# Complex-Wavelet Structural Similarity Based Image Classification

by

Yang Gao

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2012

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

**Abstract**

Complex wavelet structural similarity (CW-SSIM) index has been recognized as a novel image similarity measure of broad potential applications due to its robustness to small geometric distortions such as translation, scaling and rotation of images. Nevertheless, how to make the best use of it in image classification problems has not been deeply investigated. In this study, we introduce a series of novel image classification algorithms based on CW-SSIM and use handwritten digit and face image recognition as examples for demonstration, including CW-SSIM based nearest neighbor method, CW-SSIM based $k$ means method, CW-SSIM based support vector machine method (SVM) and CW-SSIM based SVM using affinity propagation. Among the proposed approaches, the best compromise between accuracy and complexity is obtained by the CW-SSIM support vector machine algorithm, which combines an unsupervised clustering method to divide the training images into clusters with representative images and a supervised learning method based on support vector machines to maximize the classification accuracy. Our experiments show that such a conceptually simple image classification method, which does not involve any registration, intensity normalization or sophisticated feature extraction processes, and does not rely on any modeling of the image patterns or distortion processes, achieves competitive performance with reduced computational cost.

**Acknowledgements**

It is a pleasure to thank many people who made this thesis possible.

It is difficult to overstate my gratitude to my supervisor, Dr. Zhou Wang. Without his inspiration, his patience and his profundity of knowledge, I never could have finished this work. Throughout my Master program at University of Waterloo, his provided guidance on my research, study and grad student life. I would have been lost without him. I especially want to thank Dr. Wang for his understanding and encouragement when my family came up with unexpected misfortune. It was the hardest time in my life, and I will always remember his support.

I wish to thank all my lab buddies at Image and Vision Computation Lab. They made it a stimulating and fun environment to work. I am really happy in the time with you guys. In particular, I would like to thank Abdul Rehman and Jiheng Wang for their inspiring help and advices for the research project we worked on together.

Lastly but most importantly, I wish to thank my parents, Changze Gao and Jinli Wang, and my entire family. Their unconditional love can never be replaced by the others. Though my mother could not see the completion of my thesis and my graduation, I am sure she shares our joy and happiness in another world. To my father and mother I dedicated this thesis.

# Contents

# List of Tables

# List of Figures

# Acronyms

**$k$-NN** $k$-nearest neighbor. 1, 2, 29–31, 48, 52

**CART** Classification and Regression Trees. 24

**CW-SSIM** Complex Wavelet Structural Similarity Index. 2, 3, 26–32, 34–39, 42, 43, 46, 48, 52, 53

**DMOS** Differential Mean Opinion Score. 24

**FLD** Fisher's Linear Discriminant. 10

**HVS** Human Visual System. 2, 22, 24, 26

**LDA** Linear Discriminant Analysis. 9, 10

**MARS** Multivariate Adaptive Regression Splines. 24

**MLP** Multi Layer Perception. 22

**MOS** Mean Opinion Score. 20, 22

**MSE** Mean Square Error. 2, 17, 27, 28, 39

**PCA** Principle Component Analysis. 1, 24, 25

**PSNR** Peak Signal to Noise Ratio. 24

**QDF** Quadratic Discriminant Function. 1

**RDA** Regularized Discriminant Analysis. 1

**SFS** Sequential Forward Selections. 24

**SLP** Single Layer Perception. 20

**SSIM** Structural Similarity Index. 2, 17, 20, 22, 24, 26

**SVM** Support Vector Machine. 1–3, 9, 10, 13, 24, 25, 35, 36, 42, 46, 48, 52

**TPS** Thin Plate Spline. 16

# Chapter 1

# Introduction

Image classification is a common problem in a broad range of applications. The majority of existing image classification systems contain a "feature extraction" stage as a pre-classification step. These features are typically local or global structural descriptors of the image. The subsequent classification step then works in the feature space, where a large number of classifiers may be employed, ranging from simple $k$-nearest neighbor ($k$-NN) method [5] to more advanced approaches such as affinity propogation [6], Regularized Discriminant Analysis (RDA) [7], Principle Component Analysis (PCA) mixture model [8], Quadratic Discriminant Function (QDF) [9], and kernel-based Support Vector Machine (SVM) [10] and kernel PCA methods [11]. The performance of these image classification systems is largely constrained by the extracted features, which need to be selected with great care, because "a classifier is only as good as its features". For example, since images or objects are often shifted, scaled and rotated, it is desirable to define (or design) the features so that they are invariant or robust to these changes [12]. There are also powerful machine learning algorithms, such as artificial neural networks [13] and convolutionary neural networks [14, 15], that can be employed to automatically "discover" good features from a large number of training images, where feature discovery is left to a "black box" that may be obscure and difficult to understand in intuitive ways. A limitation of these feature-based approaches is that the features are tuned to specific classification problems and are weak in their generalization capability. As a result, the features may have to undergo a new phase of design, training or selection when images with different shapes and structures are to be classified.

A different type of image classification methods are based on template matching, where the similarities between a test image and a set of templates are evaluated and used to determine the class label without employing any specific structural features of images. These approaches are conceptually simple and are often with strong generalization ability. However, the effectiveness of these methods rely heavily on the *image similarity measure*

being employed.

Recently, there has been significant progress in the design of image similarity measures [16]. In particular, the Structural Similarity Index (SSIM) [17] has been found to be a much better measure than the widely used Mean Square Error (MSE) in predicting perceptual image quality, where the similarity between a distorted and a perfect-quality reference images is used as an indicator of the quality of the distorted image. The philosophy behind SSIM is to distinguish between structural and non-structural distortions and treat them unequally, which is presumably what the Human Visual System (HVS) would do.

Despite the superior performance of SSIM over MSE, both of them are very sensitive to geometric image distortions such as small scaling, rotation, and translation. In image classification tasks, however, resistance to these distortions is crucial because it is a common practice that images are not perfectly aligned to each other before a similarity measure is computed. In order to remove this "defect" from SSIM while maintaining its advantages, the Complex Wavelet Structural Similarity Index (CW-SSIM) index was proposed [18], which is based on the correlations of phase patterns measured in the complex wavelet transform domain. The construction of CW-SSIM has some interesting connections with several computational models that account for a variety of biological vision behaviors. These models include: 1) the involvement of bandpass visual channels in image pattern recognition tasks [19]; 2) the representation of phase information in primary visual cortex using quadrature pairs of localized bandpass filters [20]; 3) the computation of complex-valued product in visual cortex [21]; 4) the computation of local energy (using sums of squared responses of quadrature-pair filters) by complex cells in visual cortex [22]; and 5) the divisive normalization of filter responses (using summed energy of neighboring filter responses) in both visual and auditory neurons [23, 24]. CW-SSIM has been shown to be a useful measure in a series of applications, including image quality assessment [25], line-drawing comparison [25], segmentation comparison [25], range-based face recognition [26] and palmprint recognition [27]. However, its use in image classification problems has not been deeply exploited [28].

In this study, we investigate CW-SSIM as a novel image classification tool in the context of handwritten digit classification and human face image classification. The robustness of CW-SSIM against small geometric distortions allows us to avoid extracting any structural features that are insensitive to these distortions or employing any preprocessing methods such as deskewing, spatial shift, scaling and rotation. A series of CW-SSIM based classification methods are introduced, including CW-SSIM $k$-NN, CW-SSIM weighted $k$-NN, CW-SSIM $k$-means, CW-SSIM Affinity Propogation and CW-SSIM SVM. Among them, CW-SSIM SVM achieves the best balance between classification accuracy and computational complexity, and is divided into two stages. In the first stage, an unsupervised clustering method is employed to divide the training images into clusters, each of which is associated with a representative image. In the second stage, a supervised learning method

based on SVM is used to maximize the classification accuracy. The performance improvement of CW-SSIM SVM is achieved with reduced computational complexity.

# Chapter 2

# Background

## 2.1 Image Classification

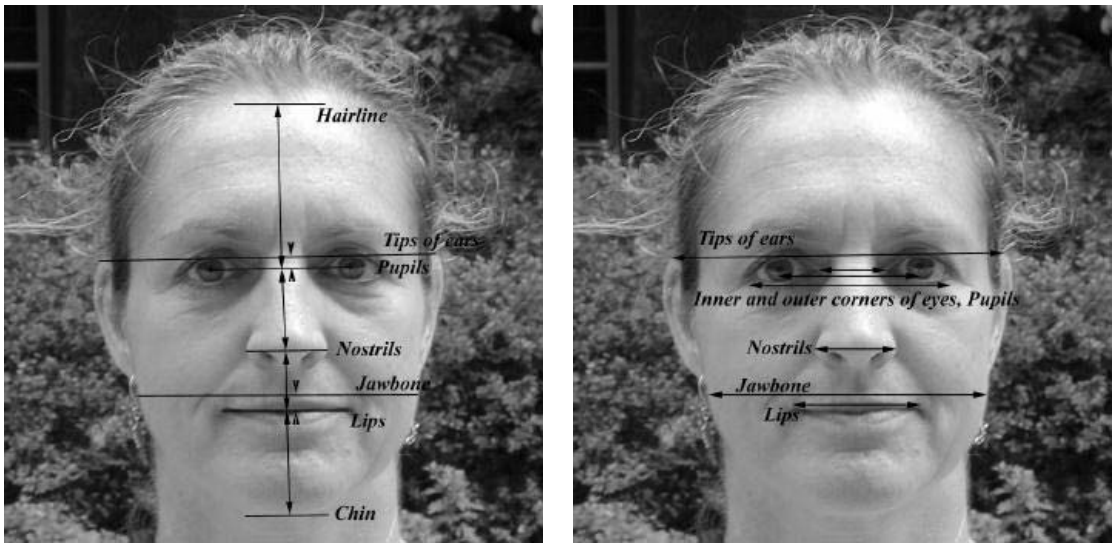
Image classification and recognition has long been researched since the beginning of the image analysis field. They lies in the "high end" of image processing world and involves the most complicated and diverse algorithms. The definition of classification is to use certain criteria to identify or distinguish different population of objects that may appear in images. One of the core component of image classification is to establish the criteria, which can vary widely in form and sophistication, ranging from example images of presumably prototypical representatives of each class, to numeric parameters from measurement, to syntactical descriptions of key features. A good example of image recognition and classification is finding a face in an image or matching that face to a specific individual, using statistical tools and numeric values. As recognition and classification can function at many different levels, a great many of specifically designed algorithms were developed to meet the need from particular levels in this field.

In genereal, computeres are sometimes better than humans at classification tasks, or at least faster, because they are not distracted by random variations in non-critical parameters and can extract meaningful statistical behavior to isolate groups. Sometimes these groups are meaningful, but in other cases they may not correspond to the intended classes. In contrary, people are much better and faster at recognition than are computers in most cases, because they can detect few critical factors that they have learned which will provide identification of familiar objects, and almost certainly do not depend on the same criteria as the statistical analysis of measurement data. Great effort has been made in the area of image recognition to simulate or approximate natural human vision but yet cannot achieve satisfactory performance on account of computer's numeric nature. But in case of classification, computer has the potential to outperform human, and thus classification

can be an ideal place to develop new imaging algorithms.

Classification techiniques in each subfields greatly differ from each other and involve an extremely broad range of complexities, from reading the bar codes to face or palmprint classification. According to the classification criteria they use, the algorithms can be categorized into appearance based methods using representative images, and fearure based methods, involves specialized definition of feature and sophisticated feature extraction procedures.

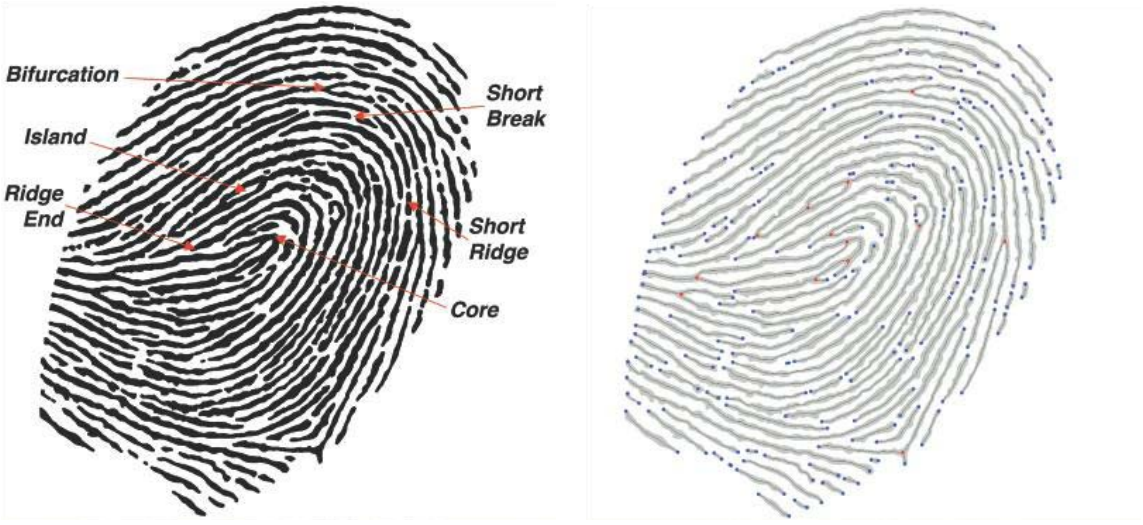
### 2.1.1 Feature Based Approach




(a) (b)

Figure 2.1: A human face labeled with several of the principal vertical and horizontal dimensions used in facial identification. From [1].

An extremely wide range of applications involve the concept of features. For instance, in some cases the target object are relatively simple and can be fully represented by one or several two-dimensional images. While in some other cases the target in the three-dimensional scenes may appear in a wide variety of presentation format as in the natural world, which is hard to "understand" for computers. Examples include automatic navigation and robotics, in which computers need to extract surface information and model the object behavior in two-dimensional images to reconstruct three-dimensional objects. The topics and goals discussed here are much more limited: to allow the image processing method to be able to recognize discrete features in essentially two dimensional scenes. If




(a) (b)

Figure 2.2: A finger print image, (a) classification of minutiae; (b) Skeleton of the image, with the ends and nodes marked. From [1].

the objects are three dimensional and can appear in different orientations, then each different two-dimensional view may be considered as a different target object. An example is that two dimensional projections of the same human face but in very different angles may not share resemblance as computer sees.

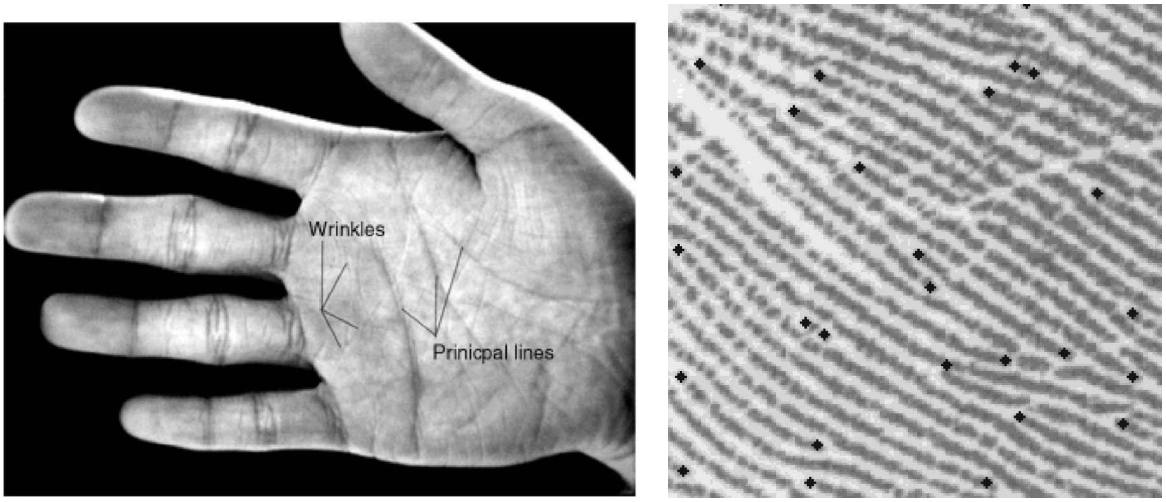
There are increasing needs for human face matching nowadays. Facial recognition and classification has become an important technique in many applications, such as screening surveillance videos for the faces of known individuals. Various features has been developed for face images. One successful approach uses ratios of vertical and horizontal distances between selected landmarks, as indicated in Figure 2.1. The attractive property of this "ratio of distance" feature is that it is relatively insensitive to the orientation of the face with respect to the camera. But when applying to face image data in real application, some of the landmarks may be obscured in any particular view, and the use of multiple combinations of dimensions needs to compensate accordingly.

It is worth noting that fully automatic identification is not the primary objective for this face image classification method. It is more likely to create a vector in a high dimensional space using the various ratios that can select a fixed but small number of the most similar faces on file, which are then presented to a human for comparison and matching. This is the same screening approach used in many other applications, some of which are described below.

"Minutiae" is a widely used feature in fingerprint classification. It can be defined as the

location and orientation of details such as branches or ends in the friction ridge pattern, as shown in Figure 2.2. Generally a small number of "minutiae" need to be extracted in fingerprint classification algorithms. They may be located either manually or by image processing. The coordinates of these features form a vector to select a group of the most similar stored prints, which a human then views.

