LUISS

Department
of Business & Management

Course of Artificial Intelligence and Machine Learning

Deepfake Generation: Evaluating State-of-the-Art Generative Networks

Prof. Marco Querini
SUPERVISOR

ID No. 232971
CANDIDATE

Academic Year 2020/2021

Contents

List of Figures

Abstract

Deepfakes are a new threat in online media information. Easy access to audio-visual content on social media, combined with the availability of modern tools and the rapid evolution of deep-learning methods, are crafting more common and convincing fake videos. This paper analyzes the problem of deepfake generation at scale, pointing out where the current state-of-the-art is. After a brief introduction on the two most common generative methods, VAEs and GANs. Most famous projects are based on complex architectures that are complicated for me to test as they require huge computational resources. Moreover, these architectures are quite bounded and do not permit clear exploitation and understanding of the specification. Thus, I propose a simplified version of the FaceSwap architecture and describe the improvements that StyleGAN has brought. Additionally, we also discuss open challenges and enumerate future directions to guide future researchers on issues that need to be considered to improve the domains of deepfake generation.

1 Introduction

Deepfakes are a new type of synthetic video, in which a subject's face is modified into a target face in order to mimic the target subject in a certain context and create authentic proofs of events that never occurred. They have the potential to significantly impact on how people determine the legitimacy of information presented online.
The quality of public discourse and the safeguarding of human rights may be influenced by such content generation and modification technologies. This is of particular importance, since deepfakes may be used by a malicious as a source of misinformation, manipulation, harassment, and persuasion. Generating and identifying manipulated media is a technically demanding and rapidly evolving challenge that requires collaborations across the entire tech industry and beyond.

This paper presents the problem of deepfakes at scale. Section 2 reviews the current literature analyzing the steps towards the current state-of-the-art. Then, section 3 describes the deepfakes trying to give them a clear and intuitive definition. In section 4, I review different methods for generating deepfakes as well as their advantages and disadvantages. I discuss challenges, research trends and directions on deepfake generation as well as multimedia problems in Section 5.

Note that in this paper I'll refer just to video-crafted deepfakes, leaving the audio sector for further researches.

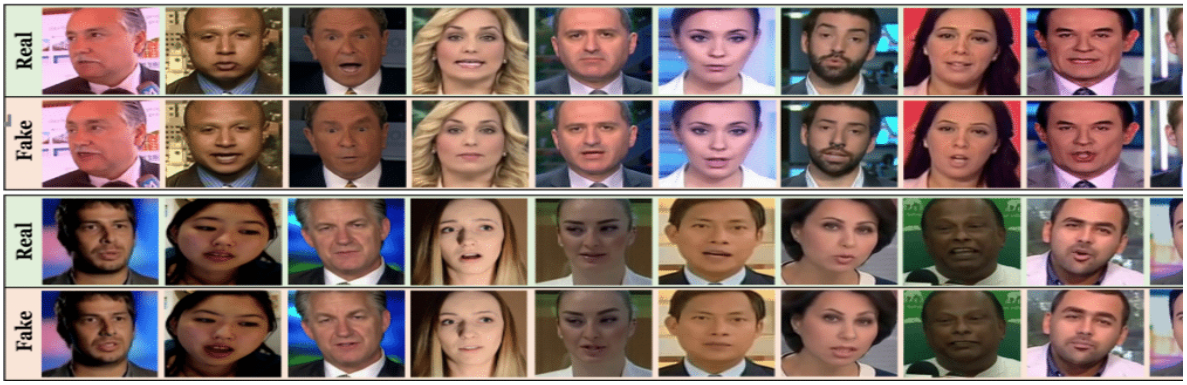

Figure 1: Deepfake generated images using a Generative Adversarial Network

2 Literature Review

In recent years, fake news has become an issue that is changing significantly the human society. In particular, deep fakes are an artificial intelligence synthetized content able to deceive the public making hard to know what to trust or not.

Academic literature started to focus on how to generate reliable contents which could have mimicked the daily contents to which we are constantly subjected. Then, thanks to the excellent result obtained in crafting new contents, a new track of existing studies concentrates on developing machine learning-based classifiers to automatically demonstrate whether a content spreading in a social media is fake or not.

2.1 Deepfake Generation

The two main schools of thoughts when referring to multimedia content creation are based on deep learning architectures such as Variational Autoencoder (VAEs) [25] or GANs [4]

The idea of using a Variational Autoencoder, initially proposed by Kingma and Welling in 2013 [12], reached a large consensus since it mitigated the regularization problem by adding a stochastic lower-dimensional space. For example RSGAN [17] stacked two different VAEs to handle separately hair

and face in the latent spaces, and then, face swapping is achieved by replacing the latent-space representations of the faces, and reconstruct the entire face image with them. LumièreNet [11] in 2019 proposed a framework based on VAEs to synthesize full pose video to learn mapping functions from the audio to video through estimated pose-based compact and abstract latent codes. The most famous VAEs' based architecture is the FaceSwap project [2]. The developers created this model based on the encoder-decoder architecture: the encoder is used to extract latent features of a face and then the decoder reconstruct them. In the period when faceswapping was invented it was considered something at the forefront. People outside the academic world where afraid of approaching to code and here the FaceSwap's intuition arises. It became the first code that everyone, also non in the academic sector, could download and run on their own. It's true that the final scopes of many developers using these architectures was not always ethical and it has sometimes been abused.

Over the time, new generative methods appeared and GANs have begun to carve out a large part of the developement patterns. Introduced in 2014 by Goodfellow et al.[4] , they have been described by Yann LeCun as "the most interesting idea in the last 10 years in Machine Learning". They keep many points in common with VAEs but enables the generation of fairly realistic synthetic images by forcing the generated images to be statistically almost indistinguishable from real ones. A GAN is composed by a generator and a discriminator: the former is continuously trained to fool the discriminator network. At the beginning the model will provide very bad results but its accuracy increases up to create an indistinguishable artificial image. Famous architecture like FaceSwap have been adapted also with a GAN version [2] and the current state-of-the-art is mainly focused on developing models GAN-centered. This model for example proposed an improved deepfake using GANs which adds to the autoencoder architecture adversarial and perpetual losses to VGGFace . The VGGFace refers to a series of models developed for face recognition which earned many atten-

tions from the academic panorama.

