Handwritten Alphanumeric Character Recognition using Jetson Nano

Aruna Rao S. L
Department of Information Technology
BVRIT HYDERABAD College of Engineering for Women
Email: ithod@bvrithyderabad.edu.in

Jaina Sri Laxmi
Department of Information Technology
BVRIT HYDERABAD College of Engineering for Women
Email: 19wh1a1276@bvrithyderabad.edu.in

Kummari Tejaswini
Department of Information Technology
BVRIT HYDERABAD College of Engineering for Women
Email: 19wh1a12b7@bvrithyderabad.edu.in

Chatragadda Bharani
Department of Information Technology
BVRIT HYDERABAD College of Engineering for Women
Email: 19wh1a1265@bvrithyderabad.edu.in

Soppadandi Spoorthi
Department of Information Technology
BVRIT HYDERABAD College of Engineering for Women
Email: 19wh1a12b4@bvrithyderabad.edu.in

***Abstract*—Handwritten Alphanumeric Character Recognition is one of the significant areas of exploration and development with a streaming number of possibilities that could be attained. The applications of alphanumeric character recognition include postal correspondence sorting, bank cheque processing, form data entry, etc. In most of the applications, major challenge lies in copying the contents from original file where the content may be in a non-editable format. The heart of the project lies within the ability to develop an efficient algorithm that can recognize handwritten alphanumeric characters which are submitted by users, which may vary in their font styles and font sizes. In order to implement this, we used EMNIST Balanced Character Dataset to train the Machine Learning model using Deep Learning. Flask is used for API and user interface. The goal is to deploy whole system on Jetson Nano Developer Kit with an optimal solution and the best accuracy which is 87%.**

***Index Terms*—Handwritten Alphanumeric Character Recognition, Character Image Processing, Deep Learning, Artificial Intelligence, Jetson Nano Development Kit**

I. INTRODUCTION

Handwritten Alphanumeric Character Recognition is the process of recognizing handwritten characters and converting them into machine-readable format. The task of recognizing handwritten characters is a challenging task because of the variability in handwriting styles. The emergence of deep learning and computer vision technologies has made this task possible with high accuracy. In this paper, we present a system for Handwritten Alphanumeric Character Recognition using Jetson Nano.

Handwritten Alphanumeric Character Recognition is an important research area of pattern recognition, and it has been used widely in applications which include: license plate number, library call number recognition in the present digital world. To build a website deployed on Jetson Nano Development Kit for high GPU processing, allows the user to write the character/digit on the Graphical User Interface, which is used to recognize by the machine learning model and predicts the output at a better accuracy rate. The problems such as the recognition of handwritten characters, which may be written in different styles when recognized, can make the job of the human easier. Alphanumeric expression recognition using

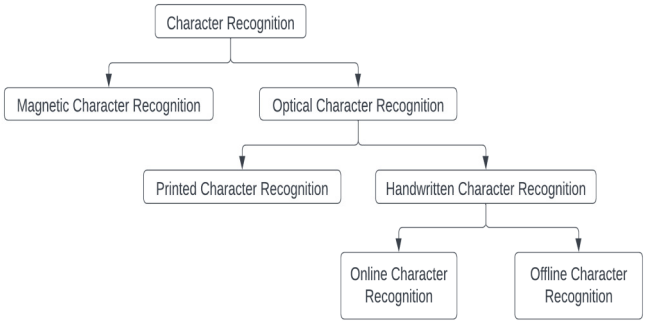

Fig. 1. Broad Classification of Recognition Systems

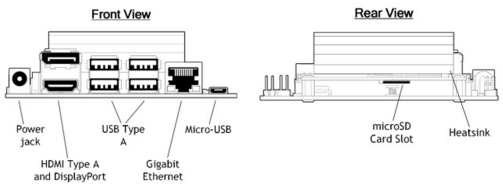

Fig. 2. Front and Rear view of Jetson Nano Development Kit

machines has become a subject of serious research.

Handwritten Digit Recognition is the capacity of a computer to interpret the manually written digits and alphabets from various sources like messages, bank cheques, papers, pictures, and so forth, and in several situations for web-based handwriting recognition on PC tablets, identifying number plates of vehicles, handling bank cheques, digits entered in any forms, etc.

