Grenze International Journal of Engineering and Technology, June Issue

Development of Android Application using Seven Segment Display for Digit Recognition using Deep Learning

C J Harshitha[1], Tejaswi K[2], Hreshikesh A[3] and Raghuram P S[4]
[1,3-4]Department of Computer Applications, JSS Science and Technology University, Mysuru, India
Email: harshi.cj@jssstuniv.in, tejaswi@jssstuniv.in, adal.hreshi@gmail.com
[2]Dayananda Sagar College of Arts, Science and Commerce, Bengaluru, India
Email: raghuram-bca@dscasc.edu.in

***Abstract*— The proposed deep learning Android application performs digit recognition on seven-segment displays. It uses a CNN model trained on a custom dataset and optimized for on-device inference using Tensor- Flow Lite (TFLite). The app will enable the user to choose a picture using the device camera or pick up a picture on the gallery. The prepro- cessing of the captured image optimizes the seven segment digits and the optimized image is sent to the trained CNN model for the classification. The identified numbers are presented on the interface of the application and they are stored on the device in SQLite database including times- tamps to enable users to access past records and analyze them. Data augmentation through rotation, scaling, and adjusting the brightness of the input image, the training of the model was performed in order to facilitate generalization. The precision, recall and F1-score are large in experimental evaluation confirming the accuracy and reliability of the of- freed approach. The lightweight aspect of the application makes it work in real-time on mobile devices with minimal resources, and the applications that could be fitted within could be automated meter reading and digital display monitoring.**

***Index Terms*— Convolutional Neural Networks · You Only Look Once · Deep Learning · Android · TensorFlow Lite.**

I. INTRODUCTION

Digit recognition has important applications such as automated meter reading, banking, and digital display interpretation. Traditional Optical Character Recognition (OCR) methods relied on image processing and template matching techniques. However, these methods often performed poorly under noise, low- quality displays, or varying illumination. With the introduction of deep learning, convolutional neural networks (CNNs) have succeeded superior performance in handwritten character and digit recognition tasks [3, 6]. Recent works have explored deploying lightweight CNN models on mobile and edge devices to enable real-time inference [3]. TensorFlow Lite (TFLite) has emerged as a widely adopted framework for deploying such models on Android applications. Several approaches have been developed for digit recognition in seven-segment displays and handwritten text. Early studies relied on traditional image processing and template-matching algorithms for optical character recognition (OCR) [1, 5].

Grenze ID: 01.GIJET.12.2.38_21

These methods were effective in controlled conditions but showed poor accuracy in noisy or low-quality images. The development and groundbreaking improvements in deep learning, convolutional neural networks (CNNs) and re- current neural networks (RNNs) have demonstrated superior performance for handwritten character and digit recognition [3, 6]. Recent works have explored deploying such models on mobile and edge devices for real-time applications [2]. Transfer learning and hybrid models combining CNN with sequence models such as BiLSTM have also been proposed for improved accuracy [10]. This pa- per presents the development of an Android application for seven-segment digit recognition using a CNN-based model. The application captures or uploads im- ages of seven-segment displays, performs inference using TensorFlow Lite, and stores recognized digits in a local database for retrieval and analysis. Experimental results demonstrate high accuracy and efficiency, making the proposed approach is well suited and has the capabilities for real-time applications such as meter reading and digital display monitoring.

II. PROPOSED WORK

The Android application is designed for digit recognition from seven-segment dis- plays using deep learning. A lightweight CNN model was trained and converted to TensorFlow Lite (TFLite) format for on-device inference. The app captures or uploads an image, processes it, performs inference, and stores recognized digits in a SQLite database.

The workflow of the application is as follows:

- The device camera allows the user to capture an image through the camera or the gallery provides a selection of an image.
- Image preprocessing is done to improve the seven segment digits in order to be recognized.
- The image is preprocessed and then fed to a CNN model that is coded with TensorFlow Lite to be inferred.
- The estimated figures are shown on the application.
- The identified digits along with a timestamp are kept in a local SQLite database.
- A user is able to see data stored in a table format, or restore the database on need.

