LOCATION IDENTIFICATION IN MECCA USING DEEP NEURAL NETWORKS

A THESIS SUBMITTED TO THE GRADUATE
SCHOOL OF APPLIED SCIENCES OF
NEAR EAST UNIVERSITY

BY
MOHAMMED ABDULGHANI TAHA

In Partial Fulfillment of the Requirements for the
Degree of Doctor of Philosophy
in
Computer Engineering

NICOSIA, 2021

MOHAMMED
ABDULGHANI
TAHA

LOCATION IDENTIFICATION IN MECCA
USING DEEP NEURAL NETWORKS

NEU
2021

LOCATION IDENTIFICATION IN MECCA USING DEEP NEURAL NETWORKS

A THESIS SUBMITTED TO THE GRADUATE
SCHOOL OF APPLIED SCIENCES OF
NEAR EAST UNIVERSITY

BY
MOHAMMED ABDULGHANI TAHA

In Partial Fulfillment of the Requirements for the
Degree of Doctor of Philosophy
in
Computer Engineering

NICOSIA, 2021

MOHAMMED ABDULGHANI TAHA: LOCATION IDENTIFICATION IN MECCA USING DEEP NEURAL NETWORKS

Approval of Director of Graduate School of Applied sciences

Prof. Dr. K. Hüsnü Can BAŞER

We certify this thesis is satisfactory for the award of the degree of Doctor of Philosophy in Computer Engineering

Examining committee in charge:

Prof. Dr. Rahib Abiyev	**Committee chairman, Department of Computer Engineering, NEU**
Assoc. Prof. Dr Kamil Dimililer	**Committee member, Department of Automative Engineering, NEU**
Assist. Prof. Dr. Elbrus Imanov	**Committee member, Department of Computer Engineering, NEU**
Assist. Prof. Dr. Kamil Yurtkan	**Committee member, Department of Computer Engineering, CIU**
Assoc. Prof. Dr. Melike Şah Direkoğlu	**Supervisor, Department of Computer Engineering, NEU**
Assist. Prof. Dr. Cem Direkoğlu	**Co-Supervisor, Department of Electrical and Electronics Engineering, METU-NCC**

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Muhammed Abdulghani, Taha

Signature:

Date:

ACKNOWLEDGMENTS

I would like to thank all people who supported me in my PhD journey and I would like to give a special thanks and prayers to my wife Merve Sumeyye Taha and to my son Badrsami Taha for their continuous support, and never forget the extreme support and fully guidance from my supervisor Assoc. Prof. Dr. Melike Şah Direkoğlu and co-supervisor Assist. Prof. Cem Direkoğlu for their efforts in helping me to write this thesis in the right way. I would like to express my heartfelt appreciation to Prof. Dr. Rahib Abiyev for his insightful and constructive suggestions throughout the planning and development of the study PhD program.

ABSTRACT

A total of nearly 2,000,000 Muslim pilgrims from all over the world flying to Mecca on a regular basis in order to achieve the Hajj worship. In glow of the irregular increase in the count of travelers, the believers from all over the world believe that it will be challenging for them to identify their location and that they may be misdirected as a result of the communication barriers that currently exist. In order to assist and follow each pilgrim in recognizing their own position, the researchers have proposed different methods. However, despite the fact that there are some researches on person localization, it still remains a challenging research problem. In this thesis, a novel method based on digital image processing and deep learning, is introduced to develop a platform to differentiate hotspot regions in Mecca. A new digital image processing algorithm is used as a pre-processing stage to enhance geographic information present in the image, and then a convolutional neural network (CNN) is employed for place recognition and localization. An extensive evaluation is conducted on a large dataset containing images of different regions in Mecca. Results show that the proposed pre-processing algorithm increases the accuracy, as well as when combined with the CNN for classification, it achieves the best localization performance.

Keywords: Artificial Intelligence; Digital Image Processing, Deep Learning; Convolutional Neural Network; Place Recognition, Person Localization.

ÖZET

Dünyanın her yerinden yaklaşık 2.000.000 Müslüman, Hac ibadetini tamamlamak için sürekli olarak Mekke'yi ziyaret eder. Hacıların sayısındaki düzensiz artış nedeniyle, dünyanın her yerinden inananlar, konumlarını tanımanın zor olduğuna ve mevcut iletişim engelleri nedeniyle yanlış yönlendirilebileceklerine ihtimal veriyorlar. Araştırmacılar, her hacıya kendi konumlarını tanımalarında yardımcı olmak ve onları takip etmek için farklı yöntemler önerdiler. Ancak, kişi lokalizasyonu ile ilgili bazı araştırmalar olmasına rağmen, hala zorlu bir araştırma problemi olmaya devam etmektedir. Bu tezde, Mekke'deki önemli bölgelerini ayırt etmek için bir platform geliştirmek için dijital görüntü işleme ve derin öğrenmeye dayalı yeni bir yöntem tanıtılmaktadır. Görüntüde bulunan coğrafi bilgiyi geliştirmek için bir ön işleme aşaması olarak yeni bir dijital görüntü işleme algoritması kullanılır ve ardından yer tanıma ve yerelleştirme için bir evrişimli sinir ağı (CNN) kullanılır. Mekke'deki farklı bölgelerin görüntülerini içeren büyük bir veri seti üzerinde kapsamlı bir değerlendirme yapılmıştır. Sonuçlar, önerilen ön işleme algoritmasının doğruluğu artırdığını ve ayrıca sınıflandırma için CNN ile birleştirildiğinde en iyi yerelleştirme performansını elde ettiğini göstermektedir.

Anahtar Kelimeler : Yapay zeka; Sayısal Görüntü İşleme, Derin Öğrenme; Evrişimli Sinir Ağı; Yer Tanıma, Kişi Yerelleştirme.

TABLE OF CONTENTS

ACKNOWLEDGMENTS iv
ABSTRACT v
ÖZET vi
LIST OF TABLES x
LIST OF FIGURES xi
LIST OF ABBREVIATIONS xiv
Chapter 1 INTRODUCTION 1
1.1 Crowd and Scene Analysis 1
1.2 Haj Season and Important Places 3
    1.2.1 Haram 4
    1.2.2 Mina 5
    1.2.3 Arafat 6
    1.2.4 Muzdalifa 7
1.3 Aims and Objectives of the Study 7
1.4 Motivation of the Study 8
1.5 The Significance of Identifying a Location in Mecca 8
1.6 Limitations of the Study 10
1.7 Problem Statement 10
1.8 Methodology 10
1.9 Study Dataset 11
1.10 Overview of the Thesis 11
1.11 Publications Related to This Study 12
Chapter 2 LITERATURE REVIEW OF PLACE RECOGNITION APPROCHES 13
2.1 Sparse Feature-based Place Recognition 13
2.2 Sequence-based Place Recognition 13
2.3 Text-based Localization and Place Recognition 15
2.4 Pure Image Retrieval 18
2.5 Topological Maps 19
2.6 Deep Learning-based Methods 19
Chapter 3 deep neural networks 27
3.1 Artificial Intelligence 27

3.2 Machine Learning 29
3.2.1 Machine Learning Types 30
  3.2.1.1 Supervised learning 30
  3.2.1.2 Unsupervised learning 31
  3.2.1.3 Reinforcement learning 31
  3.2.1.4 Semi-supervised learning 32
3.3 Artificial Neural Networks (ANN) 33
3.3.1 The Evolution of Neural Networks 34
3.3.2 Constituents of Neural Networks 35
3.3.3 Applications of ANNs 37
3.4 Support Vector Machines (SVM) 38
3.4.1 Applications of SVM 39
3.5 Deep Neural Networks 40
3.6 Deep Learning Architectures 41
3.6.1 Supervised Deep Learning Architectures 42
3.6.2 Unsupervised Deep Learning Architectures 43
3.6.3 Trends in Deep Transfer Learning 45
3.7 Convolutional Neural Networks (CNNs) 46
3.7.1 Convolutional Layer 47
3.7.2 Pooling Layer 48
3.7.3 Fully connected Layer 49
3.7.4 Activation Function 50
3.7.5 Softmax Layer 51
3.7.6 Optimization 52
Chapter 4 PROPOSED PLACE IDENTIFICATION METHOD USING COMBINED IMAGE PRE-PROCESSING AND CNN 54
4.1 Proposed Architecture 54
4.2 Image Pre-processing 55
4.2.1 Median Filter 56
4.2.2 Multithreshold Color Segmentation 60
  4.2.2.1 Segmentation based on Image Features and Functionality 61
  4.2.2.2 Thresholding 62
  4.2.2.3 Thresholding in this Study 65
4.2.3 Feature Extraction by GLCM 68
4.2.4 Feature Selection by Rough Set Approach 72

4.3 Proposed Algorithms for Feature Extraction and Classification 74
4.4 CNN Architecture 75
4.5 SVM Architecture 77
4.5.1 Kernel trick 77
4.5.2 Multi class SVM 78
4.6 ANN Architecture 79
Chapter 5 Evaluations 81
5.1 Dataset and Experimental Setup 81
5.2 Performance Metrics 85
5.2.1 Precision , Recall and Specificity 86
5.2.2 F-measure and G-mean 87
5.3 Results With Image Pre-Processing 88
5.4 Results Without Image Pre-Processing 94
5.5 Discussion of Results With and Without Image Pre-Processing 95
5.6 Comparison with Other Deep Learning based Method for Place Identification 95
Chapter 6 CONCLUSIONS AND FUTURE WORK 97
REFERENCES 98
APPENDICES 108
APPENDIX I 108
APPENDIX II 117
APPENDIX III 125
ETHICAL APROVAL DOCUMENT 125
APPENDIX V 127

LIST OF TABLES

Table 5.1: Implementation time ……..………………………………………….….. 81
Table 5.2: Results with pre-processing …………………………………………….. 90
Table 5.3: Results without preprocessing ……………………………………..…. 90
Table 5.4: Comparing results with and without pre-processing……………………… 91
Table 5.5: The proposed work is compared with a deep learning ………………….… 92

LIST OF FIGURES

Figure 1.1: Route of Haj 4
Figure 1.2: Masjid Alharam 5
Figure 1.3: Staying in Mina.. 6
Figure 1.4: Mountain of Arafat 6
Figure 1.5: Staying in Muzdalifa 7
Figure 2.1: Due to a sequence of images to query (black trajectory), the method can locate the robot of various lengths, shape and visual appearance across multiple (colored) reference sequences …………………………………………….. 14
Figure 2.2: Standard sample techniques: (a) sequence-based VPR;, (b) position recognition utilizing TextPlace; ……………………………………...… 15
Figure 2.3: Place scene text recognition.. 16
Figure 2.4: Localization using scene image texts: textual manipulation Data from the nearby shops helps us to estimate accurately the Camera place (green star)……………………………………………………………..……... 17
Figure 2.5: Results of detection of some challenging text from images 18
Figure 2.6: Descriptor-based SAES place the recognition... 20
Figure 2.7: Place Recognition Using CNN 21
Figure 2.8: Fusion-based Place Recognition 22
Figure 2.9: With a street view image query, (Lin et al., 2015) aims to find out where it was taken by comparing it to an aerial view image database on a city scale…… 23
Figure 2.10: NetVLAD layer with the CNN architecture 24
Figure 2.11: Examples of similarity maps between a query image 's local attributes, and top-ten match SPoC descriptors……………………………………………. 25
Figure 2.12: Multi-Process Fusion: Pick the best methods for processing images ……... 26
Figure 3.1: AI, Machine Learning and Deep Learning 28
Figure 3.2: Artificial Neural Network 36
Figure 3.3: Transmission transfer via dendrite inputs to axon terminal outputs in a neuron and myelinated axon……………………………………………………….. 37
Figure 3.4: Difference of ANN and DNN (Mostafa et al., 2020) 40

Figure 3.5: Performance ratio of Deep Learning and Machine Learning 41
Figure 3.6: Deep Learning Architectures (Madhavan & Jones, 2017) 41
Figure 3.7: In RNN, the structure of a single recurrent unit is shown 43
Figure 3.8: The encoder and decoder sections of a simple autoencoder are shown in this diagram 45
Figure 3.9: Schematic representation of CNN 47
Figure 3.10: 2-D Cross-correlation of CNN 48
Figure 3.11: Demonstration of Max Pooling operation in 2D space 49
Figure 3.12: Fully connected 50
Figure 4.1: General Architecture of proposed system 55
Figure 4.2: Flow of image pre-processing 55
Figure 4.3: a) The X-ray image of a computer chip with salt-and-pepper distortion has been distorted. (b) A 3 × 3 averages filter is used to reduce distortion (c) A 3 × 3 median filter is used to reduce distortion 57
Figure 4.4: Element opening and source 58
Figure 4.5: Operation of median filter 59
Figure 4.6: Results of Threshold selections 63
Figure 4.7: Two thresholds 64
Figure 4.8: Various methods for different-color thresholding 66
Figure 4.9: Flowchart of proposed image pre-processing work. 68
Figure 4.10: Flow of image pre-processing in detail 74
Figure 4.11: Proposed image classification strategy 75
Figure 4.12: Proposed CNN architecture 76
Figure 4.13: Block diagram of CNN flow 76
Figure 4.14: Block diagram of SVM 78
Figure 4.15: Block diagram of ANN 80
Figure 5.1: Recognition Datasets for Hajj and Umrah Pilgrim Events 82
Figure 5.2: videos during 2012-2019 four important places in Mecca 83
Figure 5.3: Distribution of the proposed dataset 83
Figure 5.4: Confusion matrix 86
Figure 5.5: General architecture with pre-processing 89
Figure 5.6: Specificity of the methods 89

Figure 5.7: Recall of the methods 90

Figure 5.8: Precision of the methods... 91

Figure 5.9: Gmean of the methods 91

Figure 5.10: F-measure of the methods... 92

Figure 5.11: Accuracy of the methods 93

Figure 5.12: General architecture without preprocessing.. 94

LIST OF ABBREVIATIONS

CNN :	Convolutional Neural Network
ANN :	Artificial Neural Network
SVM :	Support Vector Machine
GLCM :	co-occurring grayscale
A.I :	Artificial Intelligence
HUER :	Haj Umrah Event Recognition
VPR :	Visual Place Recognition
ML:	Machine Learning
NN:	Neural Network
DNN :	Deep Neural Network
RNN:	Recurrent Neural Network
FNN :	Feedback Neural Network
SOM :	Self-Organized Map
ENT :	Entropy
RBM :	Restricted Bolzmann Machines
ReLU	rectified linear activation function
T :	Thresholding
RGB :	Red Green Blue
NLP :	Natural Language Processing
CONTR :	Contrast
DAPSO :	Dynamic Adaptive Particle Swarm Optimization
CORREL :	Correlation
AUC :	Area Under Curve
RBF	Rdial Bsis Fnction
ECOC	kernel error Correcting Output Code

CHAPTER 1
INTRODUCTION

1.1 Crowd and Scene Analysis

Crowd analysis and scene understanding in video surveillance (Wang et al., 2007) have recently received a lot of attention. Crowded scenes can be found in films, Television shows, video recording archives, social media footage, and surveillance. Several current recognition, ability to monitor, and behavior identification techniques, which are only applicable in sparse settings, do not work well in crowded scenarios due to the massive amount of participants gathered with recurrent and powerful occlusions. As a result, several new studies have been conducted in recent years, particularly those focusing on crowded scenes. Crowd monitoring, crowd counting, pedestrian traveling time prediction, crowd feature identification, crowd behavior interpretation, and abnormality identification (Direkoglu, 2020; Direkoglu et al., 2017; Hatirnaz et al., 2020) are just a few of the topics covered.

Many current crowd monitoring studies are scene-specific, meaning that algorithms trained on one scenario could only be utilized to that scene. When moving to a new scene, data from the new scene must be acquired in order to retrain the system. It restricts the usage of these applications. Researchers have recently focused on the goal of scene independent crowd interpretation, which means that when a basic crowd method is trained, it could be used in a variety of settings without having to be retrained. Considering the intrinsically complicated crowd behaviors noticed throughout many scenes, this is not a simple task. Because there are so many different crowd scenarios, describing and comparing their dynamics is difficult (Wang & Loy, 2017).

Several researches reveal that distinct crowd systems have a set of universal qualities since various forms of crowd behaviors are based on some common concepts. Researchers presented and evaluated several standard crowd characteristics such as collectiveness, ensity, uniformity, stability and conflict from the perspective of computer vision. (Zhou et al., 2013). (F. Zhu et al., 2014) offered a more detailed list of 94 qualities to describe crowd places, themes, and activities. Recently, deep learning has had a significant success in many computer vision problems including analyzing crowds (Wang & Loy, 2017).

The accessibility of huge volumes of training data is critical to the effectiveness of deep learning algorithms. Available crowd datasets are constrained in terms of size, scene variety, and descriptions, making them unsuitable for training generic deep neural networks that may be used in a variety of scenarios. The Shanghai World Expo'10 crowd dataset (C. Zhang et al., 2016) and the WWW crowd dataset (Shao et al., 2016) were previously offered as two extensive crowd datasets.

On the other hand, Convolutional Neural Networks (CNNs) have had considerable achievements with image identification, learning dynamic attribute descriptions from videos for crowd monitoring. Although CNN is very successful in classification tasks, pre-processing of images before classification may improve the accuracy. Pre-processing is the term used to describe all changes made to data before it is fed into a machine learning or deep learning method. There are evidences that training a CNN on unprocessed images will most likely to have poor classification results as shown by (Sudeep & Pal, 2017). Pre-processing is also necessary to enhance training time (LeCun et al., 2012). A simple method of considering videos as 3D volumes and using CNN to learn features of these 3D images will not generally yield very acceptable results because of illumination, scale, orientation changes of images. Furthermore, in crowd scenes, the training method has a substantially larger computing difficulty comparing to the time required to image pre-processing. For crowd analysis, new network topologies and training methodologies must be devised such as nonparametric fine-tuning scheme (Wang & Loy, 2017).

Various festivities, such as festivals, social gatherings, live shows and sporting events, are organized on a regular basis across the world, attracting large crowds. People who gather at such events might come from many nations and cultures and have varying levels of education attitudes, and character traits. In the context of crowds, a team of individuals who have collected in one area, despite of their mother tongue or nationality, gender or occupation (Bae et al., 2018).

Yearly pilgrimage to the holy city of Mecca in Saudi Arabia, where untold numbers of people travel from all over the world to perform the ritual of Hajj (the pilgrimage). (Ilyas, 2013; Ye et al., 2019). Such large-scale gatherings of pilgrims traveling to a variety of destinations have prompted extensive efforts to investigate each and every pilgrim. In order to identify missing worshipers, to ensure the safety and assurance of believers, and to prevent any mass disasters or rushes that result in a high death toll, one of the most important aspects of the Hajj is the distinct confirmation of route zones. As a result, it is critical to figure out where devotees went missing in the past. This demand has prompted the use of cutting-edge technologies to relieve management leaders' stress.

A group of individuals who have gathered for a particular purpose or to visit a distinct activity is referred to as a "traditional" crowd. This team of people is concerned about the same thing. A large throng of pilgrims at Mina where its near to Mecca, flowing in one path and dividing a shared goal is one example (The Stoning of the Devil). The amount of time it takes a believer to throw tiny stones at the time of the ceremony has an impact on the entire group's movement; As long as one member of the group does not hurl stones at the columns, other members of the group continue to move in the direction of the columns, and new pioneers continue to join the group. The worshipers are of various ages, including youth, old, vigorous, fragile, women, and men, and move individually or in groups, according to the chiefs. This implies their speed and the portion of time they have to wrap the stone at Satan alters are not the same. These characteristics have a crucial role in obstructing pilgrims' paths and causing them to become lost. To prevent the pilgrims from going missing, a simple image processing system that does not require any large, sophisticated structure can be used, and the believers should be able to use it successfully, which is the aim of this study. It is recognized as a sacred site of images with geological information for precisely recognizing the pilgrims' position.

1.2 Haj Season and Important Places

The Hajj is considered to be the fifth pillar of Islam and is one of the most significant pilgrimages on the world. Each pilgrim makes seven counterclockwise laps around the Kaaba, then sprints back and forth between the Al-Safa and Al-Marwah mountains, drinking from the Zamzam in the process. As each pilgrim completes seven counterclockwise laps around the Kaaba, he or she then sprints back and forth between the mountains of Al-Safa and Al-Marwah, sipping water from the Zamzam spring along the way. The pilgrims then cut their hair, execute an animal sacrifice ceremony, and celebrate Eid al-Adha, a three-day global festival (see Figure **1.1** for the Hajj event and significant locations).

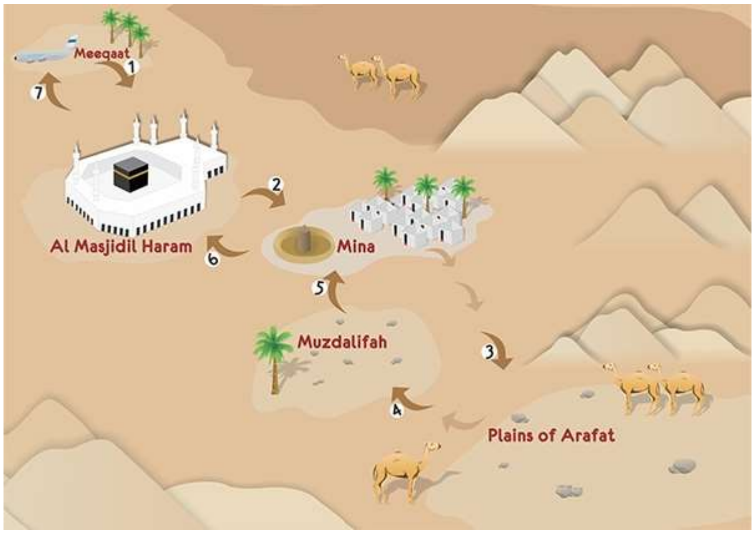


Figure 1.1: Route of Haj (Zawbaa & Aly, 2012)

As shown in Figure **1.1**, there are four hajj and ritual locations that are modelled during Hajj. These are Haram, Arafat's Mount, staying in Muzdalifah for the night, and in Mina for three nights. The following are the significant places used in this study:

1.2.1 Haram

Taking a walk 7 times counter - clockwise all around Kaaba at Mecca's Al-Masjid Al-Haram is part of the Tawaf ritual, which is performed in the holy place of Haram. When pilgrims arrive at Al-Masjid Al-arm, they either accomplish an arrival tawaf as part of their Umrah or as a welcome tawaf. Pilgrims consider Hateem, which is located on the north side of the Kaaba, on their tawaf journey. Each circuit begins with the Black Stone being kissed or touched. Pilgrims also utter a prayer while pointing to the stone. If pilgrims are unable to kiss or touch the stone due to overcrowding, they may merely point to it with their hand on each circuit. Because of the risk of dehydration, eating is not authorized, although water consumption is tolerated and encouraged (N. Mohammed, 1996).

At the area of Abraham (Muqam Ibrahim), a location within the masjid opposite the Kaaba, Tawaf is followed by two Rakaat prayers of salat. However, due to the large crowds, they could pray anywhere else in the masjid during the Hajj intervals. After pray the prayer, pilgrims drink water from the Zamzam water source well, that is kept in refrigerators throughout the Masjid.

Due to the extreme vast crowds, Tawaf the walking around the Kaba is now practiced on the 1st floor and roof of the mosque in addition to the customary rounds around the Kaaba on the floor level (N. Mohammed, 1996). See Figure **1.1**

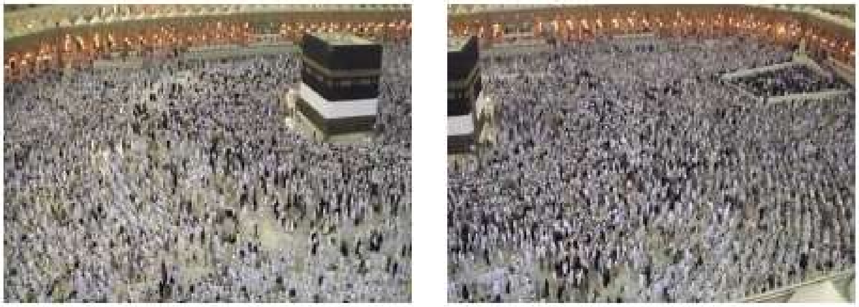


Figure 1.2: Masjid Alharam **(Zawbaa & Aly, 2012)**

1.2.2 Mina

On the 8th of Dhu al-Hijjah, after the morning prayer, pilgrims transport to Mina, where they stay for the the whole day, providing midday, noon, evening, and night prayers. During the night, they remain in their tends as shown in Figure 1.3. They leave Mina the following morning after performing morning prayers in order to move to Arafat. (Zawbaa & Aly, 2012).

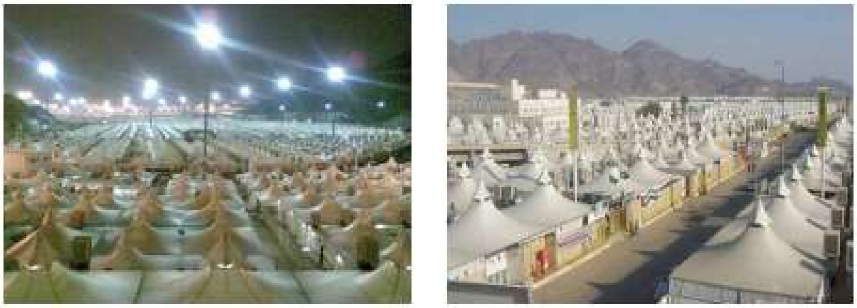


Figure 1.3: Staying in Mina **(Zawbaa & Aly, 2012)**

1.2.3 Arafat

Before noon on the 9th Dhu al-Hijjah, pilgrims arrive at Arafat, a barren and flat land 20 kilometers (12 miles) east of Mecca, where they remain in contemplative vigil, offering supplications, repenting and atoning for their sins, and seeking Grace and mercy, while listening to a sermon delivered by Islamic scholars from near Jabal al-Rahmah (The Mount of Mercy). This is referred to as "standing before God" (wuquf) as shown in Figure 1.4, and it lasts from midday until sunset. It is one of the most important rites of Hajj. Pilgrims at Masjid al-Namirah pray both midday (dhuhur) and evening (maghrib) prayers at the same time. It is regarded incorrect for a pilgrim to perform Hajj if he or she does not stay the afternoon on Arafat. (N. Mohammed, 1996).

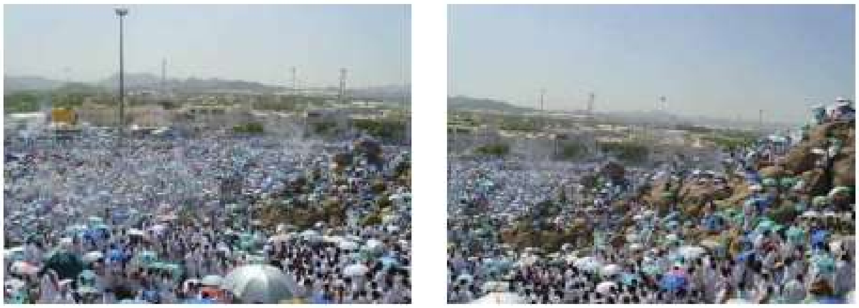


Figure 1.4: Mountain of Arafat **(Zawbaa & Aly, 2012)**

1.2.4 Muzdalifa

When leaving Arafat for Muzdalifah, pilgrims should leave before the sunset prayer has been completed, according to tradition. A area referred as Muzdalifah is located between Arafat and Mina. When pilgrims arrive, they pray Maghrib and Isha together, spend the evening and sleeping under the open sky (as depicted in Figure 1.5), and collect stones for the Devil's stoning ritual, which takes place the following day (N. Mohammed, 1996).

Figure 1.5: Staying in Muzdalifa **(Zawbaa & Aly, 2012)**

1.3 Aims and Objectives of the Study

The purpose of this research is to develop a computer vision algorithm using image pre-processing and Convolutional Neural Networks (CNNs) in order to successfully identify locations in Mecca. Image pre-processing is utilized in order to suppress unwanted distortions and/or enhancement of some important image features from images. In addition, a CNN architecture is developed which is suitable for place identification task. The proposed CNN architecture contains 9 convolutional layers and single fully connected hidden layer with a total of 2000 nodes. In order to test the proposed place identification using combined image pre-processing and CNN, a dataset is also developed for important places in Mecca. On the dataset, we used image processing techniques to remove undesired distortions and improve a few key image attributes. Denoising was conducted with median filtering, breaking the image into portions was performed with multi threshold color segmentation, feature extraction was accomplished with GLCM, and feature selection was achieved with the rough set approach. Then, the obtained pre-processed images are feed to the proposed CNN architecture for feature learning and classification. Results show that our combined image pre-processing and CNN approach achieves better classification performance comparing to classification with Artificial Neural Networks (ANNs), Support Vector Machines (SVM) and other deep learning based solutions.

1.4 Motivation of the Study

The purpose of this study is to assist worshipers in locating their precise locations so that they can complete their duties on time and in safety. Additionally, it can help to ease the operations of the government to manage the season.

The findings of the study have the potential to lead to the development of two extremely important applications. One of them is for worshipers, and the other is for the government. For example, it is possible to develop an application for worshipers that allow them to take pictures with their phones' cameras and have their location pinpointed accurately. As well as, a government-oriented application for controlling the flow of people and gathering information can be developed.

1.5 The Significance of Identifying a Location in Mecca

In most cases, the classification of a location is accomplished by a series of research. This work necessitates a number of labors that are slow, infrequent, and costly. As a result, such data is primarily accessible in a few established countries and most urban places with sufficient resources to collect and curate it; yet, such data is normally not accessible in a number of underdeveloped and emerging locales where it is the most important. The current patterns ensure that investigating urban conditions is a time-consuming and detailed challenge; even so, significant advancements in satellite technology and authority now allow for the acquisition of maximum and centered-goals symbolism of more populated areas of the globe on a daily basis. The advancements in computer vision techniques, particularly deep learning model-based image segmentation and categorization, allow for a far more accurate representation of the building system and its surroundings.

Because of the large number of worshippers in a specific geological region, such as Al Haram, Arafat, Muzdalifa, and Mina, the following problems may arise, which necessitate a solution using today's industry-accepted modern methods.

a. Medical assistance in an emergency
b. Have struggles spotting specific places such as an inn, Mina tent, supplication region, washroom, and so on.
c. When they were split from one another for any reason, they had difficulty locating their family or companions.
d. It is hard to pinpoint local hotspots in Mecca and the surrounding areas.
e. Notifying the public in the event of a crisis

By minimizing these issues, the Hajj will be a safe and appealing destination for adventurers and Hajj specialized coops. Considering the fact that various mobilized and bilingual Vanguard, Sentry, and route management squads are used and established in a scattered manner to aid and direct the explorers, those measures are insufficient to deal with the massive group situation.

Missing people, death, position recognition difficulties, lost paths, medical issues, damage, abandoning specified places, or exceeding limitations are all common occurrences during the Hajj. There are a big number of explorers who are continuously missing their path or becoming separated from their celestial ritual companions (Lu et al., 2019; S. Zhang et al., 2018).

It is proposed in this study that a novel analysis and worshipers following method be used as an initiation in order to address the difficulties and significantly reduce the stress and strain of pilgrims, their companions or group, and family members in a straightforward manner in order to provide an answer and deal with the difficulties in a straightforward manner. Also covered in this essay are the system's planning and control methods, which are known for being dependable, robust, and straightforward to use by both the general public and system administrators.

1.6 Limitations of the Study

The following technical limitations apply to this study. To begin with, there are technical restrictions, such as a lack of readily available and/or trustworthy data. Due to a lack of data or reliable data, the research was forced to limit the scope of its analysis, which posed a significant challenge in obtaining fine results. Furthermore, lack of prior research studies on the topic was another limitation. It was very hard to find good researches on Hajj and Holly Mecca in general. Another obstacle was to access suitable source code that can be used in evaluation.

1.7 Problem Statement

During the Haj season, millions of people flock to Mecca. They should visit some major destinations at different times during the Haj ritual. Because the majority of people are visiting the region for the first time, it can be difficult for them to locate the location in a timely manner.

Our research aims to use image pre-processing and deep neural network assists pilgrims in locating the four holy sites in Mecca, such as Haram, Mina, Arafat, and Muzdalifa.

1.8 Methodology

Using a combination of image pre-processing and deep neural networks, this study recognizes different locations in Mecca and its surrounding territories. A median filter is used to pre-process the captured image. Using a multiple threshold segmentation approach, the processed image is segmented. The images are then classified as various color spots. The gray level co-occurrence matrix (GLCM) was used to retrieve these color features, and selected features were chosen using the Rough set approach. The processed images are fed into a CNN network, which successfully detects Mecca's important places.

Finally, we compared our findings with two well-known and widely used machine learning techniques, ANN and SVM as well as compare with other deep learning methods.

1.9 Study Dataset

Our dataset is established from HEUR and updated videos and images from internet. We focused on four important places. The HEUR dataset is made up of videos and images shot throughout the Hajj and Umrah seasons of 2011 and 2012. HUER covers all aspects of the Hajj and Umrah rituals. Images have a pixel size of 1280x720 pixels and videos have a spatial resolution of 640x480 pixels, with average lengths of 20 seconds and 30 frames per second (Zawbaa & Aly, 2012).

We established our database from the combination of HEUR and and videos during 2012-2019 manually by ourself (Taha et al., 2021)

In our dataset we focused on four important places of Mecca where they are Haram, Mina, Muzdalifa and Arafat, where each of the places is illustrated in section 1.2. We used 500 images for each places. We used 350 images for train, 75 images for validation and 75 images for test.

1.10 Overview of the Thesis

The thesis includes the following chapters for place identification in Mecca using image pre-processing and deep neural networks:

Chapter 1: This chapter provides an overview of the crowd event scene as well as the impact of image pre-processing and deep learning. The importance of the Haj season and its major sites are then discussed. The problem statement, limitations, methodology, and dataset are all described.

Chapter 2: We will discuss traditional place recognition approaches (such as sparse feature, sequence-based, text-based, image retrieval techniques, and topological maps) as well as deep learning models (such as Convolutional Neural Networks and hybrid deep learning methods).

Chapter 3: This chapter will provide a general introduction to AI, as well as descriptions of machine learning and deep learning principles. Then, in detail, the CNN, ANN, and SVM algorithms will be demonstrated.

Chapter 4: The proposed method for place identification in Mecca using combined image pre-processing and CNN is discussed in this chapter. Median filtering, image segmentation, GLCM, and a rough set approach are all part of the suggested method. The architectures of CNN, SVM, and ANN systems will be explained as well.

Chapter 5: This chapter discusses evaluation and results. The accuracy, specificity, and other evaluation performances will be discussed and compared.

Chapter 6: The thesis' conclusion and future work will be demonstrated.

1.11 Publications Related to This Study

Taha, M. A., Direkoglu, M. S., & Direkoglu, C. (2021). Deep neural network-based detection of pilgrims location in Holy Makkah. *International Journal of Communication Systems*, e4792. https://doi.org/10.1002/dac.4792

Taha, M. A., Şah, M., & Direkoğlu, C. (2020). *Review of Place Recognition Approaches: Traditional and Deep Learning Methods*. International Conference on Theory and Application of Fuzzy Systems and Soft Computing (ICAFS2020), Advances in Intelligent Systems and Computing, vol 1306, 183–191, Springer, Cham. https://doi.org/10.1007/978-3-030-64058-3_22

CHAPTER 2
LITERATURE REVIEW OF PLACE RECOGNITION APPROCHES

Visual place recognition (VPR) methods are broadly classified into three types: narrow highlights, grouping, and in-depth learning methods (Taha et al., 2020) . In this section, we will discuss various place recognition tactics that employ traditional techniques (such as sparse feature, sequence-based, text-based, image retrieval methodologies, and topological maps) as well as deep learning algorithms (such as Convolutional Neural Networks and hybrid deep learning methods).

