

***Project Report on***

***“CAPTION BOT FOR ASSISTIVE VISION”***

*Submitted in the partial fulfillment for the requirements of the degree of*

BACHELOR OF ENGINEERING IN

COMPUTER SCIENCE AND ENGINEERING

*Submitted By*

|  |  |
| --- | --- |
| **Shawez Khan** | **1BY17CS155** |
| **Srinivas V** | **1BY17CS170** |
| **Syed Rashid Pasha** | **1BY17CS175** |
| **Vimal Kumar Sancheti** | **1BY17CS187** |

*Under the guidance of*

Dr. Satish Kumar T

Associate Professor Department of CSE, BMSIT&M



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING BMS INSTITUTE OF TECHNOLOGY & MANAGEMENT YELAHANKA, BENGALURU - 560064.

2020-2021

## BMS INSTITUTE OF TECHNOLOGY &MANAGEMENT

### YELAHANKA, BENGALURU – 560064

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**



**CERTIFICATE**

This is to certify that the Project work entitled **“Caption Bot For Assistive Vision”** is a bonafide work carried out by **Srinivas V 1BY17CS170, Rashid Pasha 1BY17CS175, Vimal Kumar 1BY17CS187,Shawez Khan 1BY17CS155** in partial fulfillment for the award of **Bachelor of Engineering Degree in Computer Science and Engineering** of the **Visvesvaraya Technological University, Belagavi** during the year 2020-2021. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in this report. The project report has been approved as it satisfies the academic requirements in respect of project work for B.E Degree.

|  |  |  |
| --- | --- | --- |
| **Signature of the Guide** | **Signature of the HOD** | **Signature of Principal** |
| Dr. Satish Kumar T | Dr. Bhuvaneshwari C | Dr. Mohan Babu G N |
| Associate Professor, Dept. of CSE, BMSIT&M | Professor & HOD, Dept. of CSE, BMSIT&M | Principal, BMSIT&M |
|  | **External VIVA-VOCE** |  |
| Name of the Examiners |  | ignature with Date |
| **1.** |  |  |
| **2.** |  |  |

# ACKNOWLEDGEMENT

We are happy to present this project after completing it successfully. This project would not have been possible without the guidance, assistance and suggestions of many individuals. We would like to express our deep sense of gratitude and indebtedness to each and every one who have helped us make this project a success.

We heartily thank our **Principal, Dr. MOHAN BABU G N, BMS Institute of Technology & Management,** for his constant encouragement in taking up this project.

We heartily thank our **Head of the Department, Dr. Bhuvaneshwari C, Department of Computer Science and Engineering, BMS Institute of Technology & Management,** for her constant encouragement and inspiration in taking up this project.

We gracefully thank our Project Coordinator and Project Guide, **Dr. Satish Kumar T, Associate Professor, Department of Computer Science and Engineering,** for his intangible support and for being constant backbone for our project.

Special thanks to all the staff members of Computer Science Department for their help and kind co-operation.

Lastly, we thank our parents and friends for the support and encouragement given to us in completing this precious work successfully.

|  |  |
| --- | --- |
| **SHAWEZ KHAN** | **1BY17CS155** |
| **SRINIVAS V** | **1BY17CS170** |
| **SYED RASHID PASHA** | **1BY17CS175** |
| **VIMAL KUMAR** | **1BY17CS187** |

ABSTRACT

In navigating and understanding an outdoor environment, our world often requires the ability to see. People with visual impairments are, therefore, faced with significant challenges in exploring these environments. Deep learning has the potential to alleviate part of the frustrations they face. In this thesis, we assess the effectiveness of using deep learning to assist people with visual impairments.

It is very interesting in how machines can automatically describe the content of images using human language. In order to gain a deeper insight of this computer vision topic, we decided to implement current state-of-the-art image caption generator Show, attend and tell: Neural image caption generator with visual attention. Our neural network based image caption generator is implemented in Python powered by Pytorch machine learning library. We have identified five major components in our pipeline: (R1) data prepossessing; (R2) Convolution Neural Network (CNN) as an encoder; (R3) attention mechanism; (R4) Recurrent Neural Network (RNN) as a decoder; (R5) Sentence Generation and evaluation.

## II

TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
|  | **ACKNOWLEDGMENT ABSTRACT**  **TABLE OF CONTENTS LIST OF FIGURES LIST OF TABLES** | **I II III IV V** |
| **1** | **INTRODUCTION** | 1 |
|  | 1.1. PROBLEM STATEMENT | 1 |
|  | 1.2. MOTIVATION | 1 |
|  | 1.2.1.VISUALLY IMPARED PEOPLE | 1 |
|  | 1.2.2.TRAFFIC LIGHT SYSTEMS | 3 |
|  | 1.2.3.RISE IN ROAD TRAFFIC ACCIDENTS | 4 |
|  | 1.3. OBJECTIVES | 5 |
|  | 1.4. EXISTING SYSTEM | 5 |
|  | 1.5. PROPOSED SYSTEM | 6 |
| **2** | **LITERATURE SURVEY** | 7 |
| **3** | **DESIGN** | 12 |
|  | 3.1. OVERALL PROCESS OF THE PROJECT | 12 |
|  | 3.2. ARCHITECTURAL DESIGN – LEVEL 0 DFD | 13 |
|  | 3.3. COMPONENT DESIGN – LEVEL 1 DFD | 14 |
|  | 3.4. COMPONENT DESIGN – LEVEL 2 DFD | 16 |
|  | 3.4.1. CNN ALGORITHM | 16 |
|  | 3.5. BEHAVIORAL DESIGN | 19 |
|  | 3.6. STRUCTURAL DIAGRAM | 19 |
| **4** | **IMPLEMENTATION** | 22 |
|  | 4.1. DOMAIN – MACHINE LEARNING | 22 |
|  | 4.2. CONVOLUTION NEURAL NETWORKS | 23 |
|  | 4.3. PROGRAMMING LANGUAGE – PYTHON | 24 |
|  | 4.4. INTEGRATED DEVELOPMENT & LEARNING |  |
|  | ENVIRONMENT | 25 |
|  | 4.5. HARDWARE REQUIREMENTS | 26 |
|  | 4.6. SOFTWARE REQUIREMENTS | 26 |
|  | 4.7. FUNCTIONAL REQUIREMENTS | 26 |
|  | 4.8. NON-FUNCTIONAL REQUIREMENTS | 27 |
|  | 4.9. DATASETS | 27 |

**TESTING** 30

30

|  |  |
| --- | --- |
| 5.1 | IMPORTANCE OF TESTING |
| 5.2 | OBJECT-ORIENTED TESTING |
| 5.3 | MANUAL TESTING |
| 5.4 | UNIT TESTING |
| 5.5 | INTEGRATION TESTING |
| 5.6 | VALIDATION TESTING |
| 5.7 | VERIFICATION AND  VALIDATION TEST CASES |
| 5.8 | UNIT TESTING |

30

31

31

31

32

33

CONCLUSION AND RESULTS 34

CODE IMPLEMENTATION AND SCREENSHOTS 36

[REFERENCES 46](#_TOC_250000)

## III

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| FIGURE NO. | CAPTION | PAGE NO. |
| 1.2.2.1 | REAL TIME TRAFFIC LIGHT SYSTEM | 3 |
| 1.4.1 | EXAMPLE OF TRAFFIC LIGHT DETECTION PROCESS | 6 |
| 3.1.1 | GENERAL FLOW OF THE DEVELOPED SYSTEM | 12 |
| 3.2.1 | LEVEL 0 DATA FLOW DIAGRAM OF THE DEVELOPED |  |
|  | MODEL | 13 |
| 3.3.1 | LEVEL 1 DATA FLOW DIAGRAM OF THE DEVELOPED |  |
|  | MODEL | 14 |
| 3.4.1 | LEVEL 2 DATA FLOW DIAGRAM OF THE DEVELOPED |  |
|  | MODEL | 16 |
| 3.4.1.1 | EXAMPLE OF AN INPUT GIVEN TO THE CNN |  |
|  | ALGORITHM | 17 |
| 3.4.1.2 | GRID FORMATION BY THE CNN FRAMEWORK | 17 |
| 3.4.1.3 | CNN ALGORITHM ARCHITECTURE | 18 |
| 3.5.1 | BEHAVIORAL DESIGN OF THE MODEL | 19 |
| 3.6.1 | STRUCTURAL DIAGRAM OF THE CNN |  |
|  |  | 20 |
| 4.2.1 | EXAMPLE OF A SIMPLE CONVOLUTION NEURAL |  |
|  | NETWORK | 24 |
| 4.3.1 | EXECUTION OF A PYTHON STATEMENT VIA |  |
|  | COMMAND LINE | 25 |
| 4.4.1 | INPUT WINDOW IN IDLE | 25 |

|  |  |  |
| --- | --- | --- |
| 4.4.2  6.1  7.1  7.2  7.3  7.4  7.5 | OUTPUT WINDOW IN IDLE  GRAPH DEPICTING VALIDATION ACCURACY VS EPOCH VALUE  SNAPSHOT OF IMPLEMENTATION 1  SNAPSHOT OF IMPLEMENTATION 2  WEB INTERFACE  FILE FOR DETECTION  VOICE OUTPUT | 26  34  40  44  44  45  45 |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| TABLE NO. | TABLE NAME | PAGE NO. |
| 5.2.1  5.3.1 | VERIFICATION AND VALIDATION TESTING  UNIT TESTING | 32  33 |

**CHAPTER 1: INTRODUCTION**

* 1. **PROBLEM STATEMENT**

Independent navigation and exploration of a city has long been a daunting task for visually impaired individuals. Guide dogs and canes have been the primary assistive devices used by the blind community to assist them with outdoor navigation. Though they are indispensable to the population, they cannot help the user with understanding new environments and navigating through them. This challenge is exacerbated when the environment the user is navigating is unknown. For this reason, they tend to remain in known environments , as they learn specific landmarks of those routes, such as intersection characteristics. Various assistive devices have been proposed to help people with visual impairments with the various tasks involved in this challenge, such as:

* Navigation through outdoor environments, e.g., talking GPS systems .
* Safe crossing at intersections, e.g., systems that employ inertial measurement sensors to provide feedback regarding heading during crossing .

. • Situational awareness, e.g., systems that provide information about points of interest around the user .

BMSIT&M, Department of CSE 2020-21

BMSIT&M, Department of CSE 2020-21

.

## MOTIVATION

* + 1. **VISUALLY IMAPRIED PEOPLE**

Navigating Around the Places

The biggest challenge for a blind person, especially the one with the complete loss of vision, is to navigate around places. Obviously, blind people roam easily around their house without any help because they know the position of everything in the house. People living with and visiting blind people [must make sure not to move things around](https://wecapable.com/kitchen-adaptations-for-visually-impaired/) without informing or asking the blind person.

