
# ADDIS ABABA UNIVERSITY

## COLLEGE OF NATURAL SCIENCES

***Mosque Building Detection Using Deep Convolutional Neural Network***

Samrawit Ergete

**A Thesis Submitted to the Department of Computer Science in Partial Fulfillment for the Degree of Master of Science in Computer Science**

Addis Ababa Ethiopia

October 2020

**ADDIS ABABA UNIVERSITY**

**COLLEGE OF NATURAL SCIENCES**

SAMRAWIT ERGETE

Advisor: Ayalew Belay (PhD)

This is to certify that the thesis prepared by *Samrawit Ergete*, titled: *Mosque Building Detection Using Deep Convolutional Neural Network* and submitted in partial fulfillment of the requirements for the Degree of Master of Science in Computer Science complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the Examining Committee:

| **Name** | **Signature** | **Date** |
|---|---|---|
| Advisor: Ayalew Belay (PhD) | _____ | _____ |
| Examiner: _____ | _____ | _____ |
| Examiner: _____ | _____ | _____ |

**Abstract**

Object detection is a computer technology related to computer vision and image processing that detects and defines objects such as humans, buildings and cars from images and videos. Object detection is breaking into a wide range of industries, with use cases ranging from personal security to productivity in the workplace. Facial recognition or face detection is one of an object detection examples, which can be utilized as a security measure to let only certain people into a classified area of building. It can also be used within a visual search engine to help consumers find a specific item they're on the hunt for.

In this work we propose detection system start from collecting and preparing data to detecting mosque building by using deep convolutional neural network (DCNN). Mosque building detection is done using Faster RCNN model.

Faster RCNN is trained on 1848 dataset collected from different websites and by directly taking pictures and splinted into 90% for training and 10% for testing.

Experimental results have proved the efficiency of the proposed technique, where the accuracy of the proposed scheme has achieved mAP of 0.70.

Keywords: Mosque building, Image processing, deep learning, detection, Faster RCNN.

**Acknowledgement**

The success and outcome of this project required a lot of guidance and assistance from many people and I am extremely privileged to have got this all along the completion of my project. All that I have done is only due to such supervision and assistance and I would not forget to thank them.

I would like to express my sincere gratitude to my advisor Dr Ayalew Belay for his continuous support, his patience, motivation, insightful comments and hard questions.

I would like to thank my mother Takelech, my sisters Yamerot and Hiwot and my brother Alemante, for your love, support and encouragement! My friends Melaku, Engdawork and Beziawit and thank you for being there for me. Thank you!

I wish to thank my father Ergete, for helping, supporting, loving, advising me in your life time. I wish you could see this. Thank you with infinity love!

Finally, I also would like to thank my loved ones, who have supported me throughout the entire process, both by keeping me harmonious and helping me put pieces together. I will be grateful forever for your love.

# Table of Contents

## List of Figures

## List of Tables

## List of Algorithm

## Acronyms and Abbreviations

| | |
|---|---|
| TP | True positive |
| TN | True negative |
| FN | False negative |
| RoI | Region of Interest |
| YOLO | You Only Look Ones |
| NMS | Non-max-suppression |
| SSD | Single shoot detection |
| DIP | Digital image processing |
| SVM | Support Vector Machine |
| mAP | Mean average precession |
| ANN | Artificial neural network |
| RPN | Region Proposal Network |
| MBD | Mosque building detection |
| SGD | Stochastic gradient descent |
| CNN | Convolutional neural network |
| DCNN | Deep convolutional neural network |
| SPPNet | Spatial Pyramid Pooling Network |
| RFCN | Region based Fully Convolutional Network |
| RCNN | Region based Convolutional Neural Network |
| Fast-RCNN | Fast Region based Convolutional Neural Network |
| Faster-RCNN | Faster Region based Convolutional Neural Network |

# CHAPTER 1 : INTRODUCTION

## 1.1 Background

A building is a structure with a roof and walls standing permanently attached to the ground to provide shelter for humans, animals, pieces of machinery, or as a performance place of human activities [1]. As needs of society increases in complexity, humans created various types of buildings to accommodate their needs. Office, retail, hospitality, school, religious, and agricultural buildings to name the few.

Mosque are grouped in the category of religious buildings. The appearance of a mosque has a unique similarity with church buildings. The main difference between the two buildings is the presence of different types of symbols on top of their dome and minaret. Mosques usually recognized or differentiated by moon or a star as a unique symbol while most churches have cross [2, 3].

Recognizing and differentiating objects is an easy task for humans, also, simulation of human intelligence processes by machines. This process includes learning (the acquisition of information and rules for using the information), reasoning (using the rules to reach approximate or definite conclusions), and self-correction.
Machines can learn is by using machine learning [4]**.** Machine learning is the study of algorithms and statistical models. Systems use machine learning to perform a task without explicit instructions, instead rely on patterns and inference done through computer vision, software engineering, and pattern recognition.

Computer vision is the process of understanding digital images and videos using computers. It seeks to automate tasks that human vision achieves. The process involves methods of acquiring, processing, analyzing, understanding digital images, and extraction of data from the real world to produce information. It also has sub-domains such as object recognition, video tracking, and motion estimation [5].

Computer vision technology has been extensively used in different industries such as automation, consumer markets, medical organizations, entertainment sectors, defense, and surveillance. The

ubiquitous and wide applications such as scene understanding, video surveillance, robotics, and self-driving cars triggered vast research in the domain of computer vision during the most recent decade.

A lot has been done on object detection but the works are subject related or feature dependent to detect an object and some works require manual feature extraction method.
In this paper, we study how we can build object detection specifically mosque buildings using automatic feature extraction method and provide a design of mosque building detection using a state-of-the-art method called Faster RCNN which can detect whether an image contains mosque or not and localize their regions if there is any.

## 1.2 Motivation

Computer vision conjunction with a robot is one of the popular applications in recent times. Robots have found their way from sealed working stations in factories to people's living and working spaces. They autonomously perform different services useful to humans, such as domestic tasks, healthcare services, entertainment, and education.

Research is progressing from special-purpose service robots such as autonomous cleaning or transport systems, to multi-functional assistive robots able to integrate diverse abilities such as person detection and tracking, postal and delivery services, reasoning, localization, navigation, object detection and recognition, planning and manipulation [6].

Among different techniques, image analysis techniques are used to determine if the object in view is a valid workpiece. Once the item has been correctly classified by the vision system, a very limited amount of data-typically only a few bits- is necessary to tell the (robot) system which of a set of predetermined sequences of actions it must follow. Aiming to implementing machine learning technology, in this research work, we process digital image with deep learning to design a mosque detection system. The research provides as one input knowledge for researchers or companies to teach robots.

## 1.3 Statement of the problem

Object detection have witnessed growing interest over the last decades due to the improvements of the performance of computer systems. A number of researches on the development of object detection system have been carried out using different techniques.

Hasan and Iman [7] and Wesam et al. [8], propose a system that could detect vehicles such as buses, cars, motorbikes, vans, and truck from images using one of deep convolutional neural network called Faster RCNN. Abdullah et al. [9],and Beril and Cem [10], propose a system that could detect building from a high-resolution area using traditional methods. Farzaneh and Esmat [11], propose hand tools like meter, nipper, screwdriver, spanner, and tongs detection system using ANN. Hui and Jinwen [12], proposed shape-based horses detection system. Changzheng et al. [13], propose a system that detects face using Faster RCNN. Erik et al. [14], Minh-Tan and Sebastien [15], Xiaodong et al. [16], Ruhan, et al. [17], Feng et al. [18], propose firearms, underground buried objects, people, landmark and ship detection system using Faster RCNN. Also, Asad et al. [19], use Fast RCNN to detect pedestrian.

However, the existing detection system, like Faster RCNN, Fast RCNN and template matching are used to detect different objects including buildings. To detect mosque building using the existed ones, they are trained on a specific feature. Therefore, it will result to bad detection result because of different feature representation of different objects [34-36], since they do not consider mosque building as one input component.

Likewise, in some researches, the image analysis part is done by shallow networks which need manual feature extraction. Using manual feature extraction is not good for classification tasks because, we specify restrictions on what features represent the input data [31].

To our best knowledge, mosque building detection system has not been discussed or addressed by any of the researchers. Hence, this research aims at developing a system that can identify the presence of mosque building and localize their regions using digital image processing, and deep learning.

Based on the problems identified the following research questions are raised.

1. Which detection mechanism is efficient to detect mosque building?
2. What type of feature can efficiently describe mosque building?

## 1.4 Objectives

### General objective

The main objective of this thesis is to propose mosque building detection that can detect whether an image contains a mosque or not and localize their regions if a mosque is detected.

### Specific objectives

To achieve the general objective, the following specific objectives are identified

- Review related literature in the area of image processing and object detection.
- Collect and prepare image data from different sources.
- Study and select suitable image detection techniques.
- Design the system architecture for the proposed system.
- Develop Prototype for the proposed system.
- Test and evaluate the prototype.

## 1.5 Methods

To achieve the objectives, the research follows a design science research methodology. The different methodologies that were used through the research time are:

**Literature Review:** Detail review and assessment will be made on object detection to understand existing techniques such as digital image processing starting from pre-processing an input dataset to the object detection processing steps. Likewise, different reviews on external mosque building architecture components will be made to select appropriate techniques and methods.

**Data collection**: Object detection requires immense knowledge source for a clear and meaningful object and event description. To achieve such training and test sets, images will be collected from different websites and by directly taking pictures and upload. Information will be gathered by conducting interviews and document analysis as well.

**Tools and Development Environments**: Free and open-source tools will be used during prototype development. To prepare datasets, LabelImg library and python code will be used to prepare annotate datasets and Keras Environment with python programing language in Anaconda will be used to implement the proposed solution.

**Testing and Evaluation**: The proposed solution will be tested and evaluated in terms of its goals and performance. The accuracy of components will be evaluated using measuring matrices called mean average precession.

## 1.6 Scope and Limitations

This study aims to find out if an image contains mosque or not and locate the regions by drawing bounding boxes if there is any. It focuses on building deep convolutional neural network to detect mosque based on external design of mosque building.

## 1.7 Application of results

Object detection has a broad area of application, especially in robotics. Developing a different detection system starting from data preparation, processing, and detecting an object is time-consuming work. Human beings can easily recognize and detect objects but, for machines such as robots to achieve this human ability, think and differentiate objects like a human, they should be trained. Amongst objects, this research chose mosque buildings which is found in the world and propose one input knowledge to teach robots to detect mosque building.

## 1.8 Thesis Outline

The research paper is subcategorized. Chapter 2 reviews concepts and methods relevant and related to the proposed approach. Chapter 3 presents related works. Chapter 4 presents the proposed detection approach. Chapter 5 implementation and experimental results and finally, chapter 6 conclude the overall work presented in this paper and draws future directions.

# CHAPTER 2 : LITERATURE REVIEW

## 2.1 Introduction

This chapter presents a review of the concepts relevant to obtain a basic understanding of the proposed research work. It gives a detail background understanding on digital image processing techniques and gives an introductory overview of a mosque and its external components.

## 2.2 Mosque

A mosque is a place of worship for Muslim settlements. It is used to perform salat in jama‘at and other religious and social services [22].

The Prophet (S) established the first mosque in Medina. After that, a good number of mosques were built in the Syrio-Byzantinc regions. Gradually they underwent further changes and accepted some more components as its own to differentiate it from other [22].

Mosques have changed by its features from the first one in Medina through those in the medieval period to those of today due to its inherent necessity. The influence of the regional styles, different variations of climate, available building materials and technology, and the mission to become comparable with the monumental and splendid buildings of other communities. It is challenging to generalize the characteristic feature of a mosque in every aspect [20].

## 2.3 Mosque Component

To understanding the external structural design and architecture of a mosque building, different reviews and experts like Shah and building design architects were interviewed. In western and central Asia and North Africa, three different types of mosque namely Arab hypostyle mosque, Persian four-iwan mosque and Indian three dome mosque was built [21].

1) Arab hypostyle mosque: which had neither an external wall nor the main gate, but exhibited two new elements: the qibla wall and mihrab on the one hand and minaret on the other. The minaret was built on a square plan and topped with a ribbed dome.
2) Persian four-iwan mosque; which has four vaulted gates arranged in cross-axial.

3) Indian three dome mosque; which has three domes and an extensive, walled courtyard, three imposing entrances, four short towers, and two high minarets longitudinally banded with red sandstone and white marble.

Besides, different external components can represent or identify mosque buildings from the existed ones [2, 20]

A) Bab Al-Sadir: is the grand entrance. In Arabic, ‘Bab’ means a gate or door and Al-Sadir meaning the frontal.

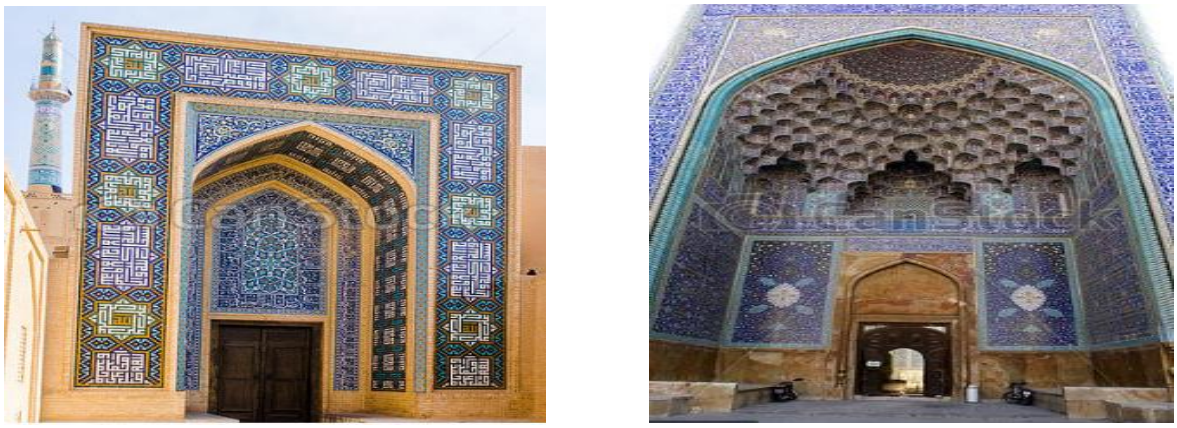


**Figure 2-1 Mosque Bab Al-Sadir**

B) Dome: also called "qubba" in Arabic. It is centrally located over the main prayer hall and there may be more than one dome to a mosque and has two practical functions; one is to echo the words of the Imam inside the mosque and the other is to cool the hot air when it rises upwards and draws in cooler air from outside.

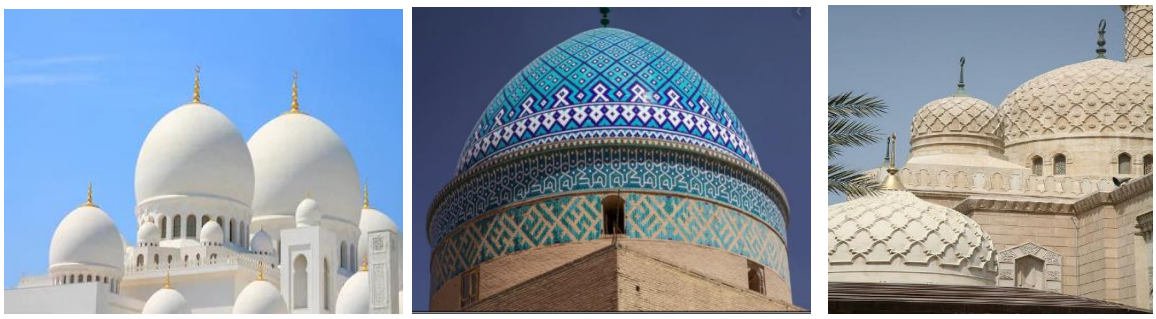


**Figure 2-2 Mosque Dome**

C) Minaret: is usually is made in a form of a circular, octagonal, or square tower which projects above the mosque with at least one balcony along its length. It is possible to

provide more than one minaret in a mosque and more than one balcony for each Minaret. Minaret has stairs or a lift leading to the upper ambulatory, which is usually covered. Nowadays the call to prayer is virtually always relayed by loudspeakers, although this is not permitted in some countries. Also, there are neither bells nor organs in Islam.

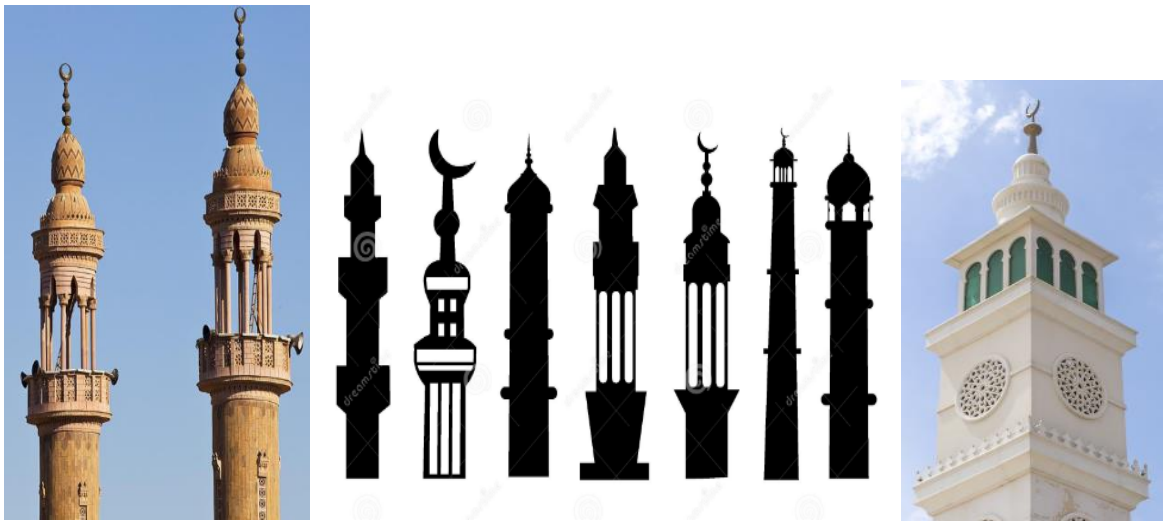


**Figure 2-3 Mosque Minaret**

## 2.4 Digital Image Processing

### 2.4.1 Image

An image to be processed by a computer should be represented using an appropriate discrete data structure. An image is a projection of a 3-dimensional scene into a 2-dimensional function, *f* (x, y), where x and y are spatial (plane) coordinates, and the amplitude of *f* at any pair of coordinates (x, y) is called the intensity or gray level of the image at that point. When x, y, and the amplitude values of f are all finite, discrete quantities, the image is called a digital image [22]. The finite number of elements in an image has a particular location and values called image elements or pixels.