2.1 Sparse Feature-based Place Recognition

Many applications for place recognition uses the features of SIFT (Lowe, 1999), SURF (Bay et al., 2008), HOG (Dalal & Triggs, 2005), Bag of Terms, probabilistic models, and other features such as 3D-scene sparse-visual object maps (Cieslewski et al., 2016). Such techniques depend on low vitality; unpredictable in the lighting modifies and diverse scenes.

2.2 Sequence-based Place Recognition

Localisation is a central component of many other automaton navigation applications, since machines require recognizing where they are to make navigation decisions (Vysotska & Stachniss, 2019). One type of localization that is also important to the success of loop-closing within the simultaneous problem of location and mapping is the ability to identify that the robot is currently at a location reported successfully, also described as "bad" location (Vysotska & Stachniss, 2019). Image matching-based localization is an influential research area, and several methods have been suggested (Churchill & Newman, 2013). One group of concepts are based on the knowledge regarding the sequence.

Nonetheless, sequence-based methods also face the obstacle that they have to have ways for successful navigation, even though the machine has diverged from a path that had been taken previously. (Vysotska & Stachniss, 2019) is a sequence-based graphical place identification study, which locates multiple image sequences against a background. A system for visual place recognition is proposed in this article, which is prepared to deal with different seasons changes, various weather conditions and also alterations in illumination. The methodology locates the robot in a diagram, shown in several image sequences that were obtained in the past at various time points. (Vysotska & Stachniss, 2019) can also locate a machine in a map that is created from images from Google Street View. Because of the implementation of an effective hashing-based image recovery policy to find probable contests in conjunction with knowledgeable searches on a network of data associations, their method strongly locates a machine and relocates rapidly if it gets lost. See the Figure 2.1.

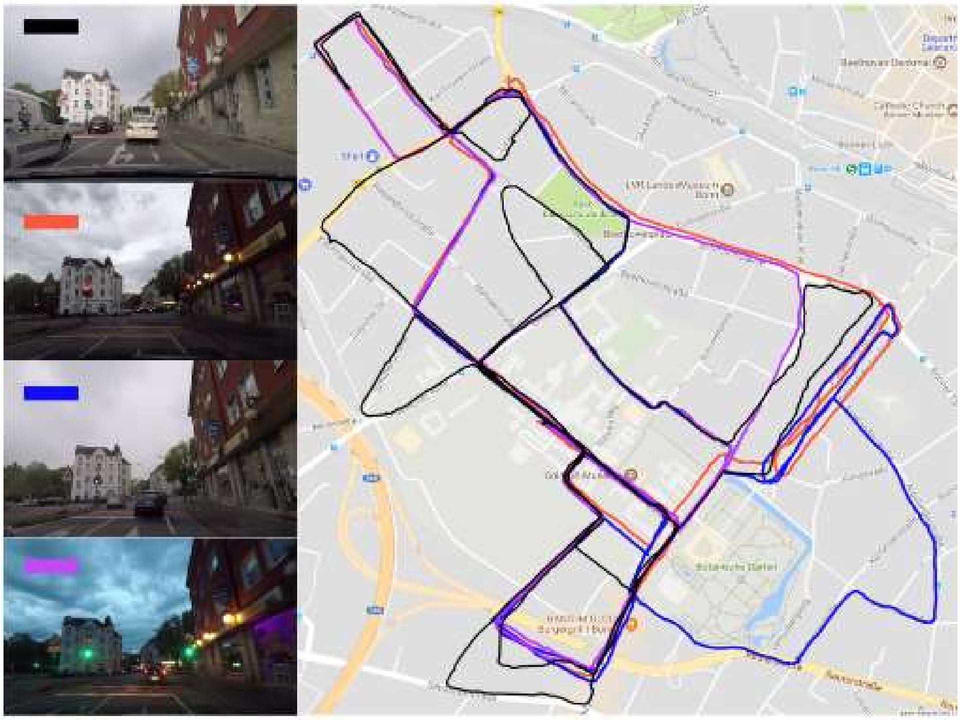
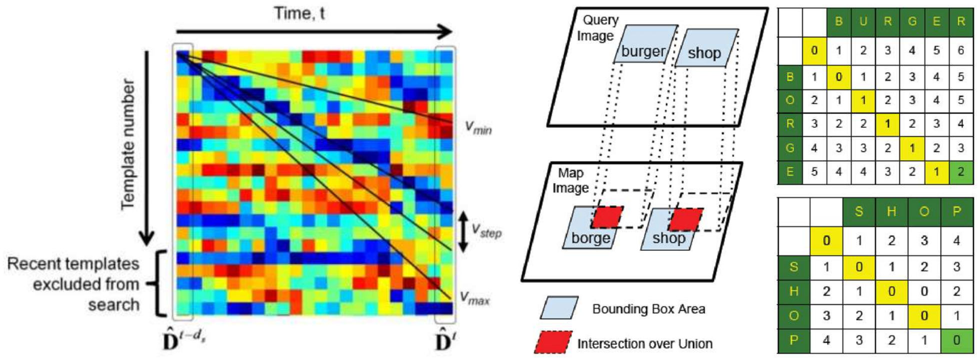


Figure 2.1: Due to a sequence of images to query (black trajectory), the method can locate the robot of various lengths, shape and visual appearance across multiple (colored) reference sequences (Vysotska and Stachniss, 2019).

In (Milford & Wyeth, 2012), a sequence-based approach for VPR is suggested, which can accommodate changing climatic situations, seasonal variation, and lighting adjustments. This technique generates a robot map by combining multiple image patterns captured at various points in period. SeqSLAM is a variant of the known robot navigation algorithm Simultaneous Localization and Mapping (SLAM). SeqSLAM generates a series of images containing the best-matching candidate position shown in Figure 2.2.



		B	U	R	G	E	R
	0	1	2	3	4	5	6
B	1	0	1	2	3	4	5
O	2	1	1	2	3	4	5
R	3	2	2	1	2	3	4
G	4	3	3	2	1	2	3
E	5	4	4	3	2	1	2

		S	H	O	P
	0	1	2	3	4
S	1	0	1	2	3
H	2	1	0	0	2
O	3	2	1	0	1
P	4	3	2	1	0

(a) (b)

Figure 2.2: Standard sample techniques: (a) sequence-based VPR; check for coherent matching sequences (Milford & Wyeth, 2012), (b) position recognition utilizing TextPlace; Levenshtein distance calculation between two query pairs and map strings (Hong et al., 2019)

2.3 Text-based Localization and Place Recognition

There are numerous works which use text-based localization to recognize the location. Texts from the scene, like traffic warnings, road markers, billboards and store road signs generally carry extensive discriminative material, they can be described as landmarks for localization (Hong et al., 2019). For robot navigation, textual information is proposed in (Wang et al. , 2015), where a text attachment method can be used to encode textual data as a landmark for the detection of the loop closure. (Radwan et al., 2016a) using numerous texts seen on a map to present a global localisation approach. (Ranganathan et al., 2013b) paired traffic signs with accurate GPS location throughout mapping and locate them by comparing the identified traffic signs to the map. The development of text detection in the wild (Jaderberg et al., 2014; Liao et al., 2018b) sets the stage the path for the use of textual information for localization and place recognition in open, complex environments.

(Hong et al., 2019) explores how to utilize high - level text content for graphical place identification and topological location. A method for visual location recognition is introduced, called TextPlace. Given the text region found in the wild, TextPlace describes places by text descriptors and builds topological maps to encapsulate the text's spatial coherence. TextPlace is the first visual place recognition method that primarily utilizes text region as descriptors to tackle position recognition in difficult scenarios such as extreme lighting shifts, perceptual aliasing, complex occlusion, and variant perspectives. Then, a whole pipeline is being built for topological metric localization on the use of high - level textual information. For place recognition, (Hong et al., 2019) makes utilisation screen texts and position topology, as illustrated in Figure 2.3.

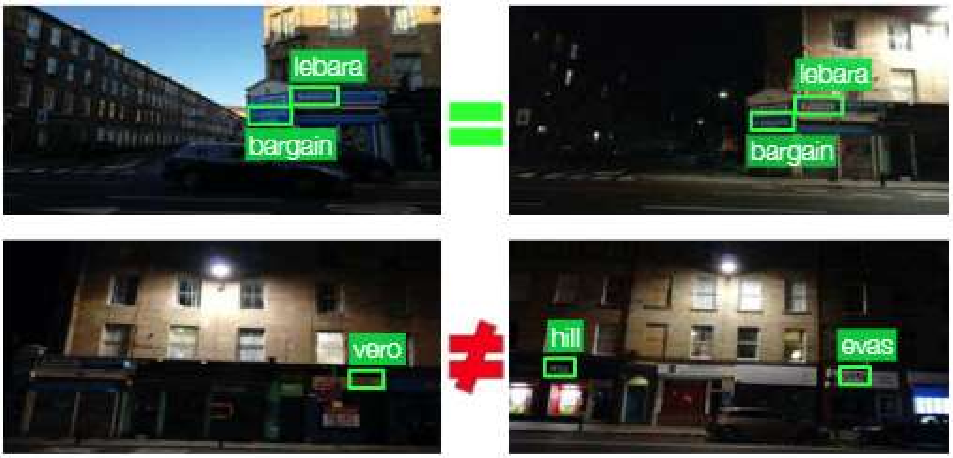


Figure 2.3: Place scene text recognition (Hong et al., 2019)

(Radwan et al., 2016a) suggests a solution that combines a standard RGB camera to detect online maps accessible to the public without using road-level images. The concept is to employ the maps' rich texts metadata data as high-level information, such as annotations from local shops and businesses. (Radwan et al., 2016a) avoids using visual - based matching characteristics in favor of using midlevel descriptions to approximate the image's existing geo-location. The emphasis is on acquiring text "in the wild" from images that are cross-referenced from the labeled map that is reachable. This allows for a new style of localization that has global range criteria, low bandwidth requirements (no images are transmitted to the network) and permanent abilities (people and companies are constantly upgrading maps). The method of Radwan et al., is broken down into three main phases as illustrated in Figure 2.4. First, a text extraction processed is applied to extract text from the images of the captured scene. The derived text will then be utilized to classify landmarks within the camera's vicinity. Eventually, a particle filter with a specific filter for a specific sensor prototype is used to get reliable assessment of position.

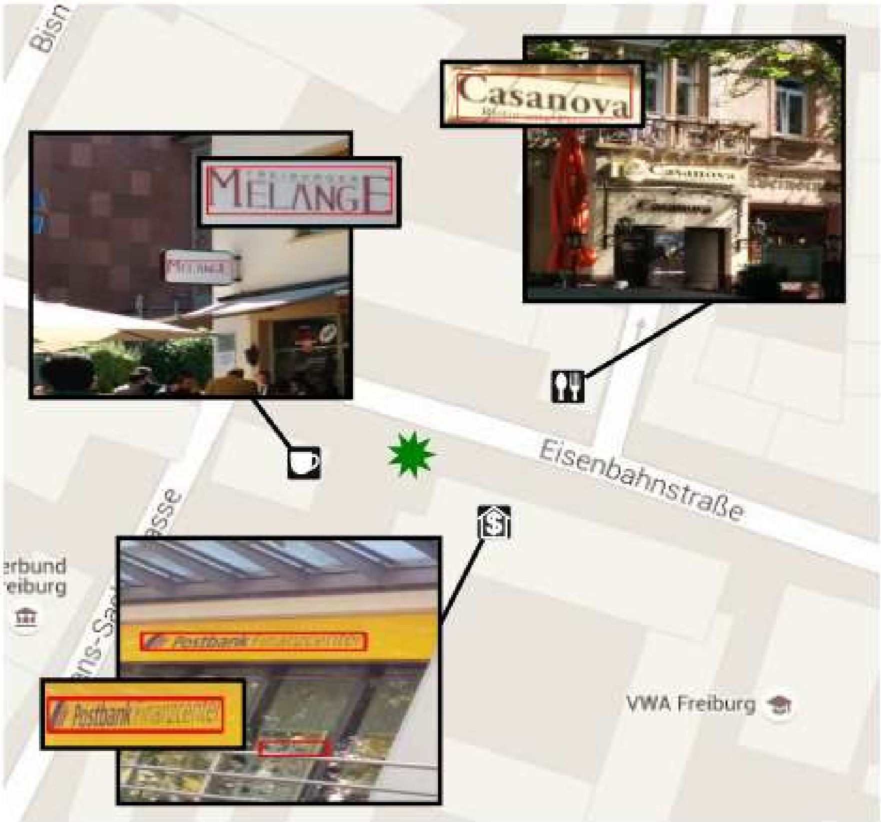


Figure 2.4: Localization using scene image texts: textual manipulation Data from the nearby shops helps us to estimate accurately the Camera place (green star) (Radwan et al., 2016).

(Ranganathan et al., 2013) locates the coordinates of GPS road signs for the purpose of identifying the current location. Recent methods, such as the (Liao et al., 2018) study, merge text identification with deep position recognition learning

In order to detect arbitrary text, (Liao et al., 2018) proposes TextBoxes++ that relies upon an end-to-end, trained, completely fully convolutional neural network. In other words, text from images is extracted using a deep learning algorithm. The basic idea is based on the proposed SSD detection algorithm. (Liao et al., 2018) propose some special designs for the adaptation of the SSD network for the efficient detection of oriented text in natural images. More specifically, (Liao et al., 2018) suggest that arbitrary text be represented by quadrilateral or oriented rectangles. Then (Liao et al., 2018) reconfigure network to estimate quadrilateral or oriented rectangle regressions from default boxes to oriented text. In order to effectively handle the text that might be intense in some areas, (Liao et al., 2018a) suggests that the default boxes be densified with vertical offsets. In addition, (Liao et al., 2018) enable kernels to manage text lines that are generally long objects better than general object tracking kernels. See Figure 2.5 for extraction of challenging text.

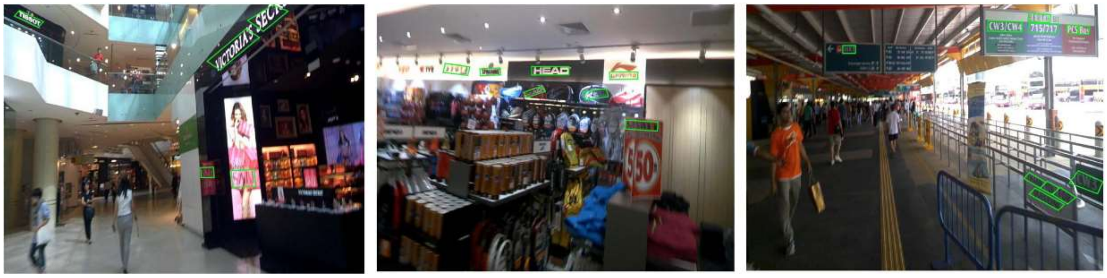


Figure 2.5: Results of detection of some challenging text from images (Liao et al., 2018)

2.4 Pure Image Retrieval

Image matching is a significant element of multiple computer vision tasks, such as object including scene recognition, 3D structure resolution, stereo interaction, and motion tracking (Lowe, 2004; Tang & Acton, 2003). In this section essential studies that use Image Retrieval principles are examined.

(Lowe, 2004) presents a method to extract characteristics from images that can be used to accurately match various views of an object or scene. The features are invariant in the size and rotation of the image and are shown to be resilient for numerous affinity variations, 3D view changes, noise introduced and light changes (Lowe, 2004).

(Tang & Acton, 2003) proposes an algorithm for retrieving images using multiple query images. (Liu & Marlet, 2012) proposes descriptors that provide imagemetric information from 1st -order to estimate the likelihood of correspondence and identify potential matches. All other information used for matching is usually limited to geometric information, i.e. relative location of the point in the image. (Liu & Marlet, 2012) the algorithm is based on intersection techniques for multiple histograms. A color histogram and a texture histogram are extracted for every query image. The intersection of multi-histogram is used to calculate the similarity of the sample images with each image in the database. The similarity classification is used to determine the images to be identified. (Liu & Marlet, 2012) using a separate peer-to- peer vocabulary tree to get a clear visual image of locations in a city dataset. A summary glossary was also used by FAB-MAP 2.0 (Cummins & Newman, 2011) to provide visually distinctive proof of a 1000 km trajectory, a probabilistic process (Glover et al., 2010).

2.5 Topological Maps

Topological techniques are attempting to decrease the number of prototypes that combine information from various regions with the accessibility of connections between them by utilizing topological techniques to do so. The neighbor's highlights and exterior appearance can be used to render these templates, for instance in TextPlace (Milford & Wyeth, 2012) and FAB-MAP (Cummins & Newman, 2011) works.

2.6 Deep Learning-based Methods

The performance of deep learning techniques in the domain of computer vision has prompted a number of preliminary investigation into their utility for visual position recognition, all using standard network features trained for other forms of recognition tasks (Chen et al., 2017a). Research works that are relied on deep learning - based method is presented in this section.

Deep learning methods have recently been used for VPR are (Lowry et al., 2016; Guo et al., 2018; Hausler et al., 2019; Kenshimov et al., 2017; Zhao et al., 2019; Zhu et al., 2018). (Zhao et al., 2019) improves feature extraction, and correlation metrics are utilised to explicitly create a network for position recognition tasks in order to handle differing appearances over time. Near the SAES metric (Zhao et al., 2019) is a convolution neural network that highlights extraction strategy. Increases potential to identify likenesses between positive and negative pairs in both positive and negative sets. Using SAES metrics, the output of frames is exceptionally pushed forward as shown in Figure 2.6.

In the experiments (Zhao et al., 2019), they test the proposed end-to-end position recognition network on Stlucia dataset (Glover et al., 2010) and Nordland dataset (Neubert et al., 2015) which are both collected to test algorithm output under apparently evolving conditions. The experiment results are stated as Area Under the Curve (AUC) and Precision-Recall(P-R) curve. The proposed approach is implemented on the basis of the Pytorch System (Paszke et al., 2019) for profound learning and the test code is open source.

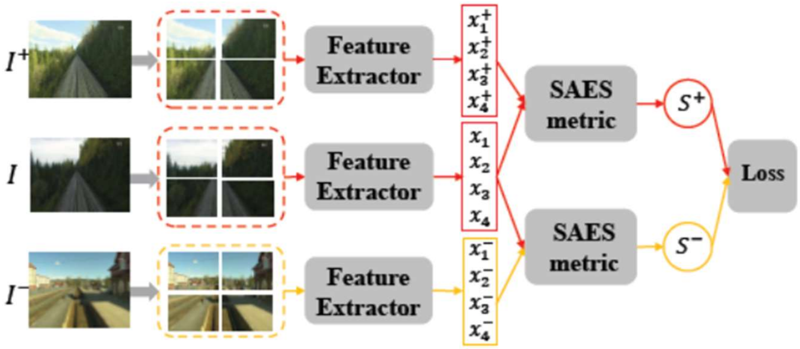


Figure 2.6: Descriptor-based SAES place the recognition (Zhao et al., 2019)

(Chen et al., 2014) present a deep learning-based location recognition algorithm that compares feature layer response from a CNN trained on ImageNet (Deng et al., 2010) and methods for filtering the corresponding location recognition hypotheses. They perform two tests, one on a 70 km benchmark location recognition dataset, and one on a point of view varying dataset, providing a quantitative comparison to two location recognition algorithms and an overview of the utility of various layers within the network for viewpoint invariance. (Gao & Zhang, 2015) offer a new approach using a modified stacked denoising auto encoder (SDA) to solve visual SLAM systems loop closure detection problem. (Bai et al., 2018) give a system that fuses AlexNet and SeqSLAM for loop closure detection.

(Zhu et al., 2018) CNN is also used; a pre-trained network model, VGG16-places365, is utilized to automatically understand and optimize image features via certain pooling, fusion, and binarization processes, and then exhibits the position recognizing similarity result with the position sequence Hamming distance shown in Figure 2.7. An algorithm is provided that determines the best candidate matching the location based on an image sequence following some of the SeqSLAM and ABLE-M ideas (Zhu et al., 2018).

In (Chen et al., 2017b), They use a multi-scale encoding method to generate context- and viewpoint-invariant features by training two CNN implementations on a large scale for the specific place recognition task. To allow this training to take place, they have created a huge Specific Places Dataset (SPED) with hundreds of examples of change of place presence at thousands of different locations, as opposed to the currently available semantic place type datasets. This new dataset allows for the development of a training regime that interprets recognition as a classification issue. They test qualified networks on many complex datasets for location recognition benchmarks.

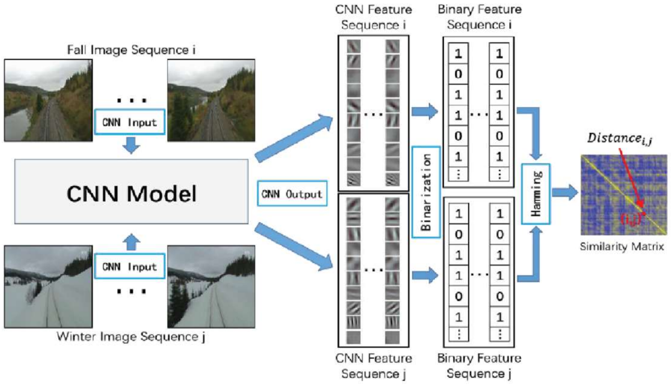


Figure 2.7: Place Recognition Using CNN (Zhu et al., 2018)

Based on deep learning, the locations (Zhou et al. , 2014) provide tremendous images of the interior and exterior scenes for training the deep attributes of scene-centered images. The Places dataset is the largest image dataset with seven million labelled images, an average of 476 categories, including numbers of scenes and locations. In addition, the most popular hierarchical and diversified database called ImageNet (Deng et al., 2010) is commonly used, that can be used for object recognition, image detection, and automatic objective clustering, etc. Because of these large repositories, ImageNet-trained deep features can also excel in various types of work (Oquab et al., 2014). Similarly, (Guo et al. , 2018) research is to pass the learning of the well-trained model based on broad Imagenet datasets, and then feed the model with a dataset to resolve the problems.

In (Guo et al., 2018), a CNN model is built in that integrates image data characteristics and mapping opportunity signals utilizing interior image databases and estimates the likelihood condition for interior location.. A feature fusion technique is built which comprised of mainly of the extraction feature module Inception V3, and the fusion and selection module function Figure 2.8. The Inception V3-based image function module is accomplished by modifying with standard images of the scene. The feature fusion and decision modules comprised of a completely layer linked and the last prediction, and the parameters are attained, and they get the parameters by means of the tests position and training set for the image.

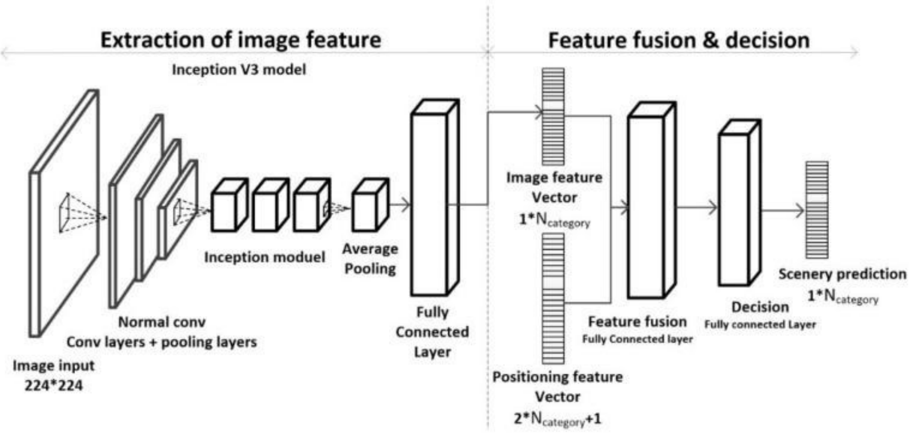

Extraction of image feature
Inception V3 model
Feature fusion & decision
Image feature Vector $1*N_{category}$
Inception moduel
Average Pooling
Fully Connected Layer
Normal conv
Conv layers + pooling layers
Image input 224*224
Positioning feature Vector $2*N_{category}+1$
Feature fusion
Fully Connected layer
Decision
Fully connected Layer
Scenery prediction $1*N_{category}$

Figure 2.8: Fusion-based Place Recognition (Guo et al., 2018)

(Kendall et al., 2015) and (Lin et al., n.d.) provide end-to-end learning for various, but related, ground-to-aerial matching tasks. (Kendall et al., 2015b) offer a six-degree real-time monocular re-location device of independence. Their platform teaches a complete CNN to minimize end-to-end detection of a 6-DOF camera from a single RGB image without requiring additional hardware or graphical optimisation. The algorithm of (Kendall et al., 2015b) runs inside and outside in real time, taking 5ms per image to predict.

Throughout this research, they use a 23-layer convolutional layer (to demonstrate that convnets is being used to fix complex issues of image plane regression. This has been made possible by exploiting the transfer of training from data on the broad classification. They (Kendall et al. , 2015b) also illustrate that the PoseNet sets up high-level features and is resistant to challenging lighting, motion blur and various cameras in which point-based SIFT registration struggles.

An even more ground-to-aerial matching (Lin et al. , n.d.), they locate a ground - level query image by comparing it to a reference aerial image database. Publicly available data is used to create a collection of 78K compatible cross-view image pairs. The critical concern for this (Lin et al., 2015) task is that the broad benchmark and appearance difference of such cross-view pairs cannot be handled by traditional computer vision approaches. They use a dataset to recognize a feature representation in which the detection views are close to each other and the misaligned views are much further apart. A conceptual solution is driven by deep learning performance in face detection and accomplishes substantial improvements over standard handcrafted features and current deep features learned from other large - scale datasets. (Lin et al., 2015) demonstrate that CNN 's effectiveness in identifying similarities among street view and aerial view images and display the potential of trained features to interpret into new locations. See Figure 2.9 for details.

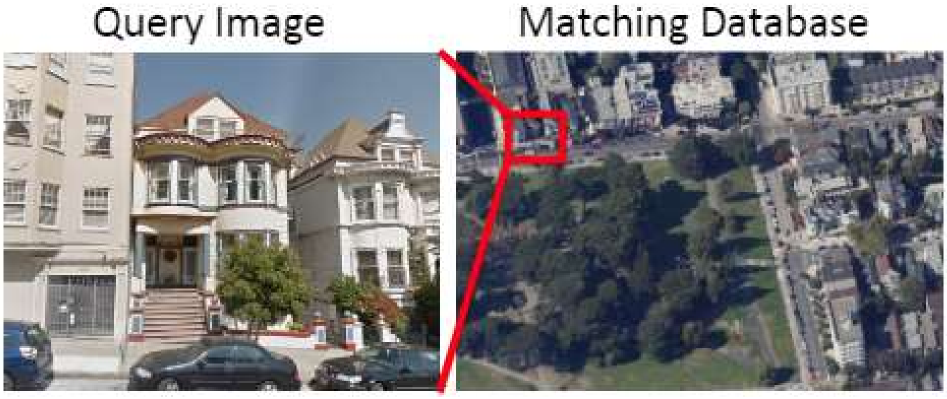


Figure 2.9: With a street view image query, (Lin et al., 2015) aims to find out where it was taken by comparing it to an aerial view image database on a city scale

Another deep learning based method based on CNN is NetVLAD (Arandjelovic et al., 2018) shown in Figure 2.10. The core element of this architecture is NetVLAD, a modified layer of the "Vector of Locally Aggregated Descriptors" (VLAD) representation of images. VLAD is commonly used for retrieving images. The layer could be conveniently interconnected into any CNN design that is well-suited to backpropagation processing (Arandjelovic et al., 2018). They establish a training method to learn end-to-end design parameters from images featuring the same places over time from Google Street View Time Machine, based on the recent badly controlled failings in the ratings. (Arandjelovic et al., 2018). The approach thus easily and accurately determines the position of a given question image.

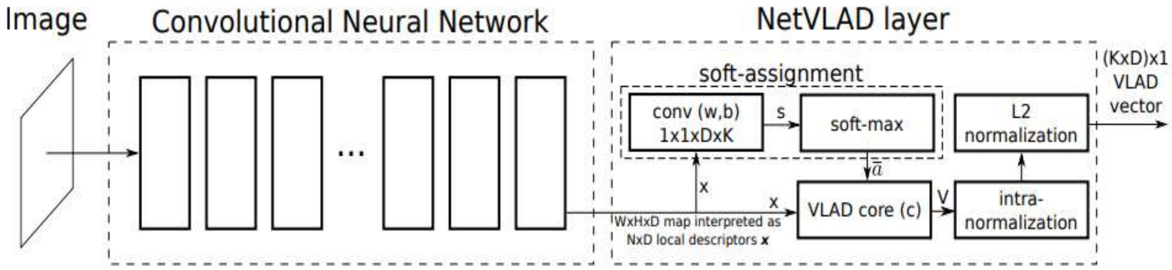


Figure 2.10: NetVLAD layer with the CNN architecture (Arandjelovic et al., 2018)

LIDAR, Sonar, RGB-Depth, and Wi-Fi are just a few of the sensors available (Jacobson et al., 2018)(Collier et al., 2013), have demonstrated the use of multiple sources of knowledge for navigation. Multiple sensors were fused using probability models (Zhang et al., 2005), standardized sensor data was concatenated by vectors (Milford & Jacobson, 2013) and standardized data was multiplied through Gaussian-distributed clusters (Jacobson et al., 2018). However, these multi-sensor techniques have the drawbacks of demanding powerful machines, and extra calibration specifications.

The idea of integrating multiple image processing processes, rather than using different sensors, has had limited work (Hausler et al., 2019). Instead of combining different image processing techniques, (Li et al., 2018) To improve performance on an image recovery challenge, the authors merged various layers of a CNN. Previous studies have used the spatial location of activations within the characteristic map space instead of implementing feature vectors based on the magnitude of CNN activations. (Babenko & Lempitsky, 2015) apply a weighting to the total pooling centered on the location of each attribute on the feature map, with the attributes closest to the system space center receiving the most weight. In another work, the combination of maximum activation co-ordinates with semantic information was used to locate across opposite points of view (Garg et al., 2018).

The investigation of possible methods of aggregating local deep characteristics in order to generate dense global characteristics for image retrieval is carried out (Babenko & Lempitsky, 2015). Firstly, they demonstrate that deep models and standard hand-engineered features have very large variations of pairwise similarities, so it is necessary to carefully assess current aggregation approaches. Such re-evaluation demonstrates that the basic aggregation procedure relies on sum pooling offers the best efficiency for deep convolutional attributes as compared to the shallow options. This method (Babenko & Lempitsky, 2015) has few parameters and carries little risk of being overfitting while learning the PCA matrix. Overall, the latest global, compact descriptor enhances four specific benchmarks. They explored many other alternative solutions for aggregating deep convolutional features into compact global definitions, and suggested a descriptor (SpoC) relying on a basic sum-pooling aggregation. Although the elements of SPoC are simple and well-known, they illustrate that the mixture of their design decisions contributes in a descriptor that delivers a huge boost over previously outlined image descriptors based on deep features shown in Figure 2.11.

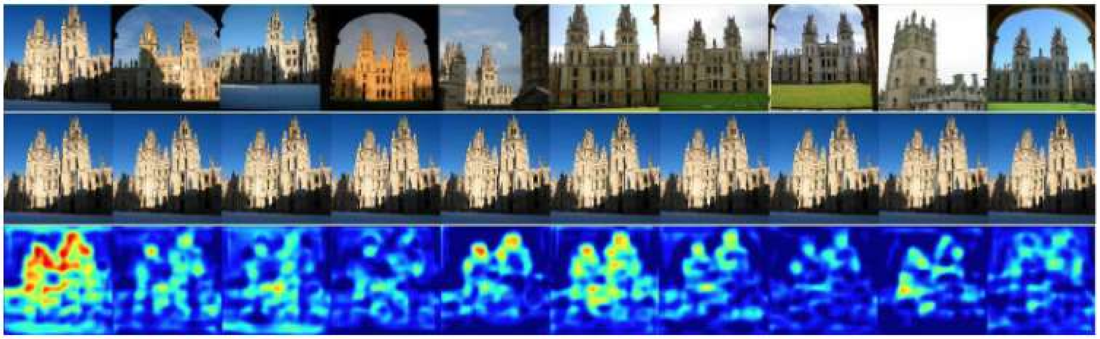


Figure 2.11: Examples of similarity maps between a query image 's local attributes, and top-ten match SPoC descriptors (Babenko & Lempitsky, 2015).

Another approach is (Hausler et al., 2019) that integrates results from several image processing approaches, such as the sum of absolute differences, oriented gradient histogram (HOG), and two deep learning features into one Hidden Markov model. This approach is called Multi-Process Fusion Figure 2.12, which automatically chooses the best environmentally safe image processing technique. During this procedure, the abilities of various image processing techniques are balanced by utilizing an automatic weighting scheme.

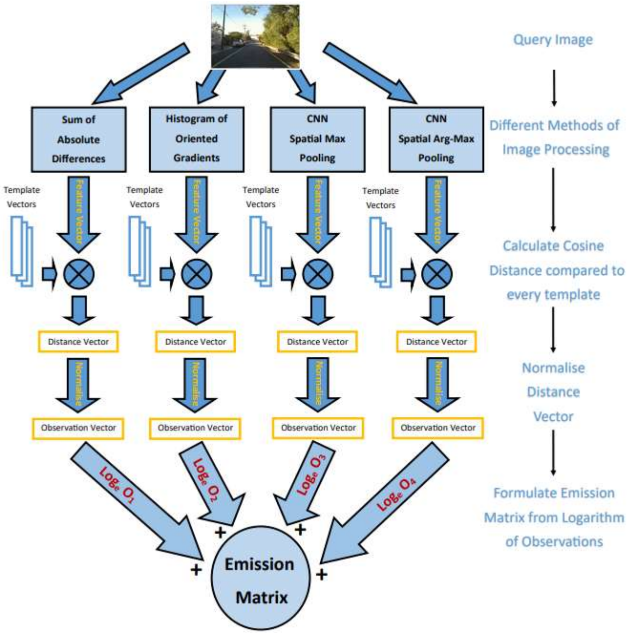


Figure 2.12: Multi-Process Fusion: Pick the best methods for processing images (Hausleret al., 2019)

CHAPTER 3

DEEP NEURAL NETWORKS

In this chapter, details of deep neural networks in particular CNNs, which is a part of this study will be discussed. Deep neural networks are sub-field of machine learning area, which is also sub-field of artificial intelligence area. Therefore, in the following sub-sections, starting from broader content of AI to narrower context of CNNs, we discuss relevant topics that are covered in this thesis.

3.1 Artificial Intelligence

Now it is the information age. Users create enormous quantities of data every day through new and ordinary innovations. As a consequence, the volume of data available has increased, especially multimedia information, is rising at an extraordinary pace. Any of this data is saved electronically and available to the community, such as Facebook members already views over 250 billion pictures and submit 350 million fresh photographs every day. Such a large number of data in our lives are a double-edged sword. On the one side, we will provide more alternatives for our inquiries if we can manage and interpret the data sufficiently. On the other hand, we can easily lose ourselves in the flood of data. Without computerized systems, the information and what we can view may take many decades. Although search engines such as Google and Yahoo are very capable of performing the text analysis well. However, the use of visual elements in search engines is still a difficult task because of the scale, orientation, illumination, resolution, orientation variations of objects.