III. IMPLEMENTATION DETAILS

The Android application that has been proposed was built on Android Studio where the major programming language was Java. The implementation of the deep learning model was in Python by using Keras and TensorFlow. A seven-segment digit dataset in the form of custom was acquired and processed by applying grayscale, noise elimination, and image downsizing methods to a constant resolution. Rotation, scaling and brightness augmentation were also used to augment data to improve model resistance. The CNN model was then trained and converted to TensorFlow Lite, which is a much smaller and faster to inference model, but is no less accurate.

The application architecture can be thought of as having three modules: (1) Image Acquisition, when the user takes an image with the device camera or reads one on the gallery; (2) Inference Module, which preprocesses the image and identifies digits with the help of the TensorFlow Lite (TFLite) model; and (3) Result Management, which displays the recognized digits on the user interface and logs them to a local SQLite database with timestamps.

The model was quantized with the TensorFlow Lite (TFLite) quantization methods to minimize the amount of computation required to run the model, since it was known to be resource-intensive and runnable on resource-intensive devices. The application is fully off-line based and therefore can be used in the field where the internet might be minimal. Figure 1 illustrates the overall workflow of the proposed application, from image capture to digit recognition and storage.

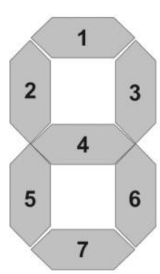


Figure 1. Workflow of the proposed 7-segment digit recognition application.

IV. Result and Discussion

The custom seven segment digit dataset was used to test the proposed Android application. This CNN model was converted to TensorFlow Lite (TFLite) and it was tested on a device based on Android and showed high accuracy and a low inference delay. The confusion matrix as shown on figure 2 represents the classification accuracy of the model on each digit.

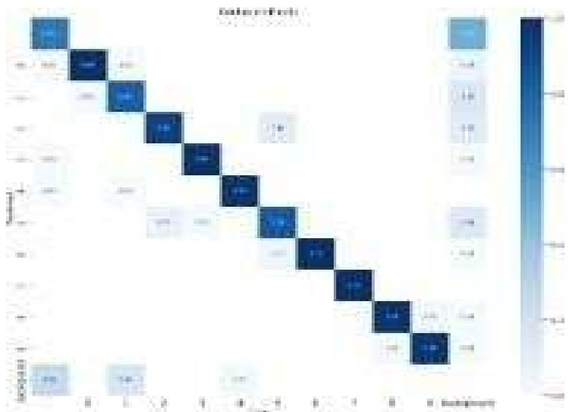


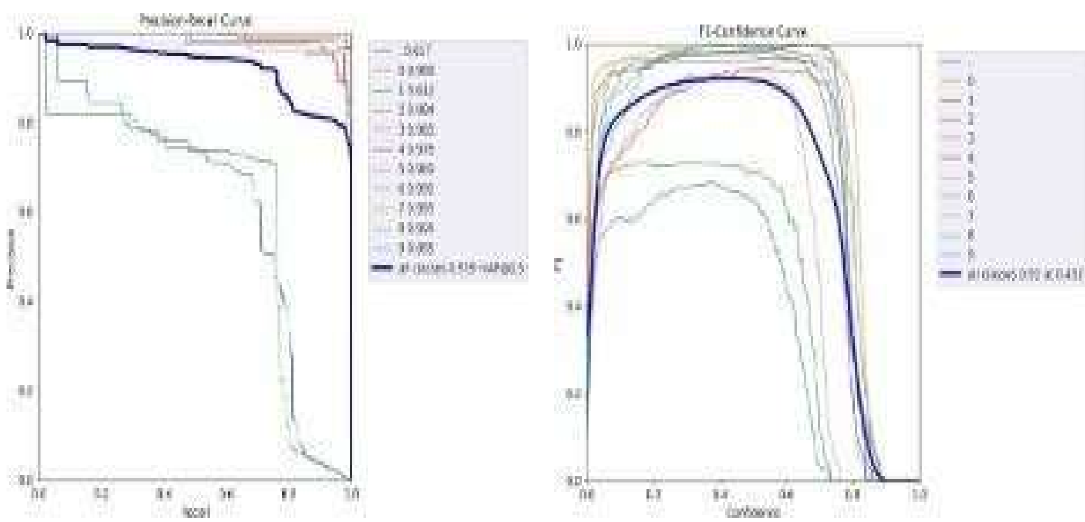
Figure 2. Confusion Matrix

Table 1 summarizes the performance metrics. The model achieved a precision of 95.6%, recall of 94.3%, and an F1-score of 94.9%. Table 2 summarizes comparative analysis from recent research.
Note: YOLOv8 results in Table 2 are included only as a benchmark comparison. The Android pipeline developed in this work integrates only the CNN model converted to TFLite, not YOLOv8.

