CERTIFICATE

CERTIFICATE

This is to certify that the project work entitled "**REAL TIME OBJECT DETECTION WITH VOICE OUTPUT USING OPENCV**" submitted to Bharathiar University in partial fulfilment of the requirements for the award of the Degree if Bachelor of Computer Science is a record of the original work done by **S. VIMALA (REG NO: 2122K0207)** under my supervision and the project has not formed the basis for the award of any Degree/Diploma/Associate/fellowship or similar title to any candidate of any university.

Internal Guide **Head of the Department**

(Dr. P.KOKILA, M.sc., M.Phil., Ph.D) **(Dr. R. PARIMALA, M.Sc., M.Phil., Ph.D.)**

Viva-Voce Examination is held on _____ L.R.G Government Arts College for Women, Tirupur - 641 604.

Internal Examiner **External Examiner**

DECLARATION

DECLARATION

I hereby declare that the project work submitted to the **Department of the Computer Science, L.R.G Government Arts College for Women, Tirupur** affiliated to Bharathiar University, Coimbatore in the partial fulfilment of the required for the award of **Bachelor of Computer Science** is an original work done by me during the sixth semester.

PLACE: TIRUPPUR **Signature of the candidate**

DATE: **(S.VIMALA)**

(REG NO: 2122K0207)

ACKNOWLEDGEMENT

ACKNOWLEDGEMENT

The world of thanks cannot compensable the facility that was provided for me, but still I would like to express my feelings of gratitude.

As first and foremost I would like to extend my thankfulness to the almightly for blessing the work of my hands, I am grateful to my parents for continued encouragement that thy had given me.

I would like to extend my profound gratitude and my sincere thanks to **Dr. M. YELIZHI, M.A., M.Phil., Ph.D.** Principal. L.R.G Government Arts College for Women, for the encouragement rendered to my during this project.

It's my privilege to thank **Dr. R. PARIMALA, M.Sc., M.Phil., Ph.D.** IN charge Head of the Department, Department of Computer Science for her valuable guidance and support throughout the project development work.

I express my deep sense of gratitude and sincere thanks to my **Dr. P. KOKILA, M.Sc., M.Phil., Ph.D**. guide Lecturer, Department of Computer Science, for providing all sorts of facilities to complete my project successfully.

I also extend my sincere thanks to all the other faculty member of the department, family and technical assistants for their co-operation and valuable guidance.

I thank our college library for providing me with many informative books that help me to enrich my knowledge to bring out the project successfully.

SYNOPSIS

SYNOPSIS

This project focuses on the development and implementation of a robust real-time object detection system employing the OpenCV library. Utilizing the MobileNet SSD architecture, a lightweight yet efficient Convolutional Neural Network, the system is capable of rapidly identifying and classifying objects within live video streams. A distinctive feature of this project is the integration of voice output to provide auditory cues corresponding to detected objects.

The methodology involves preprocessing video frames through OpenCV, applying the MobileNet SSD model for object detection, and subsequently generating synthesized voice output for each recognized object category. This fusion of computer vision and voice technology aims to enhance a accessibility for individuals with visual impairments, offering a novel means of interacting with the surrounding environment.

CONTENTS

CONTENTS

INTRODUCTION

1. INTRODUCTION

1.1 PROJECT DESCRIPTION

The main goal of this project is to detect and track various objects in a video stream. Object detection and tracking are two important tasks in computer vision that involve identifying and locating objects within an image or video stream, and then keeping track of the those objects as they move or change position. Object detection refers to the process of locating and identifying objects within an image or video frame, and is typically done using machine learning algorithms such as Convolutional Neural Networks (CNNs). These algorithms analyze the visual features of the image, such as edges, textures, and shapes, and use them to recognize specific objects or classes of objects. Objects tracking, on the other hand, involves keeping track of an object over time as it moves within a video stream. This can be challenging due to variations in lighting, changes in object appearance, occlusion by other objects, and other factors that can make it difficult to maintain a consistent tracking algorithm.

To address these challenges, tracking algorithms often use a combination of visual features, motion models, and machine learning techniques to accurately track objects and predict the future positions. Together, object detection and tracking are used in a wide range of applications, from autonomous vehicles, and surveillance systems to robotics and augmented reality. By enabling computers to detect and track objects in real time, these technologies are helping to transform the way we interact with the world around us. Object detection is well-known computer technology connected with computer vision and image processing.

Object detection is well-known computer technology connected with computer vision and image processing. With the advent of deep learning techniques, the accuracy for object detection has increased drastically. It focuses on detecting objects or its instances of a certain class(such as humans, flowers, animals) in digital images and videos .With the recent rapid development of information technology (IT), a lot of research has been carried out to solve inconveniences in everyday life, and as a result, various conveniences for people have been provided. Nevertheless, there are still many inconveniences for the visually impaired. The greatest inconveniences that a blind person feels in everyday life include finding information about objects and indoor mobility problems. They have difficulty recognizing simple objects, and it is not easy to distinguish objects that have similar forms. Previous studies included object analysis using ultrasonic sensors.

In this project, we analyze accurate object information and obtain a location using a deep learning object recognition technique. In addition, voice recognition and voice guidance technologies are synthesized so the visually impaired can know the location of the objects they want to find by speaking to the system. Object recognition algorithms are designed based on the Single Shot MultiBox Detector (SSD) structure, an object recognition deep learning model, to detect objects using a camera. In addition, voice recognition technology designed to use speech-to-text (STT) technology converts a user's vocal commands into text, from which only specific words are extracted and retrieved by the system. In the voice guidance technology, the technique of synthesizing the position of the article so it can be output, and synthesizing the name of the article, is done by using text-to-speech (TTS). In this paper, we propose an efficient object detection system to help find objects in a certain space without help from others, with special consideration for the blind.

1.2 MODULES

- ❖ OBJECT DETECTION MODEL
- ❖ OpenCV INTEGRATION
- ❖ OBJECT DETECTION PIPELINE
- ❖ TEXT-TO-SPEECH (TTS) MODULE
- ❖ OUTPUT VISUALIZATION
- ❖ VOICE OUTPUT
- ❖ INTEGRATION AND CONTROL

1.2.1 DESCRIPTION OF MODULE

OBJECT DETECTION MODEL:

- Choose a pre-trained object detection model like YOLO (You Only Look Once), SSD (Single Shot MultiBox Detector), or Faster R-CNN.
- These models can detect objects in images or video frames.

OpenCV INTEGRATION:

- Use OpenCV to capture video frames from a camera or a video file. OpenCV provides functions for image processing and manipulation, which are crucial for real-time applications.

OBJECT DETECTION PIPELINE:

- Implement a pipeline to process each frame using the object detection model. This involves detecting objects, extracting their bounding boxes, and potentially classifying them.

TEXT-TO-SPEECH (TTS) MODULE:

- Integrate a TTS engine to convert the detected objects into spoken words. Libraries like pyttsx3 or gTTS (Google Text-to-Speech) can be used for this purpose.

