Multi Scanner Android Application using Machine Learning

[1]Arati Singh, [2]Deepanshu Priy Gautam, [3]Anam Khan

[1]Student, [2]Student, [3]Assistant Professor
[1]Department of Information Technology,
[1]Galgotias College of Engineering and Technology, Greater Noida, India

Abstract: **With an increase in the need for various scanning and detecting functionality in android applications, the text recognition, QR code/barcode authentication, and object detection fields have become one of the most researched areas. In our proposed model, we tried to find a better way to provide all the three most demanding and useful functionalities in a single Android application. In our proposed model, for our first functionality, that is object detection, a Deep Convolutional Neural Network model is used to identify and detect real-time objects using Tensor Flow and perform both training and test phases. Text Scanning, our second functionality, is Optical Character Recognition (OCR), a mechanical or electronic interpretation of manually written pictures or typewritten content into machine-editable content. This functionality was implemented in our app using the Google Cloud Mobile Vision API. For our third functionality, QR code or barcode authentication, we used the Zxing and Karumi Dexter Libraries.**

Keywords**: Text Recognition, Object Detection, QR code and Barcode Authentication, Google Mobile Vision API, Zxing and Dexter Library, Convolutional Neural Network.**

I. INTRODUCTION

Machine learning is an important part of our lives in today's world. It has already begun to consume a significant portion of our daily activities. Machine learning allows a computer to "think" without having to be programmed. The demand for data and information available in printed media has skyrocketed in today's world. We have developed a number of technologies to save time, money, and energy. One of them converts softcopy JPG, JPEG, or PDF to text. This is known as text extraction. Character recognition and word and text detection are both difficult tasks. With the rise of autonomous vehicles, smart video surveillance, facial detection, and various people-counting applications, fast and accurate object detection systems are in high demand. These systems not only recognise and classify every object in an image, but also localise each one by drawing a bounding box around it. QR codes have become an important component in enabling various technical solutions and are used by millions of people worldwide every day. With the massive increase in QR code usage and the growing number of apps, the majority of which include sensitive tasks like payment and ticketing,

Using machine learning, we proposed Multi Scanner, an Android-based application. Because there aren't many Android apps that offer multiple scanning functions in a single app, we came up with the idea of creating Multi Scanner. Our application serves three major functions, which are as follows:

1. Object Detection
2. Text Recognition and Extraction
3. Scanning of QR codes and barcodes

II. LITERATURE REVIEW

In January 2013, Divakar Yadav, Sonia Sanchez-Cuadardo, and Jorge Marato used Artificial Neural Network to extract text from images in Hindi. Even pre-processing tasks are used in the literature for various considerations such as the conversion of grayscale images to binary graphics, image correction, and segmentation of the textual content of this record to lines, columns, provisions, and also at the additional level of the basic logos. G.R. Hemalakshmi, M. Sakthimanimala, and J. Salai Ani Muthu proposed Extraction of Text from an Image and Language Translation Using OCR in April 2017. Researchers used a number of steps, including scanning, pre-processing, segmentation, feature extraction, recognition, post-processing, and PDF generation, according to them. Prof. Vaibhav. V. Mainkar, Ms. Jyoti A. Katkar, Mr. Ajinkya B. Upade, and Ms. Poonam R. Pednekar created an android app for editable handwritten text in 2020. The implementation of an android application for character recognition is a broad field of study. It converts images into text using OCR technology. Developers concentrated on handwritten text and compared handwritten and printed text. Image Acquisition, Preprocessing, Segmentation, Feature Extraction, and Postprocessing are all done with the OCR Algorithm [1].

Diego Renan Bruno et al. created a real-time traffic signal sign detection system to aid vehicle drivers. Various traffic signal signs have been identified and classified by the system. The slide window technique is used to feed the input image into this system, and the CNN algorithm is used to detect traffic signal signs [2]. Davide Mulfari et al. proposed a system for identifying how the proposed mobile system distinguishes artifacts in original image retrievals from Google Image Search Engine. In this case, classification is done using Tensor flow on a Linux desktop machine. After processing, it computes each visual data point and returns a data structure with a text description.

