UiA University of Agder

Master

Smooth and Consistent Video Anonymization Using Triangular Inpainting and Optical Flow

ELIAN ALTALAB, TEKLEMARIAM WELDEHAWARIAT

SUPERVISOR

Lei Jiao

Jivitesh Sharma

University of Agder, 2022

Faculty of Engineering and Science

Department of Engineering and Sciences

Acknowledgements

This thesis is a conclusion for our master's degree in Information and Communication Technology at the University of Agder, Norway.

We want to thank our supervisors Lei Jiao and Jivitesh Sharma for their continued guidance and feedback on both the academic and technical aspects of the thesis. We would also like to thank our families and friends for supporting us spiritually in writing this thesis.

Abstract

Surveillance cameras have been deployed extensively in big cities, such as London and Shanghai. To protect people's privacy and avoid fully exposed, it is necessary to remove sensitive facial information in the surveillance footage. In this thesis, we study the anonymization of CCTV footage with face inpainting. In more detail, we employ deep neural networks to generate faces and replace the original faces in the video. Particularly, a masking method called triangular inpainting is employed to produce videos where the original faces are removed. Furthermore, we adopted an object detection method Optical Flow to ensure the smooth movement and transition of the computer generated face when masked on the original face. The thesis also tries to keep the age and gender of the generated faces to the original subjects as close as possible. To ensure that each human visible in videos is masked with a unique face throughout the whole video, we index the original face and the inpainted face for a one-to-one mapping. The designed system has been tested via extensive experiments. The results show that the human subjects are anonymized efficiently. The inpainted faces can also maintain the uniqueness and the smoothness in the video with the age and gender preserved.

Keywords: Information security, Machine Learning, Neural Network, Convolutional Neural Network, Optical Flow, GAN, Object Detection, Inpainting

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Nowadays, the requirement for security is ever needed. The number of people is continuously increasing, and with the number of crimes out there, it is not easy to identify the criminals among all the people. The best solution is to use CCTV cameras to ensure that criminals are caught more easily. According to **IHS Markit's** latest report [2], the number of surveillance cameras worldwide has reached 770 million cameras and was expected to reach 1 billion cameras worldwide by the end of 2021 [2].

A significant challenge that has been a concern for many people is their privacy. If there is a surveillance camera on every corner, their privacy is surely at risk. It does not truly matter how much the government/owners of said surveillance cameras promise the safe use of the footage. The regular pedestrians captured by these cameras would still feel uneasy knowing that someone else can see what they do whenever they are outside the comfort of their homes.

To work around such privacy-intrusive devices and methods, researchers and engineers have worked for a long time to find ways to surpass said intrusive devices. These solutions come in the form of anonymisation, which removes private information from a set of data. In this context, removing all the data that makes a person recognisable from the CCTV footage. However, these solutions come with their problems as well. These problems occur depending on what form of anonymisation is used. The two most known forms of anonymisation are blurring subjects or de-identifying subjects. Blurring the subjects is a somewhat straightforward concept that requires the faces of all human subjects in videos or images to be blurred out no matter the blurring method. However, the de-identification method requires the changing of the subject's face. This requires calculating the landmarks of the face, finding a method to change the landmarks and hopefully changing how the face looks so it would not be recognisable.

The purpose of this thesis is to benefit from the work done by researchers on different Neural networks, specifically **CIAGAN** and **DeepPrivacy** to de-identify subjects appearing in videos, as well as using other methods to ensure the most consistent preservation and inpainting the newly generated faces. Moreover, the preservation of age and gender.

1.1 Motivation

This thesis focuses on combating the privacy problems accompanying CCTV and surveillance cameras. Today's number of CCTV cameras poses a significant challenge to the privacy of any human, as it strips people from their right to keep their data and identity as a private thing. Therefore, to cope with these challenges, the design of a private system is a must to ensure that the people's right to keep their identities as a private matter.

One of these systems designed to cope with CCTV's privacy challenges is the facial anonymisation system, which allows for the erasing or altering of faces appearing in CCTV footage. However, since some level of data is still needed to ensure that CCTV footage can still help in criminal cases, the erasing or blurring of the faces would not be the optimal choice. On the other hand, altering the appearance would be a more logical choice. Systems like that already exist with the main task of generating new faces and masking the old ones with the newly generated faces.

The problem that could occur when generating new faces and inpainting them on top of the original human subjects is preserving needed data. Some data is still needed when generating new faces to cover the original faces of subjects in the footage. The data in question are the age and gender of the original subject. After all, it would be weird when we generate a new face of a 10-year-old girl and use it to cover the face of a 70-year-old man that appears in the footage. Furthermore, another thing that needs to be preserved is how the newly generated face looks throughout each video frame. Generating new faces for a human subject results in a different face generated each time the program is run, resulting in the subject having a different face in every frame.

1.2 Goal

The thesis mainly aims at proposing a method that improves the de-identification of humans in videos. The goal is to use multiple neural networks to anonymise human subjects in videos by generating new faces. The process will be done using triangular inpainting to mask the original face with the newly generated ones. Moreover, ensuring the use of the same generated face throughout the video while attempting to preserve the original age and gender of the original humans. Finally, taking advantage of Optical Flow to maintain a smooth transition between the frames.

1.3 Field of Research

Multiple methods to anonymise humans in videos have been introduced before. Each method has some unique pros, cons, result quality, and accuracy. This thesis focuses on researching some of these methods as well as combining them with triangular inpainting and optical flow to find the most optimal way of producing the desired results defined in **Section 1.2**.

1.4 Hypothesis - Statement of the problem

- **Statement 1:** How to use a de-identification neural network to anonymise one or multiple subjects in videos.
- **Statement 2:** How to preserve the same generated face for a subject throughout the entirety of a video.
- **Statement 3:** How to preserve the age and gender of the original subjects when generating new faces.

1.5 Contributions

The main contributions of this thesis are:

- We propose a novel method combining the improved and optimised anonymisation of *CIAGAN*, *DeepPrivacy* and triangular inpainting in order to anonymise multiple faces in images and videos while closely preserving the correct skin tone, gender, and age.

- Using the proposed architecture, we present a method using *DeepFace* to collect the unique faces appearing in videos to ensure the anonymised faces are consistently masked on the correct people in videos.

- Using the same architecture, we present an *Optical Flow* solution that ensures the smooth inpainting of the faces when transitioning between frames in videos.

1.6 Thesis Structure

- **Chapter 2** (**Background and Related Work**): The chapter describes the main subjects researched by the team and how they relate to the thesis. That includes some of the papers researched to show a clear state of the art description of the currently available technology that is based on the above-mentioned subjects. The chapter describes four main research fields, **Face Detection (2.1)**, **Facial Anonymization (2.2)**, **Inpainting (2.3)**, and **Optical Flow (2.4)**. As mentioned, several state of the art papers/researched were studied and included such as **CIAGAN (2.2.1)**, **DeepPrivacy (2.2.2)**, **DeepFace (2.1.2)**, **FSGAN (2.3.1)**, **SC-FEGAN (2.3.2)**, and **Deep Feature Flow (2.4.1)**.

- **Chapter 3** (**Approach and Implementation**): The third chapter presents two elements, the approach **(3.1)** chosen to reach the goal, and the implementation **(3.2)** of said approach. The approach includes a description of the architecture for the solution, why it was designed like that and how it works.
  On the other hand, the implementation section describes how the architecture introduced in the approach was built in detail. That includes everything from the preprocessing stage until the final results stage.

- **Chapter 4** (**Evaluation and Discussion**): This chapter describes in detail the experiments that were focused on. What are the experiments, how are they conducted, and what are the desired results. The second part of the chapter shows the results. The results include numerical and visual results for the experiments described in the section. The final section of the chapter is where the work and results are discussed **(4.4)**, and limitations of the work are provided as well.

- **Chapter 5** (**Conclusion and Future Work**): The final chapter is divided into two parts. The chapter contains a conclusion **(5.1)** to summarise the work and the results of the thesis. Moreover, the work to be done in the future **(5.2)** is described, how the project can be improved and what other elements and subjects could be adopted into the project to increase the quality.

Chapter 2

Background and Related Work

The de-identification is generally a complex subject, and the de-identification of moving subjects in videos is an even more complex derivative of the de-identification. Therefore, many different smaller and bigger subjects are intertwined regarding anonymisation. Furthermore, since the thesis itself is a work that picks up from what the team achieved in the pre-project building up to this thesis, the number of topics intertwined is more prominent than usual as it is a collection of the current and previous topics.

2.1 Face Detection

Facial detection is a technology that uses simple or complex artificial intelligence to detect human faces in visual media [31], whether these media are images or videos. Facial detection works by searching through the pixels of an image to detect a set of patterns that the program has been taught to assume that they belong to a human. A picture contains many different objects, and the only way that the machine can understand the difference between these objects and a human face is by following the enforced patterns. To be more precise, these patterns are called landmarks or facial landmarks. These landmarks are mathematically calculated shapes that have been implemented into the machine or the program. When an image is fed to the program, the program starts to look for patterns that resemble these pre-coded shapes. It usually starts by finding the eyes one by one, then the mouth, the nose, the eyebrows, and the iris [10]. These landmarks are the prominent landmarks of a human face. However, depending on the algorithm, the program can be taught to look for more landmarks before assuming that an object in an image is a human face. These non-primary landmarks can be the jawline, ears, and many other points scattered over the human face. Some algorithms are programmed to find the five primary facial landmarks and then find all the facial land points between them before assuming that the object given is a human face.

Facial detection is used in many other technologies that either use facial detection or configure it to produce something more significant. An important example of such technologies is security technologies, where facial detection is combined with other methods to produce systems capable of using biometrics, facial prints, and face recognition. In addition, facial detection can be used to determine many aspects of facial analysis like age, gender, race, emotions, and more [10].

