Handwritten Text to Digital Text Conversion using Various Deep Learning Models

Shiny Sherly Katuru
Computer Science
University of Central Missouri
Lee's Summit, United States of America
sxk43140@ucmo.edu

Bhavyasri Maddineni
Computer Science
University of Central Missouri
Lee's Summit, United States of America
bxm84990@ucmo.edu

Sneha Latha Kusuma
Computer Science
University of Central Missouri
Lee's Summit, United States of America
sxk53740@ucmo.edu

***Abstract*— People think it is clearer to make than to create on paper, but everything is going digital now. Consequently, HTR/HWR use is now on the rise. There are several ways to view handwriting. Segmentation for recognition, machine learning, convolutional neural networks, and recurrent neural networks are newer techniques, while character extraction, character recognition, and feature extraction are more traditional techniques. Applications for HTR/HWR include online and offline identification, signature verification, postal address interpretation, bank check processing, and writer identification.**

Handwritten Text Recognition (HTR) also known as Handwriting Recognition (HWR) is the detection and interpretation of handwritten text images by the computer. Handwritten text from various sources such as notebooks, documents, forms, photographs, and other devices can be given to the computer to predict and convert into the Computerized Text/Digital Text.

There are various applications for the HTR/HWR such as the Online recognition, Offline Recognition, Signature verification, Postal address interpretation, Bank-Check processing, Writer recognition and these are considered to be the active areas of research.

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

I. Introduction

Over the past few years, research into the recognition of handwritten documents has attracted a lot of attention. Changing over a piece of transcribed data into computerized text for sharing or saving without composing the data manually is constantly expected. As input, the proposed model transforms a photograph of handwritten text into digital text. The Convolutional Brain Organization (CNN) is utilized to concentrate on the elements of comparative articles from numerous picture tests and to characterize them. Since the text is successive information, Long Transient Memory (LSTM), an expansion of Repetitive Brain Organizations (RNN) with a more extended memory is utilized. Connectionist Temporal Classification (CTC) loss is used to deal with the various locations of the text in the image. Training is done with handwriting samples from over 600 writers and images of over 100,000 words from the IAM Handwriting Database. Subsequent to preparing for different ages, the model enrolled 94% precision and a deficiency of 0.147 on preparing information and 85% exactness and a deficiency of 1.105 on approval information.
Even though there are many technological writing tools available, many people still prefer to take their notes the old-fashioned way, on paper and with a pen. But handwriting text has disadvantages. Physical documents are challenging to share with others, store and access conveniently, and search through. Because documents are never converted to digital format, a great deal of crucial information is lost or not evaluated. We have chosen to address this issue in our project because we think that people will be able to access, search for, share, and analyze their records more effectively while still using their preferred writing style thanks to the significantly easier management of digital text compared to written text.

This project's goals include deeper investigation of the classification process and digitization of handwritten material. Since the word "handwritten text" has a fairly broad definition, we needed to define it specifically for our needs in order to focus the project's scope. In this project, we took on the challenge of categorizing the visual of any handwritten word, whether it be in block writing or cursive. This project can be integrated with algorithms that separate the words in a given line image, and those that separate the lines in a given image of a handwritten page in its entirety can be combined with these algorithms. With these additional layers, our project can take the shape of a deliverable that an end user would use. This model would be fully functional and assist the user in resolving the issue of converting handwritten documents into digital format by prompting the user to take a picture of a page of notes. Note that even though our model requires some additional layers to be added on top of it in order to produce a deliverable that is fully functional for the end user, we chose to focus on classification because we find it to be the most interesting and difficult aspect of the problem.

II. MOTIVATION

Because of widespread digitalization, handwritten text is being converted into digital text. This conversion addresses a variety of issues, including security-related issues, transferring the document, accessing the document from anywhere in the world, and physical damage to the document.
In this transformation manually write Text Recognition (HTR) or Handwriting Recognition (HWR) undertakes a significant part of identifying transcripts from information inspection images and converting them into high-level designs. The input comes from various devices, including notebooks, documents, forms, and photographs.

There are two kinds of recognition: 1. Recognition Offline 2. Online Recognition

2.1 OFFLINE RECOGNITION: In this method, the text from the images is automatically converted into digital text. The static representation of handwriting is said to be the data used in this method. For a variety of handwriting styles, this method becomes challenging.

