Research Article

Dhouha Ben Noureddine*

Handwritten digit recognition: Comparative analysis of ML, CNN, vision transformer, and hybrid models on the MNIST dataset

https://doi.org/10.1515/jisys-2024-0411
received October 01, 2024; accepted May 26, 2025

Abstract: Handwritten digit recognition (HDR) remains challenging due to variations in writing styles. To address this challenge, this study comprehensively compares ML (ML) and deep learning (DL) models. We explored a variety of approaches. We evaluated these models on the Modified National Institute of Standards and Technology (MNIST) and Extended Modified National Institute of Standards and Technology (EMNIST) datasets to assess their generalization capabilities. Initially, we investigated standalone ML and DL models trained from scratch to learn features directly. Logistic regression (LR) achieved an accuracy of 92.5% on MNIST and 86.63% on EMNIST. A multi-layer perceptron demonstrated improved performance with 98.10% accuracy on MNIST. Convolutional neural networks exhibited superior performance, reaching 99.90% accuracy on MNIST and 99.57% on EMNIST. To further enhance performance, we explored ensemble learning techniques, combining CNNs with RF (98.20 and 99.86% accuracy on MNIST and EMNIST, respectively), LR (88.67 and 99.79% accuracy on MNIST and EMNIST, respectively), and VC (99.27 and 99.83% accuracy on MNIST and EMNIST, respectively). We then introduced a ViT model, leveraging self-attention for long-range dependency modeling, achieving an accuracy of 98.70% on MNIST and 99.58% on EMNIST. Finally, we proposed a hybrid model combining CNN and ViT, that yielded the highest accuracy of 99.97% on MNIST and 98.26% on EMNIST. Throughout our experimentation, we employed various techniques such as regularization, weight initialization, and optimization strategies to improve model performance. The impact of each technique is analyzed and discussed. Overall, this study provides a comprehensive comparison of different HDR models, highlighting each approach's strengths and weaknesses. The results demonstrate the effectiveness of DL models, particularly CNNs and hybrid architectures, in achieving high accuracy in HDR.

Keywords: convolution neural networks, DL, EMNIST dataset, HDR, hybrid model, LR, ML, MNIST dataset, multi-layer perceptron, vision transformers

1 Introduction

The process of labeling input images from a given set of categories is known as image classification. It plays an important role in real-world applications such as robotics, object recognition, self-driving cars, and traffic signal processing. A core of image classification is feature extraction, which captures relevant information from images to create efficient representations for classification tasks. While computer scientists used to employ ML (ML) models that were good enough to solve this task, things have changed dramatically with the emergence of DL (DL), a subfield of artificial intelligence (AI). Its development resulted in performance levels that improved beyond all expectations, thereby eliminating much of the human engineering effort.

* **Corresponding author: Dhouha Ben Noureddine,** College of Computer and Information Sciences, Imam Mohammad Ibn Saud Islamic University (IMSIU), Riyadh, 13318, Saudi Arabia, e-mail: dnoureddine@imamu.edu.sa

Interestingly, these new technologies also include convolutional neural networks (CNNs), which have become the best-known DL-based image classification. CNNs are neural networks (NNs) designed to process image data using convolution operations available in at least one layer. They demonstrated this by learning how to create high-level abstractions from raw pixel data, allowing them to perform image classification tasks efficiently. As an alternative to CNNs in image classification, Dosovitskiy et al. [1] recently proposed the idea of using vision transformers (ViTs). Unlike CNNs, which use convolution operations to extract features, ViT works directly on image patches. This is based on the use of the attention mechanism. After that, these patches passed via a Transformer encoder block – akin to those used in NLP (NLP). This enables ViT to model long-range connections between different parts of the image, ultimately capturing more complex scene relationships than CNNs. The reason ViTs are still under development is that they show competitive performance in image classification on large datasets. These AI architectures can learn features without relying on technical filters and have the potential to improve global contextual understanding, making ViT a particularly exciting avenue for image classification.

In this article, we introduce a novel hybrid approach for HDR by combining CNNs and VViTs. The main contributions of our work are twofold:

(1) **New hybrid model**: we proposed a hybrid model that integrates CNNs and ViTs for image classification, leveraging the strengths of both architectures. This hybrid model aims to enhance feature extraction and capture both fine-grained details (via CNNs) and long-range dependencies (via ViTs).
(2) **Comparative analysis and evaluation**: we conducted a comprehensive comparative analysis of various image classification techniques in both ML and DL. Specifically, we evaluate the performance of several models, including standalone CNNs, ViTs, and the proposed hybrid model, on the widely used Modified National Institute of Standards and Technology (MNIST) and Extended Modified National Institute of Standards and Technology (EMNIST) datasets for HDR. The goal of our experiments is to provide insights into the advantages and limitations of each approach, thereby offering a deeper understanding of their capabilities in solving HDR tasks.
In our initial approach, we trained independent models from scratch, allowing them to learn features directly from the data. The second approach explored hybrid models that incorporate ensemble learning methods, where multiple models (each with distinct architectures or training strategies) are combined to enhance performance through aggregation or consensus-based decision-making. Through these strategies, we aim to investigate the strengths and weaknesses of various approaches to solving HDR problems.

The remainder of this article is organized as follows: Section 2 presents related work in image classification, particularly in the HDR field. After that, Section 3 discusses algorithms and models used in our study. Subsequently, Section 4 contains details about the dataset, the experimental setup, and our results presented therein. Then, Section 5 comprehensively discusses all the findings. Finally, our conclusions are summarized in Section 6.

2 Related work

There are many applications in computer vision (CV) where image classification is very important such as object detection, and digit recognition, medical image analysis. The MNIST handwritten digits recognition dataset is a well-explored benchmark for assessing the behavior of an image classification algorithm. Many researchers have proposed different contributions for MNIST digit recognition and achieved impressive results with diverse models such as NNs and support vector machines. For this reason, we highlight authors that specifically use the MNIST data set whereby each digit's "pixel values" represent features while the label is a number between 0 and 9. As these images have 28×28 pixels each pixel represents a feature, they become 784 features in total. Here are a few related works and notable approaches to MNIST image classification.

Reddy et al. [2] introduced a real-time system for HDR using a Support Vector Machine (SVM). Their approach consisted of two main stages: training and recognition. They trained their system using an SVM

classifier and then tested it on the MNIST dataset. The results were encouraging since they obtained 98.05% training accuracy and 97.83% test accuracy. They also applied the model for the recognition of user-provided handwritten digits in real time and obtained good results. Assegie and Nair [3] proposed a rule-based framework that is built using decision trees to classify the digits, where decision tree rules were used as rules of decision-making. Wang et al. [4] came up with an innovative way using the quantum k-nearest neighbor (kNN) algorithm. This approach employs the Grover algorithm as a search engine on the feature space to identify k-nearest neighbors and can contribute to improved accuracy for specific data types. Sheikh and Patel [5] studied the utilization of principal component analysis (PCA) and linear discriminant analysis (LDA). Their research also revealed that PCA is more successful than LDA for small datasets, meaning it is necessary to take data size into account when deciding on a technique for dimensionality reduction. For this reason, Monica and Lavanya in [6] recommended using CC (consensus clustering), which allows for obtaining clustering predictions from multiple algorithms and possibly optimizing clustering efficiency and accuracy in this way.

A CNN model designed by Assiri [7] is a simple CNN model that incorporates many methods. The dataset and the number of CNN layers were different; he used four layers for MNIST, but the only weakness was the lack of residual blocks. He applied several methods including max-pooling, dropout, data augmentation, and early stopping. Using various hyperparameter settings, the highest accuracy was obtained at 99.83% and error rate at 0.17%. EnsNet [8] is a new architecture proposed by Hirata and Fujiyoshi consisting of a base CNN along with multiple fully connected subnetworks (FCSNs). In this model, feature maps that are generated by the final convolutional layer of a base CNN are divided into different groups. Each FCSN is fed with one such group. The EnsNet is simple but attains state-of-the-art results on MNIST, achieving an error rate of 0.16% (corresponding to an accuracy of 99.84%). An et al. introduced in their study [9] simple and highly effective CNN models for the recognition of digits in the MNIST dataset. Every model has multiple convolutional layers where batch normalization and ReLU activation functions were employed in each layer while pooling layers were not used. The application of data augmentation was aimed at promoting training efficiency. Moreover, the epoch size was increased from 50 to 150, hence the training was improved. These independently trained models performed on test accuracy at levels of 99.79–99.82%. The authors investigated different ensemble learning strategies such as combining models in different orders and achieved good results. Byerly et al. proposed a novel approach for CNNs using homogeneous vector capsules (HVCs) in place of fully connected layers [10]. Unlike traditional CNNs that make use of matrix multiplication, HVCs apply element-wise multiplication that maintains the matrix dimensionality. With this approach, the subsequent CNN architecture has become much simpler and has fewer parameters. Computation cost is also reduced, along with training speed, compared to any previous state-of-the-art capsule network. They used the data augmentation technique in their experiments, similar to other studies. However, they did not consider it generic but very specific to the domain. Authors tested models individually first; after that, ensemble methods were investigated, and finally, branch weights were studied, for which weight initialization was different. All these experiments showed excellent test accuracies, but the highest one turned out to be 99.87%. Ahmed et al. [11] proposed a method that introduced a DL-based technique, namely, EfficientDet-D4, for numeral categorization. Initially, the input images are annotated to exactly show the region of interest (ROI). In the next phase, these images are used to train the EfficientNet-B4-based EfficientDet-D4 model to detect and categorize the numerals into their respective classes from zero to nine. They tested the proposed model over the MNIST dataset to demonstrate its efficacy and attained an average accuracy value of 99.83%.

Although historically CNNs have led the field of CV, recently there has been a rise of transformer-based models. Dosovitskiy et al. showed [1] that ViTs can achieve similar performance to the most advanced CNNs but with a lower demand for training resources. ViTs split images into patches, vectorize them, and then process them through a transformer encoder. In this work, they demonstrated the viability of ViTs as an effective tool for image classification tasks, specifically on the MNIST dataset. This article explores a novel approach to HDR by leveraging the complementary strengths of CNNs and ViTs. ViTs have a high capability to deal with larger datasets and offer much more reliable performance and robustness [12–14], yet depending on large datasets can also be their limitation. Meanwhile, CNNs are known to have strength in extracting features out of smaller datasets as well as display scale and shift-invariance (e.g., [15]), which makes them suitable for tasks with limited data. To surpass the level of performance and robustness in HDR, the suggested approach looks for

small or challenging datasets for ViTs only. The result of this combination leverages the feature extraction abilities of CNNs with potentially an even higher level of performance and robustness, which is characteristic of ViTs, especially when working with a larger dataset. This proposal tested on two types of images: cleaned and denoised digits, and images with their original appearance. Such a comprehensive analysis will provide valuable insights into the model's ability to handle variations in image quality, which is essential for real-world applications of HDR.

