Proceedings of TENCON 2018 - 2018 IEEE Region 10 Conference (Jeju, Korea, 28-31 October 2018)

Automatic Handwritten words on Touchscreen to Text file converter

Bidya Debnath, Adrita Anika, Mohammed Abid Abrar, Tanney Chowdhury, Rajat Chakraborty*, Asir Intisar Khan, Shaikh Anowarul Fattah, Celia Shahnaz*
Department of Electrical and Electronic Engineering
Bangladesh University of Engineering and Technology, Dhaka-1205
*Email: celia.shahnaz@gmail.com, rajat.chakraborty156@gmail.com

***Abstract*—This paper proposes a system for converting handwritten words and numbers into a text file. Our system uses A CNN based method to identify letter and digits. This Automatic system requires image preprocessing, classifying into letters and digits and saving the letter into a text file. The system consists of a touchscreen as a user interface, an Arduino board (microcontroller ATmega 2560) and MATLAB. Any type of handwriting is tested with the classifying process; we got 89% accuracy using our own dataset. Both letter and digits can be recognized and converted into text file using this process. The proposed system will lessen the labor of creating electronic documents and provide easy preservation of data**

Keywords—Touchscreen, CNN, Hand-writing recognition, AlexNet

I. INTRODUCTION

Documentation of every data available has become an important element of modern age. These data is preserved in the form of electronic document. This requires a tremendous amount of typing, especially in case of converting a handwritten document into electronic format.

Our purpose is to find a solution to ease the tedious work of typing. We propose a system which will convert handwritten text into electronic document immediately.

The system consists of a microcontroller (Arduino) and a touchscreen. The conversion is done with the help of MATLAB. When something is written on the touchscreen, it is saved as an image. This image is later changed into an alphabet using a handwriting recognition algorithm [1].

A serial communication is established between pc and Arduino through which the writing on the touchpad is extracted. The screen is composed of 320x240 pixels. Any position on the screen is given in (x, y) co-ordinate. These coordinates are read using a MATLAB code. Later, these coordinates are made into an image of the letter written on the screen.

Before going into letter and digit recognition the image taken from the touchscreen is processed to form a clear pattern. The image is resized to match the size of training images. After that an ANN (Artificial neural network) based algorithm is used to find out which letter is written.

Handwriting recognition by machine learning is an interesting field of work nowadays. Some of the effective approaches are CNN method using MATLAB's neural network toolbox [2], multilayer feedforward neural network [3]. There are two types of character recognition, (i) Online (ii) Offline. In the offline method, the pattern is captured as an image, whereas in an online method the pattern is a function of time, pressure, strokes and other physical parameters. As in our proposed system, we transfer what is written in the touchpad into an image, an offline approach is used to recognize pattern [4]

Convolutional neural networks (CNNs) are designed to learn features from data that come in the form of multiple arrays, for example, a color image. In recent days it has gained popularity among academia and industry. When large-scale training datasets are available, CNNs are capable of learning more expressive representations of image data in general object recognition [5]. Required samples for training a neural net should be 10 times the degree of freedom if trained from scratch. However getting that much data to make own dataset may not be easy.

Transfer learning [6, 7] a good choice i.e. to pre-train a CNN on a very large dataset (e.g., ImageNet, which contains 1.2 million natural images with 1000 categories), and then use the pre-trained CNN as an initialization for further fine-tuning [8, 9].

There are many models trained on a large dataset benchmarked by ImageNet classification challenge, such as ResNet, AlexNet, VGGNet [10]

For this purpose, we used pre-trained data model "AlexNet". AlexNet is a convolutional neural network, originally written with CUDA to run with GPU support, which competed in the ImageNet Large Scale Visual Recognition Challenge [11] in 2012. MATLAB has a support package of AlexNet Network model that can be used for various problems. The model is trained on more than a million images and can classify images into 1000 object categories. The network takes an image as input and outputs a label for the object in the image together with the probabilities for each of the object categories.

After loading the pre-trained model last layers that learned task-specific features are replaced with new layers to fine tune the net with our data set's specific features.

