Research Proposal
Offline text-independent writer verification and identification by learning the global feature vectors via triplet CNN

Davit Kvartskhava
dkvart17@earlham.edu
Department of Computer Science
Earlham College
Richmond, Indiana

1 ABSTRACT

Writer identification based on handwriting plays an important role in law enforcement investigations. Convolutional Neural Networks have been successfully applied to this problem throughout the last decade. Most of the research that has been done in this area has concentrated on extracting local features from handwriting samples and then combining them into global descriptors for writer retrieval/verification. This research aims to use Triplet CNNs to extract global feature vectors from images of handwritten text directly, eliminating the intermediate step involving local features. Extracting local features from small patches of handwriting samples is a reasonable choice considering the lack of big training datasets. However, the methods for aggregating local features are not perfect and do not take into account the spatial relationship between small patches of handwriting. Extracting global features from handwriting samples is not a novel idea, but this approach has never been combined with Triplet architecture. Training the CNNs to learn the global descriptors requires a large amount of training data, so I plan to use data augmentation techniques to enlarge the database by a factor of 100. The method will be evaluated on the accuracy of identification on the ICDAR 2013, CVL and IAM datasets.

2 INTRODUCTION

“Handwriting is a kind of behavioral biometrics [13].” Every person has a somewhat distinct handwriting style, which makes it possible to verify or identify a person based on their handwriting [1]. Manual forensic handwriting analysis is used by law enforcement agencies to identify the writer, and it plays a considerable role in investigations [7]. However, identifying a writer based solely on their handwriting requires a lot of human expertise and experience in addition to being very time-consuming. Hence, automating this process is a research topic of interest. The emergence of Convolutional Neural Networks has brought hope that machines will surpass the baseline set by human experts. The research on automating writer identification methods has also become relevant to analyzing historical documents as more digitized data is now available.

Signature verification can be viewed as a specific application of the writer identification task. However, in the case of signature verification, the problem space is different, as the main focus is on distinguishing between forged and genuine signatures [5]. It should be noted that because our training dataset does not include forged handwriting samples, a limitation of this study is that it will most likely fail in case of a skilled forgery.

The research in writer identification is usually divided into two sub-categories – on-line and off-line writer identification. In on-line writer identification, the dynamic information about the procedure of writing is preserved using specialized devices. In off-line writer identification, such information is not available and the only input is the handwritten text itself.

The approaches for solving the problem of writer identification can also be divided into two categories – text-independent and text-dependent methods. The text-dependent method requires the input to contain the same text as the target handwriting (or at least the same set of characters). In contrast, the text-independent method tries to solve the problem regardless of the content of handwriting.

In the last decade, Convolutional Neural Networks have become a popular choice for analyzing visual documents [8]. The groundbreaking work on object detection, OCR, face verification and many other successful applications of CNNs has revolutionized the field [9]. CNNs have also been successfully used in the writer identification problem [3, 12, 13]. Such neural networks have set the state-of-the-art baseline in terms of accuracy of identifying the writers based on their handwriting [3, 12, 13].

However, the use of CNNs is not a simple recipe for guaranteed success - different preprocessing steps, the dimensions of the input, the loss function and different formulation of the same problem often lead to different results. In this paper, I will review the methods used in the last decade to tackle the problem of writer identification with Convolutional Neural Networks and present the topic of my research below.

This proposal concentrates on off-line text-independent writer identification using CNNs. The rest of this paper talks about (3) the approach that I am suggesting, (4) related work that has been done using CNNs, (5) design and implementation strategy, (6) major risks and (7) timeline.

3 RESEARCH GOALS

My approach is to train a Convolutional Neural Network to directly learn the global representations of handwriting samples in the Euclidean space. The goal is to optimize the embeddings using the triplet architecture. This method has successfully been applied to the task of face recognition [10]. Similar work involving Triplet CNNs has been done by Keglevich et al. [7]. However, the research approach was to combine the local feature vectors through different algorithms instead of directly learning the global descriptors. The

CS388, Earlham College,
2020.

local feature vectors are produced by feeding a CNN with low dimensional patches cut out of the same handwriting sample. Hence, each handwriting sample can be characterised by a set of local descriptors. There are multiple methods for combining these local descriptors into a global vector that represents the handwriting style of a given sample. On the other hand, global feature vectors can be directly produced by CNNs if instead of small patches, CNN is fed with an entire handwriting image. Tang and Wu [12] have researched methods for optimizing the global features without aggregation of local features, but the technique that they used did not involve Triplet CNNs. The motivation for learning global descriptors as opposed to aggregating local ones is that the retrieval of local features from the small patches of the handwritten text images leads to the loss of information that might be key to identify the author with high accuracy. Aggregation methods that combine local descriptors to the global ones are not perfect, and the spatial information about the location of the patches is lost. My research is unique in that I am using triplet CNNs to learn the global descriptors to tackle the writer identification problem. The downside of feeding the CNN with large patches is that it requires more data for CNN to become accurate, so I am also employing a data augmentation technique inspired by Tang and Wu [12].

