Parshvanath Charitable Trust's
A. P. SHAH INSTITUTE OF TECHNOLOGY
(Approved by AICTE New Delhi & Govt. of Maharashtra, Affiliated to University of Mumbai)
(Religious Jain Minority)

Department of Computer Science & Engineering (AI & ML)

Object detection using TensorFlow

Aryan Kharat (23206001)
Soham Kalolikar (23206003)
Akshat Pujari (23206004)
Smith Patil (23206010)

Project Guide
Ms. Ranjita Asati

Outline

- Introduction
- Literature Survey of the Existing Systems
- Limitations of the Existing Systems
- Problem Statement
- Proposed System Design
- Framework/Algorithm
- Technologies Stack for Proposed System
- Details of Database / Input to the System
- Implementation
- Conclusion
- References

Introduction

Object detection

- It is a fundamental computer vision task that involves identifying and locating objects within images or videos.
- It recognizes what objects are present but also determining where they are in the frame, often by drawing bounding boxes around them.

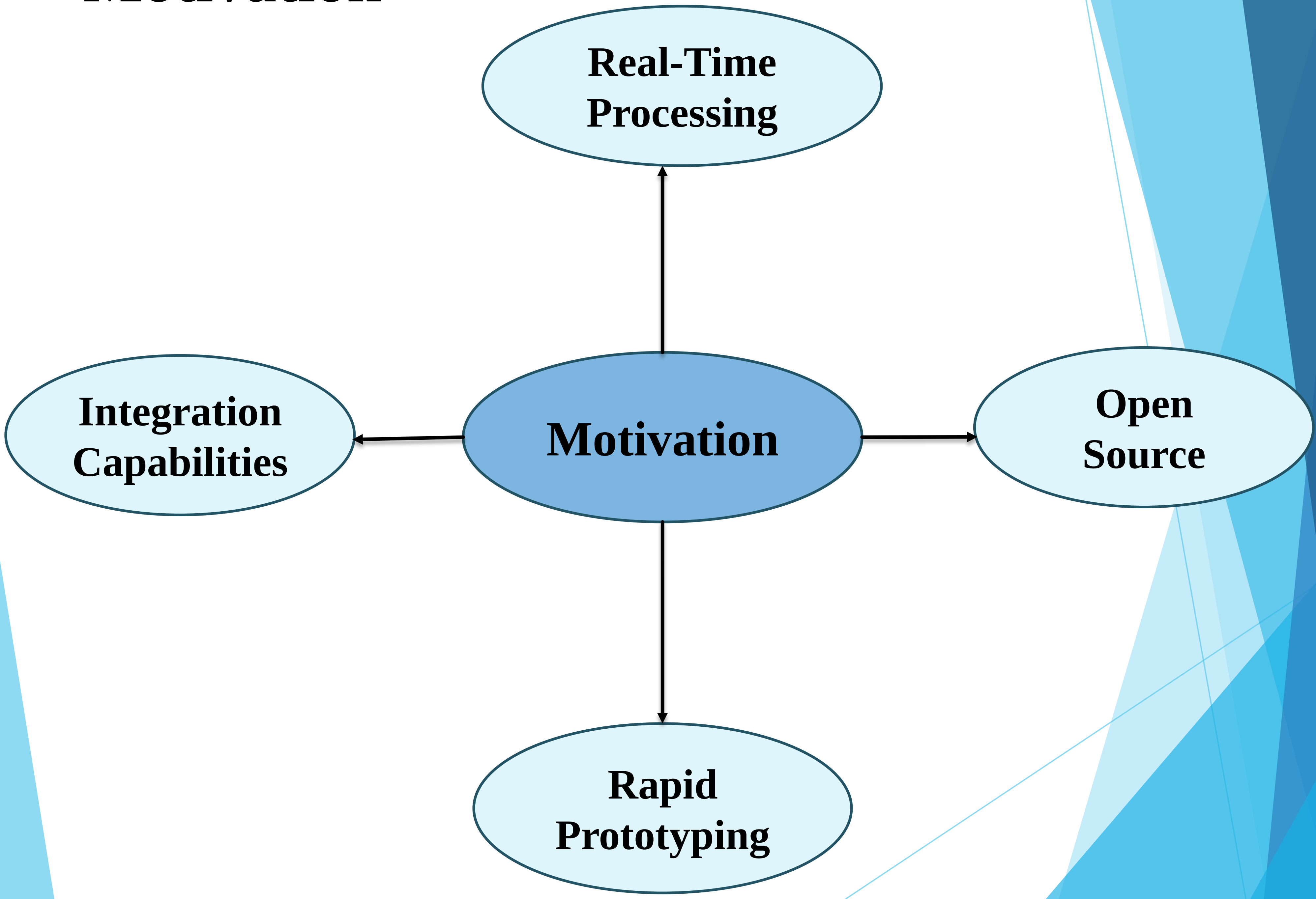
TensorFlow

- It is an open-source machine learning framework that provides tools for creating, training, and deploying object detection models.

