Selection of Robust Features for Coin Recognition and Counterfeit Coin Detection

Ali Khadair Kadam Al-Frajat

A Thesis
In the Department
of
Computer Science and Software Engineering

Presented in Partial Fulfillment of the Requirements
For the Degree of
Doctor of Philosophy (Computer Science) at
Concordia University
Montreal, Quebec, Canada

September 2018

CONCORDIA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

This is to certify that the thesis prepared

By: Ali Khadair Kadam Al-Frajat

Entitled: Selection of Robust Features for Coin Recognition and Counterfeit Coin Detection

and submitted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY (Computer Science)

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

_____ Chair

Dr. Catherine Mulligan

_____ External Examiner

Dr. Herbert Yang

_____ External to Program

Dr. Wei-Ping Zhu

_____ Examiner

Dr. Charalambos Poullis

_____ Examiner

Dr. Louisa Lam

_____ Thesis Supervisor

Dr. Ching Y. Suen

Approved by _____

Dr. Volker Haarslev, Graduate Program Director

December 17, 2018

_____

Dr. Amir Asif, Dean

Gina Cody School of Engineering & Computer Science

ABSTRACT

Selection of Robust Features for Coin Recognition and Counterfeit Coin Detection

Ali Khadair Kadam Al-Frajat, Ph.D.
Concordia University, 2018

Tremendous numbers of coins have been used in our daily life since ancient times. Aside from being a medium of goods and services, coins are items most collected worldwide. Simultaneously to the increasing number of coins in use, the number of counterfeit coins released into circulation is on the rise. Some countries have started to take different security measures to detect and eliminate counterfeit coins. However, the current measures are very expensive and ineffective such as the case in UK which recently decided to replace the whole coin design and release a new coin incorporating a set of security features. The demands of a cost effective and robust computer-aided system to classify and authenticate those coins have increased as a result.

In this thesis, the design and implementation of coin recognition and counterfeit coin detection methods are proposed. This involves studying different coin stamp features and analyzing the sets of features that can uniquely and precisely differentiate coins of different countries and reject counterfeit coins. In addition, a new character segmentation method crafted for characters from coin images is proposed in this thesis. The proposed method for character segmentation is independent of the language of those characters. The experiments were performed on different coins with various characters and languages. The results show the effectiveness of the

method to extract characters from different coins. The proposed method is the first to address character segmentation from coins.

Coin recognition has been investigated in several research studies and different features have been selected for that purpose. This thesis proposes a new coin recognition method that focuses on small parts of the coin (characters) instead of extracting features from the whole coin image as proposed by other researchers. The method is evaluated on coins from different countries having different complexities, sizes, and qualities. The experimental results show that the proposed method compares favorably with other methods, and requires lower computational costs.

Counterfeit coin detection is more challenging than coin recognition where the differences between genuine and counterfeit coins are much smaller. The high quality forged coins are very similar to genuine coins, yet the coin stamp features are never identical. This thesis discusses two counterfeit coin detection methods based on different features. The first method consists of an ensemble of three classifiers, where a fine-tuned convolutional neural network is used to extract features from coins to train two classifiers. The third classifier is trained on features extracted from textual area of the coin.

On the other hand, sets of edge-based measures are used in the second method. Those measures are used to track differences in coin stamp's edges between the test coin and a set of reference coins. A binary classifier is then trained based on the results of those measures. Finally, a series of experimental evaluation and tests have been performed to evaluate the effectiveness of these proposed methods, and they show that promising results have been achieved.

ACKNOWLEDGEMENT

First and foremost, I would like to express my sincere gratitude to my supervisor Professor Ching Y. Suen. Without his persistent guidance, support, and patience this work would not have happened. Throughout this research, I felt honored to be working with a star professor in image processing and pattern recognition. Prof. Suen was always available to openly discuss new research ideas and to take the time to review my research articles and thesis drafts. I also would like to thank Prof. Louisa Lam for her great efforts in reading the drafts of my thesis and papers and for providing precious feedbacks and corrections. Her suggestions and insights have significantly strengthened this work.

I would like to extend my gratitude to my examination committee for taking the time to read and evaluate this thesis. Their comments and feedbacks are valuable and highly appreciated.

A thank you to all friends, colleagues, and staff at the Centre for Pattern Recognition and Machine Intelligence (CENPARMI) for making this journey easier. I highly appreciate all the talks, discussions, and encouragements. Special thanks to Dr. Marleah Blom, CENPARMI's Executive Assistant, for the cheerful chats and administrative assistance. Also to Mr. Nicola Nobile, CENPARMI's research manager, for his continuous technical support.

Last but not least, I want to express my heartfelt gratitude and appreciation to my father Prof. Khadair Hmood Al-Frajat who believed in me and supported me by all means throughout my journey. I always remember his advice and words of wisdom which have enlightened and guided my way. Many thanks to my mother for all her patience, support, and encouragements, and finally, to my wife for her understanding and patience, to my daughter whose smile has always made my day, to my brother and best friend, and to my sisters for their support and encouragement.

DEDICATION

I would like to dedicate this thesis

To my parents for their unconditional love and support,

To my wife and daughter for making this journey easier,

To my brother and sisters for their encouragements,

To the best supervisor that anyone would hope for, Prof. C. Y. Suen,

Thank you all.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

DISCLAIMER

This is to declare that the author of this thesis, Ali Khadair Kadam Al-Frajat, has used his forth name, Hmood, in all his publications. The author's publications can be searched by the author name as "Ali K. Hmood".

ABBREVIATIONS

ROI	Region of Interest
HVS	Human Vision System
DCT	Discrete Cosine Transformation
SIFT	Scale Invariant Feature Transform
SURF	Speeded Up Robust Features
MSER	Maximally Stable Extremal Region
BoVW	Bag of Visual Words
GLOH	Gradient Location and Orientation Histogram
HOG	Histogram of Oriented Gradient
LBP	Local Binary Pattern
CC	Connected Component
SVM	Support Vector Machine
BPNN	Back Propagation Neural Network
KNN	K-nearest neighbors
PCA	Principle Component Analysis
LDA	Linear Discriminant Analysis
LPP	Locality Preserving Projections
CNN	Convolutional Neural Network
Conv	Convolution layer
ReLU	Rectified Linear Unit
FC	Fully Connected Layer
MSE	Mean Square Error
SNR	Signal-to-Noise Ratio
SSIM	Structural Similarities

CHAPTER 1

INTRODUCTION

1.1 MOTIVATION

Tremendous amounts of coins have been used in our daily life since the very first trading activities many centuries ago. Aside from being a medium of goods and services, coins are items most collectable worldwide. In fact, coin collection traces back to the first emperor of Rome, Caesar Augustus, who ruled between 27 B.C. and 14 A.D., where he used coins as gifts [1]. Some coins, especially unique ancient coins, can be worth more than a million dollars in today's market [2]. Therefore, a great deal of attention has been shown for numismatic applications in recent years. Most works of researchers were on coin recognition and counterfeit coin detection while some showed interest in coin grading systems. A large number of computer vision approaches have been proposed to efficiently and reliably recognize coins based on their minting time, denomination, and origin. Coin recognition has received enormous attention in recent years after the two big workshop competitions which took place in 2006 and 2007 organized by the MUSCLE CIS-Benchmark. The goal of the competitions was to classify European coins belonging to 12 countries prior to the introduction of the Euro coins.

The problem of counterfeit coin detection (or coin authentication) has been discussed as well, and various methods have been proposed to tackle the high-quality counterfeit coins detection problem. However, very few methods have been developed to automatically grade the

coins and to place each coin to its precise quality level according to a standard scale. This research concentrates on addressing coin recognition and counterfeit coin detection methods.

Coin recognition and counterfeit detection are long established research directions studied in different fields such as physics, chemistry, engineering, and computer science. Several methods have been proposed in each field. Most of existing studies recognize and authenticate coins based on physical characteristics such as metal features, thickness, diameter, size, weight, conductivity, electromagnetic field properties, and piezoelectric characteristics. The purpose of introducing the computer vision approaches in coin recognition rather than physical characteristics is grounded on various factors, such as different coins around the world can share the same physical characteristics (i.e. weight, thickness, size, and diameter) and also use the same metal type. For instance, the 2-Euro coins from different European countries share the same physical characteristics such as width, weight and diameter and are made of the same metal, and they even share a similar design on the obverse side. They only difference between the two is the design on the reverse side which is country specific. The other factor is that inserting any circular piece of metal with the same weight, metal type, and size can fool several autopay machines (e.g. public phones, autopay parking machines, or auto coin deposit machines at banks). Therefore, studying the design features on the coin surface can prevent falsification and improve the classification of coins. Additionally, computerized solutions are much more efficient, less expensive, require no human expertise, and most of them are portable.

While coin recognition can be viewed as a possible counterfeit coin detection system, there is a fundamental difference between coin recognition and counterfeit coin detection. The coin recognition problem is less challenging than coin authentication where the former requires fewer features to discriminate between different coins. Coin stamp varies significantly between different coins even for different denominations of the same country. High quality counterfeit coins share

most of the coin stamp details as the genuine coins. Therefore, coin authentication tries to have a better understanding of the coin image and captures detailed feature sets to render the misclassification rate. There are several image-based solutions proposed for coin recognition which aim at sorting the coins into their original minting country, time, and denomination.

Coin authentication aims to identify the genuine coins from the counterfeits. In fact, the number of counterfeit coins in circulation is enormous [3]. For instance, the Royal Canadian Mint announced a new device in Aug. 2015, Bullion DNA, to authenticate Gold and Silver Maple Leaf coins that were minted, after 2014 for the Gold coins and after 2015 for the Silver coins, based on the added micro-engraved security marks. There are other examples and reports that demonstrate the need of reliable solutions to automate counterfeit coins detection. In 2015, Mike Marshall, a Canadian coin collector, reported spurious collectable coins for sale on eBay's website to authorities [4]. Likewise, several governmental documents reported the anticipated number of coins and the yearly increase of counterfeit coins. For instance, about 14 million counterfeit £1 coins in the UK were in circulation during 2003 and 2004 [3], while there were more than 47 million spurious £1 coins in 2014 as the Royal Mint estimation reported. Most recently in 2017, the Royal Mint estimated that approximately one in thirty £1 coins in circulation is a counterfeit, leading to replace the current coin with a new one. Coin forgery is not limited to the UK pound; European coins generally are the most targeted due to two reasons: (1) these coins are used in several Euro Union members which makes it easier for the forger to circulate those coins, (2) the Euro coin has higher value than most of other coins.

In 2002, Germany reported a few counterfeit 2-Euro coins holding German and Belgian minting stamps. The same year, the National Central Bank reported the first counterfeit coins in France and Austria. The European Technical and Scientific Centre estimated about 2 million

counterfeit coins were circulated in European countries in 2002. A yearly report by the European Technical and Scientific Centre of the European Commissioner is released showing the yearly increase in number of detected counterfeit coins. In 2016 the number of seized coins exceeded the 2015 number by 3.15% to increase from 236,013 to 243,714.5 Euros in total value. Generally, the total number of detected and removed coins from circulation since the introduction of the Euro coins in 2002 up till 2016 is well over 2 million [5]. Therefore, the demands of an efficient and reliable counterfeit coin detection system are strong.

This research was initiated by the Danish police department through one of the leading digital forensics firms in Montreal. The Danish authorities have provided a number of coins containing both genuine and counterfeit samples. Those coins were scanned using a specialized 3D HD scanner.

1.2 OBJECTIVES

Coins are one of the most widely used currency types around the world. Certainly, coins and paper money are the only government issued currencies for daily trading transactions. The large number of coins in circulation increased the demands of highly reliable solutions to sort and authenticate coins. All existing coin recognition methods extract features from the whole coin image [6], and some of them use image matching to a reference coin [7]. Apart from image matching that requires higher computational resources, extracting features from the whole coin can be a costly process in terms of computational resources. This raises the need for an efficient method to sort a very large number of coins in a timely manner. The method should focus on certain parts of the coin rather than the entire image and extract reliable features to differentiate between different coins.

On the other hand, few studies have been proposed to archive ancient coins by recognizing texts and numbers minted on the coin surface. Those studies, manually or based on *a priori* knowledge, specified the region of interest (ROI) which contains the text. Likewise, other methods proposed for counterfeit coin detection rely merely on the text areas where the authors found significant variations in character edges between genuine and counterfeit coins. These methods extract features from the text edges along with their characteristics after manually specifying the ROI containing text. Therefore, an automated process to locate and segment those characters from the coin image for further analysis and feature extraction is needed. Character segmentation methods of other images (e.g. documents and natural images) are not applicable to coin images due to the nature of the coin image.

Nonetheless, advances in machine learning approaches, viz. deep learning, have made it feasible to extract reliable sets of features and classify images accurately. Several Convolutional Neural Networks (CNN) models have been proposed for different image types. CNN requires no feature extraction engineering to train a classifier, but it requires a large image dataset of hundreds of thousands, if not millions of images, to extract the best features and their weights. Generally, access to that large number of images is rather limited in certain domains including counterfeit coins and medical images. Therefore, transfer learning approaches have been considered to explore the possibility to fine-tune a pre-trained CNN to new image sets. The number of images required to transfer learning is much lower than training CNN models from scratch and it achieves very comparable results to the original image dataset used to train and test the CNN model. The use of transfer learning approaches to detect counterfeit coins can reduce the efforts of extracting features and it can be applied to different coin types.

Finally, the common practice of forgers is to duplicate the coin stamp by either designing a new forged stamp or copying the original by molding the genuine stamp features into a forged one. The features of the counterfeits are very close or almost identical to the genuine in high quality forged coins but are never identical. The strokes of a counterfeit stamp are always the lead to distinguish between genuine and counterfeit coins where even the high-quality forged coins have either wider, taller, detached, or missing strokes. Locating these distinguishable features needs enormous human expertise and time, given the large number of coins and the actual coin size of only a few millimeters in radius. In addition, the text on the coin is minted in different languages and origins. Thus, features should not be limited to a single coin type or language. Pixel-based differences have been used in many areas (e.g. image compression and watermarking) and showed remarkable results in measuring the differences between images based on their pixel values. Numerous methods have been proposed to measure the pixel differences between images based on different factors [8]. Combining the pixel-based and edge-based measures provide a better representation of image information to distinguish between genuine and spurious coins and is applicable to different coin types without the need for large image sets.

1.3 CHALLENGES

Coin images are unlike other images due to the coin nature and use, which increase the challenges in processing and extracting their features. The main challenges to this research can be summarized below:

1. **Coin quality**. Coins have different qualities due to wear and contamination while in circulation. Coins can vary widely from mint state coins (uncirculated) to severely damaged coins. These variations are clearly reflected on the strokes which may appear as

wider or blunt strokes while increasing the variations between coins of the same class. The degradation may diverge from worn out edges to detached strokes into invisible strokes.

2. **Coin orientation**. Coins are different in shapes but modern coins are mostly circular in shape. Due to this circular shape, no prior assumption can be made on coin orientation. Therefore, several well-known image comparison methods are not applicable to coin images.
3. **Lighting source**. Light has considerable effect on the appearance of the coin strokes due to highlight and shadow variations. These highlight and shadow are necessary in emphasizing strokes in coins where they are less than 1 mm higher than the coin background. These variations may result in obscuring some strokes.
4. **Identical color**. Coins have two main components; the foreground (stamp) and background. The colors of the foreground and background are identical in all coins, which makes it harder to differentiate between the two.
5. **Language**. Text, numbers, and symbol are the three main parts of coins. Text is mainly used by Human Vision Systems (HVS) to recognize coins, while numbers are used to identify the minting year. Symbols are the central part of the coin and usually represents something meaningful to the minting country, such as the king/queen face. However, text can be found in different languages based on the minting country. The coin recognition and counterfeit detection systems should be generic enough to be applicable to coins regardless of the country or the language.
6. **Similarity**. Unlike coins, pattern recognition of any other objects relies mainly on defining the similarities between objects and grouping them together in one class. Due to the high

similarity in the patterns of coins, coin recognition and mostly counterfeit coin detection focus on very small differences between coins to differentiate them.

7. **Dataset**. Lots of genuine and counterfeit coins are in circulation in many countries. However, accessing counterfeit coin collection is a highly challenging task. Law enforcement departments would not release counterfeit coins or their images for different reasons. These coins are dealt with as evidences and are not available to the general public. Thanks to the Danish police force who made this research possible by providing access to their counterfeit coin collection.

1.4 CONTRIBUTIONS

This thesis presents four methods crafted for coin recognition and counterfeit coin detection motivated by the literature findings and the collaboration with Danish police force. Figure 1.1 represents a summary of the proposed methods. The first method aims to segment characters from the coin images. Character segmentation is not a new topic and several methods have been proposed. But, with the challenges of the coin images that are discussed in Section 1.3, these methods are not applicable to coins [9]. Coin stamp contains other information than characters such as circular lines at the outer and inner circles of the coin, the coin symbol at the center (e.g. the head profile of the queen on Canadian coins), in addition to scratches and other noises. Therefore, a new character segmentation from coins is proposed and, to the best of our knowledge, this work is the first to address this problem. The proposed method shows promising results in defining the set of characters with different sizes and languages and segmenting them. Due to vulnerability of coin stamp edges, the method also performs additional processing to locate characters with detached or worn out strokes.

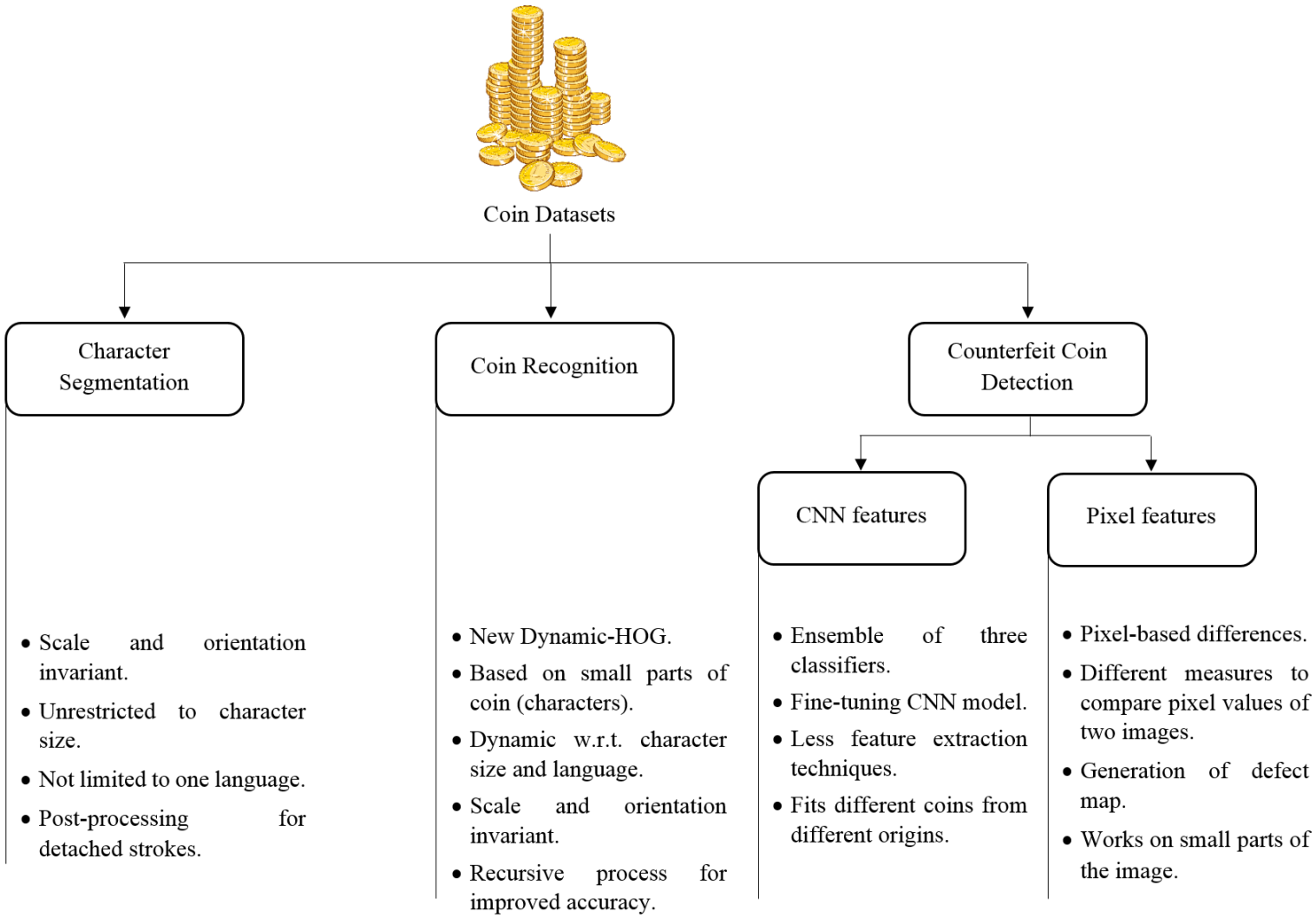


FIGURE 1-1 AN OVERVIEW OF THE PROPOSED METHODS.

An extended work was carried out to improve the well-known feature extraction method, Histogram of Oriented Gradients (HOG) [10]. Traditional HOG is limited by the extensive search process to locate objects in image in order to recognize them. The second method proposes an improved version of the HOG method by introducing a dynamic window in terms of size placed over each object w.r.t. its size. The method is applied to coin images to recognize the characters, while those characters are weighted alphabetically to identify the coin origin. The method works on identifying all candidate Regions of Interest (ROIs) in the image w.r.t. their sizes using the horizontal and vertical projection profiles. A dynamic size grid is placed over all candidate locations and then the HOG is extracted from these locations. The method uses a recursive process to identify detached and missing character strokes. We argue that the method is applicable to other objects given that reasonable edge information is preserved.

We propose two counterfeit coin detection methods based on different features. The third method proposed in this study is an ensemble method consisting of three classifiers. The first two classifiers were trained on convolutional neural network features. Due to the limited number of coin images, a transfer learning of GoogleNet model [11] was performed using a fine-tuning process to update the feature weights to be suitable for the coin images. After the fine-tuning process, the sets of features are used to train the softmax and SVM classifiers, while another classifier was trained based on the features of character strokes. The stroke width, character height and width, spaces between characters, and edge smoothness are the features for the third classifier. SVM was used to train and test those features. Finally, the coin class is decided based on the votes of the three classifiers.

The fourth method is a counterfeit coin detection method based on the pixel-based measures to track the differences between the pixel values of two images, the test and the reference images. Four pixel-based measures are used to distinguish the genuine and fake coins. Those measures are applied

on small areas of the coin where the coin is divided into concentric circular areas and then each circular area is divided vertically in half w.r.t. the centers. The pixel-based measures are found to be effective if applied to small area of the coin rather than the whole coin image. The proposed method suggests that the use of more than one reference coin, to reduce the variations between coins of the same class, enhances the classification accuracy. The evaluation results outperformed the state-of-the-art results and required lower computational cost as discussed later in Section 6.5.

1.5 THESIS OUTLINE

This thesis is organized as follows:

- Chapter 2: In this chapter, we will thoroughly discuss the computer vision methods to handle different coin problems. The existing research studies on different coin applications i.e. coin recognition, grading, and counterfeit coin detection are discussed in this chapter.
- Chapter 3: It introduces a brief overview of character segmentation from coin images, and the details of our proposed system for character segmentation from coins. This chapter presents the experimental work and its results to evaluate the proposed method, followed by a conclusion.
- Chapter 4: In this chapter, we define the coin recognition problem, and discuss in details the proposed coin recognition method using characters minted on the coin surface. The experiments, results discussion, and a comparison of our proposed method with other methods in the literature are presented.
- Chapter 5: It discusses an ensemble method which combines the outputs of three classifiers to reject counterfeit coins. It details the proposed method for counterfeit coin detection,

and illustrates the experimental work and evaluation results we achieved of the proposed method.

- Chapter 6: The differences between coin recognition and counterfeit coin detection systems are outlined in this chapter. The details of our proposed method of pixel-based feature extraction and image matching to detect spurious coins are discussed. Then, the proposed method and its parameter settings evaluation is discussed. Finally, a comparison between the results of our proposed method and those of related methods is presented to show the effectiveness of the proposed method.
- Chapter 7: This chapter reviews our research methods and summarizes how they were addressed. It then describes the directions for future work.

CHAPTER 2

LITERATURE REVIEW

Coins have been receiving increased attention by machine vision and intelligence researchers in recent years. Numerous studies have been presented in the literature for coin recognition [12, 6, 13, 14, 15, 9], grading [16, 17, 18], and authentication [19, 20, 21] systems. These studies proposed different methods and are mainly dependent on three steps: *coin segmentation*, *feature extraction*, and *classification*. Coin segmentation is the first step where the actual coin is located and segmented from the whole image. Feature extraction is the core step where several feature extraction methods have been used on coins such as *edge-based statistical features* [22], *local image features* [23], and *texture features* [13]. Finally, different machine learning methods have been used for classification such as *Bayesian fusion* [24], *K-nearest neighbor* [22], *neural network back propagation* [25], and *decision tree classifier* [26].

Generally, coin recognition systems were proposed for two types of coins: modern coins and ancient coins. Figure 2-1 shows samples of modern and ancient coins. Obverse and reverse sides are the two faces of the coin where the obverse refers to the front face and reverse refers to the back face of the coin. There are few differences between the two types where the ancient coins have some shape irregularities due to the manual minting process. Few authors suggest that recognizing ancient coins are somewhat more challenging than modern coins due to the minting process, where coins of the same class can vary based on the minting master [27]. However, research papers in the literature have used the same feature extraction and classification methods for both modern and ancient coins. Contrarily, the coin segmentation methods are different in

ancient than modern coins due to shape variations in ancient coins; a further discussion on this topic in Section 2.1.

Coin recognition is more challenging than for other objects such as pedestrian or face recognition, as discussed earlier in Section 1.3. These challenges are even more difficult for counterfeit coin detection. This chapter presents the coin recognition and counterfeit coin detection concepts together with the different methods proposed to recognize coins and identify spurious ones.

(A) OBVERSE SIDE OF OTHO SILVER DENARIUS

(B) REVERSE SIDE OF OTHO SILVER DENARIUS

(C) OBVERSE SIDE OF CANADIAN $2 COIN

(D) REVERSE SIDE OF CANADIAN $2 COIN

FIGURE 2-1 SAMPLE IMAGES OF ANCIENT AND MODERN COINS.

2.1 COIN SEGMENTATION

Coin segmentation is the process of defining the contour of the coin and masking the coin to segment it out of the image, and is also known as coin detection and coin scaling. A simple threshold operation is not suitable for coin segmentation due to the complexity of the background and the color variance between the coin and the background. Consequently, the coin segmentation requires an efficient segmentation method to find the coin contour. Therefore, there are two main methods used for coin segmentation: the *edge-based methods* [22] and *Hough transformation* [28], the latter is also used after edge-based methods to find the center and the radius for further analysis. Once coins are located, the center point and the radius are defined, coin images are scaled and background pixels are removed. Center point and radius are used for further analysis and feature extraction.

Zaharieva *et al*. [29] proposed an image-based ancient coin recognition. In this paper, the authors discussed two coin segmentation methods. Edge-based segmentation and Hough transformation methods were applied and the authors argued the capability of edge-based method to segment ancient coins but not the Hough transformation. The authors stated that it is due to the irregular shapes of ancient coins, which tend to be, but are not exactly circular. However, for modern coins, the Hough transformation method outperformed their edge-based method. In addition, a few coin specific methods have been proposed for coin segmentation, such as the method proposed by Pan *et al*. [30] that discussed a new segmentation method using an active model. However, those methods have not been adapted by any other researcher performing coin segmentation.

2.2 COIN RECOGNITION

Coin recognition is a process carried out by several systems, including the central bank for statistical purposes and other systems such as public phones, vending machines, and parking meters. Historically, coin recognition was carried out by humans to differentiate coins of different origins or denomination. With the increased number of coins in use, sorting large numbers of coins requires exhaustive human expertise and time. Thus, automated machines were developed to sort coins based on physical characteristics such as weight and size. These machines have limitations in rejecting metal pieces carrying the same physical characteristics as coins due to the absence of visual information. Hence, several image processing and machine learning methods have been proposed to take advantage of the high computational abilities of computers in handling large numbers of coins. The use of the information appearing on coins by computer vision methods have increased the recognition accuracy significantly compared to traditional features.

In 2003, the ARC Seibersdorf research center proposed the Dagobert system for coin recognition [31]. The Dagobert system was developed to sort coins belonging to 12 European countries before the introduction of the Euro coin. The actual coins were donated to charitable organization after scanning them. The images of these coins were publicly available for scientific research as the MUSCLE Coin Image Seibersdorf (MUSCLE CIS) benchmark dataset. The 2006 and 2007 have seen the peak number of studies on coin recognition after the two large competitions were held during these two years to sort the MUSCLE CIS dataset. Several methods were proposed in the two competitions, based on various feature extraction methods. Soon after, researchers have shown interest in recognizing ancient coins, which was claimed to be more challenging.

The vast majority of coin recognition methods were based on the three steps discussed earlier in this chapter. The feature extraction step is still the most challenging, since recognition accuracy highly depends on finding a suitable set of features to represent the image's information. Several methods have been used to extract edge-based features, local image features, or texture features from the coin's stamp.

2.2.1 EDGE-BASED STATISTICAL FEATURES

Edge-based statistical features is one of the feature extraction methods introduced in [29, 22]. It deduces the contour and edge distribution over the coin. In 2003, Nölle *et al*. [31] proposed a system to automatically recognize and sort large numbers of coins from different countries. The proposed method uses edge-based features as well as coin thickness and rough diameter sensors to classify coins. The authors first used edge information to determine the rotation of the test coin using fast correlation method, then three edge-based features to decide the coin class. Maaten *et al*. [22] proposed two edge-based statistical features in 2006: *distance distribution features* and *angle distribution features*.

Distance distribution features are applied to several circles around the center of the coin. In each circle, the number of edge pixels is counted and the distance from each edge to the center is registered and stored in a distance distribution histogram. On the other hand, *angle distribution features* are represented as pies from the center to the outer border of the coin. Similar to distance distribution features, the edge pixels are counted and stored in an angle distribution histogram. However, the main difference between distance and angle distribution features is the rotation: the former is rotation invariant while the latter method is rotation variant [22]. In addition, *angle-distance distribution features* combine the distance and angle distribution features of the edge

pixels. Angle-distance distribution features have also been used in the literature for classifying ancient coins in [29].