### 2.4.2 Representing Digital Image

Digital images, *f* (x, y), can be represented in three ways. The first one that images can be represented by plotting the function, with two axes determining the spatial location and the third axis being the values of *f* (intensities) as a function of the two spatial variables *x* and *y*. This

representation is useful when working with gray-scale sets whose elements are expressed as triplets of the form (x, y, z), where *x* and *y* are spatial coordinates and *z* is the value of *f* at coordinates (x, y).

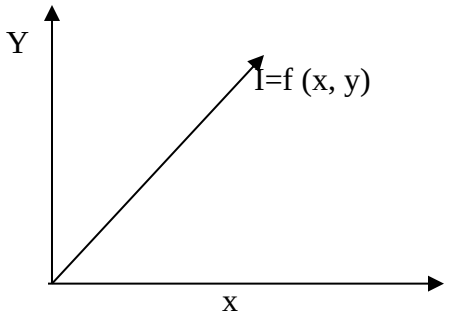



**Figure 2-4 Representation of image using coordinates**

The second type representing a digital image, could commonly appear on the photograph in which the intensity of each point is proportional to the value of *f* at that point. The third type of digital representation is by using matrix representation. A complex digital image also has a matrix representation of a two-dimensional image using a finite number of points cell elements, usually referred to as pixels (picture elements, or PEL). Each pixel is represented by numerical values: for grayscale images, a single value representing the intensity of the pixel (usually in a [0, 255] range) and for color images, three values (representing the amount of red (R), green (G), and blue (B)) are stored.

$$f(x, y) = \begin{bmatrix} f(0.0) & f(0.1) & \dots & f(0.N-1) \\ f(1.0) & f(1.1) & \dots & f(1.N-1) \\ \vdots & \vdots & & \vdots \\ f(M-1.0) & f(M-1.1) & \dots & f(M-1.N-1) \end{bmatrix}$$

**Figure 2-5 Representation of image using matrix**

M is the number of rows and N is the number of columns in some intensities (f) value.

Digital images can be processed by employing a digital computer. Processing digital images using a digital computer is known as digital image processing. Image processing involves innumerable disciplines, mainly computer science, mathematics, psychology, and physics. Other areas, such as artificial intelligence, pattern recognition, machine learning, and human vision, are also involved

in image processing. Some common examples of digital image processing are fingerprint recognition, processing of satellite images, weather prediction, character recognition, and face recognition.

There are no clear-cut boundaries in where image processing end to computer vision, however, there are three types of computerized processes: low-level, mid-level, and high-level process.
Low-level processes are those that have inputs and outputs of images characteristics and involve primitive operations such as image preprocessing to reduce noise, contrast enhancement, and image sharpening.

Mid-level processing involves tasks such as segmentation. Unlike low-level processing, in mid-level its inputs are images, but its outputs are attributes extracted from those images.

High-level processing involves making sense of a group of recognized objects like classification, tracking [23]

### 2.4.3 Fundamental Steps of Digital Image Processing

Processing an image have two methods, one is, whose input and output are images, and the Processing an image has two methods, one is, whose input and output are images, and the other is, methods whose inputs may be images but whose outputs are attributes extracted from those images.

There are different steps to process a digital image in a computer for different purposes a with different objectives. However, the fundamental steps that every image processing application passes through are discussed. It also is shown in Figure 2.7 below [24].

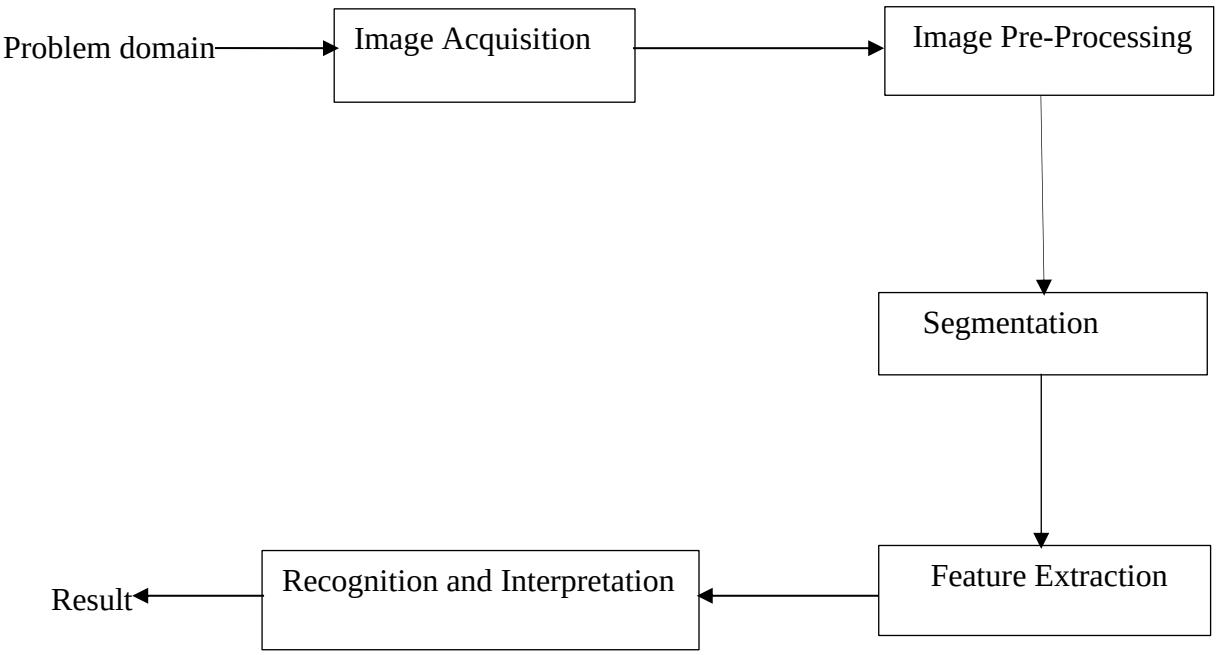



**Figure 2-6 Fundamental steps of DIP**

## 1) Image acquisition

Real-world data needs to be sensed or converted using an analog to digital converter. Therefore, in image processing, the action of retrieving or sensing an image from source is known as Image acquisition. Performing image acquisition is the first step in the working flow sequence, without an image, no processing is possible. The image that is acquired is completely unprocessed.

## 2) Image pre-processing

Pre-processing image is a low-level processing technique to bring out detail that is obscured, or simply to highlight certain features of interest in an image. Techniques that are used to process the image before they fed into high level are:

a) **Image Re-sampling:** the process of transforming an image from one coordinate system to another. The pixel dimensions and size of an image changes by removing noise, sharpening an image, and brighten an image different in the pixel.

b) **Image enhancement**: a process of adjusting digital images to be more suitable by improving the quality by directly applying techniques like removing noise, sharpening an image, and brighten an image different in the pixel.
c) **Image restoration**: recovering an image by removing or reducing degradations which are included during the acquisition of an image such as noise or out of focus blurring using prior knowledge of the degradation phenomena by applying different filtering techniques.

## 3) Image segmentation

Image segmentation is the process of partitioning the image into multiple regions or objects depending on the problem being solved and separate objects from the image background. Segmenting an image is used to assign a label to every pixel in an image so that pixels with the same labels share certain visual characteristics and stopped when the objects or regions of interest in an application have been detected resulting in a set of contours extracted from the image**.**

The image segmentation can be classified into two basic types: Local segmentation (concerned with a specific part or region of the image) and Global segmentation (concerned with segmenting the whole image) [25, 26, 27].

### 3.1) Image Segmentation Technique

There are many segmentation techniques, but they can be categorized into detection of discontinuities and detection of similarities. The segmentation technique based on detection of discontinuities is the process of partitioning an image based on abrupt changes in intensity. Examples of such algorithms include all edge detection algorithms. On the contrary, detection of similarities is based on continuities. These techniques divide the entire image into sub regions depending on some similarity rules. Examples of such algorithms include thresholding, region growing etc. Whether the segmentation technique used is discontinuity or similarity, the end result of any segmentation process is a binary image [25-27]

### A. Thresholding Method

Thresholding is computationally the oldest, cheap, and fast segmentation method which divides the image pixels to their intensity level. It is used to converts the grey scall image into binary images**.** A brightness constant or threshold can be determined to segment objects and backgrounds. The selection of these methods can be manual or automatic. It can be based on a conceptual or actual consideration of the image histogram or prior knowledge. This segmentation process is the simplest because of many objects or image regions are characterized by constant reflectivity or light absorption of their surfaces. If objects do not touch each other, and if their grey levels are distinct from background grey levels, thresholding is a suitable segmentation method. Complete segmentation can result from thresholding in simple scenes.

### B. Edge Based Segmentation Method

An image is segmented by identifying the boundaries or location where the intensity has been changed of an object by using edge detecting operators. Edge detection techniques locate the edges where either the first derivative of intensity is greater than a particular threshold or the second derivative has zero crossings. These edges mark image locations of discontinuities in grey level, color, and context.

### C. Region Based Segmentation Method

Region-based segmentation group pixels into larger regions based on the similarity according to the predefined similarity criteria. This segmentation starts from selecting an initial seed point and starts to begin the region and grow to the adjacent point and grouping the pixel or sub-regions into larger regions based on pre-defined criteria.

### D. Clustering Based Segmentation Method

Clustering Based Segmentation technique segment the image into clusters having pixels with similar characteristics.

### E. Watershed Based Methods

The watershed-based methods attempt to separate touching objects and uses the concept of topological interpretation. In this method, the intensity represents the basins having a hole in its minima from where the water spills.

### F. Partial Differential Equation Based Segmentation Method

Partial differential equation-based segmentation method (PDE) is used to enhance the edges and to remove noise. The results of the PDE method is blurred edges and boundaries that can be shifted by using close operators. This technique is appropriate for time-critical applications.

### G. Artificial Neural Network Based Segmentation Method

Artificial Neural Network Based Segmentation Method (ANN) is used to separate the required image from the background using a connected node of neural networks. Table 2.3 will summarize the segmentation techniques that were discussed before.

**Table 2:1 Comparison of different segmentation technique**

| No | Segmentation Technique | Advantage | Disadvantage |
|---|---|---|---|
| 1 | Thresholding Method | ✓ Previous information is not needed. | ✓ Spatial details are not considered.<br>✓ Highly dependent on pixels. |
| 2 | Edge Based Method | ✓ Good for images that have good contrasting between objects. | ✓ Not suitable for too many edges |
| 3 | Region Based Method | ✓ Useful to define similarity criteria. | ✓ Expensive in terms of time and memory. |
| 4 | Clustering Based Method | ✓ Useful for real problems. | ✓ Determining membership function is not easy. |
| 5 | Watershed Based Methods | ✓ Results are stable and detected boundaries are continuous. | ✓ Complex to calculate gradient. |
| 6 | PDE Based Methods | ✓ Best for time critical applications. | ✓ Computational complex. |
| 7 | ANN Based Method | ✓ Simple program. | ✓ Training time takes more time. |

## 4) Feature Extraction

Segmented pixels usually are represented and described in a form suitable for further computer processing like, feature extraction. Feature extraction is the process of describing an object by reducing them to a form suitable for computer processing called an image feature. Representing a region involves two choices: Region can be represented in terms of its external characteristics (its boundary) when the primary focus is on shape characteristics, or, internal characteristics representation (the pixels comprising the region) where the primary focus is on regional properties, such as color and texture. The most popular feature among them is color, texture, and shape of an image.

### A) Shape Feature

Recognition of image regions is one of the most important steps on the way to understanding image data and requires an exact region description in a form suitable for finding and matching shapes, recognizing objects, or making measurement of shapes. The shape of an object is determined by its external boundary abstracting from other properties such as color, content, and material composition, as well as from the object's other spatial properties, and defining the shape of an object can prove to be very difficult. The shape feature is used to encode simple geometrical forms such as straight lines in different directions. The extraction techniques are classified as region-based and contour-based.

Region identification is necessary for region description. One of the methods for region identification is to label each region (or each boundary) with a unique (integer) number; such identification is called labeling or coloring, and the largest integer label usually gives the number of regions in the image or use a smaller number of labels to ensure that no two neighboring regions have the same label; then information about some region pixel must be added to the description to provide full region reference.

The contour methods calculate the feature from the boundary and ignore its interior. Region borders must be expressed in some mathematical form.

**B) Color Feature**

Color features are defined subject to a particular color space or model and can be extracted from images or regions. It has been successfully applied to recognize images because it has very strong correlations with the underlying objects in an image. Moreover, the color feature is robust to background complication, scaling, orientation, perspective, and size of an image.

**C) Texture Feature**

Texture Feature refers to surface characteristics and appearance of an object by the size, shape, density, arrangement, the proportion of its elementary parts. It extracts feature by computing statistics of pixels or by finding the local pixels structures in the original image domain and it is measured from a group of pixels.

## 5) Recognition and Interpretation

In object recognition, it was known that the human brain processes visual information in semantic space mainly, that is, extracting the semantically meaningful features such as line-segments, boundaries, and shape [1].

As for computers, it is a challenge than humans. Recognition is the last step of the bottom-up image processing approach that must be performed after appropriate features have been detected. It is also often used in other control strategies for image understanding.

A pattern or object is an arrangement of descriptors with their respective pattern class (family of patterns that share some common properties). Therefore, Pattern or Object recognition is a process of recognizing, identifying, and localizing objects that are found in the real-world from an image and assign labels to an object based on its description and with as little human intervention as possible. No recognition is possible without knowledge.

Decisions about classes or groups into which recognized objects are classified are based on such knowledge - knowledge about objects and their classes gives the necessary information for object classification. knowledge representation data structures are mostly extensions of conventional data

structures like lists, trees, graphs, tables, hierarchies, sets, rings, nets, and matrices. Descriptions and features can be used for representing knowledge as a part of a more complex representation structure. Descriptions usually represent some scalar properties of objects and are called features. Typically, a single description is insufficient for object representation, therefore the descriptions are combined into feature vectors.

Based on the detected features in an image, one must formulate hypotheses about possible objects in the image. These hypotheses must be verified using models of objects. Not all object recognition techniques require strong hypothesis formation and verification steps.

There is a qualitative way to consider the complexity of the object recognition task would consider before computing object recognition tasks. Some of this is,

- Scene constancy: this complexity will depend on whether the images are acquired in similar conditions.
- Image-models' spaces: images may be obtained such that three-dimensional objects can be considered two-dimensional.
- Number of objects in the model database: If the number of objects is very small, one may not need the hypothesis formation stage.
- Number of objects in an image and possibility of occlusion: If there is only one object in an image, it may be completely visible. With an increase in the number of objects in the image, the probability of occlusion increases.

## 5.1) Recognition strategies

The following section discusses some basic recognition strategies used for recognizing objects in different situations.

### 1) Classification

The process of assigning land cover classes to pixels or to recognize objects based on features is known as pattern or object classification. The analyst identifies homogeneous groups of pixels having similar values and labels the groups as information classes such as water, agriculture, and forest. While generating a thematic map, thematic information is extracted with the help of

software, it is known as digital image classification. There are two types of image classification approaches in which classification is performed, supervised and unsupervised classification.

Supervised image classification is a process of identifying a class is with a remote sensing data with inputs from and as directed by the user in the form of training data. Besides unsupervised learning is a process of automatic identification of structures within remote sensing. It does not require extensive prior knowledge to study the required area. To classify a set of data into different classes or categories, the relationship between the data and the classes into which they are classified must be well understood. To achieve this by computer, the computer must be trained.

Nonparametric machine learning algorithms do not make strong assumptions about the form of the mapping function. Assumption free, they are ready to learn any functional form from the training data. Nonparametric methods are good when there is a big data with no prior knowledge, when we do not want to worry too much about choosing just the right features but require a lot more training data to estimate the mapping function and slower to train as they often have far more parameters to train. The most common classification algorithms will be discussed below [18].

Nearest Neighbor Classifiers: has no model other than storing the entire dataset and make predictions based on the most similar training patterns for a new data instance. The method does not assume anything about the form of the mapping function other than close patterns is likely to have a similar output variable.

Bayesian Classifier: has been used for recognizing objects when the distribution of objects is not as straightforward. It uses a probabilistic knowledge about the features of objects and the frequency of the objects. This classification is effective in a large number of problem domains.

Off-Line Computations: assign a class to each point in the feature space, all computations are done before the recognition of unknown objects begins called Off-Line Computations. The recognition process can be effectively converted to a look-up table and hence can be implemented very quickly.

Support Vector Machines (SVM): is one of the best-known methods in pattern and image classification. It is designed to separate a set of training images into different classes. It classifies by finding a hyperplane in N-dimensional space (N-the number of features) that distinctly

classifies the data points and predicted to belong to a category based on which side of the gap they fall.

