International Journal of Engineering & Technology, 7 (3.34) (2018) 114-117

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Medicine Box: Doctor's Prescription Recognition Using Deep Machine Learning

Dr.E.Kamalanaban[1], M.Gopinath[2], S.Premkumar[3]

[1] Professor, Department of Computer Science and Engineering, Vel Tech High Tech Dr.Rangarajan Dr.Sakunthala Engineering College, Avadi, Chennai- 600062. [2]Assistant Professor, Department of Information Techonology, Vel Tech High Tech Dr.Rangarajan Dr.Sakunthala Engineering College, Avadi, Chennai- 600062. [3]PG Scholar, Master of Computer Applications, Vel Tech High Tech Dr.Rangarajan Dr.Sakunthala Engineering College, Avadi, Chennai- 600062.
**Corresponding author E-mail: gopitamil23@gmail.com*

Abstract

A Doctor's prescription is a handwritten document written by doctors in the form of instructions that describes list of drugs for patients in time sickness, injuries and other disability problems. While we receiving a new prescription from doctor, it is unable to understand what drug name is prescribed on it. In most cases, however, we wouldn't be able to read it anyway because doctors use Latin abbreviations and medical terminologies on prescriptions that are not understandable by the general persons which make reading it very difficult. According to the National Academy of Sciences estimates that at least 1.5 million peoples are sickened, injured or killed each year by errors while reading prescription. This paper resolves the problems in doctor's prescriptions through Medicine Box, and Smart phone application that uses Conventional Neural Network (CNN) to recognize handwritten medicine names and return readable digital text. This mobile application uses TensorFlow as the machine learning library, and Custom Repository to match the partial string with the drug name. With Medicine Box, cases of misinterpretation of medicine names can be decreased. This makes the ordinary persons to understand what doctor is prescribed in the prescription and also help for pharmacists.

Keywords: *Conventional Neural Network (CNN), Histogram, Recurrent Neural Network (RNN), Smart phone application, Tensorflow*

.

1. Introduction

Nowadays the usages of Smartphone have been increased rapidly. So, implementation of newer technology on mobile devices make much better to reduce the work load for people. In the day by day human life the medicine and treatment are mandatory factors. In such cases, reading the doctor prescription always a big problem because the doctor prescription is written in the combination of Latin abbreviations and medical terminologies. Normally, prescriptions are only read by other doctors or pharmacists. Ordinary person can't understand what is in the prescription; they had big anxiety while seeing the doctor prescription.

To solve this problem and making the ordinary person to understand the handwriting of doctors prescription and to get knowledge about the medicines a deep machine learning approach through the TensorFlow is used.

TensorFlow is a free software library used for providing immense performance in the computation of numerical data. Flexibility of its architecture makes the computation deployment simple over different platforms like CPU, GPU, TPU and group of servers to mobile and edge devices [14].

TensorFlow is used to build a Conventional Neural Network with the help of collection of dataset. In addition, CNN is used to perform intelligent calculating network for recognizing doctor's prescription [11,15].

To build a CNN, a handwritten dataset called IAM is used. Because IAM is a handwriting database contains the biggest collection English handwriting images. It has 1539 pages of scanned text written by 600+ writers. Only top 200 writers with the most amounts of data are used. It is possible to use the whole dataset but it will increase the training time and size of model.

After building the CNN with dataset, sequence of features are obtained. By the help of features, the LSTM (Long Short Term memory) which is latest type of recurrent neural network for predicting characters is used to predict the character from features [8,16].

2. Methodology

2.1. Data Collection

As mentioned above, IAM dataset is used to sequence the handwriting. First, everyone needs to register to the site and download folder of images and with xml files with position of text in the image. Neural networks don't require much preprocessing of raw data in this project. So it will not make any modifications or corrections to these images however instead of passing the full image to the neural network, only small patches of text are passed.

2.2. Generating Patches of Data

To make the neural network to understand the writing style of doctors instead of passing individual sentences or words, only random patches of text are used. For generating patches from every sentence, cropped images randomly size of 113x113 are used.

Copyright © 2018 Authors. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The image shown below in Fig.1 is a collage of 8 such patches [2,3].