CrossViT, introduced by Chen et al. [16], is a model that contains the cross-attention mechanism to exchange information within the image at different scales and enhance feature extraction for better classification accuracy. It has not been explicitly evaluated against HDR datasets, but its capability of multi-scale information manipulation might help recognize hand-written digits of different sizes. In the domain of remote sensing imagery scene classification, Wang et al. [17] proposed a new model that explores the combination of ViTs with graph convolutional networks (GCNs). This study focuses on the use of ViTs in combination with other architectures to leverage their complementary strengths. In the context of HDR, this idea may be considered, where such a model would integrate explicit knowledge from a particular field. ViT for the classification of breast ultrasound images presented by Gheflati and Hassan [18] is another example showing the application of ViTs in non-natural image classification tasks. This article demonstrates the effectiveness of using ViTs for breast ultrasound image classification. Although the goal may be different from HDR, it is indicative of the versatility of ViTs for solving various image classification tasks, and therefore, ViTs could potentially be used more extensively. An alternative model was also presented by Wu et al. called CvT [19], which combines the advantages of CNNs and ViTs into a single model. CvT uses convolutional layers for efficient local feature extraction and transformer blocks to work with global information processing. Such an architecture could help achieve better performance in HDR tasks where both local and global cues are important for recognizing digits accurately. Additionally, Jayant and Vanita proposed an application of a convolutional vision transformer (CVT) for HDR [20]. To evaluate the proposed model, two datasets are considered: clean images from the EMNIST and original images from the Historical Handwritten Digit Dataset (DIDA). They achieved an accuracy of 99.89% on the EMNIST dataset using a Stochastic Gradient Descent (SGD) optimizer with a learning rate of 0.01, a batch size of 192, and a dropout rate of 0.25. However, one limitation of this study is the exclusive evaluation of single-digit datasets. This related work highlights the diverse approaches and continuous advancements in MNIST digit recognition. However, the methods for CNNs are still popular. There is a promise of new algorithms like ViTs that can improve their characteristics in accuracy and efficiency.

The researchers [21] proposed a hybrid convolutional vision transformer (CViT) architecture designed to leverage the strengths of both ViTs and CNNs. In their study uses two datasets for evaluation: the EMNIST-digit dataset, which contains cleaned images, and the DIDA dataset, which includes uncleaned images with noise and distortions. The model's performance is enhanced through cross-validation and hyperparameter tuning, demonstrating robustness and effectiveness on both cleaned and uncleaned images. CViT includes the construction of the ViT with embedded patches, layer normalization, self-attention mechanism, and the use of the Gaussian error linear unit (GeLU) activation function in the MLP layer. The model achieves a high recognition accuracy of 99.89% on the EMNIST-digit dataset and 99.81% on the DIDA dataset. The authors suggest that the model's robustness to noise and varying writing styles make it suitable for real-time applications such as postal letters and historical document digitization.

A summary of the related works explored for the digit recognition task, along with an overview of their performance and characteristics, is presented in Table 1. Even though SVMs and decision trees could still perform quite well, they may not be completely dismissed due to their respectable accuracies. However, CNNs are most often used, with EnsNet achieving the best outcomes nowadays (with only a 0.16% error rate). Capsule networks and ViTs have, however, outperformed all other methodologies by surpassing the 99.8% level. The emergence of convolutional vision transformers (CVTs) could be a good solution by merging CNN and ViT capabilities and providing better results along with robustness, mainly for small datasets. The present research establishes that this process is iterative in advancing the MNIST and EMNIST digit recognition, which would help to develop more accurate and less resource-demanding models.

Table 1: A summary table of models-related work

Method	Year	Dataset	Accuracy rate	Description	Limitations
SVM [2]	2022	MNIST	97.83%	Real-time system with training and recognition stages	Lower accuracy compared to DL approaches
Decision tree [3]	2019	MNIST	83.4%	Rule-based classification	May not be as effective for complex datasets
Quantum kNN [4]	2019	MNIST	—	k-nearest neighbor with Grover algorithm for searching neighbors	Requires further investigation on its generalizability to diverse HDR tasks
PCA/LDA [5]	2019	MNIST	86.6%	Dimensionality reduction techniques	Primarily targeted for dimensionality reduction, not specifically optimized for classification
Consensus Clustering [6]	2019	MNIST	95%	Combining multiple clustering predictions	Effectiveness might depend on the choice of individual clustering algorithms used
Simple CNN [7]	2019	MNIST	99.83%	Various techniques (max-pooling, dropout, etc.)	Lacks residual blocks, which can potentially improve performance and reduce training time
EnsNet [8]	2021	MNIST	99.84%	Base CNN with multiple Fully Connected Subnetworks	Limited exploration of hyperparameter settings and variations in the architecture
Ensemble CNNs [9]	2018	MNIST	99.79% −99.82%	Multiple CNNs with batch normalization and ReLU	Does not utilize pooling layers, which might affect feature extraction capabilities
Capsule Networks [10]	2020	MNIST	99.87%	HVCs	Limited evaluation on datasets beyond MNIST, potentially restricting generalizability
EfficientDet-D4 [11]	2023	MNIST	99.83%	The model was able to generalize well to unseen cases, indicating its effectiveness in real-world scenarios	It can be computationally expensive, especially for large datasets or real-time applications
CrossViT [16]	2021	ImageNet 1K	—	Multi-scale transformer encoders with cross-attention	Not directly evaluated on HDR datasets
ViT+GCN [22]	2020	AID	93.31%	Remote sensing scene classification	Can require large datasets for optimal performance
ViT [18]	2021	BUSI and B	86%	Breast ultrasound image classification	Can require large datasets for optimal performance
CvT [19]	2021	ImageNet	90.6%	Merging CNN and ViT strengths	Can require large datasets for optimal performance
CVT [20]	2022	EMNIST	99.89%	Combines CNN and ViT strengths	Limited to single-digit datasets, necessitating evaluation on multi-digit scenarios
CViT [21]	2024	EMNIST	99.89%	leverage the strengths of both ViTs and CNNs, CViT demonstrated robustness to variations in writing	The model's performance might vary depending on the specific characteristics of the dataset
		DIDA	99.81%	style and image artifacts	used for training

3 Materials and methods

In this section, we describe the ML and DL models used in our experiments. The ML experience is a typical process that develops from data collection and preprocessing to make it suitable for training and testing, to the implementation of algorithms itself. First, specific features are extracted from the dataset, which helps in the training and testing of models. Second, these extracted features are employed in making predictions to conform with the research question. The result will be determined by classification as it is what our study centers on, identifying English digits handwritten later. AI is a concept that denotes the capacity of machines to understand and act based on any prior training using a particular dataset that is given to them; these devices also have algorithms, which include LR, decision trees, support vector machines, k-nearest neighbors, random forests (RFs), etc. DL is a technique where artificial neural networks (ANNs) such as CNNs and recurrent neural networks (RNNs) play an important role in modeling intelligence. The dataset used in all the models was divided into training and testing datasets with a ratio of 80–20%. The MNIST dataset has 10 different classes for numerical digits from 0 to 9, while the EMNIST dataset includes 47 classes, encompassing both digits and letters of the English alphabet. For our experiments with the EMNIST dataset, we used only the digit classes (0–9) for consistency with the MNIST dataset. Different experimental strategies were adopted, as discussed later on. The details of these strategies are shown as follows:

(1) Standalone models:
   – LR algorithm uses L1 and L2 regularization.
   – Multi-layer perceptron (MLP) algorithm integrates the initialization method to facilitate learning and uses an optimizer to adjust the network's hyperparameters to improve performance.
   – CNN algorithm employs an optimizer to adjust the network's weights or hyperparameters to enhance results.
   – ViT model.
(2) Hybrid models:
   – CNN and RF.
   – CNN and LS.
   – CNN and VotingClassifier (VC).
   – CNN and ViT.

This section presents the methodology and implementation details of all models used for English HDR. We performed each model with the same hyper-parameters for fair comparison. In this article, we have implemented LR, MLPs, CNN, ViT, and hybrid models, and compared them to demonstrate the impact of feature extraction before the classifier stage.

3.1 Standalone models

3.1.1 LR

LR is a statistically supervised ML model that models the connection between input parameters and an output variable using a logistic function. The model output in binary LR is predicated on values of either 0 or 1. This technique is used to forecast the likelihood that an event will occur during a specific test. The LR problem involves employing a linear regression function to make predictions. However, the linear regression equation is inappropriate for LR, as the model's responses may not conform to the 0 and 1 requirements. This problem needs to be solved using a transformation, which is commonly known as a logit transformation. The transformation appears as follows:

$$P = \frac{1}{1 + e^{-y}}, \tag{1}$$

where P represents the probability of the event happening, e stands for the Euler number, and y denotes the standard regression equation.

We employed LR for the HDR task. In the context of image classification, the model converts each pixel into logarithmic odds (logits) as described previously. Since the classifier maps pixels directly to log odds, there are 784 inputs and 10 outputs. To address overfitting and underfitting issues, we applied regularization techniques, specifically L1 and L2. This technique aims to tune the model to reduce its error rate. Two common regularization methods are L1 and L2:

- L1 - LASSO regression which stands for Least Absolute Shrinkage and Selection Operator. It adds the absolute magnitude weighted by λ as a penalty to the loss function.
- L2 - Ridge regression uses the squared magnitude instead of absolute magnitude. Thus, it is non-sparse. We will use both in our experiments to see which one is best for our case.

3.1.2 MLP

MLP is a variation of a feedforward network that includes interconnected layers such as an input layer, one or more hidden layers, and an output layer. Each layer contains many nodes called neurons [23]. These neurons take inputs, calculate a weighted sum of these inputs along with a threshold, and then apply an activation function. The activation function used to calculate the sum of input weights and biases determines whether a neuron can be activated. When a node is activated, data are transferred to a subsequent network's layer. Otherwise, if the activation threshold is not reached, no data will be passed to the next layer. The activation function can be linear or nonlinear, depending on the specific function, often called the transfer function [24].

The second model is the MLP, a fully connected feedforward ANN. The number of nodes in the input layer is 784 (28×28), corresponding to the number of pixels in the input image. The number of nodes in the output layer represents the number of classes. During the experiments, we tested two MLP structures. The first structure consists of two hidden layers, both with 400 nodes. We trained the model by specifying initial values for the weights. Weights could be initialized in multiple ways, and using weight initialization methods instead of the default method could aid in the convergence process. We selected the following initialization methods to facilitate learning. The methods tested in the experiments are as follows:

- Zero initialization: all weights are set to zero.
- Uniform initialization: weights are randomly set to a value from the uniform distribution within a specified range. We will use the range [0, 1].
- Standard normal initialization or Nave initialization: Weights are randomly set to a value from the normal Gaussian distribution.
- Truncated normal initialization or Xavier normal initialization: this method uses the tanh activation function to set the weights.

Changing or modifying network structure is a technique used by researchers to obtain better results. In Assiri's [25] work, a different structure was used for each dataset. Modifications may include increasing network depth by adding more layers or reducing the number of layers to reduce model complexity. Another approach is to adjust the number of nodes (neurons) in each layer, thus affecting the computational load of that layer. In addition, introducing pooling layers or implementing dropout techniques on specific neurons can further change the network structure. As mentioned above, we made changes to the structure of the MLP model. The second structure has higher depth and four hidden layers but is less computationally intensive, i.e., a smaller number of nodes. The number of nodes in each layer is 100, 400, 100, and 100, respectively. We optimized this model using an optimization method, as a well-chosen optimizer can be instrumental in adjusting the network's weights or hyperparameters to enhance results. The optimizers tested in the experiments are as follows:

- SGD: it updates the weights more frequently than Gradient Descent. Since the weights are updated at each epoch, their values have high variance, making them more likely to converge faster.

