Issues in Information Systems
Volume 26, Issue 3, pp. 158-165, 2025

DOI: https://doi.org/10.48009/3_iis_2025_2025_113

Inexpensive real-time AI solutions using low-power edge devices

Brian Rodriguez, *Texas A&M University-Kingsville, nuri.yilmazer@tamuk.edu*
Nuri Yilmazer, *Texas A&M University-Kingsville, brian.rodriguez@students.tamuk.edu*
Joon-Yeoul Oh, *Texas A&M University-Kingsville,joon-yeoul.oh@tamuk.edu*

This paper explores the feasibility of deploying neural networks on low-cost embedded devices, using Raspberry Pi 5 as a platform for real-time digit recognition and generation. Leveraging accessible tools such as TensorFlow and Keras, a feed-forward neural network and a convolutional neural network (CNN) were implemented to classify handwritten digits from the MNIST dataset. A Raspberry PI Camera module enabled live digit recognition, demonstrating how AI navigated automation can be achieved with minimal hardware investment. Additionally, a generative adversarial network (GAN) was trained to produce synthetic handwritten digits, showcasing the creative potential of edge devices for content generation tasks. This proof-of-concept highlights how modern AI frameworks simplify the development and deployment of machine learning solutions on resource-constrained devices. The study underscores the potential of edge AI for business applications by offering energy-efficient, offline or online capable systems that enhance data privacy and reduce dependence on cloud infrastructure, enabling scalable and cost-effective automation in a variety of sectors.

Keywords: AI use, AI infrastructure, AI platform and tools, handwritten digit recognition, raspberry pi

Introduction

Artificial Intelligence (AI) has quickly evolved from a research-driven field into a core component of modern business processes. One of the most impactful advancements within AI is the use of neural networks for tasks such as image recognition, classification, and content generation. Convolutional Neural Networks (CNNs) have particularly demonstrated exceptional performance in computer vision tasks due to their ability to automatically learn spatial hierarchies of features from input images (Garg, Gupta, Saxena, & Sahadev, 2019). The MNIST handwritten digit dataset has become a standard benchmark for evaluating image classification models, providing a foundation for academic research and practical applications in industries seeking to automate visual data processing tasks (Dar et al., 2024). As AI tools like TensorFlow and Keras have become widely accessible, the barrier to implementing neural networks in real world scenarios has significantly lowered, allowing businesses to adopt AI driven solutions even on constrained hardware like the Raspberry Pi.

Recent studies have investigated various strategies to optimize CNN architectures for the MNIST dataset, focusing on hyperparameter tuning, data preprocessing, and training methodologies to maximize classification accuracy while minimizing computational overhead. Dar et al. (2024) emphasized the importance of techniques such as batch normalization, learning rate decay, and early stopping in enhancing CNN performance on resource-limited devices. These optimizations are of high importance when deploying AI models on embedded systems where processing power and energy consumption are key constraints. Similarly, Garg et al. (2019) demonstrated how CNNs trained on MNIST can effectively validate and classify new, unseen datasets, highlighting the versatility and adaptability of these models in dynamic business environments. The combination of lightweight neural network models with low-cost hardware presents a promising avenue for businesses to implement AI-driven automation without the need for expensive cloud infrastructure.

This paper presents a proof-of-concept implementation of neural networks on a Raspberry Pi, focusing on real-time digit recognition using CNNs and digit generation using Generative Adversarial Networks (GANs). Drawing inspiration from foundational resources on neural networks and deep learning (Nielsen, n.d.), this project demonstrates how modern AI frameworks simplify the development and deployment of machine learning solutions on embedded devices. Beyond the technical implementation, the paper discusses the potential business applications of edge AI systems, emphasizing benefits such as offline data processing, enhanced data privacy, energy efficiency, and reduced operational costs. By showcasing the feasibility of deploying neural networks on affordable hardware, this study aims to illustrate how any business can leverage AI technologies to streamline processes and improve decision-making without significant financial investment.

Convolutional Neural Networks and MNIST Classification

The MNIST dataset consists of 70,000 grayscale images of handwritten digits which is ranging from 0 to 9. Each image is preprocessed and normalized to make it standard in size to 28x28 pixels in size. Images are labeled indicating the correct digit. The dataset is typically divided into two parts and 60,000 images are used for training and remaining 10,000 are reserved for testing [7]. A sample of the MNIST data is shown below in Figure 1.

