Handwritten Text to Digital Text Conversion using Various Deep Learning Models

Maheswari Pulgam
Computer Science
University of Central Missouri
Lee's Summit, United States of America
mxp43290@ucmo.edu

Sainath Reddy Kollu
Computer Science
University of Central Missouri
Lee's Summit, United States of America
sxk05230@ucmo.edu

Rishitha Gottimukkala
Computer Science
University of Central Missouri
Lee's Summit, United States of America
rxg26050@ucmo.edu

Rahul Sagar Myakala
Computer Science
University of Central Missouri
Lee's Summit, United States of America
rxm51460@ucmo.edu

Abstract- People believe that creating anything on paper is less clear than making it, however, everything is now digital. As a result, the utilization of HTR/HWR is increasing. There are various perspectives on handwriting. Character extraction, character recognition, and feature extraction are more conventional methods, but segmentation for recognition, machine learning, convolutional neural networks, and recurrent neural networks are more recent approaches. Applications for HTR/HWR include writer identification, postal address interpretation, online and offline identification, bank check processing, and signature verification.

The computer's ability to recognize and understand handwritten text pictures is called Handwritten Text Recognition (HTR), sometimes referred to as Handwriting Recognition (HWR). Handwritten text can be fed to the computer to forecast and transform into Computerized Text/Digital Text from a variety of sources, including notebooks, documents, forms, photos, and other devices.

Many applications exist for the HTR/HWR, and they are regarded as active research fields. They include Online recognition, Offline recognition, Signature verification, Postal address interpretation, Bank-Check processing, and Writer recognition.

Keywords; - CNN, RNN, SVM, Decision Tree, Logistic Regression, Random Forest.

I. Introduction

The field of handwritten document identification has garnered significant attention in recent years. It is always required to convert a piece of recorded material into electronic text for sharing or archiving instead of writing the data by hand. The suggested model converts a picture of handwritten text into digital text as input. The Convolutional Brain Organization (CNN) is used to focus on and characterize the components of comparative articles from several image tests. An extension of Repetitive Brain Organizations (RNN) with a longer memory is used since the text is successive information, which is stored in Long Transient Memory (LSTM). To handle the different placements of the text in the image, Connectionist Temporal Classification (CTC) loss is employed. Images of more than 100,000 words from the IAM Handwriting Database and handwriting samples from more than six hundred writers are used during training. The model enrolled 94% precision and a deficiency of 0.147 on preparation information and 85% exactness and a defect of 1.105 on approval information after training for various age groups.

Despite the availability of numerous electronic writing tools, a considerable number of individuals still like taking notes the traditional way—that is, on paper with pens. However, writing text by hand has drawbacks.

Looking at, storing, and exchanging tangible documents with others can be challenging. A substantial quantity of crucial data is because papers are never converted to digital format, they are either missed or lost. We have chosen to address this issue in our project because we think that users will be able to access, search for, share, and analyze their records more effectively while still maintaining their preferred writing style because digital text is much easier to manage than written text.

Two of the project's goals are to digitize handwritten materials and carry out a more in-depth investigation of the classification process. It was challenging to narrow down the project's scope because the word "handwritten text" has a somewhat broad definition, which required us to define it specifically for our needs. For this research, we had to categorize the visual representation of each handwritten word, whether it was written in block or cursive. This project can be merged with algorithms that separate words in a given line picture, and these algorithms can be linked with algorithms that separate lines in a given image of a handwritten page. With these additional layers, our project can become a deliverable that a user would use. This fully functional model aims to address the issue of digitizing handwritten documents by asking the user to take a picture of a page of notes. Even though our methodology requires a few more layers to produce a deliverable that is fully useful for the end user, we choose to concentrate on categorization since we think it to be the most fascinating and difficult aspect of the task.

II. MOTIVATION

Handwritten text is being turned into digital text due to growing digitization. Numerous problems are addressed by this conversion, including as document transfers, global accessibility, security concerns, and physical harm to the document.

Handwritten Recognition (HWR) or Handwritten Text Recognition (HTR) plays a key role in this transformation by detecting transcripts from information inspection photos and

https://github.com/Sainath1359/NN-and-DL-final-project.git

turning them into high-level designs. Various devices, such as notebooks, documents, forms, and photos, provide the input. Two categories of acknowledgment exist: one. Recognition in an Offline Format and 2. Recognition Online

2.1 OFFLINE RECOGNITION: The text from the photos is automatically transformed into digital text using this technology. The data employed in this method is said to be a static representation of handwriting. This approach gets difficult for different handwriting types.

