

Live Object Detection Using Flutter

Mr.J. CYPTO, B. Yaswanth, G.N Sri Charan, K. Suresh, K. Vishal

Associate Professor

Department of Computer Science Engineering, SRM Institute of Science and Technology, Chennai, India

Department of Computer Science Engineering, SRM Institute of Science and Technology, Chennai, India

Department of Computer Science Engineering, SRM Institute of Science and Technology Chennai, India

Department of Computer Science Engineering, SRM Institute of Science and Technology Chennai, India

Department of Computer Science Engineering SRM Institute of Science and Technology, Chennai, India

Date of Submission: 25-11-2022

Date of Acceptance: 06-12-2022

ABSTRACT— The system's goal is to detect real-time objects that are scanned through a mobile. The proposed framework makes use of Machine Learning and Google's TensorFlow Lite object detection API. It's a useful app. Object Detection is widely utilized in several applications such as detecting vehicles, face detection, autonomous vehicles and pedestrians on streets. TensorFlow's Object Detection API is a powerful tool that can quickly enable anyone to build and deploy powerful image recognition software. Object detection not solely includes classifying and recognizing objects in an image but additionally localizes those objects and attracts bounding boxes around them. The captured images are collected and carried out to the image processing techniques and to feed into the model. And classifies the images and identifies the objects. the camera will capture the visual input of the objects and it is transferred for processing. This paper mostly focuses on detecting harmful objects like threatening objects. Real-time object detection is a vast, vibrant, and complex area of computer vision. If there is a single object to be detected in an image, it is known as Image Localization and if there are multiple objects in an image, then it is Object Detection. This detects the semantic objects of a class in digital images and videos. The applications of real-time object detection include tracking objects, video surveillance, pedestrian detection, people counting, self-driving cars, face detection, ball tracking in sports, and many more.

1. INTRODUCTION

Production of images: Image processing focuses on transforming raw images to achieve the desired result. The aim of image processing is usually to enhance images or prepare them as input for a particular task, while the goal of computer

vision is to describe and illustrate images. Noise reduction, contrast, and rotation operations, for example, are common image processing operations that can be performed at the pixel level without requiring a complex understanding of what is going on in the image. Machine Learning: Artificial Intelligence includes Machine Learning as a subset. Machine learning is the study of giving computers the opportunity to learn and create their own programmers in order to make them more human-like in their actions and decisions. This is accomplished with the least amount of human interaction possible, i.e., no explicit programming. The learning process is automated and enhanced based on the machines' experiences along the way. Machines are fed high-quality data, and different algorithms are used to create machine-learning models based on that data.

The algorithm to use is determined by the type of data available and the type of operation to be automated. Supervised Machine learning Model: In supervised learning, you use well-labeled data to train the computer. It indicates that some information has already been marked with the correct answer. It is comparable to learning that occurs in the presence of a supervisor or instructor. A supervised learning algorithm learns from labeled training data and aids in the prediction of unexpected data outcomes.

Accurate supervised machine learning can be designed, scaled, and deployed successfully. Unsupervised Machine learning Model: Unsupervised learning is a machine learning methodology in which the algorithm does not require supervision. Instead, you should let the model figure out what it needs to do on its own. By using a dataset, it detects the objects like people, mouse, laptops, etc. It is primarily concerned with unlabeled data. In comparison to supervised

learning, unsupervised learning algorithms allow you to perform more complex processing tasks. Unsupervised learning, on the other hand, can be more volatile than other natural learning approaches such as deep learning and reinforcement learning.

Flutter: Flutter is an open-source mobile SDK that allows developers to build native-looking Android and iOS apps from a single code base. Flutter has been around since Google released it in 2015, and it remained in beta until December 2018, when it was officially launched. Flutter has been generating a lot of interest since then. According to GitHub stars, Flutter is now in the top 11 app repositories.

Problem Statement - Firstly, we need to understand the problem in an efficient way. Here we are building a live object detection app using flutter with the YOLO V3 algorithm. It detects the object by first taking pictures of it comparing the pictures with the data and selecting the object which matches the dataset and showing a percentage of how accurate the object is with the data.

