Handwritten Text to Digital Text Conversion using Various Deep Learning Models

Upender Reddy Chitla
Computer Science
University of Central Missouri
Lee's Summit, United States of America
urc00520@ucmo.edu

Haripriya Eddala
Computer Science
University of Central Missouri
Lee's Summit, United States of America
Hxe61360@ucmo.edu

Sahithi Gunda
Computer Science
University of Central Missouri
Lee's Summit, United States of America
Sxg55600@ucmo.edu

Abstract-*Everything is digital these days, but many still think that making anything on paper is less clear than creating it. Consequently, there is a growing use of HTR/HWR. Diverse viewpoints exist on handwriting. While segmentation for recognition, machine learning, convolutional neural networks, and recurrent neural networks are more recent techniques, character extraction, character recognition, and feature extraction are more traditional methods. Writer identification, postal address interpretation, online and offline identification, bank check processing, and signature verification are among the applications for HTR/HWR.*

Handwritten Text Recognition (HTR), also known as Handwriting Recognition (HWR), is the computer's capacity to identify and comprehend handwritten text images. Handwritten text from a range of sources, such as notebooks, documents, forms, pictures, and other devices, can be input into the computer to forecast and convert into Computerized Text/Digital Text.

The HTR/HWR has many applications and is thought to be an active research area. These consist of writer recognition, postal address interpretation, online and offline recognition, postmark verification, bank check processing, and signature verification.

Keywords: Random Forest, CNN, RNN, SVM, Decision Tree, and Logistic Regression.

I. Introduction

In the past few years, handwritten document identification has attracted a lot of interest. Whenever any recorded material is to be shared or archived, it must always be converted into electronic text rather than being written by hand. The proposed approach uses an image of handwritten text as input and transforms it into digital text. The components of comparative articles from various image tests are focused on and characterized using the Convolutional Brain Organization (CNN). Since the text contains sequential information that is kept in long-term memory, an extension of repetitive brain organizations (RNN) with a longer memory is employed (LSTM). Connectionist Temporal Classification (CTC) loss is used to handle the text's various locations inside the image.Over 100,000 word images from the IAM Handwriting . Database and handwriting samples from more of than six hundred writers are being used during training. After training for various age groups, the model enrolled 94% precision and a deficiency of 0.147 on preparation information and 85% exactness and a defect of 1.105 on permission information.Even with all of the electronic writing tools available, a sizable portion of people still prefer to take notes the old-fashioned manner, which is on paper with pens. Writing text by hand has disadvantages, though.

It can be difficult to see, store, and share physical papers with other people. Papers are either missing or lost, and a significant amount of important data is lost since they are never converted to digital format. Because we believe that users will be able to access, search for, share, and analyze their records more effectively, we have decided to solve this issue in our project.

Digitization of handwritten materials and a deeper analysis of the classification process are two of the project's objectives. Because the term "handwritten text" has a somewhat broad connotation and we had to define it precisely for our purposes, it was difficult to specify the project's scope. Regardless of whether a word was written in block or cursive, we had to classify each handwritten word's visual representation for this study. This project can be combined with word separation algorithms in a given line image, and word separation techniques in a given image of a handwritten page can be linked with these algorithms.Our project can become a deliverable that a user would use with these extra layers. This fully functional model asks the user to snap a picture of a page of notes in an attempt to solve the problem of digitizing handwritten materials. Despite the fact that our methodology needs a few more layers to yield a product that is fully functional for the user, we focus on categorization because we find it to be the most interesting and challenging part of the work.

II. MOTIVATION

Growing digitization is converting handwritten text to digital text. This conversion solves a number of issues, such as document transfers, worldwide accessibility, security issues, and physical damage to the document.

A crucial part of this transition is played by Handwritten Recognition (HWR) or Handwritten Text Recognition (HTR), which recognizes transcripts from information inspection photos and turns them into high-level designs.

https://github.com/sahithi9999/NN-Project.git

The input is provided by a variety of devices, including notebooks, documents, forms, and pictures.
There are two types of acknowledgment: 1. Offline Recognition and 2. Online Recognition

2.1 OFFLINE RECOGNITION: With this technique, the text from the photographs is automatically converted to digital text. It is claimed that the data used in this technique is a static depiction of handwriting. This method becomes challenging for many varieties of handwriting..