OUTPUT VISUALIZATION:

- Overlay the detected objects and their labels onto the video frames. You can use OpenCV's drawing functions for this step.

VOICE OUTPUT:

- Utilize the TTS module to generate audio output corresponding to the detected objects. This output can be played in real-time as each object is detected.

INTEGRATION AND CONTROL:

- Finally, integrate all the components together to create a cohesive system. This involves coordinating the object detection pipeline with the TTS module and managing the real-time processing of video frames.

- By combining these modules, you can create a system that performs real-time object detection and provides voice output describing the detected objects.

SYSTEM SPECIFICATION

2. REQUIREMENTS SPECIFICATION

2.1 Hardware System Requirements:

Processor : Intel Dual Core 2.5 Ghz

RAM : 4 GB RAM

Monitor : 15’ inch LED Color

Hard disk : 500GB

Keyboard : Multimedia Keyboard

Mouse : Optical mouse

2.2 Software Requirements:

Operating system : Windows 11

Tool : Visual studio code

Front-End : Python

Back-End : Machine Learning

2.3 SOFTWARE DESCRIPTION

FRONT-END: PYTHON

PYTHON

Python is a high-level, interpreted and general-purpose dynamic programming language that focuses on code readability. The syntax in Python helps the programmers to do coding in fewer steps as compared to Java or C. Python is widely used in bigger organizations because of its multiple programming paradigms. They usually involve imperative and object-oriented functional programming. It has a comprehensive and large standard library that has automatic memory management and dynamic features. The software has automatic memory management and dynamic features. The software development companies prefer Python language because of its versatile features and fewer programming codes. Nearly 14% of the programmers use it on the operating systems like UNIX, Linux, Windows and Mac OS and also interactive.

Advantages or Benefits of Python

The python language had diversified application in the software development companies such as in gaming, web frameworks and applications, language development, prototyping, graphic design applications, etc. This provides the language a higher plethora over other programming languages used in the industry.

Some of the advantages are:

- **Extensive Support Libraries**

It provides large standard libraries that include the areas like string operations, Internet, web service tools, operating system interfaces and protocols. Most of the highly used programming tasks are already scripted into it the limits the length of the codes to be written in Python.

- **Presence of Third Party Modules:**

The Python Package Index (PyPI) contains numerous third-party modules that make python capable of interacting with most of the other languages and platforms.

- **Integration Feature**

Python integrates the Enterprise Application Integration that makes it easy to develop Web services by invoking COM or COBRA components. It has powerful control capabilities as it calls directly through C, C++ or Java via Python. Python also processes XML and other markup languages as it can run on all modern operating systems through same byte code.

- **Productivity**

With its strong process integration features, unit testing framework and enhanced control capabilities contribute towards the increased speed for most applications and productivity of applications. It is a great option for building scalable multi-protocol network applications.

- **Open Source and Community Development**

Python language is developed under an OSU-approved open source license, which it free to use and distribute, including for commercial purposes. Further, its development is driven by the commercial collaborates for its code through hosting conferences and mailing lists, and provides for its numerous modules.

- **Learning Ease and support Available**

Python offers excellent readability and uncluttered simple-to-learn syntax which helps beginners to utilize this programming language. The code style guidelines, PEP 8, provide a set of rules to facilitate the formatting of code. Additionally, the wide base of users and active developers has resulted in a rich internet resource bank to encourage development and the continued adoption of the language.

- **User-friendly Data Structures**

Python had built-in list and dictionary data structures which can be used to construct fast runtime data structures. Further, python also provides the option of dynamic high-level data typing which reduces the length of support code that is needed.

Simple and easy to learn

- The syntaxes of python language are very simple. Anybody can remember the python language syntaxes rules and regulations very easily.
- The elegant syntaxes of the python language makes the people to learn python in easiest manner.
- With out having any other programming language knowledge, we can learn python directly.
- The simple and powerful syntax of python language makes the programmers to express their business logic is less of code.

Platform independent

- Like java programs, python programs are also independent.
- Once we write a python program, it can run on any platform without rewriting once again.
- Python uses PVM to convert python code to machine understandable code.

High level language

- While developing python applications, developers no need to bother about memory management.
- Dynamically typed language.
- Python is a dynamically typed language. I mean, in python there is no need to declare type of a variable.
- Whenever we assign a value to the variable, based on the value the type will be allocated automatically.

Python supports POP&OOP

- Python language supports both procedural oriented programming and object oriented programming features.
- We can implement OOPs features like Encapsulation, Polymorphic, Inheritance and abstraction in python programming language.
- Python is interpreted Language
- Like PHP, Python is also interpreted language.
- Python applications do not require explicit compilation so that compiler is not require for python software.
- Python interpreter is responsible for execution of python applications.
- When ever we run a python application, the python interpreter checks for the syntax errors. If there is no syntax errors found in our code then the interpreter converts the code into intermediate code in the form of low level format and executes it.
- The intermediate code of the python applications is known as byte code.
- The extension of the byte code file is .pye (Python Compile Code).

Python is Embeddable

- We can embed the python code into other language such as C,C++,Java and Etc.,
- In order to provide the scripting capabilities to other languages we can use the python code in it.

Develop GUI& Web Application

- We can develop the GUI based applications using Python Languages.
- We can also develop the Web applications using Python Languages.

Applications of Python

- GUI base desktop applications
- Image processing and graphic design applications
- Scientific and computational applications
- Games
- Web frameworks and web applications
- Enterprise and business applications
- Operating systems
- Language development
- Prototyping

BACK-END: MACHINE LEARNING

MACHINE LEARNING

Machine learning is one of the applications of Artificial Intelligence (AI) which enables the computers to learn on their own and perform tasks without human intervention. There are numerous applications of machine learning algorithms in the field of computer vision. With the help of machine learning, formulation of some of which were previously programmed by humans, sometimes by-hand, are now being programmed without any human contribution with the help of machine learning. In the recent years, due to remarkable increase in the availability of humongous sources of data and feasibility of computational resources, machine learning has become predominant with wide range of applications in our daily lives.

TYPES

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

Supervised Learning

Supervised learning is considered to be the most elementary class of machine learning algorithms. As the name suggests, these algorithms require direct supervision. In this type of learning, the data labeled by humans is spoon-fed to the algorithm. This data contains the classes and locations of the objects of interest. Eventually, the algorithms learns from the annotated data and predicts the annotations of the new data previously not known to the algorithm, after the completion of training process.

Some of the popularly utilized supervised algorithms are:

- Neural Networks
- Decision Trees
- Random Forest
- K-Nearest Neighbors
- Linear Regression
- Logistic Regression
- Support Vector Machine

Unsupervised Learning

In the unsupervised learning, the algorithm tries to learn and identify useful properties of the classes from the given annotated data, without the help or intervention of a human. Apriori algorithms, K-means clustering, etc., are some of the common unsupervised learning algorithms.

Reinforcement Learning

In this type of learning, the machine is allowed to train itself continually using trial and error. As a result, the machine learns from past experience and attempts to capture the best knowledge possible to predict accurately. Markov decision process, Q-learning, Temporal difference, etc. are some of the examples of reinforcement learning.

