CAPSTONE PROJECT

Handwritten Digit Recognition using Deep learning

Mentor Name:

Dr.Vanitha.M

Team members:

Vasanthakumar.P(18BCS0022)

Manibharathi.S(18BCS0023)

E.Nithyashri(18BCS0070)

<u>GUIDE'S APPROVAL:</u>

CONTENTS

AIM:

This project is aimed to train a Machine learning model which can take an Image of a handwritten digit as an input and recognize it. The model is built on top of Convolutional Neural Network(CNN) with the use of MNIST data set for training and testing .

OBJECTIVE :

1. The main objective is to recognize the handwritten digits only.

2. Reduced man-power by recognizing the handwritten numbers and produce the digitized form of the digit. So that the old handwritten data in the banks, post offices etc., can be recovered and stored in the databases.

3. Proposed system serves as guide and working in digit recognition areas.

SCOPE:

1. System is designed in way to ensure that offline handwritten recognition of numeric digit is done.

2. Old and epic handwritten digits are recognized and are restored in digital form.

3. Use of Neural Network for classification.

4. Large number of training data set will improve the efficiency of the suggested approach as the model gets trained very well.

LITERATURE SURVEY (WITH PROPER REFERENCES CITED):

[1] Comparison between the online and offline handwriting recognition using digitizer technology and various algorithms like shape recognition algorithms, preprocessing and post processing techniques, and experimental systems are done and discussed.

[2] Four different approaches are used to combine several individual classifiers like Bayesian, k-nearest-neighbor to obtain the recognition of totally unconstrained handwritten numerals with 95-98.9% recognition with 0-0.90% substitution and 0.2-5% rejection.

[3] Recognition of unconstrained handwritten text is very challenging as it is hard to segment the data and the data may contain long-range bidirectional inter-dependencies. Instead of using the most approached Markov models for speech and handwriting recognition, this paper relies on a novel type of recurrent neural network(RNN) which overcomes the challenging tasks and gives accuracy of 79.7% on online data and 74.1% on offline data.

[4] This survey describes the nature of both online and offline cases of handwritten language and its transduction into electronic data. Some basic concepts and algorithms of preprocessing, character and word recognition are discussed.

[5] Using Hidden Markov model(HMM), sub-character stroke types are modeled and are concatenated and embedded to form the stochastic language model. In spite of the fact that a stochastic language model for handwriting recognition using HMM is still in development, this survey has obtained an independent recognition rate of 94.5% on 3,823 unconstrained handwritten word samples from 18 writers covering a 32 word vocabulary.

[6] This paper is based on Lancaster-Oslo/Bergen (LOB) corpus that comprises of 1 million word instances and a database with 1066 forms by 400 writers, which is a total of 82227 word instances out

of 10841 words in the vocabulary. The forms are segmented into lines and words using image-processing procedures.

[7] Using a multi-state time delay neural network, a recognition engine named NPen++ was developed at the University of Karlsruhe and Carnegie Mellon University that gives recognition rates from 96% for a 5,000 word dictionary to 93.4% on a 20,000 word dictionary and 91.2% for a 50,000 word dictionary. This engine can be run with large dictionaries as tree search and pruning techniques reduces the search space without attenuating the recognition performance.

[8] Recurrent neural networks (RNNs) with Long Short-Term memory cells gives best results in unconstrained handwriting recognition and dropout approach gives superior performance in convolutional networks. This paper has used both these techniques for the very first time in such a way that dropout doesn't affect the recurrent connections so that modeling sequences are preserved.

[9] It is an offline Roman cursive handwriting recognition system where the input is provided in form of a digit or a word or text. After a number of processing procedures, the ASCII format of the input is obtained as the output.

[10]This paper is a synopsis of the Markov-model that is widely used for offline handwriting recognition. Markov-chain or n-gram models and the crucial concepts about the hidden Markov-model(HMM) are discussed.

