Object Detection and Image Labelling using Machine Learning Technique (Android Application Using TensorFlow Library)

Likhitha P M[1], Poornashree Narayani S Kulkarni[2], Rashmi G[3], Rashmi K V[4] , Mayura D T[5]
[1234]*Student, Dept. of Information Science and Engineering, National Institute of Engineering, Mysuru, India.*
[5]*Assistant Professor, ME &BE from BAMU Aurangabad, Maharashtra, India.*

Abstract- This paper discusses about android application of an object captured in an image which is obtained in a raw data format through camera lenses which is called as Surface View Video Format and it will have different frames. Development of this application contains usage of Machine Learning technique to obtain data and the obtained data will be communicated with this Application. This will be processed using Image Processing technique. The obtained Image will be further analysed using Fritz API of Machine Learning data set. The recognition of the object will be done through pre-processed existing datasets. Tensor Flow is used as library to implement Machine Learning for this application.

Keywords- Image Processing; Machine Learning; Fritz API; TensorFlow;

I. INTRODUCTION

Object detection is one of the major factors in Robotics. Generally, the human eyes can detect the objects around it, this includes the objects captured in an image will be converted to nerve impulses to human brain and the processing and detection of object will be done. But the machines do not have such capacity to detect an object on its own. A method to detect image is required for robots to detect, process and store the information captured. This detection process will be helpful for the robots to detect the objects during the time of movements. The basic idea of this project is to develop an application that will be useful for machines that are needed for Artificial Intelligence and to analyse the objects in their surrounding environmentusing the Machine Learning technique.

This application uses Android Camera API to capture the image. The captured images will be in surface view video format from camera. The images obtained will be raw format directly from camera lens this format is called Surface View Video format. The surface view video format will contain different frames in it. The obtained image frames will be processed using image processing technique.

The pre-processed data obtained from image processing will be further analysed using Machine Leaning technique. The Fritz API is used here for the Machine Learning Dataset analysis. The recognition of the image from pre-processed data depends upon the existing sets of pre-defined objects in the datasets.

The analysed object from the live stream of images from camera will be sent back to the application in fragment view. Along with this data the time taken for the analysis of a particular object will be retrieved, the time taken is displayed in Milliseconds. This application will be useful in Robotics for analysing the type of image captured by camera using Machine Learning technique.

II. METHODOLOGY

Existing System Methodology:In Existing System, the Object detection algorithms use the features which can be used to recognize a particular object. This model is not complicated and is easy to be implemented. Here, object detection is a single regression problem which recognize directly from bounding box coordinates and class probability. Every object has got its own class such as all circles are round in shape, or square has four sides which are used to identify the objects. This system uses online repository to get the object class details. Each and every time the system communicates with online repository to recognize the object.

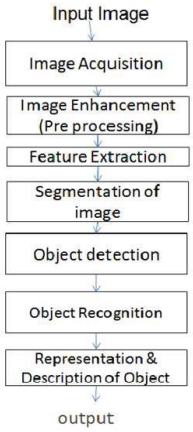

Fig.1: The primitive method of object detection in an image.

System Design of the Proposed System:

The proposed system consists of the following Modules:

- The Camera API Module
- The Image Pre-processing Module
- The Dataset Analysis Module

- The Recognition and Output Module

III. SYSTEM ARCHITECTURE

System architecture describes the behaviours, modules of the computerized system to compromise the certain requirements. Architecture shows the mechanisms of the system or building blocks.

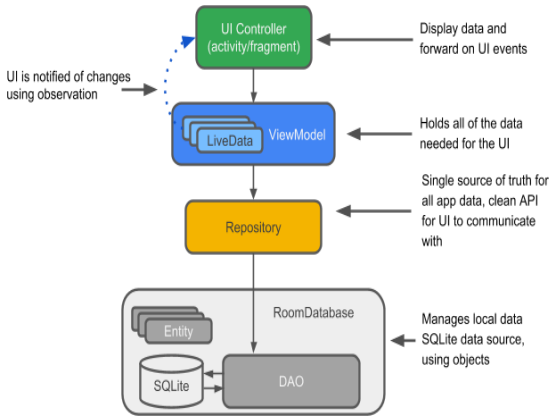

Fig.2: System Architecture of the Object Recognition in Android Application.

This system architecture has got the following layers:

- UI Controller Layer
- Live Data Layer
- Repository Layer
- Database Layer.

UI Controller Layer:

Here the UI Layer contributes to the User Interface layer, where the user captures the live images using the device hardware. This is a fragment activity where the camera frames along with the result obtained will be displayed. This layer displays the data to the user, both the input and output. Here the input will be from the camera and the output will be from the analysis of the recognized object, along with this the time taken in milliseconds will be displayed.