Importance of machine Learning

- It allows people to do tasks more quickly and effectively.
- It helps evaluating large chunks of data, easing the tasks of data scientists in an automated process.
- It allows the machine to modify are change the tasks otherwise has to be performed by a human, example changing a password or checking an account balance.
- It makes the machine to apply mathematical calculations automatically.
- It supports data extraction and interpretation of the acquired data sets.

Advantages of Machine Learning

- Easily identifies the Trends and Patters: Machine learning can track the large amount of data and finds the trends and the patterns that would not be done by humans.
- Continuous Improvement: They helps you to improving in accuracy and the efficiency that helps you to make faster and better decisions.
- Handling multi-dimensional and multi-variety data: Machine learning can handle multiple tasks and larger data sets easily in non-suitable and conditionals environments.
- Social Advertisements: Social networking sites like Facebook, Twitter, Hike can promote much needed user history.
- Allows real-world implementations: Health, Banking, Financial sectors contributes a major amount of data to be further used.

SYSTEM ANALYSIS

3. SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

Object detection is a common computer vision technique that locates and identifies specific types of objects in an image. Those types of objects can be anything from humans, animals, or vehicles to very specific items such as brands of chips or types of building materials. In some existing systems some objects are generally not detected if they're small. Objects are generally not detected if they're arranged closely together.

Here's an overview of the existing system:

- Command Line Interface: users can execute the script using command-line arguments specifying the prototxt file, pre-trained model, and minimum detection probability.
- Object detection: detected objects are filtered based on a specified minimum probability threshold. Bounding boxes are drawn around the detected objects on the frames, and their class labels and confidence scores are displayed.

DRAWBACKS

- One of the fundamental challenges in object detection is the trade-off between and accuracy.
- Objects are generally not detected if they're small.
- Objects are generally not detected if they're arranged closely together.

3.2 PROPOSED SYSTEM

- To overcome some of the pitfalls in existing system, this proposed system has been developed after careful study, as well as compared with existing system Demerits in the existing system analyzed and checked carefully, and then we found solutions for those problems.
- It empowers you to locate, identify, track, and determine the objects in the particular image or scene. The algorithm labels them and provides a robust solution of APIs to serve you specific tagging, categorization, and much more.
- Object detection also plays a vital role in content-based image retrieval, image segmentation, and video analysis, providing valuable insights and facilitating efficient data management.

FEATURES

- Object detection and instance segmentation using machine learning provides a fast and accurate means to predict the location of an object in an image or scene.
- Break down the code into modular functions or classis to improve readability and maintainability. This would make it easier to understand the extent the functionality.
- Allow users to customize the audio feedback, such as choosing different voices or adjusting the speech rate.
- Include comprehensive documentation, including a README file, to guide users on how to use the script, its requirements and potential configurations.

3.3 OBJECTIVES

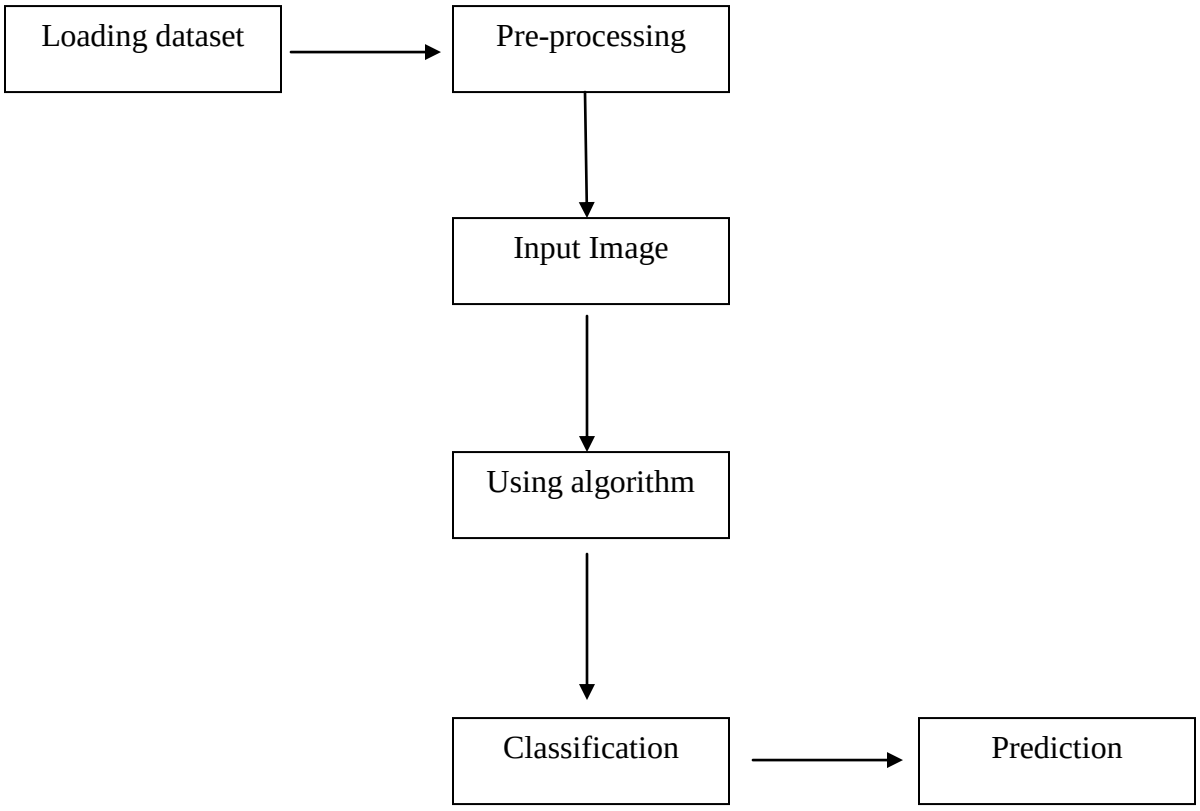
- The prime objective of this projects is to underlying factors of visual impairment and to help people suffering from visual impairment know what they are surrounded by in a very friendly cost.
- Adding on to that, the project also aims at running seamlessly without any system barrier and high usability. The accuracy of the built model is one to look out for, since it shall be dealing with detecting objects and recognizing them in real time.

SYSTEM DESIGN AND DEVELOPMENT

SYSTEM DESIGN AND DEVELOPMENT

4.1 SYSTEM DESIGN

System design is the process of defining the architecture, components, modules, interfaces and data for a system to satisfy specified requirements. One could see it as the application of systems theory to product development. There is some overlap with the disciplines of systems analysis, systems architecture and systems engineering. If the broader topic of product development "blends the perspective of marketing, design and manufacturing into a single approach to product development" then the design is the act of taking the marketing information and creating the design of the product to be manufactured. Systems design is therefore the process of defining and developing systems to satisfy requirements of the user.



