FormAssist : Deep learning methods for converting handwritten forms into digital assets

Srivathshan KS
Senior Data Analyst
PROBYTO
Coimbatore, India
srivathshan@probyto.com

Saurav Kumar
Data Analyst
PROBYTO
Guwahati, India
saurav.k@probyto.com

Shreekanth R
Data Analyst
PROBYTO
Coimbatore, India
shreekanth.r@probyto.com

Midhilesh E
Data Analyst
PROBYTO
Coimbatore, India
midhilesh.e@probyto.com

Parvej Reja Saleh
Senior Data Analyst
PROBYTO
Guwahati, India
parvej@probyto.com

***Abstract*—Customer agreement are required to follow statuary and legal requirements, which include agreements to be manually signed. In India, paper forms are still prevalent in Banking Industry. The paper forms require customers to fill a template form in capital letters and manually sign by agreeing to the terms. This creates challenge in analytical systems as the data is captured outside the system and requires time to become part of data pipeline. The future of banks is poised to be digital, however we still need historical data for train models for current data applications. This limitation is a known bottleneck in designing data applications for real time decision making. Developing Optical Character Recognition (OCR) with capabilities commensurable to that of human is still not achievable, in spite of decades of excruciating research. Due to idiosyncrasy of individual form, analysts from industry and scholastic circles have coordinated their considerations towards OCR. The work in this paper shows an efficient model to capture offline handwritten forms and convert them into digital records. The model techniques are based on deep learning methodologies and show higher accuracy for our testing set of real application forms of selected Banks. We have experimented with different feature extraction techniques to extract hand written characters in the forms. Our experimentation has evolved over time to find a generalized solution and better results. The final model uses relative position of the characters for extracting characters from the forms and Convolutional Neural Networks (CNNs) to predict the characters. The paper also discusses the serverless architecture to host the *FormAssist* as a REST API with model calibration feature to accommodate multiple types of forms.**

***Keywords*—Handwritten Forms, Optical Character Recognition (OCR), Deep Learning, Convolutional Neural Networks (CNNs), Serverless Architecture, REST API**

I. Introduction

Pattern recognition is the science of making inferences from perceptual data based on either a priori knowledge or on statistical information. It is a vital challenge in the field of computer vision and deep learning. It is generally done with feature extraction and classification. The feature extraction regularly utilizes an assortment of techniques to get a portrayal of the information and afterward utilize the classifier to arrange the information. The procedure is led physically and independently [1].

Handwritten Recognition is an area of pattern recognition which characterizes a capacity of a machine to dissect designs and distinguish the character. It has been hailed as a standout amongst the most interesting and testing branch in the field of artificial intelligence and optical character recognition [2]. An assortment of procedures and approaches have been proposed yet it still an uncertain issue. Notwithstanding, it is a testing errand particularly a handwriting recognition on form document. A few issues in handwriting recognition are because of the high ambiguity of the information, as the composed characters of every individual are unique, a few characters have a fundamentally the same as shape, disengaged or bending characters [3].

Optical Character Recognition is a procedure that can change over content, exhibit in computerized picture, to editable content. It enables a machine to perceive characters through optical components. The procedure includes some pre-handling of the picture document and after that obtaining of vital information about composed content. That learning, or information can be utilized to perceive characters [4-8]. OCR comprises of numerous stages, for example, Pre-processing, Normalization, Feature Extraction, Classification and Recognition. The contribution of one stage is the yield of following stage [9]. The errand of preprocessing identifies with the expulsion of commotion and variety in manually written.

OCR problems can be solved with various machine learning algorithms. We here have taken a real-life use case of a sample bank form to apply popular techniques to extract hand written data from it. The numbers from research looks exciting but the real-time scenario is different and more challenging. The first question from a person who needs OCR solution is the scanner type used. The final output depends on lots of factors starting from scanner model to the algorithm selection. we have tried to scrutinize them to choose the most economical and accurate solution. In our attempt to provide a complete solution to a data science problem we have built a user-friendly web portal to read the scanned paper forms and display results.

II. Related Work

Handwritten digit recognition has as of late been of extremely enthusiasm among the scientists on account of the development of different Machine Learning, Deep Learning and Computer Vision algorithms. *Anuj Dutt* and *Aashi Dutt* in their paper (*IJARCET-VOL-6-ISSUE-7-990-997)* [10], analyze the consequences of probably the most broadly utilized Machine Learning Algorithms like SVM, KNN and RFC and with Deep Learning Algorithms like multilayer CNN utilizing Keras with Theano and Tensorflow. Utilizing these, they could get the exactness of 98.70% using CNN (Keras+Theano) when contrasted with 97.91% using SVM, 96.67% using KNN, 96.89% using RFC.