Vougioukas et al. [26] proposed a temporal GAN approach with 2 discriminators and to synthesize videos frame-by-frame, they used a RNN-based generator. The video generated included facial expressions like frowns and blinks. Moreover it included precise lips movement and natural expressions.

The path to the glory for GANs was not so easy as expected. The first models generated low-resolution images and so, not so convincing by the final users. DCGAN [22] was the first architecture to replace a dense layer with a deconvolution layer in the generator. This provided a better image quality and mostly implemented new tweaks on which one could base future improvements. Another issue was related to memory constraints, and it was partially mitigated by Karras et al. with the ProGAN [10] architecture. There are some similarities with DCGAN but the latter used transpose convolutions to change the representation size. In constrast, ProGAN uses nearest neighbors for upscaling and average pooling for downscaling. ProGAN proposed an adaptive mini-batch size approach that constantly increased the resolution increasing the model complexity with new layers. The State-of-the-art was finally found with StyleGAN [8] : a new method proposed by NVIDIA's engineers and Karras team. It rethinks GANs generator architecture in a way that proposed novel ways to manipulate the image synthesis process, leaving unchanged the discriminator. It easily separates the high-level attributes of an image, such as the pose and identity, stochastic variations in generated images such as the face color, frickles, hair, and beards. The network is so allowed to perform scale-specific mixing and interpolation operations. The advantage is that the intermediate latent is free from any certain distribution restriction, and this reduces the correlation among features. StyleGAN 2 [9] improved the parent model by removing undesired parts of the generated image, such as changes in gaze direction and teeth alignment.

In the last period, new architectures tried to propose methods for editing

facial attributes, like skin color or gender, by simply adding or removing facial expressions or everyday objects (glasses) from the source image. Perarnau et al. [20] introduced an Invertible Conditional GAN (IcGAN) which matches an encoder to a cGAN. The encoder works as usual by mapping the source image into latent representation, and the cGAN reconstructs the image by using attribute vectors previously defined.
Other approaches exist but they do not enjoy the same reputation as the above described generative methods. For example, Korshunova et al. [14] implemented a convolution neural network (CNN) to extract semantic contents and use them to create the same style in another image. The loss function they proposed was a mix of the different environmental aspects (lights and style). The main problems addressed by Nirkin et al. [19] with a full convolution network (FCN) and a 3D morphable model (3DMM), were related to the fact that CNN were able to transform only one image at a time and the large dataset required for the training phase.

2.2 Deepfake Detection

As previously explained, thanks to the excellent result obtained in crafting new contents, new detection algorithms come to the rescue. Algorithms are based either on handcrafted feature extractions or deep learning-based methods. A wide choice of literature covers both these techniques. Starting with the first category, Zhang et al. [24] proposed a raw method called SURF to detect swapped faces. Further updates were performed in order to solve the two main problems: the impossibility of keeping facial expressions and the issues in video detection. Agarwal et al. [1] initiated to worry about the harm that deep fakes could bring to politics. They proposed a method for tracking facial and head movements and then extracting the presence and strength of specific action units. This idea come to mind following an hypothesis that people while speaking, have distinct (but probably not unique) facial expressions and movements. These differences then, have

been used to train the binary SVM to classify between an original and a fake face of former US President Obama. Guera et al. [6] defined an useful detection technique based on handcrafted feature to classify fake faces from video. As the previous model, they used a SVM followed by a random forest classifier to differentiate the two products. Deep fakes detection algorithms based on deep learning mitigate two important problems: they can grasp frame-to-frame temporal variations in video. Moreover, they can adapt better to the loss of information made by compression techniques.
In the sector of Face Swap and Face Reenactment, the first algorithms were proposed by Li et al. [16]. After some libraries extracted facial landmarks, several Convolution Neural Network (CNN) based models were trained to discover forged contents. Problems arises with multi-time compressed videos. To overcome these difficulties, new solutions based on the combination of Convolution Neural Network (CNN) and Recurrent Neural Network (RNN) arise. For example, Guera et al. [5] proposed a solution of a CNN to extract relevant features and the RNN to detect deep fakes. Problems with this solution occurred with videos of medium/long duration. Also the approach by Li et al. [15] has some limitations since it was based on a clue on eye blinking. In particular, what happen if there are no aberrations with the eye blinking or even they just don't exist? In the following years, several tens of academic papers tried to address this problem in a very elegant way, each making their own contribution and trying to fix the limitations of the previous model.
Final steps toward the current state-of-the-art have been made by Nguyen et al. [18] which proposed a multi-task CNN network to concurrently detect and localize fake video contents. The architecture was composed by an autoencoder for the first part and by a y-shaped decoder for the second one. Here a long chain of teoretical problems occurred: they noticed that the accuracy degrades over unseen scenarios. To fix this concept, a Forensic Transfer (FT) technique was introduced, in turns leading to an excessive computation power since large latent spaces required.

To standardize the detection process new datasets for fake visual contents have been prepared. The widely accepted are the following: UADFV [27], FaceForensic++ [23] and DeepFakeTIMIT [13].

3 Deepfakes and Disinformation

Deepfakes are synthesized,AI-generated,videos and audio through which a face can be swapped with someone else's using neural networks. It is possible to produce deepfakes in a significantly more efficient and economical manner thanks to machine learning. They are so-named because they use deep learning technology, a branch of machine learning that applies neural net simulation to massive data sets, to create a fake. Despite in the past years deepfakes were visibly doctored, advances in technology have made it harder to tell what is real and what is fake. As with many technologies, deepfakes have endured a maturity curve on the way to realizing their full potential. As algorithms improve, less source data is needed to train a more accurate deepfake.

In the present day, it is uncommon to visit social media and not stumble upon some form of edited content, be it a simple selfie with a filter, a highly embellished meme or a video edited to add a soundtrack or enhance certain elements. According to a survey the largest share of deepfake contents are in the Entertainment sector. Further studies concluded that the politics sector suffers this plague.

Deepfakes require a large amount of data to craft new contents and since public figures such as celebrities and politicians may have a large number of videos and images available online, this makes them the initial targets of deepfakes.