4.2 INPUT DESIGN

In the real-time object detection with voice output using OpenCV project, the user interface is carefully designed to provide a seamless and intuitive experience. The input design features a live video feed captured by the camera, with a responsive overlay displaying detected objects in real-time. Users can interact with the system through a clean and minimalistic control panel, allowing them to customize detection parameters and toggle voice output.

The input interface includes options for selecting specific object classes to focus on, adjusting detection sensitivity, and controlling the voice output preferences. Users can effortlessly start and stop the detection process with dedicated buttons, ensuring ease of use. The design prioritizes clarity and user-friendly interactions, making it accessible to both novices and experienced users.

Additionally, informative status indicators convey the system's performance, such as the frame rate, object count, and voice output activation. This thoughtful input design aims to enhance the overall user experience, combining the power of real-time object detection with the convenience of voice feedback through a well-crafted and intuitive interface.

4.3 OUTPUT DESIGN

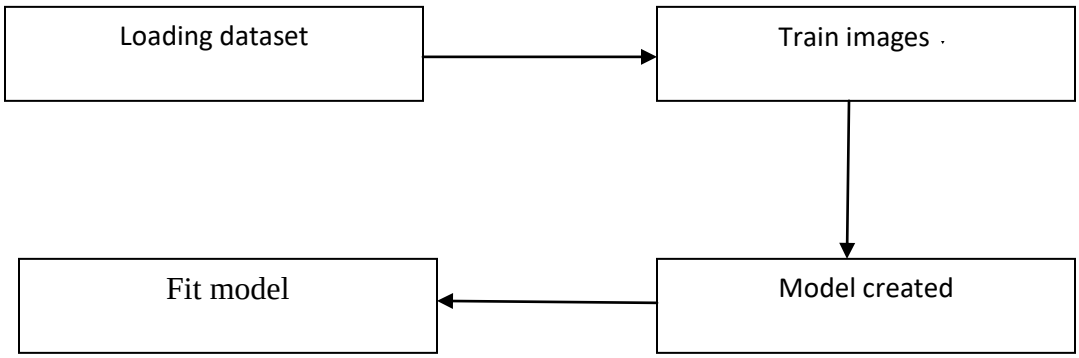
The real-time object detection with voice output project incorporates advanced computer vision techniques using OpenCV. In this innovative system, a webcam captures the live video feed, and OpenCV's object detection algorithms analyze each frame in real-time, identifying and tracking objects within the scene.

Once an object is detected, a synthesized voice output dynamically announces the type and location of the recognized object. The seamless integration of OpenCV's robust image processing capabilities with voice synthesis technology enhances accessibility and usability, making this project valuable for applications such as assistive technology, surveillance, and interactive environments.

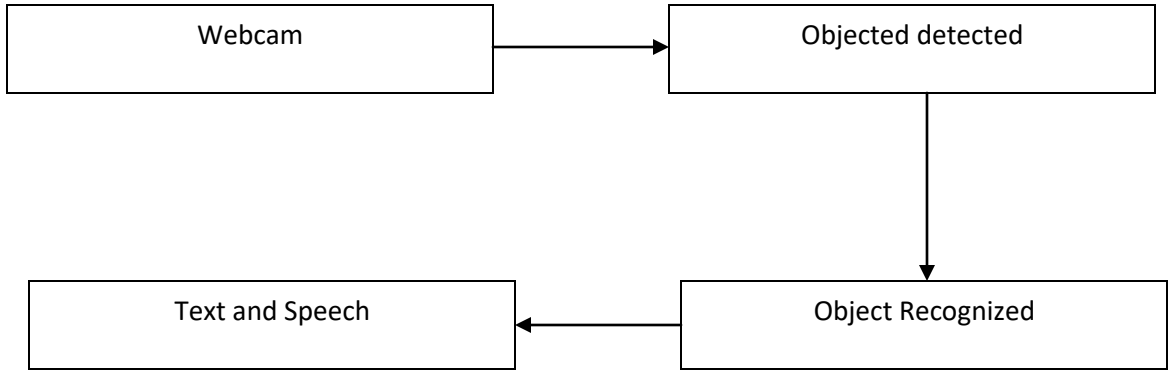
This design ensures a responsive and user-friendly experience by providing both visual and auditory feedback in real-time.

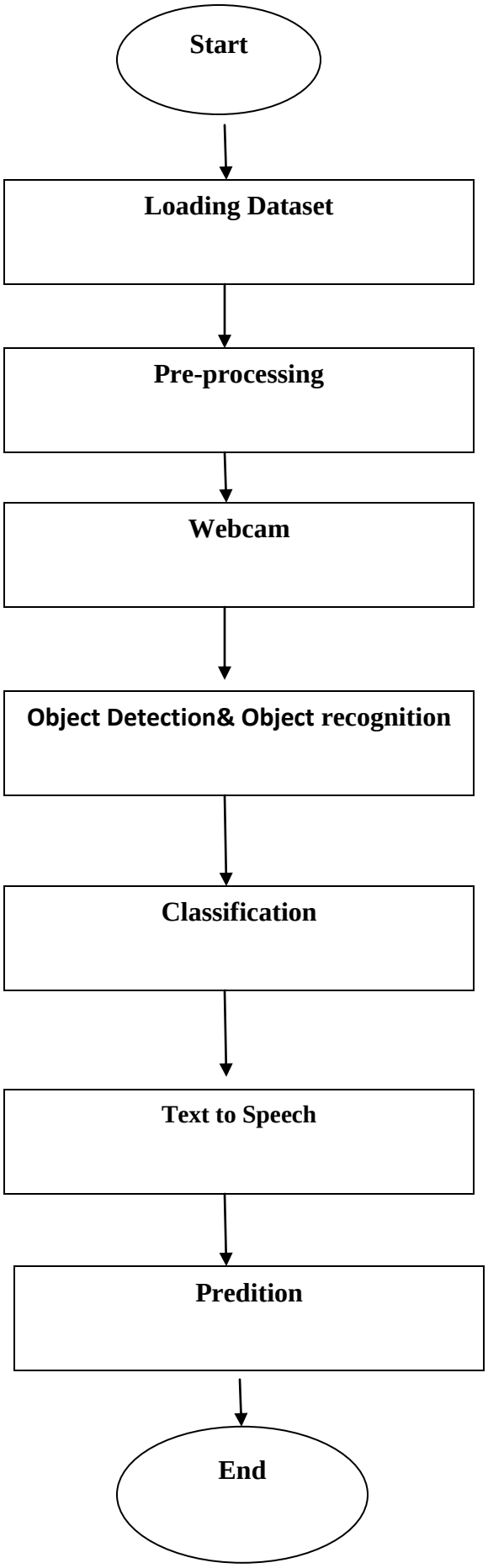
4.4 DATA FLOW DIAGRAM

Level 1



Level 2




Start
Loading Dataset
Pre-processing
Webcam
Object Detection& Object recognition
Classification
Text to Speech
Predition
End

4.5 DATABSE DESIGN

The script database implements real-time object detection using the MobileNet SSD model. It leverages the OpenCV library to process video frames, identify objects, and display the results.