Darmatasia et al. [1] proposes a work flow and a machine learning model for recognizing handwritten characters on form document. Their learning model depends on CNN as a powerful feature extraction and SVM as a high-end classifier. The proposed technique is more effective than altering the CNN with complex architecture. They have assessed some SVM and found that the linear SVM using L1 loss function and L2 regularization giving the best execution both of the accuracy rate and the calculation time. The recognition rate accomplished by the proposed strategy are 98.85% on numeral characters, 93.05% on capitalized characters, 86.21% on lowercase characters, and 91.37% on the merger of numeral and capitalized characters. While the first CNN accomplishes a precision rate of 98.30% on numeral characters, 92.33% on capitalized characters, 83.54% on lowercase characters, and 88.32% on the merger of numeral and capitalized characters. The proposed strategy was additionally validated by utilizing ten folds cross-validation, and it demonstrates that the proposed technique still can enhance the precision rate. The overall system gives an accuracy rate of 83.37% on ten different test form document.

Balci et al. [11] seeks to classify an individual handwritten word so that handwritten text can be translated to a digital form. They have utilized two principle ways to deal with achieve this errand: classifying words directly and character segmentation. For the former, they have used CNN with various architectures to train a model that can precisely classify words. For the latter, they have used LSTM with convolution to build bounding boxes for each character. They at that point pass the segmented characters to a CNN for classification, and after that reconstruct each word as indicated by the results of classification and segmentation.

Choice of classifiers and feature extraction strategies has a prime part in accomplishing most ideal classification exactness in character recognition system. In *Katiyar et al.* [12], an efficient Support Vector Machine based off-line handwritten character recognition system has been developed. It is obvious from the experimental results that the execution of SVM outperforms other state of art techniques.

Pai et al [13], presents basics of OCR method with its parts, for example, pre-processing, Feature Extraction, Classification, post-processing etc. This survey additionally examines distinctive thoughts executed before for recognition of a character.

III. Proposed Work

A leading banking institution account opening application form was considered using economical scanning options starting from a phone camera to a daily use scanner. A regular day to day used Flatbed Scanner has been chosen. The snippet of the scanned copy of the unfilled original form is shown in Fig. 1. As a fresh mind to solve a problem in OCR we started from basic level. Initially started with extracting the characters out from the forms and then applied a machine learning model to predict the characters in the form. Fig. 2 shows the snippet of the filled form.

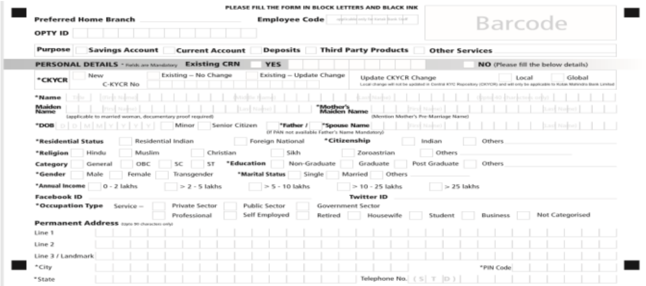

Fig. 1. Snippet of the scanned form

Fig. 2. Snippet of the filled form

IV. Work Architecture

The proposed work architecture is based on Convolutional Neural Networks (CNN) to classify the images. This algorithm takes the image as an input, passes it through a series of Convolutional, Non-linear, Pooling (down-sampling), and fully connected layers, and hence provides an output. The output can be a single class or a probability of classes that describes the image in a better way. A flowchart is shown in Fig. 3 which describes briefly about the entire proposed model.