For facial image classification, the method based on ratios of dimensions assumes that the structural information of natural human faces can be perceived by computer. A suitable database of images needs to be established, and certain machine learning methods might be involved to support such assumption. Dimensional ratios are chosen so that they are resistant to distortions. Sometimes slight changes in dimension may alter the appearance entirely, but the dimensions chosen are difficult to alter in a feature based system. This makes the ratio feature universal and stable for the classification processes. Computer is an essential part of both the facial classification and fingerprint classification process, but human interaction may still be brought into the ultimate decision making step (and sometimes the measurements as well).

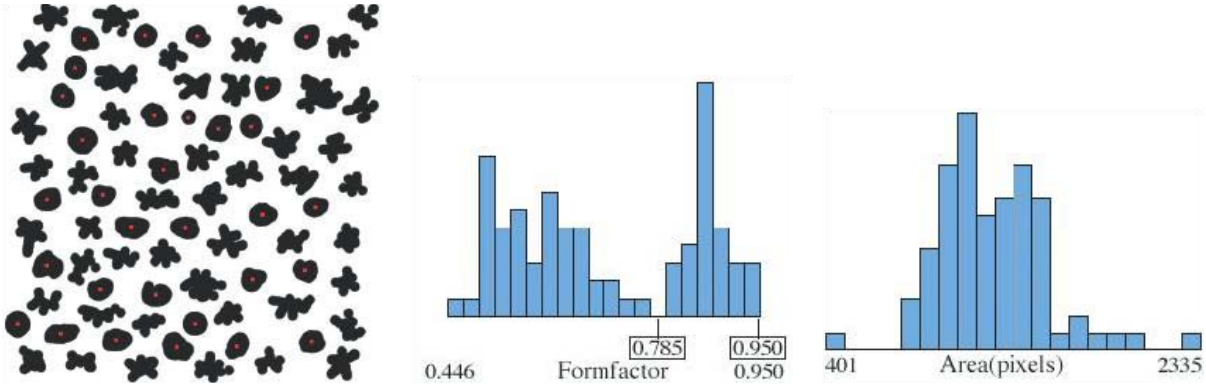



(a) (b)

Figure 2.3: Palmprint feature definitions. (a) principal lines and wrinkles; (b) a small region of a palmprint and the minutiae extracted from it. From [2]

The definition of feature and its extraction procedure may differ even for target objects of the same kind. In palmprint classification, there are many unique features in a palmprint image that can be used for personal identification. Principal lines, wrinkles, ridges, minutiae points (as shown in Figure 2.3), singular points, and the statistical property of palmprint texture are regarded as useful features for palmprint representation [29]. Various features can be extracted at different image resolutions. For features such as minutiae

points, ridges, and singular points, a high-resolution image, with at least 400 dpi, is required for feature extraction. In contrary, features like principal lines and wrinkles, can be obtained from a low-resolution palmprint image. As the definition of feature varies, the entire classification schemes for palmprint images may also distinct from each other.




(a) (b) (c)

Figure 2.4: (a) Some hand-drawn samples; (b) their histograms for formfactor; (c) The formfactor histogram shows two separate classes; the features with values greater than 0.785 are marked with red dots and are visually "rounder" than the others. The area histogram shows only a single grouping and does not distinguish two groups of features. [1]

To provide an intuitive perception of feature based classification scheme, a very simple example is shown in Fig. 2.4. which represents a very large percentage of practical applications that uses feature-specific parametric descriptions as the numeric measurement of image objects. In Fig. 2.4, the features can be grouped into two classes based on their shape ("round" and "spiky"), where we can use a simple shape descriptor:

$$\text{Form Factor} = \frac{4\pi * \text{Area}}{\text{Perimeter}^2}. \tag{2.1}$$

According to the distribution after applying form factor, all the shape objects are well-separated into two populations. Setting a threshold between the two groups can separate them into the respective classes. In Figure 2.4, the "rounder" features have been identified with a red mark.

Note that other measurement parameters such as area do not distinguish between the groups 2.4. Finding a single parametric feature description that can be successfully used to separate classes is not always possible, and when it is, selecting the best one from the many possible candidates by trial and error can be a time-consuming process. Statistical analysis programs can assist in the process.

The various classes are distinct in this simple example, so that drawing "decision lines" between them is straightforward once the range of values for the different classes has been established. This is most commonly done by measuring actual samples. Rather than just measuring "typical" specimens. Drawing the "decision lines" involves various classification methods including complex machine learning process (such as LDA, SVM, etc.), which will be discussed in the next section.

### 2.1.2 Classification Methods

In machine learning and patterm recognition, classification refers to an algorithmic procedure for assigning a given piece of input data into one of a given number of categories. Various classification methods have been developed, including statistical methods, neural networks, support vector machines, etc.

#### Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) [30] is a widely used statistical classifier. Its main idea is to find a linear combination of features which produce the greatest separation between two or more classes of target objects. Thus the resulting combination of features can be used as a linear classifier, or at least reduce the dimensionality of target objects before later classification.

Consider a set of training samples $\mathbf{x}$, each labeled as belonging to known class $y$. The classifier need to find a good predictor to label any observation with the same distribution as the given $\mathbf{x}$. As a statistical approcah, LDA first assumes that the conditional probability density function $p(\mathbf{x}|y=0)$ and $p(\mathbf{x}|y=1)$ are both normally distributed, with mean and covariance parameters $(\boldsymbol{\mu}_0, \Sigma_0)$ and $(\boldsymbol{\mu}_1, \Sigma_1)$, respectively. Under this assumption, LDA is the Bayes optimal solution which predict the observation as belonging to the second class if the ratio of the log-likelihoods is below certain threshold $T$,

$$(\mathbf{x}-\boldsymbol{\mu}_0)^T\Sigma_0^{-1}(\mathbf{x}-\boldsymbol{\mu}_0) + log|\Sigma_0| - (\mathbf{x}-\boldsymbol{\mu}_1)^T\Sigma_1^{-1}(\mathbf{x}-\boldsymbol{\mu}_1) - log|\Sigma_1| < T. \tag{2.2}$$

If the class covariance are identical, $\Sigma_0 = \Sigma_1 = Sigma$, several terms can be canceled and the classification criterion becomes:

$$\mathbf{w}\cdot\mathbf{x} < c, \text{where } \mathbf{w} = \Sigma_{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0), \tag{2.3}$$

where $c$ stands for a certain threshold constant. The result above shows that the prediction of the belonging of input $\mathbf{x}$ only depends on a linear combination of the known observations.

Fisher's Linear Discriminant (FLD) [30] is closely related to LDA, and they are often used interchangeably. Still a slight difference exists. The normally distributed classes or equal class covariance are not part of FLD's assumption. Suppose two classes of observations with means $\boldsymbol{\mu}_0, \boldsymbol{\mu}_1$ and covariance $\Sigma_0, \Sigma_1$. Fisher defined the separation between two distributions to be the ratio of the variance between the classes to the variance within the classes:

$$S = \frac{\sigma^2_{between}}{\sigma^2_{within}} = \frac{(\mathbf{w} \cdot \boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^2}{\mathbf{w}^T \Sigma_1 \mathbf{w} + \mathbf{w}^T \Sigma_0 \mathbf{w}} = \frac{(\mathbf{w} \cdot (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0))^2}{\mathbf{w}^T (\Sigma_1 + \Sigma_0) \mathbf{w}}. \tag{2.4}$$

This definition shows that we can acquire the maximum separation when:

$$\mathbf{w} = (\Sigma_0 + \Sigma_1)^{-1} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0). \tag{2.5}$$

This equation is equivalent to LDA if the normality and equal covariance assumptions are satisfied.

### Support Vector Machine

Support Vector Machine (SVM) is a classifier with discriminant function being the weighted combination of kernel functions over all training samples. As a comprehensive learning tool and linear classifier, SVM is receiving wide attention from researchers and have shown superior performance in pattern recognition and many other areas. It was first introduced in [31] as learning machines with capacity control for regression and binary classification problems.

SVM is an effective binary linear classifier. Given a set of training samples, each labeled as belonging to one of two categories, the SVM training algorithm models the sample objects behavior and build decision criteria that assigns new examples into one category or the other. The key component of SVM modeling is to discover a certain mapping pattern for the training samples, to a set of points in multi-dimensional space as their representation, so that it can construct an optimal separating hyperplane in this high-dimensional feature space (as shown in Figure 2.5). The computation of this hyperplane relies on the maximization of the margin. After modeling, new test examples can then be mapped into the same space and thus labeled as belonging to that category.

In the training stage of SVM, for a set of data of $N$ pairs $(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N)$, with $x_i \in \mathbf{R}^p$ and $y_i \in \{-1, 1\}$. The seperating hyperplane between class 1 and class $-1$ can be defined as:

$$\{x : f(x) = x^T \beta + \beta_0 = 0\}, \tag{2.6}$$

where $\beta$ is a unit vector: $\|\beta\| = 1$. $f(x)$ in Eq. 2.6 gives the signed distance from a point $x$ to the hyperplane $f(x) = x^T \beta + \beta_0 = 0$.

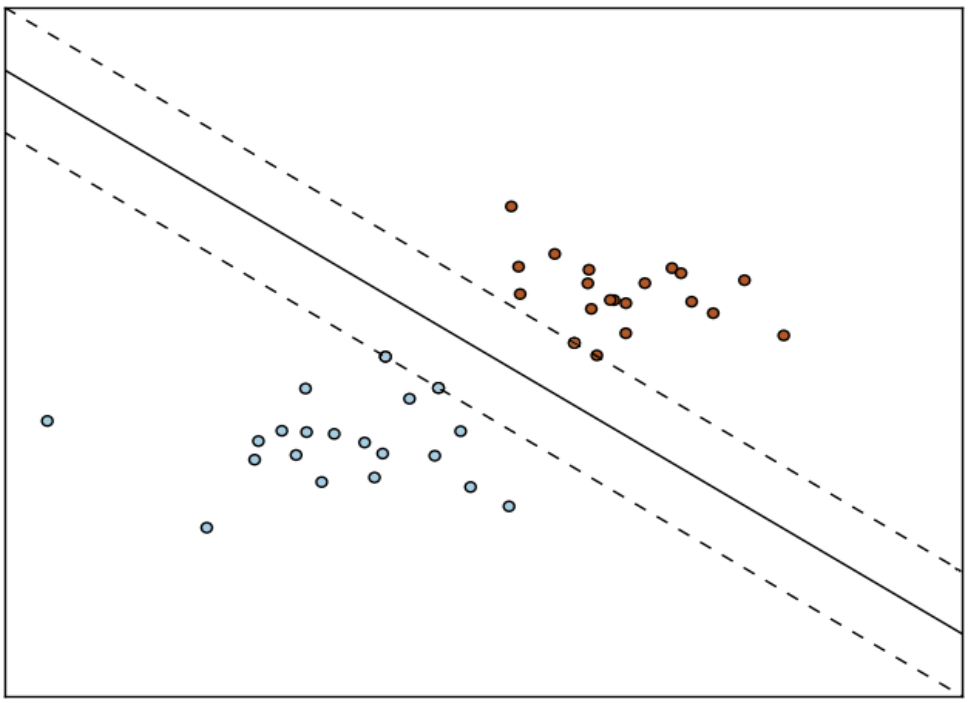


Figure 2.5: Maximum-margin hyperplane and margins for an SVM trained with samples from two classes. Samples on the margin are called the support vectors.

Assuming the data set is seperable into two classes, we can always find a function $f(x) = x^T\beta + \beta_0$ with $y_i f(x_i) > 0$, $\forall i$. Therefore it becomes possible to find the optimal hyperplane that creates the biggest margin between the training points for classes 1 and $-1$. The optimization problem

$$\begin{aligned} &\max_{\beta,\beta_0,\|\beta\|=1} \quad M \\ &\text{subject to } y_i(x_i^T\beta + \beta_0) \geq M, i = 1, \cdots, N, \end{aligned} \tag{2.7}$$

captures this concept. This problem can be rephrased as

$$\begin{aligned} &\max_{\beta,\beta_0} \quad \|\beta\| \\ &\text{subject to } y_i(x_i^T\beta + \beta_0) \geq 1, i = 1, \cdots, N, \end{aligned} \tag{2.8}$$

Class overlap is common in feature space. The slack variables can be defined as $\xi = (\xi_1, \xi_2, \cdots, \xi_N)$. Then the constraint in Eq. 2.7 can be written as

$$y_i(x_i^T\beta + \beta_0) \geq M(1 - \xi_i), \tag{2.9}$$

$\forall i, \xi_i \geq 0, \sum_{i=1}^N \leq$ constant. Then Eq. 2.8 is equivalent to:

$$\begin{aligned} &\max_{\beta,\beta_0} \quad \|\beta\| \\ &\text{subject to } y_i(x_i^T\beta + \beta_0) \geq 1 - \xi_i, \xi_i \geq 0, \forall i, \sum_{i=1}^N \leq \text{constant} \end{aligned} \tag{2.10}$$

This is the usual way the support vector classifier is defined for the nonseparable case. The problem is quadratic with linear inequality constraints, hence it is a convex optimization problem. It is computationaly convenient to re-express Eq. 2.10 in the equivalent form:

$$\begin{aligned} &\min_{\beta,\beta_0} \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^N \xi_i \\ &\text{subject to } \xi_i \geq 0, y_i(x_i^T\beta + \beta_0) \geq 1 - \xi_i, \forall i, \end{aligned} \tag{2.11}$$

The Lagrange function is:

$$L_P = \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^N \xi_i - \sum_{i=1}^N \alpha_i[y_i(x_i^T\beta + \beta_0) - (1 - \xi_i)] - \sum_{i=1}^N \mu_i\xi_i \tag{2.12}$$

which we minimize w.r.t $\beta$, $\beta_0$ and $\xi_i$. Its dual objective function is

$$L_D = \sum_{i=1}^N \alpha_i - \frac{1}{2}\sum_{i=1}^N\sum_{i'=1}^N \alpha_i\alpha_{i'}y_iy_{i'}x_i^Tx_{i'}. \tag{2.13}$$

We maximize $L_D$ subject to $0 \leq \alpha_i \leq C$ and $\sum_{i=1}^{N} \alpha_i y_i = 0$.

The linear boundaries in the input feature space can be found by the support vector classifier described above. As with other linear methods, we can make the procedure more flexible by enlarging the feature space using basis expansions such as polynomials or splines. In most cases in the enlarged space, the linear boudaries drawn by classifier can achieve better training-class separation. These boundaries project back into the original space and translate to nonlinear boundaries. Once the basis functions $h_m(x), m = 1, \cdots, M$ are selected, the procedure is the same as before. We fit the SV classifier using input features $h(x_i) = (h_1(x_i), h_2(x_i), \cdots, h_M(x))$, $i = 1, \cdots, N$, and produce the (nonlinear) function $\hat{f}(x) = h(x)^T \hat{\beta} + \hat{\beta}_0$. Thus the Lagrange dual function Eq.2.13 has the form

$$L_D = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{i'=1}^{N} \alpha_i \alpha_{i'} y_i y_{i'} \langle h(x_i), h(x_{i'}) \rangle. \tag{2.14}$$

In fact, we need not specify the transformation $h(x)$ at all, but require only knowledge of the kernel function

$$K(x, x') = \langle h(x_i), h(x_{i'}) \rangle \tag{2.15}$$

that computes inner products in the transformed space. $K$ should be a symmetric positive (semi-) definite function. Three popular choices for $K$ in the SVM literature are

$$\begin{aligned} d\text{th-Degree polynomial}: \; & K(x, x') = (1 + \langle x, x' \rangle)^d, \\ \text{Radial basis}: \; & K(x, x') = \exp(-\gamma \|x - x'\|)^2, \\ \text{Neural network}: \; & K(x, x') = \tanh(\kappa_1 \langle x, x' \rangle + \kappa_2). \end{aligned} \tag{2.16}$$

In real applications, the target objects may have more than two classes, which lead us to multi-class SVM. There are two common methods to solve a multi-class problem with binary classifiers such as SVMs: one-against-all and one-against-one. In the one-against-all scheme, a classifier is built for each class and assigned to the separation of this class from the others. For the one-against-one method, a classifier is built for every pair of classes to separate the classes two by two. Another approach to the recognition of $n$ different digits is to use a single $n$-class SVM instead of $n$ binary SVM subclassifiers with the one-against-all method, thus solving a single constrained optimization problem. Multi-class SVMs have been studied by different authors. But this method is not very popular in digit recognition applications and did not yield better performances than other classifiers. In [32], a multi-class SVM was compared to a group of binary SVMs on the USPS datasets. This multi-class SVM gave lower accuracy rates than the common methods.

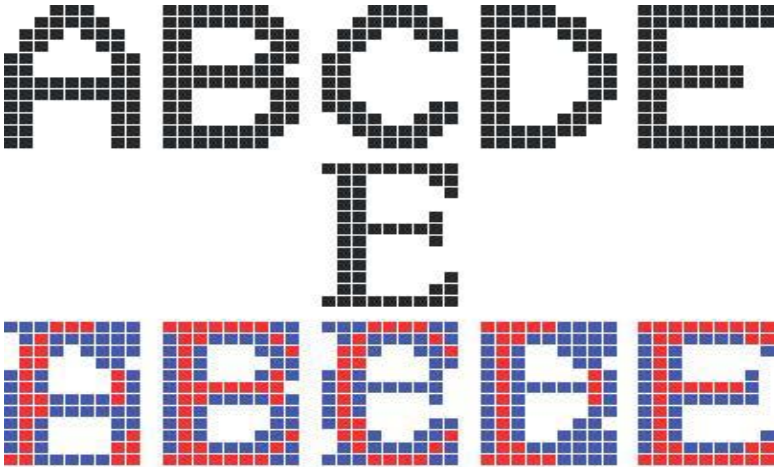


Figure 2.6: Matching each of the five letter templates (top row) with a target letter (center) produces a net score: number of matched pixels minus the unmatched ones. From [1].

