A Multi-Source Digit Recognition Framework Midterm Report - ECE 3991

Hudson Dalby

***Abstract*—Digit recognition has been a defining problem in the history of computer vision and machine learning. Convolutional Neural Networks (CNNs) have emerged as the primary solution, establishing a model for image pattern recognition that has driven advancements in document reading, postal code classification, and system automation. However, most existing systems only account for a singular visual domain—such as handwritten, printed, or digital displays—and incorrectly identify digits when applied to diverse real-world inputs. This project seeks to unify the strengths of existing frameworks by developing a single model with the ability to distinguish between different visual domains and styles in a single framework. Implementation and evaluation of this model are not yet completed, this model aims to function reliably on multiple existing datasets as well as procedurally generated datasets. This paper discusses the existing digit recognition frameworks, outlines the motivation and design of the proposed multi-domain approach, and discusses future developments intended to extend the system's adaptability and robustness**

***Index Terms*—Digit recognition, Optical Character Recognition, MNIST, SVHN, EMNIST, Convolutional Neural Network, LeNet**

I. Introduction

Digit recognition has long been regarded as a foundational problem in computer vision, tracing back to one of the earliest forms of optical character recognition (OCR) in 1914, when Emanuel Goldberg invented a device capable of reading characters and converting them into telegraph code. This system was a catalyst for development of subsequent OCR systems to automate tasks such as postal code and bank check reading.

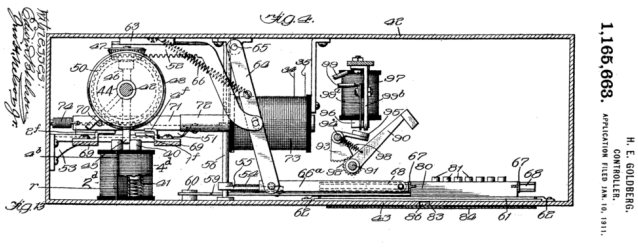

Fig. 1. Emanuel Goldberg's 1914 optical character recognition (OCR) machine [1].

The simplicity and accessibility of digit recognition appealed to many, becoming a benchmark for evaluating machine learning algorithms. This led to the conceptualization of the first Convolutional Neural Network (CNN) in 1989, which later became the LeNet series of architecture developed by Yann LeCun and his research group

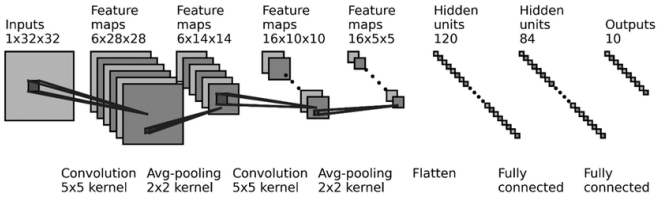

Fig. 2. Visualization of LeNet-5 architecture as illustrated in many literature sources [2].

Improvements upon the LeNet architecture drove rapid progression in both classification and visual pattern recognition, establishing CNNs as core learning frameworks for applications in modern artificial intelligence. CNNs are comprised of several key components [3] that operate synchronously to process visual data:

- **Convolutional Layers** - Apply set filters to the input image to recognize spatial features such as edges, corners, and textures.
- **Activation function** - Applied after each convolution layer to introduce nonlinearity, allowing adaptations to be made to complex pattern recognition in data.
- **Pooling Layers** - Reduce dimensions of the input data for more efficient computations and improved response to input variations.
- **Fully connected layers** - Combine extracted features for classification or regression.

Use of advanced CNN architecture and related strategies have resulted in near-human accuracy in digit recognition. Datasets such as MNIST and SVHN have provided consistent benchmarking evaluations, and models like LeNet have demonstrated incredible precision in recognition within these domains. However, most existing systems are typically trained and optimized for a single visual domain-handwritten, printed, or digital displays-and perform poorly when exposed to inputs from a visually distinct domain. This lack of adaptability limits their practicality in dynamic, real-world environments.

The motivation behind this project is to bridge this gap between specialized recognition systems by developing an adaptive framework. The envisioned system aims to achieve good performance across handwritten, printed, and digitally displayed inputs. In addition to established datasets such as MNIST and SVHN, this system will rely on procedurally generated data sets to enhance recognition under diverse conditions.