[11]Using a hidden Markov Model(HMM) continuous speech recognition system is applied to an online cursive handwriting recognition. A 1.1% word error rate is achieved for a 3050 word lexicon, 52 character, writer-dependent task and 3%-5% word error rates are obtained for six different writers in a 25,595 word lexicon, 86 character, writer-dependent task. This result was obtained by simply using the handwriting ingredients instead of speech.

[12] A database based on Lancaster-Oslo/Bergen(LOB) corpus, which consists of English sentences that includes 556 forms by 250 writers are used for off-line handwriting recognition. This paper focuses on using the linguistic knowledge over the lexicon level. These forms undergoes Preprocessing and text segmentation steps for obtaining the promising result.

[13] Least squares SVM (LS-SVM) which is easier to train as it requires only the solution to a convex linear problem, is based on the margin-maximization principle performing structural risk minimization. This paper discusses about the model selection for the LS-SVM using an empirical error criterion. This classifier is used in handwritten character recognition and also greatly demonstrates how a model selection can Improve the performance.

[14] This paper discusses about the character termination using a finite state automaton where an unknown stroke is compared with the database of strokes to match and identify the stroke. This system is used in online recognition of handwritten Tamil characters as a handwritten character is constructed via sequence of strokes.

[15] This paper discusses about the system architecture and novel components that are used on powerful machines for online handwriting recognition system in the cloud as well as on mobile devices with a bit of changes in the settings of the system. Other components such as unified time and position-based input interpretation, trainable segmentation, minimum-error rate training for feature combination, and a cascade of pruning strategies are used in this system, which currently supports 22 scripts and 97 languages. This system focuses most on the fast, touch-enabled devices with high-accuracy text entry for mobile.

[16] On binary images, a set of robust language independent features are extracted and then, lower and upper baselines subsets are derived so that the word variability will be assessed.This approach of offline handwriting recognition system using 1D HMM learns the model without even undergoing the pre-segmentation process.

[17] This paper promises that their HMM-based system for offline handwriting recognition outperforms the current state-of-the-art approach for English and French writings using sturdy methods like image object recognition: moment-based image normalization, writer adaptation, discriminative feature extraction and training, and open-vocabulary recognition.

[18] This paper uses a novel technology for 3D handwriting recognition where accelerometers and gyroscope are attached to the users' back of the hands. When the user writes in the air, these gestures are captured by the motion sensors for spotting and recognizing those gestures in further stages. A Support vector machine(SVM) is used to identify the data segments whereas the Hidden Markov models(HMM) are used to generate a text representation from the captures gestures. These individual characters are concatenated to word models.

[19] A two-pass N-best approach is taken on several UNIPEN data sets for a writer independent handwriting recognition system. This system is robust because of the combination of signal normalization preprocessing and other perpetual features. The accuracy is improved due to the melding of point oriented and stroke oriented features.

[20] This paper discuss about comparing and understanding the eastern(Japanese) and western character recognition like the preprocessing, classification and post-processing stages. This lets us understand the similar and non-similar foundation, and to develop the compact modules for integrated systems that recognizes multi-language writings.

[21] This paper compares the two strategies of SVM i.e., "one against one" and "one against all" to find out the difference and the significance. It appears that "one against all" strategy is more significant for digit recognition whereas "one against one" is comparatively easier to train and can be used for a very large number of classes. But when it comes to upper-case letters, there is a very less difference between these two strategies. Also when compared to MLP, SVM's strategies shows a better estimation than MLP.

[22] A bidirectional recurrent neural network with long short-term memory blocks approach is used alongside an objective function called Connectionist Temporal Classification (CTC)for online handwriting recognition. This system achieves 74.0% of word recognition while a system build based on HMM gives 65.4% of word recognition.

[23] This system outperforms all the other systems for the recognition of French handwriting based on three different recognition technologies.These technologies includes a framework of multi-word recognition based on weighted finite state transducers, an explicit word segmentation, and a combination of isolated word recognizes and a language model.