– RMSProp: this optimizer restricts oscillations to be vertical, allowing the learning rate to be set to a high value.
– Adagrad: in contrast to most optimizers, Adagrad changes the learning rate, enabling the model to explore a larger area in the search space.
– Adam: short for Adaptive Moment Estimation, Adam substitutes SGD and combines the best properties of the previous two methods, RMSProp and Adagrad.

3.1.3 CNNs

ANN represent an intelligent technique used for data processing and recognition. According to O'Shea and Nash [26], ANNs are primarily composed of a network of computational nodes that operate together in a distributed manner to process complicated data inputs and optimize the final output. In terms of composition, CNNs and traditional ANNs are similar. They are made up of self-optimizing neurons, each neuron receives input, processes it to produce a scalar product, and then applies a non-linear function. CNNs are regularized forms of fully connected networks, also referred to as MLPs. Every neuron in one layer of these networks is connected to every other layer's neuron. The last layer typically includes loss functions associated with classes, resulting in the entire network expressing a single perceptive score function (the weight).

CNNs consist of three distinct layers: the convolutional layer, the pooling layer, and the fully connected layer. Each of these layers processes input data in a certain way. The convolutional layer uses filters to extract features, while the pooling layer conducts either max pooling or average pooling, extracting the maximum or average value inside the filter region, respectively. The fully connected layer aggregates information from feature maps to produce the ultimate classification. CNNs have proved to be the best model when dealing with images. The structure used is two convolutional layers. Both convolutional layers' kernel sizes are 3 × 3 and have a max pooling. The input and output layers are as specified in the MLP section. The model consists of two convolutional layers each layer uses batch normalization and ReLU as an activation function (The rectified linear activation function or ReLU for short is a piecewise linear function that will output the input directly if it is positive, otherwise, it will output zero.). The last layer of the model is fully connected. The activation function for the output layer is Softmax which turns numbers aka logits into probabilities that sum to one. Softmax function outputs a vector representing the probability distributions of potential outcomes. The epoch size is 40. Figure 1 depicts the entire CNN procedure in detail.

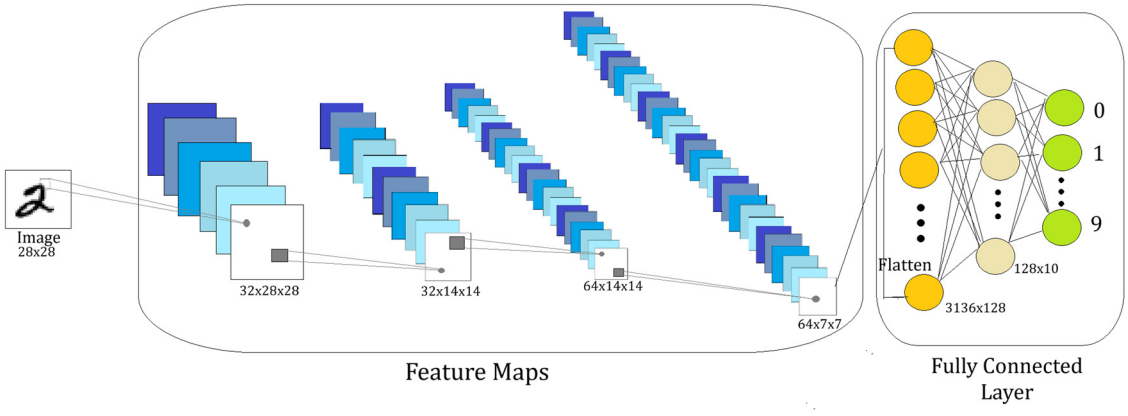

Figure 1: The architecture of the CNN model for HDR. The network consists of two convolutional layers with 3 x 3 filters and max-pooling layers. The final fully connected layer with 128 neurons is used for classification. Source: Created by the author.

3.1.4 ViT model

In model design, we closely adhered to the original ViT architecture proposed by Agrawal and Jayant [1]. A notable advantage of this intentionally simple setup is that ViT architecture, along with its pipeline implementations, can effectively remove the need for many hand-designed components like a non-maximum suppression procedure or anchor generation commonly employed in classic CNN architectures.

3.1.4.1 ViT architecture

The transformer receives embedded patches as its input. To create these embeddings, the original image is split into fixed-size patches. Padding is applied if the image dimensions are not perfectly divisible by the patch size. Afterward, the patches are flattened into one-dimensional (1D) vectors, as transformers operate on sequential input. Positional embeddings are added to preserve spatial information within the sequence. The transformer encoder consists of alternating multi-headed attention and feed-forward network (MLP) layers. Layer normalization is applied independently within each layer for stability.

A fundamental part of multi-head attention is self-attention, which receives query (Q), key (K), and value (V) vectors as input. It calculates a weight for each value vector using the dot product of Q and K, followed by the softmax function as shown in equation (2) [27]. These weights represent the relevance of each value to the current query. Finally, the self-attention mechanism produces an output by summing the weighted value vectors. This output effectively captures the relevant information from the entire sequence based on its relationship to the query.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^{\top}}{\sqrt{d}}\right)V, \tag{2}$$

where d is the hidden dimensions. The multi-head attention layer leverages its architecture to enable parallel processing and improve efficiency. It achieves this by splitting the input into smaller parts, allowing individual attention computations to occur parallelly. Following this, the MLP layer, a two-layer feed-forward network, introduces non-linearity. This layer utilizes the GeLU activation function, defined in equation (3) [28].

$$\text{GeLU}(x) = x\Phi(x) \approx 0.5x\left(1 + \tanh\left[\frac{\sqrt{2}}{\pi}(x + 0.044715x^3)\right]\right), \tag{3}$$

where $\Phi(x)$ is the Gaussian cumulative distribution.

3.1.4.2 Model architecture

To ensure uniformity during model processing, images in the dataset are first resized to a consistent size. This pre-processing step potentially enhances the model's performance by creating a standardized input format that simplifies model training and inference. Then, modular components important for constructing the ViT model are defined. These include the PreNorm module, which applies layer normalization before a function for stability, and the Residual module, which implements residual connections. Attention is a multi-headed attention module, Feedforward is a two-layer feed-forward network with GELU activation and Transformer for stacked layers of attention and feed-forward modules. Transformer encoding and image processing are the two main phases in building the ViT. In image processing, the input image is divided into patches, flattened, and projected into an embedding space, with a class token appended for global comprehension and positional encodings incorporated for spatial awareness. The embedded patches are passed through the transformer module for feature extraction in transformer encoding. The classification head then extracts the output of the class token, and the MLP head is then applied for the final classification. The training procedure entails data preparation, where an appropriate image dataset is loaded, preprocessed, and split into training, validation, and testing sets. We chose Adam as an optimizer to update model parameters and cross-entropy loss as a suitable loss function. We modified several transformer parameters to optimize the model for handwritten digit datasets. Specifically, modifying the depth parameter played a crucial role in enhancing model

performance and accuracy. Additionally, we employed a transformer module to calculate embedding patches and extract relevant features from the images.

3.2 Hybrid model

In CV, ensemble methods combine predictions from various models, they are a well-established approach to improve image classification accuracy. This article explores the application of these techniques to HDR specifically, focusing on three configurations: (1) CNN and RF, (2) CNN and VC, and (3) CNN and LR. In addition, a hybrid model combines CNN and ViT for a more robust and accurate approach.

CNN and RF: this ensemble technique uses the complementary advantages of decision trees and DL. CNNs are adept at capturing complex image features, whereas RF provides flexibility and interpretability. This combination aims to improve recognition performance by merging feature extraction capabilities with interpretability. Predictions from both individual models are aggregated to determine the final class label. This ensemble model first trains a CNN model on the MNIST dataset and then trains an RF classifier using the same MNIST dataset (respectively for the EMNIST dataset). It then makes predictions using both models and combines them using simple averaging. Finally, it evaluates the ensemble model's accuracy.

CNN and LR: this ensemble combines a DL with a traditional statistical classifier. CNNs extract high-level features, while LR performs efficient linear classification. By integrating these models, the ensemble aims to leverage the feature representation power of CNNs with the interpretability and simplicity of LR. Similar to the previous ensemble, individual model predictions are aggregated through a voting mechanism to determine the final class label.

CNN and VC: in contrast to the first two configurations, this ensemble approach involves combining predictions from two CNN models using the VC from Scikit-learn. The first CNN is described in Section 3.1.3, and the second one uses three convolutional layers with 3×3 kernel sizes, and three max pooling layers used for downsampling, ReLU activation functions, a dropout rate of 0.2, a learning rate of 0.01, 40 epochs, a batch size of 64, and a Softmax activation function for the output layer. VC acts as a meta-classifier, aggregating the predictions from the individual models into a final prediction. Each CNN model independently makes predictions for the same input data. Each CNN model independently makes predictions. Then, the predictions from both models are combined using a voting mechanism. The most common voting mechanism is hard voting, where the class with the majority of votes is chosen as the final prediction. Finally, the class with the highest number of votes (or weighted average of probabilities) is selected as the final output of the ensemble. The accuracy of the ensemble model is then evaluated on the test data.

CNN and ViT: this contribution uses a combination of CNN and ViT to recognize handwritten digits. ViTs, a relatively new architecture inspired by NLP, have demonstrated promising results in image classification tasks. Combining ViTs and CNNs can provide insights into their strengths and weaknesses. The dataset images are initially resized so that all the images are of the same size when sent to the model. It increases the model performance. Then, the images are normalized using mean and standard deviation. Normalization is also known as rescaling and helps apply the same algorithm to all images. Scaling all images contributes to total loss and provides a standard learning rate. The CNN portion extracts local features from the input images using convolutional and pooling layers. The CNN uses three convolutional layers with 3×3 kernel sizes. There are also three max pooling layers used for downsampling, which indirectly affect the depth by reducing the feature map size. However, ViT component captures global dependencies and contextual information within the images. The number of layers within the ViT (one Dense layer with 128 units) can be considered for a broader understanding of model complexity. Then, the outputs of the CNN and ViT components are fused using a fully connected layer to produce the final classification. This architecture has a moderate depth due to the use of three convolutional layers in the CNN. While the ViT does not directly contribute to depth in the same way, its presence adds complexity to the model.

4 Experiments

4.1 MNIST dataset

We have MNIST dataset to train and test our approach. MNIST is the abbreviation of The Modified National Institute of Standards and Technology, it was developed by Lecun et al. [29]. It has a large collection of handwritten digits, with 60,000 training images and 10,000 testing images. Each digit is size-normalized, gray-scaled, and fixed-size 28 × 28 images as provided in Figure 2. MNIST has been used in papers since it became available to researchers and developers in 2010. It is easy to use, has simple images, and does not need further processing before use. The MNIST dataset has been used in over 4,000 papers in the past 4 years. For 2023, the number is increasing, and the number of papers published is over 50 in January.

Figure 2: Samples of MNIST dataset [29].

4.2 EMNIST dataset

The EMNIST (Extended MNIST) [30] dataset is a collection of handwritten character recognition datasets derived from the NIST Special Database 19 and converted to a 28 × 28 pixel image format and dataset structure that directly matches the MNIST dataset. It is designed to provide a more complex and diverse set of challenges for ML models by extending the MNIST dataset, which originally consisted of digits (0–9), to include additional handwritten characters. The EMNIST dataset includes: (i) Handwritten digits (from 0 to 9), similar to MNIST. (ii) Uppercase and lowercase English letters, making it more challenging by introducing a wider range of characters. It includes multiple splits such as:

- EMNIST ByClass: contains 8,14,255 characters from 814 classes, with digits and letters.
- EMNIST ByMerge: a merged version of ByClass with fewer categories, but still more complex than the original MNIST.
- EMNIST Letters: focused on uppercase English letters, adding complexity beyond simple digit recognition.