Neural Nets: Neural Nets or Artificial Neural Networks (ANN) is an information processing paradigm that is inspired by the way biological nervous systems process information. We are constantly analyzing the world around us. Without conscious effort, we make predictions about everything we see and act upon them. At first glance, we label every object based on what we have learned in the past. Neural networks work with the same concept by adjusting the connection exist between neurons. It is composed of a large number of highly interconnected processing elements called neurons, which convert an input vector into some output.

### 2) Object detection

Object detection is another subset of object recognition strategy, aims to identify all target objects in the target image and determine the categories and position information from a still image or video data to achieve machine vision understanding [28, 3].

In earlier times, computer vision problems had a fair amount of success when machine learning was embraced. These problems were primarily solved by using traditional methods [29]. Traditional object detection algorithms are composed of three components. First, by applying sliding windows to search over the whole images, a region of interest (RoI) is generated to propose a region with object existence associated with localization. Second, feature extraction is performed by traditional hand-crafted features, such as the prevalent SIFT or HOG to extract features at the low-level feature information for each RoI. Third, the proposed feature vectors are classified by the pre-trained classifiers like SVM, Adaboost or Random Forest, and then the bounding boxes are figured out.

This detection algorithm performs three independent stages which result in large time-consumption in training and testing, adopting hand-crafted features makes detectors have no robustness to the varieties of object forms, illuminations, backgrounds, and viewpoints, global optimization is difficult to reach since the optimizations of feature extractor and classifier are separated. Also, the amount of effort required for hand-crafted features is firstly overwhelming. Moreover, it requires a fair amount of domain knowledge and is very specific to the use-case.

Therefore, due to this problem another Artificial neural network (ANN) detection algorithm is proposed, which is known as a deep convolutional neural network (DCNN).

Artificial Neural Networks (ANN) is an information processing paradigm composed of a large number of highly interconnected processing elements or network called neurons. These neural networks have the potential for solving problems in which some inputs and corresponding output values are known, but the relationship between the inputs and outputs are either not well understood or difficult to translate into a mathematical function. Hence, ANN resembles a human brain in two respects. The first property is that knowledge is acquired by the network through a learning process. The other is interneuron connection strengths known as weights are used to store the knowledge. The weights on the connection encode the knowledge of networks.

There are two basic phases in neural network operation. The training or learning phase and testing-recall or retrieval phase. In the learning phase, input data is repeatedly presented to the network, while weights are updated to obtain the desired response. In the testing phase, the trained network with frozen weights is applied to another input data that it has never used.

Learning through ANN based on the number of neurons can be classified into shallow and deep learning. Shallow learning has a smaller number of neurons where deep learning is a set of learning methods attempting to model data with complex architectures combining different non-linear transformations and learning from data sets without any manual design of feature extractors [10]. One of a powerful deep image processing learning method is a Convolutional Neural Network (CNN), use deep learning or for Deep Convolutional Neural Networks (DCNN) to perform both generative and descriptive tasks, often using machine vison that includes image and video recognition, along with recommender systems and natural language processing (NLP).

The idea for Deep Convolutional Neural Networks (DCNN) was originally published by Hinton and Sutskever, it is used to achieve state-of-the-art performance in the ImageNet Classification challenge. This research then revolutionized the field of computer vision.

A key advantage of deep learning is its ability to perform semi-supervised or unsupervised feature extraction over massive datasets. The ability to learn the feature extraction step present in deep learning-based algorithms comes from the extensive use of convolutional neural networks (ConvNet or CNN) [30, 31].

Convolutional Neural Networks or CNNs, is a special kind of neural network that uses convolution in place of general matrix multiplication in at least one of the layers for processing data. CNN is based on learning levels of representations. The higher-level concepts are defined from lower-level ones, and the same lower lever concepts can help to define many higher lever concepts. It learns multiple levels of representation and abstraction which helps to understand datasets such as images, audio, and text [32, 33, 34].

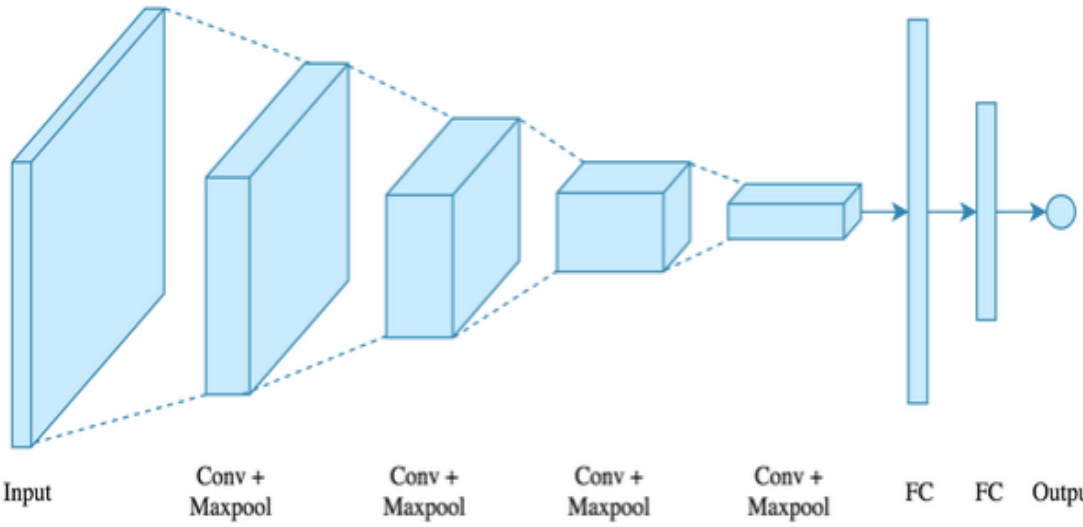



**Figure 2-7 Convolutional Neural Network**

CNN uses three main types of layers to build Convolution Network architectures: Convolutional Layer with activation functions**,** Pooling Layer, and Fully-Connected Layer.

### A. Convolutional Layer

Convolutional layers or convolutional stage is a special operation applied on a particular matrix (usually inputs which have a multidimensional array, that can be image pixels or their transformation, patterns, time series, or video signals) using another filter-matrix. The operation involves multiplying the values of a cell corresponding to a particular row and column, of the image matrix, with the value of the corresponding cell in the filter matrix. which aims at extracting local features from the input, and each kernel matrix is used to calculate a feature map or kernel map.

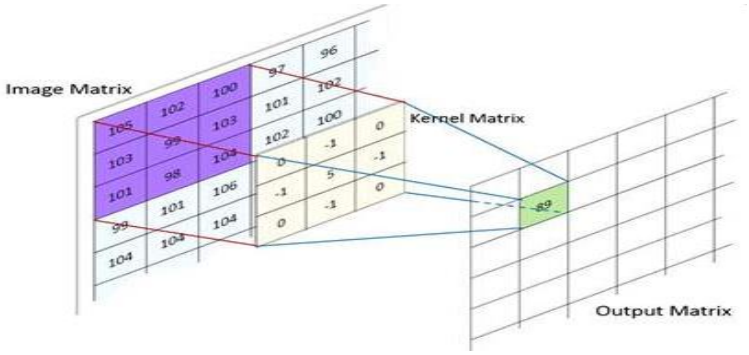



**Figure 2-8 Convolutional Layer feature extraction**

When the kernel moves across the image, scanning each pixel and converting the data into a smaller, or sometimes larger format, therefore, to assist the kernel with processing the image, a concept called padding is applied. Padding is valued which is added to the frame of the image to allow for more space for the kernel to cover the image to perform an accurate analysis of images.

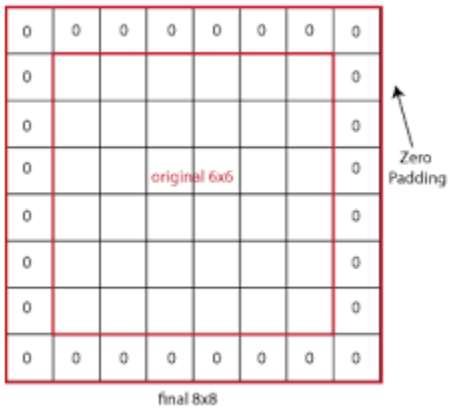



**Figure 2-9 Zero Padding**

In each neuron in the network, to determines the activation, based on whether each neuron's input is relevant for the model's prediction, a mathematical equation called activation function is attached. The purpose is to introduce non-linearity into our network because the images are made of different objects that are not linear to each other so the images are highly non-linear. There are many types of activation functions but the three common types of Activation Functions are binary, linear, and nonlinear function.

Binary Function: is a threshold-based classifier. It determines whether the neuron should be activated based on the value from the linear transformation or not. In other words, if the input to the activation function is greater than a threshold, then the neuron is activated, else it is deactivated,

its output is not considered for the next hidden layer. The binary step function can be used as an activation function while creating a binary classifier.

Linear Activation: is the activation function that takes the inputs, multiplied by the weights for each neuron, and creates an output signal proportional to the input. Linear function turns the neural network into just one layer and can be used if we have multiple outputs, not just yes and no.

Non-Linear Activation Functions: allow the model to create complex mappings between the network inputs and outputs. It is essential for learning and modeling complex data, such as images, video, audio, and data sets which are non-linear or have high dimensionality. They allow backpropagation because they have a derivative function which is related to the inputs. Hence using nonlinear Activation, we can generate non-linear mappings from inputs to outputs. Some of the most common nonlinear Activation function are: one is, Sigmoid activation function, which is used only on the output layer so that we can easily interpret the output as probabilities since it has restricted output between 0 and 1. It is computationally expensive. The other is, RELU activation function, form of f(x) = max (0, x). Compared to other activation functions, it is computationally expensive.

### B. Pooling Layer

The Pooling layer consists of performing the process of extracting a particular value from a set of values, usually the max value or the average value of all the values. This reduces the size of the output matrix. The two types of pooling method are:

**Max-Pooling:** The max value among all the values of saying a M X M part of the matrix will be taken. Thus, we are actually taking in the values denoting the presence of a feature in that section of the image to getting rid of unwanted information regarding the presence of a feature in a particular portion of the image and considering only what is required to know.
For example: if we take a feature map with 4 x 4 pixels from the convolution network as shown in the figure 2-12 below, using a 2x2 stride and taking their maximum value, a 2x2 feature map is created.

| | | | |
|---|---|---|---|
| 4 | 3 | 1 | 5 |
| 1 | 3 | 4 | 8 |
| 4 | 5 | 4 | 3 |
| 6 | 5 | 9 | 4 |

Figure 2-10 Feature map sample

| | | | |
|---|---|---|---|
| 4 | 3 | 1 | 5 |
| 1 | 3 | 4 | 8 |
| 4 | 5 | 4 | 3 |
| 6 | 5 | 9 | 4 |

Max([4, 3, 1, 3]) = 4

Figure 2-11 Max_pooling on the feature map

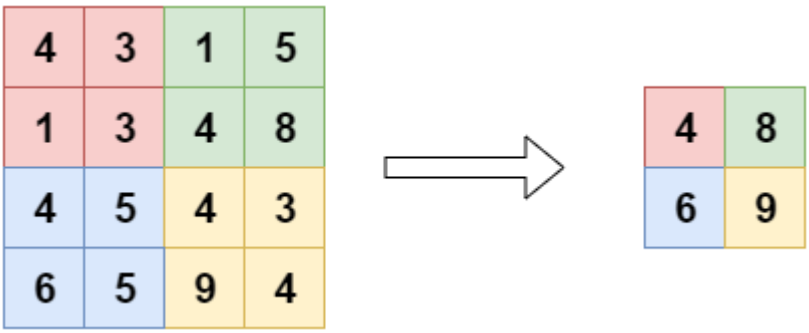



Figure 2-12 Result of pooling operation

**Average Pooling:** Calculate the average value for each patch on the feature map. For example, the same is by using a 4*4 feature map, to create a smaller size 2*2 feature map by calculating the average in the 2*2 stride will be shown in the figure below.

| | | | |
|---|---|---|---|
| 4 | 3 | 1 | 5 |
| 1 | 3 | 4 | 8 |
| 4 | 5 | 4 | 3 |
| 6 | 5 | 9 | 4 |

Avg([4, 3, 1, 3]) = 2.75

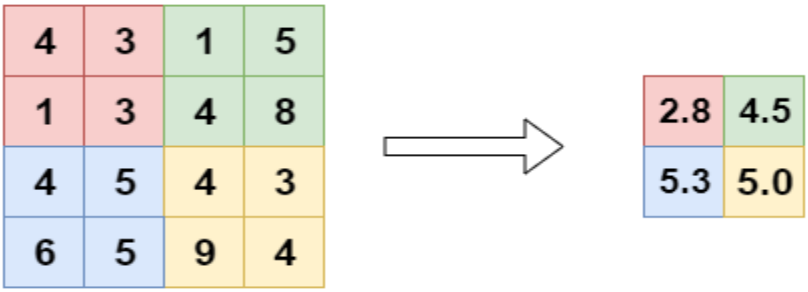



**Figure 2-13 Result of average pooling**

### C. Fully connected layer

This is the last stage of the topology of CNNs consisting of a generic multi-layer network to perform high-level reasoning. They exhibit a full connection to all the decision functions in the previous layer and convert 2D features into a one-dimensional feature vector. From these layers it is possible to extract features to train another classifier and to specify how the network training penalizes the deviation between the predicted and the true labels, various loss functions can be used. Fully connected layers possess a large number of parameters and so require powerful computational resources. Thus, the 'deep' in a 'Deep CNN' refers to the number of layers in the network. Repeating feature extraction or convolution layer process multiple times results in deeper convolutional neural networks.

DCNN can be mainly divided into two types based on the process performance, namely, one stage detection or two-stage detection.

A single-stage detection treats object detection as a simple regression problem by taking an input image and learning the class probabilities and bounding box coordinates. Such models reach lower accuracy rates but are much faster than two-stage object detectors. The most used one stage detection algorithms are YOLO (You Only Look Once) and SSD (Single Shot Multi Box Detector). Algorithms like YOLO and SSD use a fully convolutional approach in which the network can find all objects within an image in one pass (hence 'single-shot' or 'look once') through the convent [35].

### 1) You Only Look Ones (YOLO)

Is a single convolutional network to simultaneously predict multiple bounding boxes and class probabilities for predicted boxes. The input image is divided into *S x S* grid of cells and for each object that is present on the image, one grid cell is said to be "responsible" for predicting the object. Each grid cell predicts B bounding boxes and confidence scores for those boxes. These confidence scores reflect how confident the model is that the box contains an object and also how accurate it thinks the box is that it predicts. The bounding boxes having the class probability above a threshold value is selected and used to locate the object within the image [36].

### 2) Single Shoot Detection (SSD)

Take an input image which is divided into S X S cell and apply a small convolutional filter to predict object categories and offsets in bounding box locations. To detect multiple objects that are found in the same cell, it uses separate predictors (filters) for different aspect ratio detections called anchor boxes and applying these filters to multiple feature maps from the later stages of a network to perform detection at multiple scales [37].

A two-stage detection treats object detection in two stages. First, by using a Region Proposal Network (RPN), the interest of regions ware generated. Secondly, send the region proposals down the pipeline for object classification and bounding-box regression. Such models reach the highest accuracy rates but are typically slower. Two-stage detection algorithms are:

### 1) Region based Convolutional Neural Network (RCNN)

By combining high capacity convolutional neural network to bottom-up region, a region proposal with a convolutional neural network, short-form R-CNN to localize, and segment objects is designed. R-CNN is composed of different modules. First, category independent regions that are warped into a square will be proposed to define a set of candidate detections using a selective search algorithm. Second, the proposed regions will then feed to a large forwardly propagated convolutional neural network to extract the fixed length of features vector from them. Finally, based on a predefined feature map, features will be feed in to support vector machine (SVM) to classify the presence of the object within that candidate region proposal and four values which are offset values to increase the precision of the bounding box [38].

### 2) Spatial Pyramid Pooling Network (SPPNet)

Since RCNN required fixed input size, accuracy decreases. Therefore, the spatial pyramid pooling network was proposed to use the SPP layer on top of the last convolutional layer was proposed. The SPP layer pools the features and generates fixed-length vector output by aggregating at a deeper stage of the network hierarchy to avoid the need for cropping or warping at the beginning, which is then fed into the fully connected layers (or other classifiers) [39].

### 3) Region based Fully Convolutional Network (RFCN)

Region-based Fully Convolutional Network for object detection consists of shared, fully convolutional architectures as is the case of FCN [16]. To incorporate translation variance into FCN, we construct a set of position-sensitive score maps by using a bank of specialized convolutional layers as the FCN output. Each of these score maps encodes the position information with respect to a relative spatial position (e.g., "to the left of an object"). On top of this FCN, we append a position-sensitive RoIPooling layer that shepherd's information from these score maps, with no weight (convolutional/fully_conv) layers following [40].

### 4) Fast Region based Convolutional Neural Network (Fast-RCNN)

The Fast-RCNN consists of a CNN with a replaced pooling layer by an "ROI pooling" layer and its final FC layer is replaced by two branches, several classes plus a background category layer branch, and a category-specific bounding box regression branch. To train Fast-RCNN, input images are first fed into a Convolutional Neural Network to produce a similar input image sized feature map, which is shared by the last convolution layer are obtained [41].

Meanwhile, by using selective search, object proposal windows are obtained. The proposed regions on the feature map are then fed into a spatial pyramid pooling layer called the ROI Pooling layer. This layer divide features from the selected proposal windows (that come from the region proposal algorithm) into the sub-windows size and performs a pooling operation in each of these sub-windows to get a fixed-size output features size irrespective of the input size. The output features from the ROI Pooling layer are then fed into the successive FC layers to the classification branch produces probability values of each ROI belonging to K categories and one catch-all background category and make bounding boxes for the respective class.

