Signature Verification with Convolutional Triplet Network
Final Report

Word Count: 1967 Penalty: 0%

1 Introduction

The most common task in the field of forensic document analysis is authenticating signatures [1]. Currently, the tasks are done by Forensic Handwriting Experts (FHEs) who would compare the questioned signature with known authentic samples. The goal of this project is to automate the process of authenticating signatures, and by replacing human examiners with machine, a more accurate result could be achieved.

A neural network is appropriate for this task for its low financial and temporal cost. A qualified FHEs would spend years of training, testing, and getting certified before they can conduct examination that has legal effect. In comparison, once the proposed system is certified, anybody can use the signature verification and produce meaningful results. Replacing human labour with the computing power on mobile devices, signature verification could not only become cheaper but also more accessible in many occasions that were previously impractical.

2 Background & Related Work

The signature verification problem has two sub-categories: online and offline signature verification. Online signature samples contain more information than offline samples, including writing speed, acceleration, order, etc. whereas an offline signature sample is simply a static image of the signature. In this project, we mainly focus on offline signature verification, which applies to a broader and more practical use.

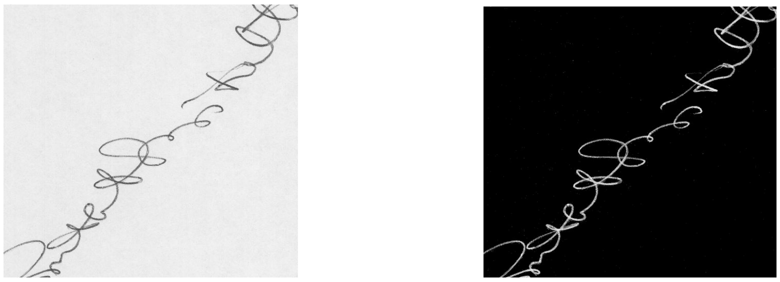
Using machine learning to automate signature verification is not a new idea. A 1994 academic journal mentioned the approach of training a neural network to solve this problem [2]. Yet, most approaches to automate signature verification are tested on specially collected data sets, which were acquired in controlled environments [3]. Simplifying the problem produces better results, but these developed models cannot reliably solve real-life cases.

In 2011, requested by the Ministry of Justice of the PR China and the Netherlands Forensic Institute, SigComp2011 were held in search of a signature verification system that can be implemented in forensic casework. Offline signatures produced in various real world environments are used to train and examine the quality of the contested systems. The highest accuracy achieved for Dutch and Chinese language is 97.67% and 80.04%, respectively [3].

3 Data and Data Processing

Scoping down the project to one specific language, we choose to use the CEDAR dataset prepared by CEDAR lab at University at Buffalo. This dataset includes 1320 original signatures (24 copies for each 55 unique signature) and 1320 forgeries, and all signatures are written in English alphabet.

To eliminate misleading factors (paper colour, paper material, ink colour, etc.) in the data samples, original RGB images are first converted to grayscale then to binary images. Yet, this process could also eliminate one important feature in the original data -- the strokes' thickness and strength. Hence, we have come up with a new solution to address this potential issue -- by setting a threshold in grayscale to binary conversion, we keep each pixel's original value if it is above the threshold, while making pixel values to zero otherwise. Images with this property is termed "partial binary" in this document. An example of the partial binary image is shown below.



Another concern with the CEDAR dataset is the limited number of signatures. There are only 55 distinct names in the dataset, and after the train-valid-test split, there are only 40 of them in training dataset. To prevent the model from overfitting, we address this issue by sampling 400 images from the training dataset and dividing each of them into 4 smaller pieces containing the same amount of black pixels. As a result, the training dataset will include images both with original dimension and truncated dimension. Note this process is only applied to the training dataset, and images in validation and testing dataset are not truncated.

At a later stage of the project, there was a need for two new datasets to further train and test the model. These two datasets are collected by SigVerComp 2011 [3], and are referred to as dataset C and D in this report. More details about train and validate on new datasets are discussed in section 7.