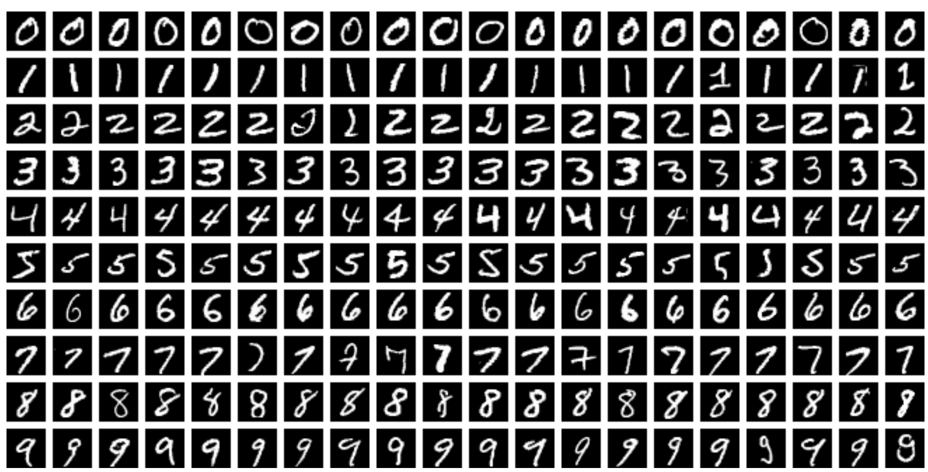

Figure 1. Sample images from MNIST test dataset [7].

Convolutional Neural Networks (CNNs) have become a cornerstone in image classification tasks, particularly with datasets like MNIST. Dar et al. (2024) emphasized the importance of hyperparameter optimization techniques such as batch normalization, learning rate decay, and early stopping to enhance CNN performance on the MNIST dataset. Their study demonstrated that careful tuning of these parameters can lead to significant improvements in classification accuracy. Similarly, Garg, Gupta, Saxena, and Sahadev (2019) validated the efficacy of CNNs trained on MNIST for recognizing handwritten digits, highlighting the model's robustness and adaptability to new datasets.

Implementing Neural Networks on Raspberry Pi

The Raspberry Pi, a low-cost, single-board computer, has gained popularity for deploying AI models at the edge. Ju et al. (2022) introduced "TripleNet," a lightweight CNN architecture optimized for Raspberry Pi, which achieved reduced inference times without compromising accuracy. Their work demonstrated that with appropriate model compression and optimization, Raspberry Pi can effectively handle image classification tasks. Additionally, optimizing deep learning models for Raspberry Pi involves techniques like pruning and quantization to reduce computational load and energy consumption, making it feasible to run complex models on resource-constrained devices (Ameen et al., 2023).

Generative Adversarial Networks for Data Generation and Recognition

Generative Adversarial Networks (GANs) are a class of machine learning models that consist of two competing neural networks: a generator and a discriminator. The generator learns to create realistic synthetic data, while the discriminator attempts to distinguish between real and generated data. Through this adversarial process, GANs can produce high-quality synthetic images, making them valuable for tasks such as image generation, data augmentation, and unsupervised learning (Popuri & Miller, 2023). Beyond image synthesis, GANs have been applied to multi-modal analysis, text-to-image generation, and improving robustness in recognition tasks, demonstrating their scalability across various domains.

Specific to handwritten digit generation, Song, Su, and Zhang (2021) developed a GAN model tailored for the MNIST dataset. Their experimental analysis focused on generating realistic handwritten digits by optimizing the architecture of both the generator and discriminator networks. Key findings from their study emphasized the impact of training stability and network design on the quality of generated images. The research showcased the potential of GANs for data augmentation in applications requiring robust image synthesis, Incorporating GANs into edge devices like Raspberry Pi enables localized content generation, enhancing data privacy and reducing reliance on cloud-based resources.

Methodology

The objective of this study was to develop and deploy neural network models for handwritten digit recognition and generation on a Raspberry Pi microcontroller. The methodology encompassed three primary phases: model development, model optimization and export, and hardware deployment with live testing.

The project began with the development of a feed-forward neural network (FFN) and a convolutional neural network (CNN) using Python libraries TensorFlow and Keras. The MNIST handwritten digit dataset was utilized for both training and validation. Images were preprocessed through grayscale normalization, resizing, and reshaping to ensure compatibility with the CNN input layers. For the CNN architecture, a series of convolutional layers were followed by max-pooling layers and fully connected layers, optimized to balance classification accuracy with computational efficiency on resource-constrained hardware. Additionally, a generative adversarial network (GAN) was designed and trained to generate synthetic handwritten digits, with training conducted on a separate workstation to manage the heavier computational load of adversarial training.