**Loss function**: Is one of the parameters required to quantify how close a particular neural network is to the ideal weight during the training process.

The classification loss $Lc_s\,(p,u)$ is given by $-\log(p_u)$ which is the log loss for the true class u of the probability of category p. The regression branch produces 4 bounding box regression offsets $t^k_i$ where $i = x, y, w,$ and $h$. $(x, y)$ stand for the top-left corner and $w$ and $h$ denote the height and width of the bounding box. The true bounding box regression targets for a class u are indicated by $v_i$ where $i = x, y, w,$ and $h$ when $u \geq 1$. The case where $u=0$ is ignored because the background classes have no ground-truth boxes.

### 5) Faster Region based Convolutional Neural Network (Faster-RCNN)

Faster RCNN is a a deep fully convolutional network with two models, one, region proposal network (RPN) for proposing regions and the other is, a Fast RCNN detector for detecting an object of interest as a single unified network [42].

The region proposal network (RPN) is a fully convolutional network to find out the possible locations of the target and each with an objectness score.

To generate region proposals, we slide a small network over the convolutional feature map output by the last shared convolutional layer. This small network takes as input an n x n spatial window of the input convolutional feature map of the lower-dimensional feature. This feature is fed into two siblings fully connected layers a box-regression layer (reg) and a box-classification layer (cls). The Region Proposal network also introduced a novel concept called Anchor boxes.

Anchor boxes are some of the most important concepts in Faster R-CNN. These are responsible for providing a predefined set of bounding boxes of different sizes and ratios for reference when first predicting object locations for the RPN. These boxes are defined to capture the scale and aspect ratio of specific object classes you want to detect and are typically chosen based on object sizes in the training dataset. Anchor Boxes are typically centered at the sliding window. For a down sampling ratio d, the feature map will have dimensions W/d * H/d. In an image, each anchor point will be separated by d spatial pixels, since we have just one at each spatial location of the feature map. Making it too low or too high can give rise to localization errors. One way to mitigate these localization errors is to learn the offsets applied to each anchor box which is the goal of the regression layer.

**Loss function:** Is one of the parameters required to quantify how close a particular neural network is to the ideal weight during the training process. To train RPNs, we assign a binary class label which predicts being an object or not to each anchor by assigning a positive label to two kinds of anchors: first, the anchor/anchors with the highest Intersection-over- Union (IoU) overlap with a ground-truth box, or second, an anchor that has an IoU overlap higher than some value n1 with any ground-truth box. Also, a negative label to a non-positive anchor is assigned if its IoU ratio is lower than some value n2 for all ground-truth boxes. Anchors that are neither positive nor negative do not contribute to the training objective. To compute bounding-box regression loss, the anchor box to a nearby ground-truth box is calculated.

The classification scores layer is a (w, h, k*2) matrix that corresponds to the foreground and background probabilities of k regions at each pixel in a feature map. Regression coefficients for the boxes layer is a (w, h, k*4) matrix that corresponds to the position offset of each bounding box relative to the preset anchor box (1 region needs to predict 4 values of the prediction area x, y, w, h).

The loss function of RPN is the sum of classification loss and bounding box regression loss. Classification loss uses cross-entropy loss to punish misclassified boxes. Regression loss uses the distance function between the true regression coefficients (calculated using the closest foreground anchor box to match the ground truth box) and the regression coefficients predicted by the network. Finally, there are N proposals or Region of Interests (ROI) from the Region proposal network. Each proposal has five values, the first one indicating the label and the rest of the four are proposal coordinates.

Region of Interest pooling (RoIPooling) maps the ROI of the image used for training to the last feature layer. After obtaining the ROI coordinates, we divide this ROI area into height x width parts, each part is pooled, finally, each pooled concatenate is input to the next layer. Through the ROI Pooling Layer operation, all ROIs, regardless of ROI size, generate a vector of fixed length to the next layer.

The problem of training is equivalent to the problem of minimizing the loss function. The procedure used to carry out the learning process in a neural network is called the optimization algorithm. Many different optimization algorithms can be algorithms that can gradient-based or not.

**Sharing features for RPN and Fast R-CNN**

To detect objects a technique that allows for sharing convolutional layers between the two networks, instead of learning two separate networks, alternating training is used. A technique in which RPN is first trained, and use the proposals to train Fast R-CNN. The network tuned by Fast R-CNN is then used to initialize RPN, and this process is iterated.

The training consists of the following steps:

1. Take input data through network parameter to extract feature from them.
2. Creating anchor targets for feature map.
3. Locations and objectness score prediction from the RPN network.
4. Taking the top N locations and their objectness scores aka proposal layer.
5. Passing these top N locations through Fast R-CNN network and generating locations and class predictions.
6. Generating proposal targets for each location suggested in step 4.
7. Using step 2 and step 3 to calculate **rpn_cls_loss** and **rpn_reg_loss.**
8. Using step 5 and step 6 to calculate **roi_cls_loss** and **roi_reg_loss.**

**Table 2:2 Summary of deep learning algorithm**

| No | Algorithm | Drawback |
|---|---|---|
| 1 | RCNN | ✓ R-CNN is slow because it performs a ConvNet forward pass for each object proposal, without sharing computation.<br>✓ Multi-stage pipeline |
| 2 | SPPNets | ✓ Faster than R-CNN because it performs a ConvNet forward pass for each object proposal, with sharing computation.<br>✓ Multi-stage pipeline |
| 3 | Fast-RCNN | ✓ Selective Search |
| 4 | Faster-RCNN | ✓ The performance of systems depends on how the previous system has performed |
| 5 | RFCN | ✓ Global average pooling layer to provide the global context |

# CHAPTER 3 : RELATED WORK

## 3.1 Introduction

This chapter presents research works done related on object detection. Research works done in the area focuses on improving a single component of an object detection system [7, 13] [16-19] or dealing with generic architecture [8-12] [14, 15] are reviewed along with different levels of the object detection process.

## 3.2 Vehicle detection

Vehicle detection system using urban nighttime and extremely dark views images is developed by [7]. To improve the computational efficiency of the detection system, the datasets was captured on a clear, blurry, occluded, and small size of bicycles, buses, cars, motorbikes, pedestrians, vans, and truck images. The images are then labeled using PASCAL VOC challenge style for later use on the two CNN architectures. VGG16 convolutional neural network architecture is used by fixing the parameters of the first four convolutional layers with suitable max-pooling layers and freed the other neural networks layers and ResNet101 convolutional neural network architecture is used by fixing the parameters of the first convolutional layer with its corresponding max-pooling layers and a single residual block to extract features from the labeled image by using pretrained weight which is trained on ILSVRC-2012-CLS datasets for image recognition and object detection contest. The Vehicle detection system is trained on Faster region-based CNN model.

During the experiment, Faster region-based CNN model-based with ResNet101 detector achieve mean average precession (map) of 0.8497.

Vehicle detection system using aerial and ortho-normalized imagery is proposed by [8]. Input images which are found in three visible color channels and one near-infrared channel in state-of-the-art object detection method called faster R-CNN is used. The input images are pre-processed by down sampled by two to obtain images with a spatial resolution of 512×512 and extract their features using VGG16. The region proposal network (RPN) is used to propose regions by using single anchor (k = 1), with a scale of 402 pixels for the 12.5 ground sampling distance (GSD) of 12.5 cm per pixel (cmpp) imagery and a scale of 202 pixels for the 25 cmpp GSD imagery. Taking

proposed regions, region-of-interest (ROI) pooling is used to select the highest value before feeding to the training model.

During the experimentation, by comparing the results of faster R-CNN detection models to the HOG+LBP+SVM models, the faster R-CNN-based models yield the highest average precession rates which are above 90%.

## 3.3 Building detection

Building and urban area detection system using very high-resolution satellite and aerial images is developed by [10]. Ikonos satellite images were used and pre-processed by up sampling the image in six to each coordinate axis using bilinear interpolation and eliminate noise using bilateral filtering. Using the pre-processed input, key points of a template was extracted by scale-invariant feature transform (SIFT) method and graphs were matched using multiple subgraph matching methods between templates and test images. Informing the graph, each key point is presented as a vertex of the graph. The unary and binary relationship between these vertices such as spatial distance and intensity values leads to edges of the graph. Finally, individual building was detected by cutting edges based on their intensity criteria and hypothesized vertices on the same buildings which have similar intensity values based on their colors. Locating separate building is done by cutting edges based on the hypothesized result.

During the experiment, 89.62% correct urban area detection performance with an 8.03% false alarm rate is obtained. Building detection performance on the test image set is 88.4% with a 14.4% false alarm rate.

Building detection using satellite images is proposed by [9]. To detect buildings, two methods were used. The first method performs fusion using corner points and DSM by first extracting and labeling local feature points using a Harris corner detector and set them a center value of W*W windows on DSM. Using these DSM windows, windows that have the highest value were selected as a kernel formation location. Finally, to detect buildings, asymmetric Gaussian probability density is used to estimate kernel density from kernel formation location and detect those with the highest density as a building. The second method is, detecting buildings using shadow points and DSM. In this method, shadow points were taken as local features. In order to extract features, the

threshold value was used to best separate and label shadow and non-shadow areas using Bimodal histogram splitting method. As in the first method, shadow points are assumed to be in the center of a WXW window on DSM. Windows with a maximum height index are then used to show a voting direction of possible building center to the final kernel density map for detecting possible building locations.

During the experiment, using corner points and DSM, their true detection (TD) and false alarm (FA) rates are 90.3% and 12.9% respectively. Besides, when they use shadow points and DSM, the true detection and false alarm rates of 86.0% and 9.6% were achieved.

## 3.4 Hand and hand tool detection

Hand, meter, nipper, screwdriver, spanner, and tongs detection system is proposed by [11]. In this work, RGB input images with a size of 480 x 720 is pre-processed using Illumination compensation to get good quality images are used. After pre-processing, images were segmented using watershed segmentation algorithm and color, texture, and geometric features are utilized.

To extract feature, Scalable Color Descriptor (SCD) images that are defined in the HSV color space are extracted by an average value of Hue under the different light conditions and 80 edge histogram descriptor (EHD) features are extracted in each object to captures the spatial distribution of edges. The average value of an object that is represented by wavelet decomposition was used. Likewise, the height and width ratio of the rectangle that surrounds the object is extracted as another feature. Finally, to select appropriate features for classification, an improved binary gravitational search algorithm (IBGSA) was used to find an optimal binary vector that each bit corresponds to a feature. Then the selected features were used in artificial neural networks (ANN) for detecting objects.

During the experiment, ANN with KNN was compared. The ANN is built with 227 nodes in the input layer, 25 nodes in the hidden layer, and 7 nodes in the output layer with IBGSA to achieve the highest result of 91.70 %. Only 61.4285 % was achieved by KNN with IBGSA.

## 3.5 Horse Detection

Horse detection system is proposed by [12]. The gPb detector is used to detect edge maps and use segment fitting tools to generate short line map to represent a texture-less image. Minimum Euclidean distance is then used to compute the distance between each line of endpoints to construct graphs according to Floyd algorithm and the angle of lines and link-lines are computed to produce shape vector to generate shape matrix by combining all vectors. Finally, to detect an object, shape matching is utilized and find the maximum bound-box. During the experiment, above 90% correct horse detection is achieved.

## 3.6 Firearm detection

Handgun and rifle firearms detection system is proposed by [14]. The firearm detection is trained on Faster RCNN using Inception v2 network to extract feature. To detect an object, the images are passed through two stages. First; a sliding window is applied to the output of the last layer of the feature extraction network, to find regions in the image that contains objects of interest. Regions with high scores from the first stage are then extracted from the feature map and fed through a classifier that predicts both the object type and a bounding box for each region.

To evaluate image classification, the research focused on only whether the image as a whole contains any firearms and achieves true positive, more than 95%.

## 3.7 Face detection

Face detection system is proposed by [13]. To improve detection accuracy, a framework named FDNet1.0 is proposed for detecting face using small scale, illumination, occlusion, background clutter, and extreme poses conditions input images on a backbone network called ResNet-v1-101 and large kernel-based deformable layers to extract high-level features.

Using the extracted feature, to propose region, anchors are designed to obtain better location samples with the aspect ratio of 1, 1.5, 2, and 16*16, 32*32, 64*64, 128*128, 256*256, 512*512 scall size based on statistical analysis on the training dataset. The anchors with the highest IoU score or IoU score with the ground truth above 0.7 are defined as positive. The anchors whose IoU score with the ground truth that is lower than 0.3 are defined as negative, while whose IoU score

above 0.3 but lower than 0.7 are ignored. Finally, using the proposed regions, object detection was done using R-CNN.

During the experiment, Faster-RCNN model with FDNet1.0 detector shows better accuracy in detecting face with 95.9% of the easy set, 94.5% of the medium set, and 87.9% of the hard set on the validation set is achieved.

## 3.8 Buried object detection

An underground buried object detection system is proposed by [15]. A hyperbola reflection in B-scan ground penetrating radar (GPR) images is used as input data for detection. The approach consists of two main stages, First, a pre-trained Convolutional Neural Network design which has 3 convolution layers of 16, 32, and 64 filters of size 5 * 5 pixels (each one is followed by a ReLu activation and a max-pooling layer of size 2 * 2 pixels) and one fully-connected layer of 64 neurons to apply transfer learning on the detection model on the grayscale Cifar-10 database is used. Next, to detect the object, Faster-RCNN (based on pre trained CNN weights) is trained and fine-tuned using both real and simulated GPR data.

During the experiment, the proposed technique is compared with the COD based on HOG and Haar-like features and provides good performance on testing real data and considerably outperforms detection.

## 3.9 People detection

People detection system called Parallel RCNN using RGB and depth data on Faster RCNN framework's is proposed by [16]. First, a raw color (RGB) image is taken and encoded a three-channel depth image as the inputs of an end-to-end deep neural network. The deep features from two kinds of images are extracted in parallel by two CNNs and then through L2 normalization, two types of features are combined. Using the extracted features, regions were proposed using RPN and classify the proposed region using Fast RCNN. During the training of Parallel RCNN, first, each stream network is trained using Faster RCNN on RGB data and encoded depth data respectively. Then the fully connected part is discarded and concatenate the convolutional parts of each stream to extract the features from each modal data. The pre-trained model for the Parallel

RCNN is trained following the steps similar to the training process of Faster RCNN using a combined parameter.

During the experiment, by comparing their model with Faster RCNN using RGB images, the experiment results indicate that by additionally using jet colormap method for encoding raw depth image, they are able to achieve mAP of 0.915. The result is 1.5% higher than that of Faster RCNN using RGB images only.

### 3.10 Disk interval detection

An automatic disk interval detection system using X-Ray images is proposed by [17]. The proposed system contains ZF network with 5 convolutional neural network layers with Faster RCNN is used. The Deep network is fine-tuned using Faster. The experiment shows average precision of 0.905 and much better performance is achieved compared to the traditional sliding window detection method on handcrafted features.

### 3.11 Pedestrian detection

pedestrian detection system is proposed by [19]. During the development, an extra convolutional layer is added to the network and the input channel is reduced from three-channel input to one. The improved method has a total of 13 layers comprising of an input layer, Serval convolution followed by ReLu, fully connected layers Softmax classifier on Fast RCNN is used and achieve a detection rate of an average of 63.42%.

### 3.12 Ship detection

*S*hip detection system using Faster RCNN is proposed by [18]. The input data, a 3x3 complex covariance matrix is pre-processed into a 3-D real vector to be able to feed into real-valued deep convolutional neural networks to extract their features. Using this output, a 3x3 spatial window of the input convolutional feature map was applied to generate a proposed region. Each sliding window was mapped to a lower-dimension feature then fed the extracted feature into two sibling fully-connected layers: one that produces Softmax probability estimates and another layer that outputs four real-valued numbers for each proposal.

## 3.13 Summary

Many researchers design different object detection systems from image. From digital and satellite images, the presence of instances and their location was detected either by directly dealing with generic architecture or by modelling a new system by improving a single component aiming to improve their speed and accuracy. To use the existed detection model to detect mosque building, some existed models are trained on a specific feature which is provided by the datasets. Therefore, it will result to bad detection result because of different feature representation of different objects [34,35,36].

Likewise, in some researches, the image analysis part is done by shallow networks which need manual feature extraction. Using manual feature extraction is not good for classification tasks because, we specify restrictions on what features represent the input data [31].

# CHAPTER 4 : DESIGN OF MOSQUE BUILDING DETECTION

## 4.1 Introduction

Object detection is used in computer vision tasks. It is applied in vast areas including robotics. Using image processing for object detection has been a wide area of research topics. This chapter presents mosque building detection (MBD) design and discusses the image analysis module. The components of each module are described along with relevant techniques, algorithms, and considerations in the proposed system.

## 4.2 MBD design consideration

The following list is considered while designing and developing the proposed MBD.

**Types of building:** This design considers building that have significant representation identified from external component and can be represented through an image. Building with internal component is not included.

**Area under consideration** When the image is captured, the whole building is captured. Not only mosque components.

## 4.3 The MBD System Architecture

The proposed system is an image analysis using deep learning. The general overview of the system stared from preprocessed input images with a multidimensional area is represented as Height× Width× Depth tensors about the occurred component of a mosque, which are passed through a pre-trained CNN up until an intermediate layer, ending up with a convolutional feature map. This technique is commonly used in the context of Transfer Learning, especially for training a classifier on a small dataset using the weights of a network trained on a bigger dataset.

Region Proposal Network (RPN) is used also. Using the features that the CNN computed, it is used to find up to a predefined number of regions (bounding boxes), which may contain objects.
Instead of detecting where objects are, the problem is modeled into two parts. For every bounding box, the following questions are asked:

- Do these bounding boxes contain a relevant object?
- How can the anchor be adjusted to better fit the relevant object?