In order to process massive amounts of data, automated processing is required, which is the aim of Artificial Intelligence (AI). Artificial Intelligence (AI) refers to the theory and creation of computer systems which normally require human ability to process data (Kurfess, 2003). At the top of the hype curve with a growing number of studies. AI has sub-fields such as machine learning, and under machine learning field, there is the more specialized field of deep learning as shown in Figure **3.1**.

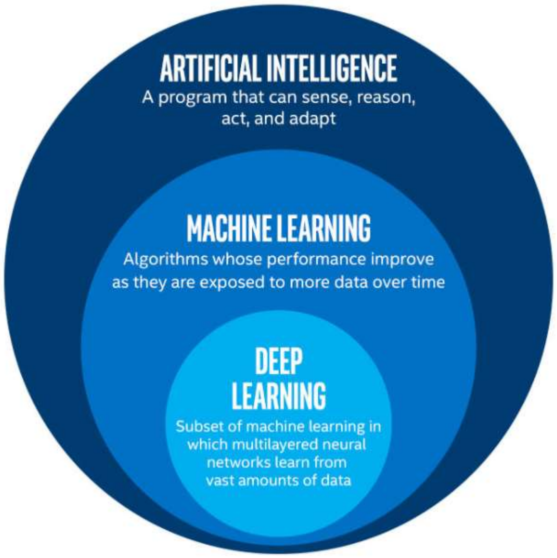

ARTIFICIAL INTELLIGENCE
A program that can sense, reason, act, and adapt
MACHINE LEARNING
Algorithms whose performance improve as they are exposed to more data over time
DEEP LEARNING
Subset of machine learning in which multilayered neural networks learn from vast amounts of data

Figure 3.1. AI, Machine Learning and Deep Learning (Srivastav, 2020)

AI considers how to develop intelligent gadgets and algorithms that could intelligently solve difficulties that are frequently viewed as human prerogatives. As a result, AI denotes a machine that mimics human action in some way (J. Russell & Norvig, 2016).

Machine learning (ML) is a subfield of AI that includes strategies for allowing machines to identify data and provide AI implementations. Machine learning is the field of study that explores how computers can improve by using data and by getting better from doing things. When using machine learning algorithms, you first collect or generate a "training set" of data that consists of multiple examples or samples to help build your model. Then, your algorithm learns or " trains " on this data to produce predictions or decisions. The use of machine learning algorithms is commonplace in all kinds of applications, including in healthcare, email filtering, speech recognition, and computer vision. In some cases, conventional algorithms are either not feasible or not preferable (Shalev-Shwartz & Ben-David, 2013).

Deep learning, also known as deep neural learning or deep neural networks, is a branch of machine learning that employs neural networks to assess a variety of characteristics using a model equivalent to that of a biological neural system. It has networks which can understand without supervision from unorganized or unidentified input (Goodfellow et al., 2016).

3.2 Machine Learning

Nowadays the technology is able to store and manipulate vast volumes of information and also reach it over a computer system from physically great distances, thanks to advancements in digital innovation. The majority of data collection equipment are now electronic and collect accurate data. As an illustration of big data collection and analysis, consider a grocery store that has hundreds of braches around the country offering thousands of products to millions of consumers. The date, customer account key, items purchased and their amounts, overall cash spent, and so forth are all recorded by the point of sale terminals. Each day, this adds to several terabytes of data. The store chain needs to be able to foresee who will buy an item in the future. The algorithm for this, once more, is unknown; it varies over time and by geographic area. Only if the data is analyzed and transformed into knowledge that we can use to make forecasts, for example, does it become useful (Alpaydin, 2020). And this is the aim of Machine Learning (ML) algorithms.

ML is the research of computing models that develop themselves over time as a result of training and information (Mitchell, 1997). It is considered to be a form of AI. ML models create a template dependent on experimental data referred to as "training data," in order to generate observations or observations and without being specifically configured (Koza et al., 1996). ML algorithms are utilized in a broad range of implementations including medicine, email filtering, and computer vision, where developing traditional methods to execute the required duties is challenging or impossible.

A branch of machine learning is strongly linked to computational mathematics which relies on achieving estimates utilizing computers. The area of machine learning benefits from the research of mathematical modeling because it provides techniques, theory, and implementation areas (Bishop, 2006).

Based on the type of the "signal" or "feedback" provided to the training program ML systems are generally categorized into four major subgroups supervised, unsupervised , reinforcement learning and hybrid (J. Russell & Norvig, 2016)

In the following sub-sections, different learning types and two important ML methods “SVM and ANN” that is used in our study will be discussed.

3.2.1 Machine Learning Types

3.2.1.1 Supervised learning

A "teacher" presents the machine with sample inputs and targeted outcomes with the purpose of learning a fundamental pattern that maps inputs to outputs. Supervised learning methods create a mathematical representation of a collection of data that includes all the sources and the outcomes that are sought. The information is referred to as training data, and it includes of a collection of training instances. Every training instance contains a single or even more sources and a supervisor signal as the expected result. Every training sample is described by an array or vector, often referred to as a feature vector, and the training data is represented by a matrix in the mathematical formula. Supervised learning techniques develop a function that may be used to predict the output associated with fresh inputs by iteratively optimizing an objective function. The algorithm will be able to accurately estimate the output for inputs that were not part of the training data if it uses an optimum function (Mohri et al., 2012).

Learning process, classification, and regression are examples of supervised learning algorithms (Mohri et al., 2012). When the outputs are constrained to a small set of data, classification techniques are employed, and regression methods are utilized when the results can contain a certain quantitative score throughout a range.

In a more abstract sense, supervised learning presents a scenario in which the "experience," a training example, has substantial information (say, the spam/not-spam labels) but is missing in the unseen "test cases" to which the gained knowledge is to be applied. In this case, the learned knowledge is used to anticipate the test data's missing information. In such instances, the environment might be thought of as an instructor who “supervises” the learner by supplying additional information (labels). However, there is no distinction between training and test data in unsupervised learning. The learner analyzes the input data with the purpose of producing a summary or compressed version of it. A classic example of such a task is clustering a data collection into subsets of comparable objects (Shalev-Shwartz & Ben-David, 2013).

3.2.1.2 Unsupervised learning

Unidentified datasets are analyzed and clustered using machine learning methods. Despite the demand for human involvement, these algorithms uncover hidden connections or data classifications

Clustering, anomaly detection, and techniques for learning latent variable models are examples of unsupervised learning algorithms. Each method employs a variety of techniques such as k-means, mixture models, DBSCAN, and OPTICS algorithms in the clustering procedure. Local Outlier Factor and Isolation Forest algorithms are two alternative techniques for anomaly detection(Mohri et al., 2012).

Unsupervised learning methods use a collection of information with only parameters and detect pattern in it, such as data point classification or clustering. As a result, the methods understand from unlabeled, unclassified, and uncategorized test data. Unsupervised learning techniques discover similarities in the data and respond depending on the existence or lack of such similarities in every new piece of information, rather than reacting to feedback. The area of density estimation in statistics, such as calculating the probability density function, is a key implementation of unsupervised learning. Unsupervised learning, on the other hand, comprises various domains that need describing and interpreting data aspects (Bishop, 2006).

3.2.1.3 Reinforcement learning

A computer algorithm communicates with complex surroundings in order to accomplish a specific task. The software is given input in the form of incentives as it directs its issue space, which it strives to optimum. (Bishop, 2006).

Reinforcement learning is a branch of machine learning that studies how program models should behave in a given context in order to increase some metric of progressive benefit. Game theory, control theory, systems studies information theory, simulation-based optimization, multi-agent systems, swarm intelligence, statistics, and genetic algorithms are among the numerous fields that study the field according to its flexibility. The surroundings are generally modeled as a Markov decision process in machine learning (MDP). Dynamic programming approaches are utilized in several reinforcement learning systems. When precise scientific models of the MDP are impossible, reinforcement learning procedures are applied. Reinforcement learning techniques are employed in automated driving and in teaching humans how to perform a game (van Otterlo & Wiering, 2012).

3.2.1.4 Semi-supervised learning

Data scientists input a limited quantity of labeled training data to an algorithm in semi-supervised learning. The algorithm then understands the data set's dimensions, which it may subsequently apply to new, unlabeled data. When algorithms are trained on labeled data sets, their performance usually enhances Labeling data, on the other hand, can be time consuming and costly. Semi-supervised learning falls somewhere between supervised and unsupervised learning in terms of performance and efficiency. Semi-supervised learning is utilized in a variety of circumstances like machine translation, fraud detection and data labeling (X. (Jerry) Zhu, 2005).

- Machine translation is the process of teaching algorithms to translate languages using a smaller set of words than a full dictionary.
- Fraud detection is the process of identifying cases of fraud when there are only a few positive examples.
- Data labeling: Algorithms trained on tiny data sets can simply add data labels to bigger ones.

Generative models, Low-density separation and Laplacian regularization are general methods used in semi-supervised learning (X. (Jerry) Zhu, 2005)

3.3 Artificial Neural Networks (ANN)

A popular ML algorithm that is also used in this thesis study is Artificial Neural Networks (ANNs). A method of programming driven by the formation of neural networks in the human brain is known as an artificial neural network. The brain, in simplistic versions, is made up of a huge amount of simple processing units (neurons) which are linked together in a complicated interaction web allowing the central nervous system to perform massively complicated calculations. ANNs are conceptual simulation structures based on this computing framework (Shalev-Shwartz & Ben-David, 2013).

In the mid-twentieth period, neural networks were suggested as a method of training. It produces an efficient training model, so it has lately been demonstrated to obtain cutting-edge efficiency on a variety of learning functions. A neural network is a directed diagram in which the nodes represent neurons and the edges represent connections between them. A measured average of the results of the neurons linked to its input edges is received as input for each neuron (Shalev-Shwartz & Ben-David, 2013).

Neural networks develop (or are trained) by interpreting instances with a defined "source" and "end," creating probability-weighted correlations among the both that are preserved inside the connection's data model. The variance within the network's interpreted outcome (often a prediction) and a goal result is normally determined when training a neural network from a provided instance. The system then changes its measured correlations utilizing this failure rate and a training law. With each modification, the neural network can generate result that is more and more close to the targeted result. The training may be terminated based on certain conditions after a reasonable amount of these changes have been made. This is called supervised learning (Shalev-Shwartz & Ben-David, 2013).

These mechanisms "train" to execute functions by looking at instances rather than being configured with undertaking instructions. ANNs are wide used in computer vision tasks. For example, for recognizing images that involves cats, by examining instance images which have been previously classified as "cat" or "no cat". Then, the method applies the findings to recognize cats in other images (Shalev-Shwartz & Ben-David, 2013).

3.3.1 The Evolution of Neural Networks

Warren McCulloch and Walter Pitts (McCulloch & Pitts, 1943) (1943) developed a mathematical framework for neural networks relying on threshold logic equations, which started the evolution of ANNs. This design allowed for the division of study into two strategies. The first strategy centered on biological mechanisms while the second focused on neural network applications of artificial intelligence This resulted to research into nerve networks and their relationship with finite automata(Kleene, 2016).

D. O. Hebb developed a training theory depending on the structure of neural brain function which was recognized as Hebbian learning, in the 1940s (Farley & Clark, 1954). Unsupervised training is referred to as Hebbian training. This led to the creation of long-term stimulation designs. Through Turing's B-type machines, scientists began extending these concepts to mathematical systems in 1948. To model a Hebbian system, Farley and Clark (Farley & Clark, 1954) (1954) utilized computing devices then known as "calculators." Rochester, Holland, Habit, and Duda developed many neural network computing devices (1956). Rosenblatt (Rochester et al., 1956) (1958) invented the input layer, a sequence detection method. Rosenblatt represented circuits not found in the simple neuron, like the special circuit, that can not be handled by neural networks at the period, using computational terminology. (Schmidhuber, 2015) Nobel laureates Hubel and Wiesel introduced a molecular paradigm in 1959 centered on their observation of two cell types in the main vision cortex basic neurons and complicated cells (Zeki, 2005). Ivakhnenko and Lapa released the Group Model of Data Processing in 1965, which was the first usable system with several levels (Schmidhuber, 2015).

Werbos' (1975) backpropagation method which facilitated realistic training of various layer nets, was a major catalyst for revived enthusiasm in neural networks and training. By changing the weights for each point, backpropagation spread the failure phrase back up across the sheets. Parallel decentralized computing was common in the 1980s by the term connectionism. The usage of connectionism to model neuronal pathways was defined by Rumelhart and McClelland (1986) (Zou et al., 2008).

3.3.2 Constituents of Neural Networks

An ANN solves challenges by combining biological concepts with sophisticated metrics. The ANNs constitute of the following:

Neurons: Artificial neurons that are conceptually developed from brain cells make up ANNs. Each synthetic neuron receives signals and generates a single result that can be transmitted to a number of different neurons. The sources may be attribute properties from a collection of outside data like images or outcomes from different neurons. The role is completed by the results of the neural network's ultimate output neurons, like identifying an item in an image (Zou et al., 2008).

To determine the neuron's result, initially the weighted number of all the input data are calculated, which is then multiplied by the quantities of the relations between the inputs and the neuron. To this number, bias expression is added. The activation is a term used to describe the weighted number. To generate the result, the weighted average is transmitted via a (usually nonlinear) activation process. Data sources, such as images and vectors, are the first inputs. The final outputs, like recognizing an item in an image complete the mission (Zou et al., 2008).

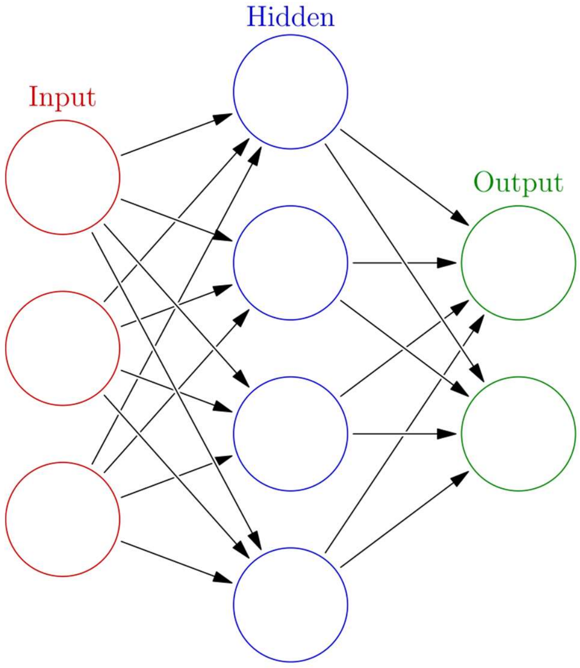


Figure 3.2: Artificial Neural Network (Zou et al., 2008)

Relations and Values: The system is made up of links, which each serves as an input to another neuron by passing the result from one node. Each relation is given a value that indicates its relative significance. Various input and output links are possible for a single neuron (Abbod et al., 2007).

Propagation method: The propagation method calculates a neuron's input as a weighted average of its prior neurons' exits and relations. To the propagation effect, a bias phrase may be applied (Dawson & Wilby, 1998).

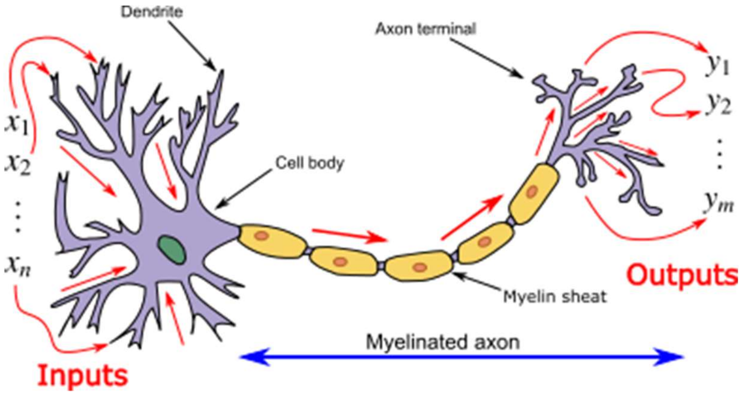


Figure 3.3: Transmission transfer via dendrite inputs to axon terminal outputs in a neuron and myelinated axon (Zou et al., 2008)

3.3.3 Applications of ANNs

ANNs have achieved utility in a variety of fields due to their capacity to predict and design nonlinear systems. System recognition and monitoring (car control, path prediction, control systems, natural product control), molecular chemistry, generalized gaming, sequence identification (gps systems, facial detection, pulse categorization, three - dimensional reconstruction, entity identification and much more), pattern recognition (gesture, voice, handwriting and typed characters, and much more) are some of the implementation fields (Mukhopadhyay, 2011). In addition, ANNs were able to identify various forms of tumors and to differentiate extremely aggressive tumor cells from less aggressive tumor types. Neural network were used to forecast base arrangements and to speed up the stability study of systems vulnerable to natural disasters (Abiodun et al., 2018).

In the geosciences, ANNs were used to construct dark models in hydrology, sea modeling and marine engineering, and geology (Abiodun et al., 2018). ANNs were used in cryptography with the aim of distinguishing among known and unknown activities. Computer science was used to recognise Android viruses recognize websites related to risk agents and identify Hyperlinks that pose a safety danger for instance (Abiodun et al., 2018). ANN devices are being researched for testing process, botnet detection, payment account theft detection, and network attack detection In physics, ANNs have indeed been suggested as a method for solving partial discrete formulas and simulating the structures of several stochastic systems (Abiodun et al., 2018).

In neural science ANNs have looked at the brief activities of young cells, how neural structure processes are derived from relationship among human neurons, and how actions can be derived from conceptual neural structures that form full components. From the specific neuron to the network level, researchers looked at the long- and short-term cognition of neural networks, as well as their relationship to cognition and training (Abiodun et al., 2018).

3.4 Support Vector Machines (SVM)

Support Vector Machines (SVMs) is another popular machine learning method that is also applied in this study. SVMs (Cortes & Vapnik, 1995) are monitored or supervised learning frameworks that process data for labeling and regression modeling in machine learning. SVMs were developed at AT&T Bell Laboratories by Vladimir Vapnik and collaborators (Boser et al., 1992, Guyon et al., 1993, Vapnik et al., 1997) and are dependent on mathematical training models or VC theory suggested by Vapnik (1982, 1995) and Chervonenkis (1974). An SVM simulation implementation creates a prototype that applies current samples to one of two classes allowing it a non-probabilistic binary linear classification model, given a collection of learning collection each labeled as corresponding to one of two subgroups. SVM transfers learning instances to places in space in order to widen the distance among the two groups as much as possible. New instances are then traced into the same space and classified according to which side of the distance they drop on (Cortes & Vapnik, 1995).

SVMs could achieve non-linear grouping as well as linear grouping by effectively translating their sources into high dimensional attribute spaces which is known as the kernel trick (Cortes & Vapnik, 1995).

If information is unlabeled, supervised learning is impossible, so an unsupervised learning method is needed wherein the data is clustered naturally into classes and additional information is mapped to these classes. The support-vector classification technique was established by Hava Siegelmann and Vladimir Vapnik to classify unlabeled data using support vector statistics improved in the support vector machines method. It is among the most commonly used grouping techniques in industrial implementations (Ben-Hur, 2008).

A support-vector machine, in more conceptual terms creates a decision boundary or set of hyperplanes in a high- or infinite-dimensional space that can be used for grouping, regression, or other purposes such as outlier detecting (Hastie et al., 2009). Conceptually, the hyperplane with the greatest gap to the closest training position of any category achieves a strong isolation because the greater the margin, the smaller the classifier's overfitting failure (Hastie et al., 2009).

3.4.1 Applications of SVM

SVMs have a wide range of implementations in a variety of fields. The following are some of the implementation areas.

In traditional analytics environments, SVMs will greatly minimize the requirement for classified training examples, making them useful in document and hyperlinks classification. With the help vector machines are used in several basic textual parsing approaches (Pradhan et al., 2004).

SVMs may also be used to achieve image categorization tasks. After only 3 to 4 cycles of validity reviews, SVMs accomplish substantially better query precision than conventional request optimization systems, according to research findings (Bannay & Guillaume, 2014). This is also valid for image clustering frameworks, especially those which use a customized form of SVM that employs Vapnik's favored method (Bannay & Guillaume, 2014).

Satellite, information such as SAR info is classified through monitored SVM (Maity, 2016). SVM can be used to identify typed letters. In biomedical and other studies, the SVM method has been commonly used as well. They have been utilized to categorize proteins, with up to 90% of the substances correctly categorized. SVM weights-based substring have been proposed as a method for SVM system analysis (Pradhan et al., 2004). In the past, support-vector system masses were also used to understand SVM versions

(Statnikov et al., 2005). A comparatively recent field of study in the biomedical fields is posthoc analysis of support-vector system models in order to classify features used by the model to make observations.

3.5 Deep Neural Networks

To solve the problem of handling large amounts data, deep learning, in other words Deep Neural Networks (DNNs) became very popular. DNNs are multi-layered Artificial Neural Networks (ANNs) that contain many hidden layers as shown in Figure 3.4.

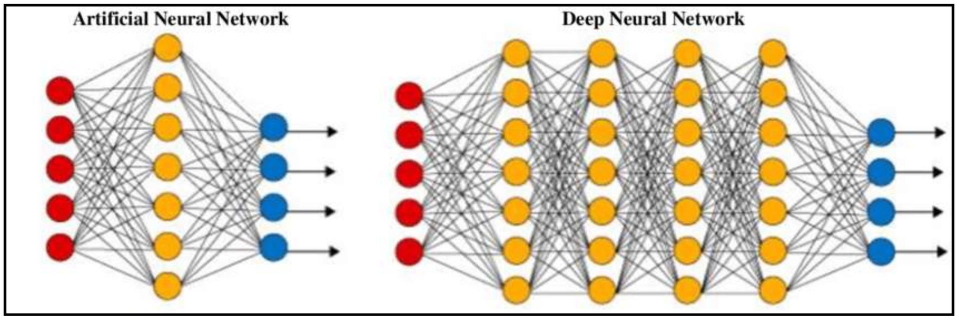


Figure 3.4: Difference of ANN and DNN (Mostafa et al., 2020)

In an ANN design, there is only one hidden layer, whereas in a DNN design, multiple hidden layers exist. DNNs have been dubbed among the best powerful methods in recent decades due to their ability to handle massive amounts of information in deeper and hidden layers with better accuracies. DNNs have recently begun to exceed traditional approaches especially in the area of pattern recognition as shown in Figure 3.5. As compared to conventional learning techniques, the success of deep learning classifiers increases significantly as the amount of data increases. When traditional simulations for machine learning encounter a particular percentage of training samples, their output stabilizes, whereas deep learning improves as the amount of data grows.

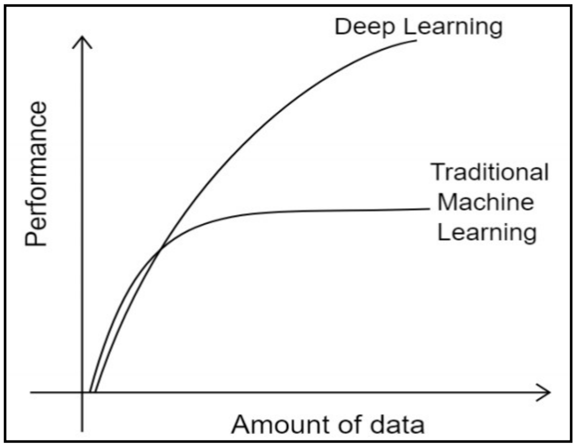


Figure 3.5: Performance ratio of Deep Learning and Machine Learning (Mathew et al., 2021)

3.6 Deep Learning Architectures

Several types of DNN architectures are proposed as a way to use AI to address traditional issues as shown in Figure 3.6. In the sub-sequent sections, these DNN architectures, in addition deep transfer learning technique are discussed briefly.

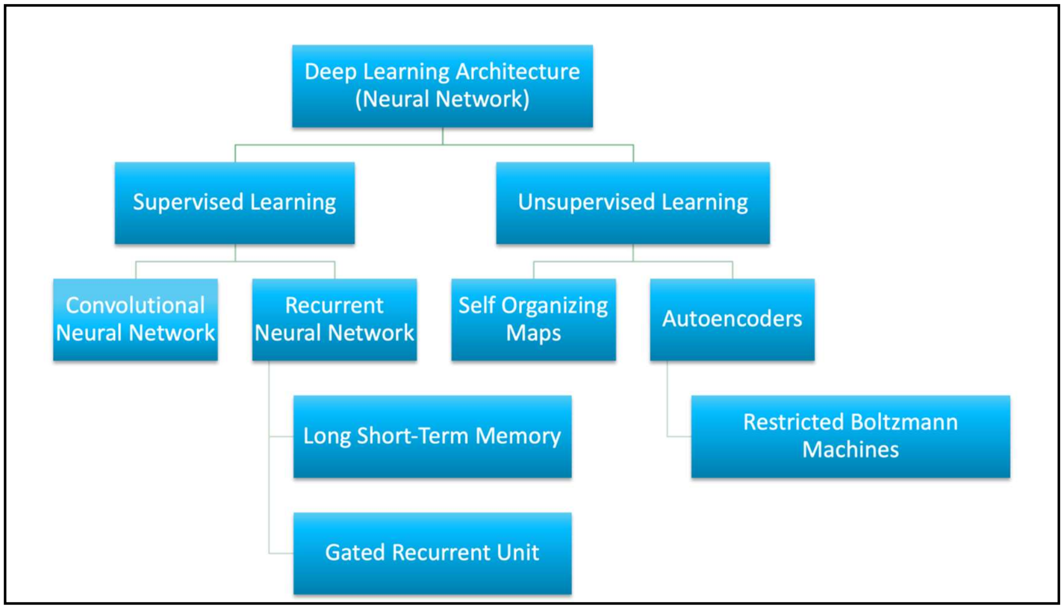


Figure 3.6: Deep Learning Architectures **(Madhavan & Jones, 2017)**

3.6.1 Supervised Deep Learning Architectures

CNNs and Recurrent Neural Networks (RNNs) are two types of the main important supervised deep learning models (Goodfellow et al., 2016) .

A CNN (or ConvNet) is a type of DNN that is generally used to interpret visual imagery in deep learning. (J. Zhu et al., 2018).

Multilayer perceptrons are supervised variants of CNNs. Multilayer perceptrons are typically completely linked networks, meaning that every neuron in one layer is linked to all neurons in the following layer. The term "convolutional neural network" refers to the network's use of the convolutional mathematical procedure. CNNs are a subtype of neural network that uses convolution rather than standard vector multiplication in at least one layer (Madhavan & Jones, 2017). In this thesis, particularly CNN is used, therefore in section 3.7, CNNs are discussed in more detail.

A RNN is a type of ANN in which nodes are connected in a directed graph that follows a temporal pattern. This enables it to behave in a temporally flexible manner. RNNs are developed from feedforward neural networks, which can interpret variable size series of inputs by using their inner state (Sherstinsky, 2018).

The name "recurrent neural network" is generally applied to two broad kinds of networks that have an identical overall design one with limited impulse and some with infinite impulse. The functionality of both types of networks is temporally variable. An infinite impulse recurrent network is a directed cyclic graph that could be unrolled and substituted with an absolutely feedforward neural network, whereas a finite impulse recurrent network is a directed acyclic graph which could be unrolled and substituted with an absolutely feedforward neural network (Djuris et al., 2013).

Further saved variables are possible in both finite and infinite impulse recurrent networks, and the memory can be controlled directly via the neural network. When another network or graph involves timing differences or has feedbacks, it can likewise be used to substitute the memory. Gated phase or gated memory refers to these regulated phases which are seen in long short-term memory networks (LSTMs) and gated recurrent units. It's also known as a Feedback Neural Network (FNN) (Sherstinsky, 2018).

Each object in the RNN has permanent links, allowing the network to store data for a greater duration. RNNs can identify features in serial data like voice, image, and script. Long short-term memory (LSTM) networks, a rather more new and evolved type of RNN, have been shown to enhance RNN sequence detection performance in especially time series type of data. (Ullah et al. 2018). See Figure 3.7.

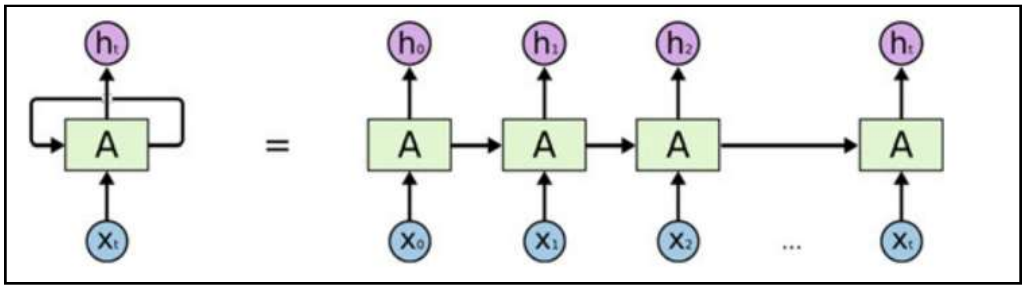


Figure 3.7: In RNN, the structure of a single recurrent unit is shown (Litjens et al., 2017).

3.6.2 Unsupervised Deep Learning Architectures

Unsupervised learning corresponds to a challenge area in which the data being utilized for training has no target labeling (J. Russell & Norvig, 2016).

Self-organized maps, autoencoders, and restricted Boltzmann machines are three unsupervised deep learning architectures discussed in this subsection.

Dr. Teuvo Kohonen established the self-organized maps (SOM) in 1982, and it was generally recognized as the Kohonen map. SOM is an unsupervised neural network that reduces the dimensionality of the feature data set to form groups. In several aspects, SOMs differ from standard artificial neural networks (Huang et al., 2019). The first notable difference is the weights are used as a node property. Following the normalization of the inputs, an unique input is picked. Every attribute of the input data is given an arbitrary weight nearly zero. The input node is now represented by these values. Many different pairings of these arbitrary values indicate different input node variants. It computes the Euclidean distance among all output nodes and the input node. The finest fitting component, or Best Match Unit BMU, is the node with the shortest length and is proclaimed to be the most precise interpretation of the input. Additional units are computed and allocated to the group that it is the distance from using these BMUs as center points. Depending on proximity, the radius of locations surrounding BMU weights is modified. The radius has been reduced. Next, no activation function is used in a SOM, and there is no notion of computing loss or back propagation since there are no goal labels to match with (Huang et al., 2019).

The exact year autoencoders were developed is unknown. LeCun discovered the first documented use of autoencoders in 1987. This type of ANN is made up of three layers: input, hidden, and output. Firstly, a suitable encoding formula is used to encode the input layer into the hidden layer. The hidden layer contains a significantly smaller set of nodes than the input layer. The condensed form of the source input is stored in this hidden layer. Using a decoder formula, the output layer attempts to recreate the input layer.

During the training period, a loss function is used to determine the gap between the input and output layers, and the weights are changed to reduce the loss Autoencoders train continuously using backward propagation, unlike standard unsupervised learning approaches in which there is no data to match the outputs with. Autoencoders are characterized as self-supervised systems as a result of this (Baldi, 2012).

There are two sections of autoencoders and deep bottleneck networks. The first section tries to compact the data to be entered into a representation that is the range is reasonably limited. The second section uses this short representation to reconstruct the original input. The autoencoder tries to extract a short summary during training that will enable for lossless reconstruction of the original data. It discovers enormously important aspects of the training outcomes unsupervised in this way. The succinct illustration of the high-dimensional input created by the autoencoder is commonly utilized as a vector of the function for clustering, indexing, and searching, as well as dimensionality reducing and attribute integrating. Figure **3.8** illustrates the encoder and decoder components of a typical autoencoder (Ahmad et al., 2019).

DBMs are probabilistic generative systems consisting of several layers of hidden, unknowns with a random distribution. The upper two layers are connected by undirected, links that are isotropic, while the lower layers are connected by top-down, guided the layer's links over them (Ahmad et al., 2019).

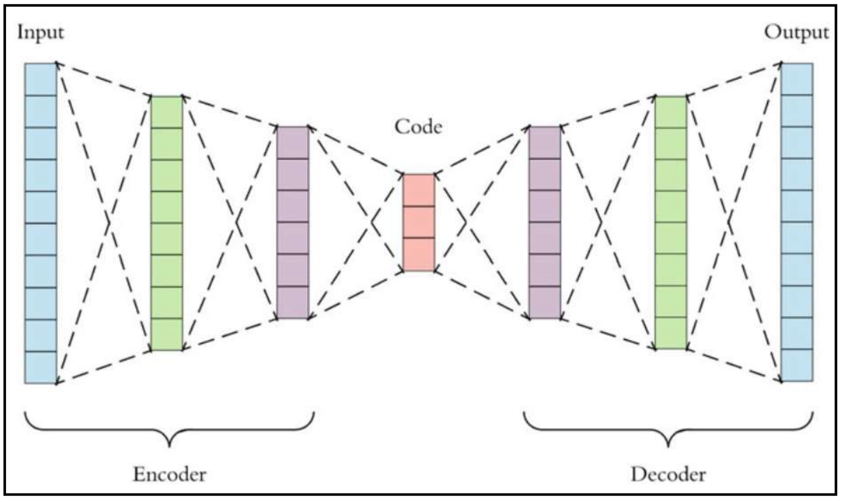


Figure 3.8: The encoder and decoder sections of a simple autoencoder are shown in this diagram (Ahmad et al., 2019).

Restricted Boltzmann Machines (RBMs) got popularized long afterwards, they were first conceived in 1986 by Paul Smolensky and were called as Harmoniums. A two-layered neural network is referred to as an RBM. Input and hidden layers are the layers. Every node in a hidden layer is linked to each node in a visible layer in RBMs. Nodes in the input and hidden layers are also coupled in a typical Boltzmann Machine. In a Restricted Boltzmann Machine, nodes within a layer are not linked because of computing sophistication (Bao et al., 2018).

3.6.3 Trends in Deep Transfer Learning

Innovative improvement is being made every year or even every week in deep learning studies . The purposes of this section, we just highlight some of the recent developments and trends in learning types (Yang, 2019).

- Self-taught learning, is an example of an algorithm to unlabeled data for the purpose of learning the representation, and then applying that learned representation to data that has been labeled. If the labeled and unlabeled data come from various distributions, then they can be various classes, and their distributions could differ as well. This definition is saying that self-taught learning is equivalent to unsupervised transfer (Yang, 2019).

- Transfer learning involves training on tasks where additional labeled data is required in order to carry over knowledge to another (but related) task. Extra labeling requirements may be costly (Yang, 2019).
- On the other hand, one-shot learning is an efficient way to learn that takes place with only one instance or a handful of instances. rather than having to learn everything from scratch, training can be improved by studying the results of prior learning, which is in general referred to as Bayesian inference Augmented and lean learning (Yang, 2019).

There are numerous architectures and software packages for ANN, CNN, and deep learning. Matlab, for example, includes the nntool, alexnet, and Googlenet (Yang, 2019).

3.7 Convolutional Neural Networks (CNNs)

The name "convolutional neural network" indicates that the network employs a mathematical operation called convolution. Convolutional networks are a specialized type of neural networks that use convolution in place of general matrix multiplication in at least one of their layers.

CNNs are generally applied for computer vision tasks such as image segmentation, categorization and object recognition in images (Abirami & Chitra, 2020). CNNs are also applied for natural language processing and optical character recognition (Abirami & Chitra, 2020). CNNs may be used to analyze sound by converting one dimension sound signal into two dimensional image data by using spectrogram function (Zaman et al., 2020). In addition, graph convolutional networks have been used to directly apply CNNs to text analytics and graph data. CNN's state-of-the-art efficiency compared to its basic methods has made it a breakthrough in a variety of industries (Abirami & Chitra, 2020).