The ISO/IEC 18004:2006 (revised in 18004:2015) standard is used to create a two-dimensional barcode known as a QR code. In his analysis, P. Bodnar summarises and presents various localization methods. Hirata and Bellusi presented an approach in 2013 that can detect QR codes with high accuracy in a large database of arbitrarily acquired images. This is a highly cited and acclaimed work that many researchers have used in their research. This paper describes a two-stage component-based detection method. Hansen et al. 2017 use the YOLO (You Only Look Once) deep Learning-based detector for 1D and 2D barcode detection. Based on tests performed with the Zxing and Zbar open-source libraries for QR code decoding, they are also able to achieve image preprocessing in terms of rotation and increase decoding performance. S. Li et al. (2018) presented a fast QR code detection method based on run-length coding. The presented algorithm is run after image binarization, for which the authors prefer the Otsu method due to its superiority under variable illumination. Blanger and Hirata presented an evaluation of deep learning techniques for detecting QR codes in natural scenes in 2019. On their own dataset, the authors demonstrated a novel adaptation of the SSD architecture and compared it to the FastQR method proposed by Bellusi et al. J. Zhang et al. 2020 present a two-layered approach for detecting barcodes (including QR codes) that not only improves detection accuracy but is also robust in complex environments and against image distortions.

III. METHODOLOGY

A. Object Detection

The goal of this paper is to identify an object with greater accuracy from a single frame in real time. The proposed system was created with Deep Neural Networks and trained with pre-generated datasets. The proposed model has two phases: training and testing. In the proposed system, we must first import data for execution. After that, the imported data is normalised. The weight and bias parameters must then be specified. The computational model is built and the output is obtained based on the specified parameter values. The loss function is declared based on the output value. In order to improve prediction, the computational model is trained and tested with a CNN network. During the training phase, various original and generative images were used as input, and the results were validated. Finally, during the testing phase, original images were used as input, and the mode correctly detected the object.

The following is the procedure for the proposed work:

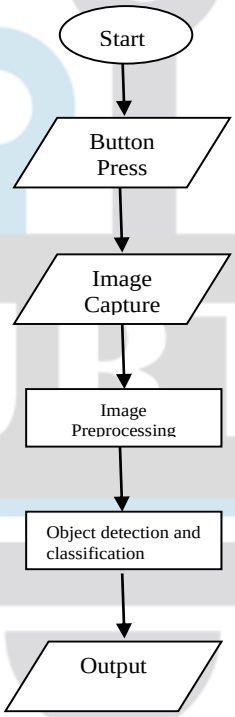

Fig.1 Mechanism flowchart of Object Detection

Image Datasets as input
Detecting the object as an output
1. Load the dataset
2. Use the mean and standard deviation to normalise the data.
3. Defining the weight and bias parameters
4. Constructing the computational model
5. Determine the loss function

6. Begin and complete the training phase
7. Assess the testing phase
8. As an output, detect the object.

The proposed system's procedure is designed in a cyclic manner using tensor flow. This cyclic algorithm is used to build the computation graph. We begin by feeding the dataset to the corresponding placeholders. The output of the computational model is then executed. The computed output is compared with the desired output, and the loss function is calculated.
The model will automatically perform back propagation and modify its variables based on the loss function.
Finally, check for the given criteria; if it is satisfied, we can stop here; otherwise, repeat the preceding steps until it meets the given criteria. The training dataset objects were saved in the file while the model was being trained.

