DLSU Research Congress 2025
De La Salle University, Manila, Philippines
June 26 to 28, 2025

Smart Recognition of Seven-Segment Displays Using CNN for Sustainable Monitoring in Green Technologies

Harish Chawla[1], Julius Alvin Librando[1], Carlos Miguel Castro[1] and Ronnel Agulto[1]
[1] *Department of Electronics and Computer Engineering*
De La Salle University - School of Innovation and Sustainability
Biñan, Laguna, Philippines
Corresponding Author: ronnel.agulto@dlsu.edu.ph

Abstract: Accurate digit recognition is an extremely useful component in modern industrial, environmental monitoring systems for automatic energy meter reading, energy consumption optimization, and anomaly detection. This paper proposes a CNN-based method for the effective recognition of numbers shown on seven-segment displays, which are commonly found in various devices to output data, emphasizing its applications in green technologies for facilitating automated monitoring and data logging. The model used a dataset of 40,000 gray-scale images representing all ten digit classes of transformed images taken from all possible angles and positions to accurately reflect a real-world setting. To increase robustness, it uses extensive data augmentation, including rotations, shifts, shearing, etc. The CNN is based on VGG-like architectures, consisting of two convolutional blocks of 3×3 filters, and is complemented with batch normalization, ReLU activations, max-pooling, and dropout layers to prevent overfitting. With an 80/20 train/validation split, classification metrics are found with precision, recall, and F1-scores around 99% for all classes. These results validate that the proposed CNN architecture is able to deliver an inexpensive and accurate solution for seven-segment display recognition to cut down on manual monitoring efforts while guaranteeing reliable data acquisition. Fast-converging and low-compute complexity indicators illustrate the feasibility of the method's inclusion on resource-limited embedded platforms. As a result, this automated digit recognition system can be integrated into IoT devices to enable reliable monitoring in smart grids, water-distribution networks, and industrial settings, thus contributing to sustainable practices as evidenced by the performance of a number of United Nations Sustainable Development Goals (SDGs).

Key Words: human-computer interaction, seven-segment display, computer vision, image recognition, convolutional neural network

1. INTRODUCTION

Seven-segment displays are ubiquitous in low-cost embedded interfaces, from multimeters and clocks to appliances and automotive indicators, arguably due to their simplicity and low cost (Wan-Fu, 2011; Thimbleby, 2013). However, manual reading is less efficient and accurate for industrial monitoring (Deshpande et al., 2023; Wannachai et al., 2020; Wannachai et al., 2022), and for sustainable IoT solutions such as smart energy and water meters requiring real-time data capture, anomaly detection is crucial with as least frequent visits as possible (Kanagarathinam & Sekar, 2016; Haseeb et al., 2024).

OCR datasets combine real and synthetic seven-segment images, which are typically created programmatically to augment training sets (Gushima & Kashima, 2023); many OCR algorithms depend on binarization through fixed or adaptive thresholding (Yousefi, 2011; Bangare et al., 2015). The architectures vary from end-to-end CNN classifiers to separate detection and recognition pipelines aimed at resource-constrained devices (Suttapakti et al., 2022). These have specialized implementations in fields such as industrial machine monitoring status (Ghugardare, 2009; Kulkarni et al., 2016), medical devices such as glucometers and pulse oximeters (Boonsim & Kanjaruek, 2023; Boonnag et al., 2023; Finnegan et al., 2019) and the counting down for traffic light recognition for autonomous vehicles (Kulkarni et al., 2018; Saini et al., 2019) with an accuracy of up to 98.2% (Shenoy & Aalami, 2018).

Low-cost optical character recognition (OCR) based IoT solutions are capable of becoming scalable, sustainable for monitoring in the domains of smart grids, water networks, and industrial infrastructures, which corresponds with UN SDGs 6, 7, and 9. Because of this, a CNN-based computer-vision method for detecting and classifying seven-segment display values of various automated monitoring tasks can be used across all scales of infrastructures and plants without imposing too much cost overhead.