When it comes to the case of working with the machine learning model or neural network model with images, it takes much time for loading, analyses, and training a model. Generally, personal computers do not support high processing and consumed more power. Instead of traditional methods to work with the computer GPU, we can work on embedded systems (IoT devices) that have the capability of high processing with low power consumption. Hence, we had chosen NVIDIA Jetson Nano Development Kit.

It is an all in an easy-to-use platform that runs in as little as 5 watts. We use Jetson Nano to run neural network models for key applications like Image Classification, Alphanumeric Character Recognition etc.

II. Related Work

A related work in the field of Handwritten Alphanumeric Character Recognition using Jetson Nano is discussed in this section.

[1] "Handwriting Numerals Recognition Using Convolutional Neural Network Implemented on NVIDIA's Jetson Nano"[1] by Huan Chen, Songyan Liu, Haining Zhang, Wang Cheng. Convolutional Neural Network (CNN) with RMSProp optimizer algorithm and Adam optimizer algorithm.The experiment is implemented on NVIDIA's Jetson Nano platform, where we compare the performance of CNN models with two different optimizer algorithms.The training accuracy is 98.25%.

[2] "Handwritten Text Recognition" by G. Louloudis et al. This survey paper provides an overview of various techniques and approaches in handwritten text recognition. It covers both online and offline recognition methods, including deep learning-based approaches.

[3] "Handwritten Character Recognition Using Recurrent Neural Networks" by Alex Graves et al. This paper explores the use of recurrent neural networks (RNNs) for handwritten character recognition. The authors introduced a novel architecture called the Long Short-Term Memory (LSTM) network, which has shown excellent performance in sequence modeling tasks.

[4] "IAM Handwriting Database" by U. Marti and H. Bunke : This database contains handwritten English text samples collected from various sources. It has been widely used for research in handwriting recognition, including the recognition of complete words or sentences.

[5] "Scrutinization of Urdu Handwritten Text Recognition with Machine Learning Approach"[2] by Dhuha Rashid Naveen Kumar Gondhi works on urdu .This paper is work done for Urdu handwritten text recognition using optical character recognition techniques. This research can also be used to compare the accuracy rates of various algorithms across multiple datasets

[6] Tanuja Kumari, Yatharth Vardan, Prashant Giridhar Shambharkar, Yash Gandhi , "Comparative Study on Handwritten Digit Recognition Classifier Using CNN and Machine Learning Algorithms"[3]. The objective of this paper is to use Convolutional Neural Networks (CNN), K-Nearest Neighbor, and Support Vector Machine to recognize isolated handwritten digits. After implementing and training the models on the same dataset and comparing the results obtained for three different models, the results show that CNN is the most optimal machine learning technique to classify handwritten digits with an accuracy of 99.59 percent.

[7] Handwritten Character Recognition Using Deep Learning (Convolutional Neural Network) 2023 by Sivakumar Venkataraman, Asherl Bwatiramba. The purpose of the research is to work with convolutional neural networks (CNNs) for Handwriting Recognition. It was found that a Convolutional Neural Networks combined with Simple Neural Network (SNN) helped in operating of well-tuned hardware with GPU. It's adequate training data can recognize numbers with an accuracy of up to 98.7 percent.

III. Proposed Methodology

The proposed methodology for Handwritten Character Recognition using Jetson Nano includes the steps mentioned below.

This can be broadly classified into three major categories, namely:

1. Jetson Nano Development Kit Setup
2. Model Development
3. Integraton and Deployment of the model on Jetson Nano

Jetson Nano Development Kit Setup: Jetson Nano is a single board computer with high performing GPU's developed by NVIDIA suitable in applications using Artificial Intelligence, Machine Learning, Robotics etc. will be used. Figure 1. shows the outline of Jetson Nano Developer Kit. The setup involves flashing Jetson Nano image on to the SD card through Balena Etcher. Figure 2. shows the flashing of Jetson Nano image on SD Card using Balena Etcher.