Commercial places can be made easily [accessible for the blinds with tactile tiles](https://wecapable.com/tactile-paving-tiles-meaning-blind-persons/). But, unfortunately, this is not done in most of the places. This creates a big problem for blind people who might want to visit the place.

Finding Reading Materials

Blind people have a tough time finding good reading materials in accessible formats. Millions of people in India are blind but we do not have even the proper [textbooks in braille,](https://wecapable.com/braille-translator/english-to-braille-converter) leave alone the novels and other leisure reading materials. Internet, the treasure trove of information and reading materials, too is mostly inaccessible for the blind people. Even though a blind person can use [screen](https://wecapable.com/best-screen-readers-free-paid/) [reading software](https://wecapable.com/best-screen-readers-free-paid/) but it does not make the Internet surfing experience very smooth if the websites are not designed accordingly. Blind person depends on the image description for understanding whatever is represented through pictures.

Blind people [do lead a ***normal life***](https://wecapable.com/blind-people-activities/) with their own style of doing things. But, they definitely face troubles due to inaccessible infrastructure and social challenges. Let us have an empathetic look at some of the daily life problems, struggles and challenges faced by the blind people.

Arranging Clothes

As most of the blind people depend on the objects’ shape and texture to identify them — arranging the laundry becomes a challenging task. Although a majority of blind people device their own technique to recognize and arrange at least their own clothes but it still is a challenging chore. This becomes a daredevil task if it’s about pairing and arranging the socks. All this is because recognizing colors is almost impossible for the persons with total blindness.

Overly Helpful Individuals

It is good to be kind and help others. But overly helpful individuals often create problems for the blind person. There are lots of individuals who get so excited to [help a disabled person](https://wecapable.com/disability-etiquette-basics/) that they forget even to ask the person whether she needs help or not. A blind person might be doing something painfully slow (from your perspective) but you should not hurry in doing the work without asking the person properly. You might end up creating some trouble for the blind person.

Getting Devices to Become Independence

The most valuable thing for a disabled person is gaining independence. A blind person can lead an independent life with some specifically designed adaptive things for them. There are lots of adaptive equipment that can enable a blind person to live their life independently but they are not easily available in the local shops or markets. [Refreshable Braille Display](https://wecapable.com/refreshable-braille-display/) is an example of such useful devices. A blind person needs to hunt and put much effort to get each equipment that can take them one step closer towards independence.

Everyone faces challenges in their life… blind people face a lot more. But, this certainly does not mean that you can show sympathy to blind persons. They too, just like any individual, take up life’s challenges and live a normal life, even if it does not seem normal to the sighted individuals.

## TRAFFIC LIGHT SYSTEMS

Traffic lights, also known as traffic signals, traffic lamps, traffic semaphore, signal lights, stop lights, robots (in [South Africa](https://en.wikipedia.org/wiki/List_of_South_African_slang_words) and most of [Africa](https://en.wikipedia.org/wiki/Africa)), and traffic control signals (in technical parlance), are signalling devices positioned at [road intersections,](https://en.wikipedia.org/wiki/Intersection_(road)) [pedestrian crossings,](https://en.wikipedia.org/wiki/Pedestrian_crossing) and other locations to control flows of traffic.

Traffic lights alternate the [right of way](https://en.wikipedia.org/wiki/Right-of-way_(traffic)) accorded to users by illuminating lamps or LEDs (Light Emitting Diode) of standard colours (red, amber (yellow), and green) following a universal [colour](https://en.wikipedia.org/wiki/Colour_code) [code.](https://en.wikipedia.org/wiki/Colour_code) In the typical sequence of colour phases:

* + The green light allows traffic to proceed in the direction denoted, if it is safe to do so and there is room on the other side of the intersection.
  + The amber light warns that the signal is about to change to red. In a number of European countries – among them the [United Kingdom](https://en.wikipedia.org/wiki/United_Kingdom) – a phase during which red and yellow are displayed together indicates that the signal is about to change to green. Actions required by drivers on a yellow light vary, with some jurisdictions requiring drivers to stop if it is safe to do so, and others allowing drivers to go through the intersection if safe to do so.
  + A flashing amber indication is a warning signal. In the [United Kingdom,](https://en.wikipedia.org/wiki/United_Kingdom) a flashing amber light is used only at [pelican crossings,](https://en.wikipedia.org/wiki/Pelican_crossing) in place of the combined red–amber signal, and indicates that drivers may pass if no pedestrians are on the crossing.
  + The red signal prohibits any traffic from proceeding.
  + A flashing red indication is treated as a [stop sign.](https://en.wikipedia.org/wiki/Stop_sign)

In some countries traffic signals will go into a flashing mode if the conflict monitor detects a problem, such as a fault that tries to display green lights to conflicting traffic. The signal may display flashing yellow to the main road and flashing red to the side road, or flashing red in all directions. Flashing operation can also be used during times of day when traffic is light, such as late at night.



Figure 1.2.2.1: Real Time Traffic Light System

## RISE IN ROAD ACCIDENTS

Road accidents—the leading cause of death by injury and the tenth-leading cause of all deaths globally—now make up a surprisingly significant portion of the worldwide burden of ill- health. An estimated 1.2 million people are killed in road crashes each year, and as many as 50 million are injured, occupying 30 percent to 70 percent of orthopaedic beds in developing countries hospitals. And if present trends continue, road traffic injuries are predicted to be the third-leading contributor to the global burden of disease and injury by 2020.

Developing countries bear a large share of the burden, accounting for 85 percent of annual deaths and 90 percent of the disability-adjusted life years (DALYs) lost because of road injury. And since road injuries affect mainly males (73 percent of deaths) and those between15 and 44 years old, this burden is creating enormous economic hardship due to the loss of familybreadwinners.

Road traffic injuries are predictable and preventable, but good data are important to understand the ways in which road safety interventions and technology can be successfully transferred from developed countries where they have proven effective. Awareness of the consequences of road traffic injuries is lagging among policymakers and the general public.

In developed countries, road traffic death rates have decreased since the 1960s because of successful interventions such as seat belt safety laws, enforcement of speed limits, warnings about the dangers of mixing alcohol consumption with driving, and safer design and use of roads and vehicles.

As in developed countries, driver impairment is an important component of road traffic accidents in developing countries. Driving at excess speeds, while under the influence of alcohol or drugs, while sleepy or tired, when visibility is compromised, or without protective gear for all vehicle occupants are major factors in crashes, deaths, and serious injuries.

In general, pedestrians, cyclists, and moped and motorcycle riders are the most vulnerable road users as well as the heaviest users of roads in poor countries. Most people who use public transportation, bicycles, or mopeds and motorcycles or who habitually walk are poor, illuminating the higher risk borne by those from less privilege.

In Asia, for instance, motorized two- and three-wheelers (such as motorized rickshaws) will make up the anticipated growth in numbers of motor vehicles.

## OBJECTIVES

The most important objective of this project is to enable visually impaired people to perform real- life object detection. The camera should be able to detect and classify the object and return a description in voice format to the visually impaired person.

It is important to ensure that the algorithm is trained in such a way that it detects the important objects for each model, and not other objects that have the same colors or features. It is also necessary to ensure that in the case of multiple objects present in a single scenario, the algorithm should be able to identify the correct object that it should follow, in order to prevent any mishaps.

## EXISTING SYSTEM

Currently, there are a number of systems that have been developed in order to perform the task of object detection for blind people. Various different methodologies and algorithms have been implemented in order to achieve the desired results.

Some of these methodologies include Single Shot Detection for Feature Extraction, Fully Convolutional Network with Clustering using the DBSCAN (Density-based Spatial Clustering of Applications with Noise) algorithm, Aggregation Channel Feature model in order to achieve object recognition using template matching, Stereo Imagery and Odometry, Classification based on Shape Characteristics, Color based Segmentation Process in order to perform object detection, Detection of Signs based on Region of Interest, Detection based on Color Density and Clustering etc.

Though the above implemented methodologies and algorithms have been successful in achieving traffic signal detection and recognition, there are many shortcomings that have been recorded in the implementation of the above systems.

Some of the shortcomings are varying labels present in the dataset, false positives leading to errors, annotations missing in the images present in the dataset, absence of classification step that reduces the accuracy of the model, detection of fake flashing luminous objects, non- parallel computation process that hampers real time requirements, harsh weather conditions not being considered, etc.

Considering the above shortcomings, the main aim of this project is to identify and classify objects in real time by considering all possible illuminations and weather conditions, in order to make the model more practical and accurate.



Figure 1.4.1: Example of Traffic Light Detection Process

## PROPOSED SYSTEM

The proposed system aims at performing real time object detection for visually impaired people. This can be achieved by implementing machine learning and image classification techniques. Artificial Neural Networks, and Convolution Neural Networks in specific, are one of the most accurate methods in order to achieve the desired result.

The basic ideology behind the proposed system/model is to develop a object detection model that is trained using a publicly available dataset. The model is trained using the various images present in the dataset, for which convolution neural networks is used. The trained model is then tested using the test dataset in order to test the working and prediction accuracy of the model. It consists of more than thirteen thousand images that have been captured in various scenarios, thus helpful in order to build a wholesome and accurate object detection model.

# CHAPTER 2: LITERATURE SURVEY

In the field of object detection, much of research has gone into the detection of traffic lights and road signs and other objects. These vary on the techniques used, the environments considered, andthe vehicles targeted for implementation on. Some of the various researches conducted over this topic are presented below.

**Ozcelik et al** have proposed A Vision Based object Detection and Recognition Approach [7]. In the proposed method, images have been taken using a camera in the first step. Image processing techniques are performed step by step to detect the objects in the received image through the computer. When traffic lights are detected, the received RGB image is converted into HSV formatto perform chromatic separation from uniform and non-chromatic elements in the image. By performing a color-based segmentation process on the obtained HSV format image, the locationsof traffic lights in the image are easily detected. The color of the traffic light is easily determined through the SVM (Support Vector Machines) classification model, which is a machine learning algorithm prepared beforehand, after the location of the traffic lights is determined in the image.

**Muller et al** have proposed Detecting objects by Single Shot Detection techniques [6]. This proposed methodology presents a deep learning approach for accurate object detection in adapting a Single Shot Detection (SSD) approach. SSD performs object proposals creation and classification using a single Convoluted Neural Networks (CNN). The original SSD struggles in detecting very small objects, which is essential for traffic light detection. But by the adaptations incorporated in this proposed methodology it is possible to detect objects much smaller than ten pixels without increasing the input image size.

