A Comparative Study of Classification Algorithms on the EMNIST dataset

Cosmic Coders
(Arnab Singha , Kanan Pandit)
Ramakrishna Mission Vivekananda Educational and Research Institute
Belur, G. T. Road, PO Belur Math, Howrah, West Bengal 711202

25/11/2024

Abstract

Accurate handwritten character recognition is key to automating digitization and interaction tasks. The EMNIST dataset, with digits and letters, challenges models due to handwriting variability. This study compares machine learning methods, for finding the optimum model for recognition of handwritten digits.

Keywords: Hierarchical Classification, EMNIST Dataset, Handwritten Character Recognition, Multi-Class Classification, Optical Character Recognition.

1 Introduction

1.1 Background

Handwritten character recognition is part of an automation process of document digitization, form processing, and text-based human-computer interaction. Handwritten character recognition is a branch of Optical Character Recognition, which converts handwritten or printed text into machine-readable formats. The EMNIST (Extended MNIST) dataset extends the widely recognized MNIST set by incorporating handwritten uppercase and lowercase letters in addition to handwritten digits. This extension poses more difficulties as it involves different handwriting styles and is therefore a very good benchmark to test the robustness of machine learning algorithms. The comprehensive nature of the dataset allows the researcher to come up with and evaluate models that can handle real-world variations in handwriting.

1.2 Motivation

The motivation behind this project emanates from the ever-increasing demand for efficient and accurate OCR systems across various industries like education, banks, and medical institutions. While deep learning models hold superiority in accuracy, traditional machine learning models such as Logistic Regression, KNN, and Random Forest are still very important and useful for interpretability, lower computational needs, and simpler implementation. This project explores the performance of such models on the EMNIST dataset in order to evaluate their effectiveness and relevance in real-world applications, especially when computational power is limited or simplicity is emphasized.

1.3 Objectives

Our key goals of this project are the preprocessing and preparation of the EMNIST dataset to feed in the problem for machine learning tasks, classification models, including Logistic Regression, Softmax Regression, KNN, Decision Tree, Random Forest, and SVM, and comparison of their strengths and limitations to deduce the best model for the text classification of handwritten character recognition. Additionally, we aim to discuss challenges in the dataset that are associated with handwriting variation and recommend improvements for future recognition systems.

2 Literature Review

In 2017,[1] Gregory Cohen et al. introduced the EMNIST dataset, an extension of the well-known MNIST dataset, tailored to handwritten letters and digits. The EMNIST dataset, derived from the NIST Special Database 19, offers compatibility with existing MNIST-based systems by preserving its format while expanding its scope to include 62 classes of characters (digits, uppercase, and lowercase letters).

The study meticulously outlines the conversion process from the original 128×128 binary images of the NIST dataset to the standardized 28×28 grayscale format. The authors present baseline benchmarks using an Extreme Learning Machine (ELM) classifier and validate the consistency of the dataset through performance comparisons between EMNIST and the original MNIST digits. Results indicate EMNIST's potential as a more challenging classification benchmark while maintaining drop-in compatibility with MNIST systems.

This work highlights the need for advanced benchmarks to accommodate evolving classification challenges, emphasizing EMNIST's utility in facilitating research in optical character recognition and complex tasks involving letters and digits.

In 2019, Alejandro Baldominos et al.[2] presented a comprehensive survey titled *"A Survey of Handwritten Character Recognition with MNIST and EMNIST,"* which reviews state-of-the-art contributions to handwritten digit and character recognition. The study extensively examines techniques applied to the MNIST dataset, a benchmark for computer vision and machine learning, as well as its extended variant, EMNIST.

The paper distinguishes between methods involving data augmentation or preprocessing and those utilizing the raw datasets. Special attention is given to convolutional neural networks (CNNs), which dominate in performance. The authors analyze techniques achieving error rates below 1% on MNIST, with some models, such as those employing DropConnect and data augmentation, reaching as low as 0.21% error. For EMNIST, which includes handwritten letters and digits, the paper reports competitive results, highlighting the utility of CNNs and advanced architectures like capsule networks for achieving accuracies exceeding 95% for letters and 99% for digits.

The survey emphasizes the evolving landscape of recognition systems, showcasing the impact of both algorithmic innovation and hardware advancements in achieving near-perfect accuracy.