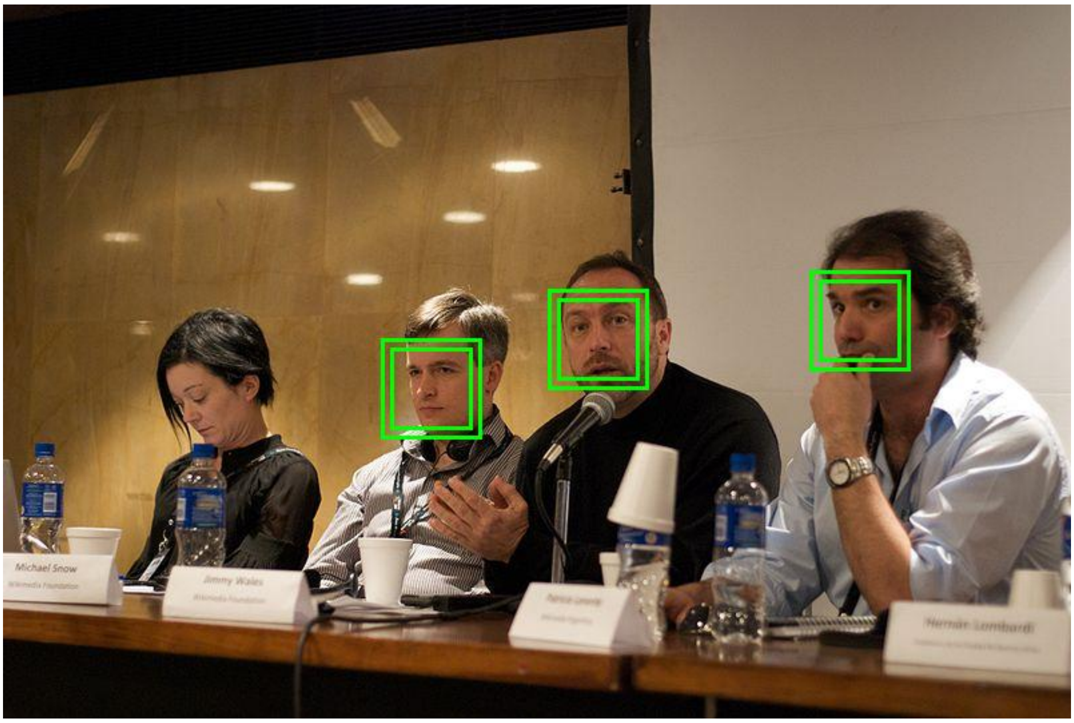


Figure 2.1: Automatic face detection with OpenCV [17].

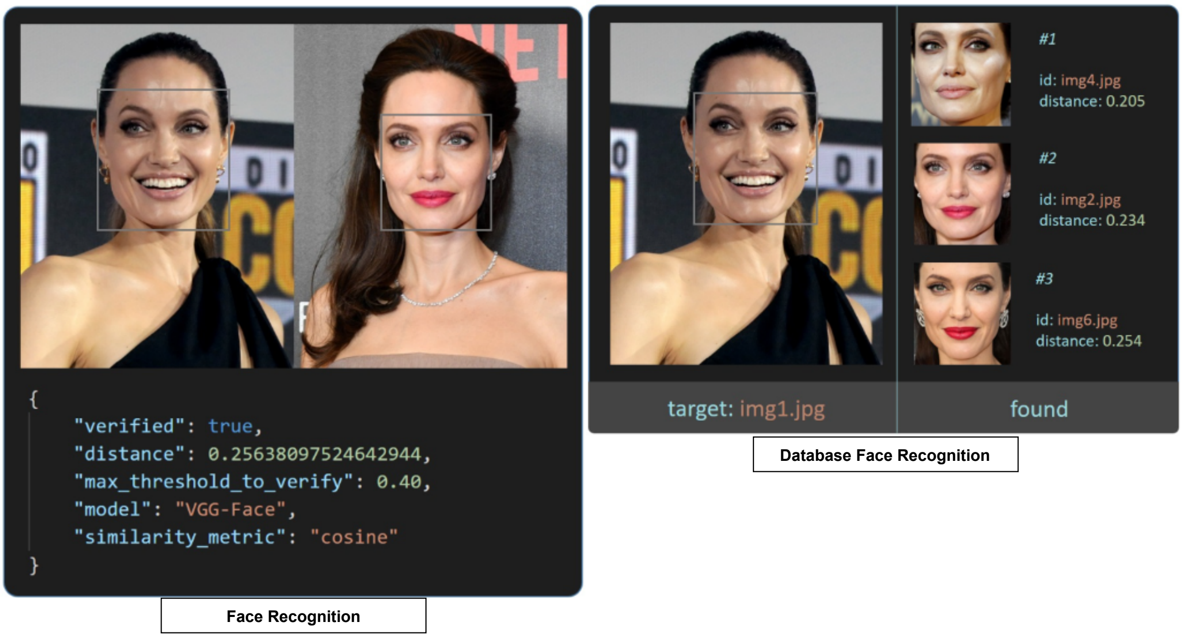
Figure 2.1 above shows how a non-complex face detection algorithm using the OpenCV library managed to detect all possible faces in an image using the five primary landmarks before drawing a rectangle surrounding these said landmarks. However, the woman on the far left part of the image was not detected because the algorithm needed clear access to the five primary landmarks before assuming that the subject was human.

2.1.1 Methods & Approaches of Facial Detection

According to a paper published in August 2018 by Ashu Kumar, Amandeep Kaur & Munish Kumar in Artificial Intelligence Review journal volume 52 [20], face detection is a generalisation of face localisation which aims to identify the location and size of faces in an image and that can be divided into two approaches, Feature-based approach and image-based approach [20].

2.1.2 DeepFace

A product of two research papers is a neural network called **DeepFace** which was first introduced in a paper published in October 2020 by **Sefik Ilkin Serengil; Alper Ozpinar** [30]. The paper introduces a lightweight facial face detection and recognition framework that utilises the power of TensorFlow and Keras to recognise human faces and align them. Furthermore, the framework allowed for the verification of subjects by comparison where two faces can be compared, and even with different poses, expressions and backgrounds, the framework could recognise if these two faces are of the same person [30]. At that stage, the framework was named Lightface.



```
{
    "verified": true,
    "distance": 0.25638097524642944,
    "max_threshold_to_verify": 0.40,
    "model": "VGG-Face",
    "similarity_metric": "cosine"
}
```

#1
id: img4.jpg
distance: 0.205

#2
id: img2.jpg
distance: 0.234

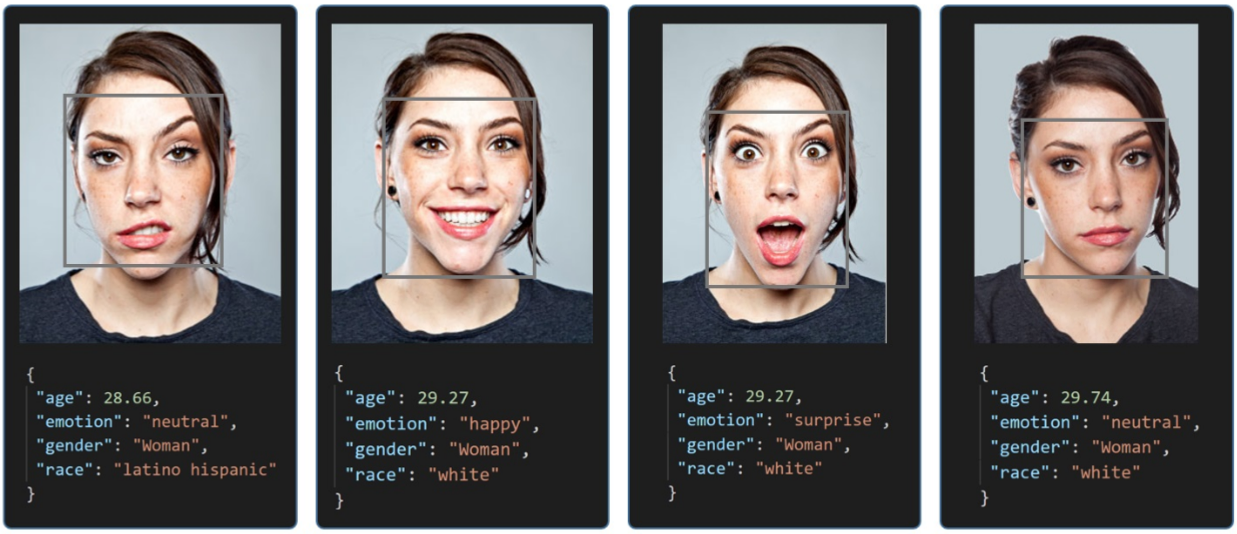
#3
id: img6.jpg
distance: 0.254

Figure 2.2: Facial recognition and verification using DeepFace [30].

Figure 2.2 above shows an example of how a picture of actress **Angelina Jolie** was verified using DeepFace against another of her images even though they have different poses, expressions and background. The image was also compared against a full database of images, where the DeepFace framework managed to detect all images of the same person.

A year later, in October 2021, another paper was published by the same authors [29]. The new paper introduced the facial attributes analysis system to the *LightFace* framework, the new advanced network was then called *DeepFace* which was the collective name for both the

LightFace and *HyperExtended LightFace* frameworks. The facial attributes analysis model allowed for the extraction of more information from detected faces. This information has been taught to the model using multiple datasets and weights that divide faces into four facial attributes, age, gender, facial expressions, and race [29]. The facial expressions and race attributes include several categories as well, and these categories are seven expressions and six races.



```
{
 "age": 28.66,
 "emotion": "neutral",
 "gender": "Woman",
 "race": "latino hispanic"
}
```

```
{
 "age": 29.27,
 "emotion": "happy",
 "gender": "Woman",
 "race": "white"
}
```

```
{
 "age": 29.27,
 "emotion": "surprise",
 "gender": "Woman",
 "race": "white"
}
```

```
{
 "age": 29.74,
 "emotion": "neutral",
 "gender": "Woman",
 "race": "white"
}
```

Figure 2.3: Facial attribute analysis for multiple faces with different expressions [29].

As shown in **Figure 2.3** above, DeepFace uses the same model for facial recognition but an extended version which allows for analysing facial attributes. These attributes are not always 100% accurate as the framework uses different facial detection models trained in different ways. This results in some models being more accurate for facial recognition and attributes. The DeepFace framework uses seven facial recognition models, VGG-Face, Facenet, OpenFace, DeepFace, DeepID, Dlib, and ArcFace. It is unnecessary to use more than one model, especially since all models present an accuracy above 90%, with OpenFace being the lowest with 93% and Facenet512 being the highest with 99.65%. The table below shows the different detection and recognition accuracy results for all the models.

Model	**Score**
Facenet512	99.65%
ArcFace	99.41%
Dlib	99.38%
Facenet	99.20%
VGG-Face	98.78%
OpenFace	93.80%
DeepID	97.05%

Table 2.1: Prediction Results For DeepFace.

