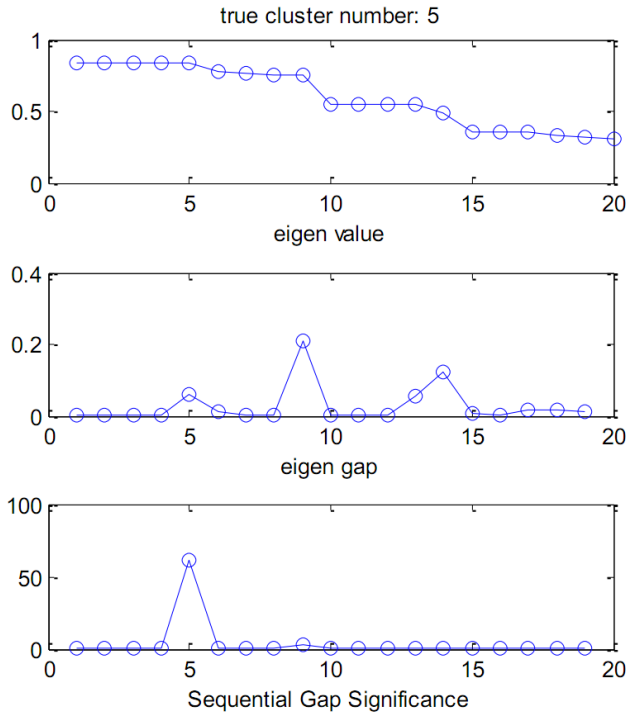


Figure 2.2: Comparison of eigenvalues (negated and shifted, least becomes greatest), eigengaps and SGS feature in characterizing the cluster structure.

in a connected component, we find the support text that has the same label and resides in the neighborhood of context patch size. Robust multilinear regression [40] is used to fit a line to the support text, treating each pixel as a point. Relative to this line, we compute the residual of the mass center of the fragment, the average residual of the support text and the standard deviation, denoted as R_c, R_{avg} and R_{std}, respectively. We define the fragment quality Q as follows:

$$Q = \frac{R_c - R_{avg}}{R_{std}} \tag{2.4}$$

If $Q < 1$, the fragment is considered a major fragment, and its label remains unchanged. Otherwise, it is a minor fragment and ready to be merged into the closest

major fragment, or the fragment with minimum Q if no major fragment exists in this connected component. With this simple postprocessing, many fragmentation errors are resolved, while most touching segmentations remain.

2.1.3 Experiments

2.1.3.1 Datasets

Two datasets were used in our experments, an Arabic field dataset and the ICDAR2009 handwriting segmentation contest dataset [31]. The Arabic field dataset contains 487 images with a total of 13,904 text lines. Images in this dataset present complex layouts and different levels of noise and degradation, which are similar to the data used in [30]. Figure 2.3 shows an example from the field dataset. The ICDAR2009 contest dataset contains 200 images with a total of 4,034 text lines. We randomly split this dataset into two parts, training and test, each with 100 images.

The traditional evaluation criterion was employed as in [31]. For a result region R and a ground truth Region G, a matchscore is computed as

$$matchscore(R, G) = \frac{\|R \cap G\|}{\|R \cup G\|} \tag{2.5}$$

where $\|\cdot\|$ denotes the cardinality of a set. The result region R is accepted if its matchscore is above the threshold T.

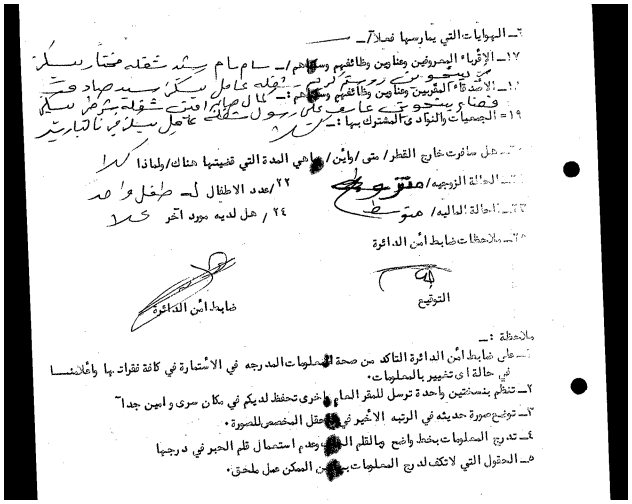


Figure 2.3: An example from the field dataset.

2.1.3.2 Results

Text line segmentation on the Arabic field dataset presents a highly challenging task [30]. The whole dataset was randomly divided into three parts: 150 for training, 100 for validation and 237 for test. The codebook was constructed on the training set, and the SVM parameters were optimized on the validation set. Due to the noisy nature of this dataset, a loose threshold $T = 0.75$ was used for evaluation to allow minor matching errors. Table 2.1 shows that the proposed approach outperformed [28] on the test set. Figure 2.4 shows a few samples of segmentation results. We can see the proposed approach can segment complex text lines.

On the ICDAR2009 dataset, the codebook was not directly constructed on the training set, instead, we first created two semi-synthesized datasets S_1 and S_2 from the training set. S_1 and S_2 are created as follows: each training image produced five copies $I_1, I_2, \ldots, I_5$ and j text lines are randomly erased on $I_j, j = 1, 2, \ldots, 5$.

Figure 2.4: Samples of segmentation results.

Then, we obtained 500 semi-synthesized images for S_1 and S_2 each. The semi-synthesized datasets provide graphs with different number of clusters for training. If we train our model directly on the original data, connections exist between bins of neighboring text lines, thus most images produce a single graph containing all text lines, and training samples of small cluster number are scarce. When text lines are randomly erased, separate graphs with different cluster numbers emerge in each semi-synthesized image, thus obtaining balanced training samples.

The codebooks and RBF kernel SVM models under different sizes of context patch and bins were constructed on S_1. Parameters of SVM models were optimized on S_2.

Table 2.1: F1 on the field dataset (%)

[28]	proposed
56.5	64.9

We simply applied the proposed method without much post-processing and optimization targeted at this dataset, so we have not achieved state-of-the-art accuracy (99.53% reported in the contest [31]). Table 2.1 shows the performance on test set with different acceptance threshold T. Lacking a full comparison given the current unavailability of [28]'s performance at $T = 0.75$, we can see the accuracy approaches 100% at the lower threshold. This indicates that main bodies of text lines are correctly detected for the most part, and many errors lie in character and stroke level segmentation. This is confirmed by careful visual examination.

Table 2.2: F1 on the ICDAR2009 contest dataset (%)

T	[28]	our method
0.95	97.8	98.3
0.75	–	99.4

2.2 Scene text line detection

Text in natural scenes carries important semantic information. Localizing text aids scene understanding, and is also relevant to a number of computer vision applications, such as internet image indexing, mobile vision, and low vision aids. Generally, text lines in natural images are curvilinear and diversified with different orientations, fonts, sizes, and scripts. We hypothesize that text can be better identified by properties of a group rather than those of individual characters (Figure 2.5). Individual image elements vary greatly and tend to cause false alarms for those methods explicitly using character models. However, a group of similar elements provides more robust statistics for discriminating text from noise. It is, therefore, natural to group image elements based on pairwise and groupwise similarity, then classify them as text or non-text regions. This can be regarded as a trade-off between top down detections and bottom up heuristic rules.

We approach the text detection problem from an image partitioning perspective and propose a general framework to detect multi-oriented scene text lines with less dependency on font or language. We aim to group similar elements first, then to identify each group as text or non-text. Specifically, we use MSERs [41] as the basic elements and partition them to segments. This results in a few important changes in the processing flow. Instead of focusing on the strong detection and filtering approaches, we use weak hypotheses for similarity clustering, followed by region-based filtering.

The elongated nature of text lines suggests that the long range dependencies

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

Figure 2.5: Intermediate results in our procedure. From top to bottom, MSER extraction, local text line hypotheses (green bounding boxes), pairwise edges in HOCC, results for HOCC, and results for texture classification (yellow bounding boxes). Different MSERs/regions are represented by different colors.

among multiple nodes can be exploited, so we explore the Higher-Order Correlation Clustering (HOCC) [42]. Weak hypotheses of local text lines can be generated based on their spatial alignment and appearance consistency with respect to their neighbors. HOCC may reject these hypotheses depending on the objective function of these similarities. This results in either a merge or a split of hypotheses, which offers a pleasant property to pursue. In another appealing property, HOCC allows for large margin training. Using structured SVM [43], the parameters of HOCC can be learned from the training data [42], and we are spared from adopting too many heuristics.

In HOCC, we use the regularization method [44] to solve the Semidefinite Programming (SDP) problem [45]. The original HOCC proposed a linear programming relaxation solution with a large number of inequality constraints. This complex linear system can be written elegantly in the SDP framework, allowing us to solve the system effectively with a large number of variables in a few seconds. After clustering, we use a texton-based texture classifier to discriminate between text and non-text areas.

We compare our method with those aiming at detecting multi-oriented and multi-language text. On a recently published dataset, our method generates promising results compared to the state of the art methods.

2.2.1 Related work

The text detection problem was studied extensively in recent work [46–51]. However, most current methods focus on building models for certain ranges of fonts and scripts, such as detection-by-recognition approaches [47, 48, 51]. The bounding boxes of the areas for potential character regions are detected and classified, and text line structures are enforced to link bounding boxes together heuristically. These kinds of approaches may not be easily adapted to multi-orientation cases.

Correlation clustering was originally proposed by [52] as a similarity clustering approach, which models pairwise relationships between entities instead of the entities themselves and does not require specifying the number of clusters. Higher-Order Correlation Clustering (HOCC) [42] improves on the original correlation clustering where higher-order relations can be incorporated. In HOCC, long range interactions can be defined by less accurate measurements (i.e., weak hypotheses), because they are regarded as soft constraints in clustering. HOCC may reject these hypotheses depending on the objective function of these similarities, which presents a notable differences between the higher order correlation clustering and other classical higher order Markov Random Field (MRF).

2.2.2 Approach

2.2.2.1 Building graphs for detection-by-clustering

In our approach, an image is represented by a graph of Maximally Stable Extremal Regions (MSERs). The detection amounts to identifying subgraphs, which contain text lines. We adopt MSER because our approach intends to perform detection for multiple languages at multiple orientations, thus we seek generic representations of text regions. Among graph partitioning algorithms, correlation based methods are particularly well suited to our problem. These methods rely on pairwise similarity, such as contrast, solidity (the ratio of contour area to its convex hull area), area ratio, and distance in images. More interestingly, prior knowledge can be incorporated as weak hypotheses of grouping. First, MSERs are extracted, then a graph of MSERs is constructed for an image. Second, MSERs are coarsely grouped by considering their consistency with neighbors, which create weak hypotheses. The correlation based grouping is described in the next section.

2.2.2.2 Extracting MSERs and building the graph

First, we compute MSERs for an input image (first row in Figure 2.5). One can observe that the sizes of MSERs vary, and the MSERs may correspond to character regions or noise. We then construct the graph of MSERs locally to avoid unnecessary edges between distant MSERs. Delaunay triangulation is employed to find pairwise edges in the graph.

2.2.2.3 Computing local consistency map

To construct the graph, we compute a local consistency map for each MSER. A local consistency map is a probability map, which shows how an MSER is consistent with its neighbors in a small patch. For each MSER, we consider a context patch, which is an image patch centered at the MSER. The MSER is referred to as the center MSER, and other MSERs on this patch are considered its neighbors. The patch has a size 7 times the width and height of the center MSER. We compute the consistency score θ between the center MSER and its neighbors as:

$$\theta = \exp(-\alpha D_{co} - \beta D_{sw}) \tag{2.6}$$

where D_{co} is the Euclidean distance between the colors of two MSERs, D_{sw} is the normalized stroke width difference, and α and β are constant parameters. RGB color is used and averaged across all pixels for each MSER. The stroke width of an MSER is estimated by the largest distance from an interior point to boundary, and it can be efficiently obtained using a Distance Transform.

Then, a local consistency map for this context patch is constructed by transforming all pixels of MSERs on this patch to their consistency scores. Figure 2.6 shows two examples of context patches and local consistency maps. In Figure 2.6 (a), the center MSER is part of the cartoon cat's face, thus a high consistency score occurs in other parts of its face and gives high intensity in the local consistency map. The text below the cartoon cat's face possesses very different color and stroke width, and shows low consistency scores. In Figure 2.6 (b), we observe high consis-

tency scores of the characters in the text line in the middle of the patch, while the character in the upper right corner has relatively low consistency scores due to the font difference.

2.2.2.4 Weak hypotheses generation

We generate hypotheses based on the local consistency map. Prior knowledge of text line includes 1) text line should be elongated, and 2) the projection profile of a text line should have higher variance. Therefore, we project local consistency map in different orientations from -90 to 85 degrees (with respect to the horizon), with an interval of 5 degrees to obtain 36 projection profiles.

In practice, the projection at each orientation is performed not on the entire local consistency map but on an oriented narrow region, whose width and length are 3 and 7 times those the selected MSER, respectively, to include less noise in the projection profile. The raw profile of a given orientation is computed by summing all the intensities in the narrow region along that orientation. Then, the raw profile is intensity-normalized by the mean of non-zero values in it, and its dimension is normalized to a predefined length by resampling to obtain the final projection profile.

The projection profile in the maximum variance orientation is intended to capture the text line structure, which usually forms a peak. The one in the orthogonal orientation tries to capture the regular intervals between characters, as can be observed in most cases. Figure 2.6 shows examples of projection profiles. The size of the patch is empirically determined and is not sensitive in our experiment.

(a) non-text (b)text

Figure 2.6: Local consistency map and the projection to two orthogonal directions.

We then concatenate the projection profiles in the maximum variance orientation and its orthogonal orientation into a feature vector, and feed the feature vector to a Random Forest classifier [53]. The training samples of projection profiles from text and non-text patches were labeled manually. If a patch is determined as positive (contains text line structure) by the Random Forest classifier, all the MSERs on this patch with a similarity above a threshold are considered a local line hypothesis.

Using local consistency maps and projection profiles, we capture the local text line structure on a patch. The projections may involve multiple text lines, because the random forest trains on various profile patterns and can handle different situations. We stress that the classification on the projection profiles need not to be highly accurate, i.e., false positives are expected, but the results are fine-tuned

by the correlation clustering.

After generating hypotheses in the entire image, there may exist two typical cases: 1) stand alone MSERs not covered by any local line hypotheses, and 2) disconnected subgraphs. The isolated MSERs are discarded, and disconnected graphs are processed separately (third row in Figure 2.5).

2.2.2.5 Correlation clustering based text detection

We briefly review the correlation clustering and introduce the higher-order clustering (HOCC). Compared to the basic correlation clustering, the key ingredient of the HOCC is the "hyperedges", which are sets of pre-defined weak hypotheses in a graph. We further provide the solution to the HOCC using semidefinite programming.

2.2.2.5.1 Correlation clustering

Given an undirected graph $G = (V, E)$, correlation clustering attempts to assign a binary label to each edge, indicating whether the two vertices are connected so they are in the same cluster. Practically, this binary label is relaxed and a rounding procedure is generally required to group the nodes.

2.2.2.5.2 Pairwise correlation clustering

Correlation clustering partitions nodes into clusters based on their pairwise similarities. Let $s_{ij}^p \in \{0, 1\}$ denote the pairwise similarity between node V_i and V_j (or on edge e_{ij}^p), and define x_{ij}^p as

$$x_{ij}^{p} = \begin{cases} 1, & \text{if } i \text{ and } j \text{ are connected} \\ 0, & o.w. \end{cases} \tag{2.7}$$

The correlation clustering problem becomes an agreement maximization formulation (Q1):

$$\begin{cases} \max\limits_{x^p} \sum\limits_{i,j} s_{ij}^{p} x_{ij}^{p} \\ s_{ij} = \mathrm{Sim}(V, i, j) = \langle \mathbf{w}^{p}, \phi_{ij}(V) \rangle \end{cases} \tag{2.8}$$

where $\phi_{ij}(V)$ denotes the features that characterize the difference between vertices i and j (Sec. 2.2.3.2), and $\mathbf{w}^p$ is the parameter vector (to be learned from training samples). Therefore, the correlation clustering becomes an integer programming problem.

In the literature, two solutions are proposed to solve Eq. 2.8. Kim et al. [42] used a number of inequalities, such as cyclic inequality constraints and odd-wheel constraints, to create physically meaningful polyhedrons. On the other hand [45], we can rewrite Eq. 2.8 to matrix form as follows

$$\begin{cases} \max\limits_{X} & trace(S^T X) \\ s.t. & X \succeq 0 \end{cases}. \tag{2.9}$$

where $S(i,j) = s_{i,j}$, and $X(i,j) = x_{i,j}$. This SDP formulation renders the problem elegant, but is limited by the number of variables in practice. We describe our solver in Sec. 2.2.2.6.

2.2.2.5.3 Higher-order correlation clustering (HOCC)

A hyperedge can be regarded as a subset of connected edges. Each hyperedge $e_k^h \in E$ contains more than one pairwise edge, i.e., $|e_k^h| \geq 2$. A hyperedge can be activated or deactivated, denoted by $x_k^h \in \{0, 1\}$, and is associated with a group-wise similarity $s_k^h = \left\langle \mathbf{w}^h, \phi^h(V)_k \right\rangle$, where $\phi_k^h(V)$ denotes the features for the kth hyperedge (Sec. 2.2.3.2). Then, the objective function becomes (Q2):

$$\begin{aligned} &\max_{x^p, x^h} \sum_{i,j} s_{ij}^p x_{ij}^p + \sum_k s_k^h x_k^h \\ =&\max_{x^p, x^h} \sum_{i,j} \left\langle \mathbf{w}^p, \phi_{i,j}^p(V) \right\rangle x_{i,j}^p + \sum_k \left\langle \mathbf{w}^h, \phi_k^h(V) \right\rangle x_k^h \\ =&\max_X \left\langle \mathbf{w}, \Phi(V, X) \right\rangle \end{aligned} \tag{2.10}$$

where $\mathbf{w}^h$ denotes the parameter vector for the hyperedge features, and $\mathbf{w}$ is concatenation of $\mathbf{w}^p$ and $\mathbf{w}^h$. $\mathbf{w}$ is learned from training data. X contains both pairwise edges and hyperedges. $\Phi(V, X)$ denotes the joint feature maps of all edges.

Binary operations are used to model the relation between higher order labels and pairwise labels. To be specific, a number of inequalities are used as follows.

$$\begin{aligned} x_k^h \leq& x_{ij}^p, \quad \forall e_{ij}^p \subset e_k^h, \\ x_k^h \geq& 1 - \sum_{i,j | e_{ij}^p \subset e^h} (1 - x_{ij}^p) \end{aligned} \tag{2.11}$$

The first inequality above ensures that the nodes in different clusters cannot be in the same activated hyperedge, and the second ensures that the nodes in the same cluster cannot have a deactivated hyperedge. Refer to [42] for more details.

2.2.2.6 Effective solution for "long-tailed" SDP

Semidefinite programming has attracted reasonable attention in recent years. Its applications range from kernel learning to low rank approximation [54]. While a few interior point based SDP packages are available (e.g., [55]), efficiency presents the major drawback when SDP is employed in real applications. Contrary to interior point based methods, boundary point based approaches, such as regularization methods, surface as alternatives. For example, the method in [44] is much easier to implement and can be tailored easily if special structures exist in SDP.

In a number of SDP problems, a "long tailed" block diagonal structure appears, being a small number of SDP variables but a large number of slack variables. Because the diagonal matrix is semipositive definite if and only if the values are nonnegative, this structure can be solved efficiently. In these cases, regularization methods become more efficient than interior point methods. This type of method is not widely known in the computer vision community, so we briefly describe one of these methods.