2.2.2 LOCAL IMAGE FEATURES

Various coin recognition systems based on *local image features* have been proposed [13, 32, 33, 34]. Local image features describe the image regions and properties in specific interest points. The *Scale-Invariant Feature Transform (SIFT)* proposed in 2004 by David G. Lowe [35] has been widely used in coin recognition [13, 33, 36, 7]. SIFT is scale and rotation invariant by definition, which makes it suitable for coin recognition systems. The SIFT method relies on the local gradient distribution for an image at different scales, while the orientation is nominated based on the peak histogram of local gradient [35]. Each SIFT feature has a location, scale, and orientation associated with it. Huber-Mörk *et al*. [37] proposed a two-stage method, where the first stage is circular shape matching to compare the shape of the coin with shape of a circle. The feature vector of this stage is a one-dimensional vector of the distance between the coin border pixels and the circle border pixels that is placed over the coin. The second stage is to extract local image features using peaks in the *Difference of Gaussians (DoG)* scale space and SIFT keypoints.

In 2008, H. Bay *et al*. [38] introduced the *Speeded-Up Robust Features (SURF)*. SURF is a scale and rotation invariant method with reduced feature vector size, where each SURF feature descriptor has 64 dimensions instead of 128 as in SIFT. Thus, it requires less comparison and computation time. M. Kampel *et al*. [32, 33] utilized the SURF feature extraction method in coin recognition systems. SURF works on determining the orientation by considering the neighborhood of the interest point. The SURF descriptor represents the distribution of Haar wavelet responses of interest point's neighborhood.

In 2002, Belongie *et al*. [39] proposed *Shape Context* as a feature descriptor. It works on locating the remaining points of each edge pixel and reconstructs the shape of each object as a set of connected points around the internal and the external boundaries of the object. Shape context has also been employed in [32] for ancient coin recognition.

In 2013, Anwar *et al*. [34] proposed another coin recognition method using the *Bag of Visual Words (BoVWs).* BoVWs considers local and statistical characteristics of the coin's texture. Anwar *et al*. applied the BoVWs method into three representations, where they divided the coin's image into circular, rectangular, and log-polar areas. Finally, BoVWs is decided by combining the BoVWs of the three areas.

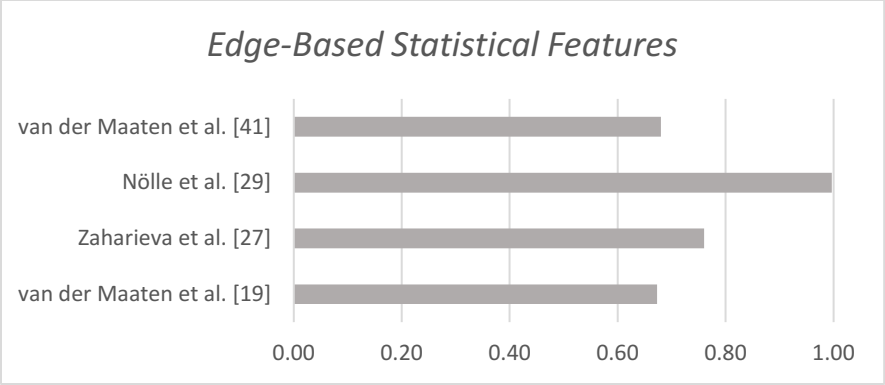
Kampel *et al*. [32] studied different local image features. The authors detect several interest points from ancient coin images and represent those interest points using local descriptors. The *Harris corner*, *Hessian-Laplace*, *Hessian-Affine*, *fast-Hessian*, *geometry-based region*, *intensity-based region*, and *difference of Gaussian* interest point detectors were explored in this study. The local feature descriptors covered by this study include Scale-Invariant Feature Transform (SIFT), *Gradient Location and Orientation Histogram (GLOH)*, shape context, and Speeded-Up Robust Features (SURF). The research paper shows an extensive evaluation conducted by considering each descriptor with all interest point detectors. The SIFT and GLOH experimentally outperformed the other descriptors.

2.2.3 TEXTURE FEATURES

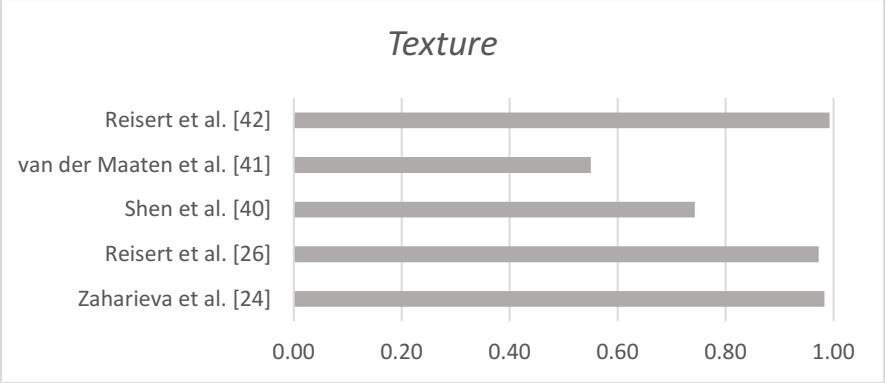
In addition to edge-based and local features, features based on the coin's texture have been proposed. Xu [40] introduced a system based on *gray level co-occurrence* matrices to extract

texture features. It relies on statistical estimation of the spatial arrangement for estimating the gray level co-occurrence. *Gabor feature* is another feature extraction method applied to coins through the literature [41]. Shen *et al*. [41] claimed that local textures are better represented using the Gabor wavelet feature and that it shows better robustness against noise. On the other hand, *gradient features* have also been introduced in coin recognition systems [28]. Reisert *et al*. [28] used the direction of the gradients and abandoned the magnitude to increase robustness against illumination and changes in image contrast. Maaten *et al*. [42] discussed the use of two contour features and two texture features as well as different combinations of the contour and texture features. Authors used multi-scale edge angle histograms and multi-scale edge distance histograms for contour features representation. On the other hand, they used the Gabor wavelet and Daubechies wavelet for the texture features. The study has used two datasets for evaluation: modern coin dataset and ancient coin dataset. The study shows that better results were achieved using contour feature, where as high as 68% accuracy rate obtained using contour feature while texture features achieved 55%. The study also suggests the recognition of modern coins achieves better accuracy than ancient coins.

Shen *et al*. [13] used the Gabor wavelets and *local binary pattern (LBP)* to extract texture features. The method works on rotating the coin image using image matching against a training coin image. The coin image is also divided into smaller areas, then the statistics of Gabor coefficients or LBP values are obtained. The authors have not used a classifier to sort the coins based on those features, but two distance measures instead. The Euclidean distance and normalized correlation, were used to measure the distance between two texture features extracted from the coin image and the reference coin image.



(A) ACCURACIES OF DIFFERENT EDGE-BASED FEATURES



(B) ACCURACIES OF DIFFERENT TEXTURE FEATURES

(C) ACCURACIES OF DIFFERENT LOCAL IMAGE FEATURES

FIGURE 2-2 ACCURACIES OF MOST CITED COIN RECOGNITION RESEARCH PAPERS

Figure 2-2 depicts the accuracies of some coin recognition systems based on three feature extraction methods: (a) edge-based statistical features, (b) texture features, and (c) local image features. The variations in accuracies between the different methods are not limited to the

efficiency of the feature extraction method used but also due to (1) the dataset used where the size of the dataset, image clarity, and whether it contains modern or ancient coins, (2) the number of classes in each dataset (the larger the number of classes, the lower the accuracy rates), and (3) the preprocessing step including noise reduction and coin segmentation.

Few research papers have proposed different feature extraction methods other than those discussed earlier in this chapter. Jain *et al*. [43] proposed a coin recognition method which focuses on sorting the coins based on their denomination only. The proposed work uses the *circular Hough transformation* method to locate the circular shape of the coin and then to define a set of coin boundary edge points. Using the boundary edge points, the method defines the radius of the coin. The authors classify the coins into their denomination based on their radii. The method achieved 94% accuracy while the number of coins used for the evaluation has not been reported. The method is still based on a physical characteristic of the coin which is the size. This means that any coin or non-coin objects, having the same size as the tested coin, can easily trick the proposed method. Allahverdi *et al*. [44] proposed a new coin recognition method for ancient coins. Their method uses *Discrete Cosine Transform (DCT)* features to train the SVM classifier. Before extracting the DCT features, a preprocessing method to segment the coin is applied. A Sobel operator is used to find the coin border edges and several morphological operators are applied to segment the coin image. The authors worked on Sasanian coins and achieved 86.2% accuracy.

Huber *et al*. [24] proposed another coin recognition system based on a two-stage process. In the first stage, the method computes a rotationally invariant description by estimating the rotational angle based on cross-correlation of the coin with a reference coin. The second stage is where the eigenspace is selected based on the diameter and thickness measurements (using

thickness sensor). The method then trains a Bayesian fusion on eigenspace features from both obverse and reverse sides. The overall reported accuracy was 93.23%.

All of the methods discussed above depend on extracting different sets of features from the whole coin image. Given the goal of introducing machine vision methods for coin recognition, finding a method that works on smaller areas of the coin to extract features instead of the whole coin could significantly reduce the resource and computational costs. Coins of different denominations, minting times, or origins have significant variations, therefore focusing on small areas of the coin that carry the most variations can render the feature vector size and provide better representation of the coin information.

C*lassification* is the last step in a coin recognition system that determines the coin denomination, minting time, or origin. [31, 32, 34] employed image processing techniques to make the decision while [13, 22, 25, 26] used machine learning methods such as Neural Network Back Propagation [25], K-nearest neighbor [13, 22], Bayesian fusion [24], or decision tree classifier [26]. Few papers have proposed coin recognition systems based on machine learning methods without a feature extraction process. Khashman *et al*. [45] proposed a coin recognition system based on neural networks where no feature extraction method was required. The research article uses a 3-layer back propagation neural network with binary output as either 0 or 1. The method only classifies two coins: 2-Euro and Turkish 1-Lira are the two coins used for training and testing the system. The method performs a set of preprocessing steps to prepare the coin images for the neural network, while the back propagation neural network consists of 400 input neurons, 25 hidden neurons and 2 output neurons. The authors stated that the experimental evaluation of their method achieved as high as 100% accuracy.

Finally, Capece *et al*. [46] proposed a new coin recognition method for mobile devices. The method uses a mobile device to capture an image of the coin and the client-server architecture to transfer the image to a remote server for recognition. A *Convolutional Neural Network (CNN)* is used on the server side to extract features and classify the coins automatically. The authors reported that their proposed method achieved as high as 100% accuracy.

2.3 CHARACTER SEGMENTATION FROM COINS

Character segmentation and word recognition is the focus of many researchers around the world. Research articles have reported the work conducted on different types of documents in various languages using various features and approaches. Compared to structured and hand-written documents, character recognition from coin images is more challenging, because the contrast between the character strokes and the background is minimal. Also, the size, font, spacing, rotation, and location of characters may vary on the same coin and between different coins. Different methods have been proposed to combine different feature extraction and machine learning methods to recognize characters of several languages. While much research has been conducted on character recognition on documents [47, 48, 49], very few researchers have tried to recognize characters on the coin surface [15, 50, 51].

The traditional character extraction methods, sorted and presented in [52], are appropriate for printed and handwritten documents, but not for natural scene or coin images [53, 54, 55]. Several improved methods have been reported on character or line segmentation, including contour tracing [56], morphological techniques [57], grayscale feature combination [58], graph-based techniques [59], Neural Network (NN) [60], and vertical and horizontal projection profiles [61]. While character segmentation from coins is essential to some counterfeit coin detection, the

methods are not suitable for coin images due to the challenges of characters on coins [15]. Some coin authentication methods (e.g. [19] and [20]) rely heavily on character properties to identify genuine coins and reject counterfeit ones. The proposed methods in these research articles consider the contour width, alignment, and spacing between characters to identify genuine coins. Yet, one performed manual character segmentation for feature analysis while the other discussed no character segmentation method.

Moreover, most coin recognition methods that have been reported are based on extracting features from the whole test coin. A few researchers worked on word recognition on coins by extracting features from character locations on the coin [15, 9, 50]. Recognizing text minted on coins has several advantages, including archiving a large number of ancient coins. Nonetheless, the researchers extracted features such as SIFT features from either a full word or character, after locating the word or the character on the coin manually, or based on a *priori* knowledge of the text location. Then the features are used to identify the words and characters. Kavelar *et al.* [15] proposed a method to spot the text on ancient Roman coins. The SIFT descriptor is used to find key points from *region of interest (ROI)* considered as the *candidate text location*. The authors check these *candidate locations* against the manually extracted SIFT features of 18 alphabets classes that exist in their Roman coin dataset. The method is proposed to recognize words based on their characters and the reported accuracy in their paper was between 29% - 53% for the limited dataset of 180 images of Roman coins.

The authors extended their work in [9] to recognize the words from a cropped candidate location from the coin image rather than word recognition from the whole coin image that was carried on in [15]. The authors have specified and cropped the ROIs containing text based on a

priori knowledge of their limited dataset. Sets of SIFT keypoints were obtained from the cropped text locations. The aim of their new method is to recognize characters and words written on Roman coins. The method compares the extracted SIFT keypoints from annotated text areas, and compares it to a set of SIFT keypoints of their manually extracted SIFT keypoints of Roman coin alphabets (centered to each character). In both papers, the SVM training process was performed on the manually extracted SIFT features and test the SIFT keypoints obtained from candidate locations. The method in [11] returned a 37.8% accuracy rate when used to recognize words. Zambanini *et al*. [50] introduced a combination of image matching and text recognition to recognize ancient coins. The proposed method uses the same work discussed in [15] to recognize text and combine it with image matching to enhance the coin recognition accuracy. The authors performed an evaluation for each method separately and compared it to the results of the combined method. The image matching has better accuracy than text recognition, while the combined method has outperformed both methods separately. Unlike previous works, O. Arandjelović [62] proposed another character recognition method based on gradient features. Small windows are placed on the obverse side of the coin. For each window, the HOG-like descriptor is found and compared against a manually annotated dataset to recognize characters. The authors also discussed the maximum likelihood of an optimal window placement to cover the whole character. The proposed solution combines image matching (for the reverse side) with character recognition (for the obverse side) to achieve a better coin recognition rate. However, the evaluation of the system was carried out on 25 coin images belonging to ancient Romans. More specifically, the system covered only the ancient Denarii coin. The recognition rate of the 25 coin dataset was not reported in this paper.

On the other hand, Pan and Tougne [51] proposed a solution for coin date detection. The proposed system requires a *priori* knowledge about the coin to locate the date. The histogram is

computed for gradients of each candidate location and is then compared to the mean histogram of synthetic date models. The system was evaluated on US coins with 900 images of individual date numbers. The recognition rate was 44% at its best. However, date segmentation has not been reported in the above-mentioned research papers and an automated method for character segmentation from coin images has not been discussed.

2.4 COUNTERFEIT COIN DETECTION

Counterfeit coin detection has always been a challenging task for law enforcement departments despite the common belief that duplicating coins is a worthless task due to the low face value of coins compared to paper notes. As discussed in Chapter 1, several reports have been released showing that hundreds of thousands of coins are counterfeit and several governments have started taking other measures to protect coins. An example is the new device introduced by the Royal Canadian Mint, Bullion DNA, to authenticate Gold and Silver Maple Leaf coins. The UK sets another example where the law enforcement departments have lacked countermeasure tools to control the counterfeit coins. The UK replaced all existing coins by a newly designed coin having some security features. The European Commission, on the other hand, has established the European Technical & Scientific Center in 2004 whose task is "*to analyze and classify every new type of counterfeit euro coin*" [5]. Figure 2-3 shows a few examples of genuine and counterfeit coins from several origins. The counterfeit coins of different countries have mismatching edges compared to genuine coins and they also have coarse noise on the background, as it appears clearly on the 2-Euro coin. Generally, for high quality counterfeit coins, there are two possible scenarios to duplicate a coin's stamp by either placing special material on the coin to copy the stamp or designing a similar stamp. In both cases, the forgery would oversight some details and small parts of the design would be missing. For Instance, along with the image in Figure 2-3(a), Canadian

authorities reported that under the magnifying glass, the bill of the loon is clearly detached from the bird's body on the Canadian $1.

Studies have been conducted on counterfeit coin detection carried out by researchers from different fields such as physics, chemistry, engineering, and computer science. Most of these studies have investigated various physical features to authenticate coins such as metal features, thickness, diameter, size, weight, conductivity, electromagnetic field properties, and piezoelectric characteristics. Size and weight can be the leading features in current automated systems such as public phone or auto parking machines.

Moreover, other researchers [63], for example, authenticate coins by using the frequencies obtained from a test coin to distinguish between spurious and genuine coins. The experimental work was conducted on real coins of 50 cents, 1- and 2-Euro denomination of euro coins. Authors [64, 65, 66] have argued the common use of acoustic validation techniques, especially in vending machines, because of the high accuracy rates of this technique. However, these frequencies rely mainly on the type of metal. Thus, if the same metal was used for the spurious coins, it will pass this test. Other methods based on X-ray fluorescence have also been proposed [67].

The X-ray methods are used to analyze the coin elements, which, in return, are used to authenticate coins. In addition to X-ray fluorescence, Denker *et al*. [68] proposed a proton induced X-ray emission (PIXE) method to authenticate coins. Generally, X-ray methods are criticized for their harmful nature and the use of X-ray fluorescence is impractical and expensive [25]. In addition, these methods are also based on characteristics of metal type used.

(A) GENUINE AND COUNTERFEIT 1-CANADIAN DOLLAR[1]

(B) COUNTERFEIT 2-EURO COIN[2]

(C) GENUINE AND COUNTERFEIT US DOLLAR

(D) GENUINE AND COUNTERFEIT DANISH KORNER

FIGURE 2-3 GENUINE AND COUNTERFEIT COINS FROM DIFFERENT COUNTRIES.

[1] http://morinvillenews.com/2012/02/07/counterfeiting-a-multi-denominational-affair/
[2] http://www.boards.ie/vbulletin/showthread.php?t=2056584474

Wang *et al*. [25] proposed a machine learning method that is based on two steps. In the first step, the test coin is rotated in all possible angles to match a reference coin using an image matching technique. The second step is to measure the relative distance between any given two points on the test coin and to compare it to the distance of the two corresponding points on the reference coin. The distance measures are then fed to a Back Propagation Neural Network (BPNN). The researchers used two sets of coins containing genuine and fake coins of the Japan Meui Year 29 (JMY) 1 Yen silver coin and Taiwan Year 1950 (TY) 2 Dime aluminum coin. The JMY coin dataset contains 16 genuine and 41 fake coins while the TY coin dataset has 10 genuine and 42 fake coins. However, the experiments were conducted on a considerably small number of coins, which allowed the authors to carefully divide the coin into smaller areas of interest of test and reference coins to carry out the comparison. The accuracy rate was 100% for the genuine and fake 1 Yen silver coins and also for the fake JMY coins while the genuine JMY coin returned 97% by misjudging one coin.

Tresanchez *et al.* [69] proposed an image-based counterfeit coin detector using an optical mouse. The optical mouse is used to partially capture the image of 2-Euro coins and to compare it to a set of reference coin images. The authors argued the benefit of using an optical mouse for its small size, low cost, and that its usage requires no expertise. However, the authors stated clearly the limitation of the optical mouse which captures a very small part of the 2 euro coins (about 1/14 of the whole coin). Comparing this small area would highly render the accuracy rate and could easily fault the classification.

Sun *et al.* [19] proposed a new counterfeit coin detection method based on combining the contour and local image features. Stroke characteristics such as the stroke width, height and width of characters, relative distances, and angles between characters were used as contour features. Local image features were extracted using the *Maximally Stable Extremal Region (MSER)* method

to match the test image with reference genuine and counterfeit coins. Although the experimental work was promising, the dataset on which the experiments were conducted was very small. The method claims no guarantee to fit other coins due to the very limited experimental dataset of 13 coins.

Liu *et al.* [21] discussed another counterfeit coin detection based on local image features. The authors compare a set of SIFT keypoints of the test image with a set of SIFT keypoints of reference coins. The SIFT descriptor is used to represent keypoints and the comparison results between a test coin with each reference coin is stored as a vector in a dissimilarity space. The authors have improved the keypoints selection process to limit the number of mismatched keypoints. However, the keypoints of a high-quality fake coin could trick the classifier. Additionally, the mechanism of how keypoints work illustrates the limitation of using such a measure to detect counterfeit coins. Nonetheless, Khazaee *et al.* [20] presented a counterfeit coin detection technique based on the 3D coin image features. The authors examined the outer ring of the coin containing characters and numbers. The height and depth information of characters and numbers obtained from the 3D images were used to authenticate coins. The method works on transforming the coin image into a new rectangular image. Then, height and depth information from each row are used as features to train the classifier. Although the reported accuracy rates in this paper was promising, the use of 3D scanner is expensive, time consuming, and requires expertise for its usage.

CHAPTER 3

CHARACTER SEGMENTATION

The increased focus on text regions contained in coins to extract their features and the properties of their characters (e.g. stroke width) to achieve higher accuracies have raised the need for a reliable character segmentation method. The traditional character segmentation methods for printed/handwritten documents and natural scenes have some assumptions that are not valid for coin images. Due to these facts, we propose a new character segmentation method based on vertical and horizontal projection profiles and dynamic adaptive mask. The method starts by scaling the coin image to fit the whole image by removing the left, right, top, and bottom margins. Then, straightening the circular shape of the coin into a rectangular shape. The horizontal and vertical projection profiles are calculated to define the average height and width of characters, then a dynamic mask is created to separate the characters.

3.1 COIN SCALING

Coin scaling is an essential step in coin recognition, grading and authentication systems. A few researchers have ignored this step and simply applied a threshold to discriminate the coin pixels from the background ones, relying on the large variance between the image background and the coin colors. However, this is not always the case and therefore, we consider coin scaling to fit the entire image. Several research articles have performed coin scaling, also called coin segmentation. The goal is to scale the coin to fit the whole image and to remove the margins outside the coin borders to facilitate feature extraction (as show in Figure 3-1(c)). In this research, we

perform coin scaling in three steps: (1*) edge detection*, (2) *morphological operations*, and (3) *Circular Hough Transform (CHT)*. Figure 3-1(a), (b), and (c) illustrate the coin scaling algorithm. The scaling algorithm detects all edges Ɛ in the image $I(\chi,\gamma)$ using *Sobel edge-detection* with a dynamic threshold [70]. The edge image I_ε is then dilated using a morphological structuring element (e.g. circular shape). Then CHT works on a 3-dimensional space where 2 dimensions represent the circle center (C_χ, C_γ) and the third dimension is the radius (R), as given in the equation (3.1) below.

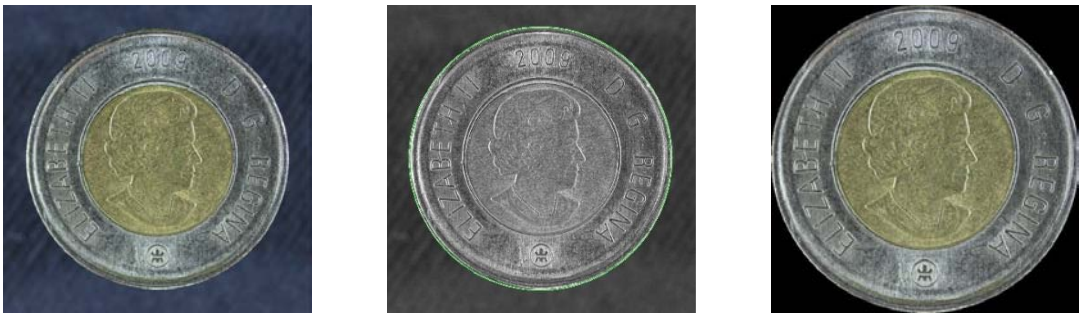
$$(\chi - C_\chi)^2 + (\gamma - C_\gamma)^2 = R^2 \tag{3.1}$$

The algorithm starts by identifying all circular shapes in the image and defining a set of center points (C_χ, C_γ) $\in I(\chi,\gamma)$ and radii R for each circle. The largest radius $r \in R$ and its corresponding center point ($C_{\chi r}$, $C_{\gamma r}$) are selected as a candidate circle of the coin. An adaptive mask $\mathcal{M}$ of size (χ,γ) is used to mask the coin. The circular shape of the coin is defined in the mask by equation (3.2):

$$\mathcal{M}(m, n) = 1 \text{ iff } (R_{max} - 0.1)^2 < m^2 + n^2 < (R_{max} + 0.1)^2$$

$$\mathcal{M}(m, n) = 0 \text{ otherwise} \tag{3.2}$$

The adaptive mask is then placed over the actual coin; the coin is cropped to fit the whole image to yield a new image. Due to the circular shape of the coin, there are remaining areas where the background can be seen. Hence, during the masking operation, the background pixels outside the actual coin pixels are set to zero as shown in Figure 2-1.



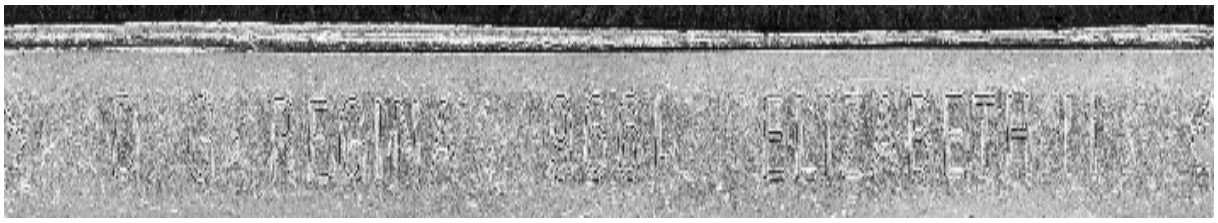
(A) CANADIAN COIN (C) LOCATING THE COIN BORDERS (E) MASKED AND SCALED COIN

(B) DANISH COIN (D) LOCATING THE COIN BORDERS (F) MASKED AND SCALED COIN

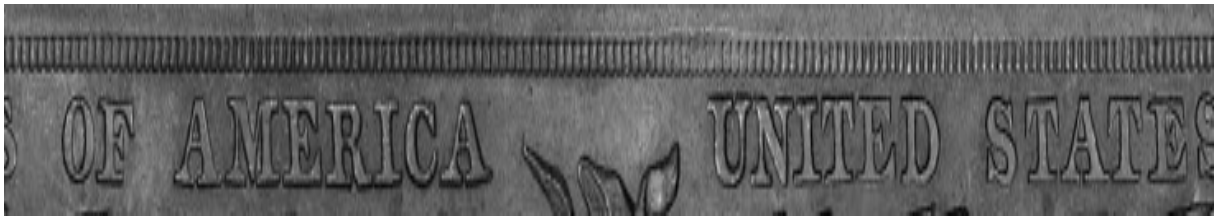
FIGURE 3-1 MASKING AND SCALING PROCESS OF CANADIAN AND DANISH COIN IMAGES.

3.2 STRAIGHTENING ALGORITHM

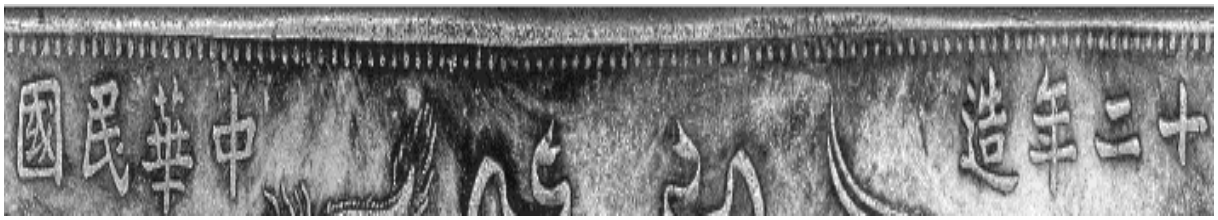
The aim of this algorithm is to overcome the orientation problem when extracting the characters. Transforming the circular shape of the coin into a rectangular shape facilitates the character segmentation process using projection profiles. Since the focus of this research is on characters on the coin surface, the straightening algorithm will consider the outer circle of the coin where the characters are located. The straightening algorithm reads the image pixels in diagonally and writes each pixel value in a new rectangular matrix.



(A) STRAIGHTENED CANADIAN COIN



(B) STRAIGHTENED US COIN



(C) STRAIGHTENED CHINESE COIN

(D) STRAIGHTENED DANISH COIN

FIGURE 3-2 SAMPLES OF STRAIGHTENED COINS USED IN OUR EXPERIMENTS.

Given the circle equation, assume $\rho(\chi, \gamma)$ is a pixel from the original image $I(h, w)$ at coordinates χ and γ, and $\overline{\rho}(\overline{\chi}, \overline{\gamma})$ is the $\rho(\chi, \gamma)$ value in the new pixel location at $\overline{\chi}$ and $\overline{\gamma}$ of the straightened image $I_s(h_s, w_s)$. The $\overline{\chi}$ and $\overline{\gamma}$ values can then be calculated by equation (3.3):

$$\begin{aligned} \overline{\chi} &= n * sin(z) + (w / 2) \\ \overline{\gamma} &= n * cos(z) + (h / 2) \end{aligned} \qquad \begin{cases} \pi < z < -\pi \\ \text{t} < n < \text{q} \end{cases} \tag{3.3}$$

where w and h are the width and height of the original image respectively. $\frac{t}{2}$ is half of the coin image width and $\frac{q}{2}$ is the width of the outer circle of the coin containing text. The straightening algorithm is an essential step in our method to define the thresholds of characters height and width. Figure 3-2 (a), (b), (c), and (d) show the straightened images of Canadian, US, Chinese, and Danish coins, respectively.

3.3 ANALYSIS AND SEGMENTATION OF CHARACTERS

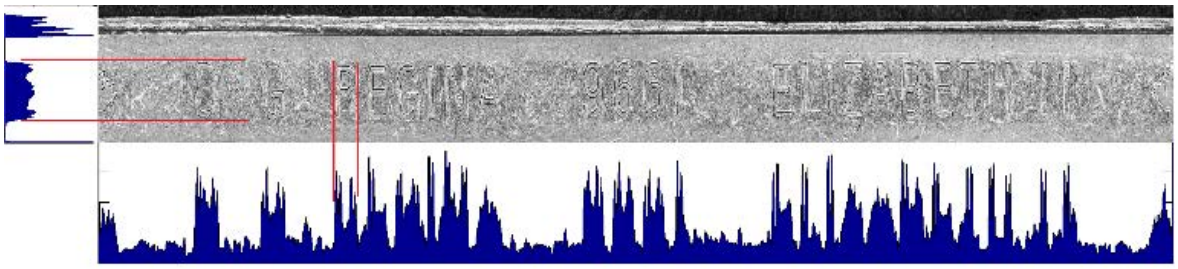
The straightened text area of the coin yields a better representation of the spacing between characters, and facilitates the use of horizontal and vertical projection profiles to detect characters. Projection profiles have been widely used in text line segmentation from documents [61]. In this research, the projection profiles method starts analyzing the text area of the coin after the straightening algorithm has been applied. The coin analysis allows the segmentation algorithm to learn the properties of the coin and to refine the parameters and thresholds for character segmentation. A 3 x 3 Gaussian filter is applied to remove noise from the coin and a Sobel filter algorithm [71] is used to obtain the edge image. Then, all connected components (blobs) are identified on the coin.