Basic architecture of a CNN model is shown in Figure 3.9. It is made up of convolutional layer, pooling layers, fully connected layers and softmax layer (output layer). Below we discuss these layers briefly.

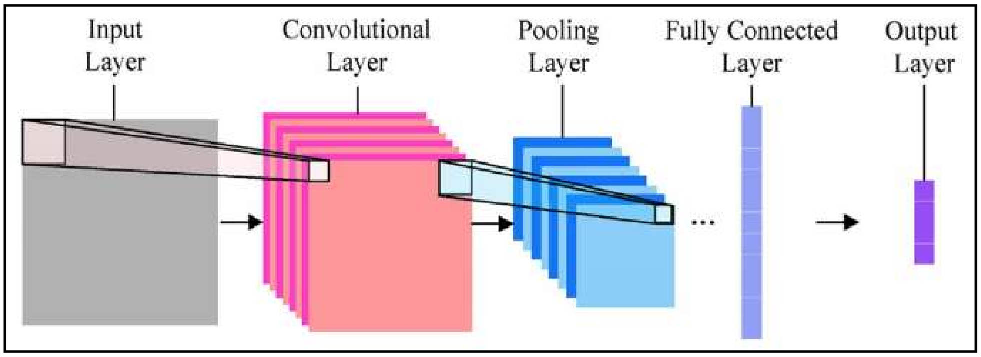


Figure 3.9: Schematic representation of CNN **(Abirami & Chitra, 2020)**

3.7.1 Convolutional Layer

The CNN's convolution layer is its brain. This layer contains filters. Filters are also called as kernels, which are used in CNN to recognize features. A filter is just a weighted matrix of data that has been taught to identify key characteristics. The convolution operation, which is an element-wise multiplication and summation among two matrices, is carried out by the filtering. By multiplying the input and filter outputs for each directed pass, the convolution layer provides either 2D or 3D filter mapping depending on the dimension of the input image. The network learns from the filter that is activated if certain properties at a specific spatial location arise as a consequence of the outcome. The ultimate output of this layer is determined by the activation map and dimensional depth of each filter.

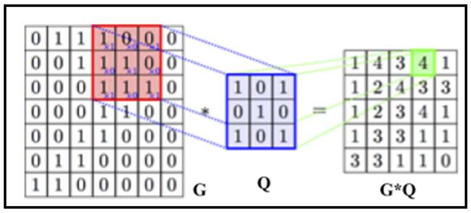
CNNs operate on a layer-by-layer basis known as convolution. Convolution, like addition, multiplication, and integration, is a mathematical process. In multiplication, two numbers are multiplied to make a third number; similarly, in convolution, two signals are multiplied to form a third signal. The discrete convolution in 2-D of two images, G and Q, is shown in the formula below:

$$F(i,j) = (G * Q)(i,j) = \sum_n G(m,n)Q(i-m, j-n) \qquad \textbf{(3.1)}$$

The sizes of the kernels are m and n. The values for each pixel are denoted by i and j. The first parameter to the convolution is generally known to as the input, and the second argument is often called to as the kernel in convolutional networks (filter). The result is known to as F, which is a feature map.

$$F(i,j) = (G * Q)(i,j) = \sum_n G(m,n)Q(i+m, j+n) \qquad \textbf{(3.2)}$$

In typically, cross-correlation, not convolution, is the process used in convolutional layers. The instance is shown in Figure 3.10. The input G is convolved with a kernel (Q), yielding a characteristic map $(G * Q)$. The quantity of feature mappings *(F)* would be determined by the quantity of kernels (filters) chosen by the creator for every convolution layer. The final layer of a convolutional network is made up of ordinary fully connected layers. Figure 3.10 depicts 2-Dcross-correlation among an input G and a kernel Q, resulting in a feature map Q. The red region that has been emphasized refers to the input's receptive field.



$$\begin{bmatrix} 0&1&1&1&0&0&0 \\ 0&0&1&1&1&0&0 \\ 0&0&0&1&1&1&0 \\ 0&0&0&1&1&0&0 \\ 0&0&1&1&0&0&0 \\ 0&1&1&0&0&0&0 \\ 1&1&0&0&0&0&0 \end{bmatrix} * \begin{bmatrix} 1&0&1 \\ 0&1&0 \\ 1&0&1 \end{bmatrix} = \begin{bmatrix} 1&4&3&4&1 \\ 1&2&4&3&3 \\ 1&2&3&4&1 \\ 1&3&3&1&1 \\ 3&3&1&1&0 \end{bmatrix}$$

Figure 3.10: 2-D Cross-correlation of CNN

3.7.2 Pooling Layer

Pooling layer is a nonlinear subsampling approach implemented by the universal nonlinear function. Max pooling, average pooling, min pooling are some examples of pooling operations. Because of its functional use, Max Pooling is the most preferred operation across them. The max pooling operation separates the input image into non-overlapping chunks, with the layers resulting in the best number in each sublayer. Figure 3.11 depicts an instance of Max Pooling, in which the operation minimizes the input data's dimension (Dumoulin et al., 2016). The pooling layer reduces the size of the middle geographical representation by not interfering with the volume of the measurement deep. Between the convolution layers, generally pooling layer is placed to decrease computational costs of training.

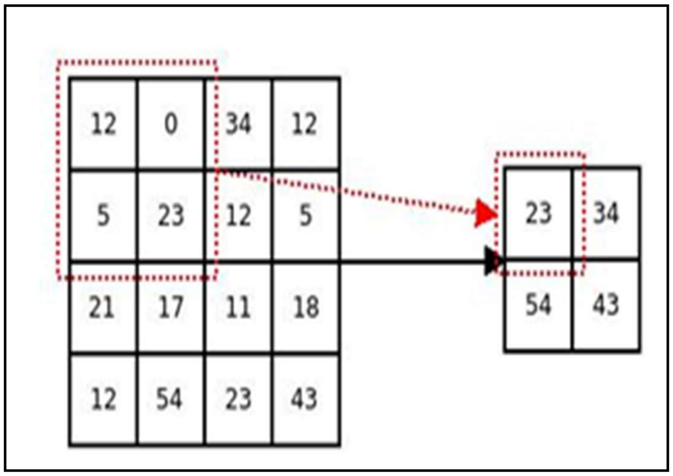


Figure 3.11: Demonstration of Max Pooling operation in 2D space

3.7.3 Fully connected Layer

The entirely linked layers in CNN is responsible for high-level reasoning. It connects the neurons in the layers above. A matrix augmentation is used to predict the operator function, and then a bias is added to equalize the 2D or 3D features and transform it to 1D quality.

In a CNN, the fully connected layers are the final layers. This completely linked layer actually provides the potential to understand the nonlinear mixture of characteristics obtained previously in the network. Each variable in this network interacts with own portion of the input. When a phrase has been successfully employed in the convolution layer, a phrase is added to estimate the end output. The depiction in Figure 3.12 is designed for the completely connected layer. The edges of the fully connected layer's architecture depict the known variables, that appear to be multiplied by the input (Wikipedia 2021). In the **Error! Reference source not found.**, the FC last layer is fully connected layer (Khiyari & Wechsler, 2016).

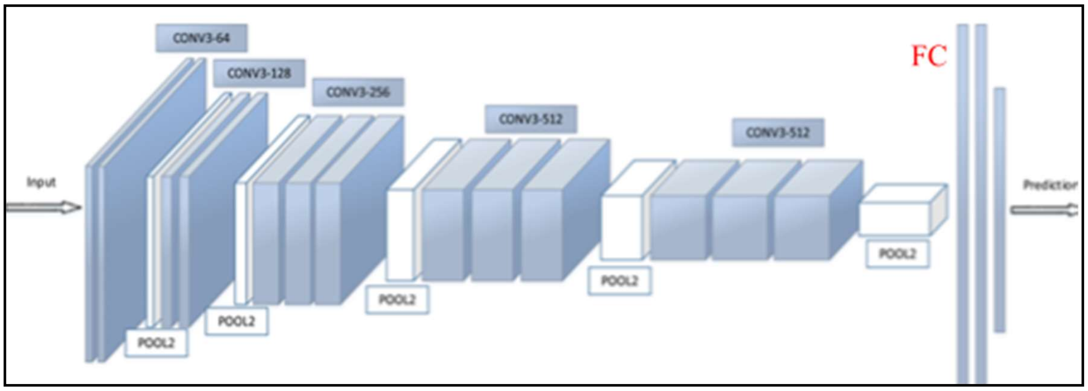


Figure 3.12: Fully connected **(Khiyari & Wechsler, 2016)**

3.7.4 Activation Function

An activation function is a function of a small value for small inputs in artificial neural networks, and a bigger value if the inputs outpace a limit. The activation function "flames" if the inputs are large enough; or else, it does nothing. In other words, an activation function functions similarly to a gate that verifies whether a received value is greater than a critical number (Teuwen & Moriakov, 2020).

Activation functions are helpful because they introduce non-linearities into neural networks, allowing them to learn complicated tasks (Teuwen & Moriakov, 2020).

Excluding nonlinearities, a neural network would calculate a linear function of its input, which would be excessively limiting. An activation function with the nonlinearity property can have a significant effect on the training performance of a neural network (Teuwen & Moriakov, 2020).

There are different activation functions such as ReLU, parameteric Relu, leaky ReLU, Sigmoid function.

The initial activation units in machine learning are sigmoid functions, which are utilized for logistic regression and simple neural network architectures.

The problem of the sigmoid function is solved (to some extent) with the tanh function; the main variation from the sigmoid function is that the curve is symetric across the origin with values ranging from -1 to 1.

Exponential Linear Units are used to accelerate the deep learning procedure by bringing the mean activations closer to zero. An alpha parameter, which must be a positive value, is employed in this procedure.

At the moment, the Rectified linear unit (Relu) is the most prevalent activation function. By introducing an activation function to each output layer, we can prevent the model from collapsing. Relu provides certain advantages, such as quick computing and gradient propagation (Ramachandranet al., 2017). These qualities make RelU the most preferred activation function in the line.

$$\boldsymbol{ReLU(x) = max(0, x), x \in R} \tag{3.3}$$

It is easy to see that $ReLU'(x) = 1\, for\, x > 0$ and that $ReLU'(x) = 0\, for\, x < 0$.

Every item of the input is subjected to a threshold operation by a ReLU layer, with any value less than zero being set to zero.

3.7.5 Softmax Layer

With a softmax layer, the input is processed using a softmax function. In neural network systems that estimate a multinomial probability distribution, the softmax function is used as the activation function in the output layer of the neural network model(Madhavan & Jones, 2017). In other words, softmax is used as the activation function for multi-class classification problems where class membership is required on more than two class labels, and where class membership is required on more than two class labels The soft max function takes an arbitrary value vector x and smothers it into a vector in the range [0,1] that accumulates to one. The formula for soft max is:

$$\boldsymbol{\sigma(x)j} = \frac{\boldsymbol{e^{xj}}}{\boldsymbol{\sum_{k=1}^{K} e^{xk}}} \tag{3.4}$$

for j = 1, . . . ,K. Where K is the number of classes. The numbers can now be used to describe a probability distribution over K possible possibilities.

3.7.6 Optimization

The following part discusses few of the effective frameworks that could be implemented to deep learning techniques to minimize training period and improve the system.

Back propagation: Backpropagation could be implemented to measure the gradient of the equation within each phase while using a gradient based approach to solve an optimization issue (Panigrahi et al., 2018).

Stochastic Gradient Descent: Gradient descent formulas that use the convex function guarantee that the desired limit is found without being stuck in a local minimum It can converge at the appropriate point in a variety of ways, based on the function's norms and training rate or phase volume (Lorraine & Duvenaud, 2018).

Learning Rate Decay: The learning level of stochastic gradient descent techniques can be changed to improve efficiency and decrease training period. The most commonly known approach is to progressively reduce the learning level in which significant adjustments can be at first and then gradually lessen the learning percentage during the training period. This helps the weights in the later stages to be fine-tuned (Ioffe & Szegedy, 2015).

Dropout: The drop out strategy may be applied to solve the overfitting troubles in deep neural networks. Through preparation, this approach is used by reducing units and their links at random (Jain, 2010a). Dropout is a normalization approach that effectively reduces overfitting and improves generalization error. Dropout increases performance on controlled learning problems in image processing computational biology textual categorization and voice understanding (Achille & Soatto, 2016).

Pooling Operation: A filter is configured in pooling operation such as Max Pooling, and it is then utilized through the nonoverlapping sub-regions of the input, with the output being the sum of the values found in the window. Pooling can minimize complexity and even the computation time of learning several variables (TAKAHASHI, 2010).

Batch Normalization: Batch normalization eliminates covariate change, allowing deep neural networks to run faster. Once the values are changed throughout the preparation, it adjusts the inputs to a layer on every mini-batch. Normalization decreases training epochs and maintains learning. The contribution from the prior induction layering can be normalized to improve the accuracy of a neural network (Jain, 2010b).

CHAPTER 4
PROPOSED PLACE IDENTIFICATION METHOD USING COMBINED IMAGE PRE-PROCESSING AND CNN

This chapter outlines the proposed method for place identification in Mecca. First, image pre-processing steps are explained. Then, different architectures are applied for feature extraction and classification using CNN, ANN and SVM. The details of these different system architectures are also explained in the sub-sequent sections.

4.1 Proposed Architecture

The recognition of distinct sites in Mecca and its surrounding regions is accomplished using a coordinated image processing strategy in this study as shown in Figure 4.1. First image pre-processing is applied; A median filter is used to preprocess the acquired images. Using a multiple threshold segmentation approach, the produced image is fragmented. The images are then classified as various color spots. The (GLCM) gray level co-occurrence matrix, was used to retrieve these color information, and feature that has been chosen were picked utilizing the Rough set technique. To properly discern the hotspots in Mecca, the processed images are feed into a CNN architecture for feature extraction and classification. We also utilized different classifiers with different architectures to demonstrate the performance against ANN and SVM (Taha et al., 2021). In the following sections, all parts of the architecture are illustrated in detail.

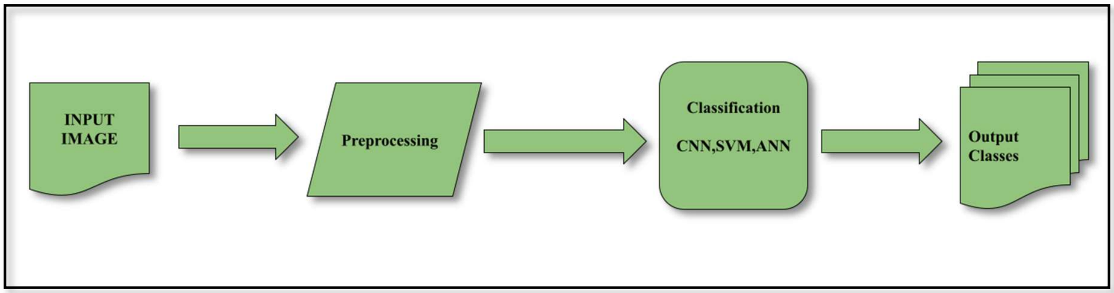


Figure 4.1: General Architecture of proposed system

4.2 Image Pre-processing

The RGB color scheme is used for all of the input images at the beginning of the process.

Information acquisition and pre-processing are the two most basic phases in image processing. When it comes to finding new places, image quality is becoming increasingly important. It is necessary to perform preliminary processing in advanced information mining, namely during the sorting and upgrading of location information. Pre-processing and image segmentation are virtually inextricably linked. Segmentation helps us to create the best image area with the highest level of precision. As you see in the Figure **4.2**, For preparing the source image, we utilized a median filter for noise reduction, multi threshold segmentation for splitting the input image into any number of parts or slices, and eliminating the specific section, GLCM for extracting feature in an image and rough set approach for feature selection. All of the pre-processing steps are performed in a parallel manner.

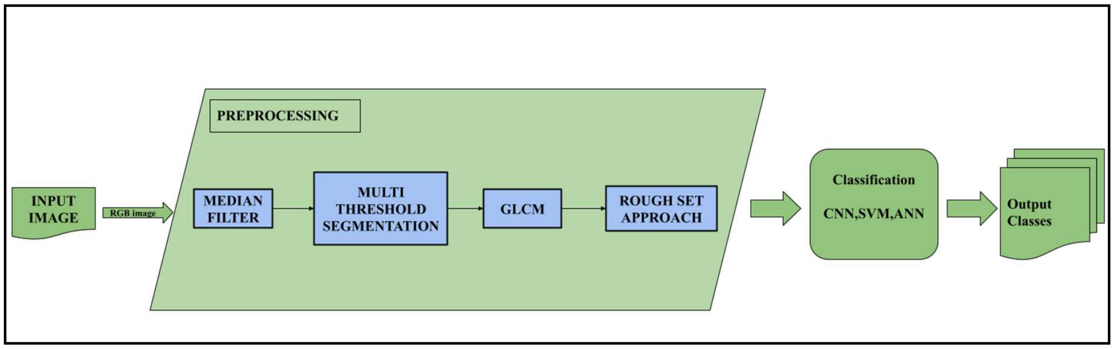


Figure 4.2: Flow of image pre-processing

4.2.1 Median Filter

For a long time, linear filters have become the main digital image analysis method. They are simple to apply and evaluate (Gabbouj et al., 1992). Linear filtering is one of the most effective methods of image enhancement available. It is a process in which a portion of the signal frequency spectrum is adjusted by the transfer function of the filter (da Silva & Mendonca, 2005).

Median filtering taking the average across a shifting window of specified number explored in the 1970s as a possible upgrade on linear filtering in the “edgy” situation. In “edgy” environments, basic median filtering is widely said to outperform linear filtering (Arias-Castro & Donoho, 2009).

The median filter is a non-linear optical filtering method for removing distortion from images and signals. This type of distortion removal is a common pre phase used to enhance the outcomes of future rendering (like image segmentation on an image). Median filtering is commonly used in digital image analysis as it retains borders thus eliminating distortion within some circumstances (Devarajan et al., 1990; Serra, 1994).

Because of its strong output for certain particular noise forms such as "Gaussian," "random," and "salt and pepper" noises, the median filter is among the most excellently order-statistic filters (Gonzalez & Woods, 2002). The middle pixel of a $M\,x\,M$ neighborhood is substituted by the median value of the corresponding window when using the median filter. It is worth noting that distortion pixels are thought to be significantly unique from the median. This form of noise can be removed using median filter (Gonzalez & Woods, 2002).

A median m of a group of values is the midpoint of the ordered range of norms it is the amount where the half of the values are smaller than m and half are larger. The median is further resilient to outliers and does not generate a new meaningless pixel value since it is a pixel number taken from the pixel neighborhood itself. It allows to avoid image information distortion and outline blurring.

To apply median filtering to a pixel in an image we must initially arrange the variables of that pixel and its peers calculate their median, and allocate that value to that pixel. For example, the median is the fifth highest amount in a 3×3 area, the 13th largest value in a 5×5 neighborhood, and so on. When many qualities in a neighborhood are similar, they are clustered together. Assume a 3×3 neighborhood has the following values (10, 20, 20, 20, 15, 20, 20, 25, 100).

The numbers are listed in the following order: $(10, 15, 20, 20, 20, 20, 20, 20, 25, 100)$, with a median of 20. As a result, the main purpose of median filters is to make places of different gray levels look more alike. An $n\ x\ n$ median filter, in reality, eliminates discrete groups of pixels that are weak or strong in comparison to their peers and have a region which is below $n^2/2$ (one-half the filter region). In this situation, "eliminated" means "driven to the neighbors' median level." Larger clusters are significantly less affected than smaller ones. (Gonzalez & Woods, 2002).

Despite the fact that the filtering of median is very beneficial, there are other filters. The median reflects the 50th percentile of a classified list of variables, but as the narrator recalls from simple statistics, ranking allows for a wide range of alternatives. The 100th degree, for instance, produces the so-called max filter, that is valuable for locating the brightest spots in a image. $R = \max\{\, z_k\ k = 1, 2, \ldots, 9\}$ is the result of a $3\ \times\ 3$ max filter. The minimum filter is the 0th level filter, which is utilized for the reverse reason (Gonzalez & Woods, 2002).

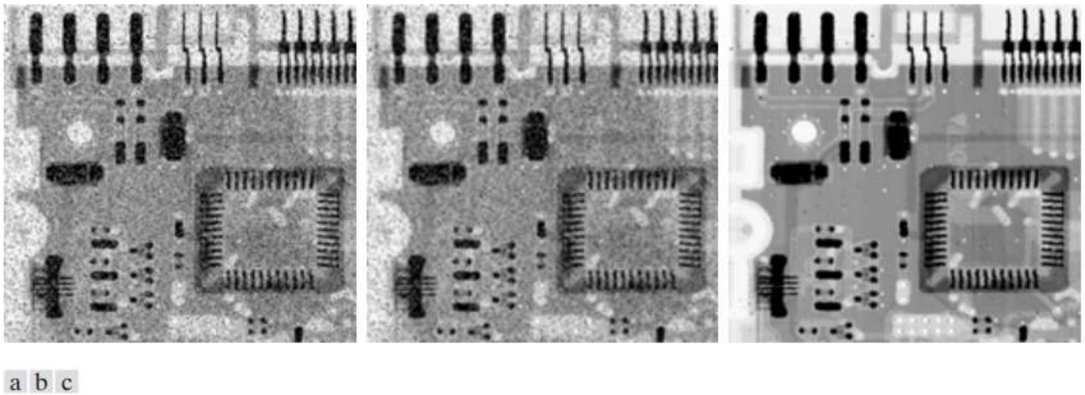


Figure 4.3: (a) The X-ray image of a computer chip with salt-and-pepper distortion has been distorted. (b) A $\mathbf{3\ \times\ 3}$ averages filter is used to reduce distortion (c) A $\mathbf{3\ \times\ 3}$ median filter is used to reduce distortion (Image courtesy of Lixi, Inc.'s Mr. Joseph E. Pascente.) (Gonzalez & Woods, 2002)

An X-ray image of a computer chip strongly distorted by salt-and-pepper distortion is shown in Figure **4.3** (a). The effect is displayed by rendering the distorted image with a 3× 3 neighborhoods mixing mask in Figure **4.3** (b) and the outcome of utilizing a 3 × 3 median filter in Figure **4.3** (c) to highlight the dominance of median filtering over average filtering in cases like this. While the image generated with the averaging filter has less apparent distortion, it comes at a substantial blurring cost. In this situation, median filtering clearly outperforms average filtering in every way. In total, median filtering outperforms averaging when it comes to removing added salt-and-pepper distortion (Gonzalez & Woods, 2002).

In the following paragraphs, the median filter used in our study is illustrated.

The images that were obtained had a low contrast and a lot of noise in their composition. Because of this, image denoising is critical for improving image quality by masking noises. Because of the noise, image quality and feature selection become hard. In this work, a non-linear median filter was employed to eliminate noise and emphasize several distinctive properties. The component closest to the window is referred to as source. 5x5 pixel window and associated source is shown in Figure **4.4**.

-	-	-	-	-
-	-	-	-	-
-	-	**Origin**	-	-
-	-	-	-	-
-	-	-	-	-

Figure 4.4: Element opening and source

The pixel window is used by the median filter to determine the output. The window can be any size and shape as long as it contains an odd number of pixels. For this investigation, a 5× 5 block shape was chosen because it is suitable to handle effectively and achieve high proficiency in the image shown in Figure **4.4**. All other window operators are less suited than the median filter. The basic idea behind this filter is to evaluate image sample estimates and determine whether they are accurate representations of the images.

All components in the image are processed, and the surrounding neighborhoods are utilized to determine whether the image graphs are accurate representations of the surrounding or not. The esteems of similar neighborhoods are organized in an even number format. The pixel values are then substituted with the filter's pixel approximation. Under normal conditions, This filter is applied through the use of a window containing an attribute with an odd value.

If the area has even pixels, the median value, which would be the midpoint of the calculated center pixel values, is chosen as the yield. The illustration below shows how this filter works in the window. The pixel values in the window are organized in ascending order in Figure **4.5**, and the median value is picked. The median value, which is the greatest midway location among the average estimates, is the primary benefit of the median filter, even if the area abnormal component has no effect on the median estimate (Figure **4.5**).

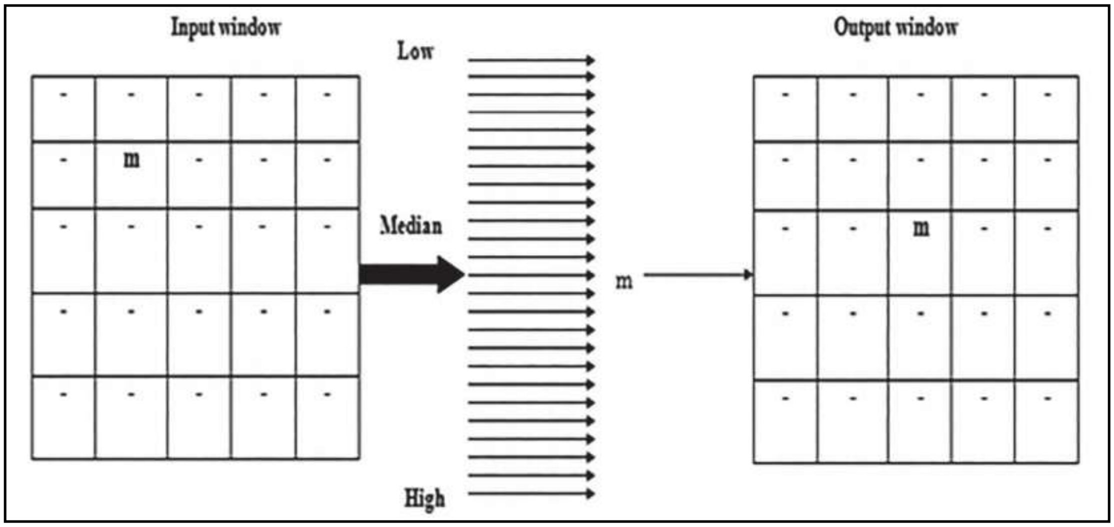


Figure 4.5: Operation of median filter

The median filter performs a complicated mathematical analysis of an image with a hazy commotion. The fluctuation in distortion is estimated in this typical image conveyance with 0 avg noise. The median filter outperforms the average filter when it comes to decreasing arbitrary noises. As a result, the pre-processed image is segmented for further processing.

4.2.2 Multithreshold Color Segmentation

The term "segmentation" refers to the process of breaking down an image into component parts or items. In total, fully independent segmentation is among the most challenging processes in image processing, and it is still a hot topic in image processing and machine vision areas. Segmentation plays a crucial function in image analysis because it is frequently the first stage that must be completed effectively prior to actually other steps such as feature extraction, classification, and definition can be undertaken (Solomon Chris, 2011).

Image segmentation is the method of dividing an electronic image into several sections (sequence of points), also known as image particles in electronic image rendering and machine technology. The aim of segmentation is to make an image more informative and simpler to interpret by simplifying and/or changing its expression. Sections and borders (lines, curves, etc.) in images are usually located using image segmentation. Image segmentation produces either a series of sections that occupy the whole image or a series of shapes derived from the image. In terms of some feature or calculated attributes such as color, intensity, or texture, each unit in an area is identical. When it comes to the similar trait, neighboring areas vary dramatically (Shapiro Linda G., 2001).

Thus, the primary aim of segmentation is to divide the image into equally exclusive areas to which we can later apply meaningful labels. The segmented objects are often referred to as the foreground, while the remainder of the image is referred to as the background. It is worth noting that there is not a single, "right" segmentation for every image. Instead, the right image segmentation is highly dependent on the forms of entity or area we want to classify. What kind of relationship does a data point need to have with its neighbors and other pixels in the image in which to be allocated to one of the regions? It is the most important topic in image segmentation, and it is generally answered in one of two ways (Solomon Chris, 2011):

Methods for creating edges and boundaries: The identification of borders as a way of examining the border among areas is the pillar of this method. It searches for strong variations among clusters of pixels as a result.

Methods that are dependent on area: This method applies pixels to an area based on how close they are to one another.

One of the really challenging functions of image analysis is segmentation of nontrivial images. The performance or failure of automated research techniques is determined by segmentation accuracy. As a result, great consideration must be given to increase the chances of correct segmentation (Rafael C. Gonzalez & Richard E. Woods., 2007).

The majority of the segmentation methodologies presented are focused on one of two fundamental principles of intensities: discontinuity or resemblance, as previously stated. The method in the first type is to divide an image into sections relying on sudden variations in intensity such as borders. The main methods in the second group are focused on segmenting an image into related regions based on a collection of predetermined requirements (Rafael C. Gonzalez & Richard E. Woods., 2007). Methods in this latter group include thresholding which is also utilized in our work, include area expanding, and area separating and combining.

4.2.2.1 Segmentation based on Image Features and Functionality

Images have three essential characteristics or attributes that can be used to segment them (R. M. Haralick & Shapiro, 1985; Rafael C. Gonzalez & Richard E. Woods., 2007; Solomon Chris, 2011)

Color: The easiest and most clear way of distinguishing among items and context in specific situations. Items that have specific color characteristics (i.e. are limited to a specific area of a color space) can be distinguished from the context. Separating an apple from an image of a blue tablecloth, for example, is a simple process.

Texture: In image analysis, texture is a very nebulous term. It does not have a specific meaning but it suits our daily definitions of a "raw" or "clean" object fairly well. As a result, texture points to the image's "standard" spatial variability in frequency or color tones across a given spatial range. The variation or other numerical periods of the density within a given zones/spatial level in the image are used to calculate a variety of texture parameters.

Motion: A strong cue may be the movement of an item in a series of image videos. Basic frame-by-frame subtraction methods are also adequate to provide a precise description of the flowing target as it occurs towards a stationary backdrop.

In brief, most image segmentation techniques can integrate and utilize data from one or more of the properties color, texture, and motion (Solomon Chris, 2011).

4.2.2.2 Thresholding

Intensity thresholding in segmentation is based on a relatively easy concept. A threshold is set a value so that points with scores higher than the threshold are allocated towards one area, while points with variables less than the threshold are allocated to some other (adjacent) area. Thresholding converts an intensity image $I(x, y)$ into a binary image $b(x, y)$ using a basic parameter (Solomon Chris, 2011). T is threshold,

$$\boldsymbol{b(x,y)} = \begin{cases} \mathbf{1} & \boldsymbol{if\ I(x,y) > T} \\ \mathbf{0} & \boldsymbol{Otherwise} \end{cases} \tag{4.1}$$

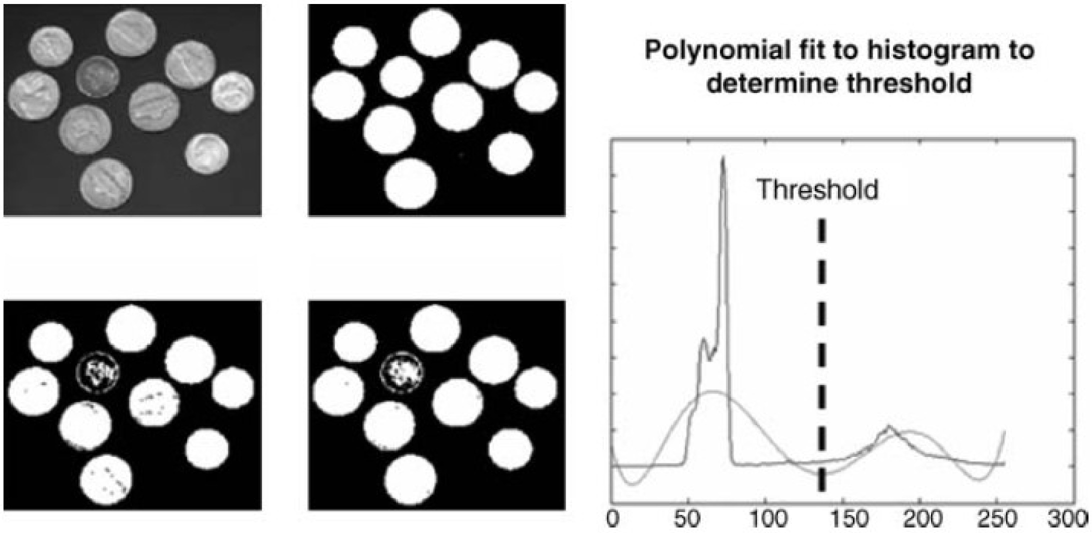


Figure 4.6: Results of Threshold selections (Solomon Chris, 2011)

The effect of frequency thresholding on an image of many coins falling on a black backdrop is shown in **Error! Reference source not found.** (upper right); every one of the coins are actually detected. The required threshold was determined remotely by experimentation in this scenario. This is permissible in a restricted range of circumstances for instance, some inspection processes which enable a human user to maintain a reasonable threshold prior to actually automatically executing a series of identical images. Numerous image analysis processes, on the other hand, necessitate complete automation and a requirement for automatically choosing a threshold is frequently needed (Gonzalez & Woods, 2002; Solomon Chris, 2011). First image (top left). Lower left: the product of automated threshold filtering based on image frequency distribution polynomial fitting. (Lower right) shows consequence of Otsu's model (Sezgin & Sankur, 2004a) for automated threshold picking. The histogram and consequence of applying a 6th order polynomial are shown on the right.

Calculating the pixel intensities and fitting a polynomial equation to it is an easy process. An appropriate threshold could be found at the lowest turning point of the graph if the sequence of the polynomial equation is carefully selected and the polynomial appropriately matches the typical form of the histogram.

Figure **4.7** (Rafael C. Gonzalez & Richard E. Woods., 2007) depicts a more challenging thresholding challenge including a histogram of three major phases, which relate to two forms of light artifacts on a black background for instance. Multiple thresholding assigns a (x, y) point to the context if $f(x, y) \leq T_1$, to one entity type if $T_1 < f(x, y) \leq T_2$, but to the other entity type if $f(x, y) > T_2$. In other words, the segmented image is provided by

$$\boldsymbol{g(x,y)}\begin{cases} \boldsymbol{a} & \boldsymbol{if\ f(x,y) > T_2} \\ \boldsymbol{b} & \boldsymbol{if\ T_1 < f(x,y) \leq T_2} \\ \boldsymbol{c} & \boldsymbol{if\ f(x,y) \leq T_1} \end{cases} \qquad \textbf{(4.2)}$$

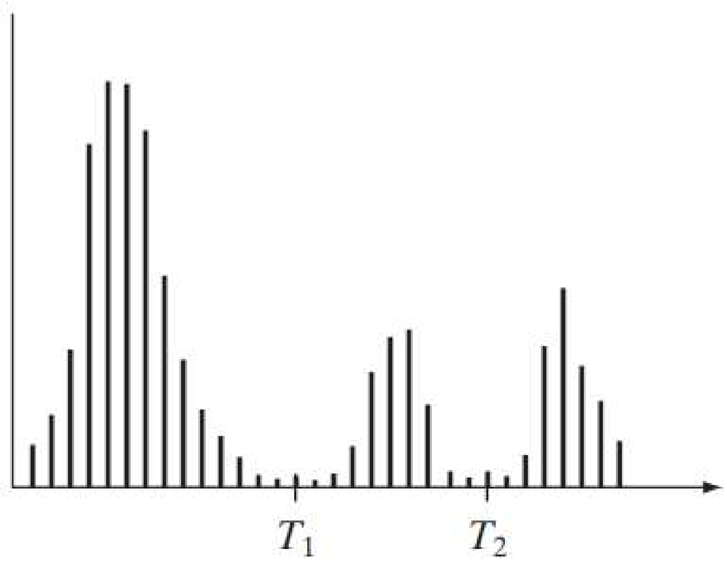


Figure 4.7: Two thresholds (Rafael C. Gonzalez & Richard E. Woods., 2007)

We can conclude perceptually from the above that the width and depth of the valley(s) splitting the histogram phases are clearly linked to the effectiveness of intensity thresholding. The following factors influence the characteristics of the valley(s): (a) the distance among spikes (the farther separated the peaks are, the stronger the likelihood of distinguishing the phases); (b) the distortion quality in the image (the methods extend as noise rises); (c) the comparative scales of objects and the background; (d) the consistency of the illumination supplier and (e) the homogeneity of the reflector (Gonzalez & Woods, 2002). Our study's segmentation will be discussed below.