The remainder of this paper is organized as follows: Section

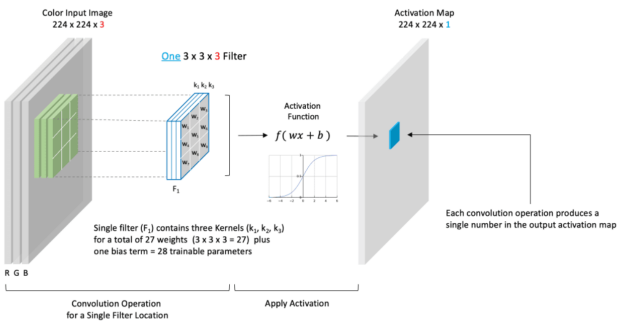

Fig. 3. Diagram of convolutional layer in modern CNNs [4].

II reviews existing digit recognition methods and discusses domain-specific limitations. Section III presents the proposed multi-domain framework, details datasets for training, and outlines key considerations that must be taken. Section IV concludes with discussion of future work and midterm report inspiration.

II. Related Work

Digit recognition has played an essential role on the evolution of computer vision, machine learning, and artificial intelligence. The introduction of the MNIST dataset the 1990's was a major milestone, establishing a standardized benchmark for evaluating system recognition capabilities. The dataset is comprised of 70,000 standardized, handwritten, black and white digits (0-9), each normalized to a size of 28x28 pixels, making it ideal for training and testing machine learning systems.

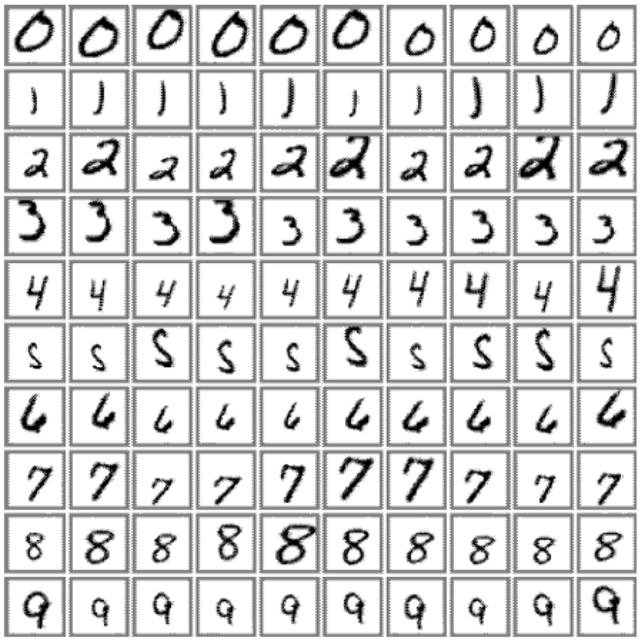

Fig. 4. Examples of digits from MNIST dataset [2].

The revolutionary LeNet-5 system [2], developed by Yann LeCun and his research group, was trained on the MNIST dataset and was able to achieve near-human accuracy in digit recognition. As one of the earliest systems utilizing a Convolutional Neural Network (CNN), LeNet demonstrated the effectiveness of deep learning in visual computing. However, its performance was limited to handwritten digits within the training dataset, inspiring the development of future CNN systems to overcome challenges for applicability beyond the MNIST dataset.

The success of the LeNet system inspired numerous domain-specific variants. A key system developed in 2013 [5] extended CNN image recognition to street view imagery by training the model on the Street View House Numbers (SVHN) dataset. The SVHN set is a collection of several hundred-thousand cropped images of real-world house numbers provided from Google Street View. This system introduced variables such as lighting, color, and perspective distortions to digit recognition. These elements added new complexity and advanced the field towards real-world application.

Fig. 5. Examples of digit imagery from SVHN dataset [4].

In 2020, further research [6] applied CNNs to seven-segment LED displays, targeting industrial and embedded applications. This approach integrated HSV-based preprocessing with deep learning to identify segmented numeric patterns in varied environments.

