Handwritten Character Recognition System Using Machine Learning And Deep Learning

Harsh Gupta[1], Adarsh Singh[2], Anurag Singh[3], Dr. Laveena Sehgal[4]
[1234]Department of Information Technology
[1234]IIMT College of Engineering, Gretaer Noida, UP, India
harshgupta5989@gmail.com ,

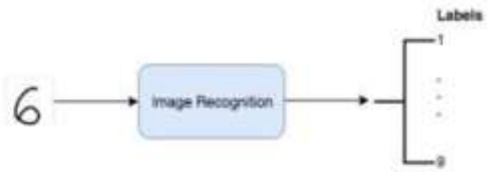
***Abstract* - This project focuses on developing a system that can recognize handwritten characters using machine learning and deep learning techniques. Handwritten character recognition helps convert written text into digital form, which is useful for document digitization and automation. The system uses image processing to clean and prepare the input, and a Convolutional Neural Network (CNN) model is trained to identify the characters. Well-known datasets like MNIST are used to train and test the model. The aim is to build an accurate and efficient system that can recognize various handwritten letters and numbers with high accuracy.**

Keywords – MNIST and EMNIST Dataset, Artificial Intelligence, Machine Learning, Deep Learning, Convolutional Neural Network, Neural Network, Image Processing, Support Vector Machine.

I. INTRODUCTION

Handwritten Character Recognition (HCR) is the process of automatically identifying and converting handwritten letters or numbers into digital text. It plays an important role in many real-world applications like reading postal addresses, scanning exam papers, or digitizing old handwritten documents. With the growing use of digital technologies, converting handwritten information into machine-readable format has become more useful and necessary.

Deep Learning (DL), a more advanced form of ML, employs multi-layered neural networks to process large datasets, enabling more accurate and complex decision-making. Unlike traditional ML algorithms, DL models can automatically extract and learn hierarchical features from raw data. This makes them particularly effective for high-dimensional data such as images, audio, and text. Deep Neural Networks (DNN), Deep Belief Networks (DBN), Convolutional Neural Networks (CNN), and Recurrent Neural Networks (RNN) are widely applied in diverse applications, including speech recognition, natural language processing (NLP), bioinformatics, medical imaging, and autonomous vehicles.



The general flow of an image recognition algorithm

The project begins with collecting or using existing datasets of handwritten characters, such as the MNIST dataset. Then, we preprocess the images by converting them to grayscale, resizing them, and removing noise. These steps help the system learn better. After that, the data is used to train a CNN model. The model learns to understand the features of each character and can later predict new, unseen characters.ML techniques include Supervised, Unsupervised, and Reinforcement Learning. While supervised learning uses labelled data for training, unsupervised learning deals with unlabelled data to identify hidden patterns. Deep Learning (DL), a subset of ML, overcomes the limitations of conventional algorithms by offering superior accuracy even on large datasets. Traditionally, recognizing handwritten characters was a difficult task for computers due to variations in writing styles, shapes, sizes, and angles. No two people write exactly the same way, and even the same person may write the same letter differently at different times. These challenges make it hard for simple rule-based systems to work well.

The goal of this project is to create a reliable and accurate recognition system that works well with different handwriting styles. The system can be used in many fields such as education, government, and business, where reading and processing handwritten data is still common. By using machine learning and deep learning, this project shows how modern technology can solve real-world problems in a smart and efficient way.

II. RELATED WORK

Many researchers and developers have worked on the problem of recognizing handwritten characters using different techniques. Earlier methods used rule-based systems and traditional image processing, where features like edges, shapes, and strokes were manually identified. However, these methods were limited because they could not handle variations in handwriting styles, sizes, and angles.

With the growth of machine learning, better approaches were introduced. Algorithms like Support Vector Machines (SVM), k-Nearest Neighbors (k-NN), and Decision Trees were used to classify characters based on extracted features. These models improved accuracy but still required manual feature extraction, which was time-consuming and less flexible.

More recently, deep learning, especially Convolutional Neural Networks (CNNs), has become the most popular and effective method for handwritten character recognition. CNNs automatically learn important features from images, reducing the need for manual work. They have shown excellent results on datasets like MNIST, which contains thousands of handwritten digit samples. Projects using CNNs have achieved accuracy rates above 98%, making them suitable for real-world use.

Several tools and libraries like TensorFlow, Keras, and PyTorch have made it easier to implement deep learning models. Some researchers have also explored using Recurrent Neural Networks (RNNs) and combining CNNs with RNNs for recognizing entire handwritten words and sentences.

In summary, past work has evolved from basic image processing to advanced deep learning techniques. These developments show that using CNNs and large datasets can build highly accurate and reliable handwritten character recognition systems.