Two different datasets 1) MATLAB 2017a digit dataset 2) Our own dataset were used to train the dataset. For our own dataset samples to train the net were few compared to the set needed to train an ImageNet. Therefore Data Augmentation [12] was used on the dataset to increase the number of sample images in the train set. Data Augmentation also helps to reduce over-fitting [13].

Our goal is not only to figure out the letter or digit but also to save it in an electronic document. So recognizing the letter and saving it in a PDF is done simultaneously.

II. HARDWARE

A. ArduinoMega 2560:

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560. It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4

UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller. It has enough processing speed for real-time serial communicating with pc. The board is power from the pc using a USB cable. Same USB cable is used for data transfer via serial communication.

B. Touchscreen (3.2" TFT LCD display module):

This module is used as the interface to write on. The input voltage of this device is 5v, but Voltage of I/O pins are 3.3 v. So it is not directly compatible with Arduino Mega.
It is also an LCD display. Whatever is written on the screen is displayed instantly underneath the writing.
To make the touchscreen functional with Arduino Mega two libraries were needed which were provided by the supplier of the devices.

C. Touch shield:

Connecting the touchscreen directly with Arduino Mega poses a problem as Mega's I/O output is 5v. Connecting touchscreen directly will fry the I/O pins. Therefore, a shield is needed to convert the output I/O voltage of Mega from 5v to 3.3 v. For this purpose, a 3.3 inch 320x240 Touch LCD (c) Shield is used between the Arduino and touchscreen.

III. METHODOLOGY

A flow chart constituting the entire process is presented in Fig.1. the flow chart shows the overall process of the work. Each segment of the chart is discussed later.

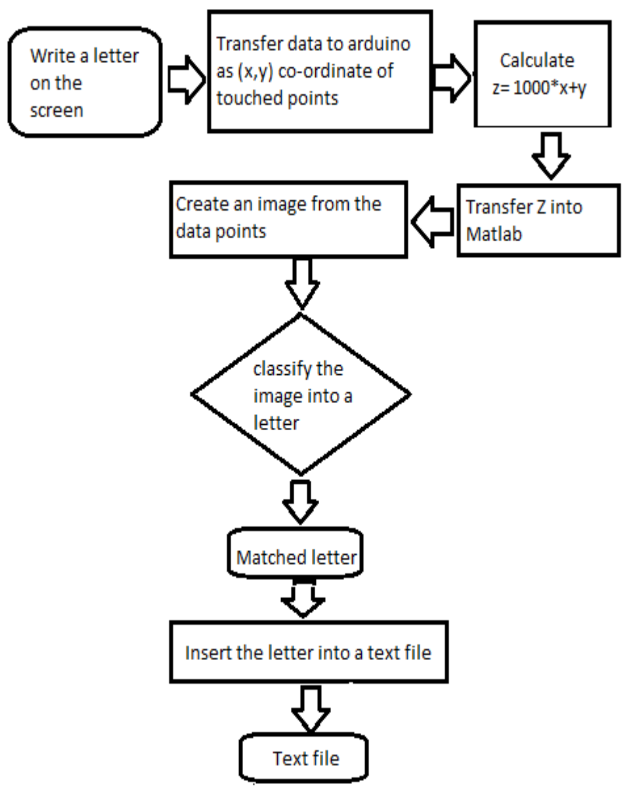


Fig 1: Flow Chart of the working process of the system

A. Calibrating the touchscreen

Before beginning to write anything on the screen the screen needs to be calibrated. If the screen is not properly calibrated the (x, y) coordinate values won't match the points actually touched on the screen. It is a must to use a sharp edge to touch the screen during calibration. Each screen can be different from the other due to slight fabrication variance; therefore calibration parameter values will be different for each frame.