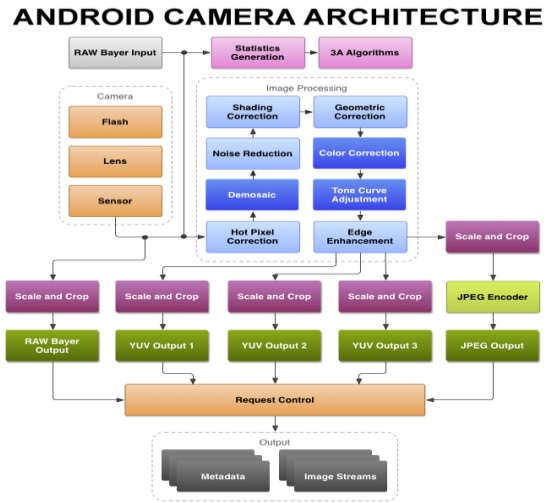

Fig.3:

Live Data Layer:

This layer will be receiving the different frames from the camera, as the name itself will explains the obtained data in this layer will be a live capturing from the device camera. Each and every time the frames gets changed this layer will keep the live data track. If the frame is changed the UI layer will be intimated about this.

Repository Layer:

This layer holds the data from Image processing module and the data will be the object model co-ordinates which are stored and sent to the Hashing Table for matching using the TensorFlow Machine Learning Algorithm. This layer is the communication between the UI and Database Layer. This layer provides a clean API service for both the camera and Datasets.

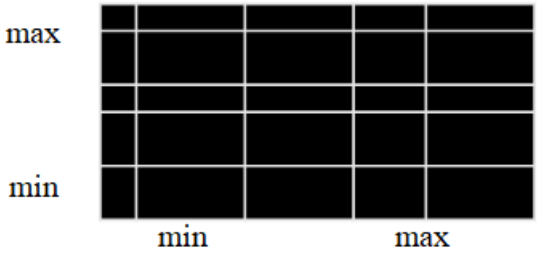

Fig.4: Hash table of co-ordinates

Database Layer:

In this database layer the datasets in Hash Table format will be kept and the matching of the Object model co-ordinates are performed. The matching process will be done using the Machine learning algorithm. The datasets will be having a copy in the Android SQLite database, the direct matching process from the hash table stored in SQLite database and the Model co-ordinates will be carried out here. The time stamp for each process will be returned back to the UI Controller Layer.

IV. IMPLEMENTATION

Camera API Module:

- Camera ManagerClass is the class used by the Android Camera to archive all the camera devices operated through Android OS.
- The settings and series of properties are defined for each camera device of its own, the characteristics are attained by its own for camera device.
- To capture camera session a session needs to be initiated, each and every time a session will be initiated during the camera capture event.

Image Processing Module:

Image formation model has got the following properties:

- Perspective projection
- Orthographic projection + scale
- Affine transformation

- From an unidentified sceneIdentify a group of features which approximately match a group of features from a model object (i.e., **correspondences**).
- Recover the **geometric transformation** that the model object has experienced and look for extra matches.

The Data Analysis Module:

- Extract a set of interest points.
- For each ordered set of three co-ordinates, non-collinear Co-ordinates points (x_1, x_2, x_3)
- Compute the co-ordinates (u,v) of the leftover features in the coordinate frame defined by the model base (x_1, x_2, x_3).
- After appropriate scaling and quantization, use the calculated coordinates (u,v**)** as an index to a two dimensional array *hash table*, and record in the corresponding hash table bin the information **(**model, (x_1, x_2, x_3))

Recognition and output Module:

- From the part extract a set of interest points.
- Choose a subjective ordered pair (x'_1, x'_2, x'_3)
- Compute the coordinates (u',v'), of the leftover feature points in the coordinate frame defined by the image base (x'_1, x'_2, x'_3**)**
- After appropriatequantization and scaling , use the calculated coordinates as an index to the hash table. For every entry (model, (x_1, x_2, x_3)) found in the equivalent bin, cast a count

V. CONCLUSION

- This application will be vigorous in detecting the object in an image captured from camera.
- This application will be able to provide the information about the image captured and time taken in milliseconds to process it.
- This application is built for Android OS.
- This application can be used for Robots to provide Artificial Intelligence about the objects in front of it and to detect them.

VI. REFERENCES

[1]. A. Krizhevsky, I. Sutskever, and G. Hinton. ImageNet classification with deep convolutional neural networks. In NIPS,2012.
[2]. M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional neural networks.
[3]. K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In ICLR, 2015.
[4]. K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. CVPR, 2016.
[5]. R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in CVPR, 2014.
[6]. J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. FeiFei. ImageNet: A large-scale hierarchical image database. In CVPR, 2009.
[7]. J. Deng, A. Berg, S. Satheesh, H. Su, A. Khosla, and L. FeiFei. ImageNet Large Scale Visual Recognition Competition 2012 (ILSVRC2012).
[8]. M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A.Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results, 2007.