4 BACKGROUND

This section describes different methods that have been used to address the writer identification problem using CNNs. Two main methods are described below: (1) training a CNN to classify the handwriting samples and (2) learning the feature vectors via CNN. In addition to that, section 4.3 reviews the methods that have been used to address the lack of training data.

4.1 Classification into writer classes

Convolutional Neural Networks have been used in two distinct ways to identify writers based on their handwriting. The first approach treats the problem as a classification task. CNN is trained through softmax loss function, where the number of output nodes corresponds to the number of users in the database. The output of each node signifies the probability that each user is the author of the handwriting. The shortcomings of this approach are that the network is not scalable and it needs to be retrained every time a new writer is registered in the database.

Xing and Qiao [13] have taken the approach mentioned above of directly training a classifier. Such a CNN outputs a vector of probabilities for a handwriting sample belonging to a specific writer in the database. Xing and Qiao extracted the patches from the lines of handwritten text. They used a specific architecture (multi-stream structure) of a neural network comprised of two dependent CNNs that share the features in some layers. The reason for using such architecture was to take advantage of the spatial relationship between different square patches. The input for this network was a pair of two adjacent patches.

4.2 Methods for obtaining encodings

A second approach for writer identification is to produce the feature vectors or encodings associated with each input image. This approach deals with the issue of scalability of the basic classifiers. The encodings are supposed to capture the unique features of the handwriting, so that the encodings themselves are enough to differentiate between two writers. This way, a feature vector can be produced for the handwriting whose author is not in the training dataset. After the feature vector has been generated, the final step is to compare it with other encodings in the database and find the one such that some measure of similarity between the encodings is minimized.

4.2.1 Encodings produced through classification.

There are different methods for obtaining encodings. An older approach starts by training a CNN with a classification layer with a task to learn to classify the handwriting samples into the writer classes [11]. The second step is to extract the penultimate layer of the network. This layer contains the features specific enough to a writer that feature vectors can be used to distinguish between handwriting samples from different authors [11].

Fiel and Sablatnig [4] used the method described above to extract the feature vectors for the writer identification. An encoding for an entire image of handwriting was obtained by averaging the encodings generated by the small patches. These encodings were later compared using 2 distance.

Christlein and Maier [3] took a similar approach to extract local feature vectors - they took the penultimate layer of a CNN as an encoding. For identification, they used cosine distance between global descriptors. They combined local feature vectors using different algorithms in order to produce a global descriptor for each handwriting sample. They compared the VLAD encoding to triangulation embedding. They also compared max pooling to sum pooling in the writer identification task. The input for the CNN was the small 32x32 patches that were randomly drawn from inside of the contours of the handwriting image.

4.2.2 Directly optimized encodings.

Another method for obtaining encodings was devised in 2015 [10], and it significantly improved the benchmark for face recognition/verification. When applied to writer identification, however, it did not improve upon the baseline set by other approaches [7]. Below, I will review both of these papers.

Schroff et al. [10] published a paper in 2015 on learning unified embeddings for face recognition. The method that they proposed produced an algorithm with a 30% lower error rate than other known approaches. They started by training a CNN with the direct aim to optimize the encodings themselves, instead of treating the problem as a classification task. They mention that the downsides of the older approach "are its indirectness and its inefficiency". The algorithm starts by picking three examples from the data — an anchor, a positive example and a negative example. Then the triplet loss function is used to maximize the distance between the encodings of the anchor and the negative example, while at the same time minimizing the distance between the anchor and the positive example. This way, the network learns to encode the images in a way that the resulting feature vector accurately represents unique features of different faces. Schroff et al. also discuss the importance of choosing the best triplets for training and propose a specific algorithm for choosing such triplets.

