Enhancing Efficiency through Barcode/QR Code Reader

## 1stLeann Joby Mathew

Department of Artificial Intelligence And Data Science ,College of Saveetha Engineering (Chennai) [leannjoby@gmail.com](mailto:leannjoby@gmail.com)

## 2stYogeshvar M

Department of Artificial Intelligence And Data Science ,College of Saveetha Engineering (chennai) [yogimanickam7988@gmaill.com](mailto:yogimanickam7988@gmaill.com)

## 3rd Augustine J

Department of Artificial Intelligence And Data

Science ,College of Saveetha Engineering (Chennai) [jaugustine0306@gmail.com](mailto:jaugustine0306@gmail.com)

**Lavanya G**, Associate Professor, Department of artificial intelligence and machine learning, Saveetha Engineering College, Chennai, India, glavanya@saveetha.ac.in

**Archana S H,** Assistant Professor/Department of artificial intelligence and data science, Saveetha Engineering College, Chennai, India, archana.s.h@saveetha.ac.in

***Abstract*— Computer Vision and Machine Learning with OpenCV or Open-Source Computer Vision Library, is a powerful open-source library to provide its services to computer vision. It contains a large number of features and functions for image processing and real time camera interaction. By integrating OpenCV with Python—a high level, interpreted, object-oriented programming language it makes for a very usable, feature real barcode and QR code reader application.**

**Pyzbar is a very important Python library that allows for the detection and decoding of barcodes and QR codes in many applications. It reads a bunch of different types of barcodes too, QR codes, UPC, EAN, Code 128, Code 39, etc. Because this flexibility enables programmers to deal with many kinds of code in one.**

**The system provides functionality for managing training data by extracting and saving QR/barcode data into a CSV file. Users can also manually add new training data and view the existing dataset, helping in building a robust collection for machine learning purposes.**

**Keywords: OpenCV, Convolutional Neural Network, Pyzbar**

**I. Introduction**

Quick and accurate data capture is simplified and human errors are reduced allowing processes like checkout and stock tracking to move quickly. Back in 2017, QR codes are 2D codes, creating a storage surface for a lot more data: URLs, text, multimedia. This versatility makes them all the better suited to many different applications, for example digital payments and secure access authentication, marketing campaigns and customer engagement. Some examples of where QR codes are commonly used in marketing for website linking to, in ticket verification in events and in healthcare for patient record and medication tracking.

As these codes are widely being used and are extremely useful, this paper presents a web-based application that uses Streamlit, OpenCV, and Pyzbar libraries to decode and detect barcodes and QR codes from both still images and live video feeds. The tool that has been developed offers an intuitive interface, that allows users to upload images or use the live webcam feed to scan for codes. Furthermore, the application provides user experience enhancement by providing functionalities to open detected codes in Google / Amazon for quick product search.

### 

### II. RELATED WORK

This project utilizes Python to address the inefficiencies of manual data entry in inventory management by automating the barcode reading process. The project employs methodologies such as image processing with OpenCV for capturing and processing images, and barcode decoding using the pyzbar library to extract encoded information from various barcode formats.[1]

Noor et al proposes a facility that introduces a security framework which provides a protected mechanism ensuring secure communication, authentication, confidentiality, and safeguarding of financial transactions. The paper concludes that the integration of QR codes with a hybrid AES-ECC algorithm significantly enhances the security and efficiency of cardless data transactions in mobile applications.[2]

Raja et al offers a model discussing the importance of integrating barcode scanning capabilities into mobile applications, particularly within the Salesforce ecosystem. This paper specifically focuses on Salesforce mobile applications, suggesting that enhancing these apps with barcode scanning functionality can significantly improve user experience.[3]

Arju et al provided a paper that provides an efficient method for recognizing and detecting barcode and QR codes together. The method automatically detects the Barcode or QR code and displays the complete information of the product. It is developed in a python environment using the OpenCV library.[4]

Qui et al’s idea reveals a rapid way to accomplish anti-counterfeit verification with a barcode. An anti-counterfeit numerical code is converted to a one-dimensional barcode using the original anti-counterfeit technology. The one-dimensional barcode is then hidden using blocking or other methods, and once the hiding mode is removed by an anti-counterfeit verifier,[5]

Liao et al proposed a barcode scanning parameter learning method, where several light emitting elements, in accordance with predetermined light emitting parameters, generate light sources one by one.[6]