B. Text Recognition and Extraction

Based on the analysis of the latest text processing Android application, can be considered in formulating the concept of application development of text processing applications with the implementation of Mobile Vision Text API. The Mobile Vision Text API uses a smartphone camera to process images and recognise existing text so that text appears on the phone screen in real-time. The ability to only scan and display text statically becomes a weakness that will be exploited in the future. In this application, a function that can change the Indonesian language to Javanese language or vice versa in certain recognised text will be developed, and the text will be displayed in real time and dynamically. As a result, a database containing text data is required to convert the language. This application's concept is to integrate text-based application architecture with the Android-based Mobile Vision Text API, which will become the translator application used by users as client and application web service that will act as an intermediary between database as server and client application.

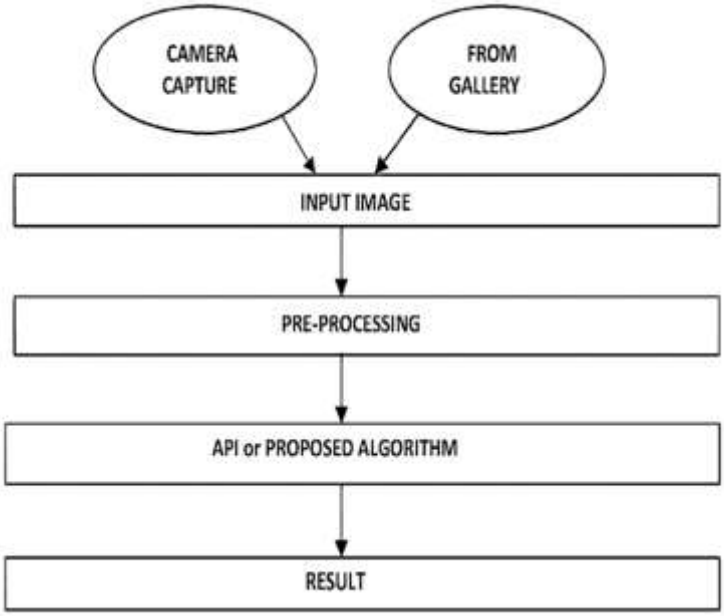

Fig.2 Text Extraction Flow Diagram

Figure 3 depicts the request and response process using the RESTful Web Services concept. The client sends HTTP requests with specific methods such as GET, POST, PUT, or DELETE. The server side receives and processes client-side requests, returning selected data in JSON or XML format [3].