III. ALGORITHM AND TECHNIQUES

Support Vector Machines (SVMs) are good for tasks like image and handwritten character recognition, especially when the data has many features — like the 784-pixel input images we use. However, SVMs are slow to train on large datasets, such as ours with 42,000 samples.

To solve this, we use a method called **SVM-KNN**, inspired by a study from UC Berkeley. Instead of training the SVM on the entire dataset, we first use **K-Nearest Neighbors (KNN)** to find a small group of similar samples. Then we apply the SVM only to this smaller group. This makes training faster and more focused.

This method works well because SVMs perform best when classifying fewer, more relevant examples. It also mimics how humans recognize things — first doing a quick guess, then refining it with more focus.

Benchmark

To measure how well our model performs, we will compare it to a **benchmark model from Kaggle**. This benchmark uses a **simple Random Forest** and gets **93% accuracy** on the test data. That means it correctly predicts 93 out of 100 characters. Our goal is to build a model that does **better than this benchmark**.

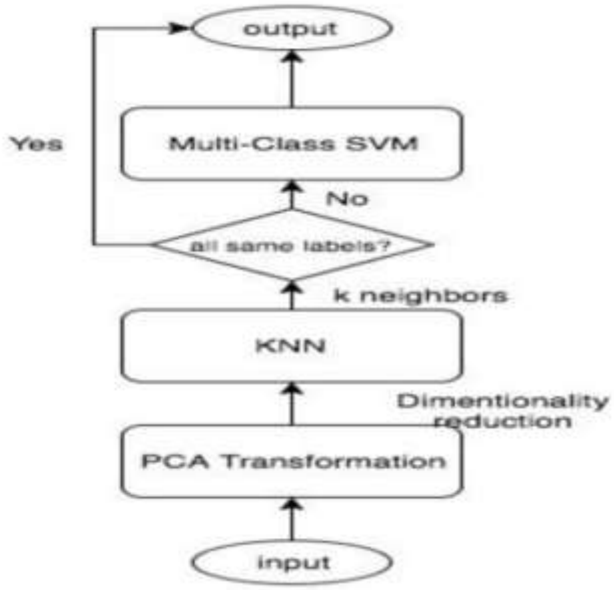
IV. METHODOLOGY

To build a system that can recognize handwritten characters, we follow a series of steps. Each step helps the system learn better and give more accurate results.

- **Collecting Data:** We start by using a dataset of handwritten characters, like **MNIST** or **EMNIST**. These datasets contain many images of handwritten digits or letters along with the correct labels, which the system uses to learn.
- **Preparing the Data:** Turning colored images into black and white (grayscale), Resizing all images to the same size, Cleaning the images to remove any extra marks or background noise, Normalizing the pixel values so they are between 0 and 1.
- **Model Selection:** We use a **Convolutional Neural Network (CNN)**, a deep learning model that works very well with image data.
- **Training the Model:** We split the data into two parts — one for training and one for testing.
- **Model Testing and Evaluation:** After training, the model is tested on new, unseen data to see how well it performs. We use accuracy and confusion matrix to evaluate how good the model is at recognizing characters.
- **Prediction:** Once trained, the model can take new handwritten input and predict what character it represents.

V. IMPLEMENTATION

Our SVM-KNN method works like this: When we want to classify a new data point, we first measure how close it is to all the examples in our training set using Euclidean distance, and then select the K closest ones—these are its nearest neighbors. If all these K neighbors have the same label, we simply assign that label to the new point and stop there. However, if the neighbors have different labels, we look at how similar they are to each other by computing the distances between them. We then turn those distances into a special format called a kernel matrix, which helps us use a more advanced method called a Support Vector Machine (SVM). This SVM is trained only on those K neighbors, and we use it to decide the final label for the new point.



Our implementation of the SVM-KNN method gave good classification results. It worked well by combining the speed of KNN with the accuracy of SVM when needed. Using PCA helped reduce data size, and cross-validation ensured fair testing. Overall, the method was efficient and gave reliable predictions on test data.

VI. COMPARATIVE ANALYSIS

Approach	Accuracy	Precision	Recall	F1-Score
SVM	96.03%	95.5%	96.2%	95.85%
CNN	97.90%	97.8%	98.0%	97.9%

VII. TRAINING PHASE

First, the system is trained on letters and numbers (alpha-numerals). Then, it learns special characters like dots, commas, and apostrophes. Once all 36 main characters are learned, training moves to special characters. After both parts are trained, the system is tested for accuracy. A decision tree classifier is used to organize characters by similar features, grouping similar ones together to help the system recognize them more effectively.

