Analysis of a high-resolution hand-written digits data set with writer characteristics

Cédric Beaulac Jeffrey S. Rosenthal

October 27, 2020

Abstract

The contributions in this article are two-fold. First, we introduce a new hand-written digit data set that we collected. It contains high-resolution images of hand-written digits, a writer identification and various writer characteristics. The data set is publicly available and is designed to create new research opportunities. Second, we perform a first analysis of this new data set. We begin with simple supervised tasks. We assess the predictability of the writer characteristics gathered, the effect of using some of those characteristics as predictors in classification task and the effect of higher resolution images on classification accuracy. We also explore semi-supervised applications; we can leverage the high quantity of hand-written digits data sets already existing online to improve the accuracy of various classifications task with noticeable success. Finally, we also demonstrate the generative perspective offered by this new data set; we are able to generate images that mimics the writing style of specific writers. The data set provides new research opportunities and our analysis establishes benchmarks and showcases some of the new opportunities made possible with this new data set.

Keywords : Computer Vision, Image classification, Writer Identification, Convolutional Neural Networks, Variational Auto-Encoders

1 Introduction

Modern computer vision algorithms have become impressively good at identifying the content of a complex image. Scanned hand-written document is an example of complex image for which many algorithms were developed. In this case the task assigned to the algorithm is to identify letters, digits and later words and sentences. In hand-written document analysis, the MNIST data set introduced by LeCun & al. [22] quickly became a benchmark for hand-written digits recognition and is now a rite of passage for any computer vision algorithm. Usually, MNIST is used for a simple task, try to identify the digit in new hand-written digit images given a training set of labelled hand-written digit images.

In this project, we explore the potential of state-of-the-art computer vision algorithm for a wider range of inference task on hand-written digits. For instance, we will tackle the writer identification problem which is a common problem in criminology or historical research. Broadly speaking, if more labels were attached to an image, could we successfully extract other interesting information out of those images ? Our goal is to rely on modern computer vision algorithms to replace feature engineering (handcrafted features). Based on the recent results obtained by Adak et al. [1] it appears that auto-derived feature outperforms feature engineering and this what we are hoping to exploit here.

We tackle the well-established task of writer identification, but also statistical inference of writer characteristics. We also discuss new research opportunities created with this data set. Our contribution is two-fold; first, we introduce and distribute a new data set that we collected, HWD+, containing hand-written digit images in high-resolution and various writer characteristics. This data set can be utilized as a standalone data set and also in conjuncture with MNIST for semi-supervised learning projects. Second, we perform a first analysis of the data set under both the supervised and semi-supervised paradigm. We also showcase how to use this data set to create interesting image generation challenges.

The remaining of this paper is organized as follows: we discuss the related publications in section 2. Section 3 introduce the new data set we collected. Following this, we introduce the algorithms used for our first analysis in section 4. Section 5 contains our analysis of the HWD+ data set.

2 Related work

In the contributed data set, we collected various characteristics about our writers and also assigned a writer ID to each writer. Consequently, one natural problem to tackle is writer identification. This problem has been extensively studied in the past and is still a relevant problem in forensics. A recent publication from Adak et al. [1] attempts to solve a writer identification problem and compares the performances of models that rely on hand-crafted feature against models with auto-derived features. Slightly before that, Xiong et al. [36] produced one of the most recent surveys comparing various modern writer identification algorithms. A result is shared across both articles [36, 1] and highlighted in a comprehensive review [32] is that auto-derived feature models perform better than feature engineering and thus we rely on auto-derived feature models in this analysis.

One of the tasks we've established for this research project was to assess the abilities of modern computer vision algorithms to infer some of the writer's characteristics. Most literature that discusses writer characteristics addresses graphology; the analysis of hand-writing patterns in order to identify psychological traits of writers. However, serious studies on graphology demonstrate that it is more a pseudo-science than anything else [21]. As a result, we focus this works on measurable characteristics such as age, gender or native language. We are interested in determining the feasibility of predicting such characteristic based on handwritten digits.

When considering the identification of the digits themselves, the MNIST data set inspired a gigantic amount of publications. The first article to discuss this data set [22] was published in 1998 and introduced the data set and compared the prediction accuracy of multiple classification methods. To two best performing algorithms were a committee of deep convolutionary neural networks (CNN) and a support vector machine (SVM) with test error rates as low as 0.7% and 0.8% respectively. This article really set the tone for future computer vision publications by establishing the sheer dominance both in terms of accuracy and memory requirements of CNNs. It was a pivotal point into explaining and empirically proving the benefits of automated feature extraction. It has also somehow established the MNIST data as an important benchmark data set.

Since then the best results obtained from a SVM algorithm was obtained in 2002 [8] with a

0.56% error rate. Simple techniques that require no training, such as KNN has achieved higher accuracy (0.54% error rate) by allowing the algorithm to search into a set of distorted images [17]. The lowest error rate (0.35%) achieved by a single NN was reported in 2010 [7]. Finally, in 2012 a committee of 35 CNNs achieved a 0.23% test error rate [6]. A problem with MNIST is that current algorithms achieve a classification accuracy that is so high that it leaves room only for marginal improvements. The true usefulness of these improvements is hard to evaluate [14] as it might be caused to details that are specific to the MNIST data set and thus aren't real improvement applicable to new problems. In other words, it is possible that MNIST has been overused and that some new models are *overfitting* this data set.

Finally, let us address related data sets. As we already mentioned, our data set is quite similar to the now-famous MNSIT data set [23]. Other digit image data sets also became quite popular such as the SVHN data set [28] which contains images of house numbers in Google Street View images. The only label included in those data set is the digit itself and supervised tasks are directed at digit classification. Our data set, HWD+, offers more opportunities since it contains various characteristics and writer identification. On top, our data set also contains high-resolution images.

For writer identification, there exist multiple text-written data set. For instance, the CEDAR (Center of Excellence for Document Analysis and Recognition) [5, 34] developed multiple data sets containing either only letters, continuous text or signatures.

There exist various massive multi-labels data sets online such as CelebA [25], DeepFashion [24] and DeepFashion2 [11] that contains images in high resolution and multiple labels. However these data sets can be overwhelmingly large and require anyone who wishes to use them to have access to multiple GPUs in order to experiment with them. Our new data set, the HWD+ data set, is much more approachable. It offers multiple labels and high resolution images but also a 28 by 28 pixel alternatives for those who wishes to simply plug in this new data set in a code already set up for MNIST or SVHN.