2.1.1TRADITIONAL Procedures

• Character Extraction: Apply a filter to the image, and then take out the people who are in it. The problem with this approach is that it returns connected characters as a single sub-image that includes every related character, which makes recognition challenging.

• Identifying Personalities: Once each character has been recovered, the computer generates the appropriate digital character.

• Derivation of Features: At this stage, the programmer needs to determine which properties would seem to be important. It will take longer to solve this problem than it would for a brain organization.

2.1.2 MODERN Strategies

- Maheswari Pulgam has contributed 20% of the

  Multi-Class Logistic Regression
  Support Vector Machine
  Decision Tree
  Random Forest

- Rishitha has contributed 30% of the work completed.

  Using Keras's MNIST Dataset
  Neural Networks
  Convolution Neural Networks
  Recurrent Neural Networks

- Sainath and Rahul have contributed 30% of the work completed.

  Using Extra Keras's EMNIST Dataset
  Multinomial Logistic Regression
  Decision Tree
  Random Forest
  Neural Networks

https://github.com/Sainath1359/NN-and-DL-final-project.git

III. MAIN CONTRIBUTIONS & OBJECTIVES

3.1 MAIN CONTRIBUTIONS

• While the conventional procedures include the division of characters, the current procedures include the division of lines.
• Focused on a recognizable AI methodology.
• Recurrent neural networks are used for several windows of the text image that overlap; convolutional neural networks are used for feature extraction.

2.2 ONLINE RECOGNITION

Text is automatically converted using this method as it is written. In this instance, sensors are used to detect the movements of the pen tip. It is claimed that this technique generates data that is a digital handwritten representation. This approach converts the impulses into matching digital characters. A writing pen, a touch screen that is integrated with the output display, and an application that recognizes and translates pen-tip movements into digital text are the components of the online handwriting recognition interface.

This method involves the following steps:

1. Classification
2. Feature extraction
3. pre-processing

OBJECTIVES:

- Finding handwritten words, lines, paragraphs, and other text in online documents is the main objective. There have been numerous evaluations and a great deal of work about handwriting recognition.
- Character Extraction: The process entails scanning the image and then extracting each individual character.
- Character Recognition: The computer outputs the corresponding digital character after extracting each individual character.

- Feature Extraction: The programmer has to identify the properties that can seem significant in this step.