Thus, this paper proposes a machine learning approach to recognize numerical values from seven-segment displays using computer vision. This study proposes a model using convolutional neural networks or CNNs to recognize and classify displayed numbers solely, which can then be used or interfaced further according to their intended use, such as in a seven-segment detection and recognition pipeline for a holistic automated data logging solution.

2. METHODOLOGY

2.1. Dataset Acquisition and Annotation

An internal database of about 40,000 grayscale images was created, with nearly 4,000 examples from each digit class (0–9). The images for the dataset acquired were collected from various seven-segment devices under varying ambient light conditions, angles of view, and display styles, with variability similar to real-world deployments. Sample images for the digit class 0 showing the variability in contrast and brightness in the dataset can be seen in Figures 1.a. to 1.c., with Figures 1.b. and 1.c. showing obstructive lighting conditions that could otherwise result in difficulties in recognition.

Figure 1. Sample images for class 0, in different lighting conditions (left to right): (a) normal conditions, (b) low contrast, and (c) high brightness with partial glare

Incorporating these variations on the training and testing datasets adds to the overall robustness of the recognition model, making it adaptive to various environments, such as indoor and outdoor deployments.

Additionally, affine image transformations are also done on the dataset for data augmentation, such as rotation and shearing, as shown in Figures 2.a. to 2.c. These transformations are especially useful for emulating different camera views and perspectives.

Figure 2. Sample images for class 0, with different transformations (left to right): (a) original image, (b) rotated image, and (c) sheared image

Figures 2.b. shows a seemingly different style or perspective of digit 0 by simply rotating the original image, while the shearing in Figure 2.c. helps emulate either a change in viewing perspective, or a subtle change in the seven segment display typeface (such as italic stylization found in some displays). These further improve the robustness of the model, and allow the model to be used in scenic applications in various positions and angles.

The 96×96 images were resized to 28×28 pixels (the input dimensions of the networks) after providing separate sub-datasets for each digit. The complete dataset was split into training (80%, ≈32000 images) and validation (20%, ≈8000 images) subsets for training via stratified sampling in order to maintain class balance and avoid sampling truncation bias.

2.2. *Preprocessing and Data Augmentation*

Before training, all pixel values were scaled by a factor of 1/255 so that their values fall in the [0,1] interval. On-the-fly data augmentation was applied via Keras's ImageDataGenerator to increase model robustness against possible affine distortions and avoid overfitting. Augmentation parameters were as follows: random rotations ±10°, horizontal and vertical shifts ±10% of the image width/height, followed by real-time rescaling.

The generator was set up with the 80/20 split for training and validation. The training samples were shuffled each epoch, while validation was executed with a fixed order, so the performance could be compared each epoch.

2.3. *Neural Network Architecture*

Based on the VGG model architectures, the proposed model stacks successive layers of 3×3 convolutions, which preserve the small number of parameters while maximizing the effective receptive field in order to learn deep feature hierarchies that have been shown to improve image classification accuracy at minimal cost.

Batch normalization is applied after each convolution to stabilize inputs to layers and allow for higher learning rates by reducing internal covariate shift, leading to faster convergence and more robust training. A 2×2 max-pooling technique is then used to progressively reduce the spatial resolution, thus providing translation invariance and aggregating prominent segment features into dense maps that match the uniform geometry of the seven-segment digits. Given the 40,000-image dataset, dropout regularization protects against overfitting by randomly turning off activations during training, which simulates an ensemble of subnetworks and promotes generalization over varied display conditions. Two stages of downsampling (28 → 14 → 7) are employed to achieve a smooth transition between capturing low-resolution, local structure and encoding outputs for global, class-specific patterns, similar to hierarchically organized biological vision systems.

The overall architecture of the proposed CNN model can be seen in Figure 3.