### 2.1.3 Appearance Based Methods

Instead of extracting certain geographical or statistical information from target images and then using them as a complete representation of the original image objects, the image object itself can be treated as the classification criteria. The example in Figure 2.6 shows an application of template matching to the letters A through E. The template consisting of black pixels that cover the shape of each target character is stored in memory. Each letter to be recognized is compared with all of the stored templates to count the number of pixels that match and the number which do not. The template that gives the highest net score (number of matches minus number of misses) is selected as the identification of the character.

The above example is merely a rough illustration of the concept of template matching. This similarity measure (number of matches pixel minus number of misses) between image objects is oversimple, which requires fixed size, location, and orientation, and in a special font designed to make them easily distinguishable. In real application, a comprehensive and balanced similarity measurement between image templates and image objects is a necessary premise.

For handwritten digit and character recognition, in most cases the target objects are binary instead of grayscale. Many algorithms have been developed for digit classification with widely varying features and classification schemes. Many of them are based on self-defined similarity measurements and template-involved schemes.

Shape context is a template based approach for image classification [3]. It provides a similarity measurement between shapes of binary image. The main idea is to pick $n$ points

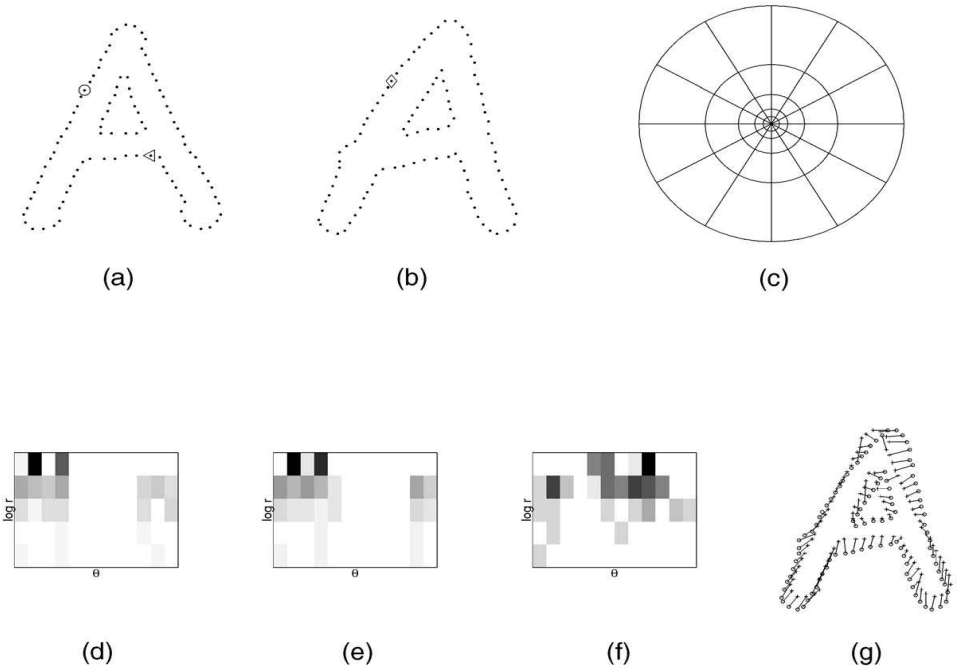



Figure 2.7: Shape context computation and matching. (a) and (b) Sampled edge points of two shapes. (c) diagram of log-polar histogram bins used in computing the shape contexts (5 bins for $\log r$ and 12 bins for $\theta$). (d), (e), and (f) Example shape contexts for reference samples marked by $\circ, \diamond, \triangleleft$ in (a) and (b). Each shape context is a log-polar histogram of the coordinates of the rest of the point set measured using the reference point as the origin (dark means large values). (g) Correspondece found using graph matching. From [3].

on the contours of a binary shape. For each point $p_i$ on the shape, consider the $n-1$ vectors obtained by connecting $p_i$ to all other points. The set of all these vectors is a rich description of the shape localized at that point. This distribution over relative positions is considered a robust, compact, and highly discriminative descriptor. A coarse histogram $h_i$ of the relative coordinates of the remaining $n-1$ points,

$$h_i(k) = \#\{q \neq p_i : (q - p_i) \in bin(k)\}. \tag{2.17}$$

This histogram is defined to be the shape context of $p_i$. An example is shown in Figure 2.7. For a point $p_i$ on the first shape and a point $q_j$ on the second shape. The cost of matching these two points can be computed as:

$$C_{ij} = C(p_i, q_j) = \frac{1}{2}\sum_{k=1}^{K}\frac{[h_i(k) - h_j(k)]^2}{h_i(k) + h_j(k)}, \tag{2.18}$$

The next step is to find a one-to-one matching that matches all pairs of point $p_i$ on the first shape and $q_j$ on the second shape that minimizes the total cost of matching,

$$H(\pi) = \sum_i C(p_i, q_{\pi(i)}). \tag{2.19}$$

The optimized matching can be found with the help of bipartite graph matching. Given the correspondence between sets of points on the two shapes a transformation $T : \Re^2 \rightarrow \Re^2$ can be estimated to map any point from one shape to the other. Several tools can be used to compute this transformation including affine model and Thin Plate Spline (TPS). A shape distance between two shapes can then be computed based on the acquired transformation model. It is defined as a weighted summation of three terms: shape contex distance, image apperance distance and bending energy. This shape distance is considered as the similarity measure between shape or binary image objects, and is applied to different applications for classification or retrieval purpose. Details of shape contex method can be found in [3].

The design of shape context based methods only focues on shape contour of binary images and does not apply on other "normal" images. This may lead our next goal to searching for a more general and efficient image similarity measure.

## 2.2 Image Similarity Measures

The performance of template matching-based image classification systems critically depends on the accuracy and robustness of the image similarity/dissimilarity measure being employed, which quantifies the closeness or departure in the image space between a query image and any selected image in the training database.

### 2.2.1 Mean Squared Error

The mean squared error (MSE) is the simplest and most widely used image dissimilarity measure [33]. For two $N$-pixel grayscale images $\mathbf{x}$ and $\mathbf{y}$ with intensity values $\{x_i|i = 1, ...N\}$ and $\{y_i|i = 1, ...N\}$, respectively, the MSE is calculated as

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(x_i - y_i)^2 . \tag{2.20}$$

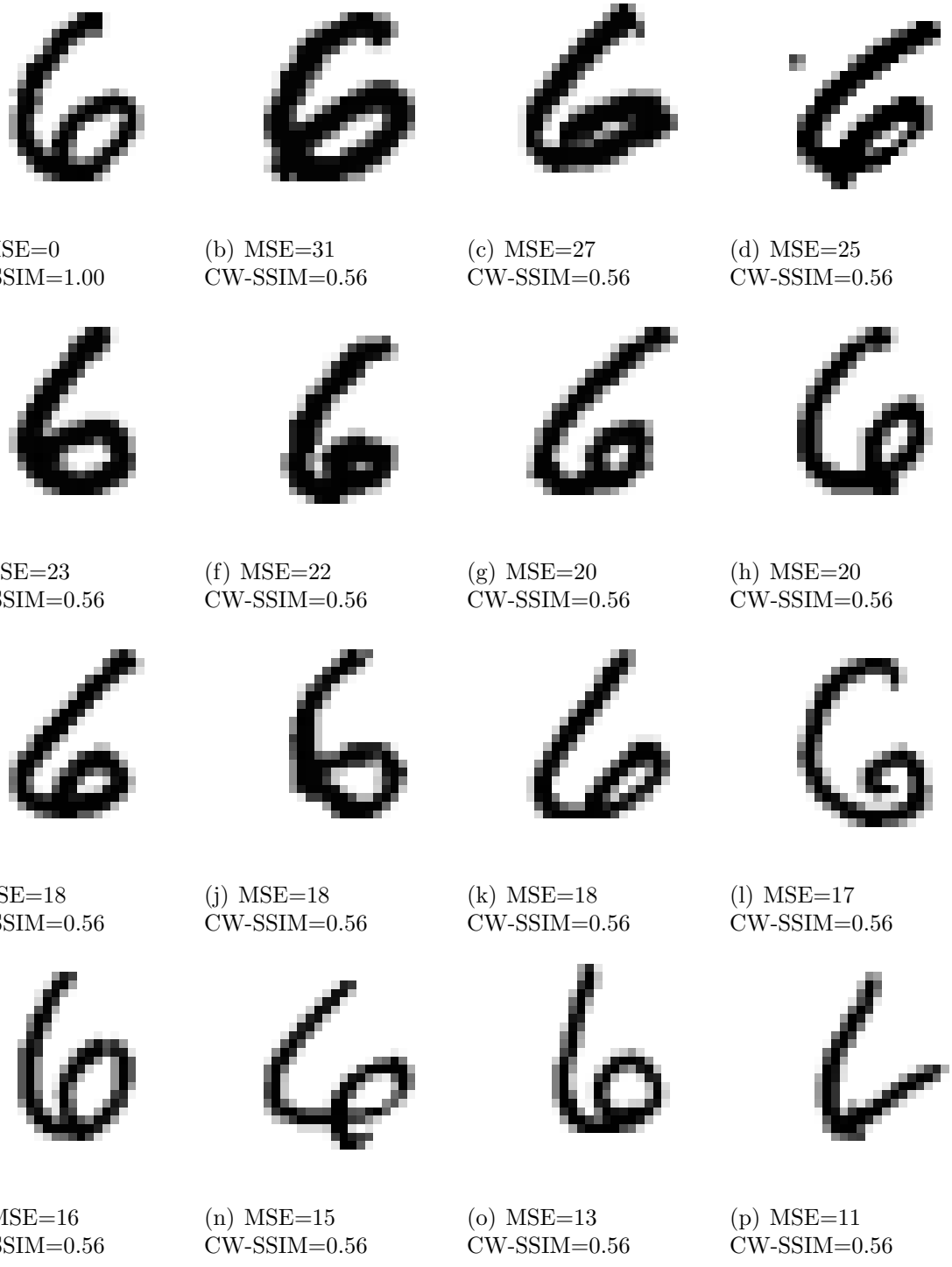
The MSE is easy to compute and has a number of desirable properties in real world applications, but it also suffers from several fundamental problems [33]. Consider the handwritten digits in Fig. 2.8, where image (a) is used as a reference and compared with every other image. Regarded as collections of pixel intensity values and compared using MSE, the images are very different. However, regarded as shapes/structures, they appear rather similar to a human observer. In such a situation, if we persist on using MSE, then we need to perform various preprocessing steps and coordinate transformations to align the image patterns beforehand [34]. However, such alignment methods are often unreliable and any mis-registration of the images may lead to erroneous results. Another example is given in Fig. 2.9, where shapes or structures between the reference image (a) and each of the other images are substantially different, but their MSE values remain the same. Therefore, in order to operationalize the notion of shape/structure similarity, with ultimate goal of using it as a basis of a robust recognition system, we need to replace MSE with a similarity measure that is based on similarity of shapes/structures between the images being compared.

### 2.2.2 Structural Similarity Indices

SSIM index is a method for measuring the similarity between two images. The SSIM index can be viewed as a quality measure of one of the images being compared, provided the other image is regarded as of perfect quality [17].

The basic spatial domain SSIM algorithm is based upon separated comparisons of local luminance, contrast and structure between an original and a distorted images. Given two local image patches $\mathbf{x} = \{x_i|i = 1, 2, \cdots, M\}$ and $\mathbf{y} = \{y_i|i = 1, \cdots, M\}$ extracted from the original and distorted images, respectively, the luminance, contrast and structural similarities between them are evaluated as

$$l(\mathbf{x}, \mathbf{y}) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \tag{2.21}$$



(a) MSE=0
CW-SSIM=1.00

(b) MSE=31
CW-SSIM=0.56

(c) MSE=27
CW-SSIM=0.56

(d) MSE=25
CW-SSIM=0.56

(e) MSE=23
CW-SSIM=0.56

(f) MSE=22
CW-SSIM=0.56

(g) MSE=20
CW-SSIM=0.56

(h) MSE=20
CW-SSIM=0.56

(i) MSE=18
CW-SSIM=0.56

(j) MSE=18
CW-SSIM=0.56

(k) MSE=18
CW-SSIM=0.56

(l) MSE=17
CW-SSIM=0.56

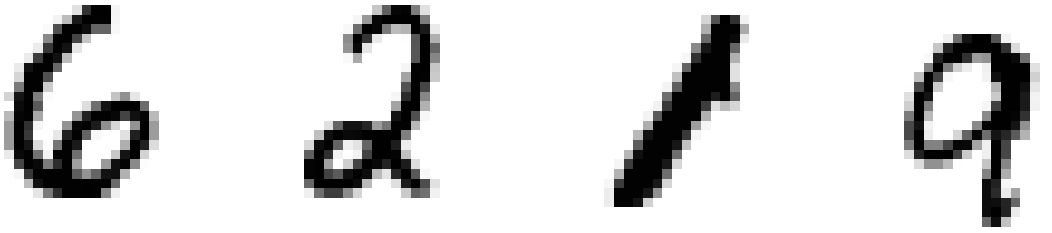
(m) MSE=16
CW-SSIM=0.56

(n) MSE=15
CW-SSIM=0.56

(o) MSE=13
CW-SSIM=0.56

(p) MSE=11
CW-SSIM=0.56

Figure 2.8: Comparison of image similarity measures MSE and CW-SSIM. (a): reference image; (b)-(p): test images with the same CW-SSIM but significantly different MSE values with respect to the reference.

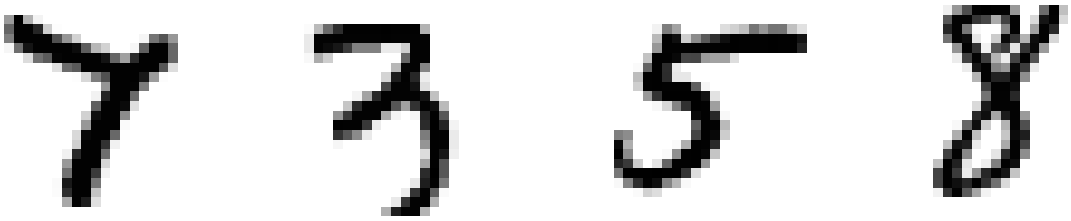


(a) MSE=0
CW-SSIM=1.00

(b) MSE=22
CW-SSIM=0.31

(c) MSE=22
CW-SSIM=0.29

(d) MSE=22
CW-SSIM=0.37

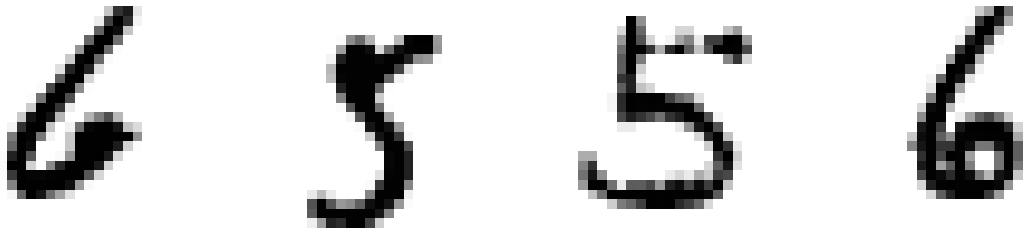


(e) MSE=22
CW-SSIM=0.32

(f) MSE=22
CW-SSIM=0.25

(g) MSE=22
CW-SSIM=0.37

(h) MSE=22
CW-SSIM=0.26

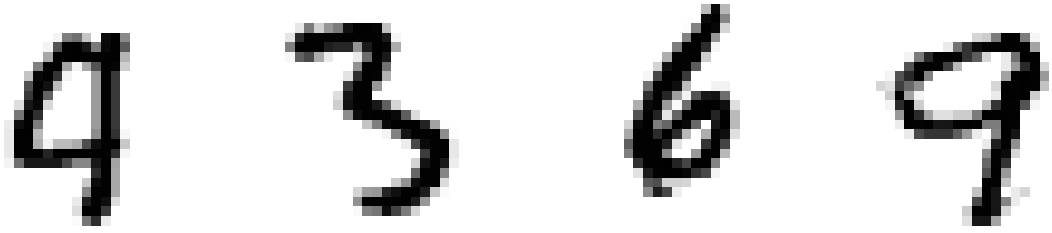


(i) MSE=22
CW-SSIM=0.46

(j) MSE=22
CW-SSIM=0.27

(k) MSE=22
CW-SSIM=0.29

(l) MSE=22
CW-SSIM=0.52



(m) MSE=22
CW-SSIM=0.40

(n) MSE=22
CW-SSIM=0.31

(o) MSE=22
CW-SSIM=0.52

(p) MSE=22
CW-SSIM=0.32

Figure 2.9: Comparison of image similarity measures MSE and CW-SSIM. (a): reference image; (b)-(p): test images with the same MSE but quite different CW-SSIM values with respect to the reference.

$$c(\mathbf{x}, \mathbf{y}) = \frac{2\sigma_{xy} + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \tag{2.22}$$

$$s(\mathbf{x}, \mathbf{y}) = \frac{\sigma_{xy} + C_3}{\sigma_x \sigma_y + C_3} \tag{2.23}$$

respectively. Here, $\mu_x$, $\sigma_x$ and $\sigma_{xy}$ represent the mean, standard deviation and cross-correlation evaluations, respectively. $C_1 = (K_1 L)^2$, $C_2 = (K_2 L)^2$, $C_3 = C_2/2$ are small constants that have been found to be useful in characterizing the saturation effects of the visual system at low luminance and contrast regions and stabilizing the performance of the measure when the denominators are close to zero. The local SSIM index is defined as the product of the three components, which gives

$$\text{SSIM}_{\text{local}} = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{2.24}$$

When this local measurement is applied to an entire image using a sliding window approach, an SSIM quality map is created. The overall SSIM value of the whole image is simply the average of the SSIM map.