Oni et al built a model which scans the barcode or QR code and displays the information on LCD. The advantage is that this model works in the dark, and has a stable and accurate performance with fast decoding speed.[7]

Burak et al proposed a medical device tracking based on clustering method which keeps track of which device belongs to which hospital and under whose responsibility, device manufacturing date etc. This QR or Barcode system is scanned via smartphones or tablets, and will be a useful application for analyzing and evaluation of inventory, efficiency and performance [8]

L. Zhang et al proposed a method that leverages YOLO's capabilities as a single-stage object detector, which provides real-time performance by simultaneously performing object detection and localization in one go.[9]

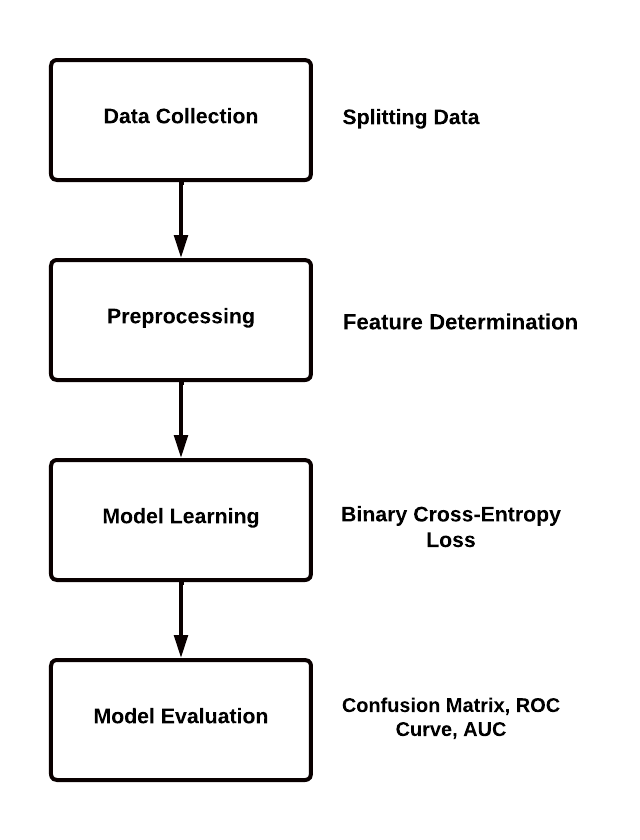
M.S. Patel et al provides a detailed overview of methods used for decoding barcodes and QR codes in images and video streams. It discusses traditional algorithms, machine learning approaches, and deep learning techniques, emphasizing their strengths and limitations. [10]