TABLE I. PERFORMANCE METRICS OF THE PROPOSED MODEL

Metric	Value
Precision	95.6%
Recall	94.3%
F1-score	94.9%

Figure 3 shows the Precision-Recall curve (a) and the F1-Confidence curve.
These plots illustrate how the model's precision, recall, and F1-score vary with different decision thresholds, confirming the robustness of the proposed approach.



(a) Precision-Recall Curve (b) F1-Confidence Curve

Figure 3. Precision-Recall and F1-Confidence curves of the proposed model.

TABLE II: SUMMARY OF EXISTING METHODS FOR DIGIT/TEXT RECOGNITION

Author (Year)	Technique Used	Dataset	Application Domain	Performance Metrics
Kulkarni (2016) & Kute	7-step image processing al- gorithm (OCR without deep learning)	Simulated & field images of 7-seg displays	Automated meter read- ing (LCD 7-segment dig- its)	~79% digit recognition accu- racy under varied conditions
Hasan et al. (2013)	Image processing + template-based methods (improved algorithm)	7-segment digital display images	Instrument display digit OCR	Improved over earlier meth- ods; accuracy not explicitly reported
Hsu et al. (2024)	CNN with transfer learn- ing (robust distributed edge learning)	Glowing LED digit im- ages (custom)	Luminous 7-seg displays (edge devices)	Improved from ~70% to >92% accuracy using robust training techniques
Mule et al. (2022)	CNN + BiLSTM sequence model	IAM Handwritten Text dataset	Handwritten text (offline HWR)	~92% character recognition accuracy
Revathy & Nath (2020)	Tesseract OCR + Google Translate API	Images printed/handwritten text of	Mobile text recognition & translation	Struggled with handwriting; low accuracy on cursive text
Mook et al. (2023)	Deep CNN (VGG16, Mo- bileNet, InceptionResNetV2)	Mixed handwritten chars & digits	General OCR (printed/handwritten chars)	InceptionResNetV2 achieved ~97.16% accuracy
Lv (2023)	CNN classifier (LeNet-style deep network)	MNIST (handwritten digits)	Handwritten digit classi fication	~99% digit classification ac- curacy (MNIST benchmark)
Promsuk et al. (2021)	OCR-based pipeline with pre- processing (HOG + Hough) and MLP classifier	Images of industrial me- ter displays (7-seg)	IIoT instrumentation (digital meter reading)	~95% accuracy in real-time recognition with noise handling
Proposed YOLOv8 Model (2025)	YOLOv8 one-stage detector (anchor-free CNN)	7-Segment LED Digit Dataset (2,300+ images)	Real-time 7-seg display reading (meters, clocks)	Precision ~95.6%, Recall ~94.3%, F1 ≈94.9% (better than YOLOv5 with 92% precision)

Note: YOLOv8 results in Table 2 are included only as a benchmark com- parison. The Android pipeline developed in this work integrates only the CNN model converted to TFLite, not YOLOv8.

III. CONCLUSIONS

This paper presented the development of an Android application for digit recog- nition from seven-segment displays using a CNN model converted to TensorFlow Lite (TFLite). The proposed system enables real-time inference on mobile de- vices and stores recognized digits in a local database for retrieval. Experimental results showed high precision, recall, and F1-score, confirming the effectiveness of the approach. Future work will focus on expanding the dataset, optimizing model size further for low-end devices, and extending the application to support multilingual OCR capabilities.