2.2 Facial Anonymization

Considering the domain of technological surveillance is ever-expanding, and places from streets, schools, and public and private places all either have or are soon to have surveillance cameras, the privacy of everyone attending these places is always at risk. Of course, these places can promise that the footage or people's privacy will never be at risk but who can assure such a thing. This is where the concept of face anonymisation comes in, the assurance that people need to be less concerned about increasing surveillance cameras. The main idea is that the faces in CCTV footage should either be removed, covered or changed. The concept has existed for a while and could be used in news reports where subjects or general people interviewed have their faces blurred out for their security or privacy. This is a form of facial anonymisation; however, the editors do a manual job in video editing software. Even though the results are always good for such videos, this would fail if used for CCTV footage. Millions of surveillance cameras worldwide produce millions of surveillance footage that would be impossible to anonymise manually.

Nevertheless, it is not impossible to speed the process up by automating it. That is where artificial intelligence comes in. Using different neural networks with trained models can make it easier to anonymise the videos by taking an extensive amount of videos and running them through the neural networks to detect human faces and automatically perform facial anonymisation. This, as mentioned before, can be done in three ways, mainly blurring the faces, removing the faces or changing the faces. The latter is the hardest one as it requires an additional model trained to generate new faces that can then be used to replace the original faces. This process is generally referred to as *De-identification* which is the most difficult of the anonymisation methods to perform. All types of facial anonymisation require the same steps to work, and these steps start by firstly detecting the human faces and then calculating the essential data points in the face. These crucial data are called landmarks and are slightly different based on the model used. However, all the models require the first landmarks: the eyes, nose, and mouth. In addition, some other data can be acquired like colours of the eyes, jawline, skin colour and more. The network then uses these landmarks to determine what to do with the face, either removing the landmarks which result in a colour covering the face, blurring the landmarks and their surroundings or extracting the landmarks and replacing them with something else.

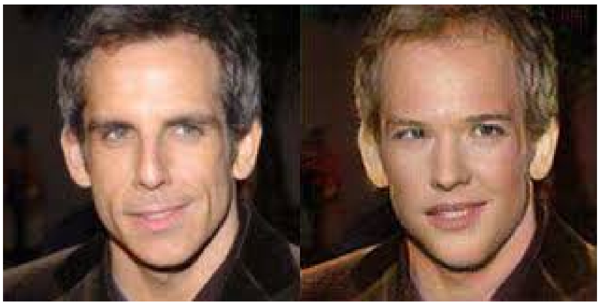


Figure 2.4: Anonymized face used a newly generated face to cover the original [27].

Figure 2.4 above shows the results of facial anonymization using models to generate a new identity. The work is the result of a work done by **Zhongzheng Ren, Yong Jae Lee, Michael S. Ryoo** in a paper they published in 2018 [27].

2.2.1 CIAGAN

One of the anonymisation networks used in this thesis is the *CIAGAN* network, which stands for **Conditional Identity Anonymization Generative Adversarial Networks**. The network was introduced in a paper published in 2020 by **Maxim Maximov, Ismail Elezi** and **Laura Leal-Taixe** [22]. The paper introduced a method to utilise *conditional generative adversarial networks* to detect and anonymise human faces in images and videos. Furthermore, the paper suggests that the *CIAGAN* network can preserve enough characteristics from the original face to generate realistic enough new faces. That also includes the use of *emporal consistency* which essentially allows the network to keep track of poses throughout a video to ensure that the newly generated faces consistent.

The model is designed to start by using the *dlib* face detector to detect the existence of human faces in an image. The algorithm then calculates the positions of these faces using box annotations to find the face in the absolute centre of the image. The faces are then cropped out of the image and secluded in a different folder while the rest of the image is discarded. Moreover, a landmark processor is used to detect the different landmarks in the cropped out face. It generally uses a combination of a five-point landmark system for the detection and 68 point landmark system for the anonymisation. Finally, the network takes this collected information. It produces two new files, one that contains the outline of the face, including the shape of the eyes, nose, and mouth, and a second file containing a mask that the network uses to determine the size and position of the whole face.

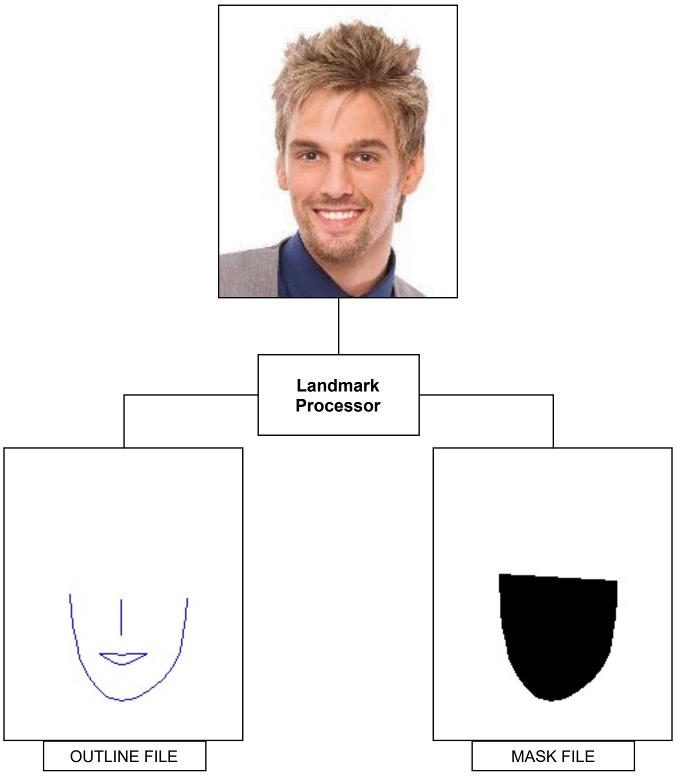


Figure 2.5: CIAGAN Landmark processor producing an outline file and a mask file.

Figure 2.5 above shows how a face is processed to produce an outline file and a mask file. The positions of the eyes, nose and mouth, as well as the shape of the face, are accurate calculated.

CIAGAN uses these two new files to generate a new identity that contains the necessary landmarks positioned in the same relative space as the position calculated in the outline file. The new identity is then masked on the subject using the same landmark positions calculated previously.

However, the CIAGAN network is not without limitations. One of its main issues is that it is

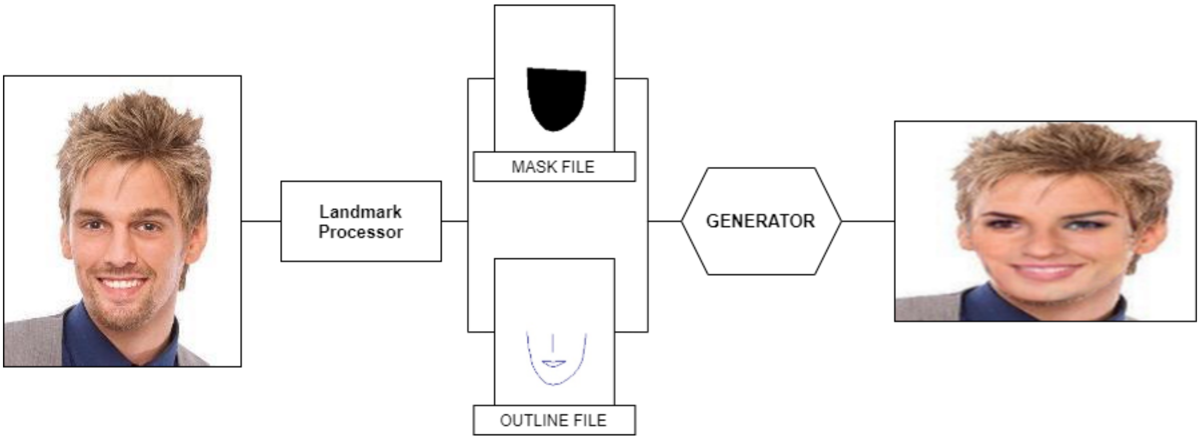


Figure 2.6: The CIAGAN network uses outline and mask files to generate a new identity.

limited to detecting and anonymising a single face from each image. The network detects the faces in an image and finds the face in the centre, and crops it out while discarding the rest. This means that the network cannot be used for crowd anonymisation or anything similar. Another problem is that the network is hardcoded and biased toward a specific dataset, that being the *CelebA* [8] dataset, which means that the code itself is designed to suit the size of the images within the dataset and thus requires configurations and tweaks to make it correctly work on another dataset. Furthermore, the paper talks about CIAGANs ability to anonymise videos which there is no proof of. The code itself does not have indications of a way to use videos as input for the network.

The team had worked on anonymisation using CIAGAN previously; the work done on the network showed that these limitations are only possible to bypass using external ways. Thus, the team used recalculation methods to ensure that all the faces in an image were used. First, all the images' subjects were cropped out and separated, then anonymised separately. To anonymise the entire image that contains all the faces, the newly generated faces are then inpainted on top of the original ones, which was done using a different method than the CIAGAN in-built inpainting, which only allows the masking of a single face.

2.2.2 DeepPrivacy

Håkon Hukkelå, Rudolf Mester, and **Frank Lindseth** from the *Department of Computer Science* at the Norwegian University of Science and Technology in Trondheim published a paper in October 2019 [15]. The paper suggests the use of a *conditional generative adversarial network* based architecture capable of detecting and anonymising a large number of faces in a given image. The architecture utilises annotation and bounding boxes to detect faces and determine the landmarks of the face. After that, the trained model generates new faces and seamlessly paints them on the original faces. The newly generated faces try to look as seamless as possible by somewhat trying to use the same skin tone as the original faces.

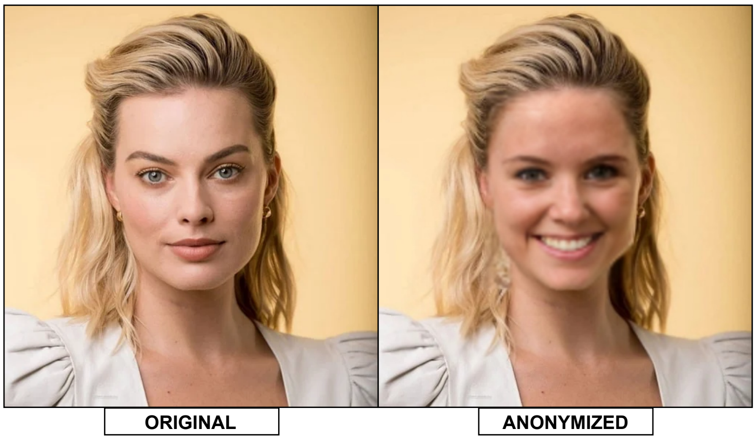


Figure 2.7: Results of anonymizing a single-face image using DeepPrivacy.