4 Architecture

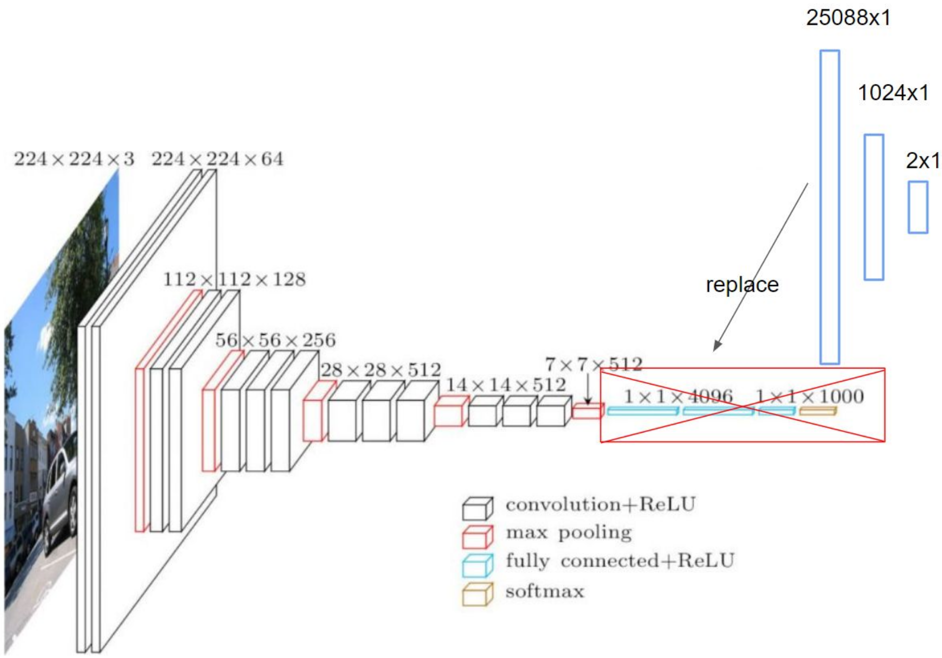


Figure 1

The adopted neural network is based on a pretrained VGG16 model. This pretrained model consists of 13 convolutional layers and 3 fully connected layers. The fully connected layers in this model is modified to suit the purpose of this project (as shown in Figure 1).

Inputting the preprocessed image into this neural network, an embedding of this image is obtained as output. The dimension of embedding is found by hyperparameter search shown in Figure 2.

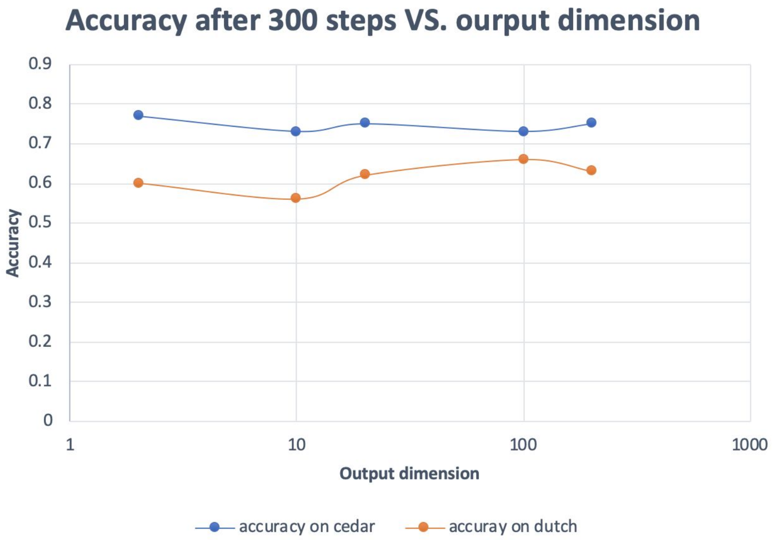


Figure 2

In training process, there are 3 images in a data instance (hence the name triplet network): Anchor (x), Positive (x+), and Negative (x-), where Anchor and Positive are authentic signatures signed by the same person, and Negative is a forged attempt for the signature. While in validation/testing, the only difference is that the third image in a data instance is the signature in question, meaning it could either be authentic or forged.