In Algorithm 1, $AX = b$ is a linear mapping of X, representing the constraints. In the literature, this may also be written as $A(X) = b$. The operation $(\cdot)_{+/-}$ denotes the projection to the positive/negative definite space. Refer to [44] for more details.

In our opinion, Algorithm 1 is an elegant solver that tackles many vision problems and has certain advantages: the implementation is straightforward (20+ lines in MATLAB), and it is an order of magnitude more efficient than other MATLAB SDP packages.

Algorithm 1 Regularization SDP.

Input:

t : Real positive scalar; S, Y : Symmetric matrix.

Z : Semipositive definite matrix.

A, b : Linear mapping, $AX = b$.

Output:

X : SDP solution for $\min trace(S^T X)$

subject to $AX = b, X \succeq 0$;

Procedure:

Repeat until $|Z + A^T y - S|$ is small

Step 1: Solve y for $AA^T y + A(Z - S) = (b - AY)/t$

Step 2: Set $X = t(Y/t + A^T y - S)_+$

Step 3: Set $Z = -(Y/t + A^T y - S)_-$

Step 4: Set $Y = X$.

To see how this formulation speeds up our problem, note that the time consuming step is the eigen-decomposition in the projection step in the internal problem (Step 2). In our formulation, since X has a long tailed structure, eigendecomposition needs to be performed only on the SDP variables. As a result, this solution performs efficiently for small problems ($\backsim$300 SDP variables), but may be slow for larger problems (e.g., more than 1,000 SDP variables).

2.2.2.7 Structural learning

Structured SVM [43] is used to learn the parameter vector $\mathbf{w}$. Consistent with previous notation, let $\{(V_n, X_n)\}$ dentoe N training samples, where V_n is the n^{th} training graph (with features), and X_n is its ground truth labels. Then, $\mathbf{w}$ is learned by:

$$\begin{aligned} &\min_{\mathbf{w},\xi} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{n=1}^{N}\xi_n \\ s.t.\ \ &\forall n, X \succeq 0, \\ &\langle \mathbf{w}, \Delta\Phi(V_n, X)\rangle \geq \Delta(X_n, X), \\ &\xi_n \geq 0 \end{aligned} \tag{2.12}$$

where $\Delta\Phi(V_n, X) = \Phi(V_n, X_n) - \Phi(V_n, X)$, and $C > 0$ is a constant. More details located in [42].

2.2.2.8 Classifying text and non-text regions

After applying HOCC, we obtain a number of image regions. Each region represents a consistent group of either text or non-text MSERs (Forth row in Figure 2.5). The matter of text/non-text will be decided on each region as a whole. We treat this step as a texture classification problem. Liu et al. [56] provide a state of the art approach based on random projections. In their approach, densely sampled image patches are projected with a random matrix and mapped to a dictionary of textons through the nearest neighbor to obtain a histogram. In our problem, we find random projections harm the classification accuracy. The reason could be that both text and non-text patterns have many variations, and random projections inevitably lose some information that may distinguish these two closely entangled classes.

Coates et al. [48] describe a method to detect text in a squared sliding window. In their method, whitened patches are encoded with an unsupervised dictionary using soft-thresholding and summed over 9 blocks to form a feature vector for this image window. Yet in our experiments, we found the naive nearest neighbor encoding outperformed soft-thresholding.

Our approach is similar in spirit as the two above. In our procedure, we separate the text and non-text patches when constructing the dictionary of textons. Through experiments we find the explicit separation of textons results in better classification performance compared to mixing text and non-text textons. Our procedure is as follows:

1. Pre-processing: Harvest 8×8 grayscale patches from text regions and non-text

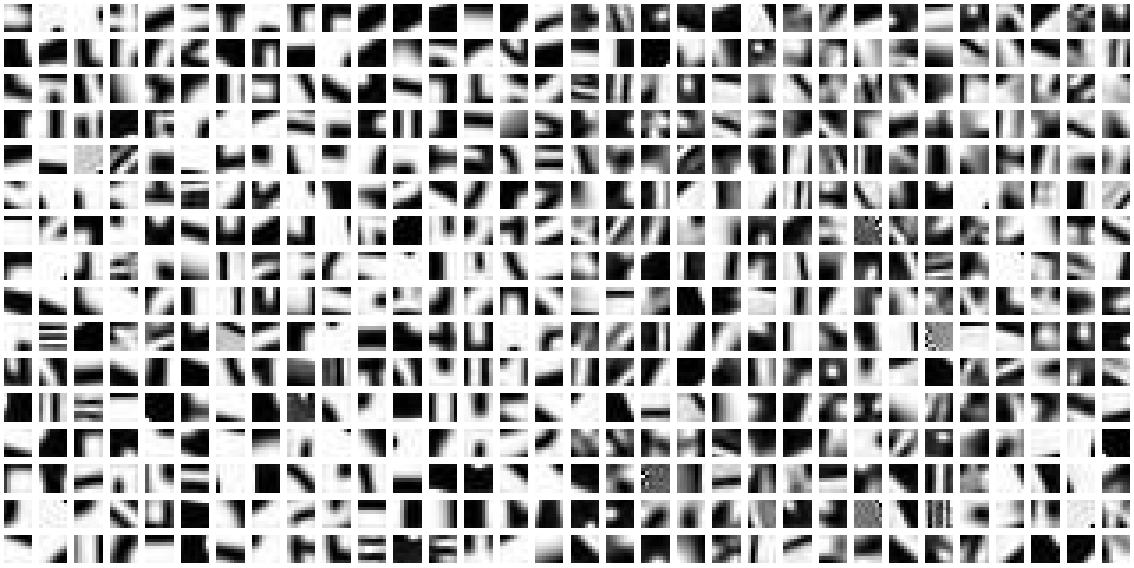


Figure 2.7: Samples of textons learned in the texture classification.

regions, and apply brightness and contrast normalization.

2. Constructing dictionary: Perform k-means on text patches and non-text patches, respectively, and combine the two sets of textons to form a dictionary.

3. Computing histogram: For each patch, find the nearest neighbor in the dictionary to form a histogram, and normalize the histograms to unit sum to produce the feature representation for an image region.

4. Classifying: Use Random Forest on the unit histograms.

Figure 2.7 shows samples of our textons. The textons corresponding to text and non-text patches share some common properties such as stroke-like patterns. However, a difference exists between statistical properties of text and non-text patches. Text and non-text patches may be encoded into different histograms, which enables the classification. As no character models or other specified features are involved, this

method is expected to generalize on text of different languages and styles.

2.2.3 Experiments

We demonstrate the effectiveness of our approach in this section. We aim to cluster similar text prior to any classification. Thus, it is natural for us to select the multi-orientation multi-scripts dataset such as the MSRA-TD500, instead of other datasets that primarily contain horizontal English text, which may be solved more effectively by strong character models and lexicons.

First, we describe two datasets used in our experiments. Then, we show the results of our method and compare them to the state of the art.

2.2.3.1 Datasets

We experiment on two datasets: the MSRA Text Detection Database (MSRA-TD500) and the OSTD dataset.

MSRA-TD500 contains 500 images of indoor and outdoor scenes. This dataset presents many challenges. First, the text is bilingual, including English, Chinese, and a mixture of the two, and they contain wide range of fonts, sizes and styles. Second, the text has arbitrary orientation. Third, the background is diversified and complex. To evaluate the performance on MSRA-TD500, we follow the protocol employed by [46]. We used 300 images for training and 200 for testing. A minimum rectangle is fit to the detected text region, and its orientation is also estimated. A ground truth rectangle can only be matched once, therefore many-to-one match is

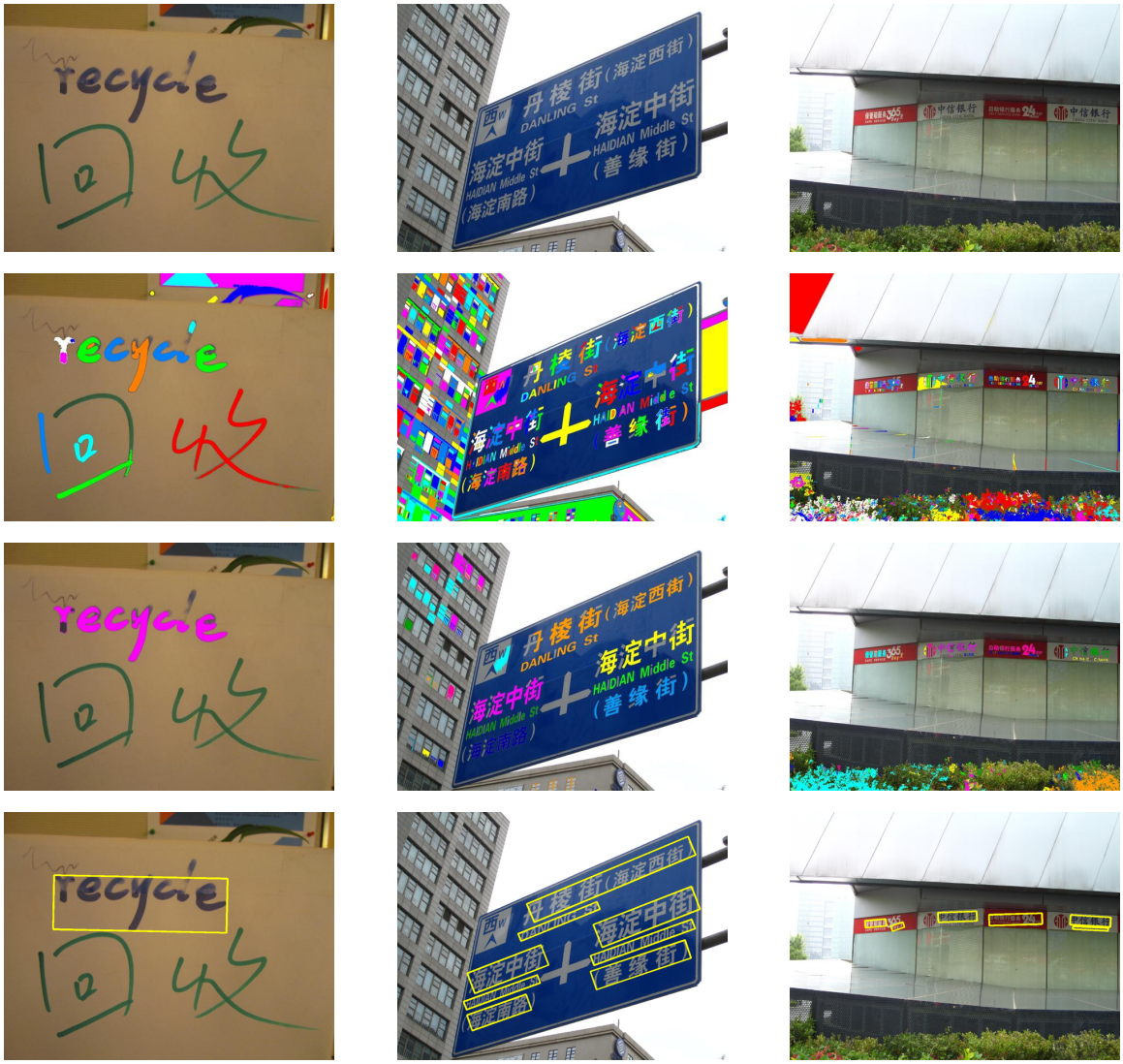
not allowed. We define the overlap ratio between a detected rectangle and a ground truth region as the ratio of the areas of their intersection and union. The rectangle is considered correct if the orientation difference is less than $\pi/8$ and the overlap ratio exceeds 0.5. We note that the 0.5 overlapping criteria differs from other text detection criteria, but is consistent with the PASCAL challenge for object detection.

The OSTD dataset contains 89 images of indoor and outdoor scenes (Figure 2.10). Text in this dataset contains diverse orientations, view perspectives, fonts, and styles. Following [46], the proposed algorithm trained on MSRA-TD500 runs on all images of OSTD. To make a fair comparison, we employ the same protocol as [57].

2.2.3.2 Features for clustering

The proposed method relies on the pairwise similarity and groupwise similarity. This section defines our features for correlation clustering in the text line detections. For pairwise features, we use a 18-dimensional features obtained in pairwise comparisons:

- Stroke width difference and ratio.
- Euclidean, χ^2, EMD [58], and L_1 distance of the RGB values.
- Euclidean, χ^2, EMD, and L_1 distance of the CIELAB values.
- Euclidean, χ^2, EMD, and L_1 distance of the contrast vectors (CIELAB values subtracting its immediate background).
- Solidity (shape area divided by convex hull area) difference.



(a) (b) (c)

Figure 2.8: More results in the MSRA dataset. From top to bottom, input image, MSER extraction, HOCC results and final detection results. Refer to Figure 2.5 for the meaning of the colors and bounding boxes.

- Area ratio difference
- Distance between two MSERs normalized by sizes and patch orientation.

For higher order features, we use a 12-dimensional features to describe group properties as follows:

- Variances in the RGB values.
- Variances in the CIELAB values.
- Variance in area normalized by the median.
- Variance in solidity.
- Variance in stroke width normalized by the median.

With the features defined, we can learn w^p and w^h for transforming the feature maps to the similarity measurement. We used the loss function in [42], but assign different weights due to the intrinsic properties of text images.

2.2.3.3 Results

We first present results for the MSRA-TD500 dataset (Figure 2.8). As shown in Figures 2.5 and 2.8, our method handles challenging cases. For example, Figure 2.5 (a) has text of various fonts and mixed languages. Text in Figure 2.5 (d) is tiny and the background (trees) renders some text difficult to detect (e.g., the road name sign). Further, the bike pattern on the traffic sign contains a similar and consistent stroke width as the text. Text in Figure 2.5 (e) is overly slanted. We also notice from 2.8 (a) that some handwritten text is correctly detected, even though their shapes and colors are less consistent compared to the printed text.

Table 2.3 shows the quantitative results. Among three methods we compared, Yao et al [46] handles oriented text explicitly, and [50] is the baseline, where the

stroke width transform was proposed. The proposed algorithm can detect text lines in different fonts and orientations. In complex backgrounds like leaves, grass, and some extremely challenging architectural patterns, we observe promising detection results. Our method achieves a higher precision with similar recall rate, compared to the state of the art method.

The HOCC plays an important role in this process, since it generates larger homogeneous regions to provide robust statistics for the discrimination between text and non-text. Avoiding local decisions is essential to the proposed approach. Without an effective grouping process, such a simple texton-based text detection algorithm has difficulty achieving superior performance. For example, [48] reports less satisfying detection rates compared to other methods using highly specialized features.

The errors mainly stem from three sources. First, some MSERs are not well extracted due to lighting, fragmentation, and blur (Figure 2.9 (a)), in the graph construction stage. Second, some text lines are too close and merged into one rectangle during clustering, due to the difficulty of resolving their local linear structures (Figure 2.9 (b)). On the other hand, text lines are broken into multiple parts when relatively large gaps exist between characters or words due to the elimination of some MSERs (Figure 2.9 (c)). Third, mistakes exist in texture classification, where artificial and plant textures may be classified as text (Figure 2.9 (d)).

We also test the proposed algorithm on the Oriented Scene Text Dataset [57]. Figure 2.10 shows some results from the OSTD dataset. We achieved the best precision rate and F measure, and our recall is comparable to the state of the art.

(a) (b)

(c) (d)

Figure 2.9: Error analysis. a) MSERs are not well extracted due to lighting; b) text lines are too close and merged; c) text lines are broken into multiple parts; d) mistakes exist in texture classification.

We observe that the text is also correctly grouped and detected, even though the properties of these text differ greatly from the MSRA-TD500 dataset.

Table 2.3: Performance comparison on MSRA-TD500.

	Precision	Recall	F-measure
Our method	0.71	0.62	0.66
Yao et al. [46]	0.63	0.63	0.60
Epstein et al. [50]	0.25	0.25	0.25
Chen et al. [59]	0.05	0.05	0.05

Table 2.4: Performance comparison on OSTD.

	Precision	Recall	F-measure
Our method	0.80	0.73	0.76
Yao et al. [46]	0.77	0.73	0.74
Yi et al. [57]	0.56	0.64	0.55
Epstein et al. [50]	0.37	0.32	0.32
Chen et al. [59]	0.07	0.06	0.06

Figure 2.10: Examples in the OSTD dataset. Detection results are shown in yellow bounding box that overlay the input images.

Chapter 3: Document Image Categorization

Classifying and grouping document images into known categories is often a prerequisite step toward document understanding tasks, such as text recognition, document retrieval, and information extraction [6]. These tasks can be greatly simplified if we know a priori the genre or the layout-type of documents. In the past, document image classification and retrieval has been researched under a number of paradigms. Two major paradigms have been extensively studied: text-content based approaches and document structure based approaches. We focus on the second paradigm, studying document structure based classification.

Previous approaches for document structure based classification have focused on finding effective visual representations. Existing approaches in the literatures mainly in their choices of local features, global representations, and learning mechanisms [7]. Various structure or layout-based features have been introduced [8–12] and are shown to be effective for document image classification and retrieval. These approaches, however, are limited to a particular class of documents such as bank forms, memos, contracts and orders. To apply existing classification systems to

other types of documents, we need to reconsider spatial features and tune it manually. Moreover, when the content and structure in documents are unconstrained, as in handwritten documents, pre-defined features may not be able to capture all variations of a particular class.

A more general approach that automatically learns different abstractions of structure hierarchy and spatial relationship among document elements is needed. Document images usually have a hierarchical structure, such as cells in rows and columns of tables, words in sentences, and sentences in paragraphs. These hierarchical patterns are often repeated in different parts of document. These properties imply the possibility of learning the layout as a combination of small groups of middle or lower level features.

We present a general approach for document image classification using CNNs. CNN is a kind of neural network that shares weights among neurons in the same layer. CNNs work well at discovering spatially local correlation by enforcing a local connectivity pattern between neurons of adjacent layers [13]. With multiple layers and pooling between layers, CNNs automatically learn the hierarchical layout features with tolerance to spatial translation, and capture repeating patterns efficiently by sharing weights.

For the task of document image classification, our CNN uses a new type of neuron, Rectified Linear Units (RLU) [60], to speed up training. We employ dropout [61] to prevent overfitting. Experiments on real-world unconstrained datasets show that our approach is more effective than previous approaches.

3.1 Related work

Byun and Lee [8] used a partial matching method in which document structure recognition and classification is applied to only part of input form images. The application of their approach is limited to *form* images and does not generalize to other types of documents. Shin and Doermann [62] defined *visual similarity* of layout structures and applied supervised classification for each specific type. They used image features, such as the percentage of text and non-text (graphics, images, tables, and rulings) in content regions, column structures, relative point sizes of fonts, density of content area, and statistics of features of connected components. For classification, they employed decision trees and self-organizing maps. Like previous approaches, the main drawback of their method stems from the features, which were designed for specific document classes (e.g., forms, letters, and articles). Additionally, given a large number of different feature types the approach is computationally slow for large scale document exploration.

Collins-Thompson and Nickolov [63] proposed a model for estimating the inter-page similarity in ordered collections of document images. They used a combination of features from text and layout, document structure, and topic concepts, to discriminate between related and unrelated pages. Text from OCR may contain errors, especially for handwritten documents, thus their approach is limited to well-structured printed documents. Joutel et al. [64] presented an approach for the retrieval of handwritten historical documents at page level based on the *curvelet transform* to compose a unique signature for each page. The approach is effective

when local shapes are important for classification, but it is likely to miss any higher level of structural saliency. In many cases, the desired similarity is embedded in global structure and relationships among different objects in document images.

