Deep Learning for Historical Document Synthesis

Master Thesis

Michaël Diatta

University of Fribourg
Boulevard de Perolles 90 - 1700 Fribourg - Switzerland

January 3, 2019

Abstract

This work tackles a particular Image-to-Image translation problem. The goal is to learn a mapping function from an input image to an output image, where the source domain (an image of a standard electronic document) is different from the target domain (an image of a historical handwritten document). The purpose is to generate an artificial historical document image that looks like a real historical document; for non-expert eyes as well as expert eyes, by transferring the "historical style" to the classical electronic document. By completing this task, it becomes possible to consider the generation of a tremendous amount of synthetic training data using only one single deep learning algorithm. Which constitutes a first step towards the creation of a massive synthetic dataset in historical documents field.

Contents

Aknowledgments

In this section, I would like to express my gratitude to the amazing two Assistants, Michele Alberti, and Vinaychandran Pondenkandath, for their friendly guidance and patience during my thesis research. I would like also thanks Prof. Dr. Marcus Liwicki, for his insights and confidence and the whole DIVA group members for their feedback during the presentation session, which was a difficult exercise for me. Finally, I would like to thanks all the people that support me during this long tail.

List of Figures

List of Tables

1 Introductory Part

This Master project called *Deep Learning for Historical Document Synthesis* has been done in the DIVA research group at the University of Fribourg, which is active in document image processing. The overall project was supervised by three members of the DIVA group, Michele Alberti, Ph.D. Student, by Vinay Chandran Pondenkandath, Ph.D. Student and finally by Marcus Liwicki, Professor at the University of Fribourg.
In the past decade, Deep Learning approaches have shown impressive results especially in computer vision tasks like image segmentation, object recognition or even image synthesis. This state of affairs is mainly due to the advent of powerful computers, to the accessibility of more massive datasets, but also to the improvement of algorithmic techniques that allow training deeper networks [11].

Unfortunately, in the field of Document Image Analysis (DIA), and more precisely for Historical Handwritten Documents Analysis (HHDIA), ground-truthed document image datasets can be a scarce resource; in such way that when it comes to take advantages of the Deep Learning breakthrough this lack of data can become a significant issue.
Indeed, a lot of DIA methods which could use Neural Networks as a learning framework to take benefits of the performance gain mentioned above require training procedures and evaluation systems that need a lot of ground-truthed documents. One alternative for using such DIA methods like Image Enhancement, Layout Analysis (region segmentation and classification), Optical Character Recognition (OCR), Binarization, or Image Restoration, conjointly to Neural Networks, is to use Annotation Softwares. Such type of software allows producing ground-truth more or less manually, after the digitalization of Handwritten Document. However, besides annotation strategies, it remains another possibility to obtain massive Historical Handwritten Document dataset: generate synthetic document images with the help of a controlled ground-truth framework.

In this Master Thesis, we want to show how it is possible to go one step further in the history of Image Synthesis Software for Historical Handwritten Documents. Contrary to several traditional methodologies, presented by the community for many years now, we try to produce a new generating framework that will take the advantages of the recent advancements in the design of generative models, but also of some style transfer algorithms. The goals of this thesis are:

- First, to brush a comprehensive panel of different architectural tracks that could lead to a more simple and more general-purpose image synthesis framework.
- Secondly, to show that the possibility of creating unlimited amounts of complex synthetic handwritten historical documents based on different ground-truthed electronic documents is a reachable goal.
- Finally, proposing and implementing three kinds of different algorithmic approaches that could lead to the draft of a practical solution, with satisfying results.

In the rest of this introductory part, we will circumscribe our Master Thesis task to what we call an Image Synthesis problem, addressing the notions of Transfer Learning, Domain Adaptation, and Image-to-Image translation. Then we will generally speak about the two main algorithmic solutions that we propose us to explore. In the second part, we will be concerned by generative models and more particularly by Generative Adversarial Networks (GANs). Finally, we will explore the idea of Neural Style Transfer, the second deep learning solution that also seems to suit the Image-to-Image translation problem we want to solve.

1.1 An Overview of the Image Synthesis Problem

In simple word, what do we expect from our deep learning framework? Basically, given two types of training datasets:

- The first one composed by classical electronic documents images called the Source Domain (see samples in Figure 18).
- The second one composed by historical handwritten documents images called the Target Domain (see samples in Figure 17).

We want to learn a mapping function that goes from the Source Domain S to the Target Domain T and vice versa. Indeed, as mentioned above, our primary purpose is to *generate synthetic historical handwritten document images* where these synthetic document images look like the original document images that come from the Target Domain.

Technically the problem is defined as an image synthesis problem with special features, where these features lead us to a particular type of transfer learning problem called Domain Adaptation, where the final goal is to perform an Image-to-Image translation task.

1.1.1 What is Transfer Learning ?

As a human being one of our characteristics is that we can retain and reuse previously learned knowledge to solve new problems. For example, learning to drive a moped at sixteen will probably help us to drive a motorbike at twenty because hopefully, we already know the Highway Code and master the gearshift. For deep learning methods, in a way, the same can occurs. As statistical models traditional deep learning technologies need to be trained on previously collected data to learn a feature space and a distribution. The learned feature space and a probability distribution represent the learned knowledge. Traditionally, when we build a deep learning framework, we train and test data that come from the same feature space and the same distribution. Conversely, transfer learning techniques are characterized by the fact that during the training and testing procedure they allow tasks and/or collections of datasets - what in the following we will call domains - and distributions to be different [29].

In the transfer learning task that we want to perform, the settings are the following. The source and the target tasks are the same, while the source and target domains are different. In the source domain, labeled data are available because our proposed framework can generate the ground-truth labels, but we have no labeled data in the target domain. Indeed, keep in mind that our purpose is to generate synthetic data with controlled ground-truthed data. We can also notice that the feature spaces between the source and the target domains are different. Technically this particular type of setting is defined by the term *transductive transfer learning* and more precisely by the term *Domain Adaptation* [29].

Now we have a better understanding of what we call a Transfer Learning task that involve a Domain Adaptation goal, we can refine our explanation and say that we are in front of a conditioned image generation problem where the goal is to generate an image in one domain given another image in the other domain, which can be defined by what in the literature [15] we recently call an Image-to-Image translation task.

1.1.2 What is an Image-to-Image Translation Task?

We can define an Image-to-Image translation problem as an image constrained synthesis process, which consist in accomplishing a pixel to pixel prediction task [15]. The goal is to transform an image represen-

tation into another plausible image representation, like for example in Figure 10.

By accepting this definition, several computer vision problems like colorization tasks, super-resolution tasks, segmentation tasks or depth estimation tasks, can be grouped under one category of problem [29, 35]. Then, instead of tackle all these particular tasks separately, designing specifics deep learning frameworks with specifics losses to find a proper way to reach a particular goal, one more clever solution is to define, a single but general enough loss function with an appropriate learning framework, like proposed in [15].

This general purpose framework that generates realistic synthetic images is called Generative Adversarial Network (GAN) [12, 14]. The other solution is to restrict our domain adaptation problem to a style transfer problem, where the goal is to learn to separate the content from the style for two given domains, to produce a synthetic output that mixed the two characteristics in one single image representation, which is called Neural Style Transfer Algorithm (NST) [9].

1.2 A Walk in the Land of Generative Adversarial Networks

Generative Adversarial Networks were introduced by Ian Goodfellow et al. in 2014 [12]. GANs technologies have encountered great success in the research area this past four years, mostly because they can generate better synthetic images than previous generative models [14]. Conceptually, think of GANs as two models of neural networks, a generator, and a discriminator, that allow generating synthetic data as output, given some input data.

However, before going too far into GANs frameworks details, we will first explain why the research community is interested in studying generative models in general and how they work. Then we will compare GANs to other generative models, explaining how they work and why they are well suited for our task.

1.2.1 The Goal in Studying Generative Models

If the goal of a machine learning algorithm is to incrementally learn from the input data he faces with the purpose of improving his performance regarding a given task [28] then, we can say that a well-suited objective for a learning algorithm is to learn how to create the data itself [14]. Generative models propose to do so:

- By learning to discover the rules that underlie the data, finding the most promising distribution to represent them, or even better...
- By directly producing samples that follow the learned distribution mimicking the real one.

For us, as in GANs NIPS 2016 tutorial session [10, p. 2], the term "generative model" is defined by several features, by:

1. p_{data} a probability distribution that describes training samples that come from a training set.
2. p_{model} an estimated probability distribution that represents the real distribution. This distribution is build by the generative models.
3. The generative models are only able to estimate p_{model} explicitly or/and are able to generates samples from p_{model}.

There are several reasons to study generative models. One of them is the possibility to be in front of multi-modal data modeling task. Where we define a multi-modal task by the fact that given one single input data the output result can lead to many different but correct solutions, like for example in future video prediction work [10].
Generative models also provide solutions for dealing with missing data. For example, if the only way to train a model is to work with missing data, like in semi-supervised learning, generative models can perform very well to fill the blank [10].
That is not all; we can also see generative models as tremendous manipulating and testing frameworks when it comes to work with high-dimensional probability distributions [10].
However, in this thesis, we are mostly interested in them by their ability to generate realistic samples of Handwritten Historical Documents [10].

After clearing what generative models are and why it is worth to study them, we will focus on how they work.

1.2.2 A Generative Models Taxonomy

By circumscribing our concern on generative models that work on the maximum likelihood principle, like GANs we can say that we are trying to define a model that provides us the better estimation of a parameterized probability distribution [12, 10]. Given a training dataset where the goal is to increase the likelihood between a probability model distribution and a probability training dataset distribution where:

- n represent the number of training samples in the dataset
- X, the training dataset
- $x^{(i)}$ a training sample coming from the training dataset
- θ the parameters of the model
- $\mathbb{E}_{x\sim p_{data}}$ the esimate of the trainging data probability distribution, we can express the probability model distribution by the following equation:

$$\prod_{i=1}^{n} p_{model}(x^{(i)};\theta) \tag{1}$$

By transfroming the formula on a log space we obtain:

$$\theta^* = \arg\max_{\theta} \sum_{i=1}^{n} \log p_{model}(x^{(i)};\theta) \tag{2}$$

$$\theta^* = \arg\max_{\theta} \mathbb{E}_{x\sim p_{data}} \log p_{model}(x|\theta) \tag{3}$$

The main difference between deep generative models consists in the way that models estimate the likelihood between the training data probability distribution and the modeled one. We can separate the models into significant categories, following [10, pp.9 - 17].
The first ones are models that try to construct the probability distribution of the models explicitly. The learning objective is then to maximize this explicit density function. As we know a density function can be either tractable in polynomial computational time or intractable. If the density is untractable, the models are constructed to find a way to approximate this density function. Which separate these explicit models in two subcategories.

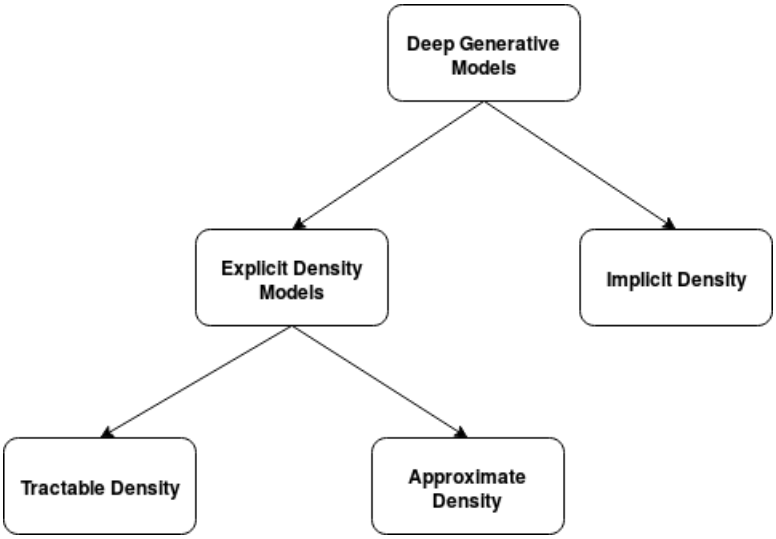


Figure 1: **Generative Models Taxonomy.** Taxonomy inspired by [10].

The other family of models uses an indirect way of working in concert with the probability distribution. We call these models implicit density models. By generating directly samples from p_{model} by transforming stochastically the current sample of p_{data} to generate a new one that should ideally come from the same distribution, like Markow Chain based models. The other subcategory of this model family proposing to generate directly new samples starting from random noise, GANs models are part of this subcategory of models. Nevertheless, GANs are flexible. It is also possible to use GANs to build explicit density. The following Figure shows the taxonomy[10].

To summarize the main differences and advantages between GANs technology and the other types of generative models [10, p. 17], we can say that:

- GANs are able to generate samples directly from the p_{model} distribution, which allows us to be independent of any representation of p_{data}.
- GANs produce, in general, better instances than other generative models.
- GANs are less constraining than other family models present in the taxonomy as it is about to design the generator function.

The main drawback coming with GANs frameworks is regarding the training procedure. Problems like mode collapse and the inability to find an equilibrium in the min-max game are well documented in the literature [12, 10, 30]. Later in this paper, we will mention the tricks we used to avoid these kinds of problems.

1.2.3 What is the Mechanics of GANs?

In this section, we will explain the critical concepts behind GANs theory. Then we will show how these concepts apply to our task.

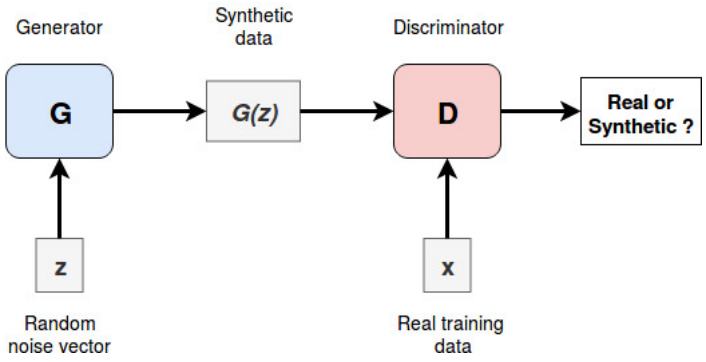


Figure 2: **A classical GAN framework.**

Two neural networks composed a GAN framework. One network is called Generator, designated by G and the other one is called Discriminator designated by D. The role of the generator is to generate new synthetic data instances while the discriminator evaluates the authenticity of instances by estimating whether an image input comes from the real training dataset or if the image input is synthetic and therefore generated by G.

We say that the two models, generator and discriminator are trained via an adversarial process because both models are trainable simultaneously through backpropagation playing what we call a two-player min-max game. We call this adversarial training a min-max game because there is two different training objectives. The discriminator D is trained to maximize the probability of classifying correctly the real images coming from the training dataset ,just as much as synthetic samples coming from G. While the generator G is trained to minimize the probability of the discriminator to make a mistake, in other words, G is optimal when D cannot distinguish real samples from a fake one [12, 10]. The main components of a traditional GAN framework are presented in the following Figure 2.

- The Generator G takes as input z a noise vector - $G(z)$.
- G is parametrized by θ - $G(z;\theta)$.
- G generates a sample as output. The synthetic instance is drawn from the model distribution p_{model} where $G(z;\theta)$ p_{model}.
- The training objective of G is to approximate p_{data}, usign p_{model}.
- D takes as input either $G(z;\theta)$ or x a training data sample coming from p_{data}, the training distribution, wich will be our case.
- $D(G(z;\theta))$ or $D(x)$ output a probability that indicate wheter the output is synthetic (0 value) or if it comes from the true distribution (1 value) .

We can now express the min-max game by the following value function, like in [12]:

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]. \tag{4}$$

In the following sections, we will call this value function the classical GAN loss function designated by the symbol: $\mathcal{L}_{GAN}$

In our case, we have a training dataset containing images of Historical Handwritten Documents, and we would like to generate synthetic images of this type of Historical Handwritten Documents. Then the role of the generator G is to generate the new synthetic document images while the discriminator's goal is to determine if the document images come from the real dataset or if they have a synthetic origin.

How conditional generative adversarial networks (cGANS) work?

We have seen that in the classical GAN implementation the output of the generator is only dependent of z, the random noise vector, implying that we have no control over what we want to be generated by the network. To improve the GAN control one solution is to add a conditional input c to the random noise vector. This technique was proposed in [27]. The synthetic image is then define by $G(c, z)$ instead of $G(z)$. No other changes occur in the overall GAN framework. The classical use of c is to add supplementary information like, textual information about the corresponding image that we want to generate for applications such text-to-image synthesis [14], or like the class of images for usual direct image synthesis task like in [4].

The Conditional Generative Adversarial Networks possibility to replace or add supplementary information as input to our GAN framework is related to what we try to achieve in this thesis. For recall, we want to give to our generator an image of an electronic document which belongs to our source domain instead of a random noise vector, this image can be conditioned by another ground-truth image that comes from the target domain - if the task is supervised for example, which is not our case but an for historical development purpose it is important to fully understand this point. Thus, we can see the ground-truth label for an image source domain as a conditional input that is part of a particular probability distribution in Figure 9. See the section 2.2 to fully understand this point now, it will be clearer in several pages.

1.3 Beside GANs, Another Possibility: Neural Style Transfer

There exist another technique that can tackle the Image-to-Image translation problem we face, at least theoretically. This method is called Neural Style Transfer algorithm.

We can think of this technique as a solution for a subcategory of an Image-to-Image translation task that focuses on texture transfer [9]. This solution is less general than a GAN one. The solution of this method consists of separating and recombining an image content with an image style using a Neural Algorithm for Artistic Style [9]. The main idea behind this algorithm is to generate new images that combine the semantic content of a target image coming from a Domain S with the appearance/style of another image coming from a Domain T. The goal is to preserve the semantic content, constraining the texture synthesis (the style of an image) by a suitable feature representation.

There are several conceptual differences between this technique, and the GAN previous one that we present, mostly regarding the way of the image is synthesized:

- First of all, we do not use GANs as an architectural based framework but instead, classical Convolutional Neural Networks (CNN), like a VGG network [33] for example, to extract the high-level semantic information.
- Secondly, this neural network framework need to be pre-trained on other tasks like classification tasks or segmentation one to extract a good feature representation of the training dataset that we choose [9].
- Finally, the framework works on a one-to-one basis that make us think to a paired dataset configuration like presented in section 2.2. By one-to-one, we want to mean that the framework is not able to

grasp the style of a whole historical document. Indeed, when it comes to applying the style to image content, we must specify a particular image that belongs to the Target Domain. It is only the style of this image that applies to the semantic content coming from the Source Domain. Conversely, a GAN framework allows theoretically to grasp the style of a whole document.

Now we have a better understanding of what deep learning techniques we are going to deal with in this paper, we will write done the outline of the rest of the thesis.

1.4 Oultline

In the next section of the thesis we will first speak about how usually the softwares tools and the researcher community deal with Handwritten Historical Documents. Then, we will explore the two deep learning techniques aforementioned using the recent literature at our disposal.
This preliminary work should allow us to perform our Image-to-Image translation task. In the third section we will define our experimental setups, looking at the particular datasets and models we need in order to achieve: a reconstruction task, a classification task our main generation task.
Next, we will focus on the training setup of the deep learning models and we will also build a quality measurement protocol to evaluate the obtained results.
Finally, we will evaluate and dicuss the results before concluding our work with some future improvement tracks.