The paper suggests that the DeepPrivacy network can also anonymise subjects in videos. This is done automatically by detecting the media and determining that it is a video, splitting the video into singular frames, and running the same process mentioned above on every frame. The last step is to merge all the newly generated frames into a video file.

Even though the network seems to work great on single images by anonymising and masking the subjects in an image, it still has some limitations and issues. These limitations become more evident when using the network on video data. The problem is the consistency of the faces in the video; the consistency of faces generally means that the network does not keep track of the subjects it already anonymised in previous frames. If a subject appears in multiple frames, the algorithm generates a new face for that subject for every frame, resulting in different new faces generated for the same person. After merging the frames into a video, the same person who appears for 100 frames will have a new face in each frame, resulting in jittering and less seamless transition throughout the video. This would not be an issue when trying to anonymise singular images. However, anonymising videos this way would kill the illusion of having fully anonymised and seamless masking that the paper suggests.

Furthermore, another issue that DeepPrivacy has is the issue of age and gender preservation. Of course, machines and humans determine age and gender differently. Even though a machine is trained to see the distinction, it can still fail to determine the difference accurately.

However, DeepPrivacy is not trained for this task, which means that the age and gender of the newly generated faces by the DeepPrivacy network are wrong, especially the age of the subjects. Nonetheless, more work on *DeepPrivacy* was done by **EMILIJA JASINSKAITE** & **ØYVOR YSTAD SKJEI** in 2021 [16]. They proposed a novel method that combines *DeepPrivacy* and attribute-driven GAN to preserve gender and age for subjects appearing in CCTV footage.

2.3 Inpainting

The concept of inpainting itself dates back hundreds of years as it was used in several applications, from repairing old paintings or replacing some parts of a painting with something different [6]. In addition, there were people dedicated to painting restorations in the previous centuries where the concept of inpainting was the primary restoration method.

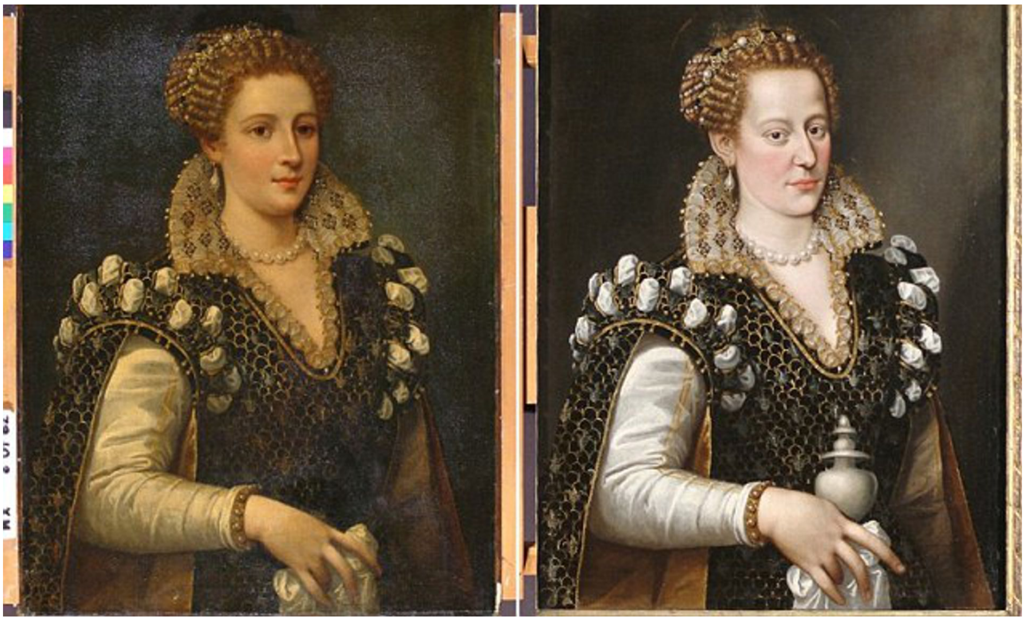


Figure 2.8: A mother (left) painted on top of her daughter's face (right) [25].

Figure 2.8 shows a painting of *Eleanor of Toledo* which, after restoration, it was revealed that her face was repainted on her daughter's Isabella Medici face. This was done in the 1500s to cover the daughter's face considering she was scandalous by their standards. The restoration was done by *Ellen Baxter* [25] and shows that only the face was changed, and all the clothes and environment were kept. This is an application of inpainting before the age of digital media.

Nowadays, inpainting is still used in digital media, which comes in the form of editing images or videos to change the facial look of subjects in these media. Moreover, it is used in some types of anonymisation, and an example would be **Deep Fake** where faces are extracted of subjects and inpainted on others while making sure that the newly inpainted faces move the same way the previous ones did.

2.3.1 FSGAN

Digital inpainting, as mentioned above, interests itself with the process of extracting something out of an image and masking something in its place. When it comes to the subject of this thesis, then digital inpainting of the human subject is the most significant focus. Much work has been performed on the concept, and one of the more ambitious works done on the subject was introduced in a paper published in 2019 by **Yuval Nirkin, Yosi Keller** and **Tal Hassner** [23]. The paper introduces a *generative adversarial network* capable of performing two tasks, **Face Swapping** and **Face Reenactment** . The former task presents itself by taking a source image and a video file, extracting the face of the human subject from the video file and replacing it with a human face from the source image [23]. The final result would be a video file containing the original human subject but with a different face. Both facial poses and expressions are still preserved from the original video file.

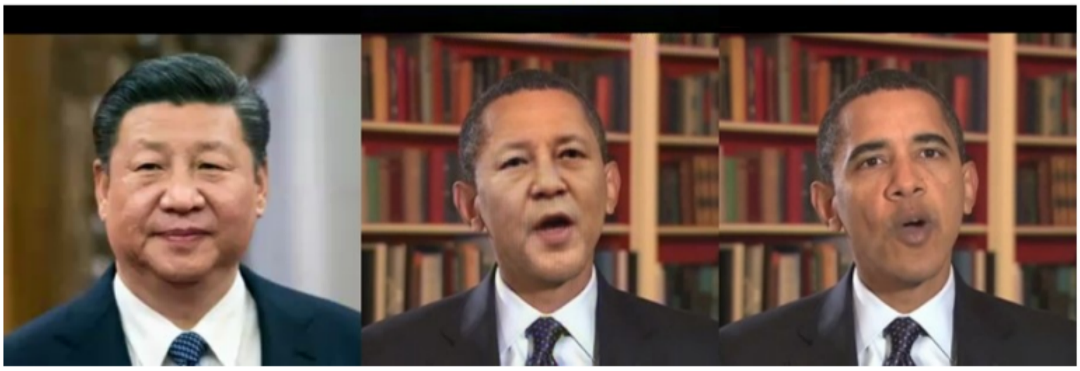


Figure 2.9: Source image (left), target video (right) and results (middle) [23].

The **Figure 2.9** shows how the results of using *FSGAN* on a source image and a target video are. The results present themselves in the middle, where the face from the source image is inpainted on top of the original human face from the video target.

The second task, face reenactment, on the other hand, takes a source image and video file and extracts the animation, expressions and poses from the video file and applies them to the source image. The result is an animated image rather than masking since the image preserves its original face and structure. *FSGAN* uses a *recurrent neural network* approach to manage and adjust poses and expressions for the face reenactment task [23].

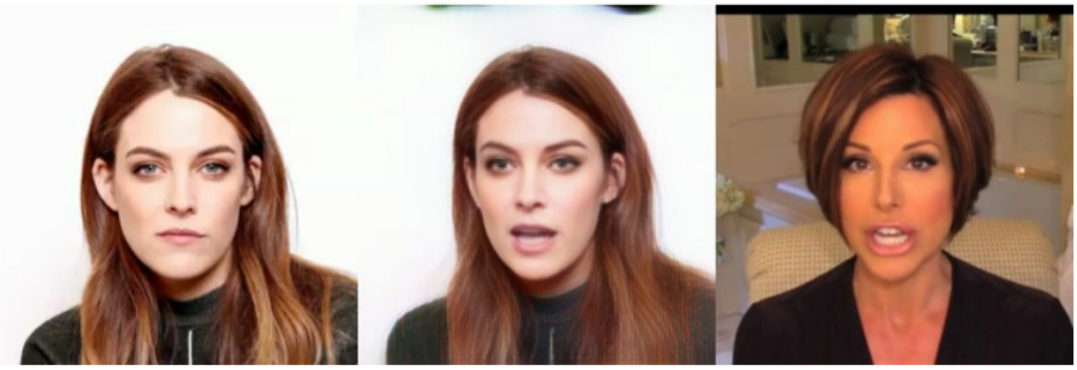


Figure 2.10: Source image (left), target video (right) and results (middle) [23].

The results of the facial reenactment are shown in **Figure 2.10** where the animation, poses, and expression from the target video are extracted and applied to the source image, which results in an animated image shown in the middle where the human subject from the source image inherits the full animation of the person in the target video.

However, even though the results from the FSGAN framework are considered outstanding by quantitative results, it still has some use limitations. Firstly, the hardware requirement for the framework to work is astonishing as it requires high-end NVIDIA GPUs with a minimum of 11GB of DRAM [11]which makes it difficult to even use for most pc-es as the number of people having access to high-end GPUs is minimal. Furthermore, the output video's resolution is always low quality, even if the GPU is high-end. Finally, the paper suggests that the framework can work with target videos with two human subjects. However, there are no quantitative results to show for this claim.