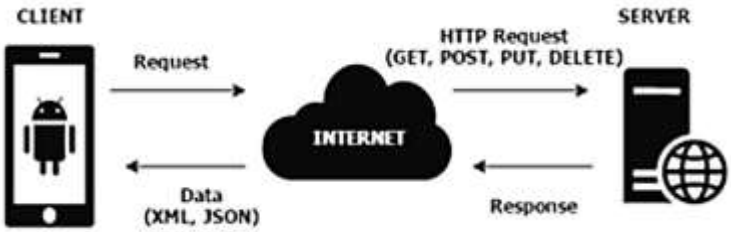

Fig.3 Workflow of RESTful Web Service

C. QR Code/ Barcode Scanning

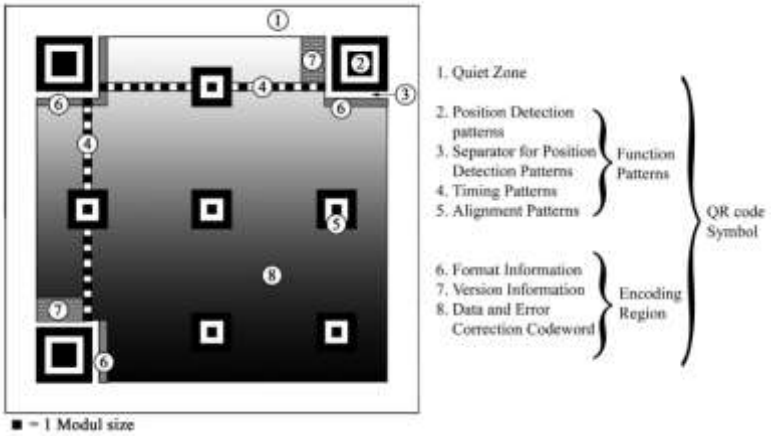

Fig.4 Structure of QR Code

Figure 4 depicts the structure of a QR code (Standardization, 2006). Every QR code is made up of dark (logical "1") and light (logical "0") modules. The modules are distributed evenly in a square net of fields, with each field the size of a single module. According to ISO/IEC18004, one module should be 4x4 px (pixels) with a print resolution of 300 dpi (dots per inch) [4].
QR codes and barcodes have become important components in enabling various technical solutions and are used by millions of people worldwide every day. With the massive increase in QR code usage and the growing number of applications, most of which involve sensitive tasks such as payment and ticketing, it is critical to understand the current state of the technology, its implementation, limitations, and future work scope.

Obtaining information from a QR code in real-world settings entails three critical steps:

1. Identification and localization
2. Image Preprocessing
3. Decoding

Identification and localization

The first step in the workflow of using QR codes and retrieving information is detecting and localising them. Hirata and Bellusi presented a method for detecting QR codes with high accuracy in a large database of arbitrarily acquired images. This is a highly cited and acclaimed work that many researchers have used in their research.

Image Preprocessing

After a QR code is detected, its image is processed to ensure optimal and accurate decoding. This is commonly referred to as pre-processing. Because of the QR code's real-world applications, pre-processing is essential not only to improve decoding accuracy but also to make it faster and simpler [5]. A. Jones demonstrated how some QR code images can be detected but not successfully decoded; he also presented a deblurring algorithm to improve recognition of low resolution images. This section will provide the reader with a comprehensive overview of the scope of pre-processing as well as the work done to overcome challenges such as blurriness, perspective difference in angle, low-resolution image, and general distortion.

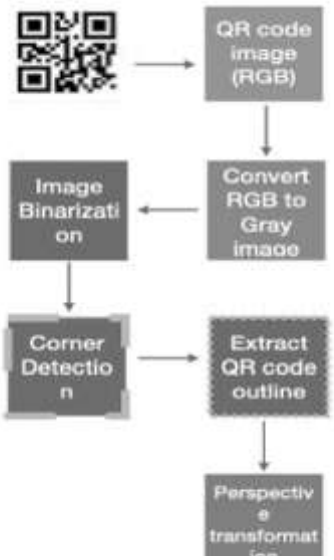

Fig.5 Image Preprocessing Flow Diagram

Decoding

Decoding is the final step in which information/data is retrieved, and it is based on the QR code's main standard architecture.

IV. RESULT AND CONCLUSION

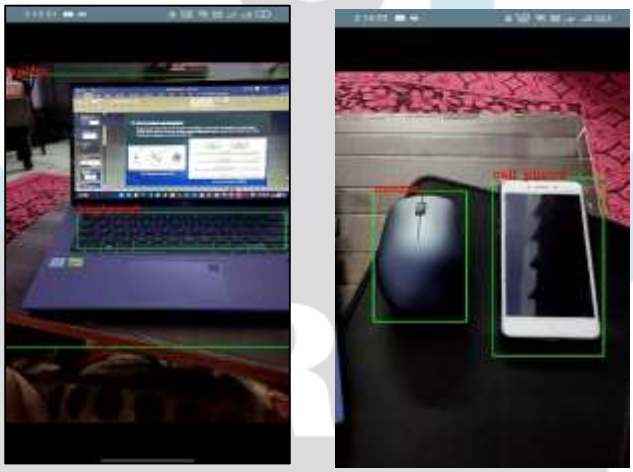

Fig.6 Working of Object Detection Module

Our neural network is trained in two steps using supervised classification learning:

1) For each object, a dataset of approximately 150 images is created.
2) The existing model is fine-tuned by training the object detector using the above dataset.

The preceding steps are repeated for each object that needs to be detected.