3 Data set

We named our Hand-Written Digits data set HWD+. The plus sign stands for the additional writer characteristics collected. To collect an interesting data set, we followed some recommendations

included in a recent article published by Rehman et al. [31]. The same authors noted in another article [32] how few existing data sets have a large enough data size to utilize modern computer vision architecture for writer identification; our data set is a contribution in that aspect.

The HWD+ data set contains 13,580 images from 97 different writers. Images were collected in a high resolution of 500 by 500 pixels in a shades-of-grey format. We also collected various information about the writers. We believe our data set has a weak signal for some variables and thus leave plenty of room for improvement in contrast to the popular MNIST data set where almost all algorithms achieve a good performance and where top-of-the-line algorithms achieve such a high accuracy that it becomes difficult to distinguish their performances.

We believe that the large resolution and the set of writer characteristics collected will lead to interesting new questions and findings. It is a unique data set that could be used in multiple fashions; this is why this section carefully explains how to the data was gathered and processed into the data set now publicly available on the first author's website.

3.1 Data gathering

To begin, we have to state that our data gathering efforts were drastically affected by the 2020 COVID-19 pandemic. The social distancing effort forced us to settle on a smaller size data set with a reduced number of writers. We also ended up with a less diversified set of writers since it was more difficult to randomly sample writers during the pandemic.

Outside of uncontrollable events we made sure to gather an interesting data set in a standardized manner. Every writer was given 2 pages containing a one inch square grid of 10 rows by 7 columns. Writers were asked to fill these pages with digits, 2 rows per digits for a total of 14 replications per digits as seen in figure 1. Every writer was given a new Sharpie pen.

Following this, the pages of handwritten digits were attached to their user identification (ID). Later we collected the following writers characteristics : (1) age, (2) biological gender, (3) height, (4) language learned in elementary school, (5) Hand used to write the digits, (6) education level and (7) main medium used to write. Characteristics(1), (2), (3) and (5) are self-explanatory. For (4) we were interested to find out if different educational system led to different digit writing styles. We initially assume a that there could be a noticeable difference between writers who

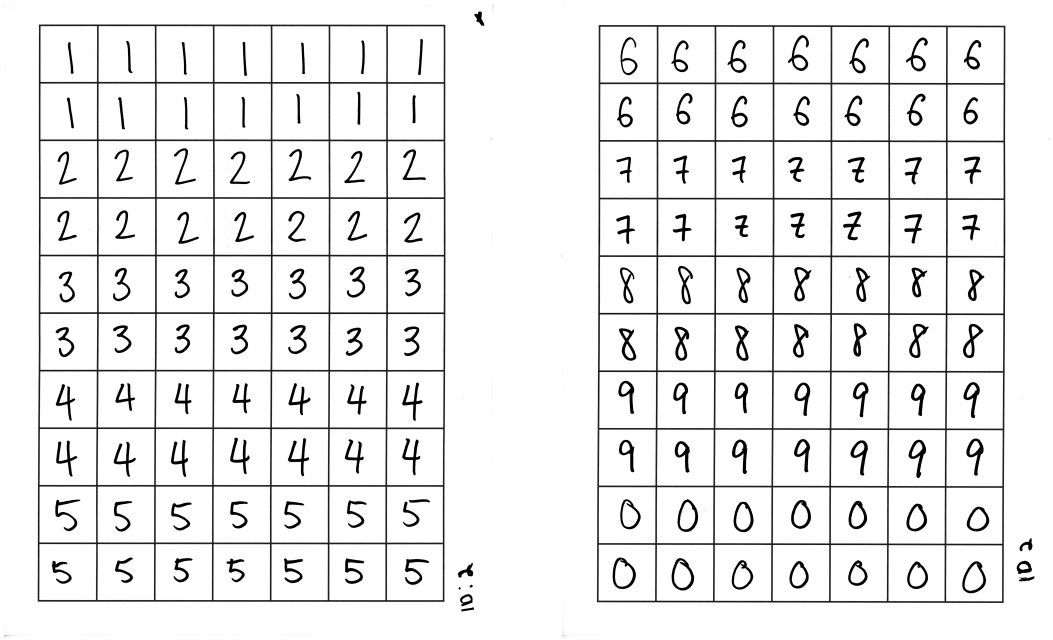


Figure 1: Example of the collected images for a single writer.

were taught with the Roman alphabet and those who were taught a Chinese or an Arabic alphabet. The educational level (6) was encoded as a four-level categorical variables were the first level represents high school, the second level means the writer completed an undergraduate program, the third level is assigned to writers who completed a master's degree or a Ph.D and we finally added a fourth level for young kids who did not complete high school yet. Finally, for the most commonly used writing medium (7), writers were asked to choose between handwriting, keyboard or other where the latest category was commonly cellphone or electronic pen.

As previously mentioned, the COVID-19 pandemic drastically slowed down our data collecting effort and at the moment of submitting this article we are still actively collecting more data. We plan to update our database frequently in the coming months in order to further increase data size. The one currently available contains 97 different writers for a total of 13,580 images.

3.2 Data processing

All of the pages collected were scanned using the same machine with the same settings: shades-of-gray and 600 pixels per inches. These pages were then processed through a script that would take off the edges of the pages and divide the grid into 600 by 600 pixels squares. We trimmed of 50 pixels off the four sides of every image to trim off the actual grid and the result is a collection of 500 by 500 pixels images.

Those images were imported in Python where they were attached to their writer ID, the seven characteristics previously discussed and the digit label. These images are stored as shades of grey images, thus they are composed of a single channel taking values between 0 and 255. When images are scanned, some of the white parts of the images lose some of their purity and thus we have set to 255 every pixel that had a value above 200. The digits were not centred, not scaled and not rotated. These 500 by 500 pixels images form the completed and less processed data set available.

For simplicity we also produced two other data sets with different images size. One data set contains 100 by 100 pixels images. This is still considered to be high resolution but it is much faster to run computer vision algorithms on these images than on their 500 by 500 counterparts. We also produced a data set of size 28 by 28 as it is the size of the images of the popular MNIST data set. This allows researchers to use already existing code set up for MNIST and simply swap data sets. The 28 by 28 data set could also be used in conjecture with the MNIST data set for some interesting semi-supervised project. The fact that it is similar to MNIST but very different at the same time should allow us to understand the problems related to the massive use of MNIST in the recent years. These image compressions were done using the open CV [3] Python library.