[24] This paper discusses about the application of neuroevolution and all the way till CNN topologies and therefore developing solutions based on genetic algorithms and grammatical evolution. MNIST data set consists of handwritten digits that are difficult even for humans to recognize. In spite of this, the system has produced an impressive result without any kind of data augmentation or preprocessing methods.

[25] Feature selection-based combination and class dependent features are used for handwriting recognition in this paper. Based upon their class separation and recognition capabilities, features are

evaluated. Also new feature vector is constructed using multiple feature vectors. The results the new feature vector helps in upgrading the recognition rate and also the selected features are effective in separating pattern classes.

[26] This paper aims at discussing about the problems in handwritten symbols, new neural network model(NNM). Two subsystems of neural networks are used particularly for gender classification for which, the bio-metrical system for Automated Factographic Information Retrieval System (AFIRS) is also designed and developed.

[27] CASIA-HWDB/OLHWDB databases are used as the trained set for the following five tasks- classification on extracted features, online/offline isolated character recognition, online/offline handwritten text recognition and obtained 93.89%, 94.77%, 97.39%, 88.76%, and 95.03% from test results respectively .

[28] For a modified CNN-RNN hybrid architecture, training is done using efficient initialization of network using synthetic data, image normalization for slant correction, and domain specific data transformation and distortion for learning the in-variants. This proposal is for recognizing the handwriting for scanned offline document images.

[29] Writing styles can vary from self-combined to various other styles. This may cause problems and that is why Genetic algorithms are proposed. A pool of images of the characters are converted into images. Graph of each character is blended so that an intermediate styles of the parent character is obtained. This method has achieved an accuracy of 98.44%.

[30] A deep neural network architecture that supports for 102 languages, has put back the segment-and-decode-based system by reducing the error rate by 20–40%. This system recognizes 10 times faster than previous system as it combines the sequence recognition with a new input encoding using Bézier curves. This paper discloses that for both the open and closed data sets,new state-of-the-art results on IAM-OnDB.

HARDWARE AND SOFTWARE REQUIREMENTS:

Hardware requirements:

1)Windows 10 with Intel Core 3 Duo/Quad/hex/Octa or higher end 64 bit processor PC or Laptop (Minimum operating frequency of 2.5GHz)

2) Hard Disk capacity of 500GB - 4TB with 4 GB RAM or above.

3) 10 Gigabit Ethernet or Bonded Gigabit Ethernet

Software requirements:

Functional requirements:

- Modified National Institute of Standards and Technology(MNIST) dataset is imported and is saved in the same directory as the program.
- A model is created. This model takes the value of the color in pixels and puts them in a one-dimensional array. A Neural network is built and activated in such a way that the number of nodes in the input layer is equal to the number of pixels in the arrays.
- From the testing set, accuracy is tested.

- For recognizing the handwritten digit, user has to input the value on an image of dimension 28 x 28 then, the value of the number written on the image is predicted in real-time.

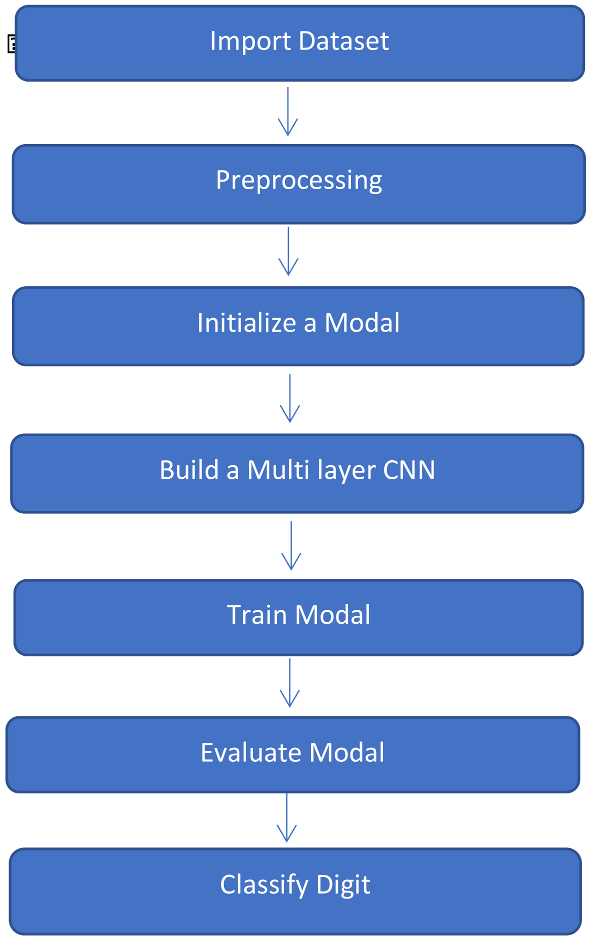
Non-functional requirements:

- Python
- Numpy (Library to handle complex Array operations)
- Keras (Machine learning Library)
- Tensorflow (Machine learning Library)

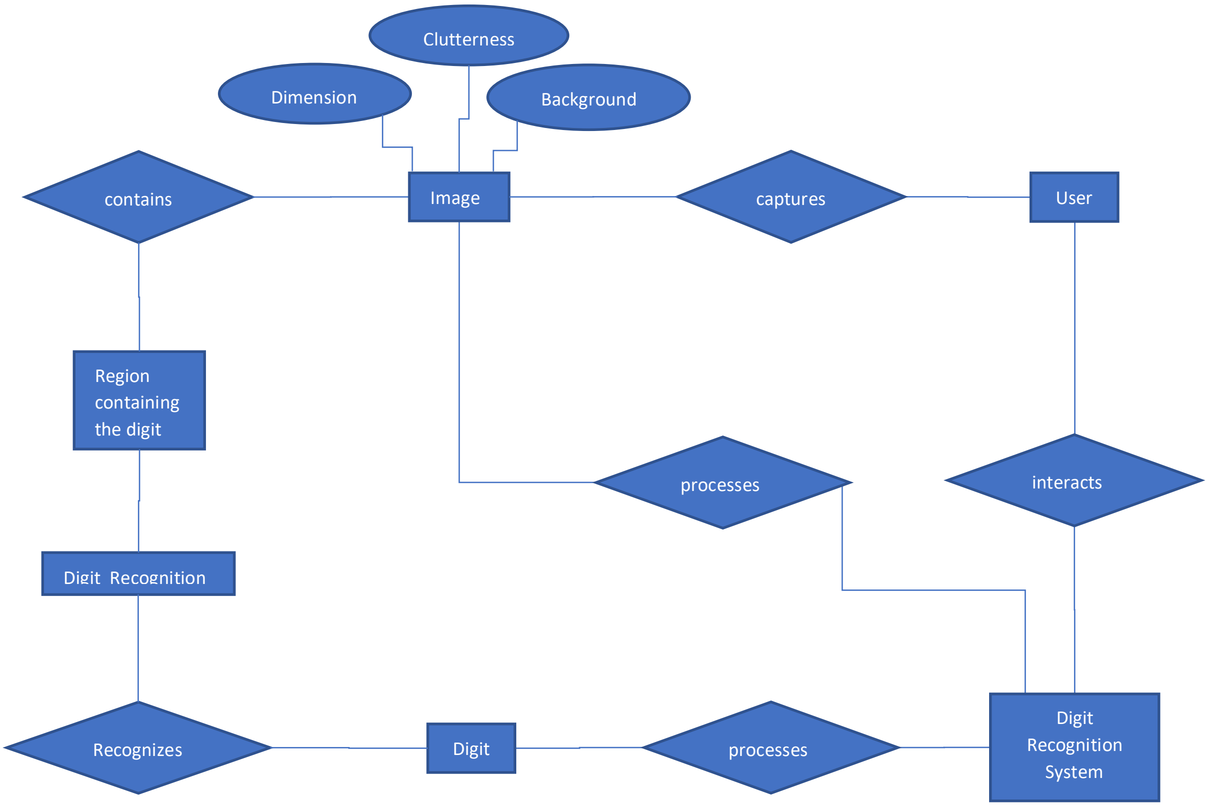
System constraints:
System takes no part in server error because the network provider can shut down the network for small maintenance work.