2.1.1TRADITIONAL Procedures

https://github.com/700748499/NN-DL-_Final_Increment.git

• Character Extraction: Include a filter on the picture and then separate the included individuals. The issue with this method is that when characters are connected, they are returned as a single sub-image with all the connected characters, making recognition difficult.
• Identifying Personalities: The computer generates the corresponding digital character after each character has been extracted.
• Derivation of Features: The programmer must identify the properties that may appear to be crucial in this step. This issue requires more improvement time than that for a brain organization.

2.1.2 MODERN Strategies

III. MAIN CONTRIBUTIONS & OBJECTIVES

3.1 MAIN CONTRIBUTIONS

• While the conventional procedures include the division or characters, the current procedures include the division of lines.

- Sneha Latha Kusuma has contributed 20% of the

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

- Bhavyasri Maddineni has contributed 30% of the work completed.

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

- Shiny Sherly Katuru has contributed 30% of the work completed.

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

OBJECTIVES:

https://github.com/700748499/NN-DL- Final Increment.git

• Centered on an AI approach with familiar elements.
• Utilization of Convolutional Neural Networks for feature extraction
• Utilization of Recurrent Neural Networks for multiple windows of the text image that overlap.

2.2 ONLINE RECOGNITION

This method converts text as it is written automatically. The pen-tip movements are detected by means of sensors in this case. This method is said to produce data that is a digital representation of handwriting. The signals are transformed into the corresponding digital characters using this method. The elements of the online handwriting recognition interface include a pen for writing, a touch surface that is integrated with the output display, and an application that determines and converts pen-tip movements into digital text.
This method involves the following steps:

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

The major goal is to identify handwritten text in online documents, including letters, words, lines, and paragraphs. The field of handwriting recognition has seen a lot of effort, and there have been several reviews.

- Character Extraction: It involves scanning of image and then individual character contained is to be extracted.

- Character Recognition: Once the individual characters are extracted, the computer outputs the corresponding digital character.

- Feature Extraction: In this step the programmer must determine the properties that may appear to be important.