The first case of a traditional deepfake occurred in 1860 when the face of southern politician John Calhoun was capably modified and replaced with the head of US President Abraham Lincoln. There is evidence of deepfakes being generated from the first years of 2000 where they've been used to swap faces of celebrities or politicians to bodies in porn images and videos. There are also cases where the deepfakes can be used for positive scenarios like creating voices of those who have lost theirs or update multimedia contents without the need of reconstruct them.

Different kind of deepfakes exists: i) face-swap, ii) lip-synching iii) puppet

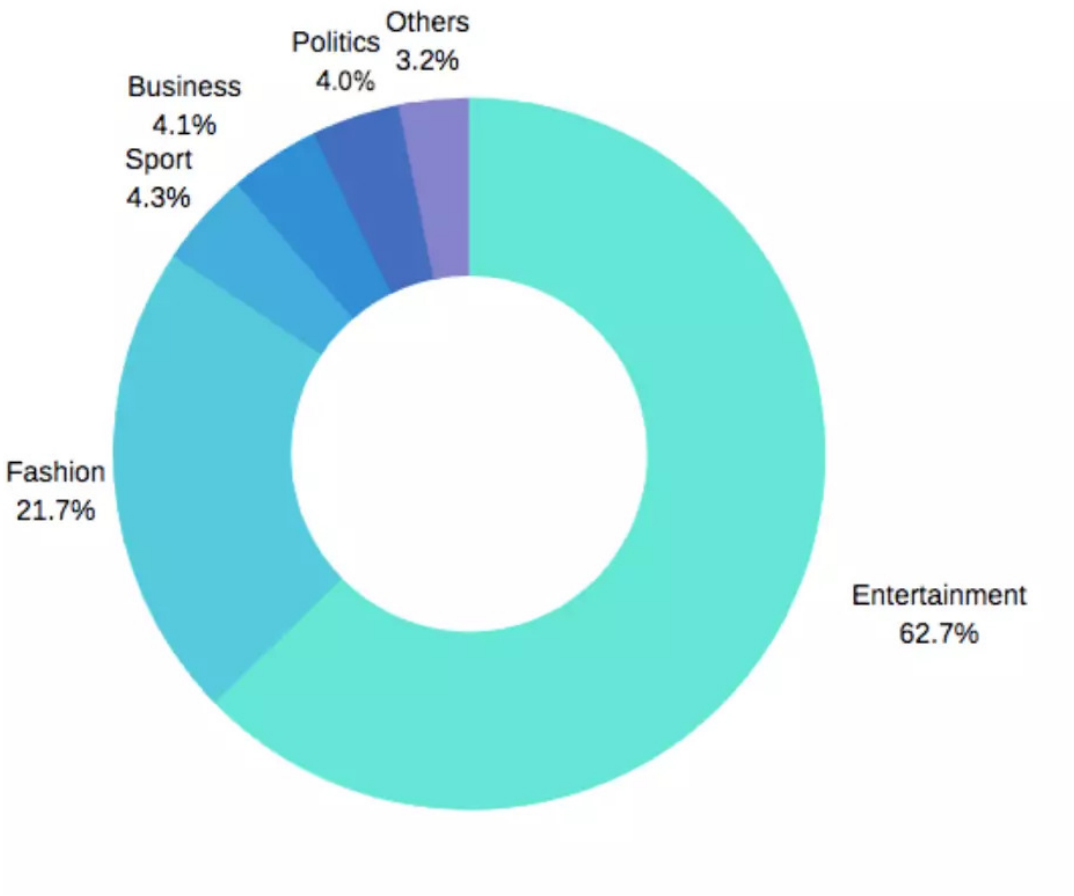

Figure 2: Distribution of deepfake targets as of June 2020 from a total of 49,081 detected videos

master and iv) face synthesis.

In face-swaps ,the face of the source person is interchanged with the one of a target person to generate a fake video, trying to portray actions which in reality have been done by the source person. This kind of deepfakes are usually generated to hit the popularity or reputation of famous personalities by showing them in a fake scheme in which they never appeared.

Lip-sync deepfakes refer to videos that are modified to make the mouth movements combining with an audio recording. Instead, Puppet-master deepfakes include videos of a target person (puppet) who is animated following the facial expressions, eye and head movements of another person (master) sitting in front of a camera. Puppet-master aim to hijack the source person's full-body in a video, and to animate it according to the

own's desire. The final type of deepfake is Face synthesis and involves the generation of photo-realistic face images and facial attribute editing. In comparison with face-swapping, facial reenactment has received more attention in the academic research literature. For instance, the Face2Face system enables real-time facial puppeteering by taking an input video of an actor's face and transferring the mouth shape and expressions onto a synthesized target face. Despite not including a voice track, output videos drew attention and triggered initial concerns over the misuse of the technology by demonstrating examples of an actor controlling the faces of Donald Trump and Vladimir Putin.

	Description
Face Swap	Source and target person interchange their faces
Lip Synching	Mouth movements combining with audio recording
Puppet Master	Video animated following facial expressions
Face Synthesis	Generation of photo-realistic face images
Voice Cloning	Generation of target speaker's voice

Table 1: Definition of the different types of deepfakes

3.1 Social Impact

Over the last years, only a few studies have examined the social impact of deepfakes. There have been dozens of studies looking at social influence from altered still images but the psychological processes and consequences of viewing deepfakes-modified video remain largely unstudied.
The core of deepfake is related to lying, which involves intentionally, knowingly, and/or purposely misleading another person.

The untruth detection research suggests that people are not surprisingly good at detecting deception and can relatively easily acquire false beliefs. In general false information spreads at higher velocity than truth.

Importantly, this level of spread is not affected by the medium in which the message is conveyed. It's the same whether the message is distributed

through text, an audio recording, or a video. Although this may seem surprising this happens because there are no reliable signals to human deception and we tend to trust what others say.

Most of the academic researches focus on how a person speak versus its body movements. Deepfakes make a shift in this sector since they not only make the verbal content change, but they also modify the visual properties of how the message was conveyed, whether this includes the movement of a person's mouth saying something that he or she actually did not, or the behavior of a person doing something that in reality has not been done.

People are growing with the awareness that these video cannot be modified or altered in any way. But, what happens when people realizes that this belief is no more valid?

The philosopher Don Fallis analyzed this problem and tried to address this issue in a theoretical and psychological way. He called it the epistemic threat of deepfakes [3]. Because of the dominance of the visual system, videos have high information carrying potential: we tend to believe what we see in a video, making the video as a "gold standard" of truth. But as deepfakes proliferate and awareness that videos can be faked spread through the population, the amount of information that videos carry to viewers is diminished.

