SBI

Automate For The Bank 2018 **#A4B2018**

Theme II

Locate Extract and Match Signatures/Handwritten Texts from multiple formats and sources

AutoSIGN

OVERVIEW

We propose a way to automate the **mechanical and tedious** task of manual verification of signatures, and the validation of other Account Informations for the process of Mandate Verification. Presently, the process suffers from drawbacks including and not limited to **significant delays**, **inaccuracies** in Signature Validation, mostly due to the involvement of significant manual labour. **Automating the process** would not only help **reduce costs**, but would also improve turnaround time and ultimately make the customers, who are the reason for the existence of the bank, **happy and satisfied**. This would then lead to **higher engagement** and **better turnovers** for the company. This will also allow the workforce to be **more productive**, allowing them to work on issues that actually require human intervention and can't be automated, yet.

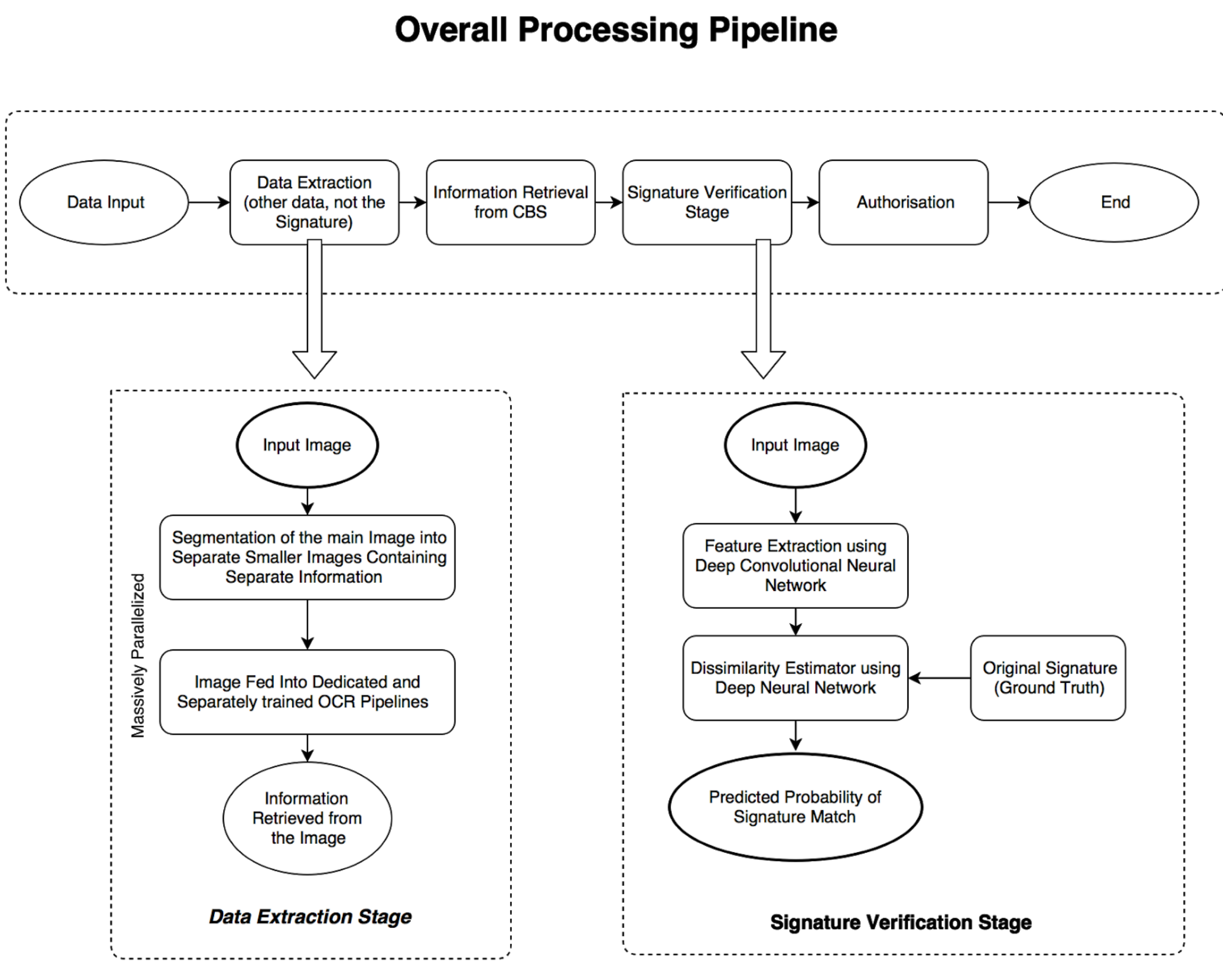
Our **Solution**, to this problem, is **AutoSIGN**.

