HELPING BLIND INDIVIDUALS: PAKISTANI CURRENCY RECOGNITION SYSTEM FOR BLIND PEOPLE USING THE YOLOv8 MODEL

Aqsa MUSHTAQ [1], Hamza Wazir KHAN [2], Muhammad SALMAN [3], Fazeel ABID [4], Jawad RASHEED [5, 6, 7, 8,*], Shtwai ALSUBAI [9,*]

Blind or visually impaired individuals face significant challenges in recognizing currency notes, especially in regions where cash transactions remain common. To address this issue, we propose a real-time currency detection system tailored for Pakistani currency notes. The system is powered by the YOLOv8 object detection model, which achieved 99.1% accuracy with an average inference time of 12 milliseconds per image, ensuring both precision and speed. Once the denomination is identified, it is communicated to the user via text-to-speech for audible feedback. The application is structured with a Streamlit-based front-end for user interaction and a Flask-based back-end API, deployed via NGROK to ensure secure, accessible cloud usage. This architecture allows real-time operation across devices without local installations. In addition to implementation, we conducted a comparative evaluation of YOLOv8 against VGG19 (val-accuracy 83%), highlighting YOLOv8's superior performance in terms of accuracy and inference speed across varied lighting and orientation conditions. The results reinforce YOLOv8's suitability for assistive technologies and provide a benchmark for future improvements in AI-powered accessibility tools.

Keywords: YOLOv8, VGG19, Pakistani Currency notes, Blind People, Image classification, Machine Learning, Deep Learning, Computer Vision

1. Introduction

Cash plays a pivotal role in our daily lives, facilitating essential activities such as purchasing goods and services, conducting small-scale business

[1] Department of Business Studies, Namal University, Mianwali, Pakistan, e-mail: aqsamushtaq2020@namal.edu.pk
[2] Department of Business Studies, Namal University, Mianwali, Pakistan, e-mail: hamza.wazir@namal.edu.pk
[3] Department of Computer Science, Namal University, Mianwali, Pakistan, e-mail: salmanbinarif@gmail.com
[4] Department of Computer Science and Information Technology, University of Lahore, Lahore, Pakistan, e-mail: fazeel.abid@cs.uol.edu.pk
[5*] Department of Computer Engineering, Istanbul Sabahattin Zaim University, Istanbul, Turkey, *Corresponding author, e-mail: jawad.rasheed@izu.edu.tr
[6] Department of Software Engineering, Istanbul Nisantasi University, Istanbul, Turkey
[7] Research Institute, Istanbul Medipol University, Istanbul, Turkey
[8] Applied Science Research Center, Applied Science Private University, Amman, Jordan
[9] Department of Computer Science, College of Computer Engineering and Sciences in Al-Kharj, Prince Sattam Bin Abdulaziz University, Al-Kharj, Saudi Arabia, e-mail: sa.alsubai@psau.edu.sa

transactions, and enabling swift exchanges [1]. In earlier times, societies relied on barter systems. Still, in this era, the absence of physical cash would severely impede our ability to perform routine tasks, especially in Pakistan. Banknotes serve as the go-to currency for minor transactions, whether for paying fares or buying items from local stores, providing a tangible, universally accepted means of exchange [2]. Despite alternative payment methods such as bank transfers, ATM transactions, and digital wallets like Jazz Cash, cash remains widely used due to its convenience [3]. However, this reliance on physical currency also exposes individuals to the risk of fraud. According to 2022 statistics, the Federal Investigation Agency (FIA) received over 100k complaints of financial fraud, with more than 40% involving monetary losses [4]. According to a survey by the Health Ministry of Pakistan, over 9 million people in Pakistan have some degree of vision impairment, ranging from mild to blindness [5]. On the other hand, the all-age prevalence of total blindness and vision impairment has steadily increased from 1990 to 2017, with projections indicating a further rise by 2025 [6].

After analyzing the stats, we know that a large portion of the population depends on others for routine tasks. It is undoubtedly troublesome for blind or visually impaired individuals to identify banknotes. However, banknotes feature Braille according to their denomination [7]. But over time, these features wear down, and blind people cannot distinguish banknotes. There is a need for a system that accurately recognizes banknotes; it will automatically improve the lives of these individuals [8]. Technologies are moving towards smartphones for blind people, including currency recognition systems to help them in financial transactions [9]. There are multiple methods to solve this problem. This paper is based on image classification using YOLOv8 (You Only Look Once) version 8 [10], [11], [12]. Fig. 1 shows the proposed scheme. First, the system will be given images of Pakistani currency notes, and it will respond with the denomination of each. This system implements transfer learning, a deep learning domain in which an already-trained model is used to train a new model on different datasets. Furthermore, text output will be converted to speech (audio feedback) to assist blind people. The actual system is the same as shown in Fig. 1. We include a Streamlit-based front-end and a Flask API back-end using NGROK for testing, demonstrating the system's modularity and scalability.

1.1. Research Questions

- Which methods can implement a versatile currency recognition system?
- How does YOLOv8 perform on the currency dataset?
- Which CNN model performs well on a small dataset?

1.2. Research Objectives

- Enhance financial independence by creating a universally accessible solution for blind or visually impaired individuals to manage currency confidently.

- Implement a versatile recognition system that adapts to diverse environmental conditions and reliably identifies Pakistani Banknotes.
- Research and implement cutting-edge technologies to expand the scope and effectiveness of the currency detection system for blind or visually impaired individuals.