4 Deepfake Generation

With the new availability in data and machine learning algorithms, new ways of generating deepfakes are arising. These application are mostly developed used deep learning techniques, in order to exploit their ability to handle high dimensional data.
The two main approaches rely on Variational Autoencoders (VAE) and Generative Adversarial Networks (GAN). In this section we are going to describe both the methods and one example for both.

4.1 Variational Autoencoders

Before describing VAE, a brief introduction on autoencoders and dimensionality reduction is required. For several reasons there is the the need to reduce the number of features of a specific dataset. It is more complicated to visualize the training set and then work on it if the number of features is large. Often, the majority of these features may proved to be tied together, and therefore superfluous. It is at this point that dimensionality reduction algorithm become relevant.
Autoencoders are a solution to address this problem. They are an unsupervised learning technique for the task of representation learning.[1]
It's composed by an encoder which is the process of recreating new features representation from the old features, and a decoder which works in the opposite direction. Autoencoders can help to mitigate the problem of dimensionality reduction since the encoder compress the relevant data from the latent space, and the decoder reconstruct them trying to avoid loss of information. The goal is to find the best encoder/decoder pair among a given family.

In particular, given two functions ϕ (e(x) - encoder) and ψ (d(e(x)) -

[1]It's a method of finding a representation of the data – the features, the distance function, the similarity function– that dictates how the predictive model will perform.

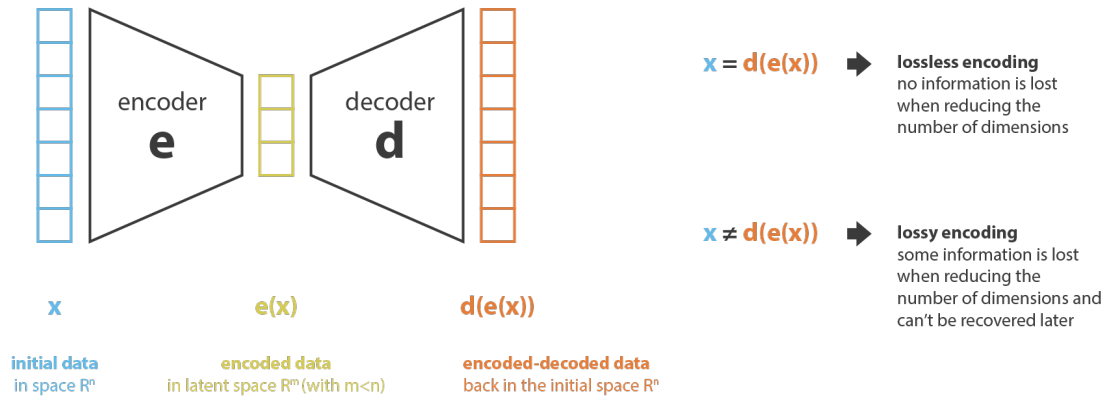

Figure 3: Autoencoders and dimensionality reduction. Source: Medium

decoder), $\phi : \chi \mapsto F$ and $\psi : F \mapsto \chi$

$$\phi, \psi = \underset{\phi,\psi}{\operatorname{argmin}}\, \epsilon \, \|\chi, \psi\| \tag{1}$$

The encoder function, denoted by ϕ, maps the original data $\chi \in R^n$, to a latent space $F \in R^m$, which is present at the bottleneck. The decoder function, denoted by ψ, maps the latent space F at the bottleneck to the output which in this case, is the same as the input function. Therefore, we are simply attempting to recreate the original image following some generalized non-linear compression.

The argument between the argmin function defines the reconstruction error measure between the input data χ and the encoded-decoded data ψ.

Autoencoders are then implemented using neural networks and consist in delineate an encoder and a decoder and to learn the best encoding-decoding scheme using an iterative optimisation process. Each iteration takes as input some data, it process them and then compare the autoencoder generated data with the initial and backpropagate the error through the architecture to update the weights of the networks: this twerk that minimise the reconstruction error is done by gradient descent over the parameters of these networks. Some problems can arise when dealing with autoencoders:

in general an autoencoder learns to capture as much information as possible rather than as much relevant information as possible. Moreover, to train an autoencoder there is need of lots of data, processing time, hyperparameter tuning, and model validation before even start building the real model with the high risk of information loss during the reconstruction phase.
Going back to our generation problem, once the autoencoder has been trained, we have an architecture but still no real way to craft any new content. One approach could be the one of sampling a point from the latent space and decode it in order to obtain a new instance. This can be done only if the latent space is enough regularized and it depends on several factors like the distribution of the data itself. Also if there will be no information loss, the high degree of freedom of an autoencoder would lead to a model overfitting, implying that some points of the latent space will give meaningless content once decoded.
To be sure that autoencoder could help us in generation of new contents we must make sure that the latent space is regular enough, and we can do this by encoding an input as a distribution over the latent space. Here the idea of variational autoencoders. By sampling from the latent space, we can use the decoder network to form a generative model capable of creating new data similar to what was observed during training.

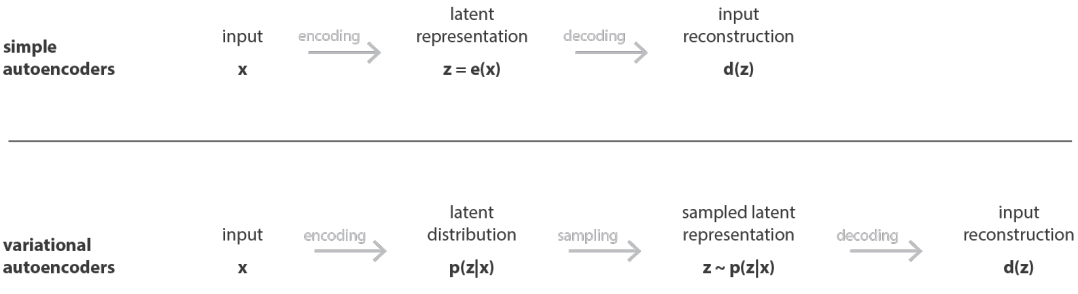

Figure 4: Variational Autoencoders. Source: Medium

Thus, the loss function that is minimised when training a VAE is composed of a term which penalizes reconstruction error, that iterates over

and over to make the encoding-decoding scheme as performant as possible, and a "regularisation term", that tends to regularise the organisation of the latent space by making the distributions returned by the encoder close to a standard normal distribution. The general regularisation term is the Kulback-Leibler [2] divergence between the returned distribution and a standard Gaussian and can be directly expressed in terms of the means and the covariance matrices of the the two distributions.