The vertical projection profile $f(x,p(x))$ is calculated to find the characters distribution on the coin, and the average width w_{avg} of connected components is found to set the average width of the dynamic mask. The horizontal projection profile $g(y,p(y))$ is also calculated to set the average height h_{avg} of the dynamic mask. The slopes between each peak and the next valley in the vertical and horizontal projection profiles are used to determine the spaces between connected components. The slopes in the vertical projection profile represent the spaces between characters, whereas the

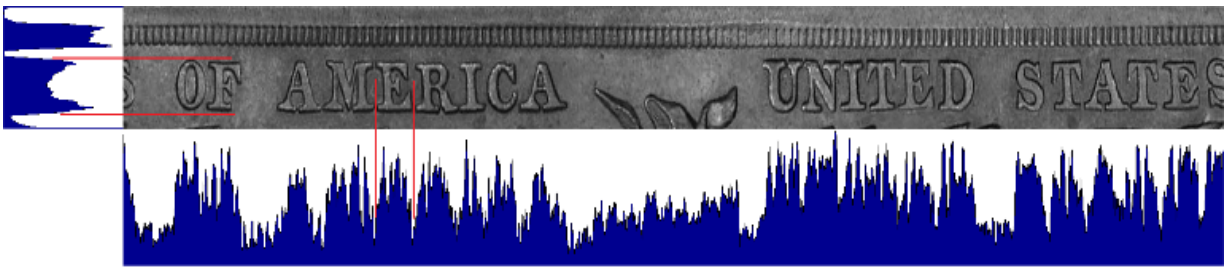
slopes in the horizontal projection profile depicts the distances between characters and other objects on the coin e.g. the outer and inner rims. The slopes are calculated by equation (3.4) below:

$$slope_{vertical} = \frac{p(x_{peak}) - p(x_{valley})}{x_{peak} - x_{valley}} \quad (3.4)$$

where $p(x)$ is the total number of edge pixels in column x, and x_{peak} and x_{valley} are the coordinates of columns where the peak and the valley are located. Figure 3-3 illustrates the vertical and horizontal projection profiles for the two different coins. The h_{avg}, w_{avg}, $slope_{vertical}$, and $slope_{horizantal}$ are defined and passed to the character segmentation algorithm.



(A) HORIZONTAL AND VERTICAL PROJECTION PROFILES OF A CANADIAN COIN



(B) HORIZONTAL AND VERTICAL PROJECTION PROFILES OF A US COIN

FIGURE 3-3 HORIZONTAL AND VERTICAL PROJECTION PROFILES OF US AND CANADIAN COINS.

The four parameters (h_{avg}, w_{avg}, $slope_{vertical}$, and $slope_{horizantal}$) that we obtained from the coin analysis are used to initiate the character segmentation algorithm together with the edge image of the coin. The algorithm checks for every connected component *lm* that falls within:

$$h_{avg} + c_1 > h_{lm} \geq h_{avg} - c_1$$

$$w_{avg} * c_2 > w_{lm} \geq w_{avg} / c_2$$

where c_1 and c_2 are constant thresholds. c_1 is the difference between h_{avg} and h_{max}, where h_{max} is the maximum height of all characters obtained from the horizontal projection profile; and $c_2 = 4$. Note that the use of different arithmetic operations for defining the range of the height and the width is due to the fact that the variation in height is much less than the variation in width from one character to another.

The algorithm starts the segmentation process by placing an adaptive mask m around each connected component that has the height and width within the range and discarding the ones that exceed the range. The mask is adaptable to the size of each character that falls within the acceptable range of characters and is shaped accordingly to cover the exact size of the character. To improve the character segmentation method, a recursive process is considered for detached character strokes. The recursive process checks every two immediate connected components that are to the right, left, above, or below each other and the sizes of both are below the maximum height h_{max} and width $w_{max}$of characters. Those two connected components are bounded by one mask for segmentation. The final step before segmentation is to check the size of the mask against the maximum height and width of characters so that *iff* the mask has $h_{max} > h_m > h_{avg-c1}$ and $w_{max} > w_m > w_{avg/c_2}$, then the character in this mask is segmented.

3.4 EXPERIMENTAL RESULTS

Our method was tested on 6 datasets containing 6 different coins from four different countries as discussed in (Section 3.4.1). The method showed an efficient performance in

segmenting characters for all coins. The final output for our method was a single character in binary and grayscale formats. Figure 3-4 shows some examples of using the proposed method to segment the characters of Canadian and Chinese coins. Each character is segmented from the binary and grayscale images of the coins. Furthermore, the use of connected components (blobs) for character segmentation is efficient and suitable for all types of coins regardless of the language used on the coin.

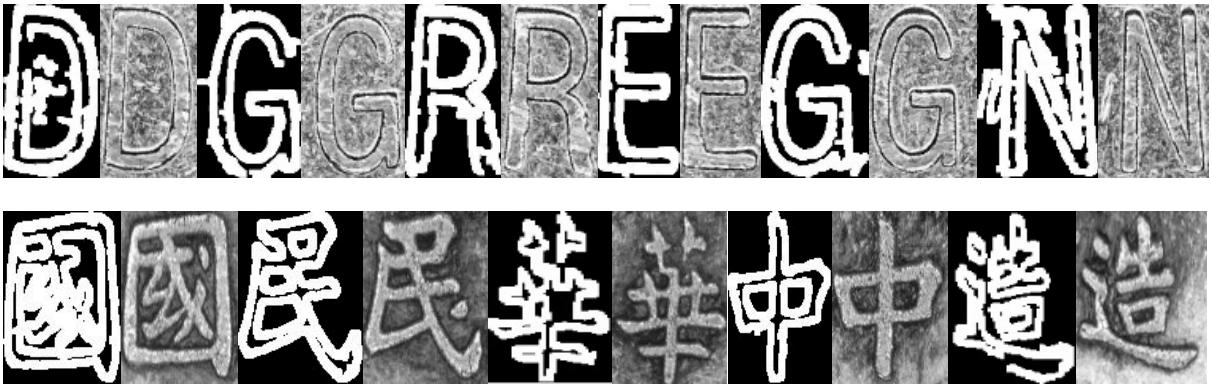


FIGURE 3-4 EXAMPLES OF SEGMENTED CHARACTERS OF CANADIAN AND CHINESE COINS USING OUR PROPOSED METHOD.

3.4.1 DATASETS

We evaluated the performance of our method on 6 datasets from 4 different origins: Canadian, Danish, Chinese, and US. The Canadian and Chinese coin images were collected and scanned at the Center of Pattern Recognition and Machine Intelligence (CENPARMI). A Canon T2i camera was used for image acquisition. It is an 18 megapixels camera with CMOS (APS-C) sensor and the lens used is EF-S 60 mm f/2.8 Macro USM. The lighting source is 2 * 135 watts, equal to (500 watts EQV each) 1000 watts and 5500K professional fluorescent light bulbs. The collected images are around 2953x2768 pixels in size and have 24-bit depth. The Canadian coin dataset consists of 2 versions on the C$2, before and after 2002. This dataset contains a total of

100 coins each with 2 sides (obverse and reverse sides) for a total of 200 images. For the Chinese coin dataset that contains 4 images, we increased the dataset by rotating the coins and adding some noise to obtain 28 images.

The Danish coin dataset was provided by the Law enforcement office and Danish authorities, containing fake and genuine coins. However, this chapter focuses on character segmentation only rather than counterfeit detection. Therefore, we used 40 images of genuine Danish coins to carry out the experimental work for the method proposed in this chapter. The original images of Danish coins were scanned in a specialized 3D HD scanner at one of the leading forensics companies. In this research, we used the 2D images which were also obtained from the same scanner. Finally, the publicly available US coin dataset were obtained from the Professional Coin Grading Service (PCGS) website that contains several US coin images. In this research, we obtained 50 images of $1 coins and 50 images of half dollar coins. A total of 348 images from 4 datasets of 6 different coins were used to evaluate the proposed method.

3.4.2 EVALUATION CRITERIA

We used the *precision*, *recall*, and *f-measure* metrics to evaluate the results of our method. We first obtained the number of ground truth characters in each coin as shown in Table 3-I. Then, we defined the *True Positive (TP)* as the number of correctly segmented characters for each image. *False positive (FP)* is the number of segmented characters that are not characters. The *False Negative (FN)* is the number of characters that are not segmented from the image. β is set equal to 1 to equalize the importance of *recall* and the *precision* when finding the *f-measure*. The *precision*, *recall*, and *f-measure* are calculated from equations (3.5), (3.6), and (3.7) as listed below:

$$precision = \frac{TP}{TP+FP} \tag{3.5}$$

$$recall = \frac{TP}{TP+FN} \tag{3.6}$$

$$f_{measure} = \frac{(1+\beta) \times precision \times recall}{\beta^2 \times (precision+recall)} \tag{3.7}$$

Additionally, the performance of the character segmentation method is evaluated by the *segmentation error rate*. The segmentation result is compared to the ground truth.

TABLE 3-I NUMBER OF CHARACTERS FOR THE 6 DIFFERENT COINS

1	Canadian (Old version)	Obverse	23
		Reverse	14
2	Canadian (New version)	Obverse	23
		Reverse	14
3	Danish	Obverse	27
		Reverse	-
4	Chinese	Obverse	10
		Reverse	9
5	US (One Dollar Coins)	Obverse	7
		Reverse	30
6	US (Half Dollar Coins)	Obverse	11
		Reverse	31

Assume C_i is the number of mismatched segments and T_j is the total number of ground truth segments. Then, the *segmentation error rate* is computed from equation (3.8) as follows:

$$Err_{seg} = \frac{C_i \times 100}{T_j} \tag{3.8}$$

If a ground truth segment is not included in the resulting segments, the ground truth segment is considered to be a mismatched segment. Thus, the segmentation error rate is a complement to the *recall*.

3.4.3 RESULTS AND DISCUSSION

We evaluated the segmentation results based on 6626 ground truth segments belonging to 348 images of 6 different coins. The results of *recall*, *precision*, and *f-measure* are represented in Table 3-II. We observe that the highest *f-value* is obtained from the segmentation of Danish coins due to the high-quality images obtained by the specialized scanner. The images of Danish coins are very clear and all the edges on the image are sharp. The lowest *f-value* resulted from the Chinese character segments. The main reasons are: (1) The Chinese coins that we worked on were over 100 years old and the quality was lower than that of other coin datasets. (2) Some of the Chinese characters are separated (detached) by nature, as shown in Figure 3-2(c), which makes it harder to segment them as a single character.

TABLE 3-II *RECALL*, *PRECISION*, AND *F-MEASURE* RATES OF CHARACTER SEGMENTATION

		Recall	***Precision***	***f-measure***
Canadian (Old version)	Obverse	0.93	0.896	0.913
	Reverse	0.905	0.876	0.89
Canadian (New version)	Obverse	0.935	0.904	0.919
	Reverse	0.91	0.874	0.892
Danish	Obverse	0.953	0.935	0.944
	Reverse	-	-	-
Chinese	Obverse	0.886	0.849	0.867
	Reverse	0.873	0.88	0.876
US (One Dollar Coins)	Obverse	0.921	0.866	0.893
	Reverse	0.948	0.89	0.918
US (Half Dollar Coins)	Obverse	0.914	0.878	0.895
	Reverse	0.929	0.894	0.911

The overall *f-measure* values were between 0.867 and 0.944, which reflects the effectiveness of the proposed character segmentation method. The segmented characters are of different sizes based on the input straight coin image size and the character height and width. The *segmentation error rate* was also reported to determine the rate of unsegmented characters from the coin image.

Figure 3-5 depicts the *segmentation error rate,* which is complement to *recall* rate. The *x-axis* represents the 6 different datasets used in our experiments in Table 3-I, while the *y-axis* depicts the *segmentation error rate* percentage. The highest segmentation error rate also was reported for the Chinese coins dataset for the reasons stated earlier. The average *segmentation error rate* is 8%, where the highest is 12.6% and lowest is 4.6%. We observed that the lowest error rate was also reported for the Danish coins dataset with the highest quality images.

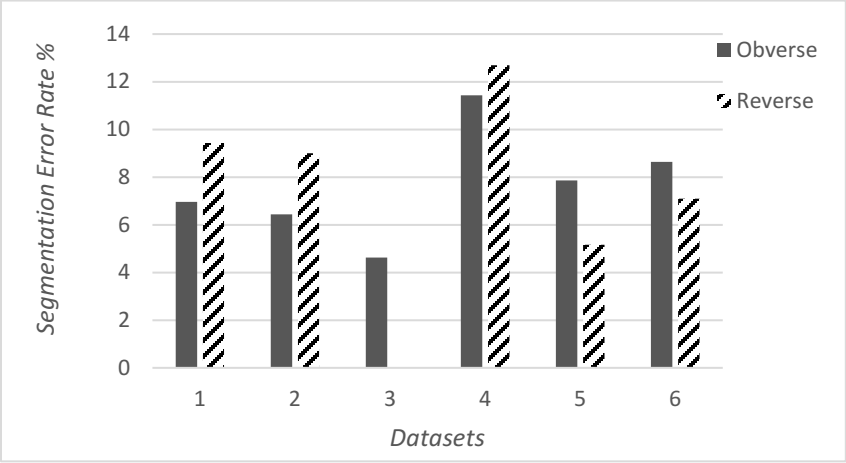


FIGURE 3-5 SEGMENTATION ERROR RATE FOR THE 6 DATASETS.

The main observations drawn by the experimental results are: (1) the use of dynamic mask improves the segmentation performance for different coins with various character sizes, (2) the coin image quality has the highest effect on segmentation results, (3) the greater the number of characters on the coin, the lower the segmentation error (since the character height and width ranges would increase and it could increase *false positive* segments), (4) the language used in coins has very little effect on the segmentation accuracy, while languages that have detached characters tend to have lower segmentation accuracy, and (5) the thresholds selected in this research and described in Section 3.4 are the optimal selection of multiple thresholds studied in this research. Changing the thresholds would considerably reduce the segmentation accuracy. The segmentation

results would remarkably decrease when applied to highly degraded and worn out coins as shown in Figure 3-6(a). In addition, the proposed method can be extended in the future to include the characters attached to the outer edge or other stamp symbols as shown in Figure 3-6(b)

(A) TWO MORGAN US DOLLAR SERIES COLLECTED FROM THE PCGS WEBSITE WITH SEVERE DEGRADATION TO THEIR CHARACTERS APPEARANCE

(B) KENNEDY AND WALKING LIBERTY US HALF DOLLAR SERIES HAVING SOME CHARACTERS ATTACHED TO THE MOTIF SYMBOL

FIGURE 3-6 EXAMPLES OF THE LIMITATION OF THE PROPOSED CHARACTER SEGMENTATION METHOD.

3.5 SUMMARY

In this chapter, we studied the problem of character segmentation from coin images. Character segmentation is an essential step for various systems that work on coins, such as coin recognition, grading, and authentication. The challenges of this work are the character orientation, the degree of degradation in coin quality, and character appearance due to highlight and shadow

variations caused by different lighting sources. We proposed a reliable solution for segmenting single characters and returning a binary and grayscale images of each character separately as shown in Figure 3-4. The proposed solution was experimentally proven to handle different languages and extract characters accurately. Experimental results suggest that the image quality of the coin can have the highest impact on the segmentation results, while the number of characters on the coin has the second highest impact on the segmentation accuracy.

CHAPTER 4

COIN RECOGNITION

This chapter presents a novel and robust method for structured object recognition based on improvements to the traditional Histogram of Oriented Gradients (HOG) method [10]. HOG is a widely-used object recognition method that has shown impressive results when applied to different objects tasks such as face and pedestrian recognition. The main drawbacks of HOG are the necessity of finding an optimal window size to fit the whole object and the exhaustive search mechanism that uses a fixed size window to slide through the whole image to locate and recognize objects. In this chapter, we propose a *dynamic-Histogram of Oriented Gradients* (*dynamic-HOG)* method that works on locating and recognizing structured objects in images. The dynamicity of the method refers to the dynamic window size w.r.t. the object size. Additionally, this method eliminates the exhaustive search by locating the objects first and then recognizing them, thus, saving a great deal of time.

The method works on structured objects due to its dependency on finding the right size of objects and placing the dynamic window. This research considers characters that are minted on coins of different languages and sizes as target objects to be recognized. Several papers have discussed the coin recognition problem in the literature and proposed solutions based on various sets of features extracted from the entire coin surface. Therefore, we have applied the *dynamic-HOG* to coin recognition. The aim is to recognize the coin based on characters minted on its surface. The character orientation is challenging due to the circular shape of coins. Therefore, we transform the coin from its circular shape into rectangular shape to facilitate character

segmentation. A set of histograms of oriented gradients is extracted for every character, based on its height and width. Also, a dimensionality reduction method is used to reduce the size of the feature vector and a multiclass SVM is used to classify the characters.

4.1 COIN RECOGNITION USING *DYNAMIC-HOG*

The proposed method is crafted for character recognition and, specifically, for characters that are minted on coins. The method captures the human vision system in recognizing coins, where humans tend to read text and numbers to identify the origin and value of any coin. The main challenges reported in most of coin recognition research works are the scale and the rotation of the coin. The focus of those papers is to develop a scale and rotation invariant coin recognition system. This research proposes a new dynamic window selection to locate characters and extract their features. The framework of the proposed system is shown in

Figure 4-1. The proposed solution is scale and rotation invariant and it works on extracting the histogram of oriented gradients from only a dynamic window using the *Dynamic-HOG* rather than the sliding window throughout the entire image.

The method starts by applying the coin scaling and straightening algorithms described in Section 3.1 and Section 3.2. Straightening the coin is an essential step for our system to overcome the rotational problem and to obtain similar gradient features for identical characters on different coins. The classifier is used to train and evaluate the *Dynamic-HOG* feature descriptor in recognizing and classifying characters. Finally, the characters are sorted alphabetically to identify the coin and classify all coins by their origins. The proposed method can greatly reduce the computational cost of coin recognition by considering small parts of the coin, specifically

characters, and also by reducing the set of features for each character using the dimensionality reduction method.

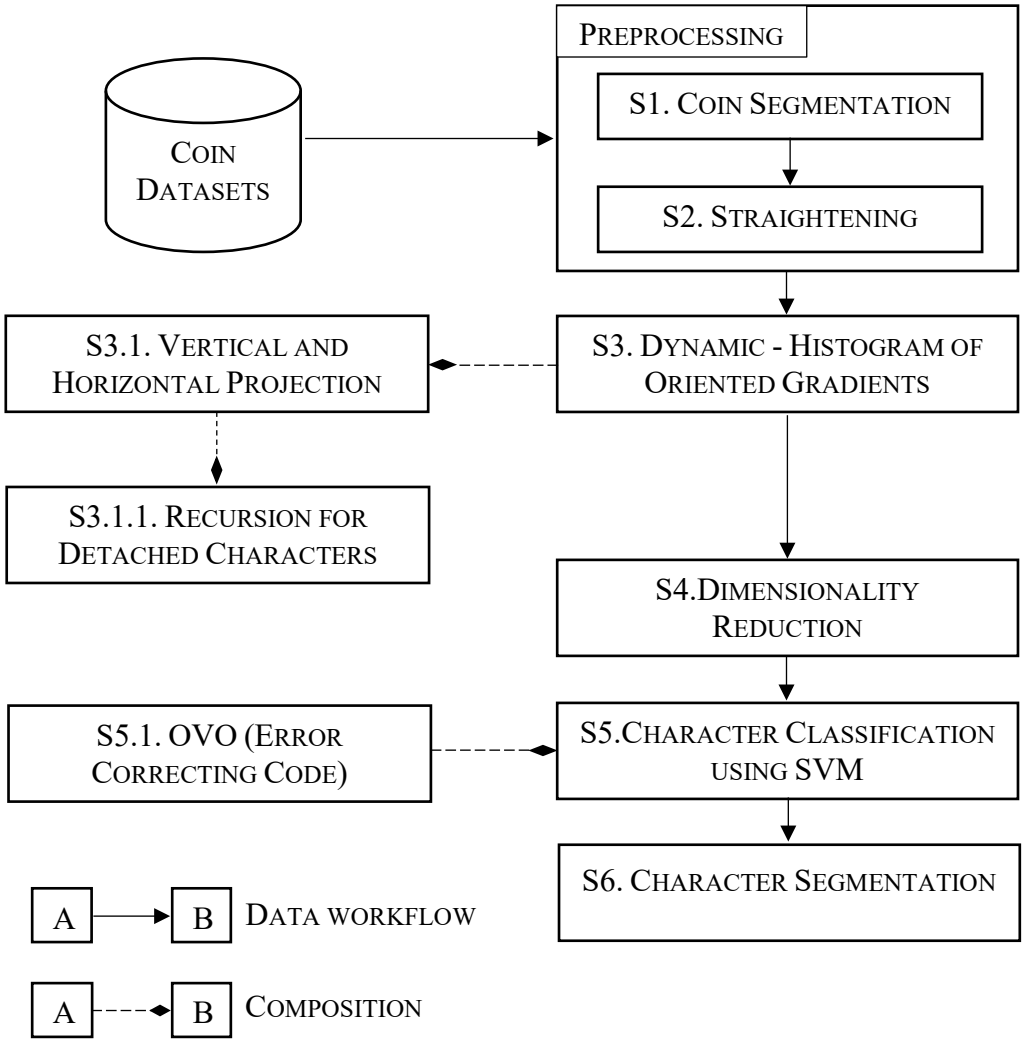


FIGURE 4-1 COIN RECOGNITION SYSTEM FRAMEWORK.

4.2 HISTOGRAM OF ORIENTED GRADIENTS

In this section, we summarize the original histogram of oriented gradients method so that the research would be self-contained. The Histogram of Oriented Gradients (HOGs) [10] is a rotation invariant feature descriptor that counts the edge orientations in all pixels of a local

neighborhood. The algorithm accumulates local 1-D histograms of gradient directions from each small set of neighborhood pixels in each cell. The algorithm then combines the histograms of neighbor cells into larger spatial regions called blocks, and accumulates the measure of local histogram. The measure of local histogram in each block is normalized to overcome any illumination, position, and shape changes. The HOGs feature vector (descriptor) is formed by combining the normalized histograms of all blocks of the image. Generally, the algorithm is based on three main stages: (1) *calculation of gradient, (2) cell orientation histograms*, and (3) *normalization*. Figure 2-1 shows an illustration of the histogram of oriented gradient process.

We used the recommended 1-D mask size of [-1 0 1] by Dalal and Triggs [10] to compute the gradient of image *I(x, y)* given in equation (4.1):

$$I_x(x, y) = I(x, y + 1) - I(x, y - 1), \text{ and}$$
$$I_y(x, y) = I(x - 1, y) - I(x + 1, y) \tag{4.1}$$

The gradient is then used to compute the magnitude μ and orientation θ in formula (7). By ignoring the direction, the gradient is transformed into polar coordinates with an angle between 0^o and 180^o.

$$\mu = \sqrt{I_x(x,y)^2 + I_y(x,y)^2}, \text{ and}$$
$$\theta = \frac{180}{\pi} \left(\tan^{-1}{}_2 \left(I_x(x,y), I_y(x,y)\right) mod\ \pi\right) \tag{4.2}$$

The cell orientation histograms are computed by dividing the window into *N* adjacent and non-overlapping cells each of *(p * p)* pixels. Then the histogram of oriented gradients is computed in each cell by considering a *B* number of orientation bins for each gradient orientation. In this research, we also considered the number of bins *(B = 9)* recommended by [10]. Despite the fact

that increasing *B* improves the performance, setting *B* to greater than 9 has an unnoticeable impact in representing the gradient information and increases the feature vector size [10]. However, with fewer orientation bins, a voting system should be considered to assign each pixel to one orientation. A pixel with an orientation close to one bin may contribute to a different bin. Therefore, the bins are numbered from 0 to $B - 1$ and each has a width of $w = \frac{180}{B}$. The i[th] bin has boundaries $[w_i, w_{(i+1)})$ and center $c_i = w_{(i+\frac{1}{2})}$ and each pixel with magnitude μ and orientation θ votes to each bin as given in equation (4.3):

$$v_i = \mu \frac{c_{(i+1)} - \theta}{w} \text{ to bin number } i = \left\lfloor \frac{\theta}{w} - \frac{1}{2} \right\rfloor$$

$$v_{i+1} = \mu \frac{\theta - c_i}{w} \text{ to bin number } (i+1) \bmod B \tag{4.3}$$

Block normalization is then the last step in the HOGs algorithm to reduce the illumination effect on the image and to normalize the gradient strengths. The histograms of oriented gradients of the blocks are normalized, given that each two neighbor blocks overlap by *C* cells. Therefore, each cell contributes to more than one block. Figure 4-3 (a) and (b) represent an example of HOGs of two Latin and non-Latin characters, respectively. The block feature ƒ contains concatenated histograms of *C* cells in each block and the normalized 𝕍 is computed by Euclidean norm as given in (4.4):

$$\mathfrak{f} = \frac{\mathfrak{f}}{\sqrt{\|\mathfrak{f}\|^2 + \epsilon}} \tag{4.4}$$

where ϵ is a small positive integer. The need of a normalization step is due to: (1) Reducing the effect of contrast changes between the same character in different images by normalizing the cell histograms, and (2) maintaining the relative magnitudes of gradient in each block where the

gradient magnitude of a cell is considered in η number of blocks due to overlap. Thus, each gradient magnitude is considered η times and has η different normalizations.

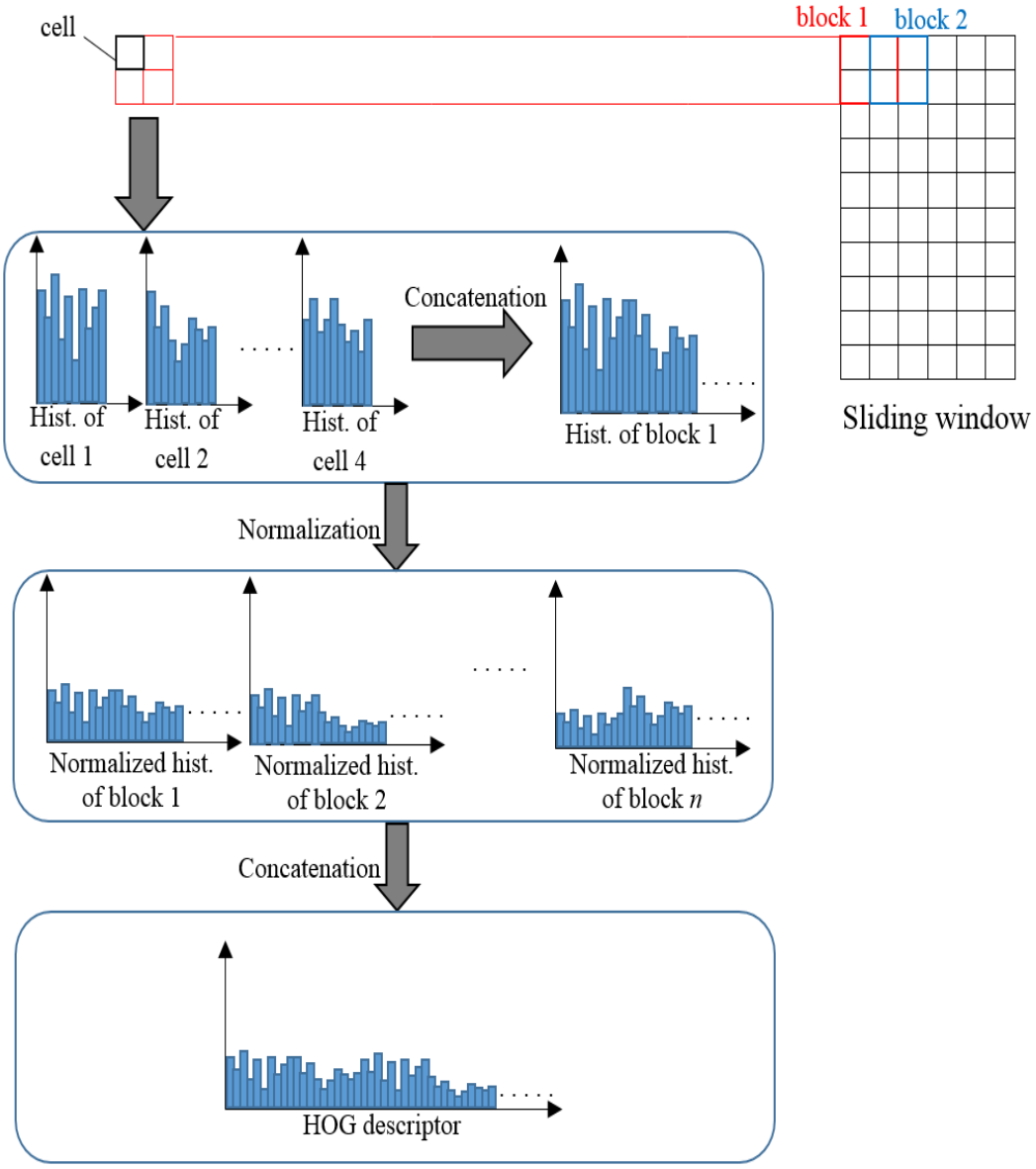
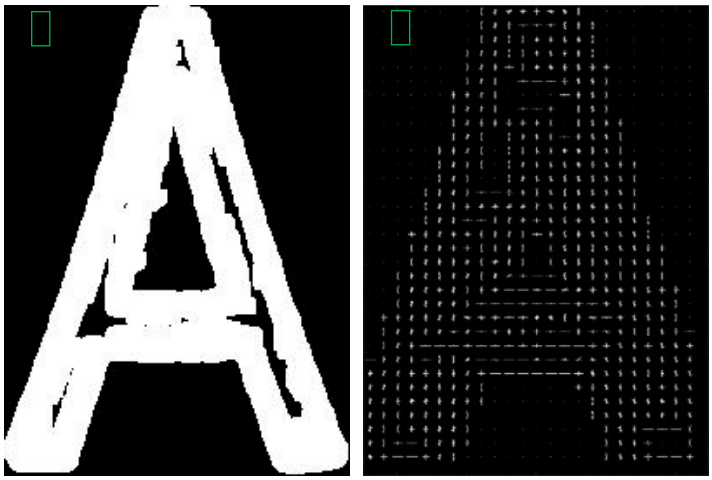
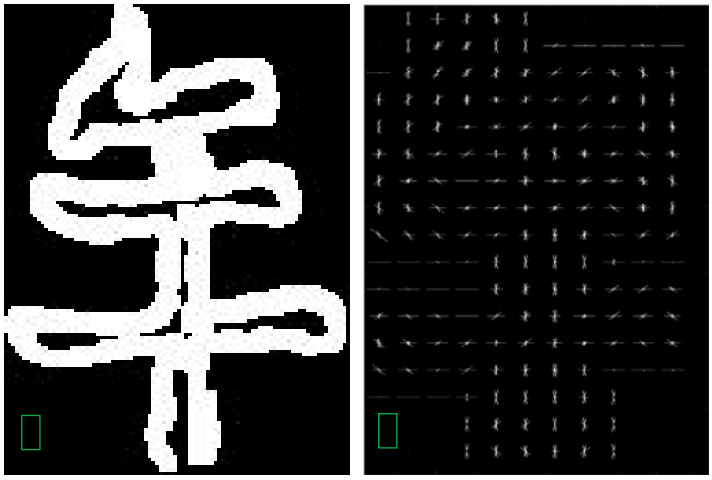


FIGURE 4-2 HISTOGRAM OF ORIENTED GRADIENTS (HOGS) FEATURE EXTRACTION PROCESS USING 3*3 BLOCK SIZE.