DETAILED DESIGN:

System Architecture:



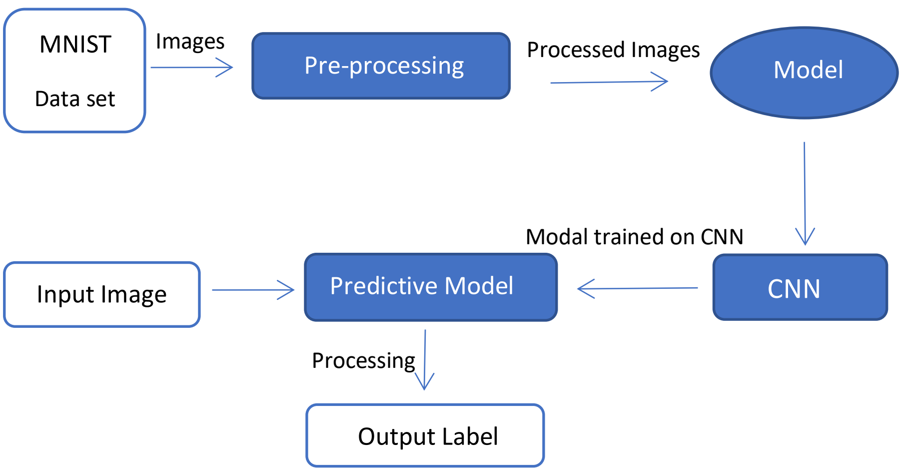
ER diagram:



Data Flow Diagram(DFD):

Level 1:

Level 2:



Module description:

Data set:

Handwritten character recognition is a research area that already contains detailed ways of implementation which include major learning datasets. MNIST data set (Modified National Institute of Standards and Technology database) is the subset of the NIST data set. MNIST is a combination of two of NIST's databases Special Database 1 and Special Database 3, which consist of digits written by high school students and employees of the United States Census Bureau, respectively. MNIST has a total of 70,000 handwritten digit images, out of which 60,000 are used for training set and 10,000 are used for test set. Each image has a dimension of 28x28 pixel and are anti-aliased. All these images have corresponding label values which tells the value of the digit.

Convolutional Neural Network(CNN):

Convolutional Neural Network is a subset of Deep learning, which is commonly used for Image recognition and classification. It is a class of deep neural networks that require minimum pre-processing. It takes the image as an input in the form of small chunks rather than taking one pixel at a time, so the network can detect irregular patterns in the image more efficiently. CNN contains 3 layers namely, an input layer, an output layer, and multiple hidden layers. Hidden layers include three types of layers:

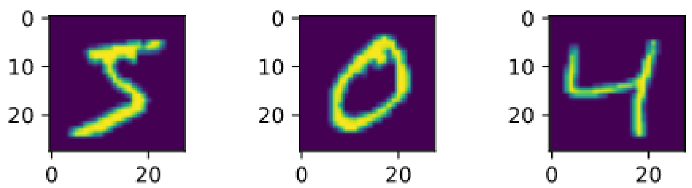
1. Convolutional layer - This layer uses a filter (also called as kernel) which is an array of weights to extract features from the input image. One layer can have many filters.

2. Pooling layer - This layer reduces the dimensions of the data coming from Convolutional layer which in return reduces the computations, number of parameters, reduces over fitting and therefore making the entire process much faster. Typically, the pooling layer is inserted between convolutional layers. It discards the activation of previous layers and hence forcing the next convolutional layers to learn from a limited variety of data

3. Fully connected layer - Output of previous layers is flattened and sent into this layer, which then uses that too classify the image into label.