Having a list of possible relevant objects and their locations in the original image, solving the problem becomes straight forward. Using the features extracted by the CNN and the bounding boxes with relevant objects, we apply Region of Interest (RoI) Pooling and extract those features which would correspond to the relevant objects into a new tensor. Finally, comes the R-CNN or detection module, which uses that information to:

- Classify the content in the bounding box.
- Adjust the bounding box coordinates.

The next section covers the details on both the architecture and loss and training for each of the components. The system architecture for the proposed system is illustrated in Figure 4-1.

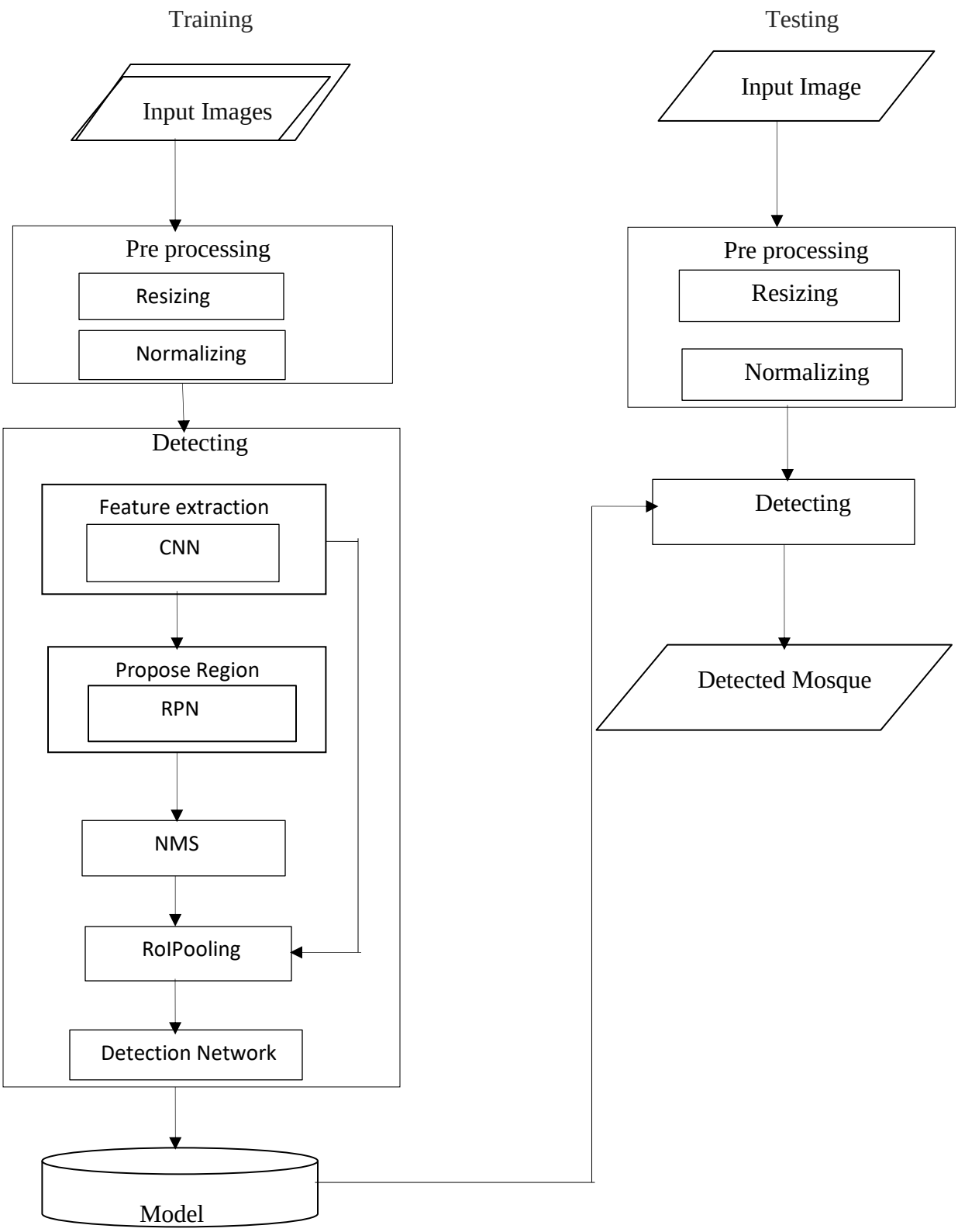



**Figure 4-1 System architecture for mosque detection**

### 4.3.1 Input Image

Input images are collected datasets that contain mosque building as an object in the image. The dataset contains different kinds of mosques based on their external architecture design. Mosque building that contains domes and circled or rectangular shaped minaets with a star, moon, different sized attached ball, or cubic shaped such as an arrow on top of them are considered. For all data, a respective text format representation or annotated image is used as input in the research.

### 4.3.2 Preprocessing

The preprocessing component is responsible to make the input images suitable for the overall image analysis activity that is provided. The main tasks done are resizing and normalizing. Resizing is one of the preprocessing techniques which brings the whole image into the same size. The image is collected from different sources with different sizes. A new image size that best reflects the contents of the image with less processing time of 300 x 300 pixels by algorithm 4-1. is used.

| |
|---|
| Input: Image |
| Output: resized Image |
| Begin: |
| For each image in dataset<br><br>Resized Image=resizing (image, target_size=300,300) |
| Return resized Image |
| END |

**Algorithm 4-1 Image resizing algorithm**

The resized image also needs to have a corresponding resized annotated dataset. Therefore, for each corresponding image, a resized annotated dataset is also produced.

Normalizing is also used to pre-process an input image before further processing. Image pixel values are an integer between the ranges of 0 to 255. Although these pixel values can be presented directly to the model, it can result in slower training time and overflow. Overflow happens when numbers get too big and the machine fails to compute correctly. Using algorithm 4-2 we normalize our data values down to a decimal between 0 and 1 by dividing the pixel values with 255.

| |
|---|
| Input: resized Image |
| Output: Normalized Image |
| Begin: |
| Divide_By=255.0<br><br>For each data in dataset<br><br>Image=float32(image)<br><br>Image=image/Divide_By |

**Algorithm 4-2 Algorithm for normalizing image**

### 4.3.3 Detecting

Detection is the process of identifying mosque buildings and localize their regions in an image. To detect an object, functions like feature extraction using CNN, RPN, NMS, RoIPooling and detection network is performed.

#### 1) Convolutional Neural Network

The convolutional neural network (CNN) also known as the base layer which is composed of layers that are responsible for feature extraction from the preprocessed input image. An input of 300x300 RGB image is passed through a stack of convolutional (Conv) layers, and filters with 3 × 3 (which is the smallest size to skim pattern). The convolution stride is fixed to 1 pixel. The spatial resolution is preserved after convolution and pooling are carried out by three max-pooling layers, which

follow the convolution layers. Max pooling is performed over a 2 × 2-pixel window, with stride 2. Figure 4-2 shows the proposed CNN feature extraction model.

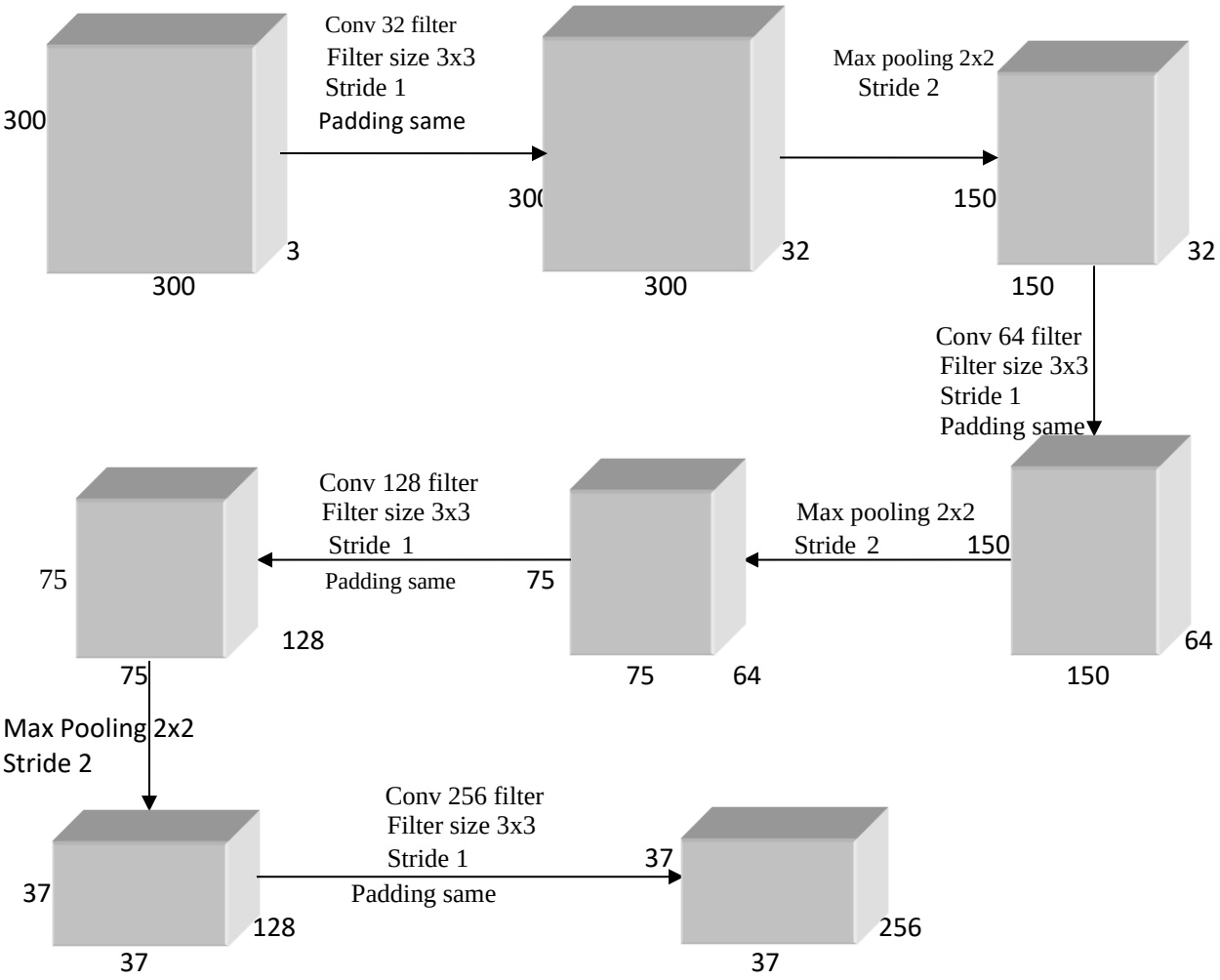



**Figure 4-2 Architecture of the CNN model**

The neurons in convolutional network are arranged in 3 dimensions width, height, and depth. The convolutional layers consist of a set of learnable filters. Every filter is arranged with width and height and extends through the full depth of the input volume.

In the first convolution layer, 300x300x3 image as input data is passed and has a filter with 3x3x3 where the last 3 represents the depth of the filter window is applied. As the filter slides over the width and height of the input volume, a 2-dimensional activation map is produced that gives the responses of that filter at every spatial position. These activation maps are then stacked along the depth dimension and produce the output volume. In the first convolution layer, 32 filters are

applied which result in a 300x300x32 activation map. The max-pooling layer will perform down sampling operation along the spatial dimensions (width, height), resulting in volume 150x150x32.

The second convolution layer accepts the result of the first convolution layer 150x150x32 and applies its 64 filters to the input which results in a 150x150x64 activation map. Max poling follows to perform down sampling and result in 75x75x64 volume.

The third convolution layer accepts the result of the second convolution layer 75x75x64 and applies its 128 filters to the input which results in a 75x75x128 activation map. Max poling follows to perform down sampling and result in 37x37x128 volume.

The final convolution layer accepts the result of the third convolution layer 37x37x128 and applies its 256 filters to the input which results in a 37x37x256 activation map.

The CNN layer is selected based on their effectiveness and efficiency in classifying building trained on two classes (mosque building and non-mosque) for later use their weight as a knowledge while training a new model. This concept is known as transfer learning and it is mostly used when the new training model has a small number of data to be processed.

**2) Region Proposal Network**

Region Proposal Network (RPN) is a small convolutional neural network used to propose a predefined number of bounding boxes or anchor box as a region proposal. RPN network learns boxes to identify classifiers and a regressor as a result. The Classifier determines the probability of a proposal having the target object and regression regresses the coordinates of the proposals.

To generate these “proposals” for the region where the object lies, a small network slides over a convolutional feature map that is the output by the last convolutional layer.

Anchor box: is a bounding box that is defined in the original image using anchor ratio and anchor point in each anchor point. Anchor point is a center point that is found in each sliding window that is defined by a convolution layer.

The future map width multiplied by feature map height, (W*H), produces several anchor points in feature map point. Feature map that is output by the last convolution layer is 37x37 pixel and produce 1369 anchor point in each feature map using the algorithm 4-3.

| |
|---|
| Input: feature map width, feature map height |
| Output: Anchor point as function Output |
| Begin: |
| Calculate number of anchor point in feature map<br><br>Anchor point = W*H //W and H is feature map width and height and Anchor point is possible anchor point in the feature map |
| Return function output |
| END |

**Algorithm 4-3 Algorithm for anchor point calculation**

A mosque building is a wide and big object in the image, 5:1, 1:1, and 4:4 anchor ration and based on the CPU processor we use, size of 32, 64, 128, 190 anchor scale are selected to produce different sized anchor boxes. The point in W × H Feature Map is found in the corresponding position in the original image and for each projected position, k Priori bounding boxes of different sizes and ratios are set and produce 12 bounding boxes for each anchor point and for all anchor points 16428 different sized anchor box is defined using the algorithm 4-4.

K= number of scall*number of ratios 4.1

Where K is number of anchor boxes in each feature map point.

| |
|---|
| Input: feature map, anchor sizes, anchor ratio |
| Output: Anchor box as function Output |
| Begin: |
| For every point in feature map<br>Calculate number of anchor point in feature map<br>Anchor box = W*H*k //W and H is feature map width and height, K is number of boxes in each anchor point and Anchor box is the possible number of anchor boxes in the feature map<br>End |
| Return function output |
| END |

**Algorithm 4-4 Algorithm for calculating anchor box**

To form 16428 anchor boxes, at each feature map point, a simple convolutional sliding window is used. A 3x3x256 sliding window is passed through each feature map point. At each sliding window, a center anchor point is used to apply different sized anchor box using anchor box and ratio. Thus, different sized 12 anchor boxes at each feature map are formed by using the algorithm 4-5, and the result of anchor formed in each feature map is sown in Figure 4-3.

| |
|---|
| Input: feature map, anchor sizes, anchor ratio |
| Output: Anchor point as function Output |
| Begin: |
| For every point in feature map<br>For each scall in anchor scale<br>Calculate anchor box size in the feature map<br>Anchor box= (feature map point, scale, scale at ratio of (width/height))<br>End<br>End |
| Return function output |
| END |

**Algorithm 4-5 Algorithm for anchor size calculation**

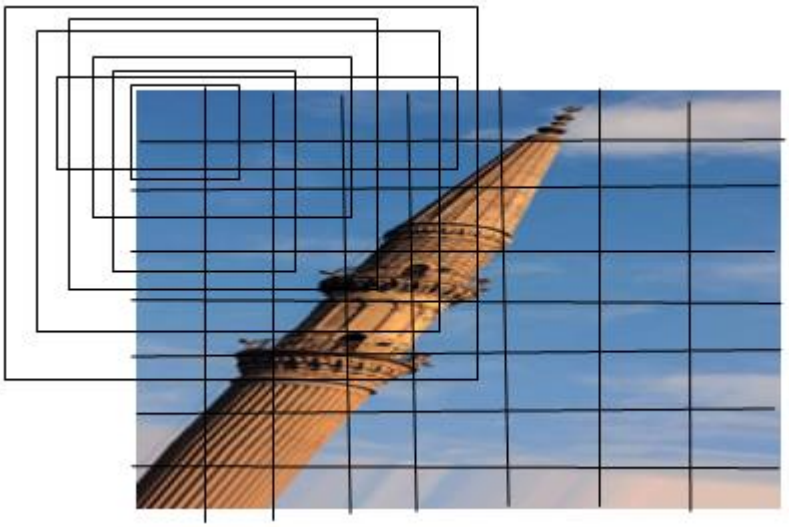


**Figure 4-3 Anchor box in single feature point**

The formed anchor box in the feature map is now used to propose a set of anchor boxes and label for each proposed region as objectness or background. To propose regions, two 1x1 convolution layers are used to replace a fully connected layer for proposing possible bounding boxes with their classifications. The convolution layer for classification, 1x1x2*12 (where 2 is number of class label and 12 is number of anchor box at each anchor point) or 1x1x24 kernel initialized with pre-trained weight, sigmoid activation function for binary output is used to output 37*37*24 label parameters as region proposal classification. Also, for regression, 1x1x4*12 (where 4 is number of box value in each coordinate to form a rectangular shape and 12 is number of anchor box at each anchor point) or 1x1x48 kernel is used initialized by pre-trained weight, the linear activation function is used to output 37*37*24 box parameters as a region proposal regression.