One could notice that we have done very few pre-processing compared to other popular data sets. To begin, we believe that size and skwedness are genuine writing characteristics that might contain valuable information about the writer and we did not want to discard that information. Thus, we decided to release the data sets detailed above with as little pre-processing as possible.

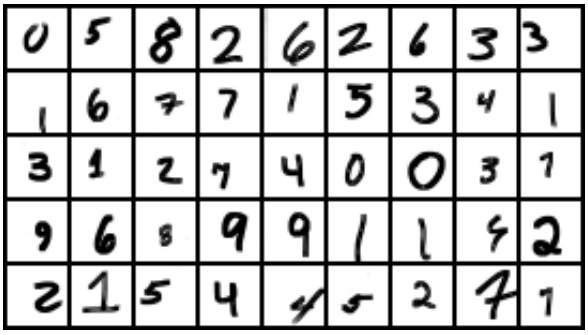


Figure 2: Sample of 45 images.

Figure 2 contains a sample of what the images in the data set look like.

4 Computer Vision Algorithms

Two models are central for our experiments. We will briefly introduce them in this section and explain why they were used. Our analysis is divided in two parts; a supervised learning analysis and an unsupervised learning analysis.

4.1 Convolutional Neural Networks for supervised learning

Multilayer neural networks (NN) have been extremely popular in recent years as a universal function estimator that can be fit using gradient-based approaches. They can be used as a model themselves or as part of other models, for instance they are used in VAEs as explain in the next section. In this project, we will use NNs as prediction function where the inputs are the pixels of an image and the output is either the label, the writer ID or any other variable we are trying to predict. This model will serve as our main supervised learning technique.

In the computer vision field, a special NN structure has been widely used; Convolutional Neural Networks (CNN) [23]. CNNs are extremely well suited for image analysis as its architecture itself is designed to incorporate spatial correlation and some degree of shifts and scale invariance. LeCun et al. [22] identify three structural aspects of CNNs that insure those properties: 1) local receptive fields, 2) shared weights and 3) spatial subsampling.

In a conventional fully connected NN, every input is passed through every node of the next

layer. In the case of image analysis, this results in every pixel of an image being inputs of every function in the first layers, this not only forces the NN to have a large number of parameters (weights) but also neglects the correlation between nearby pixels. In CNNS this is usually taken care of by convolution layers. For these layers, only a small number of nearby pixels is passed as inputs to the next layer as illustrated in Figure 3 below.

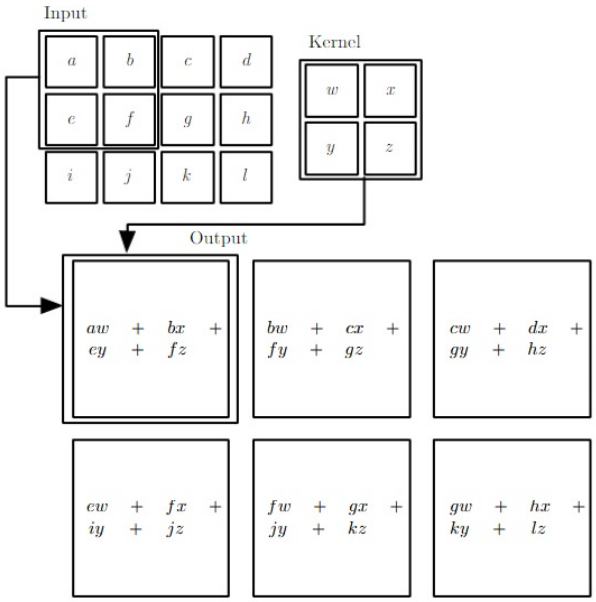


Figure 3: A visual representation of a Convolution layer provided in *Deep Learning* [12]

These layers are said to have sparse connectivity which both reduce the memory requirements and the statistical efficiency of the model. It also means faster prediction as fewer operations are needed to emit a prediction. These layers also contribute towards parameters sharing in this model.

Another typical step in a CNN is pooling. A pooling function replaces the output of a node with a summary statistics of its inputs. For instance, the max pooling operation outputs the maximum of all the inputs. The mean input is another example of possible pooling function. These pooling stages are useful at making the representation invariant to small translation within the image.

Usually a CNN contains multiple convolution layers, multiple pooling stages and fully connected layers. A detailed formulation of CNNs is available on Chapter 9 of *Deep Learning* by

Goodfellow et al. [12].

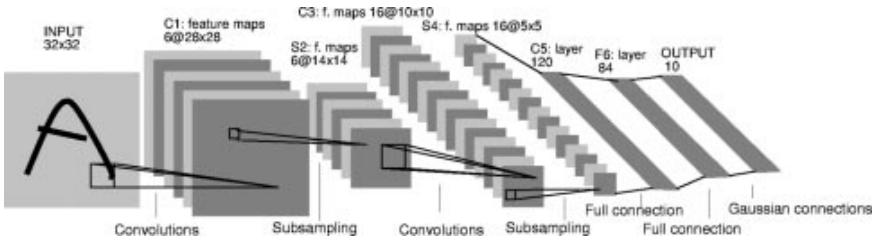


Figure 4: The representation of LeNet-5, a CNN architecture introduced by LeCun [22]

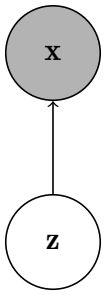
LeNet-5 illustrated in figure 4 was introduced by LeCun in the paper where MNIST was also presented [22]. It contains a succession of convolution layers, pooling stages and conclude with fully connected layer before the 10-level output.

4.2 Variational AutoEncoders for semi-supervised learning

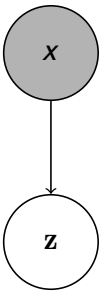
Let us briefly introduce Variational AutoEncoders (VAEs). This model is the results of parallel work from Kingma [18] and Rezende [16] on latent variable models. The MNIST data set is present throughout Kingma's thesis [19] as it provides good visualization of VAE's behaviour. We will employ VAE for semi-supervised learning tasks in our analysis.

An AutoEncoder (AE) simultaneously learns how to encode a high-dimensional observation to a different dimension latent representation and how to decode the latent representation to the full-size observation.