Keglevich et al. [7] applied this recent version of obtaining the encodings to writer identification. Again, the objective was to learn the encodings of the handwriting samples where the square distance (L2 measure) between encodings obtained from two different classes is maximized and the same measurement for the identical classes is minimized. In this paper, they incorporated an interesting algorithm for extracting the patches. They retrieved the patches around the SIFT keypoints. As they claim, based on previous research, SIFT points are such that there is enough information around them for the network to learn useful encodings. After feeding the CNN with these patches, they aggregated the vectors from different patches into one encoding. For this process of creating one feature vector per entire image of handwriting, they use VLAD [6] encodings. This approach was tested on ICDAR 13 database, and the authors report near the-state-of-the-art results.

4.3 Methods addressing the lack of data

Tang and Wu [12] proposed a novel data augmentation technique because of the necessity of large amounts of data to train a CNN. For the feature vector retrieval, they used the method of training with classification objective and extracting the last layer. They also proposed the use of joint Bayesian technique instead of square distance for the identification task. All the previous research that has been done in this area has focused on training the CNN on small image patches; however, the problem of this approach is that when local features are extracted from patches, some details about a person's writing style are lost. Learning the global features requires a lot more data, so they first extracted the words from the images of handwritten texts and then randomly permuted each word in a line. As a result, they were able to accumulate thousands of handwriting images for each writer in the dataset. They reported the best results on the CVL dataset and near state-of-the-art on ICDAR 13.

Chen et al. [2] also pointed out that CNNs need a lot of training data to achieve satisfactory accuracy in real-world applications. Data augmentation techniques do generate more data, but the downside of using such techniques is the risk of overfitting to the repeated data. Instead, they proposed a semi-supervised deep learning algorithm that learns to extract the writing style features from the mixture of labeled and unlabeled data. The patches are obtained from the original images, and VLAD encodings produce global descriptors from the local feature vectors.

5 DESIGN AND IMPLEMENTATION

This research project aims to combine the ideas of Keglevich et al. and Tand and Wu. Keglevich used the triplet architecture to optimize the local encodings directly. In contrast, Tand and Wu took the approach of learning the global descriptors, and they used the penultimate layer of CNN as a feature vector. In addition to that, Tang and Wu utilized data augmentation techniques in order to meet the data needs of learning from large input. To my knowledge, no research has been done using Triplet CNNs to directly learn the global encodings.

The dataset of labeled handwritten images will be preprocessed using binarization and denoising techniques. After that, the dataset will be randomized and divided into training, validation and test sets.

Then I will use data augmentation to enlarge the training dataset. This step will not be applied to validation or test sets. I will use the same technique as used by Tang and Wu. Each image of handwriting will be broken up into smaller patches containing the words, and these patches will be randomly permuted to produce new examples.

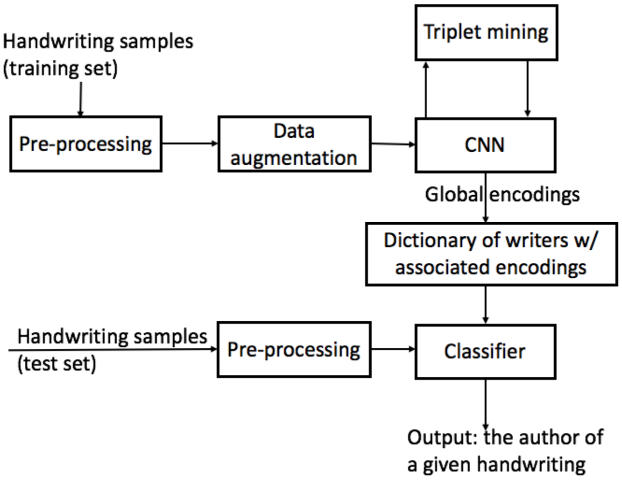
Handwriting samples (training set)
Pre-processing
Data augmentation
CNN
Triplet mining
Global encodings
Dictionary of writers w/ associated encodings
Handwriting samples (test set)
Pre-processing
Classifier
Output: the author of a given handwriting

Figure 1: The pipeline

The next step is the implementation of an online triplet mining strategy (choosing anchor, positive and negative examples). I will follow the suggestions of Schroff et al. The first step involves computing the encodings for each input in a single batch. After that, given an anchor I will find the hard positive (a positive example with the largest distance from the anchor) and hard negative examples (a negative example with the smallest distance from the anchor).

Next, I will start training a CNN to learn the encodings with given triplets of examples. This part will require a lot of iteration in order to find the best hyperparameters for the neural network. Next, I will implement the classification module. The first step to implement this module involves creating a correspondence between the writers in the database and the associated embeddings. This module will take an image of handwriting as an input, produce an embedding for that sample and look through the writer database to find a writer with an associated embedding that differs the least in L2 distance measure.