2.1.1TRADITIONAL Procedures

• Character Extraction: After adding a filter to the picture, remove the people from it. This method has the drawback of returning connected characters as a single sub-image containing all associated characters, which makes recognition difficult.
• Identifying Personalities: The computer creates the relevant digital character when each character has been recovered.
• Derivation of Features: Currently, the programmer must ascertain which properties appear to be significant. Compared to a brain organization, this challenge will take longer to solve.

2.1.2 MODERN Strategies

II. MAIN CONTRIBUTIONS & OBJECTIVES

- Haripriya Eddala has contributed 20% of the

  Multi-Class Logistic Regression
  Support Vector Machine
  Multinomial Logistic Regression

- Sahithi Gunda has contributed 30% of the work
  Using Keras's MNIST Dataset

  Neural Networks
  Recurrent Neural Network
  Decision Tree

- Upender Reddy Chitla have contributed 30% of the work completed.

  Using Extra Keras's EMNIST Dataset
  Convolution Neural Networks
  Random Forest
  Neural Networks

https://github.com/sahithi9999/NN-Project.git

3.1 MAIN CONTRIBUTIONS

• The division of lines is a part of the current procedures, whereas the division of characters is part of the conventional methods.
•Centered on an identifiable AI process.
• Multiple overlapping windows in the text image are handled by recurrent neural networks, while feature extraction is handled by convolutional neural networks.

2.2 ONLINE RECOGNITION

This approach converts text automatically as it is written. In this case, the pen tip's movements are detected using sensors. This method is said to produce data in the form of a digital handwritten representation. Using this method, the impulses are translated into corresponding digital characters. The online handwriting recognition interface consists of a writing pen, a touch screen that is integrated with the output display, and an application that detects and converts pen-tip movements into digital text.
This method involves the following steps:

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

OBJECTIVES:

- The main goal is to locate handwritten words, lines, paragraphs, and other content in online documents. Handwriting recognition has been the subject of a significant deal of research and reviews.

- Character Extraction: The procedure comprises character extraction once the image has been scanned.

- Character Recognition: The system outputs the corresponding digital character after extracting each and every individual character.

- Feature Extraction: In this step, the programmer must determine which properties may seem important.

RELATED WORKS

People have been sharing their thoughts with others for a very long time by recording them in letters, transcripts, and other media. However, the introduction of computers soon caused handwritten text to be replaced with digital writing produced by computers. People believe that a system that can digitize handwritten writing is necessary since it makes handling large amounts of data easy and quick. This kind of structure has previously been attempted to be advanced by other explorers. However, there is still a critical need for more study in this area. Numerous offline and online recognition studies have focused on the handwritten characters of the most widely spoken languages in the world, including English, Chinese, and Indian scripts like Devanagari, Malayalam, and Bangla [2–12]. All of these studies do, however, have some drawbacks, including poor accuracy, a large false positive rate, and a sluggish rate of development. evaluated how well various classifiers performed in identifying handwritten numbers. 4]. Gradient and curvature features are the most accurate for handwritten character identification tasks, according to a few character recognition studies [13]. A recent work [16] used a three-layer approach to evaluate the wavelet modifications of the input character image for handwritten Devanagari and Bangla character recognition. Rajib et al. [17] suggested a method for handwritten character identification in English that is based on the Hidden Markov Model. This method uses two distinct feature extraction techniques: local feature extraction and global feature extraction. There are multiple highlights in the global element in four, six, and four separate quantities. Form, projection, and angle highlights are some examples of these highlights. Nevertheless, by splitting the example image into nine identical pieces, nearby elements are found. The gradient feature for every block is computed using four feature vectors, yielding a total of thirty-six local features. For every sample image, this generated fly features on a local and global scale. After that, the HMM model is trained using these features. Additionally, post-handling of information is used in this strategy to minimize the cross-classification of distinct classes. Training and feature extraction take time when using this strategy. Moreover, it responds badly to such inputs when multiple characters are combined into a single image. A recognition technique based on multi-scale neural network training was put up by Velappa Ganapathy et al. [18]. This method made use of a threshold that could be customized and was determined by the minimum distance methodology in order to increase accuracy. Developing a graphical user interface (GUI) that can identify characters in the scanned image is another requirement of this technology.With a medium training level, this approach yields an accuracy of 85%. This approach required less training time because it used high resolution photos (20 28 pixels).The fuzzy membership function was employed by T. Som et al. [19] to increase the accuracy of the handwritten text recognition system. This technique normalizes text graphics to each class before applying a fuzzy approach 20 × 10 pixels. An outline of the character's bonding box is created in order to determine the text's vertical and horizontal projection. The image is cropped to a bounding box and then scaled to 10×10 pixels. Cropped photographs are then thinned with the help of the thinning operation. To create the test matrix, each of these previously processed images is stacked one after the other into a single matrix. It is determined whether the user's display of the fresh (test) photographs corresponds with the test matrix. The approach was fast, but not very accurate. A method for reducing the system's training time that makes use of a single layer neural network was presented by Rakesh Kumar et al. [20]. Segmented