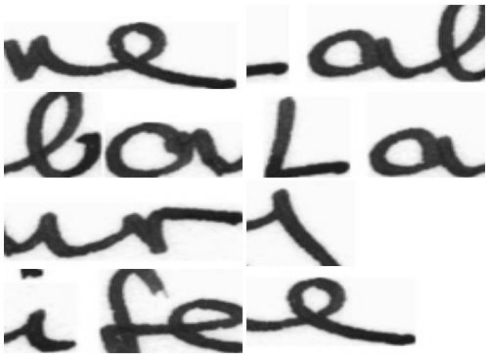


Fig. 1: Patches

With help of python, a simple generator function is created to move over each sentence, which makes cropping and generating random patches of images from it. For every image we will limit the number of patches to 70% of total patches that can be generated [1].

2.3. Predictions with Recurrent Neural Network

Recurrent Neural Network (RNN) is used to predict the next character from images. For this project, the LSTM (Long Short Term memory) cell is used from RNN. For each time one word is processed by LSTM and the forthcoming letter in name of drugs is found by calculating the probabilities of the possible values.
A vector which consists of zeroes initializes the network's memory state. So that after scanning every word it is updated easily[17, 9, 7]. In Fig.2 the 'x' is feature and the 'o' is a vector of probabilities of size length (vocabulary).

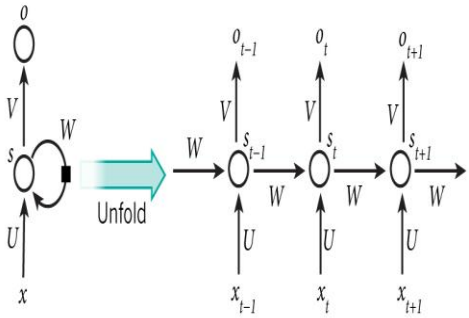


Fig. 2: Simple Recurrent Neural Network

2.4. Creating a Model

Creating a model is the important phase in machine learning approach. For this project, after completing the collection of data and some generator functions with python model is builded. It is important to implement a new environment for this TensorFlow model. Model is nothing but building a CNN in keras with TensorFlow as backend.
A standard CNN is used with multiple convolution and maxpool layers, a few dense layers and a final output layer with softmax activation. RELU activation was used between the convolution and dense layers. The size of the model needs to be proportional to the size of the data[10]. CNN model summary is provided in the table Table.1 below.
After a large bit of hyper parameter tuning it is obtained to a loss of 97% on the test dataset which the model was never exposed to see patches below that the model classified as the same writer. The shape of "t" seems very similar so would also make intuitive sense that they belong to the doctor's writing style.

Table 1: CNN Model Summary

Layer (type)	Output Shape	Param #
zero_padding2d_2 (ZeroPaddin	(None, 115, 115, 1)	0
lambda_2 (Lambda)	(None, 56, 56, 1)	0
conv1 (Conv2D)	(None, 28, 28, 32)	832
activation_7 (Activation)	(None, 28, 28, 32)	0
pool1 (MaxPooling2D)	(None, 14, 14, 32)	0
conv2 (Conv2D)	(None, 14, 14, 64)	18496
activation_8 (Activation)	(None, 14, 14, 64)	0
pool2 (MaxPooling2D)	(None, 7, 7, 64)	0
conv3 (Conv2D)	(None, 7, 7, 128)	73856
activation_9 (Activation)	(None, 7, 7, 128)	0
pool3 (MaxPooling2D)	(None, 3, 3, 128)	0
flatten_2 (Flatten)	(None, 1152)	0
dropout_4 (Dropout)	(None, 1152)	0
dense1 (Dense)	(None, 512)	590336
activation_10 (Activation)	(None, 512)	0
dropout_5 (Dropout)	(None, 512)	0
dense2 (Dense)	(None, 256)	131328
activation_11 (Activation)	(None, 256)	0
dropout_6 (Dropout)	(None, 256)	0
output (Dense)	(None, 50)	12850
activation_12 (Activation)	(None, 50)	0

Total params: 827,698
Trainable params: 827,698
Non-trainable params: 0

2.5. Training a Model

Training a model is the final stage of machine learning. Training is to make CNN work with more accuracy and high performance. Increasing the training and data will increase the performance of model.In addition, Tensor board is used to virtualize the performance model by graph [4].

2.6. Optimization

Optimization is the method of converting training trained TensorFlow model into light weight portable file so that we can use it for any mobile devices. TensorFlow lite convertor is used to convert the trained TensorFlow model into lite file(.tflite) for mobile usability. This Tensorflow lite file can use for both Android and IOS platform [17].

