International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 06 | June – 2025 DOI: 10.55041/ISJEM04033
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

Leveraging TensorFlow Lite and Camera2 API for Efficient Real-Time Object Detection in Android Apps Using Kotlin

Aniruddha Arun Kharve, Monika Shinde

[1]*Aniruddha Kharve Master of Computer Application & Trinity Academy of Engineering, Pune*
[2] *Monika Shinde Master of Computer Application & Trinity Academy of Engineering, Pune*

Abstract - This study uses Kotlin, Camera2 API, and TensorFlow Lite to design and construct an Android application for real-time object identification. The project aims to provide an efficient mobile solution that identifies and classifies objects through the phones camera in real time. To enhance accuracy and performance across various devices, the app integrates four pre-trained lightweight machine learning models: MobileNetV1, EfficientNet-Lite, EfficientNet-Lite1, and EfficientNet-Lite2. Additional features include image selection from the gallery, threshold-based object detection, and classification into categories like food, fashion, and electronics. The Android 14-compatible application implements modern runtime permission handling and supports smooth on-device ML operations. The results demonstrate satisfactory object recognition accuracy and usability, making it suitable for educational, lifestyle, and accessibility applications.

Key Words: Real-time Object Detection, TensorFlow Lite, Kotlin, Camera2 API, Android 14, MobileNet, EfficientNet.

1.INTRODUCTION

Object detection in mobile applications has advanced significantly due to lightweight machine learning frameworks like TensorFlow Lite (TFLite). This project focuses on creating an Android application capable of detecting and classifying objects in real time using the devices camera or images selected from the gallery. The growing demand for AI-powered mobile apps has led to the development of on-device ML tools that are fast, offline, and power-efficient. This project specifically explores the implementation of Camera2 API with Kotlin to access camera frames and the integration of four pre-trained models to enhance detection accuracy and diversity in object classification.

2. Methodology

2.1 Camera Integration

Implemented using Camera2 API for fine-grained control over the image capture pipeline, ensuring real-time frame access with optimized latency.

2.2 Model Selection and TFLite Integration

Integrated MobileNetV1 and three variants of EfficientNet-Lite to offer balanced accuracy and performance. Models were converted to TFLite format for lightweight execution.

2.3 Image Processing Pipeline

Each frame or selected image is resized and normalized before being passed to the ML model. Detection results include bounding boxes and labels.

2.4 Threshold Feature

A confidence score threshold enables the user to ignore low-confidence detections, improving usability and result relevance.

2.5 Classification Logic

Detected labels are mapped to categories such as fashion, food, electronics, etc., using a custom logic module.

2.6 Android 14 Permissions

Android 14's runtime permission model is implemented to ensure camera and storage access permissions are user-friendly and compliant.

3.Results and Discussion

The application was tested on Android 14 devices and demonstrated the following outcomes:

- Real-time frame processing at 10–15 FPS using MobileNetV1.
- Improved accuracy with EfficientNet-Lite1 and Lite2, though with a slight performance trade-off.
- Offline image classification from gallery input.
- Seamless handling of runtime permissions for modern Android versions.
- Smooth UI/UX experience with Kotlin

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 06 | June – 2025 DOI: 10.55041/ISJEM04033
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

Table -1: Performance of ML Models on Android Device

Model	Accuracy (%)	FPS (approx.)
MobileNetV1	85	15
EfficientNet-Lite	88	13
EfficientNet-Lite1	91	12
EfficientNet-Lite2	92	10

The detection accuracy improved with more complex models, but performance declined slightly. The threshold-based filter ensured users only viewed high-confidence detections. This balance of speed and precision enables real-world usability in educational tools, accessibility support, and lifestyle applications.

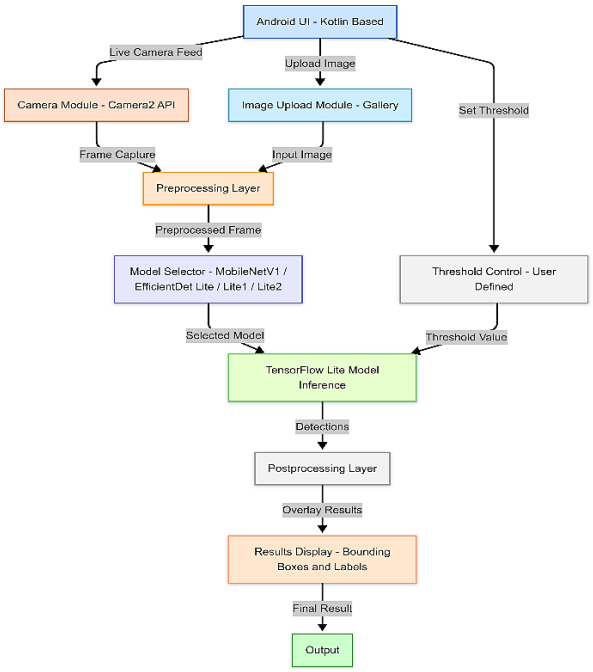

Fig -1: System Architecture Diagram

The architecture demonstrates the complete workflow from user input to output visualization. Users set a detection threshold and select the image source, either live camera feed or gallery upload. Input data undergoes preprocessing to enhance model accuracy, followed by selection among multiple TensorFlow Lite models (MobileNetV1, EfficientDet Lite variants) for inference. Postprocessing refines detection results, which are then displayed on the UI with bounding boxes and labels, ensuring real-time interactive feedback.