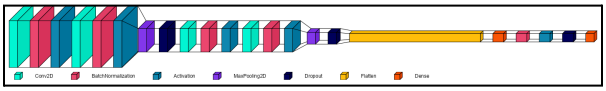

Figure 3. CNN Model Architecture

The model used for the CNN model adopts a VGG-like architecture, given its wide application in computer vision applications and its intended use case, which is much simpler and does not necessitate deeper neural network architectures that a ResNet-like architecture might provide. The benefits of alleviating the vanishing gradients problem on ResNets become more apparent on more complex image tasks such as ImageNet classification (He et al., 2016). However, ResNet's increased depth and parameter count incur greater computational overhead at both training and inference time, a factor not to be dismissed for real-time or embedded deployments of a seven-segment reader.

In contrast, with a relatively simple VGG-like architecture involving two convolutional blocks, along with well-designed regularization (batch normalization and dropout), the resulting design is more responsive and generalized, while also being faster to train and deploy compared to a ResNet variant.

2.3.1. *Initial Feature Extraction Block*

The network starts with two 32-filter 3×3 convolutional layers with the "same" padding stacked. Small kernels are selected to characterize fundamental features—edges, corners, and simple strokes—while minimizing the number of parameters. This is important because of the relatively small input size (28×28 pixels) and scale of datasets. Batch normalization normalizes

mini-batch activations after each convolution, reducing internal covariate shift and allowing high learning rates without intricate initialization. ReLU (rectified linear unit) activations are used next to introduce non-linearity, where the idea is to promote sparse gradients and avoid saturation, thus increasing convergence. A 2×2 max-pooling layer halves the spatial dimensions (to 14×14), enforcing moderate translation invariance, followed by 25% dropout, which randomly shuts down feature detectors during training and reduces co-adaptation.

2.3.2. Hierarchical Feature Refinement Block

The second block of the proposed model captures higher-level features and expands the representation capacity built on low-level features.

This time, there are two convolutional layers, each with 64 filters (3×3 kernels, "same" padding) that capture more complex patterns like segment junctions and subtle inter-digit distinctions (e.g., "8" compared to "0"). Batch normalization and ReLU follow each convolution to ensure stable, non–linear activations. A second 2×2 max-pooling layer shrinks feature maps to 7×7, which focuses on the most salient features of the input even further, while a second 25% dropout layer reduces overfitting by enforcing up to 25 independent feature detectors.

2.3.3. Classification Head

The filtered 7×7×64 feature tensor is flattened and passed on to a 256-unit fully connected layer. Batch normalization and ReLU activation still regulate activation distributions and enable deep gradient flow. This is followed by a much more aggressive 50% dropout layer that enforces robustness by simulating an ensemble of sub-networks during training.

The output layer then has 10 units and softmax as its activation function, giving class probabilities and defined decision boundaries for all digits 0 to 9.

2.4. Training Process

Training was conducted on a GPU with memory growth enabled to avoid out-of-memory errors. If no GPU is found, the processor used for training defaults to the CPU.

During the whole process, the network was compiled with the Adam optimizer (learning rate = 0.001) and categorical cross-entropy loss function to optimize accuracy. We trained for 10 epochs, with a batch size of 32, performing $\lfloor 32000/32 \rfloor = 1000$ steps per epoch and $\lfloor 8000/32 \rfloor = 250$ validation steps.

Early stopping was implemented (patience = 3) to decay overfitting, monitor validation loss and checkpoint the model (to save weights on best validation accuracy).

2.5. Evaluation

The final model was evaluated on the validation set at the end of training. Overall accuracy is reported, and precision, recall, and F1-scores across all digit classes, as defined by scikit-learn's classification_report.

To assess misclassification patterns, a confusion matrix was created, which can help in summarizing recognition discrepancies, particularly between visually similar digits (e.g., "8" vs. "0"). To confirm convergence and check for remaining overfitting, training and validation curves were plotted for both accuracy and loss.