2.7. Smartphone Application

For smart application conversion, android platform is used because android is flexible and user friendly platform for implementing machine learning in mobile devices. Android platform also provides special operation called Native Development Kit(NDK) like JDK which helps to perform machine learning operation in mobile device.

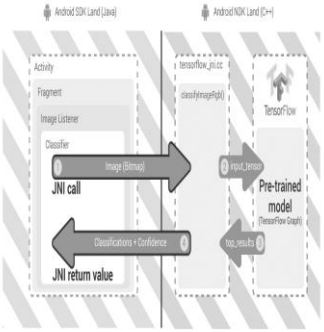


Fig. 3: Architecture Diagram of Android Application

With the model, classifier and image listener have been written which helps to perform computation. When the user opens the

camera through this application, the android NDK runs on C++ which helps to perform the recognition through classifiers [12]. The above figure Fig.3 clearly explains the process.

2.8 Repository for Medicine Name or Drugs Names

After completing the recognition through application, string is obtained as output. To confirm that the string is drug name, custom repository is used which is nothing but a large size of array contains collection drug names [5,6,13].
String matching operation is performed using java string matching concept. Finally with the help of output strings and custom repository, drugs names are validated.

3. Experimental Results

3.1 Training and Testing

Training a model with large amount of data will increase accuracy of the model. As mentioned above, we use all the generated data to train the model. In this TensorFlow, the training falls with two different things like iteration and architecture for training. For this project, we limit the iteration up to 10,000 and use the TensorFlow standard Architecture.
To virtualize the training and testing phrase in graph method we use Tensorboard that provide scenario of training and testing in the graph format. Following images illustrate the training and testing performance of the model through graph.

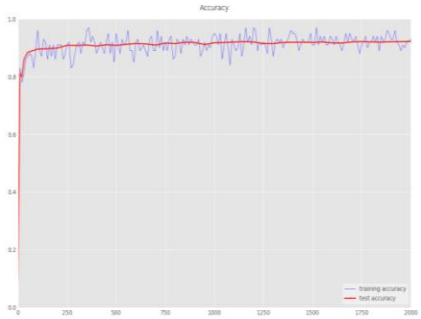


Fig. 4: Accuracy

Accuracy of model during training and testing is illustrated in Fig.4. The red line denotes the test accuracy and blue line denotes the training accuracy of the model. Increasing the dataset collection will increase the accuracy rate.

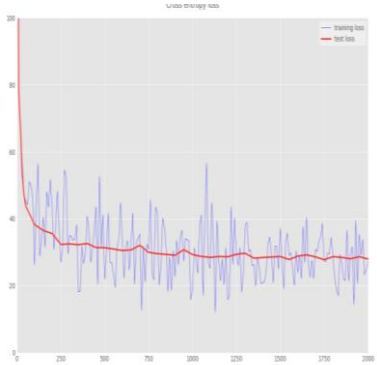


Fig. 5: Cross entropy loss

Cross entropy loss during the test and training time is depicted in Fig.5. Sometimes high loss rate also affects accuracy which should be avoided. The weights and biases of the model after completing the training and test phrases are shown in Fig.6 and Fig.7 respectively. These graphs show the recognition strength of the model.

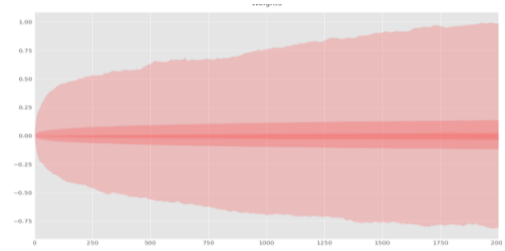


Fig. 6: Weights

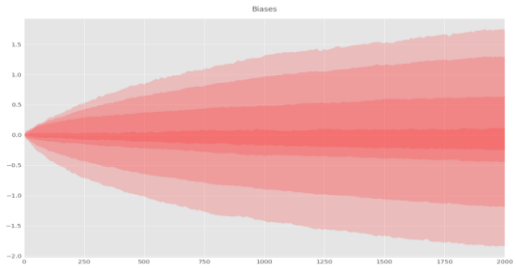


Fig. 7: Biases

3.2. Optimization and Smart Phone Application:

Optimization is the step used in this paper to optimize the model and make it as suitable file to implement in the Smartphone application. Optimizing a trained TensorFlow model make it Lite to implement into the Smartphone application.
Adam optimizer is used to optimize the trained model. The size of the model needs to be proportional to the size of the data. Three blocks of convolution -maxpool layers and couple of dense layers are used. In addition we use TensorFlow Lite Converter to the trained model into Lite weight file (.tflite).
By using Native Development Kit (NDK) we implement this trained model in android application which is most easy platform for smart phone application. Simple classifier is written in java to make the converted ftlite file to work android application[16,17]. Following images illustrate the performance in android application.