Methodology - We have designed a live object detection app that detects objects. Firstly, the application will check if the device has the access to the camera, then it will capture the live images which will be in 3D it will convert the video into single frames and it will find the objects in the frames. We have used a package named TFLITE which was available in the flutter packages in this package we have the YOLO V3 and TensorFlow Lite. After detecting the objects, it will find the object in the dataset. If the detection is greater than the expected percentage it will detect the object otherwise it will discard the detection. In pre-processing will be processed the image which was collected and will re-size play the main role of pre-processing it will make the image 640x680 it will generate a stable output and divides the image using the grid system. This is an advantage especially for data sets that are too large to store it in memory, as the required data (e.g. a batch) is loaded from the disk as and when required and then processed. After generation of tf records we create a label map that gives a unique id for each of the category to identify.

II. EXISTING SYSTEM

In the existing system, they were very few applications that were developed by the flutter, and mostly they were based on Yolo v2 which was old some objects can't detect and the detection is also

poor compared to the Yolo v3 which we used in this application. The existing system used computer-generated images which are used to detect objects but not instantly. The existing system may have data set errors that cause defects in the detection of the objects.

III. PROPOSED SYSTEM

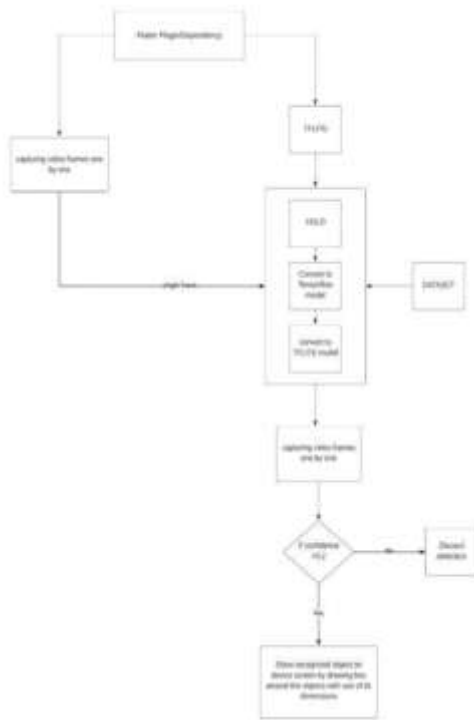
The proposed system contains main contender is it supports build a app on live object detection. It uses an algorithm called YOLO V3 algorithm. whereas the detection rate and filtering the objects result will be very high and accurate. SSD technology single shot detection where within single shot entire image capturing process will be finished it results in efficient way of 11 capturing the images. It also comprises of many complex processes like Mobile, Laptop, feature extraction etc. we can perform all these complex processes with less efficient webcams. It Easily detects the log of objects. Processing rate is higher. When there is a high processing rate the result will be efficient.

Feasibility Study - A Feasibility study is carried out to check the viability of the project and to analyze the strengths and weaknesses of the proposed system. The application is live object detection using flutter. The feasibility study is carried out in two forms.

Economic Feasibility - The proposed system does not require any high-cost equipment. This project can be developed within the available software.

Technical Feasibility - The proposed system is completely a Flutter model. The main tools used in this project are Flutter, Android Visual studio and the language used to execute the process in Dart. The above-mentioned tools are available for free and technical skills required to use this tool are practicable. From this we can conclude that the project is technically feasible.

IV. SYSTEM ARCHITECTURE



The above Figure 1, shows the work flow of our work. In this paper image data set has been created using threatening object images which are taken from google images. We have taken nearly 78 object images out of which the images split into train and test images. We train the model on train images and evaluate the model using the test performances. Firstly, we start from labelling the images. We label the images using the Labeling tool we create a rectangular box around the object which give ordinates of the object where it lies. The image data is initially stored in xml format for each image. As the number of images is more we will have a same number of xml files to avoid complexity. So, we create a CSV file that has data of all the images. Since we are working with an image dataset the size of images is more so using a binary file format to store our data has an important impact on performance of the model. Binary data occupies very little space on disk, consumes less time to copy and can be read more efficiently from disk. So, we convert data into record file format. The main advantage of using the TFLite file format is that the data can be optimized in multiple ways.

This is an advantage especially for data sets that are too large to store it in memory, as the required data (e.g., a batch) is loaded from the disk as and when required and then processed. After generation of tf records we create a label map that

gives a unique id for each of the category to identify. We have used a trained model of Faster RCNN. The default parameters changed to our use model by using config file. Model config block is about the configuration of a model.