EMNIST is often used to evaluate and benchmark ML and DL models on tasks that involve both letter and digit classification, providing a more diverse and difficult test set for recognition systems.

4.3 Experimental setup

We executed the models on a hardware platform, 12th Gen Intel(R) Core(TM) i7-1250U 1.10 GHz, equipped with 16 GB of RAM. A comparative analysis was conducted among models (LR, MLP, CNN, ViT, and hybrid models), depending on the accuracy and runtime of each method. The aim was to identify the model that achieves the highest accuracy through the training and testing of the dataset. All experiments were conducted using Google Colaboratory, which is a tool developed by Google Research. Google Colab enables users to write and execute Python code, utilizing either a CPU or a GPU. As for libraries, the Keras library, Scikit-Learn, Matplotlib, Pandas, and Numpy packages were employed to create the deep NNs in the Python programming language. The parameters "batch-size" and "epochs" provided to the "compile" function in NNs determine the duration of the training process. The "epochs" parameter signifies the number of training rounds, indicating how many times the data will be presented to the network. We fixed 20 for the number of epochs for LR, MLP, and hybrid models, 40 for the CNN model, and the ViT model which was set at 100 epochs. On the other hand, the "batch-size" parameter represents the amount of data received in each epoch. Adjusting these values provides the opportunity to enhance test results and improve the accuracy rate.

4.3.1 ViT model parameters

The most important task is finding the best hyperparameter value to fit the model. The ViT model was trained using the subsequent procedures: SGD was chosen as the optimizer with a learning rate set to 0.1 and a momentum value of 0.9. The loss function used was the Negative Log-Likelihood (NLL) function. The data loader implicitly determined the batch size, even though it was not stated in the code. The training dataset was iterated over for a predetermined number of epochs, which in this case was set to 100. During each epoch, the data loader facilitates the data processing occurring in batches. A batch of data was fed through the model to generate predictions, and then, the NLL loss between the model predictions and the actual labels was calculated. The loss function was then backpropagated throughout the model to adjust the weights, which were updated using the optimizer along with the computed gradients. Training progress was regularly monitored, and training loss was logged at regular basis, and updated as needed. The major hyperparameters that defined the particular ViT configuration used in this work for HDR were as follows:

- Image size: 28 × 28 pixels
- Patch size: 7 × 7 pixels
- Number of classes: 10 (corresponding to the MNIST dataset)
- Channels: 1 (grayscale images)
- Embedding dimension (dim), also referred to as the hidden size (D), determines the size of the feature representations within the ViT model and was set to 64 in this study.
- Depth, is the number of encoder layers. The ViT model consists of a transformer module, with the depth parameter specifying the number of transformer layers. In this implementation, the depth was set to 6, indicating the presence of six transformer layers.
- Number of heads is the attention heads per layer. The attention mechanism in the ViT model employs multiple attention heads per layer, with the number of heads controlled by the heads parameter. In this study, the number of attention heads was set to 8.
- MLP dimension (mlp_dim): is the hidden dimension in feed-forward layers. The classification head of the ViT model includes an MLP with a hidden dimension defined by the mlp_dim parameter. In this implementation, the MLP size was set to 128.
- Params are the total number of parameters in the ViT model determined by inspecting the instantiated model object. The total number of parameters in the model was found to be 213,642.

Open-source dataset was used to assess the proposed models. There are a lot of performance metrics for a classifier to show how well it performs in statistical situations. Our approaches computed evaluation metrics,

including accuracy, precision, recall, *F*1-measure, Cohen's kappa score, ROC AUC score, and the confusion matrix to evaluate the proposed models' reliability and performance.

4.3.2 Evaluation metrics

Accuracy, defined by equation (4), introduces the proportion of correct predictions to the total predictions made. The evaluation metrics are defined as TPs (TPs), true negatives (TNs), FPs (FNs), and false positives (FPs).

$$\text{Accuracy} = \frac{\text{TN} + \text{TP}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \tag{4}$$

Precision is the ratio of TPs relative to the sum of TPs and FPs. Equation (5) shows the precision formula.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{5}$$

Recall represents the ratio of correctly identified positive instances among the total number of positive instances. The evaluation metric for recall is shown in equation (6).

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{6}$$

*F*1-measure serves as a significant measure in ML, combining the predictive performances of two other metrics. Equation (7) outlines the evaluation measure for the *F*1-measure.

$$F\,1\text{-measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{7}$$

Cohen's kappa score is a statistical metric that measures the agreement between two or more raters (or observers) who are classifying items into categorical categories; in other words, it compares an observed accuracy with an expected accuracy (say random chance). Equation (8) presents the kappa score.

$$\text{Cohen's kappa} = \frac{\text{observed\_accuracy} - \text{expected\_accuracy}}{1 - \text{expected\_accuracy}}. \tag{8}$$

The confusion matrix is a table that shows the correct and incorrect predictions of our model compared to the true labels. It helps us see when the model gets confused and mixes up different classes. We used the confusion matrix to identify which specific classes of handwritten digits were most accurately detected by the model, as well as to pinpoint which classes were often mistaken for others.

The ROC AUC score is the area under the ROC curve. It sums up how well a model can produce relative scores to discriminate between positive or negative instances across all classification thresholds. The ROC AUC score ranges from 0 to 1, where 0.5 indicates random guessing and 1 indicates perfect performance.

4.4 Results

4.4.1 Results of LR model

We started our experiments with an LR model, and we implemented it individually and with regularizers L1 and L2 as mentioned before. Table 2 illustrates the results applied to the MNIST dataset.

Regularizations are typically implemented into the model to improve accuracy and average loss, as Table 2 illustrates. Compared to L1, L2 regularization yielded somewhat better results for both measures. Therefore, the use of L2 regularization is deemed more suitable in our case. Then we conducted experiments with a LR model, and we implemented it individually and with regularizers L1 and L2 as mentioned before. Table 3 illustrates the results applied to the EMNIST dataset.

Table 2: LR results for the last 20 epochs on MNIST dataset

Train and Test	Measures	Original	L1	L2
Train	Accuracy	92.95%	92.98%	92.99%
	Precision	92.75%	92.98%	92.99%
	Recall	92.95%	92.99%	93%
	*F*1-score	92.95%	92.99%	93%
	Loss	0.2561	0.2555	0.2550
Test	Accuracy	92.48%	92.49%	92.50%
	Loss	0.2681	0.2680	0.2679

Table 3: LR results for the last 20 epochs on EMNIST dataset

Train and Test	Measures	Original	L1	L2
Train	Accuracy	86.63%	87.18%	87.20%
	Precision	84.77%	84.97%	84.98%
	Recall	85.88%	86.09%	86.10%
	*F*1-score	84.18%	84.86%	84.88%
	Kappa	85.37%	85.77%	85.77%
	Loss	0.3595	0.3554	0.3552
Test	Accuracy	86.23%	86.54%	86.56%
	Loss	0.3621	0.3617	0.3616

4.4.2 Results of MLP model

Then, we implemented MLP, with two different structures as specified in the methodology section. Afterward, we tested four initialization methods on the first structure, and Xiaver had the best results. Also, four optimization methods were tested in the first structure, and Adagrad provided the best results. We experimented with combining the optimal structure with the optimal initializer and optimizer, and the combined model produced the best outcomes. Next, we applied the MLP, employing two distinct structures as outlined in the methodology section on the MNIST dataset. Then, we tested the first structure using four different initialization techniques: Xavier normal, Standard normal, Uniform, and Zero initialization with Xiaver initialization yielding the most favorable results. Similarly, four optimization methods (Adam, Adagrad, SGD, and RMSprop) were tested on the first structure, and Adagrad yielded the best results. The resulting model generated the best results by combining the optimal structure with the best-performing optimizer and initializer. The outcomes are visually represented in Table 4 applied on the MNIST dataset.

4.4.3 Results of CNN model

The third model implemented is CNN. This study determined the hyperparameters for the designed CNN model as indicated in Table 5.

Figure 3 shows the confusion matrix generated using the CNN model's architecture, which was trained on the MNIST dataset and tested on the dataset's samples.

Figure 4 shows the confusion matrix generated using the same CNN model's architecture, which was trained on the EMNIST dataset and tested on the dataset's samples.

Table 6 shows the training and testing results of the CNN model on both the MNIST and EMNIST datasets. The model achieved exceptional performance on both datasets during training, with near-perfect accuracy, precision, recall, *F*1-score, and ROC AUC scores. On the test set, the model maintained high accuracy,

Table 4: MLP results for the last 20 epochs on the MNIST dataset

Train & test	Measures	MLP									
		First structure	Second structure	Initializers				Optimizers				Combined model
				Zero	Uniform [0,1]	Standard Normal	Xavier Normal	SGD	RMSProp	Adagrad	Adam	
Train	Accuracy	98.75%	98.64%	30.68%	11.24%	74.49%	98.90%	98.75%	88.15%	99.50%	96.83%	99.14%
	Precision	98.82%	98.88%	31%	12%	75%	98.94%	98.8%	88.9%	98.14%	97%	98.35%
	Recall	98.82%	98.88%	31%	12%	75%	98.94%	98.8%	88.9%	98.14%	97%	98.35%
	*F*1-score	98.82%	98.88%	31.1%	12.4%	75.1%	98.9%	98.8%	88.9%	98.14%	97%	98.35%
	Loss	0.0146	0.0136	1.6577	2.3013	0.8213	0.0094	0.0146	0.4544	0.0015	0.1299	0.0011
Test	Accuracy	98.36%	97.47%	31.15%	11.35%	74.12%	98.02%	98.10%	88.00%	98.33%	95.98%	98.10%
	Loss	0.0491	0.0816	1.6639	2.3010	1.1867	0.0565	0.0592	0.5554	0.0499	0.1984	0.0592

Table 5: Parameters applied to the CNN models in this work

Parameters	Values
Batch size	64
Epochs	40
Dropout rate	0.2
Learning rate	0.01
Activation function	ReLu
Activation function (output layer)	Softmax
Metric	Accuracy, precision, recall, *F*1-score, Confusion matrix, ROC AUC score, and kappa score