Dependencies

- **OpenCV:** Utilized for image processing and object detection.
- **imutils:** A collection of convenience functions for OpenCV, used for resizing the frames.
- **Pyttsx3:** A text-to-speech library for python, employed to announce detected objects.

Command-Line Usage

The script can be executed from the command line using the following commands:

bash

workon cv

pytho3 real_time_object_detection.py—prototxt MobileNetSSD _ deploy.prototxt.txt—model MobileNetSSD_deploy.caffemodel

Arguments

- -p, --prototxt: Path to the Caffe 'deploy‘ prototxt file.
- -m, --model: Path to the Caffe pre-trained model.
- -c, --probability: Minimum probability to filter weak detections .

Object Classes

The script is configured to detect various objects within some classes, including aeroplane, bicycle, car, person, etc.

Color Coding

Different colors are assigned to bounding boxes based on the detected class, providing visual distinction for each object.

User Interaction

- Pressing 'q' breaks out of the main loop, terminating the script.
- Real-time updates are provided on the elapsed time and approximate frames per second (FPS) during execution.

Main Loop

- Initializes the video stream from the camera.
- Resizes each frame to a maximum width of 500 pixels for efficient processing..
- Filters out weak detections based on the specified probability threshold.
- Draws bounding boxes and labels on the frame for detected objects.
- Utilizes text-to-speech to announce the detected object's class.
- Displays the processed frame with bounding boxes and labels.

Cleanup

- Stops the video stream and releases associated resources.
- Displays elapsed time and FPS information.

Conclusion

The script showcases a practical implementation of real-time object detection, providing a foundation for further customization and integration into computer vision applications.

SYSTEM TESTING AND IMPLEMENTATION

SYSTEM TESTING AND IMPLEMENTATION

5.1 LEVEL OF TESTING

Testing can be done in different levels of SDLC. They are

5.1.1 UNIT TESTING

The first level of testing is called unit testing. Unit testing verifies on the smallest unit of software design-the module. The unit test is always white box oriented. In this, different modules are tested against the specification produced during design for the modules. Unit testing is essentially for verification of the code produced during the coding phase, and goal is to test the internal logic of the modules.

It is typically done by the programmer of the programmer of the module. Due to its close association with coding, the coding phase is frequently called "coding and testing". The unit test can be conducted in parallel for multiple modules.

5.1.2 Integration Testing

The second level of testing is called integration testing. Integration testing is a systematic technique for constructing the program structure while conducting tests to uncover errors associated with interfacing. In this, many tested modules are combined into subsystems, which are then tested. The goal here is to see if all the modules can be integrated properly.

There are three types of integration testing:

1. Top-Down Integration

Top down integration is an incremental approach to construction of program structures. Modules are integrated by moving downwards throw the control hierarchy beginning with the main control module.

2. Bottom-Up Integration

Bottom up integration as its name implies. Begins construction and testing with automatic modules.

3. Regression Testing

In this contest of an integration test strategy, regression testing is the re execution of some subset of test that have already been conducted to ensure that changes have not propagated unintended side effects.

5.2 IMPLEMENTATION

System implementation is the important state of the project when the theoretical design is turned into practical system.

5.2.1 IMPLEMENTATION PROCEDURES

The project is implemented in different phases as follows

- The first stage of the project entails installing the software needed to install the deep learning algorithms that will be used to recognize objects.
- The project's CNN must be loaded during the second phase.
- The third phase involves attach a OpenCV and active the camera and begin recording a video for detecting purpose
- Detecting the objects and placing them in a box shape are included in the fourth phase.
- In the fifth phase voice were added for the detecting objects.

1. **First Phase**

   Installing the software needed to install the deep learning algorithms that will be used to recognize objects

2. **Second Phase**

   Loading the CNN to the project

CNN (convolutional neural network)

Convolutional Neural Networks are very similar to ordinary Neural Networks from the previous chapter they are made up of neurons that have learnable weights and biases. Each neuron receives some inputs, performs a dot product and optionally follows it with a non-linearity.

Convolutional Neural Networks (CNNs) are analogous to traditional ANNs in that they are comprised of neurons that self-optimise through learning. Each neuron will still receive an input and perform a operation (such as a scalar product followed by a non-linear function)- the basis of countless ANNs. From the input raw image vectors to the final output of the class score, the entire of the network will still express a single perceptive score function. The last layer will contain loss functions associated with the classes, and all of the regular tips and tricks developed for traditional ANNs still apply.

The only notable difference between CNNs and traditional ANNs is that CNNs are primarily used in the field of pattern recognition within images. This allows us to encode image-specific features into the architecture, making the network more suited for image-focused tasks – whilst further reducing the parameters required to set up the model. One of the largest limitations of traditional forms of ANN is that they tend to struggle with the computational complexity required to compute image data.

Mobile Net Architecture

MobileNet SSD (Single Shot Multibox Detector) is a groundbreaking architecture designed for real-time object detection on resource-constrained devices, such as mobile phones and embedded systems. Introduced as an extension of MobileNet, a lightweight convolutional neural network (CNN) designed for mobile applications, MobileNet SSD strikes a balance between accuracy and efficiency.

The architecture employs a series of depthwise saparable convolutions to reuce the number of parameters and computations while preserving valuable features. This allows for faster inference without compronisin on the model's ability to detect and localize objects in images. The SSd component enhances the model's versatility by utilizing multiple convolutional layers of different scales to detect objects of various sizes. This feature rich bozes, enabling it to handle diverse object shapes and scales. The contributing to the model's robust performance. Mobilenet SSd's architecture embodies the essence of efficient and effective object detection on mobile devices, making it a pivotal development in the field of computer vision.

These convolutions separate the spatial and channel-wise operations, significantly reducing the number of parameters and computations. As a result, MobileNet SSD achieves impressive inference speeds, making it well-suited for applications that demand low latency object detection, such as augmented reality and autonomous systems.

One of the key features of MobileNet SSD is its incorporation of the Single Shot Multibox Detector (SSD) framework. This framework enables the simultaneous prediction of object classes and bounding box coordinates at multiple scales within a single forward pass of the network. By utilizing feature maps from different layers, SSD is capable of detecting objects of varying sizes. This multi scale approach enhances the models's adaptability to diverse scenarios, from small objects to larger, more prominent ones.

Additionally, MobileNet SSd employes anchor boxes, or default boxes, to further refine object localization. These anchor boxes, with different aspect ratios, act as templates for predicting bounding box coordinates, contributing to the model's precision in object localization.

In conclusion, MobileNet SSD stands as a testament to the advancements in the field of computer vision, particularly in the domain of real-time object detection on resource constrained devices. The architecture's fusion of MobileNet's efficiency with the SSD framewor's multi scale detection capabilities has paved the way for a new era in mobile based computer vision applications. As technology continues to evolve, MobileNet SSD serves as a cornerstone in the development of a efficient and accurate object detection models, enabling a myriad of applications across industries.