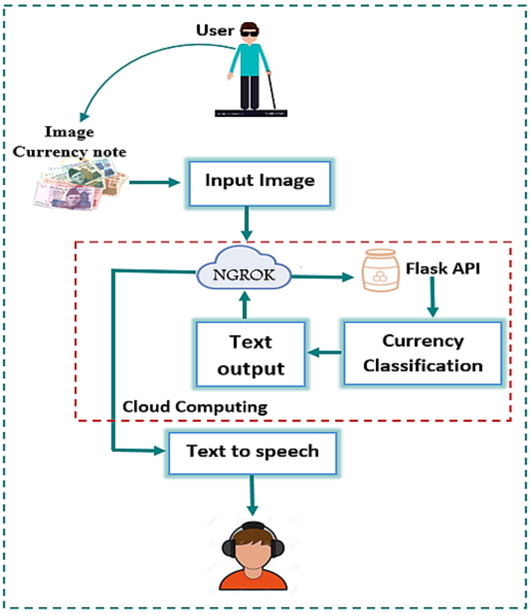

Fig. 1. The block diagram of the proposed system

A currency detection system can assist blind people with financial transactions [13]. It will reduce their dependence on others. The computer vision branch of artificial intelligence helps us develop smart technologies to solve real-world problems [8]. Machine learning and deep learning models help us develop assistive technologies for the blind and visually impaired community [14].

Traditional currency authentication relied heavily on specialized hardware sensors detecting magnetic ink, ultraviolet features, and infrared signatures [15]. However, the advent of advanced image processing and machine learning techniques has enabled software-based solutions that leverage standard cameras and computational resources. The recent explosion in deep learning capabilities, particularly convolutional neural networks (CNNs), has revolutionized this field by achieving human-level or superior performance in banknote recognition tasks [16]. Deep learning eliminated the need for manual feature engineering by automatically learning hierarchical representations directly from raw pixel data.

Recent research has explored hybrid architectures combining the strengths of deep learning and classical methods. These approaches typically use CNNs as feature extractors, feeding learned representations into traditional classifiers, such as researchers in [17] reported 99.11% accuracy in Kazakh banknote classification using ResNet-18 features with traditional ML classifiers.

VGG architectures, particularly VGG16, have demonstrated strong performance in banknote classification and counterfeiting detection tasks. Hornig et al. achieved 98.08% accuracy for denomination classification and 97.95% for fake detection on a dataset of approximately 2,572 images across six Indian denominations [18]. The VGG architecture's deep, uniform structure, combined with small 3×3 convolutional filters, enables effective learning of fine-grained texture patterns critical for distinguishing genuine from counterfeit notes.

Preprocessing images is very important for achieving high model accuracy. Some important pre-processing techniques include image blurring and filtering, such as contrast, brightness, enhancement, and sharpening. Ahmed Yousry [1] uses ORB image Pre-processing techniques built on top of the OpenCV Library. He used the Gaussian blurring method to blur the currency notes where needed. After blurring the image, he sharpens the image to highlight the boundaries.

Various CNN models have been used in past currency detection systems. Regarding Pakistani currency notes, Muhammad Imad [13] proposed a system that helps blind individuals recognize currency using the AlexNet architecture and a Support vector machine (SVM). He trained the model on six different classes. The SVM model's accuracy was 96.85%.

Various deep learning models have been explored for currency classification, including VGG16 [12], [17], [18], VGG19 [17], AlexNet [12], [18], MobileNet [17], ResNet50 [17], NesNet Large [17], and YOLO [10], [19], [10], [12], [13], [19], [20], [21], [22]. The models with the highest accuracy were VGG19 and MobileNet, achieving 91.85% and 96%, respectively [19]. The YOLOv8 model trained on Malaysian and Nepali banknotes [22] performs well compared to SSD (Single-shot detector). The YOLO series of models by Ultralytics presents an exciting prospect for the future of currency detection. Known for their speed and efficiency, YOLO models have shown promising results in studies using YOLOv8 for Malaysian and Nepali banknotes [21]. While not yet tested with Pakistani currency, these findings suggest that YOLO's potential warrants further investigation to achieve even greater accuracy in this domain.

By harnessing the power of deep learning and integrating it into smartphone technology, we can empower blind and visually impaired individuals to manage their finances with greater independence and confidence [23].

2. Proposed Methodology

Designing and implementing a currency detection system for visually impaired individuals involves key steps, including data collection, Preprocessing, model training, and system implementation. The proposed framework aims to recognize various currency denominations through a web-based testing application that provides audible feedback.

The Model will receive pre-processed data of Pakistani currency notes. The image will undergo preprocessing techniques from the OpenCV library, such as blurring (Gaussian, median, bilateral), brightening, grayscale conversion, and smoothing. The preprocessing techniques highlight the features of currency notes, remove noise, and help the model recognize them more accurately. Some preprocessed images of currency notes are shown in Fig. 2.

Fig. 2. The image of Pakistani currency when processed with different pre-processing techniques: smoothened currency note, brightened currency note, blurry currency note, and gray-scaled currency note.

In Fig. 2, Image A shows a smoothened currency note; Image B shows the output of a brightening technique; Image C shows a Blurry Currency note; and Image D shows a gray-scaled currency note. After the images are preprocessed, the dataset is given to the pre-trained Models (YOLOv8, VGG19) for Pakistani currency classification. After the model is trained on the given dataset and achieves high accuracy, its text output is converted to speech. Blind or visually impaired people will receive audio feedback to assist them with daily financial transactions. The complete proposed methodology is shown in Fig. 3.