A collection of common digit recognition systems and their benchmark performance metrics is displayed below:

Each system variant demonstrates incredibly high accuracy in data sources within their own visual domain, but performance declines with generalized data. For example, a model trained on MNIST struggles to account for environmental com-

TABLE I
BENCHMARK PERFORMANCE OF COMMON DIGIT RECOGNITION SYSTEMS

System / Method	Dataset / Domain	Result	Notes / Reference
LeNet-5 (CNN)	MNIST (handwritten)	Error: 0.8%	Foundational CNN architecture for OCR [2].
High-Performance CNN (Ciresan et al.)	MNIST	Error: 0.35%	Deep GPU-trained CNN achieving near-human accuracy [7].
Simple CNN Ensemble	MNIST	Acc: 99.87%	Ensemble averaging of lightweight CNNs [8].
Hybrid CNN + SVM	MNIST	Acc: 99.3%	CNN feature extractor with SVM classifier [9].
Domain-Adversarial CNN	MNIST ↔ SVHN	Acc: ∼97%	Cross-domain digit adaptation [10].
Capsule Network	MNIST	Acc: 99.75%	Handles rotation and spatial hierarchy [11].
MobileNetV2 (Fine-tuned)	SVHN (printed)	Acc: ∼98%	Efficient CNN for edge deployment [12].
Seven-Segment Recognition (Fuchs et al.)	LED displays	Acc: 87.17%	HSV filtering + CNN for LED digits [6].
Rule-based Segment Decoder	LED displays	Acc: 80–90%	Threshold/contour-based traditional vision approach.
Proposed Unified Multi-Source Model	MNIST, SVHN, LED (synthetic)	*(Planned)*	Cross-domain CNN using combined datasets and procedural data generation.

plexity of SVHN or LED digit recognition without algorithm retraining. Cross-domain digit recognition was first explored in 2016, achieving solid performance across multiple datasets. However, extensive training data and visual similarities were a requirement for consistent results.

Overall, prior research highlights the effectiveness and limitations of domain-specific CNNs. Despite the successful development of single-domain digit recognition, no existing framework incorporates handwritten, printed, and seven-segment-into a single model. Current datasets are largely limited to a particular visual type, and image processing techniques rarely transfer between domains. This gap in CNN development motivates the need for robust, multi-source digit recognition architecture capable of handling data to more effectively adapt to real-world environments.

III. PROPOSED APPROACH

To develop a multi-source digit recognition architecture, this project builds upon the foundational LeNet-5 architecture, as it a well-established, effective system of visual computation. The baseline system is adapted to handle multiple datasets. MNIST, SVHN, and LED digit datasets each represent a distinct visual domain. An adaptive routing system is proposed to parse each input image, classifying it as one of the most visually similar to a style from a database, then selecting the corresponding recognition algorithm.

The proposed framework will follow a modular architecture consisting of three primary components: the preprocessing unit, the adaptive routing classifier, and the recognition network. The preprocessing unit standardizes all incoming images through normalization and scaling. The adaptive router then performs domain sorting using visual such as edge density, color distribution, and contrast. Finally, the recognition component—composed of domain-specific CNN models derived from the LeNet-5 baseline—processes the directed input to produce a digit classification. This modular design allows independent optimization of each component and simplifies future expansion to new visual domains.

In later development stages of the project, the system will incorporate artificially generated images of digits in different contexts, lighting, and viewing perspectives. These procedurally generated datasets allow the router to select an algorithm to best classify the digit or, if new features are present, create a hybrid category using weighted combinations of existing algorithms. The proposed routing system will be modeled as a lightweight CNN classifier.

An important challenge to address within this multi-domain framework is input normalization. Input images from different sources, especially generated, vary in qualities such as resolution, scale, and aspect ratio. To achieve consistency in input data, bilinear interpolation will be used to manipulate the image to a consistent size across all data sets, maintaining relative pixel density to preserve features for a correct digit identification. Smaller images, such as those from the MNIST dataset , are only 28x28 pixels and will require upscaling. Conversely, larger input images with distortion and different perspectives must be downscaled without loss of identifying features. Additional factors, such as image color, must be accounted for in normalization to preserve accuracy. Developing a robust normalization system is essential to ensure cross-domain accuracy in digit recognition.

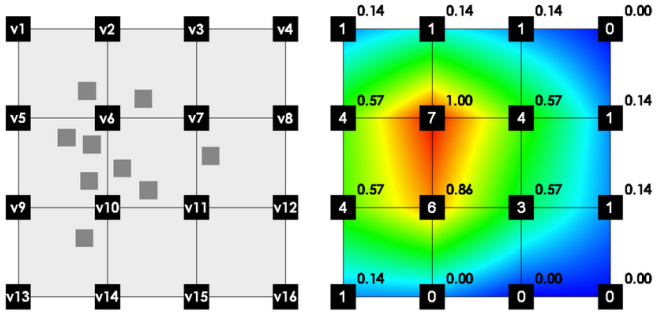

Fig. 6. Visualization of bilinear interpolation [13].