More rigorously let us define $\mathbf{x}$ as an observation of size D, in our case a single image of size 28 by 28 pixels (D=784) and $\mathbf{z}$, a latent representation (code) of size $d << D$. An AE aims to learn an encoding function $q : \mathcal{X} \to \mathcal{Z}$ and a decoding function $p : \mathcal{Z} \to \mathcal{X}$ simultaneously. These functions can take multiple forms and we can define various optimization objective functions.



(a) Generative network. It assumes $p(\mathbf{x}, \mathbf{z}) = p(\mathbf{z})p(\mathbf{x}|z)$.



(b) Inference network. Given observations $\boldsymbol{x}$ we can infer the latent variable using $q(\mathbf{z}|\boldsymbol{x})$.

Figure 5: Graphical representation of the two networks that makes up a VAE

Figure 5 is a graphical representation of the simplest VAE model. In the VAE paradigm, we assume a distribution on the two sets of random variable which leads to a full parametrized model. We start by assuming a prior on $p_\theta(\mathbf{z})$, usually this is an isotropic Normal. Then we assume a decoding distribution $p_\theta(\mathbf{x}|\mathbf{z})$ where the parameters θ are parametrized using a NN, i.e $\theta = NN_1(\mathbf{z})$. Under this model the posterior $p_\theta(\mathbf{z}|\mathbf{x})$ is intractable and thus we rely on variational inference. We define an encoding distribution $q_\varphi(\mathbf{z}|\mathbf{x})$ where $\varphi = NN_2(\mathbf{x})$ that serves as an approximation for the true posterior $p_\theta(\mathbf{z}|\mathbf{x})$.

Typically both p and q are assumed to be Normal distributions but other alternatives have been considered [19]. The system is optimized using maximum likelihood. More precisely, we maximize the Evidence Lower BOunds (ELBO), which is a lower bound of the observed data log-likelihood :

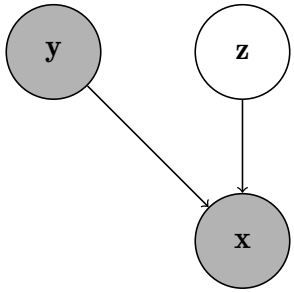
$$\begin{aligned} \ln p(\mathbf{x}) &= \mathbf{E}_{q(\mathbf{z}|\mathbf{x})}\left[\ln p(\mathbf{x})\right] \\ &= \mathbf{E}_{q(\mathbf{z}|\mathbf{x})}\left[\ln \frac{p(\mathbf{x},\mathbf{z})}{q(\mathbf{z}|\mathbf{x})}\right] + \mathbf{E}_{q(\mathbf{z}|\mathbf{x})}\left[\ln \frac{q(\mathbf{z}|\mathbf{x})}{p(\mathbf{z}|\mathbf{x})}\right] \\ &= \mathcal{L}(q,p) + KL(q||p) \geq \mathcal{L}(q,p) \end{aligned} \tag{1}$$

where $\mathcal{L}(q_\varphi, p_\theta) = \mathbf{E}_{q_\varphi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x},\mathbf{z}) - \log q_\varphi(\mathbf{z}|\mathbf{x})]$ is the ELBO that serves as objective function when we train VAEs.

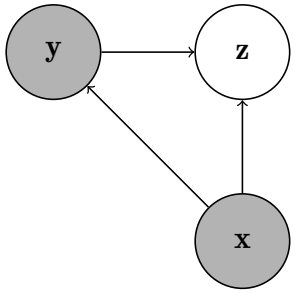
In our experiments we will be working with slight variations of the VAE where we also include a set of selected labels $\mathbf{y}$ such as the digit or the writer ID or the digit. These models were

established for semi-supervised problems. Briefly, the idea is to make use of an unlabelled data set S_u in order to improve the prediction accuracy we would get by simply using the labelled data set S_l. We will also examine the generative abilities of such model; we are curious to find out how much more control over the generative process we gain by adding labels y into the model.

We coded and experimented with two different models. To begin the M2 model proposed by Kingma [20, 19]:



(a) Generative network. It assumes $p_\theta(\mathbf{x}, \mathbf{z}, \mathbf{y}) = p_\theta(\mathbf{z})p_\theta(\mathbf{y})p_\theta(\mathbf{x}|z, y)$.



(b) Inference network. Given observations x and label y we can infer the latent variable using $q_\varphi(\mathbf{z}|x, y)$. When y is missing we can infer is using $q_\varphi(\mathbf{y}|x)$.

Figure 6: Graphical representation of the two networks that makes up the M2 model.

To obtain an objective function for semi-supervised learning we have to consider both label and unlabelled data separately. In the first case, we have the label and the resulting ELBO is :

$$\ln p_\theta(\mathbf{x}, \mathbf{y}) \geq \mathbf{E}_{q(\mathbf{z}|\mathbf{x},\mathbf{y})}\left[\ln p_\theta(\mathbf{z}) + p_\theta(\mathbf{y}) + p_\theta(\mathbf{x}|z, y) - \ln q_\varphi(\mathbf{z}|x, y)\right] = \mathcal{L}(x, y) \tag{2}$$

For unlabelled data :

$$\begin{aligned} \ln p_\theta(\mathbf{x}) &\geq \mathbf{E}_{q(\mathbf{z},\mathbf{y}|x)}\left[\ln p_\theta(\mathbf{z}) + p_\theta(\mathbf{y}) + p_\theta(\mathbf{x}|z, y) - \ln q_\varphi(\mathbf{z}, \mathbf{y}|x)\right] \\ &= \sum_y \left[q_\varphi(\mathbf{y}|x)(\mathcal{L}(x, y))\right] + \mathcal{H}(q_\varphi(\mathbf{y}|x)) = \mathcal{U}(x) \end{aligned} \tag{3}$$

where $\mathcal{H}$ is the entropy of the distribution. The bound on the entire data set is :

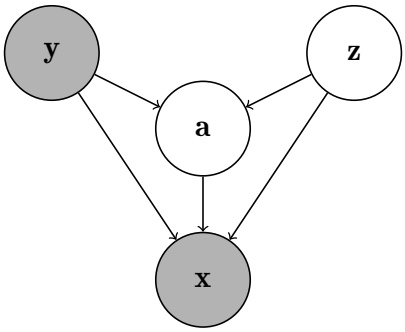
$$\mathcal{J} = \sum_{S_l} \mathcal{L}(x, y) + \sum_{S_u} \mathcal{U}(x) \tag{4}$$