B. Taking the input:

Touchscreen resolution is 320x240. i.e. 320 pixels along the x-axis and 240 pixels along the y-axis. Three buttons are created at the bottom for-

1. Newline
2. Space
3. Enter

The area free to write a letter is taken as 300x220 pixels and a rectangular box is created to mark the area.
When the screen is ready to take input a single letter is written. The whole screen is considered as an XY plane. When
Something is written on the screen (x, y) coordinates of the touched points are sent to the Arduino. The values are sent after the 'Enter' button is pressed. Pressing different buttons sends another unique value along with the coordinates. This unique value is needed for deciding whether to insert a space or a new line or new letter in the desired text file and prepares the screen for a new input. According to our process 1, 2, 3 were sent for 'Enter', 'Space' and 'newline' respectively.

C. Converting the coordinates into an image file:

The x and y coordinates are calculated into one variable for sending into MATLAB. Following equation is used
$Z = 1000*x + y$
After receiving the values in MATLAB, they are transferred back into (x, y) and an image of resolution 300x220. This image is converted into a 227x227x3 resolution for classification using pre-trained Model AlexNet. It is discussed broadly in 'Algorithm' section.

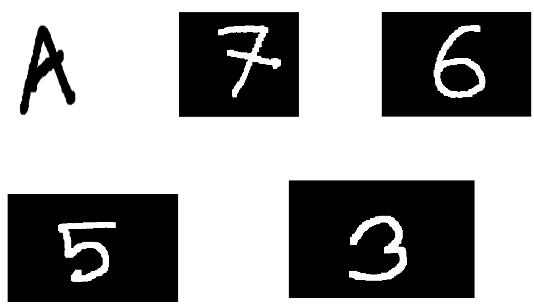


Fig 2: Images processed by MATLAB from the touchscreen co-ordinate values

D. Image processing Using Matlab:

AlexNet, a pre-trained CNN model, is used to initialize the weights of our net. By doing so, our net is able to exploit the rich hierarchy of already learned representations making it achieve a better generalization.

Two different datasets have been used for the training purpose. Firstly, we trained with only digit data that is provided in MATLAB 2017a or later versions. Secondly, we created our own database with images taken from the touchscreen.

In MATLAB, a database for digits that is from 0 to 9 is provided. Those are 28x28 sized black and white images. Buy images of dimension 227x227x3 are suitable for AlexNet. All the images in the database were processed that is resized for the training purpose accordingly.

A testing dataset was created with 15 images for each digit taken from the touchpad. This dataset of 150 images has been tested which did not show a satisfactory result.

It cannot classify digit 4 and 7 properly, but other digits are classified quite well. So, the system is at least robust.

To improve accuracy, we created our database. We took 100 images for each digit and letter from the touchscreen. As the images were taken directly, no processing was required other than dilation to improve the image quality.

Now we tested this network with a testing dataset which was created with 15 images for each digit and letter. This time, it showed a satisfactory result with 89% accuracy. It does not have the trouble of identifying any of the particular digit or letter with the second approach.

E. *Creating a text file:*

When a letter written on the screen is classified as a particular letter or digit our goal is to save it as a text file. This is done using a MATLAB code. Recalling the three buttons on the screen pressing –

1. Newline saves the letter into a new line in the text file
2. 'Space' saves the letter after space.
3. 'Enter' saves the letter immediately after the previous one.

IV. DATASET:

As stated earlier two different datasets were used to train

1. MATLAB Digits dataset
2. Images from touchscreen

A. *Matlab Digits dataset*

This dataset is provided by MATLAB. It only contains digits 0-9. There is a folder for each digit. Each folder contains 1000 images. Each image is of dimension 28x28 pixels.

Fig. 3: Images from digit data set of MATLAB 2017a

These are handwritten letters. A lot of variations are taken i.e. different handwriting and different orientation of letter. This is done so that no matter how the letter is written in the testing image the neural net can recognize it.

B. *Dataset made from Images from touchscreen:*

We made this dataset by taking input from the touchscreen. In this case, we did both letter and digit. The number of the image for each class (26 upper case letters, 10 digits) was 100.

Each image was made 227x227x3 in dimension to match the image size required for AlexNet.