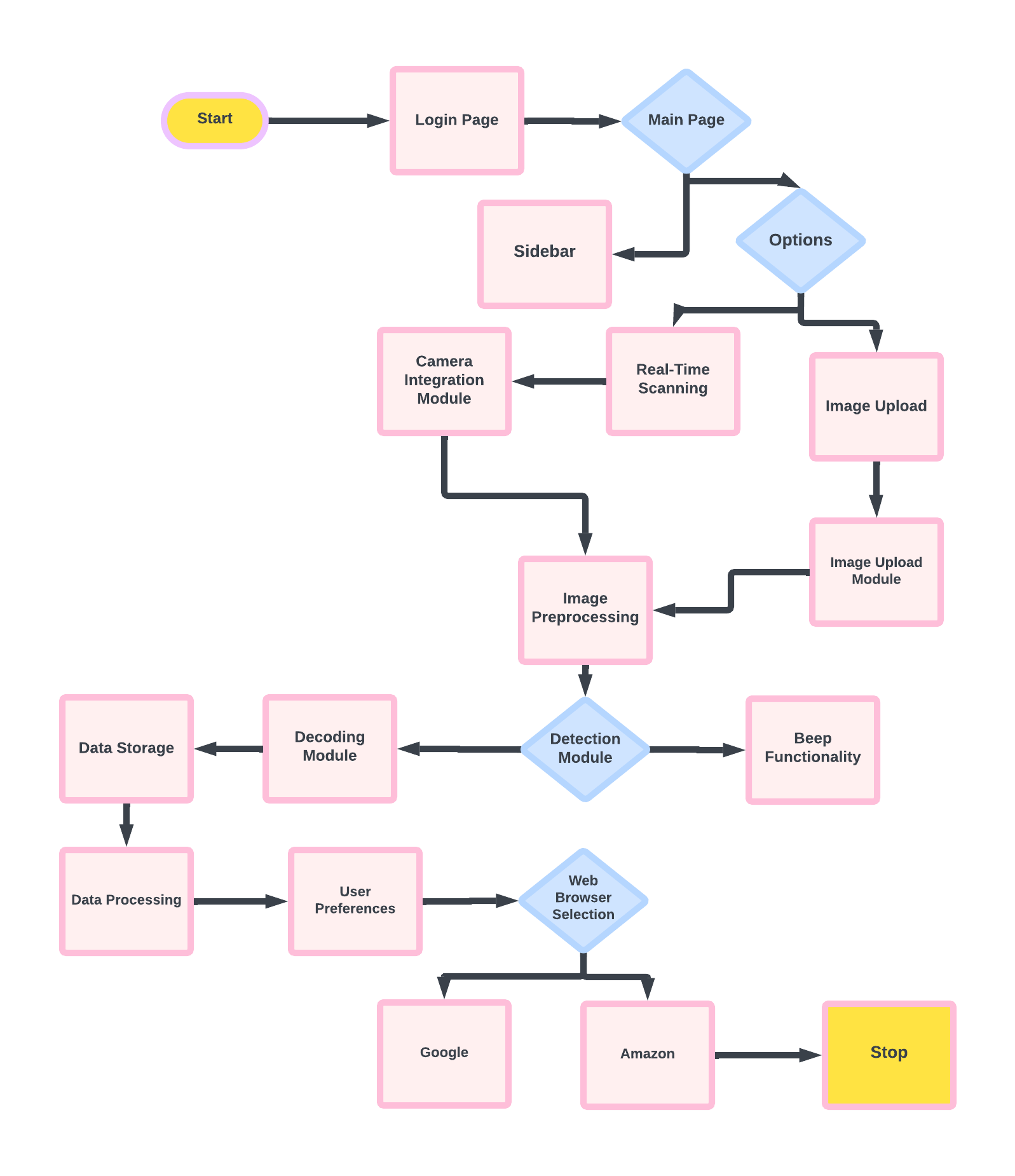
**III. METHODOLOGY**

**Data collection,** where we gather a well-defined dataset, ensuring it encompasses relevant features and target variables.Next, we perform **Data preprocessing**, which includes cleaning the dataset by handling missing values, encoding categorical variables, and normalizing numerical features as needed.

**Split it into training and testing** subsets to evaluate model performance effectively.

**Evaluate the model** using metrics such as Confusion Matrix, ROC Curve, AUC and Precession Recall Curve to assess its performance on unseen data. Finally, we **save the trained model** for future predictions.





**IV. RESULTS**

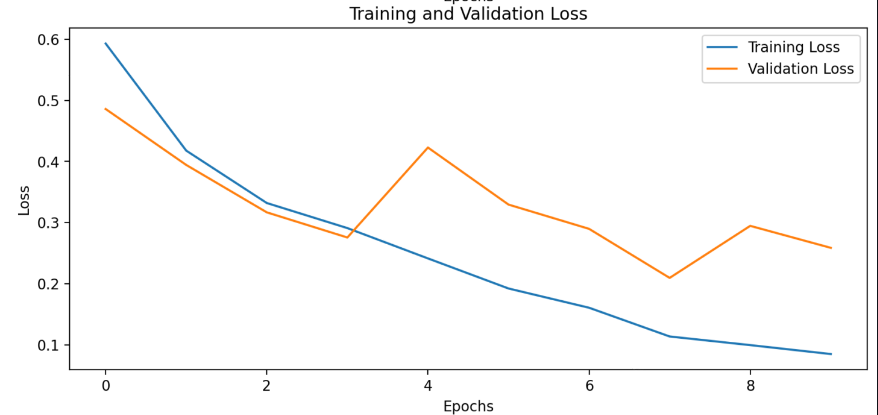


**Overall Accuracy Trends:**

The blue line representing training accuracy shows a steady increase over the epochs, reaching close to 100%. This suggests that the model is learning well from the training data.

The orange line representing validation accuracy, while more volatile, also shows improvement and stabilizes around 90 -92%, indicating a relatively good generalization ability of the model to unseen data.

The graph shows 9 epochs (from 0 to 9) on the x-axis.