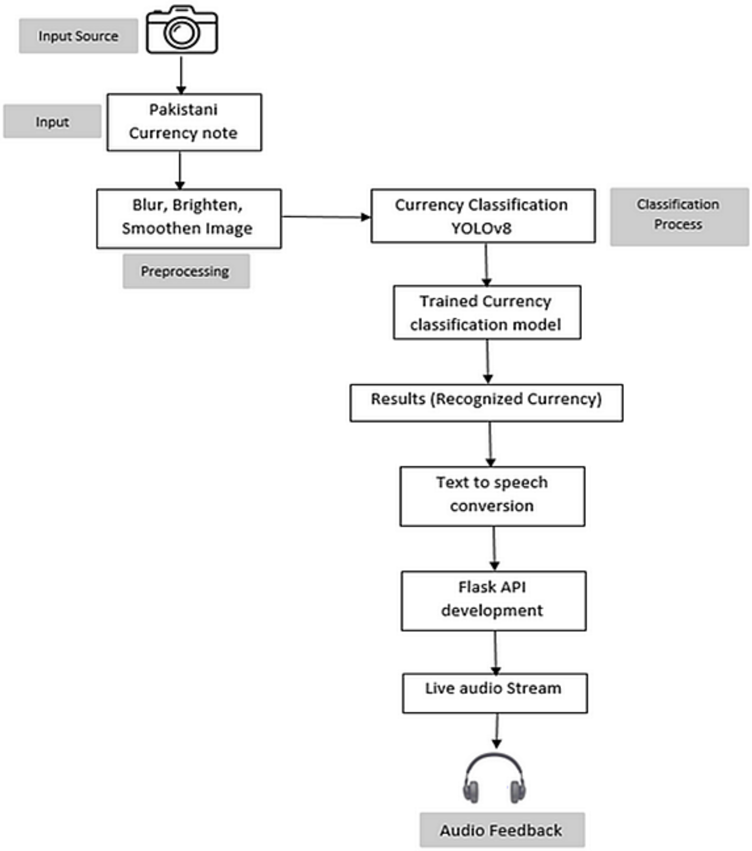

Fig. 3. The proposed methodology of the suggested scheme.

2.1. Proposed Framework

The core objective of this project is to classify currency notes from input images. When an image containing a currency note is provided, the system identifies and outputs its corresponding class label from the predefined categories: Note-10, Note-20, Note-50, Note-100, Note-500, Note-1000, Note-5000, and Invalid class. Fig. 4 shows the sample images from the dataset.

Fig. 4. The sample images for worn and damaged currency notes.

3. Data Preparation and Preprocessing

This study utilizes both primary and secondary datasets for the currency detection system. The final dataset consists of approximately 4,600 cleaned and labeled images, representing eight classes—seven Pakistani currency denominations (Note-10, Note-20, Note-50, Note-100, Note-500, Note-1000, Note-5000) and one Invalid class. Originally comprising 6,000 images, the dataset was manually cleaned by removing low-quality or misleading images to improve model performance. To ensure robustness in real-world scenarios, images were captured under various lighting conditions and angles and also included worn and damaged currency notes. For training purposes, each class contains 400 images for training, 112 for validation, and 56 for testing.

For supervised learning, we organize the dataset into distinct folders. Within this folder structure, the main data directory contains separate subdirectories, each representing a currency note class in the dataset. These subdirectories are labeled by their corresponding class (e.g., 100_class) and contain all images for that class. Each image file is given a unique name and typically uses a standard image format such as JPEG or PNG. Using this approach, we partition the dataset into training, validation, and test sets based on folder structure. Specifically, we divide the data into 70% for training, 20% for validation, and 10% for testing. The model's performance is evaluated on new image instances during testing. Table 1 outlines the dataset's sample distribution of each class for each set.

Table 1

The dataset train-test split for each class in the Pakistani currency annotation

Class Name	Training	Validation	Test
10_Class	400	112	56
20_Class	400	112	56
50_Class	400	112	56
100_Class	400	112	56
500_Class	400	112	56
1000_Class	400	112	56
5000_Class	400	112	56
Invalid/Fake	400	112	56
Total		4600	

4. Model Description

This section provides a detailed description of the YOLOv8 model and the VGG-19 architecture and explains how they are used in this study.

4.1. YOLOv8 Model Description and Working

YOLOv8 utilizes a deep CNN architecture similar to its predecessors but introduces novel features and enhancements. These include implementing a new backbone architecture, CSPNet, which offers improved efficiency and accuracy compared to previous backbones.

YOLOv8 incorporates a range of methods for object detection in images, including classification, object detection, and image segmentation. Each technique employs unique strategies to identify and locate objects within pictures.

In classification, the goal is to assign a label to the entire image indicating its primary category or class. This process helps in understanding the overall content or context depicted in the image without pinpointing each object's exact location. This approach proves beneficial for tasks where a broader understanding of the image's content is essential, even if precise object localization is not required. The structure incorporates a customized CSPDarknet53 backbone, substituting the CSPLayer in YOLOv5 with the C2f module. A spatial pyramid pooling fast (SPPF) layer aggregates features into a uniform-sized map to expedite computation. Every convolutional layer uses batch normalization and the SiLU activation function. Additionally, the head is decoupled to handle object detection, classification, and regression tasks separately. The YOLOv8 loss function combines localization, confidence, and classification losses. The classification loss specifically measures the error in predicting the class probabilities.

Classification Loss: The classification loss is typically the cross-entropy loss between the predicted class probabilities and the true class labels. For a grid cell (i,j) with a bounding box b, the classification loss can be as given in (1).

$$L_{class} = \sum_{i=0}^{S^2} \sum_{b=0}^{B} 1_{ij}^{obj} \sum_{c=1}^{C} -y_{ijb}^{(c)} \, log(\hat{p}_{ijb}^{(c)}) \tag{1}$$

Where:

- 1_{ij}^{obj} an indicator function that is 1 if the object is present in the cell (i,j) and 0 otherwise.
- $y_{ijb}^{(c)}$ is the true label (0 or 1) for class c for the bounding box b in cell ((i,j).
- $\hat{p}_{ijb}^{(c)}$ is the predicted probability for class ccc for the bounding box b in cell (i,j).