As shown in Figure 7, bilinear interpolation computes weighted averages based on the proximity of surrounding pixel intensities. This preserves continuity and prevents aliasing effects during upscaling or warping, which is particularly critical for maintaining feature clarity in small digits.

After normalization, the router anticipates the visual style most closely resembling the input data, activating the corresponding recognition algorithm. For cases that do not easily fall into an existing category, a combination of weighted algorithms from each set of visually similar data is used to infer the most probable digit. A confidence interval will be determined for the classification, and if it falls below a certain threshold,

the system will designate a new category. This dynamic system enables continual improvement and incremental learning.

An essential component of this architecture is the activation function, which allows the CNN to learn and represent complex relationships between data inputs. Without nonlinear activation, a deep network would collapse into a linear transformation regardless of its depth, severely limiting its ability to distinguish between varied visual patterns across domains.

The activation function introduces nonlinearity after each convolutional layer, allowing the system to capture and distinguish features such as curvature, sharpness, and contrast. The general activation function for CNN networks is referred to as the ReLU function, and is defined as:

$$a_{i,j,k} = \mathrm{ReLU}(z_{i,j,k}) = \max(0,\, z_{i,j,k}), \tag{1}$$

Where $z_{i,j,k}$ represents the initial value at spatial coordinates (i,j) and feature designation k. This function directly outputs the input directly only if positive, reducing remaining values to zero. This simultaneously preserves key visual features while increasing computational efficiency, keeping only portions that contribute positively to feature detection.

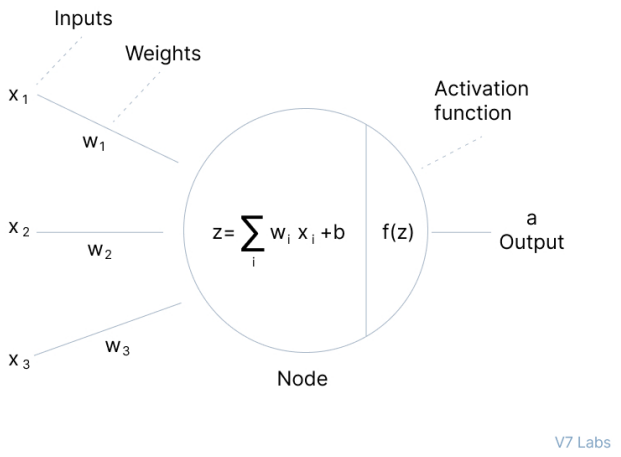

Fig. 7. Visualization of activation function (ReLU) [14].

In the proposed multi-domain system, ReLU activation functions are applied after each convolutional and connective layer. This ensures the distinct visual structures from handwritten, printed, and LED digits are effectively captured and propagated throughout the network.

The proposed system is expected to retain similar accuracy to existing CNN systems shown in Table II. However, performance is likely to vary by dataset, as systems have varying degrees of accuracy tied to the complexity of the images. The goal of this work is not to maximize accuracy of a single dataset, but to design an adaptive framework capable of transfer between handwritten, printed, and digital visual domains.

Implementation will be conducted in Python using TensorFlow or PyTorch. The training datasets will be manipulated with random rotations, scaling, and contrast variations to simulate environmental diversity. The system will run on GPU-accelerated hardware to enable parallel training of multiple domain-specific networks. All experimental configurations, including learning, dataset splits, and routing methods, will be documented for reproducibility.

TABLE II
RELATIVE ACCURACY ACROSS VISUAL DOMAINS

Domain	**Dataset Type**	**Accuracy (%)**
Handwritten (MNIST)	Black-and-white, 28×28	99.3
Printed (SVHN)	Natural scenes, color	97.8
Digital Display (LED)	Seven-segment, low contrast	87.5
Proposed Unified Model	Cross-domain hybrid	**90-95**

IV. FUTURE OF THE PROJECT

Future work on this project will focus on extending the system's adaptability, scalability, and overall performance through the inclusion of new datasets, optimization techniques, and continual learning mechanisms. The current proposed implementation integrates three primary visual domains—handwritten, printed, and digital display digits—there remains significant potential for broadening the scope of the model to encompass new forms of visual data and more complex feature variations.

A key focus of future development is the integration of additional datasets to increase variability and enhance generalization across diverse application environments. Incorporating new representations of numerical data—such as Roman numerals, tally marks, and hand gesture–based finger counting—would extend the system's capabilities beyond standard digits, enabling recognition across a broader range of contexts.