After training, all models were saved in H5 file format, facilitating seamless loading and deployment on the Raspberry Pi. While no aggressive quantization or pruning was applied during this project, standard

Keras model-saving procedures ensured that the trained weights and architecture could be efficiently transferred to the target device. Preprocessing scripts were developed to handle live image data captured by the Raspberry Pi Camera Module, standardizing inputs before feeding them into the CNN for inference. For the GAN model, only the generator component was deployed to minimize resource consumption during image generation tasks. The trained models were deployed onto a Raspberry Pi microcontroller, leveraging its GPIO interface to integrate with the Pi Camera for real-time digit recognition. Python scripts utilizing TensorFlow Lite runtime were executed on the Raspberry Pi to perform live inference, with outputs displayed via console and optional image overlay. System performance metrics, including inference time and resource utilization, were monitored to assess real-time feasibility. The CNN was able to achieve real-time digit classification with acceptable latency, while the GAN-generated synthetic digits were evaluated for visual fidelity directly on-device. This deployment demonstrated the Raspberry Pi's capability to execute deep learning tasks effectively in a low-power, offline environment.

Results and Analysis

The implementation of neural networks on the Raspberry Pi yielded promising results in both digit recognition and digit generation tasks. The convolutional neural network (CNN) achieved high classification accuracy on the MNIST dataset, maintaining reliable performance during live inference using the Pi Camera. The real-time recognition system successfully identified handwritten digits with negligible latency, averaging approximately 130 milliseconds per prediction, as observed during testing. The model exhibited exceptional accuracy, correctly classifying digits in various lighting conditions demonstrating its practical viability for embedded applications. For the generative adversarial network (GAN), the generator model was deployed onto the Raspberry Pi to produce synthetic handwritten digits. Despite the resource constraints of the device, the generated images displayed clear visual resemblance to authentic MNIST digits. While inference times were longer than those of the CNN, the system effectively generated high quality digit samples, confirming that such generative tasks are feasible on edge devices when appropriately optimized. The system uses a Raspberry Pi microcontroller connected to a Pi Camera mounted on a custom 3D-printed holder. This hardware configuration enables live image capture of handwritten digits for real-time classification as shown in Figure 2.

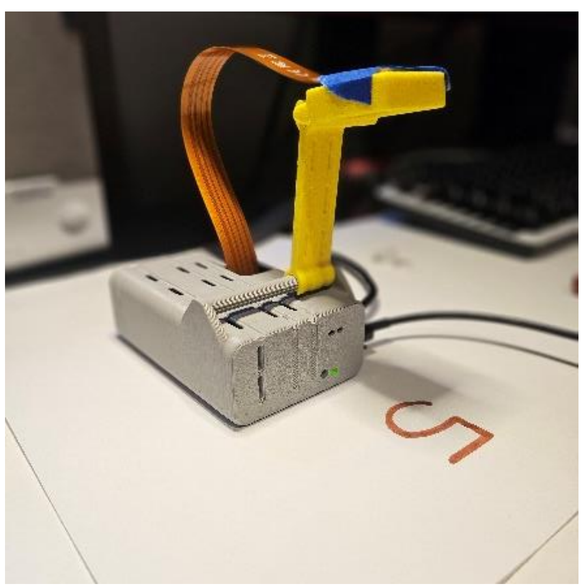

Figure 2. Raspberry Pi- handwritten digit recognition setup w/integrated Pi Camera & 3D printed Casing.

The original image of the handwritten digit '5', captured using the Raspberry Pi camera, is shown in Figure 3. This is the raw, unprocessed image, which may be in grayscale or color. Since our system is designed specifically for grayscale images, the input must first be converted accordingly. While the current design handles only grayscale inputs, it can be extended to support color images with appropriate modifications to the network architecture and preprocessing pipeline.

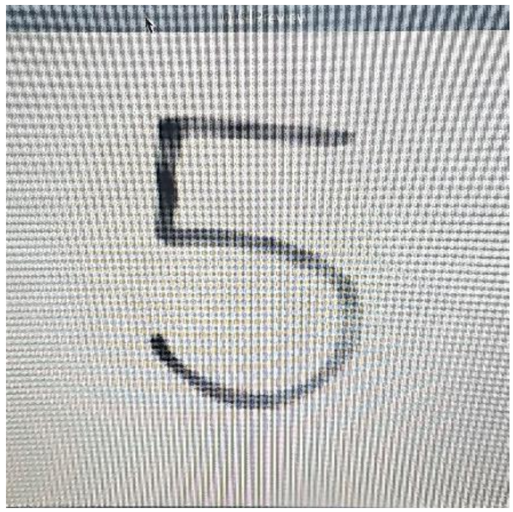

Figure 3. Captured raw image of handwritten digit "5" using the Raspberry Pi Camera.