Working on YOLOv8 in a currency detection system for blind people, after dataset cleaning and annotation, is as follows: Once we organized the data, we used a pre-trained YOLOv8 model for 30 epochs. When YOLOv8 receives an image, it divides it into a grid, usually 13×13 or 26×26, depending on the model. Each grid cell then tries to predict objects within its area. Next, the model uses a deep convolutional neural network (CNN) to extract important features from the image, often using architectures from well-known models like ResNet or Darknet. During prediction, YOLOv8 estimates the bounding box's top-left corner coordinates, width, and height to predict object locations. It also calculates a confidence score to estimate the likelihood that the predicted box (see Fig. 5) contains the object. This helps YOLOv8 get better at detecting objects.

Fig. 5. The output of the proposed system when tested with currency notes to classify those with the correct label.

YOLOv8 captures the essence of its object detection mechanism: the combined probability score for a class c and bounding box b. This equation effectively combines the object score and the class probability prediction, forming the basis of YOLO's detection process given in (2).

$$P(c) = P(object). P(c \vee object) \tag{2}$$

Where P(object) is the confidence score that an object is present in the bounding box, and P(c|object) is the probability of class c, given that an object is in the bounding box.

This equation concisely captures YOLO's core idea: predicting both an object's presence and its class for each bounding box, enabling efficient and accurate object detection and classification. For each bounding box, the class probabilities are predicted as in (3).

$$P(c_i \vee object) = \frac{e^{p_{c_i}}}{\sum_{j=1}^{c} e^{p_{c_j}}} \tag{3}$$

Where, (Pci) is the predicted logit for class i, C is the total number of classes, and P(ci|object) represents the probability that the object belongs to class iii given the bounding box contains an object.

4.2. VGG-19 Architecture

As introduced in the seminal work by Simonyan and Zisserman, VGG-19 was proposed in 2014 by the Visual Geometry Group at the University of Oxford. In their seminal paper “Very Deep Convolutional Networks for Large-Scale Image

Recognition", they demonstrated that increasing depth with small convolutional filters significantly improves image classification accuracy.

VGG-19 is one of the deeper configurations presented, with 19 weight layers in total – specifically 16 convolutional layers followed by 3 fully-connected layers. All convolutional layers use small 3×3 kernels (stride = 1, padding = 1) to preserve spatial resolution. Rectified Linear Unit (ReLU) activations are applied after every convolutional and fully-connected layer. Each group of convolutional layers (a "block") is followed by a 2×2 max-pooling layer (stride 2) that halves the spatial dimensions. It processes a 224×224 RGB image through a sequence of five convolutional blocks, each followed by 2×2 max-pooling, and concludes with three fully connected layers. The architecture consists of 16 convolutional layers using 3×3 filters (stride 1, padding 1) and 3 dense layers, totaling 19 weight layers. Thus, the full VGG-19 sequence is:

- Block 1: Conv(3×3,64) → ReLU; Conv(3×3,64) → ReLU; MaxPool(2×2, stride=2).
- Block 2: Conv(3×3,128) → ReLU; Conv(3×3,128) → ReLU; MaxPool(2×2).
- Block 3: Conv(3×3,256) → ReLU (×4 consecutive conv layers); MaxPool(2×2).
- Block 4: Conv(3×3,512) → ReLU (×4); MaxPool(2×2).
- Block 5: Conv(3×3,512) → ReLU (×4); MaxPool(2×2).
- Fully connected layers: FC1 (4096 units, ReLU); FC2 (4096 units, ReLU); FC3 (1000 units, softmax).

Each convolutional block (except Block 1) increases the channel depth (64→128→256→512→512) while the 2×2 pooling halves the spatial resolution. The network uses 3×3 conv filters with a stride of 1 and a padding of 1 throughout, a design choice that keeps the architecture uniform and the receptive field incremental. Notably, all hidden layers use ReLU nonlinearities and no local response normalization (LRN), simplifying the model.

VGG-19 is a very large model: it contains approximately 144 million trainable parameters. The majority of these parameters reside in the three fully connected layers (especially the 4096×4096 weights). The deep yet homogeneous structure (5 conv blocks + 3 FC) made VGG-19 a standard benchmark, and its pretrained weights have been widely used in computer vision research.

5. Experimental Results

For result analysis, we've evaluated a range of performance metrics, including accuracy, training loss, validation loss, and a confusion matrix. The accuracy of a model is the proportion of true instances it correctly predicts as shown in (4).

$$Accuracy = \frac{TruePositive}{TotalInstances} \tag{2}$$

5.1. Performance of VGG19 for Currency Classification

The VGG19 model is trained on Google Colab Pro, with a batch size of 32 and for 50 Epochs. We set Stochastic Gradient Descent (SGD) as the optimizer, with learning rate 1e-4, momentum 0.9, and decay 1e-6. The model results are shown in Table 2.