3. RESULTS AND DISCUSSION

Table 1. Classification Metrics

Digit	Precision	Recall	F1-Score	Support
0	1.00	0.99	1.00	802
1	1.00	1.00	1.00	798
2	1.00	0.96	0.98	801
3	1.00	0.99	1.00	797
4	1.00	1.00	1.00	799
5	1.00	1.00	1.00	800
6	0.98	0.99	0.99	800
7	1.00	1.00	1.00	799
8	0.97	1.00	0.98	803
9	1.00	1.00	1.00	798

The classification metrics, as shown in Table 1, show that the model achieves almost perfect performance values, with a precision, recall, and F1-scores approaching 1.00 in all ten digit classes. This

result indicates that the CNN rarely confuses one digit with another (high precision) and rarely misses the correct digit (high recall). Consequently, the F1-score of the model, which is the harmonic mean of precision and recall, continues to be at a near-maximum level for each digit, signifying balanced and robust recognition capabilities. Furthermore, the sufficiently high support (number of samples per class) for each digit class validates these findings as the metrics do not suffer from small sample sizes.

Overall, the results illustrate the robustness of the architecture and training strategies (data augmentation and regularization) that were used. It is important to achieve high performance metrics since misreadings may result in critical errors in the system that the seven-segment reader is interfaced with.

Table 2. Accuracy Metrics

Metric	Precision	Recall	F1-Score	Support
Accuracy	—	—	0.99	7997
Macro Ave.	0.99	0.99	0.99	7997
Weighted Ave.	0.99	0.99	0.99	7997

The trained CNN model successfully achieves perfect/near-perfect classification performance of digits in the dataset, which can be interpreted through the fact that the overall accuracy of the model is 0.99 (Table 2). In the case of macro average and weighted average, these metrics also yield 0.99 precision, recall, and F1-score, indicating balanced performance for all digit categories with no bias toward any class. This high level of consistent performance suggests that the CNN can be trusted in situations where it is equally consequential to recognize a digit correctly. A support of 7997 also confirms that the results are not skewed by small sample sizes.

To sum up, the performance listed in Table 2 confirms that the structure and training method of the network is capable of providing the consistent and accurate recognition of seven-segment digits required for real-world implementations, where a single misread digit can cause huge mistakes.

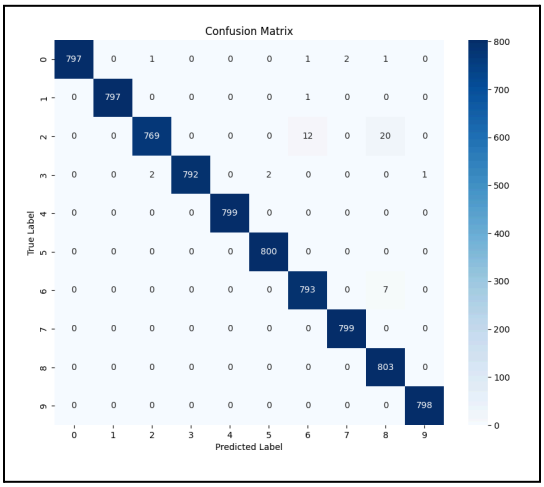

Figure 4. Confusion matrix

The confusion matrix further supports this, having a near-perfect categorization of the digits (Fig. 4). It can be noted that the digit “2” has most occurrences of miscategorization as digit “8” followed by “6”, owing to the fact that in a seven-segment display format, both digits only differ from each other by a single or double lit segment. The same can be observed from the digit “8”, where it was misclassified as digit “8” due to the same reason, having the third-highest occurrences of miscategorization.

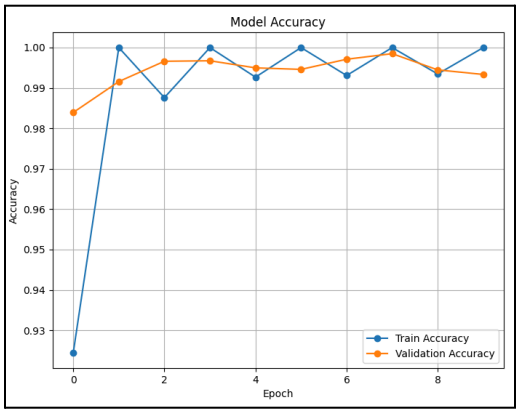

Figure 5. Line graph of accuracy vs epochs for training (blue) and validation (orange)