2 Related Work

In this section we will explore the literature at our disposal, begining with an overview of the existing softwares that allow researchers to deal with Historical Handwritten Documents. Then we will pursue with an analysis on the existing GANs technique that allow to do Image-to-Image translation, finally we will do the same for Neural Style Transfer algorithms.

2.1 Historical Document Annotation and Generation

In the field of Document Image Analysis, researchers need to have quick and accessible datasets with ground truth annotations. Unfortunately, as we said in the introductory part, these types of datasets are a scarce resource. For several years now, the community has to build different kind of software that either allows to annotate document pages or generate synthetic data to overcome this problem.

2.1.1 Annotation Softwares

A family of software that we can call Annotation Softwares as in [17], are especially build to assist researchers in the process of manually (semi-manually) annotate historical documents. The goal is to create documents associated with ground truth labels [6, 34, 36]. Why? Because ground truths, as labels of document images, are essential to the development of DIA tools and their evaluations, above all in the age of mass digitization of printed documents. Despite, the attempts to produce more and more accessible, collaborative, web-based [36, 34] and automated platforms with copyright-free datasets, the process of manual annotation using such software remains tedious and asks to have some expertise in the field of document analysis to properly handle such tools. To give an idea of what things this type of softwares can perform we will present one of such system.

Aletheia As an example of an Annotation Software, Aletheia is a ground-truthing system for large amounts of historical documents [6]. First, we will explain how the structure of a document page is represented in such type of softwares. Then, we will pursue with several strategies that allow to target and isolate the page components. Thirdly, we will highlight two approaches that allow annotating these components. Finally, we will see why the study of such tools is pertinent to our work.

A page structure is globally defined by is components that we call page elements, in descending order we find: regions - for the sake of clarity we will work with two main region types in the next paraph: graphical regions and textual regions.
Keep in mind that Aletheia supports eleven region types (text, image, graphic, line drawing, chart, separator, table, maths, noise, frame and unknown) where some regions also possess sub-types that denote logical functions, like a paragraph or heading for text regions [6]. That is not all, Aletheia supports the creation of additional detailed information concerning properties of a region and take into account the logical relations between region, such system allow to improve the performance analysis given to the algorithm some information related to the reading order of a document.

A textual region, for example, can be subdivided in lowest components text lines, words, and glyphs. Regions are the highest level of a page element and the glyphs the lowest one. When we subdivide a region into lowest level elements, we say that they are components of the regions. For example, a glyph region is a component of a word region like in Figure 4 d). Aletheia also endorses the possibility to combine in parallel a textual region with a text ground truth input. We can view this input as metadata of the textual region. Unicode or ASCII encode the textual input. Furthermore, there is the possibility to combine each

level of granularity of a textual region [6].

Now we will see that the regions mentioned above can be defined manually point by point or in a semi-automated way, using these strategies:

- The component selection tool allows the user to define which components should be a part of a region.
- The bounding box strategy consists in defining a box region of interest, manually. Then, an algorithm with defined parameters shrinks the box until the meeting of a component. Usually, we use the bounding box strategy for text regions. We show an example of this strategy in the following Figure.

Figure 3: **Bounding Box Strategy.** On the top, this is a marked region with defined parameters and on the bottom the resulting region, coming from [6].

- The smearing strategy consists in defining a region of interest manually. Then an algorithm merges all the components that lie in this area in one single polygonal region. Usually, the users employ a smearing strategy for graphics regions.

The last thing that we have to see is the two approaches that the user have to annotate the document pages:

- The use of a **Top-Down approach** consists in defining the page elements in a tree-based structure starting from the highest level components to the lowest ones.
- The use of a **Bottom-Up approch** is just the opposite way, instead of subdividing a big region into several lowest components you start from a lower level and aggregate the components of the same level to produce a component of a higher level, like in the next Figure.

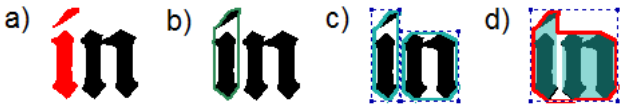


Figure 4: **Bottom-Up Strategy** - Glyphs and Words creation: a) Component selection; b) Glyph Creation; c) Glyphs Selection; d) Word Creation comming from [6].

However, such Annotation Software does not allow to perform what we try to do: *generate synthetic historical documents*. Indeed, it is true; however such software has developed together with some image synthesis tools that allow performing degradation tasks for example. This state of fact is crucial for

understanding that, Annotation Softwares and what we can call Document Images Synthesis Softwares share the same architectural structure and that they influence each other in the way of defining what is an image of a document page in terms of structure and features. In this work, we try to propose a different architectural, approach using a deep learning algorithm as Document Images Synthesis Software where the goal is to learn what defines an image of a document page without explicitly stating what the features are!

2.1.2 Images Synthesis Software

Since 1990 until nowadays, the DIA community of researchers have shown significant efforts to develop another type of software that allows generating synthetic datasets [17]; we call them Document Images Software. Such type of software allows to extend existing document images databases or to create new ones, to produce either ground-truthed controlled documents or new labeled data via some data augmentation procedures [3, 17, 26]. Indeed the use of these software has shown substantial effects on prediction performances[17]. The general procedure consists of defining a document structure via XML or LaTeX-files, combined with binarization, OCR and layout analysis algorithms. Such type of systems allow producing the synthetic document with ground truth information.

For document recognition tasks, several techniques of document image degradation used synthetic documents to improve the performance of their OCR algorithms. In the area of historical handwritten documents, some word-spotting techniques also take the advantages of using synthetic images of words to increase the system performance [25, 20]. We will now present two systems that in many points try to achieve the same objective that we are trying to perform:

DocEmul DocEmul [3] is an open source software that is used to generate structured handwritten synthetic documents which look likes real documents mainly for record counting tasks. By structured the authors mean to documents that are composed by a record-like or table-like structure (see Figure 5).

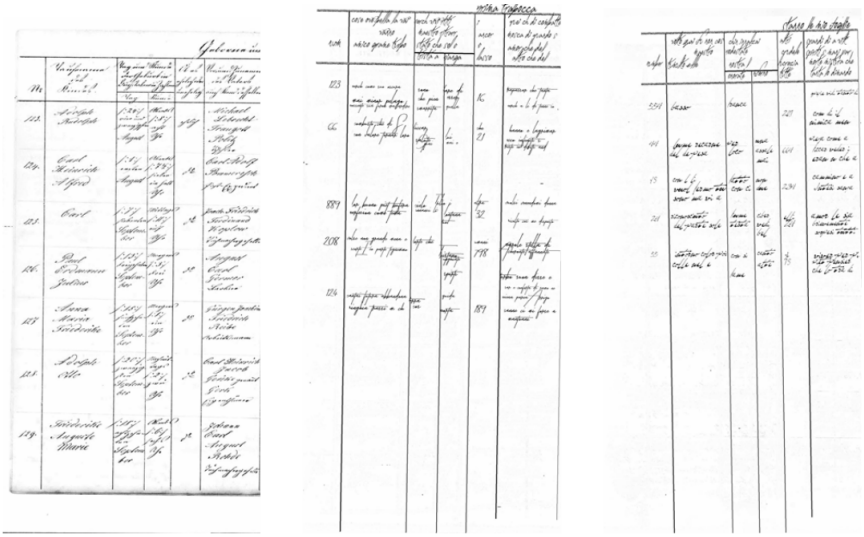


Figure 5: **DocEmul Real V.S. Synthetic Record Samples.** The first image comes from the real dataset and the two other are generated by the framework. This figure is taken from [3].

As the other frameworks mentioned above, this toolkit addresses document analysis tasks. The software is composed of several modules that perform different kind of separate tasks (see Figure 6)

- A background extractor that use binarization algorithms to localize background pixels in the document image.
- A module to structure the handwritten pages. This module allows defining a general page structure via an XML file which includes the characteristics of the document collection such as the dictionaries, the fonts, some graphics objects, the header, and the record structure. Despite some rigidity, the model allows operating some variability over these features during the generation phase.
- A module that performs some data augmentation like adding salt and pepper noise, performing some rotation and scale modification.

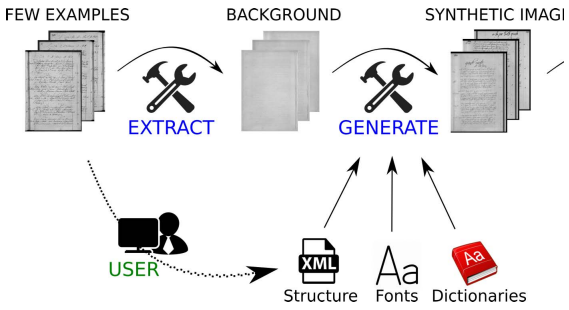


Figure 6: **DocEmul pipeline** [3].

The evaluation of the software performance shows that, by using a pre-trained CNN Deep Neural Network pre-trained model on synthetic data to fine tune the prediction during a "number of record" counting task, the results came better. Such type of performance results could be also a first goal and a proof of concept for the framework we are trying to build!

DocCreator The most similar framework to our, in a goal point of view, is DocCreator software [17]. DocCreator is a multiplatform and open source software where users can interact with the framework in an automatic or semi-automatic mode. The strength of this software is that it is the actually the only software that generates synthetic documents with ground truth that looks like real ones - which is our primary goal. Furthermore, it allows applying several degradation modes (Ink degradation, phantom character, paper holes, bleed-through, adaptive blur, 3D paper deformation, and nonlinear illumination model) to document images (see Figure 7).

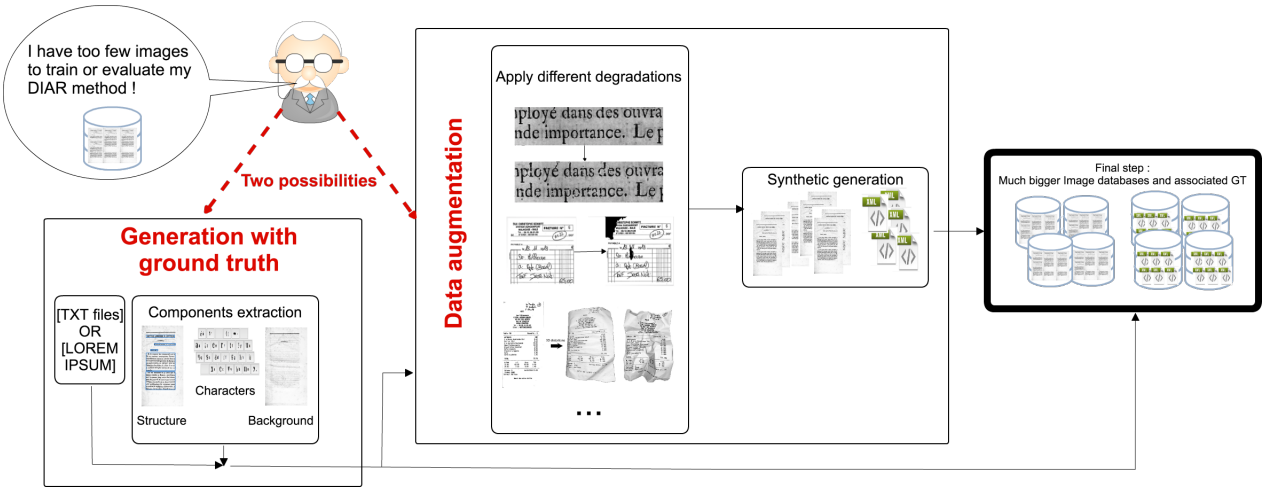


Figure 7: **DocCreator Pipeline** The pipeline shows two different ways to generate synthetic data, one way is direct the other one allow degradation procedures, figure founded in [17].

The model has shown his positive impact on performance prediction for binarization algorithm and OCR recognition, but also re-training tasks [17]. How did the researchers proceed to implement this part of the software? First of all, they use real document images coming from a true data distribution and extract from these documents three major components that will play the role of ground truth:

1. The font, extracted using Optical Character Recognition system (OCR) with a baseline management method. This system associates a Unicode value to each character of the document. Then each label is associated with a small database of images that correspond to the affiliated character. Thus the system makes it possible to create characters semi-automatically during the generation of the synthetic document, all in using some randomness.

2. The background, automatically extracted using an OpenCV algorithm that allows to in-paint the document to remove the characters.

3. The layout of the document, semi-automatically extracted using a layout segmentation method.

After the extraction procedure, the three main elements are assembled with the desired textual input to generate the synthetic image. The strength of this model is the possibility to recombine at will fonts, backgrounds, and layouts and that the generated synthetic documents look like real ones, even if some local deformation remains in the synthetic document (see Figure 8).

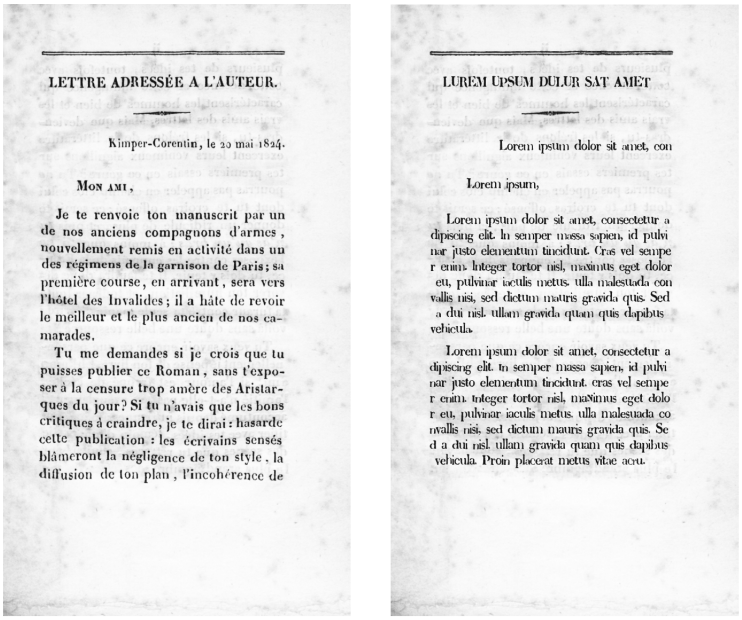


Figure 8: **DocCreator Typewritten Synthetic Document Image Generation Sample.** At the left you can see the original document image and at the right the synthetic one generated in an autonomous way with "Lorem ipsum" text, figure founded in [17].

Regarding the complexity of Historical Handwritten Documents, we are facing a more complex task when we attempt to generate synthetic documents. Behind what we can define as the structural content by the font's design, the layout of the pages, the paragraphs organization, the number of illuminations per page. It remains what we can call a style content, think to the texture information, the calligraphy of authors or the colorization for example. Such model in order to be compilant should implement such features in his architecture. But this will lead to more and more parallal algorithm creation to grasp such feature types, trying to define manually what are the components of an image. In the age of Deep Learning, maybe the time has come to let the machines defines what are the essential characteristics of a document page...

It is important, as noticed, to keep in mind that no one of these softwares, at the best of our knowledge, use Deep Neural Networks Architecture to generate these synthetic datasets, which is our goal.

2.2 Image-to-Image Translation with GANs

As we explain in section 1.1.2 an Image-to-Image translation problem is defined by an image synthesis task which consists in mapping an input image to an output image. In this subsection, we will make an overview of the most promising GANs frameworks that tackle the Image-to-Image translation problem we attempt to solve.

In the GAN perspective, the input image coupled with the ground-truth label is viewed as a constraint or a condition for the Generative Adversarial Network, as presented in the cGAN paragraph, section 1.2.3.

2.2.1 Supervised Image-to-Image Translation with Pix2Pix

Pix2Pix is a cGAN framework proposed by *Isola et al.* in [15]. We can define it as a general purpose framework that allows solving Image-to-Image translation problems using paired images. By paired images, we mean that each image in the Source Domain has a semantic equivalent image in the Target Domain, but represented with another aspect. The target images are used as ground-truth. Then, a paired image dataset consists of a collection of two types of images related in an unspoken way. The task of the algorithm is to discover this relationship. Because the images are paired such type of algorithms falls in the family of the supervised learning process.

To give a concrete example imagine a paired image couple, a sketch image composed by edges of a shoe and his corresponding photo image in a colorized and fashion way. The cGAN framework will take as input the edge image and will try to generate its correspondent as output. Then, the discriminator will then label as real or fake the couple composed by the edge image input and the synthesized photo, as shown in Figure 9. Keep in mind that the training of the discriminator is done in parallel so that the real couple of the edge input image and the ground-truthed photo also feed the discriminator.

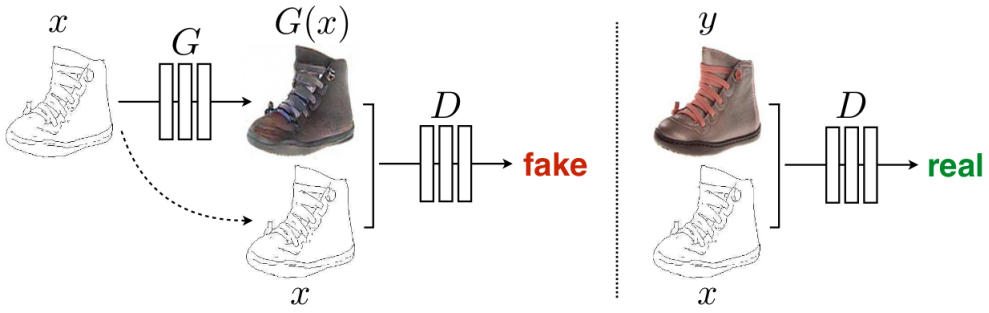


Figure 9: **Pix2Pix Image-to-Image translation.** Pix2Pix cGAN implementation framework, found in [15].

We previously said that Pix2Pix is a general purpose framework, we mean that the same GAN framework can handle different type of image processing tasks like, labels to street scene translation, labels to facade translation, black and white image to colored image translation, aerial image to map image translation, day to night translation or edges to photos translation as shown in the following Figure.

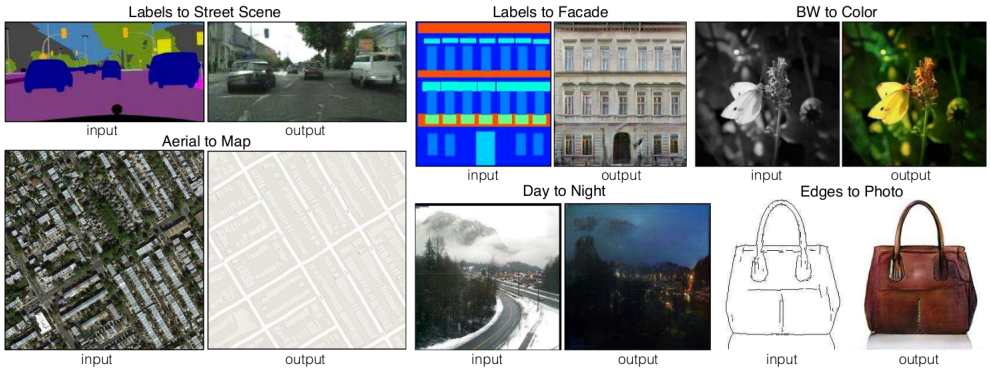
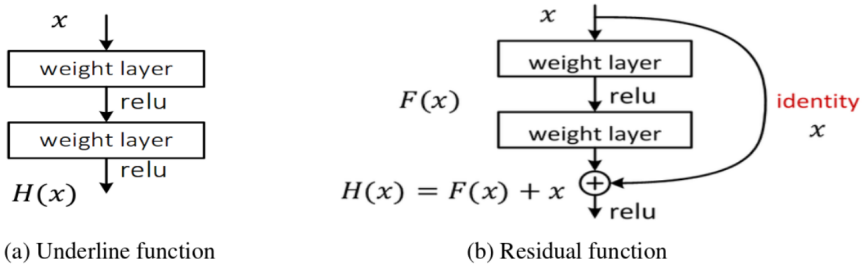


Figure 10: **Pix2pix Image-to-Image Translation Results.** Results shown over six different tasks, found in [15].