Applications

- **Security and Surveillance -** Facial Recognition , Smart Cameras
- **Autonomous Vehicles -** Self-Driving Cars
- **Augmented Reality (AR)** - Navigation

Motivation


Real-Time Processing
Integration Capabilities
Motivation
Open Source
Rapid Prototyping

Objectives:

1. To Achieve Real-Time Object Detection
2. To Support Multi-Object Detection and Tracking
3. To Improve System Robustness and Reliability
4. To Optimize Computational Efficiency

Literature Survey of the existing system

Sr. No.	Paper Title	Key Contributions	Limitations
[1]	"You Only Look Once: Unified, Real-Time Object Detection (IEEE CVPR 2016) Redmon, Joseph, Santosh Divvala, Ross B. Girshick, and Ali Farhadi"	Introduces YOLO, a single regression model for real-time object detection with high accuracy and speed.	Struggles with small object detection and can miss finer details due to its grid-based approach.
[2]	"Real-Time Object Detection with TensorFlow (IEEE CVPR 2017) Wang, Yuxin, Jifeng Dai, and R. Girshick"	Discusses integrating models like SSD and YOLO with TensorFlow for optimized real-time performance.	Implementation complexities and varying performance across different hardware platforms can be challenging.
[3]	"Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks (IEEE ICCV 2015)" Shaoqing Ren, Kaiming He, Ross B. Girshick, Jian Sun	Introduces Region Proposal Networks (RPN) for efficient object detection within a unified framework.	Still slower than single-shot models; RPN can be computationally intensive.
[4]	"EfficientDet: Scalable and Efficient Object Detection (IEEE CVPR 2020)" Zhang et al.	Presents a compound scaling method for balancing accuracy and efficiency in object detection.	Complexity in model design; scaling may lead to diminishing returns in performance for certain tasks.
[5]	"Focal Loss for Dense Object Detection (IEEE ICCV 2017)" Lin, Tsung-Yi, Priya G. Patel, and Kaiming He.	Introduces Focal Loss to address class imbalance, improving detection accuracy for hard-to-detect objects.	Adds complexity to loss function; may require careful tuning to achieve optimal results.

Sr. No.	Paper Title	Key Contributions	Limitations
[6]	"Focal Loss for Dense Object Detection (IEEE ICCV 2017)" Lin, Tsung-Yi, Priya G. Patel, and Kaiming He.	Introduced Focal Loss to address class imbalance in object detection, improving focus on hard-to-detect objects.	Primarily focuses on improving loss function; may not address other architectural weaknesses in models.
[7]	"Mask R-CNN (IEEE ICCV 2017)" Huang, Kaiming, Yi Li, and Piotr Dollár.	Extended Faster R-CNN by adding a segmentation mask prediction branch, enhancing instance segmentation accuracy.	Computationally heavier due to additional mask generation, which may affect speed in real-time applications.
[8]	"EfficientDet: Scalable Object Detection (IEEE CVPR 2020) Tan, Mingsheng, Ruoming Pang, and Qiang Chen"	Developed a scalable architecture with compound scaling for improved accuracy and efficiency in object detection.	Complexity in tuning the scaling factors; may not perform optimally in all scenarios compared to simpler models.
[9]	"YOLOv4: Optimal Speed and Accuracy of Object Detection (arXiv 2020) Bochkovskiy, Alexey, Chien-Yao Wang, and Hong-Yuan Mark Liao"	Achieved a balance of high speed and accuracy with improvements in backbone networks, feature pyramid networks, and data augmentation.	Potentially less accurate in complex scenes compared to more focused models like Mask R-CNN.