Kochi and Saitoh [65] proposed a system for identifying the type of a semi-formatted document based on important textual elements extraction and by using a flexible matching strategy for easy model generation. Bagdanov and Worring [9] approached the general problem of genre classification of printed document images using attributed relational graphs (ARGs). They used ARGs to represent the layout structure of document instances, and first order random graphs (FORGs) to represent document genres. They reported a high-accuracy on a small dataset of 130 documents consisting of 10 genres. Reddy et al. [66] addressed the form classification problem with a classifier based on the k-means algorithm. They used low-level pixel density features and adaptive boosting to classify NIST tax forms. A detailed survey on document classification based on three components, the problem statement, the classifier architecture, and the performance evaluation, can be found in Chen and Blostein [7].

Approaches based on bag-of-words (BOW) models have shown promising results on many computer vision problems, such as image classification [67], scene understanding [68], and document image classification [69, 70]. However, initial formulations typically disregard the spatial relationships of different image regions, and consider only the occurrences of visual patterns in an image. This results in a limited descriptive capability and the performance may drop significantly in presence of noise, background clutter, variation of layout and content in images. Kumar

et al. [10, 12] extended the spatial-pyramid features for document images by using a novel pooling method with horizontal-vertical partitions that are adapted to the typical layout of document images. Subsequently, methods that extend the BOW approach to incorporate spatial relationships between image regions were proposed. One of the early methods proposed the creation of spatial-pyramid features by partitioning the image into increasingly finer grids and computing the weighted histogram based kernel in each region [71]. Later, there has been a focus on selecting the optimal feature combination strategy and efficient ways to learn these local statistics, and a number of methods have been proposed [10, 12, 72, 73].

3.2 Approach

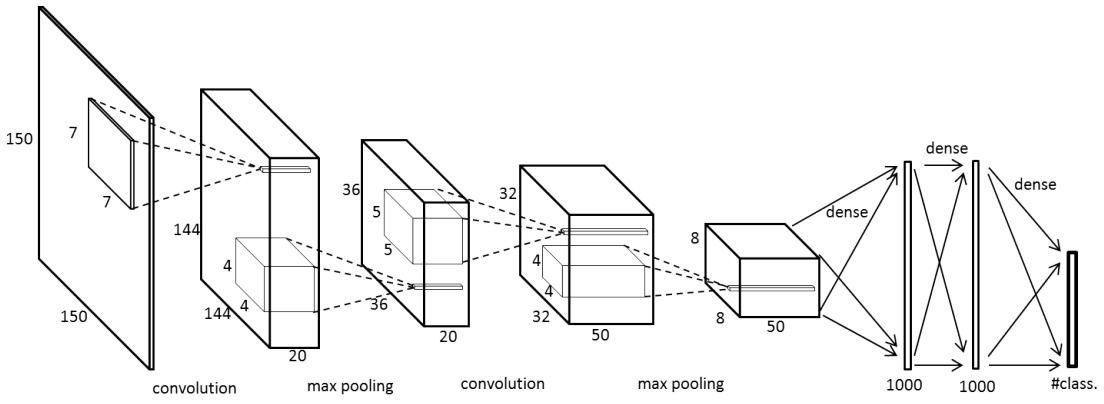


Figure 3.1: The architecture of the proposed CNN

We propose to use a CNN for document image classification. We look to learn a hierarchy of feature detectors and train a nonlinear classifier to identify complex document layouts. Given a document image, we first perform downsampling and pixel value normalization, then feed the normalized image to the CNN to predict

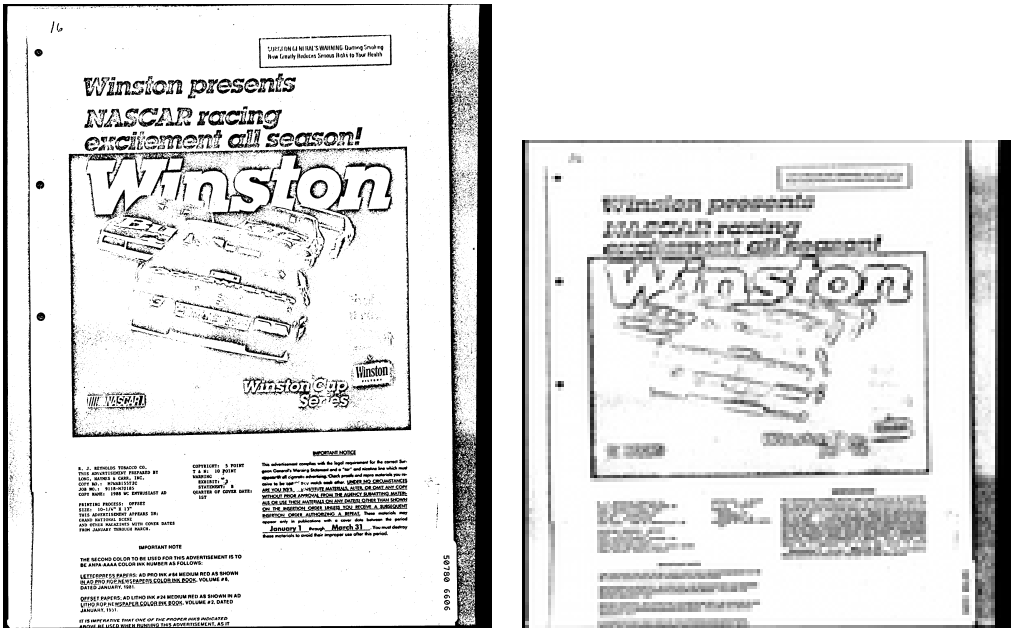
the class label.

3.2.1 Preprocessing

The resolution of document images is typically higher than 2000×2000, which is too large to be fed to a CNN given the current availability of computing resources. The large input dimension not only costs more computationally, but leads to a greater chance of overfitting. Considering that layout, instead of the details such as characters, determines the class of document images, we can reduce the input dimension by discarding details of document images, as long as the structure information is still identifiable. Specifically, document images of various sizes are all downsampled and resized to 150×150 with bilinear interpolation. At the resolution of 150×150, most characters on the document images are not recognizable but the overall layout is preserved and the locations of title, text, or table can be determined. Humans can make the same predictions on the document types as at original resolution if judging by layout only. Figure 3.2 shows the downsampled document images compared to original resolution. After downsampling, the gray scale images are divided by 255, then subtracted by 0.5, therefore normalized to the range of $[-0.5, 0.5]$.

3.2.2 Network Architecture

Figure 3.1 shows the architecture of our network, which can be summarized as $150 \times 150 - 36 \times 36 \times 20 - 8 \times 8 \times 50 - 1000 - 1000 - M$, where M is the number of



(a) (b)

Figure 3.2: (a) Original image of resolution 2544×3256 (b) Downsampled and resized to 150×150. Enlarge to see the difference in details.

classes. The input is a downsampled and normalized image of size 150×150. The first convolutional layer consists of 20 kernels, each of size 7×7, followed by a 4×4 pooling that reduces the each feature map to 36×36. The second convolutional layer contains 50 kernels each of size 5, which means each kernel is convolved with all 20 feature maps of previous layer. A 4×4 pooling comes after the second convolutional layer to produce 50 feature maps, each of size 8×8. Two fully connected layers of 1000 nodes each follow the convolution and pooling layers. The last layer is a logistic regression with softmax that outputs the probability on each class, as defined in the following equation:

$$P(y = i|x, W_1, ..., W_M, b_1, ..., b_M) = \frac{e^{W_i x + b_i}}{\sum_{j=1}^{M} e^{W_j x + b_j}} \tag{3.1}$$

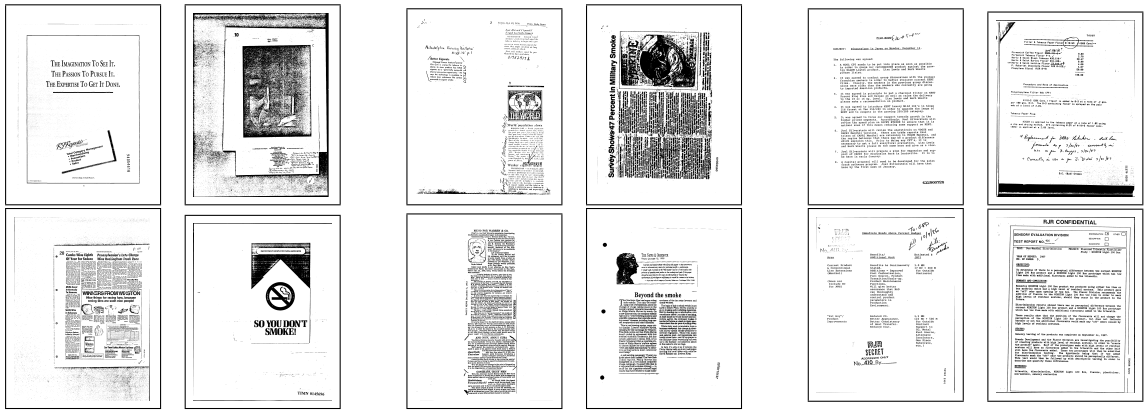
where x is the output of the second fully connected layer, W_i and b_i are the weights and biases of i^{th} neuron in this layer, and M is the number of classes. The class that outputs the max probability is taken as the predicted class, which can be described in the following equation ($\hat{y}$ denotes the predicted class):

$$\hat{y} = \arg\max_i P(y = i|x, W_1, ..., W_M, b_1, ..., b_M) \tag{3.2}$$

Instead of traditional sigmoid or tanh neurons, we use Rectified Linear Units (ReLUs) [60] in the convolutional and fully connected layers. Recent research [26] has demonstrated ReLUs bring several times speedup in training compared to using tanh units. Formally, an ReLU has an output of $f(x) = max(0, x)$, where x denotes the input. In experiments, we observe that ReLUs enable the training to complete several times faster than sigmoid neurons, and it is not so sensitive to the scale of input.

3.2.3 Training

We adopt negative log-likelihood as the loss function and perform Stochastic Gradient Descent (SGD). Recently successful neural network methods report that dropout [26, 61] improves learning. Dropout alleviates overfitting by introducing random noise to training samples. During training time, the neuron outputs of



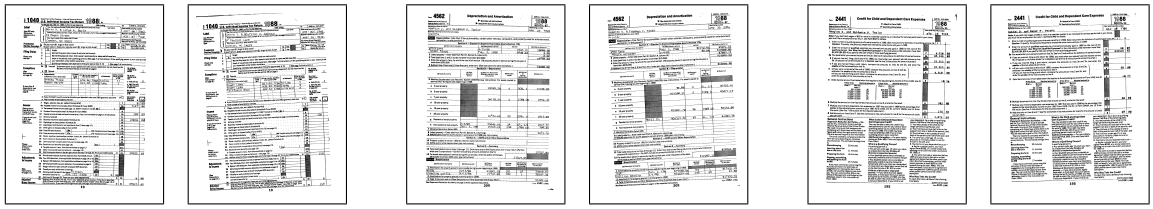
(a) *ad* (b) *news* (c) *report*

Figure 3.3: Sample images from Tobacco dataset, grouped in three genres/classes (a) *ad*, (b) *news*, and (c) *report*.

hidden layers are masked with probability of 0.5, and at test time their outputs are halved. In our experiment, we also find dropout boosts the performance for a large network. Applying dropout to all layers significantly increases the training time to reach convergence, so we only apply dropout at the second fully connected layer, i.e., half of the outputs of the second fully connected layer are randomly masked in training, and in testing the weights of the logistic regression layer are divided by 2, which is equivalent to halving the outputs of the second fully connected layer.

3.3 Experiments

We conduct experiments on two datasets to demonstrate the effectiveness of our CNN.



(a) *Form1040-1* (b) *Form4562-1* (c) *Form2441*

Figure 3.4: Sample images from from NIST tax-form dataset, grouped in three classes (a) *Form1040-1*, (b) *Form4562-1*, and (c) *Form2441*.

3.3.1 Datasets

The following two datasets were used in our experiments.

(1) Tobacco litigation dataset [74]: we used 3,482 images categorized in 10 genres(classes): *report, memo, resume, scientific, letter, news, note, ad, form,* and *email*. Figure 3.3 shows some samples of Tabacco dataset. From Figure 3.3 we observe that large inner-class variation exists, especially for the class *ad*.

(2) NIST tax-form dataset [75]: a collection of 5,590 tax-form images from National Institute of Standards and Technology, categorized into 20 classes, with labels like *Form1040-1, Form1040-2, Form4562-1, Form2441,* and so on. Figure 3.4 shows samples of NIST tax-forms.

3.3.2 Results

We mainly compare our method to previous methods SP-RF and HVP-RF [12], so we follow the same evaluation protocol. We apply the proposed CNN with the same architecture to the two datasets described above.

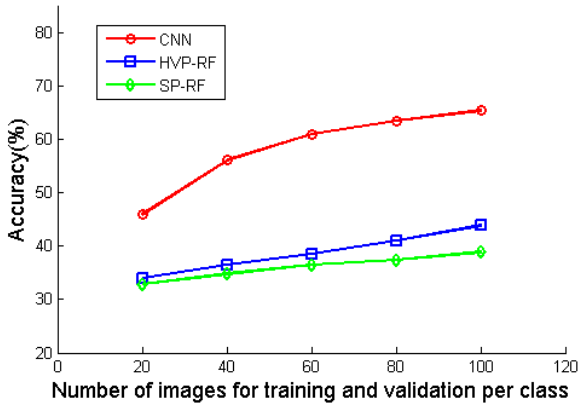


Figure 3.5: Genre classification results on Tabacco dataset (3482 images, 10 classes)

For 10 classes of images in the Tobacco dataset, we randomly select $N (N <= 100)$ images per class for training and validation, among which 80% are for training and 20% for validation, and the rest images are used for test. We vary N to determine the performance under different amount of training and validation samples. The accuracies of the proposed algorithm are obtained on 100 such random partitions of training, validation, and test, and the median accuracy is shown in Figure 3.5. The proposed approach achieves a median accuracy of 65.37% when 100 samples are used for training and validation. Our CNN consistently outperforms previous top methods SP-RF and HVP-RF [12]. A class-confusion matrix on one of the partitions is found in Table 3.1.

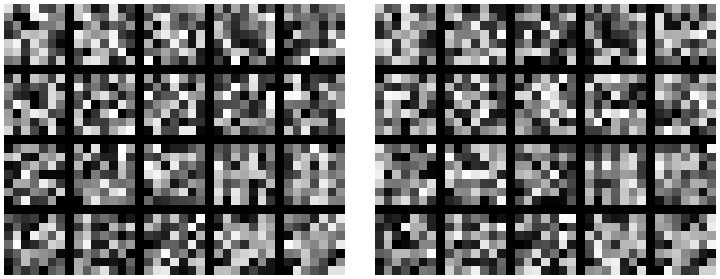
On the 20-class NIST tax-form dataset, we randomly choose one image per class (which amounts to 20 samples in total) for training, and use the rest for test. No validation set is used, because we simply use the parameters after 50 epochs of training. A median accuracy of of 100% is achieved through 100 partitions

	ad	email	form	letter	memo	news	note	report	resume	scientific
ad	**104**	0	1	1	0	9	2	2	0	3
email	1	**435**	7	3	13	0	4	3	1	0
form	2	0	**145**	5	37	7	8	7	0	14
letter	0	8	6	**297**	43	0	1	14	0	10
memo	1	7	33	51	**294**	6	3	9	0	18
news	19	1	21	13	6	**45**	8	2	0	16
note	2	10	24	8	31	5	**63**	0	0	11
report	1	15	34	65	32	11	5	**103**	5	38
resume	0	7	24	13	12	1	1	13	**13**	6
scientific	0	16	36	11	52	4	6	12	1	**45**
Accuracy (%)	80.0	87.2	43.8	63.6	56.6	51.1	62.4	62.4	65.0	28.0

Table 3.1: Class-confusion matrix for genre classification on Tobacco dataset. These are the results of one partition of training-validation-test, which produces an overall accuracy of 65.35%

of training and test, which ties with [12]. Other methods such as [76] achieved similar accuracies, but they used more training samples. We think that the proposed method achieves such high accuracies with so few training samples because the tax-form images in the same class show highly consistent layouts, and inter-class similarity is relatively low.

We visualize the kernels of the first convolutional layer learned on the Tobacco and the NIST, respectively, as shown in Figure 4.17. We do not observe obvious patterns that resemble the local structure of document images.



(a) (b)

Figure 3.6: Learned kernels in the first convolutional layer from (a) Tobacco dataset (b) NIST dataset.

3.3.3 Computational Cost

We implemented the CNN using Theano [77], which enables easy deployment on a GPU to speed up the process without much manual optimization. Our experiments were performed on a PC with 2.8GHz CPU and Tesla C1060 GPU. On average each image takes about 0.004 second processing time.

Chapter 4: Image Quality Assessment

4.1 No-Reference Image Quality Assessment using CNN

We present a CNN that can accurately predict the quality of distorted images with respect to human perception. The work focuses on the most challenging category of objective image quality assessment (IQA) tasks: general-purpose No-Reference IQA (NR-IQA), which evaluates the visual quality of digital images without access to reference images and without prior knowledge of the types of distortions present.

Visual quality is a complex yet inherent characteristic of an image. In principle, it represents the measure of the distortion compared with an ideal imaging model or perfect reference image. When reference images are available, Full Reference (FR) IQA methods [14–18] can be applied to quantify the differences directly between distorted images and their corresponding ideal versions. State of the art FR measures, such as VIF [14] and FSIM [15], achieve a high correlation with human perception.

However, in many practical computer vision applications perfect versions of the distorted images do not exist, thus NR-IQA is required. NR-IQA measures can

directly quantify image degradations by exploiting features that are discriminant for image degradations. Most successful approaches use Natural Scene Statistics (NSS) based features. Typically, NSS based features characterize the distributions of certain filter responses. Traditional NSS based features are extracted in image transformation domains using, for example, the wavelet transform [19] or the DCT transform [20]. These methods usually operate slowly due to the use of computationally expensive image transformations. Recent development in NR-IQA methods – CORNIA [21, 22] and BRISQUE [23] promote extracting features from the spatial domain, which leads to a significant reduction in computation time. CORNIA demonstrates the possibility of learning discriminant image features directly from the raw image pixels, instead of using handcrafted features.

Based on these observations, we explore using a Convolutional Neural Network (CNN) to learn discriminant features for the NR-IQA task. Recently, deep neural networks have gained researchers' attention and achieved great success on various computer vision tasks. Specifically, CNN has shown superior performance on many standard object recognition benchmarks [24–26]. One of CNN's advantages is that it can take raw images as input and incorporate feature learning into the training process. With a deep structure, the CNN can effectively learn complicated mappings while requiring minimal domain knowledge.

To the best of our knowledge, CNN has not been applied to general-purpose NR-IQA, mostly because the original CNN is not designed for capturing image quality features. In the object recognition domain good features generally encode local invariant parts, however, for the NR-IQA task, good features should be able

to capture NSS properties. The difference between NR-IQA and object recognition renders the application of CNN nonintuitive. One of our contributions is that we modify the network structure, so it can learn image quality features more effectively and estimate the image quality more accurately.

In another contribution, we propose a novel framework that allows learning and prediction of image quality on local regions. Previous approaches typically accumulate features over the entire image to obtain statistics for estimating overall quality, and have rarely shown the ability to estimate local quality, except for a simple example in [78]. By contrast, our method can estimate quality on small patchs (such as 32×32). Local quality estimation is important for image denoising or reconstruction problems, applying enhancement only where required.

We show experimentally that the proposed method advances the state of the art. Our CNN outperforms CORNIA and BRISQUE on the LIVE dataset, and it achieves comparable results with state of the art FR measures such as FSIM [15]. In addition to the superior overall performance, we also show qualitative results that demonstrate the local quality estimation of our method.