Data Visualization:

Implementation:



Sample images from MNIST Data set

Data Preprocessing **:**

Initially, the date set is separated into four different categories namely trainX, trainY, testX, testY such that X represents the feature and Y represents the label. Since a digit always comes within a category of 0 to 9, the label can be converted from a simple digit to a array of size 10 where the index which represents the digit is given 1 and rest of the indexes are given a value of 0. For example 5 would be represented as [0,0,0,0,0,1,0,0,0,0]. Then a normalization process is done ,where each pixel is converted from a value range of 0 to 255 to value range of 0 to 1. Pixels are converted into float type and then divided by 255.

IMPLEMENTATION:

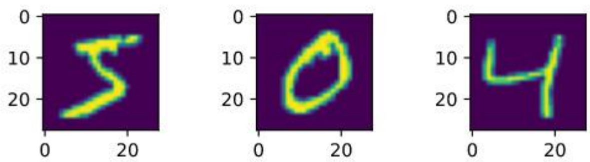
```
from tensorflow.keras.datasets import mnist
from matplotlib import pyplot
# Data set is Downloaded from Internet
(trainX, trainY), (testX, testY) = mnist.load_data()
print(f'Training: ')
print(f'x: ',trainX.shape ,end='\t')
print(f'y: ',trainY.shape)
print(f'Testing: ')
print(f'x: ',testX.shape,end='\t')
print(f'y: ',testY.shape)
# Plot first few Training Images
for i in range(3):
	pyplot.subplot(330 + 1 + i)
	pyplot.imshow(trainX[i])   # cmap=pyplot.get_cmap('gray') --> to make preview as grey
scale
pyplot.show()
for i in range(3):
	print(trainY[i], end=" ")
```

```
Training:
x:  (60000, 28, 28)     y:  (60000,)
Testing:
x:  (10000, 28, 28)     y:  (10000,)
```

```
from tensorflow.keras.utils import to_categorical
# reshape data set to have a single channel
trainX = trainX.reshape((trainX.shape[0], 28, 28, 1))
testX = testX.reshape((testX.shape[0], 28, 28, 1))
trainY = to_categorical(trainY)
testY = to_categorical(testY)
```

```python
from tensorflow.keras.datasets import mnist
from matplotlib import pyplot

# Dataset is Downloaded from Internet
(trainX, trainY), (testX, testY) = mnist.load_data()

print(f'Training: ')
print(f'x: ',trainX.shape ,end='\t')
print(f'y: ',trainY.shape)

print(f'Testing: ')
print(f'x: ',testX.shape,end='\t')
print(f'y: ',testY.shape)

# Plot first few Training Images
for i in range(3):
        pyplot.subplot(330 + 1 + i)
        pyplot.imshow(trainX[i])     #  cmap=pyplot.get_cmap('gray') --> to make preview as grey scale
pyplot.show()

for i in range(3):
        print(trainY[i], end=" ")
```

```
Training:
x:  (60000, 28, 28)      y:  (60000,)
Testing:
x:  (10000, 28, 28)      y:  (10000,)
```




```
5 0 4
```

```python
from tensorflow.keras.utils import to_categorical

# reshape dataset to have a single channel
trainX = trainX.reshape((trainX.shape[0], 28, 28, 1))
testX = testX.reshape((testX.shape[0], 28, 28, 1))

trainY = to_categorical(trainY)
testY = to_categorical(testY)
```

```
[[  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   3  18  18  18 126 136 175  26 166 255 247 127   0   0   0   0]
 [  0   0   0   0   0   0   0   0  30  36  94 154 170 253 253 253 253 253 225 172 253 242 195  64   0   0   0   0]
 [  0   0   0   0   0   0   0  49 238 253 253 253 253 253 253 253 253 251  93  82  82  56  39   0   0   0   0   0]
 [  0   0   0   0   0   0   0  18 219 253 253 253 253 253 198 182 247 241   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0  80 156 107 253 253 205  11   0  43 154   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0  14   1 154 253  90   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0 139 253 190   2   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0  11 190 253  70   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0  35 241 225 160 108   1   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0  81 240 253 253 119  25   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0  45 186 253 253 150  27   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0  16  93 252 253 187   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0 249 253 249  64   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0  46 130 183 253 253 207   2   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0  39 148 229 253 253 253 250 182   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0  24 114 221 253 253 253 253 201  78   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0  23  66 213 253 253 253 253 198  81   2   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0  18 171 219 253 253 253 253 195  80   9   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0  55 172 226 253 253 253 253 244 133  11   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0 136 253 253 253 212 135 132  16   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]]
```