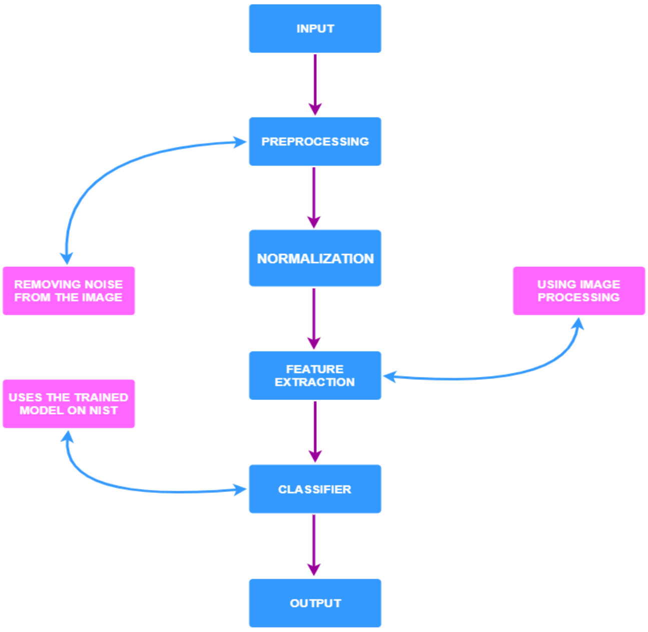

Fig. 3. Flowchart of the FormAssist Model

The weights(w) in this algorithm are updated with equation as stated below.

$$w = w_i - \alpha * (\delta L/\delta w) \quad (1)$$

w = Weight, w_i = Initial Weight, α = Learning Rate

An optimal learning rate was targeted so that the algorithm runs smoothly and fast. L is the loss function and $\delta L/\delta w$ is the derivative of the loss function with respect to w.

A. NIST DATASET

For training the alphabets and numeric models, *'NIST Handprinted Forms and Characters Database'* was used. The numeric model was trained using the MNIST dataset which is a subset of the larger NIST dataset. The MNIST dataset consists of the 10 digits. For training the alphabet model EMNIST dataset was used. Below in Fig. 4, an example of the MNIST dataset is shown.

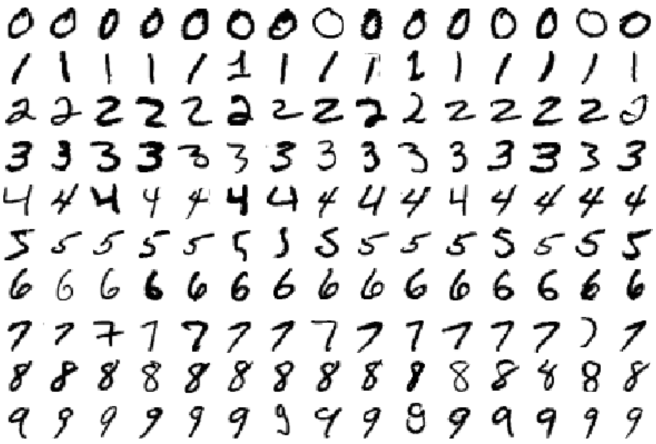

Fig. 4. Example of the MNIST dataset

There were few challenges while training the model. Numpy's *np.loadtxt()* was a very slow method for loading the dataset as compared to pandas' *pd.read_csv()*. Pandas loads the dataset exponentially faster than Numpy.

The task of minimizing the loss involves adjusting the weights so that the loss is minimal. Visually, we want to get to the lowest point in the bowl-shaped loss function as shown in Fig. 5. This figure shows the idea of minimizing the loss. Therefore, the derivatives of the loss function are taken with respect to the weights in the backpropagation step.

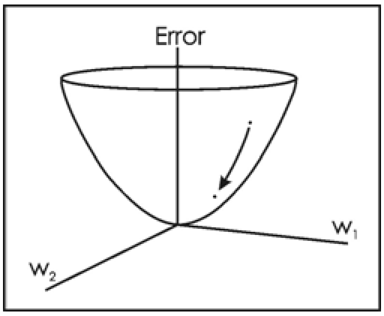

Fig. 5. Optimization algorithm works to get an optimal value of the weights so as to minimize the cost(error).

B. ACTIVATION FUNCTION : ReLU

ReLU (Rectified Linear Units) is used as the activation function for the hidden layers in the network. The function is defined in the below given equation. Fig. 6 shows the graph of the ReLU function.

$$RELU(x) = \begin{cases} 0 \text{ } if \text{ } x < 0 \\ x \text{ } if \text{ } x >= 0 \end{cases}$$