4.2 Application

Many existing real world solutions help to generate a deepfake in a fashion and reliable way. For example, tools like Faceswap [2] or DeepFaceLab [21] combines powerful architectures with easy GUIs, making the generation process available to everyone.
Problems arise since the required computational power for training exceed the common standard of developers and most of them are oblished to rely on expensive external GPUs. Moreover, the presence of a GUI makes the whole training process bounded to some fixed constraints thus leading to a bad in-depth analysis.
I'm going to propose a simplified version of FaceSwap, tested on a Google Colab Pro environment with the following characteristics.

After an initial preprocessing phase where I focus mainly on making all the facial regions of the same size, I defined the model architecture.

```
def Encoder(self):
        input_ = Input(shape=IMAGE_SHAPE)
        x = input_
        x = self.conv(128)(x)
        x = self.conv(256)(x)
        x = self.conv(512)(x)
        x = self.conv(1024)(x)
        x = Dense(1024)(Flatten()(x))
```

[2]Kullback-Leibler divergence is an information-based measure of disparity among probability distributions.

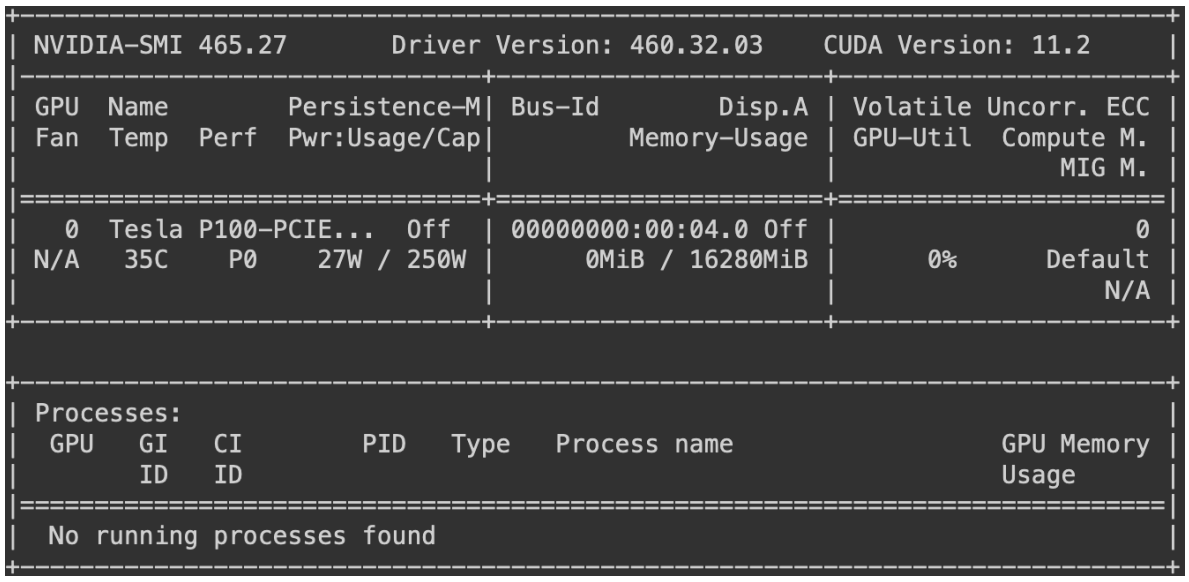

Figure 5: Google Colab Pro

```
        x = Dense(4 * 4 * 1024)(x)
        x = Reshape((4, 4, 1024))(x)
        x = self.upscale(512)(x)
        return KerasModel(input_, x)
```

Listing 1: Model Encoder

```
def Decoder(self):
        input_ = Input(shape=(8, 8, 512))
        x = input_
        x = self.upscale(256)(x)
        x = self.upscale(128)(x)
        x = self.upscale(64)(x)
        x = Conv2D(3, kernel_size=5, padding='same', activation='
    sigmoid')(x)
        return KerasModel(input_, x)
```

Listing 2: Model Decoder

A pair of autoencoders is trained on the faces of the training actors dataset. Since my computational resources were quite limited, I decided to load the weights directly from the official FaceSwap repo. This would make my model able to converge faster.
As a loss function, I opted for mean absolute error, which covers pixel-to-pixel differences between the original inputs and generated outputs.

Please notice that both the encoder and decoder contains a conv and a upscale function, as defined:

```
def conv(self, filters):
            x = Conv2D(filters, kernel_size=5, strides=2, padding='
   same')(x)
            x = LeakyReLU(0.1)(x)
            return x

def upscale(self, filters):
            x = Conv2D(filters * 4, kernel_size=3, padding='same')(x)
            x = LeakyReLU(0.1)(x)
            x = PixelShuffler()(x)
            return x
```

Listing 3: Upscale and Convolution

Upscale need to double the dimensions of the input (thanks to PixelShuffler) and the Conv function performs an inverse convolution operation. These are commonly used techniques by the state-of-the-art algorithms.

The encoder is composed by a series of convolutional layers + LeakyRelu activation layers. Note that each layer doubles in parameters, generating smaller activation maps, with a stride value of 2 reducing the map size by half after each layer. Following in the architecture, a dense layer gets a flattened vector composed by the reshaped activation map of the convolution layers. Reshape and upscale conclude the whole network.
The decoder's role is more simple, since mock the encoder's behaviour in the opposite direction. It upscales the representation of an encoded input back into an acceptable 64 x 64 output through the use of convolutional layers.

Finally, I use the trained weights of the autoencoder and facial transfer to generate a deepfake, by connecting the encoded representation of source A with the decoder of target B, resulting in the generation of an image of target B but with the facial characteristics of A.