2.3.2 SC-FEGAN

A more manual approach to digital inpainting was introduced in a paper published by **Youngjoo Jo, Jongyoul Park** [18]. The paper presents an image editing system that is semi-manual, as it requires some input from the user to be able to edit an image, but not so much that it relies solely on the user's input. The user's input comes in the form of coloured or non-coloured sketches that the system uses as a guideline to base the generation upon [18]. Furthermore, the user's sketches are used as masks to design the look of the generated areas. The sketches provided must be realistic enough to ensure that the data is translated correctly before the image completion process is executed.
Figure 2.11 shows how the *SC-FEGAN* network generates a new facial area and inpaint it

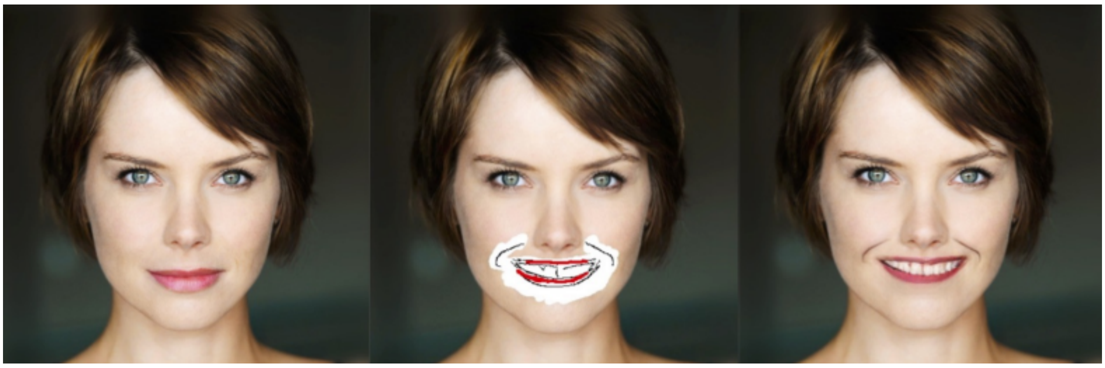


Figure 2.11: Source image (left), user provided sketch (middle) and results (right) [18].

on top of the original face based on a sketch provided by the uses. The network can even generate things that do not exists in the original image like earrings, ring, etc, if the user sketches provide such data in the sketches.

The biggest problem with this is that it is not an automatic process. The image data must come in pairs to work, a source image and a sketch mask for that specific source image. That means that the network cannot take two different source images and inpaint one on the other. Furthermore, the human subjects in both provided images must be the same. The network itself is designed as an editing network, so it makes sense that the source images must have the same person. However, in a system where anonymised faces are generated for each human subject, the *SC-FEGAN* would not be capable of inpainting the anonymised face on top of the original one.

2.4 Optical Flow

Masking faces in videos require a calculation that allows for smooth and semi-seamless mask movement to ensure that the masks move in a way where the original face is always masked at the correct time and in any frame. This task is far from simple as the algorithm does not know how a face will look in the next frame. That results in the masking algorithm waiting for the face to move first and then taking some time to re-detect the new position of the face and re-mask it. Thus, the video would look especially where there are sudden fast movements like it is slowly being masked. That means that the original face is visible for some time, and then it gets masked, which defeats the purpose of masking it in the first place.

One of the solutions that have proved successful when working on such a task is called *Optical Flow*. Optical flow is the distribution of movement velocities based on brightness patterns [12]. The optical flow itself is used in several papers and subjects within artificial intelligence as it is a detection and segmentation concept focusing on object motion, in a paper titled **The Computation of Optical Flow**, published in 1995. The authors Steven S. Beauchemin and John L. Barron of the University of Western Ontario describe the concept *Optical Flow Estimation* which is essentially a step further in the calculation of optical flow that can be used to estimate the motion between two frames [5].

Considering that the masking calculates its new position after the face appears on the screen first, optical flow estimation can be used in this instance to estimate where the face would appear before it appears. This results in the mask appearing on top of the original face when the face appears. However, for the optical flow estimation to work as flawless as possible, the movement of the face between two frames should not be too sudden, which means that if a video has a cut between two frames where the first face in the second frame is too different in position and movement from the first frame, the estimation would not be calculated correctly. In 1994, John L. Barron, D.J. Fleet, and Steven S. Beauchemin published a paper titled **Performance of Optical Flow Techniques** [4]. The paper described how different techniques could be used to calculate optical flow to determine object motion between two frames [4]. Those techniques were divided into sets. One of those sets is called the Differential Techniques. That set of techniques bases itself on the calculation of a voxel at a given location and an intensity level to produce a bright constancy constraint [24]:

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t). \tag{2.1}$$

Using Taylor Series, the formula can be developed into:

$$I(x + \Delta x, y + \Delta y, t + \Delta t) = I(x, y, t) + \frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t + higher - orderterms. \tag{2.2}$$

Thereafter, the formula can be divided by Δt to produce:

$$\frac{\partial I}{\partial x}\frac{\Delta x}{\Delta t} + \frac{\partial I}{\partial y}\frac{\Delta y}{\Delta t} + \frac{\partial I}{\partial t}\frac{\Delta t}{\Delta t} = 0. \tag{2.3}$$

Knowing that $I(x, y, t)$ from the first formula represent the intensity where (x, y, t) represent the location of a voxel, then V_x, V_y are the x, y of the optical flow of $I(x, y, t)$ [24]. The above mentioned equation is the base equation for the optical flow algorithms, however, there still is need for one or several more equations/formulas to find the actual optical flow.

2.4.1 Deep Feature Flow

Optical flow estimation is a way of accurately and smoothly detecting and following objects in a video. The concept was utilised by **Xizhou Zhu, Yuwen Xiong, Jifeng Dai, Lu Yuan**, and **Yichen Wei** of the University of Science and Technology of China. Their paper [32], published in 2017, introduces a video recognition framework that takes advantage of optical flow and network acceleration to achieve end-to-end video recognition that is more accurate and much faster than regular methods [32]. The optical flow algorithms improve the speed of object detection and segmentation in videos by calculating the movement of objects and predicting their next movement to ensure the detection boxes are always accompanying the objects in a video.

Chapter 3

Approach and Implementation

3.1 Proposed Approach

In an attempt to achieve the goal described in **Section 1.2, Chapter 1** several ideas and architectures were considered. Experiments that are typical and not for machine learning were conducted before finally settling on a method. This section explains the architecture of the project, the reasons for establishing such architecture, and how it will be used.

The method that was settled on was an architecture that makes it easier to control the process of anonymisation and inpainting while simultaneously making automatisation a bit harder. The main idea is an architecture that starts by using the *OpenCV* library to take a video and split it into frames. The algorithm goes through each frame and detects every face. Each Face in The frame is then run through a facial attribute system where the *DeepFace* network **(Subsection 2.1.2)** starts a verification process where it has a storage folder that stores unique faces from a video. The *DeepFace* network detects the face from a given frame and starts to verify it with the faces saved in the storage folder. *DeepFace* can recognise faces even with different poses and expressions, which makes it more than powerful enough to verify the same face in two frames even if the expression or the pose is different. After verifying the faces, the algorithm then either adds the face to the unique folder if it does not exist beforehand or disregards it if it already exists, then proceeds to the next face

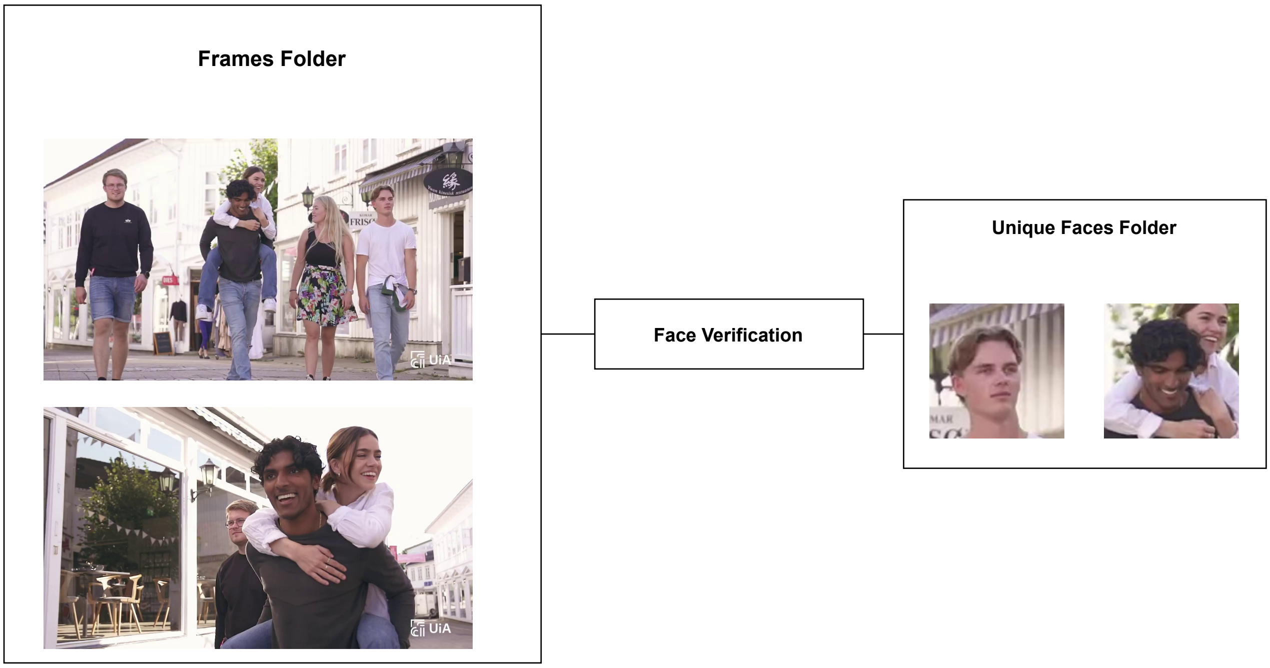


Figure 3.1: Two Images containing the same people (on the left) and what is considered unique and extracted (on the right).

Figure 3.1 above shows how in two images where some of the faces appearing in the first one appear in the second one, the algorithm recognises that the faces re-appeared in the second image; thus, they are not unique and will be disregarded.

The next stage of the network is the anonymisation which goes through the faces in the unique faces folder and anonymises each of the images there to produce a new set of images that represent the new identities to be used to make the original images. After that, the most complicated step is inpainting, which happens in two stages for each image. Firstly the network uses the verification model from the *DeepFace* network to check if a face in an image exists in the individual folder. If it does, then pick the corresponding anonymised face from the anonymised faces folder. If not, move to the next face. After finding the corresponding anonymised face, the network proceeds to inpaint the new identity on top of the original one in the image. Faces that do not get detected get skipped and remain as they are. The final step is merging the new inpainted frames into a video file.