Table 2

Epochs summary of VGG19

```
Epoch 23/50
147/147 ━━━━━━━━━━━━━━━━━━━━ 205s 1s/step - accuracy: 0.8796 - loss: 0.6343 - val_accuracy: 0.8209 - val_loss: 0.8041 - learning_rate: 1.0000e-04
Epoch 24/50
147/147 ━━━━━━━━━━━━━━━━━━━━ 206s 1s/step - accuracy: 0.8946 - loss: 0.5605 - val_accuracy: 0.7781 - val_loss: 0.9597 - learning_rate: 1.0000e-04
Epoch 25/50
147/147 ━━━━━━━━━━━━━━━━━━━━ 182s 1s/step - accuracy: 0.9020 - loss: 0.5405 - val_accuracy: 0.8003 - val_loss: 0.8835 - learning_rate: 1.0000e-04
Epoch 26/50
147/147 ━━━━━━━━━━━━━━━━━━━━ 176s 1s/step - accuracy: 0.8872 - loss: 0.5774 - val_accuracy: 0.7438 - val_loss: 1.0646 - learning_rate: 1.0000e-04
Epoch 27/50
147/147 ━━━━━━━━━━━━━━━━━━━━ 178s 1s/step - accuracy: 0.8894 - loss: 0.6011 - val_accuracy: 0.8235 - val_loss: 0.8463 - learning_rate: 1.0000e-04
Epoch 28/50
147/147 ━━━━━━━━━━━━━━━━━━━━ 175s 1s/step - accuracy: 0.9229 - loss: 0.4784 - val_accuracy: 0.8372 - val_loss: 0.9248 - learning_rate: 1.0000e-04
```

The training accuracy of VGG19 for Pakistani currency notes is 92% and the validation accuracy is 83.7%. Moreover, Fig. 6 shows the accuracy and loss curves over epochs. The training accuracy increases steadily across epochs, from approximately 12% to over 92% by the final epoch. This indicates that the model effectively learned features from the training data. The validation accuracy follows a similar trend, with initial low performance (~14%) and eventual convergence towards a stable range of 75% to 85% after epoch 15.

While the validation accuracy shows some fluctuation between epochs 17 and 26, likely due to overfitting or data noise, it remains consistently high in the later stages, demonstrating good generalization. The peak validation accuracy achieved is approximately 84%, which is competitive for real-world classification tasks involving currency notes, especially considering the challenges posed by varied backgrounds, wear and tear, and lighting conditions.

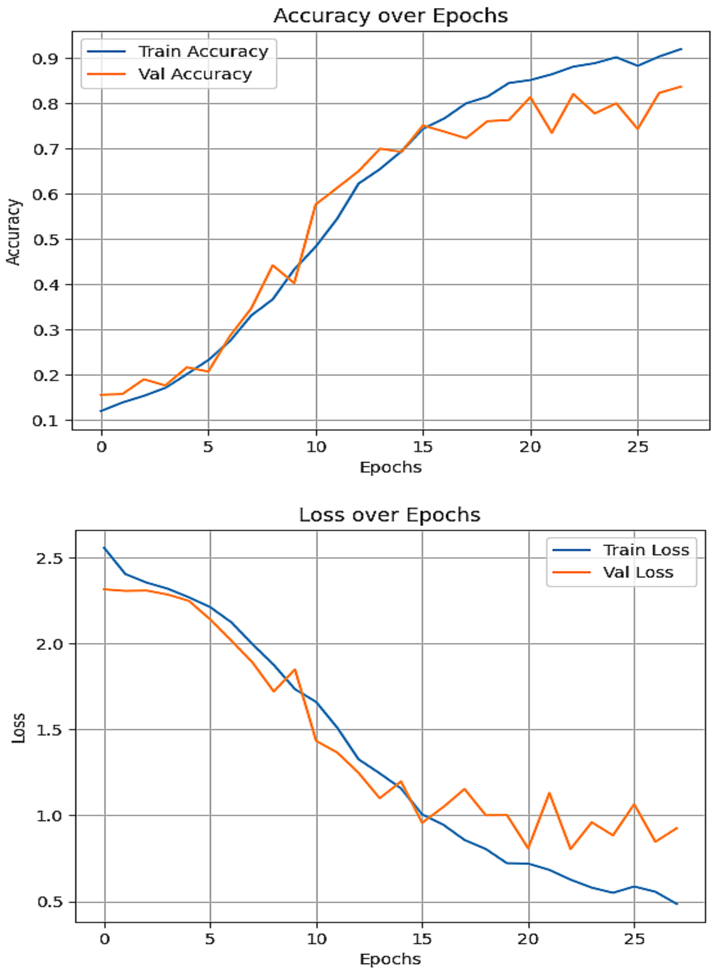

Fig. 6. The train and validation performance (accuracy and loss) curves of the VGG19 model on the classification of Pakistani currency notes.

5.2. Performance of YOLOv8 for Currency Classification

The YOLOv8 model is implemented using Google Colab Pro, which provides access to GPUs and TPUs for batch image processing and classification. We trained the model with a batch size of 64 and 30 epochs. The model's results are analyzed based on different parameters. The confusion matrix of the YOLOv8 model for the Pakistani currency detection system is shown in Fig. 7.