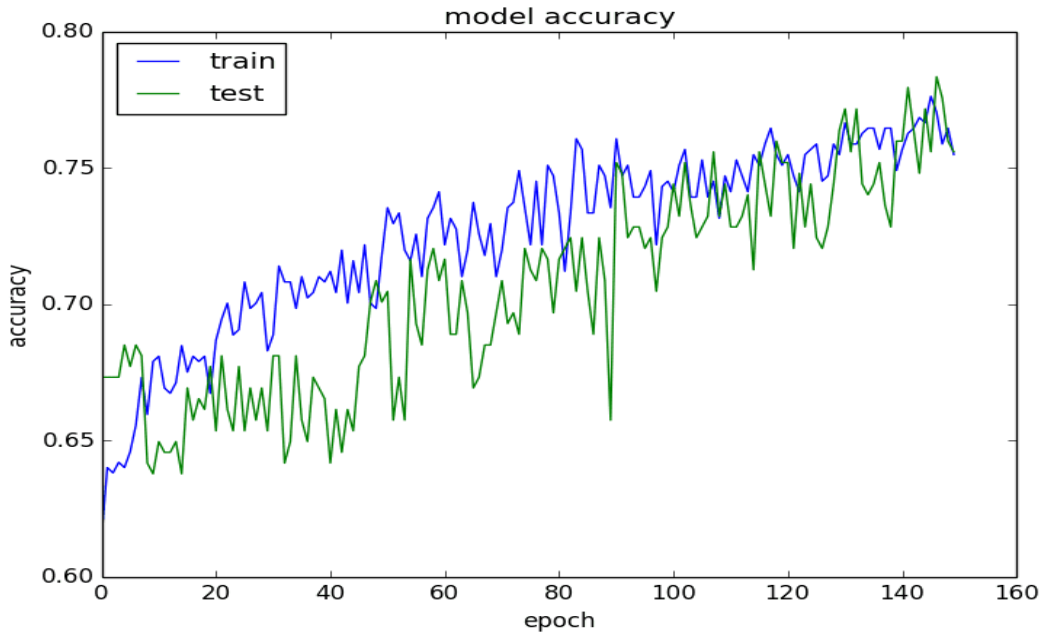

Figure 6: Accuracy

Despite the standard weaknesses related to the computing power, this architecture obtained an overall good level of performance. Some improvements include the introduction of a discriminative component as happen for the GANs. Moreover, a large number of data and better resources can lead to more test and thus to better results.

4.3 Generative Adversarial Networks

Generative adversarial networks are composed by an overall structure of two neural networks, the generator and the discriminator.

Generated samples resembling real data are supplied by the generator, which estimates the probability distribution of the real samples in order to do so. On the other hand, the discriminator estimates that the real samples is not being provided by the generator, but rather comes from the real data.

At the base of GANs lies the concept that a data generator's high

quality cannot be determined if we cannot tell fake data apart from real data

In statistics, this is called a two-sample test [3]. This makes the network able to improve the data generator until it generates something that resembles the real data and in the end fooling the classifier. The same is true if our classifier is a state-of-the-art deep neural network.

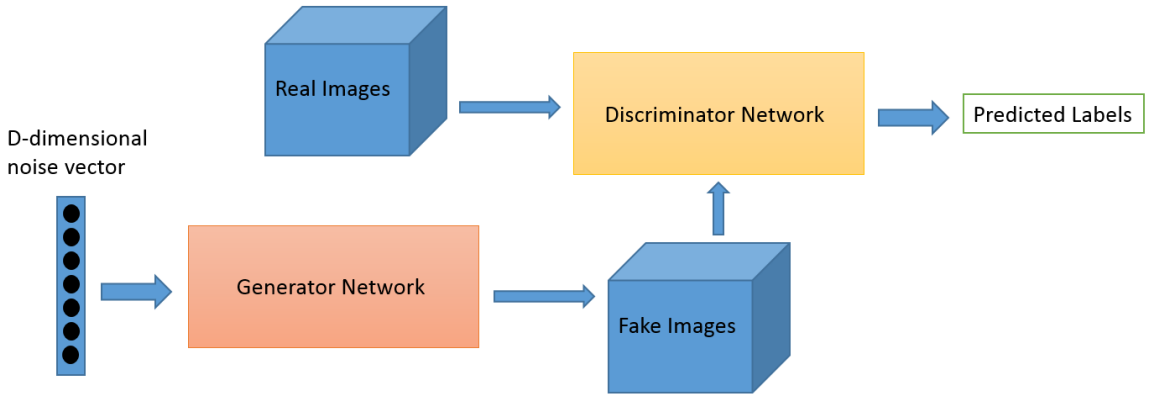

Figure 7: GAN. Source: O'Reilly

The discriminator is a binary classifier to distinguish if the input x is real (from real data) or fake (from the generator). It returns a scalar prediction $D(x)$. Assume the label y for the true data is 1 and 0 for the fake data. We train the discriminator to minimize the cross-entropy loss [4], in particular:

$$\min_D[-ylogD(x) - (1-y)log(1-D(x))] \tag{2}$$

For the generator, it first draws some random parameter z from a normal distribution $z \approx N(0,1)$. Then the function $G(z)$ is used to generate $x^{'}$. The goal of the generator is to fool the discriminator to classify $x^{'}$ as

[3]The two-sample t-test (also known as the independent samples t-test) is a method used to test whether the unknown population means of two groups are equal or not.

[4]Cross-entropy is a measure of the difference between two probability distributions for a given random variable or set of events. It measures the performance of a classification model whose output is a probability value between 0 and 1.

true data.
If the generator works well, then $D(x^{'}) \approx 1$, which results in the gradients are too small to make a good progress for the discriminator.
In this way both the generator and discriminator play a minimax game which has a objective this objective function:

$$\min_{D} \max_{G} [-E_{x \approx Data} log D(x^{'}) - E_{z \approx Noise} log(1 - D(G(z)))] \tag{3}$$

Once both objective functions are defined, they are learned jointly by the alternating gradient descent. The generator model's parameters are revised and perform a single iteration of gradient descent on the discriminator using the real and the generated images. Then an exchange of the architectures occurs. Fix the discriminator and train the generator for another single iteration. This train lasts until the generator produces good quality images.