https://github.com/sahithi9999/NN-Project.git

characters are scaled to eighty by eighty pixels. In order to improve training outcomes, data matrices.However, their result's accuracy rate is low. Another notable study by Zamora [22] is an improved version of this work, feature extraction using the diagonal approach [21]. The others used a zone-based method to extract hybrid traits from the text. As a result, speed and accuracy rose. Making use of the Euler number Approach, speed, and accuracy are improved. To lower the cross-error rate, preprocessing methods including thresholding, thinning, and filtering are used to the input image. There are three techniques used to enhance the segmentation. After segmentation, the input image is resized to 90 by 60 pixels. The texts are divided into 54 zones, each with 10×10 pixels, after the Euler number for each text is calculated. The mean value of each zone, both in terms of rows and columns, is used to create the feature vector for the character. Their approach was presented by Anshul Mehta at el [23] and is based on the heuristic segmentation method. Their method accurately identifies suitable segmentation sites between handwritten letters. This feature extraction technique makes use of Fourier descriptors. Following a successful segmentation, the discrete Fourier coefficients of the input picture (a[k] and b[k]) are discovered. Here, L stands for the input image's border points, and k is a number between zero and (L-1). This approach was used to try and categorize a total of fifty-two characters (26 upper case and 26 lower case English letters). It also provides a comparison of several classification methods. [24] published a state-of-the-art interactive method for recognizing handwritten characters. It is only necessary to involve humans when certain inputs cause confusion for the system. Human lead is increased even when exceptional precision is maintained. The only issue was that some technology did not work entirely automatically, requiring human input. Amma et al. [25] proposed a wearable input system that allows users to modify the text painted in the sir. It was a 3D reconciling handwriting acknowledgment technique. Utilizing gyroscopes, accelerometers, and motion sensors that are precisely positioned beneath the human hand.Although written data could not be applied in the same manner, the procedure made sense. The sample input picture is represented by a bit map in the feature vector of this article. Selecting the optimal feature vector is a crucial initial step in any recognition system. To facilitate accurate pattern classification, a feature extraction method is proposed that uses a small number of characteristics that are good at differentiating between distinct pattern classes. Bitmap version of the parent image maintains all significant features within a tiny neighborhood. The investigation is part of the planned as well. The proposal also looks at how the framework has been altered to account for various teaching strategies. It also illustrates how different parameter choices impact several factors, including the quantity of hidden layers and epochs. Preprocessing steps in the Character division, disturbance evacuation, normalization, and de-slanting are among the suggested techniques. This study can detect up to 95% of English characters with accuracy. The verified usability, ease of use, and high rate of acceptance of the suggested framework imply that it might be very beneficial for practical implementation.