The Triplet Network architecture is illustrated in Figure 3, where the modified VGG16 model is denoted as "Net" here. The Net is applied separately on the 3 input images to obtain their corresponding embeddings. Then, two euclidean distances are computed, they are:

- dist(x, x+): between Anchor and Positive, and
- dist(x, x-): between Anchor and Negative

In training process, the two distances are used to compute loss using the following formula:

$$L(x, x^+, x^-) = \max(0, m + dist(x, x^+) - dist(x, x^-))$$

(m = margin)

To interpret this loss function,

$$Loss = \begin{cases} 0, & 2\ dists\ are\ far\ enough \\ L(x, x^+, x^-), & 2\ dists\ are\ not\ far\ enough \end{cases}$$

While validation or testing, the two distances are then compared to determine the authenticity of the questioned signature using the formula as follows:

$$dist(x, x^-) - dist(x, x^+) \geq Margin$$

Such network is termed as "Triplet Net" equipped with Triplet Margin Loss in this report.

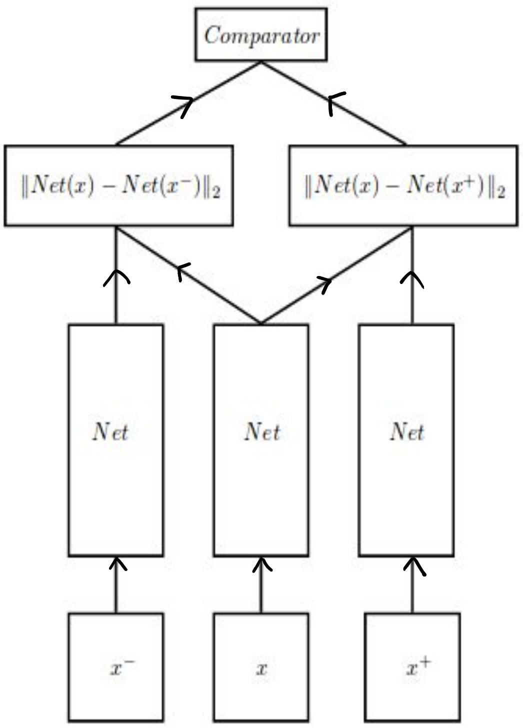


Figure 3 Triplet loss

5 Baseline Model

The baseline model adopted is similar to the triplet network architecture, but equipped with a different loss function instead of the triplet loss function mentioned in section 4. Instead of taking three images as input, the baseline model only requires two. The two-image pair could either be an Anchor--Positive pair or an Anchor--Negative pair.

The architecture of the baseline model also adopts VGG16 pretrained model. The neural network is applied separately on the 2 input images to obtain their corresponding embeddings. Then, the euclidean distance between the two embeddings are computed:

- dist(x, x+): if the input is an Anchor--Positive pair
- dist(x, x-): if the input is an Anchor--Negative pair

In training process, the computed euclidean distance is used to compute loss using the following formula:

$$L = \begin{cases} dist(x, x^{+}) & if \quad PositivePair \\ max(0, m - dist(x, x^{-})) & if \quad NegativePair \end{cases}$$

(m = margin)

While validation or testing, the two distances are then compared to determine the authenticity of the questioned signature using the formula as follows:

$$dist(x, x^{-}) - dist(x, x^{+}) \geq Margin$$

Such network is termed as “Siamese Network” [4]equipped with Pairwise Ranking Loss[4] or simply the baseline model in this report.

6 Quantitative Results

After the Triplet Net is constructed, the CEDAR dataset is used to train, validate and test the network. The network converged very quickly, stabilizing at low loss and achieving high training and validation accuracy within 500 steps. In addition, the confusion matrix shows the trained model is sensitive, serving the purpose of conducting signature verification.