To complete our brief introduction of the M2 model, one might notice that the encoding function used as classifiers $q_\varphi(\mathbf{y}|x)$ only appears in $\mathcal{U}(x)$ and thus is never actually trained on labelled data. To rectify this situation, Kingma proposed to add a term to $\mathcal{J}$ resulting in the following objective function:

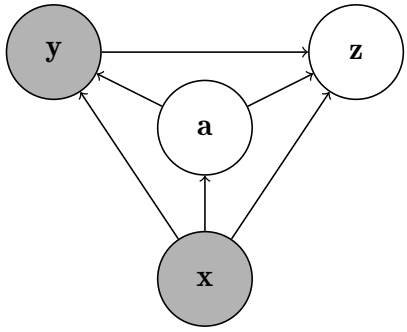
$$\mathcal{J}^\alpha = \mathcal{J} + \alpha \mathbf{E}_{S_l}\left[-\ln q_\varphi(\mathbf{y}|x)\right] \tag{5}$$

where α is an hyper-parameter that controls the relative weight between generative and discriminative learning. The bigger α is the closer we are to obtain the same classifier obtained using strictly the labelled data; in a way the whole VAE machinery can be perceived as regularization that prevents overfitting the training labelled point. More details about the M2 model can be found in various publications [20, 19, 30].

Finally, we have implemented the SDGM proposed by Maaløe et al. [26, 27, 30]. The model relies on auxiliary variables [2] to improve the expressive power of both the inference and generative model.



(a) Generative network. It assumes $p_\theta(\mathbf{x}, \mathbf{z}, \mathbf{y}, \mathbf{a}) = p_\theta(\mathbf{z})p_\theta(\mathbf{y})p_\theta(\mathbf{a}|z, y)p_\theta(\mathbf{x}|z, y, a)$.



(b) Inference network. We can infer y using $q_\varphi(\mathbf{y}|a, x)$ and the latent representation z using $q_\varphi(z|a, x, y)$.

Figure 7: Graphical representation of the two networks that makes up the SDGM model.

Figure 7 is a graphical representation of the SDGM. Similarly the objective function has a component for labelled observations, a component for unlabelled observations and an extra

term to ensure that $q_\varphi(\mathbf{y}|a, x)$ is trained with labelled observations. More details about the SDGM can be found in various publications [26, 27, 30].

5 Experiments

In this section we will tackle both supervised and semi-supervised learning problems. All of our experiments were performed using Python [35] and the Pytorch library [29]. After experimenting with multiple optimizers, we settled on Adam [20].

There are two main purposes for these experiments. First, we want to explore our data set, get to know its structure better, detect some of the patterns there might exist and establish the first benchmarks for some of the classification problems. Second, we want to showcase some of the new problems that can be tackled with this new data set.

5.1 Supervised learning

In this section, we will explore our data by establishing the first benchmarks attainable for various classification task. We approach multiple simple classification problems using three models that were previously successful; we implemented Le-Net5 [22], a deep fully-connected NN based on the work of Ciresan et al. [7], a committee of 25 CNN [6] and finally a committee of 25 deep NN. Le-Net5 [22] was selected as our basic CNN; it is introduced in the paper presenting the MNIST data set. We included a deep NN based on the work of Ciresan et al. [7] who demonstrated that a very deep and large NN performs as well as a CNN for digit prediction. We included a committee of CNN since ensemble models had the best classification accuracy on the MNIST data set. Finally, we included a committee of deep NN for comparative purposes.

Regarding possible new problems we can approach with this new data set, we will assess how higher resolution affects classification performances and how using writer characteristics as predictors can also affect the prediction accuracy. We will not address multi-label classification problems in this article, but this it is a problem that can be tackled on with this data set that couldn't be tackled with MNIST.

For the single Le-Net5 CNN and the deep NN , they will be fit 50 times, were each time we randomized which images are in the training set and the testing set. For the committee of 25

CNNs, we've fitted this model 15 times with once again randomized training and test set for each trial.

5.1.1 Image classification

It is hard to assess the prediction performances of algorithms on a new data set given the lack of comparatives. As already mentioned we wish to assess the first benchmark but also the existence of some signal; thus we will often time compare our results with the *naive classifier*, which we define here as a classifier that always votes on the majority class. Readers are invited to take a look at the descriptive statistic table in the appendix to get a rough idea of the performance of such naive classifier in this analysis.

For some of these experiments we divided our data set into a training set and a testing set in a way that both set contains every writer; the training set contains 10 images of every digit of every writer and the test set contains 4 images of every digit of every writer. We named this process *partitioning by digits*. This partitioning will be used when predicting the digit and the ID. To better assess the actual predictability of the writer characteristics we created another way to partition training data from test data; this time we split training and test sets by participants randomly assigning 70% of the writers to be in the training set and 30% in the test set. This way the females and males in the test set have never been observed. We named this *partitioning by individuals*.

	LeNet-5		Comm. LeNet-5		Deep NN		Comm. Deep NN	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
Digit	0.9399	0.0143	0.9762	0.0013	0.9192	0.0160	0.9340	0.0029
ID	0.3473	0.0136	0.6195	0.0063	0.4268	0.0077	0.5012	0.0049
Gender	0.5367	0.0183	0.5483	0.0372	0.5394	0.0208	0.5309	0.0219
Language	0.6792	0.0322	0.7621	0.0626	0.6752	0.0604	0.7149	0.0408
Hand	0.7940	0.0285	0.8304	0.0499	0.7973	0.0275	0.8232	0.0377
Education Level	0.4117	0.0222	0.4726	0.0343	0.4147	0.0393	0.4253	0.0405
Writing Medium	0.4585	0.0225	0.4782	0.0372	0.4668	0.0189	0.4714	0.0249

Table 1: Prediction accuracy for simple classification tasks

In the table 1 we can see that this data set still has a very high signal with respect to the digit. Even with a data set of much smaller size and less pre-processing than MNIST a committee of CNNs reaches close to 98% accuracy on average. Strictly for digits classification our data set is a good alternative to MNIST and should allow researchers to easily assess the impact of using a pre-processed data or not on various classification performances.