The handwritten digit is first positioned in front of the Raspberry Pi camera for image acquisition. The captured high-resolution image is then subjected to a preprocessing pipeline to enhance compatibility with the CNN model used for digit recognition. As part of this process, the image is resized and formatted to match the standard MNIST specifications, with a resolution of 28×28 pixels as shown in Figure 4.

Figure 4. Preprocessed 28x28 pixel grayscale image prepared for CNN input.

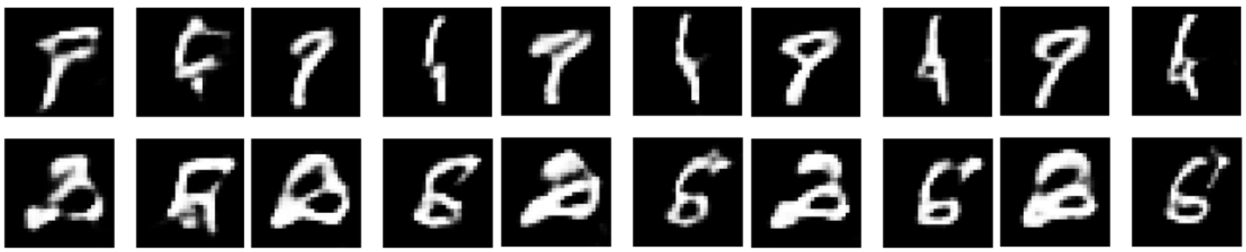

Figure 5. Sample outputs from early training epochs of the GAN model generating handwritten digits.

The generated images depict the GAN's initial attempts at replicating MNIST-style handwritten digits. At this early training stage, the outputs are of lower quality, with distorted and incomplete digit structures. Progressive refinement through continued adversarial training improves the fidelity of generated digits over time.

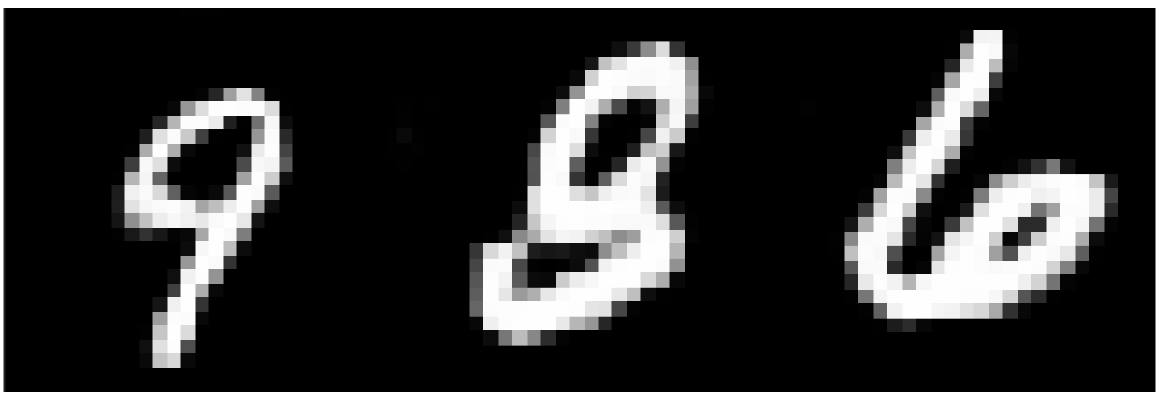

Figure 6. Handwritten digits produced by GAN model after 100 training epochs, running on Raspberry Pi.

The samples illustrate significant improvement in digit clarity and structure compared to early training outputs. The GAN successfully generates realistic MNIST-like handwritten digits, demonstrating the Raspberry Pi's capability to handle generative neural network tasks after sufficient training and model optimization.The successful deployment of CNN and GAN models on the Raspberry Pi highlights the potential that low-cost, low-power devices have for executing AI oriented tasks traditionally reserved for cloud-based infrastructures. The CNN's ability to perform real-time digit recognition illustrates how embedded systems can be used for applications such as automated data entry, form recognition, and portable scanning solutions. This capability reduces reliance on external servers, lowering latency and preserving data privacy by processing sensitive information locally.