Obviously, the most basic approach of applying SSIM into machine learning techniques is utilizing three components of SSIM, including the overall luminance comparison, contrast comparison and structure comparison, directly. Kung et al. combined the single-layer perceptron and SSIM to establish the new single-layer perceptron to predict the Mean Opinion Score (MOS) of human observers [35]. By the definition of SSIM, Eq. 2.25 is used and extend as Eq. 2.26.

$$\begin{aligned} \text{SSIM}(\mathbf{x}, \mathbf{y}) &= f(l(\mathbf{x}, \mathbf{y}), c(\mathbf{x}, \mathbf{y}), s(\mathbf{x}, \mathbf{y})) \\ &= l(\mathbf{x}, \mathbf{y})^\alpha \cdot c(\mathbf{x}, \mathbf{y})^\beta \cdot s(\mathbf{x}, \mathbf{y})^\gamma \end{aligned} \tag{2.25}$$

$$\begin{aligned} \log(\text{SSIM}(\mathbf{x}, \mathbf{y})) &= \alpha \log[l(\mathbf{x}, \mathbf{y})] + \beta \log[c(\mathbf{x}, \mathbf{y})] + \gamma \log[s(\mathbf{x}, \mathbf{y})] \\ &= \sum_{i=1}^{3} w_i x_i \end{aligned} \tag{2.26}$$

where $w_1 = \alpha$, $w_2 = \beta$, $w_3 = \gamma$, $x_1 = \log[l(\mathbf{x}, \mathbf{y})]$, $x_2 = \log[c(\mathbf{x}, \mathbf{y})]$ and $x_3 = \log[s(\mathbf{x}, \mathbf{y})]$. The proposed Single Layer Perception (SLP) model is shown in Fig. 2.10.

### 2.2.3 Other Image Similarity Measurements

#### Naive Models

The early works, especially before the invention of SSIM [17], were more concentrated on statistical analysis of images.

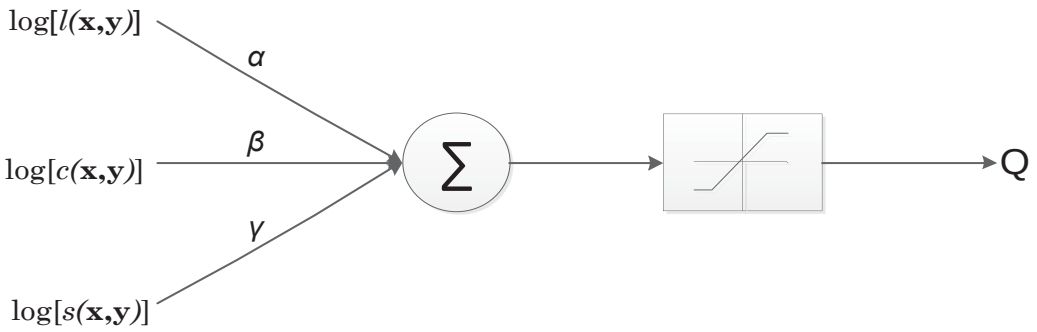



Figure 2.10: Combination of SSIM and Single-layer Perceptron

In [36], Carrai et al. proposed a model estimating the overall image quality by splitting pictures into $16 \times 16$ non-overlapping blocks. Each block is characterized by an objective metric set, which is the result of a feature-selection criterion based on statistical analysis. In detail, a quite large set of features characterizing an image has been selected:

1. First-order histogram descriptors: mean, standard deviation, skewness, kurtosis, energy and entropy.

2. Autocorrelation spread measures: profile spreads, cross-relation, second degree spread.

3. Co-occurrence matrix features: autocorrelation, covariance, inertia, absolute value, inverse difference, energy, energy coefficient (the ratio between the energy on the diagonal and energy), entropy, difference mean, difference variance, difference entropy.

In [37], a total of twenty-six image quality metrics/features are proposed. According to the type of information they use, these quality metrics are categorized into six groups:

1. Pixel difference-based measures such as mean square distortion: Mean square error, Mean absolute error, Modified infinity norm, $L * a * b * perceptualerror$, Neighborhood error and Multiresolution error;

2. Correlation-based measures, that is, correlation of pixels, or of the vector angular directions: Normalized cross correlation, Image fidelity, Czekonowski correlation, Mean angle similarity and Mean angle-magnitude similarity;

3. Edge-based measures, that is, displacement of edge positions or their consistency across resolution levels: Pratt edge measure and Edge stability measure;

4. Spectral distance-based measures, that is, the Fourier magnitude and/or phase spectral discrepancy on a block basis: Spectral phase error, Spectral phase-magnitude

error, Block spectral magnitude error, Block spectral phase error and Block spectral phase-magnitude error;

5. Context-based measures, that is, penalties based on various functionals of the multi-dimensional context probability: Rate distortion measure, Hellinger distance, Generalized Matusita distance and Spearman rank correlation;

6. HVS-based measures, that is, measures either based on the HVS-weighted spectral distortion measures or (dis)similarity criteria used in image base browsing functions: HVS absolute norm, HVS L2 norm, Browsing similarity and DC Tune.

In [4], Bouzerdoum et al. proposed a method to predict the MOS of human observers using an Multi Layer Perception (MLP). Here the MLP is designed to predict the image fidelity using a set of key features extracted from the reference and test images. The features are extracted from small blocks (say $8 \times 8$ or $16 \times 16$), and then are fed as inputs to the network, which estimates the image quality of the corresponding block. The overall image quality is estimated by averaging the estimated quality measures of the individual blocks. Using features extracted from small regions has the advantage that the network becomes independent of image size. The key features are based on the features of SSIM with some modifications. Six features, extracted from the original and test images, were used as inputs to the network: the two means, the two standard derivations, the covariance, and the mean-squared error between the test and reference blocks. They use an MLP architecture with 6 inputs, 6 neurons in the first hidden layer, 6 neurons in second hidden layer, and 1 output neuron as shown in Fig. 2.11. They used the logistic sigmoid activation function in the hidden layers and the linear activation function in the output layer.

### Singular Value Decomposition Models

We know that every real matrix $A$ can be decomposed into a product of three matrices $A = USV^T$, where $U$ and $V$ are orthogonal matrices, $U^TU = I$, $V^TV = I$, and $S = \text{diag}(s_1, s_2, \cdots)$. The diagonal entries of $S$ are called the singular values of $A$, the columns of $U$ are called the left singular vectors of $A$, and the columns of $V$ are called the right singular vectors of $A$. This decomposition is known as the $SVD$ of $A$. It is one of the most useful tools of linear algebra with several applications to multimedia including image compression and watermarking.

Applying SVD to an image matrix $X$ (size $r \times c$) yields the left singular vector matrix $U$, the right singular vector matrix $V$, and the diagonal matrix of singular values $\sigma$, i.e.,

$$\begin{aligned} U &= [u_1, u_2, \cdots, u_r] \\ V &= [v_1, v_2, \cdots, v_r] \\ \sigma &= \text{diag}(\sigma_1, \sigma_2, \cdots, \sigma_t) \end{aligned} \tag{2.27}$$

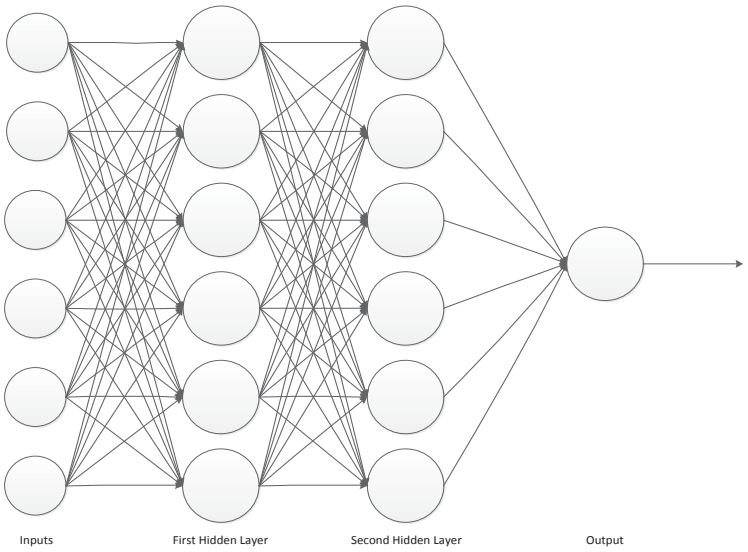



Figure 2.11: Multilayer perceptron used in [4]

where $u_i$ and $v_j$ are column vectors, whereas $\sigma_k$ is a singular value ($i = 1, 2, \cdots, r, j = 1, 2, \cdots c, k = 1, 2, \cdots t$, and $t = \min(r, c)$.). The singular values appear in descending order, i.e., $\sigma_1 > \sigma_2 > \cdots \sigma_t$.

In [38], a graphical measure and a numerical measure are proposed. The graphical measure is a bivariate measure that computes the distance between the singular values of the original image block and the singular values of the distorted image block

$$D_j = \text{Sqrt}\left[\sum_{i=1}^{n}(s_i - \hat{s}_i)\right] \tag{2.28}$$

where $s_i$ is the singular values of the original block, $\hat{s}_i$ is the singular values of the distorted block, and $n$ is the block size. If the image size is $k$, we have $(k/n) \times (k/n)$ blocks.

The numerical measure is derived from the graphical measure. It computes the global error expressed as a single numerical value depending on the distortion type

$$\text{M-SVD} = \frac{\sum_{j=1}^{(k/n)\times(k/n)} |D_i - D_{mid}|}{(\frac{k}{n}) \times (\frac{k}{n})} \tag{2.29}$$

where $D_{mid}$ represents the mid point of the sorted $D_i$s, $k$ is the image size, and $n$ is the block size.

In [39], Narwaria and Lin stated that the matrix $UV^T$ can be interpreted as the ensemble of the basis images, whereas the singular values $\sigma$ are the weights assigned to these basis images. The image structure can therefore be represented as

$$X_z = \sum_{i=1}^{z} u_i v_i^T \tag{2.30}$$

where $z(z \leq t)$ is the number of $u_i$ and $v_i$ pairs used. Each basis image (i.e., $u_i v_i^T$) specifies a layer of the image geometry, and the sum of these layers denotes the complete image structure. The first few singular vector pairs account for the major image structure, whereas the subsequent $u_i$ and $v_i$ account for the finer details in the image. As an increasing number of $u_i$ and $v_i$ pairs are used, the finer image structural details appear. $U$ and $V$ can therefore be used to represent the structural elements in images. They then use support vector regression (SVR) to map the high dimensional feature vector into a perceptual quality score by estimating the underlying complex relationship among the changes in $U$, $V$, $\sigma$, and perceptual quality score.

### Hybrid Models

The simplest hybrid approach is to directly combine different kinds of image similarity measures together. In [40], the two most widely used image quality metrics, the Peak Signal to Noise Ratio (PSNR) and the SSIM, are employed together to compute a weighted sum as the objective quality metrics. Then a neural network was used to obtain the mapping functions between the objective quality assessment metrics and subject quality assessment scores. The SVM was used to classify the images into different types which were accessed using different mapping functions.

In [41], a total of 27 full-reference image quality metrics (features) are investigated. These features fall into six categories, i.e., pixel-differences, image correlation, edge stability, spectral distance, models of the HVS, and structural similarity. The authors investigated the effectiveness of two statistical learning algorithms, namely, Classification and Regression Trees (CART) and Multivariate Adaptive Regression Splines (MARS), and one classical feature selection algorithm: Sequential Forward Selections (SFS). For CART and MARS, feature selection and estimator model optimization are performed jointly, thus the selected features maximize the correlation between estimated Differential Mean Opinion Score (DMOS) and subjective DMOS. The SFS algorithm, on the other hand, starts with the variable that is most correlated with subjective DMOS, and at each step adds a new variable that, together with the previous ones, most accurately predicts subjective DMOS via linear regression.

In [42], natural image statistics, distortion texture statistics, blur/noise statistics (patch PCA singularity, two-color prior based blur statistics and direct blur kernel and noise

estimation) are considered to be image quality features together. As many of our features are negative log histograms, the dimensionality of the features is extremely high. Therefore, PCA is performed firstly for each group of features to reduce its dimension, which is selected by cross validation. These low dimensional projections are then used to train an SVM regression model for each group of features and then three individual SVM regression outputs are used to compute a weighted linear combination of the the kernel SVM outputs.

# Chapter 3

# Methodology

## 3.1 Complex Wavelet Structural Similarity Indices

The SSIM index was originally proposed to predict human preference in evaluating image quality [17]. It provides good similarity measurement between image objects. Unfortunately, one negative feature of SSIM is its high-sensitivity to non-structural distortions, including translation, scaling, and rotation of images [18, 25]. This makes SSIM not an ideal tool for image classification in real application, where lies a strong possibility of geometric distortions of target image objects. Therefore, a new image similarity measurement, Complex Wavelet Structural Similarity Index (CW-SSIM) was developed [25].

CW-SSIM is an extension of the SSIM method to the complex wavelet domain. Its goal is to design a measurement that is insensitive to "non-structural" geometric distortions that are typically caused by nuisance factors, such as changes in lighting conditions and the relative movement of the image acquisition device, (which are very likely to occur in application data sets) rather than the actual changes in the structures of the objects. The initial inspiration of CW-SSIM is the recent discovery on HVS. In the last three decades, scientists have found that neurons in the primary visual cortex can be well-modeled using localized multi-scale bandpass oriented filters that decompose natural image signals into multiple visual channels [17]. One intereseting fact from related psychophysical research is, when performing image pattern recognition tasks, human tends to use the exact same set of visual channels [19].

The CW-SSIM measure was first proposed in [18, 25], which was built upon local phase measurements in complex wavelet transform domain. The underlying assumptions behind CW-SSIM are that local phase pattern contains more structural information than local magnitude, and non-structural image distortions such as small translations lead to

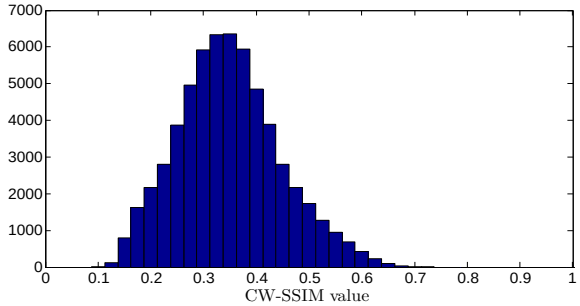
consistent phase shift within a group of neighboring wavelet coefficients. Therefore, CW-SSIM is designed to separate phase from magnitude distortion measurement and impose more penalty to inconsistent phase distortions.

Given two sets of complex wavelet coefficients $\mathbf{c}_x = \{c_{x,i} | i = 1, ..., M\}$ and $\mathbf{c}_y = \{c_{y,i} | i = 1, ..., M\}$ extracted at the same spatial location in the same wavelet subbands of the two images being compared, the local CW-SSIM index is defined as

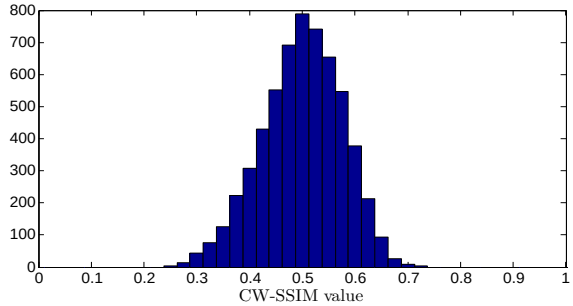
$$\tilde{S}(\mathbf{c}_x, \mathbf{c}_y) = \frac{2|\sum_{i=1}^{M} c_{x,i} c_{y,i}^*| + K}{\sum_{i=1}^{M} |c_{x,i}|^2 + \sum_{i=1}^{M} |c_{y,i}|^2 + K}. \tag{3.1}$$

where $c^*$ denotes the complex conjugate of $c$, and $K$ is a small positive stabilizing constant. The value of the index ranges from 0 to 1, where 1 implies no structural distortion (but still could have small spatial shift). The global CW-SSIM index $\tilde{S}(I_x, I_y)$ between two images $I_x$ and $I_y$ is calculated as the average of local CW-SSIM values computed with a sliding window running across the whole wavelet subband and then averaged over all subbands. It was demonstrated that CW-SSIM is simultaneously insensitive to luminance change, contrast change, and small geometric distortions such as translation, scaling and rotation [18, 25]. This makes CW-SSIM an ideal choice for image classification tasks because it is versatile and largely reduces the burden of preprocessing steps such as contrast and mean adjustment, pixel shifting, deskewing, zooming and scaling.