Next, for writer identification the performance of the committee is quite impressive in table 1, accurately predicting the writer 62% of the time, given we have a pool of 97 writers and only have hand-written digits this is quite impressive and way above the performances of the naive classifier.

Next, let us predict various characteristics collected. As we previously discussed, we used a different data set partitioning for the writer characteristics. The reason is quite simple; the CNN techniques are so good at predicting distinct style related to IDs, as shown by the high performance of the committee, that the model could map the IDs to a gender. This is not exactly identifying writing patterns that are different based on the biological gender. Thus, we implemented *partitioning by individuals* for writer characteristics to make sure the algorithm actually tries to learn an effect of the response on the writing styles that are shared among writers. We were achieving results much higher than observed in table 1, confirming our suspicions that the algorithms indirectly learned individual styles and associated them to a gender rather than actually learning an image to gender pattern.

Going back to the accuracy when predicting writer characteristics. Most of the results are worst or even at best with the naive classifier who simply selects the majority class. However, we notice that the improvement when using a committee of CNNs is statistically significant when predicting every variable except gender and writing medium in table 1; thus there might some signal for native language, handiness and education level. The improvement in table 1 is specifically important when predicting the writer ID; almost doubling the predictive performance over the simple LeNet-5. We believe that the variables for which we observe a significant increase in prediction accuracy when using a committee are predictable in a way and that this improvement is a consequence of the existence of a signal. There might be some way to further improve the prediction accuracy in order to surpass the naive classifier for those classification tasks.

We believe we achieved one of our main goals to create a data set with variables with various predictability: we can achieve high accuracy when predicting the digit, the ID seems to lead to widely different prediction performances but is definitely predictable, some characteristics, such as native language, handiness and education level, are weakly related with the images and finally the gender and usual writing medium seem to be impossible to predict using only digit images.

We also notice the good performances of the deep NN which supports the results of Ciseran et al. [7]. We also included committees of deep NNs to better understand the difference between LeNet-5 and the deep NN. We know that unstable algorithms tend to benefit more from aggregating and committees [4] and here we clearly see that LeNet-5 benefits more from the aggregation than the deep NN. This would suggest that fully connected deep NNs are more stable than CNN classifiers. In other words, with slightly different data, CNN classifiers then to be more different than deep NNs who stay relatively similar.

5.1.2 High resolution images classification

In the next two sections, we want to experiments with specific tasks that are enabled by our new data set. In this section we assess the effect of higher resolution images on the classification performances of CNNs. Being able to provide users with images as high solution as 500 by 500 pixels is something offered by very few data sets that often contain very small images. However, in this section what we call high-resolution images are 100 by 100 pixels images.

We observe a statistically significant increase in prediction accuracy when predicting the digit, the writer ID, the first language, the writer handiness and the education level of the writer in table 2. These are the exact same variables for which the committee also improved on the benchmark LeNet-5. Even though we observed predictive performance equivalent to the naive classifier for some of those variables, those improvements when using a committee or higher resolution images lead us to believe that those variables are predictable in some way. At least, there exist some signal between the images and those variables.

The results here are intuitive: the classifier benefits from higher resolution images since they are richer in information. However, it should not be surprising that it also increased the computational challenge. For instance, we could not fit committees of LeNet-5 classifiers on the high-resolution images on a single GPU (GeForce RTX 2070 Super 8Gb Ram) due to lack of

	LeNet-5 (28x28)		LeNet-5 (100x100)	
	Mean	Std	Mean	Std
Digit	0.9399	0.0143	0.9683	0.0044
ID	0.3473	0.0136	0.3675	0.0224
Gender	0.5367	0.0183	0.5354	0.0410
Language	0.6792	0.0322	0.7284	0.0441
Hand	0.7940	0.0285	0.8129	0.0355
Education Level	0.4117	0.0222	0.4466	0.0368
Writing Medium	0.4585	0.0225	0.4612	0.0234

Table 2: Prediction accuracy for simple classification tasks on low-resolution images compared to high-resolution images

memory. We can get around these problems by sending our tasks to servers online but we think it is important to make sure the algorithms we develop can run on a single computer as well; it makes the algorithms truly available to a broader audience. Additionally, as data sets get larger and larger we have to address the scalability of such algorithms. This data set offers the opportunity to analyse such scalability on a simple digit prediction task.

We can compare the gains made from richer information to the gains made from *better* algorithms. Here we see that the gains from using an ensemble of classifiers on the low-resolution data sets are bigger than the gain from using the simple technique on a richer data set. Of course we expect a committee of LeNet-5 trained on the richer data set that have even bigger gains but this is not the point here. We evaluate the benefit from using a certain algorithm over another compared to the raw gain from a richer data set; here the predictive improvement of an ensemble technique is higher than the improvement from getting a data set with twelve times as many pixels.

5.1.3 Image classification with predictors

In this section we will include some of the collected information as predictors to see how it changes the performances of the Le-Net5 classifier, once again something new that our data set enables. Moreover, we are interested in understanding the potential contribution of additional

information in images classification. For instance, we believe it would be a contribution to forensics if we establish that providing the digit (or the word) to the algorithm increases the accuracy when predicting the writer.

We will experiment with two simple tasks: in the first experiment we will try to classify images according to their digit and will incorporate the writer ID as an additional predictor and next we will do the reserve, we will classify images according to the writer ID while including the digit as an additional predictor. To do so, we include a one-hot encoding vector for writer ID or the digit in the first fully connected layer of LeNet5. In other words, the additional predictors are incorporated immediately after the convolution layers; the vector of predictors is concatenated with the vector C5 of figure 4. For this experiment, we *partition by digits* the data set.

	Images (LeNet-5)		Images + (LeNet-5)		Images (Com.)		Images + (Com.)	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
Digit	0.9399	0.0143	0.9551	0.0080	0.9762	0.0013	0.9812	0.0020
ID	0.3473	0.0136	0.3575	0.0192	0.6195	0.0063	0.6003	0.0042

Table 3: Prediction accuracy for simple classification tasks when using only the image as predictors (Images) or the image and an additional predictor (Images +)

This time we observe something slightly unexpected: including the writer ID as additional information significantly increase the accuracy when predicting the digit. However, including the digit when predicting the ID actually decreased the prediction accuracy of the committee.

Regardless, these results are encouraging and could motivate further investigation. For instance, there exist multiple way to integrate additional information in a CNN classifier and this data set offers an opportunity to explore those.