4.1.1 Related Work

Previously, researchers have attempted to use neural networks for NR-IQA. Li et al. [79] applied a general regression neural network that takes as input perceptual features, including phase congruency, entropy, and the image gradients. Chetouani et al. [80] used a neural network to combine multiple distortion-specific NR-IQA

measures. These methods require pre-extracted handcrafted features and only use neural networks for learning the regression function. Thus, they do not have the advantage of learning features and regression models in a holistic way, and these approaches are inferior to the state of the art approaches. In contrast, our method does not require any handcrafted features and directly learns discriminant features from normalized raw image pixels to achieve much better performance.

The use of convolutional neural networks is partly motivated by the feature learning framework introduced in CORNIA [21, 22]. First, the CORNIA features are learned directly from the normalized raw image patches. This implies that it is possible to extract discriminative features from spatial domain without complicated image transformations. Second, supervised CORNIA [22] employs a two-layer structure that learns the filters and weights in the regression model simultaneously based on an EM like approach. This structure can be viewed as an empirical implementation of a two layer neural network. However, it has not utilized the full power of neural networks.

Our approach integrates feature learning and regression into the general CNN framework. The advantages are two-fold. First, making the network deeper will raise the learning capacity significantly [81]. In the following sections we will see that with fewer filters/features than CORNIA, we achieve state of the art results. Second, in the CNN framework, training the network as a whole using a simple method like backpropagation enables the possibility of conveniently incorporating recent techniques designed to improve learning, such as dropout [61] and rectified linear unit [26]. Furthermore, after we make the bridge between NR-IQA and CNN,

the rapid developing deep learning community will offer a significant source of novel techniques for advancing the NR-IQA performance.

4.1.2 Approach

The proposed framework of using CNN for image quality estimation is as follows. Given a gray scale image, we first perform a contrast normalization, then sample non-overlapping patches from it. We use a CNN to estimate the quality score for each patch and average the patch scores to obtain a quality estimation for the image.

4.1.2.1 Network Architecture

The proposed network consists of five layers. Figure 4.1 shows the architecture of our network, which is a $32\times32-26\times26\times50-2\times50-800-800-1$ structure. The input is locally normalized 32×32 image patches. The first layer is a convolutional layer, which filters the input with 50 kernels each of size 7×7 with a stride of 1 pixel. The convolutional layer produces 50 feature maps each of size 26×26, followed by a pooling operation that reduces each feature map to one max and one min. Two fully connected layers of 800 nodes each come after the pooling. The last layer is a simple linear regression with a one dimensional output that gives the score.

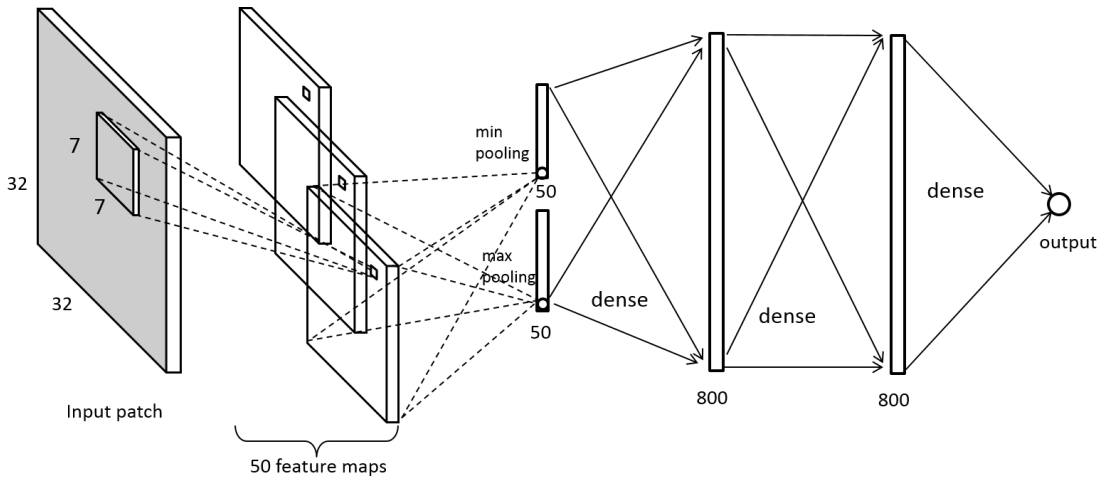


Figure 4.1: The architecture of our CNN

4.1.2.2 Local Normalization

Previous NR-IQA methods, such as BRISQUE and CORNIA, typically apply a contrast normalization. In this section, we employ a simple local contrast normalization method similar to [23]. Suppose the intensity value of a pixel at location (i, j) is $I(i, j)$, we compute its normalized value $\hat{I}(i, j)$ as follows:

$$\begin{aligned} \hat{I}(i,j) &= \frac{I(i,j) - \mu(i,j)}{\sigma(i,j) + C} \\ \mu(i,j) &= \frac{1}{(2P+1)(2Q+1)} \sum_{p=-P}^{p=P} \sum_{q=-Q}^{q=Q} I(i+p, j+q) \\ \sigma(i,j) &= \sqrt{\frac{1}{(2P+1)(2Q+1)} \sum_{p=-P}^{p=P} \sum_{q=-Q}^{q=Q} \big(I(i+p, j+q) - \mu(i,j)\big)^2} \end{aligned} \tag{4.1}$$

where C is a positive constant that prevents dividing by zero. P and Q are the normalization window sizes. In [23], it was shown that a smaller normalization window size improves the performance. In practice we pick $P = Q = 3$ so the window size is much smaller than the input image patch. Note that with this local

normalization each pixel may have a different local mean and variance.

Local normalization is important. We observe that using larger normalization windows leads to worse performance. Specifically, a uniform normalization, which applies the mean and variance of the entire image patch to each pixel, will cause about a 3% reduction in the performance.

It is worth noting that when using a CNN for object recognition, a global contrast normalization is usually applied to the entire image. The normalization not only alleviates the saturation problem common in early work that used sigmoid neurons, but also makes the network robust to illumination and contrast variation. For the NR-IQA problem, contrast normalization should be applied locally. Additionally, although luminance and contrast change can be considered distortions in some applications, we focus mainly on distortions arising from image degradations, such as blur, compression, and additive noise.

4.1.2.3 Pooling

In the convolution layer, the locally normalized image patches are convolved with 50 kernels, and each kernel generates a feature map. We then apply pooling on each feature map to reduce the filter responses to a lower dimension. Specifically, each feature map is pooled into *one* max value and *one* min value, which is similar to CORNIA. Let $R^k_{i,j}$ denote the response at location (i, j) of the feature map obtained

by the k-th filter, then the max and min values of u_k and v_k are given by

$$\begin{aligned} u_k &= \max_{i,j} R^k_{i,j} \\ v_k &= \min_{i,j} R^k_{i,j} \end{aligned} \tag{4.2}$$

where $k = 1, 2, ..., K$ and, K is the number of kernels. The pooling procedure reduces each feature map to only two scalars. Therefore, each node of the next fully connected layer takes an input of size $2 \times K$. It is worth noting that although max pooling already works well, introducing min pooling boosts the performance by about 2%.

In object recognition scenario, pooling is typically performed on every 2×2 cell. In that case, selecting a representative filter response from each small cell may keep some location information while achieving robustness to translation. This operation is particularly helpful for object recognition, as objects can typically be modeled as multiple parts organized in a certain spatial order. However, for the NR-IQA task, we observe that image distortions are often locally (if not globally) homogeneous, i.e., the same level of distortion occurs at all the locations of a 32×32 patch. The lack of obvious global spatial structure in image distortions enables pooling without retaining locations to reduce the cost of computation.

4.1.2.4 ReLU Nonlinearity

Instead of traditional sigmoid or tanh neurons, we use Rectified Linear Units (ReLUs) [60] in the two fully connected layers. Krizhevsky et al. [26] demonstrated that in a deep CNN ReLUs enable the network to train several times faster compared

to using tanh units. Here we give a brief description of ReLUs. ReLUs take a simple form of nonlinearity by applying a thresholding function to the input, in place of the sigmoid or tanh transform. Let g, w_i, and a_i denote the output of the ReLU, the weights of the ReLU and the output of the previous layer, respectively, then the ReLU can be mathematically described as $g = \max(0, \sum_i w_i a_i)$.

Note that ReLUs allow only nonnegative signals to pass through. Due to this property, we do not use ReLUs but use linear neurons (identity transform) on the convolutional and pooling layer. Min pooling typically produce negative values and we do not want to block the information in these negative pooling outputs.

4.1.2.5 Learning

We train our network on non-overlapping 32×32 patches taken from large images. For training, we assign each patch a quality score as its source image's ground truth score. We can do this because the training images in our experiments have homogeneous distortions. During the test stage, we average the predicted patch scores for each image to obtain the image level quality score. By taking small patches as input, we have a much larger number of training samples compared to using the whole image on a given dataset, which meets the particular needs of CNNs.

Let x_n and y_n denote the input patch and its ground truth score, respectively, and $f(x_n; w)$ be the predicted score of x_n with network weights w. Support Vector Regression (SVR) with ϵ-insensitive loss has been successfully applied to learn the regression function for NR-IQA in previous work [22, 23]. We adopt a similar

objective function as follows:

$$L = \frac{1}{N}\sum_{n=1}^{N} \|f(x_n; w) - y_n\|_{l_1}$$
$$w' = \min_{w} L \tag{4.3}$$

Note that the above loss function is equivalent to the loss function used in ϵ-SVR with $\epsilon = 0$. Stochastic gradient decent (SGD) and backpropagation are used to solve this problem. A validation set is used to select parameters of the trained model and prevent overfitting. In experiments, we perform SGD for 40 epochs in training and retain the model parameters that generate the highest Linear Correlation Coefficient (LCC) on the validation set.

Recently successful neural network methods [26, 61] report that dropout and momentum improve learning. We also find these two techniques boost the performance in our experiment.

Dropout is a technique that prevents overfitting in training neural networks. The outputs of neurons are typically set to zero with a probability of 0.5 in the training stage, and they are divided by 2 in the test stage. By randomly masking the neurons, dropout offers an efficient approximation of training many different networks with shared weights. In our experiments, since applying dropout to all layers significantly increases the time to reach convergence, we apply dropout only at the second fully connected layer.

Updating the network weights with momentum is a widely adopted strategy. We update the weights in the following form:

$$
\begin{aligned}
\Delta w^t &= r^t \Delta w^{t-1} - (1 - r^t)\epsilon^t \langle \nabla_w L \rangle \\
w^t &= w^{t-1} + \Delta w^t \\
\epsilon^t &= \epsilon_0 (d)^t \\
r^t &= \begin{cases} \frac{t}{T} r_e + (1 - \frac{t}{T}) r_s, & t < T \\ r_e, & t \geqslant T \end{cases}
\end{aligned}
\tag{4.4}
$$

where w^t is weight at epoch t, $\epsilon_0 = 0.1$ is learning rate, $d = 0.9$ is decay for the learning rate, $r_s = 0.9$ and $r_e = 0.5$ are starting and ending momentums, respectively. $T = 10$ is a threshold to control how the momentum changes with the number of epochs. Note that unlike [61], where momentum starts off at a value of 0.5 and stays at 0.99, we begin with a large momentum and reduce it as the training progresses. We found through experiments that this setting can achieve better performance.

4.1.3 Experiments

4.1.3.1 Experimental Protocol

Datasets: We conduct experiments on the following two datasets.

(1) LIVE [82]: A total of 779 distorted images with five different distortions – JPEG2000 compression (JP2K), JPEG compression (JPEG), white Gaussian (WN),

Gaussian blur (BLUR), and fast fading (FF) at 7-8 degradation levels derived from 29 reference images. Differential Mean Opinion Scores (DMOS) are provided for each image, roughly in the range $[0, 100]$. Higher DMOS indicates lower quality.

(2) TID2008 [83]: 1,700 distorted images with 17 different distortions derived from 25 reference images at 4 degradation levels. In our experiments, we consider only the four common distortions that are shared by the LIVE dataset, i.e., JP2K, JPEG, WN, and BLUR. Each image is associated with a Mean Opinion Score (MOS) in the range $[0, 9]$. Contrary to DMOS, higher MOS indicates higher quality.

Evaluation: Two measures are used to evaluate the performance of IQA algorithms: 1) Linear Correlation Coefficient (LCC) and 2) Spearman Rank Order Correlation Coefficient (SROCC). LCC measures the linear dependence between two quantities, and SROCC measures how well one quantity can be described as a monotonic function of another quantity. We report results obtained from 100 train-test iterations, where, in each iteration, we randomly select 60% of reference images and their distorted versions as the training set, 20% as the validation set, and the remaining 20% as the test set.

4.1.3.2 Evaluation on LIVE

On the LIVE dataset, for distortion-specific experiment we train and test on each of the five distortions: JP2K, JPEG, WN, BLUR, and FF. For non-distortion-specific experiments, images of all five distortions are trained and tested together without providing a distortion type.

Table 4.6 shows the results of the two experiments compared with previous state of the art NR-IQA methods, as well as FR-IQA methods. Results of the best performing NR-IQA systems appear in bold type. The FR-IQA measures are evaluated by using 80% of the data for fitting a non-linear logistic function, then testing on 20% of the data. We observe from Table 4.6 that our approach works well on each of the five distortions, especially on JPEG, JP2K, and FF. For the overall evaluation, our CNN outperformed all previous state of the art NR-IQA methods and approached the state of the art FR-IQA method FSIM.

We visually examine the learned convolution kernels, and find only a few kernels present obvious structures related to the type of distortion. Figure 4.17 shows the kernels learned on JPEG and all distortions combined, respectively. We can see that some blockiness patterns are learned from JPEG, and a few blur-like patterns exist for kernels learned from all distortions. But generally the kernels learned by our model do not present obvious structure compared to CORNIA [21] or CNNs trained for object recognition, probably because our model takes input as the locally normalized image patches, which resemble edge maps and own different characteristics.

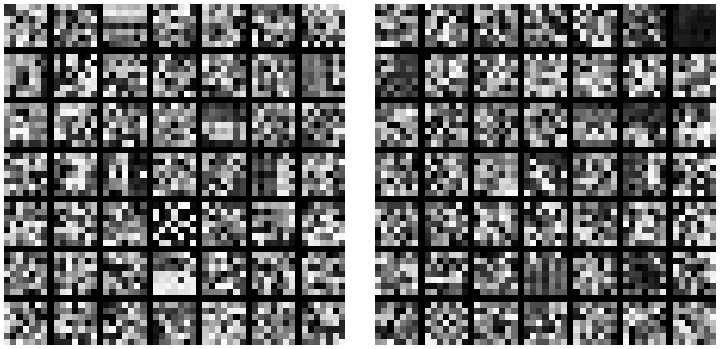
4.1.3.3 Effects of Parameters

Several parameters are involved in the CNN design. In this section, we examine how these parameters affect the performance of the network on the LIVE dataset.

Number of kernels Figure 4.3 shows how the performance varies with the

SROCC	JP2K	JPEG	WN	BLUR	FF	ALL
PSNR	0.870	0.885	0.942	0.763	0.874	0.866
SSIM	0.939	0.946	0.964	0.907	0.941	0.913
FSIM	0.970	0.981	0.967	0.972	0.949	0.964
DIIVINE	0.913	0.910	**0.984**	0.921	0.863	0.916
BLIINDS-II	0.929	0.942	0.969	0.923	0.889	0.931
BRISQUE	0.914	0.965	0.979	0.951	0.877	0.940
CORNIA	0.943	0.955	0.976	**0.969**	0.906	0.942
CNN	**0.952**	**0.977**	0.978	0.962	**0.908**	**0.956**
LCC	JP2K	JPEG	WN	BLUR	FF	ALL
PSNR	0.873	0.876	0.926	0.779	0.870	0.856
SSIM	0.921	0.955	0.982	0.893	0.939	0.906
FSIM	0.910	0.985	0.976	0.978	0.912	0.960
DIIVINE	0.922	0.921	**0.988**	0.923	0.888	0.917
BLIINDS-II	0.935	0.968	0.980	0.938	0.896	0.930
BRISQUE	0.922	0.973	0.985	0.951	0.903	0.942
CORNIA	0.951	0.965	0.987	**0.968**	0.917	0.935
CNN	**0.953**	**0.981**	0.984	0.953	**0.933**	**0.953**

Table 4.1: SROCC and LCC on LIVE. FR-IQA methods are italicized for reference.



(a) (b)

Figure 4.2: Learned convolution kernels on (a) JPEG and (b) ALL on LIVE dataset

number of convolution kernels. It is not surprising to find that the number of filters significantly affects the performance. In general, the use of more kernels leads to better performance, but insignificant performance increase is gained when the number of kernels exceeds 40.

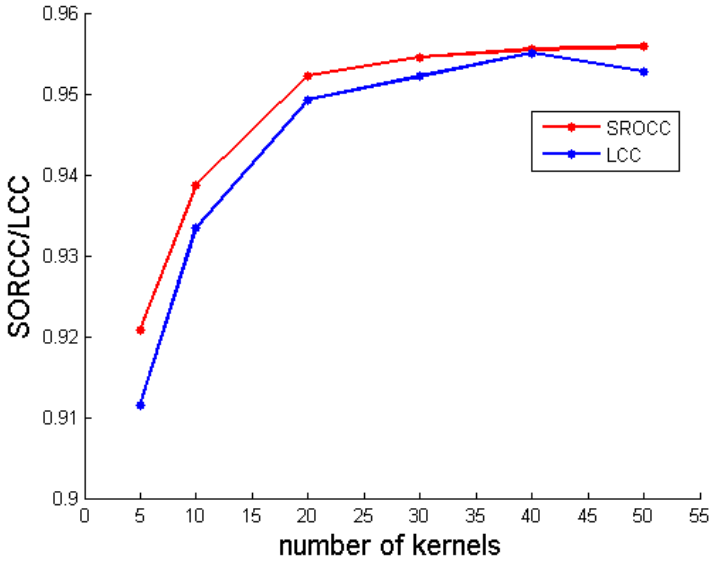


Figure 4.3: SROCC and LCC with respect to number of convolution kernels

size	5×5	7×7	9×9
SROCC	0.953	0.956	0.955
LCC	0.951	0.953	0.955

Table 4.2: SROCC and LCC under different kernel sizes

Kernel size We train and test the network with different kernel sizes while fixing the rest of structure. Table 4.2 shows how the performance changes with the kernel size, and Figure 4.2 illustrates that all tested kernel sizes show similar performance. The proposed network is not sensitive to kernel size.

Patch size In our experiment, the whole image score is simply the average score of all patches sampled, so we examine how the patch sampling strategy affects performance. This includes two aspects, patch size and number of patches per image. If we sample non-overlapping patches, larger patch size leads to fewer patches. For example, if we double the patch size, the number of patches per image will drops to one fourth of the original number. To avoid this situation, we allow overlap sampling and use a fixed sampling stride (32) for different patch sizes. Thus the number of patches per image remains roughly the same (ignoring the border effect) when patch size varies. Table 4.3 shows the change of performance with respect to patch size, indicating that a larger patch results in better performance. The performance increases slightly as the patch size increases from 8 to 48. However, larger patches lead not only to more processing time but also reduce spatial quality resolution. Therefore we prefer to use the smallest patch that yields the state of the art performance.

size	48	40	32	24	16
SROCC	0.959	0.958	0.956	0.950	0.946
LCC	0.957	0.955	0.953	0.947	0.946

Table 4.3: SROCC and LCC on different patch size

Sampling stride To observe how the number of patches affects the overall performance, we fix the patch size and vary the stride. Changing the stride does not change the structure of the network. For simplicity at each iteration of the 100 iteration experiment, we use the same model trained at stride 32, and test with different stride values. Figure 4.4 shows the change of performance with respect to the stride. A larger stride generally leads to lower performance because less image information is used for overall estimation. However, state of the art performance is still maintained even when the stride increases to 128, which roughly corresponds to 1/16 of the original number of patches. This result holds consistent with the fact that the distortions on the LIVE data are roughly homogeneous across entire image, and it also indicates that our CNN can accurately predict quality score on small image patches.