4.4 StyleGAN

The most famous GAN's applications are SyleGAN and CycleGAN [28] .
The latter represents a technique that involves the automatic training of image-to-image translation models without paired examples. Its architecture is different from other GANs in a way that it contains 2 mapping function (G and F) that acts as generators and their corresponding Discriminators (Dx and Dy).
I'd like to focus more on StyleGAN and how it's applied for deepfake generation. The StyleGAN generator no longer takes a feature from the potential range as input; instead, it uses two new references of randomness to produce a synthetic image: standalone mapping channels and noise layers.

The result is capable not only of generating photorealistic high-quality photos of faces, but also offers control over the style of the generated image at different levels of detail through varying the style vectors and noise.

Figure 8: This Person Does Not Exist

The SyleGAN architecture exploit the false past belief that improvements at the network had to be done by just improving the discriminator. As such, the generator has been somewhat neglected and remained apart.

According to the original paper proposed by Karras et al. [9], the StyleGAN is described as a progressive growing GAN architecture with five modifications, each of which was added and evaluated incrementally in an ablative study.
In particular:

- StyleGAN uses baseline progressive GAN structure, which means the volume of the generated picture increases progressively from a shallow resolution (4×4) to high resolution (1024 x1024).
- The progressive growing GAN uses nearest neighbor layers for up-sampling instead of transpose convolutional layers that are common

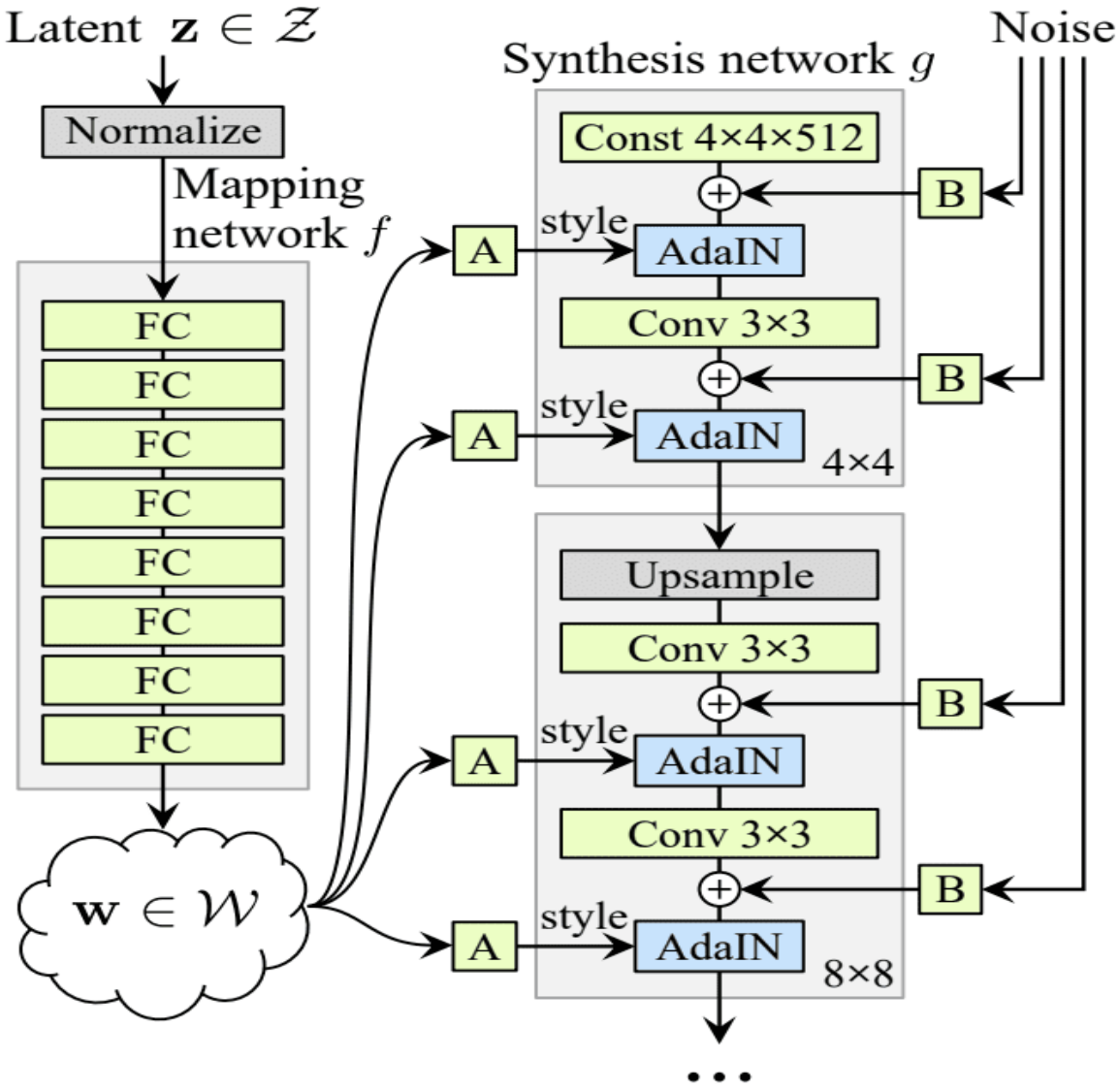

Figure 9: Overview of the SyleGAN generator

in other generator models.

- A standalone mapping network takes a random point from the latent space as input and generates a style vector.
- Subsequently it modifies the generator's model so that a point is no longer take as input from the latent space. In order to begin the image synthesis process, the model now has a constant 4x4x512 constant value input.

- Finally, a Gaussian noise is attached to each activation map before the AdaIN [7] method. A separate sample of noise for each block is evaluated based on the scaling factors of that layer.

The reason why traditional GANs have a problem with control of styles or features within the same image, and thus the reason that made StyleGAN so famous, is due to something called feature entanglement. A GAN is not as capable of distinguishing these finer details as a human, thus leading the features to become "entangled" with each other to some extent within the GAN's frame of perception.

StyleGAN is easily the most powerful GAN in existence. With the ability to generate synthesized images from scratch in high resolution and combining the homemade hardwares (NVIDIA), they are considered as the state-of-the-art in Generative networks.