These different kinds of translation problems, thanks to the job of a conditional generative adversarial network, can be handled by one single algorithm rather than several ones. Before the advent of this technology, specifics algorithms specially designed for each of the above-mentioned tasks, regarding their particularities. The adversarial loss, conditioned by the image perform an agnostic pixel to pixel mapping. The only application -dependent parameter is the dataset used to train the neural network [15]. This framework was the first that used paired image in a supervised way to demonstrate that a single cGAN framework can produce good results on a wide variety of image-to-image translation tasks that involve highly structured graphical outputs.

Framework Architecture: U-Net and PatchGAN To achieve such results, the researchers choose a *U-Net* [31] based architecture for the generator of the cGAN framework and a convolutional *PatchGAN* [21] architecture for the discriminator. Both part of the adversarial network use building block of the Convolution-BatchNorm-ReLU form following the training recommendation of [30, p. 3].

U-Net The authors choose to take a U-Net based architecture. After the normal contracting part of a classical CNN using a series of max pooling, this architecture adds successive layers. These successive layers possess more feature channels than in the contracting layer part. To do so they combined the convolutions with upsampling operators instead of pooling ones [31]. This symmetric architecture in *u-shape* allows increasing the resolution of the output. The goal is to localize high-resolution features to propagate the contextual information more efficiently [31]. The authors of Pix2Pix slightly modify the U-Net architecture with skip connections between each layer i and $i - n$ layer where n is the total number of layers.
A skip connection is a deep learning technique that allows overcoming the vanishing gradient problem in Deep Neural Networks (DNN), preserving the information flow. Researchers use skip-connections as a residual learning technique. A residual learning technique [13] represented in Figure 11) consists in replacing $H(x)$ an underline mapping function by a residual mapping function $H(x) = F(x) + x$. The residual mapping function is performed by doing an identity mapping $I(x) = x$ using a shortcut element-wise addition that connects the input to the output of a building block. In a U-Net the building block input is defined by i and the building block output by $i - n$. Because of the U-Net architecture, these skip connections allow the information to persist across the symmetric part of the network.



(a) Underline function (b) Residual function

Figure 11: **Learnink Block.** Modular building blocks for residual learning found in [13].

PatchGAN The *PatchGAN* choice of implementation is related to the ability of the framework distance loss function to grasp the structural details of images. To fully understand what each part of the network does, we will explore the objective function in details. The authors of Pix2Pix choose to mix the conditional adversarial loss function $\mathcal{L}_{cGAN}$ with a $\mathcal{L}_{L1}$ distance loss function. The conditional adversarial loss function ensures the mapping between the input and the output we want (the translation task), while the distance loss function ensures that the output of the generator, the synthetic image, remains close to the ground truth image label, which belongs to the Target Domain.

In the following conditional adversarial loss formulation: $\mathcal{L}_{cGAN}$, $z \sim p(z)$ represent the random noise that follow a given random distribution - usually a Gaussian one[1]. The random noise z is fed to the generator with a conditional input image s. The formulation $s, t \sim p_{data}(s, t)$ represent the joint probability distribution of the image s of the Source Domain S, and t the corresponding image coming from the Target Domain T feed to the discriminator D.

$$\begin{aligned}\mathcal{L}_{cGAN}(G, D) = &\mathbb{E}_{\boldsymbol{s},\boldsymbol{t}\sim p_{data}(\boldsymbol{s},\boldsymbol{t})}[\log D(\boldsymbol{s}, \boldsymbol{t})] \\ &+ \mathbb{E}_{\boldsymbol{s}\sim p_{data}(\boldsymbol{s}),\boldsymbol{z}\sim p_z(\boldsymbol{z})}[\log(1 - D(\boldsymbol{s}, G(\boldsymbol{s}, \boldsymbol{z})]\end{aligned} \tag{5}$$

The loss function $\mathcal{L}_{L1}$ is defined by the following equation:

$$\mathcal{L}_{L1}(G) = \mathbb{E}_{\boldsymbol{s},\boldsymbol{t}\sim p_{data}(\boldsymbol{s},\boldsymbol{t})}, \boldsymbol{z} \sim p_z(\boldsymbol{z})[||\boldsymbol{t} - G(\boldsymbol{s}, \boldsymbol{z})||_1]. \tag{6}$$

Unfortunately, $\mathcal{L}_{L1}$ only grasp the low frequencies of an image, this matter of fact leads the framework to generate blurry synthetic images, but less blurry than if they used a $\mathcal{L}_{L2}$ loss function instead [15, 38]. To enforce the high-frequency structure clearness, the Pix2Pix authors use a particular architecture for the discriminator called a *PatchGAN* [15, 21]. The *PatchGAN* principle is to provide a discriminator D that use local image patches that first, run faster and secondly do not penalize the entire images but only structural defects coming from the patches at a defined scale. Thus, the authors choose to restrict the discrimination between real and fake images to the computed average score of smaller $N \times N$ pixels patches [15]. Furthermore, this PatchGAN implementation plays the role of a kind of texture/style loss as explained in [21].

Finally, we are able to formulate the objective function with two loss terms balanced by the hyper-parameter lambda λ:

[1]To be more precise, regarding the *Pix2Pix* framework implementation, the authors choose to provide the noise using a dropout method instead of drawing samples from a random distribution, finding that there was no significant difference between the two methods regarding the output quality [15]

$$\mathcal{L}(G, D) = \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L1}(G). \tag{7}$$

Where we aim to solve:

$$G^*, D^* = \arg \min_G \max_D \mathcal{L}(G, D). \tag{8}$$

2.2.2 Unsupervised Image-to-Image Translation

As we just seen above supervised learning tasks need a supervised dataset, but as we said paired image datasets are a scarce resource, even more, if the goal is to perform Image-to-Image translation over more than one domain. Consequently since 2016, several frameworks [39, 19, 37, 22, 23, 5, 2, 1] try to achieve the task for unpaired image datasets. These models use a whole arsenal of different deep learning techniques going from architectural modifications to losses rearrangements. Following the GAN classification description presented in [14], we can divide the unsupervised Image-to-Image translation frameworks that possibly fit with our goal in four categories.

- The networks that use a cycle loss with at least, a bi-directional reconstruction implementation [39, 37, 19, 23, 7, 5].
- The networks that use a pair-wise distance constraint loss [1].
- The networks that use a task-specific domain adaptation auxiliary classifier like in [2].
- The networks that use Variational Auto-Encoder (VAEs) with weight sharing as generator like in [22].

One model is a particular one because he does not really take advantage of these aforementioned particularities, even if it is a GAN framework. Indeed, this SimGAN [32] model is simpler, but it will allow us to test our working hypothesis: in order to perform a domain adaptation task, for really different probability distributions, we need to implement at least one of the frameworks mentioned above. This is why we decide to first present the SimGAN implementation. Then, in order to provide a strong baseline method, and because these networks have shown tremendous results, in term of high quality of the output; we chose to focus our attention on the models based on a cycle loss that take the advantages of the bi-directional reconstruction implementation.

SimGAN model The SimGAN framework [32] is exciting for us because the goal of this architecture is to reduce the gap between synthetic images produced by a simulator (input images) and real ones coming from the *MPIIGaze* dataset, composed by gaze direction images like in the following Figure, using what they call Simulated + Unsupervised learning. The task consists in improving the realism of synthetic data produced by the simulator using unlabeled real data. The problem can be understood as an image-to-image translation problem where the goal is to produce refined gaze direction images that look like real ones starting from a synthetic simulated gaze direction input image.

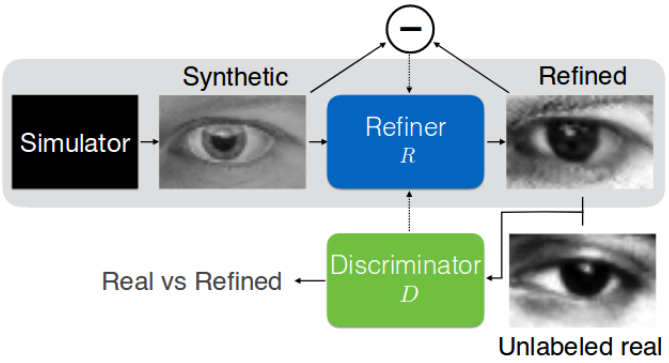


Figure 12: **SimGAN Framework.** The figure comes from [32].

They choose to use synthetic data to be able to preserve the annotations coming straight together with a given image. It is a precious property because synthetic data are generated via a controlled ground-truthed environment, avoiding an expensive and time-consuming labeling phase. Moreover, this state of affair is very similar to our implementation, as it would be explained in section 3.3.
Nonetheless, using synthetic data brings its batch of problems. The main concern is the divergence between the source domain probability distribution composed of synthetic images and the target domain probability distribution that we want to mimic, composed by the real images. One major issue with synthetic data is that a network trained with simulated images can learn to distinguish particular features only present in synthetic images. This type of learning tradeoffs can lead the network to overfitting or underfitting problems during the performance evaluation phase [12, 10].

Network Architecture This framework in his overall architecture is more close to a GAN [12] implementation and more straightforward than the previous mentioned GANs frameworks.

The authors make a lot of certain modifications compared to the original GAN implementation [12, 32]. These adjustments are relevant to conserve the annotated information coming from the simulator input, to bypass the structural artifacts and to increase the framework stability during the training. These architectural modifications consist of:

- An adversarial loss with a regularization term for the refiner network part, where the goal of the regularization term is to minimize the per-pixel difference between the feature space of the synthetic image and the feature space of the refined one with a $\mathcal{L}_{L1}$ distance norm.
- A local adversarial loss - local because the model use a PatchGAN implementation at discriminator level - for the discriminator and generator network part $\mathcal{L}_{GAN}$ where the goal is to classify several local images patches for a given pair of image samples. This technique allows diminishing the tendency of the refiner network to produce artifacts to fool the discriminator.
- An updating method applied to the discriminator part of the framework that grants to keep a history of the refined images. This technique improves the stability of the training and obliges the discriminator not to repeat the same mistakes over and over - this historic buffer is also use in the cycleGAN [39] implementation.

Results Beside a classical qualitative evaluation of the results, the authors build a user study that exhibits the high quality of the output refined images produced by the framework with what they call a *'Visual Turing Test'*. This quantitative test empower the performance evaluation of the produced output (a sample

of a refined and a real image is shown in Figure 12.
Each subject of the user study performance test has to classify 40 images: 20 images among a random selection of 50 refined images and 20 among a random selection of 50 real images. The average classification accuracy was 51.7%. This value being very close to 50%, the authors conclude that they were almost guessing during the test chance.

Cycle loss models with bi-directional reconstruction implementation These models propose to set a framework that allows discovering relations between a source domain and a target domain without the use of paired data. It was one of the first GANs solutions that propose such data setting: " two sets of images without any explicit pair labels" [19]. In the following Figure, you will see an example of the unpaired dataset used in the cycleGAN model implementation.

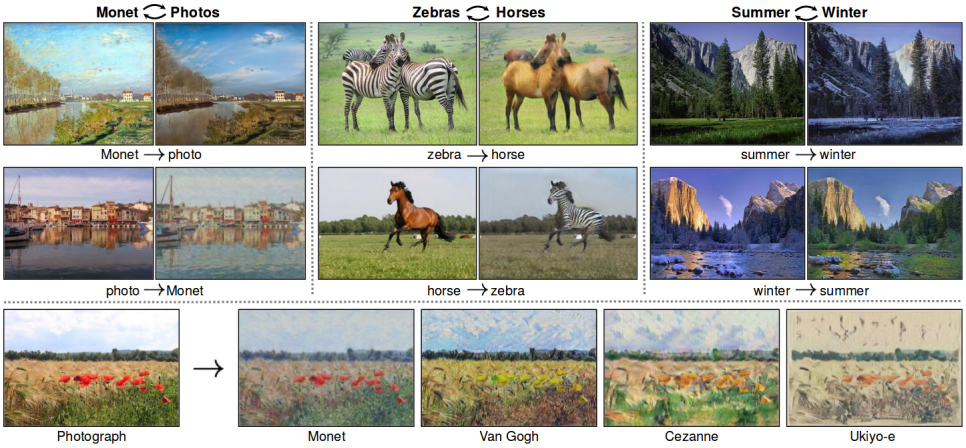


Figure 13: **cycleGAN Image-to-Image Translation Results.** These results were obtained over what we can understand as three different reconstruction tasks even if the same formulation allows to regroup these tasks in one single Image-to-Image translation problem. We retrieve a photo to painting mapping, a zebra to horse mapping, and a summer to winter landscape mapping, over six different datasets, three photos to painting datasets, a zebra to horse dataset a the summer to winter dataset. As you can see, the dataset is unpaired, indeed it is difficult to imagine equivalent photography of a Ukiyo-e painting. The Figure also expresses the bi-directional mapping function going from one domain to the other one and vice et versa. This image was found in [39].

The use of two strong constraints forces the domain mapping between a source domain and a target domain one that enforces the image-based representation across domains using the classical adversarial loss, $\mathcal{L}_{GAN}$. It is the same loss used in the Pix2Pix framework, that allows increasing the similarity between the synthetic images and the real ones. The other constraint enforces the reconstructed image to be a good representation of the input one, using also a similar loss function than in Pix2Pix, an L1 distance loss function $\mathcal{L}_{L1}$. In order to fully understand the two added components of the cycle loss model, in term of the objective function, we need to explain the architectural specificities of such type of networks.

Networks Architecture First of all, as we said, the network is a GAN based solution that takes as input an image from the source domain S and tries to generate a synthetic image belonging to the target domain T. Suppose that starting from S composed by electronic document images we want to generate synthetic but historical handwritten realistic images that looks like the ones coming from T, like presented in Figure

14. However, because the goal is to find a cross-domain relation, another generator performs the inverse mapping, this is what we call the bi-directional mapping function. For the sake of clarity, you can see on Figure 14 what is happened visually in the framework for one direction. Then, the network functions can be express by the following relations:

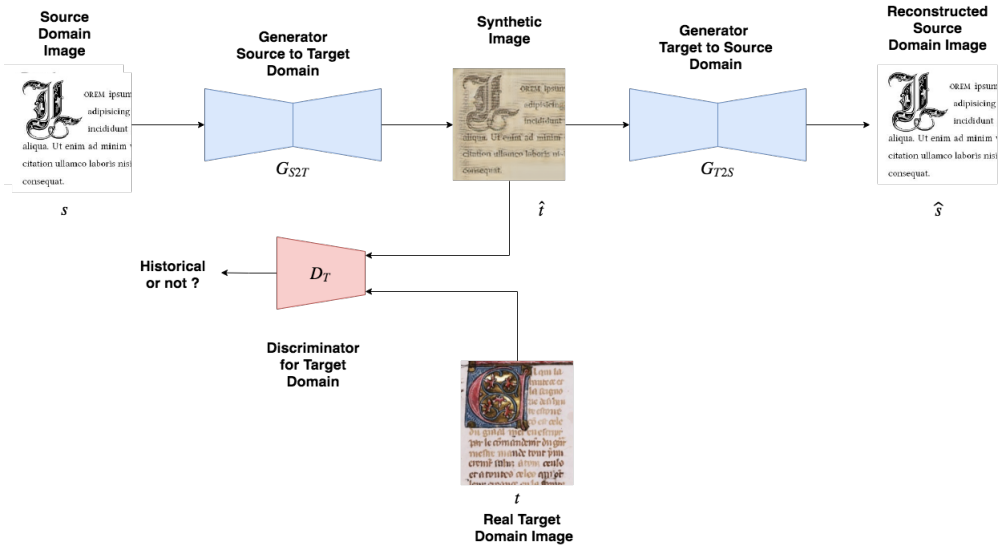


Figure 14: **cycleLoss One Side Framework Architecture.**

- A generator function G_{S2T} that perform a mapping between images from S to T and another generator function G_{T2S} that perform the inverse mapping between images from T to S in a one-to-one correspondence.
- If we express only one direction, the goal is that a real input image s coming from domain S should be translated in $\hat{t}$, a synthetic image that looks like an image coming from the domain T, by the generator G_{S2T}. The synthetic image $\hat{t}$ is then back-translated into an image that should look like the original image s coming from S. This new synthetic image $\hat{s}$ is produced by the generator G_{T2S}.
- To constrain the bidirectional mapping the authors used two different losses.
- A reconstruction loss, which is simply a L1 distance loss function [2] $\mathcal{L}_{L1}$ that enforce the original input to be as close as possible to the synthetic image produced after the successive generations. For the source domain we can express the loss by the following formluation:

$$\mathcal{L}_{L1_S} = ||s - G_{T2S}(G_{S2T}(s))||_1. \tag{9}$$

And for the target domain by:

$$\mathcal{L}_{L1_T} = ||t - G_{S2T}(G_{T2S}(t))||_1. \tag{10}$$

[2] Some models like DiscoGAN [19] used a L2 distance loss but the expirments has shown that this will lead the generator to produce blurry images.

- The cycle loss function $\mathcal{L}_{cyc}$ is just a way to express the fact that this to reconstruction loss function works, in parallal. We can now express the cycle loss by:

$$\begin{aligned}\mathcal{L}_{cyc}(G_{S2T}, G_{T2S}) = \mathbb{E}_{\boldsymbol{t}\sim p_{data_T}(\boldsymbol{t})}[||\boldsymbol{t} - G_{S2T}(G_{T2S}(\boldsymbol{t}))||_1.] \\ + \mathbb{E}_{\boldsymbol{s}\sim p_{data_S}(\boldsymbol{s})}[||\boldsymbol{s} - G_{T2S}(G_{S2T}(\boldsymbol{s}))||_1.]\end{aligned} \tag{11}$$

- A standard adversarial loss $\mathcal{L}_{GAN}$, like presented in section 1.2.3, applied by both discriminators D_S and D_T and both generators that ensure the proximity between synthetic image and a real ones for both domains. We can express the GAN loss for the G_{S2T} and D_T pair by:

$$\begin{aligned}\mathcal{L}_{GAN}(G_{S2T}, D_T) = \mathbb{E}_{\boldsymbol{t}\sim p_{data_T}(\boldsymbol{t})}[\log D_T(\boldsymbol{t})] \\ + \mathbb{E}_{\boldsymbol{s}\sim p_{data_S}(\boldsymbol{s})}[1 - \log(D_T(G_{S2T}(\boldsymbol{s})]\end{aligned} \tag{12}$$

  The same GAN loss occurr for the reverse pair G_{T2S} and D_S.