GOALS

1. To **expedite** the process of Mandate Registration by applying **Machine Vision**.
2. To **reduce the error rate** in Signature Validation, by deploying **State-Of-The-Art** Machine Learning Algorithms. (**Siamese Networks** & **One Shot Learning**)
3. To **fully automate** the above process thereby enabling human workers to work on more **fulfilling** and **insightful** (non-mechanical) tasks.

4. **Integrate** the pipeline built by us in the SBI system and **minimize the resources** needed to run the system **LIVE**.

SPECIFICATIONS



The various stages involved are described in the following pages.

Overall Processing Pipeline

Data Extraction Stage

In this stage we get the Main Input Image containing all the different mandate information in different parts. From that image, we perform **segmentation** and get the different data items (like Account Number, Account Name Holder etc.) in different small images. The required information will be present in specific parts of an image in one specific kind of mandate (for example the Account Number position in a cheque), so a **single pipeline** will work.

Now from these specific Images, by using standard Open Source **OCR Libraries** in Python, specifically **Tesseract**, we'll extract all the details.

Information Retrieval from CBS

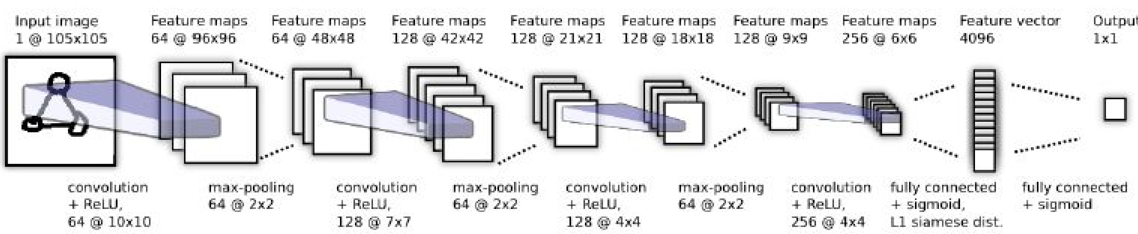
Using the Information we have already extracted from the previous stage, (just enough information to get all other information of the particular user from the **CBS API**, eg. the Account Number and the Account Holder's Name) we can get all other required information we need to validate the Mandate (images of correct Signatures, etc). This Information will be used to validate the signature later on. It will also be utilised in the last step of Mandate Validation as per the Bank Protocols. Furthermore multiple layers of encryption at every stage of the process will be added for **security** as per the need, as the information used is very **sensitive**.

Sending queries over a network is **slow** and **unpredictable**. To minimise the latency and maximise throughput, we propose a **two stage API**. A basic API to start the transaction from the mandate for a particular account. If the signature matches, we proceed to the second stage of the more advanced transaction API (explained in details in the 4th part), which will have more stringent safety checks and finally perform the transaction.

Signature Verification Stage

The Signature Verification stage will involve a novel **Deep Neural Network Architecture** by the name of **Siamese Networks**. This novel architecture has already proved its acumen by achieving superhuman capabilities in the task of Facial Recognition and Verification, and is used by **leading AI startups** around the globe in their own systems. Now, we propose to implement and **apply** this architecture to bring to the forefront its prowess as a signature detection and verification system.

The main **benefit** of this architecture is that we **don't need loads of examples** of one single kind of image (for our application loads of signatures of one person), as it just learns the "**features**" of a specific image and then **compares the features of two images** to see if they are identical or not. So this is a "**One-Shot**" method, and is **perfect** for this kind of applications.



This type of network architecture works by approximating a **non-linear distance function** between the **feature vectors** obtained from the images.