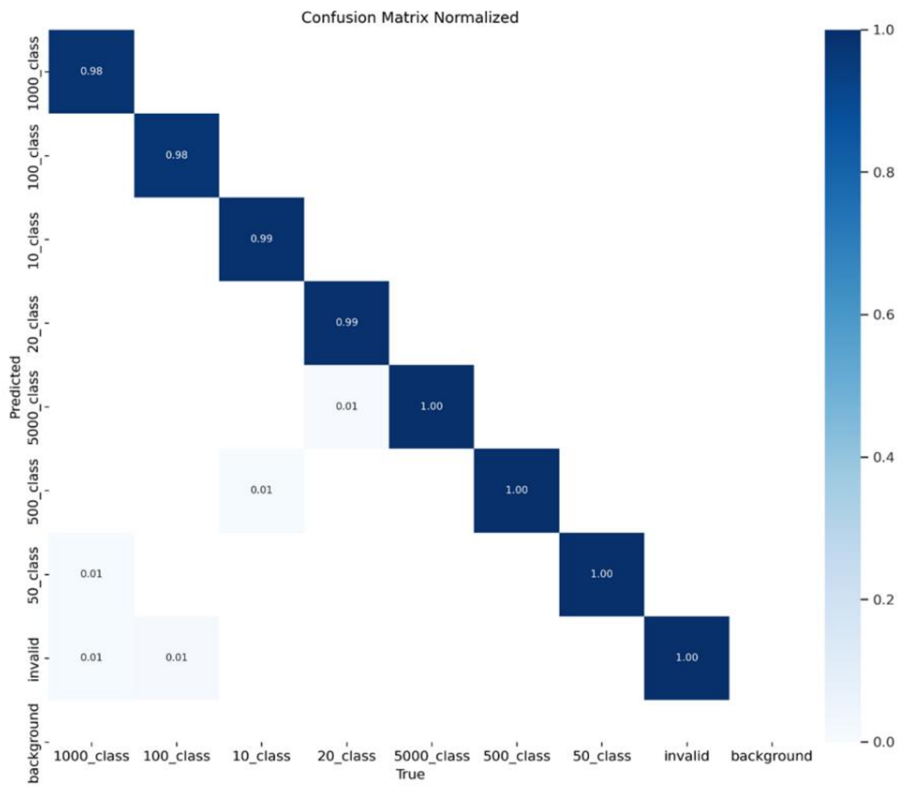

Fig. 7. The confusion matrix for the currency classification.

Looking at the confusion matrix, it shows that the model has only a 0.01% chance of mixing up 3 or 4 classes, which is negligible. This means the model trained well on the Pakistani currency note dataset with diverse images. It shows the advancement of the YOLOv8. The model can be trained well on images taken from any angle and in any environmental condition.

After the confusion matrix, training, and validation loss are crucial for analyzing the model's performance and avoiding overfitting. The training and validation graphs are shown in Fig. 8. The graphs above indicate successful learning, as the training and validation losses decrease throughout the epochs.

6. Discussion

We trained the YOLOv8 model for currency detection for 30 epochs. In Fig. 7, we show results over 15 epochs, demonstrating that the training loss (train/loss) steadily decreased throughout, indicating successful learning. This is further supported by the increasing top-1 and top-5 accuracy metrics (metrics/accuracy_top1 and metrics/accuracy_top5), which reached 99.1% and 100% respectively, by epoch 15.

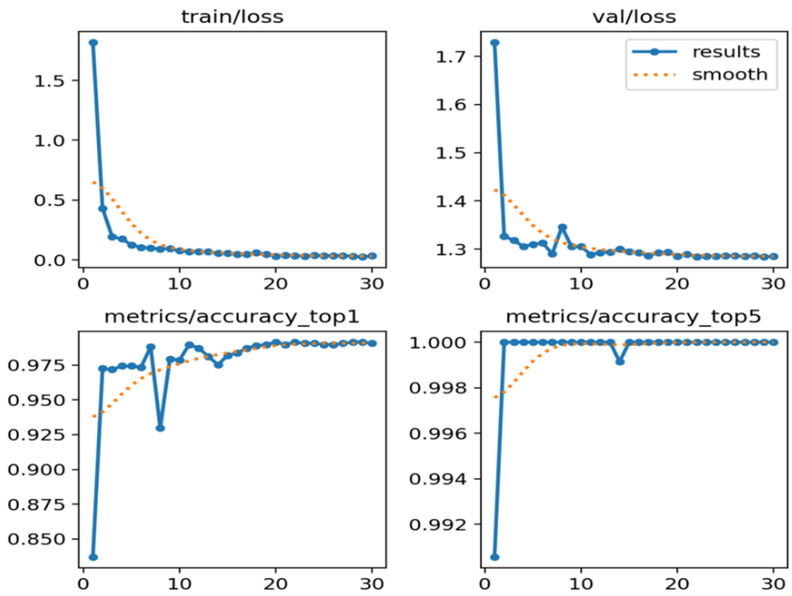

Fig. 8. The train and validation performance (accuracy and loss) curves of the YOLOv8 model on the classification of Pakistani currency notes.

In our context, top-5 accuracy means that if the correct class label is among the top 5 predictions (ranked by confidence), it is considered a correct prediction. This metric is especially useful in multi-class classification tasks (like ours with 8 currency classes), where multiple visually similar notes might be predicted. A 100% top-5 accuracy implies that the model always includes the correct class within its top 5 guesses.

However, monitoring the validation loss (val/loss) is crucial to ensure the model generalizes well to unseen data. We analyzed the difference between the training and validation losses in the graph above (see Fig. 7) to assess potential overfitting. The training loss (train/loss) and validation loss (val/loss) curves show a downward trend, suggesting the model is learning and generalizing well to unseen data. The training loss consistently decreases throughout the epochs, signifying the model effectively fits the training data. The validation loss also shows a decreasing trend, although it fluctuates slightly more than the training loss. This indicates the model is generalizing well to unseen data and avoiding overfitting the training data.

The confusion matrix (shown in Fig. 6) demonstrates the deep learning model's high performance in image classification. The rows represent the actual image classes, while the columns represent the classes predicted by the model. A total of about 200 images were classified. Looking at the confusion matrix, the model appears to have performed well at classifying currencies, with most values concentrated along the diagonal. For instance, the diagonal shows 98% correct classification for the “100_class,” meaning the model correctly classified 198 images out of the total number for the 100_currency class. Some misclassifications are negligible and can be ignored. The model correctly classified almost every class. Class labels containing a currency note (99% true-positive rate) yielded fewer false positives. These results suggest the model can effectively identify Pakistani currency notes in unseen images. However, it is important to note that the class imbalance can influence the model's overall performance in the dataset. So, the training data must be balanced.

For comparison, we also trained a VGG-19 model on the same dataset to evaluate its performance in the task of currency note classification. The VGG-19 model achieved a final training accuracy of approximately 92% and a peak validation accuracy of around 84% over 28 epochs. While the training accuracy showed a consistent upward trend, the validation accuracy fluctuated in the later epochs. Compared to YOLOv8’s nearly perfect performance, VGG-19 showed good but slightly less robust generalization. These results show that although VGG-19 can extract discriminative features for currency classification, YOLOv8 outperforms it in both accuracy and generalization, making it better suited for real-time, high-precision applications.