The deployment of the GAN model, albeit more computationally intensive, demonstrated the feasibility of on-device content generation. This has implications for tasks like synthetic data augmentation, offline visualization tools, and user-interactive AI applications where connectivity is limited. Furthermore, the works's use of TensorFlow and Keras streamlined the development workflow, allowing trained models to be exported in standardized formats and efficiently deployed on embedded hardware with minimal additional optimization.

In addition to performance, energy consumption was an important consideration. Although precise power measurements were not recorded, the Raspberry Pi's inherent design prioritizes low power usage compared to traditional GPU-powered cloud systems. This inherent energy efficiency, combined with the platform's portability and affordability, positions edge AI solutions as practical alternatives for small to medium-sized businesses seeking scalable automation.

The developed system can be adapted to recognize English letters, characters in different languages, or other types of symbols. As long as there is sufficient data available for training, validation, and testing, the neural network can be retrained for the new classification task. While the core design logic of the system remains the same, modifications may be necessary to account for the increased complexity of the new data. These adjustments might include changes to the neural network architecture, such as the number of layers, nodes, or learning rate, depending on the nature of the new dataset. This makes the system is flexible and can be generalized for various classification problems beyond handwritten digit recognition.

Overall, this proof-of-concept demonstrates that with careful model selection and hardware-aware implementation strategies, even computationally limited platforms like the Raspberry Pi can effectively perform complex AI tasks. The findings highlight the practicality of edge AI for real-world business applications, paving the way for further exploration into more demanding models and datasets on constrained hardware.

Conclusion

This study demonstrated the feasibility of deploying neural networks on inexpensive, resource-constrained devices like the Raspberry Pi for real-time handwritten digit recognition and generation. By leveraging TensorFlow and Keras, convolutional neural networks (CNNs) were successfully trained and deployed to perform live digit classification with minimal latency, while generative adversarial networks (GANs) were utilized to create realistic handwritten digit images. These results align with existing research emphasizing the effectiveness of CNNs for image classification tasks, particularly with the MNIST dataset (Dar et al., 2024; Garg et al., 2019) and extend the application to edge devices beyond traditional high-performance computing environments.

Furthermore, the generative capabilities of GANs, as explored by Song et al. (2021) and Popuri and Miller (2023), were replicated on the Raspberry Pi, underscoring the potential of embedded systems for content generation tasks. While performance optimizations such as quantization and pruning offer avenues for further enhancement, this study functioned as a proof-of-concept that highlighted the practical benefits of edge AI deployments, including reduced latency, enhanced data privacy, and operational cost savings. As energy efficiency becomes a growing concern in AI deployment, platforms like the Raspberry Pi offer a practical and scalable solution for running efficient and accessible machine learning applications directly at a data source, supporting a wide range of business and real-world use cases.

References

Dar, E. (2024), Optimizing Convolutional Neural Network Hyperparameters for MNIST Image Classification: Insights and Implications. *2024 2nd International Conference on Advances in Computation, Communication and Information Technology (ICAICCIT)*, Faridabad, India, pp. 16-21, doi: 10.1109/ICAICCIT64383.2024.10912098.

Garg,A. Gupta D., Saxena,S. & Sahadev, P. (2019). Validation of Random Dataset Using an Efficient CNN Model Trained on MNIST Handwritten Dataset. *2019 6th International Conference on Signal Processing and Integrated Networks (SPIN)*, Noida, India, pp. 602-606, doi: 10.1109/SPIN.2019.8711703.

Nielsen, M. (2015). *Neural Networks and Deep Learning*. Determination Press.

Ju, R. Lin,Y., Jian,T. & Chiang, J.-S. (2022). Efficient convolutional neural networks on Raspberry Pi for image classification. *arXiv preprint arXiv:2204.00943*. https://arxiv.org/abs/2204.00943

Ameen, S., Siriwardana, K., & Theodoridis, T. (2023). Optimizing deep learning models for Raspberry Pi. *arXiv preprint arXiv:2304.13039*. https://arxiv.org/abs/2304.13039

Popuri, A. & Miller, J. (2023). Generative Adversarial Networks in Image Generation and Recognition. *International Conference on Computational Science and Computational Intelligence (CSCI)*, Las Vegas, NV, USA, pp. 1294-1297, doi: 10.1109/CSCI62032.2023.00212.

https://en.wikipedia.org/wiki/MNIST_database