III PROPOSED FRAMEWORK

3.1.1 MULTI – CLASS LOGISTIC REGRESSION

Multiclass logistic regression and multinomial logistic regression extend the classification of logistic regression to multiclass, or more than binary, outcomes.

This type of machine learning model predicts the probability of specific results.

Alternate names of this model:

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

- Importing digits dataset from sklearn
- Loading digits dataset to the data frame
- Importing train test split method from sklearn
- Splitting total dataset into train and test datasets
- Creating the 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

Motivation One well-liked supervised technique for handling regression and classification issues is the vector machine. The main application of machine learning is in classification difficulties. The optimal line is produced by the SVM algorithm. or divide classes using an n-dimensional spatial decision boundary. As a result, expanding the appropriate category with additional data points in the future will be simple. Here, the best option boundary is represented as a hyperplane. It selects the extreme vectors or points to aid in the creation of the hyperplane. The support vectors are the extreme ones. As a result, this method is called a "Support Vector Machine." This approach is used for face identification, text categorization, and photo classification.

- Linear SVM: In this case, we employ linearly separable data. One straight line can separate two classes from the dataset; this is referred to as "linearly separable data." The classifier is known as the linear SVM classifier.

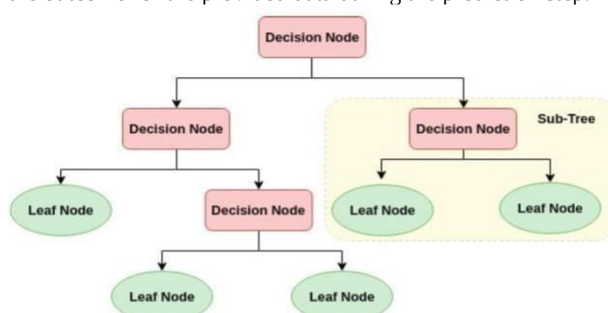
https://github.com/sahithi9999/NN-Project.git

- Non-linear SVM: In this, we use non-linearly separable data. A dataset is considered non-linearly separable if a single straight line cannot separate it into two groups. The classifier is called a Non-linear SVM classifier.
- Importing the digits dataset from sklearn

- Loading digits dataset to a data frame
- Importing train test split method from sklearn
- Importing the 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 predicted output to actual output

3.1.3 DECISION TREE

Regression and classification issues are resolved with the usage of decision trees. It can handle multi-dimensional data with accuracy. The majority of widely used classification strategies are quite simple to comprehend and use. A decision tree is a type of tree structure that resembles a flowchart, where each node represents a feature and each edge indicates an option. The process of categorizing involves two steps: prediction and learning. Using the provided training data, the model is constructed during the learning phase and used to forecast the outcome for the provided data during the prediction step.

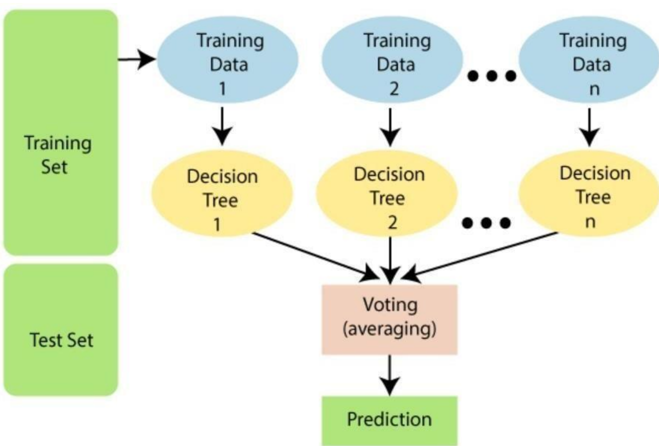


- Importing required libraries
- Importing the digits dataset from sklearn
- Loading the digits dataset to a data frame
- Data Preprocessing

- Importing train test split method from sklearn
- Splitting the total dataset into train and test dataset
- Importing Tree for the Decision Tree Classifier from sklearn
- Creating the 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