- Regarding the discriminator each synthetic image type: $\hat{s}$ and $\hat{t}$ are sent to a separate discriminator that calculate to which domain the image should belong.

- We can now build the overall loss function where we try to minimize the generators and maximize the discriminator:

$$\begin{aligned}\mathcal{L}(G_{S2T}, G_{T2S}, D_S, D_T) = \mathcal{L}_{GAN}(G_{S2T}, D_T) \\ + \mathcal{L}_{GAN}(G_{T2S}, D_S) \\ + \lambda\mathcal{L}_{cyc}(G_{S2T}, G_{T2S})\end{aligned} \tag{13}$$

  In the overall loss function, the cycle loss is parametrized by a factor λ.

- The GANs composition itself does not necessarily implement special characteristics like PatchGAN or U-Net like architecture. The classical form for GANs framework is Convolution/Deconvolution-BatchNorm-LeakyReLU (instead of ReLU) with a weight decay regularization, following GANs training recommendations [10]. The depth (total number of the layer) of the networks in both discriminator and generator part varies in function of the task we try to achieve. The authors generally chose to set the image size input and output to 256×256 pixels.

2.3 Neural Style Transfer

We will close these related work section over the Image-to-Image translation task by going deeper in the implementation details of the Neural Style Transfer Algorithm [9]. As we previously seen performing an Image-to-Image translation task is not easy work. Nevertheless, some researchers in 2016, have developed a method that allows rendering the semantic content of an image in different artistic styles, as shown in the following Figure:

Figure 15: **NST Artistic Style Transfer Samples.** The figure comes from from [9].

But how the style and the content are represented when a Convolutional Neural Network like VGG-19 [33, 9]?

2.3.1 Content Representation

The content information can be represented by a feature representation $\mathcal{F}$, at layer level l. Then for each layer in the CNN, we can represent them by a feature representation $\mathcal{F}^l$. The feature representation is composed of a matrix that expresses the number of feature maps of a given layer by the width and height size of each feature map. Then $\mathcal{F}^l_{i,j}$ express the representation of the i^{th} feature map at position j in the layer l [9]. Then if $\hat{s}$ is the generated image and s the content image, that in our case belongs to the source domain S that we give as input to our network. Then G^l and F^l are the feature representation for a given layer. The content loss can be then express by the squared error at layer level, by the following equation:

$$\mathcal{L}_{content}(\hat{s}, s, l) = \frac{1}{2}\sum_{i,j}(F^l_{i,j} - G^l_{i,j})^2 \tag{14}$$

The equation above explained why we need to pre-train the VGG model over another task like classification or object recognition before running the NST algorithm. Indeed, the goal is to learn a content representation.

2.3.2 Style Representation

The style information of an input image t, wich in our case belongs to the target domain, can be grasp using a special feature space called a Gram matrix. Where this matrix is build on the top of the feature map response for each layer l of the network. The Gram matrix simply calculate the inner product of the vectorised features map i and j for a given layer l [9], like shown in the folowing equation:

$$G^l_{i,j} = \sum_k F^l_{i,k} F^l_{j,k} \tag{15}$$

The Gram matrix allow to grasp the style while leaving aside the content representation. Then, if $\hat{t}$ is the generated image and t the input image, E_l the contribution of each layer to the mean square loss between the Gram matrix of $\hat{t}$ and t, weigthed by a given factor w_l. We can express the style loss function by the following equation:

$$\mathcal{L}_{style}(\hat{t}, t) = \sum_{l=0}^{L} w_l E_l. \tag{16}$$

To conclude, the total loss function that mixed the loss content and the style loss, over the generated image $\hat{g}$ can be expressed by:

$$\mathcal{L}_{total}(\hat{g}, s, t) = \alpha \mathcal{L}_{content}(\hat{g}, s) + \beta \mathcal{L}_{style}(\hat{g}, t) \tag{17}$$

The following figure shows what the equation make on the input images at layer level, regarding the to losses:

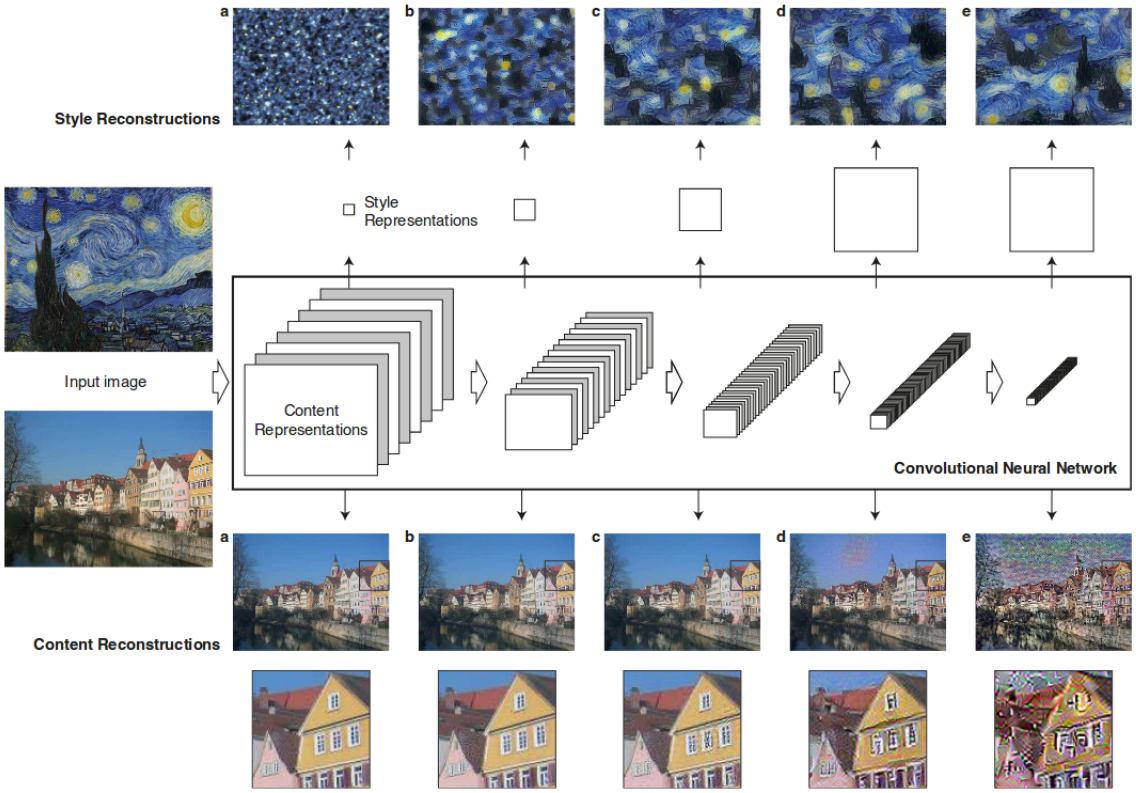


Figure 16: **Style and Content Losses Reconstructions.** The figure show the style and losses reconstructins layer by layer, the figure comes from [9].

Several researchers try to improve the original implementation presented here [9], for specific purposes, modifying the way to set the objective functions like in [16, 8, 24]. Despite the fact that some of these implementations like [24] should lead us to better results, because for example [8] improve the way of the color are preserved, we chose to use the original method in order to give to our framework a baseline landmark. After all, our goal is to find a good algorithm for our task and not really to fine tune an existing algorithm.

3 Experimental Setup

3.1 Datasets

In this section, we will present the Datasets that we choose to use. First, we speak about the original HBA Dataset using during the ICDAR 2017 competition were the historical documents that we use in this paper come from. Then, we will speak about the modification that we made over this dataset and why we choose to use particular books for particular training tasks. Thirdly, we introduce the LaTeX- framework that we use to produce our Source Domain for the generating task.

3.1.1 Original HBA Dataset

The images composing the HBA 1.0 dataset come from the Gallica digital library[3]. Originally this dataset is composed of eleven books; there are five manuscripts and six printed one. All these books are in different languages and typographies, and they were published between the twelfth and nineteenth centuries. The dataset contains composed by 4436 pages: 2,435 handwritten pages and 2,001 printed ones with 7 580'884'351 annotated ground-truthed pixels. The image format follows the Tagged Image File Format (TIFF), and the annotations are set in TXT. The annotation process of the HBA 1.0 dataset has been manually set. Only the ground truth of the learning images of nine books of all the data coming from the evaluation database is provided. The evaluation data-set of the sample database is available at the following link [4].

Originally the dataset was produced during the first edition of the International Historical Review (HBA2017) conjointly to the 14th International Conference on Document Analysis (ICDAR2017) and the 4th International Congress on Image Processing of Ancient Documents (HIP2017) in Kyoto, Japan. With the data set an evaluation protocol is provided to address specifics issues related to ancient document image analysis techniques, at pixel-level. During this competition, the goal was to evaluate and automatically deduce information coming from low-level image analysis method with a limited set of learning algorithms. Using the HBA 1.0 dataset, two tasks were evaluated as part of the HBA2017 competition: one evaluates how participating methods can discriminate textual content from graphical content at the pixel level; the other evaluates the ability of these methods to separate textual content according to fonts (e.g., lowercase, uppercase, and italic) at the pixel level.

3.1.2 Modified HBA Dataset

The original Dataset is complete. However, for our purpose, we only need six books coming from the HBA dataset, four handwritten documents, and two typewritten ones. Among these books, we made some changes.

First, we choose to remove blank pages with no text on it and the bindings document image pages; these deletions explained the differences in term of number of pages between the modified dataset and the original dataset.
Secondly, we choose not to use three grayscale documents because our goal is to handle features like, mixed ink colors that come from the illuminations or the paper texture. We also choose to not use an Italian typewritten document from 1596 and a French manuscript document from 1758 because the style of theses documents pages does not enough look like our defined domains, the source one and the target one.

[3] **Web link to the Gallica digital library:** http://api.bnf.fr/hba-un-jeu-dimages-annotees-pour-lanalyse-de-la-structure-de-mise-en-page-douvrages-anciens

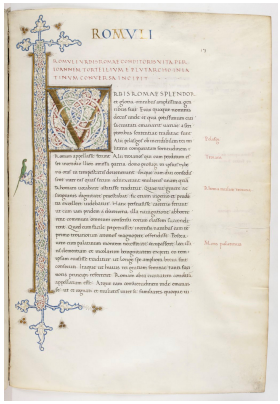
[4] **Web link to the evaluation dataset:** http://icdar2017hba.litislab.eu/index.php/evaluation/dataset-division/sample-dataset/

The following Table 3.1.2 describe the characteristics of the chosen one mentioned with the modifications documents. The following link[3] refers to the original dataset used during the ICIDAR 2017 competition.

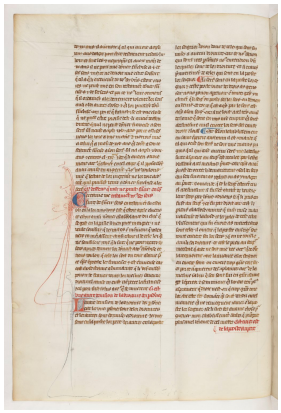
Book ID/Title	*Publication Date*	*Number of Pages*	*Book Type*	*Image Type*	*Resolution*	*Language*
Book 1 *"Plutarchus, Vitae illustrium virorum"*	1743-1774	709	Handwritten	RGB	6158 X 4267	Latin
Book 2 *"Justinien, Institutes"*	1342	464	Handwritten	RGB	3075 X 2048	French
Book 3: *"Girart d'Amiens, Meliacin ou le Cheval de fust"*	1285	331	Handwritten	RGB	2902 X 2048	French
Book 4: *"Cy commencent le Procès de Belial à l'encontre de Jhésus"*	1481	325	Typwritten with manual annotations	RGB	2860 X 2048	French
Book 5 *"Voyage pittoresque de la Grèce"*	1782-1822	324	Typewritten	RGB	3068 X 2048	French
Book 6 *"La Chartreuse de Parme"*	1839	400	Typewritten	RGB	3484 X 2048	French

Table 1: **Characteristics of the HBA Documents.** You will find the book ID and title, the publication date, the number of pages releted to the documen that we used in this work, the book type wich refers to an handwritten or a typwritten document, the image type wich refers to the color chanel showed by the document, the resolution used for the raw .jpg document imgaes and the language of the document.

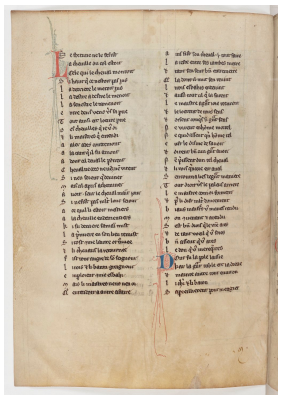
In the following Figure 17, there is a typical sample set coming from the chosen documents mentioned above. The three first books are handwritten ones where rich illuminations, manual annotations, and colored initials are present. The book four is a particular one because he looks like a handwritten one, but looking carefully we can see that he is typewritten. We choose this document because he is very similar to the electronic documents that we produce with our LaTeX- framework, see Figure 18, without the controlled ground-truth. The two others look like classical novel books.



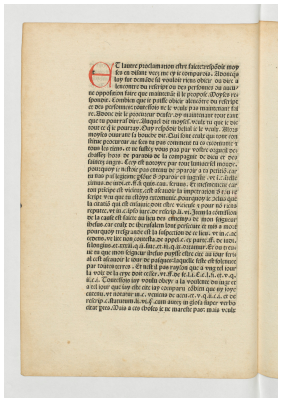
(a) Book 1
"Plutarchus, Vitae illustrium virorum"



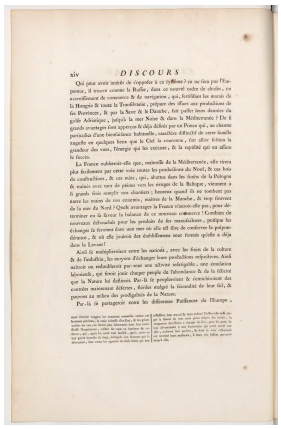
(b) Book 2
"Justinien, Institutes"



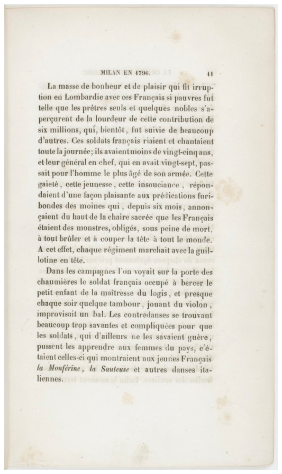
(c) Book 3:
"Girart d'Amiens, Meliacin ou le Cheval de fust"



(d) Book 4:
"Cy commencent le Procès de Belial à l'encontre de Jhésus"



(e) Book 5
"Voyage pittoresque de la Grèce"



(f) Book 6
"La Chartreuse de Parme"

Figure 17: **Raw Sample Pages of Each Book Used in This Work.**

This wide variety of books typology 3.1.2 is crucial for us because as showed in the next table our image-to-image translation goal is highly dependent of two properties: the training task we want to achieve and the probability distribution of the domain we want to grasp.

Dataset	*Book ID/Title*	*Task*
Source Domain Dataset (SDD)	***Ebook 1*** *Electronic Generated Document* ***Book 5*** *"Voyage pittoresque de la Grèce"* ***Book 6*** *"La Chartreuse de Parme"*	Reconstruction and Classification
Target Domain Dataset (TDD)	***Book 2*** *"Justinien, Institutes"* ***Book 3*** *"Girart d'Amiens, Meliacin ou le Cheval de fust"* ***Book 4*** *"Cy commencent le Procès de Belial à l'encontre de Jhésus"*	Reconstruction and Classification
Evaluation Dataset (ED)	***Book 1*** *"Plutarchus, Vitae illustrium virorum"* ***Ebook 2*** *Electronic Generated Document*	Generation

Table 2: **HBA Documents Taxonomy.** The taxonomy is made regarding the task usage and the Domain dataset affiliation.

As presented in the previous section 3.3 our main goal is to translate a document image that belongs to our source domain to the target domain. By keeping this state of matter in mind, we can define three different training sets:

- The Target Document Dataset (TDD) composed by Books 2, 3 and 4 coming from the modified HBA Dataset 3.1.2. This SDD is used for classification and reconstruction tasks. In a Domain point of view, this dataset will form the Target Domain for these given tasks. Note that book 4, which is not handwritten but belong to this dataset because the pages look like a historical one, see Figure 17.
- The Source Document Dataset (SDD) composed by book 5 and 6 coming from the modified HBA Dataset 3.1.2 and the French Electronic Document from our LaTeX-framework. This TDD is used for classification and reconstruction tasks.
- The Evaluation Document Dataset (ED) composed by the book 1 for the target domain part and by the Latin Electronic Document from our LaTeX-framework for the source domain part. This ED is used for the final generation task.

3.1.3 LaTeX-framework: Electronic Document Dataset

The goal of this framework is to help to produce the controlled source domain. The idea is to define several LaTeXspecification documents to obtain a list of file type taxonomy where we can handle and retrieve:

- The number of columns
- The number of initials

- The font type and size
- The automatically generated text

When the PDF document is produced we generate raw .jpg images using Python.

We do not spend too much time over this part of the framework for two reasons. First, the ground-truth label is useless if we are unable to mimic the style of a Historical Handwritten Images using a proper mapping function. Secondly, there already exist some framework that works with such type of system like explained in [17]. Consequently, we just implement the backbone of the framework and we produce two electronic books. One book is used in the ED dataset and the other one in the SDD dataset. In the following figure, you can see three samples of this electronic image document type.

Figure 18: **Source Domain Samples**.

3.2 Models

We chose to test three models, two GANs frameworks, the SimGAN model [32] and the cycleGAN model [39] for the aforementioned reasons. In order to test another Image-to-Image approach, we also chose to test the first and classical implementation of the Neural Style Transfer Algorithm [9] (NST). Roughly,

these three models follow the implementation recomandations given in the official papers and in the code details given by the autors. The architectural details, and training details are well documented in the code that I provide at this address [5] : if the sections 2 and 4 do not provide enough details.

3.3 Tasks

We can subdivide our tasks into three parts, the reconstruction task, the classification task and the generation task with the affiliate Datasets mentioned in previous datasets section, where:

- **The reconstruction task** that is used to pre-train the encode parts of the Generators of the cycleGAN model.
- **The classification task** that is used to pre-train the VGG-19 convolutional neural network of the Neural Style Transfer model and the discriminators part of the cycleGAN model.
- **The generation task** that is our main Image-to-Image translation task using unpaired data. This task is performed by the three models, the SimGAN model, the cycleGAN one and the Neural Style Transfer model.