7. API development and Streamlit Front-end for Testing

To evaluate the trained YOLOv8 classification model (best.pt) for Pakistani currency note recognition, a lightweight Flask API and a Streamlit-based frontend were developed for testing purposes. The Flask API is responsible for loading the trained YOLOv8 classification model, accepting image inputs, performing necessary preprocessing, and returning the predicted class label and its associated confidence score in JSON format. The backend is structured to support isolated model inference without exposing training-related components.

A Streamlit interface was built to interact with the API during development and testing (see Fig. 9). It allows uploading images of currency notes, sends them to the backend for classification, and displays the classification results, including the detected currency denomination and confidence, in real time.

This setup is intended strictly for internal testing and validation of model predictions, ensuring correctness and consistency before considering integration into any larger system.

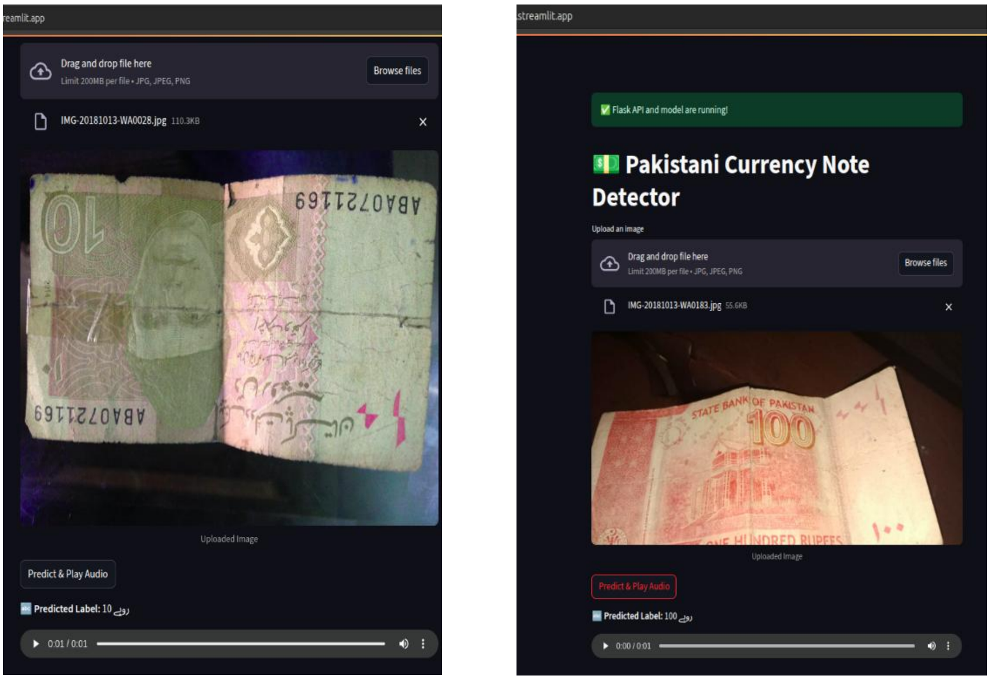

Fig. 9. Showing the testing of Streamlit. It responds in text output, along with audio feedback.

10. Conclusion and Future Directions

This research successfully developed a currency detection system utilizing the YOLOv8 deep learning model to assist visually impaired individuals in managing their finances. The system effectively recognizes various Pakistani currency denominations by processing images and providing audio feedback via text-to-speech. The YOLOv8 model achieved 99.1% top-1 and 100% top-5 accuracy after training on a balanced dataset of Pakistani currency notes, including fake notes to enhance security. The analysis of the training and validation loss curves confirms the model's ability to learn and generalize well to unseen data, avoiding overfitting.

As a comparative study, the VGG-19 model was also evaluated on the same dataset. While it achieved commendable training and validation accuracies of 92% and 84%, respectively, its performance was slightly lower and less stable than YOLOv8. This suggests that although VGG-19 can effectively classify Pakistani currency notes, YOLOv8 remains superior for real-time applications due to its higher accuracy and better generalization. Future directions may also include further comparisons with other CNN-based models to continue improving recognition accuracy, efficiency, and accessibility for visually impaired users.

This project significantly promotes financial independence for visually impaired individuals by empowering them to confidently identify currency denominations during transactions. Future research can focus on conditions, worn banknotes, and various fake notes to enhance the model's robustness. Additionally, exploring data augmentation techniques can improve the model's performance, particularly for classes with fewer images. By continuously refining the system, we can create a reliable and inclusive solution that empowers visually impaired individuals to navigate financial transactions more easily and confidently. The framework aims to recognize various currency denominations through a web-based testing application that provides audible feedback. The project emphasizes compatibility across different denominations and includes data collection, preprocessing, model selection, training, and evaluation. The chosen model undergoes fine-tuning, and the system's user interface incorporates auditory feedback. Deployment on the Google Cloud platform is planned, with a focus on legal and ethical considerations, including privacy and ethical design for inclusivity and fairness.

Future research could explore data augmentation techniques to increase the number of images and fake notes in the training dataset. This may improve the model's performance in this class and also help blind people in this regard.