**Behrendt et al** have proposed A Deep Learning Approach to objects by Detection, Tracking, and Classification [2]. This proposed methodology presents a complete system consisting of a objects detector, tracker, and classifier based on deep learning, stereo vision, and vehicle odometry which perceives objects in real-time. Within the scope of this work, three major contributions are presented.

. When selecting the confidence threshold that yield sequal error rate, the proposed system was able to detect objects as small as 4 pixels in width. The third contribution is a traffic light tracker which uses stereo vision and vehicle odometry to compute the motion estimate of traffic lights and a neural network to correct the motion estimate.

**Li et al** have proposed a object Light Recognition Technique for Complex Scene with Fusion Detections [5]. In this proposed methodology a robust object recognition model based on vision information is introduced for on-vehicle camera applications. It includes three aspects. First, in order to reduce computational redundancy, the aspect ratio, area, location, and context of traffic lights are utilized as prior information, which establishes a task model for objects recognition. Second, in order to improve the accuracy, a series of improved methods based on an aggregate channel feature method, including modifying the channel feature for each types of object and establishing a structure of fusion detectors is proposed. Third, a method of inter-frame information analysis, utilizing detection information of previous frame to modify original proposal regions is introduced, which makes the accuracy further improved.

**Saini et al** have proposed A Vision-Based object Detection and State Recognition technique [9]. The proposed technique presents a vision-based object structure detection and convolution neural network (CNN) based state recognition method, which is robust under different illumination and weather conditions. In the first step, object candidate regions are generated by performing Hue- Saturation-Lightness (HSV) based color segmentation, which are then filtered out using shape and area analysis. Further, to incorporate the structural information of object in diverse backgroundscenarios, Maximally Stable Extremal Region (MSER) approach is employed, which helps to localize the correct object structure in the image.

**Shi et al** have proposed Real-Time object Detection With Adaptive Background Suppression Filter [10]. This technique presents a novel vision-based object detection method for driving vehicles, which is fast and robust under different illumination conditions. The proposed method contains two stages: the candidate extraction stage and the recognition stage. On the candidate extraction stage, an adaptive background suppression algorithm to highlight the object candidate regions while suppressing the undesired backgrounds is proposed. On the recognition stage, each candidate region is verified and is further classified into different object semantic classes. The proposed method is then evaluated on video sequences (more than 5000 frames and labels) captured from urban streets and suburb roads in varying illumination and compared with other vision based detection approaches.

**Hamdi et al** have proposed to system for Road-signs Classification by ANN for Real-Time Implementation [3]. This system proposes a real-time algorithm for shape classification of traffic signs and their recognition to provide a driver alert system. The proposed algorithm is mainly composed of two phases: shape classification and content classification. This algorithm takes as input a list of Bounding Boxes generated in a previous work and will classify them. The traffic sign's shape is classified by an artificial neural network (ANN). Traffic signs are classified according to their shape characteristics, as triangular, squared and circular shapes. Combining color and shape information, traffic signs are classified into one of the following classes: danger, information, obligation or prohibition. The classified circular and triangular shapes are passed on to the second ANN in the third phase. These identify the pictogram of the road sign. The output of the second artificial neural network allows the full classification of the road sign.

**Abedin et el** have proposed a system for object Recognition Using Hybrid Features Descriptor and Artificial Neural Network Classifier [1]. This system represents a new approach for object Sign Recognition (OSR) system using hybrid features formed by two robust features descriptors, named Histogram Oriented Gradient(HOG) features and Speeded Up Robust Features(SURF) and artificial neural network (ANN) classifier . In the detection step, the region of interest (sign area) is segmented using color based thresholding algorithm, post processed to filter the unwanted region. Next robust features vector named Distance to Borders (DtBs) of the segmented blob is formed to verify the shape of the object . Finally the recognition of the object sign is implemented using ANN classifier upon the training of hybrid features descriptor. The proposed system simulated on offline road scene images shows a high classification rate in the recognition stage. The performance of the ANN model is illustrated in terms of cross entropy,

confusion matrix and receiver operating characteristic (ROC) curves. The simulation results illustrate that recognition using hybrid feature descriptor outperforms in all classifier and the recognition accuracy of ANN is higher than classifier stated above.

**Jung et el** have proposed a Real-time Traffic Sign Recognition System with Deep Convolutional Neural Network [4]. In this system, 6 types of traffic sign images are trained by 5 convolutional neural network architecture. In the detection phase, light-weight color-based segmentation algorithm and Hough transform algorithm are applied to extract candidate regions of traffic signs. The recognition system nearly achieves real-time performance. The recognition system is implanted into an autonomous vehicle. Different types of traffic signs are trained consistently, and development of clustering algorithm is considered as a future work for robust recognition system.

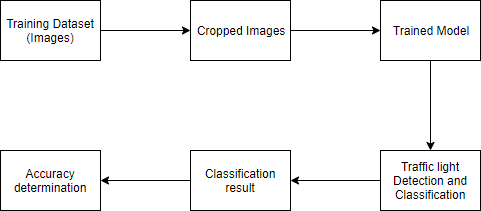
The various research works mentioned above present many methodologies for the detection of objects . But these methodologies are also hampered due to the presence of various drawbacks.The systems presented in [1], [4], [7], [10], fail in considering a vast variety of both Training and Testing datasets, that can be considered to scale the respective systems. [3] and

[6] fail on being tested in harsh weather conditions. [2] requires high quality hardware components, to achieve the expected levels of accuracy.

# CHAPTER 3: DESIGN

The following section describes the overall design of the developed real-time object detection system. Systems design is the process of defining the [architecture,](https://en.wikipedia.org/wiki/Systems_architecture) modules, interfaces, and [data](https://en.wikipedia.org/wiki/Data) for a [system](https://en.wikipedia.org/wiki/System) to satisfy specified requirements.

## OVERALL PROCESS OF THE PROJECT



Object Detection and Classification

Figure 3.1.1: General Flow of the Developed System

Figure 3.1.1 represents the overall process of the project. It depicts the general flow of the developed model. The following steps constitute the basic flow of the developed system:

* + 1. Images depicting various scenarios in a real time object environment are captured using multiple cameras. These images constitute the dataset which is used to train, validate and test the model.
    2. The images in the dataset cover a decent variety of road scenes. These images consist of many objects apart from traffic signals such as trees, vehicles, road signs, buildings, decorative lights, etc. The images are cropped to meet the bounding box dimensions mentioned in the configuration file and labeled accordingly. Now the cropped images would contain only the required object.
    3. The machine is trained using the cropped images to apply various filters and algorithms to determine the color of the traffic signal accordingly. A trained model is generated after the training process.
    4. The trained model is validated using a validation data set, which is a subset of the original dataset that is gathered. Validation is done in order to improve the accuracy of the model by adjusting its parameters.
    5. The test dataset (images) is then given as input to the trained model in order to verify the working of the trained model.
    6. The model then performs the classification of the object and it outputs the description of the object in a voice format .
    7. The extent to which the trained model performs the classification correctly constitutes to the accuracy of the classification. This accuracy of the model is output along with the classification result for a particular image.

## ARCHITECTURAL DESIGN – LEVEL 0 DFD (Data Flow Diagram)

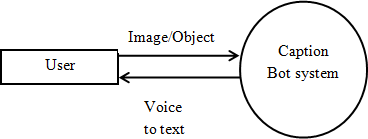


Figure 3.2.1: Level 0 Data Flow Diagram of the Developed Model

Figure 3.2.1 depicts the Level 0 Data Flow Diagram for the proposed system. A level 0 Data Flow Diagram depicts a basic overview of the proposed system/model. The proposed system’s main functionality is object detection and classification in real time. In order to perform this operation, we require a training dataset, that is used to train the designed model.

The training dataset consists of a set of images that have been captured in real time, in order to train the model to detect and classify objects. Once the proposed model is trained, a validation set is used in order to optimize the parameters of the classifier and improve the classification accuracy of the model. This is then followed by testing of the model, for which a test dataset is given as input to the model. The test dataset consists of images that the model has never seen before. This dataset is used in order to check the working of the model and calculate its accuracy of prediction. In the end, for a particular scenario, the model displays the classification result of the traffic signal along with the prediction accuracy of the model for the given dataset.

## COMPONENT DESIGN – LEVEL 1 DFD

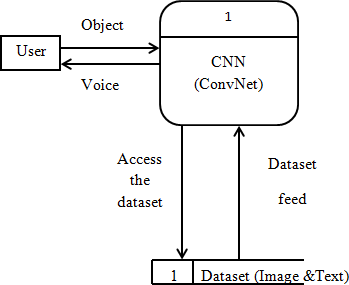


Figure 3.3.1: Level 1 Data Flow Diagram of the Developed Model

Figure 3.3.1 depicts the Level 1 Data Flow Diagram for the proposed system. A Level 1 Data Flow Diagram is more detailed than a Level 0 Data Flow Diagram. It breaks down the main processes into subprocesses that can be analyzed and improved at a deeper level. The above diagram shows how the training dataset is subject to image processing and cropping, before it can be used to train the model.

Image processing is done in order to identify the various objects in the images taken in real time, after which the signals are labeled and bounded by a box in order to perform the cropping function. This process is also known as object detection. In object detection, the main aim is to identify all objects that are present in an image and drawing bounded boxes around them.

There are two main types of object detection algorithms:

1. Algorithms based on classification
2. Algorithms based on regression

In this project, a classification-based algorithm knows as the CNN (Convolution Neural Network) Algorithm has been used to perform object detection.

This algorithm is used for real-time object detection. In this algorithm, classes and bounded boxes are predicted for the whole image in one run of the algorithm.

Feature Extraction In this step, the features in the image are extracted using principal component analysis using NumPy. CNN is used for scene classification and RNN is used for detecting objects and human attributes. Creating attributes In this step, the features extracted by the neural networks were used to define the attributes with its label strings.

## COMPONENT DESIGN – LEVEL 2 DFD

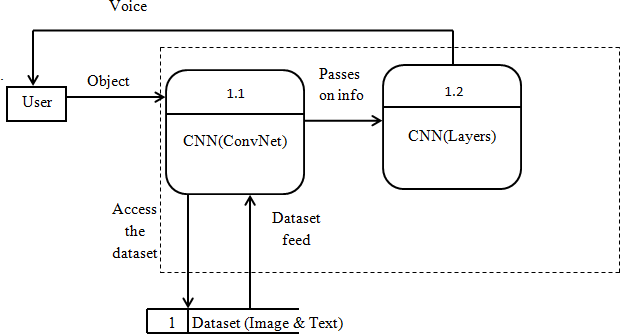


Figure 3.4.1: Level 2 Data Flow Diagram of the Developed Model

Figure 3.4.1 is a level 2 data flow diagram that shows the image processing step in detail. The object detection algorithm implemented in the developed system is known as the CNN algorithm.