REFERENCES:

[1] Tappert, C. C., Suen, C. Y., & Wakahara, T. (1990). The state of the art in online handwriting recognition. *IEEE Transactions on pattern analysis and machine intelligence*, *12*(8), 787-808.

[2] Xu, L., Krzyzak, A., & Suen, C. Y. (1992). Methods of combining multiple classifiers and their applications to handwriting recognition. *IEEE transactions on systems, man, and cybernetics*, *22*(3), 418-435.

[3] Graves, A., Liwicki, M., Fernández, S., Bertolami, R., Bunke, H., & Schmidhuber, J. (2008). A novel connectionist system for unconstrained handwriting recognition. *IEEE transactions on pattern analysis and machine intelligence*, *31*(5), 855-868.

[4] Plamondon, R., & Srihari, S. N. (2000). Online and off-line handwriting recognition: a comprehensive survey. *IEEE Transactions on pattern analysis and machine intelligence*, *22*(1), 63-84.

[5] Hu, J., Brown, M. K., & Turin, W. (1996). HMM based online handwriting recognition. *IEEE Transactions on pattern analysis and machine intelligence*, *18*(10), 1039-1045.

[6] Marti, U. V., & Bunke, H. (2002). The IAM-database: an English sentence database for offline handwriting recognition. *International Journal on Document Analysis and Recognition*, *5*(1), 39-46.

[7] Jaeger, S., Manke, S., Reichert, J., & Waibel, A. (2001). Online handwriting recognition: the NPen++ recognizer. *International Journal on Document Analysis and Recognition*, *3*(3), 169-180.

[8] Pham, V., Bluche, T., Kermorvant, C., & Louradour, J. (2014, September). Dropout improves recurrent neural networks for handwriting recognition. In *2014 14th international conference on frontiers in handwriting recognition* (pp. 285-290). IEEE.

[9] Bunke, H. (2003, August). Recognition of cursive Roman handwriting: past, present and future. In *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings.* (pp. 448-459). IEEE.

[10] Plötz, T., & Fink, G. A. (2009). Markov models for offline handwriting recognition: a survey. *International Journal on Document Analysis and Recognition (IJDAR)*, *12*(4), 269.

[11] Starner, T., Makhoul, J., Schwartz, R., & Chou, G. (1994, April). On-line cursive handwriting recognition using speech recognition methods. In *Proceedings of ICASSP'94. IEEE International Conference on Acoustics, Speech and Signal Processing* (pp. V-125). IEEE.

[12] Marti, U. V., & Bunke, H. (1999, September). A full English sentence database for off-line handwriting recognition. In *Proceedings of the Fifth International Conference on Document Analysis and Recognition. ICDAR'99 (Cat. No. PR00318)* (pp. 705-708). IEEE.

[13] Adankon, M. M., & Cheriet, M. (2009). Model selection for the LS-SVM. Application to handwriting recognition. *Pattern Recognition*, *42*(12), 3264-3270.

[14] Aparna, K. H., Subramanian, V., Kasirajan, M., Prakash, G. V., Chakravarthy, V. S., & Madhvanath, S. (2004, October). Online handwriting recognition for Tamil. In *Ninth international workshop on frontiers in handwriting recognition* (pp. 438-443). IEEE.

