HANDWRITING DETECTION

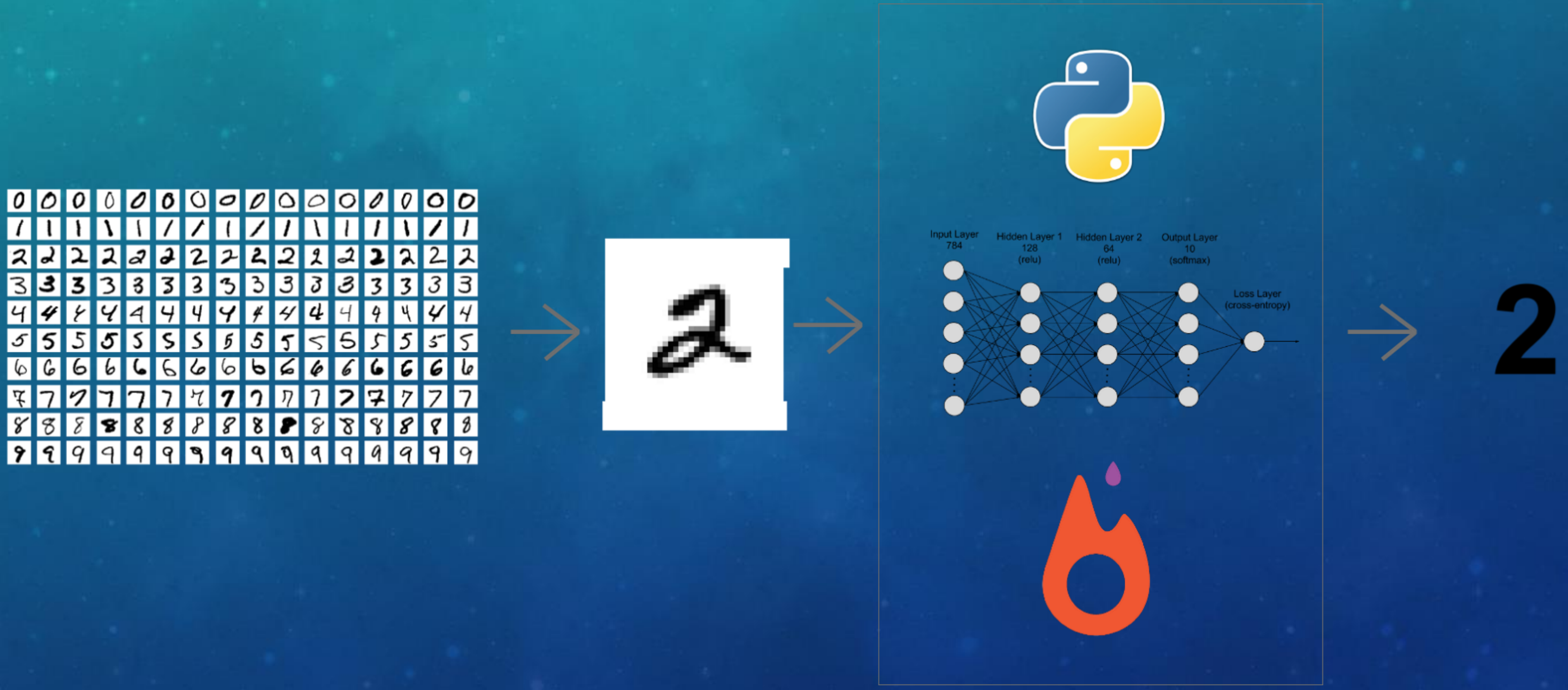
TEAM MEMBERS

- Akshat Veerwal E19CSE197
- Shad Rais E19CSE117
- Srinivas Abhishek Pratapa E19CSE037
- Suchintan Das E19CSE165

INTRODUCTION

Creating a soft copy from the hard copy is one of the major problems that the current generation is facing when it comes to technologies.

Our system would convert the people’s handwriting to digital alphabets which can be stored or used as per the user’s choice.