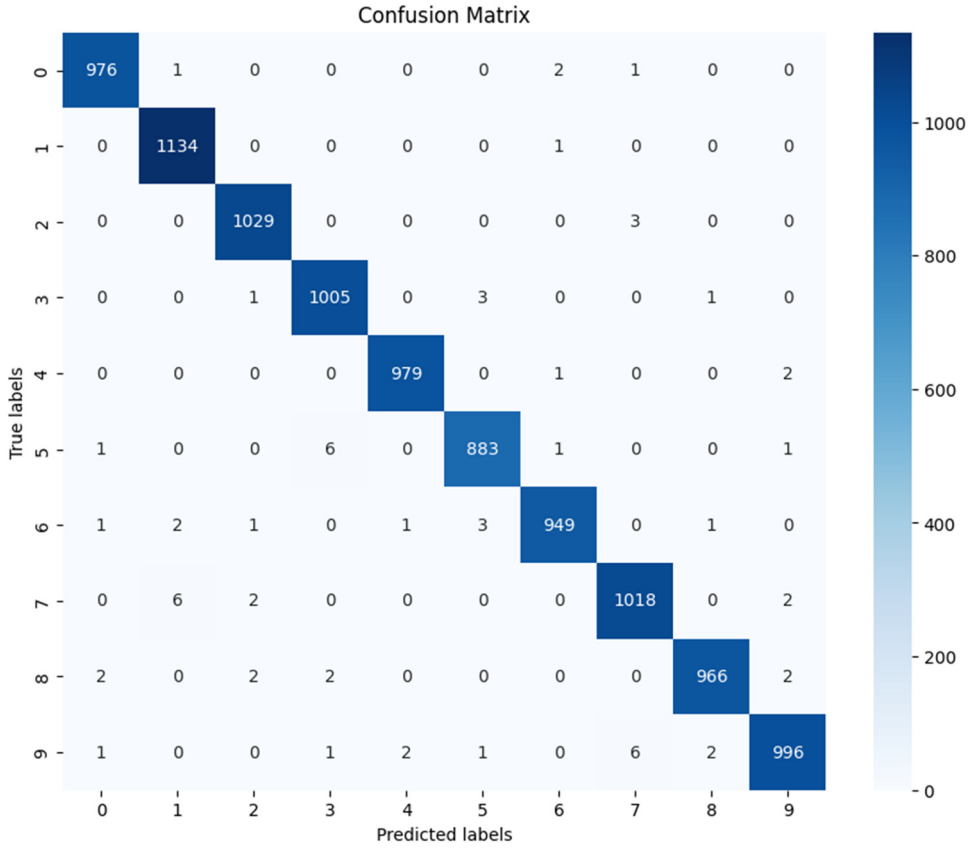

Figure 3: Confusion matrix results using CNN model on MNIST-digit dataset. Source: Created by the author.

demonstrating good generalization ability. Notably, the model achieved slightly higher accuracy and lower loss on the EMNIST dataset compared to MNIST, suggesting that the CNN model effectively learned the underlying patterns in the more challenging EMNIST dataset.

Figure 5 shows the loss and accuracy curve graphs for the CNN architecture. As the training process progresses, the curves demonstrate a clear decline in loss and a simultaneous rise in accuracy. This trend suggests that the model is effectively learning the underlying patterns and relationships within the data, allowing it to make better predictions. Notably, the graph exhibits minimal variation after the tenth iteration. This observation signifies the point of convergence, where further iterations are unlikely to lead to significant improvements in the model's performance. This can be attributed to the model having reached a state of optimal learning, where it has sufficiently adapted its internal parameters to capture the essential features of

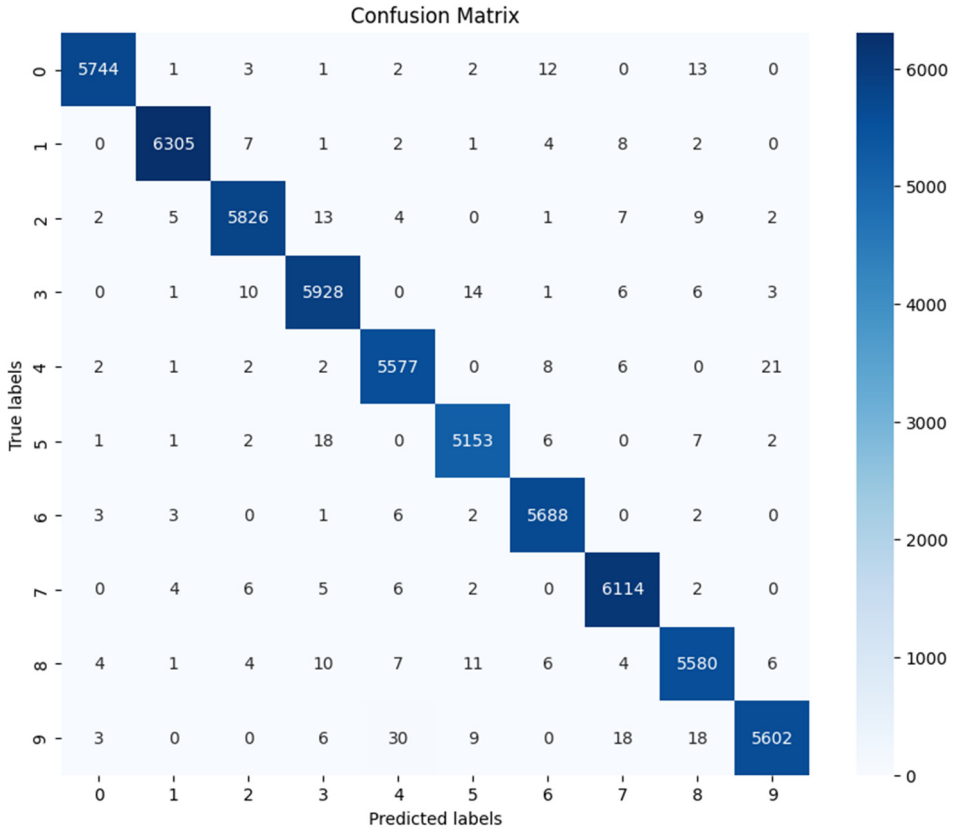

Figure 4: Confusion matrix results using CNN model on EMNIST-digit dataset. Source: Created by the author.

Table 6: CNN results for the last 40 epochs on MNIST and EMNIST dataset

Train and Test	Measures	MNSIT	EMNIST
Train	Accuracy	99.90%	99.92%
	Precision	99.31%	99.34%
	Recall	99.31%	99.34%
	*F*1-score	99.31%	99.34%
	Kappa	99.23%	99.27%
	ROC AUC	99.99%	99.98%
	Loss	1.1067×10^{-7}	0.0031
Test	Accuracy	99.23%	99.31%
	Loss	0.0761	0.0660

the data. While additional training might lead to slight fluctuations, it's unlikely to offer substantial gains in terms of loss and accuracy. Therefore, early stopping at this point could be a viable strategy to optimize training efficiency and avoid potential overfitting, which can occur if the model memorizes irrelevant details from the training data and performs poorly on unseen data.

4.4.4 Results of ViT

The fourth model implemented is the ViT model. The proposed approach achieved high accuracy. This achievement shows that ViTs have the potential to be a useful tool for HDR. In addition to accuracy, the model

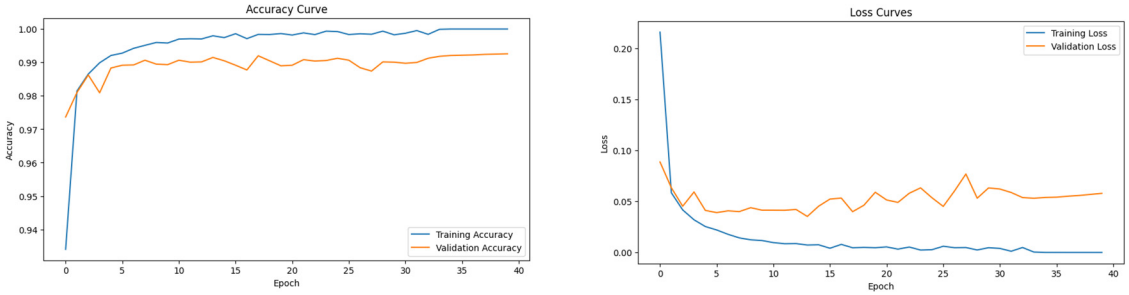

Figure 5: Graphs of the loss and accuracy curves for our CNN architecture. Source: Created by the author.

performed well in terms of precision, recall, and *F*1-score, all of which were achieved by 98.70%. This indicates that the model performs exceptionally well in detecting TPs (precision) and reducing FPs (recall), in addition to correctly predicting the correct digits the majority of the time (high accuracy). The *F*1-score, combining both concepts, further confirms the model's balanced performance. It is noteworthy that in the last fold of cross-validation, the maximum accuracy reached was considerably greater, at 99.10%. This further highlights the capability of ViTs can understand the nuances of handwritten digits. Figure 6 provides a visualization of the model's learning process, depicting the loss and accuracy curve graphs for the ViT model trained on the MNIST dataset. Analyzing these curves can provide insights into the training dynamics, such as the convergence point where the model's performance stabilizes.

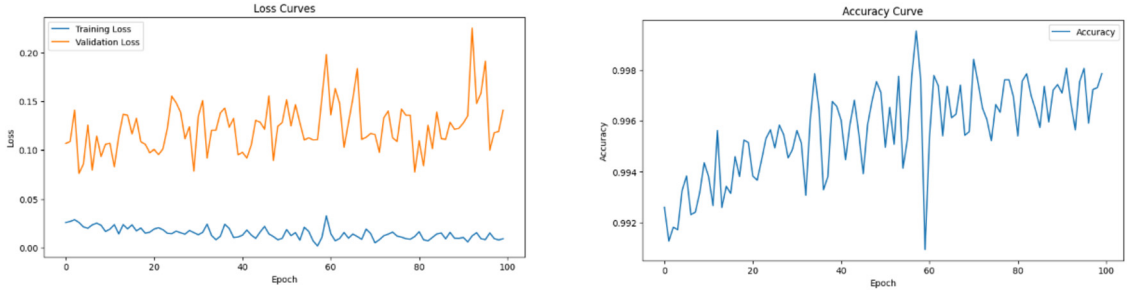

Figure 6: Graphs of the loss and accuracy curves for our ViT model. Source: Created by the author.

Figure 7 shows the confusion matrix generated using the ViT model's architecture, which was trained on the EMNIST dataset and tested on the dataset's samples.

4.4.5 Results of hybrid models

4.4.5.1 Ensemble learning

In this experiment, many ML models were combined to create hybrid models. These networks utilized a CNN for feature extraction. The extracted features were then fed to various classifiers for the final classification task. LR, RF, and VC were the ML models used. A comprehensive summary of all the obtained results is presented in Table 7.

The ensemble model CNN + VC achieved impressive results with a training and validation accuracy of 99.27%. This significantly outperformed both CNN + RF (98.20%) and CNN + LR (88.67%). These results suggest strong learning of data features and good generalization to unseen data, as evidenced by the high validation accuracy. However, evaluating the model's performance on an independent test set remains crucial for confirming its effectiveness in real-world scenarios.