REFERENCES

[1] Kulkarni, P.H., Kute, P.D.: Optical numeral recognition algorithm for seven seg- ment display. In: 2016 Conference on Advances in Signal Processing (CASP), Pune, India, pp. 397–401. IEEE (2016). https://doi.org/10.1109/CASP.2016.7746203

[2] 2. Hsu, T.-C., Tsai, Y.-H., Chu, W.C.-C.: Enhancing digit recognition for luminous images in edge computing through transfer learning with ro- bustness and fault tolerance.IEEE Transactions on Reliability(2024). https://doi.org/10.1109/TR.2024.3393424

[3] R. S. N., Kennedy Babu C.: Adaptive Residual Attention Network for Handwritten Character and Digits Recognition with Improved Energy Valley Optimizer Algo- rithm. In: 2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS), Chennai, India, pp. 1–6. IEEE (2024). https://doi.org/10.1109/ADICS58448.2024.10533644

[4] Dash, P.R., Balabantaray, R.C., Dey, R.: An Approach for Handwritten Alphanu- meric Character Recognition: Leveraging CNN for Accurate Recognition. In: 2024 1st International Conference on Cognitive, Green and Ubiquitous Computing (IC- CGU), Bhubaneswar, India, pp. 1–6. IEEE (2024). https://doi.org/10.1109/IC-CGU58078.2024.10530699

[5] Hasan, M.M., Rahaman, A., Maswood, M.M.S., Rahman, M.M.: An improved method for 7-segment numeric character recognition. In: 2013 International Con- ference on Informatics, Electronics and Vision (ICIEV), Dhaka, Bangladesh, pp. 1–IEEE (2013). https://doi.org/10.1109/ICIEV.2013.6572607

[6] Mook, C.E., Lee, C.P., Lim, K.M., Lim, J.Y.: Handwritten Character and Digit Recognition with Deep Convolutional Neural Networks: A Comparative Study. In: 2023 11th International Conference on Information and Commu- nication Technology (ICoICT), Melaka, Malaysia, pp. 137–141. IEEE (2023). https://doi.org/10.1109/ICoICT58202.2023.10262721

[7] He, K.: Handwritten Digit Recognition Based on Convolutional Neural Network. In: 2023 IEEE 3rd International Conference on Electronic Technology, Commu- nication and Information (ICETCI), Changchun, China, pp. 16–19. IEEE (2023). https://doi.org/10.1109/ICETCI57876.2023.10176680
[8] Lv, X.: Handwritten Digit Recognition Based on Deep Learning Algorithms. In: 2023 International Conference on Internet of Things, Robotics and Dis- tributed Computing (ICIRDC), Rio De Janeiro, Brazil, pp. 476–481. IEEE (2023). https://doi.org/10.1109/ICIRDC62824.2023.00093
[9] Revathy, S., Nath, S.: Android Live Text Recognition and Translation Applica- tion using Tesseract. In: 2020 4th International Conference on Intelligent Comput- ing and Control Systems (ICICCS), Madurai, India, pp. 1259–1263. IEEE (2020). https://doi.org/10.1109/ICICCS48265.2020.9120973
[10] Mule, H., Kadam, N., Naik, D.: Handwritten Text Recognition from an Image with Android Application. In: 2022 IEEE Fourth International Conference on Advances in Electronics, Computers and Communications (ICAECC), Bengaluru, India, pp. 1–5. IEEE (2022). https://doi.org/10.1109/ICAECC54045.2022.9716714
[11] Garcia, M.B., Claour, J.P.: Mobile Bookkeeper: Personal Financial Man- agement Application with Receipt Scanner Using Optical Character Recog- nition. In: 2021 1st Conference on Online Teaching for Mobile Edu- cation (OT4ME), Alcalá de Henares, Spain, pp. 15–20. IEEE (2021). https://doi.org/10.1109/OT4ME53559.2021.9638794
[12] Shiddieqy, H.A., Adiono, T., Syafalni, I.: Mobile Client-Server Approach for Handwriting Digit Recognition. In: 2019 International Symposium on Electron- ics and Smart Devices (ISESD), Badung, Indonesia, pp. 1–4. IEEE (2019). https://doi.org/10.1109/ISESD.2019.8909448
[13] 7-Segment Digits Dataset, https://universe.roboflow.com/bhautik-pithadiya/7- segment-display-gxhnj/dataset/5, last accessed 2025/07/25
[14] Android AI Developer Guide, https://developer.android.com/ai, last accessed 2025/07/25
[15] TensorFlow Lite, https://www.tensorflow.org/lit, last accessed 2025/07/25