I am planning to implement a Triplet CNN with the keras library in Python. I will also be using the OpenCV library to detect the handwriting contours and extract the patches, including words (this step is necessary for data augmentation).

6 MAJOR RISKS

1. Data augmentation techniques make the CNN overfit to the training dataset.

I can address this issue by reducing the amount of artificial data produced. In addition to that, I can combine multiple datasets to get large amount of training data.

2. Training takes a long time and I'm not able to run enough experiments.

I can run the experiments in parallel. I could also make use of the cloud services that provide the environments suitable for deep learning.

7 TIMELINE

Week 1: Download the datasets. Create the working environment on lovelace to make use of GPGPUs: download all the necessary packages (keras with tensorflow backend, opencv), set up a jupyter notebook to run on lovelace. Binarization of images. Split the dataset between training, development, test sets. Finding contours of text using OpenCV. Start working on the first draft of the paper.

Week 2: Start training the CNN to learn the feature vectors on a fraction of data. Make sure that CNN has enough layers to overfit a small number of examples. Produce preliminary results of accuracy for training and development sets to decide what kind of work I should prioritize next. Try some data augmentation techniques, closely examine the augmented images to make sure everything's working fine. Submit the first draft of the paper.

Week 3: Finish up augmenting the data. Start training the CNN on the combined dataset. Implement the identification module. All of the major pieces of the software should be ready.

Week 4: Check the results to see if we're making progress. Employ some regularization mechanisms (L1, L2) Produce the accuracy measures for each dataset, and the aggregated results based on soft and hard criterion. Continue working on the paper.

Week 5: Make sure that the code is clean and is working as expected and produce README. Submit the second draft of the paper.

Week 6: submit the second release of the code. Revise the paper, start working on a poster.

Week 7: Have everything ready. Let others review the paper and the poster. Based on the feedback, make the necessary changes.

Week 8: Buffer time.

ACKNOWLEDGMENTS

I would like to thank David Barbella.

REFERENCES

[1] Marius Bulacu and Lambert Schomaker. 2007. Text-independent writer identification and verification using textural and allographic features. *IEEE transactions on pattern analysis and machine intelligence* 29, 4 (2007), 701–717.

[2] Shiming Chen, Yisong Wang, Chin-Teng Lin, Weiping Ding, and Zehong Cao. 2019. Semi-supervised feature learning for improving writer identification. *Information Sciences* 482 (2019), 156–170.

[3] Vincent Christlein and Andreas Maier. 2018. Encoding CNN activations for writer recognition. In *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*. IEEE, 169–174.

[4] Stefan Fiel and Robert Sablatnig. 2015. Writer identification and retrieval using a convolutional neural network. In *International Conference on Computer Analysis of Images and Patterns*. Springer, 26–37.

[5] Luiz G Hafemann, Robert Sabourin, and Luiz S Oliveira. 2017. Learning features for offline handwritten signature verification using deep convolutional neural networks. *Pattern Recognition* 70 (2017), 163–176.

[6] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. 2010. Aggregating local descriptors into a compact image representation. In *2010 IEEE computer society conference on computer vision and pattern recognition*. IEEE, 3304–3311.

[7] Manuel Keglevic, Stefan Fiel, and Robert Sablatnig. 2018. Learning features for writer retrieval and identification using triplet CNNs. In *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. IEEE, 211–216.

[8] YD Li, ZB Hao, and Hang Lei. 2016. Survey of convolutional neural network. *Journal of Computer Applications* 36, 9 (2016), 2508–2515.

[9] Weibo Liu, Zidong Wang, Xiaohui Liu, Nianyin Zeng, Yurong Liu, and Fuad E Alsaadi. 2017. A survey of deep neural network architectures and their applications. *Neurocomputing* 234 (2017), 11–26.

[10] Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 815–823.

[11] Yi Sun, Xiaogang Wang, and Xiaoou Tang. 2015. Deeply learned face representations are sparse, selective, and robust. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2892–2900.

[12] Youbao Tang and Xiangqian Wu. 2016. Text-independent writer identification via CNN features and joint Bayesian. In *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. IEEE, 566–571.

[13] Linjie Xing and Yu Qiao. 2016. Deepwriter: A multi-stream deep CNN for text-independent writer identification. In *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. IEEE, 584–589.