1) Data Acquisition:: Internet has abundant resources to gather datasets and there are multiple types of datasets. The dataset chosen for this usecase is the EMNIST Balanced Charater Dataset which consists of equal number of alphabets

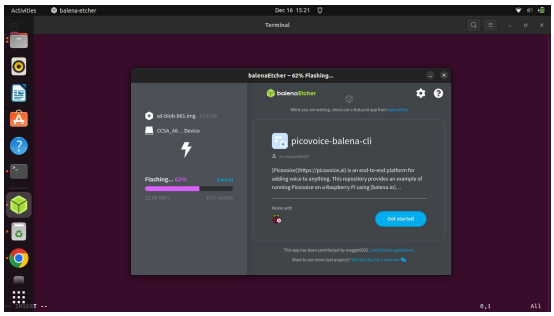

Fig. 3. Flashing into NVIDIA Drivers using Balena Etcher

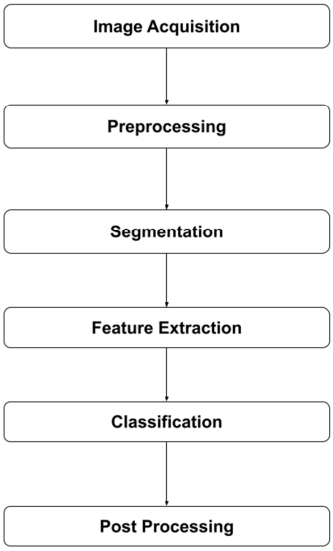

Fig. 4. Steps involved in training and build the model

and digits.

2) Preprocessing:: Preprocessing is a required stage, as it helps in improving accuracy in order to provide valid and better results. This involves; Data cleaning which is done to remove inconsistent data, outliers, null values etc from the dataset. It improves the accuracy of the data by reducing errors in the data. It also involves techniques such as data transformation, integration and reduction.

3) Feature Extraction:: Input data is transferred to a set of features from the pixels obtained from the characters. Dimensionality reduction is a part of feature extraction which ensures that the large data is reduced into the required form. In order to extract relevant features from the images, such as shape or size etc.

Model Development: Deep learning model Keras is used to develop the model. Keras is a Convolutional Neural Network API which is a part of TensorFlow. TensorFlow and Keras utilizing CUDA and cuDNN libraries, can leverage the computational power of GPUs to accelerate the training and inference processes of deep learning models to speed up deep learning models.

Training: The EMNIST Balanced Dataset is splitted into train and test datasets. During the training process, the model undergoes both forward and backward propagation iterations for the specified number of epochs. The forward propagation involves passing the training data through the model's layers, making predictions based on the current weights and biases. The predicted outputs are then compared to the actual labels using the specified loss function, such as the Sparse Categorical Cross Entropy.

After the forward propagation, the model performs backward propagation to compute the gradients of the loss with respect to the model's weights and biases. These gradients indicate the direction and magnitude of weight updates required to minimize the loss. The optimizer, in this case, the Adam optimizer, utilizes these gradients to update the model's parameters and adjust the weights accordingly. This process happens iteratively. The Optimizer and the algorithm used is ADAM.

Evaluation: Once the model has been trained, evaluation is performed. This evaluation involves applying the trained model to the test dataset. The test dataset contains examples that were not used during the model training process. Model Prediction is a part of the evaluation where the trained model is used to make predictions on the test dataset. The model takes the input examples from the test dataset and performs forward propagation to generate predictions or classifications. Various metrics are calculated to evaluate the model's performance alike accuracy, precision etc. These metrics provide quantitative measures of how well the model performs on the test dataset.

Integration and Deployment of the model on Jetson Nano: The trained model is then deployed on Jetson Nano Development Kit. Figure . shows the integration of trained model into jetson Nano.

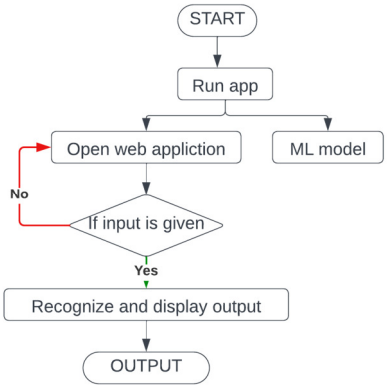

Fig. 5. Architecture of Handwritten Alphanumeric Character Recognition using Jetson Nano

IV. Experimental Results

The experimental results obtained from the study on ”Handwritten Alphanumeric Character Recognition using Jetson Nano” are presented and analyzed in this section.
The following are the results observed: Train the dataset, EMNIST Balanced Character Dataset. Set up the main route for the Flask Web Application. Now, Compile and Load the model which was trained previously. The trained model post integration with Jetson Nano Development Kit, which uses ADAM Algorithm provides the output as shown in the below graphs:

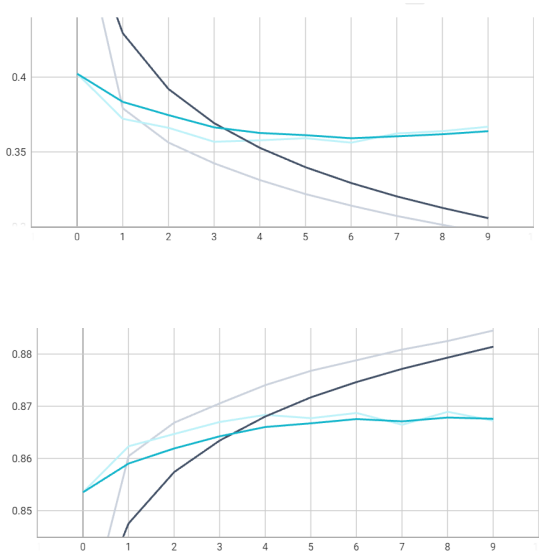

Fig. 6. Epoch values after training the model

The character is drawn on the editable space on the web application. Now, the following process happens:
A Flask route is defined which accepts HTTP POST requests. It retrieves the image data submitted from the client-side. Specifically, it gets the base64 encoded image data.
Then the encoded image is decoded using the ‘base64‘ and ‘numpy‘ libraries. It first splits the base64 encoded data string to extract the actual image data, which is then decoded using the ‘base64‘ library.
The input image is first converted to grayscale. Then, the image is resized to a 28x28 pixel image. After resizing, the image is normalized by dividing each pixel value by 255.0.

Finally, the preprocessed image is used to obtain the predicted class probabilities for each of the 62 classes (digits, uppercase letters, and lowercase letters). The predicted class probabilities are returned as a NumPy array.
Then the character recognition task is performed by predicting the character drawn on the canvas image. The predicted character and its probability are stored.
To do the prediction, the input image needs to be preprocessed first. This is done by converting the color image to grayscale. The grayscale image is then resized to (28,28) pixels, as this is the size expected by the EMNIST model.

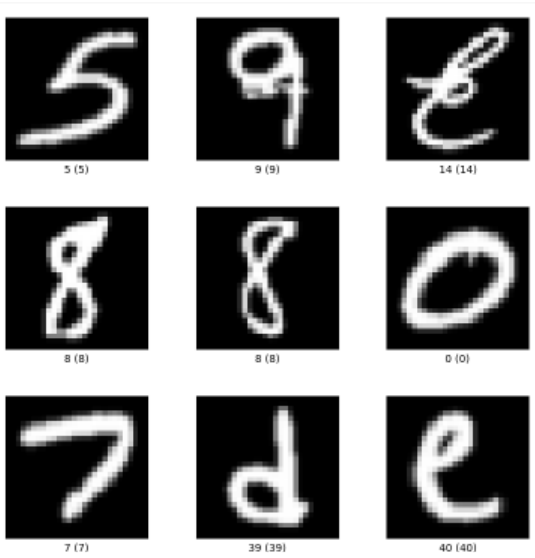

Fig. 7. Characters and digits in Trained Dataset

Next, the image is normalized by dividing each pixel value by 255.0 to scale it between 0 and 1. Finally, the image is fed as input to the EMNIST model.
The model predicts the probability of the digit and displays the closest and highest probability character on the screen and it’s probability.
Finally, the trained model generates a good accuracy which is 86.857%.

V. Conclusion

This paper achieved the recognition of Handwritten Alphanumeric Characters based on the concept of deployment of ML model on Jetson Nano. This project Alphanumeric Character Recognition is deployed on Jetson Nano which supports integration and working with the machine learning model using Deep Learning methodologies like using PyTorch framework, runs multiple neural networks parallelly help high-resolution sensors flask is for making API calls and user interface, to provide convenience for the users to convert the non-editable handwritten characters or digits into an editable format project can be used. This system involves three crucial steps - Preprocessing, Feature Extraction, and Classification. The project development follows four modules – NVIDIA Jetson Nano, Implementation of Machine Learning Algorithms, User Interface and Integration.
The Web Application is designed in such a way that the user will be provided an Interface where they can write a character or a digit using their free hand. The saved Machine Learning model is loaded again. It then classifies if the given input is a digit or an alphabet and predicts the output on recognizing the Handwritten Character as either an Alphabet or a numerical Digit. This method can be embedded into varied applications like library call numbers or license plates and can use in handwriting Character Recognition. So it has good practical value.