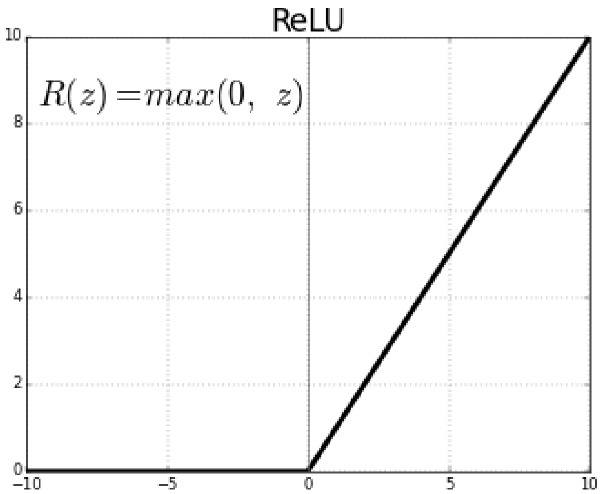

Fig. 6. ReLU function graph

C. DROPOUT

To prevent overfitting in our model, dropout regularization is used. Dropout randomly shuts down some neurons in the network during forward propagation and backward propagation. This ensures that weights are not too large, and the model is not overfitting. In Fig. 7, the dropout process is shown briefly.

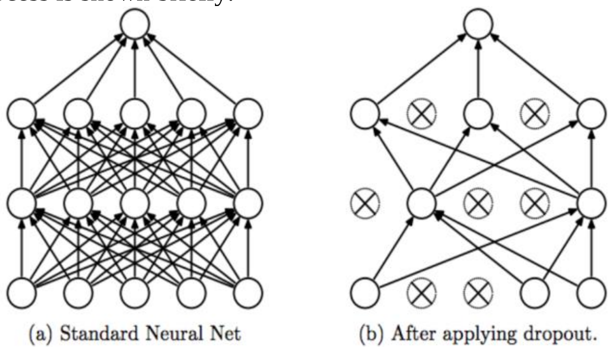

Fig. 7 Example of Dropout process

D. DATA AUGMENTATION

Before training the model, data augmentation is used to make full use of the dataset. In data augmentation, the image is randomly distorted in random manners to create new additional data, which helps to train the model better. In Fig. 8, digit 6 is shown as an example of data augmentation.

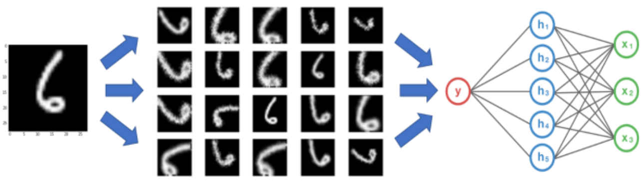

Fig. 8. Example of Data Augmentation of digit 6

E. OPTIMIZATION

Adam is used as the optimization algorithm. It is an extension of Stochastic Gradient Descent and is used to update weights in the neural network by iterating through the dataset. Adam has the combined advantages of optimization algorithms, AdaGrad and RMSprop.

In the article [14], a good representation of Adam algorithm can be seen. The advantages of using Adam on non-convex optimization problems are stated below [14].

- Straightforward to implement
- Computationally efficient
- Little memory requirements
- Invariant to diagonal rescale of the gradients
- Well suited for problems that are large in terms of data and/or parameters
- Appropriate for non-stationary objectives
- Appropriate for problems with very noisy/or sparse gradients
- Hyper-parameters have intuitive interpretation and typically require little tuning.

The performance of various optimization algorithms [15] is shown in Fig. 9. It is clearly seen that, Adam performs the best. While training with Adam Optimization algorithm, we get the lowest cost (the pink line).

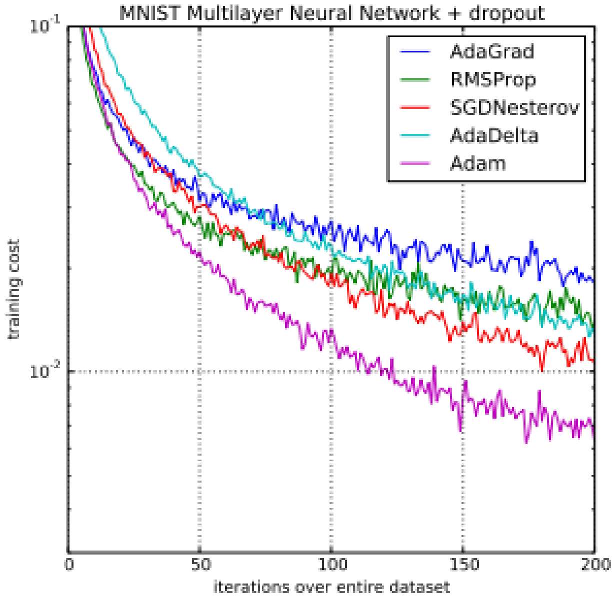

Fig. 9. Comparison of Adam to Other Optimization Algorithms [15]

F. FEATURE EXTRACTION

Feature extraction is one of the most important part in optical character recognition. Here we try to extract the most important features from the image and a good algorithm for feature extraction can significantly improve the accuracy of the model. The feature extraction method is shown in Fig. 10.