Simultaneously, to train RPN, a valid anchor box from predefined anchor boxes or proposed regions, a valid anchor is selected for later use to calculate the loss between the proposed region and the truth value. To select a valid anchor, unwanted anchors are removed first based on their coverage area in the image. Anchor boxes that cross the boundary of an image (300x300 pixel) are ignored and only anchors inside the boundary are kept for further process by using algorithm 4-6 and the results are shown in Figure 4-4.

| |
|---|
| Input: Image width, Image height, anchor boxes |
| Output: Valid Anchor as function Output |
| Begin: |
| For every anchor in the feature map<br>Calculate valid anchor box<br>anchor boxes= select (anchor boxes, target size <=300,300)<br>End |
| Return function output |
| END |

**Algorithm 4-6 Algorithm for valid anchor box selection**

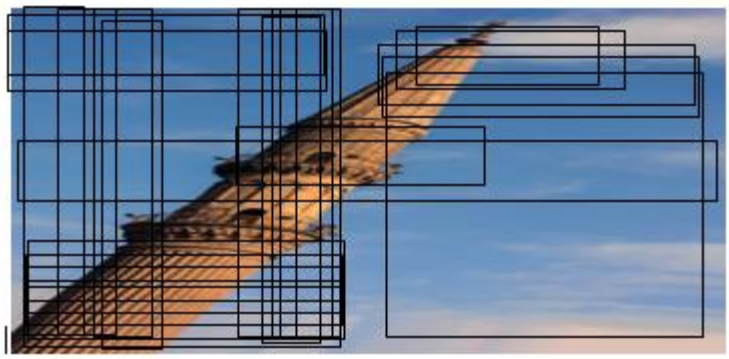
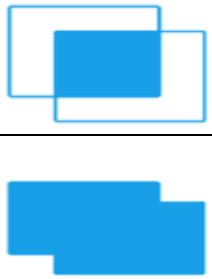


**Figure 4-4 Valid anchor box**

Anchor boxes can be label as a foreground if an anchor that contain objects and background if the anchor does not contain object using intersection over union (IOU). We choose the anchors whose overlap with the ground truth box is greater than RPN POSITIVE OVERLAP = threshold value as foreground label and whose overlap with any ground truth box is lower than RPN NEGATIVE OVERLAP = threshold value as a negative label or background label.

$$IOU = \frac{\text{Area of Overlap}}{\text{Area of Union}} \quad \text{..... [4.2]}$$



The defined anchor box set as objectness (foreground) if the mosque is found on it and for boxes that do not contain a mosque is set to as background. Anchor box with IOU confidence score value greater than 0.6 is considered as mosque and boxes with less than 0.4 confidence values are set to background. Lowering or increasing the confidence score will cause labeling false region and loss of region, therefore, the average value is chosen and anchors with a confidence score between 0.4 and 0.6 are not considered using the algorithm 4-7.

| |
|---|
| Input: anchor, Ground truth, threshold |
| Output: foreground, background as function Output |
| Begin: |
| For every box in proposed region<br>Calculate IOU from ground truth value<br>IOU = anchor n GT // where anchor is proposed anchor box in image, GT is ground<br>anchor U GT truth n is intersection u is union<br>End<br>For every anchor in the image<br>label anchor as foreground and background<br>foreground =anchor[max_overlap>=0.6] //mosque<br>background =anchor[min_overlap<0.4] //not mosque or background<br>End |
| Return function output |
| END |

**Algorithm 4-7 Algorithm for labeling anchor box**

The labeled anchor has a greater number of negative overlap than positive. So, we randomly down sample the number of the foreground or positive region and background or negative region with batch size 256 for each as a batch size to maintain a balanced ratio between foreground and background anchors during training.

The RPN uses all the anchors selected for the mini-batch to calculate the classification loss using binary cross-entropy. Only those minibatch anchors marked as foreground are used to calculate

the regression loss. For calculating the targets for the regression, I used the foreground anchor and the closest ground truth object and calculate the correct difference value needed to transform the anchor into the object.

### 3) Non-Max-Suppression

The proposed regions that are proposed by region proposal layer called region of interest (ROI) are used for the next task, detecting an object. To compute mosque detection from the proposed regions, unwanted ROI needs to be first removed by using non-max-suppression (NMS). NMS is the removal of regions that are overlapped based on the IOU value with the ground truth.

Using equation 4.2, to select regions that intersect the truth value classification cause between sensitivity and specificity is considered. If we set the confidence score very low, our sensitivity will be high and specificity will be, low. Besides, if we use a high confidence score, sensitivity will be decrease and specificity will increase. Therefore, we carefully select specific 0.5 confidence score to achieve average region. Similarly, for further process, a limited number of regions (300 boxes) called ROI samples are subsampled for faster processing based on the CPU capacity we have as a batch size for later use. The algorithm for calculating NMS is shown in algorithm 4-8.

| |
|---|
| Input: RPN, Ground truth, Threshold, Maximum-box |
| Output: ROI function Output |
| Begin: |
| For every box in ROI<br>Calculate non max suppression to select ROI input with the highest thrush hold value to the ground truth<br>RPN_Threshold = RPN.apply_threshold (Threshold)<br>Select limited number of ROI from RPN_Threshold<br>ROI= RPN_Threshold.apply_Selection (Maximum-box)<br>End |
| Return function output |
| END |

**Algorithm 4-8 Algorithm for ROI Sampling selection**

### 4) RoIPooling

After RPN, we get proposed regions with different sizes. Different sized regions mean different sized CNN feature maps. It's not easy to make an efficient structure to work on features with different sizes. Region of Interest Pooling can simplify the problem by reducing the feature maps into the same size. Unlike Max-Pooling which has a fix size, ROI Pooling splits the input feature map into a fixed number (let's say k) of roughly equal regions, and then apply Max-Pooling on every region. Therefore, the output of ROI Pooling is always k regardless the size of input.

The same as feature extracted from image dataset, based on proposed regions we compute the corresponding ROI by reusing the existing convolutional feature map with pre-trained

initialization and to feed them to the fully convolutional layer for the detection, they must be resized into the same value by pooling ROI (18x18x256).

### 5) Detection Network

Fast RCNN or detection layer use the pooled ROI then fed into two fully convolutional layers with a depth of 500 using RELU activation function. To detect the final object, LINEAR activation function for bounding box coordinate regression and SoftMax activation function since we only have two classes (mosque, background) is used to classify bounding box. The detection layer contains two losses: classification loss which represents category loss, and regression loss which represent bounding boxes location loss. classification loss is computed with a cross-entropy of two categories (mosque or no mosque) and regression loss is similar to RPN, using linear activation but only 4 values are participant in the gradient calculation.

The Faster RCNN is trained with Stochastic Gradient Descent with momentum, setting the momentum value to 0.9 and takes 49 hours with HP Intel core i5 -6200u CPU. The total loss information calculated propagated backward starting from the output layer to a fully convolutional layer with fixed convolutional layer neurons in the hidden layer. The neurons of the hidden layer only receive a fraction from the total loss, based on the contribution each neuron has contributed to the original output. This process is repeated, layer by layer, until all the neurons in the network have received a loss signal that describes their relative contribution to the total loss. This is done by calculating the partial derivate of the cost function with relative to the weight of each neuron. The detection loss is used for initialization for RPN layer by fixing a unique value for it and train the RPN again. The regions that are proposed by the second RPN is used as an input for the detection layer with initialized weight from RPN and detect an object. This process is repeated, layer by layer, until all the neurons in the network have received a loss signal that describes their relative contribution to the total loss. Algorithm 4-9 shows a backward propagation algorithm used while training the CNN model.

| |
|---|
| Input: network assigned input and output |
| Output: weight update parameter |
| Begin: |
| Propagate the errors backward through the network<br><br>For all fully connected layer in RPN layers<br><br>Calculate the nodes signal error from Detection layer<br><br>Keep shared layer fixed // CNN is shared layer<br><br>Update unique nodes weight in the full convolution network<br><br>End<br><br>For all fully connected layer in Detection layers<br><br>Calculate the nodes signal error from RPN layer<br><br>Keep shared layer fixed // CNN is shared layer<br><br>Update unique nodes weight in the full convolution network<br><br>End |
| Return weight update parameter |
| END |

**Algorithm 4-9 Algorithm for backward propagation**

## 4.4 Summary

Object detection is a process of identifying the presence of an object in the image and predicting the location of an object in the image. We processed the detection system with image processing using deep learning. The proposed system can achieve effective detection using different mediums of input.

For each input, appropriate processing methods are used. The images are preprocessed using resizing and normalizing method and detected based on a trained Faster RCNN model. The Faster RCNN model constitutes convolutional neural network to extract features of an input image, region proposal network to propose regions and detection network from Fast RCNN to detect an object based on the proposed region. Based on the result from the detection model the system reaches a final detected output by predicting the existence of a mosque with their location area from an image.

# CHAPTER 5 : IMPLEMENTATION AND EVALUATION

## 5.1 Introduction

This chapter discusses the experiments carried out to test the effectiveness of the proposed system. The data set used, results achieved in the classification and detection process and the system performance will be discussed.

## 5.2 Dataset preparation

To evaluate the proposed system, datasets were gathered that are used for classification and detection models. The gathered data is passed in different processes before we feed them into our models.

A) Data collection

Images are gathered from different websites and by taking pictures taken going to places in Addis Ababa and Bishoftu. Different mosques, and non-mosque images specifically church buildings with similar features like a mosque are collected. The collected data is evaluated by four Shah and one building architect designer to validate. The correct datasets were collected before further processing is done.

B) Data Preprocessing

The collected images are first cleaned in a form that is not related to our labeled images and since deep learning performance is based on the number of data, the algorithm is fed. Dataset is enlarged using a technique called image augmentation. The datasets contain images that have high- and medium-quality images. Image augmentation like horizontal and vertical flip and rotate augmentation is used. Likewise, according to the model, different activities that are suitable input for our models is performed.

Classification model: is used for the sake of transfer learning. A trained CNN is used to detect an object in the image. CNN classifies the datasets that are used according to their label.

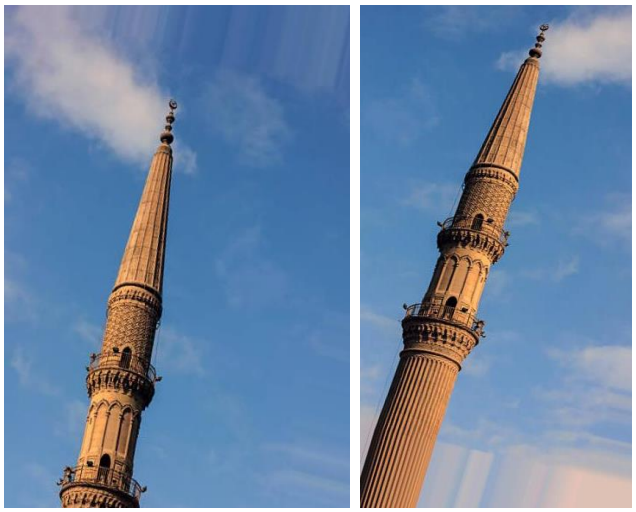


Figure A

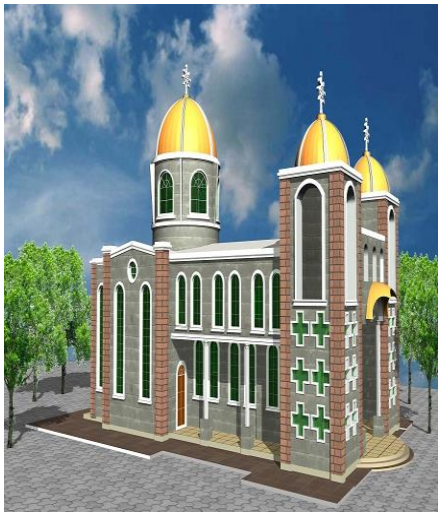


Figure B

**Figure 5-1 Figure A contain mosque image and figure B contain non mosque image sample**

Detection model: since object detection models take labeled images or a ground truth value for localizing an LabelImg image, one type of labeling or annotation tool based on the scope of the project is used. Mostly for object detection and localization, tools that could draw rectangle boxes for localizing are preferred. Therefore, we choose a rectangular bounding box annotation technique to define the location or out targeted object in the image. Rectangular boxes are determined by upper left x and y coordinate and lower left x and y-axis coordinate. To label data, graphical image annotator tool licensed by MIT to label our image format output with .xml format is used. Likewise, we evaluate the prepared '.xml' file which in graphical form has a bounding box that locates the location of the mosque is evaluated before further processing. To represent the data for this project, all the '.xml' format is converted to its corresponding 'txt' formatted dataset using python code. Figure 5.2 shows the '.xml' format representation of our dataset.

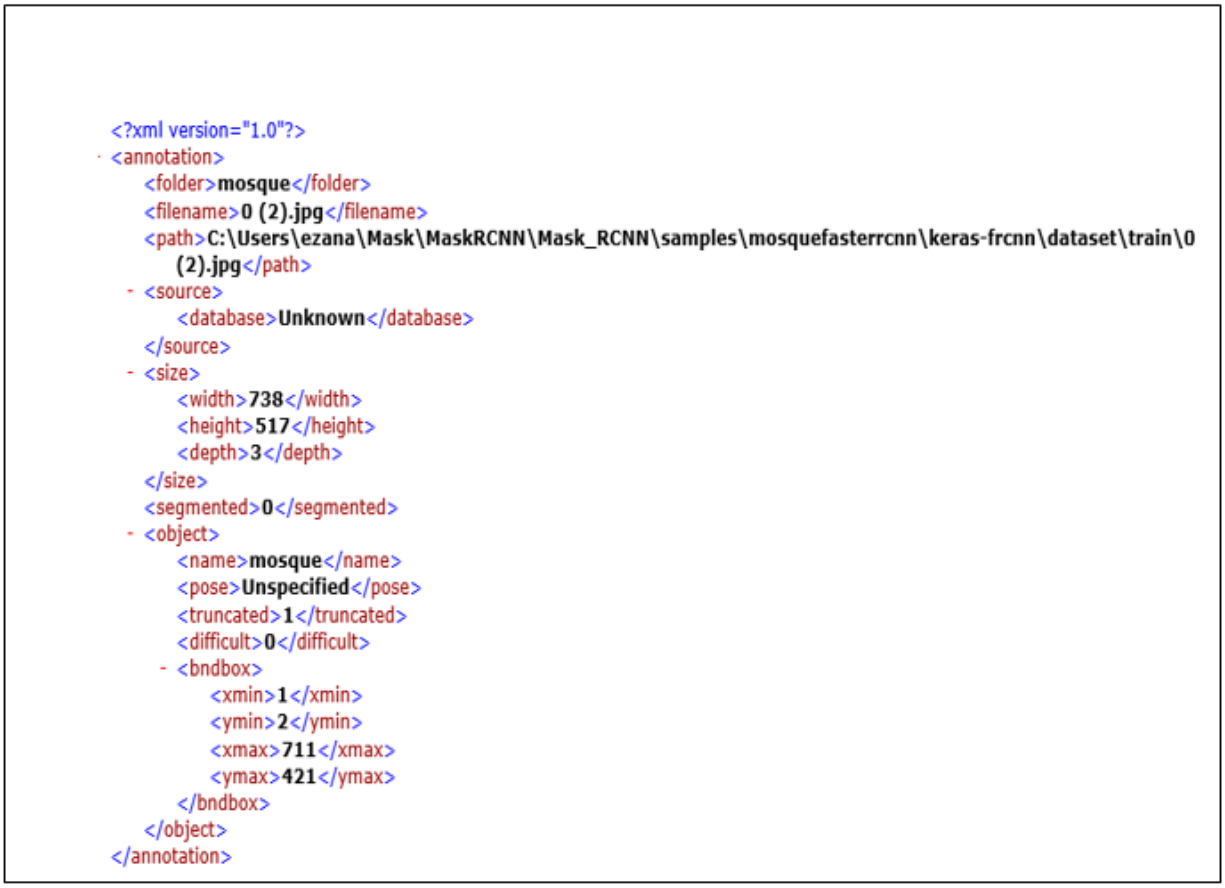



**Figure 5-2 Annotated image sample**

The research model requires a large amount of labeled data. But getting enough data is a major problem in our cases which leads to the use of other techniques to expand the dataset. Data augmentation provides a means for increasing the quantity of training data available for machine learning and is particularly relevant when training deep learning systems from scratch [41]. To combat the high expense of collecting thousands of training images, horizontal flip, vertical flip and rotational augmenting is used to expose our classifier and detection to a wider variety of transformed images to make the detection more robust.

**Table 5:1 Collected data to train classification model**

| No | Image Type | Total Collected data | Collected and augmented data |
|---|---|---|---|
| 1 | Mosque | 800 | 1848 |
| 2 | Non-mosque | 800 | 1848 |

**Table 5:2 Collected data to train detection model**

| No | Image Type | Total Collected data | Collected and augmented data |
|---|---|---|---|
| 1 | Mosque | 800 | 1848 |

Using a standard data classification method, augmentation on the dataset is applied. To train a convolutional neural network for later use as transfer learning, dataset from these images, 80% is used for training and 20% is used for testing [43]. Detection model is trained with 90% of the dataset and 10% for testing the model is used.

## 5.3 Development Environment

The image analysis is done on Anaconda Keras. To augment image data, Augmenter package is used. Augmenter is a python package made available under the terms of the MIT license. The package emphasis providing operations that are typically used in the generation of image data for machine learning problems.

## 5.4 System Evaluation

Evaluation is done on the model because the accuracy of the system is affected by the performance of the detection model. Performance evaluation of the detection model is done using testing datasets for evaluating using deep learning evaluation techniques, mean average precession (mAP) matrix.

### A) Classification Evaluation

Convolutional Neural Network is used to classify an object for later use as a transfer learning. CNN model is trained with two class labels to classify binary classification as image contains either a mosque building or no mosque building in the image. Figure 5-3 is used for classifying image.