In 2021, Asha B. Shetty et al.[3] proposed a comprehensive approach to handwritten digit and character recognition, utilizing the MNIST and EMNIST datasets. The research integrates Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks to achieve robust performance in Optical Character Recognition (OCR). The system processes input images through stages of preprocessing, feature extraction, and classification, ensuring improved accuracy and efficiency.

The study presents a detailed methodology, highlighting the advantages of CNN and LSTM in feature extraction and classification tasks. Preprocessing techniques include noise reduction, contrast enhancement, and elimination of irrelevant portions, preparing the data for optimal algorithm performance. Benchmark results demonstrate high accuracies of 98.23% (CNN) and 98.35% (LSTM) for MNIST, and 85.17% (CNN) and 85.71% (LSTM) for EMNIST datasets.

This work underscores the effectiveness of deep learning algorithms in OCR applications, while addressing challenges such as variability in handwriting styles. The authors emphasize the practical applications of this research in areas like user authentication and automated document digitization.

In 2023,[4] a study titled *"Using Machine Learning Models and Deep Learning Networks for Handwritten Numbers and Letters Recognition"* by Sarmad Hamzah Ali presents a comprehensive exploration of classification algorithms and deep learning techniques for the recognition of handwritten characters. The study evaluates eight recognition technologies, employing Scikit-learn classifiers and deep learning neural networks with datasets such as MNIST, EMNIST, and CoMNIST.

The research outlines methodologies including sequential and convolutional neural networks, image preprocessing via OpenCV, and parameter tuning for classifiers like SVM, KNN, and MLP. Results demonstrate that deep learning models, particularly CNNs, achieve superior recognition accuracy, with precision levels of up to 98% for numerical digits, while challenges persist in recognizing Cyrillic characters (accuracy ranging from 75.6% to 84.8%). The incorporation of CUDA technology significantly enhances the efficiency of CNN training.

This work emphasizes the importance of dataset preprocessing and parameter optimization in improving recognition accuracy, establishing CNNs as the most effective model for handwritten character recognition.

There are many other works have been carried out on this dataset so far for on the context of recognition of hand written characters. Most of the models use Convolutional Neural Netwoks are used for better results.

Table 1: Comparative Study of Handwritten Recognition Papers

Author's Name	**Year**	**Models**	**Advantages**	**Drawbacks**
Gregory Cohen et al.	2017	Extreme Learning Machine (ELM)	Introduced the EMNIST dataset as a challenging extension to MNIST; ensured compatibility.	Baseline benchmarks only; lacked exploration of advanced models for better accuracy.
Alejandro Baldominos et al.	2019	CNNs, Capsule Networks	High accuracy for MNIST and EMNIST; detailed evaluation of modern architectures.	Limited focus on extended datasets like EMNIST and less attention to handwriting variability.
Asha B. Shetty et al.	2021	CNN, LSTM	Achieved high accuracy for MNIST and EMNIST; included preprocessing, feature extraction, and classification.	Moderate performance on EMNIST; lacks experimentation with advanced optimization or augmentation techniques.
Sarmad Hamzah Ali	2023	Sequential Neural Networks, CNN	High precision for numerical digits (98%+); CUDA integration for faster model training.	Struggles with accuracy for Cyrillic characters; performance declines with more complex datasets.

3 Data Description

3.1 Source

The dataset is collected from Kaggle and is part of the Extended MNIST (EMNIST) dataset series. It significantly extends the original MNIST dataset by including a broader variety of classes and features. It offers an easily accessible and versatile platform for machine learning practitioners to develop, test, and benchmark models for character recognition which are often used in Optical Character Recognition (OCR) tasks. We have selected the 'Byclass' split of the total dataset.

3.2 Features

The dataset contains grayscale images of 28×28 representing handwritten characters. It contains 784 pixel values each one as a feature of the dataset.
The pixel values are ranging from 0 to 255 and represents the gray scale of color spectrum.
Total data is splited into train and test set by default.
The size of the Training is 697,932 samples.
The size of the Test is 116,323 samples.

3.3 Target Variables

The dataset includes 62 distinct classes:

- Digits: 0-9.
- Uppercase letters: A-Z.
- Lowercase letters: a-z.