IV. Related Works

For an exceptionally long time, people have recorded their thoughts in letters, transcripts, and other formats to share them with others. Nevertheless, the advent of computers quickly led to the replacement of handwritten text with digital writing generated by computers. People perceive a need for a system that can convert handwritten writing into digital text since it makes processing such data quick and simple. Several explorers have previously attempted to advance this kind of framework. Nevertheless, there is still a vital need for additional research in this field. The handwritten characters of the most extensively used languages in the world, such as English, Chinese, and Indian scripts including Devanagari, Malayalam, and Bangla, have been the subject of several offline and online recognition research [2–12]. All of these studies do, however, have certain shortcomings, such as low precision, a higher rate of false positives, and a slow change pace. compared the performance of several classifiers for the recognition of handwritten digits. 4]. According to a few character recognition investigations, gradient and curvature properties are the most accurate for handwritten character recognition tasks [13]. In order to examine the wavelet transformations of the input character image for handwritten Devanagari and Bangla character identification, a three-layer technique was utilized in a recent work [16]. A Hidden Markov Model-based system for handwritten character recognition in English was proposed by Rajib et al. [17]. The two different feature extraction techniques used in this method were global and local feature extraction. Global element has numerous highlights in the amounts of four, six, and four independently. These highlights include projection highlights, angle highlights, and form highlights. Even so, neighboring elements are identified by dividing the sample image into nine identical blocks. Four feature vectors are used to calculate the gradient feature for each block, resulting in a total of thirty-six local features. This produced fly features (local and global) for each sample image. These features are then supplied into the HMM model to train it. This method also uses information post-handling to reduce the cross-classification of different classes. With this method, training and feature extraction are time-consuming. Furthermore, in response to such inputs, it performs poorly when several characters are concatenated into a single image. Velappa Ganapathy et al. [18] proposed a recognition strategy based on multi-scale neural network training. To improve accuracy, this approach used a customizable threshold that was calculated using the minimum distance methodology. This method also requires building a graphical user interface (GUI) that can recognize characters in the scanned image. This method gives 85% accuracy with a medium training level. Since this method made use of images with a high resolution (20 28 pixels), it required less training time.T. Som et al. [19] used the fuzzy membership function to improve the handwritten text recognition system's accuracy. This method uses a fuzzy approach for each class after normalizing text pictures to 20×10 pixels. To identify the text's vertical and horizontal projection, a bonding box is formed around the character. The image is resized to 10×10 pixels after it has been cropped to a bounding box. Next, with the aid of the thinning operation, cropped photos are thinned. All these previously processed images are arranged one after the other into a single matrix to generate the test matrix. The user's presentation of new (test) photos is checked to see if it matches the test matrix. Although the approach was quick, its accuracy was poor. Rakesh Kumar et al. [20] presented a technique that uses a single layer neural network to shorten the system's training time. Characters that are segmented are resized to eighty-by-eighty pixels. To enhance training performance, data normalization is applied to the input matrices. Yet, the accuracy rate of their result is low an enhanced version of this work, feature extraction utilizing the diagonal approach [21], is another noteworthy piece of work by Zamora [22]. The others extracted hybrid features from the text using a zone-based approach. Accuracy and speed increased as a result. Utilizing the Euler number Approach, velocity, and precision are enhanced. Preprocessing techniques such as thresholding, thinning, and filtering are applied to the input image to reduce the cross-error rate. Three methods are applied to improve the segmentation. The input image is scaled to 90 X 60 pixels after segmentation. Following the calculation of each text's Euler number, the texts are split into 54 zones, each of which has 10×10 pixels. The character's feature vector is derived from the mean value of each zone, both in terms of rows and columns. Their concept, which is based on the heuristic segmentation method, was put up by Anshul Mehta at el [23]. Their approach finds valid segmentation points between handwritten letters with good accuracy. Fourier descriptors are used in this method of feature extraction. The discrete Fourier coefficients of the input picture (a[k] and b[k]) are found after a successful segmentation. Here, k is a number between zero and (L-1) and L denotes the border points of the input image. Fifty-two characters in total (26 upper case and twenty-six lower case English letters) were attempted to be categorized using this method. Additionally, it offers a comparison of several classification techniques. [24] put out a cutting-edge interactive technique for handwritten character recognition. Human assistance is only required for those inputs that confuse the system. It raises human lead even with excellent precision maintained. The sole problem was that human interaction was required because the technology did not function fully automatically. A wearable input system that let users change the text painted in the sir was proposed by Amma and colleagues [25]. It was a handwriting acknowledgment method using 3D reconciliation. With the use of motion sensors, accelerometers, and gyroscopes positioned precisely behind the human hand, the handwriting motions were wirelessly recorded. The method made sense, but written data could not be applied in the same way. This article's feature vector is a bit map representation of the sample input picture. In any recognition system, choosing the best feature vector is an essential first step. By using a limited number of features that are effective in distinguishing between different pattern classes, the proposed feature extraction method aims to support precise pattern classification. The parent image's bit map version preserves every important feature in a small neighborhood. The planned includes the inquiry as well. The proposed additionally includes an examination of the modifications made to the framework because of different learning methods. It also shows how various parameter selections affect a number of variables, such as the number and size of hidden layers and epochs. Preprocessing steps in the recommended technique include character division, commotion evacuation, normalization, and de-slanting. Up to 95% of English characters can be accurately detected by this study. The recommended framework's confirmed usability, convenience of use, and high rate of acceptance suggest that it might be incredibly helpful for real-world application.

III PROPOSED FRAMEWORK

3.1.1 MULTI – CLASS LOGISTIC REGRESSION

The classification of logistic regression is generalized to multiclass, or with more than binary outcomes, by multiclass logistic regression and multinomial logistic regression.

This kind of machine learning model forecasts the likelihood of certain outcomes.