4.1.3.4 Cross Dataset Test

Tests on TID2008 This set of experiment is designed to test the generalization ability of our method. We follow the protocol of previous work [21, 23] to investigate cross dataset performance between the two datasets by training our CNN

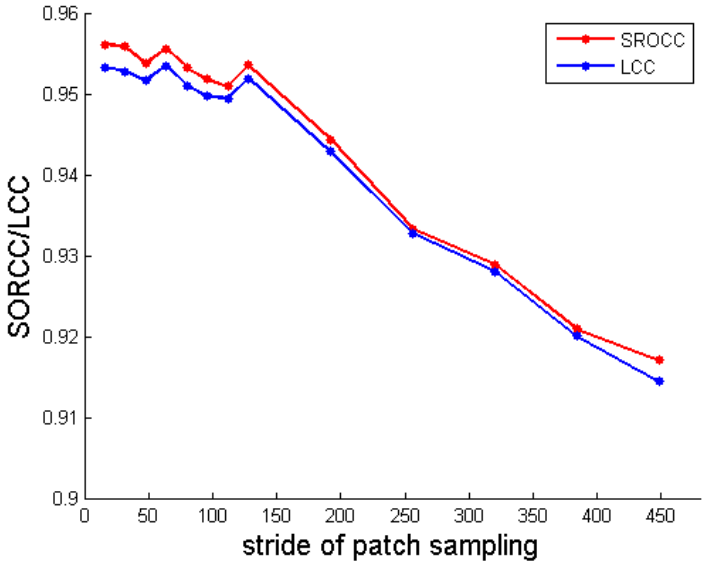


Figure 4.4: SROCC and LCC with respect to the sampling stride

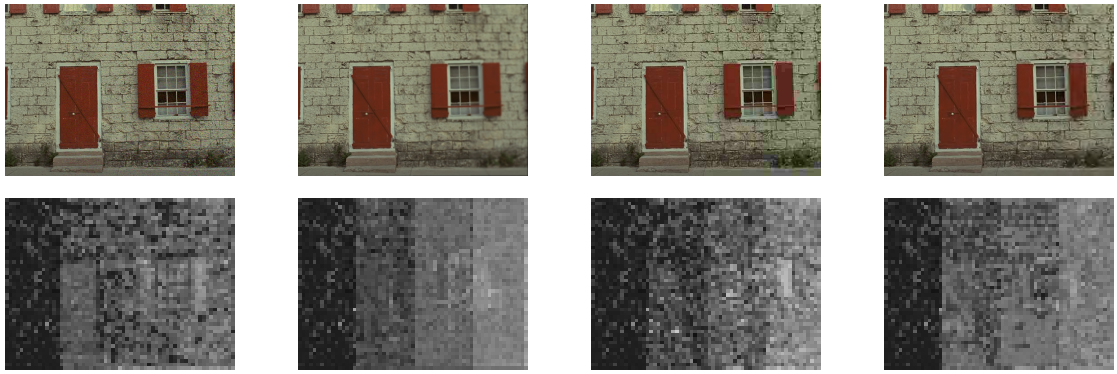
on LIVE and testing on TID2008[1]. This experiments examines only the four types of distortions that are shared by LIVE and TID2008 are examined. The DMOS scores in LIVE range from 0 to 100, while the MOS scores in TID2008 fall in the range 0 and 9. To make a fair comparison, we adopt the same method as [21] to perform a nonlinear mapping on the predicted scores produced by the model trained on LIVE. A nonlinear mapping based on a logistic function is usually applied to FR measures for transforming the quality measure into a certain range. We randomly divide the TID2008 into two parts of 80% and 20% for 100 times. Each time, 80% of data is used for estimating parameters of the logistic function and 20% is used for testing, i.e., evaluating the transformed prediction scores. Results of the cross dataset test are shown in Table 4.4, where we observe that our CNN outperforms

[1]We have observed that some images in TID2008 share the same content as images in LIVE. However, their resolutions are different.

	CORNIA	BRISQUE	CNN
SROCC	0.890	0.882	**0.920**
LCC	0.880	0.892	**0.903**

Table 4.4: SROCC and LCC obtained by training on LIVE and testing on TID2008

previous state of the art methods.



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

Figure 4.5: Synthetic examples and local quality estimation results. The color images contain distortions in (a) WN, (b) BLUR, (c) JPEG, and (d) JP2K. Each color image is divided into four parts, and three of them are distorted in different degradation level. The grayscale images show the local quality estimation results, where brighter pixels indicate lower quality.

4.1.3.5 Local Quality Estimation

Our CNN measures the quality on small image patches, so it can be used to detect low/high quality local regions as well as giving a global score for the entire image.

We select an undistorted reference image from TID 2008 (which is not included in LIVE) and divide it into four vertical parts. We then replace the second to the fourth parts with distorted versions at three different degradation levels. Four synthetic images are generated in this way, one for each types of distortions including WN, BLUR, JPEG, and JP2K. We next perform local quality estimation on these synthetic images using our model trained on LIVE. We scan 16×16 patches with a stride of 8 and normalize the predicted scores into the range $[0, 255]$ for visualization. Figure 4.5 shows the estimated quality map on the synthetic images. We can see that our model properly distinguishes the clean and the distorted parts of each synthetic image.

To better examine the local quality estimation power of our model, we consider several types of distortions in TID2008 that are not used in previous experiments, and find three types that can only affect local regions: JPEG transmission, JPEG2000 transmission, and blockwise distortion. From TID2008, we again choose several images not shared by LIVE, and test on their distorted versions with the these three distortions. Figure 4.6 shows the local quality estimation results. We find our model locates the distorted regions with reasonable accuracy and the results generally match human judgment. Our model locates the blockwise distortion

extremely well, although this type of distortion is not contained in the training data from LIVE. In the images of the third row in Figure 4.6, the stripes on the window are mistaken as a low quality region. We speculate that this occurs because the local patterns on the stripes resemble blockiness distortion. Contextual information may be needed to overcome such problems.

4.1.3.6 Computational Cost

Our CNN is implemented using Theano [77]. With Theano we can easily run our algorithm on a GPU to speed up the process without much optimization. Our experiments are performed on a PC with 1.8GHz CPU and GTX660 GPU. We measure the processing time on images of size 512×768 using our model of 50 kernels with 32×32 input size, and test the model using the part of those strides that give the state of the art performance in the experiments on LIVE. Table 4.5 shows the average processing time per image under different strides. Note that our implementation is not fully optimized. For example, the normalization process for each image is performed on the CPU in about 0.017 sec, which represents a significant portion of the total time. From Table 4.5 we can see that, with a sparser sampling pattern (stride greater than 64), real time processing can be achieved while maintaining state of the art performance.

Figure 4.6: Local quality estimation results on examples of non-global distortion from TID2008. Row 1 shows JPEG transmission errors, row 3 shows jpeg200 transmission errors, and row 5 shows local blockwise distortion. Grayscale images in row 2, 4, and 6 show the local quality estimation results, where brighter pixels indicate lower quality.

stride	32	64	96	128
time(sec)	0.114	0.041	0.029	0.023

Table 4.5: Time cost under different strides.

4.2 Simultaneous Estimation Of Image Quality And Distortion Via Multi-task Convolutional Neural Networks

In most standard image quality assessment datasets, distorted images are grouped into classes according to the types of distortion, such as Gaussian noise, blur, and compression methods, and the images in each group are degraded with different levels of the same distortion. While many image quality assessment methods have been proposed, it is interesting to notice that most methods estimate quality scores without fully utilizing the distortion type information. Some methods can only work well given certain types of distortion, while other methods ignore distortion types in training when estimating the quality for a collection of images with different types of distortion.

Identifying the distortion type is an important task for NR-IQA. A quality score cannot fully characterize the distortion present in an image. A much better description will result if both the distortion type and quality score are determined. Take Figure 4.7, for example, where the images in the same column own very similar quality scores, but have different types of distortions (Gaussian blur and JPEG compression, respectively).




Gaussian blur (84.6) Gaussian blur (68.5) JPEG2000 (74.1) JPEG2000 (48.6)




JPEG (83.1) JPEG (67.8) white noise (73.5) white noise (48.1)

Figure 4.7: Images in each column have very similar image quality scores, but they have different distortion types. Higher score denotes worse quality.

A few methods, such as [19, 23], attempted a two-stage framework. They first perform distortion identification, i.e., classify the images into one of the distortions, then perform the distortion-specific quality estimation. However, experiments in [23] show that such a two-stage approach does not produce results superior to the distortion-blind approach.

This disappointing evidence may be explained in two completely different ways: 1) distortion type does not facilitate image quality assessment, or 2) distortion identification and quality estimation are two correlated tasks. Explicitly separating these two tasks into two stages may not be optimal to exploit the extra information

provided by the distortion type. We argue that the latter is more likely to be the reason.

We use a multi-task Convolutional Neural Network (CNN) to address this problem. The two tasks, quality estimation and distortion identification, are learned simultaneously with one CNN. For multi-task learning, the literature typically specifies a main task and several extras. In this section, however, we are optimizing both tasks, therefore we use the term primary and secondary tasks. Since quality estimation is more widely addressed, we treat it as the primary task and distortion identification as the secondary task.

Neural networks can be designed conveniently for multi-task learning conveniently. It is natural to create a network that has shared hidden layers, and multiple output layers for different tasks [84]. In this section, we further show that network structures also influence the multi-task performance. For instance, we empirically show that simply increasing the number of tasks based on the state of the art structure may not lead to optimal solutions, because the structure was tuned heavily to one of the tasks heavily.

We propose a CNN structure that shares internal representation among the tasks. Specifically, our CNN contains two convolutional layers with linear neurons, one ordinary pooling and one max-min pooling, two fully connected layers equipped with Rectified Linear Units (ReLU), and two output layers for the two tasks. Compared to the architecture described in the last section, we added one more layer of convolutional filtering and adjusted the neuron number in fully connected layers accordingly. The proposed model has nearly 90% less parameters, yet it has strong

representation power for both tasks and yields balanced results.

We show through experiments that the our multi-task CNN achieves the similar or better performance on image quality estimation and distortion identification compared to the state of the art.

4.2.1 Related work

As described in the last section, a large portion of previous research efforts on NR-IQA focus on Natural Scene Statistics (NSS) based features, including DIIVINE [19], BLIINDS-II [85], and BRISQUE [23]. It is worth noting that DIIVINE and BRISQUE use a two-stage framework. They first perform distortion identification, i.e., classify the images into one of the distortions, then they perform the distortion-specific quality estimation. However, experiments in [23] show that such a two-stage approach is not superior to the distortion-blind approach.

Multi-task learning using neural networks has been studied for decades. Abu-Mostafa [86] suggested one multi-task setting called catalytic hints, where one task is the internal features used by other tasks. Caruana [84] studied on multi-task learning with neural networks on a number of problems. He demonstrated that learning multiple correlated tasks at the same time may significantly improve the performance of the main task.

4.2.2 Approach

We believe that image quality estimation and distortion identification are two correlated tasks. Therefore, the critical question becomes how to utilize both channels' information in the training to improve the performance. We design a multi-task CNN that simultaneously learns to predict quality score and identify distortion.

We first describe a naive extension of our previous CNN architecture (described in the last section) to a multi-task CNN. Then, we discuss the issues in the network and present our multi-task network by increasing filter layers and adjusting hidden node numbers. While it is unlikely that all tasks can achieve the best performance at the same time [84], we aim to generate balanced outputs that improve both tasks.

A number of methodologies can train the multi-task network. The network can be jointly trained simultaneously using all tasks, or it can be trained iteratively. In this section, we jointly train the network, but use different learning rates for different tasks in their private layers. While it is unlikely that all tasks can achieve the best performance at the same time [84], our goal is to generate balanced outputs that improve both tasks.

4.2.2.1 Overall process

We convert the color image to grayscale, and normalize the image similar to [23], and divide the image into patches (typical size 32×32 pixels). We then make predictions on each patch using a multi-task CNN, and aggregate patch predictions to obtain the result for the image. Assuming that the distortion is homogeneous

across the entire image, the image quality score is simply the average of patches' quality scores. The image distortion is decided by a majority voting of the patches, i.e., the most frequently occurring distortion on patches determines the distortion of the image.

Dividing an image into patches provides a large quantity of training samples for the CNN. Accurate predictions on patches lead to superior performance on the image level, especially for distortion identification, as we will demonstrate in the experiments section.

4.2.2.2 From single to multi-task architectures

In this section, we refer to our CNN architecture in the last section as IQA-CNN. Then, we naively extend it to IQA-CNN+, a multi-task variant by directly adding a minor task in the output layer, as a baseline. As described previously, IQA-CNN for quality score estimation has one convolutional layer, one pooling layer, two fully connected layers, and one output layer. We extend this structure for the multi-task by adding a classification layer, referring to it as IQA-CNN+. The secondary task for distortion identification shares the same structure as the one for quality score estimation, except that the output layer is a multi-class logistic regression.

Unfortunately such a simple extension is not ideal for the multi-task scenario. The reason is three-fold. First, the IQA-CNN+ has a shallow convolutional structure (one layer), which makes the feature learning less efficient compared to deeper structures, because the filters share no patterns. Second, given such a large number

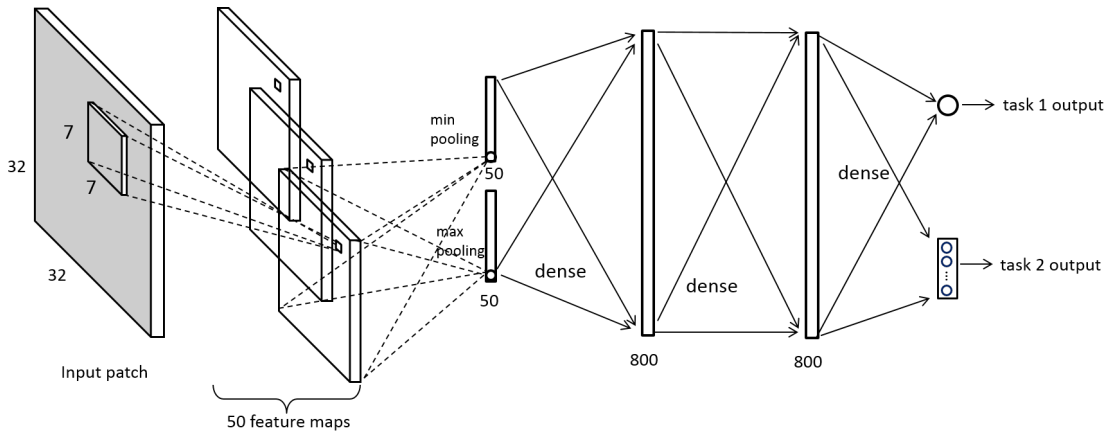


Figure 4.8: The architecture of the IQA-CNN+ as a naive multi-task extension to IQA-CNN.

of neurons, too many parameters need to be learned in the network. The larger model size makes it less practical, or even impossible in some applications. Third, the arrangement of the fully connected layers may not facilitate multiple tasks when it is specially tuned for one task.

4.2.2.3 IQA-CNN++: a compact multi-task network

In the multi-task setting, we aim to estimate the quality and identify the distortion in one network. Therefore, the network needs to learn features shared by both tasks, and the importance of the two tasks need to be balanced. We also prefer a compact model that is significantly smaller in size but has a comparable performance. As a result, we propose the following two modifications: 1) increasing the number of convolutional layers while reducing the receptive field of the filters, and 2) modifying the fully connected layers to have a "fan-out" shape with significantly fewer neurons. We refer to our new model as IQA-CNN++.

The structure of IQA-CNN++ is shown in Figure 4.9. Our new architecture contains two convolutional layers, each with a pooling, followed by two fully connected layers and one output layer. The first convolutional layer contains 8 kernels each of size 3×3, followed by a 2×2 pooling. The second convolutional layer contains 32 kernels, each of size $3 \times 3 \times 8$. The 32 feature maps obtained by the second convolutional layer are pooled to 32 max and 32 min values, i.e., each feature map is pooled to one max and one min value, which form 64 inputs for the next layer. Again, no nonlinear neurons are used in the convolutional layers. There are 128 and 512 Rectified Linear Units (ReLUs) in the two fully connected layers, respectively. Both the linear regression layer and logistic regression layer exist in the last part of the multi-task network, and they both take as inputs the second fully connected layer's outputs, i.e., the two tasks share all internal structures.

Through some simple computation we determine that IQA-CNN (and IQA-CNN+) has approximately 7.2×10^5 learnable parameters (weights of neurons). By comparison our IQA-CNN++ consists of roughly 7.7×10^4 learnable parameters, which reduces the model size by 90%. Despite a significant reduction in size, the IQA-CNN++ still shows excellent performance as shown in later experiments.

4.2.2.4 Multi-task CNN learning

In the single-task learning, the training stage is simple. We define a loss function for the task and use the Stochastic Gradient Descent (SGD) and back-propagation to minimize the loss, as well as updating the network parameters.

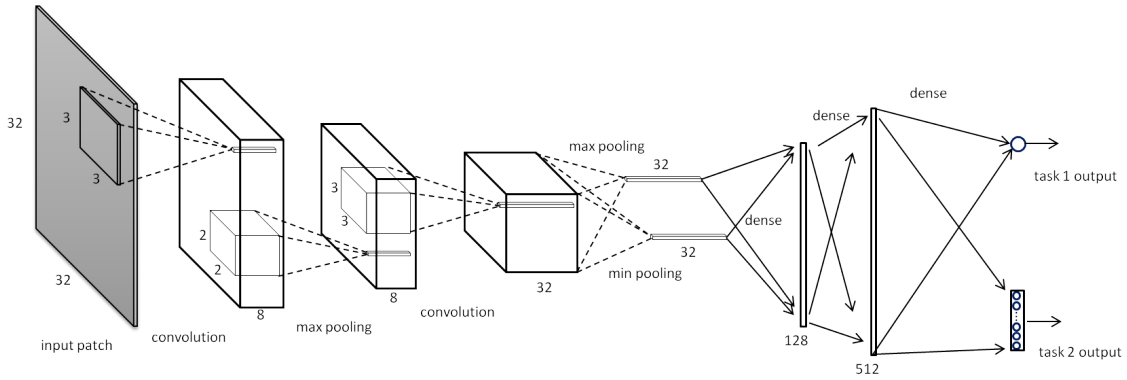


Figure 4.9: The architecture of the IQA-CNN++ for simultaneously image quality estimation and distortion identification.

In the multi-task setting, training is similar. Define quality score estimation as task 1, and let x_n and y_n denote the input patch and its ground truth score, respectively, and let $f(x_n; W)$ be the predicted score of x_n with network weights W. The loss L_1 is measured using the l_1 norm of the prediction error. The optimization objective for task 1 is defined as follows:

$$\min_W L_1 = \min_W \frac{1}{N} \sum_{n=1}^{N} \|f(x_n; W) - y_n\|_1 \tag{4.5}$$

where N is the total number of patches.