The target labels are encoded as integers, each corresponding to one of the 62 classes, facilitating compatibility with classification models. The class mapping is from the ACSII values of the characters to 0 to 61 integers sequentially. The mapping is as follows:

Character	**ASCII range**	**Label Range**
Digits (0-9)	48 to 57	0 to 9
Uppercase letters (A-Z)	65 to 90	10 to 35
Uppercase letters (a-z)	97 to 122	35 to 61

The dataset exhibits an imbalanced distribution, reflecting real-world frequencies of character usage, adding complexity to the classification task.

4 Data Preprocessing

4.1 Data Loading

We loaded the Train and Test dataset from the respective CSV file to a pandas DataFrame, both having 785 feature columns including the 28×28 pixel data and one column as the label.

4.2 Train-Test Split

The dataset is by default splitted into a Train and Test set with 697,932 and 116,323 samples respectively. However, the datasets are imbalanced.

4.3 Resizing the Dataset

We faced difficulty in handling the dataset with total 785 feature (having 28×28 images). So we resized the images using the Lanczos resampling method 14×14 pixels for reducing the computational complexity while retaining key structural details such as edges and shapes. We flattened the images into one-dimensional arrays, resulting in 196 features per sample.

4.4 Saving the Preprocessed Data

After resizing the images we saved both the Training and the Test sets as CSV files (train_byclass_new.csv and test_byclass_new.csv). And we loaded the sets again as new DataFrames.

4.5 Normalization

To ensure numerical stability during training and accelerate convergence we used normalization technique over the pixel intensity values. We scaled the pixel intensity values, originally ranging from 0 to 255, a range of [0, 1].