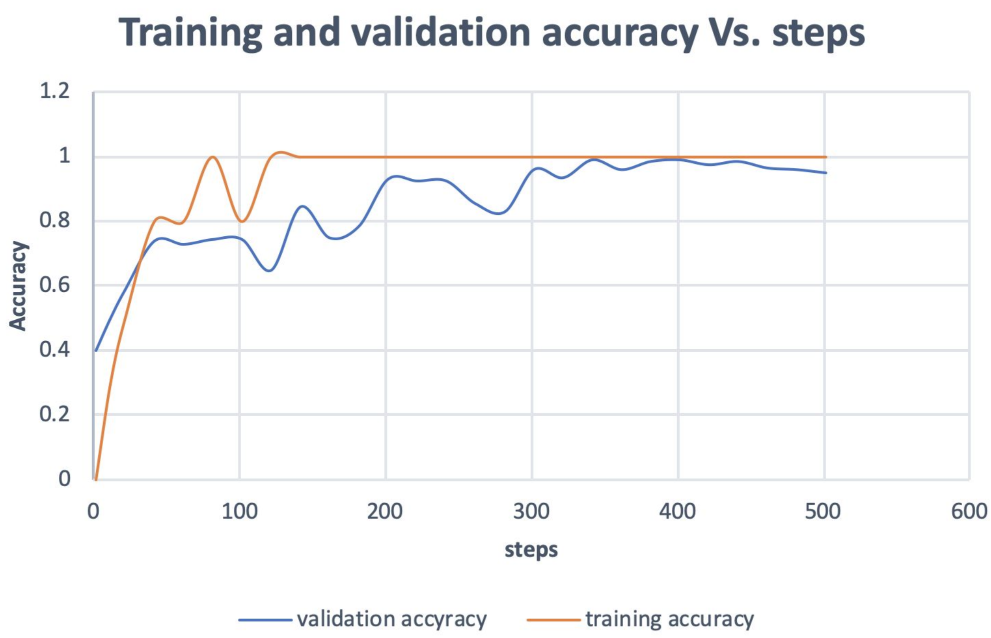


Figure 4. Plots of training and validation loss vs. # of steps (upper) as well as training and validation accuracy vs. # of steps (lower) when applying Triplet Net to CEDAR dataset

	Auth	Forg
Auth	233	17
Forg	3	247

Figure 5. Confusion matrix when testing Triplet Net on CEDAR (sensitivity = 98.8%)

The quantitative results of the Triplet Net and baseline model (both trained and tested on CEDAR) is summarized in Table 1. Triplet Net performs better than the baseline model in every metric.

	Baseline Model	Triplet Net
Training Accuracy	70%~80%	100%
Validation Accuracy	66%~68%	95+%
Testing Accuracy on CEDAR dataset	65%	95+%
Testing Accuracy on dataset D (trained with CEDAR)	52 %	56 %

Table 1. A summary of the quantitative results up to this stage

Although the trained model achieved good results on CEDAR dataset, further testings are conducted to examine the applicativity of this model to solving real-world signature verification problems. We found another dataset D to test the trained model. The results are displayed in Figure 6.

	Auth	Forg
Auth	145	105
Forg	114	136

Figure 6. Confusion matrix when trained on CEDAR and tested on a new dataset D (sensitivity = 54.4%)

At this stage, the trained model performed well on the signature images from CEDAR dataset but did significantly poorer on a new dataset D. That was when we realized using CEDAR as our training set was not representative enough. Hence, we integrated another dataset C into the training dataset, with the hope that the model can generalize better with the addition of data with different characteristics, and thus improve the testing accuracy when applied to dataset D.