Define distortion identification as task 2, where the negative log-likelihood is used as the loss. Let z_n denote the ground truth of distortion type for patch x_n, and $g(x_n, z_n; W)$ denote the negative log likelihood of x_n. Let L_2 denote the loss and the optimization objective for task 2 can expressed as :

$$\min_W L_2 = \min_W \frac{1}{N} \sum_{n=1}^{N} g(x_n, z_n; W) \tag{4.6}$$

In the multi-task learning, the loss function can be formulated as a weighted

sum of each task's loss. Suppose there are M tasks. Let α_i and L_i denote the weight and loss for task i, respectively. L is the overall loss, then the multi-task optimization objective is formulated as:

$$\min_W L = \min_W \frac{1}{N} \sum_{n=1}^{N} (\alpha_1 \left\| f(x_n; W) - y_n \right\|_1 + \alpha_2 g(x_n, z_n; W)) \tag{4.7}$$

If $\alpha_m = 0$, the learning is identical to task m. We employ SGD and backprop to approximately minimize the loss, so in practice the network parameters are updated using the weighted sum of the gradients from all tasks. Let w^i denote the ith network parameter, D_m^i be the gradient for w^i from task m, and p be the learning rate. The following updating rule applies for each iteration:

$$w^i \leftarrow w^i - p \sum_{m=1}^{2} \alpha_m D_m^i \tag{4.8}$$

We choose α_m according to two criteria: first, they should to be large enough to update each individual task effectively; second, not too large to collectively produce enormous updating on the network weights. In practice, since we are mainly interested in task 1, we first pick the best α_1, then pick the largest α_2 that can work with α_1.

Similar to the training of our previous model, we apply dropout only at the second fully connected layer. Half of the inputs of the second fully connected layer are randomly masked in training, and in testing their weights are divided by 2. We also use momentum for robust training.

4.2.3 Experiments

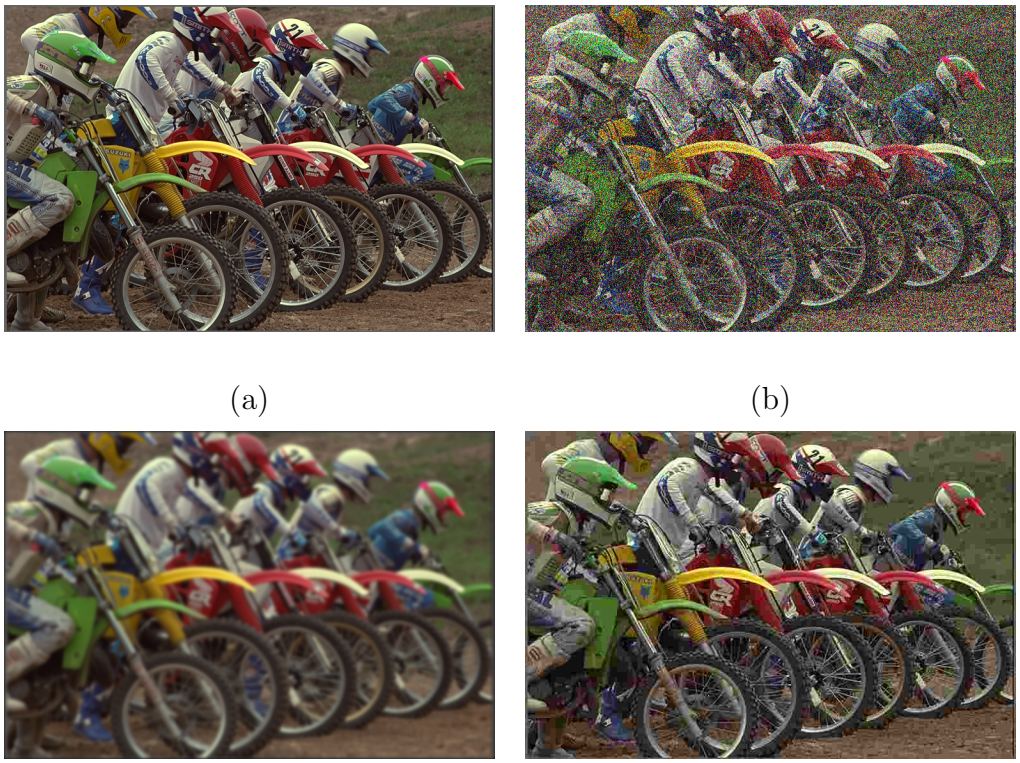
We present experiments on three standard datasets and measure the performance in using the standard metric. Particularly, we compare the multi-task CNNs against other models and our previous method. Cross dataset experiments are also conducted to demonstrate the generalization power of the proposed approach.

4.2.3.1 Experimental Protocol

Datasets: Our experiments employ the following three datasets.

(1) LIVE [82]: A total of 779 distorted images with five different distortions – JPEG2000 compression (JPEG2K), JPEG compression (JPEG), white Gaussian (WN), Gaussian blur (BLUR), and fast fading (FF) at 7-8 degradation levels derived from 29 reference images. Differential Mean Opinion Score (DMOS) is provided for each image, roughly in the range $[0, 100]$. Higher DMOS indicates lower quality.

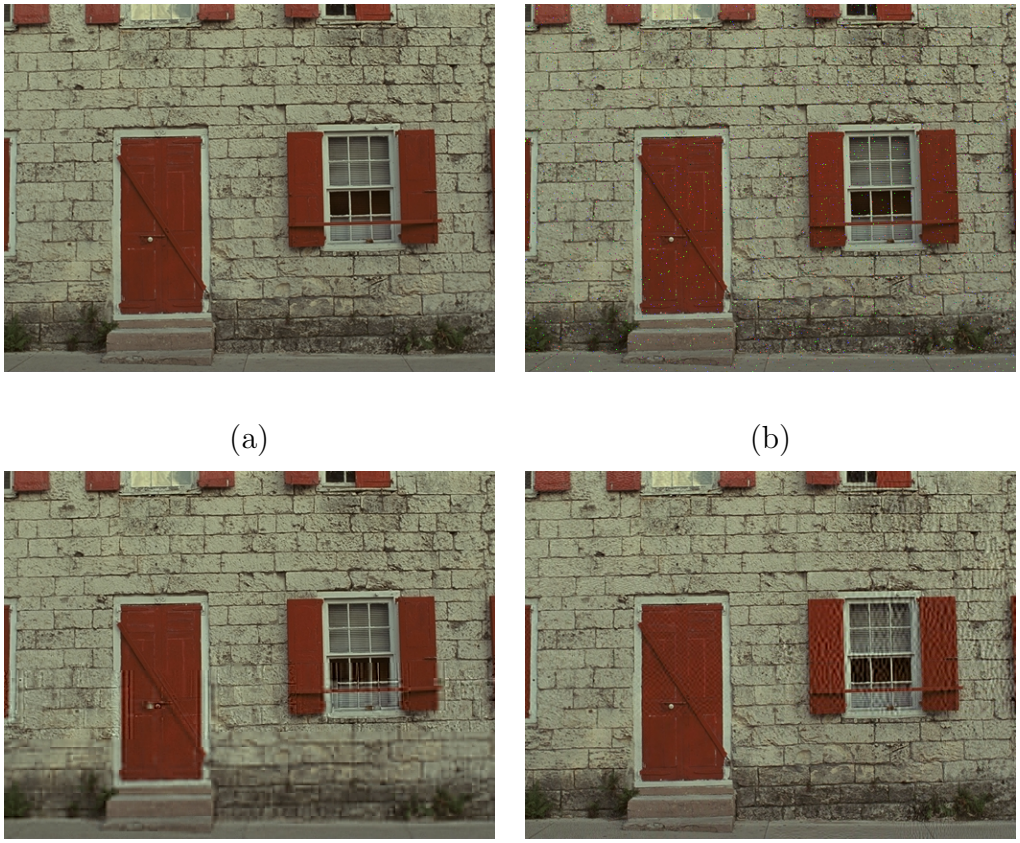
(2) TID2008 [83]: 1,700 distorted images with 17 different distortions derived from 25 reference images at 4 degradation levels. Each image is associated with a Mean Opinion Score (MOS) in the range $[0, 9]$. Contrary to DMOS, higher MOS indicates higher quality. In our experiments, we use the first 13 distortions including: additive Gaussian noise (WN), additive noise in color components (WNC), spatially correlated noise (SCN), masked noise (MN), high frequency noise (HFN), impulse noise (IN), quantization noise (QN), Gaussian blur (BLUR), image denoising (IDN), JPEG compression (JPEG), JPEG2000 compression (JPEG2K), JPEG transmission errors (JPEGTE), JPEG2000 transmission errors (JP2KTE).



(a) (b)

(c) (d)

Figure 4.10: Sample images from the LIVE dataset: (a) clean reference image, (b) white Gaussian noise, (c) Gaussian blur, and (d) jpeg compression.



(a) (b)

(c) (d)

Figure 4.11: Sample images from the TID2008 dataset: (a) clean reference image, (b) impulse noise, (c) jpeg transmission noise, and (d) jpeg2000 transmission noise.

(3) CSIQ [87]: 30 original images distorted using six different types of distortions at four to five levels each, resulting in a total of 866 distorted images. DMOS is provided for each distorted image. The six distortions are JPEG compression (JPEG), JPEG-2000 compression (JP2K), global contrast decrements (CONTRAST), additive pink Gaussian noise (FNOISE), additive white Gaussian noise (WN), and Gaussian blur (BLUR). We also use only four distortions that appear in LIVE dataset, i.e., JPEG2K, JPEG, WN, and BLUR.

Evaluation: We use these two measures to evaluate the performance of the quality estimation: 1) Linear Correlation Coefficient (LCC), which measures the linear dependence between two quantities, and 2) Spearman Rank Order Correlation Coefficient (SROCC), which measures how well one quantity can be described as a monotonic function of another quantity. For the distortion identification task, we simply compute the classification accuracy. We report results obtained from 100 train-test trials, where in each trial we randomly select 60% of reference images and their distorted versions as the training set, 20% as the validation set, and the remaining 20% as the test set.

4.2.3.2 Evaluation on LIVE

On the LIVE dataset, we train and test on all five distortions together. We compare our multi-task CNNs with previous methods on both quality estimation and distortion identification tasks. Then, we discuss the difference between our multi-task CNNs. Figure 4.10 shows a few samples from the LIVE dataset, and we

can see the distortions are distributed homogeneously.

4.2.3.2.1 Multi-task CNNs vs previous models

Multi-task CNNs utilize both DMOS and distortion types for training and accomplish the two tasks simultaneously, as described in the above section. As a comparison, all previous methods use DMOS as ground truth for quality estimation training, and use distortion types independently for distortion identification only if needed.

In Table 4.6, we compare the performance of multi-task CNNs with previous methods, including DIIVINE, BLIINDS-II, BRISQUE, CORNIA, and IQA-CNN. Note that BLIINDS-II and BRISQUE reported the classification accuracy of distortion types, but CORNIA's classification result was not directly available. Thus we used CORNIA's implementation and obtained the classification accuracy. Performance of full reference methods appear in Table 4.6 for reference. The top three results are highlighted in red, blue, and green, respectively.

Quality score estimation: Table 4.6 shows that in the quality estimation task multi-task CNNs (IQA-CNN+ and IQA-CNN++) outperformed the non-CNN based methods. Both multi-task CNNs achieved similar performance compared to IQA-CNN.

Distortion identification: For the distortion identification task, both multi-task CNNs achieved much higher accuracy. Compared with the state of the art, the gains are approximately 5% and 8%, respectively. IQA-CNN++ achieves the best performance (0.951) among all competitors.

	LCC	SROCC	Class. Acc.
PSNR	0.856	0.866	-
SSIM	0.906	0.913	-
FSIM	0.960	0.964	-
DIIVINE	0.917	0.916	-
BLIINDS-II	0.930	0.931	0.838
BRISQUE	0.942	0.940	0.886
CORNIA	0.935	0.942	0.875
IQA-CNN	0.953	0.956	-
IQA-CNN+	0.953	0.953	0.921
IQA-CNN++	0.950	0.950	0.951

Table 4.6: Performance of quality estimation and distortion identification on LIVE. Full-Reference methods are italicized. The top three results are highlighted in red, blue, and green, respectively.

The appealing performance of the multi-task CNNs in distortion identification partly stems from the voting of patches. On the patch level, we observe a classification accuracy around 0.88, therefore the correct prediction is highly likely to collect the most votes and appear as the final prediction on image level. From this aspect we can see that being able to predict on small image patches with a reasonable accuracy contributes greatly to the superior performance on distortion identification.

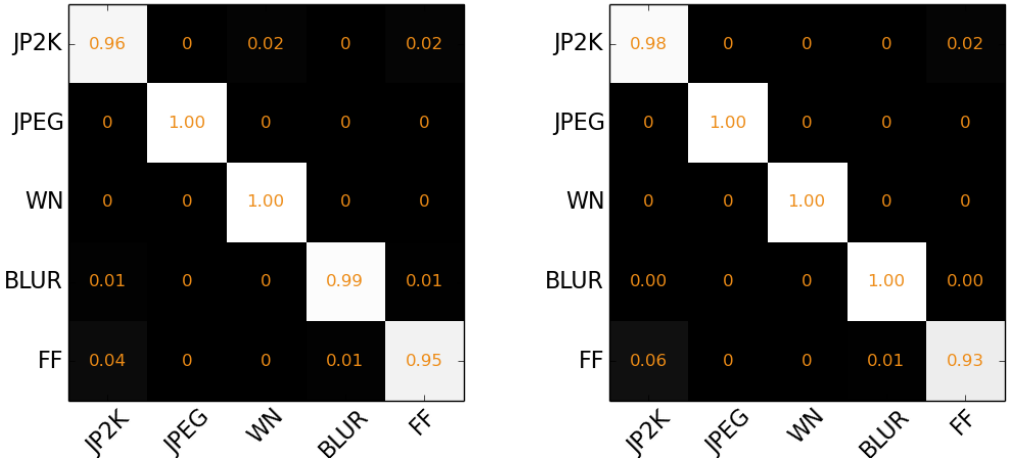
4.2.3.2.2 Comparisons on different multi-task CNNs

Comparing two multi-task CNNs, the IQA-CNN++ performs notably well in distortion identification, with a 3% gain over IQA-CNN+, while the two mthods show similar performance in quality estimation. To further analyze the influence of the network architecture, we compute the confusion matrices for both CNNs in Figure 4.12. For IQA-CNN+, one type (JPEG2K) performs relatively poorly. In comparison, IQA-CNN++ performs well on all distortions without failing on any particular one.

4.2.3.3 Evaluation on TID2008

On the TID2008 dataset, we train and test on 13 distortions together, similar to the experiments on LIVE described above. In Table 4.7 we compare the performance of IQA-CNN+, IQA-CNN++, and other methods with available results.

From Table 4.7 we observe that both IQA-CNN+ and IQA-CNN++ outperform the previous methods by a large margin. For the quality estimation task,



(a) IQA-CNN+ (b) IQA-CNN++

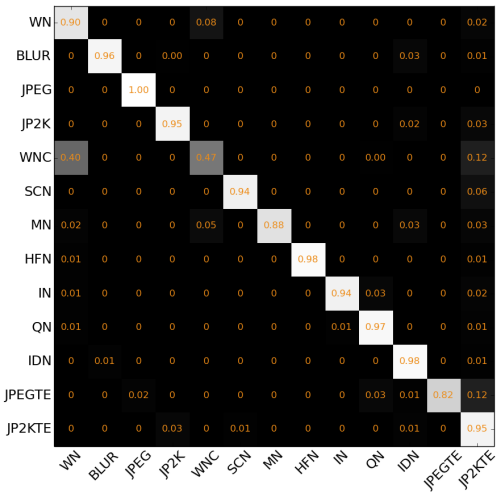
Figure 4.12: Confusion matrices on LIVE dataset.

IQA-CNN++ achieves an LCC and SROCC of 0.880 and 0.870, showing advantage over the IQA-CNN+ as well as other methods. On the distortion identification side, IQA-CNN++ achieves the highest accuracy of 0.929 while other methods are below 0.9.

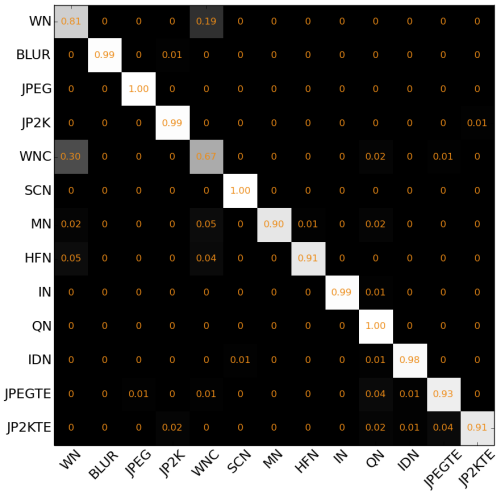
Some of the 13 distortions in this experiment are not homogeneous on the entire image, e.g., impulse noise (IN), JPEG transmission errors (JPEGTE), and JPEG2000 transmission errors (JP2KTE) occur either very sparsely or concentrate on a small fraction of the image, as shown in Figure 4.11. IQA-CNN++ handles these issues well. From the confusion matrix (Figure 4.13b), notice that IQA-CNN++ achieves a recall higher than 0.9 on almost every distortion (except WN and WNC), which leads to an overall better performance. Both IQA-CNN+ and IQA-CNN++ become confused on WN and WNC, but which is understandable because we work on grayscale images where WNC is almost indistinguishable from

	LCC	SROCC	Class. Acc.
PSNR	0.652	0.669	-
SSIM	0.857	0.878	-
FSIM	0.913	0.926	-
CORNIA	0.837	0.813	0.862
IQA-CNN	0.873	0.862	
IQA-CNN+	0.870	0.861	0.889
IQA-CNN++	0.880	0.870	0.929

Table 4.7: Performance of quality estimation (measured in LCC and SROCC) and distortion identification (measured in classification accuracy) on 13 distortions of TID2008. Full-Reference methods are italicized. Red and blue denote the top two results, respectively



(a) IQA-CNN+



(b) IQA-CNN++

Figure 4.13: Confusion matrices on the 13 distortions of TID2008.

WN.

Training on these locally distorted images can prove difficult, because two images of different degradation levels or distortion type may produce the same clean patches with different quality scores and distortion types, which makes the training data noisy. Despite this issue, the proposed methods performs well. This fact demonstrates that the proposed approaches have a strong learning ability that is robust to noise in samples.

4.2.3.4 Cross Dataset Test

To observe how well our methods generalize, we conduct cross dataset tests. Training and validation are performed on LIVE, then the obtained model is tested on TID2008 and CSIQ without adapting any parameters. Both the quality estimation and the distortion identification tasks are tested. The three datasets cover different types of distortions so we use only the four distortions common to all the three datasets, JPEG2K, JPEG, WN, and BLUR.

Since the model trained on LIVE produces a quality score in the same range as DMOS, which is different from MOS in TID2008, we follow the tradition as in [21] to apply a nonlinear mapping on the predicted quality scores on TID2008. No mapping is applied for CSIQ because it also uses DMOS as the ground truth quality measure. Tables 4.8 and 4.9 show the results of the cross dataset tests on TID2008 and CSIQ, respectively. For image quality estimation task, the IQA-CNN++ trained on LIVE achieves an LCC/SORCC of 0.895/0.906 on TID2008 and 0.928/0.936 on CSIQ,

	LCC	SROCC	Class. Acc.
PSNR	0.776	0.901	-
SSIM	0.817	0.903	-
FSIM	0.952	0.954	-
CORNIA	0.890	0.880	0.920
IQA-CNN	0.903	0.920	-
IQA-CNN+	0.893	0.912	0.890
IQA-CNN++	0.895	0.906	0.933

Table 4.8: Performance of quality estimation (measured in LCC and SROCC) and distortion identification (measured in classification accuracy) on four common distortions of TID2008, using models trained on LIVE. Full-Reference methods are italicized. The top three results are highlighted by red, blue, and green, respectively.

outperforming other methods. IQA-CNN++ also shows the best best performance on the distortion identification task for both datasets.