4.2.2.3 Thresholding in this Study

Image segmentation is a bit-by-bit process for separating an input image into any number of segments or fragments and deleting a specific section for creating the source image. This is a method of dividing a shaded image into discrete sections, with the areas conveying location information. To accomplish thresholding for gray images, the brightness scope values are frequently characterized by the original image.

Thresholding has traditionally been accomplished by evaluating a range of brightness values in the main image, then selecting the pixels within that range as relating to the foreground and ignoring all other pixels as relating to the background, which is a straightforward method. When such an image is displayed, it is usually represented as a binary or two-level image (Sezgin & Sankur, 2004b). The following is the general rule for pixel thresholding at different grey levels:

$$\boldsymbol{gs(x,y)} = \begin{cases} \mathbf{0}, & \boldsymbol{fs(x,y) < T} \\ \mathbf{1}, & \boldsymbol{fs(x,y) \geq T} \end{cases} \tag{4.3}$$

In this equation, T represents the threshold value, $fs(x,y)$ represents the original pixel value, and g(x, y) represents the pixel value that was obtained after thresholding..

The Equation 4.3 will change, when there are more than one thresholding value at a time as follow:

$$\boldsymbol{(x,y)} = \begin{cases} \mathbf{0}, & \boldsymbol{fs(x,y) < t_1} \\ \mathbf{1}, & \boldsymbol{t_1 \leq fs(x,y) \leq t_2} \\ \mathbf{0}, & \boldsymbol{fs(x,y) > t_2} \end{cases} \tag{4.4}$$

t1 represents the lower threshold level and t2 represents the higher threshold level.

Multiple variables are used to classify each pixel in color images, which permits for multi-thresholding (Otsu, 1979) to be applied to the image. RGB values are used in color imaging to distinguish each pixel from another.

The RGB pixel characteristics are being investigated in order to obtain the image's most important characteristics. For instance, if we are focused in green areas (also known as Forests), the color thresholding algorithm should be able to extract the pixels that are green in color and dismiss the pixels that are other colors relying on the color knowledge. If we're only interested in blue areas (the sky), the color thresholding method should be able to extract blue color from other items while rejecting pixels from blue areas. The proposed method's procedure is outlined below.

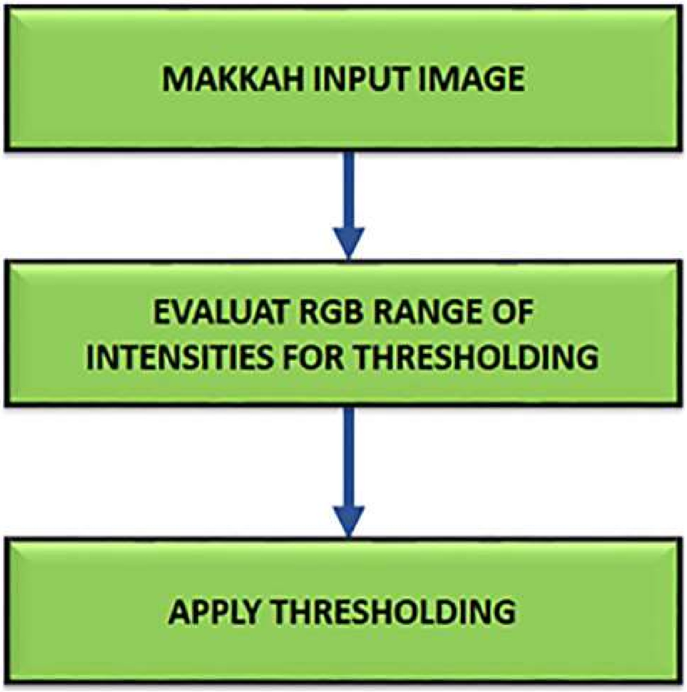


Figure 4.8: Various methods for different-color thresholding

We use formulas 4.5, 4.6 and 4.7 respectively, determining the thresholding range of multiple colors using the maximum and minimum RGB variety.

$$gs(x,y)\begin{cases} fs(x,y), 0 \leq red(x,y) \leq t_r \\ g1(x,y), red(x,y) > t_r \end{cases} \tag{4.5}$$

$$gs(x,y)\begin{cases} fs(x,y), T_g \leq green(x,y) \leq 255 \\ g1(x,y), green(x,y) < t_g \end{cases} \tag{4.6}$$

$$gs(x,y)\begin{cases} fs(x,y), 0 \leq blue(x,y) \leq t_b \\ g1(x,y), blue(x,y) > t_b \end{cases} \tag{4.7}$$

The gray element range is g1(x, y), the red element range is red(x, y), the green element range is green(x, y), and the blue element range is blue(x, y).

The basic equation given in Equations (4.3) and (4.4) has been partially altered to convert the approach of grey level thresholding to color thresholding, as shown in Equations (4.5), (4.6), and (4.7). In the new equations each RGB element is regarded separately from the others.

For more precise image segmentation, a multiple stage thresholding method is used on RGB color images shown in Figure 4.9.

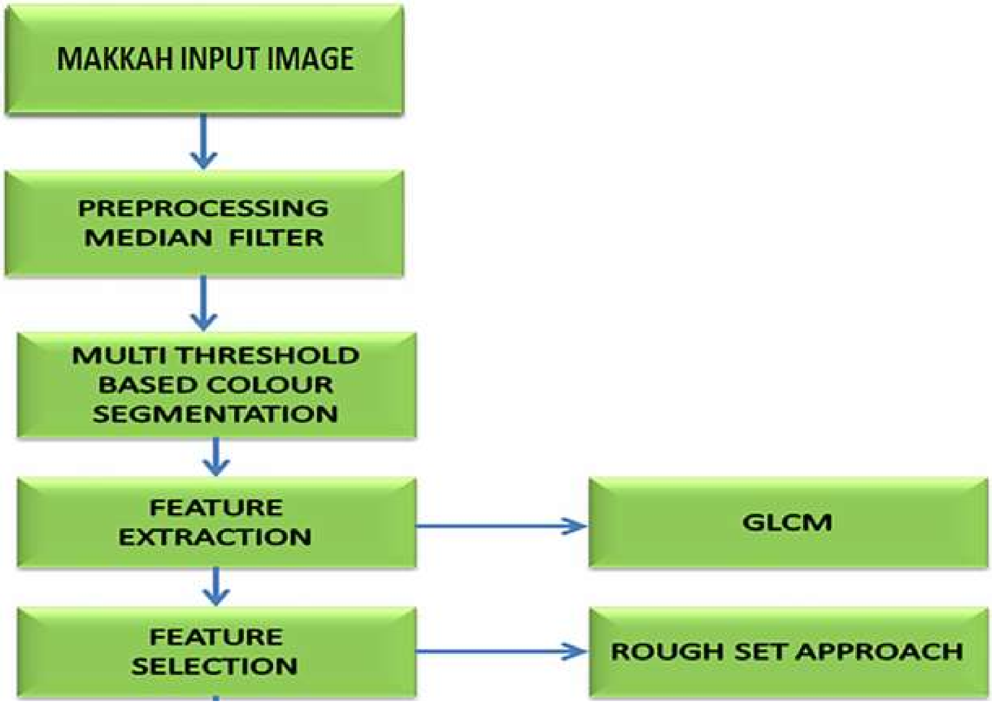


Figure 4.9: Flowchart of proposed image pre-processing work.

4.2.3 Feature extraction by GLCM

Feature extraction is a systematic process that defines the related structure details found in a pattern in order to make the process of classifying the pattern easier. Feature extraction is a method of dimensionality removal used in pattern recognition and image processing. The primary aim of feature extraction is to remove the most important details from the existing information and display it in a lower-dimensional space (Kumar & Bhatia, 2014). Whenever an algorithm's inputs is very huge to analyze and is assumed of being repetitive (lots of data but little information), the data is converted into a restricted description collection of features. Feature extraction is the process of transforming input data into a collection of properties. It is assumed that the features collection will derive the necessary details from the source data in order to accomplish the required function utilizing this simplified description rather of the maximum scale input if the features retrieved are wisely selected (Kumar & Bhatia, 2014) .

After pre-processing and achieving the required degree of segmentation, a feature extraction method is used to separate features from the sections, which is then accompanied by classification and post-processing strategies. It is critical to concentrate on the feature extraction process because it has a visible effect on the identification framework's performance (Kumar & Bhatia, 2014) .

Texture is a valuable feature for distinguishing regions of interest in an image. Haralick et al. (Robert M. Haralick et al., 1973) suggested Grey Level Co-occurrence Matrices (GLCM) as one of the first approaches for extracting features in 1973. It has been commonly utilized in several texture analysis implementations since then, and it has stayed an essential feature extraction tool in the edge detection field. Haralick removed fourteen attributes from the GLCMs to describe texture (Robert M. Haralick, 1979).

To obtain reasonable precision with limited resources, image pre-processing techniques are required to improve the consistency of images. Image pre-processing entails steps such as distortion reduction and edge detection, resulting in an immediate abnormality detection system. In computer vision issues, the Gray Level Co-occurrence Matrix (GLCM) texture features are commonly included. The second-order numerical knowledge of gray levels among neighboring pixels in an image is represented by GLCM (Seal et al., 2018).

The GLCM examines 2 points and their relationship, that is known as a 2nd order layer. The gray value association provided by a kernel filter is stored in the co-occurrence matrix. Certain paths characterize the adjacent points in the transition (e.g., 0, 45, 90 degrees, etc.). It is also possible for the path to be negative (reverse). The location is determined by the gray rate measures being identical (Bethanney Janney et al., 2020).

Let's say there's a M co-occurrence matrix with N dimensions, and the entities' parameters and positions are i, j (Mall et al., 2019).

Contrast: The contrast is a measurement of the image's frequency domain, or the relationship between pixel intensity and its neighbors across the image. Also it defines how much difference there is in the image at a regional level. The variation in brightness between each item and other items in similar area of vision is used to calculate the contrast. The equation 4.8 defines the contrast function of an image ‘Contr.' as follows:

$$\boldsymbol{Contr} = \sum\sum |\boldsymbol{i} - \boldsymbol{j}|^{2}\boldsymbol{p}(\boldsymbol{i},\boldsymbol{j}) \qquad \textbf{(4.8)}$$

The cell is on the diagonal and $(i-j)=0$ when i and j are equal. These variables are assigned a weight of 0 since they indicate pixels that are identical to their neighbors (no contrast). There is a minor contrast if i and j differ by one, and the weight is one. When I and j differ by two, the contrast increases and the weight increases to four. As $(i-j)=0$ grows, the weights continue to grow exponentially.

Correlation: A pixel's correlation with its neighboring pixel across the full image is measured by the correlation function of an image. The value of a negative correlation of an image ranges between -1 to 1 for a completely positive image while the value for a static image is infinity. Below Equation defines the ‘Correl' function of an image.

$$\boldsymbol{Correl} = \sum_{\boldsymbol{m}=1}^{\boldsymbol{M}}\sum_{\boldsymbol{n}=1}^{\boldsymbol{N}} \boldsymbol{C_n}(\boldsymbol{i},\boldsymbol{j})\,\frac{(i-\mu_x)(j-\mu_y)}{\boldsymbol{\sigma_x\sigma_y}} \qquad \textbf{(4.9)}$$

Here, the terms $\mu_x\mu_y$ and $\sigma_x\sigma_y$ denote the means and standard deviations of the summed cooccurrence matrix Cn, which denotes a long horizontal and vertical spatial plane, and a long horizontal spatial plane, respectively.

Entropy: Variation is used to describe the texture of the image, while entropy tests the level of homogeneity among pixels inside the image. Entropy is significantly inversely linked to energy, with images with higher gray levels having higher entropy. In the below equation, the entropy function of an image ‘Ent' is described as shown below.

$$\textbf{Ent} = \sum_{\boldsymbol{i,j}=0}^{\boldsymbol{N}-1} \boldsymbol{P_{i,j}}(-\boldsymbol{ln}\ \boldsymbol{P_{i,j}}) \qquad \textbf{(4.10)}$$

Entropy is commonly categorized as a first-degree metric, but it should be categorized as a "zeroth"-degree measure instead! For the reason that $\boldsymbol{P_{i,j}}$ is always between 0 and 1, because it is a probability, and a probability can never be either greater than 1 or smaller than 0. As a result, the value of **ln** $(\boldsymbol{P_{i,j}})$will always be zero or negative. The greater the absolute value of **ln** $(\boldsymbol{P_{i,j}})$, the smaller the value of $\boldsymbol{P_{i,j}}$

Sum of squares variance: The distribution of the gray level total range of the image is calculated by the total of squares variance. The default deviation a first-order numerical measure is closely associated with this heterogeneity. The variability of the image increases as the gray scale value varies from its average value. In equation below, the sum of squares variance function of an image called 'Variance' is described as follows:

$$\mathbf{Variance} = \sum_{i=1}^{N_g}\sum_{j=1}^{N_g} (\mathbf{i} - \boldsymbol{\mu})^2 \, \mathbf{p}(\mathbf{i},\mathbf{j}) \qquad \textbf{(4.11)}$$

Where p(i,j) is the (i,j) -th entry of the normalized GLCM, N_g is the total number of gray levels in the image.

Sum of average: The sum average function calculates the mean of an image's grayscale total range. The 'Sum average' function of an image is described in the following equation:

$$\mathbf{Sum\ average} = \sum_{i=2}^{2N_g} \mathbf{i}\mathbf{p}_{x+y^{(i)}} \qquad \textbf{(4.12)}$$

Where $P_{x+y}$is the gray level sum distribution. Also it is related to the distribution of the sum of co-occurring pixels in the image

GLCM is used to capture the features of an image. There are a variety of approaches for extracting the attributes that allow the pixel to correspond to the actual image. Using the gray range of m and n, the co-event array $C\ (m, n)$ determines the occurrence of noise pixels. When the gray count is increased to improve quality, it makes the process more complex. Various processes such as contrast, correlation, entropy, and variance are extracted to reduce the length of the features.

4.2.4 Feature selection by rough set approach

Feature selection techniques have been essential elements of the learning model in recent years, helping to eliminate unnecessary and duplicate features (Bolón-Canedo et al., 2015).

The selection of selected properties which are most predictive of a given result is referred to as feature selection. This is an issue that can be seen in a variety of fields, including machine learning, signal processing, and, more recently, bioinformatics/computational biology. While working with massive datasets with tens or hundreds of thousands of parameters, feature selection is one of the most critical and difficult challenges (Banerjee et al., 2006).

Individual and subset assessment are the two main techniques to feature selection. Individual assessment, also known as feature rating, evaluates item attributes by adding values to them based on their relative importance. Subset analysis, on the other hand, generates nominee function subsets based on a search technique. Aside from that, function selection approaches can be broken down into three categories: filters, wrappers, and embedded methods (Bolón-Canedo et al., 2015).

The goal of feature selection strategies is to reduce the numbers of parameters in a database that are unnecessary or duplicate. Feature selection, unlike other dimensionality reducing approaches, keeps the initial attributes following reduction and selection. The advantages of feature selection are numerous: it enhances additional analysis by eliminating distracting data and outliers it speeds up and reduces the expense of post-analysis, it facilitates information representation and it offers a clearer interpretation of the fundamental mechanism that provided the data (Banerjee et al., 2006).

Rough set is a numerical method for dealing with inaccurate and incomplete information. Machine learning, data mining, and information exploration have all used it. The so-called feature selection, which is particularly useful for classification challenges is among the implementations of Rough set theory in machine learning. This is accomplished by identifying a collection of features that can be reduced. The reduce collection is a subset of all characteristics that keeps the identification precision of the initial parameters (Anaraki & Eftekhari, 2013).

When indicating the accurate input image, this technique aids in the discovery of characteristics with a low subset and high reliability. The issue is determining which feature to use by decreasing undesired noises or mismatching features. Relevance and duplication are used to obtain the sub feature. Picking significant characteristics of input images characterizes the important characteristics of input images.. See the pre-processing method in detail in Figure **4.10**.

A step known as Rough Set Approach (Banerjee et al., 2006) is used to point out the dependent information and to lower the points in the datasheet. Subsets like maximum and minimum are taken. Consider $T\ (U, A, C, D)$ the rules having with U as a global object, A as a set of crude highlights, C as a lot of condition feature, D as a decision feature, and $C, D\ \subseteq\ A$. For a subjective set $P\ \subseteq\ A$, a confusion connection is

$$\boldsymbol{IND}(\boldsymbol{P}) = \{(\boldsymbol{x}, \boldsymbol{y}) \in \boldsymbol{U}^{*}\boldsymbol{U}: \forall \boldsymbol{a}\ \in \boldsymbol{P}, \boldsymbol{a}(\boldsymbol{x}) = \boldsymbol{a}(\boldsymbol{y})\} \tag{4.13}$$

If $P\ \subseteq\ c$ and $x\ \subseteq\ v$ then bottom and top approximate of X that regards to P is expressed as

$$\boldsymbol{PX} = \{\boldsymbol{x} \in \boldsymbol{U}: [\boldsymbol{x}]_{\boldsymbol{IND}(\boldsymbol{P})} \subseteq\ \boldsymbol{X}\} \tag{4.14}$$

$$[\boldsymbol{x}]_{\boldsymbol{INDP}(\boldsymbol{P})} = \{\boldsymbol{y}\ \in \boldsymbol{U}: \boldsymbol{a}(\boldsymbol{y}) - \boldsymbol{a}(\boldsymbol{x}), \forall \boldsymbol{a}\ \in \boldsymbol{P}\} \tag{4.15}$$

The above equation is the equivalent section of $x\ in\ U/IND(P)$. The $P\ +\ ve$ area of D is the universe U that was differentiated with certainty to 1 class of utilizing characteristics of P. Hence, the characteristics are picked by the approximate by using rough set approach and some for dividing.

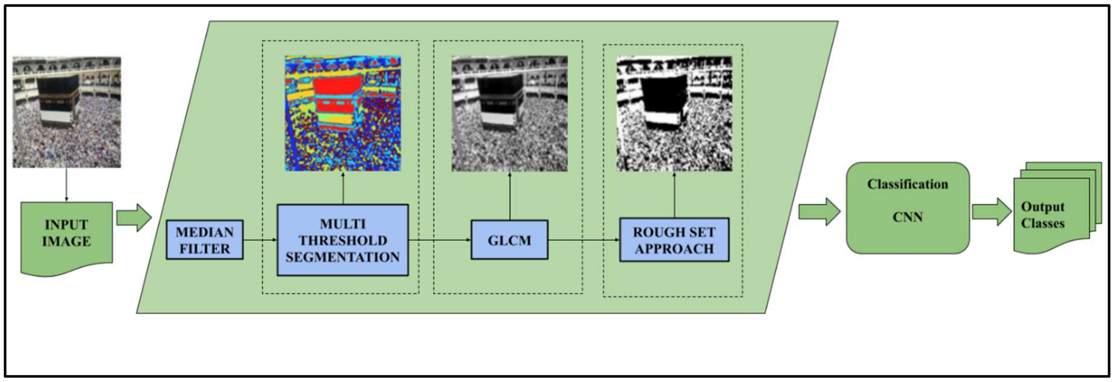


Figure 4.10: Flow of image pre-processing in detail

4.3 Proposed Algorithms for Feature Extraction and Classification

When a computer analyzes an image and determines which "class" it belongs to, this is known as image classification. A class is basically a term, such as "vehicle," "animal," "building," "location," and so forth (Gonzalez & Woods, 2002).

Raw pixel data was used to classify images in the beginning. Computers would decompose images into individual pixels as a result of this. The issue is that two images of the same subject can appear to be very different (Gonzalez & Woods, 2002). They can have a variety of backgrounds, angles, positions, and so on. This makes it difficult for computers to accurately "see" and classify images. Deep learning and machine learning methods made the problem easier to solve.

Image categorization is a supervised learning problem in machine learning and deep learning. Labeled example images are utilized to train a model to detect a set of target classes (things to identify in images). (Mitchell, 1997).

In this work we used ANN and SVM machine learning algorithms to compare our results with our proposed deep learning CNN architecture as shown in the **Error! Reference source not found.** The details of these architectures are illustrated in the sub-sequent sub-sections.

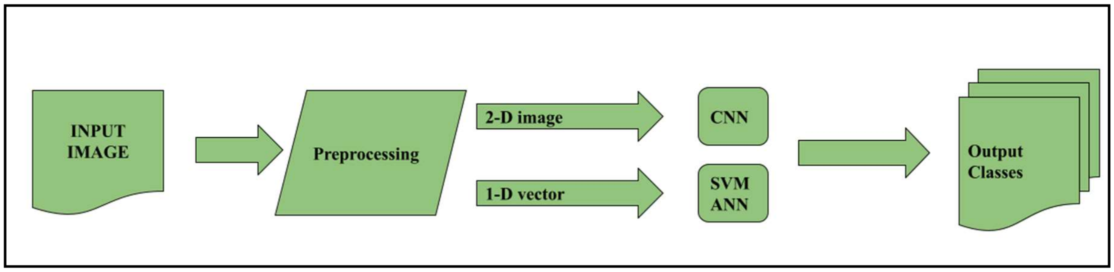


Figure 4.11: Proposed image classification strategy

4.4 CNN Architecture

CNN is a supervised learning approach with a feed forward architecture that is inhibited by the human brain's neural activity. It is a multilayer perceptron alternative that requires minimal filtering methods and requires less training. CNN applies both feature learning (using initial layers) and classification (using the last layers). The trained information filters the input images at several stages. To locate the image characteristics, the input travels through all phases (convolution, pooling, and fully connected layers). In our study we observe that when layer counts are raised, the training step will be more effective. Proposed CNN architecture have nine convolution layers, eight pooling layers, and two fully connected layer. In each convolutional layer contains 32 filters, size of kernel 5x5, valid padding activation function is Relu6 as shown in the Figure **4.12** and Figure **4.13** for details. In each pooling layer, a window of 3x3 pixel is used with 2 stride.

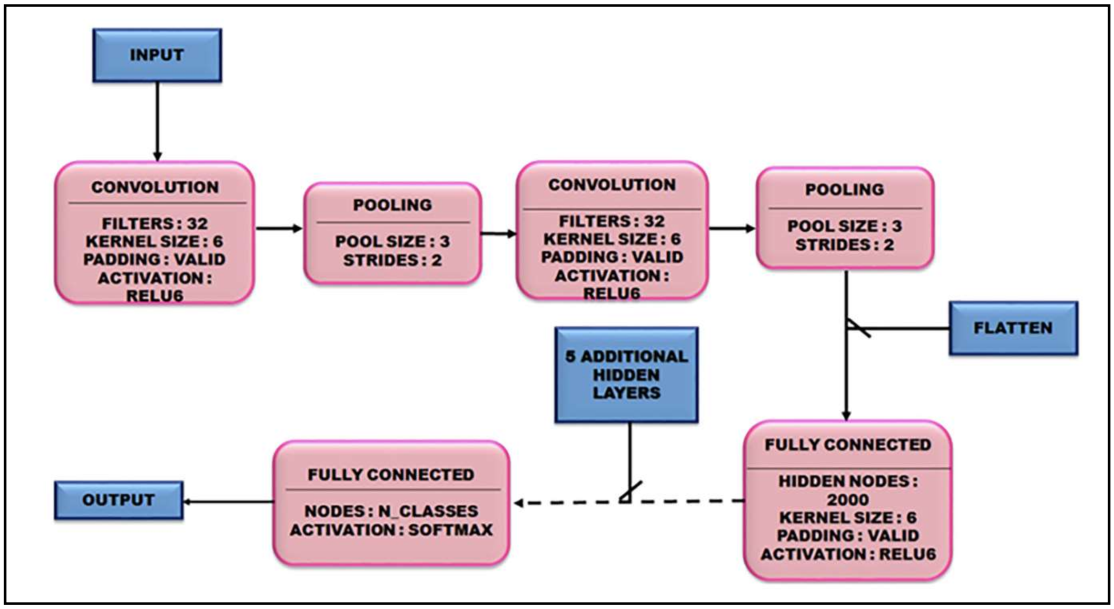


Figure 4.12: Proposed CNN architecture

The classification of places in Mecca is done using CNN in this study. We use the Mecca dataset as it is explained in the Evaluation section. Both training and validation datasets are required for supervised learning. In the training stage, some prehandling actions are required. Figure **4.13** depicts the work procedure of our work.

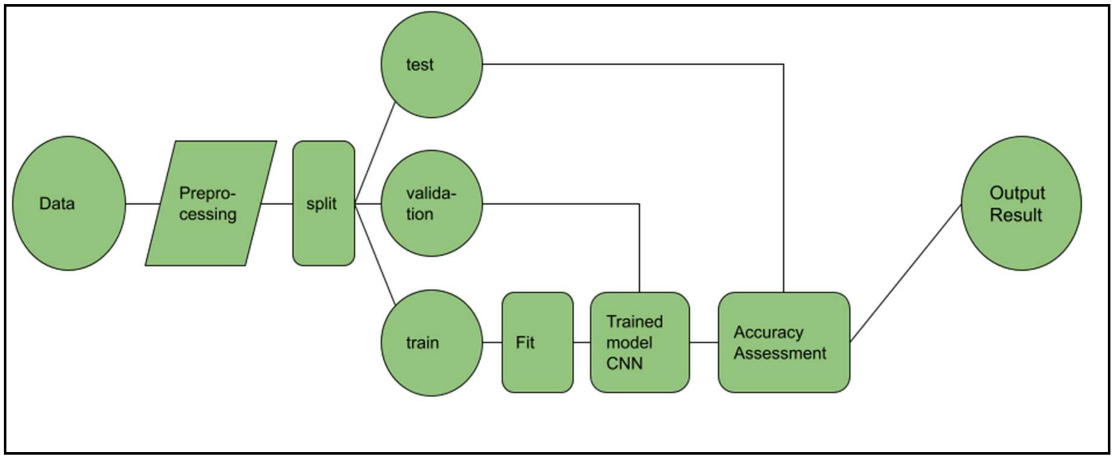


Figure 4.13: Block diagram of CNN flow

4.5 SVM Architecture

There are numerous motives for comparing our proposed method with SVM, but only a few of them have been discussed here.

SVM is a very useful method when we don't have a lot of knowledge about the data. SVM can be applied to a variety of data types, including images, text, and audio. It can be applied to data that is not evenly distributed and whose distribution is undetermined. When there is a straightforward illustration of separation between classes, SVMs commonly do not struggle from the condition of overfitting and accomplish remarkably well. It is possible to find numerous actual implementations for SVM in the actual world such as Sentiment Analysis of Emotions in Speech or Video or Image, Cancer Diagnosis, Handwriting Recognition, and so on. Numerous latest researches have demonstrated that the SVM has commonly been able to achieve maximum classification accuracy achievement over other methods of classification. (Chidambaram & Srinivasagan, 2018).

In classification and regression, support vector machines are a type of supervised learning method. SVM creates $(n-1)$ hyperplanes for n-dimensional space, which are used to divide datasets into different classes. Maximum-margin hyper-plane SVMs were created with linear classification in mind. In (Cortes & Vapnik, 1995), introduced a method for classifying non-linearly separable data by applying kernel tricks to current maximum margin SVMs. The overall structure of the technique has not changed, however the dot product employed in the prior SVM has been replaced by a kernel function. If y denotes the class level to which x belongs, then the input to the classifier is the sequence $(x1, y1), (x2, y2) \ldots \ldots \ldots (xn, yn)$.

4.5.1 Kernel trick

SVMs are also referred to as kernelized SVMs mainly due to the kernel that transforms the input data space into a higher-dimensional space. It makes use of existing features, performs some changes, and adds new ones. Polynomial Kernel, Gaussian Kernel, Radial Basis Function (RBF), Laplace RBF Kernel, Sigmoid Kernel, and Anove RBF Kernel are all popular kernels. (Huh, 2015). As you see in the Figure **4.14**, RBF kernel is used in our work due to the best results.

4.5.2 Multi class SVM

Most early machine learning methods, such as perceptron learning and SVM, were created with binary classification in mind. Researchers have suggested a variety of multi-classification methodologies over time, based on a conjunction of many binary classifiers or trying to solve the entire classification as an optimization problem.

Most early machine learning algorithms, such as perceptron learning and SVM (Thirumala et al., 2019), were built for binary classification. Multi-classification problems, on the other hand, are highly common in real-world applications. Researchers have suggested a variety of multi-classification methods over time, based on a combination of numerous binary classifiers or treating the entire classification as an optimization problem.

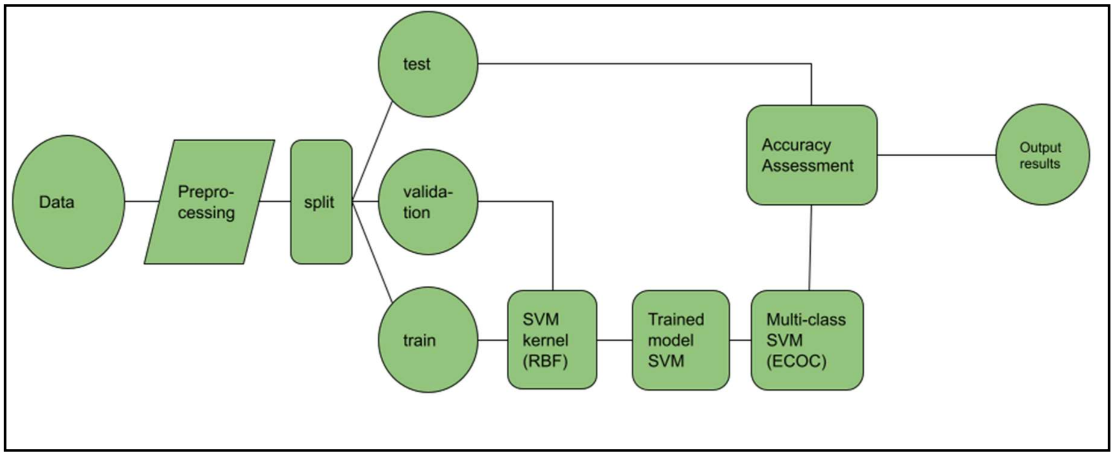


Figure 4.14: Block diagram of SVM

SVM multiclass learning, for example, is usually done in one of two methods. The first method involves building a set of binary SVM classifiers and then combining them to forecast which class a new input vector belongs to. One-versus-rest $(1 - v - r)$, one-versus-one $(1-v-1)$, Decision Directed Acyclic Graph (DDAG), and Error Correcting Output Code (ECOC) are the most well-known of these algorithms (Thirumala et al., 2019). The second approach treats the entire dataset as a single optimization problem and attempts to solve it in a single step, such as the "Crammer and Singer" approach (Thirumala et al., 2019). This method, however, is remarkably challenging and computationally expensive to implement due to its sophistication, and it is rarely used in real-world SVM multi-classification implementations. (Thirumala et al., 2019).

We preferred the first method for multi class issue. We used Error Correcting Output Code (ECOC) due to the best results. See the Figure **4.14**.

4.6 ANN Architecture

There are a variety of reasons for contrasting our proposed method with the ANN, but only a couple of them have been mentioned here.

Artificial Neural Networks with the ability to learn by themselves and produce output that is not restricted to the input that is supplied to them. They have the ability to perform numerous processes in parallel without impacting the overall efficiency of the model(Yang, 2019). Due to the potential of ANNs to process a large number of inputs and infer hidden as well as complex, non-linear relationships, they are a promising tool for machine learning (Yang, 2019). Beside that, the vast majority of Deep Learning models are part of a wider family of machine learning methods are relied on artificial neural networks (Teuwen & Moriakov, 2019). Also, Artificial Neural Networks are effectively being utilized in a broad range of implementations varying from face recognition to decision-making also a significant contribution in image and character identification is served by artificial neural networks (ANNs) (Yang, 2019).

The three layers artificial neural network used in this study are as follows: In our work we used 256x256 vector as input, two hidden layers: first one has 20 unit, second one has 10 units and four output layer, since we have for attributes for classification.

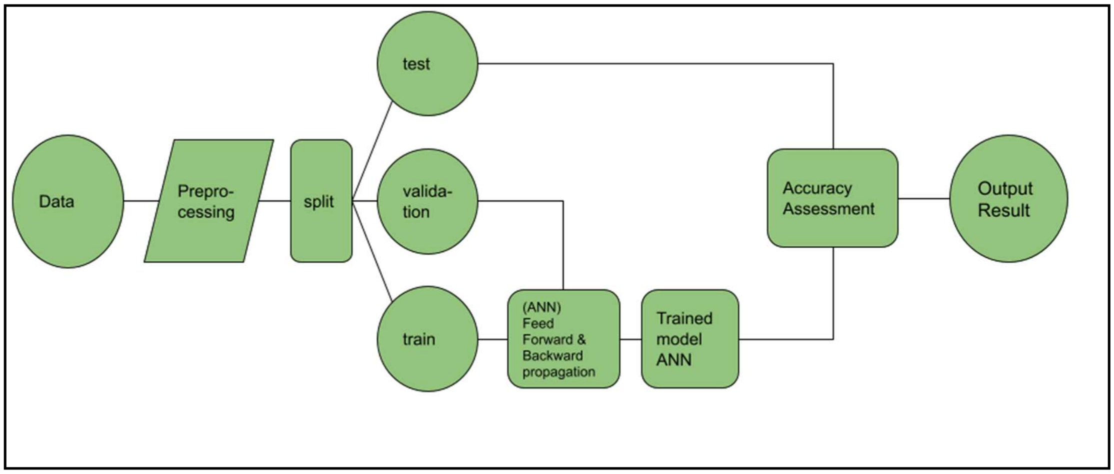


Figure 4.15: Block diagram of ANN

All of our algorithms flow are in the same approach, as seen in Figure **4.15**, the dataset is first divided into three sections: "test," "validation," and "train." The training procedure then begins. Overfitting is solved by "validation" dataset. The trained model is then tested using the "test" dataset.

CHAPTER 5

EVALUATIONS

This chapter discusses the evaluations carried out to assess the performance of the proposed combined image pre-processing and CNN architecture for place identification in Mecca. First, the dataset and experimental settings are discussed in section 5.1. Then, evaluation metrics formulas are given in section 5.2. In the subsequent sections, experiments with other popular methods such as ANN and SVM are presented. The proposed method is also compared with a deep learning based place identification method.

5.1 Dataset and Experimental Setup

Our dataset is established from HEUR and updated videos and images from internet. We focused on four important places. The HEUR dataset is made up of videos and images shot throughout the Hajj and Umrah seasons of 2011 and 2012. The Hajj and Umrah ritual events (rotating in Tawaf around Kabaa, performing Sa'y between Safa and Marwa, standing on Mount Arafat, resting overnight in Muzdalifah, staying two or three days in Mina, and throwing Jamarat) are all covered by HUER. Images have a pixel size of 1280x720 pixels and videos have a spatial resolution of 640x480 pixels, with average lengths of 20 seconds and 30 frames per second (Zawbaa & Aly, 2012). See Figure **5.1**.