Among the machine learning algorithms utilized in the supervised learning methodology is the Random Forest Algorithm. This technique is applied in machine learning to address regression and classification issues. This is based on a method that combines multiple classifiers to solve a complex problem and enhance the performance of the model. The Random Forest classifier chooses the average of multiple decision trees applied to various dataset subsets in order to increase the dataset's prediction accuracy. It uses the majority votes of each decision tree's projections to anticipate the outcome rather than depending on just one decision tree. The amount of trees in the forest increases accuracy, avoiding the overfitting problem.



- 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

https://github.com/sahithi9999/NN-Project.git

3.2 USING KERAS'S MNIST DATASET

3.2.1 NEURAL NETWORKS

ARCHITECTURE OF A NEURAL NETWORK:

In Neural Network there are three types of architectures.

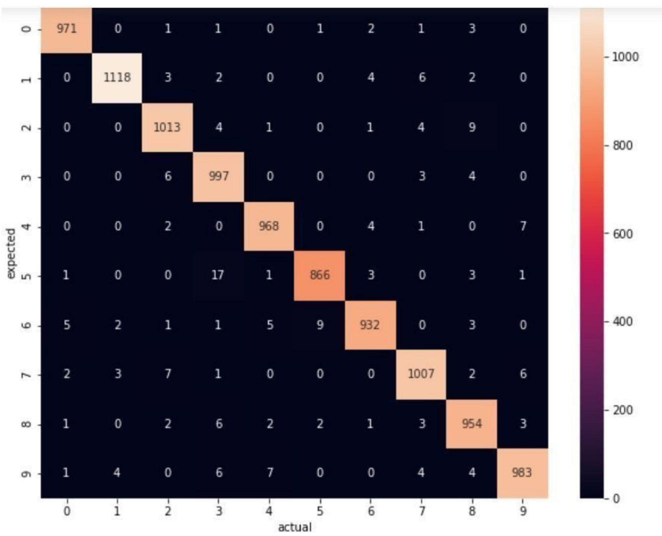
1.Single-Layer Feedforward Network: A layer of neurons in the output is projected onto an input layer of source codes. This kind of network is referred to as an acyclic or feedforward network. It alludes to the output layer's computation neurons. As a result, it is called a single layer. The input layer is not used for any computation. Therefore, it is not included in the total.

2.Multi-Layer Feedforward Network: In addition to the input and output layers, this network also has one or more hidden layers. These nodes are sometimes referred to as hidden neurons or hidden units. The output and the external input are separated by the concealed layer. For the length of the network that survives, fixed input layer nodes provide the input signal to the hidden layer, and the output from the subsequent layer gets data from the buried layer, and so on.

3. Recurrent Networks: This network can be compared to the feedforward network. Its one or more feedback loops are the main point of differentiation.

- 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

The way the neurons in this feed-forward artificial neural network connect to one another is patterned after the structure of the visual cortex. This network uses several layers of arrays to process data. This kind of neural network is used for image and face identification. CNN processes the images without concentrating on feature extraction, instead using the two-dimensional array as input. This network is based on three main concepts: local respective fields, pooling, and convolution.

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

https://github.com/sahithi9999/NN-Project.git

3.2.3 RECURRENT NEURAL NETWORKS

Recurrent neural networks are a kind of artificial intelligence that identify patterns in data sequences, such as text, handwriting, and spoken speech. The back propagation technique is used to train the model. Back propagation is called back propagation over time because it happens for every timestamp. Networks with Long Short-Term Memory (LSTMs): These special neural networks are mostly used to learn long-term dependencies in sequence prediction problems. Thanks to a feedback connection, it can handle the entire data sequence except for individual data points (images).