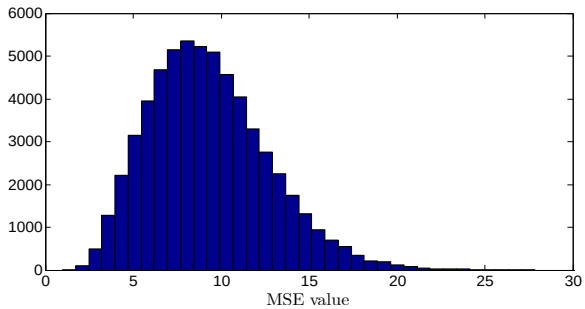
The performance of CW-SSIM is in clear contrast to that of MSE in the examples shown in Figs. 2.8 and 2.9. In Fig. 2.8, although there are notable variations in the spatial locations, orientations and thickness of the strokes in the test digit '6' images, they share similar structures, and consistently high CW-SSIM values are obtained, while there are significant differences in MSE values. In Fig. 2.9, the test images represent different digits and have very different structures, but they share the same MSE value with respect to the reference image (a), making it impossible to select the right images (i)(l)(o) out of all test images. By contrast, CW-SSIM easily distinguishes images (i)(l)(o) among all test images because its CW-SSIM value is clearly the highest. Still, there may be doubts that the cases shown in Figs. 2.8 and 2.9 are merely isolated exception. Thus to demonstrate CW-SSIM outperforms MSE as an image similarity measure for classification tasks, we randomly selected 10 digit images from MNIST database (consists of 60000 training digit images and 10000 testing digit images, each labeled as digit from 0 to 9), and calculated their CW-SSIM and MSE value with all the other 60000 samples. If the distribution of CW-SSIM or MSE value between the selected image and other images in the same class (i.e. a selected '6' with all '6's in the 60000 data set) is distinguishable with the distribution on the entire database, then we can show that this measurement can achieve better recognition result in stastistical means. The distributions and the means, medians, and variances are shown in Fig 3.1. It is clear that two CW-SSIM based distribution in (a) and (b) have



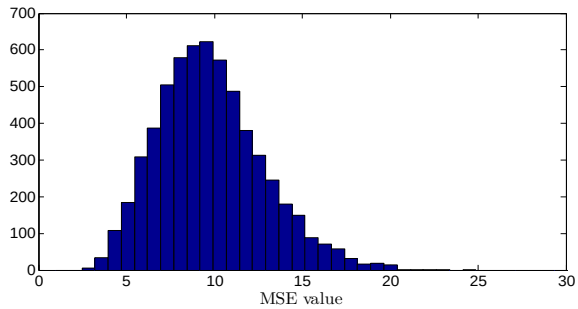

(a) CW-SSIM value with all 60000 images
Mean=0.35, Median=0.34, Variance=0.0095.




(b) CW-SSIM value with images in the same class
Mean=0.50, Median=0.50, Variance=0.0057.




(c) MSE value with all 60000 images
Mean=9.29, Median=8.96, Variance=11.40.




(d) MSE value with images in the same class
Mean=9.79, Median=9.49, Variance=9.15.

Figure 3.1: Comparison of CW-SSIM and MSE distributions

much more difference in their shape as well as in their parameters. The two MSE versions, comparatively, are nearly identical.

These illustrative examples demonstrate the power of CW-SSIM, which does not require any pre-registration process but still provides consistently reasonable comparisons. This inspires us to use CW-SSIM as the image similarity measure in image classification tasks.

## 3.2 CW-SSIM Based Image Classification Methods

Since CW-SSIM is a new similarity criterion introduced to the field, a series of image classification methods may be developed. In this section, we start from simple nearest neighbor algorithms to more sophisticated methods that lead to improved performance or reduced complexity. Here we present our algorithms with handwritten digit and face image recognition as our application in mind. However, since CW-SSIM is an efficient similarity measurement for most natural images, the CW-SSIM based methods are not confined to

the applications mentioned in this chapter. Th general approach should apply to many others as well.

### 3.2.1 CW-SSIM Based Nearest Neighbor Methods

Given a set of $N$ training images $\{I_i|i = 1, ..., N\}$ and their associated class labels $\{l_i|i = 1, ..., N\}$ (in the case of digit recognition, there are 10 classes, each representing a digit between 0 and 9, i.e., $l_i \in [0, 9]$), the most straightforward way of applying CW-SSIM for image classification is to find the image $I_j$ in the training image set that is "closest" to a test query image $I_q$ in CW-SSIM sense and use $l_j$ to label the query image. This CW-SSIM based nearest neighbor (CW-SSIM NN) classifier can be expressed as

$$l(I_q) = l_j, \text{ where } j = \arg\max_{i\in[1,N]} \tilde{S}(I_q, I_i)\,. \tag{3.2}$$

Indeed, due to the desirable properties possessed by CW-SSIM, this conceptually simple algorithm can achieve very good performance, especially when the training set is large, as will be shown in Section 4.1.

The CW-SSIM NN classifier can be easily generalized to a CW-SSIM $k-$NN classifier. Given the $k$ nearest neighbors of $I_q$ (denoted by $\{I^{(i)}|i = 1, ..., k\}$ and with class labels $\{l^{(i)}|i = 1, ..., k\}$) in the training image set in terms of CW-SSIM, we use a majority vote to decide on the class label assigned to $I_q$:

$$l(I_q) = \arg\max_{j\in[0,9]} \sum_{i=1}^{k} \delta(j, l^{(i)})\,. \tag{3.3}$$

where $\delta$ is a function such that $\delta(a, b) = 1$ if $a = b$ and $\delta(a, b) = 0$ otherwise.

The $k$-NN approach only considers the first $k$ nearest neighbors of the query image in the full training image set. This can be interpreted as weighting the full sorted image set by a hard-weighting function which has value 1 for the first $k$ images and 0 for the rest. It has been shown that soft-weighted $k-$NN can perform better than hard-weighted $k-$NN [43, 44]. Therefore, we extend our CW-SSIM $k$-NN classification approach to a CW-SSIM weighted $k$-NN classifier, where the weight $w_i$ is determined based on how close $I_q$ is to the $i$-th image in $k$ neighbors:

$$l(I_q) = \arg\max_{j\in[0,9]} \sum_{i=1}^{k} \tilde{S}(I_q, I^{(i)})\delta(j, l^{(i)})\,. \tag{3.4}$$

Since the basic $k$-NN approach only consider the $k$th neighborhood, and the similarity information between $I_q$, $I^{(i)}$ and all other images are abandoned in the process. We have

made a lot of effort on developing a number of $k$-NN variant algorithms which are described as follows.

As the $k$-NN algorithm just takes the first $k$ neighboring elements from the entire data set, and "trust" them blindly even some of them may be ambiguous or written in uncommon ways. These "weird" neighbors may interfere the decision model and lower the classification accuracy. Thus we introduce a "confidence" concept for each data in the data set. There are different ways to define the confidence, one of them is:

$$c(I_i) = \frac{\sum_{s=1}^{K} \delta(l_i, l^{(s)})}{K}. \tag{3.5}$$

where $K$ can be a predetermined number as the self-defined neighborhood for each $I_i$. With a higher ratio of neighbor images labeled as the same class, this data better represents its class and is hence more "confident". Another way to define confidence takes the distribution described in Fig. 3.1 into consideration. As stated previously in section 3.1, the two distributions of CW-SSIM values between target image and the entire data set, and between target and other image of the same class, can be used as an indicator of how well the target image can represent its own class. The numerical difference between them can be defined as "confidence" for each image object. The distributions are normalized to calculate the Kullback-Leibler Divergence (KLD),

$$c(I_i) = D_{KL}(P||Q) = \sum_i P(x) ln \frac{P(x)}{Q(x)}. \tag{3.6}$$

where $P(x)$ and $Q(x)$ denotes the two CW-SSIM value distribution. Regardless of how the "confidence" is defined for each object, the confidence-weighted $k$-NN algorithm can be written as:

$$l(I_q) = \arg\max_{j \in [0,9]} \sum_{i=1}^{k} c(I^{(i)}) \delta(j, l^{(i)}) \,. \tag{3.7}$$

Inspired by the shape context method [3], which describes the shapes using a histogram of relative position of points on the shape, and evaluate the difference between shapes based on matching cost of their histogram matrices, we also built a histogram-like matrix to describe the target image. Instead of storing the Euclidean distance and corresponding angles between different points on the binary shape as in shape context method, the image similarity distance, CW-SSIM values between target and other images in the data set, and labels of th images become the basic elements of our histogram. As shown in Fig 3.2, for target image $I_i$, the histogram,

$$h_i(k,t) = \#\{q \neq i, l_q = t : \tilde{S}(I_q, I_i) \in bin(k)\}, t \in [0,9]. \tag{3.8}$$

where $l_q$ denotes the class or label $I_q$ is assigned to. $Bin$s can be a set of uniformly divided intervals. Since CW-SSIM has a numerical range from 0 to 1, in our application $bin(k)$s are set to [0, 0.1], [0.1, 0.2], ..., [0.9, 1]. This martix carries over-detailed information describing the distance between the target image and other images from different classes. In the example shown in Fig. 3.2, according to the matrix the target image is most similar to other '6's since it is a '6' itself. Also, it is relatively less similar to '1's. This matrix actually describes the statistical feature of the target image. For image $I_i$ and $I_j$, the similarity distance between can be computed as the matching cost of two matrices:

$$\tilde{C}(I_i, I_j) = \frac{1}{2} \sum_{k=1}^{K} \sum_{t=0}^{9} \frac{[h_i(k,t) - h_j(k,t)]^2}{h_i(k,t) + h_j(k,t)}, \tag{3.9}$$

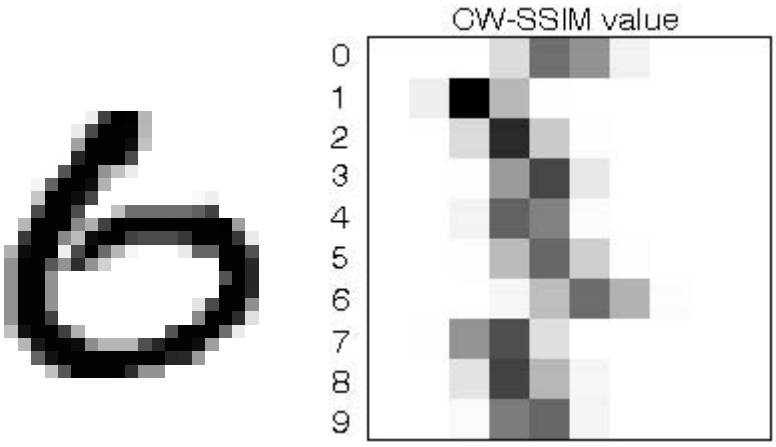
The matching cost $\tilde{C}$ can be used as the weight $w_i$ in a weighted $k$-NN classifier. In some cases, this measurement works better than the CW-SSIM weight described in Eq. 3.10, especially for some bizarre images. In Fig. 3.3, we computed the CW-SSIM matrix for the reference '3' image (a) and its first four neighboring images. The only other '3' in its neighborhood is image (c) in the third place. Image (e)(i) both have a higher CW-SSIM value and may lead to misclassification if we continue to use the weight defined in 3.10. However their CW-SSIM matrix are giving more satisfactory results that the matrix of image (c) have a much lower distance with the reference image than (e)(i). It will benefit the $k$-NN decision model if we set the weight of $k$-NN algorithm to $\tilde{C}$:

$$l(I_q) = \arg\max_{j \in [0,9]} \sum_{i=1}^{k} \tilde{C}(I_q, I^{(i)}) \delta(j, l^{(i)}) \,. \tag{3.10}$$

Though it may improve the performance of $k$-NN algorithm for some ambiguous image, we discovered that the improvement is not decisive but introducing matrix-based calculation will greatly increase the time consumption of our method. Taking the complexity into account, this CW-SSIM matrix-involved scheme should be adopted with care in practical applications.

### 3.2.2 CW-SSIM Based K-Means Method

A major problem with the nearest neighbor based methods described above is that they demand for CW-SSIM calculations of the query image with respect to all images in the training set. This could be computationally extremely expensive and thus prohibit its use in real-world applications. Classical methods for clustering such as $k-$means [45] have been used frequently in numerous vision applications [46]. Here we develop a CW-SSIM $k-$means clustering method to extract *typical* structures or *representatives* from




(a) Target image

(b) Proposed histogram for reference sample

Figure 3.2: Proposed histogram describing the target image.

the training image set and subsequently use these representatives to perform classification of the test query image with the help of nearest neighbor based methods.

$k-$means is an iterative algorithm that contains two steps in each iteration − updating the centroid for each cluster and updating cluster label for each sample image. Here we perform these two steps using CW-SSIM as the similarity criterion in replace of the typically used Euclidian distance. Given a set of $R$ training images $\{I_i | i = 1, ..., R\}$ that belong to a cluster, $C$, the centroid of the cluster is updated as

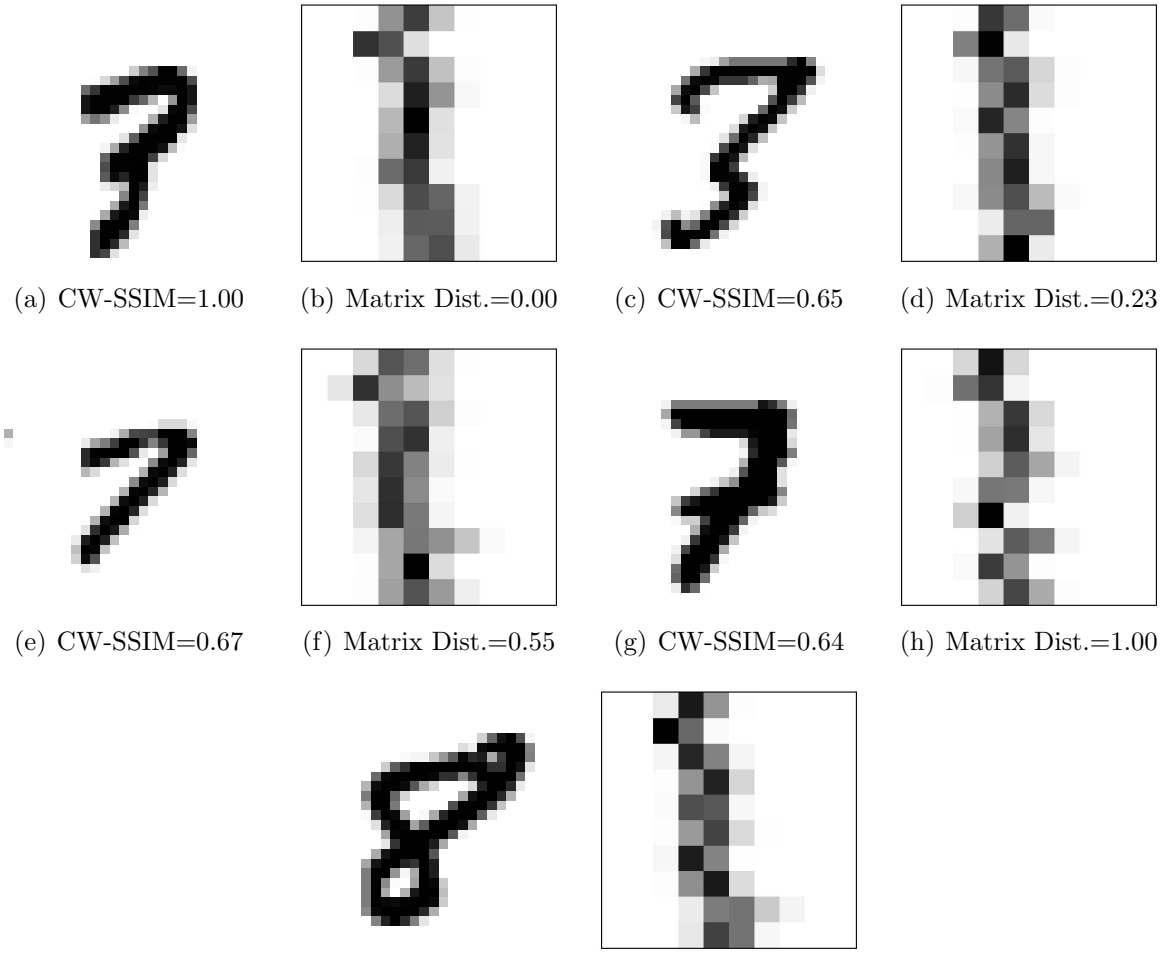
$$I_c\,,\ \text{where } c = \arg\max_{i\in[1,R]} \sum_{j\in[1,R]} \tilde{S}(I_i, I_j)\,. \tag{3.11}$$

Here the centroid $I_c$ is not really the "mean" of all the images in the cluster, because CW-SSIM is not a valid distance metric in the image space and there is no simple definition of the notion of "mean" in terms of CW-SSIM. Rather, it is a representative image selected from all images in the cluster that on average is most similar to all other images in CW-SSIM sense. Given $Z$ clusters with centroids $I_c^{(1)}$, $I_c^{(2)}$, ..., $I_c^{(Z)}$, the cluster label updating step is performed by reassigning the membership of each image $I_i$ for $i = 1, ..., N$ by

$$I_i \in C_j\,,\ \text{where } j = \arg\max_{j\in[1,Z]} \tilde{S}(I_i, I_c^{(j)})\,, \tag{3.12}$$

where $C_j$ denotes the set of all images belonging to the $j$-th cluster.