The final HOGs feature vector is derived from concatenating the normalized block feature $\mathbb{V}$. Figure 4-2 illustrates the cell and block representation and the normalization process in the histogram of oriented gradients algorithm.



(A) CANADIAN $2 COIN'S CHARACTER OF SIZE (208*104) AND ITS HOG REPRESENTATION



(B) CHINESE COIN'S CHARACTER OF SIZE (139*94) AND ITS HOG REPRESENTATION

FIGURE 4-3 EXAMPLES OF HOGS DESCRIPTOR USING 8*8 PIXELS IN EACH CELL AND 2*2 BLOCK SIZE.

4.3 DYNAMIC-HOG

The key features of HOG descriptor are the division of the image into cells, overlapping blocks, and normalization step. However, selecting the cell and block sizes play vital roles in determining the feature representation accuracy while finding the best window size to fit the whole object are the main challenges in HOG method [72]. On the other hand, avoiding the exhaustive sliding window originally introduced in HOG method to search for objects in the image is the

focus in this research. Therefore, we introduce the *dynamic-HOG* method for character recognition. The key feature of the proposed method is to have a dynamic window size placed over each character w.r.t. its height and width. Characters vary in size where, for instance, the character 'm' is wider than the character 'i', and 'c' is shorter than 'f'. Déniz *et al*. [72] studied the window size and the placement problem and proposed a method based on different window scales. Yet, their method uses a predefined static window size (scale). We argue that the use of a dynamic window size improves the results of HOG for different characters. The dynamic-HOG can be applied to any structured object as long as the horizontal and vertical projection profiles can be obtained accurately.

4.3.1 SELECTION OF WINDOW SIZE

Window size selection uses the *vertical* and *horizontal projection profiles* to determine the height and width of each character. This step starts by applying a 3 x 3 Gaussian filter to remove noise and the Sobel filter algorithm [71] to obtain a binary image. The vertical projection profile *f(x,p(x))* is applied to determine the characters distribution on the coin and to define the average width $w_{avg.}$ If $v_{a,b}$ denotes a value pixel at coordinates (*a, b*) in image *I(x,y)*, the vertical projection value $(p(b))$ at the b^{th} column of *I(x,y)* can be calculated by (4.5). On the other hand, the horizontal projection profile *g(y,p(y))* is also defined to set the average height h_{avg} of connected components (blobs).

$$p(b) = \sum_{a=0}^{x-1} v_{ab} \tag{4.5}$$

After defining the w_{avg} and h_{avg} of all connected components, the spaces between characters are determined by calculating the slopes between each peak and the next valley. Slopes are defined

on the vertical projection profile to determine the spaces between characters by the following formula (4.6):

$$Slope_{vertical} = \frac{\mathrm{p}(x_{peak}) - \mathrm{p}(x_{valley})}{x_{peak} - x_{valley}} \quad (4.6)$$

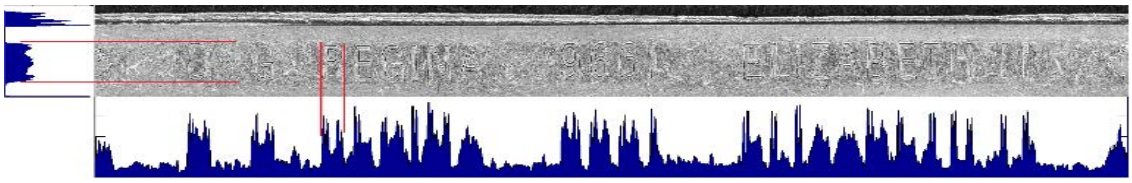
where $p(x)$ is the total number of white pixels in column x, and x_{peak} and x_{valley} are the coordinates of columns where the peak and valley are located.

Figure 4-4(a) shows the vertical and horizontal projection profiles of a Canadian coin. In addition to h_{avg} and w_{avg}, the maximum height h_{max} and maximum width w_{max} of connected components are defined, see Figure 4-4(b). The algorithm then checks for all connected components CC with height h_{CC} and width w_{CC} such that:

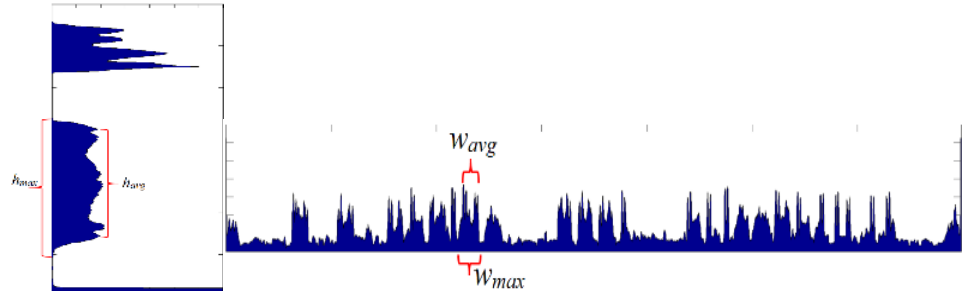
$$h_{avg} + \mathcal{E}_1 > h_{CC} \geq h_{avg} - \mathcal{E}_1$$

$$w_{avg} * \mathcal{E}_2 > w_{CC} \geq w_{avg} / \mathcal{E}_2$$

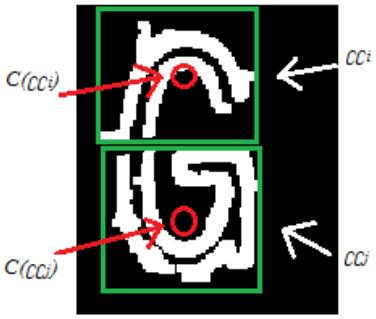
where $\mathcal{E}_1$ and $\mathcal{E}_2$ are constant thresholds. $\mathcal{E}_1$ is the difference between h_{avg} and h_{max}, where h_{max} is the maximum height of all characters obtained from the horizontal projection profile; and $\mathcal{E}_2 = 4$. Note that the use of different arithmetic operations for defining the range of the height and the width is due to the fact that variation in height is much less than the variation in width from one character to another.



(A) HORIZONTAL AND VERTICAL PROJECTION PROFILES OF A CANADIAN COIN



(B) HORIZONTAL AND VERTICAL PROJECTION PROFILES WITH H_{AVG}, H_{MAX}, W_{AVG}, AND W_{MAX} DENOTED ON THEM



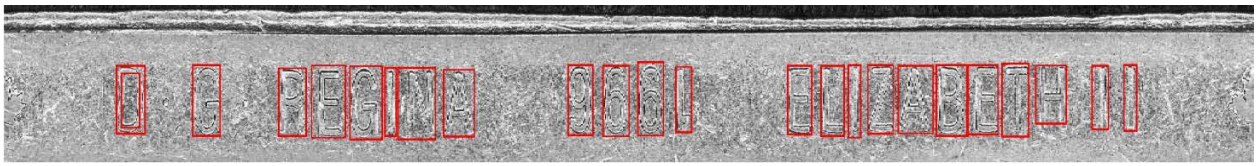
(C) DETACHED CHARACTER G WITH TWO DETECTED CONNECTED COMPONENTS CC_I AND CC_J

FIGURE 4-4 ILLUSTRATION OF THE *DYNAMIC-HOG* RECURSIVE PROCESS.

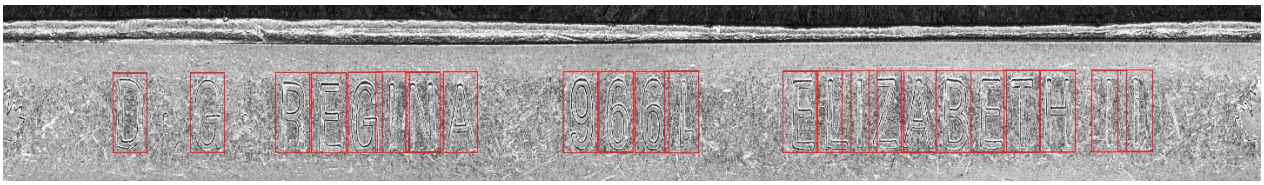
The *Dynamic-HOG* method starts by placing a bounding window around each connected component that has the height and the width within the range and discards the ones that extend outside the range. To improve the character localization, we apply a recursion process to detect all connected components CC that occur below the width and height ranges, but within the maximum height h_{max} of all characters that we obtain from the horizontal projection profile. These CC commonly occur due to degradations on the coin surface which result in broken character strokes. For each $CC_i \subseteq CC$, the method checks the distance d between the center $C(CC_i)$ and the center of the immediate window C(CC_j), where $CC_j \subseteq CC$, as shown in Figure 4-4(c). If d is less than h_{max} / α

for a CC_j that occurs above or below the CC_i or is less than w_{max} / α for a CC_j that occurs left or right the CC_i, where α (equals 2 for our coin datasets obtained experimentally as discussed in Section 4.6) is the threshold of acceptable distance between the CC_i, $CC_j \subseteq CC$, then the two blobs are combined in one window and checked again against the original range of acceptable characters width and height. However, the size of CC_i and CC_j should not be less than h_{avg} / β and w_{avg} / β where β experimentally set to 4 for the height and the width.

The window size is specified from the dynamic adaptive window placed over each character in the previous step. The HOG feature set is obtained from each window and is used for classification. Since the HOG features are extracted from the exact location of characters based on the dynamic bounding window, there will be minimal redundancy and noise features in the HOG vector. Therefore, the dynamic window size improves the recognition rate of each character and reduces the error rate of character segmentation as discussed in Section 4.6. Figure 4-5(a) shows a fixed size window over characters; we argue that even if an exact placement of window is achieved, the window that covers one character may also cover another character or part of it. On the other hand, a dynamic window covers each character based on its size as well as characters that are missing part of their stroke; for instance, the character H (shown in Figure 4-5(b)) has the bottom part of the stroke erased due to circulation effects. The fixed size window approach yields a number of overlapping windows: each window may fit the character strokes exactly or fit the character strokes and the next character strokes when the character width is considerably smaller than the bounding window size. The dynamic window size is very applicable to characters and works for coins with different sizes of characters.



(A) DYNAMIC WINDOW SIZE USING THE HORIZONTAL AND VERTICAL PROJECTION PROFILES



(B) STATIC WINDOW SIZE SELECTED BASED ON THE LARGEST CHARACTER SIZE AND FITTED MANUALLY OVER ALL CHARACTERS

FIGURE 4-5 COMPARISON BETWEEN DYNAMIC AND MANUAL WINDOW PLACEMENT OVER CHARACTERS.

4.4 DIMENSIONALITY REDUCTION

The *Dynamic-HOG* features are calculated based on the orientation histograms of edge intensity (bins) in a local region. Therefore, the feature vector size can be calculated by multiplying the number of bins B in each cell C, by the number of cells C in each block Υ; and finally, by the number of blocks Υ in each window W. The final HOG feature vector size can be calculated by formula (4.7) below:

$$Size_{\mathbb{V}} = \Upsilon_{vertical} * \Upsilon_{horizontal} * C * B \tag{4.7}$$

Given that in Figure 4-3(a), 8*8 pixels were used for each cell and 2*2 cells in each block, it resulted in 25 blocks vertically and 12 blocks horizontally. The total number of HOG features in the feature vector is:

$$Size_{\mathbb{V}} = 25 * 12 * 4 * 9 = 10{,}800$$

The feature vector has a rich descriptor as shown in Figure 4-3. Each block contains the radius and weight of the bins to form a small illustration of the 9 bins, where each bin has an angle between 0 to 180 degrees and a normalized relative weight. However, the 10,800 features is a large number that contains noise and irrelevant information due to the overlapping blocks or non-edge areas, such that in the green rectangles in Figure 4-3 where those areas have no orientation and weight information. In addition, due to the different sizes of the bounding windows of each character, the feature vector size would vary and thus, we add zeroes at the end of each vector to match the length of the longest feature vector. Therefore, the goal of using dimensionality reduction methods is to reduce the feature vector size, increase the learning accuracy, improve result comprehensibility, and avoid overfitting of data.

Several dimensionality reduction methods for machine learning have been presented, such as *Linear Discriminant Analysis (LDA)* [73], *Principal Components Analysis (PCA)* [74], and *Locality Preserving Projections (LPP)* [75]. The dimensionality reduction methods work on projecting the feature vector into a lower dimensional space while maintaining important properties and relationships between data to achieve a higher classification accuracy. The higher the dimensionality, the greater the computational cost and the lower the performance. Therefore, eliminating the redundant and noise features from the feature vector advances the classification accuracy and reduces the time and memory consumptions.

Linear discriminant analysis (LDA) works to maximize the inter-class similarity and intra-class dissimilarity. LDA is a supervised dimensionality reduction technique that performs a linear transformation of the original data in the D-dimensional space into a subspace of d dimensions where $d < D$. Let $F=\{f_1, f_2, f_3, \ldots, f_n\}$ be a matrix of data points belonging to K classes. LDA finds

a linear mapping *L* of matrix *F* that maximizes the *Fisher criterion.* Assume z is the transformation vector such that $z^T f_i = y_i$, the objective function of LDA is computed as given in equation (4.8):

$$z = arg \max_z \frac{z^T S_b z}{z^T S_w z} \tag{4.8}$$

where $S_w = \sum_{i=1}^{K} (\sum_{j=1}^{n_i} (f_{i_j} - m_i)(f_{i_j} - m_i)^T)$ and $S_b = \sum_{i=1}^{K} n_i (m_i - \bar{m})(m_i - \bar{m})^T$ are scatter matrices of the average within a class and between classes, respectively. n_i is the number of data points in class i, and m_i is the mean vector of class i, while f_{i_j} is the j[th] data point of class i, and $\bar{m} = \frac{1}{n} \sum m_i$ is the overall mean.

On the other hand, the principal component analysis (PCA) transforms the feature vector dimensional space into a subspace of the original dimension space. PCA is an unsupervised dimensionality reduction method. The principal components are formed from the eigen decomposition of the covariance matrix $\sum$ of the original data. They represent the linear combination of the original features data. The first small set of principal components account for the most variance of the feature vectors where the first principal component accounts for the greatest variance possible in the data and so on. Let $f_1, f_2, f_3, \ldots, f_n$ be the original data points in the *D* dimensional space. The goal is to project these data into a subspace with *d* dimensions such that $d < D$. Now, the transformation vector z is obtained by the objective function in equation (4.9):

$$z = arg \max_z \sum_{i=1}^{n} (y_i - \bar{y})^2 \tag{4.9}$$

where $y_i = z^T f_i$ and $\bar{y} = \frac{1}{n} \sum c_i$. The PCA considers the eigenvectors of the covariance matrix with highest eigenvalues.

The PCA has the advantage of faster matrix multiplication due to the linear transformations which, in return, requires lower computational cost. Also, the PCA is data-dependent, which relies on the data only to find the maximum variance between them. The PCA has been widely used in computer vision applications [72] and combined with HOG to detect human, eyes, and cars in several research papers.

Locality preserving projection (LPP) was introduced by He and Niyogi [75]. LPP is also an unsupervised dimensionality reduction method that works on projecting the original data points into a maximum variance while preserving the local neighborhood information. LPP adapts a graph-based method to find a linear mapping of nonlinear *Laplacian Eigenmap*. Let $F = (f_1, f_2, f_3, \ldots, f_n)$ be the matrix of n data points in a D dimensional space. The goal is to map $y = (y_1, y_2, y_3, \ldots, y_n)^T$ with d dimensional subspace on the original data points to ensure that if f_1 and f_2 are close then y_1 and y_2 are close. The transformation vector z is computed from the objective function of LPP in equation (4.10):

$$z = arg \min_{z} \sum_{i=1}^{n} \sum_{j=1}^{n} (y_i - y_j)^2 W_{ij} \tag{4.10}$$

where y_i is a 1-dimensional representation of f_1. W_{ij} is the weight of the edge connecting nodes i and j, and has value 0 if there is no edge.

The LDA, PCA, and LPP dimensionality reduction methods are tested on our system to reduce the feature vector size and to compare their accuracy results.

4.5 SUPPORT VECTOR MACHINES

The reduced feature vectors obtained from the dimensionality reduction method is now the input to the *Support Vector Machine* (SVM) classifier. The SVM was introduced by Vapnik [76] and since then, the SVM showed excellent results in classification systems. The SVM was proposed first to be a linear classifier. SVM works on dividing only two classes by a hyperplane placed between them and the goal is to maximize the distance of each class elements from the hyperplane. Later in 1990's, the SVM was redesigned to work on non-linear (multiclass) classifications. However, we used the SVM to classify the characters and the coins in return based on the experimental evaluation performed (in Section 4.6).

The non-linear SVM uses the *one versus all (OVA)* or the *one versus one (OVO)* techniques to support the multiclass prediction. Given C number of classes, the one versus all works on training a classifier for each class where only samples of class $c_i \in C$ are positive and ignores C/c_i where samples of all other classes are negative [77]. The later introduced *OVO* outperforms the *OVA* in performance. Therefore, we use the *OVO* technique in this research to achieve a multiclass SVM on the extracted feature set. The *OVO* trains each classifier, from the *k(k-1)/2* classifiers constructed for *k*-class SVM, with data from the c_i, $c_j \in C$. OVO works on classifying each two classes c_i and c_j at a time and ignores the $C/\bar{c}$ where $\bar{c} = c_i \cup c_j$; the classifier considers c_i as the positive class while c_j is the negative. Then each vector is classified to either c_i or c_j.

A linear kernel function is chosen for the proposed system over the radial basis kernel function to achieve a better accuracy, faster computation, and less overfitting. Moreover, given the *k(k-1)/2* classifiers we obtain at the end of the training, each feature vector is assigned to a class

based on majority votes. The positively classified characters are then segmented from the coin and used to identify the coin's minting country.

Moreover, we use other widely used classifiers such as the *K-nearest neighbors (KNN)*, *decision tree*, and *Naïve Bayes* classifiers to compare the classification results and analyze the classification accuracy rates as discussed in Section 4.6.

4.6 EXPERIMENTAL RESULTS

The goal of our experiments is to evaluate the character recognition from coins using the *Dynamic-HOG* descriptor. Datasets play an important role in evaluating the system performance. Hence, the experiments are conducted on datasets represented in two different forms. The datasets are compiled from 5 different coins belonging to four different countries as discussed in Section 3.4.1. The Canadian, Chinese, Danish, and US coins are used. The Canadian coins dataset consists of 2 versions of the C$2 based on their release dates which are before and after 2002. However, in the proposed solution, we identify both versions of the coin as Canadian coin since both versions bear the same set of characters on their surface. Unlike the two versions of the Canadian coin, the half dollar and the one-dollar US coins have different characters on the surface. Thus, we recognize them separately as two different coins belonging to two different classes. The total of 828 images belonging to 5 coins (Table 4-I) are used in two forms for evaluation: (1) the 828 images used as 9 sets each belongs to either the obverse or the reverse side of coins except for the Danish coins that have obverse side only, and (2) the 828 images separated into 7 sets, each with a mixture of coins to evaluate the coin recognition rate. In the two forms of dataset separation, we used 70% of the images for training and 30% for testing.

The proposed solution works on selecting an appropriate window size for all characters w.r.t. their sizes and extracts a feature vector for each character using the HOG feature descriptor. Then, the dimensionality of all feature vectors are reduced and used as the input to the classifier for training. Finally, the correctly classified characters are segmented from the coin and weighted alphabetically to indicate the coin class to which they belong.

In this experiment, sets of feature vectors are extracted from each dynamic window using the HOG descriptor and are used to recognize characters. Let C_i be the number of correctly recognized characters. T_j is the total number of classified characters. S_d is the set of ground truth characters. We first obtained the number of ground truth characters in each coin as shown in Table 4-I. There is total number of 18,257 ground truth characters belonging to 828 images of 5 different coins. Given the *f-measure* in equation (4.11), the *recall*, *precision*, and *f-measure* values for each dataset are shown in Figure 4-6(a). The *recall*, *precision*, and *f-measure* values obtained for the obverse and reverse sides of each of the 5 different coins (except for the Danish coin having only an obverse side) confirm the effectiveness of the *Dynamic-HOG* descriptor.

$$precision = \frac{C_i}{T_j} \quad , \qquad recall = \frac{C_i}{S_d}$$

$$f - measure = \frac{(1+\beta) * precision * recall}{\beta^2 * (precision + recall)} \tag{4.11}$$

We set β equal to 1 to equalize the importance of *recall* and the *precision* when finding the *f-measure*.

The *f-value* shows a high accuracy rate in recognizing characters. The HOG settings for this experiment are 8*8 pixels in each cell and 2*2 cells in each block based on the experiment

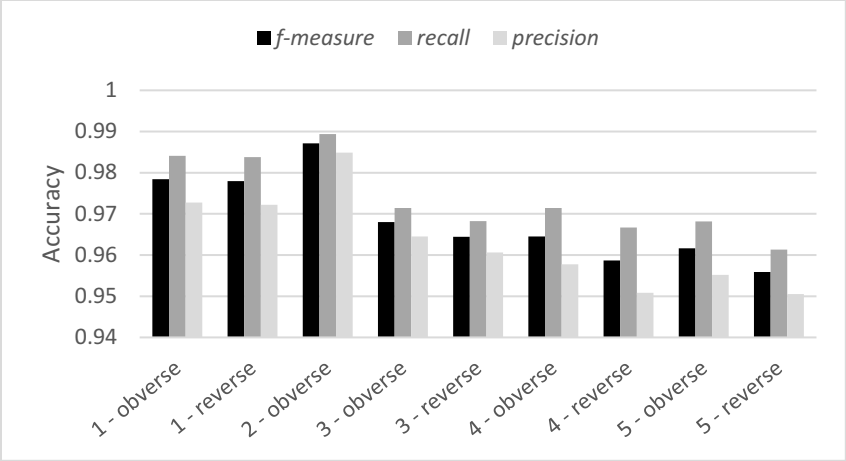
suggestion shown in Figure 4-6(b). The optimal window size is selected for each character as described in Section 4.3.1 while the settings for the recursive process is shown in Figure 4-7. Characters belonging to the Danish coins returned the highest accuracy rate due to (a) clarity of the images with clear strokes representing each character, and (b) the filled stroke of the characters where the stroke being a thick white line on black background. The characters of the Canadian coins returned the second highest accuracy due to minimal noise around and between the character strokes. The lowest rate was reported for the US coins with a 1.87% drop in *f-value* compared to the Danish coins. The quality of US character images was the lowest and contained the highest noise level. In addition, the US coins have the smallest image size of 500*500 which makes it harder to determine the exact size of the bounding window where noise has higher effect on characters. In addition, we investigated the relationship between the parameter settings and the recognition rate.

An interval set of parameters are selected for HOG (Figure 4-6(b)). The goal of this experiment is to evaluate the best parameter setting for an optimal recognition rate. The HOG with different C and Υ parameters (cell size and block size, respectively) are applied to our 5 datasets. However, in this experiment we used the Danish coin dataset to evaluate different settings of HOG since it has the best quality images among the datasets. Figure 4-6(b) illustrates the performance of character classification using different C and Υ settings. Four different C values and three Υ values for the number of pixels in each cell and number of cells in each block settings, respectively, were considered. The experiment shows the lowest *f-measure* accuracy reported for C = 2*2 pixels and Υ = 4*4 cells settings where the *f-value* was 0.966, while the highest *f-value* returned for the C = 8*8 pixels and Υ = 2*2 cells settings, which has a 1.3% increase in recognition accuracy compared to C = 2*2 pixels and Υ = 4*4 cells settings.

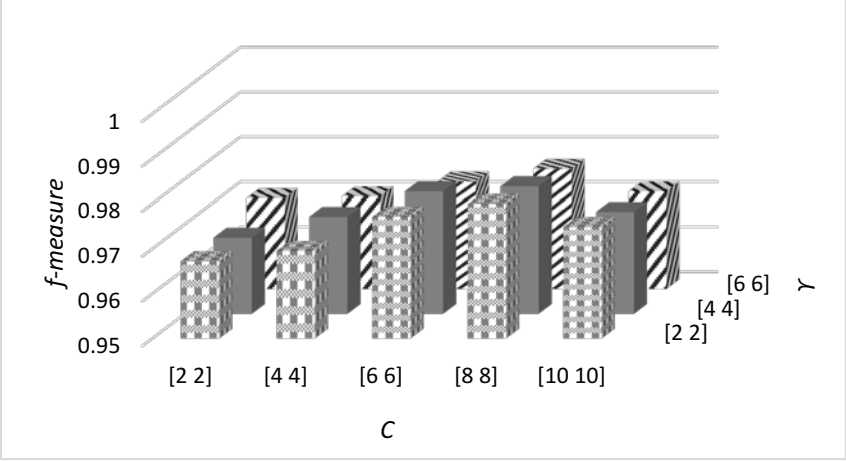
TABLE 4-I NUMBER OF CHARACTERS FOR THE 5 DIFFERENT COINS

1	Canadian	Obverse	200 images	23
		Reverse	200 images	14
2	Danish	Obverse	300 images	27
		Reverse	-	-
3	Chinese	Obverse	14 images	10
		Reverse	14 images	9
4	US ($1 Coins)	Obverse	25 images	7
		Reverse	25 images	30
5	US (50¢ Coins)	Obverse	25 images	11
		Reverse	25 images	31

The returned results confirm the findings of Dalal and Triggs [10], where the larger the block size can cause reduced ability to suppress local illumination and lower gradient information which results in a lower the recognition accuracy. Reducing the number of pixels in each block enables capturing informative local pixels features especially with the normalization of histograms step in the HOG method. On the other hand, a smaller number of pixels in each cell has higher redundant and noise information that would negatively affect the recognition rate. In this research, we selected the optimal settings of C and Υ (C = 8*8 and Υ = 2*2) to carry out the experimental evaluation.



(A) *RECALL*, *PRECISION*, AND *F-MEASURE* VALUES FOR THE OBVERSE AND REVERSE SIDE FOR EACH OF THE 5 COINS IN OUR DATASET



(B) *F-MEASURE* VALUES OF DIFFERENT C AND Υ SETTINGS FOR *DYNAMIC-HOG* DESCRIPTOR

FIGURE 4-6 EXPERIMENTAL RESULTS FOR DIFFERENT *DYNAMIC-HOG* PARAMETERS.

The edges play an important role in image representation, the only drawback is the continuity of edges and the noise. Therefore, this method applies a recursive process to combine two or more contours that belong to a single character. The recursive process relies on the distances d between two immediate contours, and hence, deciding the best distance to represent single character is the challenge. Figure 4-7 depicts the different settings of the d value for the height and width used to capture single character contours. The recursive process with different d_{height} and d_{width} parameters (as discussed in Section 4.3.1) are applied to our 5 datasets. However, in this

experiment we also used the Danish coin dataset to illustrate the evaluation of different settings of the distances between any two immediate contours. The change of *d* for the height has lower impact than the *d* in width. This is mainly due to the empty spaces above and below the characters being larger than to the left and the right of characters. Also, the height range of characters is much smaller than the width range which in return eliminates adding large noise edges to the top or the bottom of characters when we check again in the recursive process.

We also notice that the best results were returned when $d_{width} = 0.5$, which reveals that the separation of character contours commonly occurs in a vertical style. The experimental results revealed the lowest *f-measure* reported for $d_{height} = 1$ and $d_{width} = 1$ settings where the *f-value* was 0.823. The highest *f-measure* returned for the $d_{height} = 0.5$ pixels and $d_{width} = 0.5$ settings with 15.69% increase in recognition accuracy compared to $d_{height} = 1$ and $d_{width} = 1$ settings.

Moreover, the use of small blocks in the proposed *Dynamic-HOG* method significantly increases the feature vector size to better describe the characters. Therefore, the use of a dimensionality reduction method was a necessity to reduce the feature vector size while maintaining valuable features information to classify the characters. The evaluation of different dimensionality reduction methods was carried out to study the method that best fits our datasets and to determine its effectiveness in maintaining important features to discriminate between characters while removing redundant and noise features.

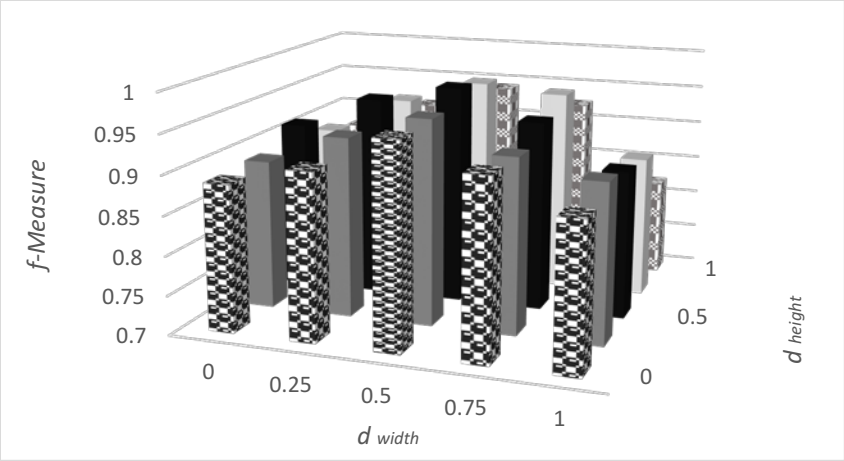


FIGURE 4-7 *F-MEASURE* RATES OF DIFFERENT *D* SETTINGS FOR HEIGHT AND WIDTH FOR THE *RECURSIVE* PROCESS

Experiments were run for the three dimensionality reduction methods, (PCA, LDA, and LPP) to evaluate the performance of each method. Figure 4-8 depicts the *f-measure* values of classifying feature vectors of characters after reducing the dimensionality of those vectors. The results reflect the optimal selection of the number of attributes for each method after several experiments. The results suggest that PCA preserves the highest discrimination information of all methods. PCA shows a significant improvement, not only for the vector size, but also for the classification accuracy compared to the raw data. The second form of our dataset is used in this experiment where the coins are placed into 7 sets. In other words, we created 7 datasets of mixed coins of the 5 different coins in our original dataset. The reason for this setting is to evaluate the performance of our system whose goal is to recognize and discriminate between different coins.