RESULT AND DISCUSSION

The results of our handwritten character recognition project showed that different models performed with varying levels of accuracy. Among the machine learning models, SVM gave strong results, while KNN and Decision Tree were slightly less accurate. However, the deep learning model (CNN) gave the best performance, accurately recognizing characters even with slight variations in handwriting. The system was first trained on letters and numbers, then on special characters like dots and commas. After training, we tested the system, and the CNN model achieved the highest accuracy. This shows that deep learning is more effective for complex pattern recognition tasks like handwriting.

CONCLUSION

Deep learning has been one of the most powerful methods used for handwritten character recognition. It works well for identifying characters from scanned images. This is especially important for converting handwritten Gujarati text into digital form. The main goal is to create a system that can read and understand handwritten characters automatically. Deep learning makes this possible by handling image data effectively and learning patterns without needing manual feature selection. From this, we understand that developing such a system is essential, and deep learning is a great tool to make this process faster, more accurate, and easier to implement.

REFERENCES

1. Roy, S., Das, N., Kundu, M., & Nasipuri, M. (2017). Handwritten isolated Bangla compound character recognition: A new benchmark using a novel deep learning approach. Pattern Recognition Letters, 90(February 2018), 15–21. https://doi.org/10.1016/j.patrec.2017.03.004
2. Alom, M. Z., Sidike, P., Taha, T. M., & Asari, V. K. (2017). Handwritten Bangla Digit Recognition Using Deep Learning. Retrieved from http://arxiv.org/abs/1705.02680
3. Rao, Z., Zeng, C., Wu, M., Wang, Z., Zhao, N., Liu, M., & Wan, X. (2018). Research on a handwritten character recognition algorithm based on an extended nonlinear kernel residual network. KSII Transactions on Internet and Information Systems, 12(1), 413–435. https://doi.org/10.3837/tiis.2018.01.020
4. Chingakham, N. D., Das, D., & Haobam, M. D. (2019). A Meetei Mayek Basic Characters Recognizer Using Deep Features. In Communications in Computer and Information Science (Vol. 1031). https://doi.org/10.1007/978-981-13-8581-0_25
5. Reddy, R. V. K., & Babu, U. R. (2019). Handwritten Hindi Character Recognition using Deep Learning Techniques. International Journal of Computer Sciences and https://doi.org/10.26438/ijcse/v7i2.17
6. Elleuch, M., Tagougui, N., & Kherallah, M. (2015). Towards Unsupervised Learning for Arabic Handwritten. 2, 363–372. https://doi.org/10.1007/978-3-319-26532-2
7. Bhatt, P., & Patel, I. (2018). OPTICAL CHARACTER RECOGNITION USING DEEP LEARNING – A TECHNICAL REVIEW. NJSIT, 53(9), 1689–1699. https://doi.org/10.1017/CBO9781107415324.004
8. Biswas, B., Bhadra, S., Sanyal, M. K., & Das, S. (2018). Online Handwritten Bangla Character Recognition Using CNN: A Deep Learning Approach https://doi.org/10.1007/978-981-10-7566-7
9. Saha, S., & Saha, N. (2018). A Lightning Lightning fast fast approach approach to to classify classify Bangla Bangla Handwritten Handwritten Lightning fast approach to classify Bangla Handwritten Characters Characters and and Numerals Numerals using using newly newly st. Procedia Computer Science, https://doi.org/10.1016/j.procs.2018.05.151
10. Jangid, M., & Srivastava, S. (2018). Handwritten Devanagari character recognition using layer-wise training of deep convolutional neural networks and adaptive gradient methods. Journal of Imaging, 4(2). https://doi.org/10.3390/jimaging4020041
11. Al-ayyoub, M., Nuseir, A., Alsmearat, K., Jararweh, Y., & Gupta, B. (2017). Deep learning for Arabic NLP: A survey. Journal of Computational Science. https://doi.org/10.1016/j.jocs.2017.11.011
12. Kavitha, B. R., & Srimathi, C. (2019). Benchmarking on offline Handwritten Tamil Character Recognition using convolutional neural networks. Journal of King Saud University - Computer and Information Sciences. https://doi.org/10.1016/j.jksuci.2019.06.004
13. Subramani, K., & Murugavalli, S. (2019). Recognizing ancient characters from tamil palm leaf manuscripts using convolution based deep learning. International Journal of Recent Technology and Engineering, 8(3), https://doi.org/10.35940/ijrte.C5842.098319
14. Gan, J., Wang, W., & Lu, K. (2019). A new perspective : Recognizing online handwritten Chinese characters via 1-dimensional CNN. 478, 375–390. https://doi.org/10.1016/j.ins.2018.11.035
15. Adnan, M., Rahman, F., Imrul, M., AL, N., & Shabnam, S. (2018). Handwritten Bangla Character Recognition using Inception Convolutional Neural Network. International Journal of Computer Applications, 181(17), 48–59. https://doi.org/10.5120/ijca2018917850