Other names of this model:

- polytomous LR
- multiclass LR
- softmax regression
- multinomial logit
- maximum entropy
- conditional maximum entropy

- Importing the digits dataset from sklearn
- Loading the digits dataset to a data frame
- Importing train test split method from sklearn
- Splitting the total dataset into train and test datasets
- Creating a model for Logistic Regression
- Training the model
- Checking the score of model
- Predicting some of the test data
- Comparing the predicted output to actual output

1.1.2 SUPPORT VECTOR MACHINE

Encouragement The vector machine is a popular supervised method for solving regression and classification problems. Classification problems are the principal use of machine learning. The SVM algorithm creates the ideal line. or an n-dimensional spatial decision boundary to split classes. It will therefore be easy to add more data points in the future to the relevant category. Here, we have a hyperplane, which is a best option boundary. It helps create the hyperplane by choosing the extreme vectors or points. The extreme vectors are the support vectors. Thus, this technique is known as a

https://github.com/Sainath1359/NN-and-DL-final-project.git

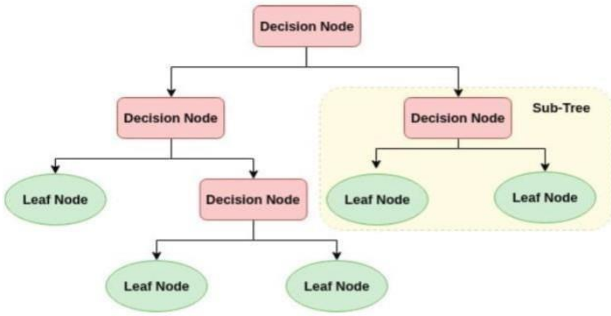
"Support Vector Machine." Text categorization, photo classification, and face identification are all done with this method.

- Linear SVM: We use data that is linearly separable in this. If two classes can be distinguished from the dataset by a single straight line, The term for it is "linearly separable data." The linear SVM classifier is the name of the classifier.
- Non-linear SVM: We employ non-linearly separable data in this. Non-linearly separable data is defined as datasets that cannot be divided into two groups by a single straight line. A Non-linear SVM classifier is the name given to the classifier.

- Importing digits dataset from sklearn
- Loading the digits dataset to a data frame
- Importing train test split method from sklearn
- Importing SVC model from sklearn
- Data Preprocessing
- Splitting the total dataset into train and test dataset
- Creating a model for SVM
- Training the model
- Checking the score of model
- Predicting some of the test data
- Comparing the predicted output to actual output

3.1.3 DECISION TREE

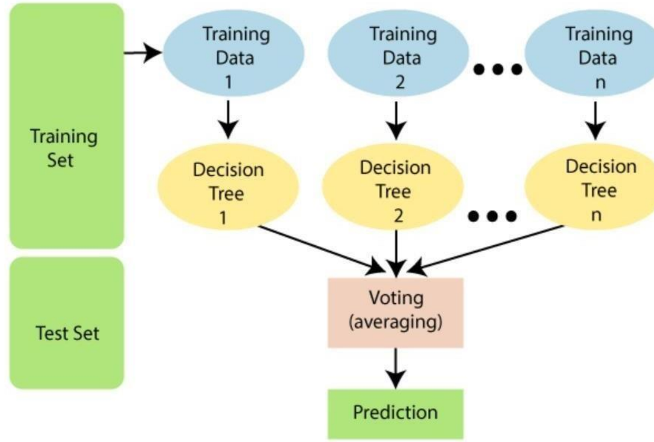
Decision trees are used to solve problems with classification and regression. It has the accuracy to handle multi-dimensional data. Most popular classification techniques are quite easy to understand and apply. A decision tree is a kind of tree structure that looks like a flowchart, with each edge denoting a choice and each node representing a feature. There are two steps in the categorizing process: learning and prediction. In the learning stage, the model is built using the given training data, and in the prediction step, it is used to predict the result for the given data.



- Importing required libraries
- Importing the digits dataset from sklearn
- Loading the digits dataset to a data frame
- Data Preprocessing
- Importing train test split method from sklearn
- Splitting the total dataset into train and test dataset
- Importing Tree for Decision Tree Classifier from sklearn
- Creating a model for Decision Tree Classifier
- Training the model
- Checking the score of model
- Predicting some of the test data
- Comparing the predicted output to actual output

3.1.4 RANDOM FOREST

The Random Forest Algorithm is one of the machine learning algorithms used in the supervised learning approach. In machine learning, this method is used to resolve problems with regression and classification. This is predicated on an approach that combines numerous classifiers to address a complicated issue and improve the model's performance. To improve the dataset's prediction accuracy, the Random Forest classifier selects the average of several decision trees applied to different dataset subsets. Without relying on a single decision tree, it makes predictions about the output based on the majority votes of each tree's projections. Accuracy grows together with the number of trees in the forest, preventing the overfitting issue.