REFERENCES

[1] *A. Yousry, M. Taha, and M. Selim,* Currency Recognition System for Blind people using ORB Algorithm, Int. Arab J. e-Technology, vol. 5, no. 1, pp. 33–40, 2018, [Online]. Available: https://www.researchgate.net/publication/329487411

[2] *G. Rösl and F. Seitz*, On the Stabilizing Role of Cash for Societies, no. 167, 2022, [Online]. Available: www.imfs-frankfurt.de

[3] *State Bank of Pakistan*, Payment Systems Review, State Bank Pakistan, pp. 1–15, 2022

[4] S. Abbasi, "Pakistan;s web of cyber scammers," DAWN, Jul. 16, 2023. [Online, Last Accessed: September 15, 2025]. Available: https://www.dawn.com/news/1764628

[5] Over 9 Million People Are Visually Impaired in Pakistan: Health Ministry," ProPakistani, 2022, [Online, Last Accessed: September 15, 2025]. Available: propakistani: https://propakistani.pk/2022/09/07/over-9-million-people-are-visually-impaired-in-pakistan-health-ministry/

[6] Hassan, B., Ahmed, R., Li, B., Noor, A., & Hassan, Z. ul. (2019). A comprehensive study capturing vision loss burden in Pakistan (1990-2025): Findings from the Global Burden of Disease (GBD) 2017 study. PLOS ONE, 14(5), e0216492. https://doi.org/10.1371/journal.pone.0216492

[7] Pkrevenue,"Currency notes have braille features for visually impaired persons: SBP". Last Accessed: September 15, 2025. Available:https://pkrevenue.com/currency-notes-have-braille-features-for-visually-impaired-persons-sbp/

[8] R. Jadhav et al., "Currency Recognition using Machine Learning," Int. Res. J. Eng. Technol., pp. 204–211, 2022, [Online]. Available: www.irjet.net

[9] RAZ, "6 Top Cell Phones for People Who are Blind or Visually Impaired in 2023," RAZ Mobility. [Online]. Available: https://www.razmobility.com/assistive-technology-blog/6-top-cell-phones-for-people-who-are-blind-or-visually-impaired-in-2023/

[10] A. Jawale, K. Patil, A. Patil, S. Sagar, and A. Momale, "Deep Learning based Money Detection System for Visually Impaired Person," in 2023 7th International Conference on Intelligent

Computing and Control Systems (ICICCS), IEEE, May 2023, pp. 173–180. doi: 10.1109/ICICCS56967.2023.10142586.
[11] C. H. Wai, "Development and evaluation of a banknote reader for visually impaired people using deep learning-based object detection" no. June, 2023.
[12] N. U. Foysal, T. J. Thamid, M. S. Uddin, M. F. Islam, M. N. Islam, and F. Al Faisal, "Advancing AI-based Assistive Systems for Visually Impaired People: Multi-Class Object Detection and Currency Classification," in 2023 International Conference on Information and Communication Technology for Sustainable Development (ICICT4SD), IEEE, Sep. 2023, pp. 438–442. doi: 10.1109/ICICT4SD59951.2023.10303628.
[13] M. Imad, F. Ullah, and M. A. Hassan, "Pakistani Currency Recognition to Assist Blind Person Based on Convolutional Neural Network," J. Comput. Sci. Technol. Stud., pp. 12–19, 2020.
[14] M. Ijaz, A. Iqbal, U. A. Raza, and R. M. Aftab, "A Survey on Currency Identification System for Blind and Visually Impaired," Int. J. Adv. Trends Comput. Sci. Eng., vol. 10, no. 2, pp. 989–996, 2021, doi: 10.30534/ijatcse/2021/721022021.
[15] O. T, "Counterfeit Currency Detection System using Image Processing and Deep Learning," Int. J. Sci. Technol., vol. 16, no. 3, Sep. 2025, doi: 10.71097/IJSAT.v16.i3.8486
[16 M. A. Siddiki, M. N. Hossain, K. Akhter, and M. R. Rahman, "Bangladeshi Currency Identification and Fraudulence Detection Using Deep Learning and Feature Extraction," Int. J. Comput. Sci. Mob. Comput., vol. 12, no. 1, pp. 1–13, Jan. 2023, doi: 10.47760/ijcsmc.2022.v12i01.001
[17] U. Sadyk, M. Yerzhan, R. Baimukashev, and C. Turan, "Comparative analysis of machine learning approaches in Kazakh banknote classification," Bull. Electr. Eng. Informatics, vol. 13, no. 5, pp. 3552–3567, Oct. 2024, doi: 10.11591/eei.v13i5.8004
[18] H. Prakash, A. Yadav, U. P, C. Jha, G. K. Sah, and A. Naik, "Deep Learning approaches for Automated Detection of Fake Indian Banknotes," in 2023 IEEE International Conference on Integrated Circuits and Communication Systems (ICICACS), Feb. 2023, pp. 1–5. doi: 10.1109/ICICACS57338.2023.10100265
[19] V. Abburu, S. Gupta, S. R. Rimitha, M. Mulimani, and S. G. Koolagudi, "Currency recognition system using image processing," 2017 10th Int. Conf. Contemp. Comput. IC3 2017, vol. 2018-Janua, no. 4, pp. 1–6, 2017, doi: 10.1109/IC3.2017.8284300
[20] S. Mahesh Bahrani, "Deep Learning Approach for Indian Currency Classification," Int. J. Eng. Appl. Sci. Technol., vol. 5, no. 6, pp. 335–340, 2020, doi: 10.33564/ijeast.2020.v05i06.049
[21] N. Buhl, "YOLO models for Object Detection Explained," ENCORD. [Online, Last Accessed: September 15, 2025]. Available: https://encord.com/blog/yolo-object-detection-guide/
[22] M. M. T. Anjali Dhami, Chandrika Deo, Dristi Gautam, "Comparative analysis of YOLO and SSD models for Nepali currency detection - A Project Report," no. 750407, 2023
[23] K. Balachandran, "Computer Vision and Deep Learning Based Assistance System for The Visually Impaired," no. December, pp. 20–23, 2023.