- Importing the required libraries
- Loading the mnist dataset
- Data preprocessing
- Creating Recurrent Neural Network Model
- Adding more Layers
- Compiling the model
- Training the model
- Predicting few elements from 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 the 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 the 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 the sklearn Random Forest Classifier
- Loading emnist dataset
- data preprocessing
- Building a Random Forest Classifier model
- Training 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 accuracy of the model
- Displaying input images
- Creating the list to map the predicted output to the corresponding digital character.
- Predicting a few elements from the test dataset
- Comparing predicted output with actual images
- CONVOLUTION NEURAL NETWORKS
- Importing the necessary libraries
- Loading emnist dataset
- Data Preprocessing
- Creating Convolution Neural Network Model
- Adding more layers
- Compiling the model
- Training the model
- Evaluating the model for test cases

https://github.com/sahithi9999/NN-Project.git

- Displaying the input images
- mapping the anticipated output to the appropriate digital character by creating a list.
- Predicting few elements from the test dataset
- Comparing the predicted output to the actual outputs

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 digital character.
- 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 given image

III. 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 per Class	~ 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")
```

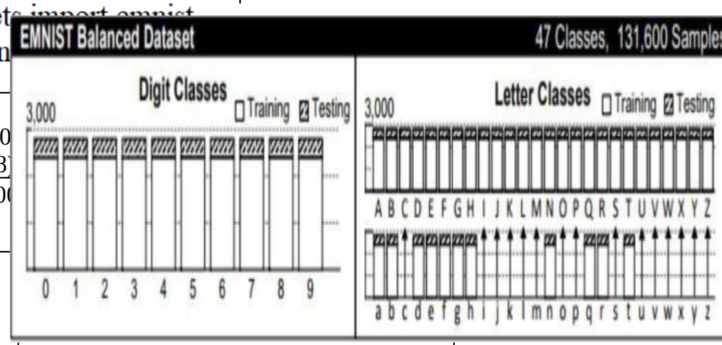
The dataset contains 28x28 image of a digit.

Classes	10
x_train	(60000, 28, 28)
y_train	
x_test	(10000 28)
y_test	(100

Extra Keras MNIST Dataset

To load the dataset, we use:

```
from extra-keras-datasets import emnist
(input_train, target_train ... d')
```



Classes	47

B. Data Preprocessing

Preprocessing is necessary before handwritten photo datasets can be utilized in machine learning. The dataset needs to be organized and cleaned in order to make it easier to use for the machine learning

https://github.com/sahithi9999/NN-Project.git

algorithm. First, all false data points should be removed. Inaccurate data entry or mistakes made throughout the data collection procedure can lead to invalid data points. Invalid data points could potentially arise from outliers in the data collection process. Outliers are data points that substantially differ from the remainder of the sample. They could distort the results of the machine learning algorithm if they are not removed.

Due to missing values and/or noisy data, the quality of the raw data could be worse than the quality of the final projection. Preprocessing is therefore required to improve the suitability of the data for mining and analysis of the three kinds of smoking habits. This includes redundant value reduction, feature selection, and data discretization. Regarding BMI, a sizeable fraction of individuals (25%) are fat, whereas 18% are overweight. the ranking score that the chosen feature relevance technique produced for the balanced data.

C. further clarifies the importance of BMI. There were 201 missing Body Mass Index (BMI) feature values in the original dataset. The mean BMI for the whole dataset was used to fill in these figures. Moreover, it was shown that more than 30% of the population did not smoke; this finding may have been caused by incomplete or missing data regarding feature values. Due to the overwhelming volume of information, it was decided to reclassify those people using some assumptions in order to prevent omissions. The Unknown values for those under the age of eighteen were modified to never because they are no longer as likely to smoke as they formerly were. As a result, the dataset had 909 fewer ok unknowns than it did 1544 times earlier.

Additionally, the numbers for each sort of employment were reclassified, moving from "children" to "never worked." This is due to the possibility that children have dreams of "never working" and the fact that they weren't meant to be classified as laborers in the first place.

D. Data Preparation

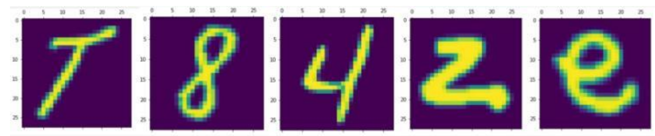
The second step is to normalize the data. Standardizing the data is converting each data point to the same unit of measurement. This is important because it ensures that the machine learning system compares data fairly. The final step is to combine the data sets. This is necessary if the data collection is divided into many files. The fourth step is to label the data. This is necessary if the data set hasn't been labeled already. Labeling the data is the process of assigning a name to each data piece. The sixth step is to remove any duplicate data points. Duplicate data points have the potential to alter the output of the machine learning algorithm.

The sixth step involves splitting the data into training and testing sets. The training set is used to train the machine learning algorithm. The testing set is used to assess the accuracy of the machine-learning system. The eighth step is formatting the data. In the event that the format of the data is unsuitable for the machine learning algorithm, this is required. The seventh step is to filter the data. This is necessary if the data collection is too large to use using the machine learning method. The ninth step is to normalize the data. Making changes to data so that the mean is zero and the standard deviation is one is known as normalization. This is important because it ensures that the machine learning system compares data fairly.

The ninth step involves choosing the machine learning method. The machine learning algorithm is the one that will be utilized to extract knowledge from the data set. The eleventh step is choosing the parameters for the machine learning algorithm. The parameters are the configurations that the machine learning algorithm will use to infer conclusions from the data set. The twelfth step involves putting the machine learning algorithm into action. The machine learning algorithm is applied to the data set at this point. In the twelfth phase, the results of the machine learning algorithm are evaluated. At this point, the accuracy of the machine-learning algorithm is evaluated. At stage fourteen, the machine learning algorithm is adjusted if needed.

This stage involves adjusting the machine learning algorithm based on the evaluation's results. The fifteenth step involves repeating the fourteen and sixth phases until the machine learning system reaches the desired accuracy.

IV. RESULTS