IV. Related Works

Humans have written their thoughts down in the form of letters, transcripts, and other forms for a very long time; to pass them on to other people. However, with the invention of computers, digital writing produced by computers soon replaced handwritten text. Because it makes processing such data quick and easy, individuals sense the need for a system that can turn handwritten writing into digital text. In the past, several explorers made an effort to promote such a framework. However, there is still a critical need for more study in this area. Numerous recognition studies have been conducted for offline and online handwritten characters of the most widely used languages in the world, including English, Chinese, and Indian scripts, such as Devanagari, Malayalam, and Bangla [2-12]. However, each of these studies has some drawbacks, including slow change speed, poor precision, a higher rate of false [] conducted a performance comparison of several classifiers for handwritten digit recognition. 4]. The most accurate characteristics for handwritten character recognition tasks are gradient and curvature [13] features. a few character recognition experiments. A recent study [16] used a three-layer approach to analyze the wavelet transforms of the input character image for handwritten Devanagari and Bangla character recognition. Rajib and others [17] proposed a Hidden Markov Model-based English handwritten character recognition system. Global and local feature extraction were the two distinct feature extractions utilized in this approach. Worldwide element incorporates many highlights like angle highlights, projection highlights and shape highlights in the quantities of four, six and four separately. Though nearby elements are determined by partitioning the example picture into nine equivalent blocks. Each block's gradient feature is calculated using four feature vectors, giving a total of 36 local features. For each sample image, this resulted in fty features (local and global). After that, in order to train the HMM model, these features are fed into it. Information post handling is likewise used by this technique to diminish the cross classi□cation of various classes. Training and feature extraction take a significant amount of time using this approach. In addition, it performs poorly when multiple characters are combined in a single image in response to such inputs. The multi scale neural network training-based recognition approach was suggested by Velappa Ganapathy et al. [18]. This method employed a selectable threshold that was computed using the minimum distance methodology in order to increase accuracy. This approach also entails the creation of a GUI that can identify characters throughout the scanned picture. With a medium degree of training, this approach offers 85% accuracy. This technique required less training time since it used pictures with a large resolution (20 28 pixels).The fuzzy membership function was utilized by T. Som et al. [19] to increase the precision of the handwritten text recognition system. In this method, text images are normalized to 20 × 10 pixels and then fuzzy approach is used to each class. Bonding box is created around the character in order to determine the vertical and horizontal projection of the text. Once the image is cropped to a bounding box, it is re-scaled to the size of 10×10 pixels. Then, cropped images are thinned by the help of thinning operation. In order to create the test matrix, all these pre-processed images are placed into a single matrix; one after another. When new (test) images are presented by the user, it is tested for the matching against the test matrix. The method was fast but it provides a low accuracy. Rakesh kumar et al [20] proposed a method in order to reduce the training time of system by utilizing a single layer neural network. Segmented characters are scaled to 80 X 80 pixels. Data normalization is performed on the input matrices to improve the training performance. But their result has a low accuracy rate.Other notable work proposed by Zamora includes feature extraction using the diagonal method [21], an improved version of this work [22]. The others used zone based hybrid feature extraction from the text. Doing so, led to improvement in speed and accuracy. By using Euler number approach, speed and accuracy are improved. Many preprocessing like Thresholding, thinning and □ltering operations are performed on the input image so that cross error rate can be minimized. Three techniques are utilized for better segmentation. After segmentation, the input image is resized to the size of 90 X 60 pixels. Then after, Euler number is calculated for each text and then they are divided into 54 zones, such that each contains 10×10 pixels. The average value of each zone (row and column wise) is used as the feature vector of the character.Their idea was proposed by Anshul Mehta at el [23] and is based on the heuristic segmentation method. Their method does a good job of identifying legitimate segmentation points between handwritten letters. This method for feature extraction employs Fourier descriptors. Following a successful segmentation, the input image's discrete Fourier coefficients (a[k] and b[k]) are determined. Here, L stands for the input image's border points, and k ranges from zero to (L-1). This approach attempted to categorize 52 characters in total (26 upper case and 26 lower case English letters).It also provides a comparative analysis of different classi□cation methods. [24] proposed a novel interactive method for recognizing handwritten characters. Only the inputs that cause confusion for the system necessitate human intervention. Despite maintaining high accuracy, it increases human lead. The only issue was that the system did not operate entirely automatically and necessitates human intervention. Amma and co. 25] proposed a wearable input system that allowed users to alter the text drawn in the sir. It was a 3D reconciliation technique for penmanship acknowledgment. The handwriting motions were wirelessly recorded with the use of motion sensors, accelerometers, and gyroscopes placed exactly to the back of the human hand. Although the approach was sound, written data could not be used in the same way. The feature vector in this article is a bit map representation of the input picture sample. The optimal feature vector selection is a crucial step in any recognition system. The suggested feature extraction approach seeks to assist in accurate pattern categorization by employing a small amount of features that are efficient in differentiating pattern classes. The bit map version of the parent picture retains all of the essential details in a compact neighborhood. The inquiry is also included in the planned.The analysis of the changes brought about in the framework as a result of various learning systems is also included in the suggested. It also demonstrates how different parameter choices have an impact on a variety of factors, including the quantity and size of hidden layers as well as epochs. The suggested technique's preprocessing comprises things like commotion evacuation, character division, normalization, and de-slanting. This study attempts to detect English characters up to 95% accurately. The suggested framework may prove to be very beneficial for practical usage due to its verifiable ease of use, usability, and high rate of acceptance.

III PROPOSED FRAMEWORK

3.1.1 MULTI – CLASS LOGISTIC REGRESSION

Multiclass Logistic Regression / Multinomial Logistic Regression generalizes the classification of Logistic Regression to Multiclass, i.e., with more than binary outcome.

The ML model of this type predicts the probabilities for different 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

Support A well-liked supervised technique for classification and regression issues is the vector machine. In machine learning, classification issues are its main application. The optimal line is made by the SVM algorithm. or a decision boundary to divide classes in n-dimensional space. As a result, adding additional data points to the appropriate category in the future will be simple. A best choice boundary is a hyperplane, which is what we have here. It selects the extreme vectors or points, which aids in the creation of the hyperplane. Support vectors are the extreme vectors. This method is hence called a "Support Vector Machine." This method is used for face identification, picture classification, and text categorization.