As mentioned above our main goal is to perform the Image-to-Image translation task where we can model our ambition in a two-step schematic way like in the following Figure:

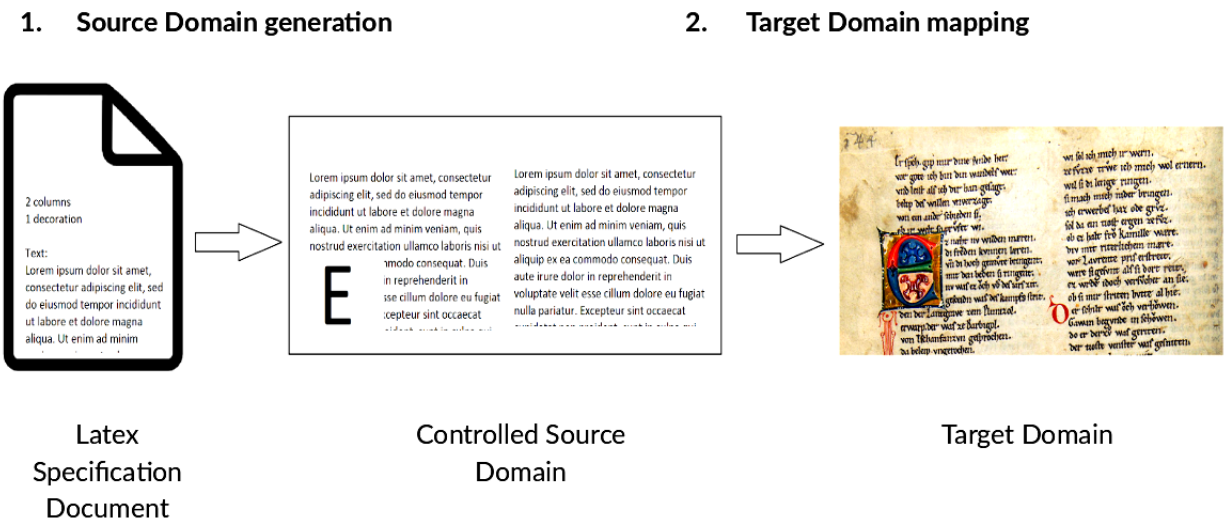


Figure 19: **A Two Steps Generation Task.** The first step consist of creating the controlled source domain samples using a latex specification document then we have to achieve the second step wich consist in learning the mapping function between the source domain and the target domain.

During the first step, we achieve the source domain generation using a latex specification document file that should produce a controlled source domain document sample. In the second step, we have to learn a mapping function between the source domain and the target domain, where the goal is to perform the transfer learning task that we call Image-to-Image translation task. For doing so, we chose to use the aforementioned models.

[5] https://github.com/aalbec/master_thesis

4 Training Setup

Just to recall we have three tasks to perform like mentioned in the previous section, for more details regarding the datasets affiliate to each tasks see Table 3.1.2.

We then have three models to train, the SimGAN, the cycleGAN and the NST model over two image setting modes: a raw image setting mode and a random-crop image setting mode.
For the seek of clarity, the image setting mode defines the type of input that will be given to the network. The raw setting mode represents the fact that the entire image of a document page has been resized to $[256 \times 256]$ pixels, like in Figure 22. The random-crop image setting mode represents the fact that over a resized document image of $[3086 \times 2132]$ and center cropped by $[1886 \times 1232]$ we randomly select an image crop of $[256 \times 256]$ pixels. This pixel setting is important because it allows having the almost the same number of lines and word in a cropped image for the Source Domain and the Target Domain of the Evaluation Dataset (ED), which is an essential component of our experimental setup like explained in the previous section.
The three aforementioned models are trained on the raw training setting mode for the reconstruction task as shown in the following Table.

Model	*Task*	*Image Settings*
SimGAN	Generation	Raw Random-Crop
cycleGAN	Generation	Raw Random-Crop
NST	Generation	Raw Random-Crop

Table 3: **Raw Training Mode.** This table show the raw training setting mode for the three translation models, the task that they performed and the image setting mode used.

Additionally to that, two models among them, the cycleGAN and the NST - more precisely their sub-models- are tested over another training settings mode: the pre-training mode, with the corresponding task mode, like shown in the following Table.

Model	*Sub-Model*	*Task*	*Image Settings*
cycleGAN	Generators	Reconstruction	Random-Crop
	Discriminators	Classification	Random-Crop
NST	VGG-19	Classification	Raw Random-crop

Table 4: **Pre-training Mode.** This table show the pre-training setting mode for the cycleGAN and NST translation models. The particualr pre-training tasks, that imply reconstrution and classification are performed by the sub-model parts of the respective main models given the mentioned image setting modes.

We will now see in details the training setup for each model, tasks, and image settings mode starting by the GANs frameworks and pursuing the NST model.

4.1 GANs Frameworks

4.1.1 SimGAN

This GAN model is the simple one in term of architecture, but also in term of loss functions. The purpose of this model is mainly to isolate the properties of the loss function in order to test some hypothesis. Indeed, we do not expect that this model outperforms the cycleGAN one. The goal is to show that without a cycle consistency loss the model is unable to perform an Image-to-Image translation task. Some experimental results presented in [32, 39] shown that the model performs well when the domains are similar in term of data distribution. Our goal is to show that it's probably not the case for our domains where the style and semantic content of the source and target domains are not so close.

Raw Training Details After a test period, using several settings we decide to train the SimGAN model over the training dataset of the Evaluation Dataset (ED), that is composed by 600 document images for both domains the source and the target ones, during fifty epochs. In such way that, during twenty epochs the refinery is only trained with the L1 distance loss function $\mathcal{L}_{L1}$ that is used as a self-regularization term and the classical GAN loss function the classical GAN loss function $\mathcal{L}_{GAN}$. Then during five epochs, we train the discriminator over randomly selected training samples coming from the refiner and from the target domain of the ED training part. Finally, we train both model parts one after the other during twenty-five epochs. We are obliged to try several ratios of training epochs between the refiner and the discriminator because of GANs well know training issues, indeed the settings are quite difficult to manage. Because, the goal is to find an equilibrium between the discriminator and the generator, all in avoiding mode collapse problems. Actually, we don't want that the discriminator become too strong for example which was the case for this implementation.
We keep the same settings for the random-crop image setting mode, we just adjust the PyTorch transform of the model like set in Section 4.
We use a batch size of four, mainly because of computational restrictions, a learning rate of 0.001 and a delta of 0.1 for the $\mathcal{L}_{L1}$ distance loss parameter. A For more training details you can see the code at this address[6].

4.1.2 cycleGAN

This GAN model is more complex than the SimGAN in term of architecture, but also in term of loss functions. The purpose of this model is to show that at least we need a more complex objective function to perform the Image-to-Image translation task.
In the cycleGAN implementation, we opted for a cycle consistency loss $\mathcal{L}_{L_{cyc}}$ with a regularization term of 10, in addition to the L1 distance loss function $\mathcal{L}_{L1}$ and the classical GAN loss function $\mathcal{L}_{GAN}$ that is slightly modified. Instead of using the classical one, we follow the recommendations of the cycleGAN authors replacing the negative log-likelihood objective by a least-square loss. This modification increases the training stability of the network and shows better results in term of synthetic output results [39]. The cycleGAN model also opts to an implementation that allows grasping the two directions of the mapping function regarding the domains; where the Image-to-Image translation task is performed from the Target Domain to the Source Domain and from the Source Domain to the Target Domain. The cycle loss with the bi-directional mapping ensures the transfer learning. We chose this GAN framework implementation instead of the other implementations presented in the related work section 2.2.2 because of the obtained results over highly complex image types like shown in the cycleGAN paper [39].

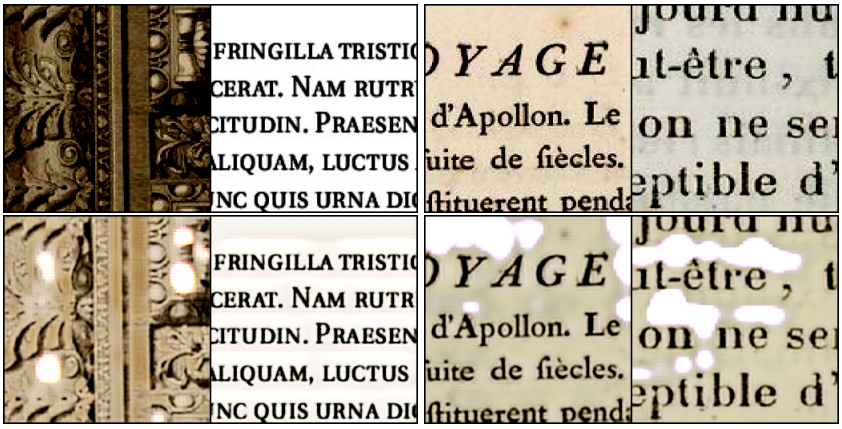
[6] `https://github.com/aalbec/master_thesis`

Raw Training Details After an early testing phase, using several different datasets we decide to train the cycleGAN model over the training dataset defined in the Evaluation Dataset (ED). We chose to restrict the Target Domain to only one book, instead of a set of several books, because during the early testing phase we discover that using a collection of different historical handwritten documents lead the model to uniform the representation of target domain books. This matter of fact lead us to several drawbacks, actually, we were unable to spot specifics features like annotations types or, font types, that represent the specificity of each particular historical document books.
In term of the training period, we chose the train the model during fifty epochs from scratch, for both directions. We follow the recommendation of using a history buffer that stores the 50 previously synthetic generated images by the generators in order to update the discriminator and reducing the model oscillation [39]. We use a batch size of one, a learning rate of 0.0002 and linearly decay the rate to zero after 25 epochs. We reduce the number of epochs mainly because the of the size of our evaluation dataset and to avoid overfitting risks. Like for the previous model we keep the same settings for the random-crop image setting mode. We just adjust the PyTorch transform of the model like set in Section 4.
For more training details you can see the code at this address [7].

Pre-training details In order to pre-train the cycleGAN model, we build to distinct tasks for the generator part and discriminator part of the cycleGAN model. We pre-train the encoder parts of the generators over a reconstruction task using a simple auto-encoder for both domains, the source one and the target one using respectively the Source Domain Dataset (SDD) and the Target Domain Dataset (TDD), but only for a random-crop image setting with the same previous PyTorch transform. We chose to train the encoder part of the generators during only 25 epochs because the generators in are huge in term of network size like showed in section 2.2.2
We also at first time decide to pre-train the two discriminators using the same datasets over a classification task. Nevertheless, after a quick testing phase, we decide to give-up the pre-training phase for the discriminators, because this part of the model became too strong compared to the generators. Indeed, the training phase became more and more unstable, and the equilibrium was not even approached, which lead the generators to output either white or black synthetic outputs.

In the following Figure, you will find the results obtained after the pre-training phase of the source domain generator G_{S2T}, that we train during 25 epochs over a batch size of 4 with a learning rate of 0.001. We can notice the presence of white holes in the reconstructed images. This is probably due to the fact that the among the set there is one book that is generated by our source domain latex generator module. This book is the only one with a white background, auto-encoder try to standardize the background color over the epochs this can explain why we retrieve these artifacts.

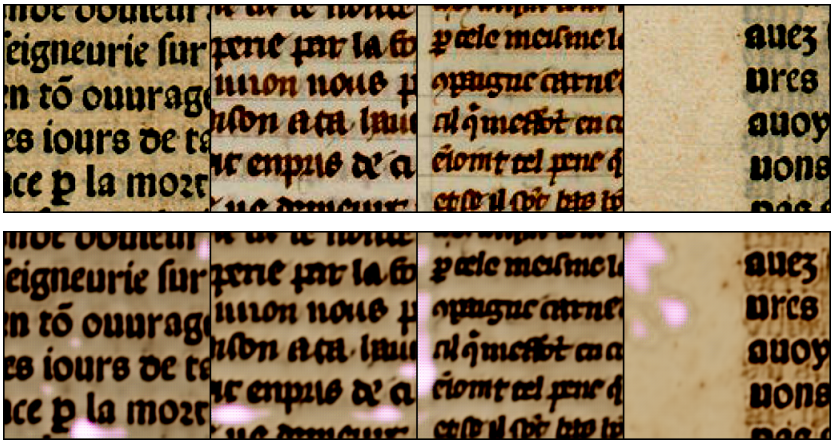
[7] `https://github.com/aalbec/master_thesis`



Source Domain Input Image V.S. Reconstructed Source Images at Epoch 25

Figure 20: **Source Domain Reconstruction Task Using a Simple Auto-Encoder.** The four top images represent the source domain input images and the bottom one the reconstructed source domain images after 25 epochs. The SDD dataset is composed of three books, the first and the third images come from the same book who integrates some illustrations.

In the following Figure, you will find the results obtained after the pre-training phase of the target domain generator G_{T2S}, that we train during 25 epochs over a batch size of 4 with a learning rate of 0.001. We can notice the presence of colored artifacts hole in the reconstructed images after 5 epochs, where the network is still learning how to well reconstruct an image. After 25 epochs the artifacts disappear. We can see in Figure (b) that the colors have vanished during the reconstruction process. This matter of fact explains why in the final results the network seems to be unable to colorize the synthetic sample. We also notice and confirm the background uniformization effect mentioned above, with an increase of the contrast.



(a) Target Input Images V.S. Reconstructed Target Images at Epoch 5

(b) Target Input Images V.S. Reconstructed Target Images at Epoch 25

Figure 21: **Target Domain Reconstruction Task Using a Simple Auto-Encoder.** The four top images represent the target domain input images and the bottom one the reconstructed target domain images after 5 epochs for point (a) and after 25 epochs for point (b). The TDD dataset is composed of three books, the second and the third images come from the same book.

In both domains, the reconstruction is quite perfect, we can infer that the model encodes the essential features of the presented books, indeed you will see in the results section that the pre-training phase has a good effect over the quality of the random-cropped synthetic samples.

4.2 Neural Style Transfer Algorithm

For the Neural Style Transfer Algorithm model we use a VGG-19 convolutional neural network where the loss function minimize the loss content $\mathcal{L}_{content}$ and the style loss $\mathcal{L}_{style}$ using some weighting factor for the style and content reconstruction that follow the L-BFGS numerical optimization strategy in order to keep a defined ratio between the style and the content importance. This model only supports a one-to-one mapping. The training details are dependent on the pre-training phase of the model; other architectural

details are the code details, see [8].

Raw Training Details Initially, the VGG-19 network is pre-trained with ImageNet for object recognition and localization tasks. When we run the algorithm the model uses the pre-trained VGG-19 network implemented with the PyTorch library, only the last layers of the network are re-initialized.

Pre-Training Details In order to improve the results of the VGG-19 network, we chose to train the convolutional neural network over our own TDD dataset composed by three book classes on a classification task, during 25 epochs for the two image setting modes, raw, and random-crop. The batch size is set to 4, the learning rate to 0.001 and the momentum to 0.9.

4.3 Quality Measurements

In the litterature [39, 37, 19, 5, 9], regarding unsupervised Image-to-Image translation tasks, there are two ways to evaluate the performance of the models, a qualitative one and a quantitative one.

4.3.1 Qualitative methods

The qualitative method is used in almost all scientific research and is mainly based on the subjective perceptual appreciation of the authors regarding a bunch of human criterions which are not formally defined, like for example, the blurriness of the output, the style preservation of the target domain. Even if this method does not offer objective and measurable guarantees concerning the quality of the output images, these approaches are appropriate for identifying the forces and defects of a given model, first of all at the early steps of a framework implementation, but also for further improvement purposes.

4.3.2 Quantitative methods

The quantitative methods are likewise used in almost all scientific research. We can divide them into two subcategories, the quantitative perceptual studies that are human-based and the quantitative metrics that are machine-based.

Quantitative perceptual studies Generally, quantitative perceptual studies proposed like in [39, 37, 5] are based on the use of the Amazon Mechanical Turk (AMT) tool. The AMT tool allows accessing to random people to ask them to answer online surveys in exchange for money. The baseline used protocol during these surveys was originally developed by *Isola et al.* [15].

A series of real and synthetic images pair is shown to the participants. The goal is to guess which one is the real one. The session is divided into two steps. The first step is a training stage. During a defined number of trials, feedback is given to the participant for him to know that he has labeled the image correctly. The second step consists of the remaining trials of the session. This step is used to evaluate the number of times the participant was wrong and thus allow to rate the algorithm performance.
This quantitative protocol as a qualitative one can also lead to some biases. First, the hard problem is to define a good ratio between the number of training trials and the real experimental ones with the goal to avoid that humans participants, because of the training stage, begin to identify the regularities of synthetic outputs to quickly. The second bias to avoid is the non-reproducibility of the experimental setup, regarding the number of participants, the chosen image-pairs, the expertise of the participant for examples.

[8] `https://github.com/aalbec/master_thesis`

Quantitative mechanical metrics The other quantitative metrics are non-human-dependant, but more dataset dependent. Indeed, how a mechanical metric can evaluate the graphical realism of an image? Basically, it depends on the dataset and on the previous work done by the scientific community, inasmuch as datasets used by the researchers - the Cityscape dataset used in [39] or the Label-Facades dataset used in [39, 37] - were previously designed for other tasks than Image-to-Image translation, like layout segmentation one, for example. Then it becomes possible to use some classical metrics used for such tasks, for segmentation tasks, for example, semantic segmentation metrics like, per-pixel accuracy, per-class accuracy or some Fully-Convolutional Networks (FCN) that allow comparing FCN-scores over the target domain and the synthetic one, are well suited. Unfortunately, these methods also provide their disadvantages like shown in the aforementioned literature [39, 37, 5], in some case because these metrics are task-specific they are unable to grasp what we can call the "cognitive realism" of an image, so they should not be used alone.

4.3.3 Which method and wich metric for our case?

To evaluate the performance results of our tested frameworks over Image-to-Image translation task using unpaired dataset, we choose to use a qualitative evaluation procedure. A qualitative approach at this early step of our research is the best approach.
Indeed, because the visual quality of translation results is not higher enough to fool a human, a human quantitative evaluation will only show us what we already know: the output results are not similar enough to the target domain. Even if we chose on purpose a ground-truthed historical dataset - HBA-modified- that allow the use of segmentation algorithms, we chose to not use the other mechanical metrics for two main reasons:

- The major part of our work is done one $[256 \times 256]$ random-cropped images which are not well suited for this DIA metric.
- The results can lead us to choose the wrong baseline model for future work, considering that our main goal is to fool a human.

Therefore, conscious of the lack of reproducibility and the subjectivity bias of the qualitative evaluation procedures we chose to build a standardized protocol, keeping in mind further possible uses of synthetic historical handwritten images for DIA tasks.

4.3.4 Protocol

We evaluate the output images over five characteristics, two general characteristics the absence of blurriness and the absence of artifacts in the output images. One source domain characteristic that focuses on the preservation of the structural content of the source domain image. Finally, two target domain characteristics that focus on the synthetic handwritten historical document image regarding what we have all the semantic content and the style content. The specification of each character is defined in the following list:

- The **Abscence of Blurriness** in the output images.
- The **Abscence of Artifacts** in the output images.
- The **Preservation of Source Domain Structural Content** of the source domain. Where the structural content is defined by the number of lines, the presence of initials, the number of paragraphs, the number of words.

- The **Preservation of the Target Domain Semantic Content**. Where the semantic content is defined by the font shape, the readability of letters, the readability of words, the presence or absence of annotations.
- The **Preservation of the Target Domain Style Content**, where the style content englobes mainly the texture, the colors, the overall similarity impression, the background similarity with the preservation of some paper degradation effects, the initial similarity.

Each of these features will be rated from zero to five, where five means that the criterion is fully satisfied and zero means that the criterion is not at all satisfied.