Limitations of Existing Systems

- **Difficulty with Small Objects**: Detecting small objects can be challenging.
- **Complexity of Implementation**: Setting up and fine-tuning the object detection model can be complex and time-consuming.
- **Dependency on Quality Data**: The effectiveness of object detection systems is heavily reliant on the quality and diversity.
- **Sensitivity to Environmental Changes**: Object detection systems can be sensitive to changes in lighting, weather.

Problem statement

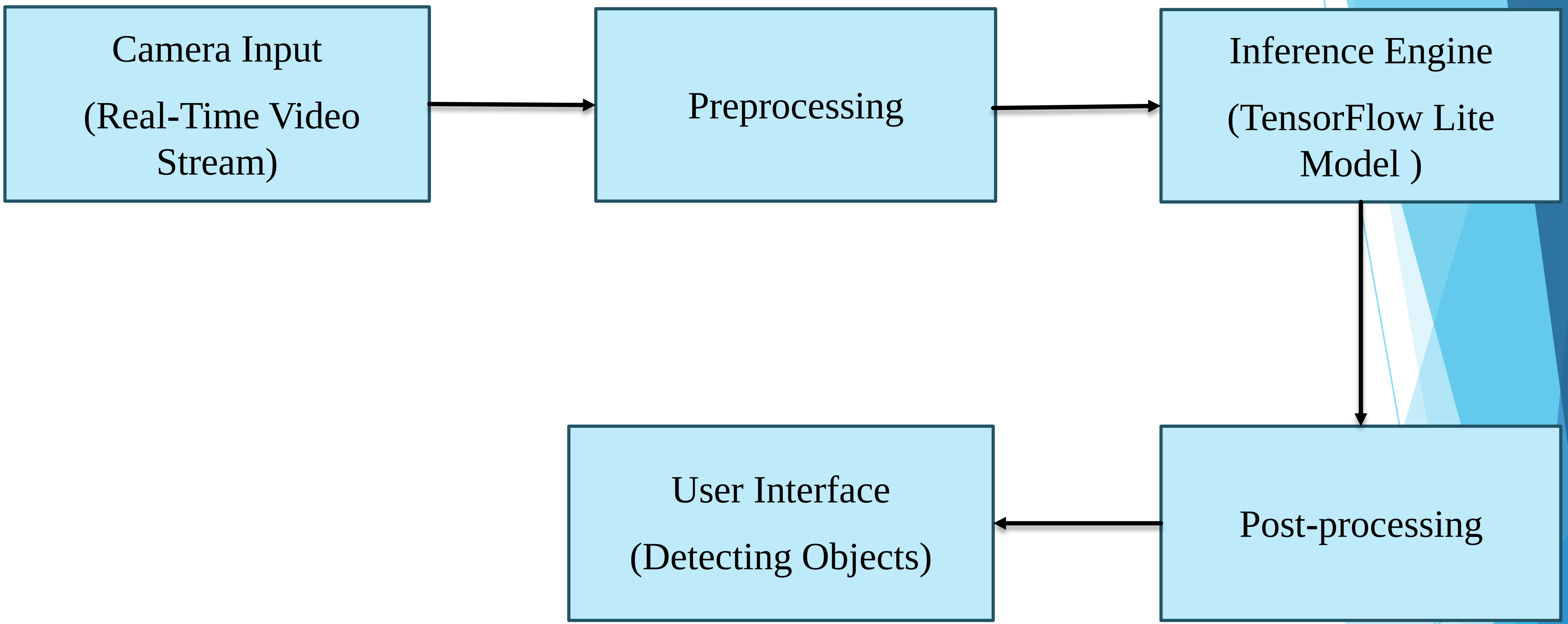
- In today’s rapidly evolving technological landscape, real-time object detection has become a critical component in various applications such as autonomous vehicles, security systems, and augmented reality.

- The project aims to develop an efficient object detection system using TensorFlow that accurately identifies and localizes multiple objects in real-time across various environments.

- Key challenges include selecting and optimizing suitable models, preparing diverse training data, enhancing detection of small and overlapping objects, and ensuring real-time inference.

- The system will feature a user-friendly interface for visualizing results and will be designed for easy integration with existing applications.

Proposed System Design



General Architecture Diagram

Proposed System Design

- **Camera Input** : Captures real-time video.
- **Preprocessing** : Prepares the video frames for the model.
- **Inference Engine** : TensorFlow Lite model (e.g., MobileNet V1) runs on the CPU.
- **Post-processing** : Applies filters and thresholds to the model's output.
- **Output** : Displays detected objects with details like bounding boxes and inference time.
- **User Interface** : Allows users to control various settings like threshold, max results, etc.