OpenCV
HELLO
WORLD
Keras
TensorFlow
2.0

RELATED WORKS

- **Handwritten character Recognition using Neural Network: This project's purpose was to take handwritten characters that are scanned, processed, and a neural network algorithm that is trained with them which would then find patterns, and then change the character . The goal of this research was to create software that would recognise the characters.**
- **Recognition of Handwritten Characters Using Gradient Features : Any recognition system must include feature extraction as a component. The purpose of feature extraction was to determine the pattern's characteristics. The gradient depicts the extent and direction of the most significant intensity shift in a small area surrounding each pixel. An attempt was made to recognise characters, and the recognition accuracy was found to be 94 percent. Gradient Features were utilised for recognition because of the simplicity of the logic, convenience of use, and high recognition rate.**
- **Matlab's Neural Network Toolbox is used to recognise characters: Recognition of handwritten text has been one of the most active and hard areas of research In the field of image processing and pattern recognition,. It has a wide range of uses, including reading aids for the bank checks, and the conversion of any handwritten document into structured text. Any recognition system must include feature extraction as a component. The purpose of feature extraction was to determine the pattern's characteristics. The first phase is image acquisition, which entails scanning the image, noise filtering, smoothing, and normalisation, and then preparing the image for segmentation, which entails dividing the image into sub images. Feature Extraction increases the rate of recognition and reduces misclassification.**
- **Character recognition utilising a character geometry-based feature extraction approach: This work describes a feature extraction technique that is based on geometry that can be used in segmentation-based word recognition systems. The suggested system extracts the character contour's geometric features. These properties are based on the character skeleton's basic line types. The system's output is a set of feature vectors. The feature vectors obtained from a training set were then utilised to train a Neural Network-based pattern recognition engine, allowing the system to be benchmarked.**

PROPOSED METHODOLOGY

- As per the project need, we were required for a highly populated dataset to train our model because this branch of detection is majorly based on good testing of the model on a good dataset.

- Firstly, we split the data into two halves, one was for the labels and another one for the images. Based on the labels of the data like L we read the whole y data for getting the image. For this thing we are using the split function that is inbuilt csv. After successful splitting of data we reshaped the data based on the training x values to make the data ready to be displayed as an image which we would be giving to the "KERAS" module of python. During this process we shape both our testing and training data on the same constraints itself. Initially in the dataset that we are using we have 784 columns of pixel data which can be converted to a 28x28 pixels. The values on the dataset are in floating point . But the issue is that we can't use floating point values on the main function, we need an integer values so we created a loop to map the integer values with the characters of the dataset.

- Now to visualize the data we needed some form of graphical representation. Bar graphs are widely used for visualizing such types of data, because we need to check the number of alphabet and their integer values traffic in short in the dataset. It's important to know the traffic of a character because no dataset is perfect. If we need to know that why a model is not working fine with some characters in the near future we need to keep an eye on the training part of the model. By knowing the traffic of a character in a dataset we get an idea that the model is weak when it comes to some specific characters because the traffic of those characters is very less. This is very useful in scalability because if a model has not received enough training and testing in a certain areas one can use other datasets for better outputs.

- Now we need to shuffle the data to make it ready for training. Shuffling is done direly using the inbuilt function of python. We are not shuffling the testing data as we didn't had the need to shuffle it.

- Keras is now used to change that rough data into an image. The shape of the show function is 3x3 in short there will be 9 plots in one time. Plots are alphabets in this case.