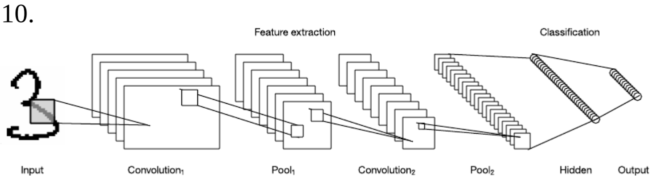

Fig. 10. Example of Feature Extraction

The challenge in this work was the boundaries of box, which is not clear in the scanned copy. This becomes hard to extract character based in the boxes. A more robust approach is used to extract the characters based on the relative position of the black markers on the four corners of the form. The position of each characters in the form is manually taken in to account. OpenCV is used to read the image files and each image is divided into three equal parts. For matching the black boxes in the original image and the scanned image, OpenCV's *cv2.matchTemplate* is used. Template Matching is a technique for looking and finding the location of a template image in a bigger image. It essentially slides the template image over the input image and compares the template and patch of input image under the template image. After template matching the images are converted into grayscale images using *cv2.cvtColor* & *cv2.BGR2GRAY*.

G. IMAGE THRESHOLDING

There is some noise in the image which can decrease the efficiency and accuracy of the model. To prevent this, image thresholding is used. In this process we set a threshold value, and the pixel values that are greater than this threshold value are set to white (a value of 255) and other pixel values are set to black (a value of 0). Fig. 11 is an example of different types of Image Thresholding.

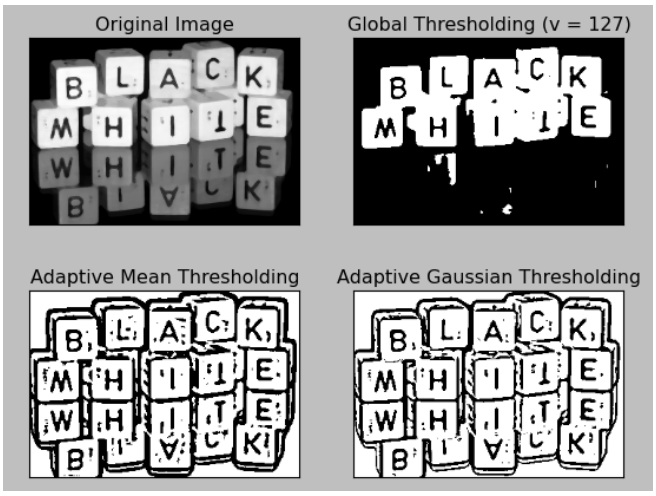

Fig. 11 Example of different types of image thresholding

H. CLASSIFIER

The training part is done using Keras API. Keras is a high-level neural network API written in Python language, capable of running on top of TensorFlow, CNTK and Theano.

V. RESULTS

Confusion Matrix are printed that provides the percentage of accuracy with which each digit and each alphabet has been recognized. A confusion matrix defines a specific table that allows the visualization of the performance of an algorithm by providing the accuracy corresponding to each of the input and output classes [16].

The model is tested on a few sample forms and confusion matrix is obtained. Confusion matrix of digits & alphabets are shown in Fig. 12 and Fig. 13 respectively.