To seriously evaluate a model the examinator will:

- Only chose images that come from the test part of the Evaluation Dataset (ED).
- Chose from two to five idiomatic triad samples to evaluate a model, where the triad is composed by the source domain image sample, the target domain image sample, and the synthetic output image.
- Ensure that the idiomatic triad samples fulfill the properties of the aforementioned characteristics in order to evaluate the model performance at the best.
- For one model, evaluate both: raw $[256 \times 256]$ images and $[256 \times 256]$ random cropped images idiomatic triad samples sets.
- Give a note to the model from one to twenty-five for raw and random cropped images sets. The note of the two sets is then added in one unique note that goes from zero to fifty.

The goal of the protocol is to emphasize the strength and weakness of a model and to provide a standard method in order to improve the quality of synthetic samples for future analysis and implementation rounds.

5 Image-to-Image Translation Task Results and Discussion

This Image-to-Image Translation task results section is divided in two subsections, where the first subsection focuses on GANs models results, starting from the SimGAN implementation folowed by the cycleGAN one. The second subsection evaluates the results obtained by the Neural Style Transfer algorithm.

5.1 Generative Adversarial Networks Models

We chose to start by the worst results obtained following the SimGAN model implementation [32]. The results was so bad that we partially respect the evaluation protocol. Instead of runing the evaluation over the test part of the Evaluation Dataset (ED), we chose to take the last training samples obtained during the training phase for the evaluation. Next, we explore the results obtained with the cycleGAN model. With this implementation we totally respect the evalauation protocol. Additionally, we try to pre-train the model with an auto-encoder for the encoding parts of the generators, trying to improve the previous random-cropped obtained results; thus we re-evaluate the random-cropped results. For doing that, we use the Target Domain Dataset (TDD) and the Source Domain Dataset (SDD).

5.1.1 SimGAN results

Raw Images $[256 \times 256]$ After a training phase of 50 epochs, over $[256 \times 256]$ resized document images coming from the target and the source domain of the ED dataset we obtain inconclusive results. This network implementation is unable to correctly refine the document images as wished.

(a) Target Domain sample *(b) Source Domain sample* *(c) Refined Sample*

Figure 22: **Idiomatic SimGAN Raw Samples Triad** Triad of document images composed by a sample coming from the ED, one coming from the target domain, the second one from the source domain and the third one is the generated output sample. We call the generated output sample: “Refined sample”, because the SimGAN generator is called ”Refiner”. The refined sample should exemplify the Image-to-Image translation objective.

General remarks The blurriness is not really different from the source domain, we only note some blurry effects around the initials and the overall textual information in the refined samples. Concerning the structural content, at first glance is perfectly preserved, we can observe the presence of the initials, the two-column mode is also conserved, the space between paragraphs, the number of the line seem also

to be the same than in the source domain. Nevertheless, the structural content is way too close to the source domain samples. Our goal is not to produce some variations in the source domain it is to perform a mapping between two different domains. The semantic content and the style content of the target domain is not at all grasped by the model. Even the background color is not modified at all. To be sure that we grade correctly the refined samples we will look more closely to some refined samples with a six scale zoom and comment the presence of some pixel-level artifacts.

Zoomed Raw Images $[256 \times 256]$ If we look closely in Figure 23, some colored artifacts are introduced by the refiner inside the textual content and around the initials, at the pixel level. We can also notice the presence of other artifacts at the top and bottom of the document pages. These artifacts are probably due to the fact that the ED images are resized by $[256 \times 307]$ with the PyTorch transform before a random cropping of $[256 \times 256]$.

We can conclude that the model obtains a raw sample mark of 12.

- **Abscence of Blurriness** in the output images : 3.
- **Abscence of Artifacts** in the output images: 2.
- **Preservation of Source Domain Structural Content**: 5.
- **Preservation of the Target Domain Semantic Content**: 1.
- **Preservation of the Target Domain Style Content**: 1.

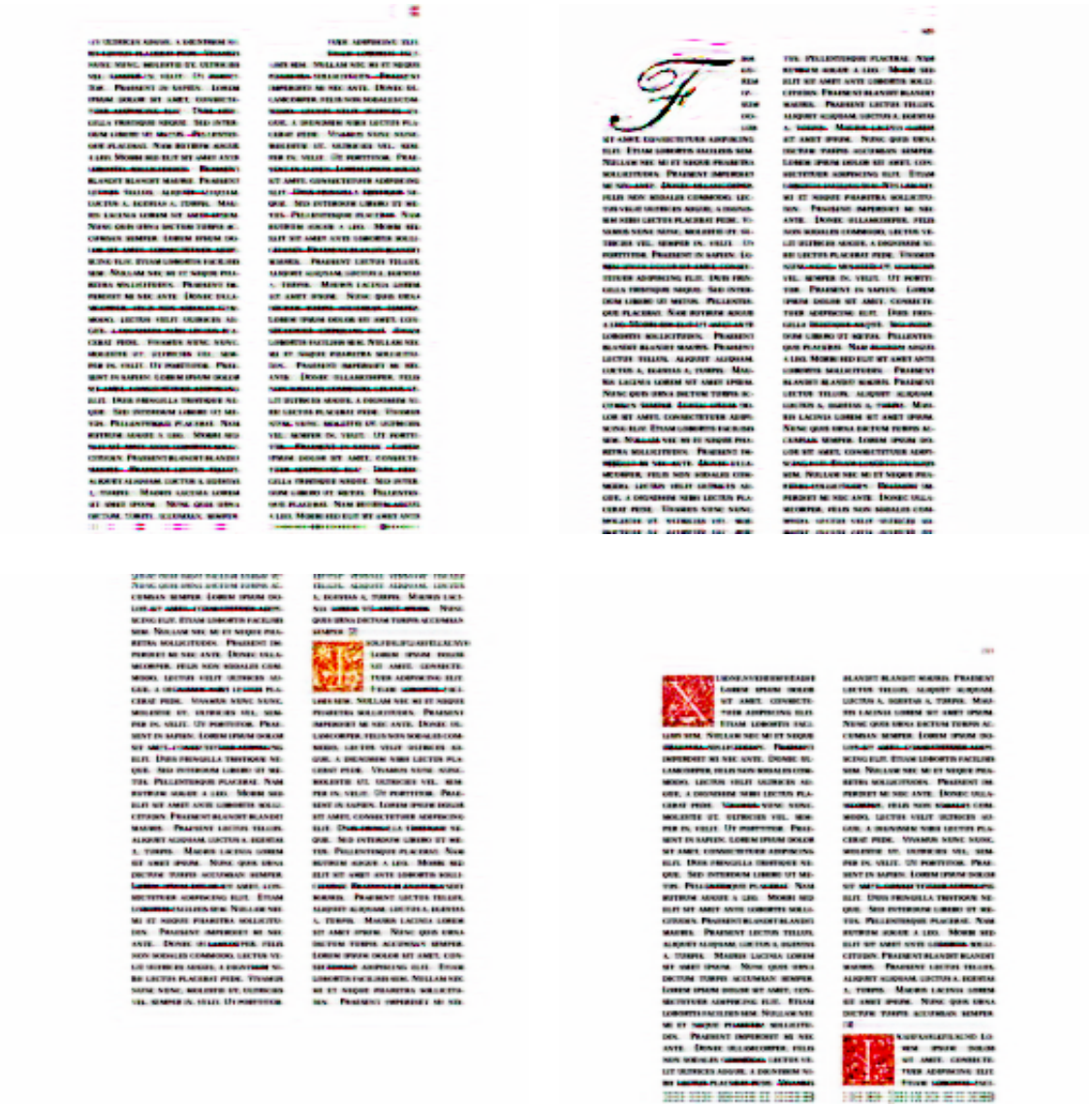
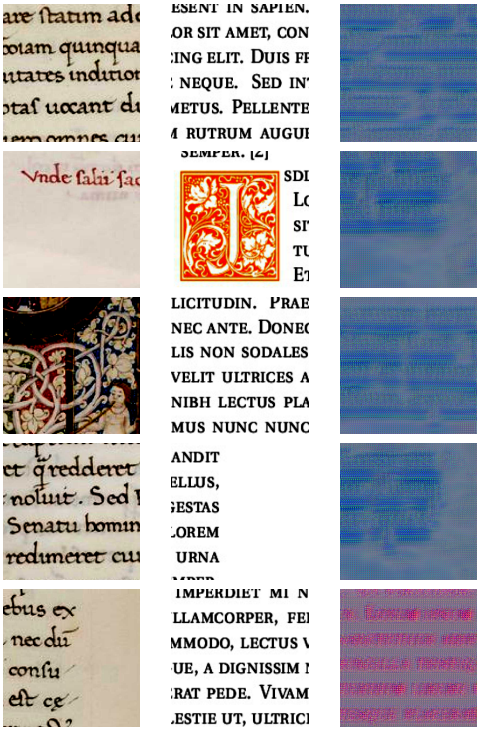


Figure 23: **Zoomed Raw Refined Samples.** These four samples exemplify the presence of colored artifacts in the semantic and structural content of refined sample images. The images seem to be unaligned because of the white background but, they express the same $[256 \times 256]$ pixel dimension.

Random Cropped Images $[256 \times 256]$ After a training phase of 50 epochs, over $[3086 \times 2132]$ resized, center cropped $[1886 \times 1232]$ and finally random cropped $[256 \times 256]$ document images coming from the target and the source domain of the ED dataset, we obtain worse results than for the previous section with the raw images.

General remarks We can not say if there is a blurry effect or not because we are not able to repair any structural, semantic or style content. Indeed, the refined output samples are unreadable. Regarding the structural content, we can notice that the shape of the words, the spaces between lines and the number of

lines are not the same. On these samples, we only notice the presence of structured artifacts where the color channel seems to change between blue and red. This effect could indicate that the network faces to a well known GANs training problems [10]. Indeed, when a GAN model is unable to converge the parameters oscillate, these oscillations can explain the color channel switch. Concerning the artifacts, it is likely that the generator collapses, which explains the limited varieties of samples and the persistence of the same shapes in the artifacts.



(a) Target Domain Samples *(b) Source Domain Samples* *(c) Refined Samples*

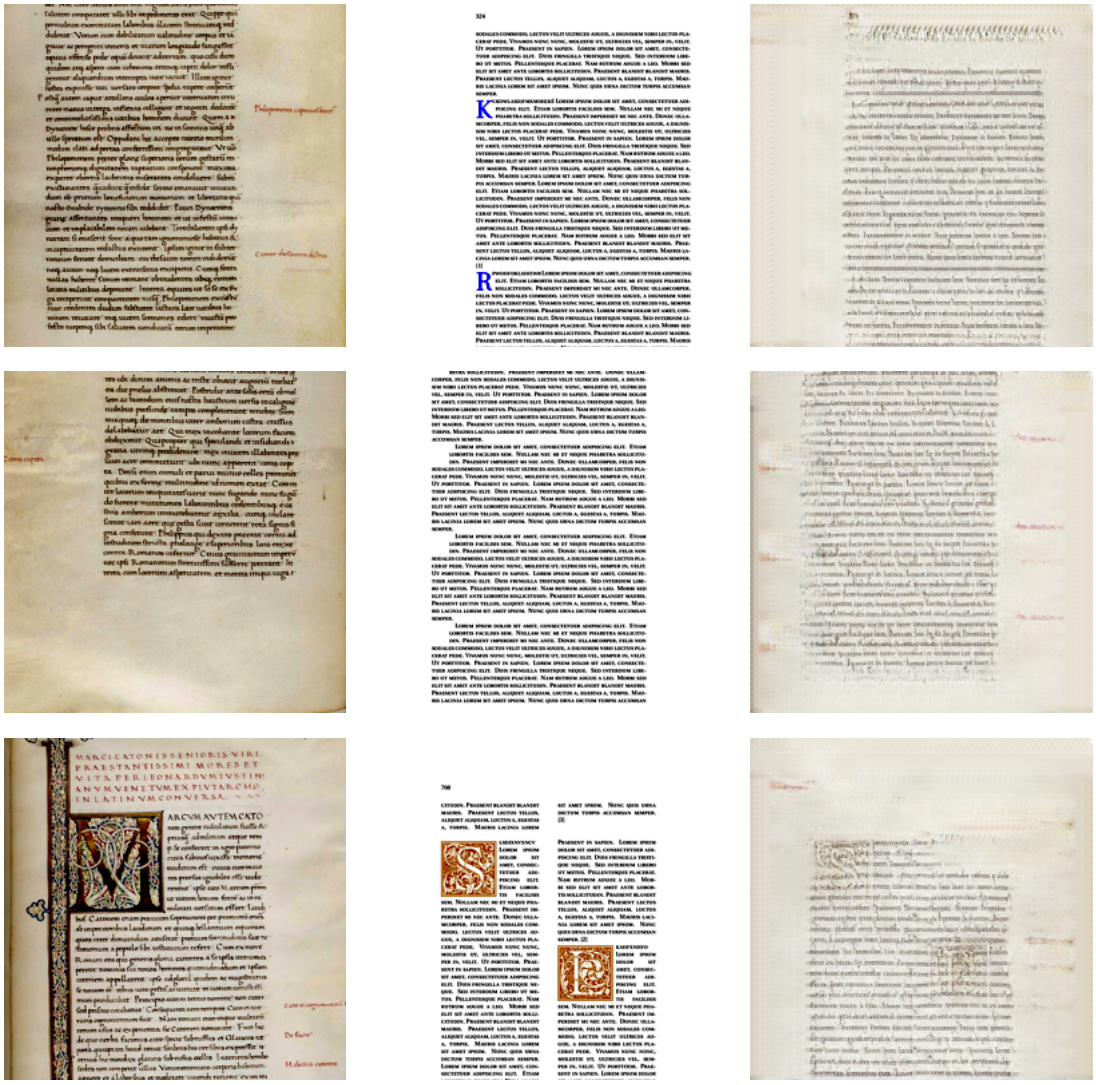
Figure 24: **Idiomatic SimGAN Random Cropped Samples Triad.** Triad of document images composed by samples coming from the ED, in the first coloumn there is the samples coming from the target domain, in the second one the samples coming from the source domain and in the third one there is the generated output samples. We call the generated output samples: "Refined samples", because the SimGAN generator is called "Refiner". The refined sample should exemplify the Image-to-Image translation objective.

We can conclude that this network implementation, just as much for raw samples, if not more, is unable to correctly refine the document images to a finer granularity. Indeed, the network is unable to produce neither, a refined sample that looks like a sample of the target domain nor, a refined sample that keep the semantic content of the source domain.
Regarding the previous configuration, with the raw images in Figure 22, we are entitled to expect the framework to produce a similar output type, but it is not the case. In a technical point of view this matter of fact can be explained, not only by the classical GANs training issues, but also by the fact that in term of learning objective the SimGAN model only use a self-regularization loss $\mathcal{L}_{L1}$, which is simply a distance loss that minimizes the distance between the source domain samples and the refined buffered

samples. It seems conceivable that this model is unable to transfer the style from one domain to another one if the probability distributions of the two domains are too distinct which is our case. Indeed, in the original implementation, the two domains were really close the informational space was reduced to eye gaze information. We can also notice that the architecture of the SimGANs is far more complex than the following one. Then, the model obtains a random-crop samples mark of 0, and an overall one of 12.

5.1.2 cycleGAN results

Raw Images $[256 \times 256]$ We chose to train the cycleGAN network during 50 epochs, over $[256 \times 307]$ resized document images, followed by a random crop of $[256 \times 256]$ coming from the target and the source domain of the ED dataset. Even if the resolution is low, this network and dataset setting show satisfying synthetic results, as shown in the following Figure.



(a) Target Domain Samples

(b) Source Domain Samples

(c) Synthetic Samples

Figure 25: **Idiomatic cycleGAN Raw Samples Triad** Triad of document images composed by three idiomatic samples coming from the ED, the first colomn express the samples coming from the target domain, the second ones from the source domain and the third ones are the generated synthetic output samples. The synthetic samples should exemplify the Image-to-Image translation objective.

General remarks Concerning the blurriness of the generated synthetic output samples, we can argue that the blurry effect is slightly more present than in the target domain samples. The effect is concentrated over the textual information. On the artifacts absence point of view, we note the presence of some checkboard artifacts all around the margins of the generated document, but also at the top and at the bottom of the documents. Regarding the structural content expressed by the source domain samples, we note that he is not well preserved, inasmuch as the presence of the initials is hidden by some textual artifacts at the pixel level, look at samples one and three in Figure 25. It may be also noted that the

paragraph separations with the line breaks are only partially respected, but this can be explained by the fact that there is no paragraphs and line breaks in the target domain dataset distribution. Still, concerning the structural content, we can also notice that the two column-mode, present in the third source domain sample, does not end up in the synthetic generated sample and that the number of lines present in the synthetic sample is, consequently, probably different from the source domain sample. We think that this matter of fact is also due to the target domain dataset distribution. Regarding the semantic content, we can note that at this level of details, still in Figure 25, the font shape and the readability of letters and words is quite similar than in the target domain samples. We also remark the presence of colored annotations in the margins of the generated synthetic samples. The overall style content of the target domain is preserved, we retrieve the texture, some color information and we also note several paper degradation effects, like in the synthetic sample number two. Concerning the background color, unfortunately, we notice that the brown color is less strong than in the target domain. Finally, regarding the initials, we notice that the style of them is not preserved like in the synthetic sample number one.

Zoomed Raw Images $[256 \times 256]$ If we look closely, like in Figure 26, we remark in a more obvious way the artifacts presence. Regarding the semantic content, we also notice that the shape of the font is quite dissimilar to the target one. We remark that the annotation sometimes is not colored and that they are less readable than the textual information, because of a blurry effect. Regarding the style content, it's now clear that the initials are superimposed on the textual information.