3. Third phase

In third Phase involves attaching a camera to opencv and utilizing to activate the camera and begin recording a video for detecting purposes.

4. Fourth Phase

Detecting the objects and placing them in a box shape are included in the fourth Phase

Object Detection

Object detection finds and identifies things in images, and it's one of the biggest accomplishments of deep learning and image processing. One of the common approaches to creating localizations for objects is with the help of bounding boxes. You can train an object detection model to identify and detect more than one specific object, so it's versatile.

The models are usually trained to detect the presence of specific objects. The constructed models can be used in images, videos, or real-time operations. Even before the deep learning methodologies and modern-day image processing technologies, object detection had a high scope of interest. Certain methods had success with object detection, and there were relatively few other competitors in this field.

Object detection makes use of the special and unique properties of each class to identify the required object. While looking for square shapes, the object detection model can look for perpendicular corners that will result in the shape of a square, with each side having the same length. While looking for a circular object, the object detection model will look for central points from which the creation of the particular round entity is possible.

5. Fifth Phase

In this phase we add voice to object detection. Text-to-speech method were added in this phase

Text-to-speech

Text-to-speech is a type of speech synthesis application that is used to create a spoken sound version of the text in a computer document, such as a help file or a web page. TTs can enable the reading of computer display information for the visually challenged person, or may simply be used to argument the reading of a text message. Current TTS applications include voice enaled e-mail and spoken prompts in voice response systems.

TTS is often used with voice recognition programs. Like other modules the process has got its own relevance on being interfaced with, where finds its own operations based on image processing schemes. So once image gets converted to text and thereby it could be converted from text to speech. Character recognition process ends with the conversion of text to speech and it could be applied at anywhere.

Algorithms based on classification they work in two stages. In the first step, we're selection from the image interesting regions. Then we're classifying those regions using convolutional neural networks. This solution could be very slow because we have to run prediction for every selected region. One of the most important algorithms are the convolutional neural network and the another faster-RCNN.

CONCLUSION

6. CONCLUSION

An accurate and efficient object detection system has been developed which achieves comparable metrics with the existing state of the art system. This project uses recent techniques in the field of computer vision and deep learning. Custom dataset was created using labeling and the evaluation was consistent. This can be used in real time applications which require object detection for pre-processing in their pipeline.

An important scope would be to train the system on a video sequence for usage in tracking applications. Addition of a temporally consistent network would enable smooth detection and more optimal than per-frame detection.

FUTURE ENHANCEMENT

7. FUTURE ENHANCEMENT

Object detection is a key ability for most computer and robot vision system. Although great progress has been observed in the last years, and some existing techniques are now part of many consumer electronics or have been integrated in assistant driving technologies, we are still far from achieving human-level performance, in particular in terms of open-world learning. It shoud be noted that object detection has not been used much in many areas where it could be of great help. As mobile robots, and in general autonomous machine, are starting to be more widely deployed the need of object detection systems is gaining more importance. Finally, we need to consider that we will need object detection systems for nano-robots or for robots that will explore areas that have not been seen by humans, such as depth parts of the sea or other planets, and the detection systems will have to learn to new object classes as they are encountered. In such cases, a real-time open-world learning ability will be critical.

BIBLIOGRAPHY

8. BIBLIOGRAPHY

Reference Books

Websites

1. https://www.python.org/
2. https://www.learnpython.org/
3. https://www.chatgpt.org/
4. https://cloud.google.com/text-to-speech/

Books

1. Lutz, M. (2013). Learning Python, 5th Edition (5 edition). Beijing: O'Reilly Media.
2. S. Kanimozhi, G. Gayathri, and T. Mala, "Multiple Real-time object identification using single shot Multi-Box detection", 2019 Internal Conference on Computational Intelligence in Data Science (ICCIDS), Chennai, India, 2019.
3. Selman TOSUN, Enis KARAARSLAN "Real-Time Object Detection Application" IEEE Conferences 2018.
4. Ms. R. Vnitha and Ms. A. Theerthana. "Design And Development of Hand Gesture Recognition System For Speech Impaired People", 2018.
5. Shangeetha, R. K., V. Valliammai, and S. Padmavathi. "Computer vision-based approach for Indian Sign Language character recognition". Machine Vision and Image Processing (MVIP), 2018 International Conference on IEEE, 2018.
6. Geethapriya. S, N. Duraimurugan, S.P. Chokkalingam, "Real-time Object Detection", International Journal of Engineering and Advanced Technology (IJEAT).
7. Hammad Naeem, Jawad Ahmad and Muhammad Tayyab, "Real-time Object detection and Tracking", IEEE.
8. Dave, H., Gobse, A., Goel, A., & Bairagi, S. (2020). OCR Text Detector and Audio Convertor, int J. Res. Appl. Sci. Eng. Technol, 8, 991-999.

APPENDICES

SAMPLE CODING

To run the script, execute the following commands

workon cv

python3 real_time_object_detection.py --prototxt MobileNetSSD_deploy.prototxt.txt --model MobileNetSSD_deploy.caffemodel

```
# import the necessary packages

from imutils.video import VideoStream

from imutils.video import FPS

import numpy as np

import argparse

import imutils

import time

import cv2

import pyttsx3


# construct the argument parser and parse the arguments

ap = argparse.ArgumentParser()

ap.add_argument("-p", "--prototxt", required = True,  help = "path to Caffe 'deploy' prototxt file")
```