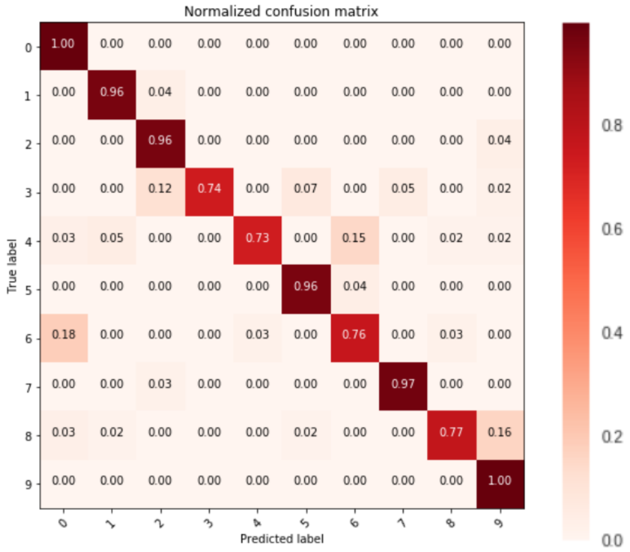

Fig. 12. Confusion Matrix of 10 digits

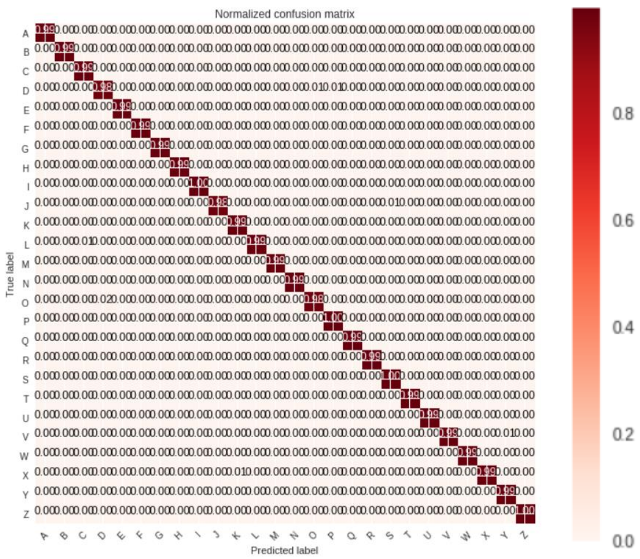

Fig. 13. Confusion Matrix of 26 alphabets

Most of the characters shows an excellent accuracy which more than 90%. Few similar looking characters were giving a lower but good accuracy above 75%. The details are shown in Table. 1 below.

TABLE I. ACCURACY OF THE CHARACTERS

CHARACTERS	ACCURACY	ACCURACY (in %)
0, 1, 2, 5, 7, 9 C, D, E, G, H, K, L, N, T, U, W, X, Y, Z	EXCELLENT	90 +
6, 7 J, M, O, R	VERY GOOD	80-90
3, 4 A, B, F, I, P, Q, S, V	GOOD	70-80

A web-based UI is created so that a user can view and edit the response. Then it can be saved in a JSON file. This entire framework is hosted on a webpage, shown in Fig. 16.

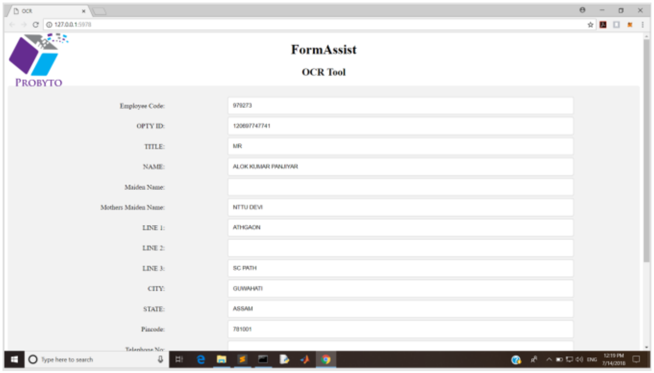

Fig. 14. Screenshot of the web-based UI of FormAssist

VI. CONCLUSION

The improved performance of OCR with the latest developments in Deep Learning have opened up scope for high value business use cases in Banking and Insurance Industry. The impact of improved OCR methodologies will be far reaching in near future, mainly due to the need of data for real time analytics. The above discussed methodology is implemented in Probyto's Business solution for industrial use – FormAssist.