## 3.4.1 Convolution Neural Network ALGORITHM

* A **Convolutional Neural Network (ConvNet/CNN)** is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other.
* The pre-processing required in a ConvNet is much lower as compared to other classification algorithms.
* While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics.
* The architecture of a ConvNet is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex.
* Individual neurons respondto stimuli only in a restricted region of the visual field known as the Receptive Field. A collection ofsuch fields overlap to cover the entire visual area.
* The objective of the Convolution Operation is to **extract the high-level features** such as edges, from the input image. ConvNets need not be limited to only one Convolutional Layer. Conventionally, the first ConvLayer is responsible for capturing the Low-Level features such as edges, color, gradient orientation, etc. With added layers, the architecture adapts to the High- Level features as well, giving us a network which has the wholesome understanding of images in the dataset, similar to how we would.
* There are two types of results to the operation — one in which the convolved feature is reduced in dimensionality as compared to the input, and the other in which the dimensionality is either increased or remains the same. This is done by applying **Valid Padding** in case of the former, or **Same Padding** in the case of the latter.

CNN first takes an input image:

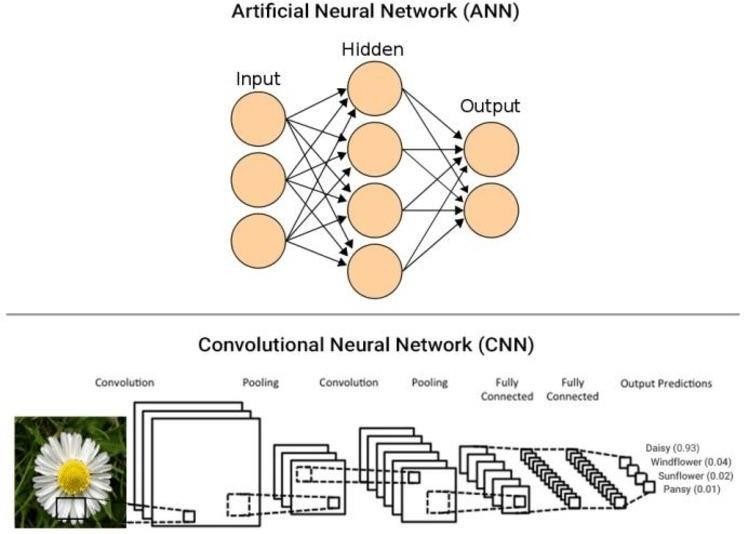


Figure 3.4.1.1: Example of an Input Given to the CNN Algorithm

Step 1: Convolution layer The Kernel

* + The role of the ConvNet is to reduce the images into a form which is easier to process, without losing features which are critical for getting a good prediction
  + if input image size is 5\*5 dimension and Kernel/filter dimension is 3\*3

then the output i.e. convolved feature is 3\*3 dimension(so we have reduced the image dimension) Step 2: Pooling layer.

* + Similar to the Convolutional Layer, the Pooling layer is responsible for reducing the spatial size of the Convolved Feature. This is to decrease the computational power required to process the data through dimensionality reduction. Furthermore, it is useful for extracting dominant features which are rotational and positional invariant, thus maintaining the process of effectively training of the model.

Step 3: Flattening.

* + Now that we have converted our input image into a suitable form for our Multi-Level Perceptron, we shall flatten the image into a column vector. The flattened output is fed to a feed-forward neural network(ANN) and back propagation applied to every iteration of training.

Step 4: Full Connection.

* + Adding a Fully-Connected layer is a (usually) cheap way of learning non-linear combinations of the high-level features as represented by the output of the convolutional layer. The Fully-Connected layer is learning a possibly non-linear function in that space.
  + From the training dataset(Labelled images) our CNN will identify and learn the features of images and when a new image is given to our CNN will classify into a right category.
  + Once CNN classifies an image then we will convert the text (category) into voice so that the person can easily come to know what is in front of him/her.
    - An image is nothing but a matrix of pixel values, right? So why not just flatten the image (e.g. 3x3 image matrix into a 9x1 vector) and feed it to a Multi-Level Perceptron for classification purposes? Uh.. not really.
    - In cases of extremely basic binary images, the method might show an average precision score while performing prediction of classes but would have little to no accuracy when it comes to complex images having pixel dependencies throughout.
    - A ConvNet is able to **successfully capture the Spatial and Temporal dependencies** in an image through the application of relevant filters. The architecture performs a better fitting to the image dataset due to the reduction in the number of parameters involved and reusability of weights. In other words, the network can be trained to understand the sophistication of the image better.

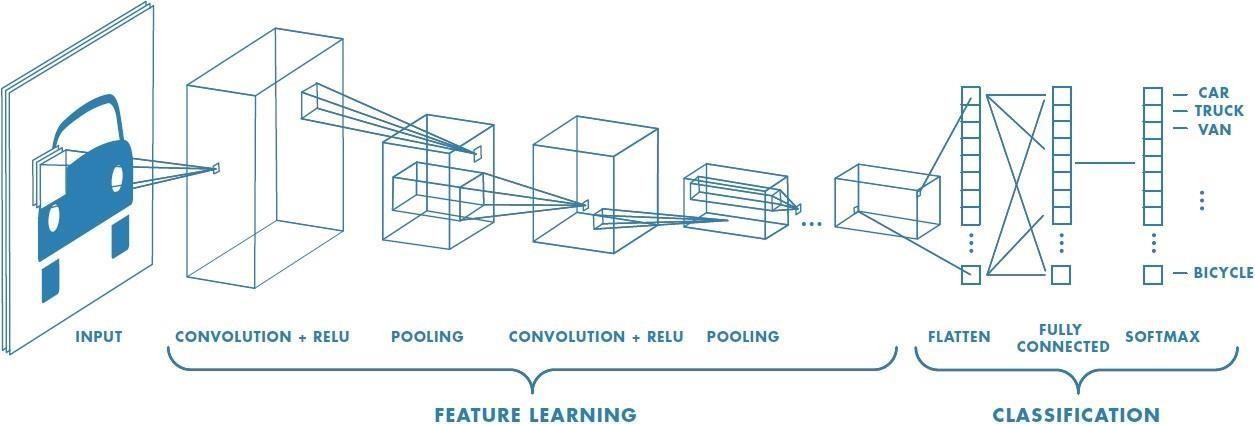


Figure 3.4.1.3: CNN algorithm Architecture

## BEHAVIORAL DESIGN

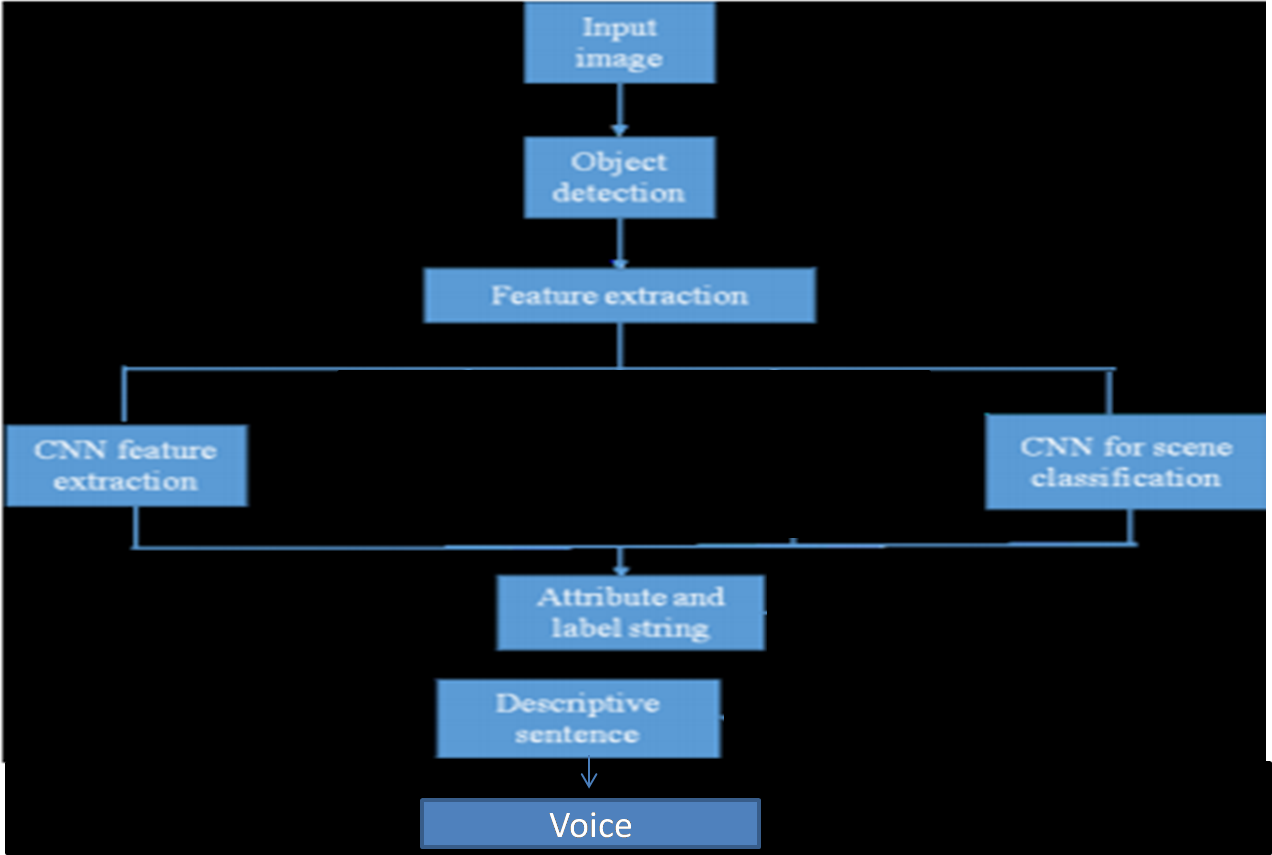


Figure 3.5.1: Behavioral Design of the Model

Figure 3.5.1 depicts the behavior of the developed system in different stages.

## STRUCTURAL DIAGRAM

A structure diagram is a conceptual modeling tool used to document the different structures that make up a system such as a database or an application. It shows the hierarchy or structure of the different components or modules of the system and shows how they connect and interact with each other. It is a tool used to guide developers to ensure that all parts of the system work as intended in relation to all the other parts.

Modules at top level called modules at low level. Components are read from top to bottom and left to right. When a module calls another, it views the called module as black box, passing required parameters and receiving results.

.