- Linear SVM: In this we use linearly separable data. If a single straight line can separate the dataset into two classes, it is known as linearly separable data. Classifier is known as linear SVM classifier.

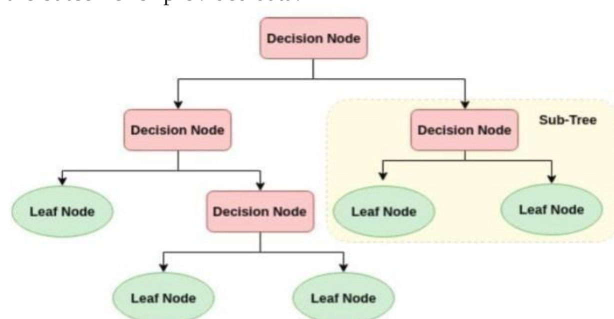
https://github.com/700748499/NN-DL- Final Increment.git

- Non-linear SVM: In this we use non-linearly separable data. If a single straight line cannot separate the dataset into two classes, it is known as non-linearly separable data. Classifier is known as Non-linear SVM classier.

- 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

Regression and classification issues are solved using decision trees. It can accurately handle data with several dimensions. Popular categorization methods are fairly simple to comprehend and interpret. A decision tree is a tree structure that resembles a flowchart; each node represents a feature, and each edge a choice. The categorization method consists of two steps: learning and prediction. The model is created based on provided training data in the learning stage, and in the prediction step, it is utilized to forecast the outcome for provided data.

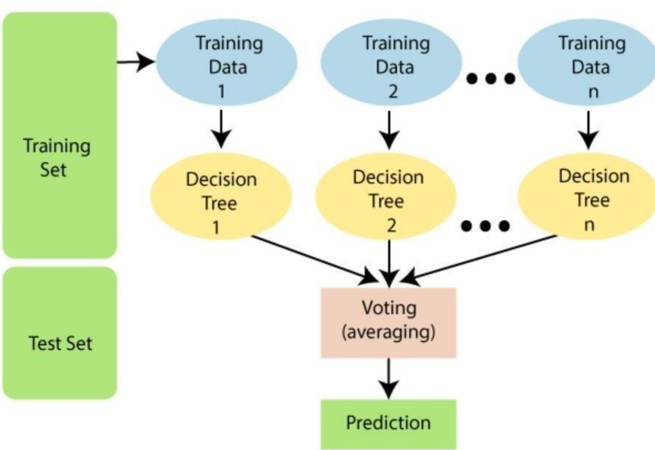


- Importing required libraries
- Importing the digits dataset from sklearn
- Loading the digits dataset to a data frame
- Data Preprocessing

- 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 supervised learning method includes the machine learning algorithm known as the Random Forest Algorithm. This approach is used to solve classification and regression issues in machine learning. This is based on a method that combines many classifiers to tackle a challenging problem and enhance the performance of the model. Random Forest is a classifier that uses many decision trees on various subsets of the dataset, with the average being selected to increase the dataset's prediction accuracy. It predicts the output based on the majority votes of predictions of each tree without depending on one decision tree. As the number of trees increases in the forest the accuracy also increases and prevents the overfitting problem



- Iimporting 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: Computing systems use biological neural networks as inspiration to carry out a variety of tasks on a large amount of data. Artificial neural networks are this. To get the best outcomes by adjusting the inputs, several algorithms are utilized to comprehend the relationships in the dataset. To provide the required results, neural networks are trained. Different models are used to the data in order to anticipate future outcomes. Each of the nodes is linked to the others through The nodes are linked to one another so that it can function similarly to the human brain. Data are clustered and classified using correlations and hidden patterns.

ARCHITECTURE OF A NEURAL NETWORK:

In Neural Network there are three types of architectures.