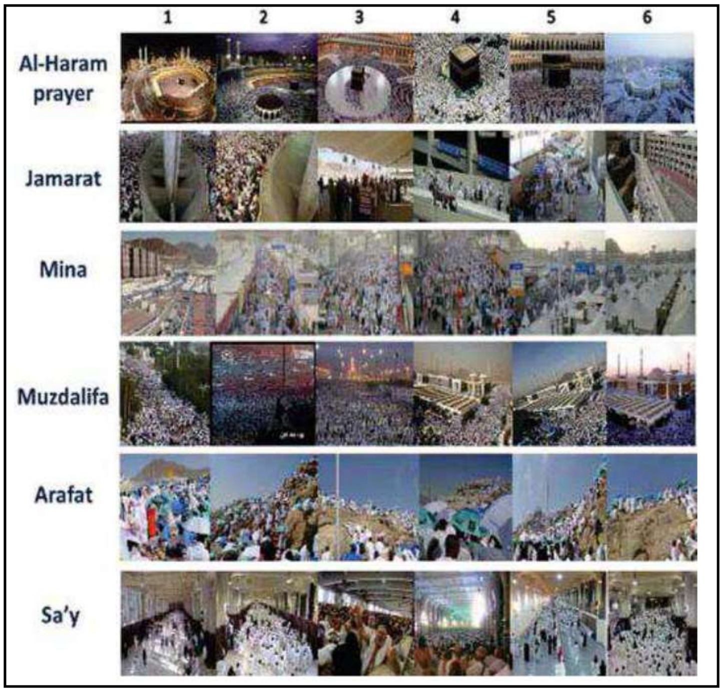


Figure 5.1: Recognition Datasets for Hajj and Umrah Pilgrim Events (Zawbaa & Aly, 2012)

We established our database from the combination of HEUR (Zawbaa & Aly, 2012) and and videos during 2012-2019 periods manually by ourself (Taha et al., 2021) as shown in the figure below:

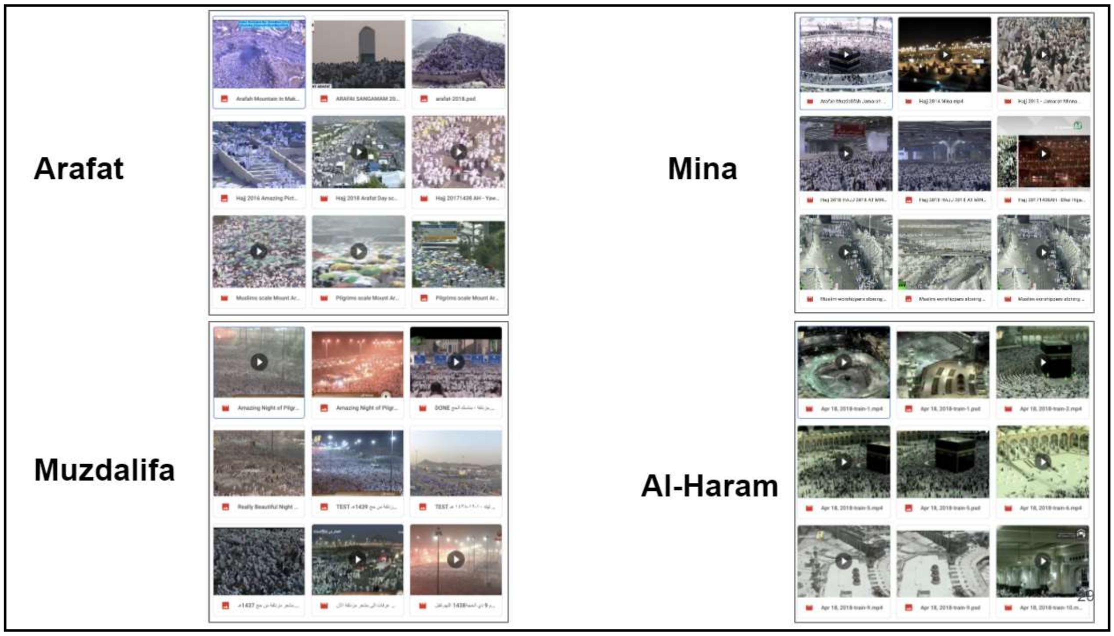


Figure 5.2: Videos during 2012-2019 four important places in Mecca (Taha et al., 2021)

In our dataset we focused on four important places of Mecca where they are Haram, Mina, Muzdalifa and Arafat. In chapter 1 we illustrated each of the places. We used 500 images for each places. We used 350 images for train, 75 images for validation and 75 images for test as shown in Figure 5.3.

350 images
75 images
75 images
Training
Validation
Test

Figure 5.3: Distribution of the proposed dataset

In the experiments, we assess the impact of image pre-processing on the place identification accuracy in our dataset. For this purpose, the performance of CNN with pre-processing (CNN+pre-processing), ANN with pre-processing (ANN+pre-processing) and SVM with pre-processing (SVM+pre-processing) are compared. In addition, we compare the classification performances when no image pre-processing is utilized for these three methods such as CNN without pre-processing (CNN), ANN without pre-processing (ANN) and SVM without pre-processing (SVM). In the experiments, the same evaluation settings are used such as the same training, validation and test images are utilized by all methods. In addition, the same image pre-processing methods are applied. The only difference is the use of different classifiers in CNN, ANN and SVM.

Proposed CNN architecture have nine convolution layers, eight pooling layers, and two fully connected layer. In each convolutional layer contains 32 filters, size of kernel 5x5, valid padding activation function is Relu6 as shown in Figure **4.12** (Taha et al., 2021). In each pooling layer, a window of 3x3 pixel is used with 2 stride. While training, stochastic gradient descent is used with the training learning rate of 0.01, 20 batch size and 500 epochs.

SVMs are called kernelized SVM due to their kernel that converts the input data space into a higher-dimensional space. We used (RBF) radial basis function kernel and error Correcting Output Code (ECOC) for multi class learning.

The ANN architecture's input layer is made up of a 256x256 vector . It has two hidden layers, the first of which contains 20 units and the second of which contains 10 units, with Sigmoid providing as the activation function.

Table 5.1: Implementation time on Intel(R) Core™ i7-4720HQ CPU@ 2.60GH processor (windows 10)

Methods	Time (second)
Median Filtering	1449.372672
Pre-processing	2356.706304
ANN	38.05874
SVM	12.27066
CNN	37405.874

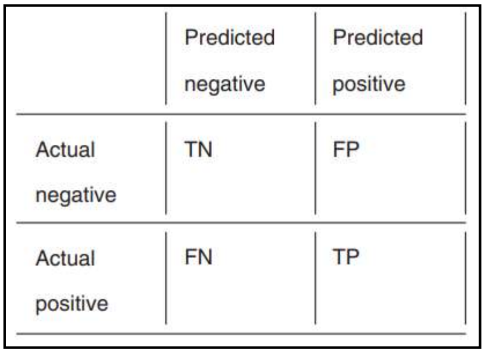
Table **5.1** shows the actual running time in seconds for each methods, on a matlab implementation on Intel(R) Core™ i7-4720HQ CPU@ 2.60GH processor (windows 10) or Nvidia K40 GPU for 2000 images during training phase.

5.2 Performance Metrics

Accuracy is one of the most widely utilized criteria while performing categorization. The following equation is used to estimate the accuracy of a simulation (via a confusion matrix).

$$\boldsymbol{Accuracy} = \frac{\boldsymbol{TN+TP}}{\boldsymbol{TN+FP+FN+TP}} \tag{5.1}$$

"TN" refers for True Negative, and it displays the percentage of correctly identified negative cases. Likewise, "TP" refers to True Positive, that denotes the number of correctly identified positive instances. The terms "FP" and "FN" stand for False Positive and False Negative values, respectively. "FP" stands for False Positive value, which would be the number of exact negative examples classified as positive, and "FN" stands for False Negative value, which is the multitude of real positive examples categorized as negative (Kulkarni et al., 2020). This information can also be utilized for displaying confusion matrix. Confusion matrix graphically displays how accurately a classification is predicted and it is a widely used metric. It can be used to solve problems involving binary and multiclass categorization (Branco et al., 2015). Confusion matrices are used to depict counts based on expected and true values shown in Figure 5.4. The result "TN" refers for True Negative, and it displays the percentage of correctly identified negative cases. Likewise, "TP" refers to True Positive, that denotes the number of correctly identified positive instances The terms "FP" and "FN" stand for False Positive and False Negative values, respectively. "FP" stands for False Positive value, which would be the number of exact negative examples classified as positive, and "FN" stands for False Negative value, which is the multitude of real positive examples categorized as negative (Kulkarni et al., 2020).



	Predicted negative	Predicted positive
Actual negative	TN	FP
Actual positive	FN	TP

Figure 5.4: Confusion matrix

In the following subsections other metrics that are used in our proposed work are illustrated.

5.2.1 Precision , Recall and Specificity

Precision and recall are two extensively employed and successful classification measures. Precision indicates how well the algorithm predicts positive outcomes. The positive predictive value is another name for it. The sensitivity of a model is also called as recall, and it is used to quantify the intensity of a model's ability to identify favorable results (Kulkarni et al., 2020).

The precision and recall equations can be found below:

$$\boldsymbol{Precision = \frac{TP}{TP+FP}} \qquad \textbf{(5.2)}$$

$$\boldsymbol{Recall = \frac{TP}{TP+FN}} \qquad \textbf{(5.3)}$$

Precision evaluates how much a positive class item in the dataset is detected as a positive class example by the classifier, whereas recall evaluates how much a positive class occurrence in the dataset is anticipated as a positive class example by the classifier (Kulkarni et al., 2020).

The proportion of negatives that are correctly identified (i.e., the proportion of individuals who do not have the condition (unaffected) who are accurately identified as not having the ailment) is referred to as specificity (Trevethan, 2017).

Specificity is a measurement of a test's ability to detect true negatives. The percentage, or ratio, of genuine negatives out of all the samples that do not have the condition is known as specificity, selectivity, or true negative rate (true negatives and false positives) (Trevethan, 2017).

$$\boldsymbol{Specificity = \frac{TN}{FP+TN}} \tag{5.4}$$

5.2.2 F-measure and G-mean

When only the positive class's performance is taken into account, two metrics are crucial: True Positive Rate (TP_{rate}) and Positive Predictive Value (PPV). True Positive Rate (TP_{rate}) is defined as recall (R) in information retrieval, as indicated in equation (5.2).

Precision (P) denotes the percentage of relevant objects recognized in equation (5.3) and is used to calculate positive predictive value (Sun et al., 2009).

F-measure (F) is the integration of these two measures as an average (Sun et al., 2009):

$$\boldsymbol{F - measure = \frac{2RP}{R+P}} \tag{5.5}$$

Both True Positive Rate (TP_{rate}) and True Negative Rate (TN_{rate}) are supposed to be significant concurrently when it comes to both classes' performance. (Kubat et al., 1998) preferred G-mean defined as follow:

$$\boldsymbol{G - mean = \sqrt{TP_{rate}.TN_{rate}}} \tag{5.6}$$

G-mean assesses how well a learning algorithm performs in both of these categories (Sun et al., 2009).

To evaluate the performance of machine learning based classification algorithm, statistical performance metrics such as precision, recall, specificity, f1-measure, negative predictive values are used apart from the accuracy. These metrics reveal more details on how well the classifier is doing than just using accuracy as a measure, especially when the test data is unbalanced. The most common way to compute these metrics is through the construction of confusion matrix, which is a form of tabular representation of how the classifier correctly or incorrectly classify a set of test data. Usually in confusion matrix, the column of the matrix represents the classifier predictive scores of a particular class while its row gives the actual classes of the test data. Correctly classified samples are those samples in the diagonal elements of the matrix and misclassification are in the non-diagonal elements of the confusion matrix (Kulkarni et al., 2020).

For multi-classification problem, metrics are computed for each class separately and their average will give the overall performance of the classifier. For instance, for four classes (Haram, Arafat, Mina and Muzdalifa) classification problem, to estimate metrics (e.g. precision and recall) for class Haram, class Haram will be assigned a Positive class label while the remaining classes (Arafat, Mina and Muzdalifa) will be assigned Negative class label. For class Arafat, Arafat will take the Positive class and Haram, Mina and Muzdalifa will be the Negative classes and so on. In this way the four statistical quantities i.e., True Positive (TP), False Positive (FP), True negative (TN) and False Negative (FN) samples are estimated which are used to compute the performance metrics for each class. Then for each metric, average of the classes is taken.

5.3 Results With Image Pre-Processing

First, we show the evaluations results when image pre-processing is employed as shown in Figure 5.5.

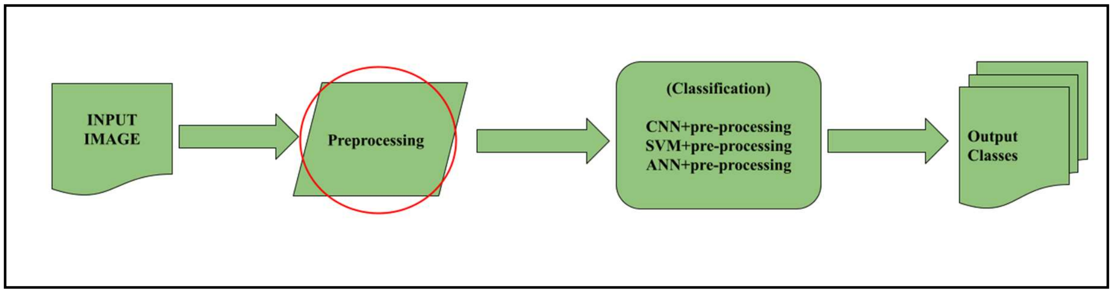


Figure 5.5: General architecture with pre-processing

Specificity, recall, precision, Gmean, f-measure, and accuracy have all been used to verify the performance of the proposed task. The advanced study's overall categorization presentation is differentiated using two current methodologies. On the x-axis, the normalized value of the number of iterations is plotted. The score of the performance measure grew as the number of iterations increasing. Advanced work has a maximum specificity of 98 percent, while subsist work has a specificity of 80 percent and 92 percent, respectively (Figure 5.6) (Taha et al., 2021).

In the figures 5.6 to 5.11, we demonstrate how we tested different methods with varying numbers of iterations during training and test the model for different number of iterations. It is important to note that the scale of iterations is 100 to 1. It means on the figures, x-axis 1 means 100 iteration during training.

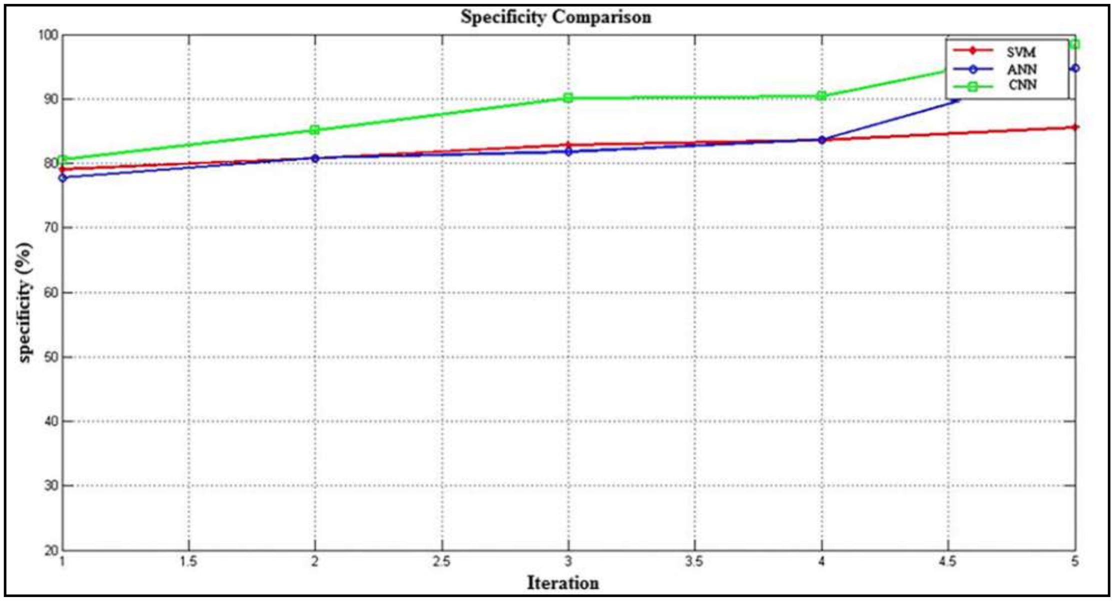


Figure 5.6: Specificity of the methods

The advanced work's "Recall" is collated with prior work until the number of iterations reaches four. When compared to previous techniques, the highest recall of the suggested method has not improved significantly, but it does vary linearly from iteration 3 to 5 (Taha et al., 2021). (Figure 5.7).

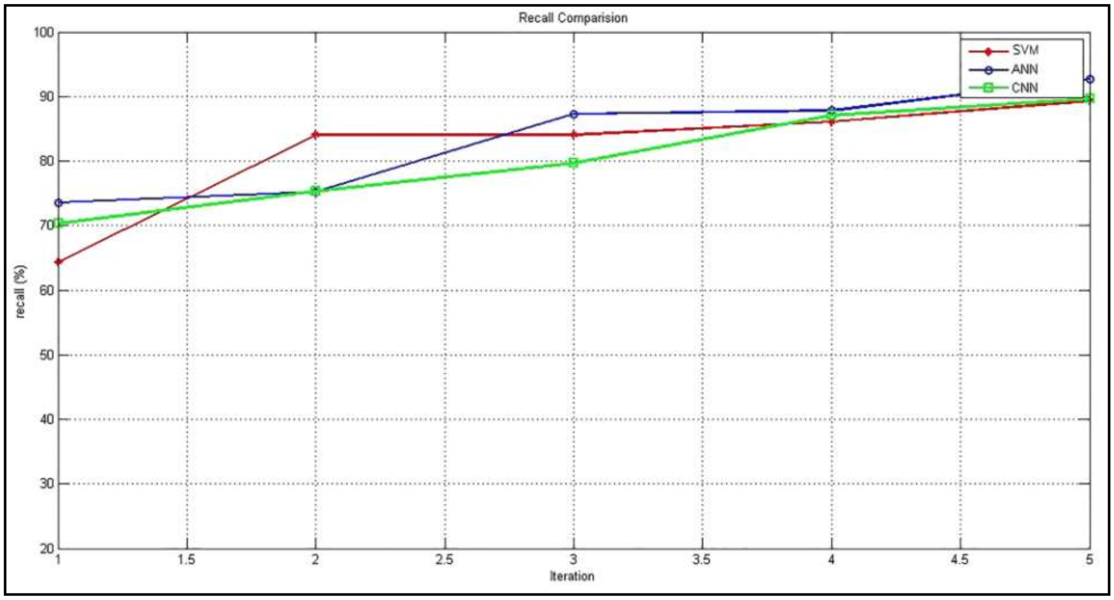


Figure 5.7: Recall of the methods

The precision value, on the other hand, has been found to be enhanced for all iterations, reaching a high of 90%. (Figure 5.8).

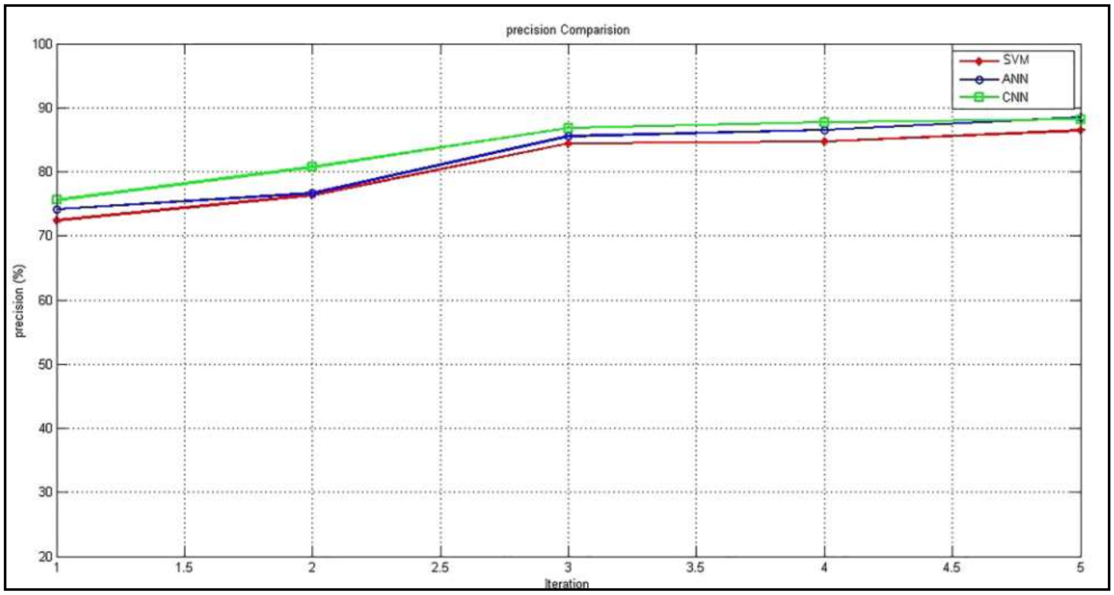


Figure 5.8: Precision of the methods

Remarkably, the Gmean score for the suggested technique hits 100% when the iteration is increased above 4. The Gmean value is maintained at a constant value along the iteration line from 3 to 4. (Figure 5.9).

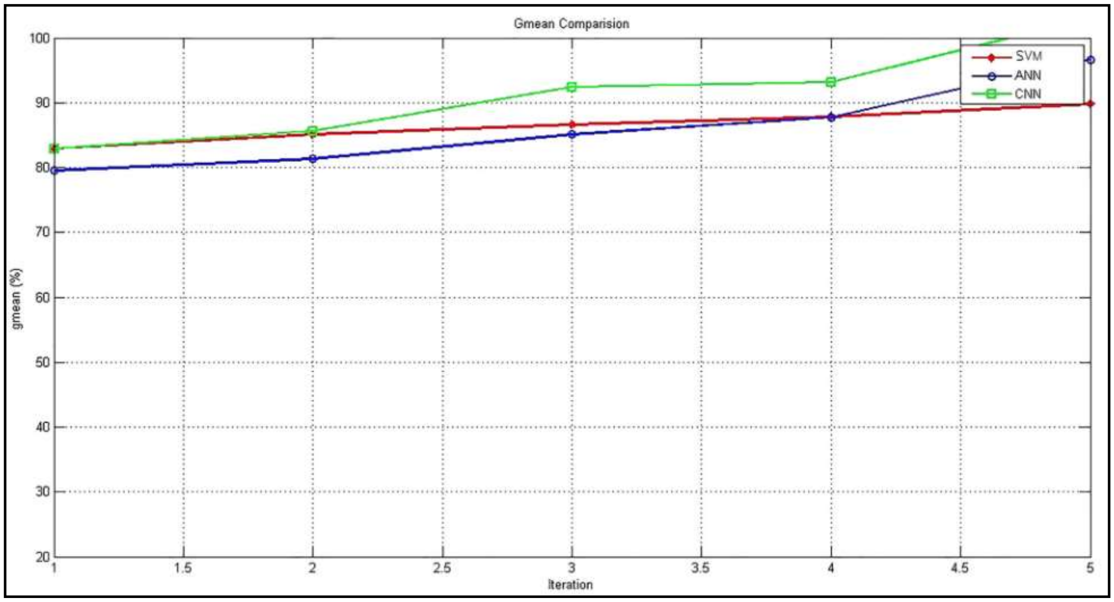


Figure 5.9: Gmean of the methods

Until the iteration reaches 3, the f-measure productivity has been steadily increasing from 73 percent to 88 percent. Since the number of iterations grows, the performance does not meet expectations, as it falls somewhere between the values of the previous approaches. (Figure 5.10).

Finally, the proposed method's accuracy performance has balanced the aforementioned drawbacks (Figure 5.10 and Figure 5.11). For the whole cycle of iterations, the performance is continuously increased over the existing approaches. The proposed method achieves a maximum accuracy of 90 percent, whereas traditional algorithms only achieve 84 percent and 80.5 percent accuracy.

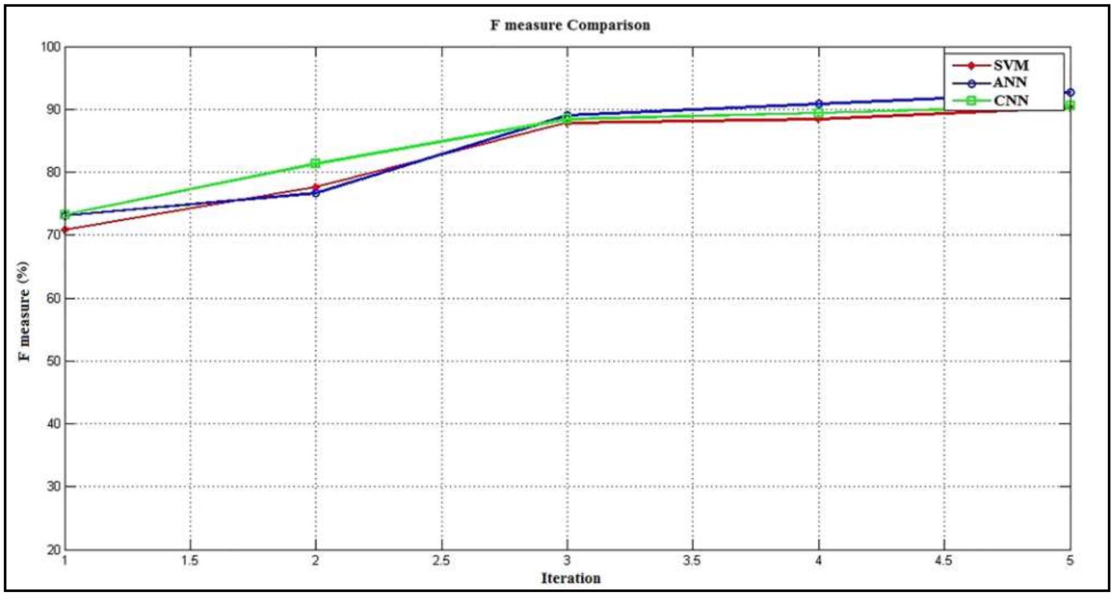


Figure 5.10: F-measure of the methods

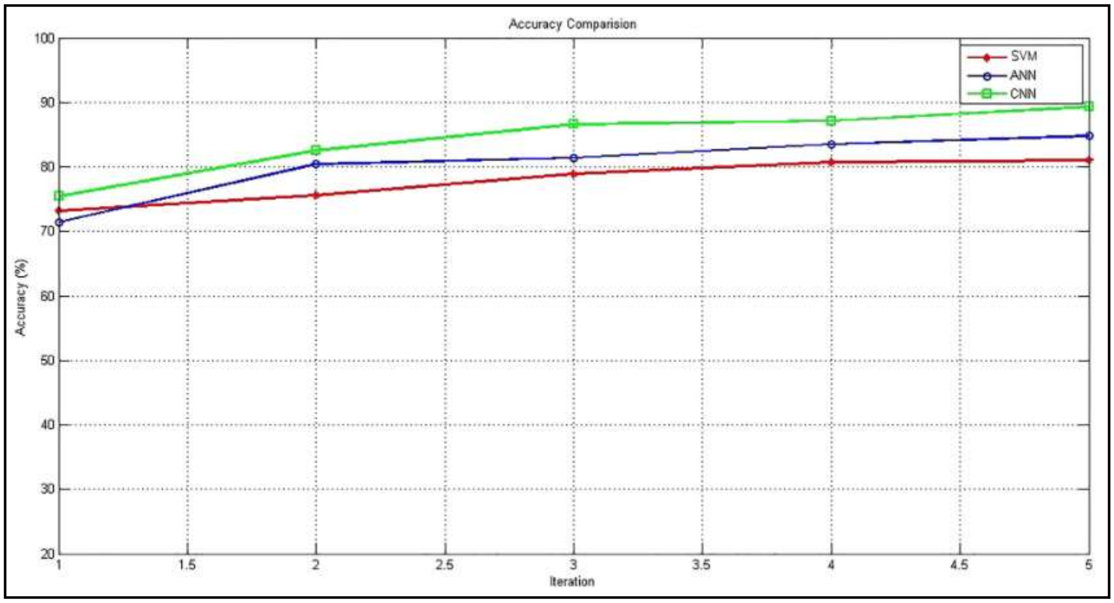


Figure 5.11: Accuracy of the methods

In Table **5.2**, results of all methods using image pre-processing are combined. It is observed that CNN+pre-processing achieves significantly better performance especially for specificity, g-mean and accuracy metrics comparing to ANN+pre-processing and SVM+pre-processing.

Table 5.2: Results with pre-processing

Methods	CNN+pre-processing	ANN+pre-processing	SVM+pre-processing
Specificity	98%	92%	80%
Recall	90%	92%	90%
Precision	90%	90%	90%
G-mean	100%	97%	90%
Accuracy	90%	84%	80.50%

5.4 Results Without Image Pre-Processing

In this section we show the evaluations results without pre-processing as shown in Figure 5.12.

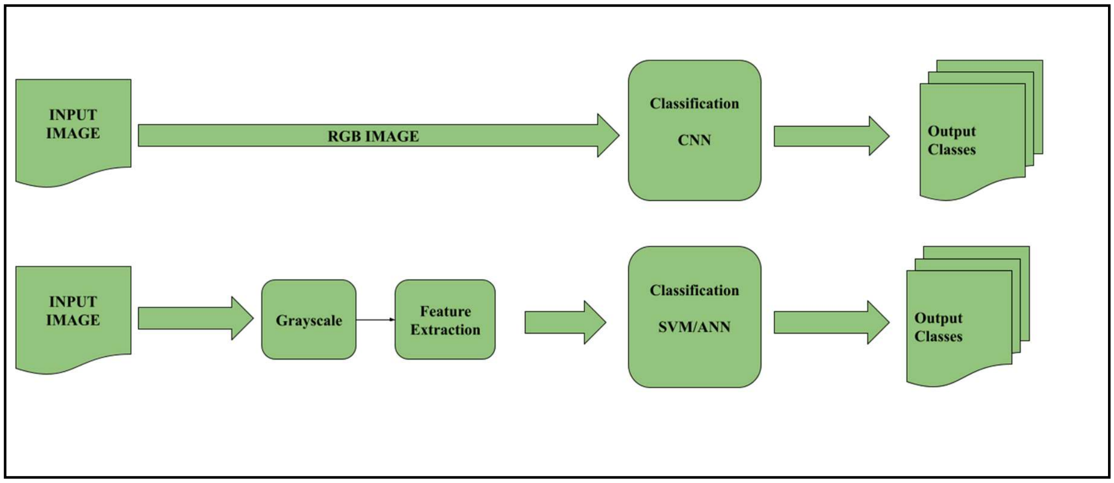


Figure 5.12: General architecture without preprocessing

In Table **5.3** shows the results without applying pre-processing methods. The performance decreased for all methods especially CNN. Because CNN works fine with large number of datasets (Goodfellow et al., 2016).

Table 5.3: Results without preprocessing

Methods	CNN	ANN	SVM
Specificity	67%	72%	69%
Recall	62%	72%	73%
Precision	65%	75%	73%
G-mean	71%	76%	74%
Accuracy	63%	73%	71%

5.5 Discussion of Results With and Without Image Pre-Processing

In this section, we demonstrate the differences in accuracy among methods that include and do not include pre-processing. As shown in Table 5.4, the accuracy enhances when pre-processing techniques applied.

Table 5.4: Comparing results with and without pre-processing

Methods	Accuracy with pre-processing	Accuracy without pre processing
CNN	90%	63%
ANN	84%	73%
SVM	80.50%	71.50%

5.6 Comparison with Other Deep Learning based Method for Place Identification

We compared our method with a known CNN based place recognition method (J. Zhu et al., 2018). (J. Zhu et al., 2018) uses VGG16 network to recognize 365 different places. The network has 16-weight-layer architecture, with 13 convolutional layers and three fully-connected layers. Because the Places dataset comprises over 10 million photos divided across 365 distinct scene types, the last fully-connected layer's dimensions is 365. These 13 convolutional layers are organized into five sections, with every section having the same data dimension for each layer. After each section, there is a max-pooling layer that runs across a 2x2 pixel window with stride 2. We modified the last layer of VGG16 network's dimension to 4 because our dataset focuses on 4 important places in Mecca.

Table 5.5: The proposed work is compared with a deep learning based related work

Methods	Accuracy
Proposed work with pre-processing	90%
Proposed work without pre-processing	63%
(J. Zhu et al., 2018)	61%

We applied our dataset to the CNN network of (J. Zhu et al., 2018) . In Table 5.5, we realized that our method, pre-processing with CNN, has a better result. (J. Zhu et al., 2018) has good results on their own dataset that is composed of one million images. Our dataset has smaller size in comparison to their dataset size. Since our dataset's size is small, the performance of a complex CNN architecture, such as VGG16, is low.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

In this thesis, a technique has been introduced to address the problems with detection and checking a specific pilgrim location in an extremely sensitive and crowded environment of holy Mecca. A novel image pre-processing algorithm is presented to improve the performance of place recognition in Mecca. When images are input to the convolutional neural network (CNN) after the proposed pre-processing stage, it achieves the best performances. Extensive evaluation is conducted to illustrate the effectiveness of the proposed framework. The proposed pre-processing stage is also experimented with artificial neural networks (ANN) and support machine classifier (SVM), and compared to the other deep learning based related work. The suggested technique, which combines the proposed pre-processing with the CNN classifier, is straightforward and offers significantly smarter and more functional outcomes in a more accurate and effective sense by alleviating the challenges of the worshipper, while also assisting in reducing the load placed on the advisory committee during the holy Hajj pilgrimage.

In this work, we categorize four major religious sites in Mecca, including the Haram, Muzdalifa, Mina, and Arafat. In the future, we can expand the number of places that can be classified, such as Jamarat, Sa'y, and Miqat, to assist worshippers in finding their locations correctly.

This work can be optimized by using extra sensor such as Mobile Phone Antenna GPS module. Data captured by GPS module can be processed and integrated to the algorithm presented in this thesis for place recognition and localization. Different data fusion algorithms can be studied and experimented to reduce error and optimize results.