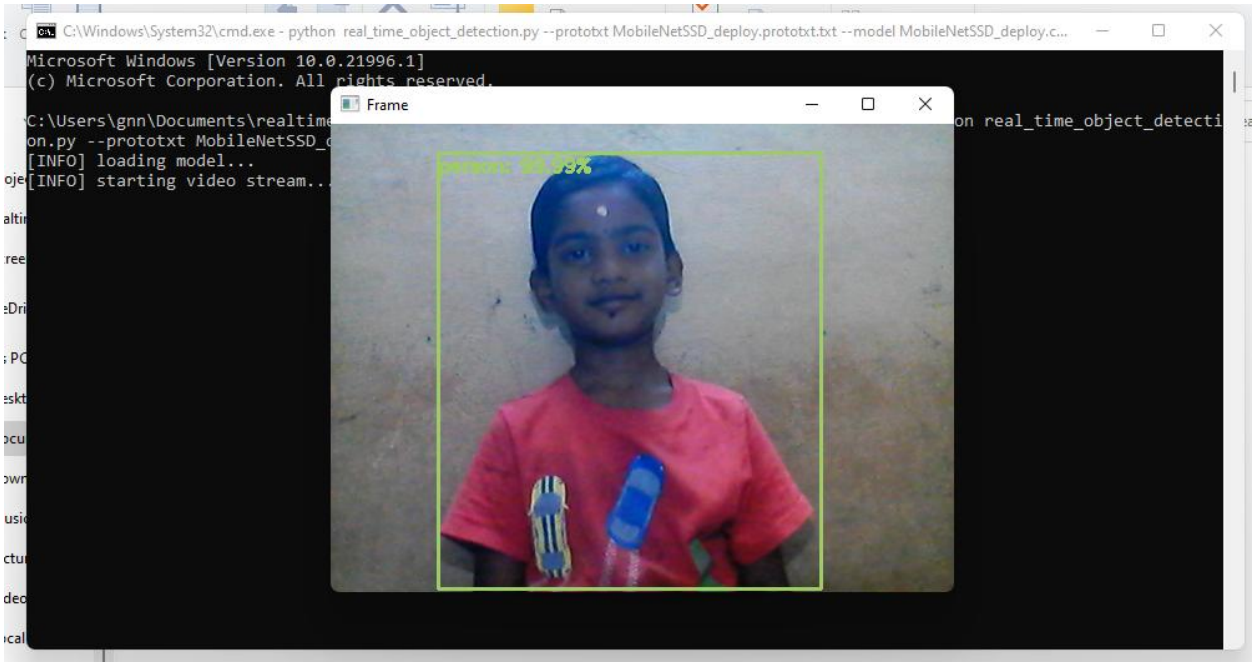
```
ap.add_argument("-m", "--model", required = True, help = "path to Caffe pre-trained model")

ap.add_argument("-c", "--probability", type = float, default = 0.2, help = "minimum probability to filter weak detections")

args = vars(ap.parse_args())


# initialize the list of class labels MobileNet SSD was trained to detect

# and generate a set of bounding box colors for each class

CLASSES = ["background", "aeroplane", "bicycle", "bird", "boat", "bottle", "bus", "car", "cat", "chair", "cow", "diningtable",

"dog", "horse", "motorbike", "person", "pottedplant", "sheep", "sofa", "train", "tvmonitor"]


COLORS = np.random.uniform(0, 255, size = (len(CLASSES), 3))


# load our serialized model from disk

print("[INFO] loading model...")

net = cv2.dnn.readNetFromCaffe(args["prototxt"], args["model"])


# initialize the video stream, allow the cammera sensor to warmup,

# and initialize the FPS counter

print("[INFO] starting video stream...")

vs = VideoStream(src=0).start()
```

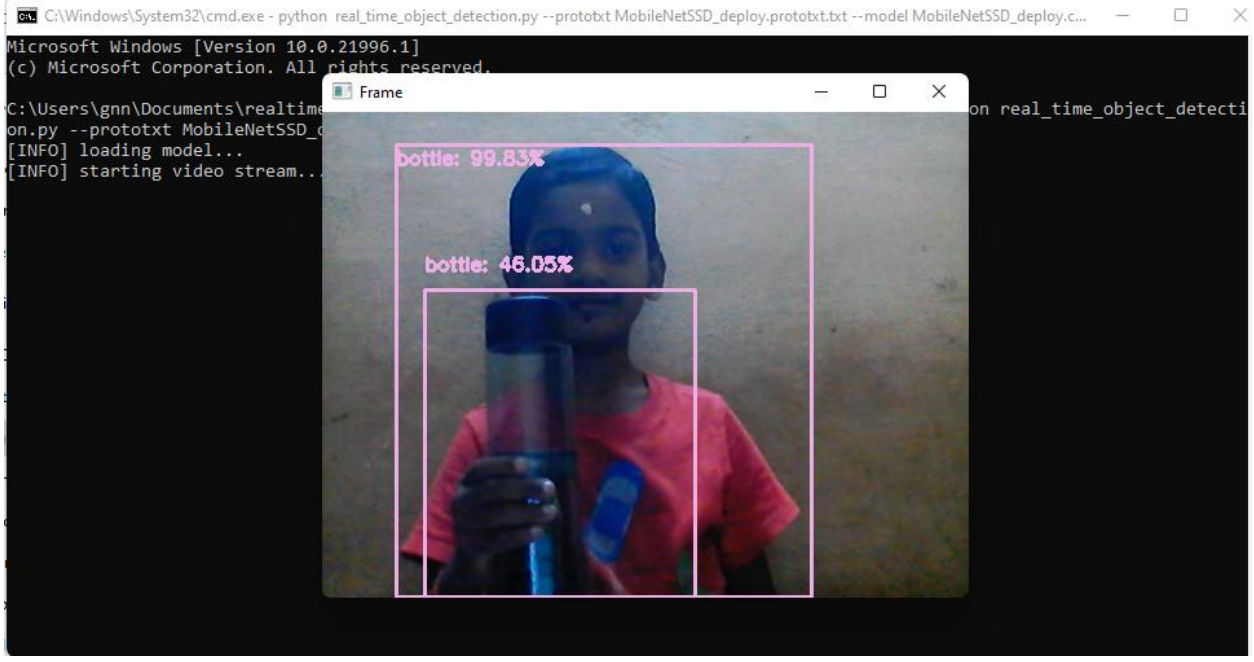
```
time.sleep(2.0)
fps = FPS().start()

# loop over the frames from the video stream
while True:
	# resize the video stream window at a maximum width of 500 pixels
	frame = vs.read()
	frame = imutils.resize(frame, width=500)

	# grab the frame dimensions and convert it to a blob
	# Binary Large Object = BLOB
	(h, w) = frame.shape[:2]
	blob = cv2.dnn.blobFromImage(cv2.resize(frame, (300, 300)), 0.007843, (300, 300), 127.5)

	# pass the blob through the network and get the detections
	net.setInput(blob)
	detections = net.forward()

	# loop over the detections
```