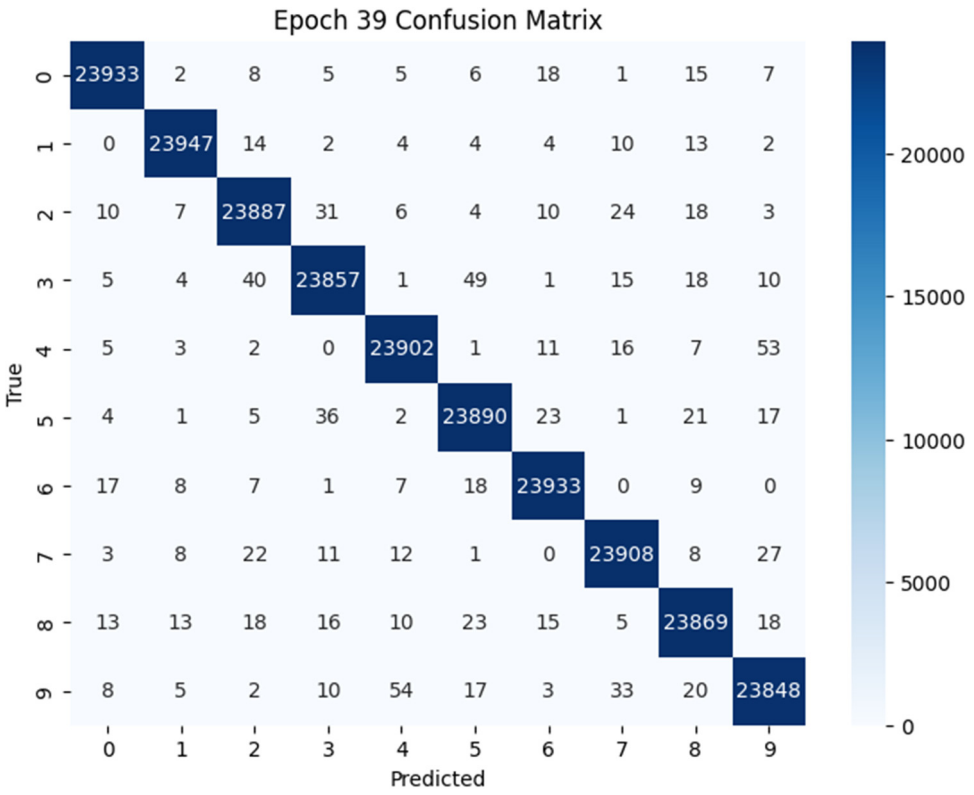

Figure 7: Confusion matrix results using ViT model on EMNIST-digit dataset. Source: Created by the author.

Table 7: Hybrid ML models trained on MNIST dataset for the last 20 epochs

Model	Precision (%)	Recall (%)	*F*1-score (%)	Accuracy (%)
CNN + LR	89.05	89.05	89.05	88.67
CNN + RF	98.97	98.97	98.97	98.20
CNN + VC	99.53	99.53	99.53	99.27

In this experiment, multiple ML models were combined to create hybrid models, leveraging the strengths of each algorithm. CNN was employed for feature extraction from the EMNIST dataset, capturing spatial hierarchies in the handwritten digits. The extracted features were then fed into various classifiers to perform the final classification task. The classifiers used in this study include LR, RF, and a VC, which combines predictions from multiple models for improved performance. A comprehensive summary of all the results obtained from these hybrid models is presented in Table 8.

Table 8: Hybrid ML models trained on EMNIST dataset for the last 20 epochs

Model	Precision (%)	Recall (%)	*F*1-score (%)	Accuracy (%)
CNN + LR	99.56	99.56	99.56	99.79
CNN + VC	99.62	99.62	99.62	99.83
CNN + RF	99.68	99.68	99.68	99.86

The results presented in Table 8 highlight the strong performance of the hybrid ML models trained on the EMNIST dataset over the last 20 epochs. The CNN combined with RF achieves the highest metrics across all evaluation criteria, with precision, recall, *F*1-score, and accuracy all reaching 99.68% and an accuracy of 99.86%. This suggests that the combination of CNN for feature extraction and RF for classification provides the most effective synergy in handling EMNIST data. The CNN + VC and CNN + LR models also perform exceptionally well, with accuracies of 99.83 and 99.79%, respectively, and slightly lower precision, recall, and *F*1 scores. While all models exhibit near-perfect results, the minor variations between them indicate that the choice of classifier impacts performance, with RF proving to be the most optimal in this case. These results suggest that combining DL with ensemble techniques or simpler classifiers can yield robust performance in handwritten character recognition tasks.

4.4.5.2 CNN and ViT

The hybrid model combining CNN and ViT achieved impressive results with a training and validation accuracy of 99.97%. This significantly outperformed both CNN (99.90%) and ViT (98.70%). Figure 8 shows the confusion matrix generated using the architecture of the hybrid model combining CNN and ViT, which is trained on the MNIST and tested on the dataset's samples.

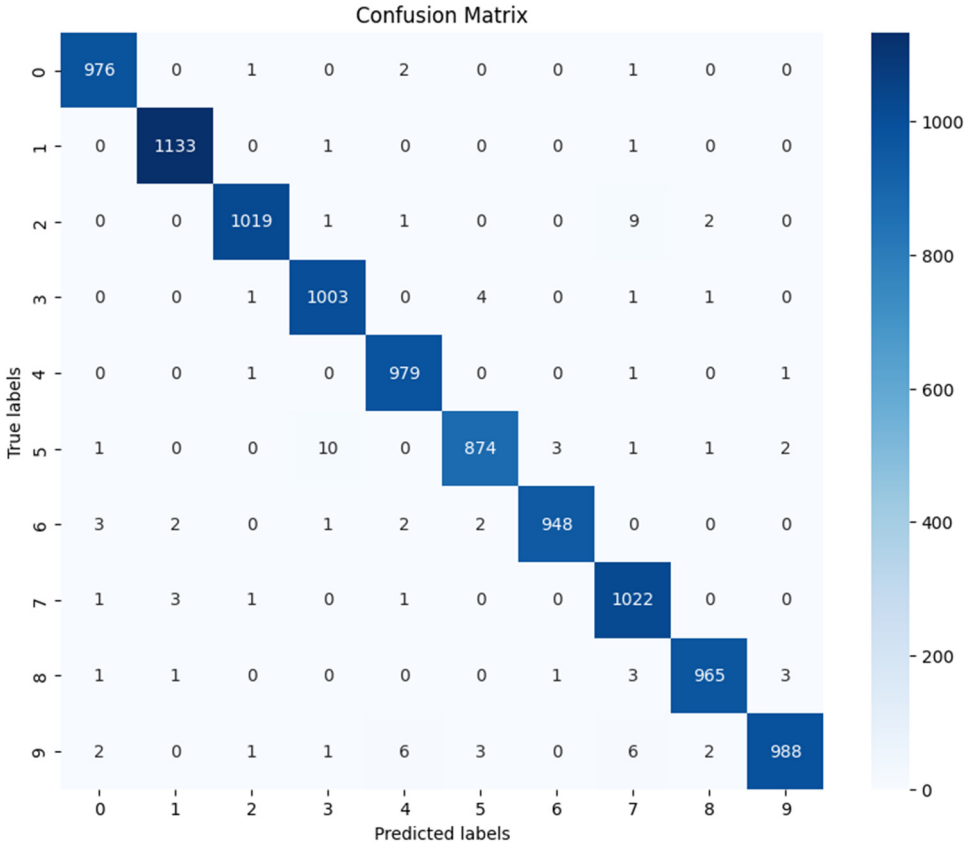

Figure 8: Confusion matrix results using hybrid model combining CNN and ViT on MNIST-digit dataset. Source: Created by the author.

Figure 9 provides ROC curves and AUC scores that indicate that the model is performing exceptionally well across all ten classes. The high AUC scores, approaching 1.0 for most classes, suggest that the model has excellent discriminatory power. This means the model can accurately distinguish between positive and negative instances for each class. While the overall performance is impressive, there might be slight variations

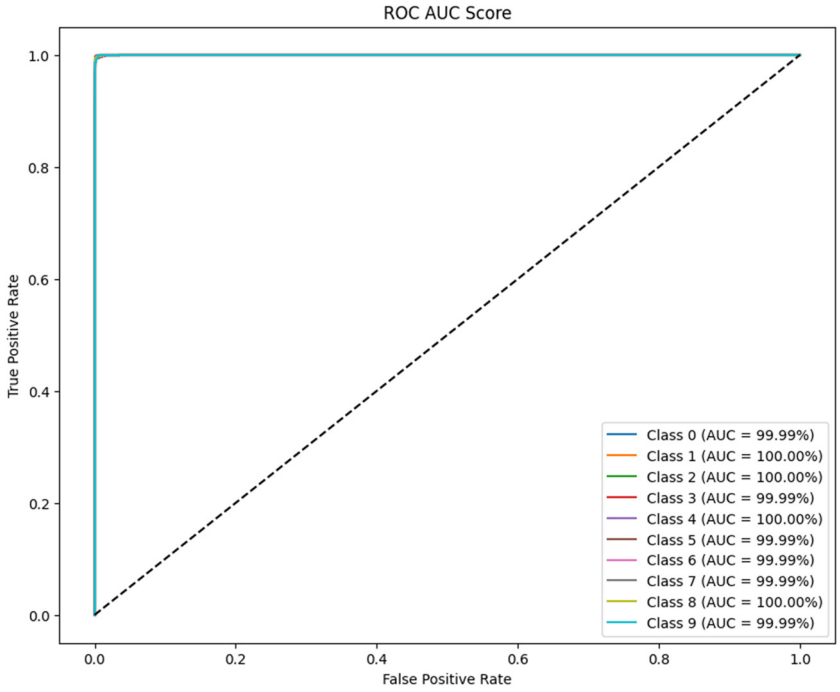

Figure 9: Evaluation of a multi-class classifier using ROC curves and AUC metrics for the MNIST dataset. Source: Created by the author.

in the AUC scores across different classes. However, in this specific case, all classes exhibit near-perfect performance. True positive rate (TPR): the proportion of positive instances that are correctly identified as positive. False positive rate (FPR): the proportion of negative instances that are incorrectly identified as positive. An ideal classifier would have an ROC curve that hugs the top-left corner, indicating high sensitivity and specificity. The AUC score quantifies the overall performance of the classifier, with a higher AUC indicating better performance.

The hybrid model combining CNN and ViT achieved impressive results with a training and validation accuracy of 98.26% trained on a different dataset (EMNIST). Figure 10 shows the confusion matrix generated using the architecture of the hybrid model combining CNN and ViT, which is trained on the EMNIST dataset and tested on the dataset's samples. We fixed 80 for the number of epochs for this hybrid model (CNN + ViT), and we used the early stop technique to prevent the overfitting.

Figure 11 illustrates the ROC curves and AUC scores that indicate that the model is performing exceptionally well across all 26 classes. The high AUC scores, approaching 1.0 for most classes, suggest that the model has excellent discriminatory power. This means the model can accurately distinguish between positive and negative instances. While the overall performance is impressive, there are slight variations in the AUC scores across different classes. Some classes, like Class 17, have slightly lower AUC scores compared to others. This could be due to various factors such as class imbalance, data quality, or model complexity.

5 Discussion

5.1 Comparative analysis of the proposed models

This study aimed to develop a high-performing model for accurate recognition of English handwritten digits. The employed methodology and achieved results raise several important considerations. The evaluation metrics used in this research include accuracy, precision, recall, $F1$ score, Cohen's kappa score, and ROC AUC score. Table 9 demonstrates the performance of the CNN model along with other proposed models.