5.2 Semi-supervised learning

In this section, we tackle two semi-supervised tasks using the two semi-supervised learning model introduced in section 4.2: the M2 model presented by Kingma and Wellington [20] and the SGDM model established by Maaløe et al. [26, 27, 30]. Our first problem is to perform a

semi-supervised analysis where we use the MNIST data set as unlabelled observations. We demonstrate an improvement in the prediction accuracy for the labels we collected in our data set when using the M2 model and the additional unlabelled data from MNIST.

The second task we focus on is image generation. We will briefly discuss and demonstrate the generative abilities of the SGDM model using our data set. The multiple labels allows us to turn multiple *control knobs* which imbues the generative process with much more control; thus we make a significant contribution in what we call *controllable content generation*.

5.2.1 Semi-Supervised classification

Here we use the M2 model described in section 4.2 to predict the Digit and the ID in our images while increasing our data set size with some unlabelled images, the MNIST data set. In our implementation of the M2 model $q_\varphi(y|x)$ is parametrized by a LeNet5 CNN. We assess the improvement produced when including new unlabelled data compared to the results previously obtained in section 5.1.1 when using a single LeNet5.

This gives us a great perspective on semi-supervised classification. It is said that it is possible to leverage unlabelled points from other data sets to improve the accuracy over the simple classifier and we have argued this is due to some regularization. However fitting the compression and decompression machinery does increase the run time needed to fit such semi-supervised model.

	LeNet-5		M2	
	Mean	Std	Mean	Std
Digit	0.9399	0.0143	0.9542	0.0060
ID	0.3473	0.0136	0.4174	0.0099

Table 4: Prediction accuracy of the semi-supervised M2 model trained on the HWD+ and MNIST data set compared to LeNet-5 trained on HWD+.

Table 4 shows a significant increase in accuracy when using the semi-supervised model. These results are impressive given how standardised the MNIST data set is; it is widely different from our data set and the standardize procedure reduces the difference between writers. As we previously discussed the second term of the objective function presented in equation 5 trains

the classifier on labelled data and is precisely what we trained in previous sections. Further investigation on how the first term serves as regularization should lead to interesting results.

More generally, we believe those results are extremely encouraging; theses models could be better tuned to the task at hand. We will also investigate further in a subsequent research project the idea of forming committee of classifiers fit under the semi-supervised paradigm.

5.2.2 Generative perspectives

In this section we want to showcase the opportunity, our data set offers for controllable (conditional) image generation. In the example below we have fitted the SDGM model described in section 4.2 with both the ID and the Digit as labels $\mathbf{y}$. Since the model is fitted for generative purpose, we use all of our data points, which are labelled, and the classifier $q_\varphi(x)$ is completely irrelevant here. What we truly want, is to train $p_\theta(\mathbf{x}|z, a, y)$ to generate images that are both good looking and that respect the conditions imposed by $\mathbf{y}$, in other word the image contains the right digit and style. Other details of the images randomized through $\mathbf{z}$ and $\mathbf{a}$.

To showcase our results we have produced a series of figure, each one represents a different ID, the first four columns is a sample of four real images and the six following columns are generated images. We have selected the digits one, two, four, seven and nine has they exhibit large difference in style from one write to another.

Figure 8: Generated images for ID #12

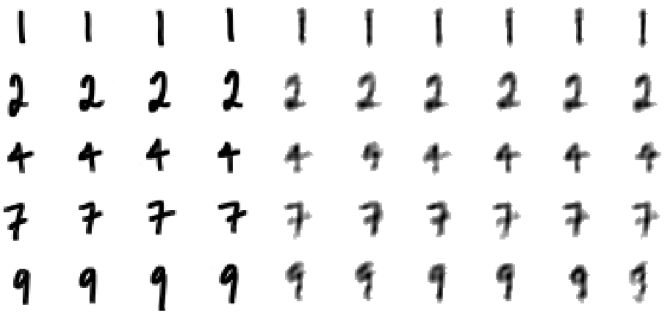


Figure 9: Generated images for ID #14

Figure 10: Generated images for ID #29

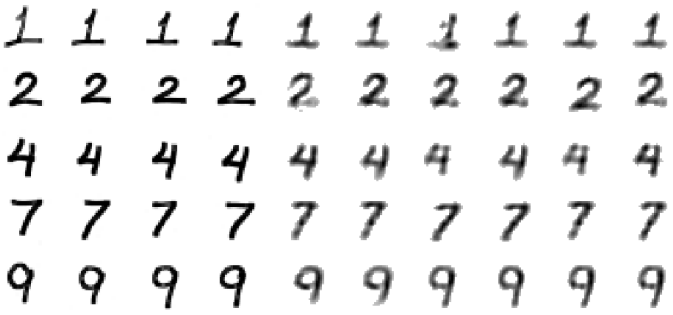


Figure 11: Generated images for ID #70

The generator seemed to have learned very well the effect of the digit input. We see that the generated digits are distinguishable and are appropriate. This was to be expected based on previous experiments [20, 19].

Additionally, the SDGM also learned very well the writing styles of the various writers as observed in figures 8,9,10 and 11. We observe that the size of generate images respect the size of the true images as well as multiple details such as serifs and angles. For instance, the images of *ones* generated by the SDGM model has serif for ID #12 and ID #70 and not the other two. Similarly the images of *fours* are either open or close depending on the style of the writer. Moreover, sometimes the *seven* has a little bar, sometimes the tail of the digit *nine* is curved and more. Overall this is quite a controllable generation success. We already knew it was possible to generate images of a specified digit but the writer ID is something more subtle and those images prove that the VAE model is able to grasp and mimic what makes the writing styles different.

The images are a bit blurry but this is a well-known problem for VAE generated images [33, 15, 10, 9] and a problem we are not trying to fix in this article. The generative process could be further improved with new upcoming VAE structure [10] or other generative models such as GAN [13] which do not suffer as much from the blurry images problem.

These results are preliminary but they highlight the capacity of some well-developed generative model to grasp subtle writing styles and the opportunity that our data provide to experiment with such generative models.

6 Conclusion

In this article we introduced a brand new data set, HWD+, which contains about 10 000 high-resolution images of hand-written digits attached to a set of labels containing the digit, the writer ID and various writer characteristics. The data set has been carefully collected and processed and is publicly available online.