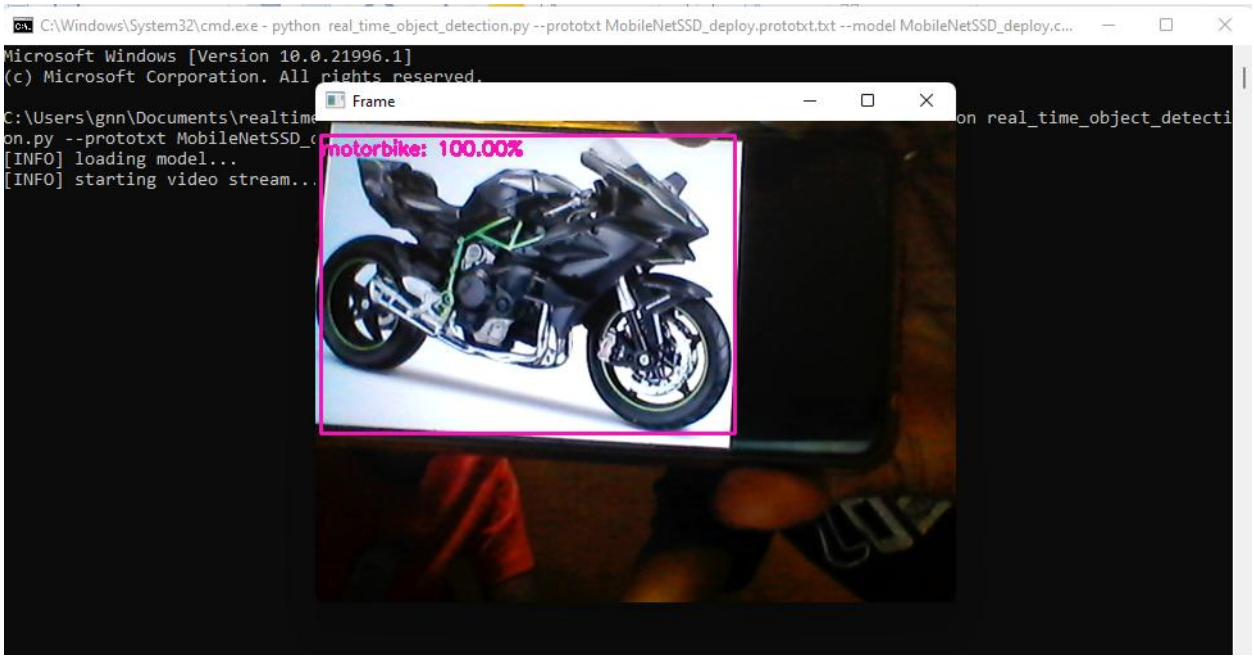
```
for i in np.arange(0, detections.shape[2]):
                # extract the probability of the prediction
                probability = detections[0, 0, i, 2]

                # filter out weak detections by ensuring that probability is
                # greater than the min probability
                if probability > args["probability"]:
                        # extract the index of the class label from the
                        # 'detections', then compute the (x, y)-coordinates of
                        # the bounding box for the object
                        idx = int(detections[0, 0, i, 1])
                        box = detections[0, 0, i, 3:7] * np.array([w, h, w, h])
                        (startX, startY, endX, endY) = box.astype("int")

                        # draw the prediction on the frame
                        label = "{}: {:.2f}%".format(CLASSES[idx], probability * 100)
                        cv2.rectangle(frame, (startX, startY), (endX, endY), COLORS[idx], 2)
                        y = startY - 15 if startY - 15 > 15 else startY + 15
                        cv2.putText(frame, label, (startX, y), cv2.FONT_HERSHEY_SIMPLEX,
0.5, COLORS[idx], 2)
```

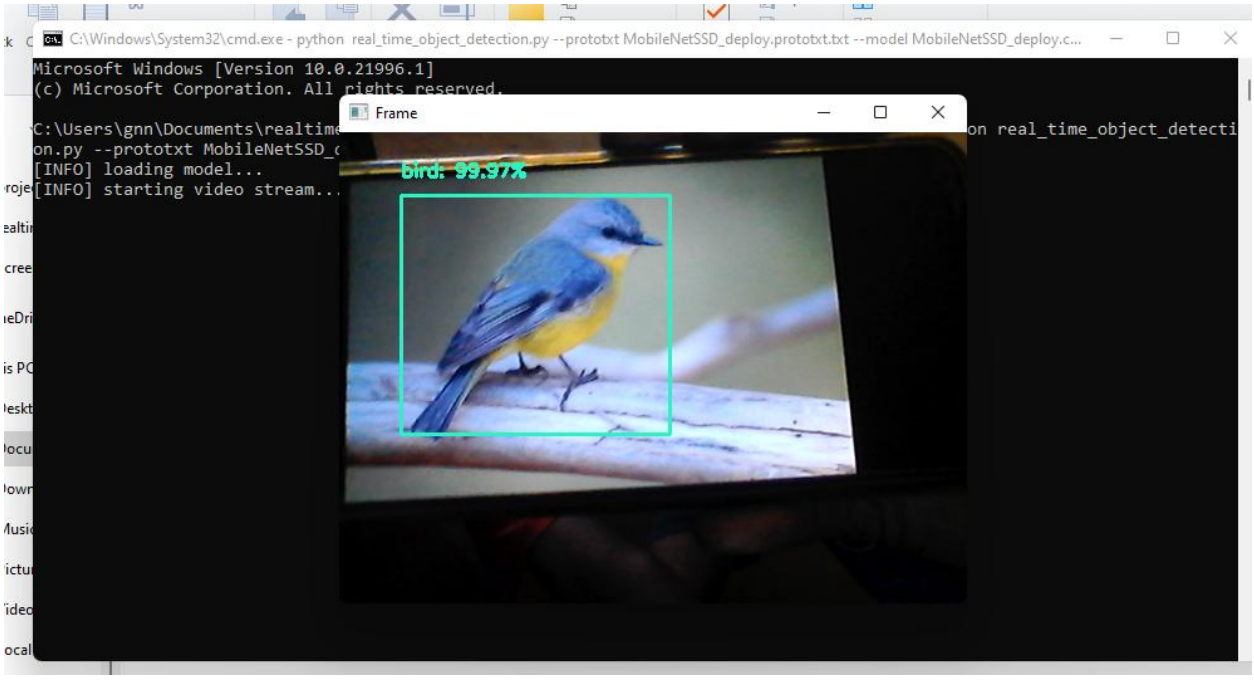
```
                    engine = pyttsx3.init()
                     engine.say(CLASSES[idx])
                     engine.runAndWait()

        # show the output frame
        cv2.imshow("Frame", frame)
        key = cv2.waitKey(1) & 0xFF

        # if the 'q' key was pressed, break from the loop
        if key == ord("q"):
                break
        # update the FPS counter
        fps.update()
# stop the timer and display FPS information
fps.stop()
print("[INFO] elapsed time: {:.2f}".format(fps.elapsed()))
print("[INFO] approx. FPS: {:.2f}".format(fps.fps()))
# cleanup
cv2.destroyAllWindows()
vs.stop()
```

SAMPLE OUTPUT






C:\Windows\System32\cmd.exe - python real_time_object_detection.py --prototxt MobileNetSSD_deploy.prototxt.txt --model MobileNetSSD_deploy.c...
Microsoft Windows [Version 10.0.21996.1]
(c) Microsoft Corporation. All rights reserved.
C:\Users\gnn\Documents\realtime
on.py --prototxt MobileNetSSD_
[INFO] loading model...
[INFO] starting video stream...
Frame
motorbike: 100.00%


C:\Windows\System32\cmd.exe - python real_time_object_detection.py --prototxt MobileNetSSD_deploy.prototxt.txt --model MobileNetSSD_deploy.c...
Microsoft Windows [Version 10.0.21996.1]
(c) Microsoft Corporation. All rights reserved.
C:\Users\gnn\Documents\realtime
on.py --prototxt MobileNetSSD_
[INFO] loading model...
[INFO] starting video stream...
Frame
bird: 99.97%


C:\Windows\System32\cmd.exe - python real_time_object_detection.py --prototxt MobileNetSSD_deploy.prototxt.txt --model MobileNetSSD_deploy.c...
Microsoft Windows [Version 10.0.21996.1]
(c) Microsoft Corporation. All rights reserved.
C:\Users\gnn\Documents\realtime
on real_time_object_detecti
on.py --prototxt MobileNetSSD_
[INFO] loading model...
[INFO] starting video stream...
Frame
aeroplane: 99.95%