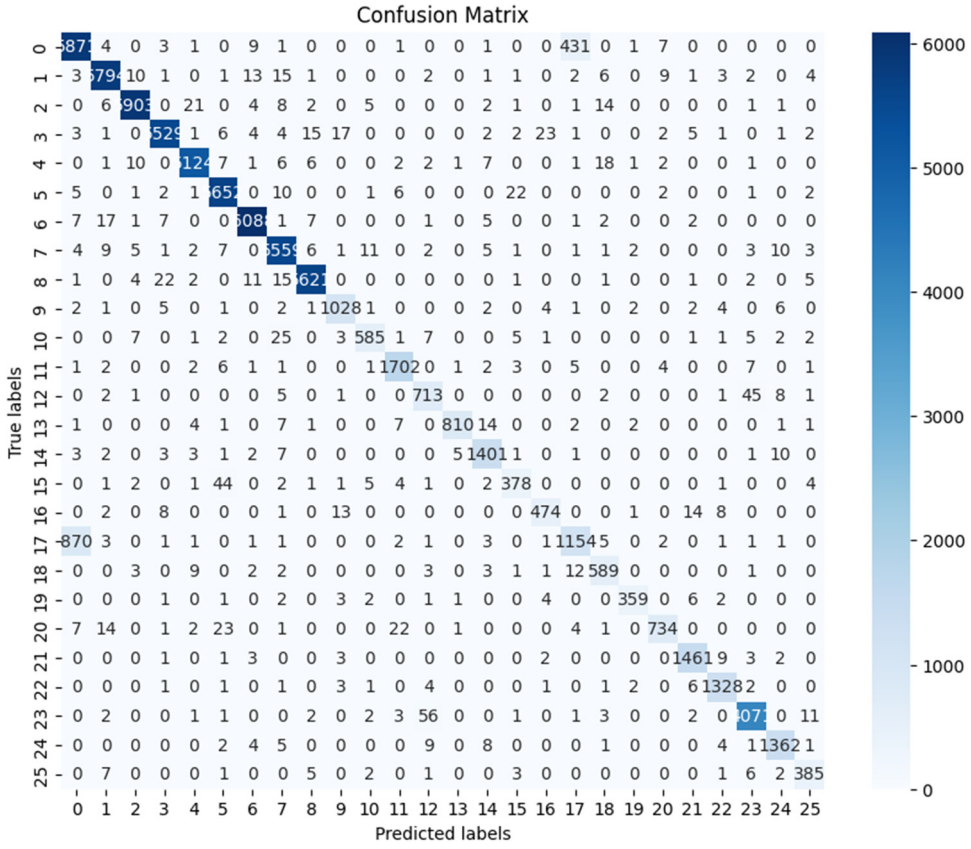

Figure 10: Confusion matrix results using hybrid model combining CNN and ViT on EMNIST dataset. Source: Created by the author.

First, the selected standalone models–LR MLP, CNN, and ViT–achieved varying levels of performance (92.95, 99.14, 99.90, 98.70%, respectively). CNN consistently demonstrated superior performance due to its inherent ability to learn spatial features critical for digit identification. LR provided a reasonable baseline, with performance enhanced slightly through regularization. Increasing the number of training epochs from 20 to 200 yielded less than a 2% accuracy improvement, indicating that most learning occurred early in training. For MLP, three experimental techniques were employed. The first compared two network structures: one with fewer layers but more neurons, which outperformed the deeper alternative. The second examined four initialization methods; zero, uniform, and standard normal initializations degraded performance, whereas Xavier's method had minimal impact. The third tested four optimization methods; Adagrad slightly improved accuracy (by 0.23%) but also increased loss (by 0.0074). Overall, MLP outperformed LR, suggesting that deeper learning models are more effective for this task. CNN achieved an impressive 99.90% accuracy in HDR. Its performance can be attributed to: (i) effective feature extraction, (ii) robustness to variations in image quality, and (iii) the quality and quantity of training data. The ViT model also performed well (98.70% accuracy), though CNN outperformed it in terms of accuracy. However, ViT's potential for capturing global dependencies provides a valuable complementary capability.

Second, hybrid models incorporating RF, VC, and LR with CNNs yielded mixed outcomes. The CNN + Voting Classifier model performed well (99.27%), but the improvement over standalone CNN (99.90%) was marginal. Conversely, the CNN + LR combination performed poorly (88.67%), indicating that LR may not be suitable for ensemble use in this context. Nonetheless, the ensemble models generally outperformed individual models in various metrics, supporting the notion that combining models can enhance performance.

Finally, the CNN + ViT ensemble consistently outperformed the individual models and other ensemble combinations in terms of accuracy, precision, recall, *F*1-score, and ROC AUC score. This indicates that the combination of CNNs and ViTs effectively leveraged their complementary strengths for HRD. CNNs excel at extracting local features from images, while ViTs are adept at capturing global dependencies. Combining these

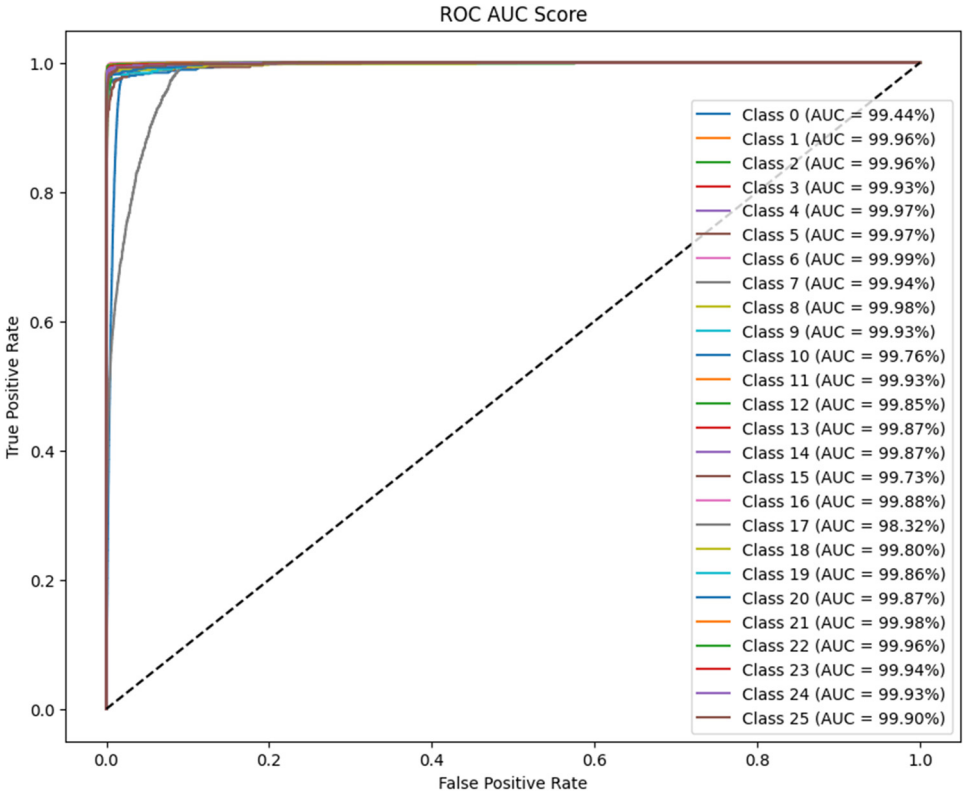

Figure 11: Evaluation of a multi-class classifier using ROC curves and AUC metrics for the EMNIST dataset. Source: Created by the author.

Table 9: Recognition performance of the proposed models for the MNIST dataset

Pre-trained models	Accuracy (%)	Precision (%)	Recall (%)	*F*1-score (%)	Cohen's kappa (%)	ROC AUC score (%)	Loss
CNN + LR	88.67	89.05	89.05	89.05	88.10	88.94	0.4635
LR	92.95	92.99	93.00	93.00	91.45	92.75	0.2679
CNN + RF	98.20	98.97	98.97	98.97	97.90	98.00	0.0599
ViT	98.70	98.95	98.95	98.95	97.73	98.80	0.0406
MLP	99.14	98.82	98.82	98.82	97.86	98.21	0.0532
CNN + VC	99.27	99.53	99.53	99.53	98.12	99.59	0.00235
CNN	99.90	99.31	99.31	99.31	99.23	99.99	0.0011
CNN + ViT	**99.97**	99.99	99.99	99.99	98.97	99.99	1.1067×10^{-7}

two architectures allows for a more comprehensive understanding of the image content. CNNs and ViTs have different strengths and weaknesses. By combining them, the ensemble can benefit from the advantages of both models, leading to improved performance. Ensembles can help reduce overfitting by introducing diversity and averaging the predictions of multiple models. The CNN + ViT ensemble might have better generalization capabilities, making it more robust to variations in unseen data. Overall, the CNN + ViT ensemble demonstrates a significant improvement in HDR performance compared to the individual models and other ensemble combinations. This suggests that combining CNNs and ViTs is a promising approach for HRD tasks.

In conclusion, while this study demonstrates the potential of various approaches for HDR, the results highlight the importance of careful model selection and hyperparameter tuning. Further investigation into deeper CNN architectures and exploring other advanced techniques like data augmentation could be important for pushing the boundaries of performance. This extended paragraph provides a critical analysis and

highlights several areas for future improvement, drawing attention to potential limitations and alternative solutions. It also emphasizes the need for further exploration and optimization.

Table 10 presents the performance of pre-trained models on the EMNIST dataset. LR performed the weakest (86.63% accuracy), while CNN + ViT achieved 98.26% accuracy with strong precision, recall, and *F*1 scores. The standalone ViT model reached 99.58% accuracy, setting a benchmark. Hybrid models (CNN with LR, VC, and RF) also showed excellent results, with accuracies from 99.79 to 99.86% and low loss values. Among them, CNN + RF achieved the highest accuracy (99.86%) and an ROC AUC score of 99.99%.

Table 10: Recognition performance of the proposed models for the EMNIST dataset

Pre-trained models	Accuracy (%)	Precision (%)	Recall (%)	*F*1-score (%)	Cohen's kappa (%)	ROC AUC score (%)	Loss
LR	86.63	84.77	85.88	84.18	85.37	85.63	0.3595
CNN + ViT	98.26	96.00	96.00	96.00	96.12	99.83	0.0479
ViT	99.58	99.58	99.58	99.58	99.72	99.78	0.0931
CNN + LR	99.79	99.56	99.56	99.56	99.66	99.81	0.0787
CNN + VC	99.83	99.62	99.62	99.62	99.71	99.89	0.0521
CNN + RF	99.86	99.68	99.68	99.68	99.77	99.99	0.0131
CNN	99.92	99.31	99.31	99.31	99.23	99.98	0.0031

Models combining CNN with LR, VC, and RF also show outstanding performance, with accuracies ranging from 99.79 to 99.86%, and low loss values, demonstrating that hybrid models improve classification. Among them, CNN + RF achieves the highest accuracy of 99.86% and a near-perfect ROC AUC score of 99.99%. The CNN model alone leads in accuracy (99.92%) and shows the lowest loss, indicating its exceptional fit and effectiveness for the task.

Overall, CNN-based models, particularly CNN and CNN + RF, show the best performance, with accuracy near 100% and minimal loss, proving CNNs to be highly effective for the EMNIST dataset. The ViT model excels for precision and recall, while hybrid models combining CNN with other classifiers like LR and RF further enhance performance, demonstrating the power of DL models for complex image recognition tasks.

5.2 Comparative analysis with state of the art

Our proposed models achieve competitive performance compared to the state-of-the-art approaches presented in Table 11. Even though some standalone models, such as MLP (98.36%), attain promising accuracy, they are not as accurate as the best models, such as EnsNet (99.84%) and Capsule Networks (99.87%). This demonstrates how well DL – in particular, CNN-based methods – performs in HDR. Interestingly, our ensemble learning strategy using CNN with VC (99.27%) achieves comparable or even surpasses the accuracy of several related works. Our ViT model (98.70%) also demonstrates competitive performance, potentially requiring less training data compared to other ViT-based methods like CvT (90.6%), as suggested by the limitations mentioned in the table. Our CNN model's performance (99.90%) is comparable to or better than the Simple CNN model reported in the study of Assiri [7] (99.83%). CNNs and ViTs are both powerful architectures for image classification tasks. Our proposed CNN model achieved very high accuracy (99.90%), while the CNN + ViT ensemble demonstrated even better results (99.97%), which is comparable to or surpasses the performance of many state-of-the-art methods reported in the literature (e.g., EnsNet, Capsule Networks, EfficientDet-D4). Overall, our work shows the potential of several approaches to tackle the HDR problem and provides interesting options. We conclude that DL models, especially the hybrid models combining CNN and ViT, consistently outperform traditional ML methods.