- Importing the digits dataset from sklearn
- Loading the digits dataset to a data frame
- Data preprocessing
- Importing train test split method from sklearn
- Splitting the total dataset into train and test dataset
- Importing Linear model for Logistic Regression from sklearn
- Creating a model for Logistic Regression
- Training the model
- Checking the score of model
- Predicting some of the test data
- Comparing the predicted output to actual output

3.2 USING KERAS'S MNIST DATASET

3.2.1 NEURAL NETWORKS

Neural Networks: Biological neural networks serve as an inspiration for computing systems, which use them to perform a wide range of tasks on massive amounts of data. This is known as artificial neural networks. Many algorithms are used to understand the relationships in the dataset to optimize the results by modifying the inputs. Neural networks are trained to produce the necessary outcomes. To predict future results, various models are used to the data. Every node is connected to every other node via It can operate similarly to the human brain because the nodes are connected to one another. Correlations and hidden patterns help classify and cluster data.

ARCHITECTURE OF A NEURAL NETWORK:

https://github.com/Sainath1359/NN-and-DL-final-project.git

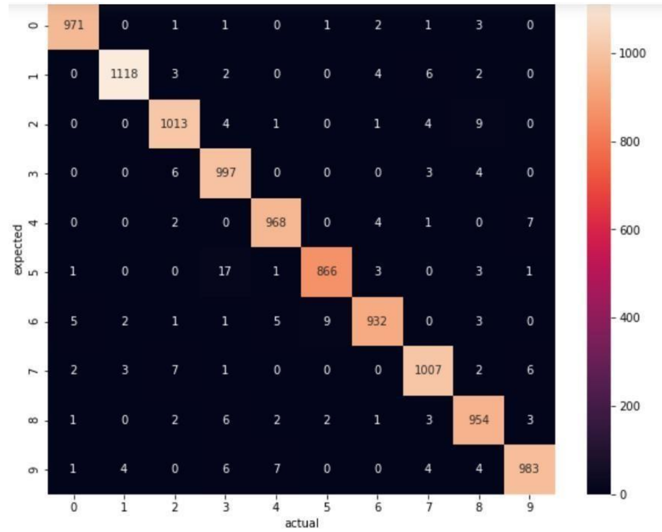
In Neural Network there are three types of architectures.

1.Single-Layer Feedforward Network: An input layer of source codes is projected onto a layer of neurons in the output. We call this type of network an acyclic or feedforward network. It refers to the computation neurons in the output layer. It is referred to be a single layer as a result. There is no computation done on the input layer. As a result, the total does not include it.

2.Multi-Layer Feedforward Network: This network has one or more hidden layers in addition to the input and output layers. The nodes in this layer are also known as hidden units or hidden neurons. The hidden layer is introduced in between the output and the external input. Fixed input layer nodes supply the input signal to the hidden layer, and the output from the next layer receives information from the buried layer, and so on, for the duration of the surviving network.

3.Recurrent Networks: The feedforward network and this network are comparable. The primary distinction is that it has one or more feedback loops.

- Importing necessary Libraries
- Loading mnist dataset
- Data preprocessing
- creating Neural Network Model
- Adding Layers
- Compiling Model
- Training Model
- Evaluating model for test dataset
- Importing necessary libraries
- Predicting output for few elements
- Comparing predict output to actual output.
- Creating a confusion matrix for model analysis
- Pictorial representation of confusion matrix for better understanding



3.2.2 CONVOLUTION NEURAL NETWORKS

This feed-forward artificial neural network is patterned like the visual cortex in terms of how its neurons link to one another. This network processes data using multiple layers of arrays. For faces and images, this type of neural network is employed recognition. CNN takes the two-dimensional array as input and processes the images without focusing on feature extraction. Three key ideas underpin this network: pooling, convolution, and local respective fields.