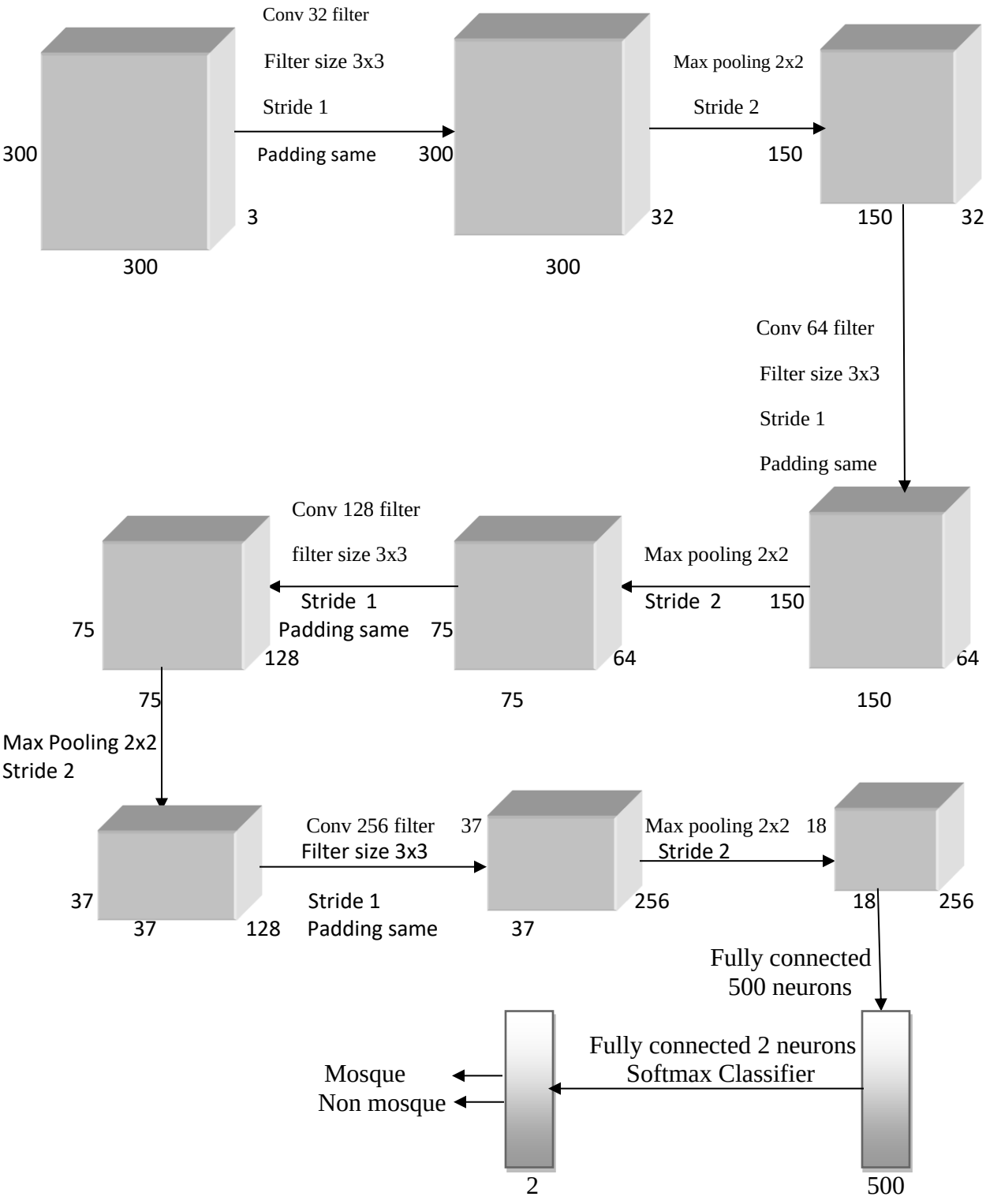



**Figure 5-3 CNN architecture**

The classification model is trained by taking a preprocessed input image size of 300x300 and image pixel values are an integer between the ranges of 0 to 255, it can result in slower training time and overflow. Using Algorithm 4.5 the data is normalized to values down to a decimal between 0 and 1 and fed them to the convolutional neural network.

In the first convolution layer, 300x300x3 image as input data is passed and has a filter with 3x3x3 where the last 3 represents the depth of the filter window is applied. As we slide the filter over the width and height of the input volume, we will produce a 2-dimensional activation map that gives the responses of that filter at every spatial position. These activation maps are stacked along the depth dimension and produce the output volume. In the first convolution layer, 32 filters are applied which result in a 300x300x32 activation map. The max polling layer will perform down sampling operation along the spatial dimensions (width, height), resulting in volume 150x150x32.

The second convolution layer accepts the result of the first convolution layer 150x150x32 and applies its 64 filters to the input which results in a 150x150x64 activation map. Max poling follows to perform down sampling and result in 75x75x64 volume. Its function is to progressively reduce the spatial size of the representation to control overfit, to reduce the number of parameters and computation in the network.

The third convolution layer accepts the result of the second convolution layer 75x75x64 and applies its 128 filters to the input which results in a 75x75x128 activation map. Max poling follows to perform down sampling and result in 37x37x128 volume. Its function is to progressively reduce the spatial size of the representation to control overfit, to reduce the number of parameters and computation in the network.

The fourth convolution layer accepts the result of the third convolution layer 37x37x128 and applies its 256 filters to the input which results in a 37x37x256 activation map. Max poling follows to perform down sampling and result in 18x18x256 volume. The final fully connected layer computes the class scores, resulting in a volume of size 1x1x2, where the 2 numbers correspond to a class score, among the categories of our dataset.

During training, it is observed that adding more layers does not improve the performance on our dataset. It leads to overfitting, increased memory consumption, and computation time. The

removal of one layer from the model results in poor performance because the model will not generalize enough with a smaller number of layers in the model.

The performance of the model will be poor either by overfitting or underfitting the provided data. The training of the model is plotted to see the possibility of overfitting and underfitting in the model. Overfitting happens when a model learns the detail and noise in the training data to the extent that it negatively impacts the performance of the model on new data. Figure 5-4 will show the result of trained model

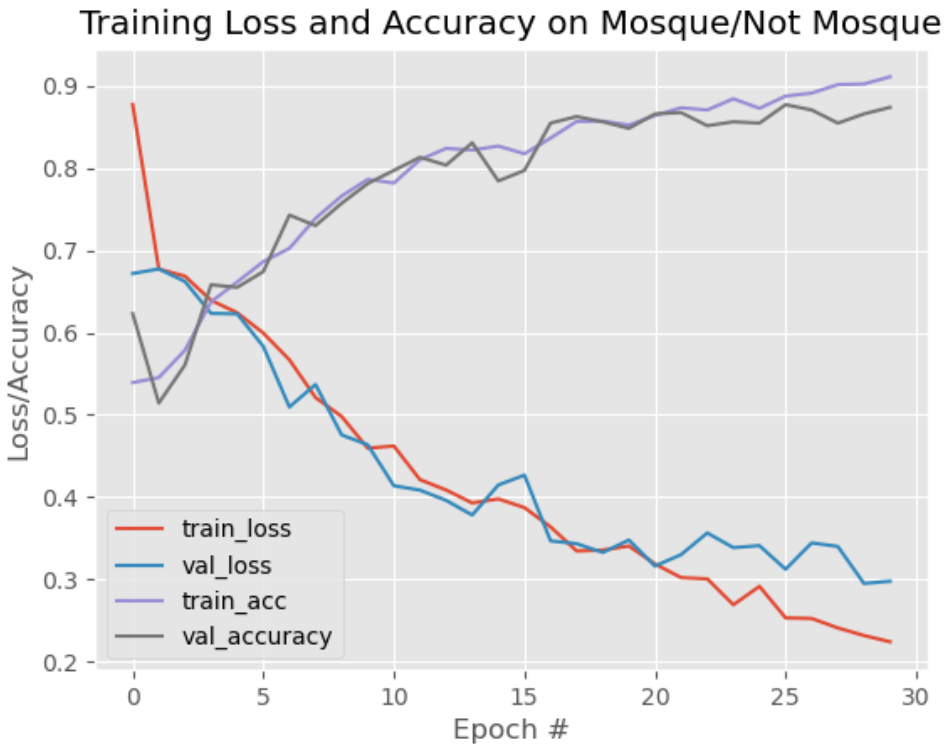



**Figure 5-4 Plot of training and testing accuracy and validation loss**

The model achieves 93% accuracy in 30 epochs. Continuing the training above 30 epochs, the model tries to learn the data. The noise and the performance are not changing but overfitting happens.

### B) Detection evaluation

The performance of the detection model will be measured the same as the classification is evaluated based either by overfitting or underfitting the data. The training of the model is plotted to see the possibility of overfitting and underfitting in the model. Our detection model is not overfitting as shown in the following figures.

Detection of the object introduced four loss, two from RPN, and the other two from Fast RCNN model. As mentioned before, RPN model has two outputs. One is for classifying whether it's an object and the other one is for bounding boxes' coordinates regression. From figure 5-5 we can see that the objectness score proposed by RPN is decreased through learning.

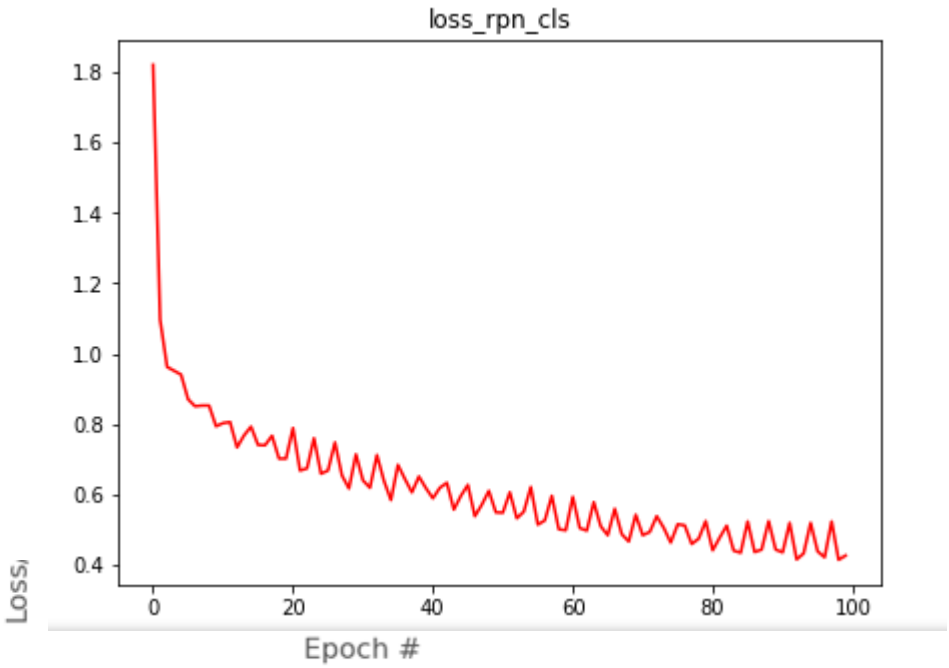



**Figure 5-5 Plot of epochs VS loss for RPN classifier output**

From figure 5-6, RPN regression for bounding box coordinate regression where the RPN propose region shows fast learning through the training time.

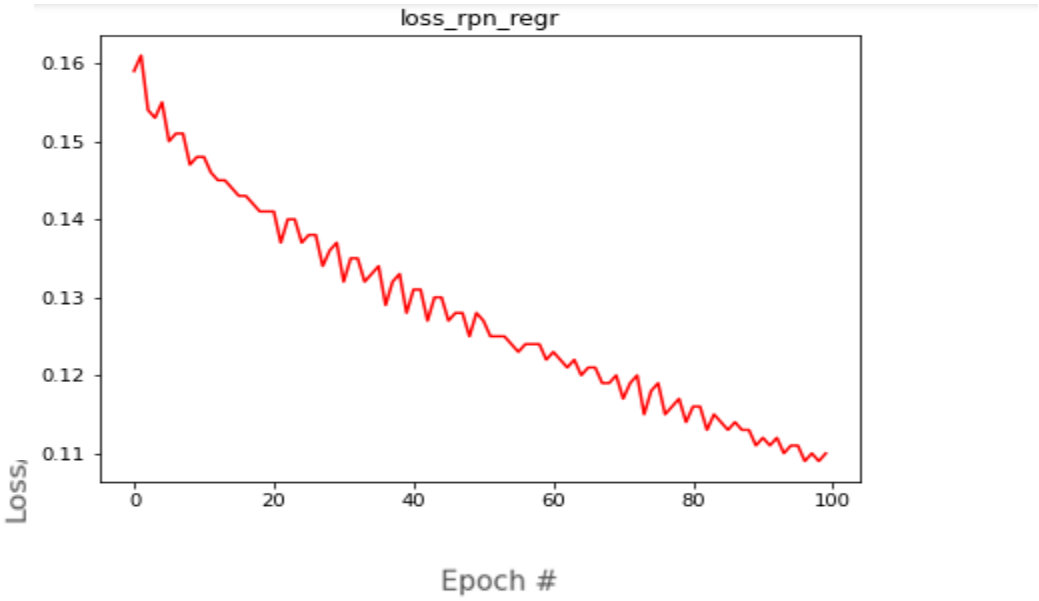



**Figure 5-6 Plot of epochs VS loss for RPN regression output**

The similar learning process is shown in Classifier model. Compared with the two plots for boxes' regression, figure 5-7 show classification model classify with a similar tendency.

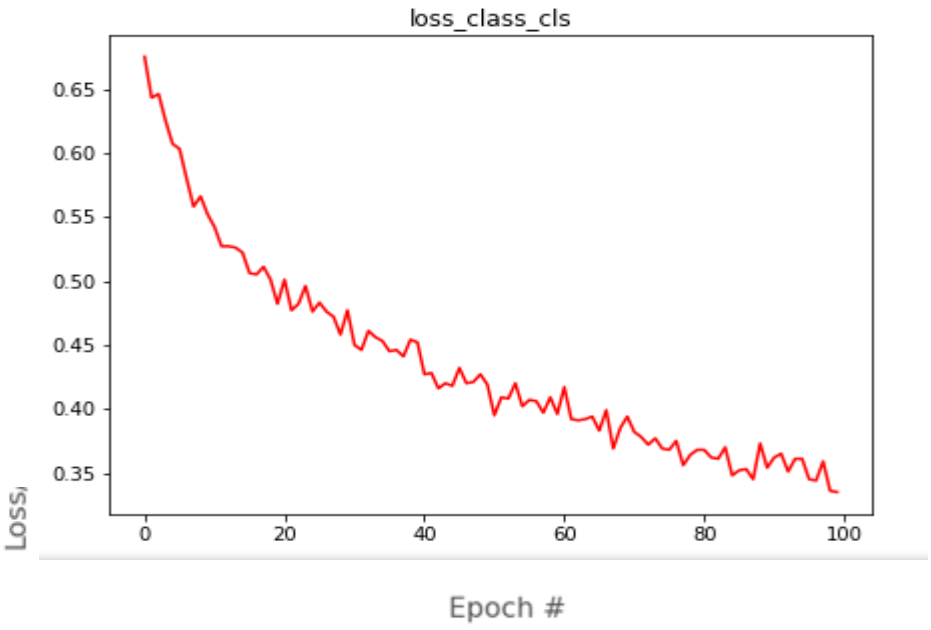



**Figure 5-7 Plot of epochs VS loss for Detection classification output**

The similar learning process is shown in Classifier model. Compared with the two plots for boxes' regression, figure 5-8 show classification model regress or localize the object with a similar tendency.


**Figure 5-8 Plot of epochs VS loss for Detection regression output**

In Figure 5.9, we can see that total loss is also decreased through learning. The total loss is the sum of RPN classification loss, RPN regression loss, detection classification loss and detection regression loss and it has a decreasing tendency.

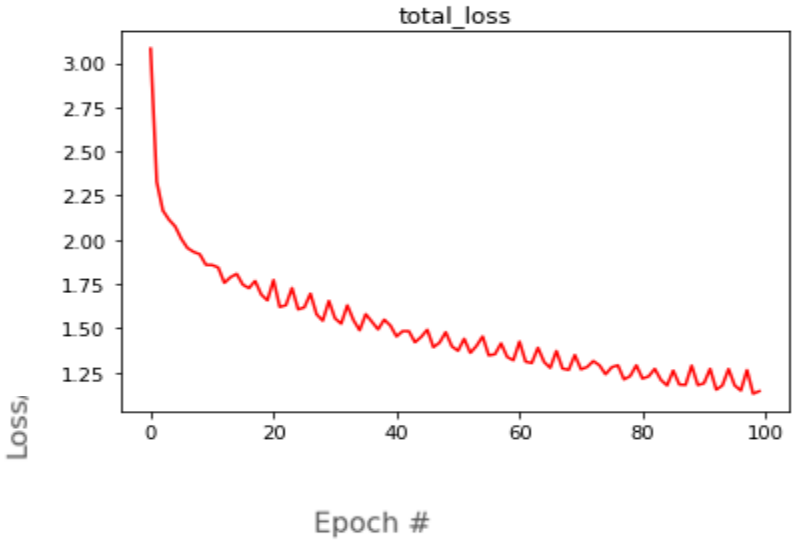



**Figure 5-9 Plot of epochs VS total loss from two models**

In Figure 5.10, the training accuracy of the model to detect the mosque from the provided data is increased from 0.57 to 0.88.