4.2.3.5 Discussion

The experiments show that the proposed multi-task CNNs generally achieve the state of the art performance on both image quality estimation and distortion identification. For the task of image distortion identification, the IQA-CNN++ consistently outperforms others on all three datasets, with comparable performances in the quality estimation. This demonstrates that the proposed multi-task learning

	LCC	SROCC	Class. Acc.
FSIM	0.961	0.962	-
CORNIA	0.914	0.899	0.768
IQA-CNN	0.913	0.923	-
IQA-CNN+	0.910	0.918	0.730
IQA-CNN++	0.928	0.936	0.783

Table 4.9: Performance of quality estimation (measured in LCC and SROCC) and distortion identification (measured in classification accuracy) on four common distortions of CSIQ, using models trained on LIVE. Full-Reference methods are italicized. The top three results are highlighted in red, blue, and green, respectively.

framework has excellent and balanced performance, which makes it more suitable as an overall solution.

4.2.3.6 Computational Cost

Convolutional networks typically require a large amount of computational resources. The implementation of the proposed multi-task CNN is based on Theano [77] which enables the computation to be deployed easily on GPU. We measured the processing time on images of size 512×768. On a PC with 2.8GHz CPU and Tesla C1060 GPU, IQA-CNN+ and IQA-CNN++ take 0.017 and 0.013 seconds, respectively, on average to process an image.

4.3 Document Image Quality Assessment

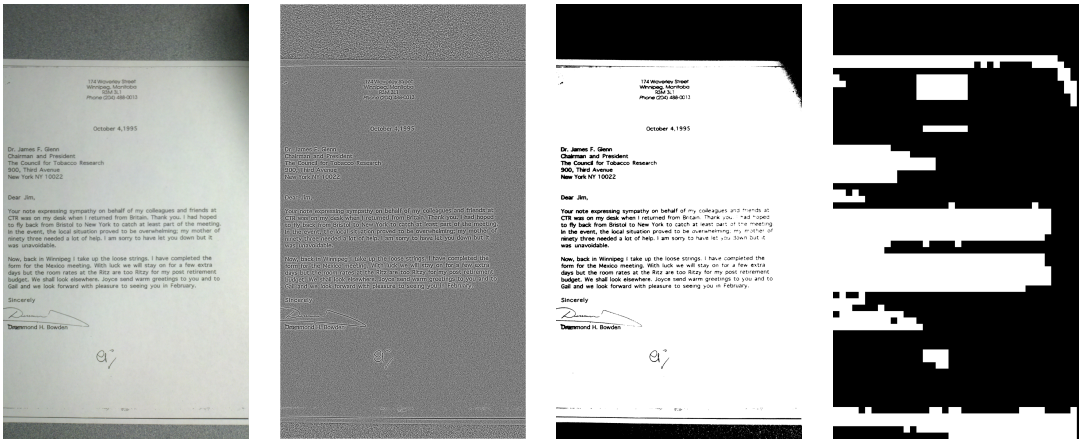
Document quality has a direct impact on the optical character recognition (OCR) performance. Thus, it is desirable to estimate document quality before applying OCR, and Document image quality assessment (DIQA) has attracted attention from OCR research community. In this section, we assume the quality of a degraded document image is directly correlated with the performance of OCR software it uses. A document quality prediction system can be used in many practical applications [88]. For example, it can filter highly degraded document image for which the OCR system cannot handle, or it can be used to select high quality document frames in a video capture system [89]. When applying a document enhancement method, we may be able to avoid further degradation under the guidance of a quality measure.

We present a CNN based method for DIQA, which provides a more unified and principled method for feature learning and regression. Taking the advantage of the homogeneous nature of typical distortions, we divide document images into patches to increase the number of training samples significantly. This allows our method to work for large images. An efficient patch selection process is employed and only informative patches are input to the CNN. Our method demonstrates state of the art performance on two document quality datasets.

4.3.1 Related Work

Early work on DIQA focused on deriving solutions for specific types of document degradations and relied on hand-crafted features. In [90–92], several quality factors for typewritten document images are proposed including: Font Size (FS), Small Speckle Factor (SSF), Stroke Thickness Factor (STF), White Speckle Factor (WSF) and Broken Character Factor (BCF). These metrics were computed empirically based on connected-components and were chosen because they may have a high correlation with the OCR error rate. They have been used to predict OCR accuracy and to choose the best restoration method for preprocessing. However, these metrics cannot be directly applied to general document distortions for the following reasons: 1) the computation of these metrics depends on font size. Thus they are effective only under the assumption that the sizes of individual characters are similar. However, a complex document image may contain characters of different font or stroke sizes, and some scripts such as Arabic typically show varying stroke sizes. 2) The touching of handwritten characters could relate to the writing style of writer and not related directly to quality.

Recently, Kumar et al. [93] proposed a sharpness measure for camera-captured document images, which is specifically designed to measure the blur distortion and may only be applied to camera-captured document images. The first general-purpose method seen in the literature is the feature learning method introduced by Ye et al. [94], which is an extension of the CORNIA system [21]. This method is based on an unsupervised feature learning that can automatically learn discriminant features for



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

Figure 4.14: (a) A document image of size 1860×3264, (b) local normalization result (intensity rescaled for better visualization), (c) binary map obtained from the original image using Otsu's method, and (d) mask of non-constant 48×48 patches (white).

different types of document degradations. Unfortunately, this is a rather empirical feature learning solution.

4.3.2 Approach

We introduce the overall process of estimating document image quality. We preprocess a grayscale document image with local normalization, crop the image into patches, use the CNN to estimate quality scores for selected patches, and average the scores to obtain a score for the image. We aim to predict quality scores that correlate with OCR accuracies as much as possible.

4.3.2.1 Preprocessing

Preprocessing is typically required for general image quality assessment to be robust to intensity and contrast change. As in [23], we perform a local normalization over the entire image. Each pixel is subtracted by the mean and divided by standard deviation of the pixels in a surrounding window. Figure4.14(a) and (b) show a document image and its local normalization result.

4.3.2.2 Patch sifting

We perform Otsu's binarization [95] on the raw image and obtain a binary map corresponding to foreground and background. We crop patches from the pre-processed (i.e., locally normalized) images and check their corresponding patches on the binary map. If the patch on the binary map is constant, i.e., all ones or all zeros, then the patch is discarded. The patch size is chosen to be larger than the typical stroke width, meaning text patches are preserved. Most patches discarded in this manner are background patches or non-text foreground patches. Figure 4.14 (c) shows the result of Otsu's binarization, and Figure 4.14 (d) shows the locations of patches selected after sifting.

Given that we predict quality with respect to OCR performance, we would like to focus on the patches that contain characters. In our document dataset, most image content is either text or background, thus we need to discard the background patches and use the rest for quality estimation.

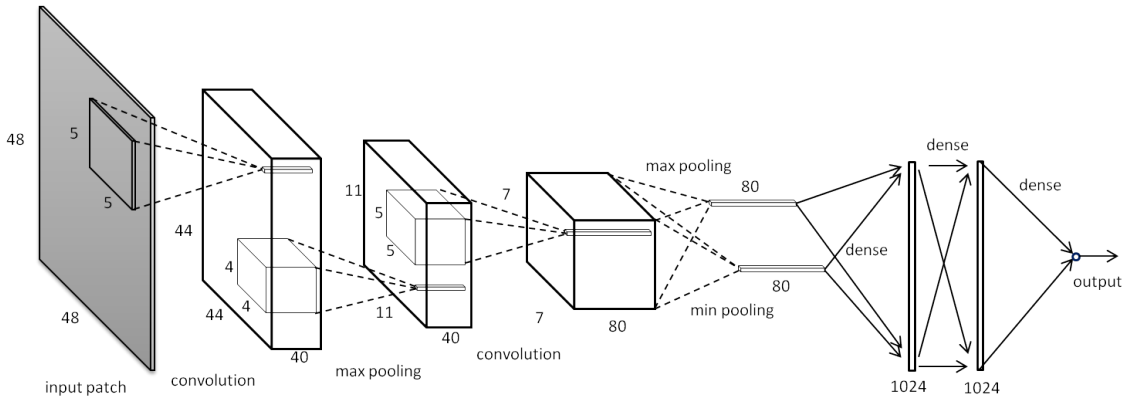


Figure 4.15: The architecture of the proposed CNN.

4.3.2.3 Network Architecture

Once the patches are obtained, we feed them into a network. Figure 4.15 shows the architecture of the proposed network. The network contains two convolution and pooling layers, two fully connected layers, and one output layer. The input is sifted patches of 48×48. The first convolution layer contains 40 kernels, each of 5×5, followed by a 4×4 max pooling, then the second convolution layer with 80 kernels each of 5×5. Following the second convolution layer is a special max-min pooling that we will explain later. Each of the two fully connected layers contains 1024 nodes. The last layer is a linear regression that outputs the quality score.

We use Rectified Linear Units (ReLUs) [60] as the neurons in the two fully connected layers. Formally ReLUs can be expressed as $f(x) = max(0, x)$, where x denotes the input. ReLUs allow positive signals to pass and suppress the negative signals. Compared to traditional sigmoid or tanh neurons, ReLUs are robust to input range and leads to faster training as demonstrated in [26]. Note that no nonlinear transform is applied in convolution or pooling layers, or equivalently, a

linear neuron ($f(x) = x$) is applied, because in experiments we did not observe any benefit from using ReLUs in the convolution layers.

As previously mentioned, a special max-min pooling after the second convolution layer is used. Specifically, each feature map obtained by the second convolution layer is pooled into *one* max value and *one* min value. Suppose there are 80 kernels in the second convolution layer, after max-min pooling we obtain 80 max values and 80 min values for a total of 160 outputs. The max-min pooling discards the location information of features, but the filter responses characterized by maxs and mins are sufficient to capture the quality's statistics.

4.3.2.4 Learning Procedure

By cropping images into patches we have plenty of training samples. Most importantly the training patches are all labeled, and we can use the labels of patches the same as the ground truth OCR scores of their original images.

During training, for each patch we try to train the network to predict a score close to the ground truth. The error between the last layer's output (predicted score) and the patch's ground truth is measured by the l_1 norm. We use Stochastic Gradient Decent (SGD) and backpropagation to solve the minimization and update the parameters. Training is performed on minibatches of samples for a given number of epochs, and we select the model parameters that achieve the best performance on the validation set.

4.3.3 Experiments

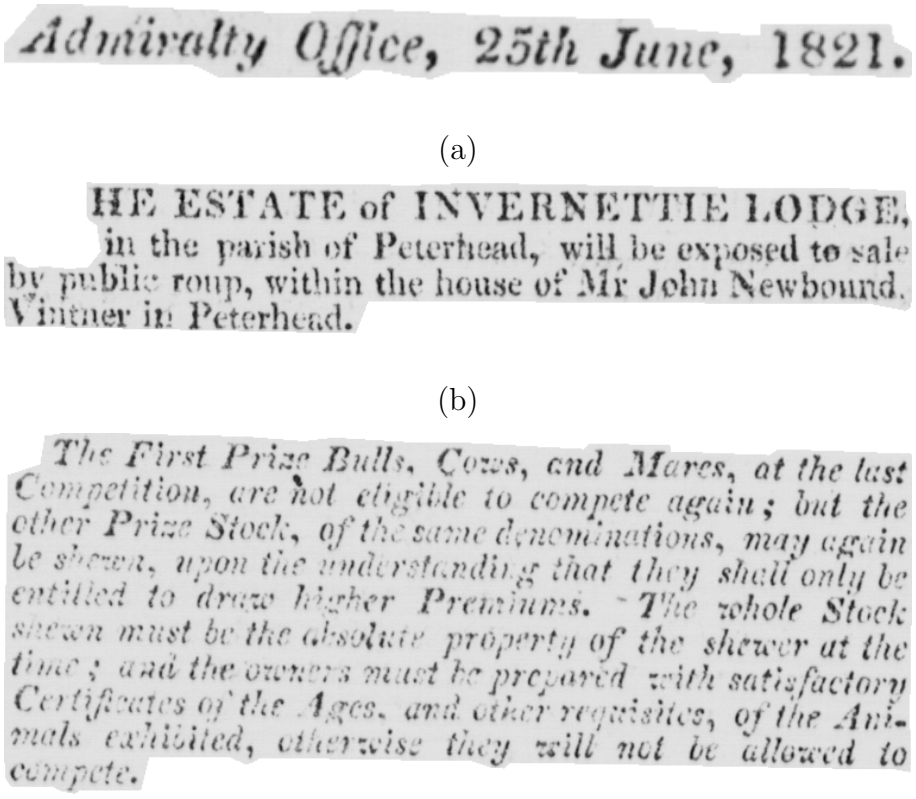
4.3.4 Dataset and protocol

Datasets: We conduct experiments on the following two datasets.

(1) Sharpness-OCR-Correlation (SOC) dataset [96]: a total of 175 color images with resolution 1840×3264. These images are captures of 25 documents using a cell phone camera. Each document contains machine printed English, and 6-8 photos, with varying focal lengths, were taken to generate different levels of blur. Figure 4.14(a) shows a sample image of this dataset. A commercial OCR software (ABBYY Fine Reader) was performed on each of the 175 images, and the OCR results were evaluated by the ISRI-OCR evaluation tool [97] to obtain OCR accuracies in the range $[0, 1]$. The OCR accuracy is the ground truth for each image in our quality assessment task.

(2) Newspaper dataset [94]: a total of 521 grayscale images with various resolution. These images represent a subset of a historical collection and contain machine printed English and Greek. Each image in this dataset is a text region instead of an entire page. The OCR accuracies were obtained for each image using ABBYY Fine Reader and ISRI-OCR in the same way as the SOC dataset. On this dataset, the OCR performance is mainly affected by broken strokes. Figure 4.16 shows sample images from this dataset.

Evaluation protocol: Following the tradition in natural image quality assessment, we compute the correlation between the predicted quality scores and ground truth



(a)

(b)

(c)

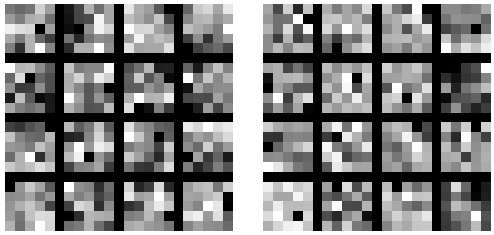
Figure 4.16: Sample images from the Newspaper dataset

	BRISQUE-L [23]	CORNIA [21]	CORNIA-SF [22]	CNN
LCC	0.904	0.937	0.927	**0.950**
SROCC	0.836	0.862	0.854	**0.898**

Table 4.10: Median LCC and SROCC over 100 random sampling experiments on SOC dataset

OCR accuracies. Specifically, we use the Linear Correlation Coefficient (LCC) and the Spearman Rank Order Correlation Coefficient (SROCC) to evaluate the performance of the proposed algorithm and compare it to previous methods. LCC is a measure of the degree of linear relationship between two variables. SROCC measures how well the relationship between two variables can be described using a monotonic function.

In our experiments, we randomly sample 60% of the data as the training set, 20% as the validation set, and leave the remaining 20% as a test set. This random split of the dataset is reasonable for the Newspaper dataset, but not on the SOC dataset. The images of the SOC dataset are organized in groups, in which each group contains only images taken from the same document. Thus, the random split is conducted at the group level. For both datasets, this random split of data is repeated 100 times, each time the LCC and the SROCC are computed, and we report the median LCC and SROCC.



(a) (b)

Figure 4.17: Learned convolution kernels on (a) SOC dataset, and (b) Newspaper dataset

4.3.5 Evaluation

Evaluation on SOC: On the SOC dataset, the image resolution is high, so we use a relatively large patch size of 48. With this patch size we obtain roughly 9×10^4 training patches, 3×10^4 validation patches, and 3×10^4 test patches. We show the experimental results and compare with previous approaches in Table 4.10. The proposed method achieved a higher LCC and SROCC than other competing methods.

We visualize the learned filters in the first convolution layers, in Figure 4.17(a). The learned filters do not show patterns immediately intuitive to human. Some patterns tend to change intensity gradually along a direction that may correspond to blurred character boundary, while most patterns seem to be of irregular structure. We believe this happens because that the filters are learned from locally normalized images instead of the original images.

Evaluation on Newspaper: The image size varies greatly in the Newspaper dataset. Images can be as small as 569×38, and the characters in the images

	BRISQUE-L	CORNIA	CORNIA-SF	CNN
LCC	0.722	**0.751**	0.735	0.731
SROCC	0.709	0.725	0.708	**0.726**

Table 4.11: Median LCC and SROCC over 100 random sampling experiments on Newspaper dataset

appears smaller than those in SOC. Thus, a smaller patch size of 32 is used for this dataset. Approximately 6×10^4 patches exist for training, and 2×10^4 each for validation and test. Table 4.11 shows the experimental results on the Newspaper dataset. Our method achieved similar performance compared to the state of the art. All competing methods show a decrease of performance on this dataset. The major noise present in the Newspaper dataset arises from eroded/broken strokes, which are likely part of the document instead of being introduced by the imaging process. This inherent distortion mainly contains the semantics, thus it may not be measured well by those methods that typically focus on statistics of low level features.

We also visualize the first convolution layer filters learned on the Newspaper dataset in Figure 4.17(b). It is difficult to see obvious structures from the learned filters.

Chapter 5: Summary of Contributions

We have addressed three problems: text line detection, document image categorization, and No-Reference image quality assessment.

5.1 Handwritten and Scene Text Line Detection

In Chapter 2 we addressed the problem of text line detection in handwritten documents and natural images. Our contributions include:

1. The development of a graph-based method for text line segmentation which uses image-patches in the training data to obtain the contextual evidence needed for detecting text lines in a new document images.

2. A novel Sequential Gap Significance feature combined with a support vector machine to make accurate predictions of the number of clusters in Normalized Cuts.

3. A general framework to detect multi-oriented scene text lines with less dependency on font or language. Instead of focusing on the strong detection and filtering approaches, this new framework explores the Higher-Order Correlation Clustering to exploit elongated nature of text lines and to consider long

range dependencies.

4. Texton-based method for classification of candidate text lines on the region level, which provides stable statistics and achieves state of the art performance in comparison with competing methods that aim at detecting multi-oriented and multi-language text.

5.2 Document Image Categorization

In Chapter 3 we addressed the problem of document image categorization. Our contributions include:

1. A general approach for document image genre classification using CNNs. The introduction of CNNs for document image genre classification largely reduces the needs of hand-crafted features or domain knowledge. The hierarchical nature of document images make CNNs an excellent solution for their classification.

2. The demonstration of our CNN's state of the art performance on standard benchmarks. With very little feature engineering and limited training data, our method outperform the previous best method by a large margin.

5.3 No-Reference Image Quality Assessment

In Chapter 4 we addressed the problem of NR-IQA. Our contributions include:

1. A CNN based approach to general-purpose NR-IQA. CNN has not previously

been applied to general-purpose NR-IQA, because the original CNN is not designed to capture image quality features. The difference between NR-IQA and object recognition makes the application of CNN nonintuitive. Our method bridges the gap between NR-IQA and CNN, and it opens the door to a broad range of deep learning methods.

2. Demonstration of local image quality estimation ability. Previous approaches typically accumulate features over the entire image to obtain statistics for estimating overall quality, and have rarely shown the ability to estimate local quality. Our method can estimate quality on small patchs, which is important for many image enhancement applications such as denoising or reconstruction.