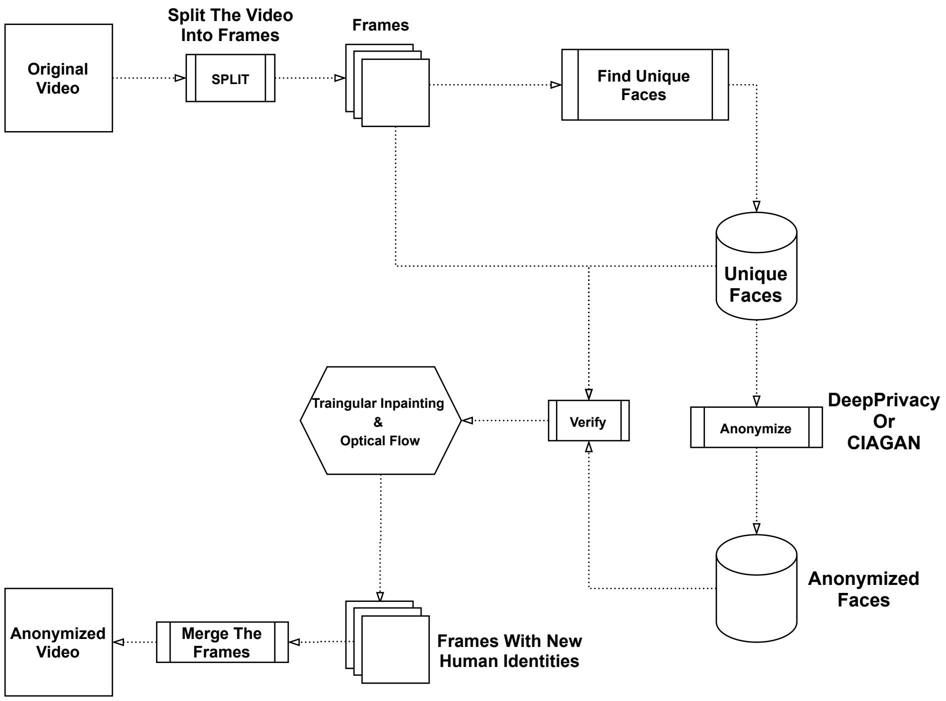


Figure 3.2: The main architecture for the network.

The architecture for the network is shown in **Figure 3.2** above, where the cycle of each video starts by splitting the video into frames, extracting unique faces, anonymising said unique faces using either *CIAGAN* or *DeepPrivacy* to see different results, start the inpainting process by assigning every face in each frame for its corresponding anonymised face, mask the original faces with the corresponding anonymised faces using triangular inpainting and finally merge the frames to produce a new video file. To explain the process of the architecture. First, a video file is taken and split into frames. Each frame is then run through the verification system, where the faces are extracted. The extracted faces are saved into a folder called unique faces. If a face already exists there, then its replica from a different frame would not be saved there again. The faces in the unique faces folder will be anonymised using either *DeepPrivacy* or *CIAGAN*, and the results are saved in the *Anonymised faces* folder. The next step is a verification process where the original frames are taken. Each face in there would receive an anonymised version used for the same person throughout the video's entirety. The penultimate step performs the triangular inpainting on the frames, followed by optical flow. The result would be a set of frames where the faces are anonymised correctly before these frames are merged into a video file.

3.1.1 Dataset

The team proposes the use of the **Hollywod Human Actions (HOHA)** [21] dataset to access a plethora of short video clips. These video clips are extracted from Hollywood movies, both old and new. The dataset was introduced first in a paper published in 2008 by **Ivan Laptev, Marcin Marszalek, Cordelia Schmid** and **Benjamin Rozenfeld** [21]. Other than movie clips, the dataset contains annotations and different subsets of movie clips on different occasions. However, since the annotations, training and testing subsets are not needed, the only thing used from the dataset is several randomly selected video clips. The durations of these clips range from two seconds to almost three minutes. Regardless of the duration of the videos, the dataset was chosen because of the video diversity it contains. The resolutions, number of people appearing, environment, poses, and movement all vary a lot in the *(HOHA* dataset).

3.2 Implementation

The execution of face inpainting begins with splitting videos into frames. Distinctive faces of those frames are gathered in a separate folder. Using the *DeepPrivacy* model, the unique faces are de-identified. All the faces in the video frames are inpainted with those de-identified unique faces before the inpainted frames are merged to become a video.

3.2.1 Video Pre-Processing

Video configuration runs in two steps, splitting the video into frames and merging the frames to produce a video after the entire process of inpainting or anonymisation is finished.

Splitting the video into frames is carried out by opening the video file or camera using the *OpenCV*, a computer vision python library. By reading frame by frame, each frame is saved using *cv.imwrite()*, a function of the *OpenCV* python library. The user can decide what number of frames to extract by choosing to extract all frames or once every specific amount of time [14]. The algorithm of video splitting is shown below.

Algorithm 1 Splitting a video into frames.

```
Require: import cv2 package of OpenCV library
  videoCap ← full path of the video
  while videoCap is open do
    success, frame ← read videoCap
    # success is false if there is no frame again in videoCap
    if success is false then
      break
    else
      save each frame into a folder
    end if
  end while
  then release videoCap
```

After anonymising and inpainting all the frames, they are merged to recreate the video as it was before splitting, but now all the faces are de-identified. All the de-identified frames are read one by one to a *NumPy* array using *cv2.imread()*. Thereafter, the image array is written to a video file using the *OpenCV* method, *cv2.VideoWriter* [13]. The algorithm for merging frames into a video is shown below. The *filename* is a file where the output video is stored, *fourcc* is the 4-character code of codec used to compress the frames, *fps* is the frame rate of the video stream, and *frameSize* is the height and width of the frame.

Algorithm 2 Merging frames into a video.

```
Require: import cv2, numpy, and glob
  width ← any chosen width size for an image
  height ← any chosen height size for an image
  fps ← desired frame rate
  filename ← path to output video
  fourcc ← cv.VideoWriter_fourcc
  outWriter ← cv2.VideoWriter(filename, fourcc, fps, (width, height)
  for each frame in frames do
    img ← cv2.imread (frame)
    outWriter.write(img)
  end for
  then release outWriter
```

3.2.2 Image Verification

The *DeepFace* framework is used for image verification. The verify function verifies face pairs as identical or different people by taking two image paths as inputs [30].

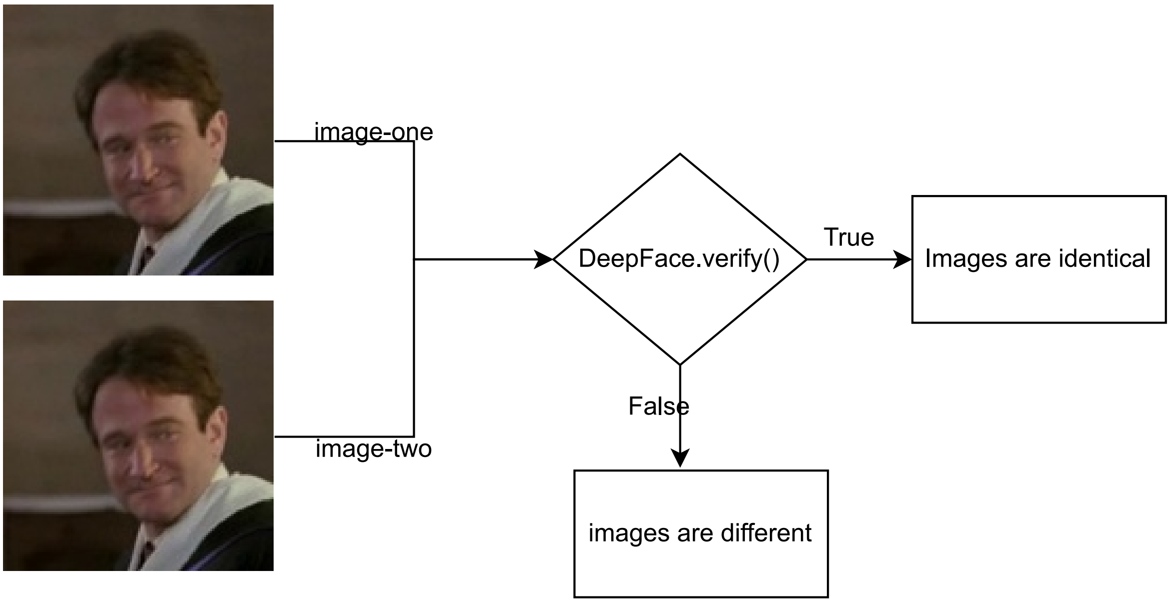


Figure 3.3: Verify if the two images are identical.

Algorithm 3 verify whether image-1 and image-2 are the identical or not.

```
Require: DeepFace: deep learning facial recognition system
  DeepFace.verify (image-1, image-2)
```

The verification function decides whether the two face images are identical or not by finding the distance between their vector representations. If the distance is less than a threshold value, the two faces are categorised as the same person value [30].

verified : True

distance : 4.440892098500626e-16

threshold : 0.4

model : “VGG-Face”

detector_backend : “opencv”,“similarity_metric”:“cosine”

Table 3.1: Verification result of two images.

Models **VGG-Face, OpenFace, Google FaceNet**, and **Facebook DeepFace** are supported in the *DeepFace* framework. By default, the *VGG-Face* model is used [30].

3.2.3 Inpainting

The pipeline for the face inpainting consists of several phases: finding unique faces from all frames of a video => de-identifying found unique faces => detect a face in the video frame => find this face among the unique faces and fetch its corresponding de-identified face => find landmark points of both faces, the one from the video frame and its de-identified face => find triangles in both faces => integrating the de-identified face into the original face from the video frame, and finally, seamless cloning and optical flow.

Find unique faces

The initial phase in face inpainting is to find all the unique (distinctive) faces throughout the video. *Dlib frontal face detector* is used to detect the frontal faces in the images. It performs the face detection based on *HoG* (directional gradient histogram) feature descriptor with the linear *SVM* (support vector machine) machine learning algorithm [3].

Algorithm 4 Finding all the unique faces in the video frames.

```
Require: built detection model (shape_predictor_68_face_landmarks model), face recog-
  nition model DeepFace
  unique_faces ← folder containing unique faces
  detector ← dlib.get_frontal_face_detector()
  faces ← detector(image)
  for each face in faces do
    DeepFace.verify(face) with all saved unique faces
    if a face does not exist then
      add the face into unique_faces folder
    else
      continue
    end if
  end for
```

After detecting each face in the video frame, the algorithm checks whether the face is already saved as a unique face or not. The *DeepFace* verification function, as defined in **Subsection 3.2.2**, is used to compare the detected face with each of all saved unique faces. If the detected face is not unique, it will be saved as a new distinct face.

De-identify found unique faces

The unique faces saved from all the frames of the video are de-identified using *DeepPrivacy*. The *DeepPrivacy* model is based on a *conditional generative adversarial network*, and it generates new faces based on the original pose and image background [15].