The above clustering algorithm group images in the training set without considering their class labels. As a result, a "bad" or outlier sample image may be clustered to a group



(a) CW-SSIM=1.00 (b) Matrix Dist.=0.00 (c) CW-SSIM=0.65 (d) Matrix Dist.=0.23

(e) CW-SSIM=0.67 (f) Matrix Dist.=0.55 (g) CW-SSIM=0.64 (h) Matrix Dist.=1.00

(i) CW-SSIM=0.67 (j) Matrix Dist.=0.41

Figure 3.3: Comparison of CW-SSIM value and histogram matrix distance between reference image (a) and other images. The matrix distance is normalized to 1.

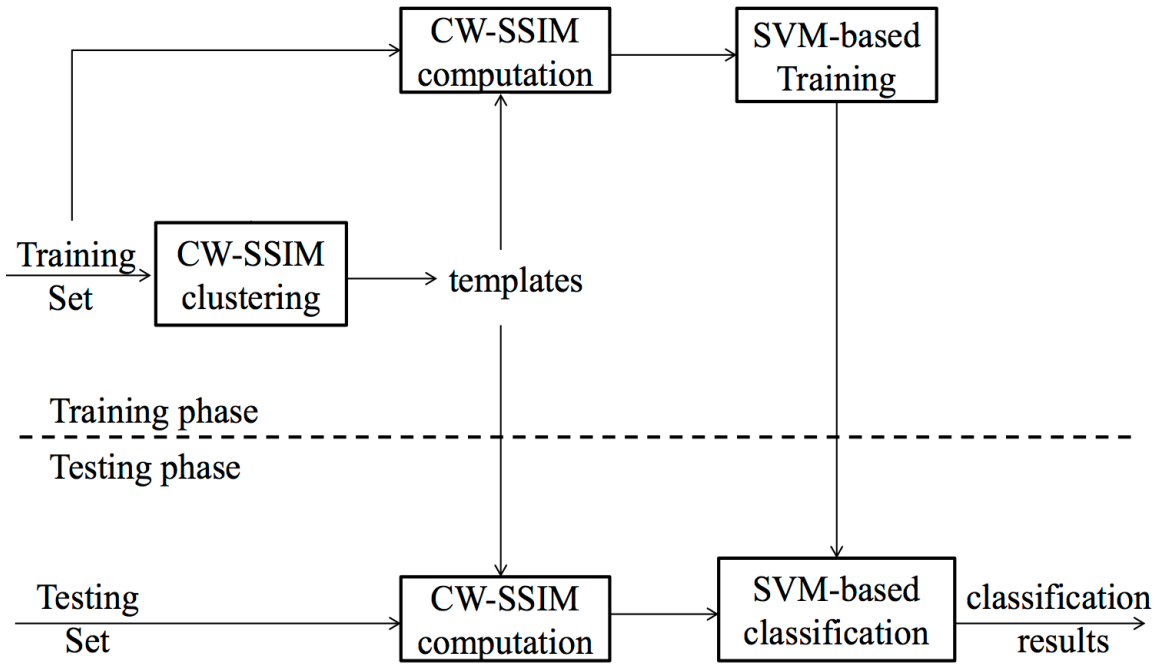



Figure 3.4: Framework of the proposed CW-SSIM SVM method.

of sample images that are similar in structure but have different class labels. To avoid such situations, for each training image $I_t$ with class label $l_t$, we examine its $k$ nearest neighbors $\{I_t^{(i)} | i = 1, ..., k\}$ with class labels $\{l_t^{(i)} | i = 1, ..., k\}$ and compute the frequency of these neighbors that have the same class labels as $I_t$

$$\tilde{p}(I_t) = \frac{\sum_{i=1}^{k} \delta(l_t, l_t^{(i)})}{k}. \tag{3.13}$$

We then exclude the training images from the $k$-means clustering process with $\tilde{p}(I_t) < T_p$, where $T_p$ is a preset threshold. Our experiments show that this training image pruning approach helps improve the classification results.

The $k$-means clustering process provides us with a set of cluster centroids or representative images. We can then apply the same weighted $k$-NN method for image classification as described in Eq. (3.10). The only difference is that the full training set used in Eq. (3.10) is replaced by the set of representative images. This leads to a much more efficient CW-SSIM weighted $k$-means image classification method.

### 3.2.3 CW-SSIM Based Support Vector Machine Method

Motivated by the success of the SVM method [10] in a variety of pattern recognition tasks, we develop a CW-SSIM SVM image classification algorithm. The general structure of the algorithm is illustrated in Fig. 3.4, where the training phase consists of two main stages − an unsupervised clustering stage and a supervised SVM learning stage.

In the first stage, the training images are divided into clusters and one representative image (or template) is selected for each cluster. It is useful to be aware that there could be many different writing styles of the same digit, thus it makes sense to group the training images not only by their class labels, but also by their styles or structures. CW-SSIM is an ideal tool for this task because images originated from the same digit and written with the same style are likely to be shifted, scaled, and/or rotated versions of each other. Our unsupervised clustering method works as follows. First, we calculate a matrix $\mathbf{C}$ of size $N \times N$, which contains the CW-SSIM values of every image with every other image in the training set. Each column of this matrix is a vector $\mathbf{s}_i = \{\tilde{S}(I_i, I_j)|j = 1, ...N\}$ that contains the CW-SSIM values between the $i$-th image and all other images in the training set. This vector may be considered as "features" of the $i$-th training image (though not the descriptive features of image structures typically used in many other image classification methods). The clustering process starts by taking the whole training set as one cluster and defines the centroid of the cluster as

$$I_c^{(1)}\,,\ \text{where } c = \arg\max_{i\in[1,N]} \sum_{j\in[1,N]} \tilde{S}(I_i, I_j)\,. \tag{3.14}$$

Now assume that we are at a stage where we have $M$ clusters with centroids $I_c^{(1)}$, $I_c^{(2)}$, ..., $I_c^{(M)}$, respectively (the initial stage corresponds to $M = 1$ case). We decide on whether to create a new cluster by checking whether

$$\min_{i\in[1,N]} \max_{j\in[1,M]} \tilde{S}(I_i, I_c^{(j)}) > T\,, \tag{3.15}$$

where $T$ is a predefined threshold. If this is satisfied, then we can stop with the current number of clusters and use the corresponding centroids as representative images for the clusters. Otherwise, we define a new cluster centroid as

$$I_c^{(M+1)} = I_m\,,\ \text{where } m = \arg\min_{i\in[1,N]} \max_{j\in[1,M]} \tilde{S}(I_i, I_c^{(j)})\,, \tag{3.16}$$

and let $M = M + 1$. After a new cluster is added, we reassign the membership of each image $I_i$ for $i = 1, ..., N$ by

$$I_i \in C_j\,,\ \text{where } j = \arg\max_{j\in[1,M]} \tilde{S}(I_i, I_c^{(j)})\,, \tag{3.17}$$

where $C_j$ is the collection of all images belonging to the $j$-th cluster. The new centroid for each class $j \in [1, M]$ is then updated by

$$I_c^{(j)} = I_m \,, \text{ where } m = \arg\max_{I_i \in C_j} \sum_{I_k \in C_j} \tilde{S}(I_i, I_k) \,. \tag{3.18}$$

This is followed by the next stage of judgement on whether a new cluster should be created, as in Eq. (3.15).

In the second stage of the training phase, we have the representative templates at hand. We can then describe any training image using a length-$M$ vector of CW-SSIM values between the training image and all templates. Since every training image has a class label associated with it, this is a supervised learning problem. In particular, we develop a classifier by using support vector machines (SVM) with Gaussian kernels, which has been proven to be a powerful classifier of excellent generalization capability. Details of the SVM learning algorithm can be found in [10].

The testing part of our CW-SSIM SVM classification algorithm is straightforward. For each test query image, we compute its CW-SSIM values with respect to all templates, resulting a length-$M$ vector of CW-SSIM values. We then feed this vector to the SVM classifier, which produces a classification result.

### 3.2.4 CW-SSIM Based Support Vector Machine Method Using Affinity Propogation

Our two-staged scheme consists of an unsupervised clustering method to select templates, and a supervised learning stage based on SVM. Since SVM has been proved to be a practical and precise tool to model data behavior in various applications, to improve the performance of this scheme, our next step is focused on the clustering stage. The common clustering approach, like $k$-means, uses learning data to learn a set of centers such that a relatively small squared error between data points and nearest center can be achieved. Taking $k$-means as an example, it begins with an initial set of randomly seleted templates and iteratively refines this set so as to decrease the sum of squared errors. However, this works well only in case of a small number of clusters and chances are good that at least one random inintialization is close to a good solution.

A novel and efficient clustering algorithm, affinity propagation, was proposed in 2007 [6], which claimed to address a solution to the randomness of clustering initialization and found clusters with much lower error and time consumption than other methods. At the very beginning of the clustering procedure, it views all data points as potential templates. By viewing each data point as a node in a network, they devised a method that recursively transmits real-valued messages along edges of the network until converge to a good set of

templates and corresponding clusters. On the basis of formulas which targeting minima of an appropriately chosen energy function, these message are updated every iteration, and their magnitude reflects the current affinity that one data point has for choosing another data point as its templates.

In general, affinity propagation takes a collection of real-valued similarities between data points as input, where the similarity $s(i,k)$ indicates the eligibility of index $k$ to be suited as the template for data point $i$. In our case, this similarity matrix can be set to the CW-SSIM value matrix $s(i,j)$ between image $i$ and $j$, $i \neq j$. Moreover, affinity progation also takes as input a real number $s(k,k)$ for each data point $k$ so that a larger values of $s(k,k)$, the so-called "preferences", can lead to a higher possibility for data point $k$ to be chosen as templates. The number of identified templates is influenced by the values of the input preferences.

Two kinds of real-valued messages are exchanged between data points. The "responsibility" $r(i,k)$, sent from data point $i$ to candidate template point $k$, reflects the quantative evidence for how eligible $k$ is to serve as the template for point $i$, considering other potential template for point $i$. The "availablility" $a(i,k)$, sent from candidate template point $k$ to point $i$, reflects the quantative evidence for how suitable it would be for point $i$ to choose point $k$ as its tempalte. The availabilities are initialized to zero: $a(i,k) = 0$. The responsibilities and availabilities are then calculated as follows:

$$r(i,k) \leftarrow s(i,k) - \max_{k' \text{s.t.} k' \neq k} \{a(i,k') + s(i,k')\}. \tag{3.19}$$

$$a(i,k) \leftarrow \min\{0, r(k,k) + \sum_{i' \text{s.t.} i' \notin \{i,k\}} \max\{0, r(i',k)\}\}. \tag{3.20}$$

Thus in the first iteration, $r(i,k)$ is set to the input similarity $s(i,k)$, minus the largest of the similarities between point $i$ and other candidate templates. In the later iterations when some points are appropriately assigned to other templates, their availabilities will drop below zero, as prescribed by the rule above. These negative availabilities will decrease the effectiveness of some input similarities $s(i,k')$ in Eq.3.21, excluding the corresponding candidate images from template pool. For $k = i$, the responsibility $r(k,k)$ is set to the input $s(k,k)$, the preference that data point $k$ to be chosen as a template, minus the largest of similarities between point $i$ and all other candidate template. This "self-responsibility" reflects quantative evidence for the possibility of point $k$ being a template, based on its input preference value tempered by how inappropriate it is to be assigned to another template.

The availability $a(i,k)$ is set to the "self responsibility" $r(k,k)$, plus the sum of the positive responsibilities candidate template $k$ receives from other points. Only the positive

portions of incoming responsibilities are added, because it is only necessary for a good template to represent some data points well, no matter how poorly it represents other data points. A negative $r(k,k)$ indicates that point $k$ is currently better suited as belonging to another template rather than being an template itself, and its availability as a template can be increased if some other points have positive responsibilities for point $k$ being their template. The total sum is thresholded under zero to limit the influence of strong incoming positive responsibilities. The "self-availability" $a(k,k)$ is updated differently:

$$a(k,k) \leftarrow \sum_{i' \text{s.t.} i' \neq k} \max\{0, r(i',k)\}. \tag{3.21}$$

The equation above reflects quantative evidence that point $k$ is a template, based on the positive responsibilities sent to candidate exemplar $k$ from other points. These update rules only require simple, local computations.

Availabilities and responsibilities can be combined to identify templates at any stage of affinity propagation. For point $i$, the value of $k$ that maximizes $a(i,k) + r(i,k)$ either identifies point $i$ as a template if $k = i$, or identifies the data point that is the exemplar for point $i$. The number of message-passing iterations can be pre-defined. It can also be terminated after changes in the messages fall below a threshold, or after the local decisions stay constant for some number of iterations.

In our application, we use affinity propagation in the clustering stage instead of the algorithm in Section 3.2.3. The CW-SSIM values between training images are used to set the similarity matrix $s(i,j)$. After several iterations the train set will converge to a certain number of clusters, and the corresponding templates will be used in later stage as shown in Figure 3.4.

# Chapter 4

# Experimental Result

## 4.1 Handwritten Digit Image Classification

Our experiments were performed on the MNIST database of handwritten digit images [47], which has been the most widely used benchmark in the literature. The database includes 60,000 training and 10,000 test samples. All images have been size-normalized and centered in a 28 $\times$ 28 box. Some sample images from MNIST database are shown in Figures 4.1.

The performance of all the methods is given in Table 4.1 for different sizes of training set. Each experiment is performed 5 times with training data selected randomly from all training data. The average values of the 5 experiments are presented in the Table. First, we compare the performance of MSE and CW-SSIM based nearest neighbor methods. The results for MSE NN and CW-SSIM NN with different numbers of training images are shown in the second and third rows of Table 4.1. It appears that CW-SSIM alone, as a "raw" similarity measure (without any machine learning process involved), can achieve very good performance (less than 3% error rate) which is significantly better than the performance of MSE. As expected, the performance of CW-SSIM $k$-NN method can be improved when the values of $k > 1$ are considered, as can be observed from the results in Table 4.1. Using CW-SSIM weighted $k$-NN (denoted as CW-SSIM (w) $k$ -NN) helps us further improve the performance as the error rate reduces to 1.73 % when the whole MNIST training data set is used for classification. Figures 4.2 and 4.3 show the performance of CW-SSIM $k$-NN and CW-SSIM weighted $k$-NN, respectively, as a function of training set size for the values of $k$ = 1,3,5,7. It can be observed that the best performance is achieved for the value of $k$ = 5. Therefore, we use $k = 5$ for all the experiments where $k$-NN is used as a classifier.

Second, we test the performance of, more practical, template based methods for different number of training images. For the results presented in Table 4.1, we learned 1150 representatives for each template based method. It can be observed that CW-SSIM pruned