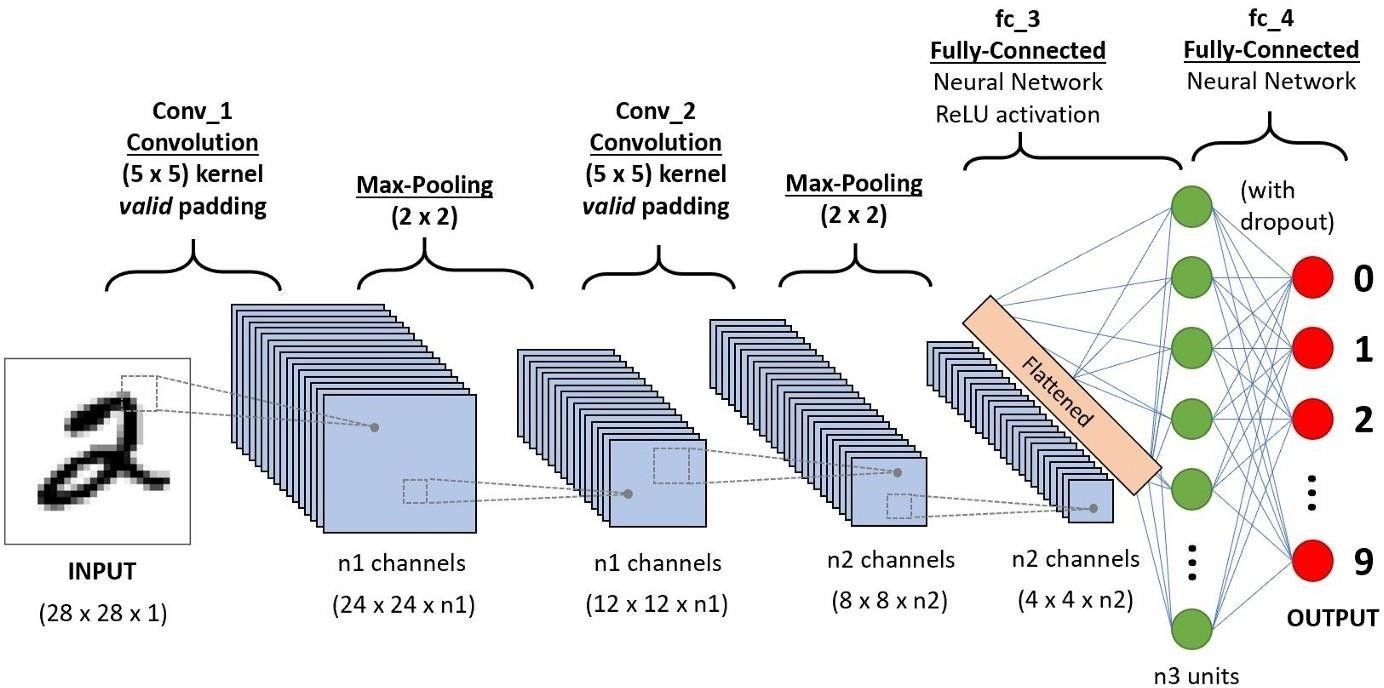


Figure 3.6.1: Structural Diagram of the CNN

# CHAPTER 4: IMPLEMENTATION

## DOMAIN – MACHINE LEARNING

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it learn for themselves.

The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide. The primary aim is to allow the computers learn automatically without human intervention or assistance and adjust actions accordingly. Machine learning enables analysis of massive quantities of data. While it generally delivers faster, more accurate results in order to identify profitable opportunities or dangerous risks, it may also require additional time and resources to train it properly.

Machine learning algorithms are often categorized as supervised or unsupervised.

* + - **Supervised machine learning algorithms** can apply what has been learned in the past to new data using labelled examples to predict future events. Starting from the analysis of a known training dataset, the learning algorithm produces an inferred function to make predictions about the output values. The system is able to provide targets for any new input after sufficient training. The learning algorithm can also compare its output with the correct, intended output and find errors in order to modify the model accordingly.
    - In contrast, **unsupervised machine learning algorithms** are used when the information used to train is neither classified nor labelled. Unsupervised learning studies how systems can infer a function to describe a hidden structure from unlabelled data. The system doesn’t figure out the right output, but it explores the data and can draw inferences from datasets to describe hidden structures from unlabelled data.
    - **Semi-supervised machine learning algorithms** fall somewhere in between supervised and unsupervised learning, since they use both labelled and unlabelled data for training – typically a small amount of labelled data and a large amount of unlabelled data. The systems that use this method are able to considerably improve learning accuracy. Usually, semi-supervised learning is chosen when the acquired labelled data requires skilled and relevant resources in order to train it / learn from it. Otherwise, acquiring unlabelled data generally doesn’t require additional resources.
    - **Reinforcement machine learning algorithms** is a learning method that interacts with its environment by producing actions and discovers errors or rewards. Trial and error search and delayed reward are the most relevant characteristics of reinforcement learning. This method allows machines and software agents to automatically determine the ideal behaviour within a specific context in order to maximize its performance. Simple reward feedback is required for the agent to learn which action is best; this is known as the reinforcement signal.

## CONVOLUTION NEURAL NETWORKS

Convolution neural networks (CNN) or connectionist systems are computing systems vaguely inspired by the [biological neural networks](https://en.wikipedia.org/wiki/Biological_neural_network) and [astrocytes](https://en.wikipedia.org/wiki/Astrocytes) that constitute animal [brains.](https://en.wikipedia.org/wiki/Brain) The neural network itself is not an algorithm, but rather a framework for many different [machine](https://en.wikipedia.org/wiki/Machine_learning) [learning](https://en.wikipedia.org/wiki/Machine_learning) algorithms to work together and process complex data inputs. Such systems "learn" to perform tasks by considering examples, generally without being programmed with any task- specific rules. For example, in [image recognition,](https://en.wikipedia.org/wiki/Image_recognition) they might learn to identify images that contain cats by analyzing example images that have been manually [labeled](https://en.wikipedia.org/wiki/Labeled_data) as "cat" or "no cat" and using the results to identify cats in other images. They do this without any prior knowledge about cats, for example, that they have fur, tails, whiskers and cat-like faces. Instead, they automatically generate identifying characteristics from the learning material that they process.

An CNN is based on a collection of connected units or nodes called [artificial neurons,](https://en.wikipedia.org/wiki/Artificial_neuron) which loosely model the [neurons](https://en.wikipedia.org/wiki/Neuron) in a biological brain. Each connection, like the [synapses](https://en.wikipedia.org/wiki/Synapse) in a biological brain, can transmit a signal from one artificial neuron to another. An artificial neuron that receives a signal can process it and then signal additional artificial neurons connected to it.

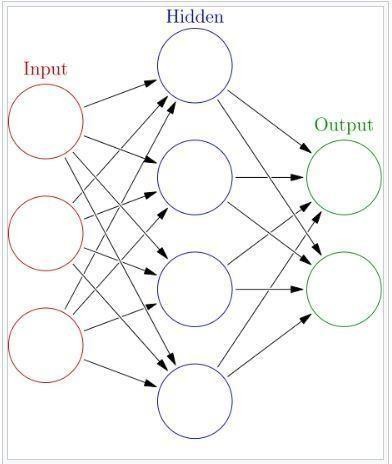


Figure 4.2.1: Example of a Simple Convolution neural network

Figure 4.2.1 represents a simple Convolution neural network with a single hidden layer. Here, each circular node represents an artificial neuron and an arrow represents a connection from the outputof one artificial neuron to the input of another. The connections between artificial neurons are called edges. Artificial neurons and edges typically have a weight that adjusts as learning proceeds.

## PROGRAMMING LANGUAGE – PYTHON

The programming language used in order to develop this entire project is Python. Python is an interpreted, high-level, general-purpose programming language. It has a design philosophy that emphasizes code readability, notably using significant whitespace. Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including procedural, object- oriented, and functional programming. Python’s large standard library, commonly cited as one of its greatest strengths, provides tools suited to many tasks. It includes modules for creating graphical user interfaces, connecting to relational databases, unit testing etc. Python can also serve as a scripting language for web applications.

From development to deployment and maintenance, Python helps developers be productive and confident about the software they’re building. [Benefits that make Python the best fit](https://steelkiwi.com/blog/why-python-django-are-your-top-choice-for-web-development/) for [machine](https://steelkiwi.com/blog/what-is-machine-learning/) [learning](https://steelkiwi.com/blog/what-is-machine-learning/) and AI-based (Artificial Intelligence) projects include simplicity and consistency, access to great libraries and frameworks for AI and machine learning (ML), flexibility, platform independence, and a wide community. These add to the overall popularity of the language. Python offers concise and readable code. While complex algorithms and versatile workflows stand behind machine learning and AI, Python’s simplicity allows developers to write reliable systems. Developers get to put all their effort into solving an ML problem instead of focusing on the technical nuances of the language.

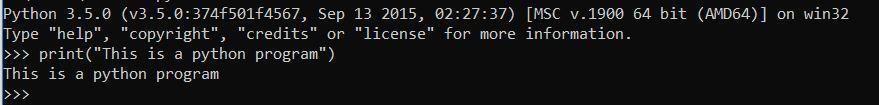


Figure 4.3.1: Execution of a Python Statement via Command Line

* 1. **INTEGRATED DEVELOPMENT AND LEARNING ENVIRONMENT** IDLE (short for integrated development environment or integrated development and learning environment) is an [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) for [Python,](https://en.wikipedia.org/wiki/Python_(programming_language)) which has been bundled with the default implementation of the language. It is packaged as an optional part of the Python packaging with many [Linux distributions.](https://en.wikipedia.org/wiki/Linux_distributions) It is completely written in Python and the [Tkinter](https://en.wikipedia.org/wiki/Tkinter) GUI toolkit. It is cross-platform and avoids feature clutter.

Its main features are:

* Multi-window text editor with [syntax highlighting,](https://en.wikipedia.org/wiki/Syntax_highlighting) autocompletion, smart indent and other.
* Python shell with syntax highlighting.
* Integrated debugger with [stepping,](https://en.wikipedia.org/wiki/Program_animation) persistent [breakpoints](https://en.wikipedia.org/wiki/Breakpoint), and call stack visibility.