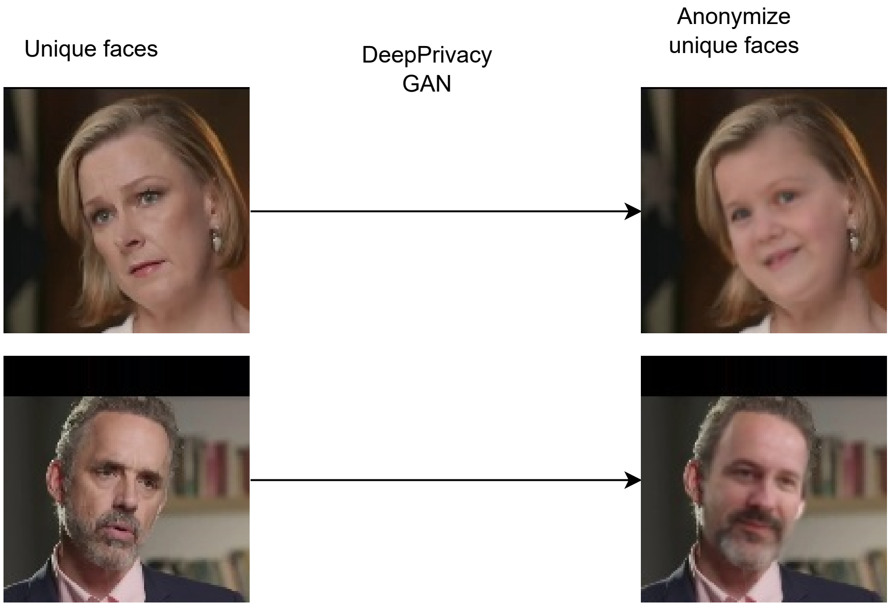


Figure 3.4: De-identifying all unique faces.

Detecting faces

Identically, as for finding the unique face, the *Dlib frontal face detector* is used to detect the frontal face of a face from the video frame. The algorithm for detecting a face in a frame is shown in **Algorithm 4** above. After detecting the face in the video frame, using the verification method defined in **Algorithm 3**, this face is being found among the unique faces saved from all the video frames as unique faces. It executes by looping through all the unique faces and verifying each with the face from the video frame. Then, the de-identified face corresponding to the unique face is fetched to be used for inpainting over the face in the video frame.

Landmark points

The next phase is the face landmark detection of both the original face from the video frame and its de-identified face.

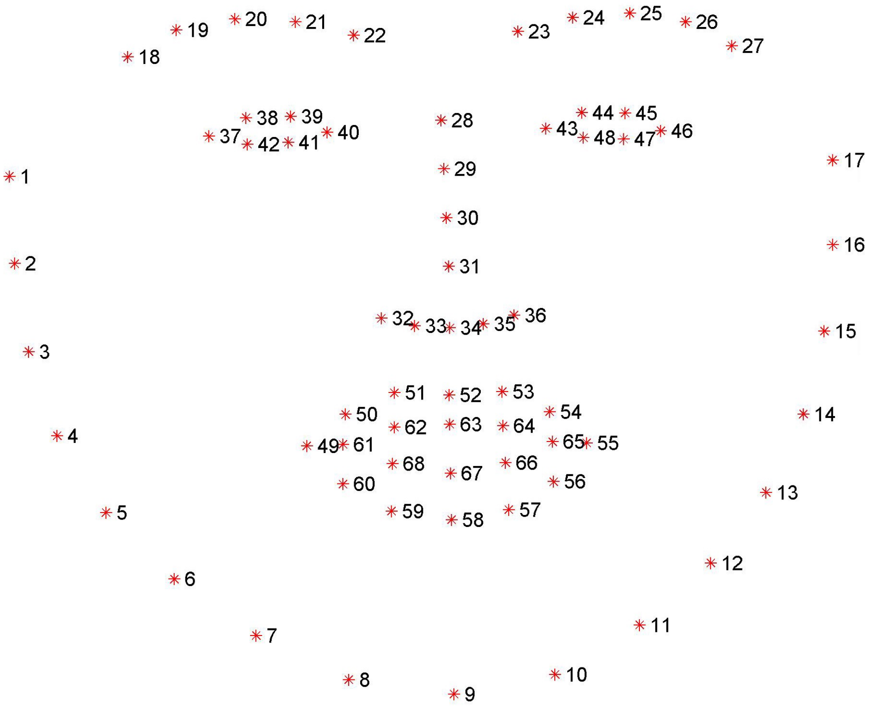


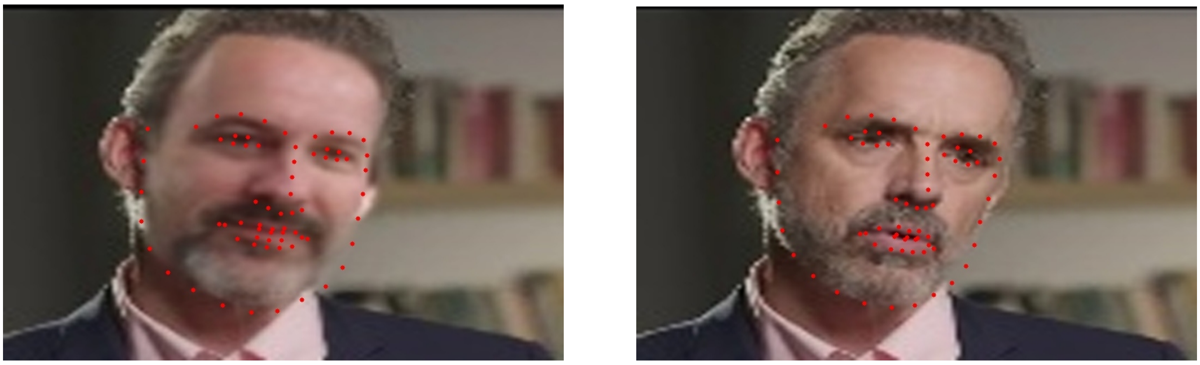
Figure 3.5: Processed 68 facial landmarks on a detected face [28, p. 4].

Using the *dlib* pre-trained face keypoint detector, the key points of the individual face in the image are detected. Those face key points are used to locate and represent the most important area of the face, like the eyes, nose, eyebrow, jawline, and mouse. The shape prediction method is used to detect significant facial structures. As shown in **Figure 3.5** above, the *dlib* estimates the location of 68 coordinates corresponding to the facial points on a face.

Algorithm 5 Predicting landmarks of face in an image.

```
Require: image: each face in an image
  predictor ← getting the 68 face_landmarks shape predictor
  for every face in an image do
    landmarks ← predictor(image, face)
    landmarks_points ← empty array for saving landmark points of each face
    for point in 68_landmarks do
      x_coordinate ← landmarks.part(point).x
      y_coordinate ← landmarks.part(point).y
      # add both coordinates into the array list
      landmarks_points.append(x_coordinate, y_coordinate)
    end for
  end for
```

The landmarks predictor takes the whole image and the specified face to predict the landmark points. By looping throw the 68 landmark points, the x and y landmark coordinates are obtained as shown in **Figure 3.6** below.



(a) (b)

Figure 3.6: a) De-identified face. b) Original face from video frame.
The location of 68 coordinates corresponding to the facial points on both faces.

Convex hull

After detecting and predicting all the landmark points of the face, the *convex hull* of those landmarks can be drawn. The *convex hull* is the face border of landmark points that encloses all of the points in the set. As shown in **Figure 3.7** below, all the landmarks are on and inside the convex hull.

Algorithm 6 Convex hull of the landmarks of the face.

```
Require: OpenCV function convexHull, numpy array
  points ← array of landmarks_points
  convexhull ← cv2.convexHull(points)
  cv2.polylines(img1, [convexhull], True, (255, 0, 0), 2)
```

Using an OpenCV *convexHull* function, a convex shape contour can be drawn. It takes the landmark point sets and returns the convex hull of those sets.

It is the mask of this *convex hull* of the de-identified face which is going to be inpainted over the original face.

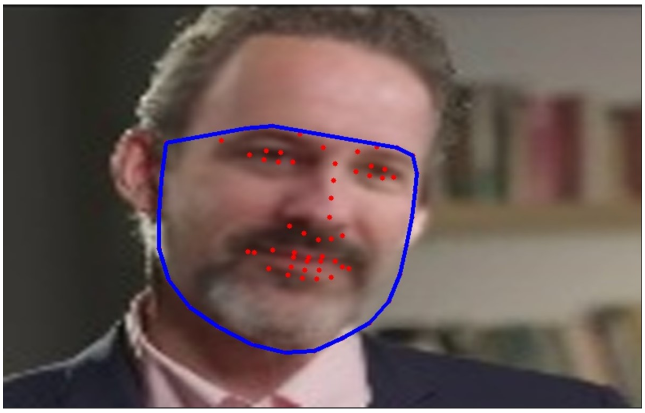


Figure 3.7: The convexhull which encloses all the landmark points in a face.

Triangulation

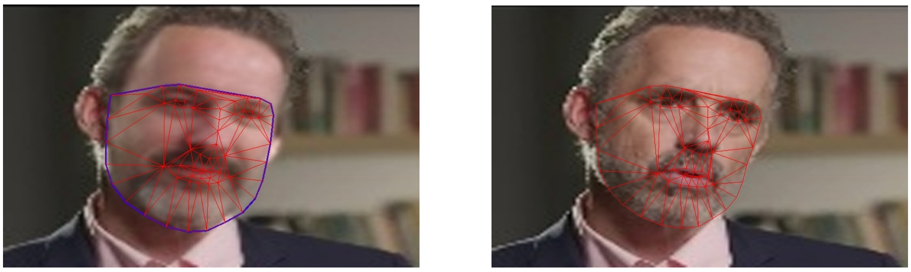
Both faces, the original face from the video frame and its de-identified face, have different sizes and resolutions. Therefore, cutting out the face from the de-identified face and putting it into the original face is not the most suitable way of face-swapping. In addition, it is not a good practice to change the size and resolution of the de-identified face right away to the size and resolution of the original face because the face would lose its original proportions. So, instead, both faces are divided into triangles where the triangles have the same patterns. The same patterns mean that triangles in both faces have the same connection points. Then, both the triangles are extracted and warped.

A Delaunay triangulation is used to subdivide the faces into triangles. It subdivides a set of points in a face into triangles where the landmark points become vertices of the triangles [9].