```
mapping = {0:"0", 1:"1", 2:"2", 3:"3", 4:"4", 5:"5", 6:"6", 7:"7", 8:"8", 9:"9", 10:"a",
           11:"b", 12:"c", 13:"d", 14:"e", 15:"f", 16:"g", 17:"h", 18:"i", 19:"j", 20:"k",
           21:"l", 22:"m", 23:"n", 24:"o", 25:"p", 26:"q", 27:"r", 28:"s", 29:"t", 30:"u",
           31:"v", 32:"w", 33:"x", 34:"y", 35:"z", 36:"a", 37:"b", 38:"d", 39:"e", 40:"f",
           41:"g", 42:"h", 43:"n", 44:"q", 45:"r", 46:"t"}
output = model.predict_classes(x_test[20:29])
for element in output:
    print(mapping[element])
```

```
t
8
4
z
e
f
r
x
b
```

https://github.com/sahithi9999/NN-Project.git

Dataset	Model	Description	Accuracy
Sklearn's Digit	Multinomia l 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/sahithi9999/NN-Project.git

Keras's MNIST	Convolution Neural 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/sahithi9999/NN-Project.git

	Recurrent Neural Networks	Layer 3: type=LSTM, activation=sigmoid Layer 4: type=Dense, activation=relu Layer 5: type=Dropout Layer 6: type=Dense, activation=softmax Optimizer=Adam, Learning rate=1e-3, Decay=1e-5 Loss=sparse_categorical_cross entropy Epochs=5	98.18%
	Multinomial Logistic 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%

		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/sahithi9999/NN-Project.git

CONCLUSION

A proposed technique for digitizing handwritten text has been created and assessed. A comparison with existing relevant works has been done.
Character extraction can be used to recognize and alter words using this technique.
It is also useful for identifying and transforming phrases by breaking words apart and eliminating letters.

REFERENCES

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

[3]. [1]Karthik Dutta, [2]Praveen Krishnan, [3]Mahesh Mathew, [4]C.V. Jawahar,
"Improving CNN-RNN Hybrid Networks for Handwriting Recognition," CVIT

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

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

https://github.com/sahithi9999/NN-Project.git