Figure 4.4.1: Input Window in IDLE

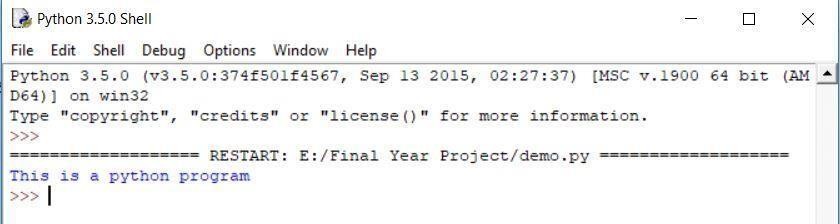


Figure 4.4.2: Output Window in IDLE

## HARDWARE REQUIREMENTS

* PROCESSOR: any processor with a speed of 1.5GHz or more
* RAM: 8GB RAM or more
* HARD DISK: 50GB or more
* Standard Input Device
* Standard Output Device

## SOFTWARE REQUIREMENTS

* OPERATING SYSTEM: Windows 8 or higher
* FRONT END TOOL: IDLE
* PROGRAMMING LANGUAGE: Python
* COMPILER: Integrated within IDLE

## FUNCTIONAL REQUIREMENTS

* The system should be able to process all images present in the dataset with equal importance and priority.
* The system should be able to identify objects present in the images, in order to perform model training.
* The system should be able to train the model based on the objects that have been detected and cropped in the processing stage.
* The validation set should be able to increase the accuracy and learning of the model by fine tuning the model and classifier parameters.
* The model should be able to make correct predictions when the test set is presented to the model.

## NON-FUNCTIONAL REQUIREMENTS

* PERFORMANCE: The developed object detection system should be able to train the classifier within a certain amount of time, and the training time should decrease as the number of iterations increases.
* SCALABILITY: The developed system should be able to train the model and predict the output on a variety of datasets, apart from the one used in this project.
* RELIABILITY: The developed system should produce accurate results, thus proving to be reliable to the users and the society.
* USABILITY: The developed system should be usable by people from different backgrounds, irrespective of their knowledge of the system.
* ENVIRONMENTAL: The developed system should be beneficial to the society, and should operate towards the betterment of the people and the surrounding environment.
* INTEROPERABILITY: The developed system should be able to run on various platforms, irrespective of the platform on which it was designed.

## DATASETS

Data is of utmost importance when it comes to machine learning algorithms and techniques. Machine learning algorithms are mathematical models that learn to find patterns in the input that is fed to them. This input is nothing but large amounts of data (instances). A collection of instances is known as a dataset.

In the field of machine learning, there are basically three categories of datasets:

1. Training set
2. Validation set
3. Test set

The training data set in Machine Learning is the actual dataset that is used to train the model for performing various actions. The training dataset is a collection of examples used for learning, in order to fit the parameters of a classifier.

The validation set is used in order to tune the parameters of the classifier. It is basically a collection or a dataset of examples that is used to tune the hyper parameters of a classifier.

It is sometimes also called the development set or the dev set. The validation set should follow the same probability distribution as that of the training set.

Finally, the performance of the classifier or the model is tested on an unseen test set. The test set consists of data that is used to provide an unbiased evaluation of the final model or the classifier. This test set is used to check whether the model is responding or working appropriately or not.

So, in general, a machine learning algorithm is a mathematical model that learns to find patterns in the input that is fed to it. This input is referred to as the training data. A validation data set is used in order to tune the parameters of the model and increase its accuracy and performance. Once a machine learning algorithm learns the underlying patterns of the training data, it needs to be tested on fresh data that it has never seen before. This is also known as the test data. It has to be ensured that the test dataset belongs to the same distribution as the training data. Another important point to note is that during the training process, only the training set or the validation set is available. The test set should not be used while training the classifier.

The dataset being used in this is an accurate dataset for vision-based object detection.

The scenes cover a decent variety of road scenes and typical difficulties:

* Busy street scenes inner-city
* Suburban multilane roads with varying traffic density
* Dense stop-and-go traffic
* Road-works
* Strong changes in illumination/exposure
* Overcast sky with light rain

This dataset contains 13427 camera images at a resolution of 1280x720 pixels. The camera images are provided as raw 12bit HDR (High Dynamic Range) images taken with a red-clear-clear-blue filter and as reconstructed 8-bit RGB (Red-Green-Blue) color images. The RGB images are provided for debugging and can also be used for training. The training set consists of 5093 images and the test set consists of 8334 images.

# CHAPTER 5: TESTING

Software testing is defined as an activity to check whether the actual results match the expected results and to ensure that the software system is [Defect](https://www.guru99.com/the-unconventional-guide-to-defect-management.html) free. It involves execution of a software component or system component to evaluate one or more properties of interest.

Software testing also helps to identify errors, gaps or missing requirements in contrary to the actual requirements. It can be either done manually or using automated tools.

Before the system goes live, it must be tested thoroughly. Testing reveals the presence of bugs in the system. Testing has two parts:

* + Verification – Does it implement a specific function correctly?
  + Validation – Does it confirm to the user requirements?

## IMPORTANCE OF TESTING

Testing is the measurement of software quality hence, one of the most important stages in software development. It involves executing an implementation of the software and its operational behavior to check that it is performing as required. One of the main goals of testing is to have a minimum number of test cases that will find a majority of the implementation errors.

## Object-Oriented Testing

In an object-oriented system four levels of testing can be identified:

* + - Testing the individual operations associated with objects.
    - Testing individual object classes.
    - Testing clusters of objects.
    - Testing the object-oriented

## Manual Testing

Manual testing is the process of manually testing software for defects. It requires a tester to play the role of an end user and use most of all features of the application to ensure correct behavior.

To ensure completeness of testing, the tester often follows a written test plan that leads them through a set of important test cases. A key step in the process of software engineering is testing the software for correct behavior prior to release to end users.

## Unit Testing

Unit testing is a method by which individual units of source code are tested to determine if they are fit for use. A unit is the smallest testable part of an application. In procedural programming a unit may be an individual function or procedure. Unit tests are typically written and run by software developers to ensure that code meets its design and behaves as intended.

The goal of unit testing is to isolate each part of the program and show that the individual parts are correct. A unit test provides a strict, written contract that the piece of code must satisfy. As a result, it affords several benefits. Unit tests find problems early in the development cycle. Testing cannot be expected to catch every error in the program: it is impossible to evaluate every execution path in all but the most trivial programs.

This is the first level of testing in which different modules are tested against design specification. Unit testing is done for the verification of the code produced during the coding phase and to test the internal logic of the modules. After coding, each module were tested and run individually. All the GUIs where data entry is required were tested found no errors.

## Integration Testing

Integration testing is a systematic technique for constructing the program structure while conducting test to uncover errors associated with interfacing. Many tested modules are intended into a subsystem, which is then tested. The different modules were linked together and tested. Correctness and accuracy were confirmed.

## Validation Testing

Validation testing provides the final assurance that the software meets all functional, behavioral and performance requirements. The software is completely assembled as a package, interfacing errors have been uncovered and collected, and a final series of software test –validation test may begin. Validation succeeds when the software works as per the user requirement.

Validation phase usually produces minor modifications in the software to deal with errors and failures that are uncovered. A test case is a dummy set of data used to input a system and test it.

## VERIFICATION AND VALIDATION TEST CASES

The following test cases were implemented as part of the testing process.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SL No | Test Case Description | Input | Expected Results | Test Case Results |
| 1 | Validation of training dataset | Training Images | All images displayed without failure | Pass |
| 2 | Verification of the image cropping function | Raw Images | Cropped Images consisting of only objects | Pass |
| 3 | Building of a train  model with interruption | Cropped Images | Trained Model | Fail |
| 4 | Building of a train model without interruption | Cropped Images | Trained Model | Pass |
| 5 | Predicting the test images using the trained model | Testing Images | Images with objects are detected | Pass |

Table 5.2.1: Verification and Validation Testing

From Table 5.2.1, it can be inferred that the building of the trained model failed, when there was an interruption that occurred.

## UNIT TESTING

During the building of the model using the train dataset, various Epoch values were tested, where an Epoch is one complete presentation of the data set to be learned to a learning program.

The following table contains the various Epoch values tested and their respective obtained resultant accuracies:

|  |  |  |
| --- | --- | --- |
| **Epoch Value** | **Validation Accuracy** | **Validation Loss** |
| 10 | 0.9325 | 0.1998 |
| 20 | 0.9375 | 0.1878 |
| 30 | 0.9474 | 0.1792 |
| 40 | 0.9417 | 0.1774 |
| 50 | 0.9431 | 0.1745 |

Table 5.3.1: Unit Testing

From the values present in Table 5.3.1, it can be seen that for an epoch value more than 30, the validation accuracy starts reducing, which is an indication of overfitting of data by 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. This means that the noise or random fluctuations in the training data is picked up and learned as concepts by the model. The problem is that these concepts do not apply to new data and negatively impact the model’s ability to generalize. Hence this overfitting of the data has to be avoided.

# CHAPTER 6: CONCLUSION AND RESULTS

After going through the entire process of researching, implementing and testing the developed model, it can be concluded that Convolution Neural Network Technique proves to be a very efficient method for the detection and classification of objects in real time.

One of the most important findings was during the training process. It was found that model training failed if the training process was interrupted in between. Due to the nature of the training process and the characteristics of the algorithm being used, it is necessary that the training process remains uninterrupted in order to make sure that model training completes successfully.

Another important finding was encountered while building the model. As mentioned in Table 5.3.1, the model was tested with different epoch values in order to monitor its performance at various stages. One epoch is when an entire dataset is passed forward and backward through the neural network only once. A training process may consist of more than one epoch. As the number of epochs increases, a greater number of times the weights are changed in the neural network.