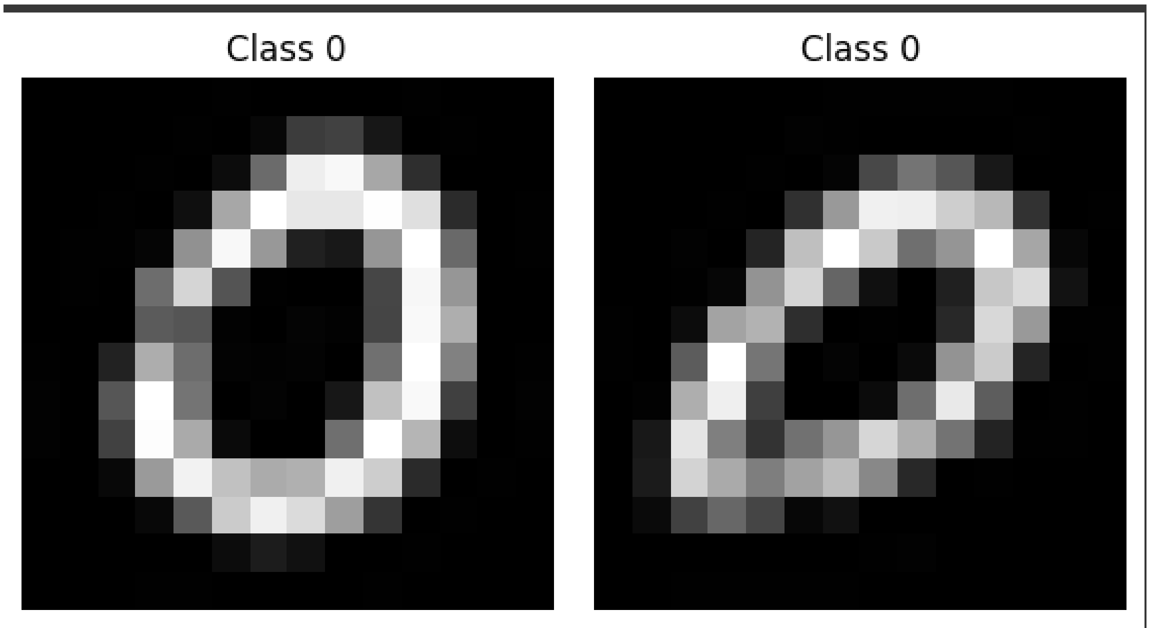

Figure 1: Image after size reduction Digit: 0

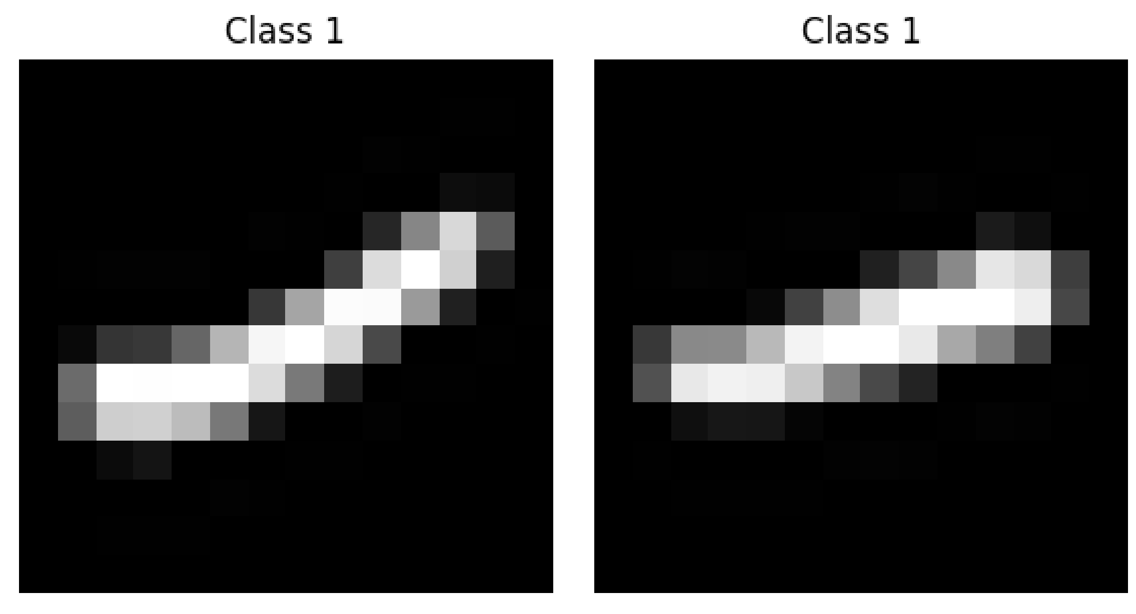

Figure 2: Image after size reduction Digit: 1

4.6 Checking for Missing and Nan Values

We checked for missing values in the dataset. The dataset contains no missing values. Also there is no outliers in the dataset.

4.7 Analyzing Class Distribution

The dataset contains 62 classes, but the distribution is imbalanced. Certain classes, such as frequently used digits or letters, are overrepresented, while less common characters are underrepresented.

4.8 Visualization

We plotted random samples of resized images using matplotlib to confirm their resolution, quality, and alignment. Also we plotted class distributions using bar plots and pie charts to highlight imbalances.

4.9 Structuring the Dataset for Specific Tasks

We implemented a custom two-layer model, for which we subdivided the data into three groups: digits (0-9), uppercase (A-Z) and lowercase(a-z) letters.

4.10 Final Output of Preprocessing

After all these processes both datasets are clean, normalized, and saved in a format ready for machine learning model training.

5 Methodology

5.1 Algorithms Used

1. Logistic Regression
2. Softmax Regression
3. Decision Tree
4. Random Forest
5. A custom Two-Layer model using Softmax Regression

5.2 Justification

5.2.1 Logistic Regression

Logistic Regression is a linear model used for binary or multi-class classification problems. Specifically, we use multinomial logistic regression to handle the multi-class nature of the EMNIST dataset. This model works by fitting a linear decision boundary between classes and outputting a probability distribution over the 62 classes, which is achieved through the Softmax function.

5.2.2 Softmax Regression

Softmax Regression is used to handle multi-class problems by applying the Softmax function to the output. In this case raw model outputs (logits) are converted into probabilities, ensuring they sum to 1. Softmax regression is essentially logistic regression, but generalized for multi-class classification. We applied Softmax regression because it provides a natural way to handle multi-class problems where each class is mutually exclusive. EMNIST dataset has 62 distinct classes, this algorithm is good for selecting the most likely character (class) for each image. Moreover, Softmax ensures that the outputs are probabilistic, which is beneficial for tasks where we want a clear ranking of possible classes.

5.2.3 Decision Tree Regression

Decision Trees are a non-linear model that splits the data into subsets based on feature values at each node. Each decision node represents a question about a feature, and branches correspond to different outcomes based on feature values. The final leaf nodes hold the class labels. We applied Decision trees because they can capture complex, non-linear relationships between features and classes. Decision trees are capable of handling interactions between features without requiring any transformations or feature engineering makeing them a good fit for the EMNIST dataset, which involves complex handwritten character shapes. Additionally, decision trees can handle categorical features well and can be pruned to avoid overfitting, providing interpretability and flexibility. We chose a maximum depth of 16 to balance model complexity and prevent overfitting while still capturing the necessary patterns in the data.