Fig. 8: Loading prescription

The working model of the android application is depicted in Fig.8 and Fig.9 where Fig.8 shows the loading of prescription that is needed to scan and Fig.9 shows how the application is recognizing the doctor's handwriting.

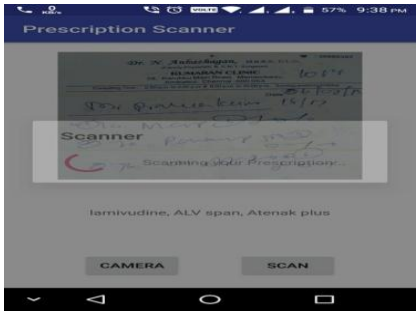


Fig. 9: Recognition of prescription

4. Conclusion

In this paper, we provide a smart way to recognize the doctor's prescription using deep machine learning. By using the Recurrent Neural Network (RNN) neighborhood character is predicted. This application will help for those people who suffer while seeing the doctor's prescriptions. The key role of this paper lies on data used for training purpose. The more accuracy of training data increases the performance of the application. By training the large data with large iteration we can perform the application efficiently.

References

[1] S. Nomura, K. Yamanaka, O. Katai, H. Kawakami, and T. Shiose, "A novel adaptive morphological approach for degraded character image segmentation," Pattern. Recogn., vol. 38, no. 11, pp. 1961–1975, Nov. 2005.

[2] Rehman, and T. Saba, "Performance analysis of character segmentation approach for cursive script recognition on benchmark database," Digit. Signal Process., vol. 21, no. 3, pp. 486-490, May 2011.

[3] Multilingual Character Segmentation and Recognition Schemes for Indian Document Images Parul Sahare1 and Sanjay B. Dhok1.

[4] Text Extraction Using Artifical Neural Network" M UsmanRaza, Ata UllahKhawajaMoyeezUllahGhori,SajjadHaider.

[5] C.Z. Shi, S. Gao, M.T. Liu, C.Z. Qi, C.H. Wang, and B.H. Xiao, "Stroke detector and structure based models for character recognition: A comparative study," IEEE Trans. Image Process., vol. 24, no. 12, pp.4952-4964, Aug. 2015.

[6] Lapuschkin, S., et al., The LRP toolbox for artificial neural networks.Journal of Machine Learning Research, 2016. 17(114): p. 1-5.

[7] N. Nikolaou, M. Makridis, B. Gatos, N. Stamatopoulos, and N. Papamarkos, "Segmentation of historical machine-printed documents using adaptive run length smoothing and skeleton segmentation paths," Image Vision. Comput., vol. 28, no. 4, pp. 590–604, Apr. 2010.

[8] J. Tian, R. Wang, G. Wang, J. Liu, and Y. Xia, "A two-stage character segmentation method for Chinese license plate," Comput.Electr.Eng., vol. 46, pp.539–553, Aug. 2015.

[9] V. Bansal, and R.M.K. Sinha, "Segmentation of touching and fused Devanagari characters," Pattern. Recogn., vol. 35, no. 4, pp.875–893, Apr. 2002.

[10] L. Zheng, A.H. Hassin, and X. Tang, "A new algorithm for machine printed Arabic character segmentation," Pattern. Recogn.Lett., vol. 25, no. 15, pp.1723–1729, Nov. 2004.

[11] R. Manmatha, J.L. Rothfeder, "A scale space approach for automatically segmenting words from historical handwritten documents," IEEE Trans. Pattern Anal. Mach. Intell., vol. 27, no. 8, Aug. 2005.

[12] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna,"Rethinking the Inception Architecture for Computer Vision," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2818-2826.

[13] Rosaly B. Alday and Ruel M. Pagayon MediPic: A Mobile Application for Medical Prescriptions in 2015

[14] TensorFlow Brief reference: https://www.tensorflow.org/

[15] TensorFlow Text classification: https://www.tensorflow.org/tutorials/text_classification_with_tf_hub

[16] Recurrent Neural Nextwork: https://www.tensorflow.org/tutorials/recurrent

[17] Android Application: https://www.tensorflow.org/mobile/ , https://www.tensorflow.org/mobile/android_build