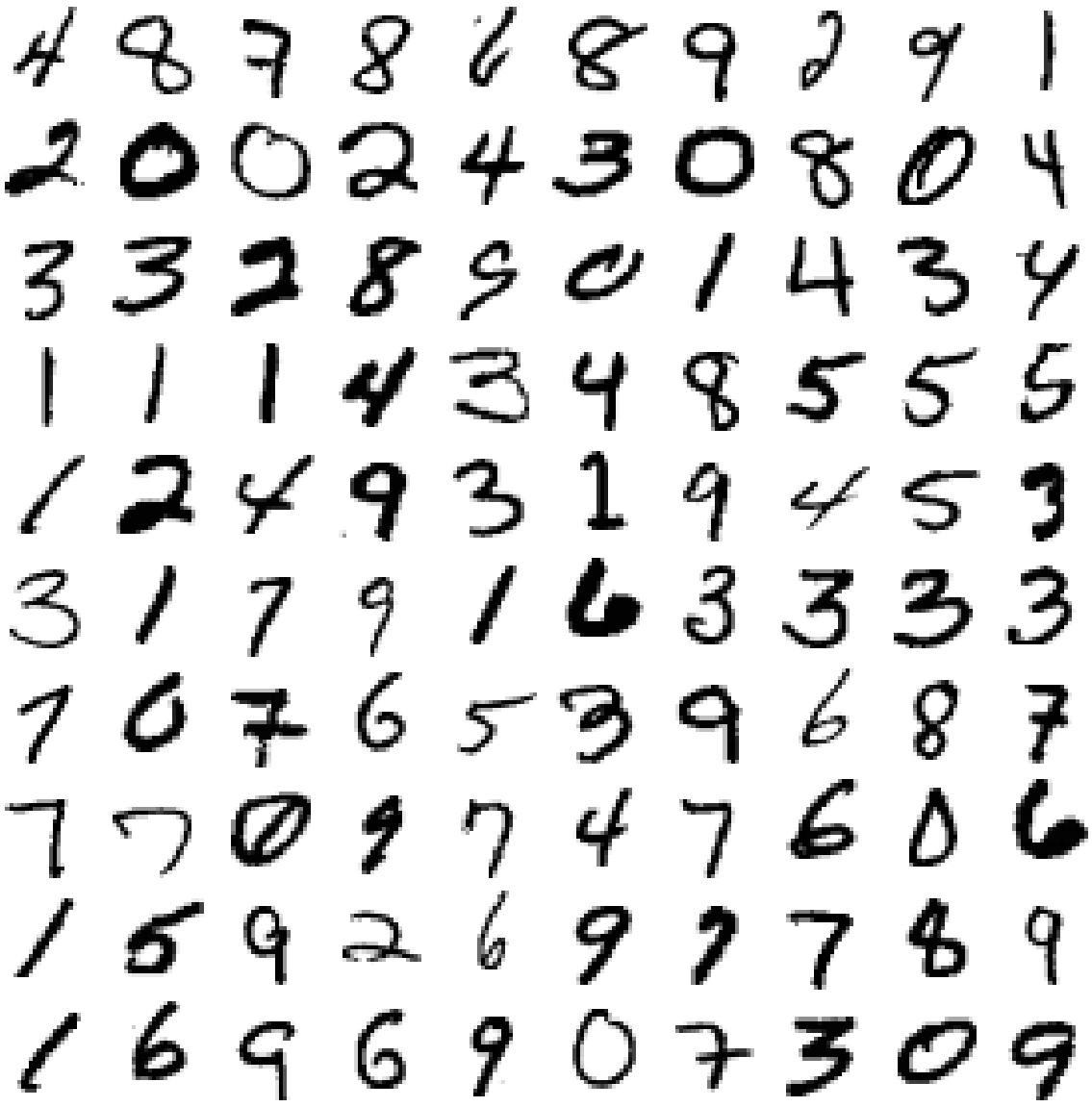


Figure 4.1: Sample images from MNIST database

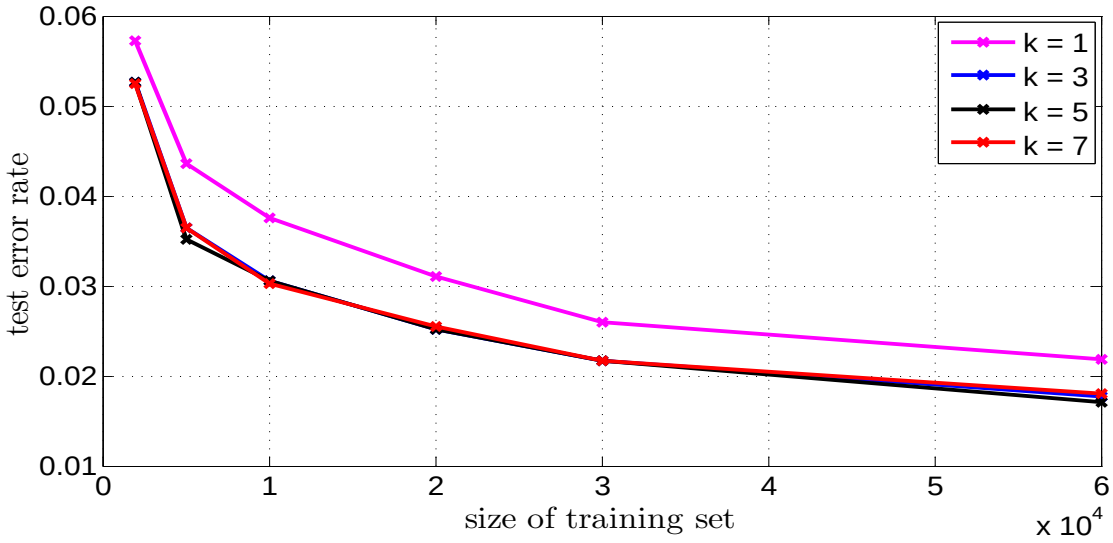



Figure 4.2: Performance of CW-SSIM $k$-NN method as a function of training set size for different values of $k$

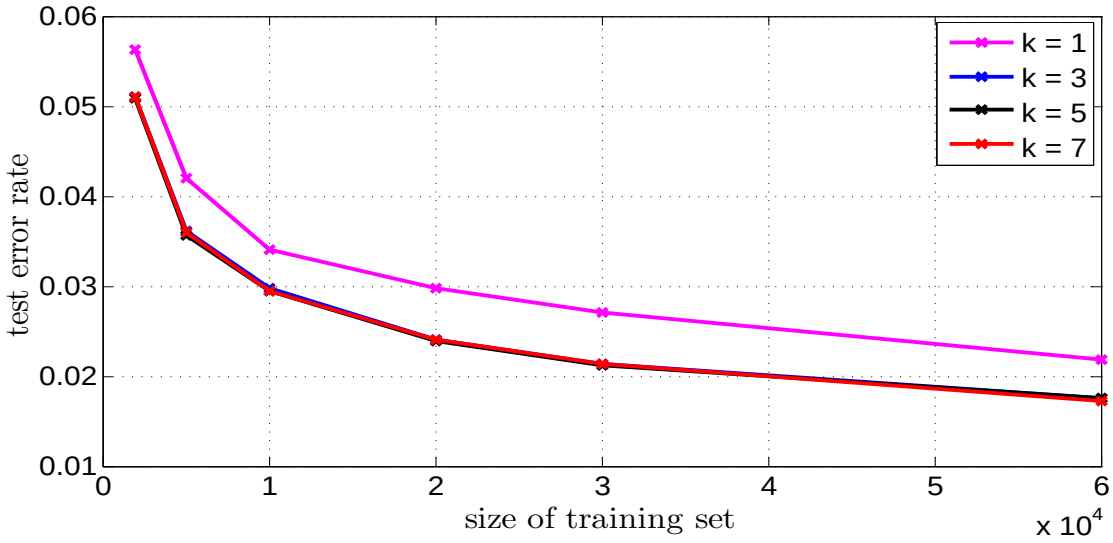



Figure 4.3: Performance of CW-SSIM weighted $k$-NN method as a function of training set size for different values of $k$

$k$-means (denoted as CW-SSIM (p) $k$-means) performs better than the method without pruning. The performance difference is higher for smaller sizes of training sets because pruning is expected to be more effective when the number of training images are lower in number. The value of the threshold, $T_p$, is set to be 0.5 which means that the training images that can not be correctly classified using the training set based on $k$-NN classifier are ignored. Test error rate of 4.56 % suggests that the similarity measure helps us to achieve high accuracy even when a small fraction of training set is used for classification. Our CW-SSIM SVM algorithm outperforms aforementioned template based methods. An SVM is a binary classifier with discriminant function being the weighted combination of kernel functions over all training samples. For multi-class classification, binary SVMs are combined in either one-against-others or one-against-one (pairwise) scheme [48]. Note that in the clustering stage, the resulting number of clusters (and thus templates) varies with different choices of the threshold value $T$. The recognition error rate as a function of the number of templates is shown in Fig. 4.4. It can be observed that using a very small number of templates (38 out of 60,000 training images), the CW-SSIM SVM algorithm can achieve around 95% of accuracy. The error rate further decreases with the increasing number of templates, which collect more variations of representative structures. Some of the learned templates are shown in Fig. 4.5, where we can see that the templates are fairly different from each other even within each digit category, representing different writing styles.

Table 4.1: Performance comparisons based on recognition error rate

| Training samples | 2000 | 5000 | 10000 | 20000 | 30000 | 60000 |
|---|---|---|---|---|---|---|
| MSE NN | 12.57% | 10.41% | 9.56% | 8.23% | 7.62% | 6.92% |
| CW-SSIM NN | 5.72% | 4.35% | 3.75% | 3.41% | 2.50% | 2.18% |
| CW-SSIM $k-$NN | 5.26% | 3.65% | 3.05% | 2.51% | 2.17% | 1.77% |
| CW-SSIM (w) $k-$NN | 5.08% | 3.57% | 2.95% | 2.39% | 2.12% | 1.73% |
| CW-SSIM $k-$means | 7.48% | 6.65% | 6.04% | 5.59% | 5.45% | 4.74% |
| CW-SSIM (p) $k-$means | 7.16% | 5.99% | 5.71% | 5.42% | 5.21% | 4.56% |
| CW-SSIM SVM | 6.02% | 4.24% | 3.70% | 2.81% | 2.45% | 1.91% |

Table 4.2: Time saving by using CW-SSIM SVM as compared to CW-SSIM NN

| Training samples | 2000 | 5000 | 10000 | 20000 | 30000 | 60000 |
|---|---|---|---|---|---|---|
| Time savings | 88.60% | 95.24% | 97.57% | 98.76% | 99.20% | 99.61% |

The proposed CW-SSIM SVM method achieves lower error rate than the other two template based methods for all the sizes of training set. The performance improves with the size of the training set. When all 60,000 training images are used, the error rate is

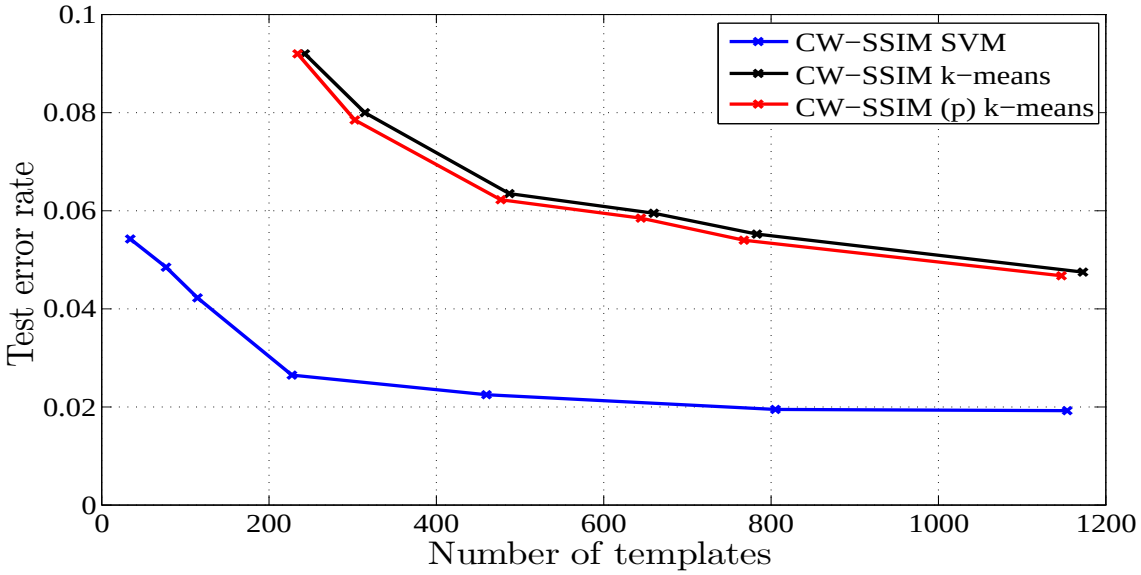



Figure 4.4: Recognition error rate comparison of template-based proposed methods as a function of the number of templates.

reduced to less than 2%. It is important to mention that such improvement in recognition accuracy is obtained with largely reduced computational complexity because only a very small percentage of the images (i.e., the selected templates) need to be compared as compared to the methods that calculate CW-SSIM with all the images in the training set. As reported in Table 4.2, the time saving could be as high as 99.6%. Our non-optimal MATLAB implementation on a Intel Q9400 @ 2.66GHz computer in single core mode takes about 2.5 seconds to classify a test image using 228 templates. It has the potential to achieve real-time performance with code optimization and hardware implementation. Some of the misclassified digits are shown in Figure 4.6. As can be observed that many of them are ambiguous and/or uncharacteristic, with obviously missing parts or strange strokes. Although there exist other recognition systems that achieved higher accuracy [47], they typically involve preprocessing stages (e.g., deskewing and denoising) and/or training and testing algorithms that are much more complicated in terms of both algorithm implementation and computational complexity.

Compared with other CW-SSIM based methods, affinity propagation can be used as an alternative way, and a better way proved by our experiment results, for clustering the training images. The experiments were also performed on the MNIST database of handwritten digit images. The performance of all the methods is given in Table 4.3 as well as Figure 4.7 for different sizes of training set. Each experiment is performed 5 times with training data selected randomly for each experiment. The average values of

G

Figure 4.5: Sample templates learned from MNIST training set.

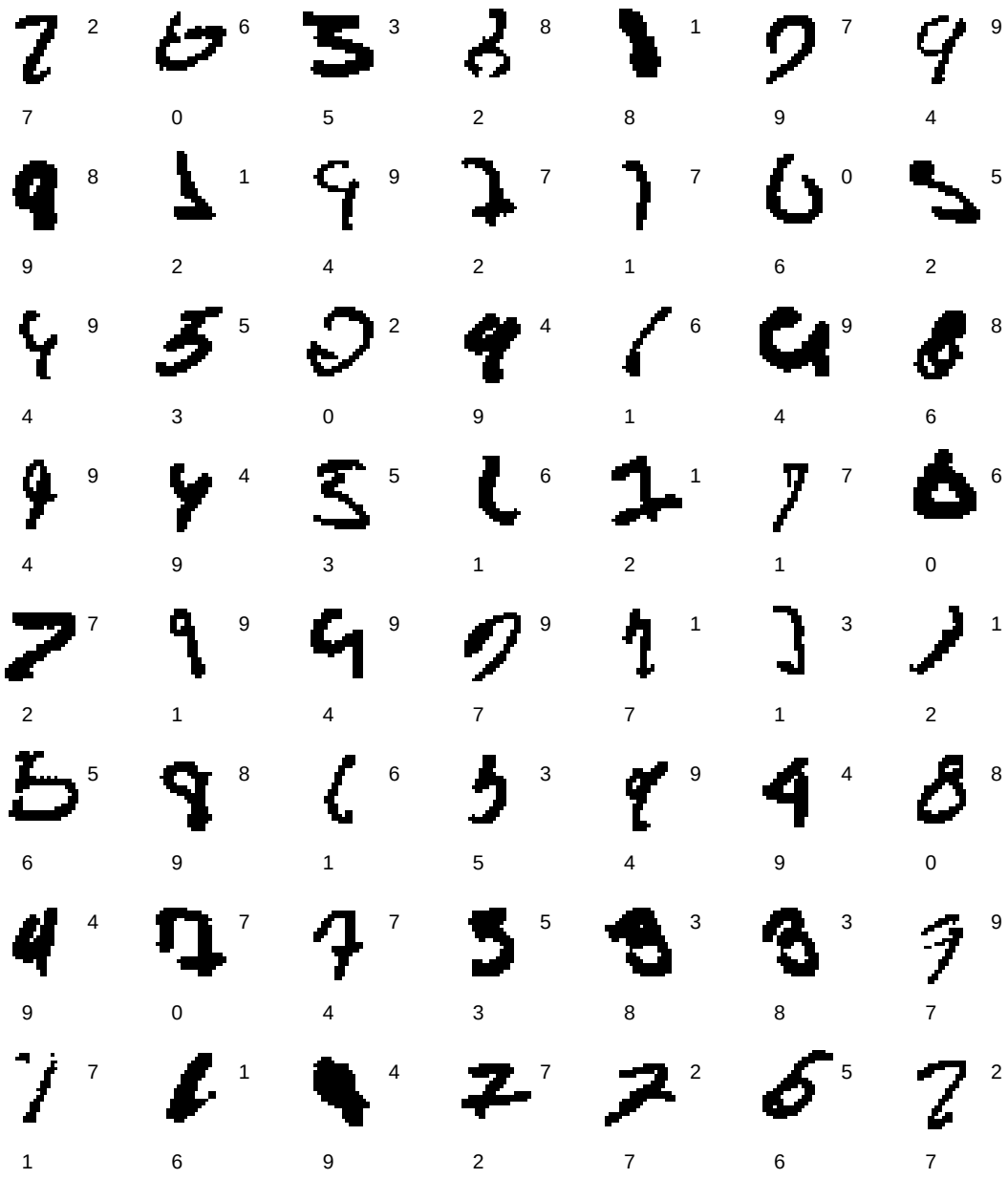



Figure 4.6: Samples of misclassified test digits using proposed method. True label is given in the top right corner and the assigned label in given at the bottom of each image

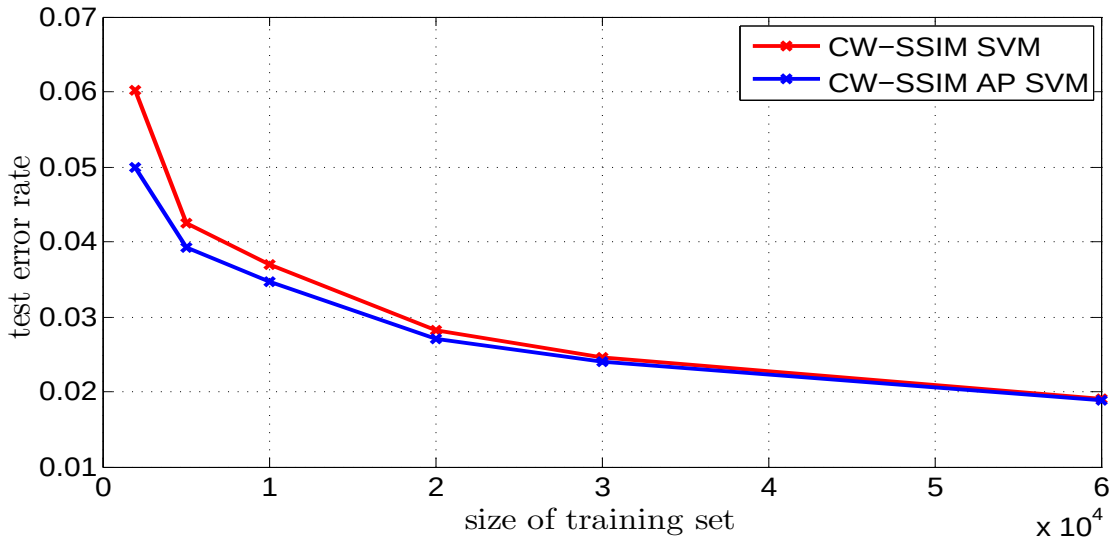



Figure 4.7: Performance of CW-SSIM based SVM method as a function of training set size

the 5 experiments are presented in the Table. As shown in Figure 4.7, the performance of our CW-SSIM based SVM scheme is better with affinity propagation, especially when the training set size is relatively small. Affinity propagation is an efficient algorithm, and it offers better templates, which means more complete and typical representation of all training images, for our later stage. The improvement of accuracy decreases as the training set grows, because in that case, our former algorithm (described in Section 3.2.3) can achieve better results by increasing the number of clusters, which may finally cover almost all data points (used to be 30% more than affinity propagation). By using a relatively small number of templates (some are shown in Figure 4.8), the SVM method with affinity propagation can achieve better classification accuracy. In addition, it is notable that the affinity-propagation-involved scheme also achieve a much lower time consumption of clustering procedure. Comparing to the former method, affinity propagation can save more than two orders of magnitude less calculation time cost under the same computation environment.