Framework

❖ **Frameworks:**

- **TensorFlow Lite**: Ideal for running TensorFlow models on mobile devices. It’s optimized for performance on mobile and embedded devices.

- **Kotlin:** It plays a critical role in bridging the TensorFlow Lite model with the Android application's user interface and real-time processing capabilities.

Framework

- **Convolutional Neural Networks (CNNs):** It is used to process the input images/videos and generate predictions about the presence and location of objects.
- **Single Shot MultiBox Detector (SSD):** This algorithm is responsible for predicting bounding boxes and class probabilities directly from the feature maps of the CNN, allowing real-time detection.
- **Depthwise Separable Convolutions (Used in MobileNet):**used to make the model efficient enough to run on devices with limited computational resources.
- **Non-Maximum Suppression (NMS):**It is used to select the best bounding boxes among multiple overlapping boxes predicted by the model.

Technology Stack for Proposed System

1. Android Studio Giraffe (2022.3.1)

The latest version of Android Studio offers enhanced UI design tools and a faster emulator for improved development efficiency.

2. Kotlin

Kotlin is a concise and modern programming language that enhances productivity with its null safety and interoperability with Java.

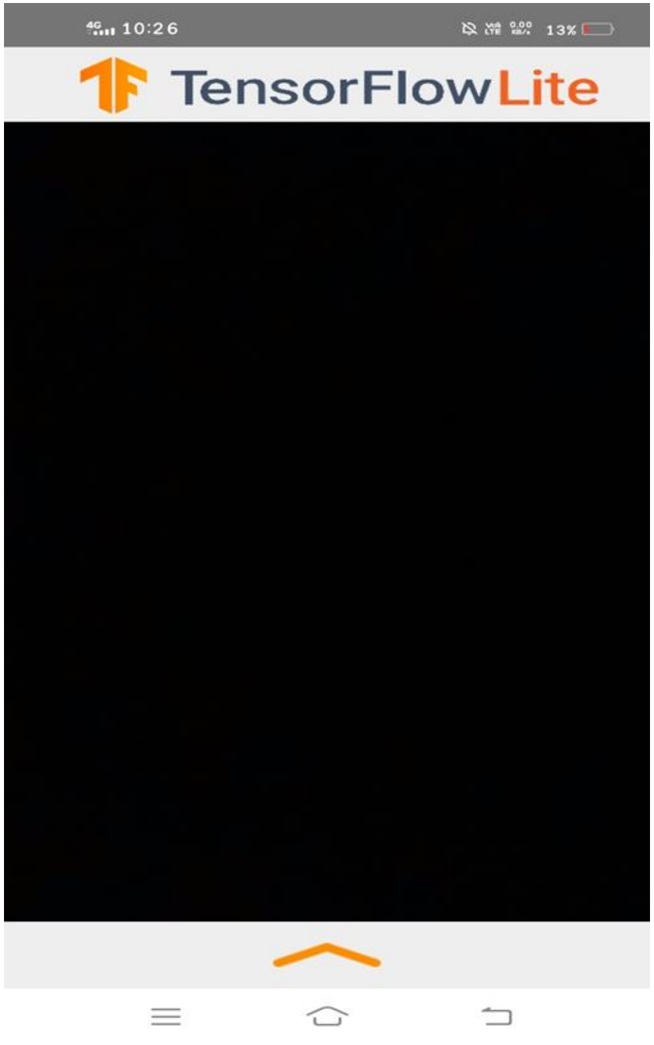
3. TensorFlow Lite

TensorFlow Lite enables on-device machine learning, allowing for quick inference and improved user privacy.

Details of Database / Input to the System

- **COCO (Common Objects in Context):** Large-scale dataset with over 330,000 images and 80+ object categories, widely used for object detection and segmentation tasks.
- It provides diverse, real-world scenarios for training and evaluating models, making it ideal for this project.
- The primary application is to train models for detecting and classifying objects within images.
- It is also used for tasks like instance segmentation, where the goal is to delineate each object instance within an image.
- **Download Link:** https://cocodataset.org/#download