- Importing necessary libraries
- Loading mnist dataset
- Data preprocessing
- Creating Convolution Neural Network Model
- Adding Layers
- Compiling the model
- Training the model
- Predicting few elements of test dataset
- Comparing predicted output to the actual output

3.2.3 RECURRENT NEURAL NETWORKS

A type of artificial intelligence known as a recurrent neural network recognizes patterns in data sequences, including spoken words, handwriting, and text. For model training, the back propagation technique is employed. Since back propagation occurs for each timestamp, it is referred to as back propagation over time. Long Short-Term Memory Networks (LSTMs): These unique neural networks are mostly employed in sequence prediction issues to learn long-term dependencies. With the exception of individual data points (pictures), it can process the complete data sequence thanks to a feedback connection.

https://github.com/Sainath1359/NN-and-DL-final-project.git

- Importing required libraries
- Loading the mnist dataset
- Data preprocessing
- Creating Recurrent Neural Network Model
- Adding Layers
- Compiling the model
- Training the model
- Predicting few elements from the test dataset
- Comparing the predicted output to the actual output

3.3 USING EXTRA KERAS'S EMINIST DATASET

3.3.1 MULTINOMIAL LOGISTIC REGRESSION

- Importing necessary libraries
- Import logistic regression from sklearn linear model.
- Loading emnist dataset
- Data pre-processing
- Creating a model for logistic regression
- Training the model
- Checking the score of the model

3.3.2 DECISION TREE

- Importing necessary libraries
- Importing Tree for Decision Tree Classifier from sklearn
- Loading emnist dataset
- Data Pre-processing
- Creating a model for Decision Tree Classifier
- Training the model
- Checking the score of a model

3.3.3 RANDOM FOREST

- Importing necessary libraries
- Importing Random Forest Classifier from sklearn
- Loading emnist dataset
- data preprocessing
- Creating a model for Random Forest Classifier
- Training a model
- Checking the score of a model

3.3.4 NEURAL NETWORKS

- Importing necessary libraries
- Loading emnist dataset
- Data Preprocessing
- Creating Neural Network Model
- Adding Layers
- compiling model
- Training Model
- Evaluating the accuracy of the model
- Displaying the input images
- Creating a list to map the predicted output to the corresponding digitalcharacter.
- Predicting few elements from the test dataset
- Comparing the predicted output with the actual images
- CONVOLUTION NEURAL NETWORKS
- Importing necessary libraries
- Loading emnist dataset
- Data Preprocessing
- Creating Convolution Neural Network Model
- Adding layers
- Compiling the model
- Training the model
- Evaluating the model for test cases
- Displaying the input images
- Creating a list to map the predicted output to the corresponding digitalcharacter.
- Predicting few elements from the test dataset
- Comparing the predicted output to the actual outputs

https://github.com/Sainath1359/NN-and-DL-final-project.git

3.3.5 RECURRNT NEURAL NETWORKS

- importing necessary libraries
- Loading emnist dataset
- Data Preprocessing
- Creating a Recurrent Neural Network Model
- Adding layers
- Compiling the model
- Training the model
- Evaluating the model for the test cases
- Displaying the input images
- Creating a list to map the predicted output to the corresponding digitalcharacter.
- Predicting few elements from the test dataset
- Comparing the predicted output to the actual output

3.4 USING A SAMPLE DATASET OF HANDWRITTEN WORDS

3.4.1 CONVOLUTION NEURAL NETWORKS

- Importing required libraries
- Loading the dataset to data frame
- Data Preprocessing
- Creating Convolution Neural Network
- Adding Layer
- Compiling the model
- Training the model
- Data Preprocessing
- Predicting the given image
- Data Preprocessing
- Predicting the given image

V. Methodology

A. Datasets

1. SCIKIT LEARN'S DIGIT DATASET

To load the dataset, we use:

```
sklearn.datasets.load_digits(*, n_class=10, return_X_y=False, as_frame=False)
```

The dataset contains 8x8 image of a digit.

Classes	10
Samples perClass	~ 180
Total Samples	1797
Dimensionality	64
Features	integers 0 – 16

Keras MNIST Dataset

To load the dataset, we use:

```
tf.keras.datasets.mnist.load_data(path="mnist.npz")
```

The dataset contains 28x28 image of a digit.