5.2.4 Random Forest Regression

Random Forest is an ensemble learning method that combines multiple decision trees to improve classification performance. Each tree in the forest is trained on a random subset of the data, and the final prediction is made by aggregating the results from all trees, typically using majority voting for classification. We applied Random Forest to improve model accuracy by reducing overfitting and variance, which are common challenges in individual decision trees. By averaging predictions from multiple trees, Random Forest tends to generalize better and is less sensitive to noise in the data. It is useful for our choosen high-dimensional dataset EMNIST, where the complexity of the images may lead to overfitting in single decision trees.

5.2.5 Custom Two-Layer Model Using Softmax

We applied this approach as it provides a justified method for handling the complexity of the EMNIST dataset by breaking down the problem into smaller, more manageable tasks. By categorizing the data into three broad groups—digits, uppercase letters, and lowercase letters—the model simplifies the classification task. Instead of treating all 62 classes equally, this first layer reduces the problem space, allowing specialized classifiers to focus on fewer categories. This modular approach enables the model to handle the complexity of handwritten characters more efficiently.

The hierarchical structure is effective in managing specific tasks. By addressing one class group at a time, the hierarchical model can reduce the overall complexity of the classification task, leading to better generalization. The specialized classifiers can learn more focused and relevant features for each group (for example numerical digits or uppercase letters), which improves performance compared to trying to classify all characters at once.

This approach allows for more efficient training and prediction, as classifiers are tailored to specific groups, reducing the computational burden on each model. It also provides flexibility to incorporate different strategies for each group (e.g., more complex models for difficult categories like lowercase letters).

5.3 Architecture of the Two-Layer Model

The two-layer model is doing the classification in two steps. It is initially classifying the input into three classes as digits, uppercase and lowercase letter. Then there are further three classifiers that are classifying the input into the actual class i.e. the actual character it is. Following is the diagram of this architecture -

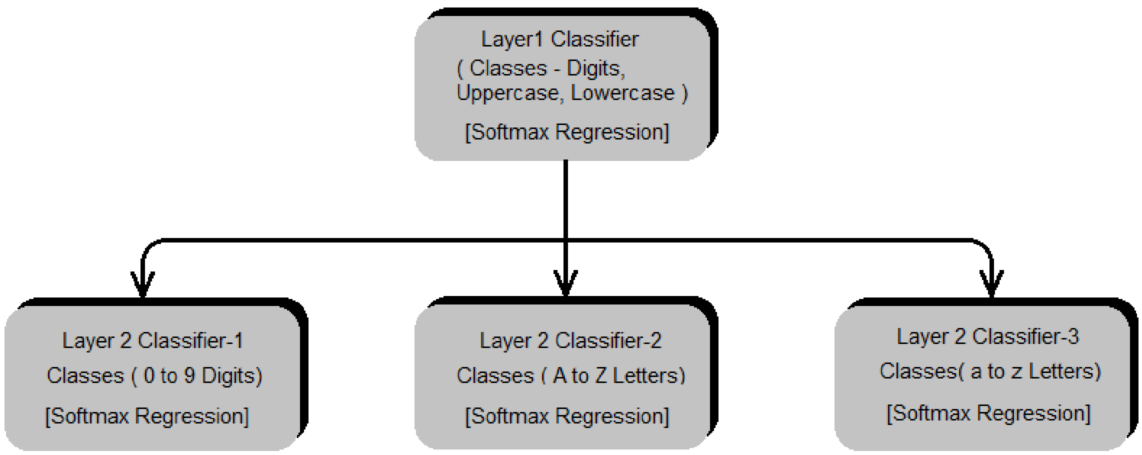

Figure 3: Architecture of Two-Layer Model

6 Implementation

6.1 Tools and Libraries

- Python
- NumPy
- Pandas
- Matplotlib
- Scikit-learn
- Google Drive
- Seaborn

- PIL (Python Imaging Library)
- Google Colab

6.2 Parameters

We have used some useful hyperparameters for the two of the implemented models.

Softmax Regression

max_iter = 1000.
Ensures that the optimization algorithm converges.