5 Conclusions and Future Research

I analyzed the current state-of-the-art but it's clear that there is still, however, room for improvement.
Future research can include working with a larger dataset and better computational resources. The generative models are data-driven, and therefore they reflect the learned features during training in the output. To generate high-quality deepfakes a large amount of data is required for training. Moreover, I think the academic research is moving toward a specific direction, and in particular the identity leake issue. The preservation of target identity is a problem when there is a significant mismatch between the two subjects of the analysis. Finally, additional improvements in environental condition can help to achieve better results: most of the current synthetic media generation focuses on a frontal face pose. In facial reenactment,for good results the face is swapped with a lookalike identity.However,it is not possible to always have the best match, which ultimately results in identity leakage.
Present deepfake generation focuses on the face region only, however the next generation of deepfakes is expected to target full body manipulations,such as a change in body pose,along with convincing expressions. Enviromental changes like illumination or surrounding scenes can result in strange artifacts in the final result. Moreover, the presence of wobble between different frames makes impossible to distinguish a temporal consistency over the whole generation process.
Online media are view under a new eye in these years because of their potential positive and negative impact on our society. The trade-off between deepfake generation and detection will not end in the foreseeable future,although impressive work has been presented for the generation and detection of these technologies. My future work is to repeat the video generation process using a more homogeneous set of images generated by the GAN and then to compare the quality of the generated deepfakes.

6 References

[1] Shruti Agarwal et al. "Protecting World Leaders Against Deep Fakes". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. June 2019.

[2] *FaceSwap Project*. `https://github.com/deepfakes/faceswap`.

[3] Don Fallis. *The Epistemic Threat of Deepfakes*. 2020. eprint: `1812.08685` (cs.CV).

[4] Ian J. Goodfellow et al. "Generative Adversarial Networks". In: (2014). arXiv: `1406.2661 [stat.ML]`.

[5] David Guera and E. Delp. "Deepfake Video Detection Using Recurrent Neural Networks". In: *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)* (2018), pp. 1–6.

[6] David Güera et al. *We Need No Pixels: Video Manipulation Detection Using Stream Descriptors*. 2019. arXiv: `1906.08743 [cs.LG]`.

[7] Xun Huang and Serge Belongie. *Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization*. 2017. arXiv: `1703.06868 [cs.CV]`.

[8] Tero Karras, Samuli Laine, and Timo Aila. *A Style-Based Generator Architecture for Generative Adversarial Networks*. 2019. arXiv: `1812.04948 [cs.NE]`.

[9] Tero Karras et al. *Analyzing and Improving the Image Quality of StyleGAN*. 2020. arXiv: `1912.04958 [cs.CV]`.

[10] Tero Karras et al. *Progressive Growing of GANs for Improved Quality, Stability, and Variation*. 2018. arXiv: `1710.10196 [cs.NE]`.

[11] Byung-Hak Kim and Varun Ganapathi. *LumièreNet: Lecture Video Synthesis from Audio*. 2019. arXiv: `1907.02253 [cs.LG]`.

[12] Diederik P Kingma and Max Welling. *Auto-Encoding Variational Bayes*. 2014. arXiv: 1312.6114 [stat.ML].

[13] Pavel Korshunov and Sebastien Marcel. *DeepFakes: a New Threat to Face Recognition? Assessment and Detection*. 2018. arXiv: 1812.08685 [cs.CV].

[14] Iryna Korshunova et al. "Fast Face-Swap Using Convolutional Neural Networks". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. Oct. 2017.

[15] Yuezun Li, Ming-Ching Chang, and Siwei Lyu. *In Ictu Oculi: Exposing AI Generated Fake Face Videos by Detecting Eye Blinking*. 2018. arXiv: 1806.02877 [cs.CV].

[16] Yuezun Li and Siwei Lyu. *Exposing DeepFake Videos By Detecting Face Warping Artifacts*. 2019. arXiv: 1811.00656 [cs.CV].

[17] Ryota Natsume, Tatsuya Yatagawa, and Shigeo Morishima. *RSGAN: Face Swapping and Editing using Face and Hair Representation in Latent Spaces*. 2018. arXiv: 1804.03447 [cs.CV].

[18] Huy H. Nguyen et al. *Multi-task Learning For Detecting and Segmenting Manipulated Facial Images and Videos*. 2019. arXiv: 1906.06876 [cs.CV].

[19] Yuval Nirkin, Yosi Keller, and Tal Hassner. *FSGAN: Subject Agnostic Face Swapping and Reenactment*. 2019. arXiv: 1908.05932 [cs.CV].

[20] Guim Perarnau et al. *Invertible Conditional GANs for image editing*. 2016. arXiv: 1611.06355 [cs.CV].

[21] Ivan Perov et al. *DeepFaceLab: A simple, flexible and extensible face swapping framework*. 2020. arXiv: 2005.05535 [cs.CV].

[22] Alec Radford, Luke Metz, and Soumith Chintala. *Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks*. 2016. arXiv: 1511.06434 [cs.LG].

[23] Andreas Rössler et al. *FaceForensics++: Learning to Detect Manipulated Facial Images*. 2019. arXiv: 1901.08971 [cs.CV].

[24] Brandon M. Smith and Li Zhang. "Joint Face Alignment with Non-Parametric Shape Models". In: *Proceedings of the 12th European Conference on Computer Vision - Volume Part III*. ECCV'12. Florence, Italy: Springer-Verlag, 2012, pp. 43–56. DOI: 10.1007/978-3-642-33712-3_4. URL: https://doi.org/10.1007/978-3-642-33712-3_4.

[25] Pascal Vincent et al. "Extracting and Composing Robust Features with Denoising Autoencoders". In: *Proceedings of the 25th International Conference on Machine Learning*. ICML '08. Helsinki, Finland: Association for Computing Machinery, 2008, pp. 1096–1103. ISBN: 9781605582054. DOI: 10.1145/1390156.1390294. URL: https://doi.org/10.1145/1390156.1390294.

[26] Konstantinos Vougioukas, Stavros Petridis, and Maja Pantic. "Realistic Speech-Driven Facial Animation with GANs". In: *International Journal of Computer Vision* 128 (May 2020). DOI: 10.1007/s11263-019-01251-8.

[27] Xin Yang, Yuezun Li, and Siwei Lyu. *Exposing Deep Fakes Using Inconsistent Head Poses*. 2018. arXiv: 1811.00661 [cs.CV].

[28] Jun-Yan Zhu et al. "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networkss". In: *Computer Vision (ICCV), 2017 IEEE International Conference on*. 2017.