A high-level overview of fine-tuning the existing model with regard to custom objects is provided below:

1. Create a training set. This data is a large collection of images that have been resized to meet the specifications.
2. Produce an XML file that describes the objects in the images. This is known as labelling.
3. Convert all XML files to TFRecord records.
4. Creating the model's configuration file.
5. Train the model with the image TFRecord files and the configuration file.

TensorFlow has the following advantages over other frameworks:
1. Simple deployment.
2. Better GPU support when compared to other models.

3. It provides high-level APIs for modelling.
4. It is extremely simple to make radical and extreme changes.

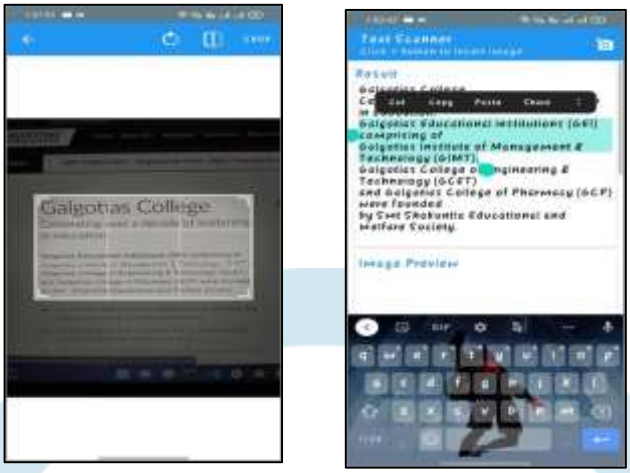

Fig.7 Working of Text Scanner Module

Google Vision API has a lot of potential, and it's getting better all the time. Its powerful Machine Learning and large image database contribute to its position as one of the best current market solutions. Landmark Detection has a bright future in labelling and organising photos taken on a trip. In terms of Face Detection, it will be useful in future research on people's emotions when exposed to a specific environment or product. There is still a significant disadvantage with this API. It cannot be used in real-time systems because communication with Google's servers introduces significant latency.

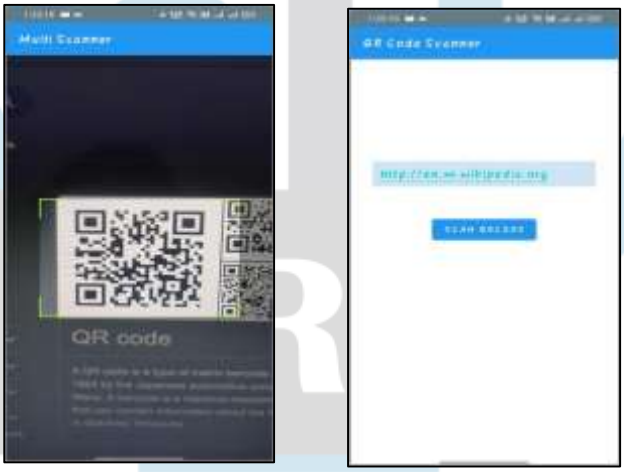

Fig.8 Working of QR Code Scanner Module

We used Zxing for QR/ Barcode Scanning because it is a cross-platform third-party barcode scanning library. Assisted Selling wraps the ZXing project in an isolated library (Barcode Scanner) and displays the ZXing interface. It is a Java-based barcode image processing library with portability to other languages. It supports 1D product barcodes, 1D industrial barcodes, and 2D barcodes. Zxing is an abbreviation for Zebra Crossing, and it is a popular open-source API for integrating QR (Quick Response) Code processing. It is a Java-based barcode image processing library with portability to other languages. It can read 1D product, 1D industrial, and 2D barcodes. Google uses web search to obtain millions of indexable barcodes from ZXing. It also serves as the basis for Android's Barcode Scanner app and is integrated into Google Product and Book Search. ZXing (pronounced "zebra crossing") is an open-source, multi-format 1D/2D barcode image processing library written in Java with portability to other languages. ZXing is also a Google open source project.