The comparative analysis with established methods from the literature, as shown in the second part of the Table 11, demonstrates that the CNN + RF model on EMNIST (99.86%) and CNN (99.92%) outperform many

Table 11: Comparative analysis with state of the art

Method	Accuracy rate (%)
SVM [2]	97.83
Decision tree [3]	83.40
PCA/LDA [5]	86.60
Consensus clustering [6]	95
Simple CNN [7]	99.83
EnsNet [8]	99.84
Ensemble CNNs [9]	99.79–99.82
Capsule networks [10]	99.87
EfficientDet-D4 [11]	99.83
ViT + GCN [22]	93.31
ViT [18]	86
CvT [19]	90.60
CVT [20]	99.89
CViT [21]	99.89–99.81
CNN + LR (MNIST)	88.67
LR (MNIST)	92.50
MLP (MNIST)	98.10
CNN + RF (MNIST)	98.20
ViT (MNIST)	98.70
CNN + VC (MNIST)	99.27
CNN (MNIST)	99.90
CNN + ViT (MNIST)	99.97
LR (EMNIST)	86.63
CNN + ViT (EMNIST)	98.26
ViT (EMNIST)	99.58
CNN + LR (EMNIST)	99.79
CNN + VC (EMNIST)	99.83
CNN + RF (EMNIST)	99.86
CNN (EMNIST)	99.92

traditional approaches like SVM (97.83%), Decision Tree (83.40%), and PCA/LDA (86.60%), highlighting the significant advances DL has made in image recognition tasks. For instance, CNN (EMNIST) (99.92%) outperforms Ensemble CNNs (99.79–99.82%) and EfficientDet-D4 (99.83%), both of which are highly respected methods in image classification. CNN + RF (EMNIST) (99.86%) also rivals other top models such as Capsule Networks (99.87%) and outperforms models like SVM and ViT + GCN, which achieved accuracy rates of 93.31 and 86%, respectively. ViT (EMNIST) (99.58%) performs better than ViT (MNIST) (98.70%), indicating that the model generalizes well to more complex datasets. In summary, the CNN-based models (particularly CNN and CNN + RF) not only excel in terms of accuracy but also outperform a wide array of traditional and state-of-the-art methods, reaffirming the effectiveness of DL for complex image recognition tasks. The ViT model, alone or combined with CNN, demonstrates high performance as well, further reinforcing its potential in visual recognition tasks.

This comparison highlights how competitive our proposed approaches are in the larger context of HDR research. While well-known methods like CNNs show good performance, our work investigates other techniques such as ensemble learning, ViTs, and combining CNN with ViT providing useful choices for future research and improvement in the field.

6 Conclusion

We provided a comprehensive comparison of ML and DL models for HDR, offering valuable insights into their strengths and weaknesses. We investigated standalone models trained from scratch, ensemble learning, and

ViTs. By comparing these models, we aim to identify the most effective methods for this task and provide insights into their potential applications. Our findings show how well DL techniques, in particular our proposed approaches, perform when it comes to obtaining high accuracy on HDR tasks. Our proposed model combining CNN and ViT demonstrated even better results (99.97%), which surpasses the performance of many state-of-the-art methods reported in the literature, the accuracy of our proposed CNN model (99.90%) is comparable to or better than many literature methods, our CNN with VC ensemble achieved also a remarkable accuracy (99.27%), showcasing the potential of combining multiple models for improved performance. Furthermore, our ViT model demonstrated encouraging outcomes (98.70%), indicating its viability as an alternative approach, especially given its capacity to function well with smaller datasets.

In future work, we aim to explore the performance of these models on larger and more diverse datasets that could provide further insights. We will investigate new CNN and ViT architectures that lead to even better results like the hybrid Light-Weight ViT. We plan to apply transfer learning techniques to leverage pre-trained models on larger datasets to improve efficiency and performance. And, we aim to explore the deployment of these models in real-world applications, such as automated document processing or optical character recognition.

Funding information: The authors state no funding involved.

Author contributions: Conceptualization, D.B.N.; data curation, D.B.N.; formal analysis D.B.N.; investigation, D.B.N.; validation, D.B.N.; writing original draft preparation, D.B.N.; writing review and editing, D.B.N. All authors have read and agreed to the published version of the manuscript.

Conflict of interest: The authors state no conflict of interest.

Data availability statement: The datasets used in this study are publicly available at MNIST Dataset and EMNIST Dataset.

References

[1] Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, et al. An image is worth 16 × 16 words: transformers for image recognition at scale, in: International Conference on Learning Representations. 2021. p. 293–97. doi: 10.48550/arXiv.2010.11929.

[2] Reddy BP, Reddy RS, Vasem PS, Venkatesh P, Rajashekaran S. Handwritten digit recognition using SVM algorithm in machine learning. Int J Creative Res Thoughts. 2022;10(6). Accessed 30 October 2024. [Online]. Available: https://ijcrt.org/papers/IJCRT22A6520.pdf.

[3] Assegie T, Nair P. Handwritten digits recognition with decision tree classification: A machine learning approach. Int J Electr Comput Eng. 2019;9(5):4446–51. doi: 10.11591/ijece.v9i5.pp4446-4451.

[4] Wang Y, Wang R, Li D, Adu-Gyamfi D. Improved handwritten digit recognition using quantum k-nearest neighbor algorithm. Int J Theoret Phys. 2019;58(7):2331–40. doi: 10.1007/s10773-019-04124-5.

[5] Sheikh R, Patel M. Handwritten digit recognition using different dimensionality reduction techniques. Int J Recent Tech Eng. 2019;8(2):999–1002. doi: 10.35940/ijrte.B1798.078219.

[6] Monica RF, Lavanya K. Handwritten digit recognition of mnist data using consensus clustering. Int J Recent Tech Eng. 2019;7(6):1969–73. Accessed 21 June 2024. [Online]. Available: https://www.ijrte.org/wp-content/uploads/papers/v7i6/F2408037619.pdf.

[7] Assiri S. A simple CNN model for MNIST handwritten digits classification. Int J Adv Comput Sci Inform Tech. 2020;11(5):1517–24.

[8] Hirata Y, Fujiyoshi H. EnsNet: An ensemble of convolutional neural networks for robust digit recognition. 2021. Accessed 10 July 2024. [Online]. Available: https://arxiv.org/abs/2008.10400.

[9] An H, Sun C, Wang X. Deep ensemble convolutional neural network for digit recognition, in 15th International Conference on Computer Science and Information Technology (ICCSIT). 2018. Vol. 10. p. 309–13.

[10] Byerly A, Dollár P, Goodman N, Srinivasan P, Zitnick CL, LaTeX: Capsule networks with homogeneous vector capsules, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020. p. 6012–21.

[11] Ahmed SS, Mehmood Z, Awan IA, Yousaf RM. A novel technique for handwritten digit recognition using deep learning. J Sensors. 2023;2023:2753941. doi: 10.1155/2023/2753941.
[12] Bhojanapalli S, Chakrabarti A, Glasner D, et al. Understanding robustness of transformers for image classification, in IEEE/CVF International Conference on Computer Vision (ICCV). 2021. p. 211–21. doi: 10.1109/ICCV48922.2021.01007.
[13] Naseer M, Ranasinghe K, Khan SH, Hayat M, Shahbaz Khan F, Yang MH. Intriguing properties of vision transformers. Adv Neural Inform Proces Syst. 2021;34:23296–308. doi: 10.1109/ICCV48922.2021.01007.
[14] Paul S, Chen P. Vision transformers are robust learners. 2021. Accessed 18 July 2024. [Online]. Available: https://arxiv.org/abs/2008.10400.
[15] LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proc IEEE. 1998;86(11):2278–324. doi: 10.1109/5.726791.
[16] Chen C, Fan Q, Panda R. Crossvit: cross-attention multi-scale vision transformer for image classification. 2021. Accessed 04 August 2024. [Online]. Available: https://arxiv.org/abs/2103.14899.
[17] Wang J, Gao Y, Shi J, Liu Z. Optical remote sensing scene classification based on vision transformer and graph convolutional network. Acta Photon. Sin. 2021;50(11).
[18] Gheflati B, Hassan R. Vision transformer for classification of breast ultrasound images. 2021. Accessed 11 August 2024. [Online]. Available: https://arxiv.org/abs/2110.14731.
[19] Wu H, Xiao B, Noel C, Liu M, Dai X, Yuan C, et al. Cvt: introducing convolutions to vision transformers. 2021. Accessed 24 October 2024. [Online]. Available: https://arxiv.org/abs/2103.15808.
[20] Agrawal V, Jayant J. Convolutional vision transformer for handwritten digit recognition. Durham, NC, USA: Research square Company; 2022. doi: 10.21203/rs.3.rs-1560520/v1.
[21] Agrawal V, Jagtap J, Patil S, Kotecha K. Performance analysis of hybrid deep learning framework using a vision transformer and convolutional neural network for handwritten digit recognition. MethodsX. 2024;12(102554). doi: 10.1016/j.mex.2024.102554.
[22] Hong D, Gao L, Yao J, Zhang B, Plaza A, Chanussot J. Graph convolutional networks for hyperspectral image classification. 2020. Accessed 13 August 2024. [Online]. Available: https://arxiv.org/abs/2008.02457.
[23] Dixit R, Kushwah R, Pashine S. Handwritten digit recognition using machine and deep learning algorithms. Int J Comput Appl. 2020;176(42):27–33. doi: 10.5120/ijca2020920550.
[24] Chigozie N, Winifred I, Anthony G, Stephen M. Activation functions: Comparison of trends in practice and research for deep learning. 2018. [Online]. Available: https://arxiv.org/abs/1811.03378.
[25] Assiri Y. Stochastic optimization of plain convolutional neural networks with simple methods. 2020. Accessed 27 August 2024. [Online]. Available: https://arxiv.org/abs/1511.08458.
[26] O'Shea K, Nash R. An introduction to convolutional neural networks. 2015. Accessed 31 August 2024. [Online]. Available: arXiv:1511.08458.
[27] Riaz N, Arbab H, Maqsood A, Nasir K, Ul-Hasan A, Shafait F, et al. Conv-transformer architecture for unconstrained off-line urdu handwriting recognition. in Research Square, 2022. [Online]. Available: doi: https://doi.org/10.21203/rs.3.rs-1514700/v1.
[28] Hendrycks D, Gimpel K. Gaussian error linear units (gelus). 2016. Accessed 07 September 2024. [Online]. Available: https://arxiv.org/abs/1606.08415.
[29] Lecun Y, Cortes C, Burges CJ. MNIST handwritten digit database. 2010. Accessed 15 September 2024. [Online]. Available: http://yann.lecun.com/exdb/mnist.
[30] Greg C, Sadegh A, Jonathan T, Andre VS. EMNIST: Extending MNIST to handwritten letters. 2017. Accessed 25 September 2024. [Online]. Available: https://arxiv.org/abs/1702.05373.