**Volatility in Validation Curves:**

The validation accuracy for each fold fluctuates more than the training accuracy, with sharp rises and falls across the epochs. This fluctuation is common, especially when using dropout, which adds randomness during training and can result in some variability in validation performance. The

The sharp fluctuations suggest that the validation accuracy

is more sensitive to noise or minor variations in the data.

**Training and Validation Gap:**

There is a small gap between the training and validation

accuracies. Training accuracy is consistently higher than

validation accuracy across all folds, suggesting that the

model performs better on the data it has seen during

training.

However, the gap is not very large, implying that the

model is not heavily overfitting. This is likely due to the

use of dropout and batch normalization, both of which are

regularization techniques aimed at reducing overfitting.

**V. CONCLUSION**

This Streamlit app provides a convenient interface for training and evaluating a convolutional neural network (CNN) to classify QR codes and barcodes from image datasets. Users can upload CSV files containing image paths and corresponding labels, and the app preprocesses the images by resizing and normalizing them. It builds and trains a simple CNN model with two convolutional layers, tracks training progress, and evaluates performance on a test set. The app also visualizes key metrics such as accuracy, loss, confusion matrix, ROC curve, and precision-recall curve, offering a comprehensive overview of the model's performance. This tool simplifies the process of QR/barcode classification with real-time feedback and easy-to-interpret results.

**VI. REFERENCE**

[1] Noor, Hamad., Abbas, Abdulazeez, Abdulhameed., Mudhafar, H., Ali. (2023). Enhancing Security and Efficiency through QR Integration with Hybrid AES-ECC Algorithm in Mobile Apps for Cardless Data Transactions. Al-Iraqia Journal of Scientific Engineering Research, 2(4) doi: 10.58564/ijser.2.4.2023.124

[2] Raja, Patnaik. (2022). Enhancing Mobile App Functionality and User Experience With BarcodeScanner API. doi: 10.47363/jaicc/2022(1)301

[3] Arju, Aman., Aryan, Singh., Ayush, Raj., Sandeep, Raj. (2020). An Efficient Bar/QR Code Recognition System for Consumer Service Applications. 127-131. doi: 10.1109/ZINC50678.2020.9161778

[4] Qiu, Yi. (2017). Method of using barcode for realizing anti-counterfeit verification quickly.

[5] Liao, Deng-Zhen. (2015). Optimized barcode scanning parameter learning method of barcode reader, and barcode reader.

[6] Oni, O., A.., A., O., Musibau., G., A., Wahab., A., G., Koleoso. (2024). Building of Barcode and QR Code Reader Device. 4(1):105-111. doi: 10.52589/ajste-opzmzn49

[7] Burak, Tiryakioglu., Gulgun, Kayakutlu., Irem, Duzdar. (2016). Medical device tracking via QR code and efficiency analyze. 3115-3128. doi: 10.1109/PICMET.2016.7806549

[8] L. Zhang, Y. Liu, and T. Chen, "Real-Time Barcode and QR Code Detection Using YOLO and OpenCV," 2021.

[9] M. S. Patel and A. R. Singh.Barcode and QR Code Decoding in Images and Video Streams: A Comprehensive Review -2019

[10] Wang, Y., Zhang, H., & Liu, X. (2022). Efficient Barcode and QR Code Detection with Convolutional Neural Networks.

[11] Bai, Y., & Zhang, H. (2020). A novel method for barcode recognition based on machine learning. *Journal of Optical Communications and Networking*, 12(4), 257-265. doi:10.1364/JOCN.12.000257

[12] Tseng, S. S., & Lin, Y. H. (2018). QR Code-based interactive learning in a mobile environment: A study of the impacts on student learning engagement. *Computers & Education*, 127, 1-10. doi:10.1016/j.compedu.2018.08.005

[13] Aman, A., & Ali, A. (2021). A framework for secure mobile payment system using QR code technology. *International Journal of Information Security*, 20(2), 103-115. doi:10.1007/s10207-020-00523-6

[14] Feng, Z., & Wang, Y. (2019). Barcode detection based on deep learning. *IEEE Transactions on Image Processing*, 28(8), 4122-4133. doi:10.1109/TIP.2019.2894741

[15] Abdulrahman, A. A., & Ali, S. H. (2021). Secure data transmission using QR codes and blockchain technology. *Journal of Information Security and Applications*, 58, 102758. doi:10.1016/j.jisa.2021.102758

.