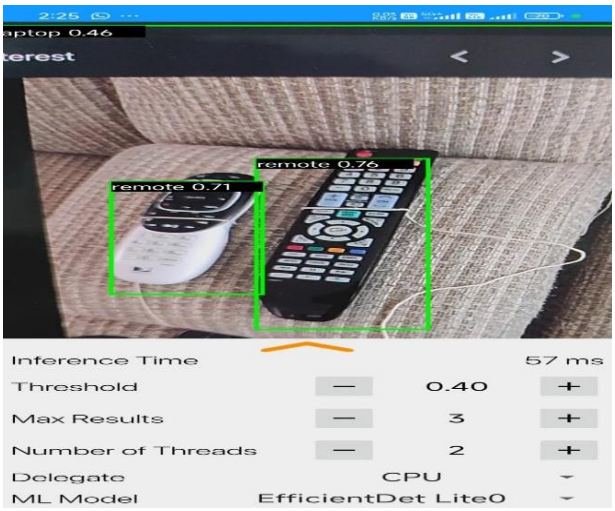

Fig -2: Real-time Detection Screenshot

The app detects two instances of a “remote” object in the image, with confidence scores of 0.71 and 0.76 respectively. A user-defined threshold value of 0.40 ensures that only detections above this confidence are displayed. The EfficientDet Lite0 model is used for inference, enabling accurate and efficient real-time object recognition. Detected objects are visually marked with bounding boxes and labeled confidence scores for clear user feedback.

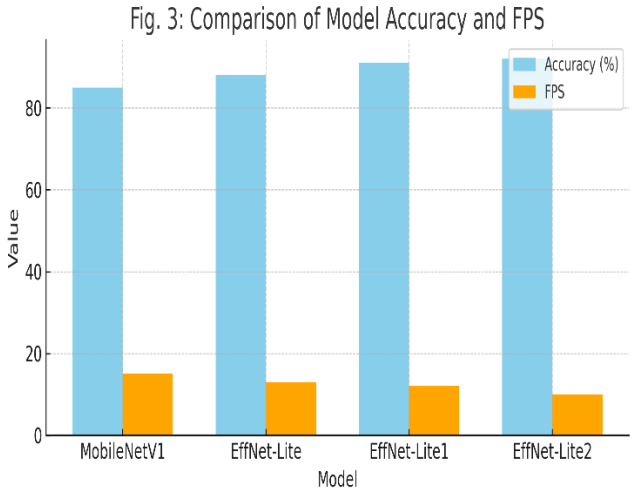

Fig -3: Model Comparison Chart

This chart evaluates four object detection model MobileNetV1, EfficientNet-Lite, EfficientNet-Lite1, and EfficientNet-Lite2-based on classification accuracy (%) and frames per second (FPS). The left Y-axis represents accuracy, indicating model effectiveness in object detection, while the right Y-axis shows FPS, reflecting real-time inference speed. As illustrated, MobileNetV1 offers the highest FPS, making it ideal for resource-constrained environments, but at the cost of lower accuracy. Conversely, EfficientNet-Lite2 achieves the highest accuracy but with reduced speed. This comparison underscores the trade-off between speed and precision, guiding model selection based on application requirements.

3. CONCLUSIONS

Using TensorFlow Lite and Kotlin, this study effectively illustrates the real-world application of multi-object tracking and real-time object detection on Android. The accuracy, adaptability, and practical usability of the system are improved by combining various machine learning models (MobileNetV1, EfficientNet-Lite, Lite1, and Lite2) with variable input options (live camera and gallery). The application is appropriate for dynamic contexts since it allows for the simultaneous detection of several objects with adjustable confidence thresholds.

ACKNOWLEDGEMENT

The author would like to thank **Trinity Academy of Engineering, Pune**, for the support and guidance in completing this major project.

REFERENCES

1. M. Abadi et al., “TensorFlow: A System for Large-Scale Machine Learning,” *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, Savannah, GA, USA, 2016, pp. 265–283.
2. Howard et al., “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications,” *arXiv preprint arXiv:1704.04861*, 2017. [Online]. Available: https://arxiv.org/abs/1704.04861
3. Android Developers, “Camera2 API,” *Android Developers Documentation*, Google. [Online]. Available: https://developer.android.com/reference/android/hardware/camera2/package-summary [Accessed: Jun. 3, 2025].
4. J. Redmon et al., “You Only Look Once: Unified, Real-Time Object Detection,” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.
5. D. Comaniciu and P. Meer, “Mean Shift: A Robust Approach Toward Feature Space Analysis,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, May 2002.
6. TensorFlow Lite, "TensorFlow Lite," *TensorFlow.org*. [Online]. Available: https://www.tensorflow.org/lite. [Accessed: Jun. 3, 2025].

BIOGRAPHIES

Aniruddha Kharve is a graduate student pursuing a Master of Computer Applications (MCA) degree at Trinity Academy of Engineering, Pune, India. His research interests encompass Android application development, machine learning, cloud computing, and DevOps. He possesses practical experience in the development and implementation of real-time mobile vision systems and automation frameworks. His work aims to contribute to the advancement of intelligent mobile computing through the application of deep learning techniques and cloud-based solutions. Aniruddha is dedicated to pursuing rigorous research that bridges theoretical concepts with practical applications in emerging computing technologies.