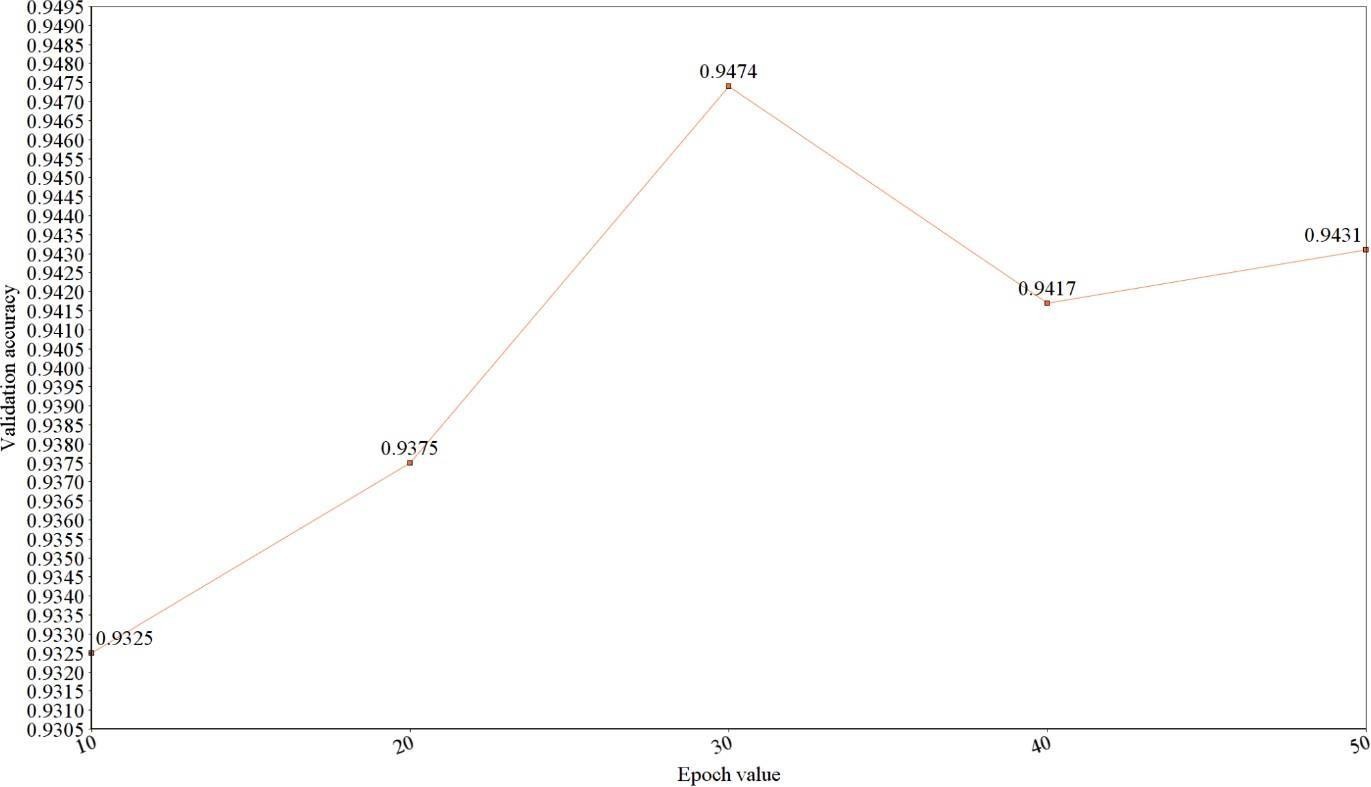


Figure 6.1: Graph depicting Validation Accuracy vs Epoch Value

Another important result was found during the prediction process. The developed model was able to detect multiple objects in a single image, and perform the prediction process on the object that is nearest to the object under consideration. This is important as the real-world scenario works in the exact same way.

# CHAPTER 7: CODE IMPLEMENTATION AND SCREENSHOTS

1. app.py

### CODE:

#!/usr/bin/env python

from flask import Flask,render\_template,request,send\_from\_directory app = Flask( name )

import os import keras

import numpy as np

from keras.preprocessing import image from gtts import gTTS

path1 = "D:/final-year project/Flask/static/model1/" path2 = "D:/final-year project/Flask/static/model2/" path3 = "D:/final-year project/Flask/static/model3/" path4 = "D:/final-year project/Flask/static/model4/"

@app.route("/") def hello():

return render\_template('index.html')

@app.route("/Model1",methods=['GET','POST']) def model1():

if request.method == 'POST':

f=request.files['file'] f.save(os.path.join(path1,f.filename)) p=os.path.join(path1,f.filename)

cnn = keras.models.load\_model('model4-traffic.h5')

import numpy as np

from keras.preprocessing import image

test\_image = image.load\_img('D:/final-year project/Flask/static/model1/traffic.jpg', target\_size = (64, 64))

test\_image = image.img\_to\_array(test\_image) test\_image = np.expand\_dims(test\_image, axis = 0) result = cnn.predict(test\_image)

if result[0][0] == 1:

prediction = 'red'

tts = gTTS("Traffic Signal is red,Cross the road as soon as possible") tts.save('D:/final-year project/Flask/static/music/1.wav')

else:

prediction = 'Green'

tts = gTTS("Traffic Signal is Green, Please do not move.") tts.save('D:/final-year project/Flask/static/music/1.wav')

return render\_template('audio.html') return render\_template('Model1.html')

@app.route("/Model2",methods=['GET','POST']) def model2():

if request.method == 'POST':

f=request.files['file'] f.save(os.path.join(path2,f.filename)) p=os.path.join(path2,f.filename)

cnn = keras.models.load\_model('model1-bus.h5') import numpy as np

from keras.preprocessing import image

test\_image = image.load\_img('D:/final-year project/Flask/static/model2/bus.jpeg', target\_size = (64, 64))

test\_image = image.img\_to\_array(test\_image) test\_image = np.expand\_dims(test\_image, axis = 0) result = cnn.predict(test\_image)

if result[0][0] == 1:

prediction = 'No-Rush'

tts = gTTS("Bus is empty You can get into Bus")

else:

tts.save('D:/final-year project/Flask/static/music/1.wav')

prediction = 'Rush'

tts = gTTS("Bus is Rush Please do not get into bus") tts.save('D:/final-year project/Flask/static/music/1.wav')

return render\_template('audio.html') return render\_template('Model2.html')

@app.route("/Model3",methods=['GET','POST']) def model3():

if request.method == 'POST':

f=request.files['file'] f.save(os.path.join(path3,f.filename)) p=os.path.join(path3,f.filename)

cnn = keras.models.load\_model('model3-person.h5') import numpy as np

from keras.preprocessing import image

test\_image = image.load\_img('D:/final-year project/Flask/static/model3/person.jpeg', target\_size = (64, 64))

test\_image = image.img\_to\_array(test\_image) test\_image = np.expand\_dims(test\_image, axis = 0) result = cnn.predict(test\_image)

if result[0][0] == 1:

prediction = 'person'

tts = gTTS("There is person in-front of you") tts.save('D:/final-year project/Flask/static/music/1.wav')

else:

prediction = 'No-person'

tts = gTTS("There is No-person in-front of you") tts.save('D:/final-year project/Flask/static/music/1.wav')

return render\_template('audio.html') return render\_template('Model3.html')

@app.route("/Model4",methods=['GET','POST']) def model4():

if request.method == 'POST':

f=request.files['file'] f.save(os.path.join(path4,f.filename)) p=os.path.join(path4,f.filename)

cnn = keras.models.load\_model('model2-dog.h5') import numpy as np

from keras.preprocessing import image

test\_image = image.load\_img('D:/final-year project/Flask/static/model4/dog.jpeg', target\_size = (64, 64))

test\_image = image.img\_to\_array(test\_image) test\_image = np.expand\_dims(test\_image, axis = 0) result = cnn.predict(test\_image)

if result[0][0] == 1:

prediction = 'No-dog'

tts = gTTS("There is No dog in-front of you") tts.save('D:/final-year project/Flask/static/music/1.wav')

else:

prediction = 'dog'

tts = gTTS("There is dog in-front of you") tts.save('D:/final-year project/Flask/static/music/1.wav')

return render\_template('audio.html') return render\_template('Model4.html')

if name == ' main ':

app.run(debug=True)