REFERENCES

Abbod, M. F., Catto, J. W. F., Linkens, D. A., & Hamdy, F. C. (2007). Application of Artificial Intelligence to the Management of Urological Cancer. In *Journal of Urology* (Vol. 178, Issue 4, pp. 1150–1156). https://doi.org/10.1016/j.juro.2007.05.122

Abiodun, O. I., Jantan, A., Omolara, A. E., Dada, K. V., Mohamed, N. A. E., & Arshad, H. (2018). State-of-the-art in artificial neural network applications: A survey. In *Heliyon* (Vol. 4, Issue 11, p. e00938). Elsevier Ltd. https://doi.org/10.1016/j.heliyon.2018.e00938

Abirami, S., & Chitra, P. (2020). Energy-efficient edge based real-time healthcare support system. *Advances in Computers*, *117*(1), 339–368. https://doi.org/10.1016/BS.ADCOM.2019.09.007

Achille, A., & Soatto, S. (2016). Information Dropout: Learning Optimal Representations Through Noisy Computation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *40*(12), 2897–2905. http://arxiv.org/abs/1611.01353

Ahmad, J., Farman, H., & Jan, Z. (2019). Deep Learning Methods and Applications. In *SpringerBriefs in Computer Science* (pp. 31–42). Springer. https://doi.org/10.1007/978-981-13-3459-7_3

Alpaydin, E. (2020). *Introduction to Machine Learning* (Fourth Edition). https://mitpress.mit.edu/books/introduction-machine-learning-fourth-edition

Anaraki, J. R., & Eftekhari, M. (2013). Rough set based feature selection: A review. *IKT 2013 - 2013 5th Conference on Information and Knowledge Technology*, 301–306. https://doi.org/10.1109/IKT.2013.6620083

Arias-Castro, E., & Donoho, D. L. (2009). Does median filtering truly preserve edges better than linear filtering? *The Annals of Statistics*, *37*(3), 1172–1206. https://doi.org/10.1214/08-AOS604

Babenko, A., & Lempitsky, V. (2015). *Aggregating Deep Convolutional Features for Image Retrieval*. http://arxiv.org/abs/1510.07493

Bae, J. W., Shin, K., Lee, H. R., Lee, H. J., Lee, T., Kim, C. H., Cha, W. C., Kim, G. W., & Moon, I. C. (2018). Evaluation of Disaster Response System Using Agent-Based Model with Geospatial and Medical Details. *IEEE Transactions on Systems, Man, and*

Cybernetics: Systems, *48*(9), 1454–1469. https://doi.org/10.1109/TSMC.2017.2671340

Baldi, P. (2012). *Autoencoders, Unsupervised Learning, and Deep Architectures* (Vol. 27). JMLR Workshop and Conference Proceedings. http://proceedings.mlr.press/v27/baldi12a.html

Banerjee, M., Mitra, S., & Anand, A. (2006). Feature Selection Using Rough Sets. In *Multi-Objective Machine Learning* (pp. 3–20). Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-33019-4_1

Bannay, F., & Guillaume, R. (2014). Towards a Transparent Deliberation Protocol Inspired from Supply Chain Collaborative Planning. *Communications in Computer and Information Science*, *443 CCIS*(PART 2), 335–344. https://doi.org/10.1007/978-3-319-08855-6_34

Bao, L., Sun, X., Chen, Y., Man, G., & Shao, H. (2018). Restricted Boltzmann machine-assisted estimation of distribution algorithm for complex problems. *Complexity*, *2018*. https://doi.org/10.1155/2018/2609014

Ben-Hur, A. (2008). Support vector clustering. *Scholarpedia*, *3*(6), 5187. https://doi.org/10.4249/scholarpedia.5187

Bethanney Janney, J., Roslin, S. E., & Kumar, S. K. (2020). Analysis of skin lesions using machine learning techniques. In *Computational Intelligence and Its Applications in Healthcare* (pp. 73–90). Elsevier. https://doi.org/10.1016/b978-0-12-820604-1.00006-6

Bishop, C. (2006). *Pattern Recognition and Machine Learning* . Springer. https://www.springer.com/gp/book/9780387310732#aboutAuthors

Bolón-Canedo, V., Sánchez-Maroño, N., & Alonso-Betanzos, A. (2015). *A Critical Review of Feature Selection Methods* (pp. 29–60). Springer, Cham. https://doi.org/10.1007/978-3-319-21858-8_3

Branco, P., Torgo, L., & Ribeiro, R. (2015). *A Survey of Predictive Modelling under Imbalanced Distributions*. http://arxiv.org/abs/1505.01658

Chidambaram, S., & Srinivasagan, K. G. (2018). Performance evaluation of support vector machine classification approaches in data mining. *Cluster Computing 2018 22:1*, *22*(1), 189–196. https://doi.org/10.1007/S10586-018-2036-Z

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297. https://doi.org/10.1007/bf00994018 da Silva, E. A. B., & Mendonca, G. v. (2005). Digital Image Processing. *The Electrical Engineering Handbook*, 891–910. https://doi.org/10.1016/B978-012170960-0/50064-5

Dawson, C. W., & Wilby, R. (1998). Une approche de la modélisation pluie-deblt par ies réseaux neuronaux artificiels. *Hydrological Sciences Journal*, *43*(1), 47–66. https://doi.org/10.1080/02626669809492102

Devarajan, G., Aatre, V. K., & Sridhar, C. S. (1990). Analysis of median filter. *Proceedings of 16th Annual Convention and Exhibition of the IEEE in India, ACE 1990*, 274–276. https://doi.org/10.1109/ACE.1990.762694

Direkoglu, C. (2020). Abnormal Crowd Behavior Detection Using Motion Information Images and Convolutional Neural Networks, *IEEE Access*, vol. 8, pp. 80408-80416, 2020, doi: 10.1109/ACCESS.2020.2990355.

Direkoglu, C., Sah. M., & O'Connor, N. E. (2017). Abnormal crowd behavior detection using novel optical flow-based features," *IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2017, pp. 1-6, doi: 10.1109/AVSS.2017.8078503.

Djuris, J., Ibric, S., & Djuric, Z. (2013). Neural computing in pharmaceutical products and process development. In *Computer-Aided Applications in Pharmaceutical Technology* (pp. 91–175). Elsevier. https://doi.org/10.1533/9781908818324.91

Farley, B. G., & Clark, W. A. (1954). Simulation of self-organizing systems by digital computer. *IRE Professional Group on Information Theory*, *4*(4), 76–84. https://doi.org/10.1109/TIT.1954.1057468

Gabbouj, M., Coyle, E. J., & Gallagher, N. C. (1992). An overview of median and stack filtering. *Circuits Systems and Signal Processing*, *11*(1), 7–45. https://doi.org/10.1007/BF01189220

Gonzalez, R. C., & Woods, R. E. (2002). *Digital image processing*. https://www.worldcat.org/title/digital-image-processing/oclc/48944550?page=citation

Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). *Deep learning*. https://doi.org/10.4258/hir.2016.22.4.351

Haralick, R. M., & Shapiro, L. G. (1985). Image segmentation techniques. *Computer Vision, Graphics, and Image Processing*, *29*(1), 100–132. https://doi.org/10.1016/S0734-189X(85)90153-7

Haralick, Robert M. (1979). Statistical and structural approaches to texture. *Proceedings of the IEEE*, *67*(5), 786–804. https://doi.org/10.1109/PROC.1979.11328

Haralick, Robert M., Dinstein, I., & Shanmugam, K. (1973). Textural Features for Image Classification. *IEEE Transactions on Systems, Man and Cybernetics*, *SMC-3*(6), 610–621. https://doi.org/10.1109/TSMC.1973.4309314

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning*. Springer New York. https://doi.org/10.1007/978-0-387-84858-7

Hatirnaz, E., Sah, M. & Direkoglu, C (2020). A novel framework and concept-based semantic search Interface for abnormal crowd behaviour analysis in surveillance videos. *Multimedia Tools and Applications,* 79, 17579–17617 (2020). https://doi.org/10.1007/s11042-020-08659-2

Huang, J., Dong, Q., Gong, S., & Zhu, X. (2019). *Unsupervised Deep Learning by Neighbourhood Discovery*. https://github.com/raymond-sci/AND.

Huh, M.-H. (2015). Kernel-Trick Regression and Classification. *Communications for Statistical Applications and Methods*, *22*(2), 201–207. https://doi.org/10.5351/csam.2015.22.2.201

Ilyas, Q. M. (2013). A NetLogo Model for Ramy al-Jamarat in Hajj. *Undefined*.

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *32nd International Conference on Machine Learning, ICML 2015*, *1*, 448–456. https://arxiv.org/abs/1502.03167v3

J. Russell, S., & Norvig, P. (2016). *Artificial Intelligence: A Modern Approach*. Pearson Education. https://b-ok.asia/book/3704484/55dad1

Jain, A. K. (2010a). Data clustering: 50 years beyond K-means. *Pattern Recognition Letters*, *31*(8), 651–666. https://doi.org/10.1016/j.patrec.2009.09.011

Jain, A. K. (2010b). Data clustering: 50 years beyond K-means. *Pattern Recognition Letters*, *31*(8), 651–666. https://doi.org/10.1016/j.patrec.2009.09.011

Khiyari, H. el, & Wechsler, H. (2016). Face Recognition across Time Lapse Using Convolutional Neural Networks. *Journal of Information Security*, *07*(03), 141–151. https://doi.org/10.4236/JIS.2016.73010

Kleene, S. C. (2016). Representation of Events in Nerve Nets and Finite Automata. In *Automata Studies. (AM-34)* (pp. 3–42). Princeton University Press. https://doi.org/10.1515/9781400882618-002

Koza, J. R., Bennett, F. H., Andre, D., & Keane, M. A. (1996). Automated Design of Both the Topology and Sizing of Analog Electrical Circuits Using Genetic Programming. In *Artificial Intelligence in Design '96* (pp. 151–170). Springer Netherlands. https://doi.org/10.1007/978-94-009-0279-4_9

Kubat, M., Holte, R. C., & Matwin, S. (1998). Machine learning for the detection of oil spills in satellite radar images. *Machine Learning*, *30*(2–3), 195–215. https://doi.org/10.1023/a:1007452223027

Kulkarni, A., Chong, D., & Batarseh, F. A. (2020). Foundations of data imbalance and solutions for a data democracy. In *Data Democracy: At the Nexus of Artificial Intelligence, Software Development, and Knowledge Engineering* (pp. 83–106). Elsevier. https://doi.org/10.1016/B978-0-12-818366-3.00005-8

Kumar, G., & Bhatia, P. K. (2014). A detailed review of feature extraction in image processing systems. *International Conference on Advanced Computing and Communication Technologies, ACCT*, 5–12. https://doi.org/10.1109/ACCT.2014.74

Kurfess, F. J. (2003). Artificial Intelligence. In *Encyclopedia of Physical Science and Technology* (pp. 609–629). Elsevier. https://doi.org/10.1016/B0-12-227410-5/00027-2

LeCun, Y. A., Bottou, L., Orr, G. B., & Müller, K.-R. (2012). Efficient BackProp. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *7700 LECTURE NO*, 9–48. https://doi.org/10.1007/978-3-642-35289-8_3

Lin, T. Y., Cui, Y., Belongie, S., & Hays, J. (2015). Learning deep representations for ground-to-aerial geolocalization. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *07-12-June-2015*, 5007–5015. https://doi.org/10.1109/CVPR.2015.7299135

Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., van der Laak, J. A. W. M., van Ginneken, B., & Sánchez, C. I. (2017). A Survey on Deep Learning in Medical Image Analysis. *Medical Image Analysis*, *42*, 60–88. https://doi.org/10.1016/j.media.2017.07.005

Lorraine, J., & Duvenaud, D. (2018). Stochastic Hyperparameter Optimization through Hypernetworks. *ArXiv*. http://arxiv.org/abs/1802.09419

Lu, Y., Xie, S., & Wu, S. (2019). Exploring Competitive Features Using Deep Convolutional Neural Network for Finger Vein Recognition. *IEEE Access*, *7*, 35113–35123. https://doi.org/10.1109/ACCESS.2019.2902429

Madhavan, S., & Jones, M. T. (2017, November 8). *Deep learning architectures* . https://developer.ibm.com/technologies/artificial-intelligence/articles/cc-machine-learning-deep-learning-architectures

Maity, A. (2016). *Supervised Classification of RADARSAT-2 Polarimetric Data for Different Land Features*. http://arxiv.org/abs/1608.00501

Mall, P. K., Singh, P. K., & Yadav, D. (2019, December 1). GLCM based feature extraction and medical X-RAY image classification using machine learning techniques. *2019 IEEE Conference on Information and Communication Technology, CICT 2019*. https://doi.org/10.1109/CICT48419.2019.9066263

Mathew, A., Amudha, P., & Sivakumari, S. (2021). Deep learning techniques: an overview. *Advances in Intelligent Systems and Computing*, *1141*, 599–608. https://doi.org/10.1007/978-981-15-3383-9_54

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, *5*(4), 115–133. https://doi.org/10.1007/BF02478259

Mitchell, T. (1997). *Machine Learning* . Hill McGraw. http://www.cs.cmu.edu/~tom/mlbook.html

Mohri, M., Rostamizadeh, A., & Talwalkar, A. S. (2012). Foundations of Machine Learning. *Undefined*.

Mostafa, B., El-Attar, N., Abd-Elhafeez, S., & Awad, W. (2020). Machine and Deep Learning Approaches in Genome: Review Article. *Alfarama Journal of Basic & Applied Sciences*, *0*(0), 0–0. https://doi.org/10.21608/ajbas.2020.34160.1023

Mukhopadhyay, S. (2011). Artificial neural network applications in textile composites. *Soft Computing in Textile Engineering*, 329–349. https://doi.org/10.1533/9780857090812.4.329

N. Mohammed, M. (1996). *Hajj & 'Umrah from A to Z*. https://openlibrary.org/books/OL806387M/Hajj_%CA%BBUmrah

Otsu, N. (1979). THRESHOLD SELECTION METHOD FROM GRAY-LEVEL HISTOGRAMS. *IEEE Trans Syst Man Cybern*, *SMC-9*(1), 62–66. https://doi.org/10.1109/TSMC.1979.4310076

Panigrahi, A., Chen, Y., & Kuo, C.-C. J. (2018). Analysis on Gradient Propagation in Batch Normalized Residual Networks. *ArXiv*. http://arxiv.org/abs/1812.00342

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., … Chintala, S. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. *Advances in Neural Information Processing Systems*, *32*. http://arxiv.org/abs/1912.01703

Pradhan, S., Ward, W., Hacioglu, K., Martin, J. H., & Jurafsky, D. (2004). *Shallow Semantic Parsing using Support Vector Machines* *. http://www.cis.upenn.edu/

Rafael C. Gonzalez, & Richard E. Woods. (2007). *Digital Image Processing* . https://www.scirp.org/(S(351jmbntvnsjt1aadkposzje))/reference/ReferencesPapers.aspx?ReferenceID=768956

Rochester, N., Holland, J. H., Haibt, L. H., & Duda, W. L. (1956). Tests on a cell assembly theory of the action of the brain, using a large digital computer. *IRE Transactions on Information Theory*, *2*(3), 80–93. https://doi.org/10.1109/TIT.1956.1056810

Schmidhuber, J. (2015). Deep Learning in neural networks: An overview. In *Neural Networks* (Vol. 61, pp. 85–117). Elsevier Ltd. https://doi.org/10.1016/j.neunet.2014.09.003

Seal, A., Bhattacharjee, D., & Nasipuri, M. (2018). Predictive and probabilistic model for cancer detection using computer tomography images. *Multimedia Tools and Applications*, *77*(3), 3991–4010. https://doi.org/10.1007/s11042-017-4405-7

Serra, J. (1994). Morphological filtering: An overview. *Signal Processing*, *38*(1), 3–11. https://doi.org/10.1016/0165-1684(94)90052-3

Sezgin, M., & Sankur, B. (2004a). Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging*, *13*(1), 146–165. https://doi.org/10.1117/1.1631315

Sezgin, M., & Sankur, B. (2004b). Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging*, *13*(1), 146–165. https://doi.org/10.1117/1.1631315

Shalev-Shwartz, S., & Ben-David, S. (2013). Understanding machine learning: From theory to algorithms. In *Understanding Machine Learning: From Theory to Algorithms* (Vol. 9781107057135). Cambridge University Press. https://doi.org/10.1017/CBO9781107298019

Shao, J., Loy, C. C., Kang, K., & Wang, X. (2016). Slicing convolutional neural network for crowd video understanding. *Proceedings of the IEEE Computer Society*

Conference on Computer Vision and Pattern Recognition, *2016-December*, 5620–5628. https://doi.org/10.1109/CVPR.2016.606

Shapiro Linda G., S. G. C. (2001). *Computer Vision* (illustrated, Vol. 580). Prentice Hall. https://books.google.iq/books/about/Computer_Vision.html?id=FftDAQAAIAAJ&redir_esc=y

Sherstinsky, A. (2018). Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network. *Physica D: Nonlinear Phenomena*, *404*. https://doi.org/10.1016/j.physd.2019.132306

Solomon Chris, B. T. (2011). *Fundamentals of Digital Image Processing: A Practical Approach with Examples in Matlab | Chris Solomon, Toby Breckon | download* (1st ed.). Wiley-Blackwell. https://2lib.org/book/1054860/6c258c

Srivastav, S. (2020). *Artificial Intelligence, Machine Learning, and Deep Learning. What's the Real Difference? | by Sushant Srivastav | The Startup | Medium*. https://medium.com/swlh/artificial-intelligence-machine-learning-and-deep-learning-whats-the-real-difference-94fe7e528097

Statnikov, A., Aliferis, C. F., Tsamardinos, I., Hardin, D., & Levy, S. (2005). A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis. *Bioinformatics*, *21*(5), 631–643. https://doi.org/10.1093/bioinformatics/bti033

Sudeep, K. S., & Pal, K. K. (2017). Preprocessing for image classification by convolutional neural networks. *2016 IEEE International Conference on Recent Trends in Electronics, Information and Communication Technology, RTEICT 2016 - Proceedings*, 1778–1781. https://doi.org/10.1109/RTEICT.2016.7808140

Sun, Y., Wong, A. K. C., & Kamel, M. S. (2009). Classification of imbalanced data: A review. *International Journal of Pattern Recognition and Artificial Intelligence*, *23*(4), 687–719. https://doi.org/10.1142/S0218001409007326

Taha, M. A., Direkoglu, M. S., & Direkoglu, C. (2021). Deep neural network-based detection of pilgrims location in Holy Makkah. *International Journal of Communication Systems*, e4792. https://doi.org/10.1002/dac.4792

Taha, M. A., Şah, M., & Direkoğlu, C. (2020). *Review of Place Recognition Approaches: Traditional and Deep Learning Methods*. 183–191. https://doi.org/10.1007/978-3-030-64058-3_22

TAKAHASHI, T. (2010). *Statistical max pooling with deep learning*. https://patents.patsnap.com/v/US10013644-statistical-max-pooling-with-deep-learning.html

Teuwen, J., & Moriakov, N. (2019). Convolutional neural networks. In *Handbook of Medical Image Computing and Computer Assisted Intervention* (pp. 481–501). Elsevier. https://doi.org/10.1016/B978-0-12-816176-0.00025-9

Teuwen, J., & Moriakov, N. (2020). Convolutional neural networks. *Handbook of Medical Image Computing and Computer Assisted Intervention*, 481–501. https://doi.org/10.1016/B978-0-12-816176-0.00025-9

Thirumala, K., Pal, S., Jain, T., & Umarikar, A. C. (2019). A classification method for multiple power quality disturbances using EWT based adaptive filtering and multiclass SVM. *Neurocomputing*, *334*, 265–274. https://doi.org/10.1016/j.neucom.2019.01.038

Trevethan, R. (2017). Sensitivity, Specificity, and Predictive Values: Foundations, Pliabilities, and Pitfalls in Research and Practice. *Frontiers in Public Health*, *5*, 307. https://doi.org/10.3389/fpubh.2017.00307

van Otterlo, M., & Wiering, M. (2012). Reinforcement learning and markov decision processes. In *Adaptation, Learning, and Optimization* (Vol. 12, pp. 3–42). Springer Verlag. https://doi.org/10.1007/978-3-642-27645-3_1

Wang, X., & Loy, C. C. (2017). Deep Learning for Scene-Independent Crowd Analysis. In *Group and Crowd Behavior for Computer Vision* (pp. 209–252). Elsevier Inc. https://doi.org/10.1016/B978-0-12-809276-7.00012-6

Yang, X.-S. (2019). Neural networks and deep learning. *Introduction to Algorithms for Data Mining and Machine Learning*, 139–161. https://doi.org/10.1016/B978-0-12-817216-2.00015-6

Ye, W., Cheng, J., Yang, F., & Xu, Y. (2019). Two-Stream Convolutional Network for Improving Activity Recognition Using Convolutional Long Short-Term Memory Networks. *IEEE Access*, *7*, 67772–67780. https://doi.org/10.1109/ACCESS.2019.2918808

Zaman, K., Sah, M., & Direkoglu, C. (2020). Classification of Harmful Noise Signals for Hearing Aid Applications using Spectrogram Images and Convolutional Neural Networks. *4th International Symposium on Multidisciplinary Studies and Innovative Technologies, ISMSIT 2020 - Proceedings*. https://doi.org/10.1109/ISMSIT50672.2020.9254451

Zawbaa, H., & Aly, S. A. (2012). Hajj and Umrah Event Recognition Datasets. In *undefined*.

Zeki, S. (2005). BRAIN AND VISUAL PERCEPTION The story of a 25-year collaboration By David H. Hubel and Torsten N. Wiesel 2004. New York: Oxford University Press. Price £29.99 ISBN 0-19-517618-9. *Brain*, *128*(5), 1226–1229. https://doi.org/10.1093/brain/awh507

Zhang, C., Kang, K., Li, H., Wang, X., Xie, R., & Yang, X. (2016). Data-Driven Crowd Understanding: A Baseline for a Large-Scale Crowd Dataset. *IEEE Transactions on Multimedia*, *18*(6), 1048–1061. https://doi.org/10.1109/TMM.2016.2542585

Zhang, S., Zhang, S., Huang, T., & Gao, W. (2018). Speech Emotion Recognition Using Deep Convolutional Neural Network and Discriminant Temporal Pyramid Matching. *IEEE Transactions on Multimedia*, *20*(6), 1576–1590. https://doi.org/10.1109/TMM.2017.2766843

Zhao, C., DIng, R., & Key, H. L. (2019). End-To-End Visual Place Recognition Based on Deep Metric Learning and Self-Adaptively Enhanced Similarity Metric. *Proceedings - International Conference on Image Processing, ICIP*, *2019-September*. https://doi.org/10.1109/ICIP.2019.8802931

Zhou, B., Tang, X., & Wang, X. (2013). Measuring crowd collectiveness. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 3049–3056. https://doi.org/10.1109/CVPR.2013.392

Zhu, F., Wang, X., & Yu, N. (2014). Crowd tracking with dynamic evolution of group structures. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *8694 LNCS*(PART 6), 139–154. https://doi.org/10.1007/978-3-319-10599-4_10

Zhu, J., Ai, Y., Tian, B., Cao, D., & Scherer, S. (2018). Visual Place Recognition in Long-term and Large-scale Environment based on CNN Feature. *IEEE Intelligent Vehicles Symposium, Proceedings*, *2018-June*. https://doi.org/10.1109/IVS.2018.8500686

Zhu, X. (Jerry). (2005). *Semi-Supervised Learning Literature Survey*. https://minds.wisconsin.edu/handle/1793/60444

Zou, J., Han, Y., & So, S. S. (2008). Overview of artificial neural networks. In *Methods in Molecular Biology* (Vol. 458, pp. 15–23). Humana Press. https://doi.org/10.1007/978-1-60327-101-1_2

APPENDICES

APPENDIX I

The following demonstration system is based on MATLAB simulation and is demonstrated in detail (Taha et al., 2021).

The images of Mecca's most essential sites, including Al-Haram (Image 1), Arafat (Image 2), Muzdalifa (Image 3), and Mina (Image 4), are obtained from (Taha et al., 2021). As depicted in the image, the crowds of pilgrims in these locations are so dense that recognizing each and every entity is a difficult task. To process further, our proposed method takes into account the sensitive crowded input places listed below (Figure I.1)

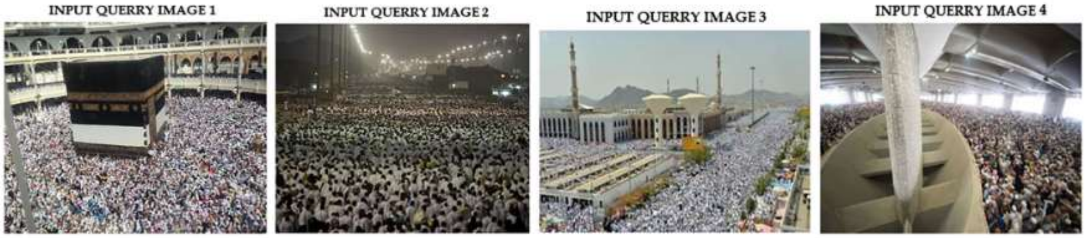


Figure I.1: Input images

Noise is created when there is a difference in the intensity value of the pixel during image capture movement. Such noises must be eliminated at this point. The median filter examines the input pixel and replaces it with the median value of neighboring pixels whose pattern corresponds to the window shown in Figure I.2.

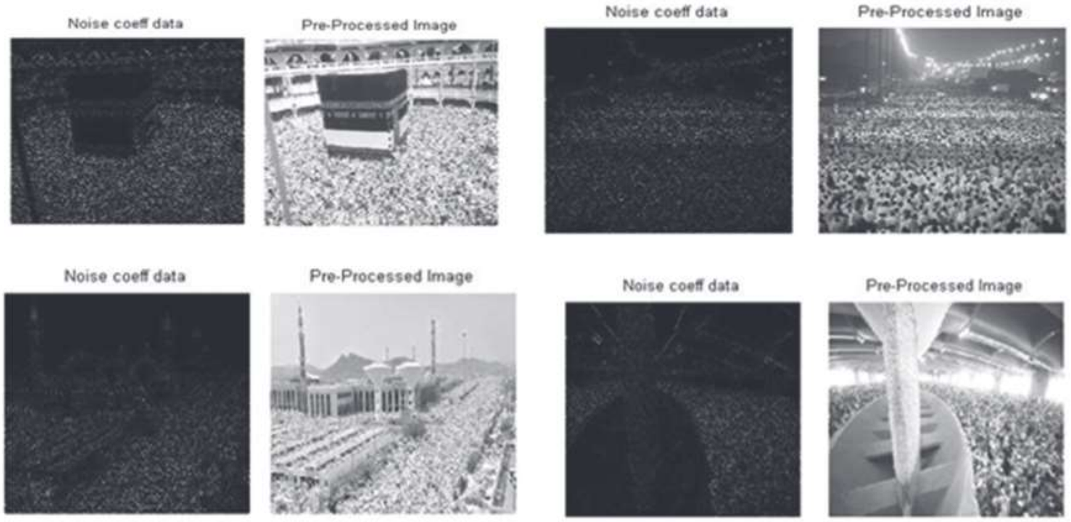


Figure I.2: Pre-processing (noise removal) of the images that are being used as input

Binarization reduces the data content (256 shades) of a color image or greyscale image to a black and white image. The easiest and most efficient method for assigning the 2 levels to pixels that are beneath or higher the threshold esteem is non-linear thresholding. The threshold value separates the histogram into two parts: object and background. Using global and local thresholding methods, the complexity of determining the ideal threshold value is decreased (Figure I.3).

The various frequency components in an image are classified into numerous classifications in order to focus the image energy disparately into separate frequency bands. It is stated that, while the vitality thickness of the various wavelet sub bands is undeniably different, causing rises by separating the images into factually disparate recurrence clusters of data, it is also obvious that the knowledge on the high recurrence sub-bands is exceptionally organized in the spatial domain (Figure I.4).

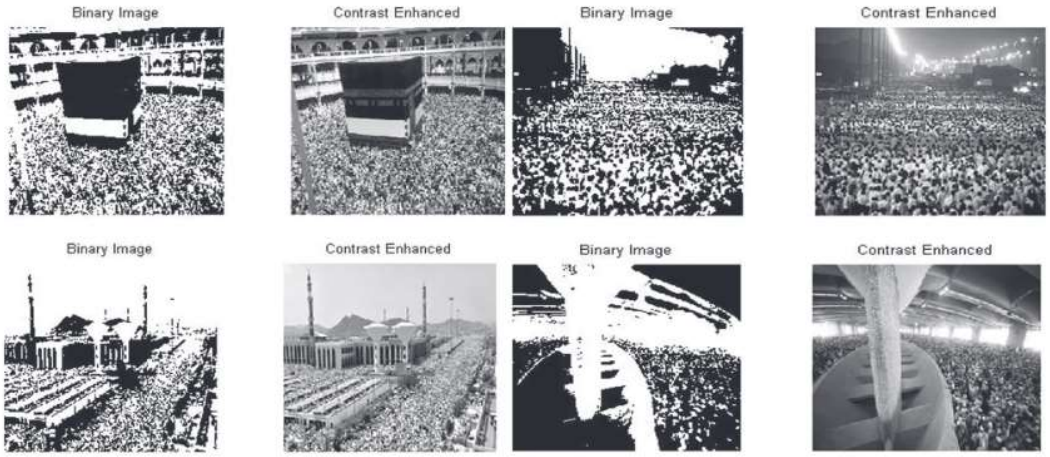


Figure I.3: Pre-processing (binarization) of input images

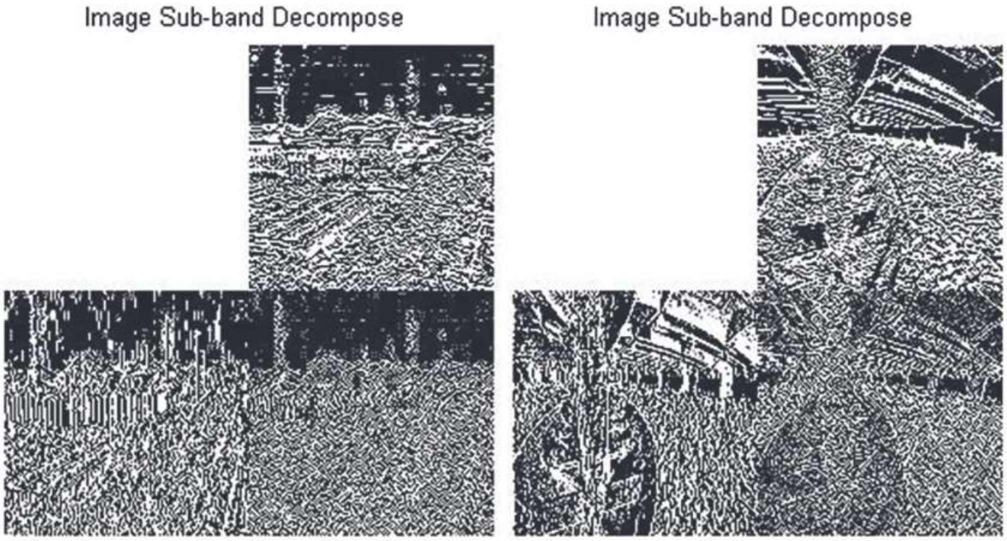


Figure I.4: Decomposition of input images

Image compression has been used to focus energy in a small region and decorrelate information. The wavelet conversion degrades the restricted vitality signal in the spatial space into a variety of functions as a standard in the symmetrical spatial area. In the quantified spatial area, the signal's characteristics are assessed. Unlike the conventional

FFT, the wavelet transformation explores functions in the evaluated spatial space and timing area, which has a better neighborhood limit of recurrence and time (Figure I.5).

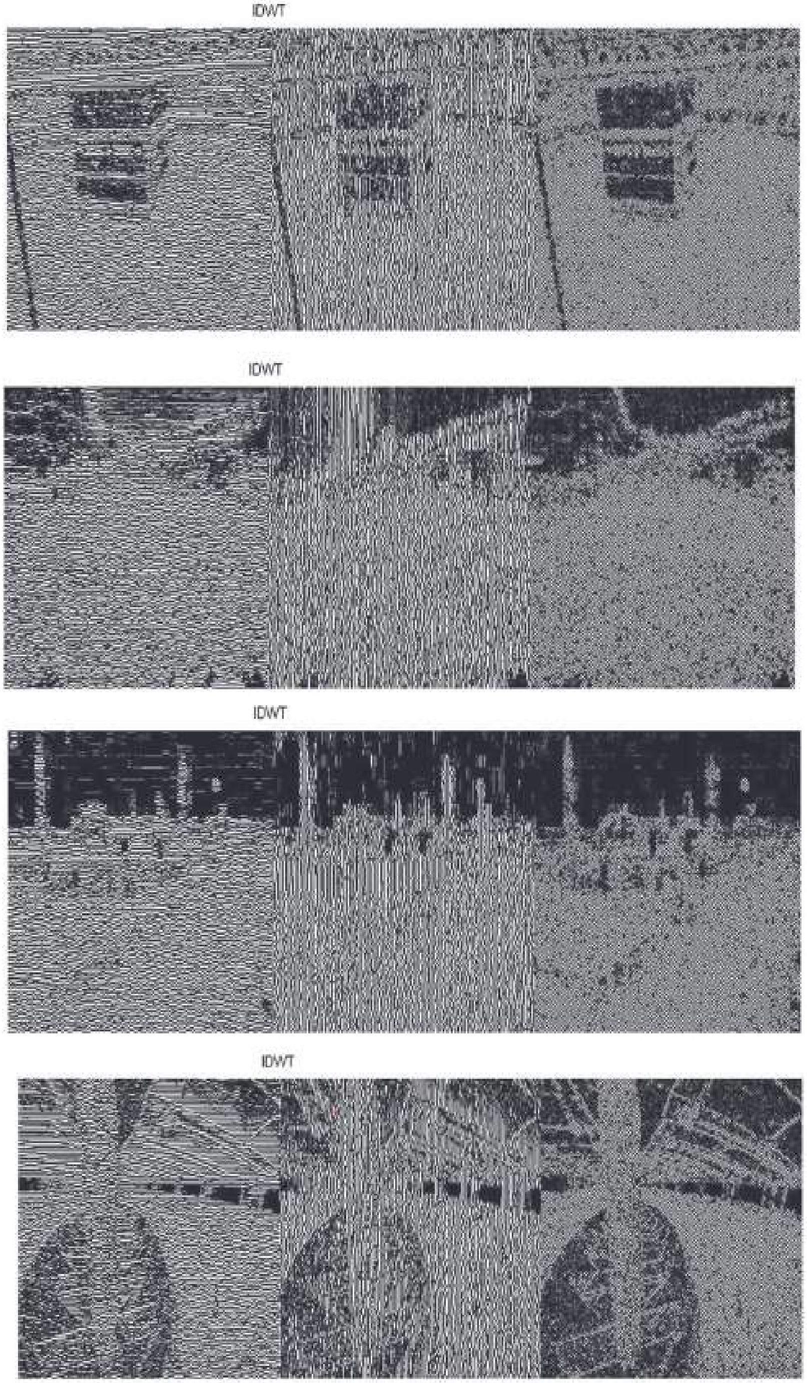


Figure I.5: IDWT of input images

Segmentation is the process of dividing an input image into different portions in order to convert the delineation of a diagram into some meaningful form. We suggest multithreshold-based segmentation in our work. We thought about identifying four separate places. Neighborhood areas are segmented based on contour information such as color, amplitude, and textures by marking the edges of each shape present in the image (Figure I.6).