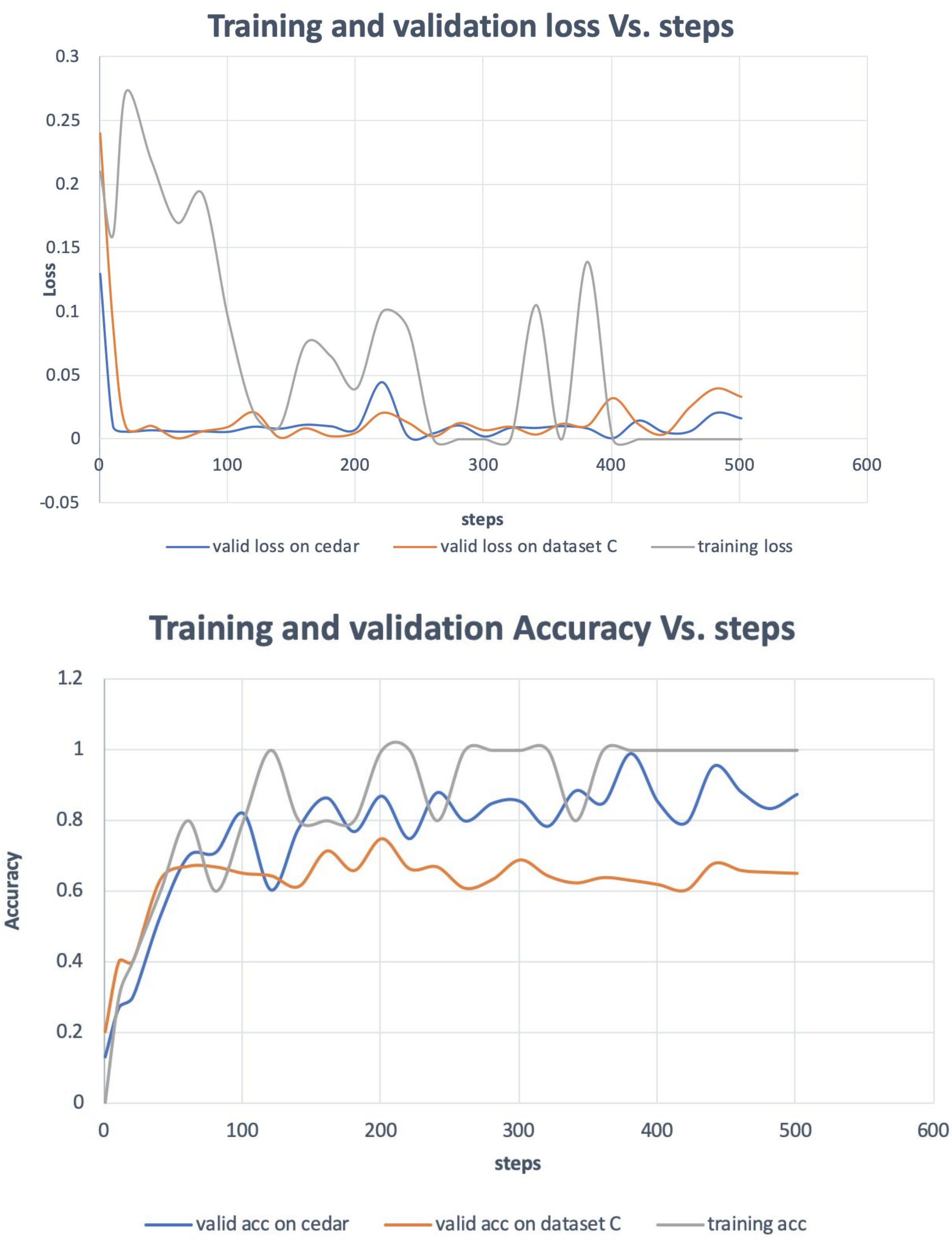


Figure 7. Plots of training and validation Loss vs. # of steps (upper) as well as training and validation accuracy vs. # of steps (lower) when applying Triplet Net to a new dataset C

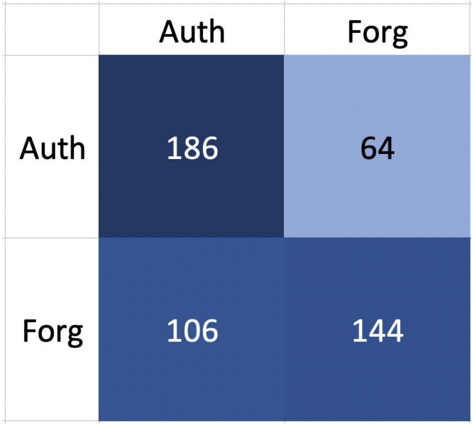


Figure 8. Confusion matrix when trained on CEDAR & dataset C and tested on dataset D (sensitivity = 74.4%)

Indeed, with the addition of more data with distinct characteristics (pen thickness, hand-writting styles, more distinct signatures, etc.) in the training dataset, the model improved on its accuracy and sensitivity when tested by dataset D. The quantitative results are summarized below.

	Baseline Model	Triplet Net
Training Accuracy	70%~80%	100%
Validation Accuracy	66%~68%	95+%
Testing Accuracy on CEDAR dataset	65%	95+%
Testing Accuracy on dataset D (trained with CEDAR)	52 %	56 %
Testing Accuracy on dataset D (trained with CEDAR & dataset C)	60%	66%

Table 2. The final summary of the quantitative results

7 Qualitative Results

We further explored our model by printing out some example signatures that our model can and cannot recognize. Given Figure 9(a) , 9(b), 10(a)and 10(b) as authentic, our model can recognize 9(c) is a forged signature, but cannot recognize 10(c) is forged.
By comparing two forgery signatures, we can see that 9(c) has different stokes and letters from 9(a) and 9(b), while 10(c) is very similar to 10(b). We can conclude that our model has learned to detect forgeries in a way that is very close to human. Also, our model can produce similar predictions as human.