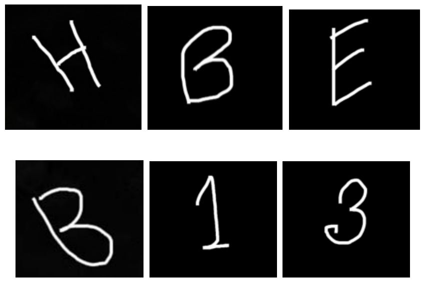


Fig 4: Sample Images of the dataset that has been created using Touchscreen

V. ALGORITHM:

A. *Training*

For both of the datasets, the same procedure was followed to create a train model.

After loading the dataset, it is split into training and testing set randomly,70% training and 30% validation.

Instead of training the model from scratch pre-trained model AlexNet is loaded. This method gives better accuracy than training from scratch.

Last three layers of the AlexNet are replaced with a fully connected layer, a softmax layer, and a classification layer. Learning rate was set at 20%.

For optimizing the neural net using back propagation the hyperparameter called minibatch size was taken as 100. It was the optimal value, taking large mini batch made the procedure slower and taking smaller value made the neural net nosier [14].

The MATLAB digit dataset contains 1000 images, which is sufficient for training a neural net, but our data set is very small compared to that.

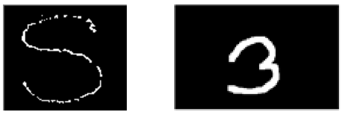
As we used pre-trained model, the necessity of large dataset is reduced. To improve the accuracy, our approach is to generate augmented data before training the classifier. A MATLAB code was run to do this. Data augmentation performs-

1. Rotation
2. Scaling
3. Reflection
4. Shear
5. Translation transformations

On the images of given data set. It helps increase dataset image number as well as prevents overfitting. Training network process uses an augmented image data store to transform training data for each epoch.

B. *Testing*

Before going into testing with the dataset containing images from the touchscreen some pre-processing were required. When one wrote a letter on the screen the points were not connected as we can see in the following image.



Fig, 5: (a) Original letter written on the touchscreen after taking into MATLAB (b) image after a dilation

This created problem while classifying. Therefore we performed dilation on this image to smooth the gaps. Disk dilation was used in this case.

As the classification was done using AlexNet, the testing images had to be resized in 227x227x3. Concatenating a 227x227 image with dimension three made the image 3 dimensional.

After performing dilation and resize the testing data set was put into classification.

VI. TESTS AND RESULTS

Our main objective was to identify correctly which letter is written on the touchscreen. Training with the database provided by MATLAB gave good accuracy but did not perform well enough with the images taken from the touchscreen. Moreover, we could only classify digit s with this data set

Whereas after training the model with our dataset the accuracy percentage became lower as the dataset did not have a large number of data. But performance improved as the training and test set originated from the same source. We could identify both letters and digits this time.

The result is given in the table below.

Table 1: Accuracy of training and test from Given dataset

Model	Dataset	Accuracy
AlexNet	MATLAB Digit dataset	98%
	Images from touchscreen	82%

Table 2: Accuracy of testing new inputs

Training dataset	Test set	Accuracy
MATLAB digits dataset	Image from touchscreen	82%
Images from touchscreen	Image from touchscreen	89%

15 letters were written and converted into images to create the test set.

VII. HUMANITARIAN IMPACT

This system can be further developed to be more user-friendly by using a bigger screen. Therefore input won't be limited to only one letter on screen rather it will serve as a diary. It lessens the trouble of typing.

One can simply write down his/her thoughts on the screen as he/she would write with pen and paper and the text will be preserved as an electronic document for later use. So it will be very helpful when quick typing and documentation of data is needed.

Using this algorithm this proposal can later be turned into a mobile app also. Then there will be no requirement of an external touchscreen.

VIII. CONCLUSION

Our system is developed on an image processing algorithm using CNN (convolutional neural network). To reduce the necessity of huge dataset we used pre-trained data model AlexNet. Most of the Digit data sets are developed from scanning handwritten letters on paper. Those are different from our images. Therefore we created our data set by taking input in the touchscreen, but the number of sample images was few. It was not a practical solution to take 1000 input for each class ourselves. Data augmentation provided a better solution. Using these processes we achieved a good percentage of accuracy. The hardware set up we used was not very high end. We plan to develop our system further so that we can turn this into a handy tool. This system will save a lot of time. Moreover, people will be free of tedious job of typing.