The distance between similar images (i.e. signatures belonging to the same person) is ideally zero and the distance between the signatures of two different people is as large as possible. The deep neural network is trained on triplets of images, i.e. (First Image - Signature of Person A, Second Image - another signature of person A which is in same manner and has very small differences, and the Third Image - Signature of Person B which is different from Person A). The optimization is to minimize the distance between the **Feature Vectors** of 1st and 2nd images and maximize the difference of Feature Vectors of 2nd and 3rd image. This objective function is known as the **Triplet Loss Function**.

The training of the Siamese Network would require a lot of **hand-crafted** triplet images to make sure the Network learns different aspects of Signature Detection and make sure that they get even **borderline cases** correct. Since the degree of model overfitting is determined by both its power and the amount of training it receives, providing a convolutional network with more training examples can reduce overfitting. Since these networks are usually trained with all available data, one approach is to either generate new data from scratch or perturb existing data to create new ones. For example, input images could be rotated by a few degrees to create new examples with the same label as the original, and this would better detect non-aligned signatures. This process is known as Data Augmentation.

We even need to manually feed some negative cases and train the network on it to better detect Signature Fraud, and to adapt to new ways of fraud. These synthetically generated data will help the Deep Neural Network model **converge faster** even on a relatively small data set.

Reference :

https://pdfs.semanticscholar.org/e669/55e4a24b611c54f9e7f6b178e7cbaddd0fbb.pdf

Why Siamese Network is the Best Choice for Signature Verification

- The entire image won't be sent over the network. Only the feature vector for the signature will be transmitted. This not only **reduces** the **bandwidth usage** but it also minimises the latency since a low footprint file is streamed over the network.
- Transmitting feature vectors is also **highly secure**, when compared to transmitting raw image files containing signature over the network. This is because the image is **vectorised** with a one way function (like a cryptographic hash function) which is learned from the data at hand. Thus, the function can't be inverted to find the original image. This is equivalent to sending encrypted data over network as these features are very **high dimensional**, and even if someone has our source code, it will be **impossible** to reconstruct the signature from the feature vector. As these features are actually linear combinations of actual human-recognisable features, it's impossible even for us to even understand the features. Also, anyone trying to perform a **Man-in-the-middle (MITM)** attack will be left clueless as to the contents of the transmission.
- We don't need **more than two signatures** for one person to train the network.
- The network can be trained online even in **production** to keep **improving** its performance with every input signature it is fed in with.
- In the primary database of the bank, we can now keep only the feature vector of a signature instead of the full images. This will **save a lot of space** and it'll be a lot lesser than the current database space usage since storing just a few integers in-place are much more efficient than storing images or links to images.
- This is the **State-Of-The-Art** system, with a **significant accuracy improvement** over the manual signature checking systems which are currently used.

Authorisation

Looking upon the Signature Validation and Account Name Validation **confidence scores** (0-100), if they comply to certain **thresholds**, and if the other mandate details are **matched perfectly**, we pass the final information to the Bank whether the Mandate is accepted/rejected. We will also incorporate multiple fraudulent activity checks along the way to stop fraudulent transactions from being executed.

Demo of Minimal Viable Product

Source Code Repository: https://github.com/AyanSinhaMahapatra/Signature_Validation_System

Please feel free to reach us for any further clarification at not.boring.ai@gmail.com.

Scope For Future Improvements

As an **improvement** to the current Data Extraction stage, we would like to **propose** a novel method to expedite the process of data extraction by **utilizing** the technology of **Optical Mark Recognition (OMR)** for the purpose of mandate forms. This will not only improve the overall accuracy of the system built also increase the speed of the system as we will overcome the need for the localisation stage of the entity and object detection stage of the processing pipeline. Further, Optical Mark Recognition (OMR) technology will reduce the probability of **typographical errors** occurring from the end user thereby **decreasing** the mismatch **error rates** of the system.

About Ourselves

Arpan Bhowmick (https://www.linkedin.com/in/arpan9/)

Ayan Sinha Mahapatra (https://www.linkedin.com/in/ayansinhaju/)

Dibyadip Chatterjee (https://www.linkedin.com/in/dibyadip-chatterjee/)

Sayan Goswami (https://www.linkedin.com/in/sayan-goswami/)

Bachelor of Engineering, Electronics and Telecommunication Engineering

UG - II, Jadavpur University, India