PCA returned a 1% average improvement over the raw data in recognition accuracy. Overall, PCA returned the best classification accuracy when 800 features were considered. LDA reported the second highest accuracy and also slightly outperformed the raw data of 10,800 features with 24 features. The LPP achieved its best accuracy rate using 300 features and returned the lowest accuracy rate among other methods with the second highest running time of 51.4

seconds after the raw data that required 84.2 seconds of processing time. The PCA needed 39 seconds average for feature extraction, training and testing. Finally, the LDA was the fastest, due to the lowest number of features needed, with 29.9 seconds average.

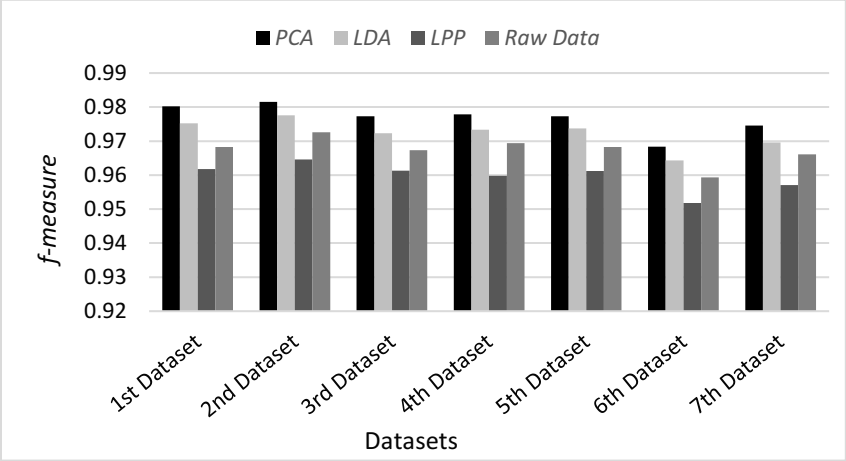


FIGURE 4-8 *F-MEASURE* RATES FOR RAW DATA AND DIMENSIONALITY REDUCTION METHODS.

Furthermore, different classification algorithms are used in the experiments to evaluate the effectiveness and accuracy for each algorithm in classifying our datasets. Table 4-II depicts the *recall (R)*, *precision (P)*, and *f-measure (f)* values of four classification algorithms. We used the well-known classifiers such as SVM, Naïve Bayes, Decision Tree, and KNN to classify the reduced feature vector using PCA. The Naïve Bayes classifier returned the lowest classification accuracy on our datasets and returned the second highest training and testing time of 40 seconds on average.

For example, Table 4-II shows the *f-measure* value of the Naïve Bayes classification rate on the 1st dataset has dropped by 2.14% compared to the SVM classification rate, while other datasets returned almost the same degradation level in accuracy. The KNN is tested with different k number of neighbors and the best classification rate was obtained using $k = 3$ and achieved the second lowest classification accuracy. Overall, KNN has an average of 1.74% drop in *f-value*

compared to SVM. However, the KNN registered the fastest training and testing time among classifiers with an average time of 34 seconds, followed by SVM with 39 seconds and achieved the highest classification accuracy rate. The results of this experiment suggest that the use of SVM would be optimal to classify the PCA feature vectors of segmented characters. Thus, we employed the SVM classifier for the proposed system.

TABLE 4-II *RECALL*, *PRECISION*, AND *F-MEASURE* RATES FOR DIFFERENT CLASSIFIERS

Datasets	**SVM**			**Naïve Bayes**			**Decision Tree**			**KNN**		
	P	*R*	*f*	*P*	*R*	*f*	*P*	*R*	*f*	*P*	*R*	*f*
1st	0.979	0.981	**0.980**	0.956	0.961	0.958	0.967	0.968	0.968	0.961	0.963	0.962
2nd	0.980	0.982	**0.981**	0.958	0.962	0.960	0.969	0.969	0.969	0.963	0.965	0.964
3rd	0.976	0.978	**0.977**	0.953	0.958	0.955	0.964	0.965	0.965	0.958	0.961	0.960
4th	0.976	0.978	**0.977**	0.954	0.958	0.956	0.965	0.965	0.965	0.959	0.962	0.960
5th	0.976	0.978	**0.977**	0.953	0.958	0.955	0.965	0.965	0.965	0.959	0.961	0.960
6th	0.967	0.969	**0.968**	0.944	0.949	0.946	0.955	0.956	0.956	0.949	0.953	0.951
7th	0.973	0.975	**0.974**	0.950	0.955	0.953	0.962	0.962	0.962	0.956	0.959	0.957

Finally, we perform a character segmentation based on correctly classified characters to confirm the results of character recognition and for further processing such as studying the character characteristics (i.e. stroke width) in coin authentication as discussed in [19]. The character segmentation process is performed on correctly classified blobs using the *Dynamic-HOG*. In order to evaluate the character segmentation, the number of characters in each coin is identified as shown in Table 4-I. Let C_i be the number of correctly segmented characters. T_j is the total number of extracted segments (characters and non-characters). S_d is the number of ground

truth characters. We find the *recall*, *precision*, and *f-measure* values as given in equation (14). Additionally, the performance of the character segmentation step is evaluated by the *segmentation error rate*. The segmentation result is compared to the ground truth. Let W_i be the number of mismatched segments. Then the *segmentation error rate* is computed by equation (4.12) as follows:

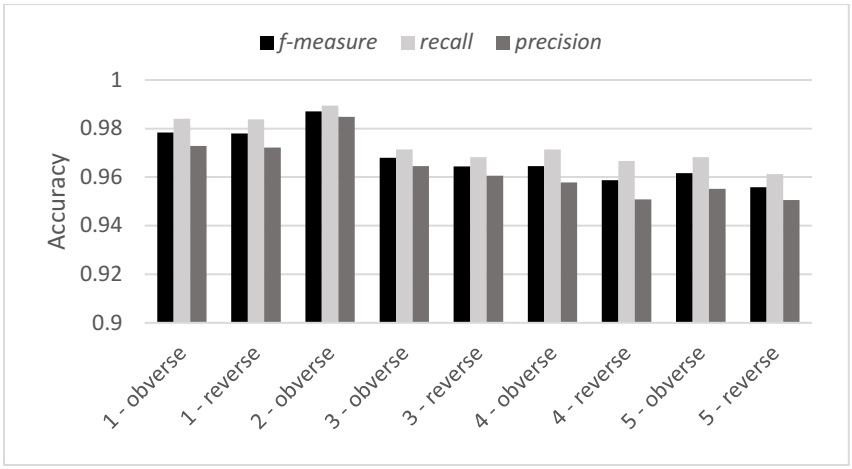
$$Err_{seg} = \frac{W_i}{S_d} \tag{4.12}$$

where W_i is the number of ground truth segments that were not returned by the character segmentation method. Thus, the segmentation error rate complements the *recall* measure.

The results of *recall*, *precision*, and *f-measure* are represented in Figure 4-9(a). The *x-axis* represents the 5 different coins images each with both sides of each coin (except for the Danish coins) used in our experiments. The *y-axis* depicts the accuracy rate of the evaluation methods. We observe that the highest *f-value* and lowest *segmentation error rate* obtained are for character segmentation of Danish coins due to the high-quality images obtained by the specialized scanner. The overall *f-measure* values were between 0.95 and 0.98 based on positively recognized blobs. The evaluation results reflect the effectiveness of the proposed character segmentation.

Moreover, the *segmentation error rate* was also reported as shown in Figure 4-9(b). The highest segmentation error rate was reported for the US coins dataset due to the fact that: (1) The US coins dataset contains more than one series of coins where the character shape has a slight change. (2) The US coins dataset has the smallest image size. Therefore, scratches and noise were present on the US coin images more than images of other datasets. (3) The US coins dataset has a

small set of images; thus, the segmentation method is more vulnerable to a higher number of true negative segments.



(A) *RECALL*, *PRECISION*, AND *F-MEASURE* VALUES FOR CHARACTER SEGMENTATION OF THE 5 DIFFERENT COINS DATASETS

(B) *SEGMENTATION ERROR RATE* FOR THE PROPOSED CHARACTER SEGMENTATION METHOD

FIGURE 4-9 *EXPERIMENTAL RESULTS* FOR THE PROPOSED CHARACTER SEGMENTATION.

Generally, the average *segmentation error rate* is 0.026, where the highest is 0.0387 and lowest is 0.010. In addition, we noticed that the more characters on the coin, the lower the segmentation error. This is because the character height and width ranges that we determine for window size selection (in Section 4.3.1) would increase and this increases the *false positive* segmentation. On the other hand, the coin recognition rate is more vulnerable when there are very

few characters on the coin. The proposed solution shows an efficient character segmentation approach that the segmented characters can be used in other coin applications such as coin authentication or grading.

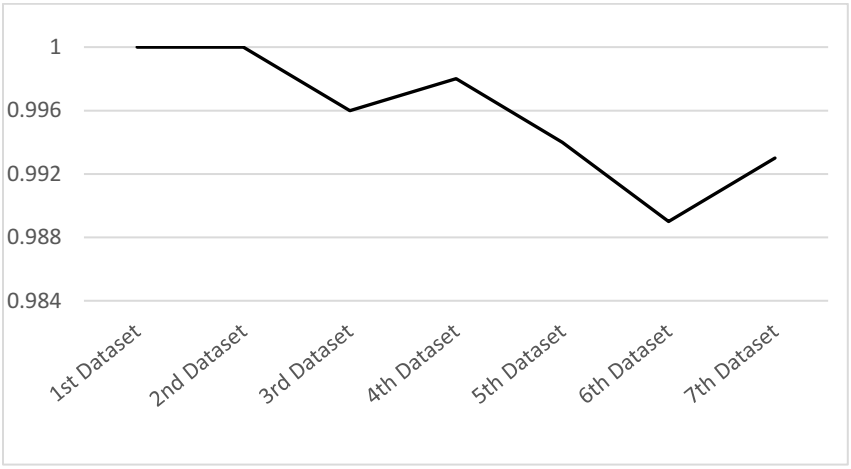


FIGURE 4-10 *NORMALIZED* NUMBER OF COIN RECOGNITION RATE.

Furthermore, the coin recognition has been assessed using *f-measure* based on characters recognized by *dynamic-HOG* method. The aim is to measure the accuracy rate of recognizing coins using features from subparts of the image to reduce the computational cost. After recognizing characters from the coin image using the proposed method, the coin is recognized based on those characters. Each coin has its own text that refers to the minting country, which is the part that is used by human vision system to recognize the coin. Therefore, those characters can reliably reveal the coin's origin after recognizing them. Figure 4-10 depicts the *normalized* number of correctly recognized coins over the 7 datasets of mixed coin types to evaluate the performance. The results demonstrate the effectiveness of the proposed method and suggest that the proposed method outperforms the state-of-the-art methods. The recognition rate was at 100% for the first and second datasets of mixed coin types while the lowest was for the sixth dataset at 98.9% accuracy. The

overall performance of the proposed method from all coins is 99.5% which confirms the reliability of using subparts of the coin to recognize it rather than the whole coin image.

Nonetheless, numerous research works have been conducted on coin recognition, very few of which introduced coin recognition based on the characters minted on the coin. Therefore, we compare the results of the proposed solution with the results of other related work that studied the character recognition problem on coins. Although these studies tested different coin types, the method reported in this work shows a better performance than these presented in the literature.

TABLE 4-III PREVIOUS CHARACTER-BASED COIN RECOGNITION METHODS COMPARED WITH OUR PROPOSED METHOD

Research Paper	**Feature extraction Method**	**Dataset (Coins)**		**Character Segmentation**	**Recognition Rate**
		Domain	Size		
[15]	Word recognition using SIFT descriptor	Ancient Roman coins	180	NA	53%
[9]	Character recognition using SIFT descriptor	Ancient coins	990 Char.	Manually specified ROI	75.6%
[50]	Word recognition using SIFT descriptor and image matching	Ancient Roman coins	464	NA	40% (word recognition only)
[51]	Binary gradient map and connected component properties	US coins	375	Manually specified ROI	44% real 92% synthetic
[62]	HOG-like descriptor for characters and image matching	Roman Imperial denarii	25	HOG-like descriptor	NA
Proposed Method	Dynamic-HOG descriptor	Canadian, Danish, Chinese, US	828	Dynamic-HOG	98.15%

Table 4-III shows the performance of our method as well as these of other methods. The six directly related research works have shown promising results to the field. Four of these research works focused on ancient coins and two used modern coin datasets. In addition, four methods proposed character segmentation in which two of them specified the region of interest (ROI) of character locations manually, while [62] used HOG-like descriptors to locate and extract characters automatically. However, [62] has not reported any results regarding the character recognition rate. The first three methods in the table used the SIFT descriptors to represent features of characters. The reported results suggest a better performance for SIFT features when applied to the exact ROIs that contain characters as shown in [9] while a lower recognition rate resulted for approximate candidate locations of characters [15, 50].

Pan and Tougne [51] worked on identifying the numbers on the coin to recognize the minting year of the coin. The authors used a binary gradient map of the segmented characters to study the blob areas and its properties. The utilization of such an approach is not reliable for circulated coins as shown in their accuracy results of 44% on real coins and 92% for synthetic coins. On the other hand, the *Dynamic-HOG* descriptor used in the proposed solution returned the highest recognition accuracy of up to 98.15%. Bounding each character (w.r.t. its size) with a dynamic window and extract the HOG features from this dynamic window shows a significant improvement over methods in related work. It also reduced the number of *false positive* characters by specifying the range of acceptable height and width for characters.

Finally, the recognition time has not been reported in other methods, while in our method, it required 39 seconds for every dataset. The running time for all of our experiments includes

reading the image dataset, window size selection, HOG feature extraction, dimensionality reduction, and classification.

4.7 SUMMARY

In this chapter, we studied the problem of coin recognition based on recognizing the characters minted on both sides of the surface. The character segmentation is an essential step for various systems that work on coins such as coin recognition, grading, and authentication. Therefore, we proposed a *Dynamic-HOG* descriptor for character recognition and improved segmentation. The proposed solution was evaluated on various coin types, including a publicly available dataset. The challenges of this work are the character orientation, character sizes, heavily degraded coin quality, and character clarity problems due to highlight and shadow variations caused by different lighting sources. The proposed method works on recognizing the characters after placing a bounding window over each character w.r.t. its size from the coin image to recognize the coin in return. The proposed solution was experimentally proven to be capable of handling different languages and extracting characters accurately. Experimental results suggest the image quality of the coin can have the highest impact on the segmentation and recognition results while the number of characters on the coin also has an impact on the recognition and segmentation accuracy. The *Dynamic-HOG* descriptor shows robust feature extraction for characters while PCA works well on reducing the feature vector size and increasing the classification rate.

CHAPTER 5

ENSEMBLE METHOD FOR COUNTERFEIT COIN DETECTION

This chapter discusses a robust counterfeit coin detection method to tackle the advancements and sophistications of modern forgery methods. The proposed method in this research uses an ensemble method of three classifiers in which the first two are based on transfer learning by fine-tuning a pre-trained convolutional neural network (CNN), and the third classifier is trained on features and measures of characters on the coin surface. CNN is a well-established image classification technique that extracts and trains several image features for a given classification task. CNN requires a very large training set, even with transfer learning technique, in order to achieve a high accuracy. In counterfeit coins research, such a large number of counterfeit coins is not available due to security reasons. Therefore, the proposed method uses a fine-tuned pre-trained CNN to classify and extract features to train another classifier while it combines other features obtained from characters that are minted on the surface to achieve a higher precision.

The fine-tuning process customizes the general image features obtained from the original images used for training the CNN (i.e. natural image), into accustomed features that are suitable for coins. In addition, characters represent one of the two major parts of the coin stamp and are used mainly by human vision system to recognize and authenticate coins. The ensemble method combines the classification results of two classifiers trained using features from convolutional

layers and a third classifier trained on character features e.g. distances between characters, stroke width, height and width of characters. The ensemble method achieved promising accuracy rates, demonstrating the reliability of combining fine-tuned CNN classification results and other selected features from characters in coin authentication. The method is evaluated on a real-life dataset of Danish coins as part of our collaborative work with a digital forensic firm and the Danish authorities.

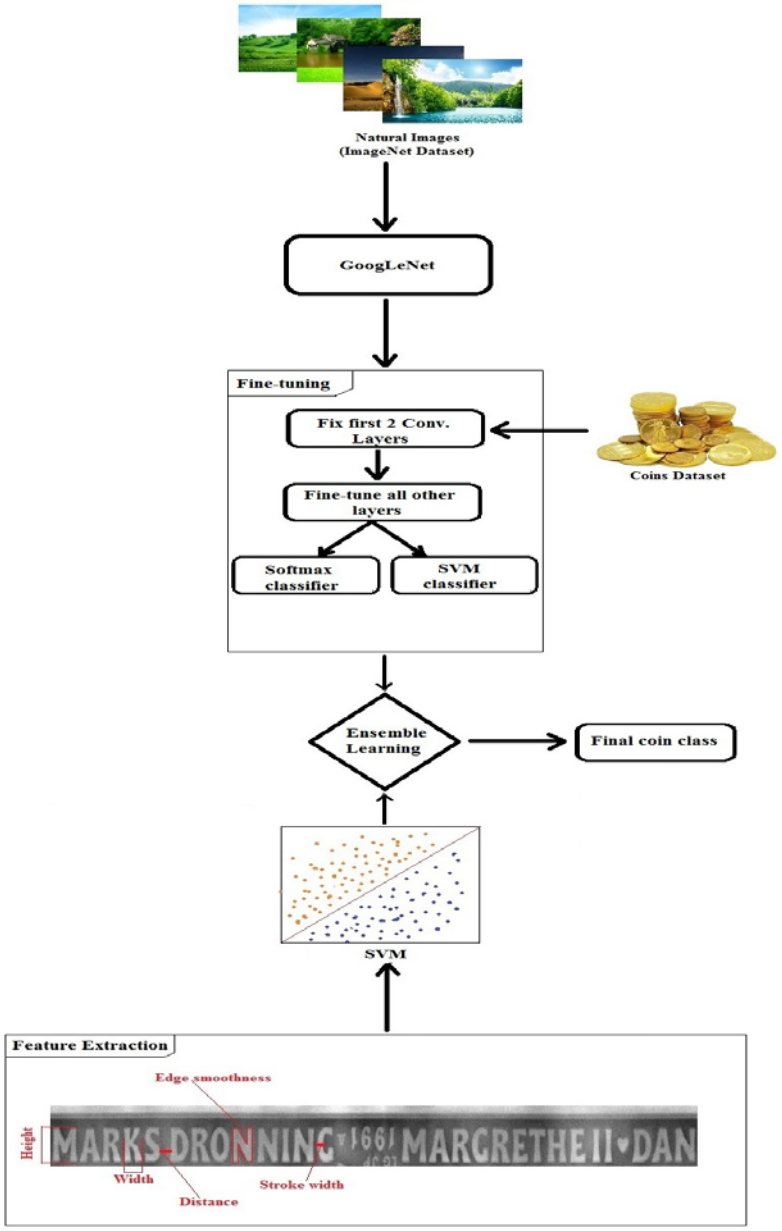


FIGURE 5-1 FRAMEWORK OF THE PROPOSED METHOD.

5.1 ENSEMBLE METHOD

The growing needs for coin authentication system have received the attention of researchers to develop reliable and cost-effective methods. The previous methods were based on physical characteristics and features of coins e.g. weight and size which are naïve compared to the new methods used by forgers nowadays. Therefore, this research proposes a new counterfeit coin detection method based on combining convolutional neural networks and other features mainly considered by human experts. Convolutional neural networks (CNNs) have been the topic of interest to many researchers for its classification results and optimization. CNN is a deep learning method that implicitly extracts features from images and learns comprehensive data from the image [78]. Training CNNs from scratch to perform an optimal feature extraction and to classify images requires a large number of images (tens of thousands if not millions), which is not feasible in our study. Hence, many researchers with limited datasets have used pre-trained CNNs on a different image dataset i.e. natural image and transfer learning from those CNNs to the new dataset. The transfer learning methods used in other domains such as face recognition or medical image classification reported the state-of-the-art results.

Like coin recognition systems, the proposed method starts by coin scaling (extraction), where the goal is to scale the coin to fit the whole image and remove the unnecessary background and marginal information. The coin scaling process has been performed to eliminate feature extraction from areas occurring outside the coin's border as discussed in Section 3.1. Figure 5-1 represents the general framework of the proposed method.

Additionally, the circular shape of the coin adds an extra challenge for CNN where extracting features of two coins belonging to the same class but having different rotational angles

yields different feature sets. Hence, all coins are rotated based on one reference coin w.r.t. the dataset they belong to. The process of rotating each test coin is performed on its binary image, where, at every rotation, it is compared to the reference coin and the distance between their pixels are registered. The minimum distance between each pixel of the reference image with the corresponding pixel in the test image is computed. Euclidian distance [79] is adopted to find the minimum distance between pixels in both images. Let $\bar{I} \in I, \bar{J} \in J$ be the two partial images of the test and the reference coins, respectively, where each is *m* by *n* in size, the Euclidian distance E_d is given by formula (5.1):

$$E_d^2(\bar{I},\bar{J}) = \sum_{i=1}^{m}\sum_{j=1}^{n}(\bar{I}_{ij} - \bar{J}_{ij})^2 \tag{5.1}$$

The whole image is then rotated based on the minimal Euclidian distance value obtained. The rotational step is crucial to adapt robust features from the reference coin and compare it to the identically rotated test coin.

The GoogLeNet architecture [11] is used to fine-tune the features to better extract features from coins. The features from convolution layers are used to train two classifiers: the softmax and SVM. Additionally, we used another SVM classifier trained on a set of character features from the coin. These include: *characters height*, *width*, *stroke size*, *edge smoothness*, *spaces between characters*, and *distances between the characters and the top and bottom rings*. A dynamic adaptive mask is used to handle the measures of each character separately. The dynamicity of the adaptive mask is achieved through finding the height and width of each character and then, decide the mask size w.r.t. the character size. These features are extracted only after applying the straightening algorithm which aims at transforming the circular shape of the coin into a rectangular shape. It should be noted that only part of the coin image, where the text appears, is transformed

as discussed in Section 3.2. The posterior probabilities of softmax and two SVM classifiers are used to determine the actual class of the coins.

5.2 CONVOLUTIONAL NEURAL NETWORK

Convolutional neural network (CNN) is the new era in machine learning. CNNs are a machine learning technique that uses deeper networks to implicitly extract features from images. Deep networks comprise different layers to extract more sophisticated information to better understand the image and need no engineering methods to extract features. CNN has four main operations that can be used in different orders based on the CNN architecture. *Convolution*, *Non-Linearity (Rectified Linear Unit)*, *Pooling or Sub Sampling*, and *Classification (Fully Connected Layer)* are the four operations.

Convolution: the convolutional layer is the prime layer to extract features from images. It extracts features using a small sliding window throughout the whole image. The weights of the sliding window is the same in each convolutional layer. Different number of sliding window sizes are often used in single CNN architecture. Figure 5-2 illustrates the convolutional layer process using a 3x3 sliding window. The stride is a setting of the number of pixels to slide the filter over the image. If stride is set to 1, the filter slides 1 pixel every time, as shown in Figure 5-2. Convolutional neural networks consist of multiple layers to improve the feature extraction and to build a better representation of these features. The initial layers learn generic features such as color and edge information, while the latter layers focus on learning image specific features.

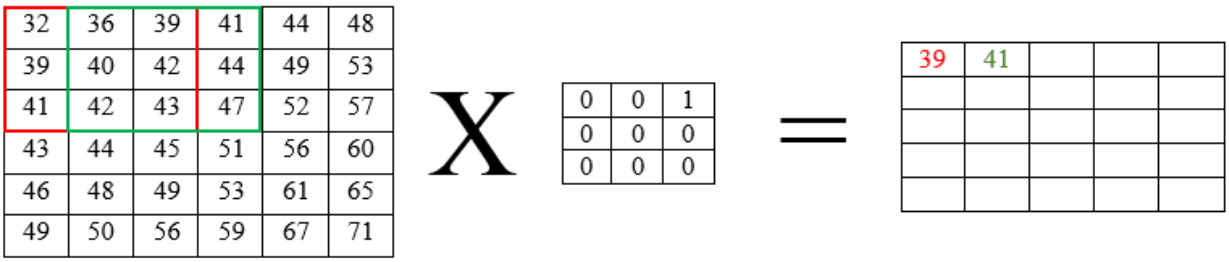


FIGURE 5-2 CONVOLUTIONAL LAYER CONCEPT

The sliding window is called the feature detector, kernel, or more commonly known as the filter. The filter is sliding through the image w.r.t. the stride setting and the output of every slide is the element-wise multiplication of pixel values by the filter values. The output of this process is written into a new matrix called the convolved feature or the feature map.

Rectified Linear Unit (ReLU): a non-linear operation whose goal is to transfer all negative value of pixels by zero in the feature map. The output of the ReLU layer is that $\forall \rho \in feature\ map;\ \rho = max(zero, \rho)$ where ρ is the pixel value.

Pooling or Sub Sampling: the goal of performing the pooling process is to reduce the dimensionality of the feature map while maintaining the most valuable and unique features. There are several pooling methods that can be used i.e. max, average, and sum. In max pooling, for example, another sliding window is defined to slide through the feature map and in every position, the maximum number from the feature values in the sliding window is returned in a reduced feature map. The same process applies for average pooling, where the average of the feature values is returned, and for sum pooling, which returns the sum of feature values in the sliding window. However, max pooling is the most widely used method due to its experimentally proven better results. The pooling operation also uses stride to define the number of pixels to slide the filter and

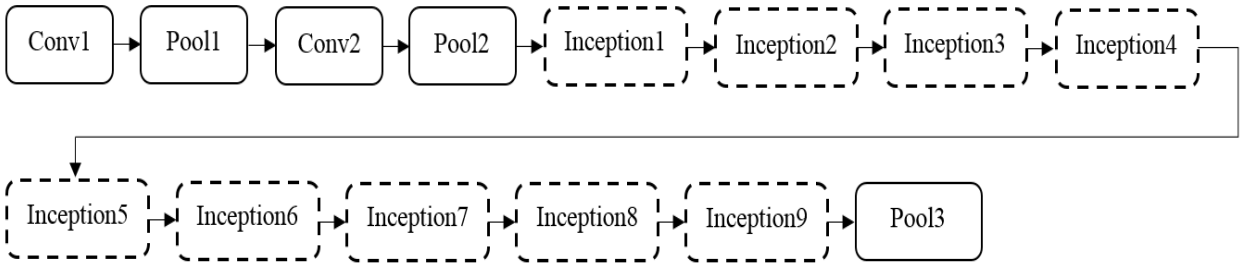
pooling stride can be different than the stride setting for the convolutional filter. Figure 5-3 shows the concept of max pooling operation using a 2x2 window with the stride set to 1.

FIGURE 5-3 MAX POOLING ILLUSTRATION

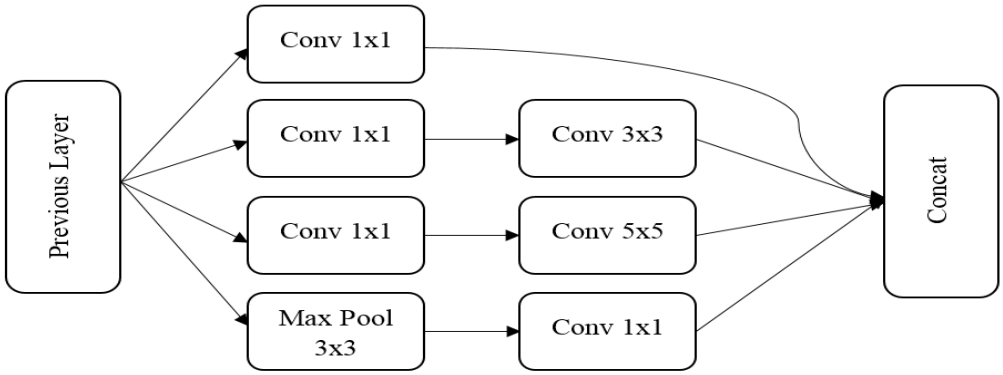
Fully Connected Layer: the fully connected (FC) layer is the classification layer and it uses the softmax activation function. This layer can be replaced by other classifiers such as SVM. By definition, the fully connected layer refers to connecting every neuron in one layer to every neuron in the next layer. The fully connected layer is usually placed after the convolutional and pooling layers, while the output of these layers is the input to the fully connected layers. The softmax is the activation function used in in the fully connected layer and it produces the probability of assigning each neuron to a class where the sum of all probabilities is 1.

Different architectures have been discussed in the literature w.r.t. the layers used, their number, and the layer distributions. GoogLeNet is one of those architectures introduced by Szegady *et al.* [11] in 2015 and it has outperformed all other architectures for the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) in 2014. GoogLeNet architecture introduced a network-within-network concept by using a new module, "*Inception*", which is a subnetwork containing a number of convolutional filters of different sizes and dimensions working in parallel and concatenating their outputs. GoogLeNet has two convolutional layers, three pooling layers, and nine "*Inception*" layers where each "*Inception*" layer comprises 6 convolution layers and 1

pooling layer making it a very deep network. Figure 5-4 illustrates the GoogLeNet convolution, pooling, and inception layers. The input image dimension of GoogLeNet is 224 × 224. The CNN architecture works by capturing a sparse representation of image information while gradually reducing dimensionality. GoogLeNet have shown better results in comparison to other CNN architectures and needs fewer trainable weights [11].