We have done a first analysis of the data set; we have shown that our data contains variable with different predictability making it an excellent benchmark for testing new computer vision algorithms. We also especially considered classification tasks that were made possible with

our new data set such as including additional predictors in classification tasks or using higher-resolution images.

We have also done a few unsupervised and semi-supervised analysis. We have shown the potential use of our data set in a semi-supervised classification task in tandem with the MNIST data set; the SDGM model produced a more accurate LeNet-5 classifier. We have also shown the potential of our multi-label data set for controllable image generation.

Our primary goal for the future is to invest more time developing controllable generative models and we will use our data set to do so. We believe our data set is the perfect testing ground for new creative controllable generative models. Additionally, we will investigate further the benefits of integrating MNIST for semi-supervised task given the positive results we've obtained so far.

Acknowledgement

The authors gratefully acknowledge the financial support from the Natural Sciences and Engineering Research Council (NSERC) of Canada and the Ontario Student Assistance Program. The authors would also like to recognize the contribution of the 150 participants who returned data sheets; without them this new data base wouldn't exist.

References

[1] C. Adak, B. B. Chaudhuri, and M. Blumenstein. An empirical study on writer identification and verification from intra-variable individual handwriting. *IEEE Access*, 7:24738–24758, 2019.

[2] Felix V Agakov and David Barber. An auxiliary variational method. In *International Conference on Neural Information Processing*, pages 561–566. Springer, 2004.

[3] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[4] Leo Breiman. Heuristics of instability and stabilization in model selection. *Ann. Statist.*, 24(6):2350–2383, 12 1996.

[5] Sung-Hyuk Cha and Sargur N Srihari. Assessing the authorship confidence of handwritten items. In *Proceedings Fifth IEEE Workshop on Applications of Computer Vision*, pages 42–47. IEEE, 2000.

[6] Dan C. Ciresan, Ueli Meier, and Jürgen Schmidhuber. Multi-column deep neural networks for image classification. *CoRR*, abs/1202.2745, 2012.

[7] DC Ciresan, U Meier, LM Gambardella, and J Schmidhuber. Deep big simple neural nets excel on handwritten digit recognition. corr abs/1003.0358 (2010). *Google Scholar Google Scholar Digital Library Digital Library*.

[8] Dennis Decoste and Bernhard Schölkopf. Training invariant support vector machines. *Mach. Learn.*, 46(1–3):161–190, March 2002.

[9] Garoe Dorta, Sara Vicente, Lourdes Agapito, Neill DF Campbell, and Ivor Simpson. Structured uncertainty prediction networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5477–5485, 2018.

[10] Garoe Dorta, Sara Vicente, Lourdes Agapito, Neill DF Campbell, and Ivor Simpson. Training vaes under structured residuals. *arXiv preprint arXiv:1804.01050*, 2018.

[11] Yuying Ge, Ruimao Zhang, Lingyun Wu, Xiaogang Wang, Xiaoou Tang, and Ping Luo. A versatile benchmark for detection, pose estimation, segmentation and re-identification of clothing images. *CVPR*, 2019.

[12] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[14] David J. Hand. Classifier technology and the illusion of progress. *Statist. Sci.*, 21(1):1–14, 02 2006.

[15] Huaibo Huang, Ran He, Zhenan Sun, Tieniu Tan, et al. Introvae: Introspective variational autoencoders for photographic image synthesis. In *Advances in neural information processing systems*, pages 52–63, 2018.

[16] D. Jimenez Rezende, S. Mohamed, and D. Wierstra. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. *ArXiv e-prints*, January 2014.

[17] Daniel Keysers, Thomas Deselaers, Christian Gollan, and Hermann Ney. Deformation models for image recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(8):1422–1435, 2007.

[18] D. P Kingma and M. Welling. Auto-Encoding Variational Bayes. *ArXiv e-prints*, December 2013.

[19] Diederik P. Kingma. *Variational Inference & Deep Learning : A New Synthesis*. PhD thesis, Universiteit van Armsterdam, 10 2017.

[20] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[21] RICHARD J. KLIMOSKI and ANAT RAFAELI. Inferring personal qualities through handwriting analysis. *Journal of Occupational Psychology*, 56(3):191–202, 1983.

[22] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[23] Yann LeCun. Generalization and network design strategies. 1989.

[24] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[25] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[26] Lars Maaløe, Casper Kaae Sønderby, Søren Kaae Sønderby, and Ole Winther. Improving semi-supervised learning with auxiliary deep generative models. In *NIPS Workshop on Advances in Approximate Bayesian Inference*, 2015.

[27] Lars Maaløe, Casper Kaae Sønderby, Søren Kaae Sønderby, and Ole Winther. Auxiliary deep generative models. *arXiv preprint arXiv:1602.05473*, 2016.

[28] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[29] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[30] Rastin Rastgoufard. Multi-label latent spaces with semi-supervised deep generative models. 2018.

[31] A. Rehman, S. Naz, M. I. Razzak, and I. A. Hameed. Automatic visual features for writer identification: A deep learning approach. *IEEE Access*, 7:17149–17157, 2019.

[32] Arshia Rehman, Saeeda Naz, and Muhammad Imran Razzak. Writer identification using machine learning approaches: A comprehensive review. *Multimedia Tools Appl.*, 78(8):10889–10931, April 2019.

[33] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In *Advances in neural information processing systems*, pages 3483–3491, 2015.

[34] Sargur N Srihari, Sung-Hyuk Cha, Hina Arora, and Sangjik Lee. Individuality of handwriting. *Journal of forensic science*, 47(4):1–17, 2002.

[35] Guido Van Rossum and Fred L Drake Jr. *Python tutorial*. Centrum voor Wiskunde en Informatica Amsterdam, The Netherlands, 1995.

[36] Yu-Jie Xiong, Yue Lu, and Patrick S. P. Wang. Off-line text-independent writer recognition: A survey. *International Journal of Pattern Recognition and Artificial Intelligence*, 31(05):1756008, 2017.

Appendices

Table of descriptive statistics

Biological Gender	Male 46	Female 51		
Handiness	Right 84	Left 13		
Language (education)	French 75	English 16	Other 6	
Education Level	No high school 7	High school 13	Bachelor 55	Graduate 22
Usual writing medium	Hand 44	Keyboard 45	Other 8	

Table 5: Table of occurence at the time of submission. The data set contains 97 writers.