3. Simultaneous distortion identification and quality estimation. We developed a multi-task CNN that accurately identifies distortion and estimates quality on small image patches. Our multi-task CNN achieves a complete solution for NR-IQA problems and outperforms previous methods by a large margin on distortion identification.

4. State of the art performance on natural images and document images. Our CNN based approaches obtain high correlations with human opinions on natural image benchmarks (LIVE, TID2008, and CSIQ), and with OCR accuracy on the document image benchmarks (SOC and Newspaper).

5.4 List of Publications

1. L. Kang, P. Ye, L. Yi and D. Doermann. "Simultaneous Estimation of Image Quality and Distortion via Multi-task Convolutional Neural Networks." Intl. Conf. on Image Processing (ICIP 2015), September 2015.

2. F. Wang, L. Kang and Y. Li. "Sketch-based 3D Shape Retrieval using Convolutional Neural Networks." IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2015), June 2015.

3. L. Kang, P. Ye, Y. Li and D. Doermann."Convolutional Neural Networks for No-Reference Image Quality Assessment." IEEE Conf. On Computer Vision and Pattern Recognition (CVPR 2014), pp. 1733-1740, June 2014.

4. L. Kang, Y. Li and D. Doermann. "Orientation Robust Text Line Detection in Natural Images." IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2014), pp. 4034-4041, June 2014.

5. L. Kang, J. Kumar, P. Ye, Y. Li and D. Doermann. "Convolutional Neural Networks for Document Image Classification." Intl. Conf. on Pattern Recognition (ICPR 2014), pp. 3168-3172, August 2014.

6. L. Kang, P. Ye, Y. Li and D. Doermann. "A Deep Learning Approach to Document Image Quality Assessment." Intl. Conf. on Image Processing (ICIP 2014), pp. 2570-2574, October 2014.

7. P. Ye, J. Kumar, L. Kang and D. Doermann. "Real-time No-Reference Image

Quality Assessment based on Filter Learning." IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2013), pp. 987-994, June 2013.

8. P. Ye, J. Kumar, L. Kang and D. Doermann. "Unsupervised Feature Learning Framework for No-reference Image Quality Assessment." IEEE. Conf. on Computer Vision and Pattern Recognition (CVPR 2012), pp. 1098-1105, June 2012.

9. L. Kang, D. Doermann, H. Cao, R. Prasad and P. Natarajan. "Local Segmentation of Touching Characters using Contour based Shape Decomposition." Document Analysis Systems (DAS 2012), pp. 460-464, March 2012.

10. L. Kang, J. Kumar, P. Ye and D. Doermann. "Learning Text-line Segmentation using Codebooks and Graph Partitioning." Intl. Conf. on Frontiers in Handwriting Recognition (ICFHR 2012), pp. 63-68, September 2012.

11. J. Kumar, L. Kang, D. Doermann and Wael Abd-Almageed. "Segmentation of Handwritten Textlines in Presence of Touching Components." Intl. Conf. on Document Analysis and Recognition (ICDAR 2011), pp. 109-113, September 2011.

12. L. Kang and D. Doermann. "Template based Segmentation of Touching Components in Handwritten Text Lines." 11th Intl. Conf. on Document Analysis and Recognition (ICDAR 2011), pp. 569-573, September 2011.

Bibliography

[1] Zhidong Lu, R. Schwartz, P. Natarajan, I. Bazzi, and J. Makhoul. Advances in the bbn byblos ocr system. In *Document Analysis and Recognition, 1999. ICDAR '99. Proceedings of the Fifth International Conference on*, pages 337 –340, Sept. 1999.

[2] R.M. Haralick. Document image understanding: geometric and logical layout. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on*, pages 385 –390, June 1994.

[3] S. Messelodi and C.M. Modena. Automatic identification and skew estimation of text lines in real scene images. *Pattern Recognition*, 32(5):791 – 810, 1999.

[4] David Doermann, Elena Zotkina, and Huiping Li. Gedi - a groundtruthing environment for document images. In *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*, DAS '10, New York, NY, USA, 2010. ACM.

[5] Marc Petter, Victor Fragoso, Matthew Turk, and Charles Baur. Automatic text detection for mobile augmented reality translation. In *ICCV Workshops*, pages 48–55, 2011.

[6] A. Dengel and F. Dubiel. Clustering and classification of document structure-a machine learning approach. In *International Conference on Document Analysis and Recognition*, volume 2, pages 587–591, 1995.

[7] Nawei Chen and Dorothea Blostein. A survey of document image classification: problem statement, classifier architecture and performance evaluation. *International Journal Document Analysis Recognition*, 10(1):1–16, 2007.

[8] Yungcheol Byun and Yillbyung Lee. Form classification using dp matching. In *Proceedings of the 2000 ACM Symposium on Applied Computing - Volume 1*, pages 1–4, 2000.

[9] A.D. Bagdanov and M. Worring. Fine-grained document genre classification using first order random graphs. In *Document Analysis and Recognition, 2001. Proceedings. Sixth International Conference on*, pages 79–83, 2001.

[10] J. Kumar, P. Ye, and D. Doermann. Learning Document Structure for Retrieval and Classification. In *International Conference on Pattern Recognition*, pages 1558–1561, 2012.

[11] Siyuan Chen, Yuan He, Jun Sun, and Satoshi Naoi. Structured document classification by matching local salient features. In *International Conference on Pattern Recognition*, pages 653–656, 2012.

[12] J. Kumar, P. Ye, and D. Doermann. Structural similarity for document image classification and retrieval. *Pattern Recognition Letters*, 2013.

[13] Yann Lecun, Lon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *IEEE proceedings*, pages 2278–2324, 1998.

[14] H. R. Sheikh, A. C. Bovik, and G. de Veciana. An information fidelity criterion for image quality assessment using natural scene statistics. *IEEE Transactions on Image Processing*, 14(12):2117–2128, Dec. 2005.

[15] Lin Zhang, D. Zhang, Xuanqin Mou, and D. Zhang. FSIM: A feature similarity index for image quality assessment. *IEEE Transactions on Image Processing*, 20(8):2378–2386, 2011.

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

[17] Z. Wang and Q. Li. Information content weighting for perceptual image quality assessment. *IEEE Transactions on Image Processing*, 20(5):1185–1198, 2011.

[18] Wufeng Xue, Lei Zhang, Xuanqin Mou, and Alan C. Bovik. Gradient magnitude similarity deviation: A highly efficient perceptual image quality index. *arXiv:1308.3052*, 2013.

[19] Anush Krishna Moorthy and Alan Conrad Bovik. Blind image quality assessment: From natural scene statistics to perceptual quality. *Image Processing, IEEE Transactions on*, 20(12):3350–3364, 2011.

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

[21] Peng Ye, Jayant Kumar, Le Kang, and David Doermann. Unsupervised Feature Learning Framework for No-reference Image Quality Assessment. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1098–1105, 2012.

[22] Peng Ye, Jayant Kumar, Le Kang, and David Doermann. Real-time no-reference image quality assessment based on filter learning. In *Intl. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 987–994, 2013.

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

[24] Dan Claudiu Ciresan, Ueli Meier, and Jürgen Schmidhuber. Multi-column deep neural networks for image classification. In *Computer Vision and Pattern Recognition*, pages 3642–3649, 2012.

[25] Koray Kavukcuoglu, Pierre Sermanet, Y-Lan Boureau, Karol Gregor, Michaël Mathieu, and Yann LeCun. Learning convolutional feature hierachies for visual recognition. In *Advances in Neural Information Processing Systems (NIPS)*, 2010.

[26] Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2012.

[27] Yi Li, Yefeng Zheng, D. Doermann, S. Jaeger, and Yi Li. Script-independent text line segmentation in freestyle handwritten documents. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(8):1313 –1329, Aug. 2008.

[28] J. Kumar, Le Kang, D. Doermann, and W. Abd-Almageed. Segmentation of handwritten textlines in presence of touching components. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pages 109 –113, Sept. 2011.

[29] U. Pal and S. Datta. Segmentation of bangla unconstrained handwritten text. In *Document Analysis and Recognition, 2003. Proceedings. Seventh International Conference on*, pages 1128 – 1132, Aug. 2003.

[30] V. Manohar, S.N. Vitaladevuni, Huaigu Cao, R. Prasad, and P. Natarajan. Graph clustering-based ensemble method for handwritten text line segmentation. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pages 574 –578, Sept. 2011.

[31] B. Gatos, N. Stamatopoulos, and G. Louloudis. Icdar 2009 handwriting segmentation contest. In *Document Analysis and Recognition, 2009. ICDAR '09. 10th International Conference on*, pages 1393 –1397, July 2009.

[32] N. Sebe, I. Cohen, A. Garg, and T Huang. *Machine Learning in Computer Vision*. Springer, 2005.

[33] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):888–905, August 2000.

[34] Jayant Kumar, Wael Abd-Almageed, Le Kang, and David Doermann. Handwritten arabic text line segmentation using affinity propagation. In *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*, DAS '10, pages 135–142, New York, NY, USA, 2010. ACM.

[35] Zhixin Shi, S. Setlur, and V. Govindaraju. A steerable directional local profile technique for extraction of handwritten arabic text lines. In *Document Analysis and Recognition, 2009. ICDAR '09. 10th International Conference on*, pages 176 –180, July 2009.

[36] Fei Yin and Cheng-Lin Liu. Handwritten chinese text line segmentation by clustering with distance metric learning. *Pattern Recognition*, 42(12):3146 – 3157, 2009. ¡ce:title¿New Frontiers in Handwriting Recognition¡/ce:title¿.

[37] F. Jurie and B. Triggs. Creating efficient codebooks for visual recognition. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pages 604 – 610, Oct. 2005.

[38] Peng Ye, Jayant Kumar, Le Kang, and David Doermann. Unsupervised feature learning framework for no-reference image quality assessment. In *Computer Vision and Pattern Recognition, 2012. CVPR 2012. IEEE Conference on*, page to appear, June 2012.

[39] Ulrike Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, Dec. 2007.

[40] William Dumouchel and Fanny O'Brien. Computing and graphics in statistics. chapter Integrating a robust option into a multiple regression computing environment, pages 41–48. Springer-Verlag New York, Inc., New York, NY, USA, 1991.

[41] Jiri Matas, Ondrej Chum, Martin Urban, and Tomás Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and vision computing*, 22(10):761–767, 2004.

[42] Sungwoong Kim, Sebastian Nowozin, Pushmeet Kohli, and Chang D. D. Yoo. Higher-order correlation clustering for image segmentation. In *NIPS*. 2011.

[43] I. Tsochantaridis, T. Joachims, T. Hofmann, Y. Altun, and Y. Singer. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6(9), 2005.

[44] Jérôme Malick, Janez Povh, Franz Rendl, and Angelika Wiegele. Regularization methods for semidefinite programming. *SIAM J. on Optimization*, 20(1):336–356, April 2009.

[45] Micha Elsner and Warren Schudy. Bounding and comparing methods for correlation clustering beyond ILP. In *In NAACL-HLT Workshop on Integer Linear Programming for Natural Language Processing (ILPNLP) 2009*, 2009.

[46] Cong Yao, Xiang Bai, Wenyu Liu, Yi Ma, and Zhuowen Tu. Detecting texts of arbitrary orientations in natural images. In *CVPR'12*, pages 1083–1090, 2012.

[47] L. Neumann and J. Matas. Real-time scene text localization and recognition. In *IEEE CVPR*, pages 3538–3545, 2012.

[48] A. Coates, B. Carpenter, C. Case, S. Satheesh, B. Suresh, Tao Wang, D.J. Wu, and A.Y. Ng. Text detection and character recognition in scene images with unsupervised feature learning. In *ICDAR 2011*, pages 440 –445, sept. 2011.

[49] Jung-Jin Lee, Pyoung-Hean Lee, Seong-Whan Lee, Alan L. Yuille, and Christof Koch. Adaboost for text detection in natural scene. In *ICDAR*, pages 429–434, 2011.

[50] Boris Epshtein, Eyal Ofek, and Yonatan Wexler. Detecting text in natural scenes with stroke width transform. In *IEEE CVPR*, pages 2963–2970, 2010.

[51] Kai Wang, Boris Babenko, and Serge Belongie. End-to-end scene text recognition. In *ICCV'11*. IEEE, 2011.

[52] Nikhil Bansal, Avrim Blum, and Shuchi Chawla. Correlation clustering. *Machine Learning*, 56(1-3):89–113, 2004.

[53] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[54] Ganzhao Yuan, Zhenjie Zhang, Bernard Ghanem, and Zhifeng Hao. Low-rank quadratic semidefinite programming. *Neurocomput.*, 106:51–60, April 2013.

[55] R. H. Tucuncu, K. C. Toh, and M. J. Todd. Solving semidefinite-quadratic-linear programs using sdpt3. *Mathematical Programming*, 95:189–217, 2003.

[56] Li Liu and P.W. Fieguth. Texture classification from random features. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(3):574–586, 2012.

[57] Chucai Yi and YingLi Tian. Text string detection from natural scenes by structure-based partition and grouping. *Image Processing, IEEE Transactions on*, 20(9):2594–2605, 2011.

[58] Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. A metric for distributions with applications to image databases. ICCV'98, 1998.

[59] Xiangrong Chen and Alan L Yuille. Detecting and reading text in natural scenes. In *CVPR'04*, volume 2, pages II–366. IEEE, 2004.

[60] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, pages 807–814, 2010.

[61] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. Arxiv, 2012.

[62] C. Shin and D. Doermann. Document image retrieval based on layout structural similarity. In *International Conference on Image Processing, Computer Vision and Pattern Recognition*, pages 606 – 612, 2006.

[63] K. Collins-Thompson and R. Nickolov. A clustering-based algorithm for automatic document separation. In *SIGIR Workshop on Information Retrieval and OCR: From Converting Content to Grasping, Meaning*, pages 1–8, 2002.

[64] G. Joutel, V. Eglin, S. Bres, and H. Emptoz. Curvelets based queries for cbir application in handwriting collections. In *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*, volume 2, pages 649–653, 2007.

[65] T. Kochi and T. Saitoh. User-defined template for identifying document type and extracting information from documents. In *International Conference on Document Analysis and Recognition*, pages 127–130, 1999.

[66] K. V. Umamaheswara Reddy and Venu Govindaraju. Form classification. In *SPIE Document recognition and Retreival*, volume 6815, pages 1–6, 2008.

[67] Christian Wallraven, Barbara Caputo, and Arnulf Graf. Recognition with local features: the kernel recipe. In *IEEE International Conference on Computer Vision*, pages 257–264, 2003.

[68] Pedro Quelhas, Florent Monay, J-M Odobez, Daniel Gatica-Perez, Tinne Tuytelaars, and Luc Van Gool. Modeling scenes with local descriptors and latent aspects. In *International Conference on Computer Vision*, volume 1, pages 883–890, 2005.

[69] E. Barbu, P. Héroux, S. Adam, and É. Trupin. Using bags of symbols for automatic indexing of graphical document image databases. *Ten Years Review and Future Perspectives: Graphics Recognition*, pages 195–205, 2006.

[70] J. Kumar, R. Prasad, H. Cao, W. Abd-Almageed, D. Doermann, and P. Natarajan. Shape Codebook based Handwritten and Machine Printed Text Zone Extraction. In *International Conference on Document Recognition and Retrieval*, pages 7874:1–8, 2011.

[71] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2169 – 2178, 2006.

[72] Jianchao Yang, Kai Yu, Yihong Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1794 –1801, 2009.

[73] Yi Yang and S. Newsam. Spatial pyramid co-occurrence for image classification. In *International Conference on Computer Vision*, pages 1465 –1472, 2011.

[74] D. Lewis, G. Agam, S. Argamon, O. Frieder, D. Grossman, and J. Heard. Building a test collection for complex document information processing. volume 2006, pages 665–666, 2006.

[75] D.L. Dimmick, M.D. Garris, and C.L. Wilson. Nist structured forms reference set of binary images (sfrs), 1991.

[76] Siyuan Chen, Yuan He, Jun Sun, and S. Naoi. Structured document classification by matching local salient features. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 653–656, 2012.

[77] James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. Theano: a CPU and GPU math expression compiler. In *Proceedings of the Python for Scientific Computing Conference (SciPy)*, Jun. 2010.

[78] Wufeng Xue, Lei Zhang, and Xuanqin Mou. Learning without human scores for blind image quality assessment. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 995–1002, 2013.

[79] Chaofeng Li, A.C. Bovik, and Xiaojun Wu. Blind image quality assessment using a general regression neural network. *IEEE Transactions on Neural Networks*, 22(5):793–799, 2011.

[80] A. Chetouani, A. Beghdadi, S. Chen, and G. Mostafaoui. A novel free reference image quality metric using neural network approach. In *Int. Workshop Video Process. Qual. Metrics Cons. Electron.*, pages 1–4, Jan. 2010.

[81] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(8):1798–1828, 2013.

[82] H. R. Sheikh, Z. Wang, L. Cormack, and A. C. Bovik. LIVE image quality assessment database release 2. Online, http://live.ece.utexas.edu/research/quality.

[83] N. Ponomarenko, V. Lukin, A. Zelensky, K. Egiazarian, M. Carli, and F. Battisti. TID2008 - a database for evaluation of full-reference visual quality assessment metrics. *Advances of Modern Radio Electronics*, 10:30–45, 2009.

[84] Rich Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.

[85] Michele A Saad, Alan C Bovik, and Christophe Charrier. Blind image quality assessment: A natural scene statistics approach in the DCT domain. *Image Processing, IEEE Transactions on*, 21(8):3339–3352, 2012.

[86] Yaser S Abu-Mostafa. Learning from hints in neural networks. *Journal of complexity*, 6(2):192–198, 1990.

[87] Eric C. Larson and Damon M. Chandler. Most apparent distortion: Full-reference image quality assessment and the role of strategy. *Journal of Electronic Imaging*, 19(1):011006–011006–21, 2010.

[88] Peng Ye and David Doermann. Document Image Quality Assessment: A Brief Survey. *2013 12th International Conference on Document Analysis and Recognition*, pages 723–727, August 2013.

[89] Jayant Kumar, Raja Bala, Hengzhou Ding, and Phillip Emmett. Mobile Video Capture of Multi-page Documents. *2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 35–40, June 2013.

[90] L.R. Blando, J. Kanai, and T.a. Nartker. Prediction of OCR accuracy using simple image features. *Proceedings of 3rd International Conference on Document Analysis and Recognition*, 1:319–322, 1995.

[91] Andrea Souza, Mohamed Cheriet, Satoshi Naoi, and Ching Y Suen. Automatic filter selection using image quality assessment. In *Document Analysis and Recognition, 2003. Proceedings. Seventh International Conference on*, pages 508–512. IEEE, 2003.

[92] Michael Cannon, Patrick Kelly, and Judith Hochberg. Quality assessment and restoration of typewritten document images. *International Journal on Document Analysis and Recognition*, 2(2-3):80–89, December 1999.

[93] Jayant Kumar, Francine Chen, and David Doermann. Sharpness estimation for document and scene images. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 3292–3295. IEEE, 2012.

[94] Peng Ye and David Doermann. Learning features for predicting ocr accuracy. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 3204–3207. IEEE, 2012.

[95] A threshold selection method from gray-level histograms. *Systems, Man and Cybernetics, IEEE Transactions on*, 9(1):62–66, Jan 1979.

[96] Jayant Kumar, Peng Ye, and David Doermann. A Dataset for Quality Assessment of Camera Captured Document Images. *International Workshop on Camera-Based Document Analysis and Recognition (CBDAR)*, pages 39–44, 2013.

[97] Ray Smith. Isri-ocr evaluation tool. `http://code.google.com/p/isri-ocr-evaluation-tools/`.