(A) GOOGLENET [11] CONSISTING OF TWO CONVOLUTIONAL, THREE POOLING, AND NINE INCEPTION LAYERS



(B) SAMPLE INCEPTION LAYER (INCEPTION5) REPRESENTATION [11]

FIGURE 5-4 REPRESENTATION OF GOOGLENET CNN ARCHITECTURE.

GoogLeNet was originally trained to extract features and classify natural images into 1000 classes. The filter weights are optimized to extract features from the ImageNet dataset of natural images. The fine-tuning process applies a continuous backpropagation learning to optimize the filter weights according to the new image dataset. The fine-tuning process that is used to transfer learning in this research keeps the first two convolution layers fixed, as those layers extract generic

type of features (i.e. edges), and fine-tunes all intermediate layers. While replaces the last fully connected (FC) layer intended for the 1000 classes of the original dataset used to train GoogLeNet into a new FC layer of two classes to fit our dataset of genuine and counterfeit classes. The original filter weights of the natural images are fine-tuned to optimize the weights for the coins dataset through backpropagation and stochastic gradient descent to find the optimal filter weights. On the other hand, the feature vectors from the fine-tuned layers are also used to train SVM classifier beside the new FC layer that uses softmax as activation function.

Assume that coin dataset *D* of *x* images, fine-tuning is an iterative process to update the filter weights *w* to reach the minimum error rate. As suggested in the literature, we reduced the learning rate and the best results achieved at learning rate $l = 6 * 10^{-4}$ for our dataset. Given that the original filter weights of pre-trained CNN are reasonably good and achieved a good classification results on the original dataset. The fine-tuning process uses lower learning rate to improve updating the filter weights responsibly.

5.3 FEATURES EXTRACTION FROM CHARACTERS

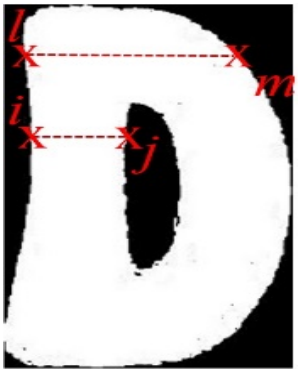
Characters represent a major part of the coin and they are the main characteristic used by human to recognize coins and by experts to authenticate them. The most common method to forge a coin is to strike a coin using a fake die that is, in return, molded from original coin stamp. This method results in a small variation in salient width which is reflected as the edges in a coin image. In addition, extracting features from characters is influenced by the findings from the literature, where counterfeit coin detection methods have been developed based on features from their characters. To find the characters on the coin, we have used the same work we proposed in Chapter 3 to locate the characters but not extracting them as shown in Figure 5-5(a).

After characters are located on the coin a set of features are extracted and used to classify coins. The stroke width, edge smoothness, character height and width, number of pixels, and spaces between characters. Stroke width calculation is inspired by the stroke width transformation method [80] where for every edge boundary pixel *i*, the algorithm traces the neighboring pixels until it reaches another edge boundary pixel *j* Figure 5-5(b). However, the proposed technique works on reading the pixels horizontally, but not vertically. Also, we consider the other gradient direction other than the actual stroke width such as the line between points *l* and *m* that spans over the whole character in Figure 5-5(b). The number of pixels between each edge boundary pixels is stored as stroke width. The character height and width are specified by the horizontal and vertical profiles that specify the size of the dynamic adaptive mask around each character.

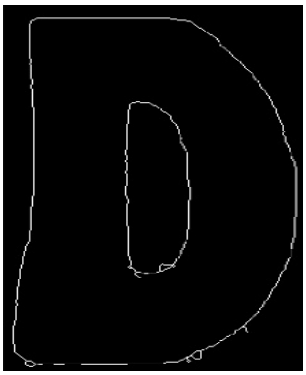
Moreover, the total number of edge pixels (stroke pixels) is calculated as there is deviation in character sizes between character of the genuine and counterfeit coins. In addition, the spaces between characters are calculated by taking the distance between the centers of every two immediate adaptive masks, as shown in Figure 5-5(d). Finally, the edge smoothness is found by first applying a thinning algorithm to reach one-pixel edges of the character stroke as shown in Figure 5-5(c). Then the number of thinned edge pixels are calculated as well as the pixels contained in the thinned edges.

These sets of features are then used for training a linear support vector machine (SVM), since we have only genuine and counterfeit classes. SVM works on finding a hyperplane separation with maximal margins between two classes. We employed the linear SVM with radial basis function (RBF) kernel to authenticate coins.

(A) DANISH COIN IMAGE AFTER APPLYING THE ADAPTIVE MASK TO LOCATE CHARACTERS



(B) MEASURING THE STROKE WIDTH OF CHARACTERS



(C) AFTER APPLYING THE THINNING ALGORITHM

(D) CALCULATING THE DISTANCES BETWEEN CHARACTERS

FIGURE 5-5 FEATURE EXTRACTION TECHNIQUES FROM CHARACTERS.

5.4 CLASSIFICATION

The classification results, along with their posterior probabilities of the three classifiers, are used to decide the class of a coin image. The feature sets from fine-tuned GoogLeNet are used to: (1) train the FC layer and the softmax function which discriminates classes based on the largest values in each feature vector while suppresses the less significant values, and (2) train a one-vs-one SVM classifier. Additionally, the character feature sets are used to train another on-vs-one SVM classifier. The posterior probabilities for both SVM classifiers are estimated and used with probabilities of the softmax function to improve the final classification results using $\hat{c} = \frac{\sum_1^k P_{i,k}(c)}{k}$,

where c is the classification result of classifier k for image i. $P_{i,k}(c)$ is the posterior probability of c for image i given by classifier k. $\hat{c}$ is the final classification results given to image i.

5.5 EXPERIMENTAL RESULTS AND DISCUSSIONS

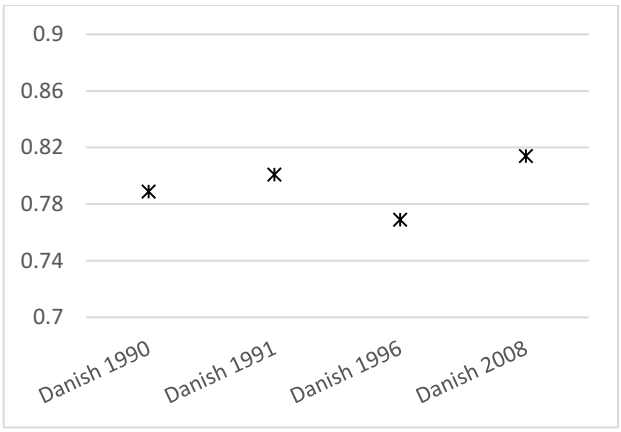
The proposed method was evaluated on different datasets that contain genuine and counterfeit Danish coins. The Danish coin datasets comprise the obverse side of Denmark 20 Korner of different years, as shown in Table 5-I. The original coin datasets provided by the Danish law enforcement have fewer number of coins than the one used in this research. The four datasets (Danish 1990, Danish 1991, Danish 1996, and Danish 2008) contain a total number of 173 counterfeit and 325 genuine coins, and are not equally divided between the four datasets. This number of coins is not enough to achieve good results and to fine-tuning a CNN architecture, especially not for learning CNN architecture from scratch. In addition, the large variation between the number of genuine and counterfeit coins in the datasets could highly affect the classification accuracy. Therefore, the original images of the four datasets have been augmented via a number of random transformations, including image rotation and noise addition. Data augmentation in convolutional neural networks has been widely used for its effectiveness in enhancing the feature learning process [81, 82]. Then, the datasets are randomly partitioned into training, validation and test sets as 60%, 20%, and 20% split respectively. The training set is used to train each classifier, while the evaluation of classifiers is conducted using the testing set. Furthermore, the validation set is used to set the parameter settings and further update the filter weights. In addition, the impact of using any single classification results and the parameters involved in achieving the proposed method is considered in the evaluation process.

We evaluated the three classifier results and the ensemble methods proposed in this chapter. The fine-tuned GoogLeNet with softmax (FTmax), the fine-tuned GoogLeNet with SVM (FTsvm), and character features with SVM (CFsvm) are the three classification methods based on different feature sets.

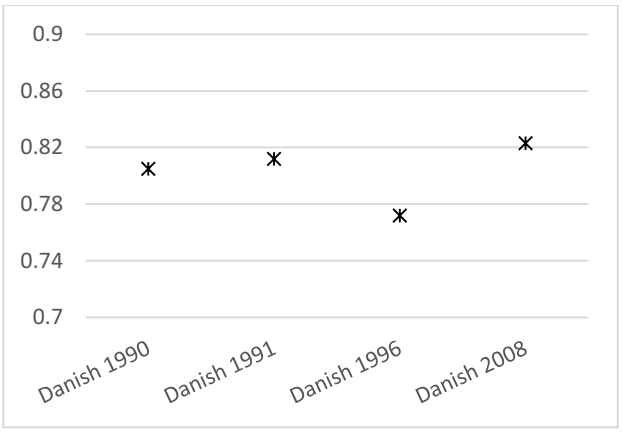
TABLE 5-I DANISH COIN DATASETS

Datasets	**Training Set**		**Validation Set**		**Testing Set**	
	Genuine	Counterfeit	Genuine	Counterfeit	Genuine	Counterfeit
Danish 1990	1144	1097	383	367	383	367
Danish 1991	1144	1097	383	367	383	367
Danish 1996	1144	1097	383	367	383	367
Danish 2008	1144	1097	383	367	383	367

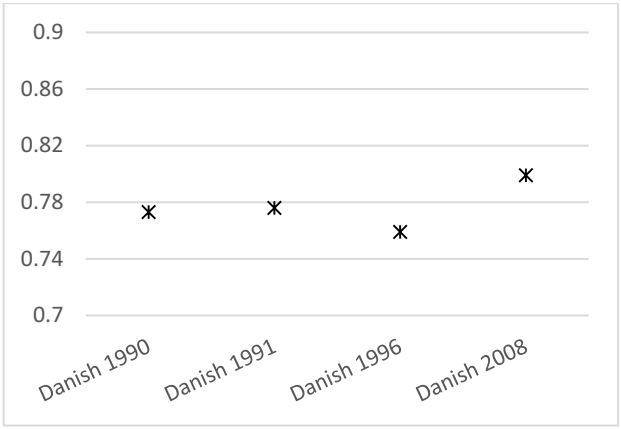
On the other hand, the proposed method (PM) has shown an improvement to the prediction of counterfeit coins as illustrated in Figure 5-6. These graphs demonstrate the precision results of the three classifiers and the proposed ensemble method. They also depict the precision rates of the four datasets used to evaluate the results of the three classifiers separately and represent the results of the ensemble method proposed in this chapter. The character feature sets are classified using binary SVM and are shown to return the lowest precision rate among other methods. This low precision rate can be seen as a result of the lower number of character features when compared to the number of CNN features, as well as confirm the findings in literature that the convolutional layers show a better representation of image features than hand crafted features.



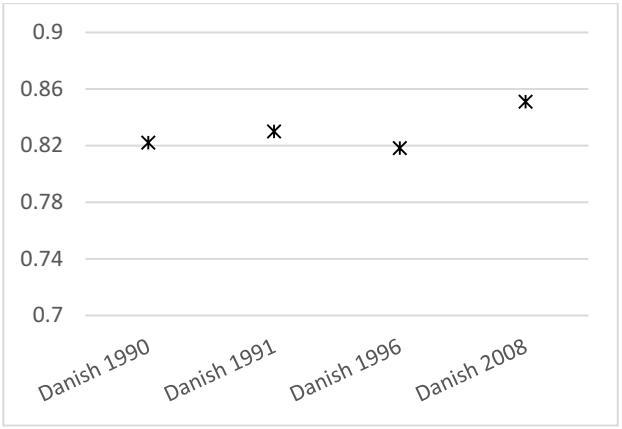
(A) FINE-TUNED GOOGLENET AND SOFTMAX



(B) FINE-TUNED GOOGLENET AND SVM



(C) CHARACTER FEATURES AND SVM



(D) THE PROPOSED ENSEMBLE METHOD

FIGURE 5-6 PRECISION RATES OF THREE CLASSIFICATION RESULTS AND THE PROPOSED METHOD.

The second lowest rates were obtained from the fine-tuned GoogLeNet feature set with softmax function that showed an improvement of 1% - 3.3% in precision rates when compared to SVM results of character features. Yet, both the fine-tuned GoogLeNet with softmax function and character features with SVM were below the precision results of fine-tuned GoogLeNet with SVM where the latter shows the best classification results among the other two methods and an average improvement of 0.97% over FTmax results. Our ensemble model has returned a precision rate as high as 85.1% for the Danish 2008 dataset.

We have also evaluated the number of correctly classified images using different combinations of two methods from the four methods. Figure 5-7 illustrates the normalized results of correctly classified coin images by a pair of methods. Every method has been combined with the other three methods and the results are reported in the column chart in Figure 5-7. The results of combining the method with itself has not been reported in the figure and the value is set to 0 to avoid any confusion.

The results suggest combing any two methods improves the classification results. The fine-tuned GoogLeNet with softmax and fine-tuned GoogLeNet with SVM combination shows a better classification results than using any single method. It also shows 1.49% more true positives than a combination of fine-tuned GoogLeNet with softmax and character features with SVM.

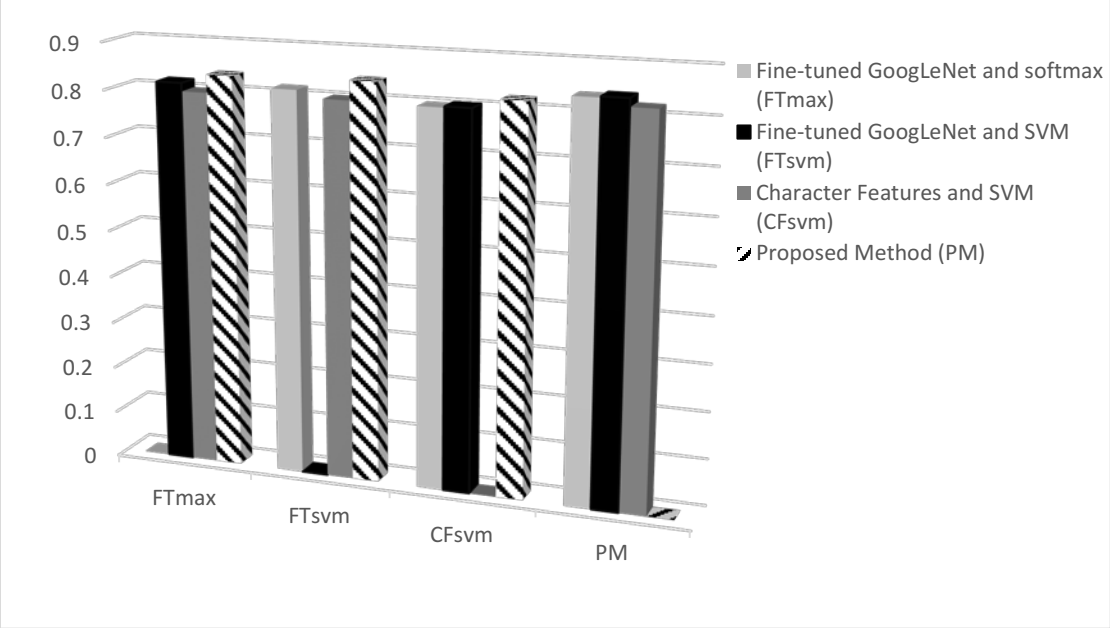


FIGURE 5-7 NORMALIZED NUMBER OF CORRECTLY CLASSIFIED COINS BY A COMBINATION OF TWO METHODS.

On the other hand, the proposed method achieves the highest number of correctly classified images when combined with any other method, where the combination of the proposed method and fine-tuned GoogLeNet with SVM shows the highest normalized number of correctly classified coins with a rate of 0.843.

Additionally, we compared the proposed method to the methods discussed in the literature in terms of performance and equipment. The proposed system achieved a promising precision rates using non-overlapping training, validation, and testing sets. The method required less hand-crafted features and achieved better results using deep learning features. The character feature sets were influenced by the results of the literature that achieved a high accuracy based on character features [19]. However, the use of the whole coin images to train the convolutional layers of a pretrained CNN have achieved a higher precision than features from characters only. In addition, the proposed ensemble method of combining the results of three classifiers achieved an even higher precision and was capable to distinguish counterfeit coins from genuine with less equipment requirements, as it was proposed by [20] using 3D image information.

However, although some methods in the literature have achieved higher accuracy than the proposed method, they require a hand-crafted set of features that can vary from coin to another. The proposed method is the first to use deep learning techniques on counterfeit coin detection and returned a promising results to the field. Training and tuning a CNN model suited for coins can improve the results if an appropriate number of coin images are available. Hundreds of thousands of coin images are required to build a tuned CNN model for coins from scratch that were not available for this research.

5.6 SUMMARY

In this chapter, we studied the problem of counterfeit coin detection and proposed an ensemble method of three classification results. The proposed method uses a fine-tune process to optimize feature extractors of pretrained CNN and transfer learning to improve feature extraction from our coin dataset. The extracted features from fine-tuned GoogLeNet were used to train two classifiers, softmax and SVM, and the classification probabilities of these classifiers are registered for the different coins. In addition, sets of features were extracted from characters minted on the coin and these sets were used to train another SVM classifier. The ensemble method was evaluated on different coin datasets that comprise genuine and counterfeit coins. The method was experimentally proven to be applicable to the counterfeit coin detection problem. Experimental results suggest that the use of the ensemble method can considerably improve the classification rate and enhance the classification probabilities to distinguish between coin images of different classes as shown in Section 5.5.

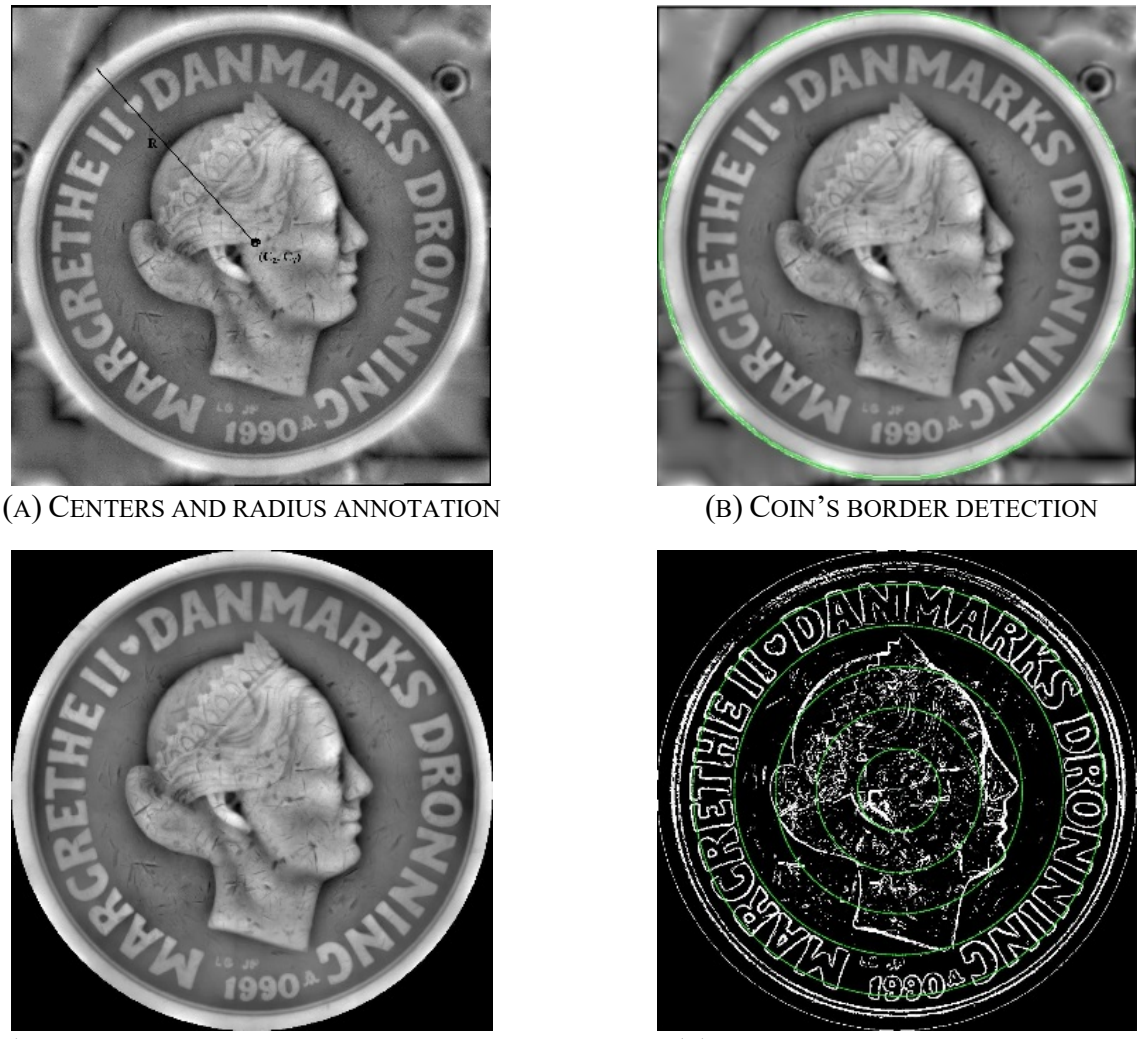
CHAPTER 6

COUNTERFEIT COIN DETECTION BASED ON EDGE FEATURES

Counterfeit coin detection systems are developed to differentiate genuine coins from fake ones. An efficient method based on a reliable set of features and single classifier could significantly reduce the computational costs. It has been experimentally established by other related works [19, 20] that the edges of counterfeit coin stamp are the prime indicator to distinguish between genuine and counterfeit coins, since even the high-quality forged coins have wider, taller, detached or missing strokes. This chapter describes a robust method for counterfeit coin detection based on coin stamp differences between genuine and counterfeit coins.

A set of measures based on edge differences are proposed in this chapter. The proposed method compares the *edge width*, *edge thickness*, *number of horizontal and vertical edges*, and *the total number of edges,* and *number of pixels* a test coin and multiple genuine reference coins. Additionally, the *Signal-to-Noise Ratio (SNR)*, *Mean Square Error (MSE)*, and *Structural Similarity (SSIM)*, which are well-known measures to track the differences between two images, are also applied to the coin image. The sets of features are then placed into an index space where each vector represents the features of one test coin and a reference coin. The final feature vector represents the feature set of one test coin and is computed by averaging the feature value of vectors in the index space.

The proposed method achieved high precision and recall rates, demonstrating the effectiveness and robustness of the selected edge features in authenticating coins. The method was evaluated on a real-life dataset of Danish coins as part of a collaborative effort.



(A) CENTERS AND RADIUS ANNOTATION (B) COIN'S BORDER DETECTION

(C) MASKING AND SEGMENTING THE COIN (D) CONCENTRIC CIRCULAR DIVISION

FIGURE 6-1 SEGMENTATION AND ROTATION OF DANISH COINS.

6.1 SELECTION OF EDGE FEATURES

The proposed solution starts by applying the coin segmentation and scaling methods to fit the coin over the entire image as discussed in Section 3.1. Then, it rotates the test coin image to match the reference coin using image matching. Figure 6-1 illustrates the segmentation, scaling,

and concentric circular division for the rotation of a Danish coin. The circular shape of the coin presents the challenge of comparing pixel-based values and generating the defect map. The ideal solution is to rotate the coin in all possible 360 degrees and find the exact match to a reference coin. However, comparing all pixels of the image 360 times is a costly process. Thus, we divide the image into circular areas as shown in Figure 6-1(d) and consider only the first two circular areas containing text and numbers to rotate over the respective areas on the reference coin. The circular area size is decided based on our experimental work and is equal to 1/6 of the original image size. The minimum distance between each pixel of the divided reference image with the corresponding pixel in the test image is computed. Euclidian distance [79] is adopted to find the minimum distance between pixels in both images. Let $\bar{I} \in I, \bar{J} \in J$ be the two partial images of the test and the reference coins respectively, where each is m by n in size, the Euclidian distance E_d is given by formula (6.1):

$$E_d^2(\bar{I},\bar{J}) = \sum_{i=1}^{m}\sum_{j=1}^{n}(\bar{I}_{ij} - \bar{J}_{ij})^2 \tag{6.1}$$

The entire image is then rotated based on the minimal Euclidian distance value obtained. The rotational step is crucial to extract robust edge features from the test coin and to compare it to the reference coin.

After rotating the test coin, a set of edge-based measures, focusing on structural information of edges while neglecting the intensities of edges, is extracted. The minimum and maximum edge widths, edge thickness, total number of edges, horizontal edges, vertical edges, and total number of pixels in the test coin that are not in the reference coin are computed in small areas of the coin. In addition, the widely used Signal-to-Noise Ratio (SNR), Mean Square Error

(MSE), and Structural Similarity (SSIM) are also used to assess the differences between the test coin and several reference coins.

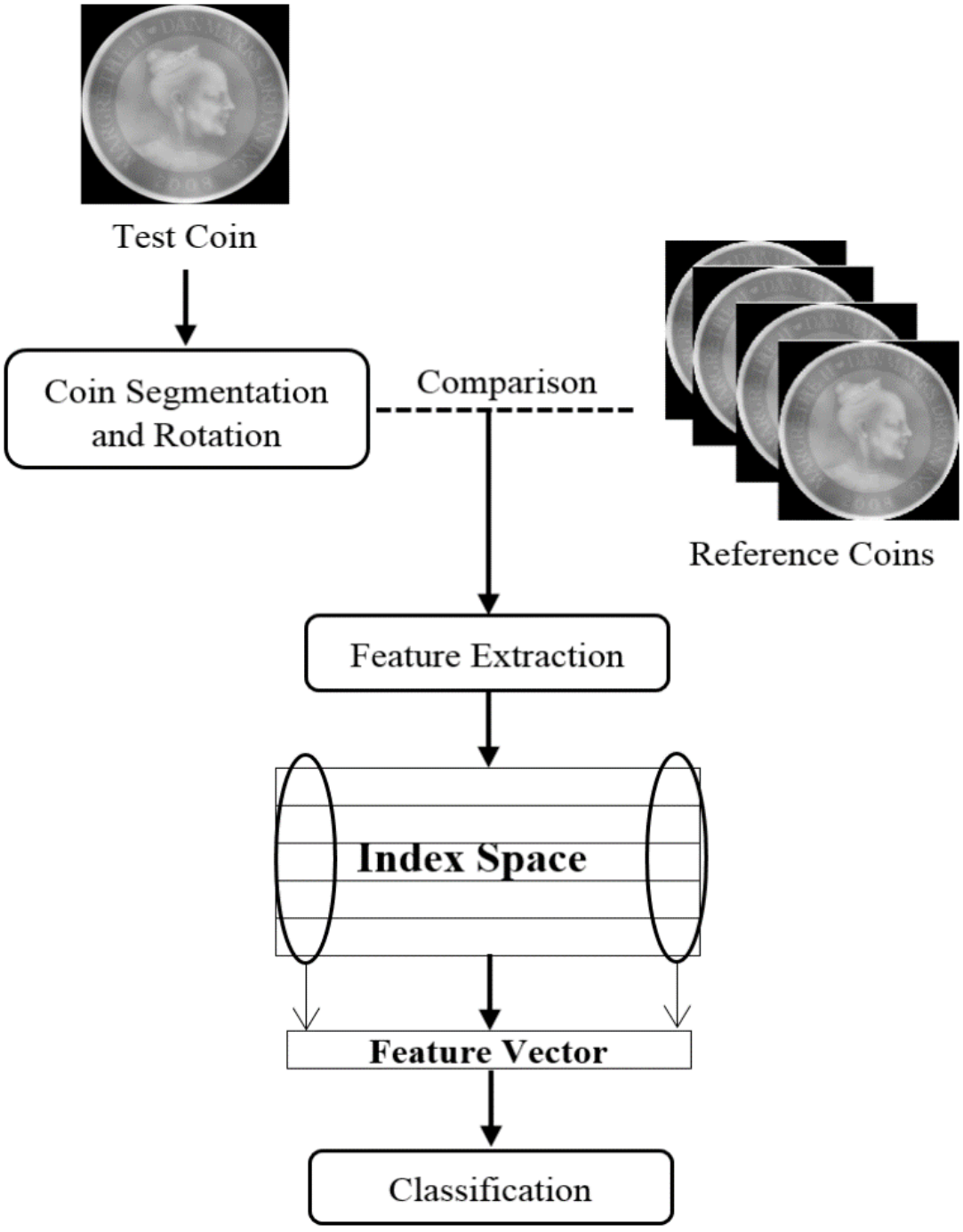


FIGURE 6-2 FRAMEWORK OF PROPOSED METHOD.

The nine measures, obtained from comparing a test coin with each reference coin, are represented in one feature vector in the index space. The index space has a similar number of vectors to the number of reference coins assigned. The final feature vector for each test coin is the average value of the vectors from the index space. Moreover, these features are applied to small areas of the edge map image to capture the edge characteristics in each small area. Figure 6-2 represents the framework of the proposed research. Finally, a classifier is trained and tested on the final feature vector.

6.2 FEATURE EXTRACTION

A great deal of efforts in developing image quality measurements based on known characteristics of human vision system (HVS) have been employed to evaluate the differences between two images. The comparison of pixel values has been widely used in several computer vision applications including digital forensics [83]. The most commonly used comparison is based on the RGB values in each pixel. In contrast, few methods are based on comparing the edge differences between the two images. This research focuses on the pixel value in edge images to measure the existence of edge values in genuine and counterfeit coins. To that purpose, the minimum and maximum edge widths, edge thickness, total number of edges, horizontal edges, vertical edges, and total number of pixels in the test coin that are not in the reference coin, or vice versa, are computed in small areas of the coin.

In addition, the widely used Signal-to-Noise Ratio (SNR), Mean Square Error (MSE), and Structural Similarity (SSIM) are used to measure the pixel differences between both images. The SNR and MSE are mainly used in images to measure their quality, after applying modifications such as compression, watermarking, or information hiding [83]. The SNR and MSE are computed

as a single value for the whole image to estimate the quality. However, the focus of this research is to find the edge differences of the coin stamp between the genuine and fake coins. Thus, the coin is divided into concentric circular areas to compare the pixels in each small area of the test coin with its corresponding pixels in the reference coin. Moreover, each circular area is further divided into two areas vertically w.r.t. the center point which yields half circular areas, hereafter in this paper referred to as regions of interest. Figure 6-4 (a) illustrates a Danish coin divided into a number of regions of interest. Such division allows capturing a detailed edge information and better represents the image information.