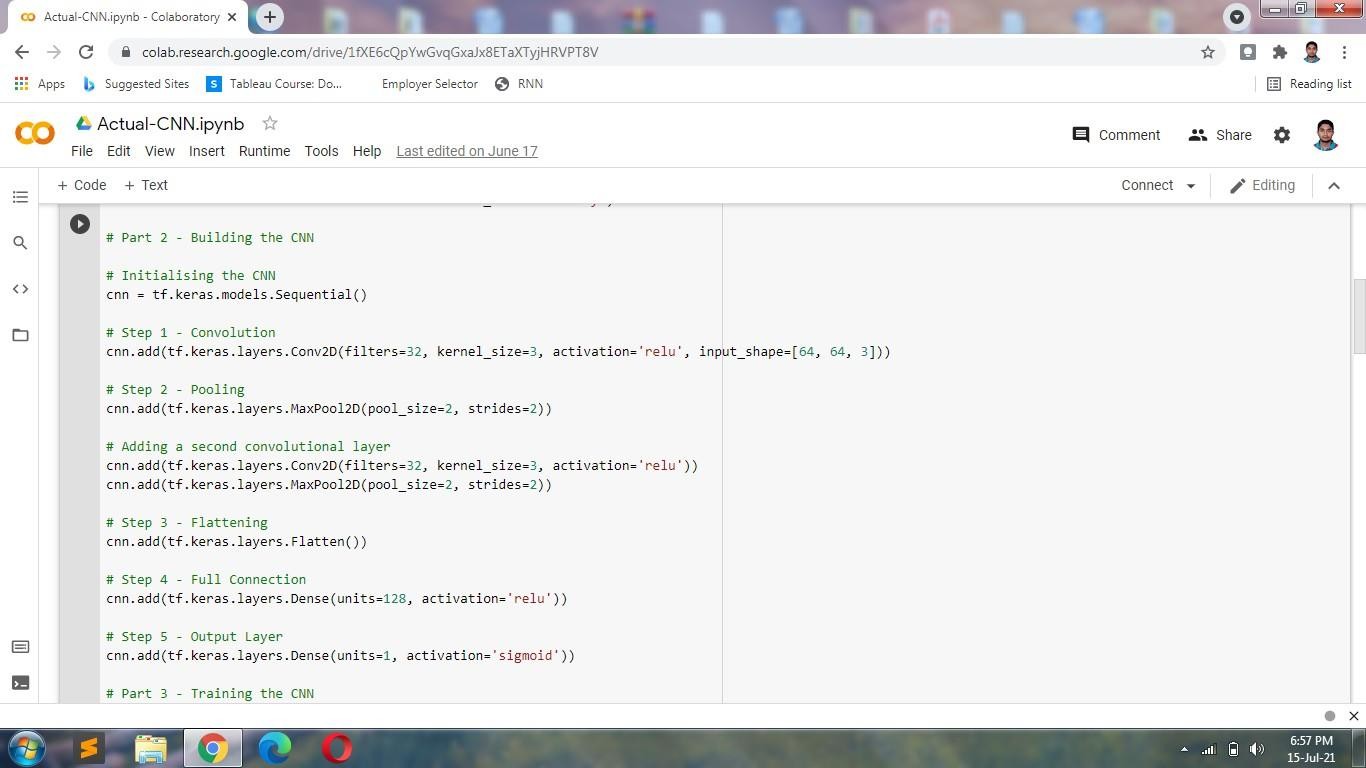


Figure 7.1: Implementation 1

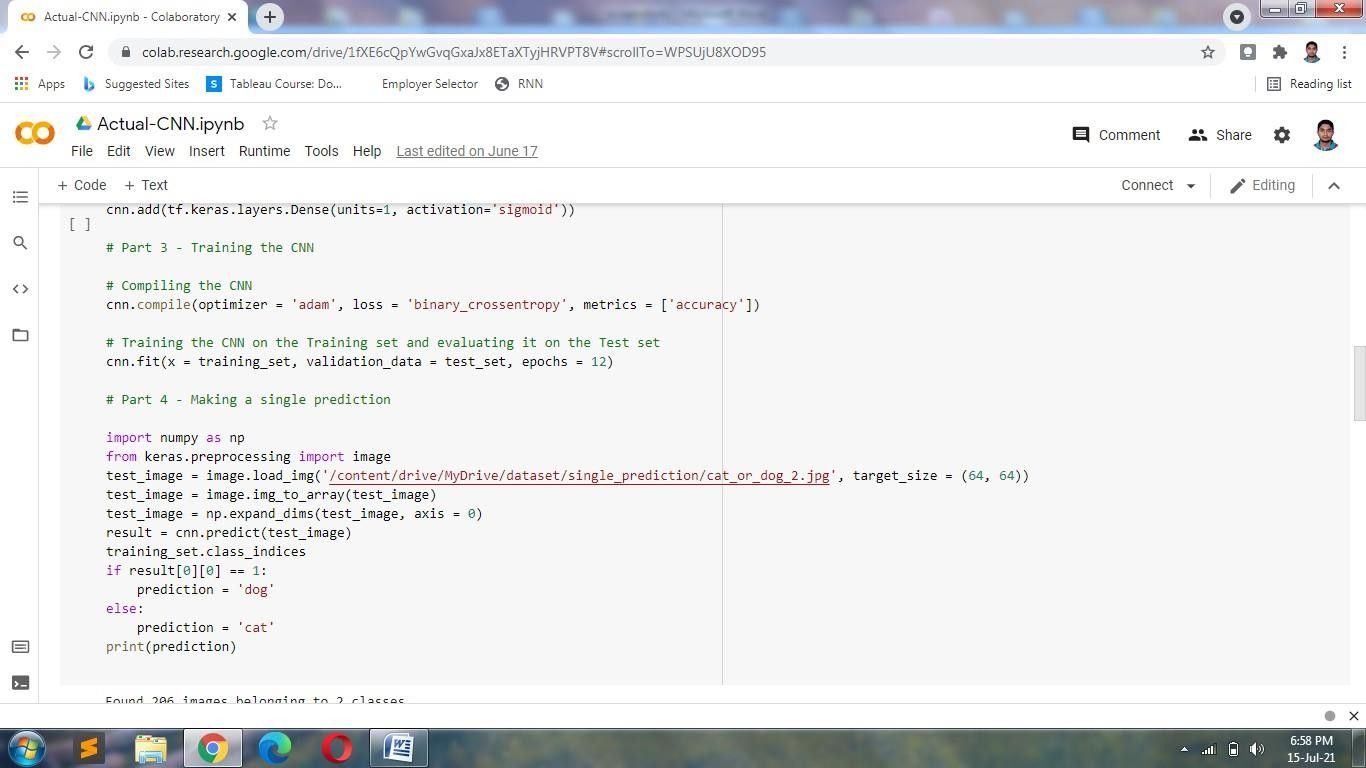


Figure 7.2: Implementation 2

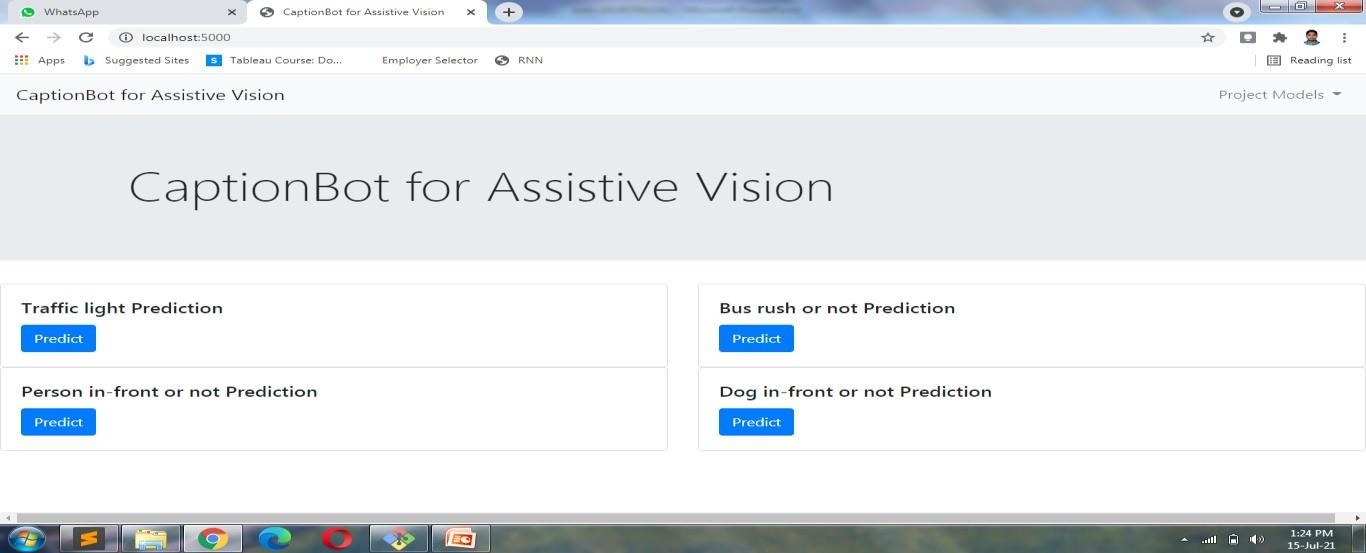


Figure 7.3: Web Interface

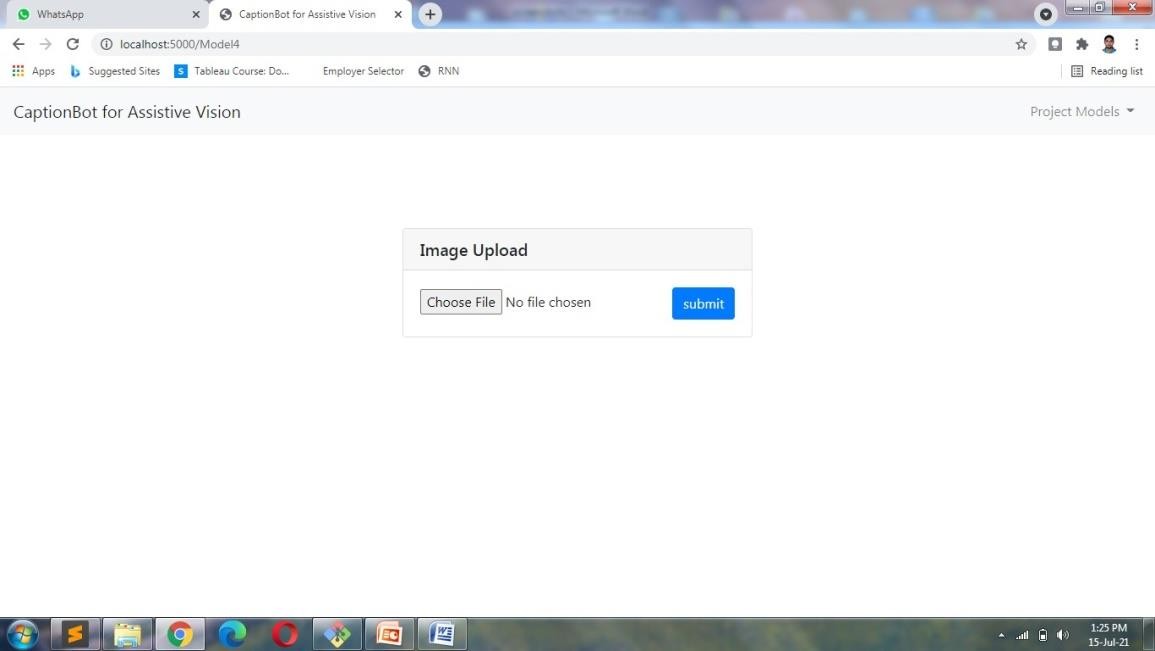


Figure 7.4: File for Detection

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Figure 7.5: Voice Output | | |

# REFERENCES

1. Abedin, Tingke Shen, Amlan Kar, Sanja Fidle. " Learning to Caption Images Through a Lifetime by Asking Questions." *2019 19th International Conference on Computer and Information Technology (ICCIT)*. IEEE, 2019.
2. Behrendt, [Hongje Seong](https://ieeexplore.ieee.org/author/37087015739); [Junhyuk Hyun](https://ieeexplore.ieee.org/author/37085631260); [Euntai Kim](https://ieeexplore.ieee.org/author/37277101900) "A deep learning approach to object: Detection, tracking, and classification." *2019 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2019.
3. Qiuyuan Huang ,Pengchuan Zhang Dapeng ,WuLei Zhang et al. " Turbo Learning for CaptionBot and DrawingBot *2020 International Conference on Control, Automation and Diagnosis (ICCAD)*. IEEE, 2020.
4. Jung, Seokwoo, et al. "Real-time Traffic Sign Recognition system with deep convolutional neural network." *2016 13th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*. IEEE, 2016.
5. Li, Xi, et al. "Traffic light recognition for complex scene with fusion detections." *IEEE Transactions on Intelligent Transportation Systems* 19.1 (2018): 199-208.
6. Hao Yang , Chunfeng Yuan ,*, Li Zhang, Yunda Sun, Weiming Hu*and Stephen J. Maybank " STA-CNN: Convolutional Spatial-Temporal Attention Learning for Action Recognition ." IEEE, 2020.
7. M Manoj krishna, M Neelima , M Harshali , M Venu Gopala Rao . " Image classification using Deep learning ." IEEE, 2020.
8. Priyanka, D., et al. "Traffic light and sign detection for autonomous land vehicle using Raspberry Pi." *2017 International Conference on Inventive Computing and Informatics (ICICI)*. IEEE, 2017.
9. Saini, Sanjay, et al. "An efficient vision-based traffic light detection and state recognition for autonomous vehicles." *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2017.
10. Shi, Zhenwei, Zhengxia Zou, and Changshui Zhang. "Real-time traffic light detection with adaptive background suppression filter." *IEEE Transactions on Intelligent Transportation Systems* 17.3 (2016): 690-700.
11. Tran, Tai Huu-Phuong, et al. "Real-time traffic light detection using color density." *2016 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia)*. IEEE, 2016.
12. Yudin, Dmitry, and Dmitry Slavioglo. "Usage of fully convolutional network with clustering for traffic light detection." *2018 7th Mediterranean Conference on Embedded Computing (MECO)*. IEEE, 2018.

## Signature of Guide