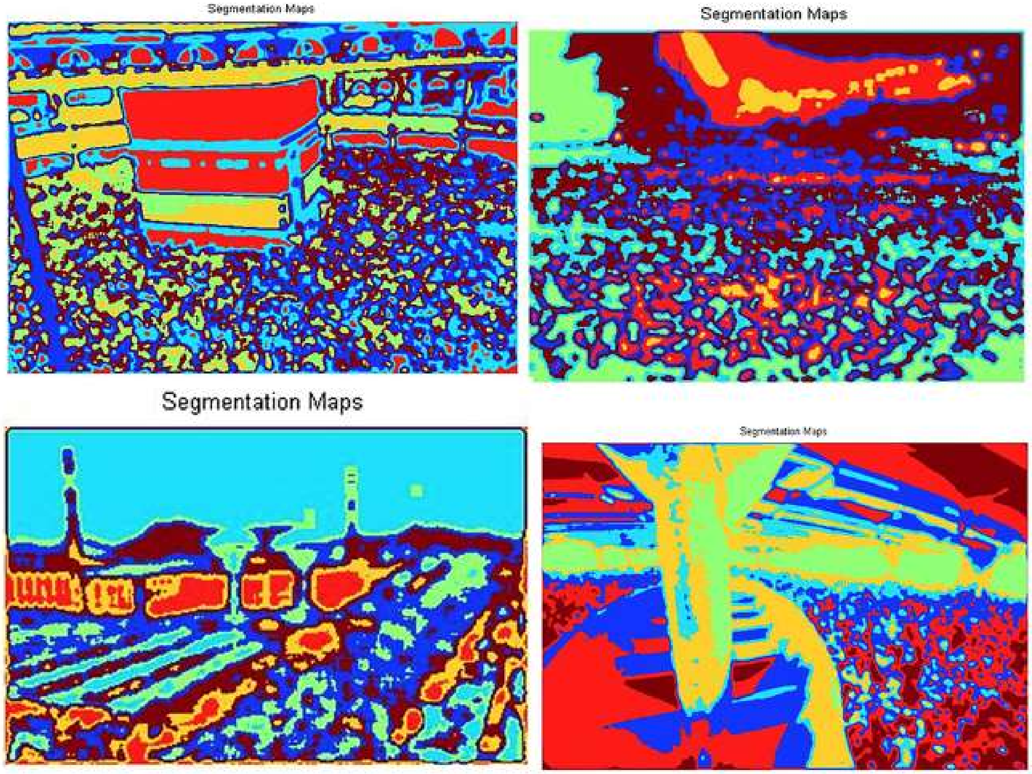


Figure I.6: Segmentation of input images

The image is quite far decreased in size by mapping, which transforms the image coordinates into different areas. After mapping into a 2D or 3D conversion, the characteristics can be clearly obtained (Figure I.7).

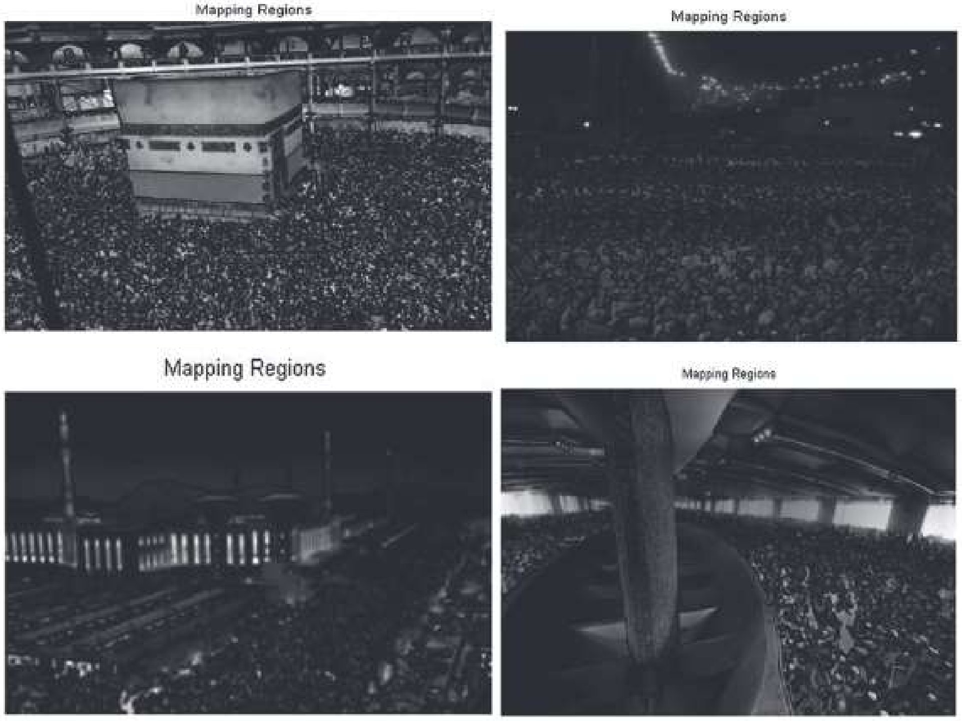


FIGURE 14 Mapping of input images

Figure I.7: Mapping of input images

Utilizing their edge data, the needed field can be labeled. The image above has been processed to order information with low energy at high frequency and high energy at low frequency. Utilizing trained date sequences, the edges can clearly define location information (Figure I.8).

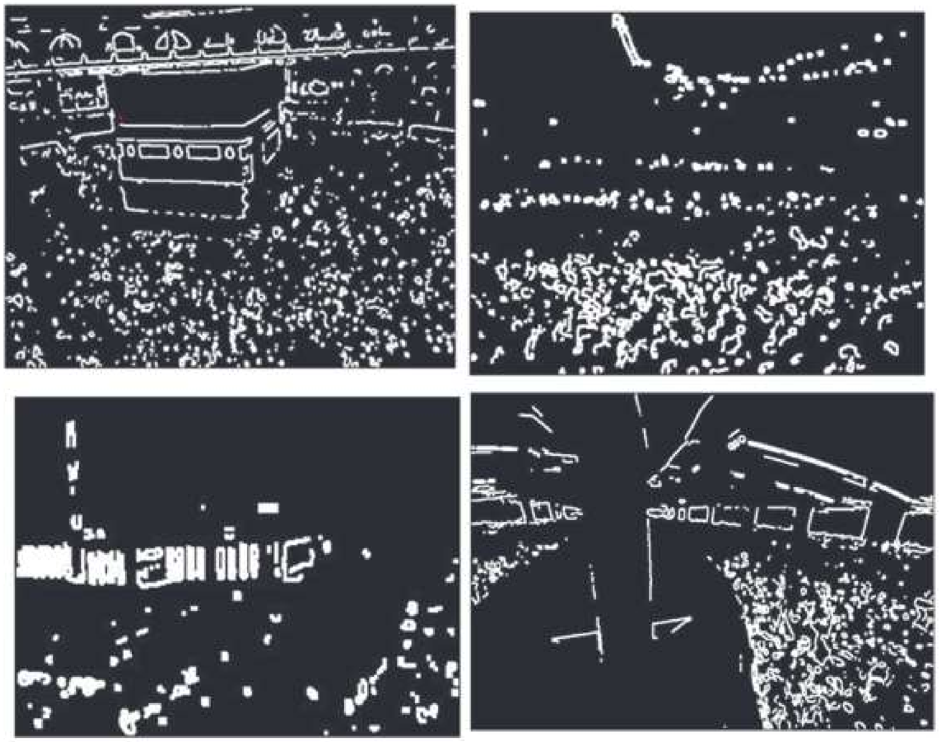


Figure I.8: Edge masking of input images

Categorizing the images improves the operation of the innovative study. For good functioning, a CNN network is suggested in this work. The number of steps required for each image varies. The number of steps required for the Al-Haram image is 158, but the Arafat image requires 171 iterations, and the Mina image requires a maximum of 218 iterations (Figure I.9).

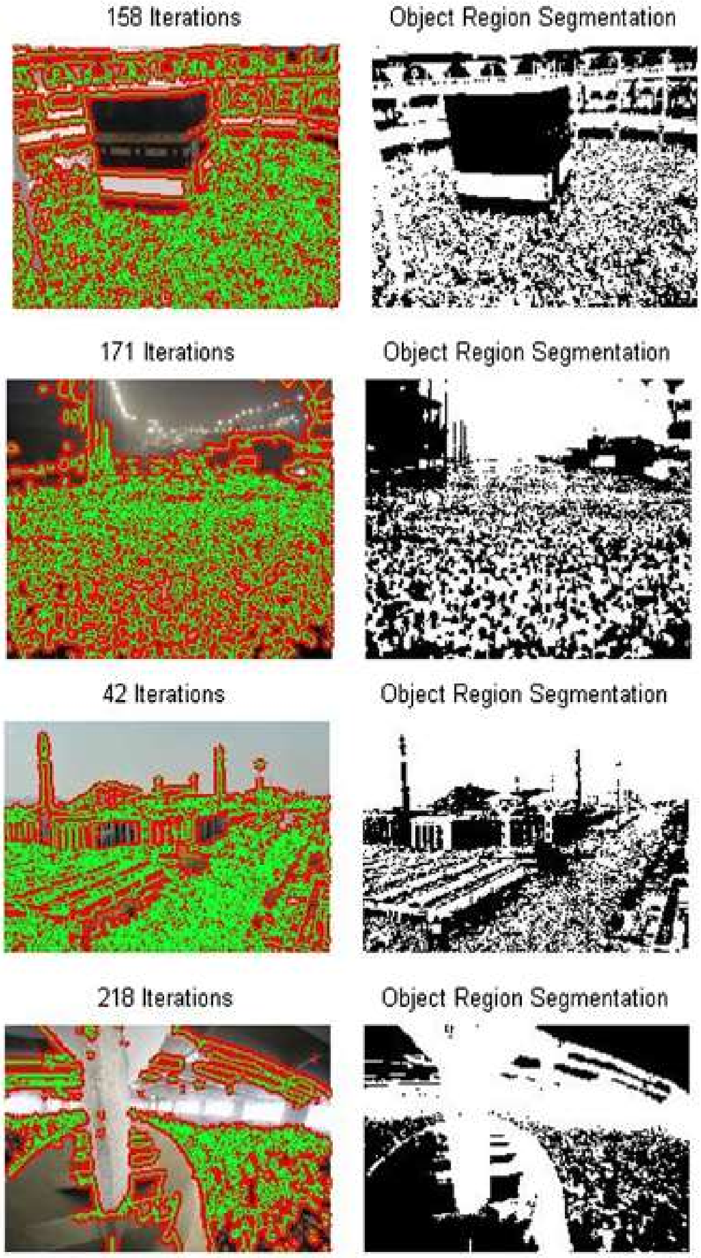
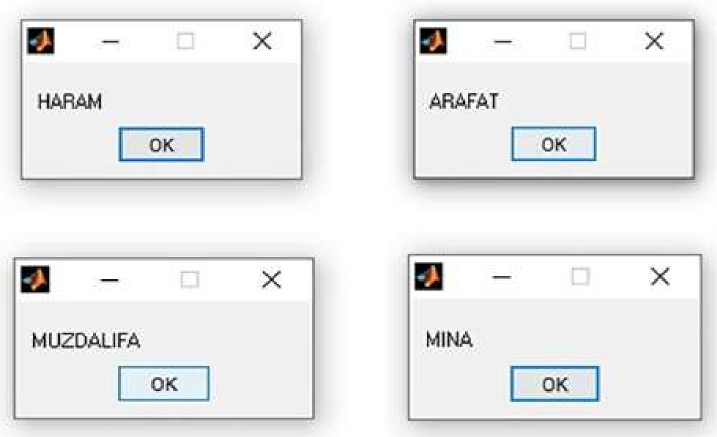

158 Iterations
Object Region Segmentation
171 Iterations
Object Region Segmentation
42 Iterations
Object Region Segmentation
218 Iterations
Object Region Segmentation

FIGURE 16 Classification of input images

Figure I.9: Classification of input images

The locations have been clearly identified by our system after completing all of the preceding steps (Figure I.10).



HARAM

ARAFAT

MUZDALIFA

MINA

Figure I.10: Output (place recognition) of proposed work.

APPENDIX II

SAMPLE PROGRAM CODE

```
main.m
clc;
clear all;
close all;
%[filename,pathname]=uigetfile({'*.bmp;*.tif;*.tiff;*.jpg;*.jpeg;*.gif','
IMAGE Files(*.bmp,*.tif,*.tiff,*.jpg,*.jpeg,*.gif)'},'Chose Image File');
[filename,pathname]=uigetfile({'*.png','IMAGE Files'},'Chose Image
File');
img=imread(cat(2,pathname,filename));
%inp_data = imresize(img,[300 300]);


inp_data=img;
figure;
imshow(inp_data);
title('Input image');
%img = imread(inp_data);
dim = size(inp_data);
width = dim(2);
height = dim(1);
nBins=5;
winSize=7;
nClass=6;
outImg = subfun(inp_data, nBins, winSize, nClass);
figure;
imshow(outImg);
title('Segmentation Maps');
colormap('default');
coeff=fspecial('gaussian', 3, 3);
filt_image = imfilter(inp_data,coeff,'symmetric','conv') ;
color_frm = makecform('srgb2lab', 'AdaptedWhitePoint',
whitepoint('d65'));
lab = applycform(filt_image,color_frm);
l = double(lab(:,:,1));
a = double(lab(:,:,2));
b = double(lab(:,:,3));
li = cumsum(cumsum(l,2));
ai = cumsum(cumsum(a,2));
bi = cumsum(cumsum(b,2));
salienc_map = zeros(height, width);
sm2 = zeros(height, width);
for j = 1:height
    yo = min(j, height-j);
    y1 = max(1,j-yo); y2 = min(j+yo,height);
    for k = 1:width
        xo = min(k,width-k);
        x1 = max(1,k-xo); x2 = min(k+xo,width);
        invarea = 1.0/((y2-y1+1)*(x2-x1+1));
        lm = inter_summa(li,x1,y1,x2,y2)*invarea;
        am = inter_summa(ai,x1,y1,x2,y2)*invarea;
        bina_map = inter_summa(bi,x1,y1,x2,y2)*invarea;

        salienc_map(j,k) = (l(j,k)-lm).^2 + (a(j,k)-am).^2 + (b(j,k)-
bina_map).^2;
    end
end
figure;
imshow(salienc_map,[]);
img = (salienc_map-min(salienc_map(:)))/(max(salienc_map(:))-
min(salienc_map(:)));
figure;
imshow(img);
title('Mapping Regions');
im1=img;
im1=medfilt2(im1,[3 3]); %Median filtering the image to remove noise%
BW = edge(im1,'sobel'); %finding edges
[imx,imy]=size(BW);
msk=[0 0 0 0 0;
     0 1 1 1 0;
     0 1 1 1 0;
     0 1 1 1 0;
     0 0 0 0 0;];
B=conv2(double(BW),double(msk));
L = bwlabel(B,8);
mx=max(max(L));
[r,c] = find(L==17);
rc = [r c];
[sx sy]=size(rc);
n1=zeros(imx,imy);
for i=1:sx
    x1=rc(i,1);
    y1=rc(i,2);
    n1(x1,y1)=255;
end
figure,imshow(im1);
figure,imshow(B);
title('Edge Masking');
seg = subfunct(inp_data,'whole',400,0.02,'vector');


Train.m
clc
clear all
close all
warning off

train_db = 'ARAFAT\';
class_name = 'ARAFAT';

count = 1;
count = featureExtraction(train_db,class_name,count);

train_db = 'HARAM\';
class_name = 'HARAM';
count = featureExtraction(train_db,class_name,count);
```

```
train_db = 'MINA\';
class_name = 'MINA';
count = featureExtraction(train_db,class_name,count);

train_db = 'MUZDALIFA\';
class_name = 'MUZDALIFA';
count = featureExtraction(train_db,class_name,count);


display('Training Finished !');

Testing.m
clc
clear all
close all
warning off

[filename, pathname] = uigetfile( {'*.png','Testing ,,Image'; ...
   '*.*',  'All Files (*.*)'}, ...
   'Read a file');
in_img = [pathname filename]; %
aa=imread(in_img);
    [m,n,x]=size(aa);
    if x==3
    res_img = rgb2gray(imresize(aa,[256 256]));
    else
    res_img = (imresize(aa,[256 256]));
    end
X = double(res_img);
[cA1,cH1,cV1,cD1] = dwt2(X,'haar');
sx = size(X);
A1 = idwt2(cA1,[],[],[],'haar',sx);
H1 = idwt2([],cH1,[],[],'haar',sx);
V1 = idwt2([],[],cV1,[],'haar',sx);
D1 = idwt2([],[],[],cD1,'haar',sx);

k = 7;
[IDX,C] = kmeans(double(X),k);
centers_val = mean(C,2);

cooccur_matri = graycomatrix(res_img,'Offset',[2 0;0 2]);
stats = invar_feat_extrc(cooccur_matri,0);
energy = stats.energ;
entrophy = stats.entro;
contust = stats.contr;
autoCorr = stats.autoc;
prob = stats.maxpr;
feat1 = [energy entrophy contust autoCorr prob centers_val'];
%feat1.val = [energy entrophy contust autoCorr prob centers_val'
res_img(1,:)];
[dist_val1,outclass] = invar_match(feat1);
msgbox(outclass);

function count = featureExtraction(train_db,class_name,count)

files = dir([train_db '*.png']);
```

```
for i = 1:length(files)
    [dir_name] = files(i).name;

    files2 = [train_db dir_name];
    aa=imread(files2);
    [m,n,x]=size(aa);
    if x==3
    res_img = rgb2gray(imresize(aa,[256 256]));
    else
    res_img = (imresize(aa,[256 256]));
    end
    imm=imread(files2);

    X = double(res_img);
    [cA1,cH1,cV1,cD1] = dwt2(X,'haar');
    sx = size(X);
    A1 = idwt2(cA1,[],[],[],'haar',sx);
    H1 = idwt2([],cH1,[],[],'haar',sx);
    V1 = idwt2([],[],cV1,[],'haar',sx);
    D1 = idwt2([],[],[],cD1,'haar',sx);

    k = 7;
    [~,C] = kmeans(double(X),k);
    centers_val = mean(C,2);

    cooccur_matri = graycomatrix(res_img,'Offset',[2 0;0 2]);
    stats = invar_feat_extrc(cooccur_matri,0);
    energy = stats.energ;
    entrophy = stats.entro;
    contust = stats.contr;
    autoCorr = stats.autoc;
    prob = stats.maxpr;
   % feat.val = [energy entrophy contust autoCorr prob centers_val'
res_img(1,:)];
    feat.val = [energy entrophy contust autoCorr prob centers_val'];
    feat.class = class_name;
    save(['Feature_Dir\feature_' num2str(count)],'feat');
    count = count+1;
end

function [out] = invar_feat_extrc(glcmin,pairs)

if ((nargin > 2) || (nargin == 0))
   error('Too many or too few input arguments. Enter GLCM and pairs.');
elseif ( (nargin == 2) )
    if ((size(glcmin,1) <= 1) || (size(glcmin,2) <= 1))
       error('The GLCM should be a 2-D or 3-D matrix.');
    elseif ( size(glcmin,1) ~= size(glcmin,2) )
        error('Each GLCM should be square with NumLevels rows and
NumLevels cols');
    end
elseif (nargin == 1)
    pairs = 0;
    if ((size(glcmin,1) <= 1) || (size(glcmin,2) <= 1))
       error('The GLCM should be a 2-D or 3-D matrix.');
    elseif ( size(glcmin,1) ~= size(glcmin,2) )
```

```
        error('Each GLCM should be square with NumLevels rows and
NumLevels cols');
    end
end


format long e
if (pairs == 1)
    newn = 1;
    for nglcm = 1:2:size(glcmin,3)
        glcm(:,:,newn)  = glcmin(:,:,nglcm) + glcmin(:,:,nglcm+1);
        newn = newn + 1;
    end
elseif (pairs == 0)
    glcm = glcmin;
end

size_glcm_1 = size(glcm,1);
size_glcm_2 = size(glcm,2);
size_glcm_3 = size(glcm,3);


out.autoc = zeros(1,size_glcm_3); % Autocorrelation: [2]
out.contr = zeros(1,size_glcm_3); % Contrast: matlab/[1,2]
out.corrm = zeros(1,size_glcm_3); % Correlation: matlab
out.corrp = zeros(1,size_glcm_3); % Correlation: [1,2]
out.cprom = zeros(1,size_glcm_3); % Cluster Prominence: [2]
out.cshad = zeros(1,size_glcm_3); % Cluster Shade: [2]
out.dissi = zeros(1,size_glcm_3); % Dissimilarity: [2]
out.energ = zeros(1,size_glcm_3); % Energy: matlab / [1,2]
out.entro = zeros(1,size_glcm_3); % Entropy: [2]
out.homom = zeros(1,size_glcm_3); % Homogeneity: matlab
out.homop = zeros(1,size_glcm_3); % Homogeneity: [2]
out.maxpr = zeros(1,size_glcm_3); % Maximum probability: [2]

out.sosvh = zeros(1,size_glcm_3); % Sum of sqaures: Variance [1]
out.savgh = zeros(1,size_glcm_3); % Sum average [1]
out.svarh = zeros(1,size_glcm_3); % Sum variance [1]
out.senth = zeros(1,size_glcm_3); % Sum entropy [1]
out.dvarh = zeros(1,size_glcm_3); % Difference variance [4]
out.denth = zeros(1,size_glcm_3); % Difference entropy [1]
out.inf1h = zeros(1,size_glcm_3); % Information measure of correlation1
[1]
out.inf2h = zeros(1,size_glcm_3); % Informaiton measure of correlation2
[1]
out.indnc = zeros(1,size_glcm_3); % Inverse difference normalized (INN)
[3]
out.idmnc = zeros(1,size_glcm_3); % Inverse difference moment normalized
[3]


glcm_sum  = zeros(size_glcm_3,1);
glcm_mean = zeros(size_glcm_3,1);
glcm_var  = zeros(size_glcm_3,1);
u_x = zeros(size_glcm_3,1);
u_y = zeros(size_glcm_3,1);
```

```
s_x = zeros(size_glcm_3,1);
s_y = zeros(size_glcm_3,1);

p_x = zeros(size_glcm_1,size_glcm_3); % Ng x #glcms[1]
p_y = zeros(size_glcm_2,size_glcm_3); % Ng x #glcms[1]
p_xplusy = zeros((size_glcm_1*2 - 1),size_glcm_3); %[1]
p_xminusy = zeros((size_glcm_1),size_glcm_3); %[1]

hxy  = zeros(size_glcm_3,1);
hxy1 = zeros(size_glcm_3,1);
hx   = zeros(size_glcm_3,1);
hy   = zeros(size_glcm_3,1);
hxy2 = zeros(size_glcm_3,1);

for k = 1:size_glcm_3

    glcm_sum(k)  = sum(sum(glcm(:,:,k)));
    glcm(:,:,k)  = glcm(:,:,k)./glcm_sum(k);
    glcm_mean(k) = mean2(glcm(:,:,k));
    glcm_var(k)  = (std2(glcm(:,:,k)))^2;

    for i = 1:size_glcm_1

        for j = 1:size_glcm_2

            out.contr(k) = out.contr(k) + (abs(i - j))^2.*glcm(i,j,k);
            out.dissi(k) = out.dissi(k) + (abs(i - j)*glcm(i,j,k));
            out.energ(k) = out.energ(k) + (glcm(i,j,k).^2);
            out.entro(k) = out.entro(k) - (glcm(i,j,k)*log(glcm(i,j,k) +
eps));
            out.homom(k) = out.homom(k) + (glcm(i,j,k)/( 1 + abs(i-j) ));
            out.homop(k) = out.homop(k) + (glcm(i,j,k)/( 1 + (i - j)^2));

            out.sosvh(k) = out.sosvh(k) + glcm(i,j,k)*((i -
glcm_mean(k))^2);


            out.indnc(k) = out.indnc(k) + (glcm(i,j,k)/( 1 + (abs(i-
j)/size_glcm_1) ));
            out.idmnc(k) = out.idmnc(k) + (glcm(i,j,k)/( 1 + ((i -
j)/size_glcm_1)^2));
            u_x(k)          = u_x(k) + (i)*glcm(i,j,k); % changed
10/26/08
            u_y(k)          = u_y(k) + (j)*glcm(i,j,k); % changed
10/26/08

        end

    end
    out.maxpr(k) = max(max(glcm(:,:,k)));
end

for k = 1:size_glcm_3

    for i = 1:size_glcm_1
```

```
        for j = 1:size_glcm_2
            p_x(i,k) = p_x(i,k) + glcm(i,j,k);
            p_y(i,k) = p_y(i,k) + glcm(j,i,k); % taking i for j and j for
i
            if (ismember((i + j),[2:2*size_glcm_1]))
                p_xplusy((i+j)-1,k) = p_xplusy((i+j)-1,k) + glcm(i,j,k);
            end
            if (ismember(abs(i-j),[0:(size_glcm_1-1)]))
                p_xminusy((abs(i-j))+1,k) = p_xminusy((abs(i-j))+1,k)
+...
                    glcm(i,j,k);
            end
        end
    end


end

for k = 1:(size_glcm_3)

    for i = 1:(2*(size_glcm_1)-1)
        out.savgh(k) = out.savgh(k) + (i+1)*p_xplusy(i,k);

        out.senth(k) = out.senth(k) - (p_xplusy(i,k)*log(p_xplusy(i,k) +
eps));
    end

end

for k = 1:(size_glcm_3)

    for i = 1:(2*(size_glcm_1)-1)
        out.svarh(k) = out.svarh(k) + (((i+1) -
out.senth(k))^2)*p_xplusy(i,k);

    end

end
for k = 1:size_glcm_3
    for i = 0:(size_glcm_1-1)
        out.denth(k) = out.denth(k) -
(p_xminusy(i+1,k)*log(p_xminusy(i+1,k) + eps));
        out.dvarh(k) = out.dvarh(k) + (i^2)*p_xminusy(i+1,k);
    end
end

for k = 1:size_glcm_3
    hxy(k) = out.entro(k);
    for i = 1:size_glcm_1

        for j = 1:size_glcm_2
            hxy1(k) = hxy1(k) - (glcm(i,j,k)*log(p_x(i,k)*p_y(j,k) +
eps));
```

```
            hxy2(k) = hxy2(k) - (p_x(i,k)*p_y(j,k)*log(p_x(i,k)*p_y(j,k)
+ eps));
        end
        hx(k) = hx(k) - (p_x(i,k)*log(p_x(i,k) + eps));
        hy(k) = hy(k) - (p_y(i,k)*log(p_y(i,k) + eps));
    end
    out.inf1h(k) = ( hxy(k) - hxy1(k) ) / ( max([hx(k),hy(k)]) );
    out.inf2h(k) = ( 1 - exp( -2*( hxy2(k) - hxy(k) ) ) )^0.5;
end

corm = zeros(size_glcm_3,1);
corp = zeros(size_glcm_3,1);
for k = 1:size_glcm_3
    for i = 1:size_glcm_1
        for j = 1:size_glcm_2
            s_x(k)  = s_x(k)  + (((i) - u_x(k))^2)*glcm(i,j,k);
            s_y(k)  = s_y(k)  + (((j) - u_y(k))^2)*glcm(i,j,k);
            corp(k) = corp(k) + ((i)*(j)*glcm(i,j,k));
            corm(k) = corm(k) + (((i) - u_x(k))*((j) -
u_y(k))*glcm(i,j,k));
            out.cprom(k) = out.cprom(k) + (((i + j - u_x(k) -
u_y(k))^4)*...
                glcm(i,j,k));
            out.cshad(k) = out.cshad(k) + (((i + j - u_x(k) -
u_y(k))^3)*...
                glcm(i,j,k));
        end
    end
    s_x(k) = s_x(k) ^ 0.5;
    s_y(k) = s_y(k) ^ 0.5;
    out.autoc(k) = corp(k);
    out.corrp(k) = (corp(k) - u_x(k)*u_y(k))/(s_x(k)*s_y(k));
    out.corrm(k) = corm(k) / (s_x(k)*s_y(k));
end


function [inx,outclass] = invar_match(feat1)
dir_name = 'Feature_Dir\';

files = dir([dir_name '*.mat']);

all_dist = [];
classname = {};
for i = 1:length(files)
    file = [dir_name files(i).name];
    load (file)
    dist_val = sum(sum((double(feat.val(:,8:end))-
double(feat1(:,8:end))).^2)).^0.5;
    classname{i} = feat.class;
    all_dist = [all_dist;dist_val];
end

[val inx] = min(all_dist);

outclass = classname{inx};
```

APPENDIX III

ETHICAL APROVAL DOCUMENT

Date: _2__/_8__/_2021___

To the **Graduate School of Applied Sciences**

The research project titled “Place Identification in Mecca using Deep Neural Networks” has been evaluated. Since the researcher(s) will not collect primary data from humans, animals, plants or earth, this project does not need to go through the ethics committee.

Title: Assoc Prof Dr

Name Surname: Melike Sah Direkoglu

Signature:

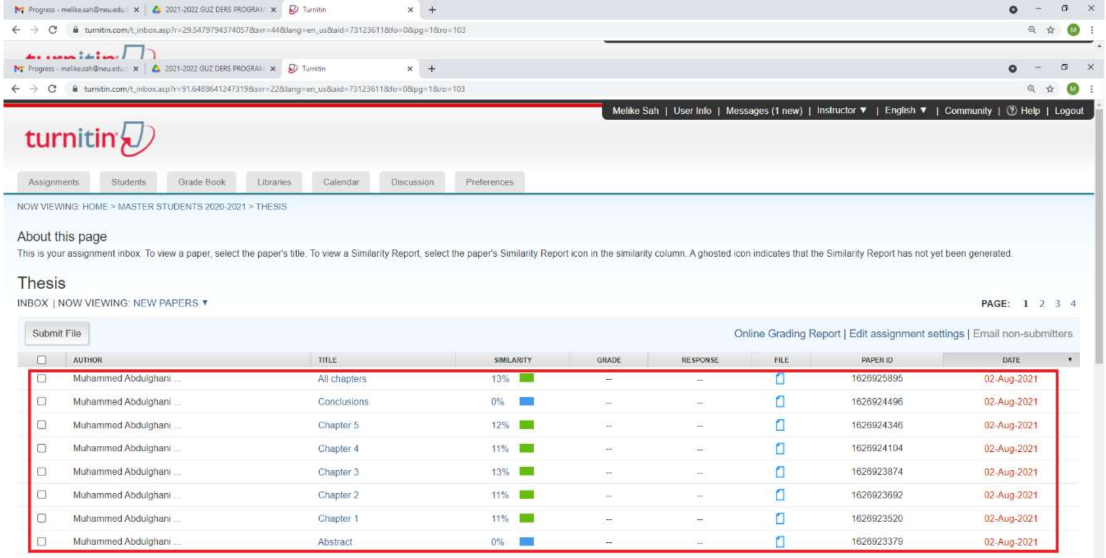
Role in the Research Project: Supervisor

APPENDIX IV

Similarity Report

Chapters	Percentages
Abstract.doc/docx	0%
Chapter 1.doc/docx	11%
Chapter 2.doc/docx	11%
Chapter 3.doc/docx	13%
Chapter 4.doc/docx	11%
Chapter 5.doc/docx	12%
Conclusions.doc/docx	0%
*All.doc/docx	13%

*All.doc/docx document must include all your thesis chapters (except cover page, table of contents, acknowledge, declaration, references, appendix, list of figures, list of tables, and abbreviations list).


Progress - melike.sah@neu.edu
2021-2022 GUZ DERS PROGRAM
Turnitin
turnitin.com/t_inbox.asp?r=29.5479794374057&svr=44&lang=en_us&aid=73123611&fo=0&pg=1&ro=103
turnitin.com/t_inbox.asp?r=91.6488641247319&svr=22&lang=en_us&aid=73123611&fo=0&pg=1&ro=103
Melike Sah | User Info | Messages (1 new) | Instructor ▼ | English ▼ | Community | Help | Logout
turnitin
Assignments
Students
Grade Book
Libraries
Calendar
Discussion
Preferences
Submit File
Online Grading Report | Edit assignment settings | Email non-submitters
PAGE: 1 2 3 4

NOW VIEWING: HOME > MASTER STUDENTS 2020-2021 > THESIS

About this page

This is your assignment inbox. To view a paper, select the paper's title. To view a Similarity Report, select the paper's Similarity Report icon in the similarity column. A ghosted icon indicates that the Similarity Report has not yet been generated.

Thesis

INBOX | NOW VIEWING: NEW PAPERS ▼

☐	AUTHOR	TITLE	SIMILARITY	GRADE	RESPONSE	FILE	PAPER ID	DATE ▼
☐	Muhammed Abdulghani ...	All chapters	13% 🟩	--	--	🗋	1626925895	02-Aug-2021
☐	Muhammed Abdulghani ...	Conclusions	0% 🟦	--	--	🗋	1626924496	02-Aug-2021
☐	Muhammed Abdulghani ...	Chapter 5	12% 🟩	--	--	🗋	1626924346	02-Aug-2021
☐	Muhammed Abdulghani ...	Chapter 4	11% 🟩	--	--	🗋	1626924104	02-Aug-2021
☐	Muhammed Abdulghani ...	Chapter 3	13% 🟩	--	--	🗋	1626923874	02-Aug-2021
☐	Muhammed Abdulghani ...	Chapter 2	11% 🟩	--	--	🗋	1626923692	02-Aug-2021
☐	Muhammed Abdulghani ...	Chapter 1	11% 🟩	--	--	🗋	1626923520	02-Aug-2021
☐	Muhammed Abdulghani ...	Abstract	0% 🟦	--	--	🗋	1626923379	02-Aug-2021

Regards,

Assoc. Prof Dr Melike Sah Direkoglu.

APPENDIX V

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name	: Taha, Mohammed Abdulghani
Nationality	: Iraq
Date and Place of Birth	: 17 of August 1987, Erbil
Marital Status	: Married

EDUCATION

Degree	Institution	Year of Graduation
M.Sc.	BAU, Department of Computer Engineering, Istanbul, Turkey.	2013
B.Sc.	Salahaddin University, Department of Software Engineering, Erbil, Iraq.	2010

WORK EXPERIENCE

Year	Place	Enrollment
Aug, 2017-present	Department of Computer Engineering	Lecturer
Jan, 2014-2016	Versian Company for Programming	Programmer
Sep, 2013-2014	Department of Computer Engineering	Lab Assistant
Jan, 2011-2012	IDB Company for Programming.	Programmer

FOREIGN LANGUAGES

English, Turkish, Arabic and Turkish

PUBLICATION IN INTERNATIONAL REFFEREED JOURNALS IN (IN COVERAGE OF SSCI/SCI-EXPANDED AND AHCI):

- Taha, M. A., Direkoglu, M. S., & Direkoglu, C. (2021). Deep neural network-based detection of pilgrims location in Holy Makkah. *International Journal of Communication Systems*, e4792. https://doi.org/10.1002/dac.4792

- S. Ibrahim, Taha, M. A., Ibrahim, A. A. (2021) Diabetic Retinopathy Detection Using Local Extrema Quantized Haralick Features with Long Short-Term Memory Network. *Inernational Journal of Biomedical Imaging*

PUBLICATION IN INTERNATIONAL REFFEREED JOURNALS IN (IN COVERAGE OF British Education Index, ERIC, Science Direct, Scopus, IEEE)

- Taha, M. A., Şah, M., & Direkoğlu, C. (2020). *Review of Place Recognition Approaches: Traditional and Deep Learning Methods*. International Conference on Theory and Application of Fuzzy Systems and Soft Computing (ICAFS2020), Advances in Intelligent Systems and Computing, vol 1306, 183–191, Springer, Cham. https://doi.org/10.1007/978-3-030-64058-3_22
- Taha, M. A., (202) Hybrid algorithms for spectral noise removal in hyper spectral images. *AIP Conference Proceedings*

THESIS

Master

- Taha, M.A (2013). C3Net Algorithm Using Dynamic Bayesian Network. Master Thesis, Bahcesehir University, Comuter Engineering, Istanbul, Turkey.

Lisans

- Taha, M. A. (2010). A Kurdish Search Engine. Undergraduate project (B. Sc.), Salahaddin University, Faculty of Engineering, Software Department, Irak, Erbil.

COURSES GIVEN (*from 2010 to 2021*)

- Object Oriented Programing (English)
- Web Design (English)
- Web Programming (English)
- Introduction To Programming (English)
- Academic Debate and Critical Thinking (English)

HOBBIES

- Reading, Traveling, Walking , Making Coffee.