Classes	10

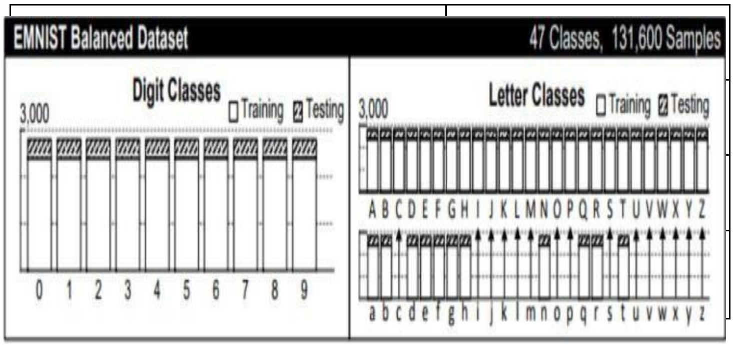
https://github.com/Sainath1359/NN-and-DL-final-project.git

x_train	(60000, 28, 28)
y_train	(60000,)
x_test	(10000, 28, 28)
y_test	(1000,)

Extra Keras MNIST Dataset

To load the dataset, we use:

```
from extra-keras-datasets import emnist
(input_train, target_train), (input_test, target_test) = emnist.load_data(type='balanced')
```



B. Data Preprocessing

Handwritten picture datasets require preprocessing in order to be used in machine learning. To make the dataset easier to utilize for the machine learning algorithm, it should be cleaned and arranged. Eliminating any erroneous data points is the first step. Errors in the data collection process or inaccurate data entry can result in invalid data points. Outliers in the data collection might potentially result in invalid data points. Data points that deviate significantly from the rest of the collection are called outliers. If they are not eliminated, they may skew the machine learning algorithm's findings.

The quality of the raw data may be worse than the quality of the final forecast due to missing values and/or noisy data. To make the data more appropriate for mining and analysis of the three categories of smoking behaviors, preprocessing is therefore necessary. This covers data discretization, feature selection, and redundant value reduction. When it comes to BMI, a considerable proportion of people (25%) are obese, whereas 18% are overweight. The ranking score in the balanced data provided by the selected feature relevance technique.

C. explains the significance of BMI in addition. The dataset originally contained 201 Body Mass Index (BMI) feature values that were missing. These figures were filled in using the mean BMI for the entire dataset. Furthermore, it was discovered that over 30% of the population did not smoke; this finding could be explained by either missing data or a lack of information regarding feature values. Owing to the sheer amount of data, it was determined to reclassify those individuals based on a few presumptions so as to avoid omissions. Since people under the age of eighteen are less likely to smoke now than they were in the past, the Unknown values for such individuals were changed to never. Consequently, the dataset contained 909 fewer ok unknowns than it did 1544 times earlier. The values for each employment type were reclassified as well, going from "children" to "never worked." This is because youngsters may have aspirations of "never working" and because they should not have been considered a labor type in the first place.

D. Data Preparation

Norming the data is the second phase. Converting each data point to the same unit of measurement is known as standardizing the data. This is crucial because it guarantees that the comparisons made by the machine learning algorithm are fair. Combining the data sets is the third stage. If the data collection is separated into several files, this is required. Labeling the data is the

https://github.com/Sainath1359/NN-and-DL-final-project.git

fourth stage. If the data set is not already tagged, this is required. Giving each data point a name is the process of labeling the data. Eliminating any duplicate data points is the sixth step. The machine learning algorithm's output may be distorted by duplicate data points.

Dividing the data into training and testing sets is the sixth stage. The machine learning algorithm is trained using the training set. The machine-learning algorithm's accuracy is evaluated using the testing set. Formatting the data is the eighth stage. If the data is not in a format that the machine learning algorithm can use, this is required. Filtering the data is the seventh stage. If the data set is too big to use with the machine learning technique, then this is required. The data must be normalized as the ninth step. To normalize data is to make adjustments such that the standard deviation is one and the mean is zero. This is crucial because it guarantees that the comparisons made by the machine learning algorithm are fair.

Selecting the machine learning algorithm is the tenth stage. The algorithm that will be used to extract knowledge from the data set is known as the machine learning algorithm. Selecting the machine learning algorithm's parameters is the eleventh stage. The settings that the machine learning algorithm will employ to draw conclusions from the data set are known as the parameters. Executing the machine learning algorithm is the twelfth stage. In this stage, the data set is subjected to the machine learning algorithm. The machine learning algorithm's output is assessed in the twelfth phase. This is the stage where the machine-learning algorithm's accuracy is assessed. If necessary, the machine learning algorithm is modified at the fourteenth stage. In this step, the evaluation's findings are used to modify the machine learning algorithm. Until the machine learning algorithm achieves the required accuracy, the fifteenth step entails repeating the previous fourteen and sixth phases.