Decision Tree

max_depts = 16
We use this parameter for limits the depth of the tree, preventing it from growing too large and overfitting the training data. A maximum depth of 16 strikes a balance between model complexity and generalization.

Random Forest

n_estimators = 500
Creates 500 decision trees in the forest.

max_leaf_nodes = 20
Limits the number of leaf nodes to 20 in each tree.

n_jobs = -1
Uses all the available CPU cores.

6.3 Training Process

The total EMNIST byclass dataset was splited by default in training and test. We reduced the dimensions of the images, as a result the dataset's features got reduced. We also normalized the data.

We used these sets to train the models (Linear Regression, Softmax Regression, Decision Tree and Random Forest) using the the scikit-learn library. We trained the Two-Layer model using the Softmax Regression at each layer for separate classes of classifiers. The first layer of the dataset is categorized into three groups: digits (0-9), uppercase letters (A-Z), and lowercase letters (a-z). The first-layer classifier predicts the group of each input. In the second layer of the model, specialized classifiers were trained to handle fine-grained classification within each group.

7 Results

Evaluation Metrics

After training, testing, and evaluating various machine learning models on the EMNIST dataset, we analyzed their performance using key metrics such as training and test accuracy, precision, recall, and F1 score. These metrics provided insights into the effectiveness of each model in handling the complexities of handwritten character recognition. The results of our evaluation are summarized in the table below.

Table 2: Model Performance Comparison

Model	**Training Accuracy**	**Test Accuracy**	**Precision**	**Recall**	**F1**
Logistic Regression	72%	72%	0.69	0.72	0.70
Softmax Regression	73%	73%	0.71	0.73	0.71
Decision Tree	81%	71%	0.69	0.71	0.70
Random Forest	48%	48%	0.38	0.48	0.36
Two-Layer	-	58%	0.54	0.58	0.54

Comparison of performance of different models

Logistic Regression

Training and test accuracy (**72%**) with balanced precision (**0.69**) and recall (**0.72**).

F1 score (**0.70**) reflects moderate success in classification across all classes.

Solid baseline model, reliable but less effective than Softmax Regression for multi-class problems.

Softmax Regression

Achieved the highest test accuracy (**73%**) and best overall F1 score (**0.71**), balancing precision (**0.71**) and recall (**0.73**).

Slightly better results than Logistic Regression.

Decision Tree Classifier

Scored the highest training accuracy (**81%**) but experienced a drop in test accuracy to (**71%**), indicating overfitting.

Precision (**0.69**) and recall (**0.71**) were lower than those of the regression models.

Effective at learning complex patterns but struggled with generalization.

Random Forest Classifier

Underperformed across all metrics, with the lowest test accuracy (**48%**) and F1 score (**0.36**).

Precision (**0.38**) and recall (0.48) were significantly lower, indicating frequent large errors.

Least effective model due to insufficient hyperparameter tuning.

Two Layer Classifier

its Breaks down the classification task into manageable subsets, improving efficiency.

Achieves (**58%**)accuracy, F1 Score (**0.54**),Precision (**0.54**) and recall (**0.58**) showing balanced precision and recall.

Provides a foundation for enhancing hierarchical models with better techniques.

Best Performing Model : Softmax Regression

It achieved the highest test accuracy (**73%**), demonstrating good generalization.

Balanced precision (**0.71**) and recall (**0.73**), leading to the best F1 score (**0.71**).

8 Discussion

The study analyzes the EMNIST dataset using several classification algorithms, including Logistic Regression, Softmax Regression, Decision Trees, Random Forests, and a two-layer hierarchical classifier. Logistic Regression achieved **72%** training and testing accuracy, indicating its capacity to learn generalized patterns from the dataset. Its limited performance reflects the simplicity of the model in handling complex datasets with **62 classes**, where inter-class similarities may pose challenges. Softmax Regression performed slightly better than Logistic Regression with **73%** training and testing accuracy. The use of multinomial loss makes it more suited for multi-class classification, but the results suggest that it might not fully capture the nuances of all class distinctions.

The Decision Tree Classifier achieved **81%** training accuracy but reduced to **71%** test accuracy, suggesting overfitting to the training data. The shallow depth (set at **16**) likely constrained the model's ability to generalize while preventing it from over-complexity. The Random Forest Classifier significantly underperformed with **48%** test accuracy, despite being an ensemble of trees. This performance is likely due to the **max_leaf_nodes** limit (20) and possibly suboptimal hyperparameters, which restricted its learning capacity.