Since the test and reference coin images are identically rotated, a defect map is generated by a simple pixel value difference. The pixel value difference is a straightforward method that generates a defect map image containing all pixels in one image that are not in the other and vice versa. The defect map image $I(\chi, \gamma)$ has the same number of pixels as $K(\chi, \gamma)$ and $J(\chi, \gamma)$, and each pixel $p(x, y)$ has a value computed by equation (6.2) below:

$$p(x,y) = \begin{cases} |I_{xy} - J_{xy}| & iff\ I(x,y) > 0 \text{ and } J(x,y) > 0, \\ 0 & otherwise \end{cases} \tag{6.2}$$

Figure 6-3 shows a sample result of generated defect map of comparing two genuine and two fake coins with a single reference coin. The figure justifies the use of multiple reference coins to render the variations between within class coins (Figure 6-3 (a)), it also justifies the applicability of the selected measures to track the differences between the edges of a test and reference genuine coin (Figure 6-3 (b)). Now, let defect map *I(χ,γ)* be divided into small circular areas. The set of measures are applied to each region of interest $\overline{I}(\overline{\chi}, \overline{\gamma}) \in I(\chi,\gamma)$ and the results are is stored in the index space.

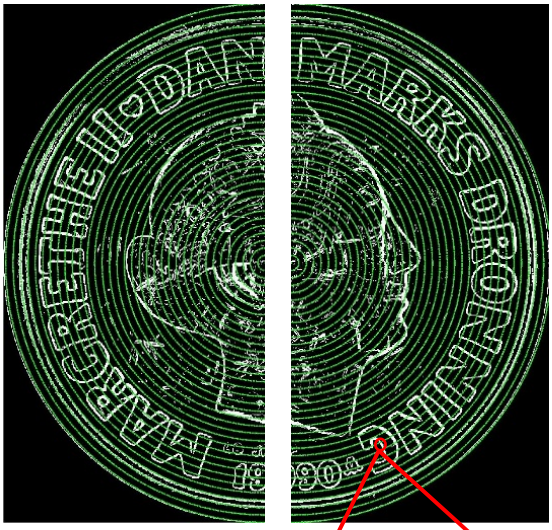
(A) A DEFECT MAP RESULTS FROM COMPARING TWO GENUINE TEST COINS WITH A GENUINE REFERENCE COIN

(B) A DEFECT MAP RESULTS FROM COMPARING TWO FAKE TEST COINS WITH A GENUINE REFERENCE COIN

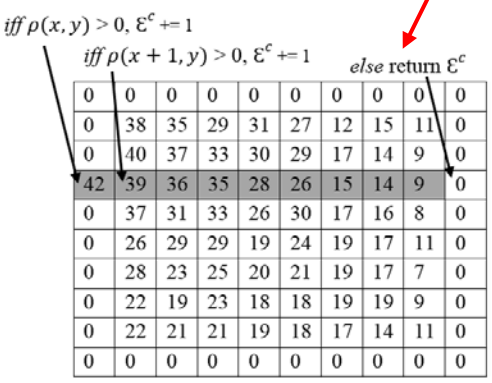
FIGURE 6-3 SAMPLE DEFECT MAPS

The set of measures are applied to the defect map image to extract the edge information. A similar method to the stroke width transformation is proposed in this research. The focus of the stroke width transformation method is finding the average width of text strokes based on variations between text and non-text edge sizes [80]. In coins, we have a mixture of edges which contain text, numbers, and other stamp features such as head profile. Therefore, the edge width count ε^c proposed here reads each non-zero pixel and adds up the number of horizontal consecutive non-

zero pixel as illustrated in Figure 6-4 (b). Edge width count can be defined as the length of a straight line from an edge pixel to another along the horizontal direction.

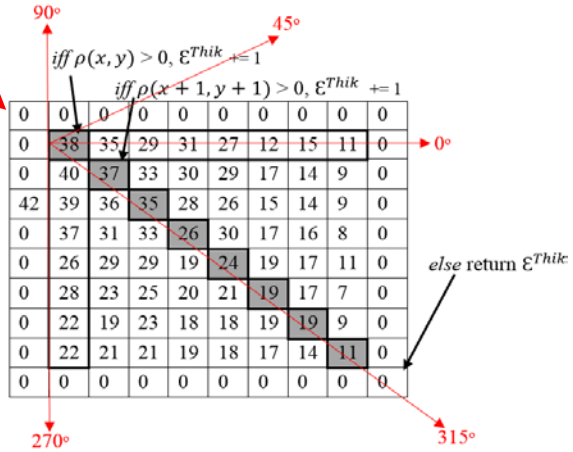


(A) REGIONS OF INTEREST REPRESENTATION



0	0	0	0	0	0	0	0	0	0
0	38	35	29	31	27	12	15	11	0
0	40	37	33	30	29	17	14	9	0
42	39	36	35	28	26	15	14	9	0
0	37	31	33	26	30	17	16	8	0
0	26	29	29	19	24	19	17	11	0
0	28	23	25	20	21	19	17	7	0
0	22	19	23	18	18	19	19	9	0
0	22	21	21	19	18	17	14	11	0
0	0	0	0	0	0	0	0	0	0

(B) AN ILLUSTRATION OF EDGE WIDTH COUNT MEASURE



0	0	0	0	0	0	0	0	0	0
0	38	35	29	31	27	12	15	11	0
0	40	37	33	30	29	17	14	9	0
42	39	36	35	28	26	15	14	9	0
0	37	31	33	26	30	17	16	8	0
0	26	29	29	19	24	19	17	11	0
0	28	23	25	20	21	19	17	7	0
0	22	19	23	18	18	19	19	9	0
0	22	21	21	19	18	17	14	11	0
0	0	0	0	0	0	0	0	0	0

(C) AN ILLUSTRATION OF EDGE THICKNESS MEASURE

FIGURE 6-4 REPRESENTATION OF REGIONS OF INTEREST AND EDGE WIDTH COUNT MEASURE.

Assume x is the *horizontal-axis* and y is the *vertical-axis*. Let $\rho(x,y) \in I(\chi,\gamma)$ be a non-zero pixel in any given $\overline{I}(\overline{\chi}, \overline{\gamma})$, the algorithm reads the consecutive pixels $\rho(x+i,y)$. The edge width count ε^c is increased by 1 *iff* the value of $\rho\ (x+i,\ y) > 0$ where $1 < i < \overline{\gamma}$. The highest and the lowest edge width values and locations of these edges w.r.t. $\overline{\gamma}$ are returned for each region of

interest. In stroke width transformation, a single value is returned representing the stroke width. We argue the interest in the highest and lowest edge width values is due to the mixture of edges on the coin where the strokes of the head profile and text have different widths and the wear effect on these strokes vary from coin to another. Therefore, considering the highest and lowest widths depict richer edge information. The edge width count proposed in this paper reveals the horizontal and diagonal width of edges depending on the angle of the edge w.r.t. the circular shape of the coin. Such measures have a better representation of edge information in counterfeit coin detection.

The edge has multiple information that can be represented in several forms to provide better understanding of the edge features. Edge thickness is another edge feature extraction algorithm proposed in this research. Edge thickness counts the largest number of connected pixels that form a single edge. This measure returns the largest and smallest numbers of consecutive pixels belonging to the biggest and smallest edges in each region of interest. Given $\rho(x, y)$, a non-zero pixel in any region of interest $\bar{I}(\overline{\chi}, \overline{\gamma}) \in I(\chi,\gamma)$, the algorithm counts the consecutive pixels $\rho(x+i, y+j)$, and the edge thickness ε^{Thik} is increased by 1 *iff* the value of $\rho\ (x+i, y+j) > 0$ where $1 < i < \overline{\gamma}$ and $1 < j < \overline{\chi}$. The algorithm reads the next pixel in horizontal, vertical, or diagonal direction based on the *x* and *y* values, as illustrated in Figure 6-4(c), where the horizontal, vertical, and diagonal consecutive pixels are counted and the longest value (highlighted pixels in this example) is returned for each edge. The edge thickness measure has a good representation of edge information and it reveals the total number of connected pixels in each direction. However due to the mixed nature of edges and a possible presence of noise, the largest and smallest edge thicknesses along with their directions are returned while the average is neglected. The direction is one of the 5 possible angles depends on the change in the *x* and *y* values.

On the other hand, the total number of edges in every $\overline{I}(\overline{\chi}, \overline{\gamma})$ is another good measure to capture similarities between coins belonging to the same class. The total number of edges $\mathcal{N}^{\varepsilon}$ counts the number of edges of lengths that are higher than certain threshold in each region of interest. The algorithm has been extended further to count the number of horizontal edges $\mathcal{N}_h^{\varepsilon}$and vertical edges $\mathcal{N}_v^{\varepsilon}$. The algorithm starts by reading the edge pixels in all directions *iff* the number of consecutive non-zero pixels is greater than λ threshold then $\mathcal{N}^{\varepsilon}$ is increased by 1, while the number of horizontal edges $\mathcal{N}_h^{\varepsilon}$ counts the number of horizontal edges having width greater than λ and the number of vertical edges $\mathcal{N}_v^{\varepsilon}$ counts the number of vertical edges having length greater than λ. The λ threshold heavily relies on the number of pixels in each circular area ϑ that is set to divide the coin into concentric circular areas, as discussed earlier in this chapter. The λ threshold equals $\vartheta/2$.

The number of pixels is also computed in every region of interest. Several counterfeit coin images have been released by authorities in which all of them are found to have mismatching edges with the genuine coins. Therefore, measuring the pixel differences between the test and the reference coin images is essential to establishing an acceptable range of pixel differences between coins in the same class and differentiating them from coins of another class. The number of pixels $\mathcal{N}^{\rho}$ is computed in each region of interest from the defect map $\overline{\Delta}\,(\overline{\chi}, \overline{\gamma})$ by equation (6.3):

$$\mathcal{N}^{\rho} = \sum_{i=1}^{\overline{\chi}} \sum_{j=1}^{\overline{\gamma}} \overline{\Delta}\,(i\,, j) \tag{6.3}$$

The number of pixels in every region of interest is added to the feature vector in the index space.

TABLE 6-I DANISH COIN DATASETS

Datasets	Total No. of Coins		Training Set		Testing Set	
	Genuine	Counterfeit	Genuine	Counterfeit	Genuine	Counterfeit
Danish 1990	1132	1132	792	792	340	340
Danish 1991	1132	1132	792	792	340	340
Danish 1996	1132	1132	792	792	340	340
Danish 2008	1132	1132	792	792	340	340
Total	4528	4528	3168	3168	1360	1360

Moreover, each region of interest of the test coin $\overline{I}(\overline{\chi}, \overline{\gamma}) \in I(\chi,\gamma)$ and the reference coin $\overline{J}(\overline{\chi}, \overline{\gamma}) \in J(\chi,\gamma)$ are compared pixel to pixel by the SNR and MSE formulas (6.4) and (6.5) given below:

$$SNR = 10 * \log_{10} \frac{\sum_{i=1}^{\overline{\chi}} \sum_{j=1}^{\overline{\gamma}} \overline{I}_{ij}^{\,2}}{\sum_{i=1}^{\overline{\chi}} \sum_{j=1}^{\overline{\gamma}} \left(\overline{I}_{ij} - \overline{J}_{ij} \right)^2} \tag{6.4}$$

$$MSE = \frac{1}{\overline{\chi} * \overline{\gamma}} \sum_{i=1}^{\overline{\chi}} \sum_{j=1}^{\overline{\gamma}} \left(\overline{I}_{ij} - \overline{J}_{ij} \right)^2 \tag{6.5}$$

The SNR and MSE values are somewhat complementary as both compare pixel to pixel values and have different representations of the pixel difference values. The goal is to increase the weight of pixel difference values for classification, and the different representations will render the variation between coins of the same class. In addition, the SNR and MSE have single value for each region of interest. Therefore, the smaller the circular area, the better representation of edge information, the higher the accuracy rates. The size of each concentric circular area decides the

size of feature vector as well. Hence, the experimentally chosen size of 50 pixels as the width of each circular area in this research is debated in the discussion on experimental results in Section 6.5.

Another image quality measure, used to track differences between the test and the reference coin, is based on comparing the perceived change in structural information of the reference and the test signals. The structural similarity (SSIM), proposed by Wang *et al.* [84], extracts structural information that are highly perceptual by the human vision system and measures the structural information changes between the two images. The luminance similarity (LS), contrast similarity (CS) and structural similarity (SS) are computed in each small block, and the product of these three similarities is used to predict the differences between the two images. The SSIM has been widely used in assessing the image quality. SSIM index is the value obtained from SSIM that reflects the image quality. SSIM index has a value between 0 and 1, where 1 indicates the highest quality.

In this research, we applied the SSIM on each region of interest of the test and reference coins and the SSIM index is expanded to return a value between 0 and 1000 to increase the variation range between images of different classes.

6.3 INDEX SPACE

Both counterfeit and genuine coins can appear in variant qualities, from mint state to highly degraded coins. These variations may arise from coin wear or contamination caused by daily use. The variation appears clearly on the edge strokes, which protrude from the background of the coin. Therefore, considering multiple reference coins having different wear levels to compare to every test coin can highly render those variations in between interclass coins.

The feature extraction described in Section 6.2 is performed by comparing the test coin with each reference coin. The feature sets resulting from feature extraction methods are stored in one vector in the index space. In other words, each vector in the index space contains all features extracted from comparing the test coin with one reference coin. Let I_T be the total number of coin images and $\mathcal{R} = \{I_{\mathcal{R}1}, I_{\mathcal{R}2}, \ldots, I_{\mathcal{R}\mathrm{N}}\} \in I^{\mathrm{T}}$ be N reference coins, the mapping of each reference coin $I_{\mathcal{R}i}$ to the feature set in the indexing space obtained by comparing the test coin to a reference coin is defined as:

$$\Phi_N^{\mathcal{R}} : \begin{matrix} I_{\mathcal{R}1} \\ \vdots \\ I_{\mathcal{R}N} \end{matrix} \rightarrow \begin{pmatrix} \mathcal{F}_{\mathcal{R}_1 1} & \cdots & \mathcal{F}_{\mathcal{R}_1 k} \\ \vdots & \ddots & \vdots \\ \mathcal{F}_{\mathcal{R}_N 1} & \cdots & \mathcal{F}_{\mathcal{R}_N k} \end{pmatrix} \tag{6.6}$$

where $\mathcal{F}_{\mathcal{R}_i k}$ is a single feature value obtained by comparing the test and the i^{th} reference coin. k is the total number of features obtained from the feature extraction methods. N is the total number of reference coins.

Therefore, the number of vectors in the index space depends on the number of reference coins. Sets of experiments have been conducted to determine the number of reference coins necessary to reduce the variation between coins of the same class. Finding the best setting for the number of reference coins is discussed in the experimental evaluation Section 6.5.

The final feature vector prepared for the classifier is computed from the index space. The feature values stored in the final feature vector are the average values of each feature from the different vectors.

6.4 CLASSIFICATION

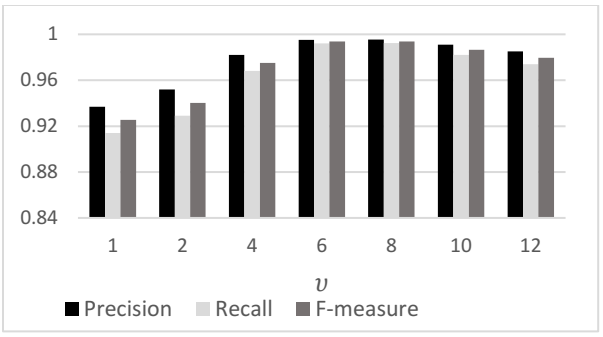
Counterfeit coin detection systems are deployed to decide the coin class to be either genuine or counterfeit. The high-quality counterfeit coins have minor variances to the genuine ones, which requires further caution to distinguish between them and the genuine coins. In addition, the counterfeit coins are produced by different sources of forgery, which increases the variations between the within class coins. Therefore, a further reduction in variations between coins of the same class is necessary. The principal component analysis (PCA), introduced earlier in Section 4.4, to transform the feature vector dimensional space into a subspace that has better distinguishable feature values.

Moreover, a two-class classifier is used to train a set of features from genuine and counterfeit coins. In this research, we employ the *Support Vector Machine* (SVM) classifier with a linear kernel function is selected for the proposed system for better accuracy, faster computation, and less overfitting. Moreover, one can argue that having a single class for counterfeit coins is not reasonable, since there are multiple sources of forgery. The proposed method considers only rejecting counterfeit coins while having a second classifier for rejected coins can help classifying the coins into multiple classes based on the source of forgery. However, this process goes beyond the scope of this research.

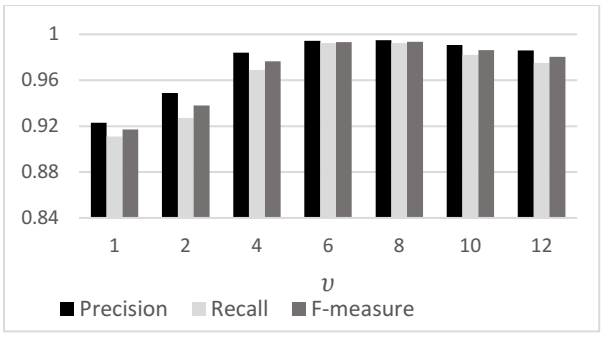
6.5 EXPERIMENTAL RESULTS

The goal of our experiments is to evaluate the performance and accuracy of the proposed counterfeit detection method. Different settings of parameters configuration were tested and evaluated to achieve the highest accuracy. Datasets play an important role in evaluating the system performance. Thus, the experiments are conducted on real life datasets prepared by the Danish authorities containing both counterfeit and genuine coins. The images were scanned by specialized scanner in one of the leading digital forensic firms. However, the original images of the four datasets have been augmented using a random transformation, including image rotation and noise addition. The coin images were split into training and testing in which 70% of the images were used for training and 30% for testing as shown in Table 6-I. The number of pixels in every region of interest is added to the feature vector in the index space. The reference coins selected for training and testing are genuine coins with different qualities in terms of contamination, orientation, and wear. However, all images were rotated based on one reference coin w.r.t. the datasets as discussed in Section 6.1.

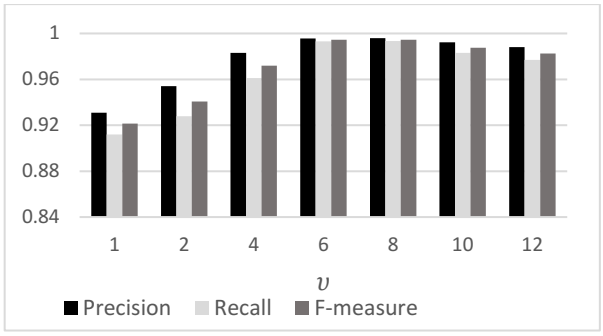
The proposed solution works by extracting different features from the coin edges and representing those features in the index space. Then, the final feature vector is computed for each test coin and used as the input to the classifier for training. Finally, the coins are classified into two classes, genuine and counterfeit, based on pixel-based edge features.



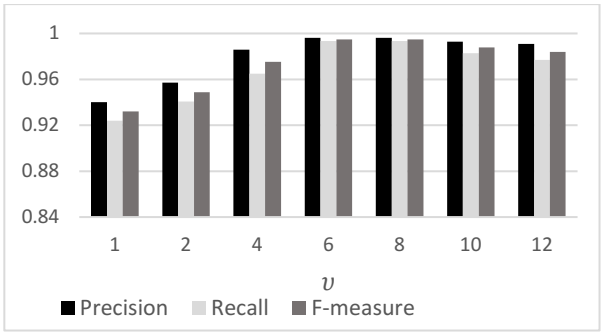
(A) DANISH 1990



(B) DANISH 1991



(C) DANISH 1996



(D) DANISH 2008

FIGURE 6-5 *RECALL*, *PRECISION*, AND *F-MEASURE* RATES FOR DIFFERENT υ SETTINGS W.R.T. THE FOUR DATASETS.

In this experiment, sets of feature vectors are extracted from each test coin against the reference coins. Then, the final feature vector is computed by averaging the feature vectors in the index space. We employed *recall*, *precision*, and *f-measure* to evaluate the performance of the proposed method.

Several parameter settings of the proposed method were evaluated to select the best parameter settings; the parameter settings that achieved the highest *f-measure* values were selected for training and testing the classifier. The number of reference coin images to minimize the in-class variations were evaluated. Different number of reference images υ were selected to evaluate the performance of the proposed method. Figure 6-5 represents the *recall*, *precision*, and *f-measure* values for different υ selections on each dataset. The lowest *f-measure* value was obtained by

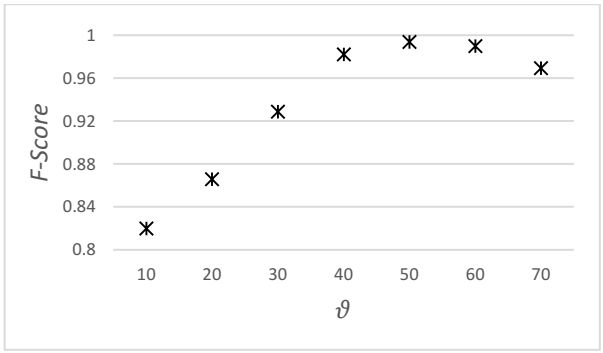
comparing the test image with one reference coin. The low *f-measure* value confirms the necessity of comparing the test coin with multiple reference coins to reduce the same class variations.

However, the classification rate improved gradually as the value of υ increased. The highest *f-measure* value was obtained when comparing the test coin with six reference coins. A 7.63% improvement in *f-measure* value was returned using six reference coins instead of one reference coin. Based on the dataset, the accuracy remains steady, or improved 0.0002 at its best when υ is set to 8. Such increase in υ was not considered in this research to maintain the number of reference coins required to the lowest possible that achieves the highest accuracy rates. Meanwhile, a further increase in the number of reference coins slightly reduced the *f-measure* rate.

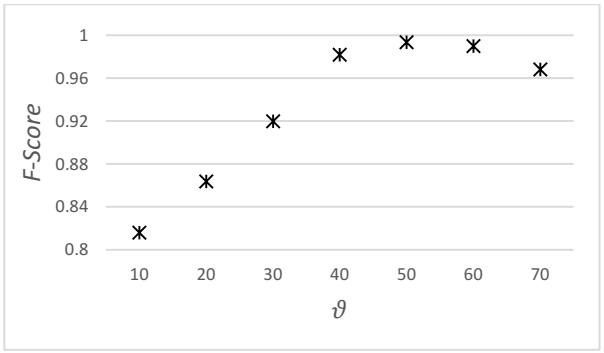
A 1.42% drop in the *f-measure* value was reported when υ is set to 12. Therefore, we conclude that the increase would have a negative impact if the experimentally reported threshold is exceeded due to the increase in noise level. Overall, the four datasets performed well and returned the highest accuracy by comparing the test coin image with six reference coins. υ is set to equal six, which returned the highest accuracy in the rest of the experimental work.

Additionally, the number of concentric circular areas in each coin was evaluated to select the parameter settings when dividing the coin into small regions of interest, as discussed in Section 6.2. The threshold for the number of concentric circular areas was evaluated based on the width of each circular area in terms of number of pixels. In other words, ϑ is the number of pixels deciding the width of each circular area. Therefore, an interval set of values of ϑ were evaluated to find the best parameter setting. Figure 6-6 illustrates the performance of the proposed method for different numbers of circular areas in each dataset. The feature extraction methods proposed in this research

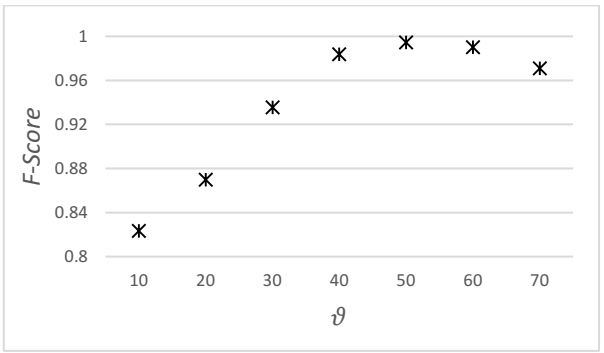
are based on pixel value comparison. Thus, the number of pixels for the width of each region of interest has a strong impact on the classification accuracy rate.



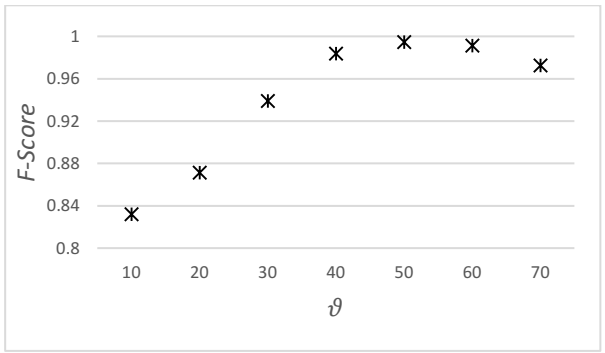
(A) DANISH 1990



(B) DANISH 1991



(C) DANISH 1996



(D) DANISH 2008

FIGURE 6-6 *F-MEASURE RATES* FOR DIFFERENT ϑ SETTINGS W.R.T. THE FOUR DATASETS.

The *f-measure* values captured in Figure 6-6 for different ϑ settings suggest that considering 10 pixels in each circular area would return the lowest accuracy rate. In addition to the higher computational cost when selecting smaller circular areas, a very small ϑ setting affects the feature representation and produces a larger feature vector. The unnecessary large feature vector can play a vital role in triggering overfitting issues. The *f-measure* value steadily increases along with the ϑ value, and achieves the highest *f-measure* rate when ϑ is equal to 50. Moreover, the captured *f-measure* rates for the four datasets are almost the same where the highest *f-measure* drop returned when ϑ is equal to 10. The *f-measure* rate starts to decrease as the number of pixels

increases above 10 pixels. The *f-measure* rate drops approximately 2.44% when the width of each circular area reached 70 pixels, reflecting an increase of 1.62% of the overall *f-measure* rate reduction when ϑ equals 60.

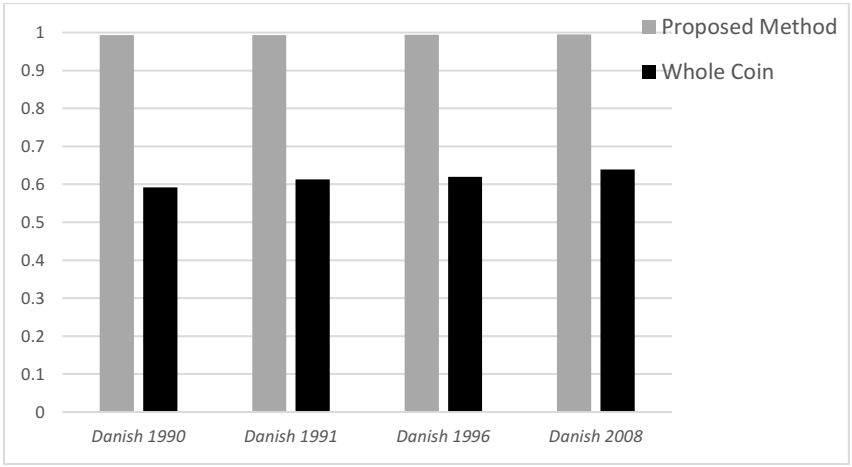
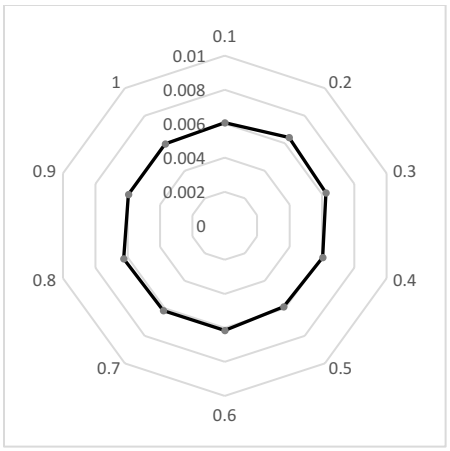
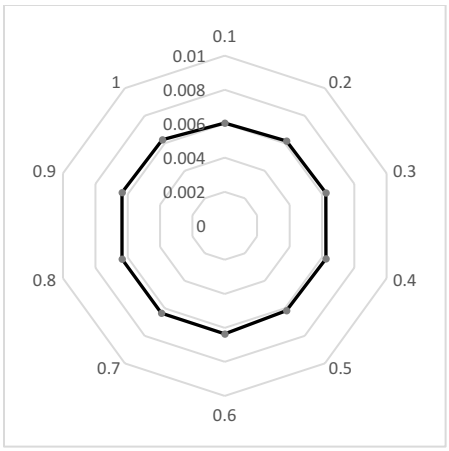


FIGURE 6-7 *F-MEASURE* RATES OF THE PROPOSED METHOD AND FEATURES FROM THE WHOLE COIN IMAGE OF THE FOUR DATASETS.

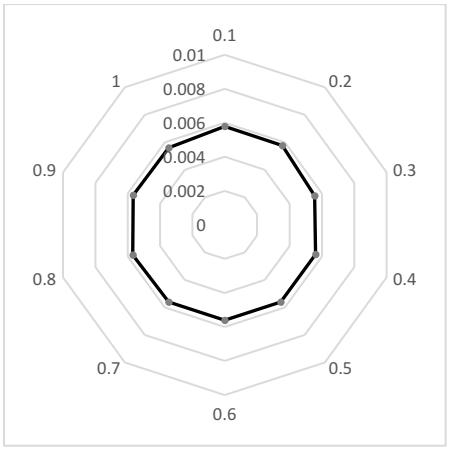
Generally, the proposed method reports a high accuracy rate that outperforms the state-of-the-art results. The focus on small circular areas out of the entire coin image added better representation of the feature set. In order to reflect the advantages of the proposed method, a comparison between the discussed method in this paper and features extracted from the whole coin image was conducted. Figure 6-7 depicts the *f-measure* rates of the proposed method and features extracted from the whole coin image. The features of the whole coin image are extracted using the same feature extraction method of this work (Section 6.2). However, the feature extraction methods (except edge width count and edge thickness, which returns 4 values each) of this research return a single value for each circular area. Thus, applying it to the whole coin image would return 15 feature values only, which makes it an unsuitable and prone to overfitting.



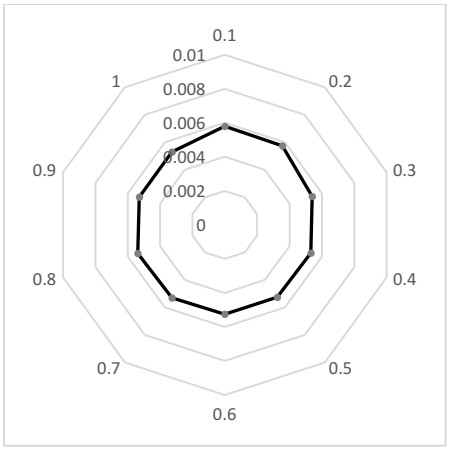
(A) DANISH 1990



(B) DANISH 1991



(C) DANISH 1996



(D) DANISH 2008

FIGURE 6-8 *CLASSIFICATION ERROR RATE* IN EACH DATASET.