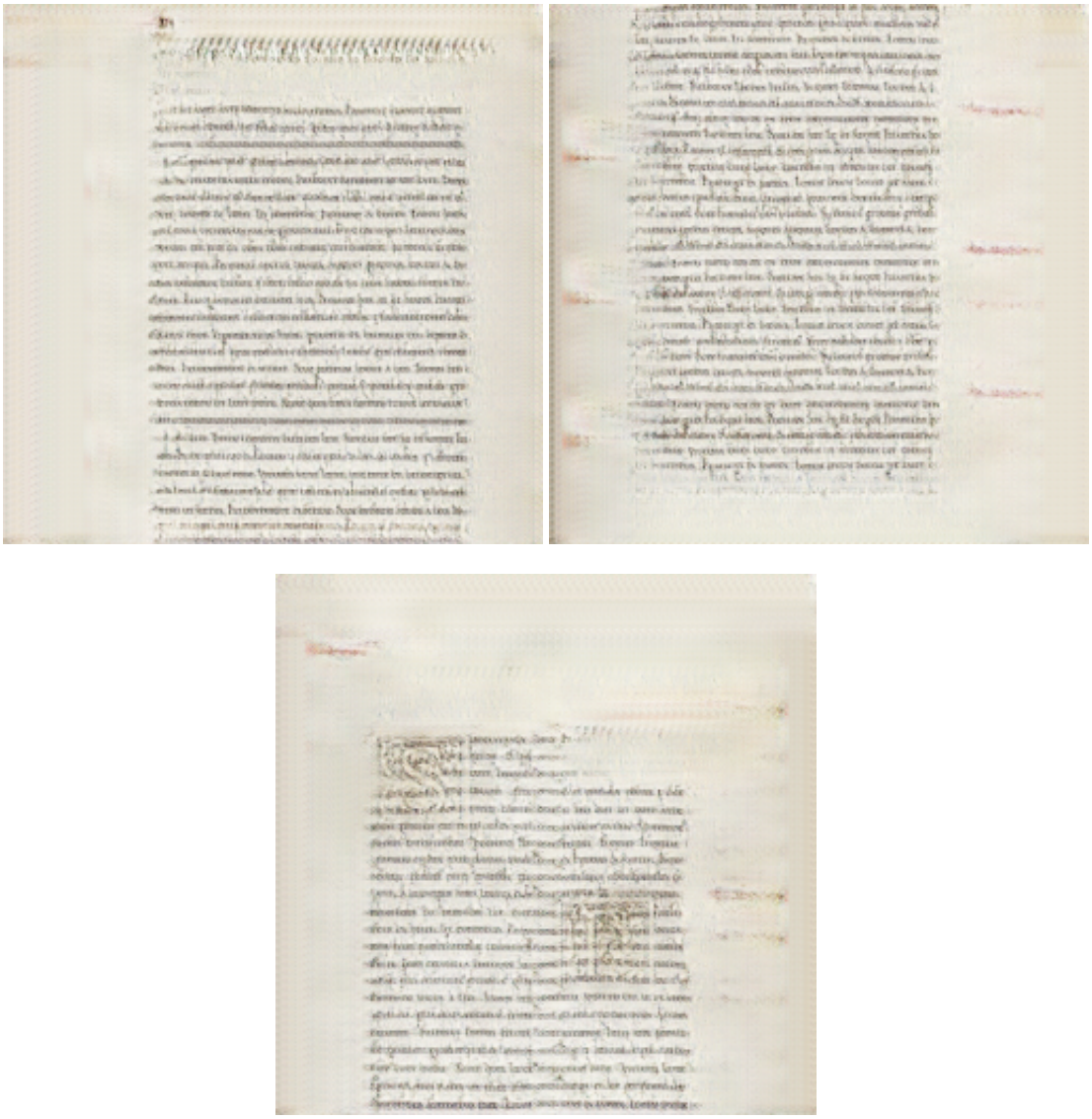


Figure 26: **Zoomed Raw Synthetic Samples.** These three zoomed samples, at a scale of one point six, show what type of characteristics we can retrieve in the synthetic generated output samples.

We can conclude that the model performs quite well with a raw sample mark of 14. Even if the mark is not so different than for the SimGAN framework the points are well balanced which represent better what we try to achieve in this work.

- **Abscence of Blurriness** in the output images : 3.
- **Abscence of Artifacts** in the output images: 3.
- **Preservation of Source Domain Structural Content**: 3.

- **Preservation of the Target Domain Semantic Content**: 3.
- **Preservation of the Target Domain Style Content**: 3.

Random Cropped Images $[256 \times 256]$ After a training phase of 50 epochs, over $[3086 \times 2132]$ resized, center cropped $[1886 \times 1232]$ and finally random cropped $[256 \times 256]$ document images coming from the target and the source domain of the ED dataset, we obtain satisfying results that show us with finer details what the model can learn at pixel-level. We have shown in the following Figure the results obtained over four idiomatic testing samples.

General remarks Concerning the blurriness of the random-cropped generated synthetic output samples we observe any sign of blurriness if we compare the samples to the target domain ones. Regarding the presence of artifacts, we note some font upper mark that is present at the top of the word in the generated synthetic samples one and four. This artifact presence can be explained by the fact that there are some calligraphic accents over some target domain samples, like in target samples one and two. The structural content expressed by the sample two is too preserved, there is almost no font or initial change, the number and the shape of words and lines are almost fully preserved. In sample three, we observe that the font shape is more curved than in sample two. This preservation effect can be explained by the presence of initials in the source domain sample, where this particular feature presence modify the way that the generated sample is produced by the generator. Indeed, samples one and four express structural modifications regarding the font shapes, the number of lines and words that look like the target domain more than the source domain and, if you look to the source domain samples you can note the absence of initials - this fact is also verified by the other testing samples (for more results samples look at the experimental results present in the code link). Concerning the semantic content, we can note that the readability of letters and words is not possible for the generated synthetic samples, even if the overall font style is quite similar to target samples. On style matter, we retrieve the texture but the color present in the source domain are absent in the target domain. This can be explained by the fact that the color information is lost over the training process like shown in the analysis of the pre-training phase in subsection 4.1.2. Concerning the background color, the color is less strong than in the target domain, this can also be explained by this loss of information regarding the color, indeed the network tends to standardize the color to a single mode of expression.

(a) Target Domain Samples

(b) Source Domain Samples

(c) Synthetic Samples

Figure 27: **Idiomatic cycleGAN Random Cropped Samples Triad.** Triad of document images composed by samples coming from the ED, in the first coloumn there are samples coming from the target domain, in the second one samples coming from the source domain and in the third one there are generated synthetic output samples. The generated synthetic samples should exemplify the Image-to-Image translation objective.

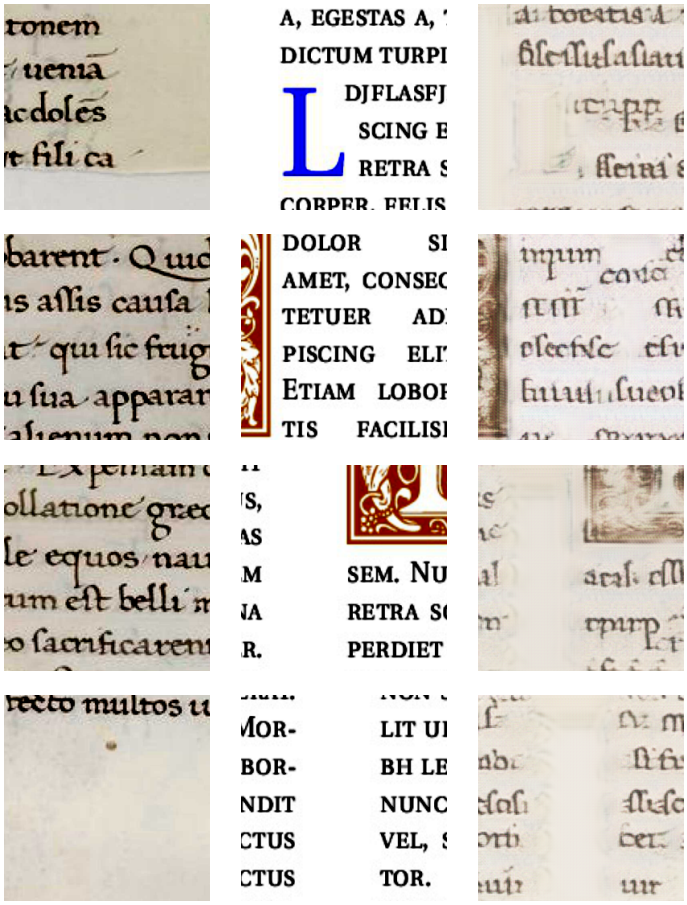
Even if there is a range of improvement regarding the semantic and structural content, in particular, concerning the font shape and the respect of the features constraints mentioned in the evaluation protocol for properties like words shape, words and lines number. We also note that the line base is well aligned, which is an important characteristic for line base DIA task for example. We can conclude that the model performs quite well with random-cropped samples with a mark of 18.

- **Abscence of Blurriness** in the output images : 5.
- **Abscence of Artifacts** in the output images: 4.
- **Preservation of Source Domain Structural Content**: 3.

- **Preservation of the Target Domain Semantic Content**: 3.
- **Preservation of the Target Domain Style Content**: 3.

Pre-trained cycleGAN With Random Cropped Images $[256 \times 256]$ First, we pre-train the model with an auto-encoder for the encoder part of the generators of 25 epochs, over $[3086 \times 2132]$ resized, center cropped $[1886 \times 1232]$ and finally random cropped $[256 \times 256]$ document images coming from the Target Domain Dataset and the Source Domain Dataset. Then we train the model with the pre-trained generator over 50 epochs, using $[3086 \times 2132]$ resized, center cropped $[1886 \times 1232]$ and finally random cropped $[256 \times 256]$ document images coming from the target and the source domain of the ED dataset, we obtain more satisfying results than the previous setting. We showed, in the following Figure, the results obtained over four idiomatic testing samples.

General remarks The blurriness of the random-cropped generated synthetic output samples is more present than with the previous setting. Regarding the artifacts, they are less present and rarer than in the previous setting, the ratio seems to match to what we retrieve in the target domain, but some new character repetition series seems to appear in samples one and two. We notice that the structural content expressed by the samples embrace the goals that we try to achieve and that was not fulfilled by unpre-trained model. Indeed, we the number of words and lines, even the shape of words match to the source domain when initials are present. The two-column mode is also preserved as shown in sample four. But there is still some mismatch with the blue initials like we notice in the Figure 26, we can link this state of affairs to the color issues that we mentioned during the analysis of the previous results. Concerning the semantic content and the style content, no major effects are expressed except that the background color seems to be a little bit more nuanced, like shown in sample four.



(a) Target Domain Samples *(b) Source Domain Samples* *(c) Synthetic Samples*

Figure 28: **Idiomatic Pretrained cycleGAN Random Cropped Samples Triad.** Triad of document images composed by samples coming from the ED, in the first coloumn there are samples coming from the target domain, in the second one samples coming from the source domain and in the third one there are generated synthetic output samples. The generated synthetic samples should exemplify the Image-to-Image translation objective.

This trained cycleGAN model has shown some improvements regarding the structural content and the style one. If we think to our main thesis goal which was to full a human examinator, we are still far away from our objective regarding the style and semantic content of the synthetic samples. Our second objective is to allow researchers to perform some DIA tasks over this synthetic data, we have to admit that with this output quality it is still not possible to use this synthetic data for some DIA tasks like OCR analysis for example. Thus, we can conclude that the model performs better than the previous setting, because of the good balance between the protocol characteristics, with a random-cropped samples final mark of 18, and an overall one of 32.

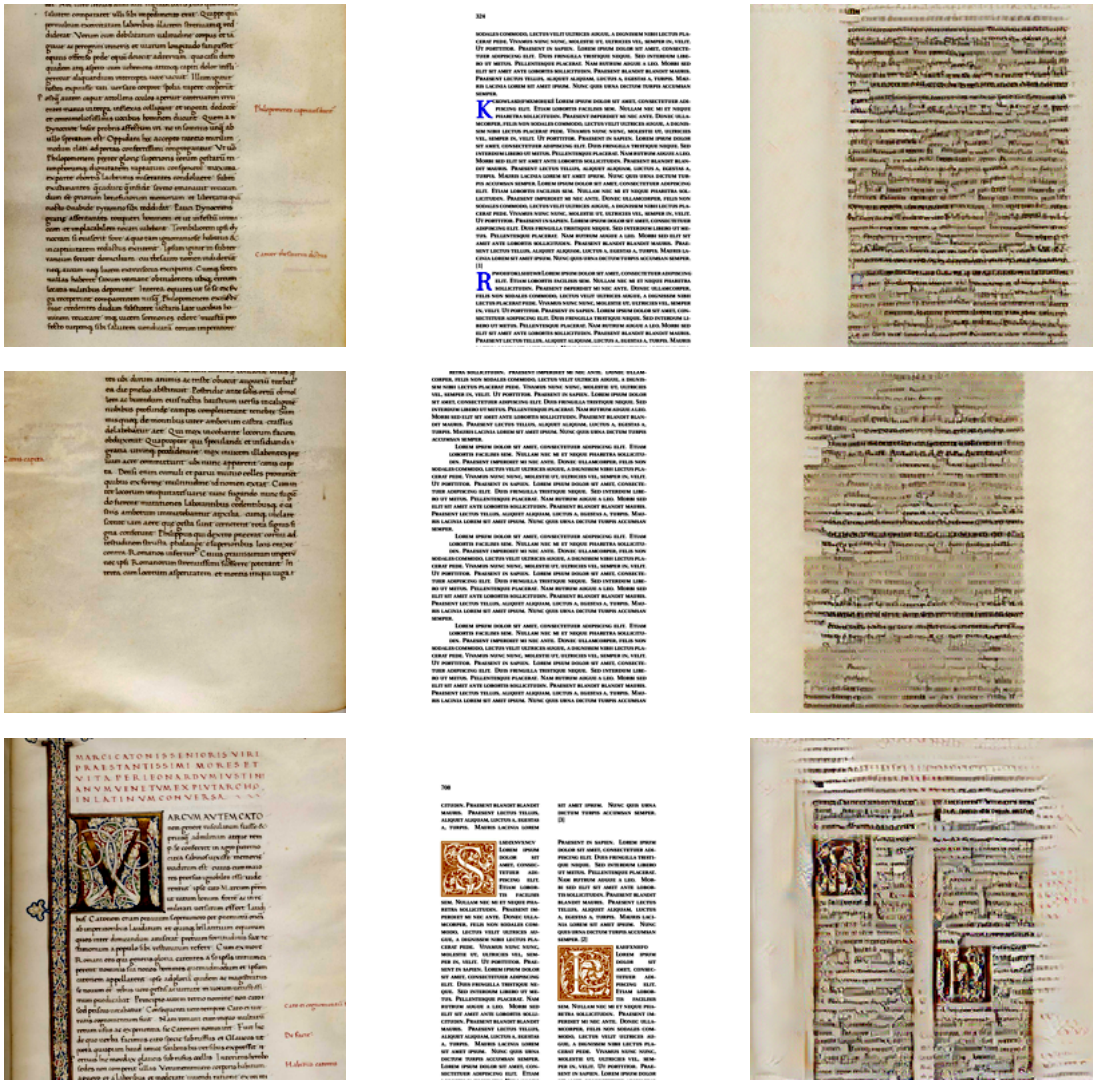
- **Abscence of Blurriness** in the output images : 4.

- **Abscence of Artifacts** in the output images: 4.
- **Preservation of Source Domain Structural Content**: 4.
- **Preservation of the Target Domain Semantic Content**: 3.
- **Preservation of the Target Domain Style Content**: 3.

5.2 Neural Stlye Transfer Algorithm Results

Finally, we will explore the results of the Neural Style Transfer algorithm model following the evaluation protocol. We have try to improve the raw and random-cropped results using a pre-trained VGG-19 model over a classification task using the Target Domain Dataset (TDD), instead of applying the VGG-19 model pre-trained with ImageNet; thus we re-evaluate the results with the evaluation protocol.

ImageNet VGG-19: Raw Images $[256 \times 256]$ We chose to apply the Neural Style Transfer Algorithm [9], over $[256 \times 307]$ resized document images, followed by a random crop of $[256 \times 256]$ coming from the target and the source domain of the testing part of the ED dataset. Comparing to the results obtained by the cycleGAN, for raw images, we clearly understand that the algorithm does not seems to perform what we expect from him, like in the three samples shown in the following Figure. Keep in mind that we run the algorithm with an ImageNet pre-trained model.



(a) Target Domain Samples

(b) Source Domain Samples

(c) Synthetic Samples

Figure 29: **Idiomatic NST Raw Samples Triad - Pretrained with ImageNet** Triad of document images composed by three idiomatic samples coming from the ED, the first colomn express the samples coming from the target domain wich represent the style that we want to learn, the second ones from the source domain which represent the content that we want to preserve and the third ones are the generated synthetic output samples. The synthetic samples should exemplify the Image-to-Image translation objective.

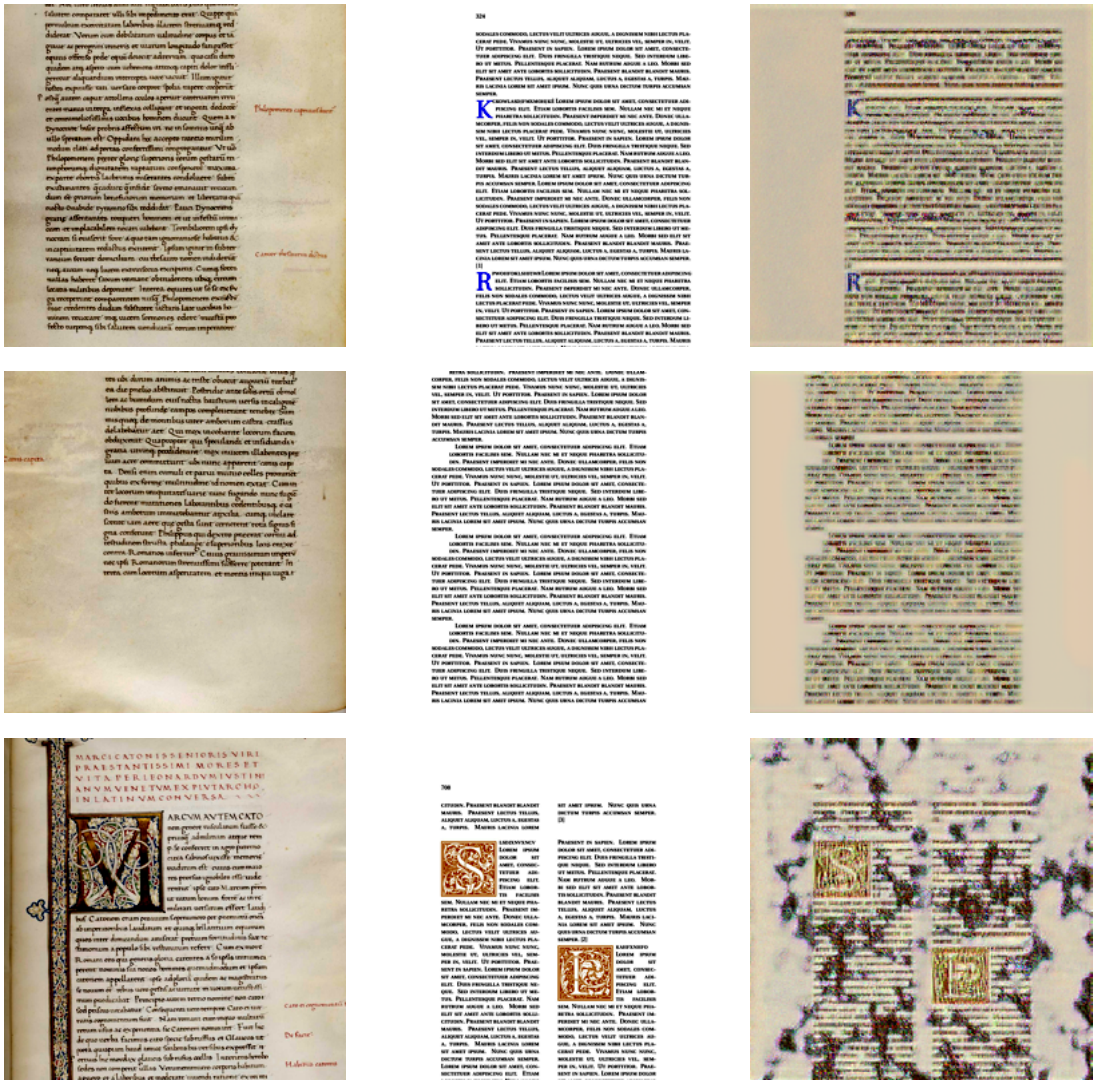
General remarks Concerning the blurriness of the generated synthetic output samples, we remark the presence of blurry stain marks. The effect is distributed over all the textual information. This blurry effect is accompanied by the presence of several artifacts at the top, bottom and right margins of the documents like shown in sample three. It also seems that some colored artifacts are present in the third sample, they are distributed over all the textual information. In the whole the structural content is well preserved, we can recognize the page structure of each sample of the source domain, even for the two-

column mode, but because of the blurry effect, the paragraphs and lines outlines are poorly defined. The baselines become shaky with the use of this algorithm. The initials in sample one are totally absent in the synthetic samples. Concerning the semantic content, we can notice that the font shape and the readability of letters and words are only partially preserved. In the third sample, we also notice the presence of artifacts that look like annotations. The overall style content of the target domain is also partially preserved, we retrieve the texture but not the color information, the paper degradations are also absent. The background color is more unified and similar to the target domain than in the cycleGAN samples. Finally, regarding the initials, the style and the semantic content is not preserved at all, even if we can guess and infer the meaning of the initial stains.