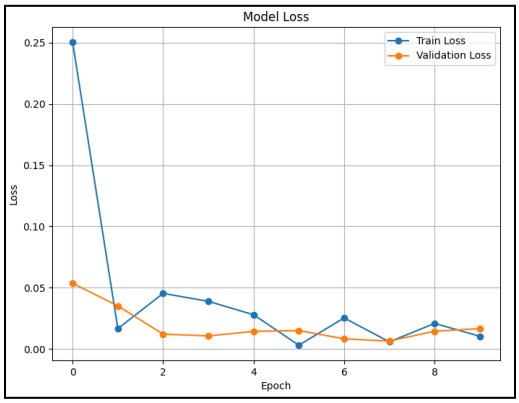

Figure 6. Line graph of loss vs epochs for training (blue) and validation (orange)

After training the model for 10 epochs, the training accuracy started fluctuating while the validation accuracy settled at around 99% accuracy (Fig. 5), while the training and validation loss continued to decrease (Fig. 6). The model does not necessarily need too much training, as increasing the epochs beyond 10 leads to signs of overfitting, where there is an observed increase in validation loss; this has been determined during the model's training and not shown in the figures.

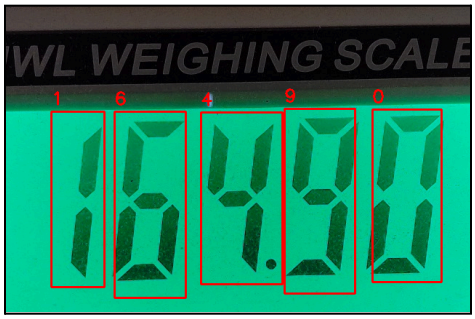

Figure 7. Digital weighing scale with seven-segment display reading

The model was then tested on recognizing values on scenic images of seven-segment displays displaying real values. Figure 7 shows the identification of numerical values displayed on a digital weighing scale seven-segment display, outputting the predicted values. These values can then be processed further according to the particular use case.

Overall, the trained CNN model can ultimately be deployed and interfaced with computing devices, and may undergo further optimization for use in low-power embedded AI devices for automated readout and logging of values displayed in seven-segment format after bounding boxes or Region of Interest (ROI) have been specified (Fig. 7). Ideally, the model can be paired with a separate seven-segment detection model to automate the ROI selection process and included in a seven-segment reader pipeline.

4. CONCLUSIONS

Overall, the study successfully implemented a convolutional neural network with an architectural design roughly similar to VGGNet. The performance metrics were relatively high, reaching around 99% precision, recall, and F1-scores for all classes (digits 0-9 inclusive), indicating that all classes were more than adequately classified correctly, with minimal to negligible instances of false positives nor missed classifications per category. Additionally, the resulting model exhibited minimized losses for both training and validation, which indicates that the model is generalized and most likely not overfitted, and is expected to perform still adequately well when used on previously unseen image inputs.

In line with the study's theme on sustainability, the approach proposed in the study enables efficient data acquisition and management, potentially for use in smart grids, water distribution networks, and various industrial environments. Future directives for this research may include the addition of a seven-segment detection model as well as letters/characters used in hexadecimal notation in the recognition model, although its application is largely more limited. There can also be other training setup alternatives for future research. For instance, while ImageDataGenerator met the project's requirements, the tf.data API can be considered for better performance and scale. These improvements will strengthen the idea of the model being a low-cost, low-power alternative, not just a response to operational need, but an active contributor to sustainability in technology-based monitoring-driven initiatives.

5. REFERENCES

Bangare, S. L., Dubal, A., Bangare, P. S., & Patil, S. (2015). Reviewing Otsu's method for image thresholding. International Journal of Applied Engineering Research, 10(9), 21777-21783.

Boonnag, C., Ittichaiwong, P., Saengmolee, W., Seesawad, N., Chinkamol, A., Rattanasomrerk, S., ... & Wilaiprasitporn, T. (2023). PACMAN: A framework for pulse oximeter digit detection and reading in a low-resource setting. IEEE Internet of Things Journal, 10(15), 13196-13204.