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

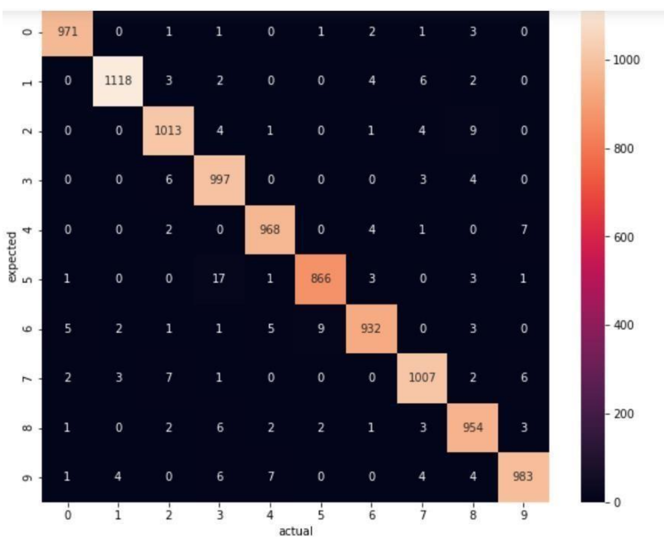
2.Multi-Layer Feedforward Network: With the exception of the input and output layers, this network has one or more hidden layers. These layer's nodes are also referred to as hidden neurons or hidden units. Between the output and the external input is where the hidden layer is inserted. The input signal to the hidden layer is provided by fixed input layer nodes, and the output from the hidden layer is provided to the following layer, and so on throughout the remaining network

3.Recurrent Networks: This network is similar to feedforward network.The maindifference is that it has at least one feedback loop

- Importing necessary Libraries

https://github.com/700748499/NN-DL- Final Increment.git

- 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 connection pattern between the neurons in this unique breed of feed-forward artificial neural network is modeled after that of the visual cortex. This network uses numerous layers of arrays for data processing. This kind of neural network is used for face and image recognition. Without concentrating on feature extraction, CNN accepts the two-dimensional array as input and acts on the pictures directly. This network relies on three fundamental concepts: local respective fields, convolution, and pooling.

- 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

Recurrent Neural Network is a kind of artificial Intelligence in which the patterns in the data sequences such as text, handwriting, and spoken words are identified. It uses back propagation algorithm for training the model. It is known as back propagation through time as back propagation happen for every timestamp. Long Short-Term Memory Networks (LSTMs): These are the special kind of neural networks that are used for learning long-term dependencies

mainly in sequence prediction problems. It has a feedback connection that is itcan process the entire data sequence apart from single data points (images).

- 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

https://github.com/700748499/NN-DL- Final Increment.git

- 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

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

https://github.com/700748499/NN-DL- Final Increment.git

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.

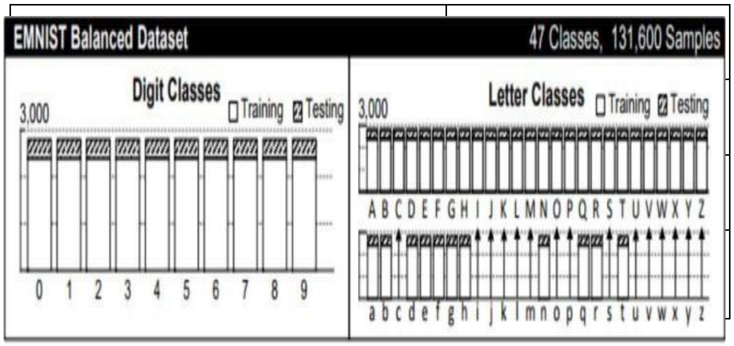
Classes	10
x_train	(60000, 28, 28)
y_train	(60000,)
x_test	(10000, 28, 28)
y_test	(1000,)

https://github.com/700748499/NN-DL- Final Increment.git

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*

Preprocessing handwritten image datasets is important for machine learning. The dataset should be cleaned and organized in a way that is easy to use for the machine learning algorithm. The first step is to remove any invalid data points. Invalid data points can be caused by errors in the data collection process or by incorrect data entry. Invalid data points can also be caused by outliers in the data set. Outliers are data points that are far from the rest of the data points in the set. They can distort the results of the machine learning algorithm if they are not removed.