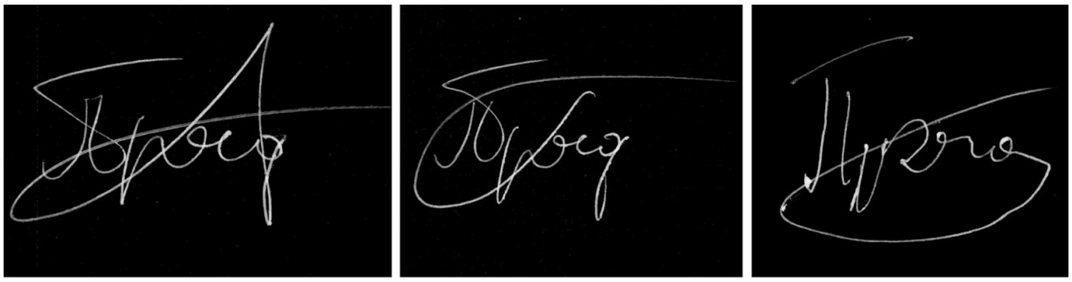


Figure 9(a)(b)(c). A data instance our model can recognize. The three signatures are ((a)authentic-(b)authentic-(c)forgery)

Figure 10(a)(b)(c). A data instance our model can not recognize. The three signatures are ((a)authentic-(b)authentic-(c)forgery)

8 Discussion and Learnings

(This report is structured in a way that some of the findings and learnings are discussed in section 6. Here is a summary of the important learnings in this project)

With the help of the two new things we tried in this project, the model performed well. The Triplet Net trained with the triplet loss function achieved high accuracy; and transfer learning reduced the training time.

However, many things caught us on surprise. For example, the model converged much quicker than expected; and performance of the same model across several datasets can be vastly different.

To improve, the training dataset could be more representative. We saw that the model generalize better after an addition of data varieties in the dataset.

9 Ethical Framework

Considering the adoption of the proposed machine learning model to verify signatures as an action, this section provides an ethical examination of the action using four criteria as follows:

Beneficence:

Comparing the cost for training a model to training a qualified FHE, the model can learn with much more samples in much less time. In addition, a successful implementation of this model can produce meaningful results with lower cost compared to hiring FHEs, which is expensive and is not always accessible.

Nonmaleficence:

A wider adoption of the model by the public would also further reduce forgery attempts to gain personal interests.

Autonomy:

When installed on mobile devices, the model can support the users to conduct signature verification anytime anywhere with the computation power available on mobile devices.

Justice:

Since this approach to solve signature verification problem is attempting to automate the work currently done by people, there exists the possibility of replacing workers with this model, forcing them out of work and taking away their financial sources. There are students who are currently in the process of learning and becoming experts in this industry; and instructors who have worked hard to possess the knowledge and achievements they have today. The adoption of this model would bring damages to their routine lives, penalizing them unfairly for their career choice.

References

[1] Bradford, R.R., Bradford, R. “Introduction to Handwriting Examination and Identification.” Nelson-Hall (1992)

[2] L. Franck and R. Plamondon, *Automatic Signature Verification: The State of the Art*. 1994.

[3] M. Liwicki *et al.*, "Signature Verification Competition for Online and Offline Skilled Forgeries (SigComp2011)," *2011 International Conference on Document Analysis and Recognition*, Beijing, 2011

[4] G.Koch, R.Zemel, & R.Salakhutdinov. (2015). Siamese Neural Networks for One-Shot Image Recognition. ICML - Deep Learning Workshop, 7(11), 956–963. DOI : https://doi.org/10.1017/CBO9781107415324.004

[5] Vassileios Balntas, Edgar Riba, Daniel Ponsa and Krystian Mikolajczyk. Learning local feature descriptors with triplets and shallow convolutional neural networks. In Richard C. Wilson, Edwin R. Hancock and William A. P. Smith, editors, Proceedings of the British Machine Vision Conference (BMVC), pages 119.1-119.11. BMVA Press, September 2016.

Permissions

Permission to post video: yes for both group members

Permission to post final report: yes for both group members

Permission to post source code: yes for both group members