REFERENCES

1 S. Venkataraman and A. Bwatiramba, "Handwritten Character Recognition Using Deep Learning (Convolutional Neural Network)," in *Proceedings of the International Conference on Computer Vision*, February 2023.

2 D. Rashid and N. K. Gondhi, "Scrutinization of Urdu Handwritten Text Recognition with Machine Learning Approach," in *Emerging Technologies in Computer Engineering: Cognitive Computing and Intelligent IoT*, V.E. Balas et al. (eds), vol. 1591, Communications in Computer and Information Science, Springer, Cham, May 2022.

3 T. Kumari, Y. Vardan, P. G. Shambharkar, and Y. Gandhi, "Comparative Study on Handwritten Digit Recognition Classifier Using CNN and Machine Learning Algorithms," in *6th International Conference on Computing Methodologies and Communication (ICCMC)*, March 2022.

4 H. Chen, S. Liu, H. Zhang, and W. Cheng, "Handwriting Numerals Recognition Using Convolutional Neural Network Implemented on NVIDIA's Jetson Nano," in *Communications, Signal Processing, and Systems*, April 2020, pp. 529–535.

5 R. Dey, R. C. Balabantaray, and S. Mohant, "Sliding window based offline handwritten text recognition using edit distance," in *Multimedia Tools and Applications*, vol. 81, pp. 22761–22788, May 2021.

6 K. Sakthi Sudhan, S. Mohanraj, and T. V. P. Sundararajan, "A Survey of Handwritten Character Recognition with MNIST and EMNIST," Computer Science Department, Universidad Carlos III of Madrid, 28911 Leganes, Madrid, August 2019.

7 S. Ahlawat and A. Choudhary, "Hybrid CNN-SVM Classifier for Handwritten Digit Recognition," *Procedia Comput. Sci*, vol. 167, pp. 2554–2560, 2020.

8 B. Debnath et al., "Automatic Handwritten words on Touchscreen to Text file converter," in *IEEE Reg. 10 Annu. Int. Conf. Proceedings/TENCON*, vol. 2018-October, pp. 219–223, 2019.

9 A. Beltrán and S. Mendoza, "Efficient algorithm for real-time handwritten character recognition in mobile devices," in *2011 8th International Conference on Electrical Engineering, Computing Science and Automatic Control*, 2011, pp. 1–6, doi: 10.1109/ICEEE.2011.6106583.

10 H. A. Shiddieqy, T. Adiono, and I. Syafalni, "Mobile Client-Server Approach for Handwritten Digit Recognition," in *2019 International Symposium on Electronics and Smart Devices (ISESD)*, 2019, pp. 1–4, doi: 10.1109/ISESD.2019.8909448.

11 R. Vaidya, D. Trivedi, S. Satra, and P. M. Pimpale, "Handwritten Character Recognition Using Deep-Learning," in *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*, 2018, pp. 772–775, doi: 10.1109/ICICCT.2018.8473291.

12 H. Du, P. Li, H. Zhou, W. Gong, G. Luo, and P. Yang, "WordRecorder: Accurate Acoustic-based Handwriting Recognition Using Deep Learning," in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, 2018, pp. 1448–1456, doi: 10.1109/INFOCOM.2018.8486285.

13 D. Kavitha and P. Shamini, "Handwritten Document into Digitized Text Using Segmentation Algorithm," *An International Journal of Advanced Computer Technology*, 2016.

14 S. B. K. S., V. Bhat, and A. S. Krishnan, "SolveIt: An Application for Automated Recognition and Processing of Handwritten Mathematical Equations," in *2018 4th International Conference for Convergence in Technology (I2CT)*, 2018, pp. 1–8, doi: 10.1109/I2CT42659.2018.9058273.

15 T. Mantoro, A. M. Sobri, and W. Usino, "Optical Character Recognition (OCR) Performance in Server-Based Mobile Environment," in *2013 International Conference on Advanced Computer Science Applications and Technologies*, 2013, pp. 423–428, doi: 10.1109/ACSAT.2013.89.

16 V. V. Mainkar, J. A. Katkar, A. B. Upade, and P. R. Pednekar, "Handwritten Character Recognition to Obtain Editable Text," in *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)*, 2020, pp. 599–602, doi: 10.1109/ICESC48915.2020.9155786.