VI. Results

https://github.com/Sainath1359/NN-and-DL-final-project.git

Dataset	Model	Description	Accuracy
Sklearn's Digit	Multinomial Logistic Regression	Default parameters	97.55 %
	Support Vector Machine	C=1.1	99.16 %
	Decision Tree	Default parameters	100 %
	Random Forest	n_estimators=35 criterion=entropy	98.6 %
	Neural Networks	Layer 1: type=dense,activation=relu Layer 2: type=Dense, activation=sigmoid Optimizer=adam Loss=sparse_categorical_cross entropy	98.08%

		Epochs=7	
		Layer 1: type=Conv2D, activation=relu Layer 2: type=MaxPooling2D	

https://github.com/Sainath1359/NN-and-DL-final-project.git

Keras's MNIST	ConvolutionNeural Networks	Layer 3: type=Dropout Layer 4: type=Conv2D, activation=relu Layer 5: type=MaxPooling2D Layer 6: type=Dropout Layer 7: type=Flatten Layer 8: type=Dense, activation=sigmoid Layer 9: type=Dense, activation=softmax Optimizer=adam	99.27 %
		Loss=sparse_categorical_cross entropy Epochs=10	
		Layer 1: type=LSTM, activation=relu Layer 2: type=Dropout	

https://github.com/Sainath1359/NN-and-DL-final-project.git

		Layer 3: type=LSTM, activation=sigmoid Layer 4: type=Dense, activation=relu Layer 5: type=Dropout	
	RecurrentNeural Networks	Layer 6: type=Dense, activation=softmax Optimizer=Adam, Learning rate=1e-3, Decay=1e-5 Loss=sparse_categorical_cross entropy Epochs=5	98.18%
	MultinomialLogistic Regression	Default Parameters	69.42%
	Decision Tree	Default Parameters	58.90%
	Random Forest	Default Parameters	80.92%
	Neural Networks	Layer 1: type=Flatten Layer 2: type=Dense,activation=relu	83.35%

https://github.com/Sainath1359/NN-and-DL-final-project.git

		Layer 3: type=Dense, activation=softmax Optimizer=adam Loss=sparse_categorical_cross entropy Epochs=10	
Extra Keras's EMNIST	Convolution Neural Networks	Layer 1: type=Conv2D Layer 2: type=MaxPooling2D Layer 3: type=Dropout Layer 4: type=Flatten Layer 5: type=Dense, activation=relu Layer 6: type=Dense, activation=softmax Optimizer=adam Loss=sparse_categorical_entropy Epochs=10	85.07%

https://github.com/Sainath1359/NN-and-DL-final-project.git

CONCLUSION

A suggested method for converting handwritten text to digital text has been developed and evaluated. There has been a comparison with relevant works made available.

This technique can be expanded to identify and transform words through character extraction.

By splitting words and removing the letters, it can also be used to identify and transform phrases.

REFERENCES

[1]. 1Ahmed Mahi Obaid, 2Hazem M. EI Bakery, 3M.A. Eldosuky, 4A.I. Shehab, "Handwritten Text Recognition System Based on Neural Network",International Journal of Advanced Research in Computer Science & Technology (IJARCST), Vol. 4, Issue 1, 2016

[2]. 1Ayush Purohit, 2Shardul Singh Chauhan, "A Literature Survey on Handwritten Character Recognition", International Journal of Computer Science and Information Technologies (IJCSIT), Vol. 7, Issue 1, 2016

[3]. 1Karthik Dutta, 2Praveen Krishnan, 3Mahesh Mathew, 4C.V. Jawahar,

"Improving CNN-RNN Hybrid Networks for Handwriting Recognition," CVIT

[4]. 1Batuhan Balci, 2Dan Saadati, 3Dan Shiferaw, "Handwritten TextRecognition using Deep Learning", Stanford

[5]. 1Yi-chao Wu, 2Fei Yin, 3Zhuo Chen, 4Cheng-Lin Liu, "Handwritten Chinese Text Recognition Using Separable Multi-Dimensional Recurrent Neural Network", International Conference on Document Analysis and Recognition, 2017

https://github.com/Sainath1359/NN-and-DL-final-project.git