[15] Keysers, D., Deselaers, T., Rowley, H. A., Wang, L. L., & Carbune, V. (2016). Multi-language online handwriting recognition. *IEEE transactions on pattern analysis and machine intelligence*, *39*(6), 1180-1194.

[16] El-Hajj, R., Likforman-Sulem, L., & Mokbel, C. (2005, August). Arabic handwriting recognition using baseline dependant features and hidden markov modeling. In *Eighth International Conference on Document Analysis and Recognition (ICDAR'05)* (pp. 893-897). IEEE.

[17] Kozielski, M., Doetsch, P., & Ney, H. (2013, August). Improvements in rwth's system for off-line handwriting recognition. In *2013 12th International Conference on Document Analysis and Recognition* (pp. 935-939). IEEE.

[18] Amma, C., Georgi, M. & Schultz, T. Airwriting: a wearable handwriting recognition system. *Pers Ubiquit Comput* 18, 191–203 (2014).

[19] Jianying Hu, Sok Gek Lim, Michael K. Brown,Writer independent on-line handwriting recognition using an HMM approach, Pattern Recognition.

[20] Jäger, S., Liu, C. L., & Nakagawa, M. (2003). The state of the art in Japanese online handwriting recognition compared to techniques in western handwriting recognition. *Document Analysis and Recognition*, *6*(2), 75-88.

[21] Milgram, J., Cheriet, M., & Sabourin, R. (2006, October). "One against one" or "one against all": Which one is better for handwriting recognition with SVMs?. In *tenth international workshop on Frontiers in handwriting recognition*. Suvisoft.

[22] Liwicki, M., Graves, A., Fernàndez, S., Bunke, H., & Schmidhuber, J. (2007). A novel approach to on-line handwriting recognition based on bidirectional long short-term memory networks. In *Proceedings of the 9th International Conference on Document Analysis and Recognition, ICDAR 2007*.

[23] Menasri, F., Louradour, J., Bianne-Bernard, A. L., & Kermorvant, C. (2012, January). The A2iA French handwriting recognition system at the Rimes-ICDAR2011 competition. In *Document Recognition and Retrieval XIX* (Vol. 8297, p. 82970Y). International Society for Optics and Photonics.

[24] Baldominos, A., Saez, Y., & Isasi, P. (2018). Evolutionary convolutional neural networks: An application to handwriting recognition. *Neurocomputing*, *283*, 38-52.

[25] Oh, I. S., Lee, J. S., & Suen, C. Y. (1999). Analysis of class separation and combination of class-dependent features for handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *21*(10), 1089-1094.

[26] Kulik, S. D. (2015). NEURAL NETWORK MODEL OF ARTIFICIAL INTELLIGENCE FOR HANDWRITING RECOGNITION. *Journal of Theoretical & Applied Information Technology*, *73*(2).

[27] Yin, F., Wang, Q. F., Zhang, X. Y., & Liu, C. L. (2013, August). ICDAR 2013 Chinese handwriting recognition competition. In *2013 12th international conference on document analysis and recognition* (pp. 1464-1470). IEEE.

[28] Dutta, K., Krishnan, P., Mathew, M., & Jawahar, C. V. (2018, August). Improving cnn-rnn hybrid networks for handwriting recognition. In *2018 16th international conference on frontiers in handwriting recognition (ICFHR)* (pp. 80-85). IEEE.

[29] Kala, R., Vazirani, H., Shukla, A., & Tiwari, R. (2010). Offline handwriting recognition using genetic algorithm. *arXiv preprint arXiv:1004.3257*.

[30] Carbune, V., Gonnet, P., Deselaers, T., Rowley, H. A., Daryin, A., Calvo, M., ... & Gervais, P. (2020). Fast multi-language LSTM-based online handwriting recognition. *International Journal on Document Analysis and Recognition (IJDAR)*, 1-14.