PROPOSED METHODOLOGY

- Now after the training it's necessary to take a summarize the data. To achieve that we used summary function which is an inbuilt function for getting a summary of the whole thing including layers and all. The model is saved using save in-built function.
- After training it was necessary to test the model on the dataset for which I used predict in-built function of keras which let's us get a text recognition.
- The dataset in a whole performed really well we were able to achieve a good accuracy on the model. There were many instances the model really didn't performed that well , due to some lack of data. But most of the alphabets performed well during the testing of the model. Modules that we used for the training ,testing etc. are-

1. Keras (For whole CNN)
2. Numpy (For all graphical representations)
3. Pandas (Same role as Numpy but also used for calculations part)
4. Sklearn (Training and testing of model)
5. Matplotlib (Similar purpose to Pandas)

EXPERIMENTAL RESULTS

- The model works really good with normal alphabets of english and also works with a good precision over all bold alphabets if written clearly in a well manner. The accuracy that is reached highest by the model is 60%-70% when an image is provided. If a person has a unique way of writing the alphabets, it totally depends on the person on a whole as the writing can be categorized between good to worse scale but on a whole if the writing has a decent enough clarity then our model would detect the alphabet correctly which is quite good in terms of performance if we watch from a lens of CNN accuracy. The time taken by the model is actually not that big to make the prediction. It's customized in a decent enough way that a person using the model don't have to wait for a very long time to get the output from the model. On an average, it gives the result under 5 seconds which is good enough speed keeping in consideration the complexity of the project.

PROJECT FINDINGS

- Our model which is based on Convolutional Neural Network(CNN) reached an accuracy of 95% on validation dataset which is really good when compared to other competitors accuracy on their validation dataset. The model achieves an accuracy of 50-60% on user's data which is really not that high but when we keep into the consideration the errors and problems of precision a CNN model faces in today's date ,model got a decent enough score in this field also. If we check other competitors model they also achieved a decent enough accuracy which we got an idea when we tried using those model and gave little bit complex alphabets to recognize by the model. But we also compared our model to a similar project which was based on Matlab and had an accuracy of 97.8% on validation dataset.

CONCLUSIONS

- We created a 6-layer CNN model using tensorflow which was able the read the characters as well numerics with an accuracy of 50%-60%.
- Furthermore, we will be adding more languages to convert between them and help the people read written documents in languages which they are not able to comprehend.

REFERENCES

- ➢ **Chirag I Patel,Ripal Patel, Palak Patel.**
- **https://www.ijser.org/viewPaperDetail.aspx?MAY1114**
- S. Basavaraj Patil, N. V. Subbareddy ‘Neural network based system for script identification in Indian documents’ in Sadhana Vol. 27, Part 1,February 2002, pp.83–97.
- T. V. Ashwin, P. S. Sastry ’A font and size-independent OCR system for printed Kannada documents using support vector machines’ in Sadhana Vol. 27, Part 1, February 2002, pp. 35–58.
- Kavallieratou, E.; Fakotakis, N.; Kokkinakis, G.,’ New algorithms for skewing correction and slant removal on word-level [OCR]’ in Proceedings of ICECS '99.
- Simmon Tanner, “Deciding whether Optical Character Recognition is Feasible”.
- Matthew Ziegler, “Handwritten Numeral Recognition via Neural Networks with Novel Preprocessing Schemes”
- ➢ **AshutoshAggarwal, Rajneesh Rani, RenuDhir**
- **https://www.researchgate.net/publication/280063550_Handwritten_Devanagari_Character_Recognition_Using_Gradient_Features**
- ➢ **Kauleshwar Prasad, Devvrat C. Nigam, AshmikaLakhotiya and DheerenUmre.**
- **https://www.researchgate.net/publication/317174185_Enhancement_of_ANN-Based_Offline_Hand_Written_Character_Recognition_Using_Gradient_and_Geometric_Feature_Extraction_Techniques**
- ➢ **Sheng Wang.**
- **https://www.researchgate.net/publication/355711639_Handwritten_Font_Image_Design_System_Based_on_Deep_Learning_Algorithm**
- ➢ **Hitesh Mohapatra (KL University)**
- ➢ J.Pradeep , E.Srinivasan and S.Himavathi
- https://arxiv.org/ftp/arxiv/papers/1103/1103.0365.pdf
- (PDF) Handwritten Character Recognition (HCR) Using Neural Network (researchgate.net)