IX. REFERENCES

[1] Impedovo, S. et al. (1991) "Optical character recognition," International Journal of Pattern Recognition and Artificial Intelligence, Vol. 5(1-2), pp. 1-24

[2] R. Arnold and P. Miklós, "Character recognition using neural networks," 2010 11th International Symposium on Computational Intelligence and Informatics (CINTI), Budapest, 2010, pp. 311-314. doi: 10.1109/CINTI.2010.5672225

[3] J. Pradeep, E. Srinivasan and S. Himavathi, "Neural network based handwritten character recognition system without feature extraction," 2011 International Conference on Computer, Communication and Electrical Technology (ICCCET), Tamilnadu, 2011, pp. 40-44. doi: 10.1109/ICCCET.2011.5762513

[4] Rahul Kala, Harsh Vazirani, Anupam Shukla and Ritu Tiwari "Offline Handwriting Recognition using Genetic Algorithm" IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 2, No 1, March 2010 ISSN (Online): 1694-0784 ISSN (Print): 1694-0814

[5] Russakovsky, Olga; Deng, Jia; Su, Hao; Krause, Jonathan; Satheesh, Sanjeev; Ma, Sean; Huang, Zhiheng; Karpathy, Andrej; Khosla, Aditya; Bernstein, Michael; Berg, Alexander C.; Fei-Fei, Li. / ImageNet Large Scale Visual Recognition Challenge. In: International Journal of Computer Vision. 2015; Vol. 115, No. 3. pp. 211-252

[6] H. C. Shin, Holger R. Roth, Mingchen Gao, Le Lu, Ziyue Xu, Isabella Nogues, Jianhua Yao, Daniel Mollura, Ronald M. Summers "Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning," in *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1285-1298, May 2016. doi: 10.1109/TMI.2016.2528162

[7] Jie Lu, Vahid Behbood, Peng Hao, Hua Zuo, Shan Xue, Guangquan Zhang "Transfer learning using computational intelligence: A survey", Knowledge-Based Systems, Elsevier, Volume 80, May 2015, Pages 14-23

[8] Zhang, J.; Xia, Y.; Wu, Q.; Xie, Y. Classification of Medical Images and Illustrations in the Biomedical Literature Using Synergic Deep Learning. Comput. Sci. 2017; arXiv:1706.09092.

[9] Radenović F., Tolias G., Chum O. (2016) "CNN Image Retrieval Learns from BoW: Unsupervised Fine-Tuning with Hard Examples." In: Leibe B., Matas J., Sebe N., Welling M. (eds) Computer Vision – ECCV 2016. ECCV 2016. Lecture Notes in Computer Science, vol 9905. Springer, Cham

[10] Alfredo Canziani, Adam Paszke, Eugenio Culurciello "An Analysis of Deep Neural Network Models for Practical Applications" arXiv:1605.07678

[11] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton,"ImageNet classification with deep convolutional neural networks" NIPS'12 Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1Pages 1097-1105

[12] Perez, Luis & Wang, Jason. (2017). "The Effectiveness of Data Augmentation in Image Classification using Deep Learning." arXiv:1712.03689v1 [cs.CV] 11 Dec 2017

[13] Sebastien C. Wong, Adam Gatt, Victor Stamatescu, Mark D. McDonnell "Understanding data augmentation for classification: when to warp?" arXiv:1609.08764v2 [cs.CV] 26 Nov 2016

[14] Mu Li, Tong Zhang, Yuqian Chen, Alexander j.Smola "Efficient mini-batch training for stochastic optimization" KDD'14 proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining-Pages 661-670 New York, New York, USA- August 24-27,2014 ISBN: 978-1-4503-2956-9 doi> 10.1145/2623330.2623612