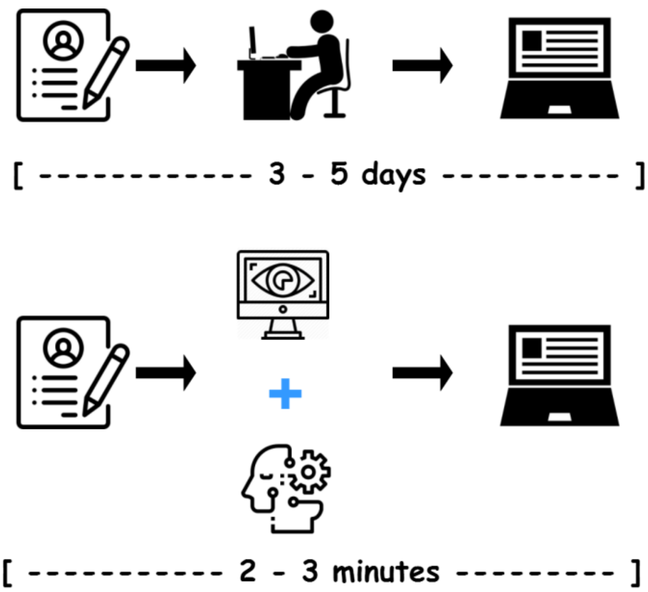

Fig. 15. FormAssist: End to End Solution for Handwritten Forms using Deep Learning

The application helps business to archive the digital data and provide near real-time data for deploying data analytics application. The algorithm is part of research team and keep on updating with latest developments in Deep Learning.

ACKNOWLEDGMENT

This work is supported by PROBYTO Data Science and Consulting Pvt Ltd and is being developed during Data Science Summer Camp 2018.

REFERENCES

[1] Darmatasia and Mohamad Ivan Fanany, *"Handwriting Recognition on Form Document Using Convolutional Neural Network and Support Vector Machines (CNN-SVM)"*

[2] A. Jindal and M. Amir, *"Automatic classification of handwritten and printed text in ICR boxes,"* Souvenir 2014 IEEE Int. Adv. Comput. Conf. IACC 2014, pp. 1028– 1032, 2014.

[3] N. Sharma, T. Patnaik, and B. Kumar, *"Recognition for Handwritten English Letters : A Review,"* vol. 2, no. 7, pp. 318–321, 2013

[4] Dan ClaudiuCires¸an and Ueli Meier and Luca Maria Gambardella and JurgenSchmidhuber, *"Convolutional Neural Network Committees for Handwritten Character Classification"*, 2011 International Conference on Document Analysis and Recognition, IEEE, 2011

[5] GeorgiosVamvakas, Basilis Gatos, Stavros J. Perantonis, *"Handwritten character recognition through two-stage foreground sub-sampling"* ,Pattern Recognition, Volume 43, Issue 8, August 2010

[6] Shrey Dutta, Naveen Sankaran, PramodSankar K., C.V. Jawahar, *"Robust Recognition of Degraded Documents Using Character N-Grams"*, IEEE, 2012

[7] Naveen Sankaran and C.V Jawahar, *"Recognition of Printed Devanagari Text Using BLSTM Neural Network"*, IEEE, 2012

[8] Yong-Qin Zhang, Yu Ding, Jin-Sheng Xiao, Jiaying Liu and Zongming Guo1, *"Visibility enhancement using an image filtering approach"*, Zhang et al. EURASIP Journal on Advances in Signal Processing 2012

[9] Bhatia, *"Optical Character Recognition Techniques: A Review"*, International Journal of Advanced Research in Computer Science and Software Engineering 4(5), May - 2014, pp. 1219-1223

[10] Anuj Dutt, AashiDutt, *"Handwritten Digit Recognition Using Deep Learning"*, IJARCET, Volume 6, Issue 7, July 2017, ISSN: 2278 – 1323

[11] Batuhan Balci, Dan Saadati, Dan Shiferaw, *"Handwritte n Text Recognition using Deep Learning"*

[12] Gauri Katiyar, Ankita Katiyar, Shabana Mehfuz, *"Off-Line Handwritten Character Recognition System Using Support Vector Machine"*, American Journal of Neural Networks and Applications 2017; 3(2): 22-28

[13] Nikhil Pai, Vijaykumar S. Kolkure, *"OPTICAL CHARACTER RECOGNITION: AN ENCOMPASSING REVIEW", IJRET,* Volume: 04 Issue: 01 | Jan-2015

[14] Jason Brownlee, *"Gentle Introduction to the Adam Optimization Algorithm for Deep Learning"*, Deep Learning, Machine Learning Mastery

[15] Diederik Kingma, Jimmy Ba, *"Adam: A Method for Stochastic Optimization"*, University of Toronto, 2015 ICLR paper (poster)

[16] Alsaad, A., 2016. *Enhanced root extraction and document classification algorithm for Arabic text* (Doctoral dissertation, Brunel University London).