These config files are used to configure parameters to the initial setting for some of the computer codes. Here, each Model Config specifies one model to be served, including its name and the path. After the initialization of everything we train our based on our system configuration. When each training phase has begun, the loss will be reported. As the training method progress, it will begin high and get smaller and lower. We are training on model Faster RCNN Inception V2 model. It started at 3 and fell down rapidly. In this we have trained nearly 5000 steps until loss is constantly less than 0.1.

It took almost twelve hours to train the model. The model may train faster on powerful CPU and GPU. After completing the training process, we export an inference graph by using checkpoints that are created while training a model.

This checkpoint won't contain any description of the computation defined by the model. Weights from these checkpoint files are inserted into variable operations. After finishing training from the Client script, we access the api from our side by giving the test images to detect the objects in an image.

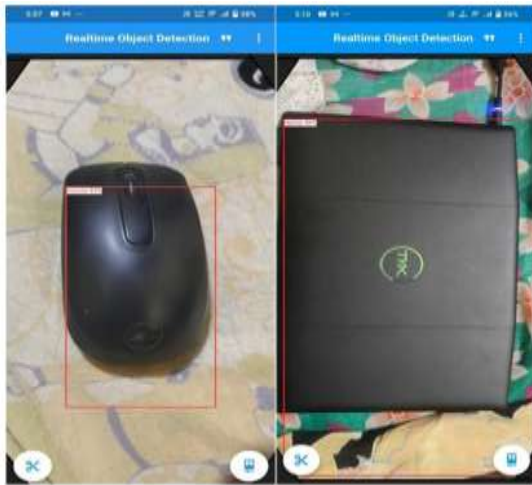
These images are led to the tensor flow serving server. TensorFlow serving is a versatile, high-performance serving system intended to produce environments different machine learning models. TensorFlow Serving makes deploying fresh techniques or algorithms and experiments straightforward while maintaining same server design and APIs. The input image we have given are returned with Bounding boxes around the object.

Typically, developers interact with Flutter through the Flutter framework, which provides a modern, reactive framework written in the Dart language. It includes a rich set of platform layout, and foundational libraries, composed of a series of layers working from the bottom to the top, we have: Basic foundational classes, and building block services The rendering layer provides an abstraction for dealing with layout. With this layer, you can build a tree of renderable objects. You can manipulate these objects dynamically, with the tree automatically updating the layout to reflect your changes. The widgets layer is a composition abstraction. Each render object in the rendering layer has a corresponding class in the widgets layer. In addition, the widgets layer allows you to

define combinations of classes that you can reuse. This is the layer at which the reactive programming model is introduced. The Material and Cupertino libraries offer comprehensive sets of controls that use the widget layer's composition primitives to implement the material.

V. MODEL DESCRIPTION

1. Creating data sets of objects –
 The collection of data is the first step in Live object detection construction.



2. Training Image –
 Training the model with the data sets collected for live object detection.

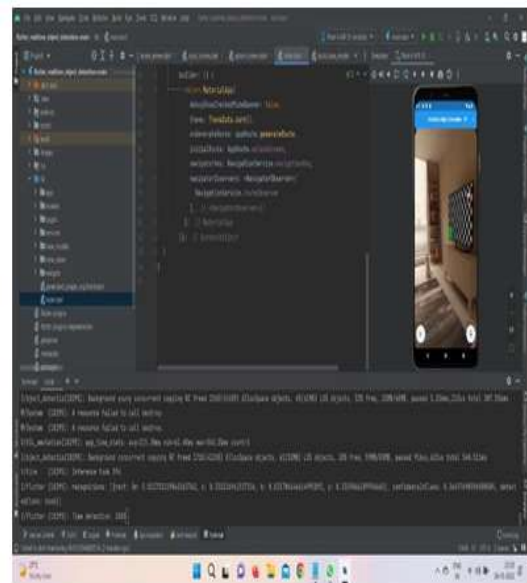


Figure represents the UML diagram, Firstly the application will find the camera that was present in the device, and then it will capture the live video and it will be doing pre-processing like resizing the frames and it will generate the stable

output of the image. After the frame will be sent to the TFLITE and it will identify the object using the dataset and divides the image using the grid system it will draw the drawing box around the object and it will detect the object.