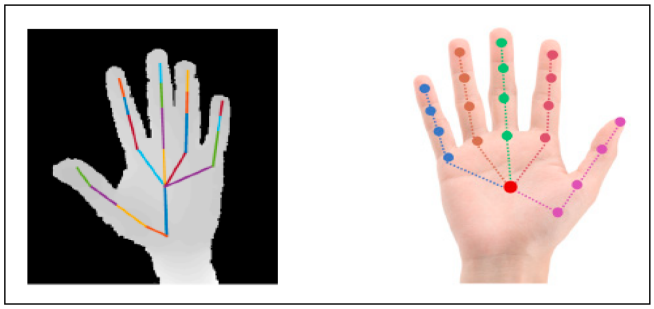

Fig. 8. Visualization of gesture recognition [15].

An existing gesture recognition system could be incorporated into the framework to extend its functionality beyond written digits. Integrating gesture-based input would enable recognition of hand and finger movements corresponding to numerical values, allowing the framework to operate within adaptive, human-centered systems such as assistive interfaces or contactless input devices.

Another area of interest is the implementation of continual learning and self-adaptation. Instead of retraining the entire model when exposed to new data sources, the system could

incrementally update only the relevant submodules or add new domain classifiers. This modular retraining approach would enable seamless integration of future datasets.

Finally, future iterations of the project would include comprehensive performance benchmarks directly against modern architectures, not only assessing accuracy, but speed and efficiency. Following these evaluations, development on a lightweight application of the framework for embedded digit recognition applications.

A. Midterm Report

This report serves as a hybrid between the IEEE research paper format covered in class, and a personal exploration of machine learning concepts. The goal was to gain familiarity with formal research writing and technical communication while investigating a topic that has long been an area of interest to me—computer vision. Rather than presenting completed experimental results, this work focuses on understanding the structure and methodology of a research paper, emphasizing the reasoning and planning stages of a proposed system design.

The motivation to study a multi-source digit recognition framework came from curiosity about how visual data recognition is achieved. At first, the idea of implementing a visual learning model seemed beyond my current experience, given the depth of machine learning theory and rapidly accelerating complexity of systems. However, through the process of researching architectures and data handling strategies, I gained valuable insight into how these systems operate and how such a project might be approached in the future.