Because of missing values and/or noisy data, the quality of the raw data may be worse than the quality of the final forecast. Therefore, data preparation is required to make it more suitable for mining and analysis of the three types of smoking behaviors. This includes redundant value reduction, feature selection, and data discretization. Regarding BMI, a significant portion of individuals (25%) fall into the obese category, whereas 18% are overweight. The ranking score given by the chosen feature relevance technique in the balanced data additionally accounts for the significance of BMI. 201 Body Mass Index (BMI) feature values were initially missing from the dataset. The mean BMI for the whole dataset was calculated to fill in these numbers. Additionally, it was found that more than 30% of the population does not smoke, which might be interpreted as either missing data or insufficient information on the feature values. Due to the volume of data, it was decided to re-categorize those people by making certain assumptions in order to prevent leaving out any information. The Unknown values existing in those under the age of 18 were altered to never since they have a lower likelihood of smoking today than they did when they were younger. As a result, there were 909 fewer ok unknowns in the dataset as opposed to 1544 before. Another reclassification was changing the values for each employment type from "children" to "never worked." This is due to the fact that children shouldn't have been thought of as a labor type in the first place and may reflect ideals of "never working.

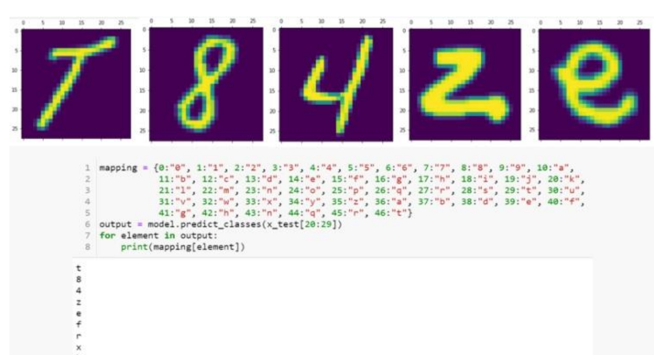
C. *Data Preparation*

The second step is to standardize the data. Standardizing the data means that all of the data points are converted to the same unit of measurement. This is important because it ensures that the machine learning algorithm is comparing apples to apples. The third step is to merge the data sets. This is necessary if the data set is divided into multiple files. The fourth step is to label the data. This is necessary if the data set is not already labeled. Labeling the data means assigning a name to each data point. The fifth step is to remove any duplicate data points. Duplicate data points can distort the results of the machine learning algorithm.

The sixth step is to split the data into training and testing sets. The training set is used to train the machine learning algorithm. The testing set is used to test the accuracy of the machine-learning algorithm. The seventh step is to format the data. This is necessary if the data is not in a format that the machine learning algorithm can use. The eighth step is to filter the data. This is necessary if the data set is too large to use for the machine learning algorithm. The ninth step is to normalize the data. Normalizing the data means adjusting the data so that the mean is zero and the standard deviation is one. This is important because it ensures that the machine learning algorithm is comparing apples to apples.

The tenth step is to choose the machine learning algorithm. The machine learning algorithm is the algorithm that will be used to learn from the data set. The eleventh step is to choose the parameters for the machine learning algorithm. The parameters are the settings that the machine learning algorithm will use to learn from the data set. The twelfth step is to run the machine learning algorithm. This is the step where the machine learning algorithm is actually run on the data set. The thirteenth step is to evaluate the results of the machine learning algorithm. This is the step where the accuracy of the machine-learning algorithm is determined. The fourteenth step is to modify the machine learning algorithm if necessary. This is the step where the machine learning algorithm is modified based on the results of the evaluation. The fifteenth step is to repeat the steps from six to fourteen until the machine learning algorithm reaches the desired accuracy.

VI. Results



https://github.com/700748499/NN-DL- Final_Increment.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/700748499/NN-DL- Final Increment.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/700748499/NN-DL- Final Increment.git](https://github.com/700748499/NN-DL-%20Final%20Increment.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/700748499/NN-DL- Final Increment.git

Extra Keras's EMNIST		Layer 3: type=Dense, activation=softmax Optimizer=adam Loss=sparse_categorical_cross entropy Epochs=10	
	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%

Conclusion

A proposed handwritten text to digital text conversion has been designed and tested. A comparison with related work has been presented.

- This model can be extended to recognize and convert words by extracting its characters.
- It can also be extended to recognize and convert sentences by segmenting words and thereby extracting the characters.

https://github.com/700748499/NN-DL- Final Increment.git

References

[1]. [1]Ahmed Mahi Obaid, [2]Hazem M. EI Bakery, [3]M.A. Eldosuky, [4]A.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 TextRecognition 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/700748499/NN-DL- Final Increment.git