The Two-Layer Hierarchical Model categorized data into three groups: digits, uppercase letters, and lowercase letters, achieving **65**% test accuracy. Specialized classifiers for the digit category reached **94**% accuracy, for the uppercase letters category reached **86**% accuracy, and for the lowercase letters category reached **82**% accuracy. However, after combining the three separate classifiers, the model achieved a precision score of **0.54**, recall of **0.58**, and an F1 score of **0.54**. These results indicate the potential of hierarchical classification in managing dataset complexity but also reveal areas for improvement and indicate that errors in earlier stages propagate through the hierarchy, affecting overall performance.

Limitations of the Study

Data Imbalance

Certain classes, especially underrepresented ones, contributed to reduced model performance and generalization issues.

Feature Preprocessing

While resizing images to **14x14** and normalizing improved processing efficiency, this might have led to a loss of important fine-grained details.

Limited Hyperparameter Optimization

Logistic Regression and Random Forests, for instance, could benefit from additional optimization to explore better configurations.

Model Scalability

Decision Trees and Random Forests struggled with scalability due to the dataset's size, affecting their potential.

Evaluation Metrics

The reliance on macro and weighted averages in the classification report doesn't fully highlight the disparity in class-specific performances, especially for minority classes.

Computational Constraints

Training models like Random Forests with larger trees and more splits might have yielded better results but at a higher computational cost.

Lack of Cross-Validation

One significant limitation of this study is the absence of cross-validation during model evaluation. Cross-validation is a standard technique to assess model performance by splitting the dataset into multiple subsets, ensuring robust generalization and reducing the risk of overfitting. However, due to the large size of the EMNIST dataset, implementing cross-validation was computationally expensive and time-intensive. Instead, the dataset was split into training and testing sets for evaluation. While this approach provides a snapshot of model performance, it does not guarantee the robustness and consistency of the results across different splits. Future studies could address this limitation by employing stratified sampling or parallelized cross-validation methods to maintain efficiency while ensuring thorough evaluation.

Inability to Use Support Vector Machines (SVM)

Another limitation of this study is the exclusion of Support Vector Machines (SVM) as a classification algorithm. While SVMs are known for their effectiveness in high-dimensional spaces and their ability to handle complex decision boundaries, they are computationally expensive for large datasets like EMNIST. Training an SVM on such a dataset requires significant memory and processing time, especially when using non-linear kernels. This computational overhead made it infeasible to include SVMs in our analysis. Future research could explore strategies such as sampling, dimensionality reduction, or using approximate SVM techniques to mitigate these challenges and incorporate SVMs into the evaluation framework.

Suboptimal Performance of the Two-Layer Model

The two-layer hierarchical classification model, while conceptually promising, did not achieve satisfactory results across all categories. Although the first layer, which segmented data into digits, uppercase, and lowercase letters, achieved a moderate test accuracy of **65%**, subsequent specialized classifiers struggled to maintain high performance. The hierarchical approach introduced compounding errors; miss-classifications at the first layer propagated to the second layer, negatively impacting overall accuracy. Additionally, this approach may have been limited by the loss of fine-grained features during preprocessing and the lack of advanced techniques like neural network-based classifiers. Future improvements could involve integrating deeper learning architectures or ensemble strategies within the two-layer framework to enhance its effectiveness.

9 Conclusion

In this project, we explored the application of machine learning algorithms for handwritten character recognition using the EMNIST ByClass dataset. Despite the challenges posed by the dataset's size, complexity, and class imbalance, the study provided valuable insights into model performance, strengths, and limitations.

Key Takeaways

Logistic Regression and Softmax Regression: Logistic Regression and Softmax Regression achieved moderate accuracies of **72%** and **73%**, demonstrating their utility as baseline models but highlighting their limitations in handling complex data.

Decision Tree and Random Forest: Decision Trees achieved high training accuracy (**81%**), but overfitted, with test accuracy dropping to **71%**. Random Forest, constrained by computational limitations, achieved a low test accuracy of **48%**, failing to leverage its full potential.