Boonsim, N., & Kanjaruek, S. (2023, September). An integrated technique for detecting seven-segment digits on medical devices. In 2023 27th International Computer Science and Engineering Conference (ICSEC) (pp. 1-4). IEEE.

Deshpande, S., Padalkar, S., & Anand, S. (2023). IIoT based framework for data communication and prediction using augmented reality for legacy machine artifacts. Manufacturing Letters, 35, 1043-1051.

Finnegan, E., Villarroel, M., Velardo, C., & Tarassenko, L. (2019). Automated method for detecting and reading seven-segment digits from images of blood glucose metres and blood pressure monitors. Journal of medical engineering & technology, 43(6), 341-355.

Ghugardare, R. P., Narote, S. P., Mukherji, P., & Kulkarni, P. M. (2009, January). Optical character recognition system for seven segment display images of measuring instruments. In TENCON 2009-2009 IEEE Region 10 Conference (pp. 1-6). IEEE.

Gushima, K., & Kashima, T. (2023, September). Automatic generation of seven-segment display image for machine-learning-based digital meter reading. In PHM Society Asia-Pacific Conference (Vol. 4, No. 1).

Haseeb, H., Hassan, M. T., Iftikhar, A., & Asmat, A. (2024). Automated Detection and Recognition of Seven-Segment Digits from Electric Meters Utilizing Digital Image Processing and Machine Learning. VFAST Transactions on Software Engineering, 12(4), 87-98.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

Kanagarathinam, K., & Sekar, K. (2019). Text detection and recognition in raw image dataset of seven segment digital energy meter display. Energy Reports, 5, 842-852.

Kulkarni, P. H., Kute, P. D., & More, V. N. (2016, January). IoT based data processing for automated industrial meter reader using Raspberry Pi. In 2016 International Conference on Internet of Things and Applications (IOTA) (pp. 107-111). IEEE.

Kulkarni, R., Dhavalikar, S., & Bangar, S. (2018, August). Traffic light detection and recognition for self driving cars using deep learning. In 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA) (pp. 1-4). IEEE.

Saini, S., Nikhil, S., Konda, K. R., Bharadwaj, H. S., & Ganeshan, N. (2017, June). An efficient vision-based traffic light detection and state recognition for autonomous vehicles. In 2017 IEEE Intelligent Vehicles Symposium (IV) (pp. 606-611). IEEE.

Shenoy, V. N., & Aalami, O. O. (2018, April). Utilizing smartphone-based machine learning in medical monitor data collection: Seven segment digit recognition. In AMIA Annual Symposium Proceedings (Vol. 2017, p. 1564).

Suttapakti, U., Titijaroonroj, T., Nunsong, W., & Kakanopas, D. (2022, May). Seven Segment Display Detection and Recognition via Deep Learning Technique. In 2022 19th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON) (pp. 1-4). IEEE.

Thimbleby, H. (2013, April). Reasons to question seven segment displays. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 1431-1440).

Wan-Fu, H. (2011, September). The design of a six-digit digital clock with a four-digit seven-segment display module. In 2011 International Conference on Electrical and Control Engineering (pp. 2656-2659). IEEE.

Wannachai, A., Boonyung, W., & Champrasert, P. (2020). Real-Time Seven Segment Display Detection and Recognition Online System Using CNN. In Bio-inspired Information and Communication Technologies: 12th EAI International Conference, BICT 2020, Shanghai, China, July 7-8, 2020, Proceedings 12 (pp. 52-67). Springer International Publishing.

Wannachai, A., Boonyung, W., Yawootti, A., Nuangpirom, P., & Munsin, R. (2022, July). Seven-segment Display Automatic Detection and Interpretation System using CNN-GO. In 2022 6th International Conference on Green Technology and Sustainable Development (GTSD) (pp. 793-798). IEEE.

Yousefi, J. (2011). Image binarization using Otsu thresholding algorithm. Ontario, Canada: University of Guelph, 10, 9.