To perform such an experiment, the 15 features of every comparison between the entire test and reference coin images are added to one vector, followed by the 15 features of the second comparison with another reference coin image and so on. A total of 90 features (consisting of 15 features x 6 reference coins) in every vector are used to classify the coins. Let $\mathcal{L}$ be a feature vector and $\mathcal{L} = \{f_1(I,J_1), f_1(I,J_2), \ldots, f_n(I,J_1), \ldots, f_n(I,J_k)\}$, where $f_n(I,J_k)$ is the k^{th} feature from comparing a test image I and a reference image J, and n is the total number of reference coin images. The Euclidian distance method is used to classify the feature vector $\mathcal{L}$ into a genuine and counterfeit coin. The overall performance is illustrated in Figure 6-7.

The findings of this experiment confirm the findings of the previous experiment, which concluded that the larger the circular area is, the lower the *f-measure* rate. From the literature, considering features from specific regions of the coin is the common approach for counterfeit coin detection systems. The feature vector has better image information representation by considering smaller details.

Furthermore, the general performance of the proposed method was evaluated. Figure 6-8 represents the overall performance of the proposed method on the four datasets. In this experiment, the training and testing sets were split into 10 equal sets for each dataset and used to train and evaluate the classifier. The performance was additionally evaluated by the *classification error rate*.

The *classification error rates* were reported over the 10 splits of each dataset. The average error rate was 0.00585, which reflects the effectiveness of the selected feature extraction methods. The *classification error rate* slightly decreased or increased between different splits of the same dataset and between splits of other datasets. The steadily changing rates confirm the robustness of the proposed method when used for different coin types. The lowest *classification error rate* was reported to Danish 2008 dataset at 0.00525 while the highest was 0.00641 belonging to Danish 1991 dataset. Overall, the proposed method achieved a high accuracy rate that outperformed the accuracy rates reported in the literature, with a lower computational cost. The extracted features are domain independent and can fit any coin type regardless of the language or color of the coin. It also has a lower sensitivity to coin wear and contaminations due to daily use of the coin.

Table 6-II shows the performance of our method as well as that of the other methods discussed in the literature. The four directly related research works have shown promising results to the field. These research works proposed different feature extraction and representations to

authenticate genuine coins and reject counterfeit ones. One of these research works focused on 3D coin images and three used the 2D representation of coin images. Three related works ([19, 20] and the method proposed in Chapter 5) focused on the textual area of the coins. The authors in [69] used an optical mouse to capture small parts of the coin image and to compare it to a reference coin. The authors worked on a dataset of 2-Euro coins. The proposed method took advantage of the low cost and ease-of-use of the optical mouse. Their method achieved good accuracy rates, but suffered from the limitation of the optical mouse scanner that could only capture 1/14 of the entire coin size at a time.

TABLE 6-II PREVIOUS COUNTERFEIT COIN DETECTION METHODS COMPARED WITH OUR PROPOSED METHOD

Research Paper	Proposed Method	Accuracy Rate
Tresanchez et al. [69]	Optical mouse	97%
Sun et al. [19]	Character measures and local image features	~97%
Khazaee et al. [20]	Height and depth information of test edges obtained from the 3D representation of coin images	99.3%
Ensemble Method [Chapter 5]	Ensemble method of two deep learning features and one text edges features	85.1%
Liu *et al.* [21]	SIFT descriptor applied to local areas	98.6%
Proposed Method	Pixel-based features based on test and a reference coin comparison	99.42%

Sun et al. [19] proposed a combination of text characteristics and local image features to represent the feature vector. The text characteristics included stroke width, height and width of characters, relative distances, and angles between characters. While maximally stable extremal

region (MSER) was used for texture features. Their proposed method achieved promising results, but was experimentally tested on 11 genuine and 2 fake coins only.

Khazaee *et al.* [20] proposed another feature set based on height and depth information obtained from the 3D image representation of the coin. The height and depth information of textual area of the coin was considered in their study. A straightening algorithm was applied to first transform the text area from circular to rectangular, and then, the height and depth data were extracted. The method has achieved better results than [19] but was limited to a 3D coin image and requires special equipment to capture such an image. Hmood *et al.* [85] proposed another counterfeit coin detection method using convolutional neural network and text features. They achieved encouraging results, but their method suffers from a high computational cost and requires a large number of images to retrain the CNN on another coin type. Finally, Liu *et al.* [21] used the SIFT descriptor to represent the keypoints information that are extracted from local area of the images. The author stated that the use of smaller areas rather than the whole coin image has improved the overall classification rate. The proposed method in [21] relies solely on the SIFT keypoints to classify the genuine and fake coins. However, the keypoints of a high-quality fake coin could trick the classifier. Additionally, the mechanism of how keypoints work illustrates the limitation of using such a measure to detect counterfeit coins.

Our proposed method returned the highest classification accuracy of up to 99.42% among the other research studies in the literature. The number of reference coins and the number of circular areas have a high impact on the overall performance. The selected features in this research provides a good representation of pixel-based differences and shows a significant improvement over other methods discussed in related works. It also reduced the number of *false positives* by

using the index space to render the variations between interclass coins and increase it for intraclass ones.

6.6 SUMMARY

In this chapter, we studied the problem of counterfeit coin detection. We proposed *edge-based* differences method for coin authentication using set of edge pixels differences measures. The proposed solution was evaluated on four real life coin datasets containing genuine and counterfeit coins. The challenges of this work are the coin orientation, heavily degraded coin quality, illumination, wear, and scratches. The proposed method compares the edges of test coin with edges a reference coin. Nine different measures are considered in this study, including minimum and maximum edge widths, edge thickness, total number of edges, horizontal edges, vertical edges, and total number of edge pixels in the test coin that are not in the reference coin, as well as Signal-to-Noise Ratio (SNR), Mean Square Error (MSE), and Structural Similarity (SSIM). These measures are applied to small concentric circular areas of the coin. The method was experimentally proven to be reliable in rejecting counterfeit coins and authenticating genuine ones having different coin qualities. Experimental results suggest having more than one reference coin can render the variations between coins of the same class caused by circulation, therefore improving the accuracy. The pixel-based difference measures show a robust feature representation for counterfeit coin detection with a much lower computational cost.

CHAPTER 7

CONCLUSION AND FUTURE WORK

The purpose of this thesis is to study, design, and implement coin recognition and counterfeit detection systems focusing on a smaller and yet reliable set of features. Character segmentation from coins, coin recognition, and different counterfeit coin detection are presented and implemented in this thesis and the evaluation results of these systems are illustrated and discussed. While the evaluation results were reliable and promising and the systems addressed the coin recognition and counterfeit coin detection problems, there are still some challenges and limitations. Therefore, several suggestions are discussed in Section 7.2 for further research and improvements.

7.1 CONCLUSIONS

The implementation of computer vision methods for coin recognition and counterfeit coin detection are more challenging than it is for other images, such as face and pedestrian recognition, for reasons discussed earlier in this thesis. The vast majority of coins are round and share similar structure such as a symbol (i.e. head profile), which appears in the center of the coins within the inner circle, while text and numbers are distributed around the outer circle. There are several research works for counterfeit coin detection and coin recognition that are studying and analyzing the character features of coins. However, we found no prior automated solution that segment characters from coins before analyzing and extracting features from them. To the best of our knowledge, we proposed the first method to automatically locate and segment characters from coin

images (Chapter 3). This method is scale and rotation invariant while being able to process characters of different languages [86].

Moreover, in order to implement a reliable system to distinguish between coins of different classes, we proposed a coin recognition system that captures the human vision system processing by considering the text area of the coin (Chapter 4). Text is one of the two parts of the coin's visual information and human tends to read the textual area to classify coins. Therefore, a dynamic-HOG method is discussed, which finds the height and width of each character and applies a dynamic mask w.r.t. the character height and width. Finally, a set of Histogram of Oriented Gradients (HOG) is extracted for each character. Placing a dynamic mask to characters is a recursive process, where after initiating the heights and widths of characters, the method considers the partially detached character strokes. The proposed method achieved high accuracy rates and significantly reduced the recognition time by considering characters only rather than the whole coin image [87, 88].

An ensemble method is proposed to detect counterfeit coins based on the majority voting of three classifiers. The method takes advantage of the advancements in machine learning methods by introducing convolutional neural networks. Two classifiers are trained on convolutional features while a third classifier is trained on character features (Chapter 5). The final class of a coin is decided based on a voting system based on the probabilities of the three classifiers. The obtained results are promising and confirm the effectiveness of the proposed method [85, 89].

In order to reduce the computational costs of the counterfeit coin detection system, we proposed a new system based on edge features (Chapter 6). Edge mismatches between genuine and counterfeit coins are the major lead to distinguishing between different classes. The edges of

counterfeit coins are wider, thinner, missing, or misplaced if compared to genuine coins, as discussed in the literature. This thesis proposed several precise edge measures to capture those differences and the demonstrated results have shown the reliability of the proposed method in capturing counterfeit coins [90].

7.2 FUTURE WORK

This section discusses possible refinements and improvements to the proposed methods, and also suggests some further research for future work.

The preliminary work for the proposed methods focused on scaling the coin to fit the whole coin image under the assumption that the coin is circular. This assumption holds for the vast majority of modern coins, yet there are a few modern and many ancient coins that are of different shapes. A few studies have been discussed this aspect in the literature, but no standard solution was adapted by other researchers. Further studies can be conducted on coins of non-circular shapes to precisely locate the actual coin border edges and to scale the coin.

The proposed method of character segmentation from coins shows a considerable decrease in accuracy when applied to coins whose characters are connected to other stamp edges. The projection profiles find the height and width of each character based on the peaks and valleys to demonstrate the character strokes and spaces between them. However, characters attached to other stamp edges show barely visible valleys between characters which makes it hard to decide the actual size of the character. Further studies can be proposed on deciding the actual size of characters by tuning the projection profiles parameters or by refining the character strokes before applying the projection profiles.

The proposed dynamic-HOG is argued in this thesis to work on other structured objects that have uniform heights and widths and not limited to characters on coins. Therefore, a further evaluation on structured objects other than characters can be carried out to recognize these objects at a similar accuracy to the one reported in this thesis.

In this thesis, a fine-tuning process to transfer learning of existing CNN has been studied. CNN is an evolvable research direction and several architectures are proposed for different purposes. Studying and developing a new CNN architecture that has less computational cost yet efficient in classifying genuine and counterfeit coin could also be a promising research direction.

Finding a counterfeit coin dataset is the most challenging task in counterfeit coin detection. The proposed methods can also be further analyzed to develop a non-reference counterfeit coin detection. The method can start by studying a non-reference coin recognition, then extended to find more precise features for counterfeit detection.

Finally, there are multiple sources for coin forgery, as stated by several governmental reports, which include the number of different forge sources. These reports established the number of sources of forgery based on the variations between counterfeit coins themselves. Therefore, identifying the source of forgery is still a concern, yet no work has been performed and no solution has been developed to address this issue. Investigating this research direction would be the first to study and propose a solution to identify the sources of different forgeries.

REFERENCES

[1] C. Suetonius, The lives of the twelve Caesars, Penguin Classics, London, England, 2003.

[2] N. T. Elkins, "A survey of the material and intellectual consequences of trading in undocumented ancient coins: a case study on the North American trade," *Frankfurter elektronische Rundschau zur Altertumskunde,* vol. 7, pp. 1-13, 2008.

[3] C. R. Gagg and P. R. Lewis, "Counterfeit coin of the realm–Review and case study analysis," *Engineering Failure Analysis,* vol. 14, no. 6, pp. 1144-1152, 2007.

[4] J. Robitaille, "Counterfeit coins cause for concern among Canadian collectors,," Canadian Coin News, 31 March 2015. [Online]. Available: http://canadiancoinnews.com/counterfeit-coins-cause-concern-among-canadian-collectors/. [Accessed 10 January 2018].

[5] The European Technical and Scientific Centre , "The protection of euro coins in 2016," European Commission, 2016.

[6] S. Kim, S. H. Lee and Y. M. Ro, "Image-based coin recognition using rotation-invariant region binary patterns based on gradient magnitudes," *Journal of Visual Communication and Image Representation,* vol. 32, pp. 217-223, 2015.

[7] S. Zambanini and M. Kampel, "Automatic coin classification by image matching," in *Proceedings of the 12th International Symposium on Virtual Reality, Archaeology and Cultural Heritage VAST*, Prato, Italy, 2011.

[8] M. A. Qureshi and M. Deriche, "A bibliography of pixel-based blind image forgery detection techniques," *Signal Processing: Image Communication,* vol. 39, pp. 46-74, 2015.

[9] A. Kavelar, S. Zambanini and M. Kampel, "Reading ancient coin legends: object recognition vs. ocr," in *Proceedings of the 37th Annual Workshop of the Austrian Association for Pattern Recognition (OAGM/AAPR)*, Innsbruck, Austria, 2013.

[10] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, San Diego, USA, pp. 886-893, 2005.

[11] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich, "Going deeper with convolutions," in *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, USA, pp. 1-9, 2015.

[12] A. Vadivelan and K. Praveen, "Implementation of coin recognition by extracting the local texture features with Gabor wavelet and LBP operator," in *Proceedings of the International Conference on Electrical Engineering and Computer Sciences (EECS)*, Hong Kong, pp. 135-139, 2014.

[13] L. Shen, S. Jia, Z. Ji and W.-S. Chen, "Extracting local texture features for image-based coin recognition," *IET Image Processing,* vol. 5, no. 5, pp. 394-401, 2011.

[14] D. Pawade, P. Chaudhari and H. Sonkambale, "Comparative study of different paper currency and coin currency recognition method," *International Journal of Computer Applications,* vol. 66, no. 23, pp. 26-31, 2013.

[15] A. Kavelar, S. Zambanini and M. Kampel, "Word detection applied to images of ancient roman coins," in *Proceedings of the 18th International Conference on Virtual Systems and Multimedia (VSMM)*, Milan, Italy, pp. 577-580, 2012.

[16] P. Atighehchian, "Coin wear estimation and automatic coin grading," Masters thesis, Concordia University, Montreal, 2017.

[17] R. Bassett, ""Machine assisted visual grading of rare collectibles over the Internet," in *Proceedings of the CSIS Student Research Day*, New York, USA, pp. 1-12, 2003.

[18] R. Bassett, P. Gallivan, X. Gao, E. Heinen and A. Sakalaspur, "Development of an automated coin grader: a progress report," in *Proceedings of the 8th Annual Mid-Atlantic Student Workshop on Programming Languages and Systems*, New York, USA, pp. 15.1-15.10, 2002.

[19] K. Sun, B.-Y. Feng, P. Atighechian, S. Levesque, B. Sinnott and C. Y. Suen, "Detection of counterfeit coins based on shape and letterings features," in *Proceedings of the 28th ISCA International Conference on Computer Applications in Industry and Engineering*, San Diego, USA, pp. 165-170, 2015.

[20] S. Khazaee, M. Sharifi Rad and C. Y. Suen, "Detection of counterfeit coins based on modeling and restoration of 3d images," in *Proceedings of the 5th International Symposium Computational Modeling of Objects Presented in Images: Fundamentals, Methods, and Applications*, Niagara Falls, USA, pp. 178-193, 2016.

[21] L. Liu, Y. Lu and C. Y. Suen, "An image-based approach to detection of fake coins," *IEEE Transactions on Information Forensics and Security,* vol. 12, no. 5, pp. 1227-1239, 2017.

[22] L. J. P. van der Maaten and P. J. Poon, "Coin-o-matic: A fast system for reliable coin classification," in *Proceedings of the Muscle CIS Coin Competition Workshop*, Berlin, Germany, pp. 7-18, 2006.

[23] S. Zambanini, A. Kavelar and M. Kampel, "Classifying ancient coins by local feature matching and pairwise geometric consistency evaluation," in *Proceedings of the 22nd International Conference on Pattern Recognition (ICPR)*, Stockholm, Sweden, pp. 3032-3037, 2014.

[24] R. Huber, H. Ramoser, K. Mayer, H. Penz and M. Rubik, "Classification of coins using an eigenspace approach," *Pattern Recognition Letters,* vol. 26, pp. 61–75, 2005.

[25] J. -P. Wang, Y. C. Jheng, G. M. Huang and J. H. Chien, "Artificial neural network approach to authentication of coins by vision-based minimization," *Machine Vision and Applications,* vol. 22, no. 1, pp. 87-98, 2011.

[26] Paul Davidsson, "Coin classification using a novel technique for learning characteristic decision trees by controlling the degree of generalization," in *Proceedings of the 9th International Conference on Industrial & Engineering Applications of Artificial Intelligence & Expert Systems*, pp. 403-412, 1997.

[27] M. Zaharieva, R. Huber-Mörk, M. Nölle and M. Kampel, "On ancient coin classification," in *Proceedings of the International Conference on Virtual Reality, Archaeology and Intelligent Cultural Heritage*, Aire-la-Ville, Switzerland, pp. 55-62, 2007.

[28] M. Reisert, O. Ronneberger and H. Burkhardt, "An efficient gradient based registration technique for coin recognition," in *Proceedings of the Muscle CIS Coin Competition Workshop*, Berlin, Germany, pp. 19-31, 2006.

[29] M. Zaharieva, M. Kampel and S. Zambanini, "Image based recognition of ancient coins," in *Proceedings of International Conference on Computer Analysis of Images and Patterns*, Springer Berlin Heidelberg, pp. 547-554, 2007.

[30] X. Pan, K. Puritat and L. Tougne, "A new coin segmentation and graph-based identification method for numismatic application," in *Proceedings of the International Symposium on Visual Computing: Advances in Visual Computing*, Las Vegas, USA, pp. 185-195, 2014.

[31] M. Nölle, H. Penz, M. Rubik, K. Mayer, I. Holländer and R. Granec, "Dagobert-a new coin recognition and sorting system," in *Proceedings of the 7th Internation Conference on Digital Image Computing-Techniques and Applications (DICTA'03)*, Syndney, Australia, pp. 329-338, 2003.

[32] M. Kampel and M. Zaharieva, "Recognizing ancient coins based on local features," in *Proceedings of the International Symposium on Visual Computing*, Las Vegas, USA, pp. 11-22, 2008.

[33] M. Kampel, R. Huber-Mork and M. Zaharieva, "Image-based retrieval and identification of ancient coins," *IEEE Intelligent Systems,* vol. 24, no. 2, pp. 26-34, 2009.

[34] H. Anwar, S. Zambanini and M. Kampel, "Supporting ancient coin classification by image-based reverse side symbol recognition," in *Proceedings of the International Computer Analysis of Images and Patterns*, York, UK, pp. 17-25, 2013.

[35] David G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision,* vol. 60, no. 2, pp. 91-110, 2004.

[36] S. Zambanini and M. Kampel, "Coarse-to-fine correspondence search for classifying ancient coins," in *Proceedings of the ACCV Workshop*, Daejeon, Korea, pp. 25-36, 2012.

[37] R. Huber-Mörk, S. Zambanini, M. Zaharieva and M. Kampel, "Identification of ancient coins based on fusion of shape and local features," *Machine Vision and Applications,* vol. 22, no. 6, pp. 983-994, 2011.

[38] H. Bay, T. Tuytelaars and L. V. Gool, "Speeded-up robust features (SURF)," *Computer Vision and Image Understanding,* vol. 110, no. 3, pp. 346-359, 2008.

[39] S. Belongie, J. Malik and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 24, pp. 509-522, 2002.

[40] C. Xu, "Research of coin recognition based on bayesian network classifier," *Advances in information Sciences and Service Sciences(AISS) ,* vol. 4, no. 18, pp. 395-402, 2012.

[41] L. Shen, S. Jia, Z. Ji and W.-S. Chen, "Statictics of Gabor features for coin recognition," in *Proceedings of the International Workshop on Imaging Systems and Techniques (IST)*, Shenzhen, China, pp. 295-298, 2009.

[42] L. van der Maaten and E. Postma, "Towards automatic coin classification," in *Proceedings of the EVA-Vienna*, Vienna, pp. 19-26, 2006.

[43] N. Jain and N. Jain, "Coin recognition using circular Hough transform," *International Journal of Electronics Communication and Computer Technology (IJECCT),* vol. 2, no. 3, pp. 101-104, 2012.

[44] R. Allahverdi, A. Bastanfard and D. Akbarzadeh, "Sasanian coins classification using discrete cosine transform," in *Proceedings of the 16th CSI International Symposium on Artificial Intelligence and Signal Processing (AISP)*, Shiraz, Iran, pp. 278-282, 2012.

[45] A. Khashman, B. Sekeroglu and K. Dimililer, "Rotated coin recognition using neural networks," *Analysis and Design of Intelligent Systems using Soft Computing Techniques,* vol. 41, pp. 290-297, 2007.

[46] N. Capece, U. Erra and A. V. Ciliberto, "Implementation of a coin recognition system for mobile devices with deep learning," in *Proceedings of the 12th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, Naples, Italy, pp. 186-192, 2016.

[47] L. Chen, S. Wang, W. Fan, J. Sun and S. Naoi, "Beyond human recognition: A CNN-based framework for handwritten character recognition," in *Proceedings of 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, Kuala Lumpur, Malaysia, pp. 695-699, 2015.

[48] B. P. Chacko, V. V. Krishnan, G. Raju and P. B. Anto, "Handwritten character recognition using wavelet energy and extreme learning machine," *International Journal of Machine Learning and Cybernetics,* vol. 3, no. 2, pp. 149–161, 2012.

[49] Y. M. Alginahi, "A survey on Arabic character segmentation," *International Journal on Document Analysis and Recognition (IJDAR),* vol. 16, no. 2, pp. 105–126, 2013.

[50] S. Zambanini, A. Kavelar and M. Kampel, "Improving ancient roman coin classification by fusing exemplar-based classification and legend recognition," in *Proceedings of the International Conference on Image Analysis and Processing*, Naples, Italy, pp. 149-158, 2013.

[51] X. Pan and L. Tougne, "Topology-based character recognition method for coin date detection," *International Journal of Computer, Electrical, Automation, Control and Information Engineering,* vol. 10, no. 10, pp. 1752 - 1757., 2016.

[52] R. G. Casey and E. Lecolinet, "A survey of methods and strategies in character segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 18, no. 7, pp. 690-706, 1996.

[53] C. Mancas-Thillou and B. Gosselin, "Character segmentation-by-recognition using log-Gabor filters," in *Proceedings of the 18th International Conference on Pattern Recognition (ICPR)*, Hong Kong, China, pp. 901-904, 2006.

[54] S. Zambanini, M. Kampel and M. Schlapke, "On the use of computer vision for numismatic research," in *Proceedings of the 9th International Symposium on Virtual Reality, Archaeology and Cultural Heritage*, Braga, Portugal, pp. 17-24, 2008.

[55] T. E. de Campos, B. R. Babu and M. Varma, "Character recognition in natural images," in *Proceedings of the International Conference on Computer Vision Theory and Applications*, Lisbon, Portugal, pp. 273–280, 2009.

[56] A. Nikolaidis and C. Strouthopoulos, "Robust text extraction in mixed-type binary documents," in *Proceedings of the 10th IEEE Workshop on Multimedia Signal Processing*, Queensland, Australia, pp. 393-398, 2008.

[57] B. Roy and R. K. Chatterjee, "Historical handwritten document image segmentation using morphology," in *Proceedings of National Conference on Emerging Trends in Computing and Communication*, Kolkata, India, pp. 123-131, 2014.

[58] J. Sun, Y. Hotta, K. Fujimoto, Y. Katsuyama and S. Naoi, "Grayscale feature combination in recognition based segmentation for degraded text string recognition," in *Proceedings of the First International Workshop on Camera-Based Document Analysis and Recognition*, Seoul, Korea, pp. 39-44, 2005.

[59] A. M. Elgammal and M. A. Ismail, "A graph-based segmentation and feature extraction framework for Arabic text recognition," in *Proceedings of the Sixth International Conference on Document Analysis and Recognition*, Seattle, USA, pp. 622-626, 2001.

[60] B. Verma, "A contour character extraction approach in conjunction with a neural confidence fusion technique for the segmentation of handwriting recognition," in *Proceedings of the 9th International Conference on Neural Information Processing (ICONIP'OZ)*, Singapore, pp. 2459-2463, 2002.

[61] M. Khayyat, L. Lam, C. Y. Suen, F. Yin and C.-L. Liu, "Arabic handwritten text line extraction by applying adaptive mask to morphological dilation," in *Proceedings of the 10th IAPR International Workshop on Document Analysis Systems (DAS)*, Montreal, Canada, pp. 100-104, 2012.

[62] O. Arandjelović, "Reading ancient coins: automatically identifying denarii using obverse legend seeded retrieval," in *Proceedings of the European Conference on Computer Vision*, Firenze, Italy, pp. 317-330, 2012.

[63] A. Gavrijaseva, O. Martens and R. Land, "Acoustic Spectrum Analysis of Genuine and Counterfeit Euro Coins," *Elektronika Ir Elektrotechnika,* vol. 21, no. 3, pp. 54-57, 2015.

[64] J. D. Snider, E. M. Zoladz and C. S. Carmine, "Acoustic coin sensor". US Patent US9279716B2, 08 March 2016.

[65] A. Wischnath, "Method of testing the validity of coins in a coin". German patent 102004038153, 25 August 2006.

[66] Mototsugu Suzuki, "Development of a simple and non-destructive examination for counterfeit coins using acoustic characteristics," *Forensic Science International,* vol. 177, no. 1, pp. e5-e8, 2008.

[67] I. Sandu, C. Marutoiu, I. G. Sandu, A. Alexandru and A. V. Sandu, "Authentication of old bronze coins I. Study on archaeological patina," *Acta Universitatis Cibiniensis Seria F, Chemia,* vol. 9, no. 1, pp. 39-53, 2006.

[68] A. Denker, J. Opitz-Coutureau, M. Griesser, R. Denk and H. Winter, "Non-destructive analysis of coins using high-energy PIXE," *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms,* vol. 226, no. 1, pp. 163-171, 2004.

[69] M. Tresanchez, T. Pallejà, M. Teixidó and J. Palacín, "Using the Optical Mouse Sensor as a two-Euro Counterfeit Coin Detector," *Sensors,* vol. 9, no. 9, pp. 7083–7096, 2009.

[70] W. K. Pratt, Digital Image Processing, John Wiley and Sons, Inc., New York, USA, 1991.

[71] I. Sobel and G. Feldma, "A 3x3 isotropic gradient operator for image processing," *a talk at the Stanford Artificial Project in,* pp. 271-272, 1968.

[72] O. Déniz, G. Bueno, J. Salido and F. De la Torre, "Face recognition using histograms of oriented gradients," *Pattern Recognition Letters,* vol. 32, pp. 1598–1603, 2011.

[73] K. Fukunaga, Introduction to statistical pattern recognition (2nd edition), Academic Press Professional, San Diego, USA,1990.

[74] I. T. Jolliffe, Principal component analysis (2nd edition), Springer-Verlag, New York, USA, 2002.

[75] X. He and P. Niyogi, "Locality peserving projections," in *Proceedings of the Advances in Neural Information Processing Systems*, Vancouver, Canada, pp. 153-160, 2003.

[76] V. N. Vapnik, Statistical Learning Theory, John Wiley and Sons, New York, USA, 1998.

[77] R. K. Eichelberger and V. S. Sheng, "Does one-against-all or one-against-one improve the performance of multiclass classifications?," in *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence*, Bellevue, USA, pp. 1609–1610 2013.

[78] Y. LeCun, Y. Bengio and G. Hinton, "Deep Learning," *Nature,* vol. 521, no. 7553, pp. 436-444, 2015.

[79] P.-E. Danielsson, "Euclidean distance mapping," *Computer Graphics and Image Processing,* vol. 14, no. 3, pp. 227-248, 1980.

[80] B. Epshtein, E. Ofek and Y. Wexler, "Detecting text in natural scenes with stroke width transform," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, San Francisco, USA, pp. 2963-2970, 2010.

[81] S. C. Wong, A. Gatt and V. Stamatescu, "Understanding data augmentation for classification: when to warp?," in *Proceedings of the International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, Gold Coast, QLD, Australia, pp. 1-6, 2016.

[82] L. Perez and J. Wang, "The effectiveness of data augmentation in image classification using deep learning," in *arXiv preprint arXiv:1712.04621*, pp. 1-8, 2017.

[83] Ali K. Hmood, Z. M. Kasirun, H. A. Jalab, G. M. Alam, A. A. Zidan and B. B. Zaidan, "On the accuracy of hiding information metrics: counterfeit protection for education and important certificates," *International Journal of Physical Scineces,* vol. 5, no. 7, pp. 1054-1062, 2010.

[84] Z. Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing,* vol. 13, no. 4, pp. 600-612, 2004.

[85] Ali K. Hmood, T. V. Dittimi and C. Y. Suen, "Counterfeit coin detection using stamp features and convolutional neural network," in *Proceedings of the International Conference on Pattern Recognition and Artificial Intelligence*, Montreal, Canada, pp. 273-279, 2018.

[86] Ali K. Hmood, T. V. Dittimi and C. Y. Suen, "Scale and rotation invariant character segmentation from coins," in *Proceedings of the 14th International Conference on Image Analysis and Recognition (ICIAR)*, LNCS, Springer, Montreal, Canada, pp. 153-162, 2017.

[87] Ali K. Hmood, C. Y. Suen and L. Lam, "Dynamic-HOG descriptor for structured object recognition: case study on coins," in *In proceedings of the 6th International Workshop on Image Mining – Mathematical Theory and Applications (IMTA)*, Montreal, Canada, pp. 688 – 694, 2018.

[88] Ali K. Hmood, C. Y. Suen and L. Lam, "An enhanced histogram of oriented gradient descriptor for numismatic applications," *Pattern Recognition and Image Analysis, Springer,* vol. 28, no. 4, pp. 569–587, 2018.

[89] Ali K. Hmood and C. Y. Suen, "An ensemble of character features and fine-tuned convolutional neural network for spurious coin detection," in *a chapter in Language Processing, Pattern Recognition, and Intelligent Systems*, World Scientific Publishing, 2018, In Press.

[90] Ali K. Hmood and C. Y. Suen, "Selection of edge-based features for counterfeit coin detection,". *Submitted,* 2018.