We can conclude that the model performs badly. The idea of mixing the style content coming from the target domain and the semantic content coming from the source domain is a bad idea. Indeed, the style transfer is performed and uniformly applied over the all image document without any regard for the semantic content of the target domain. Conceptually, we can understand that the style of a historical document is highly correlated to his content, even if for artistic style transfer this seems to be a good way of thinking to the translation problem. We can argue that hoping that this particular way [9] to separate the content from the style for documents is a chimera of the mind - for more details over the model, implementation sees subsection 2.3. We give to the ImageNet VGG-19 raw images model implementation a mark of 8.

- **Abscence of Blurriness** in the output images : 1.
- **Abscence of Artifacts** in the output images: 1.
- **Preservation of Source Domain Structural Content**: 2.
- **Preservation of the Target Domain Semantic Content**: 2.
- **Preservation of the Target Domain Style Content**: 2.

Target Domain Dataset (TDD) VGG-19: Raw Images $[256 \times 256]$ We chose to apply the Neural Style Transfer Algorithm [9], over $[256 \times 307]$ resized document images, followed by a random crop of $[256 \times 256]$ coming from the target and the source domain of the testing part of the ED dataset. This time the VGG-19 convolutional neural network has been pre-trained over the target domain dataset for a classification task. This pre-training phase has canceled the general blurry effect obtained in the previous results, but the overall result is not satisfactory.



(a) Target Domain Samples

(b) Source Domain Samples

(c) Synthetic Samples

Figure 30: **Idiomatic NST Raw Samples Triad - Pretrained with TDD** Triad of document images composed by three idiomatic samples coming from the ED, the first column express the samples coming from the target domain which represent the style that we want to learn, the second ones from the source domain which represent the content that we want to preserve and the third ones are the generated synthetic output samples. The synthetic samples should exemplify the Image-to-Image translation objective.

General remarks Comparing to the previous results we remark that the blurry effect with stain presence has vanished. Regarding the artifacts, they are dissimilar to the previous one; now we can only notice their presence at pixel-level and they are colored like in sample one and two. The structural content of the source domain samples is better preserved, the outlines of paragraphs and lines are now well defined. The baselines are stable due to the disappearance of the blurry effect. The initials in sample one are now well defined. Concerning the semantic content, we notice that the font shape and

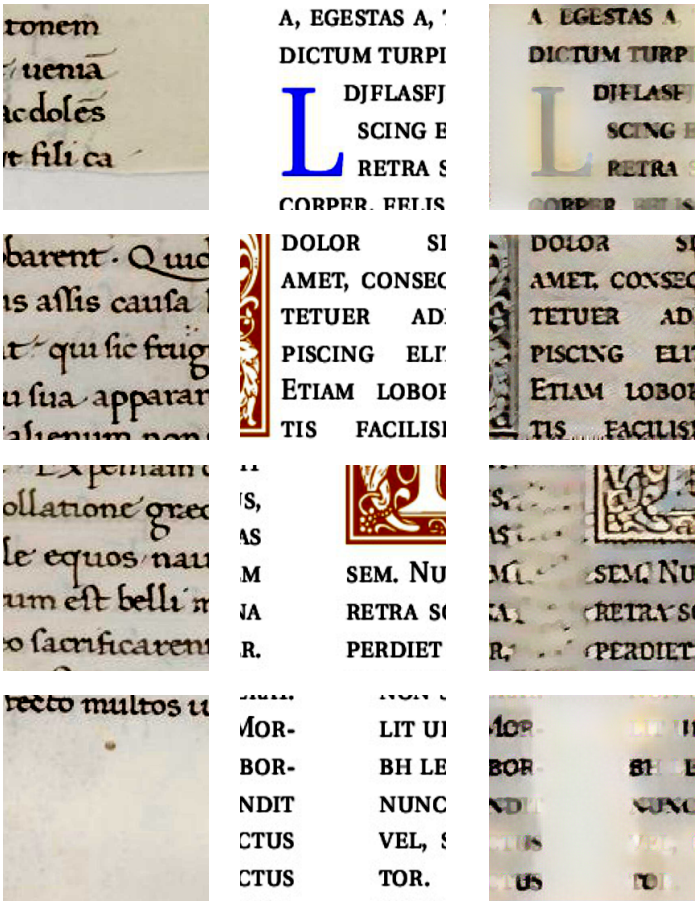
the readability of letters and words are more preserved but too similar to the source domain in the sense of the SimGAN raw results. The annotation is now totally absent. Regarding the style content, we can see in the third sample what we explain in the conclusive part of the previous evaluation. When a huge initial with some decorations is present in the target domain the overall style is applied and mixed to the all synthetic document, which is not what we want. The color of the background is now too much standardized.

We can conclude again that the model performs badly, but for different reasons that are due to the pre-training phase. By training the model to classify document images coming from the TDD dataset we train the VGG-19 to learn the feature space of documents. By doing this the network is now able to recognize and separate the textual information from the background information which leads us to a more precise output regarding the structural information. Nevertheless, as we said in the past evaluation, the style is still applied to the overall image without distinction which leads us to poor results where the synthetic data is highly dependent to the target domain sample. We give to the VGG-19 raw images model implementation, pretrained with TDD, a mark of 11.

- **Abscence of Blurriness** in the output images : 4.
- **Abscence of Artifacts** in the output images: 1.
- **Preservation of Source Domain Structural Content**: 4.
- **Preservation of the Target Domain Semantic Content**: 1.
- **Preservation of the Target Domain Style Content**: 1.

ImageNet pre-trained VGG-19 model: Random Cropped Images $[256 \times 256]$ We apply the Neural Style Transfer Algorithm over $[3086 \times 2132]$ resized, center cropped $[1886 \times 1232]$ and finally random cropped $[256 \times 256]$ document images coming from the target and the source domain of the testing part of the ED dataset using the VGG-19 model pre-trained with ImageNet dataset. We show, in the following Figure, the results obtained over four idiomatic testing samples.

General remarks The blurry effect of the random-cropped generated synthetic output samples is light. We notice the presence of stain artifacts in sample one and pixel artifacts in sample three. We notice that the structural content expressed by the samples is too close to the source domain samples for sample one and three. The sample four show us that the two-column mode is preserved but some words seem to vanish because of the way the style is computed by the domain, indeed some letters are filled with wight ink. The semantic content in the synthetic sample is also too close to the source domain, indeed even if words and letters are almost perfectly readable the font is too much similar to the Arial font. Regarding the style the color is absent, we note the presence of a binarization effect in sample three where we can clearly recognize the defect in the way that the style of the target image is computed because the blank line is filled with some wight and black pixels. Regarding the initial, we notice that the model does not seems to learn the colors, this is quite normal because the colors are mainly dependent on the style content expressed by the target domain samples. Finally, we retrieve some interesting background dissimilarities in term of color over the set of generated synthetic outputs.



(a) Target Domain Samples *(b) Source Domain Samples* *(c) Synthetic Samples*

Figure 31: **Idiomatic NST Random cropped Samples Triad - Pretrained with ImageNet** Triad of document images composed by three idiomatic samples coming from the ED, the first colomn express the samples coming from the target domain wich represent the style that we want to learn, the second ones from the source domain which represent the content that we want to preserve and the third ones are the generated synthetic output samples. The synthetic samples should exemplify the Image-to-Image translation objective.

We can conclude again that the model performs badly, for the same conceptual reasons than in the previous NST settings. Regarding the general remarks following our evaluation protocol, we give to the ImageNet pre-trained VGG-19 random-cropped images model a mark of 11.

- **Abscence of Blurriness** in the output images : 3.
- **Abscence of Artifacts** in the output images: 2.
- **Preservation of Source Domain Structural Content**: 4.
- **Preservation of the Target Domain Semantic Content**: 1.

- **Preservation of the Target Domain Style Content**: 1.

TDD pre-trained VGG-19 model: Random Cropped Images $[256 \times 256]$ We apply the Neural Style Transfer Algorithm over $[3086 \times 2132]$ resized, center cropped $[1886 \times 1232]$ and finally random cropped $[256 \times 256]$ document images coming from the target and the source domain of the testing part of the ED dataset using the VGG-19 model pre-trained with the Target Domain Dataset. We show, in the following Figure, the results obtained over the same four idiomatic testing samples than in the previous evaluation. The results are better than in the previous setting but miss the important translation work.

General remarks There is no particular blurry effect, or at least it is very difficult to distinguish this effect when an initiative is present in the synthetic sample like in samples two and four. As for the blurry effect, the distinction between artifacts and s̈tyle pixel" modifications cannot be clearly made. Regarding the structural content expressed by the samples it is worse than the previous setting, indeed the structure suffers no major structural modifications comparing to the source domain. It should be good news if the semantic content was modified, but it is not the case at all. We can easily remark that the shape of the font, words, and line is the same than in the source domain even if our purpose is to mimic the semantic content of the target domain. Finally, concerning the style, we notice the presence of color at tiny pixel levels when the initials with huge decorations are present in the sample. This is not the case in sample one for example when the initial is just colored, even if the outline of the initial is lightly colored, this matter of fact can be explained by the fact that the style is highly correlated to the target domain like we previously said, where no color is present. To conclude the background variety has been improved comparing to the previous setting.

(a) Target Domain Samples *(b) Source Domain Samples* *(c) Synthetic Samples*

Figure 32: **Idiomatic NST Random cropped Samples Triad - Pretrained with ImageNet** Triad of document images composed by three idiomatic samples coming from the ED, the first colomn express the samples coming from the target domain wich represent the style that we want to learn, the second ones from the source domain which represent the content that we want to preserve and the third ones are the generated synthetic output samples. The synthetic samples should exemplify the Image-to-Image translation objective.

We can conclude again that the model performs better than the previous one because of the adapted pre-training phase; nevertheless, for the same conceptual reasons than aforementioned, the model misses the domain transfer point. Regarding the general remarks and following our evaluation protocol, we give to the TDD pre-trained VGG-19 random-cropped images model a mark of 15. The overall model with the best implementation which is the TDD pre-trained one obtain a total score of 26.

- **Abscence of Blurriness** in the output images : 4.
- **Abscence of Artifacts** in the output images: 4.
- **Preservation of Source Domain Structural Content**: 4.

- **Preservation of the Target Domain Semantic Content**: 1.
- **Preservation of the Target Domain Style Content**: 2.

6 Conclusion

Based on the results we obtained we can say that the well suited model for future DIA tasks seems to be the GANs frameworks, more particularly the tested cycleGAN model that use the bi-directional reconstruction cycle loss. Regarding the goals that we have set at the begining of the work, we think that we meet two of the three requirements.

Indeed, we have brush a comprehensive panel of the different architectural tracks, the generative one and the neural style one. These tracks lead us to a more simple and general-purpose framework than the existing one that allows to perform Image-to-Image translation task in one single move. Even if the results are perfectible, we think that it's a promising approach. The main drawback for the cycleGAN framework is the image resolution constraint. Actually, the results obtained with the raw image setting mode are enthusiastic, unfortunately, the computational time required to train the model is too expensive. We can conclude that the proposed implemented approaches, tested in this work, lead us to an early practical solution for Image-to-Image Historical Handwritten Document translation task.

But the last goal, which consists to show the possibility to create an unlimited amount of complex synthetic document is not reached. Indeed, there is some limitation due to dataset constraints. First of all the cycleGAN model seems to be unable to learn to generate realistic documents if the target document dataset is composed of more than one book. This matter of fact suppose that for each book we want to mimic the style we must at least re-train the model during a certain amount of time.
Secondly, actually, the synthetic gound-truth generator is not developed enough to produce real usable annotation for DIA tasks like layout segmentation task for example, mainly because we chose to spend the major part of our time over the mapping function.
Finally, we need to find a way to balance the dataset samples in order to have a good ratio of colored documents, but also a good ratio of initials, in both domain the source one and the target one.

7 Future Work

On the GANs side We think that we should try other architecture type, like Variational Autoencoder in order to add some variety in the output samples for example. We also think that the I/O resolution of the document images is the main drawback, in order to overcome that one solution should be to use a ProgressiveGAN approche like in [18].

On the NST side We don't think that the way to compute the style of an image using a Gram matrix is well suited for our task. This deep learning algorithm works well with artistic style transfer because we can decorrelate the style from the content of a painting for example, but we do not think that we can do the same with Historical Document the content and the style are too intricate. It could be a good solution to think to a specific Neural Style Transfer for document images.

In a general point of view We think that the next step is to try to test the results obtained with the best model over some classification tasks for example. The goal is to use synthetic data as a data-augmentation technique in order to see if a neural network can better classify document images with this synthetic data than without them. It will give us some additional motivation to pursue this interesting work.

References

[1] Sagie Benaim and Lior Wolf. One-Sided Unsupervised Domain Mapping. *CoRR*, abs/1706.00826, 2017.

[2] Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. Unsupervised Pixel-Level Domain Adaptation with Generative Adversarial Networks. *CoRR*, abs/1612.05424, 2016.

[3] S. Capobianco and S. Marinai. DocEmul: A Toolkit to Generate Structured Historical Documents. *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, 01:1186–1191, Nov 2017.

[4] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets. *CoRR*, abs/1606.03657, 2016.

[5] Yunjey Choi, Min-Je Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation. *CoRR*, abs/1711.09020, 2017.

[6] Christian Clausner, Stefan Pletschacher, and Apostolos Antonacopoulos. Aletheia - An Advanced Document Layout and Text Ground-Truthing System for Production Environments. *2011 International Conference on Document Analysis and Recognition*, pages 48–52, 2011.

[7] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial Feature Learning. *CoRR*, abs/1605.09782, 2016.

[8] Leon A. Gatys, Matthias Bethge, Aaron Hertzmann, and Eli Shechtman. Preserving Color in Neural Artistic Style Transfer. *CoRR*, abs/1606.05897, 2016.

[9] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image Style Transfer Using Convolutional Neural Networks. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[10] Ian Goodfellow. NIPS 2016 Tutorial: Generative Adversarial Networks. *CoRR*, abs/1701.00160, 2017.

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

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

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[14] He Huang, Phillip S Yu, and Changhu Wang. An Introduction to Image Synthesis with Generative Adversarial Nets. *arXiv preprint arXiv:1803.04469*, 2018.

[15] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-Image Translation with Conditional Adversarial Networks. *CoRR*, abs/1611.07004, 2016.

[16] Yongcheng Jing, Yezhou Yang, Zunlei Feng, Jingwen Ye, and Mingli Song. Neural Style Transfer: A review. *CoRR*, abs/1705.04058, 2017.

[17] Nicholas Journet, Muriel Visani, Boris Mansencal, Kieu Van-Cuong, and Antoine Billy. DocCreator: A New Software for Creating Synthetic Ground-Truthed Document Images. *Journal of imaging*, 3(4):62, 2017.

[18] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive Growing of GANs for Improved Quality, Stability, and Variation. *CoRR*, abs/1710.10196, 2017.

[19] Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim. Learning to discover cross-domain relations with generative adversarial networks. *arXiv preprint arXiv:1703.05192*, 2017.

[20] Thomas Konidaris, Basilios Gatos, Kostas Ntzios, Ioannis Pratikakis, Sergios Theodoridis, and Stavros J. Perantonis. Keyword-guided word spotting in historical printed documents using synthetic data and user feedback. *International Journal of Document Analysis and Recognition (IJDAR)*, 9:167–177, 2007.

[21] Chuan Li and Michael Wand. Precomputed Real-Time Texture Synthesis with Markovian Generative Adversarial Networks. *CoRR*, abs/1604.04382, 2016.

[22] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised Image-to-Image Translation Networks. *CoRR*, abs/1703.00848, 2017.

[23] Ming-Yu Liu and Oncel Tuzel. Coupled Generative Adversarial Networks. *CoRR*, abs/1606.07536, 2016.

[24] Fujun Luan, Sylvain Paris, Eli Shechtman, and Kavita Bala. Deep Photo Style Transfer. *CoRR*, abs/1703.07511, 2017.

[25] J. Mas, A. Fornes, and J. Llados. An Interactive Transcription System of Census Records Using Word-Spotting Based Information Transfer. *2016 12th IAPR Workshop on Document Analysis Systems (DAS)*, 00:54–59, April 2016.

[26] Carlos AB Mello and Rafael D Lins. Generation of images of historical documents by composition. *Proceedings of the 2002 ACM symposium on Document engineering*, pages 127–133, 2002.

[27] Mehdi Mirza and Simon Osindero. Conditional Generative Adversarial Nets. *CoRR*, abs/1411.1784, 2014.

[28] T.M. Mitchell. *Machine Learning*. McGraw-Hill International Editions. McGraw-Hill, 1997.

[29] Sinno Jialin Pan, Qiang Yang, Wei Fan, and Sinno Jialin Pan (ph. D. A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*, 2010.

[30] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *CoRR*, abs/1511.06434, 2015.

[31] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. *CoRR*, abs/1505.04597, 2015.

[32] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Josh Susskind, Wenda Wang, and Russell Webb. Learning from Simulated and Unsupervised Images through Adversarial Training. *CoRR*, abs/1612.07828, 2016.

[33] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR*, abs/1409.1556, 2014.

[34] Hao Wei, Kai Chen, Mathias Seuret, Marcel Würsch, Marcus Liwicki, and Rolf Ingold. DIVADIAWI – a Web-based Interface for Semi-automatic Labeling of Historical Document Images. *Digital Humanities 2015*, 2015.

[35] Xian Wu, Kun Xu, and Peter Hall. A survey of image synthesis and editing with generative adversarial networks. *Tsinghua Science and Technology*, 22(6):660–674, 12 2017.

[36] Marcel Würsch, Rolf Ingold, and Marcus Liwicki. DivaServicesA RESTful web service for Document Image Analysis methods. *Digital Scholarship in the Humanities*, 32(1):150–156, 2017.

[37] Zili Yi, Hao (Richard) Zhang, Ping Tan, and Minglun Gong. DualGAN: Unsupervised Dual Learning for Image-to-Image Translation. *ICCV*, pages 2868–2876, 2017.

[38] Hang Zhao, Orazio Gallo, Iuri Frosio, and Jan Kautz. Loss Functions for Neural Networks for Image Processing. *CoRR*, abs/1511.08861, 2015.

[39] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. *CoRR*, abs/1703.10593, 2017.