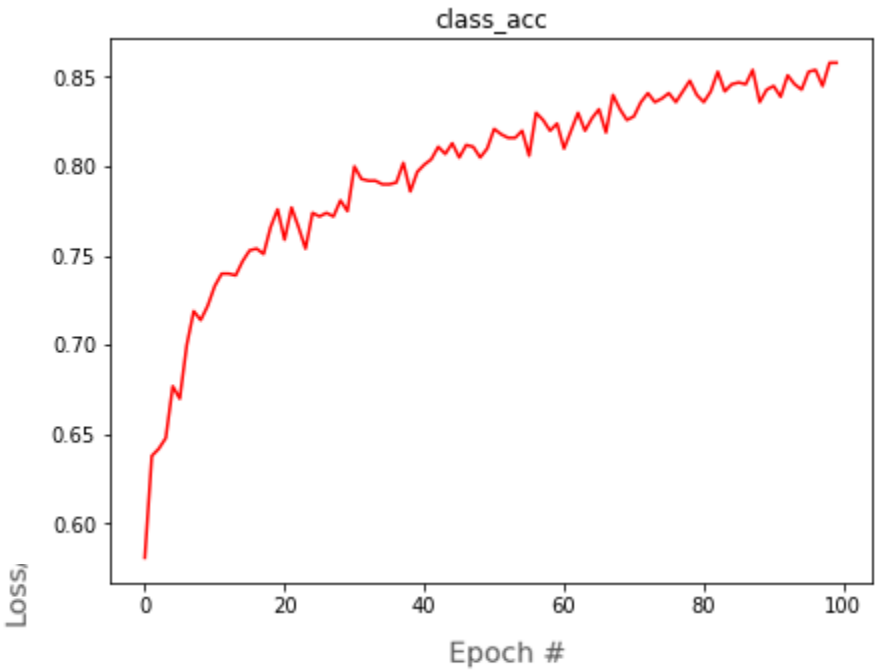



**Figure 5-10 Plot of epochs VS proposed model**

## 5.5 Analysis of Results

Object detection systems make predictions in terms of a bounding box and a class label. The mean Average Precision (mAP) is a popular measuring metric for evaluating the accuracy of object detectors by calculating an estimated average precession (AP) [44].

The assessment of object detection methods is mostly based on the precision P and recall R concepts, respectively defined as

$$P = \frac{\text{TP}}{\text{TP} + \text{FP}} = \frac{\text{TP}}{\text{all detections}} \quad 5.1$$

$$R = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{\text{TP}}{\text{all ground truths}} \quad 5.2$$

Where TP is true positive, FP is false positive and FN is false negative.

Precision is the ability of a model to identify only relevant objects (percentage of correct positive predictions). Recall is the ability of a model to find all relevant cases (all ground-truth bounding boxes). It is the percentage of correct positive predictions among all given ground truths.

The precision_ recall is an inconsistent value, posing challenges to an accurate measurement of its AUC. This is circumvented by processing the precision _ recall curve to remove the inconsistency behavior before AUC estimation. All-point interpolation is used to calculate the average of precession by adding precession in all points correspondingly. The mean Average Precision (mAP) score is calculated by taking the mean AP over all classes and/or overall IoU thresholds, depending on different detection challenges that exist. During evaluating our model, we can achieve a mAP of 0.70. The system output result during testing is shown in Figure 5-11 below.

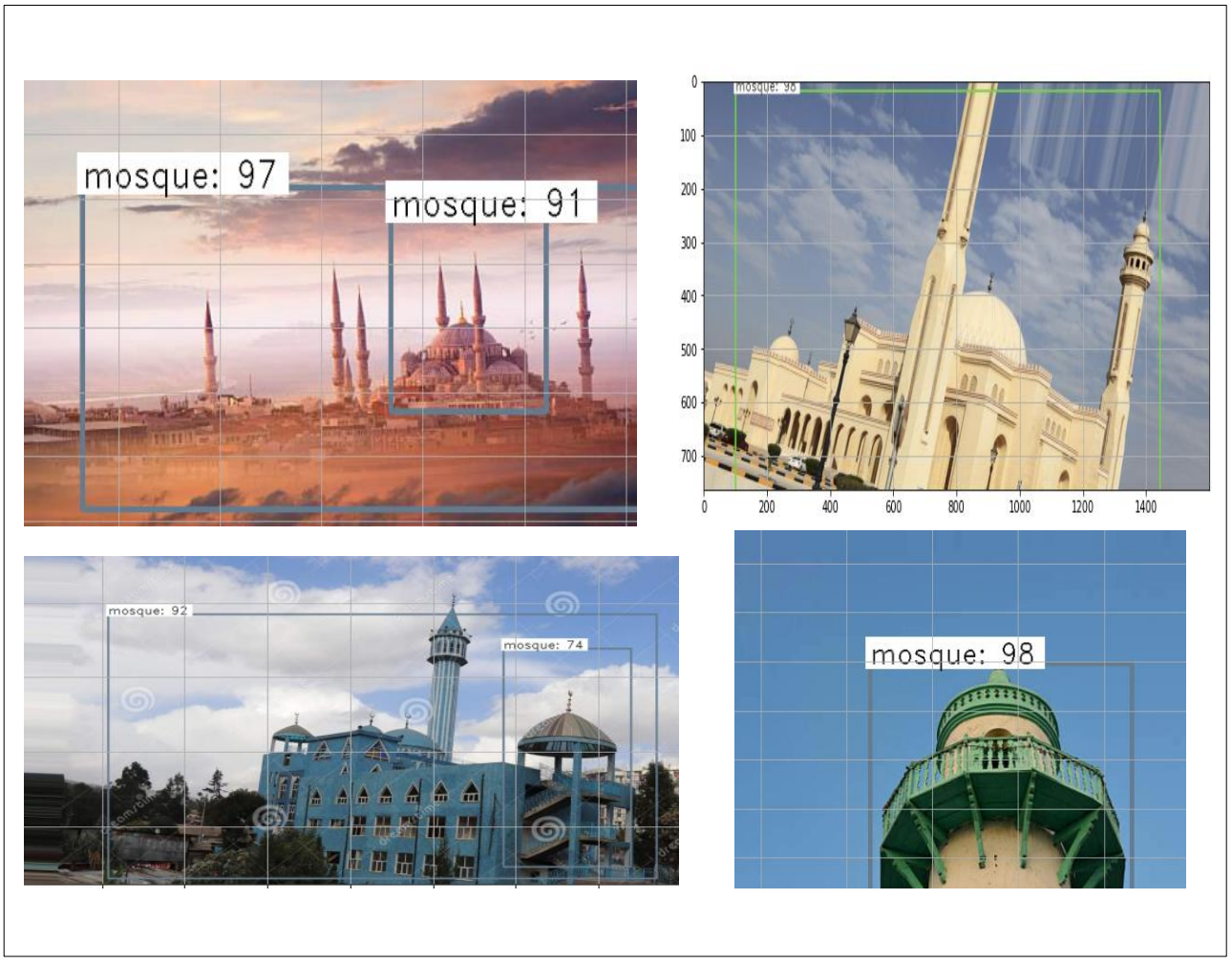



**Figure 5-11 Mosque image detection result**

## 5.6 Comparison with Region-based Fully Convolutional Network

Comparison is done with the same dataset and parameter with architectural difference in the model. The RFCN model is trained with Stochastic Gradient Descent with momentum, setting the momentum value to 0.9 and takes four days to train with HP Intel core i5 -6200u CPU and achieve mAP of 0.53.

The difference in the performance of the two models came with their architecture. Region-based Fully Convolutional Network (RFCN) use region proposal network to propose region and full convolution layer after RoIPooling are removed. The RFCN poor performance came because the model over fit the training data.

# CHAPTER 6 : CONCLUSION AND FUTURE WORK

## 6.1 Conclusion

Numerous researches are done in the field of object classification, detection, and recognition for processed images and real-time detection. In recent times, numerous research works are done by incorporating images of buildings, human, and face detection. Some of the researches use manual feature extraction to detect instances and some use real imagery datasets that have different processing strategies. Images that are taken by satellites are used for object detection but the distance where the picture is taken makes the image look indistinguishable. Satellite images are easy for general object detection. One of the goals of object detection is the localization of an object in addition to classification. Using existing model to detect mosque building results in a bad localization result since they are different in size.

Human beings can easily recognize and detect objects but, for machines such as robots to achieve this human ability, think like a human and differentiate objects, they should be trained. Amongst objects, this research chose a mosque building which is found in the whole world and propose a model called mosque building detection as one knowledge area.

A detection approach is proposed using image processing in deep learning models. In this research, mosque detection is done by processing images. Images are first preprocessed by resizing and normalize them for suitable processing. The preprocessed image feature is extracted using a proposed shared convolutional layer for proposing regions. Using the proposed region and the shared layer the research can detect mosque building in the image. A human-like capability of detecting an object is achieved by training Faster_RCNN model. The final result of detection is drawing bounding or rectangular boxes in an image where the specific object is. In this research, when a picture is presented, a mosque building is distinguishably located and labeled in the box as a mosque image. The detector model is evaluated by mAP and achieve 0.70 with 88% training accuracy.

## 6.2 Contribution

During the research time, the works that we contribute are:

- ✓ Mosque and non-mosque image classification dataset are collected and prepared.
- ✓ Image classification model is developed and evaluated.
- ✓ Object detection model is developed and evaluated.

## 6.3 Future work

This research work explores different areas that can be further improved for better computer visioning. This thesis opens opportunities for further research on

- ✓ Increasing the number of datasets to improve the performance and functionality of the system.
- ✓ Design instantly segment mosque building for better useability of computer visions.
- ✓ Increasing the dataset to include other building type to improve domain specific detection.

## References


[1] Frederick.S, Ricketts.T, Merritt and a. Jonathan, BUILDING DESIGN, New York, San Francisco, Washington, D.C. Auckland, Bogota´, Caracas, Lisbon, London, Madrid, Mexico City, Milan, Montreal, New Delhi, San Juan, Singapore, Sydney, Tokyo and Toronto: Library of Congress, 2000.

[2] A. David, Metric Handbook Planning and Design Data, British Library, 2019.

[3] "MathWorks," [Online]. Available: https://in.mathworks.com/solutions/image-video-processing/object-recognition.html. [Accessed 02 04 2020].

[4] E. Wolfgang, "Introduction to Artificial Intelligence," Germany, Springer Nature, 2018..

[5] L. Daniel, "Development of Automatic Maize Quality Assessment System Using Image Processing Techniques," Department of computer Science, Addis Ababa University, Addis Ababa, Ethiopia, Addis Ababa, 2015.

[6] T. Carme, Computer Vision Theory and Industrial Applications, Spain, Berlin, Heidelberg, New York,London, Paris, Tokyo,Hong Kong, Barcelona and Budapest: Springer-Verlag Berlin Heidelberg, 1992.

[7] H. F. a. I. Mohammed, "Night Time Vehicle Detection," *ResearchGate,* vol. Volume 21, no. Issue 2, p. 143–165, 2012.

[8] W. Sakla and G. K. a. N. Mundhenk, "Deep Multi-Modal Vehicle Detection in Aerial ISR Imagery," in *2017 IEEE Winter Conference on Applications of Computer Vision*, Santa Rosa, CA, USA, 2017.

[9] A. H. Özcan, C. Ünsalan and R. a. Peter, "Building detection using local features and DSM data," in *2013 6th International Conference on Recent Advances in Space Technologies (RAST)*, Istanbul and Turkey, 2013.

[10] Sırmac¸ and U. B. a. Cem, "Urban Area and Building Detection Using SIFT Keypoints and Graph Theory," *IEEE Transactions on Geoscience and Remote Sensing,* vol. Volume: 47, no. Issue: 4, pp. 1156 - 1167, April 2009.

[11] Rashedi and A. F. a. Esmat, "Object detection in images using artificial neural network and improved binary gravitational search algorithm," in *2015 4th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS)*, Zahedan and Iran, 2015.

[12] Xiao and J. a. H. Wei, "A SHAPE-BASED OBJECT CLASS DETECTION MODEL USING LOCAL SCALE-INVARIANT FRAGMENT FEATURE," in *2014 IEEE International Conference on Image Processing (ICIP)*, Paris and France, 2014.

[13] Z. Changzheng and T. X. a. Dandan, "Face Detection Using Improved Faster RCNN," China, 2018.

[14] E. Alexander, M. Vasileios, V. Fabio and M. Z. a. Kamer, "Firearm Detection and Segmentation Using an Ensemble of Semantic Neural Networks," in *2019 European Intelligence and Security Informatics Conference (EISIC)*, 2020.

[15] M.-T. P. a. Sebastien, "Buried object detection from B-scan ground penetrating radar data using Faster-RCNN," in *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, Valencia, Spain, 2018.

[16] R. Xiaodong and Z. S. a. Yi, "Parallel RCNN: A Deep Learning Method for People Detection Using RGB-D Images," in *2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI 2017)*, China, 2017.

[17] S. Ruhan, O. William, W. Raymond, S. Mark, D. Capoferri, B. Kenneth, G. Alexander, R. Robert, H. Adam and C. J. a. Vipin, "Intervertebral disc detection in X-ray images using faster R-CNN," 2017.

[18] F. Zhou, W. Fan and Q. S. a. M. Tao, "Ship Detection Based on Deep Convolutional Neural Networks for Polsar Images," in *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, Valencia and Spain, 2018.

[19] A. Ullah, H. Xie, Muhammad and O. F. a. Z. Sun, "Pedestrian Detection in Infrared Images Using Fast RCNN," in *2018 Eighth International Conference on Image Processing Theory, Tools and Applications (IPTA)*, Xi'an and China, 2018.

[20] N. Muhammad, Mosque Architecture: Formulation of Design Criteria and Standards in the Context of Bangladesh, Bangladesh, 2000.

[21] R. Stegers, Sacred Buildings, German: Birkhäuser, 2008.

[22] Woods and R. G. a. Richard, Digital Image Processing, United States of America: Addison-Wesley Longman Publishing, 2008.

[23] D. Hailemichael, Development Of Automatic Maize Quality Assessment System Using Image Processing Techniques, Addis Ababa: Master's Thesis. Department of computer Science, Addis Ababa University, Addis Ababa, Ethiopia, 2015.

[24] G. Tigistu, "Automatic Flower Disease Identification using image processing," in *AFRICON 2015*, Addis Ababa, Ethiopia, Sept 2015.

[25] D. K. a. Yadwinder, "Various Image Segmentation Techniques: A Review," *International Journal of Computer Science and Mobile Computing*, vol. Vol.3, pp. 809-814, 2014.

[26] S. Krishna and K. S. a. Akansha, "A Study Of Image Segmentation Algorithms For Different Types Of Images," in *A Study Of Image Segmentation Algorithms*, Ghaziabad and India, 2010.

[27] M. Umaa and S. a. Mala.C, "An expermental study and analysis of different image segmentation techniques," in *Elsevier Ltd*, India, 2013.

[28] D. Cao and Z. C. a. L. Gao, "An improved object detection algorithm based on multi-scaled and deformable convolutional neural networks," pp. 2-22, 2020.

[29] H. Qian and J. X. a. J. Zhou, "Object Detection Using Deep Convolutional Neural," in *2018 Chinese Automation Congress (CAC)*, Xi'an and China, 2018.

[30] A. Krizhevsky and I. S. a. G. E. Hinton, "ImageNet Classification with Deep Convolutional," in *Communications of the ACM*, May 2017.

[31] R. Santiago, T. d. Menezes, R. Marrocos and M. a. H. Maia, "Object Recognition Using Convolutional Neural Networks," in *TENCON 2018,2018 IEEE Region*, Jeju, Korea (South), 2018.

[32] T. Liu, S. Fang, Y. Zhao and P. W. a. J. Zhang, "Implementation of Training Convolutional Neural Networks," Beijing, China, 2015.

[33] N. Ivars, "Deep Convolutional Neural Networks: Structure,Feature Extraction and Training," in *Riga Technical University*, Latvia, 2017.

[34] Teilo and O. a. Keiron, "An Introduction to Convolutional Neural Networks," pp. 1-11, 2 12 2015.

[35] Ionescu and P. S. a. R. Tudor, "Optimizing the Trade-Off between Single-Stage and Two-Stage Deep Object Detectors using Image Difficulty Prediction," in *2018 20th International*

*Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*, Timisoara and Romania, 2018.

[36] J. Redmon and R. G. a. S. Divvala, "You Only Look Once: Unified, Real-Time Object Detection," 9 5 2016.

[37] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed and C.-Y. F. a. A. C. Berg, "SSD: Single Shot MultiBox Detector," in *Springer International Publishing*, 2016.

[38] R. Girshick, J. Donahue and T. D. a. J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," 22 10 2014.

[39] K. He, X. Zhang and S. R. a. J. Sun, "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. Volume: 37, no. Issue: 9, pp. 1904 - 1916, 29 12 Sept. 1 2015.

[40] J. Dai, Y. Li and K. H. a. J. Sun, "R-FCN: Object Detection via Region-based Fully Convolutional Networks," in *30th Conference on Neural Information Processing Systems*, Spain, 2016.

[41] G. Ross, "Fast R-CNN," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile, 18 February 2016.

[42] S. Ren, K. He and R. G. a. J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," 6 1 2016.

[43] Simon and K. D. a. Richard.M, "Optimally splitting cases for training and testing high dimensional classifiers," 2011.

[44] R. Padilla, S. L. Netto and E. a. Silva.A, "A Survey on Performance Metrics for Object-Detection Algorithms," in *ResearchGate*, Brazil, 2020.

[45] Getahun Tigistu, “Automatic Flower Disease Identification using image processing”, Master’s Thesis. Department of computer Science, Addis Ababa University, Addis Ababa, Ethiopia.

## Appendix A: Interview Questions

Interview Questions

1. Where does mosque could be found?
2. Does mosque structure differ from place to place?
3. Are there different types of mosque building?
4. Does mosque have a same structure appearance?
5. Does mosque have same size?
6. What are the common external components that could differ it from existed type of building?
7. Does all mosque have same external component?

**Declaration**

I, the undersigned, declare that this research is my original work and has not been presented for degree in any other university, and that all sources of materials used for the research have been acknowledged.

**Declared by:**

Name: Samrawit Ergete

Signature: _____

Date: _____

**Confirmed by advisor:**

Name: Ayalew Belay (PhD)

Signature: _____

Date: _____

Place and date of submission: Addis Ababa University, October 2020.