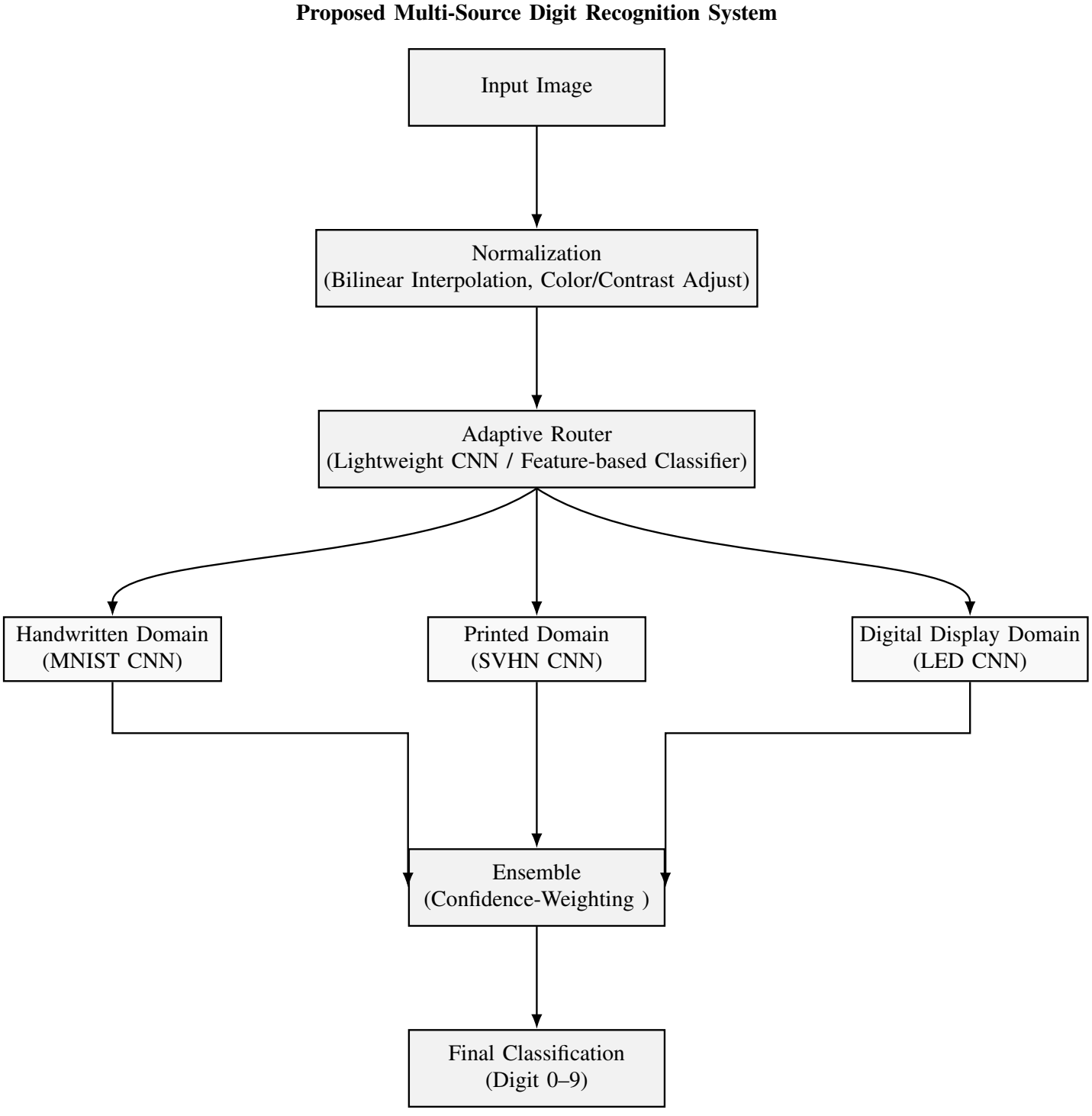

Fig. 9. Proposed multi-source digit recognition architecture. The pipeline proceeds top-to-bottom: images are normalized, routed to a domain-specific CNN (MNIST, SVHN, or LED), integrates them into the algorithmic data, and outputs the determined classification.

REFERENCES

[1] E. Goldberg, "Machine for reading characters and converting them into telegraph code." German Patent No. 293,853, 1914. Often regarded as the first optical character recognition system.

[2] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[3] B. Kromydas, "Understanding convolutional neural networks (cnn)." https://learnopencv.com/understanding-convolutional-neural-networks-cnn/, 2023.

[4] Y. Netzer, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Street view house numbers (svhn) dataset — example images." http://ufldl.stanford.edu/housenumbers/, 2011. Accessed: 2025-10-16.

[5] I. Goodfellow, Y. Bulatov, J. Ibarz, S. Arnoud, and V. Shet, "Multi-digit number recognition from street view imagery using deep convolutional neural networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2013.

[6] R. Fuchs, J. Wojak, O. Gühne, and N. Teichert, "Robust recognition of seven-segment digits using classical and deep learning methods," *Sensors*, vol. 20, no. 15, p. 4270, 2020.

[7] D. Ciresan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3642–3649, 2012.

[8] S. An and J. Kim, "Ensemble of simple convolutional neural networks for mnist digit recognition," *arXiv preprint arXiv:2008.10400*, 2020.

[9] D. Ferrari and A. Lira, "Hybrid cnn-svm model for handwritten digit recognition," *Pattern Recognition Letters*, vol. 112, pp. 61–67, 2018.

[10] Y. Ganin and V. Lempitsky, "Domain-adversarial training of neural networks," *Journal of Machine Learning Research*, vol. 17, no. 59, pp. 1–35, 2016.

[11] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, pp. 3856–3866, 2017.

[12] A. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenetv2: Inverted residuals and linear bottlenecks," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4510–4520, 2018.

[13] J. F. Vicent, "Visualization based on grids with bilinear interpolation (figure)." https://www.researchgate.net/figure/sualization-based-on-grids-with-bilinear-interpolation_fig2_283288506, 2015. Accessed: 2025-10-16.

[14] V. Labs, "Activation functions in neural networks — illustrative diagram." https://www.v7labs.com/blog/neural-networks-activation-functions, 2021. Accessed: 2025-10-16.

[15] M. Oudah, A. Al-Naji, and J. Chahl, "Hand gesture recognition based on computer vision: A review of techniques," *Journal of Imaging*, vol. 6, no. 8, p. 73, 2020.