Implementation



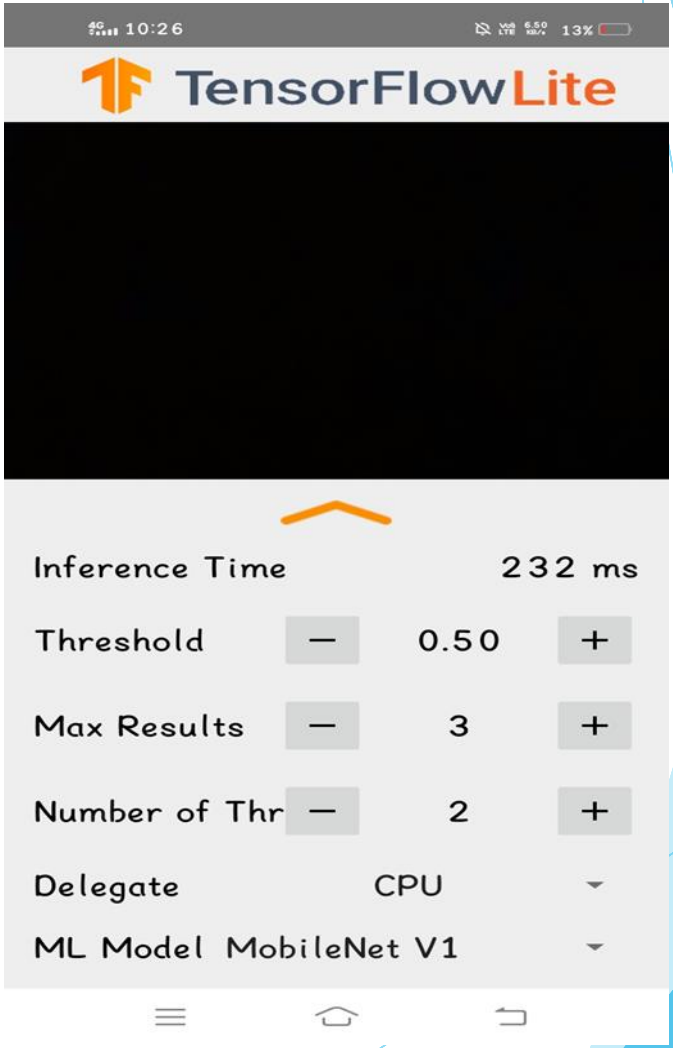


Fig.5.3.3: Output 1

Fig.5.3.4: Output 2

Implementation

Fig.5.3.4: Output 3

Conclusion

In conclusion, our object detection system has successfully achieved its goals, resulting in a fully functional and optimized solution capable of real-time performance. Built using TensorFlow Lite and Kotlin, the system efficiently processes live video feeds, detecting and classifying objects in real-time with high accuracy. Key optimizations, such as model quantization and multi-threading, have been implemented to ensure the system runs smoothly on mobile and resource-constrained devices.

Reason for the changes

References

[1] "You Only Look Once: Unified, Real-Time Object Detection (IEEE *Conference on Computer Vision and Pattern Recognition*. 2016) Redmon, Joseph, Santosh Divvala, Ross B. Girshick, and Ali Farhadi"

[2] "SSD: Single Shot MultiBox Detector (IEEE *European Conference on Computer Vision*. 2016) Liu, Wei, Dragomir Anguelov, Dumitru Erhan, Cristian Rodriguez, and Sermanet Pierre"

[3] "Real-Time Object Detection with TensorFlow (IEEE *Conference on Computer Vision and Pattern Recognition*. 2017) Wang, Yuxin, Jifeng Dai, and R. Girshick"

[4] "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks (IEEE *International Conference on Computer Vision* 2015)" Shaoqing Ren, Kaiming He, Ross B. Girshick, Jian Sun

[5]"EfficientDet: Scalable and Efficient Object Detection (IEEE *Conference on Computer Vision and Pattern Recognition*. 2020)" Zhang et al.

[6]"Focal Loss for Dense Object Detection (IEEE ICCV 2017)" Lin, Tsung-Yi, Priya G. Patel, and Kaiming He.

[7] "Mask R-CNN (IEEE ICCV 2017)" Huang, Kaiming, Yi Li, and Piotr Dollár.

[8] "EfficientDet: Scalable Object Detection (IEEE *Conference on Computer Vision and Pattern Recognition*. 2020) Tan, Mingsheng, Ruoming Pang, and Qiang Chen"

[9] "YOLOv4: Optimal Speed and Accuracy of Object Detection (arXiv 2020) Bochkovskiy, Alexey, Chien-Yao Wang, and Hong-Yuan Mark Liao"

Thank You…!!