Table 4.3: Performance comparisons based on recognition error rate

| Training samples | 2000 | 5000 | 10000 | 20000 | 30000 | 60000 |
|---|---|---|---|---|---|---|
| CW-SSIM SVM | 6.02% | 4.24% | 3.70% | 2.81% | 2.45% | 1.91% |
| CW-SSIM AP SVM | 5.00% | 3.93% | 3.46% | 2.71% | 2.40% | 1.89% |

Figure 4.8: Sample templates clustered using affinity propagation.

Table 4.4: Performance comparisons based on recognition error rate

| Training images | 800 | 700 | 600 | 500 | 400 | 300 | 200 |
|---|---|---|---|---|---|---|---|
| Testing images | 100 | 200 | 300 | 400 | 500 | 600 | 700 |
| CW-SSIM AP $k-$NN | 5.00% | 5.50% | 7.33% | 12.25% | 20.75% | 23.83% | 29.14% |
| CW-SSIM SVM | 2.25% | 3.33% | 4.47% | 6.25% | 7.75% | 10.76% | 15.23% |
| CW-SSIM AP SVM | 0.00% | 0.00% | 1.33% | 2.00% | 2.80% | 6.25% | 10.14% |

## 4.2 Face Image Classification

We next study the problem of face image classification using CW-SSIM based methods. We used CW-SSIM based methods with and without affinity propagation to identify test images among 900 grayscale images (samples shown in Figure 4.9) extracted from the Olivetti face database. Olivetti database, or ORL database, contains a set of face images and was used in various face image applications. There are ten different images from each of 40 distinct subjects. For some subjects, the images were taken at different times, with varying lighting conditions, facial expressions (open or closed eyes, smiling or not smiling) and facial details (glasses or no glasses).

The test results are shown in Table 4.4. Each experiment is performed 5 times with training data selected randomly from training data, and the average values of 5 experiments are presented in the Table. To demonstrate our CW-SSIM based "Affinity propagation + SVM" method is a more efficient and accurate method than other methods, we first compare the performance of our affinity-propagation-involved SVM scheme and the SVM scheme in Section 3.2.3. Figure 4.10 gives a direct perception which prove in clustering stage affinity propagation can achieve much better results than the others. The results are also shown in the second and third rows of Table 4.4. It appears that affinity-propagation-involved SVM scheme uniformly achieve much lower error rate, and also as expected, save more than 60% computation time. It is also worth mentioning that with less than half of all data (400 training image out of 900, other 500 as testing set), the "affinity propagation + SVM" scheme can obtain a classification accuracy of more than 97.2% using less than 50 templates. A number of template samples are shown in Figure 4.12.

We then test the performance of our "affinity propagation + SVM" method and $k$-NN algorithm, with different values of $k$, using the same templates clustered using affinity propagation. Figure 4.11 show that SVM significantly outperforms a series of $k$-NN methods, proving SVM is a much better tool to model training data behavior and produce competitive test results.

In addition, this face image application is also sufficiently fast to achieve real-time performance on a single-core MATLAB implementation.

Figure 4.9: Samples of 900 images extracted from Olivetti database.

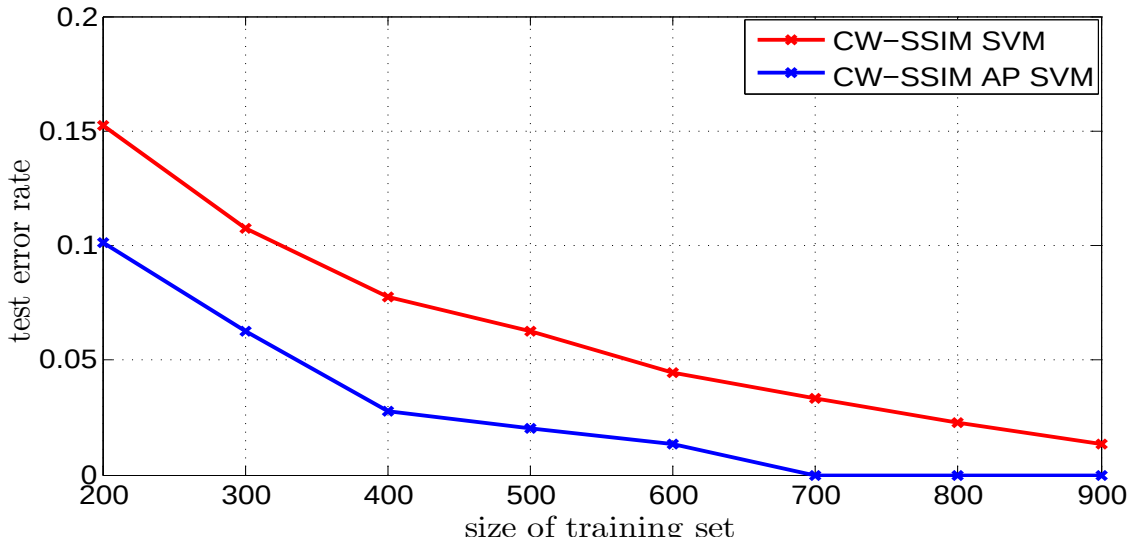



Figure 4.10: Performance of CW-SSIM based SVM method as a function of training set size

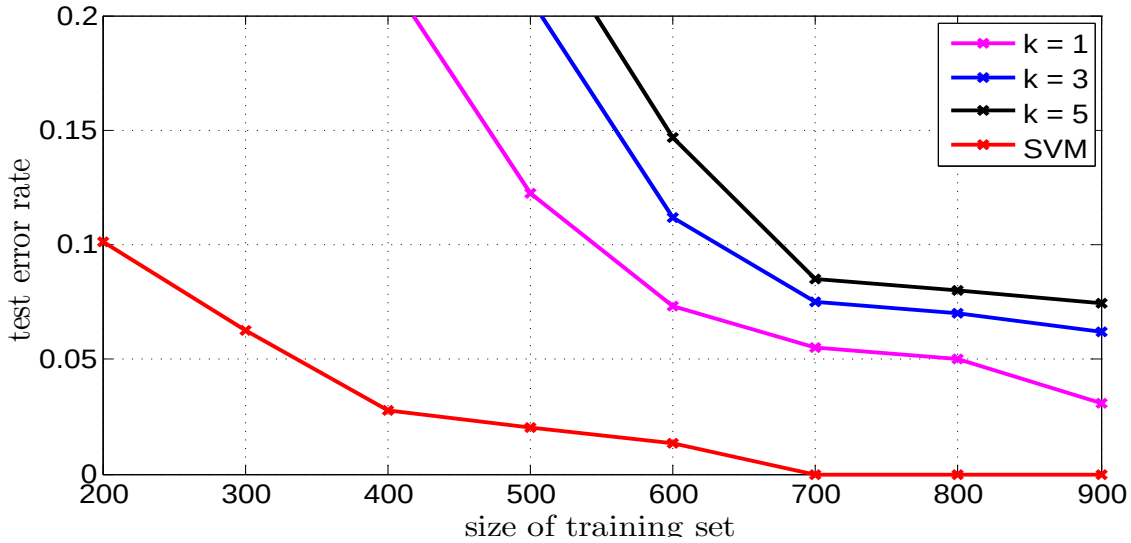



Figure 4.11: Performance of CW-SSIM Based SVM and CW-SSIM $k$-NN method as a function of training set size for different values of $k$

Figure 4.12: Samples templates clustered using affinity propagation from Olivetti database.

# Chapter 5

# Conclusions

In the past several decades, many algorithms have been developed for quite diverse subfields of image classification. They may greatly differ from each other in many ways, and it looks improbable to migrate a classification scheme, including specified feature definition and extraction process, to another different application. Therefore it is natural to think of developing a universal approach for general-purpose image classification task. We decided to use template based scheme, and after proving CW-SSIM's excellent performance for image classification tasks, we chose CW-SSIM as the image simlarity criterion.

We studied the problem of image classification using CW-SSIM, which is connected with a number of computational models in biological vision and is robust to small geometric distortions of images. We use digit image and face image classification as examples and propose a series of CW-SSIM based algorithms, including CW-SSIM based nearest neighbor method, CW-SSIM based $k$ means method, CW-SSIM based support vector machine method (SVM) and CW-SSIM based SVM using affinity propagation, which do not rely on any normalization, registration or image structure description-based feature extraction processes, and do not involve any statistical modeling of the image patterns or distortion processes, but achieve competitive performance in recognition accuracy. These properties make the proposed algorithms readily adapted to a broad range of image classification problems.

We put a lot of effort on finding the optimal CW-SSIM based algorithm. According to the experiment on handwritten digit and human face image classifications, CW-SSIM based support vector machine using affinity propagtion is currently achieving the most satisfactory outcome among other $k$-NN, $k$-means, or SVM only schemes. Yet the present work is still at the initial stage of a brand new research direction, and can be extended in a number of ways in the future.

One obvious extension would be applying the CW-SSIM based algorithm for other image classification applications, such as fingerprint and palmprint classification, and other

types of digit or face image classification (i.e., more complicated and detailed face image under different context). Successfully migrate the same algorithm to those applications in different subfield of image classification will better validate the generality of template based scheme along with CW-SSIM as the similarity measurement.

In addition, some details of our scheme, including the similarity index and the clustering method, can be adjusted to achieve better results. We may explore the potential of developing a better similarity measurement for image classification tasks using weighted or multi-scale CW-SSIM. As we are using raw CW-SSIM values as the only input of affinity propagation, it is also possible that a different settings of affinity propagation or a combination of CW-SSIM, affinity propagation and other techniques can improve the clustering efficiency or reduce the computational complexity.

# Bibliography


[1] John C. Russ. *The Image Processing Handbook 6th Edition.* CRC Press., Raleigh, NC, USA, 2002.

[2] David Zhang, Wai-Kin Kong, Jane You, and Michael Wong. Online palmprint identification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25:1041–1050, 2003.

[3] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(4):509–522, April 2002.

[4] A. Bouzerdoum, A. Havstad, and A. Beghdadi. Image quality assessment using a neural network approach. *Proc. IEEE Int. Sym. Signal Processing and Information Technology*, pages 330–333, 2004.

[5] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, January 1967.

[6] Brendan J. Frey and Delbert Dueck. Clustering by passing messages between data points. *Science*, (315):972–976, 2007.

[7] Jerome H. Friedman. Regularized Discriminant Analysis. *Journal of the American Statistical Association*, 84(405):165–175, 1989.

[8] Hyun-Chul Kim, Daijin Kim, and Sung Yang Bang. A numeral character recognition using the pca mixture model. *Pattern Recognition Letters*, 23(1-3):103 – 111, 2002.

[9] Fumitaka Kimura, Kenji Takashina, Shinji Tsuruoka, and Yasuji Miyake. Modified quadratic discriminant functions and the application to chinese character recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(1):149 – 153, jan. 1987.

[10] C. Burges. A tutorial on support vector machines for pattern recognition. *Knowledge Discovery and Data Mining*, 2:1–47, 1998.

[11] Vladimir N. Vapnik. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995.

[12] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[13] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. *Learning internal representations by error propagation*, pages 318–362. MIT Press, Cambridge, MA, USA, 1986.

[14] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, November 1998.

[15] P.Y. Simard, D. Steinkraus, and J.C. Platt. Best practices for convolutional neural networks applied to visual document analysis. In *Document Analysis and Recognition, 2003. Proceedings. Seventh International Conference on*, pages 958 – 963, aug. 2003.

[16] Zhou Wang and A. C. Bovik. *Modern Image Quality Assessment*. Morgan & Claypool Publishers, March 2006.

[17] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Processing*, 13(4):600–612, April 2004.

[18] Zhou Wang and Eero P. Simoncelli. Translation insensitive image similarity in complex wavelet domain. In *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing*, Philadelphia, PA, March 2005.

[19] J. A. Solomon and D. G. Pelli. The visual filter mediating letter identification. *Nature*, 369:395–397, 1994.

[20] Pollen D. A. and Ronner S. F. Distinctive image features from scale-invariant keypoints. *Science*, 212:1409–1411, 1981.

[21] I. Ohzawa, G. DeAngelis, and R. Freeman. Stereoscopic depth descrimination in the visual cortex: Neurons ideally suited as disparity detectors. *Science*, (249):1037–1041, 1990.

[22] E H Adelson and J R Bergen. Spatiotemporal energy models for the perception of motion. *Journal of Optical Society of America*, 2(2):284–299, Feb 1985.

[23] D. J. Heeger. Normalization of cell responses in cat striate cortex. *Visual Neuroscience*, (9):181–197, 1992.

[24] O. Schwartz and E. P. Simoncelli. Natural signal statistics and sensory gain control. *Nature: Neuroscience*, 4(8):819–825, August 2001.

[25] Mehul P. Sampat, Zhou Wang, Shalini Gupta, Alan C. Bovik, and Mia K. Markey. Complex wavelet structural similarity: A new image similarity index. *IEEE Trans. Image Processing*, 18(11):2385–2401, November 2009.

[26] Shalini Gupta, Mehul P. Sampat, Zhou Wang, Mia K. Markey, and Alan C. Bovik. Facial range image matching using the complex wavlet structural similarity metric. *Proc. IEEE Workshop on Applications of Computer Vision*, February 2007.

[27] L. Zhang, Z. Guo, Z. Wang, and D. Zhang. Palmprint verification using complex wavelet transform. *Proc. IEEE Int. Conf. Image Proc.*, September 2007.

[28] G. Fan, Zhou Wang, and Jiheng Wang. CW-SSIM kernal based random forest for image classification. *Proc. SPIE Visual Comm. and Image Processing*, July 2010.

[29] D. Zhang, Wai-Kin Kong, J. You, and M. Wong. Online palmprint identification. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(9):1041 – 1050, sept. 2003.

[30] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188, 1936.

[31] J.A.K. Suykens and J. Vandewalle. Least squares support vector machine classifiers. *Neural Processing Letters*, 9:293–300, 1999.

[32] E. J. Bredensteiner and K. P. Bennett. Multicategory classification by support vector machines. *Computational Optimization and Applications*, 1999.

[33] Z. Wang and A. C. Bovik. Mean squared error: Love it or leave it? A new look at signal fidelity measures. *IEEE Signal Processing Magazine*, 26(1):98–117, January 2009.

[34] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, April 2002.

[35] Chih hsien Kung, Wei sheng Yang, and Chih ming Kung. A study on image quality assessment using neural networks and structure similarty. *Journal of Computers*, 6(10), 2011.

[36] P. Carrai, I. Heynderickz, P. Gastaldo, and R. Zunino. Image quality assessment by using neural networks. *Proc. IEEE Int. Sym. Circuits and Systems*, 5, 2002.

[37] Ismail Avcibas, Bulent Sankur, and Khalid Sayood. Statistical evaluation of image quality measures. *Journal of Electronic Imaging*, 11(2):206–223, 2002.

[38] Alesandr Shnayderman, Alexander Gusev, and Ahmet M. Eskicioglu. An svd-based gray-scale image quality measure for local and global assessment. *IEEE Trans. Image Processing*, 2006.

[39] Manish Narwaria and Weisi Lin. Video quality assessment using temporal quality variations and machine learning. *Proc. IEEE Int. Conf. Multimedia and Expo*, 0:1–6, 2011.

[40] Wenrui Ding, Yubing Tong, Qishan Zhang, and Dongkai Yang. Image and video quality assessment using neural network and svm. *Tsinghua Science and Technology*, 13(1):112 – 116, 2008.

[41] T. H. Falk, Yingchun Guo, and Wai-Yip Chan. Improving robustness of image quality measurement with degradation classification and machine learning. *Proc. IEEE Asilomar Conf. on Signals, Systems, and Computers*, December 2007.

[42] Huixuan Tang, Neel Joshi, and Ashish Kapoor. Learning a blind measure of perceptual image quality. *CVPR*, 2011.

[43] J.E.S. Macleod, A. Luk, and D.M. Titterington. A re-examination of the distance-weighted k-nearest neighbor classification rule. *IEEE Transactions on Systems, Man and Cybernetics*, 17(4):689 –696, july 1987.

[44] Jakub Zavrel. An empirical re-examination of weighted voting for k-nn. In *Proceedings of the 7th Belgian-Dutch Conference on Machine Learning*, pages 139–148, 1997.

[45] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1967.

[46] Baback Moghaddam and Alex Pentland. Probabilistic visual learning for object detection. In *ICCV*, pages 786–793, 1995.

[47] Yann Lecun and Corinna Cortes. The MNIST database of handwritten digits, 2010.

[48] U. Kressel. *Pairwise classification and support vector machines*, pages 255–268. MIT Press, Cambridge, MA, USA, 1999.