3. Testing the model –

In this model first we will be taken as video as the input and the video is sent to TFLite, by using the YOLO V3 algorithm the video is converted into frames and matches the object in the dataset, adding the model parameters, collecting the model, and therefore the last is saving the model for the long run prediction method.



VI. CONCLUSION

We have concluded that by using flutter and TFLITE to detect the objects in real-time this application can be used in the car if there are any objects to apply breaks if the person near to car or above crashes to open air bags. This application has passed all the test cases by detecting the objects, image classifiers are used to classify images to detect the objects. The accuracy of the application is very good and can detect objects at any time. This application can be used in multiple ways like to use in cars or in security cameras when any person visits the place.

Future Enhancements - This project can be extended in many ways like in security cameras for the person who is visiting the place at night time. This also can be used not only to apply the brakes to the car or vehicle but also to decrease the speed. It will be more help full to disability people mostly blind people who can be set to their glasses so by

using text to speech they can hear object names which were around them.

VII. RESULT



REFERENCE

- [1] Mansour Ahmadi, Angelo Sotgiu, and Giorgio Giacinto. "IntelliAV: Toward the Feasibility of Building Intelligent Anti-malware on Android Devices". In: Machine Learning and Knowledge Extraction. Ed. by Andreas Holzinger, Peter Kieseberg, A Min Tjoa, and Edgar Weippl. Cham: Springer International Publishing, 2017, pp. 137–154.
- [2] James Bergstra and Yoshua Bengio. "Random Search for Hyperparameter Optimization". In: *J. Mach. Learn. Res.* 13 (Feb. 2012), pp. 281–305. ISSN: 1532-4435.
- [3] Yu, Liyan, Xianqiao Chen, and Sansan Zhou. "Research of Image Main Objects Detection Algorithm Based on Deep Learning." In 2018 IEEE 3rd International Conference on Image, Vision and Computing (ICIVC), pp. 70- 75. IEEE, 2018.
- [4] Kim, Jung Uk, Jungsu Kwon, Hak Gu Kim, Haesung Lee, and Yong Man Ro. "Object Bounding Box-Critic Networks for Occlusion-Robust Object Detection in Road Scene." In 2018 25th IEEE International Conference on Image Processing (ICIP), pp. 1313-1317. IEEE, 2018
- [5] K. K. Hati, P. K. Sa and B. Majhi, "Intensity Range Based Background Subtraction for Effective Object Detection," in *IEEE Signal Processing Letters*, vol. 20, no. 8, pp. 759-762, Aug. 2013
- [6] H. V. Ravish Aradhya, Mohana and Kiran Anil Chikodi, "Real time objects detection and positioning in multiple regions using single fixed camera view for video surveillance applications," 2015 International Conference on Electrical, Electronics, Signals, Communication and Optimization (EESCO), Visakhapatnam, 2015, pp. 1-6.
- [7] J. Li, F. Li, and M. Zhang, "A Real-time Detecting and Tracking Method for Moving Objects Based on Color Video," in 2009 Sixth International Conference on Computer Graphics, Imaging and Visualization. IEEE, 2009, pp. 317–322.
- [8] F. Labrosse, "Short and long-range visual navigation using warped panoramic images," *Robotics and Autonomous Systems*, vol. 55, no. 9, pp. 675–684, 2007
- [9] Boukhary, Shady. Flutter Clean Architecture Pub Package. April 30, 2019. Accessed on September 19, 2019. [Online]. Available: https://pub.dev/packages/flutter_clean_architecture
- [10] Boukhary, Shady. Hotter'n Hell Mobile Application. May 28, 2019. Accessed on September 19, 2019. [Online]. Available: <https://github.com/ShadyBoukhary/Axion-Technologies-HnH>.
- [11] I. B. Lisa Maria Garnweidner-Holme, Inaki Garitano, Josef Noll and Mirjam Lukasse "Designing and developing a mobile smartphone application for women with gestational diabetes mellitus followed-up at diabetes outpatient clinics in Norway " vol. 3, pp. 310-323, May 21, 2015.
- [12] Alsaleh, Saad, and Fares Alshamari. "Exploring Factors Affecting Patients Adoption of mHealth Services in the Kingdom of Saudi Arabia." *International Journal of Health, Wellness & Society*, vol 6, no. 4, 2016.