Two-Layer Hierarchical Model: This hierarchical model offered a good approach by dividing the dataset into simpler categories. However, it achieved only **65%** accuracy in the first layer, with misclassifications propagating and impacting subsequent layers. This underscores the need for more robust implementations in hierarchical classification frameworks.

Challenges

In our dataset, class imbalance significantly affected all models, with rare classes being misclassified more frequently. Computational constraints further limited the exploration of advanced models like Support Vector Machines (SVMs) or hyperparameter tuning for existing algorithms.

Reflection on Meeting the Objectives

The objectives of evaluating multiple models, comparing their performance, and addressing challenges like overfitting and class imbalance were met through comprehensive analysis. Metrics such as accuracy, precision, recall, F1-score, and confusion matrices were used to identify strengths and weaknesses of each model. While the models provided insights into handling handwritten character recognition, they fell short of achieving robust, high-accuracy results for this complex and imbalanced dataset.

Future Research and Improvements

Deep Learning Models: Integrating Convolutional Neural Networks (CNNs) into the EMNIST dataset would greatly enhance performance, as CNNs excel in automatically extracting hierarchical features from image data. These networks can recognize complex patterns in handwritten characters, such as edges and shapes, which traditional machine learning models struggle to capture. CNNs may be effective for our EMNIST dataset for improved accuracy and generalization. By leveraging techniques like data augmentation, CNNs can also address class imbalance, helping the model better handle underrepresented classes. This integration could significantly boost the system's recognition capabilities.

Address Class Imbalance: To address class imbalance in the EMNIST dataset, techniques like oversampling and class-weighted loss functions can be explored. Oversampling duplicates samples from underrepresented classes. Class-weighted loss functions give more importance to misclassifications in minority classes, improving model focus on these classes. These methods can enhance fairness and accuracy for this dataset.

Model Optimization: To enhance model performance on the EMNIST dataset, cross-validation and advanced hyperparameter tuning should be used. Cross-validation ensures better generalization by training the model on different data subsets. Hyper-parameter tuning, through methods like grid or random search, helps find optimal model settings, improving accuracy and robustness. These techniques will lead to more reliable and effective models.

Hybrid Models: To boost accuracy, hybrid models can combine traditional machine learning methods with deep learning approaches, like CNNs, for better feature extraction and prediction. Additionally, using advanced ensemble techniques like XGBoost can improve performance by combining multiple weak learners, leading to stronger and more accurate models.

Scalability: To address computational constraints, distributed learning techniques can be implemented. These techniques allow the training process to be spread across multiple machines or GPUs, enabling the use of more resource-intensive algorithms like deep learning models. Distributed learning helps handle large datasets efficiently, speeds up training, and makes it feasible to explore more complex models, improving both scalability and performance.

References

[1] Cohen, Gregory, et al. "EMNIST: Extending MNIST to handwritten letters." 2017 international joint conference on neural networks (IJCNN). IEEE, 2017.

[2] Baldominos, Alejandro, Yago Saez, and Pedro Isasi. "A survey of handwritten character recognition with mnist and emnist." Applied Sciences 9.15 (2019): 3169.

[3] Shetty, Asha B., et al. "Recognition of handwritten digits and English texts using MNIST and EMNIST datasets." International Journal of Research in Engineering, Science and Management 4.7 (2021): 240-243.

[4] Ali, Sarmad Hamzah. "USING MACHINE LEARNING MODELS AND DEEP LEARNING NETWORKS FOR HANDWRITTEN NUMBERS AND LETTERS RECOGNITION." CENTRAL ASIAN JOURNAL OF MATHEMATICAL THEORY AND COMPUTER SCIENCES 4.12 (2023): 9-17.

[5] EMNIST (Extended MNIST) from Kaggle.com https://www.kaggle.com/datasets/crawford/emnist

Acknowledgement

We would like to place on record our deep sense of gratitude to Br Bhaswarachaitanya (Tamal Maharaj), Assistant Professor of Computer Science Department(Data Science), RKMVERI, Belur Math, Howrah, West Bengal, for his generous guidance, help and useful suggestions. We express our sincere gratitude to him for his guidance, continuous encouragement in the present work.

We are extremely thankful to Br Bhaswarachaitanya (Tamal Maharaj) for providing ourselves infrastructural facilities to work in, without which this work would not have been possible.

Arnab Singha (1st year BDA, B2430032)
Kanan Pandit (1st year BDA, B2430051)