Conclusion

In less than two decades, mobile applications have grown rapidly. Information technology is permeating human life and playing an important role in changing the lives of so many people around the world by providing efficient, fast, consistent, and authentic tools in the form of internet and mobile applications.

This project is an Android Mobile Application, which is a mobile-based application with multiple scanning functionalities such as text extraction, QR/Barcode scanning and authentication from a real-time captured image using the camera or from saved images, and real-time object detection.
Moreover, This application will also be used to scan and extract information from objects, texts, or anything else.
This application has numerous applications. For example, it can be used to retrieve information from a business card and save it directly to the phone, or to retrieve text from an image captured by an Android device and save it for later use. This application will also make it easier to retrieve text from books and edit it, as well as convert images such as slides into notes. So, this will save us a lot of time because it extracts all of the text from the image and allows us to use it directly or modify it. Text scanners save time and labour by converting information from printed materials to digital data stores. For example, a document scanner can be used to recover, retrieve, and preserve old or valuable documents. Scanned documents are fantastic. They allowed us to archive stacks of paper into folders, which took up far less space and were infinitely easier to organise, move, and copy.
Our application can also validate any QR or Barcode by scanning it and retrieving relevant information or site details. QR codes are frequently used to track information about products in a supply chain, to trace coronavirus, and link directly to product pages online. For example, if you were looking for the exact dress a model was wearing in a poster, a QR code could take you directly to the web page where you could purchase it.
Our application can also be used to detect any object in real-time by accurately labeling them.

V. Future Scope

Due to time constraints, we only focused on text images and documents with no skew that contain Latin Alphabet during the implementation of this capstone project's Android-based devices. We were unable to extend our efforts to make our application process text documents in Arabic or Berber, for example. In the future, we hope to incorporate other languages such as Japanese, Chinese, Arabic, or any other language that does not use the Latin alphabet. We will also focus on the text with skew to ensure that the application is efficient and useful in the future.
In the future, we will also try to implement object detection functionality in a way that is more useful in day-to-day life in the form of a handy android application, as well as use object detection functionality for advanced purposes like fast motion detection, accurate counting of crowd, detection of anomalies etc.
As part of our future work on QR Barcode Authentication Functionality, we will add a quick payment facility for users, allowing them not only to authenticate any QR or Barcode of any product, but also to make quick and easy payments from their smart phones anywhere they want.

VI. Acknowledgment

The author would like to express our deepest gratitude for our mentor, Mrs. Anam Khan who guided us throughout this project with her technical skills and immense knowledge in this field.

References

[1] B.Y. Panchal, G. Chauhan, S.R. Panchal et al., An investigation on feature and text extraction from images using image recognition in Android, Materials Today: Proceedings, https://doi.org/10.1016/j.matpr.2021.06.237.
[2] G. Padmapriya, B. Santhosh Kumar, M.N. Kavitha et al., Deep convolutional neural network for real time object detection using tensor flow, Materials Today: Proceedings, https://doi.org/10.1016/j.matpr.2021.02.671.
[3] Fariz Dzulfiqar Nurzam, Emha Taufiq Luthfi, "Implementation of Real-Time Scanner Java Language Text with Mobile Vision Android Based" 2018 International Conference on Information and Communications Technology (ICOIACT), IEEE.
[4] Laslo Tarjan, Ivana Šenk, Srdjan Tegeltija, Stevan Stankovski, Gordana Ostojic, "A readability analysis for QR code application in a traceability system", http://dx.doi.org/10.1016/j.compag.2014.08.015 0168-1699/ 2014 Elsevier.
[5] V. Jain, Y. Jain, H. Dhingra, D. Saini, M.C. Taplamacioglu, M. Saka, "A SYSTEMATIC LITERATURE REVIEW ON QR CODE DETECTION AND PRE-PROCESSING", International Journal on "Technical and Physical Problems of Engineering" (IJTPE), Iss. 46, Vol. 13, No. 1, Mar. 2021.