Algorithm 7 Triangulation: subdivide a face into triangles

```
  rectangle ← cv2.boundingRect(convexhull)
  subdivide ← cv2.Subdiv2D(rectangle)
  insert all the landmarks_points into subdivide (subdivide.insert(landmarks_points)
  triangles ← subdiv.getTriangleList()
  triangles ← np.array(triangles, dtype=np.int32)
```

An OpenCV *boundingrect* function gets the *convexhull* of the face and constructs a rectangle of it. Then, using an OpenCV subdiv2d function, the rectangle subdivides into several rectangles. The subdivide function returns the triangle list where the landmark points are vertices of the triangles by taking the landmark points. As shown in **Figure 3.8** below, both faces are divided into triangles. All landmark points are connected and there is no remaining empty points within any triangle.

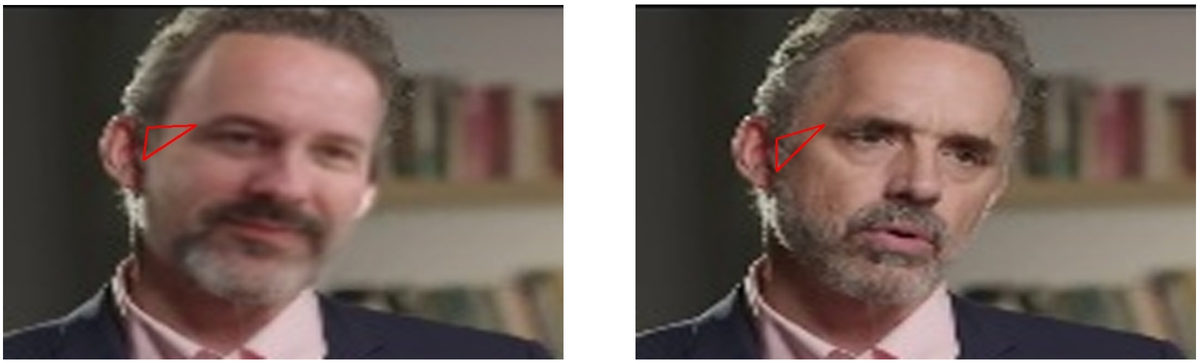


(a) (b)

Figure 3.8: a) Triangulation of De-identified face. b)Triangulation of Original face from video frame.

Extract and Warp triangles

If we see the first triangles of both faces, those corresponding triangles have different sizes. Before starting with the swapping process, triangles on both faces must warp and have the same size.



(a) (b)

Figure 3.9: a) De-identified face. b) Original face from video frame.

The warping process is done with an *OpenCV* function called *affine transform*, *getAffineTransform*. First, the triangle of the de-identified face is taken and extracted. Then, this triangle is transformed to fit with the size and resolution of its corresponding triangle in the original face [7].

Algorithm 8 Extract and Warp triangles

```
Require: landmark points for both the source(de-identified) and destination(original) face.

  # Getting the corresponding points of both faces and output the transformation matrix
  transform_matrix ← cv2.getAffineTransform(De-identified_points, original-points)
  Apply affine transformation to de-identified face with warpAffine() function.
  result ← cv2.warpAffine(de-identified, transform_matrix, output_face_size)
```

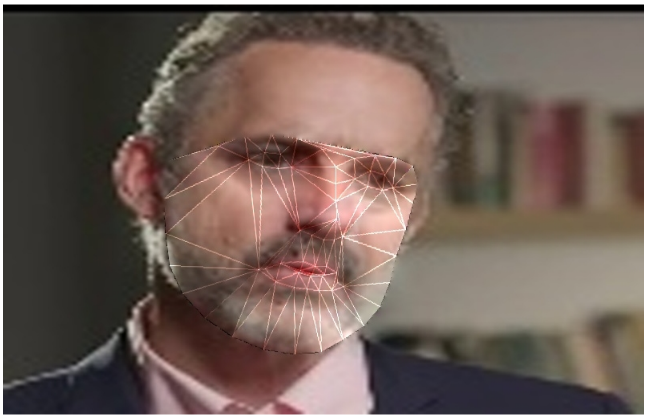


Figure 3.10: All the triangles of the de-identified face are warped into the triangles of the original face from video frame. Then, the original face is replaced with the warped de-identified face.

Seamless Cloning

Once all the triangles in the de-identified face are extracted and warped, they link together. As shown in **Figure 3.10** above, the de-identified face is ready and replaced the original face. The process is done by cutting out the original face to make space for the de-identified face, and then the original image without the face and the de-identified face linking together. In addition to the swapping process of the faces, the colour of the de-identified face should adjust or fit with the destination's original face. An *OpenCV* built-in function, *seamless clone*, makes this adjustment automatically. It copies the face from the de-identified image and pastes it into the original image, making a composition that looks seamless and natural [26].

Algorithm 9 Seamless cloning: adjusting the colors so that the de-identified face fits the original face.

```
Require: mask of the destination(original) face that should be replace, central point of
  original face, and flags(cloning type)
  result_image ← cv2.seamlessClone(de-identified-face, original-face, mask, center, flags)
```

Where **mask** is the size of the clipped face from the original face that will be replaced with a warped de-identified face, and the **centre** is the central location of the de-identified face in the original face. The final anonymized face is shown in **Figure 3.11** below.

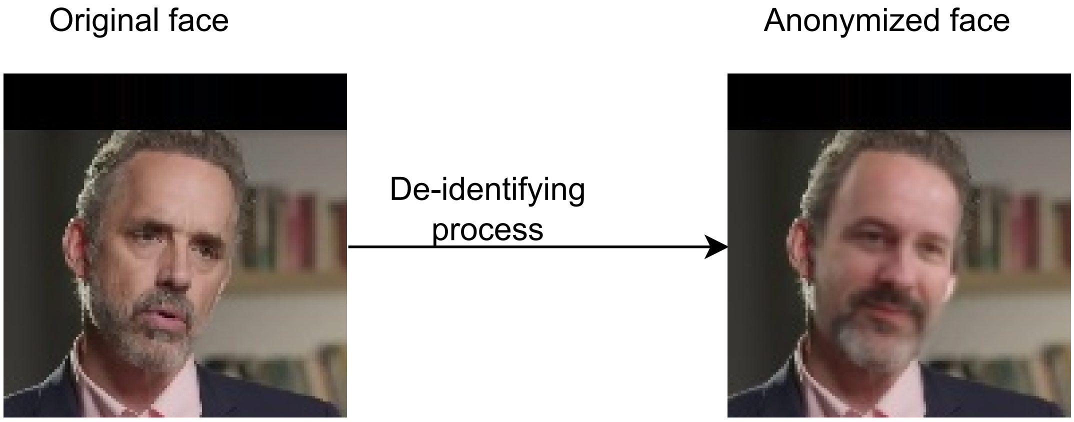


Figure 3.11: De-identifying original face into anonymize face.

Optical Flow

The video frames are inpainted or de-identified perfectly, and it seems realistic. However, when all those de-identified frames merge to become a video, the faces shake. The reason for this is that when the face moves in the video, the inpainted de-identified face, which is on the top of the original face, follows after it with a different velocity, making the glitching visible. To make it stable, an optical flow with the *Lucas-Kanade* method is used. The *Lucas-Kanade* method uses a *differential* method for optical flow estimation that tries to predict the next position of an image based on the previous and current position and velocity of the image.

Algorithm 10 Calculates an optical flow for a feature set using the iterative Lucas-Kanade method with pyramids.

```
Require: prev_image_gray, hull_prev, hull_current, convexHull of current_image
  current_image_gray ← cv2.cvtColor(current_image, cv2.COLOR_BGR2GRAY)
  next_hull ← cv2.calcOpticalFlowPyrLK(prev_image_gray, current_image_gray, hull_prev, hull_current)
  # setting the current hull & image into previous hull & image for estimating the hull of the next face.
  hull_prev ← np.array(hull_current, np.float32)
  prev_image_gray ← current_image_gray
```

By getting and combining the details of the previous points of the images along with the current points, the next position is predicted. It becomes the combination of the detected and the predicted landmark.

Chapter 4

Evaluation and Discussion

The testing of the framework is divided into multiple experiments to ensure that all the sub goals are achieved. The experiments represent the different parts of the framework from simple image anonymization to the more advanced video anonymization, inpainting, and facial preservation.

The results of the above-explained experiments where achieved using the following hardware:

Hardware	**Specification**
GPU	Nvidia 3060 TI
Memory	16 GB Ram
CPU	Intel Core i5 11400F
GPU Memory	8 GB

Table 4.1: Testing Hardware.

4.1 Inpainting The De-Identified Faces

Since the work done on the framework was done in stages based on smaller steps rather than leaps, one of the first things to be worked on was anonymising and inpainting a single image. The first experiment uses the *DeepPrivacy* network to anonymise the face appearing in the image. After masking, the face will then be inpaintied using triangular inpainting and reshaped to suit the original face's shape. The experiment process looks like **Figure 4.1** below. The original image is firstly taken by the *DeepPrivacy* network. The inpainting algorithm uses the results from the network and the original image to produce a semi-realistic, correctly masked anonymised image.

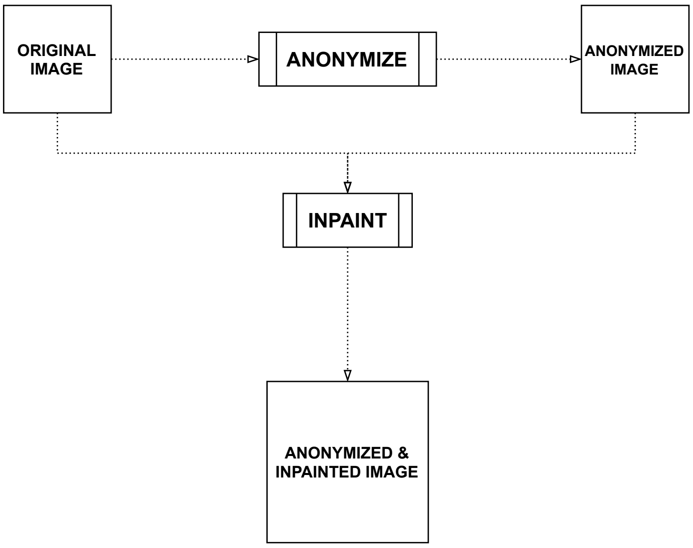


Figure 4.1: The process of which a face from a single image is taken by the system, anonymized using either *DeepPrivacy* or *CIAGAN* and inpainted on the original image.

The experiment did not require pre-processing or even verification for the uniqueness of faces and subjects. That is because its purpose was to anonymise human subjects in an image. What was needed was an image containing visible faces, an anonymised version of each of the faces appearing in the images, and the anonymised version of these faces. Using the image shown in **Figure 4.2** below, the system extracts all the faces, which in this situation, there is only one face available for extraction.

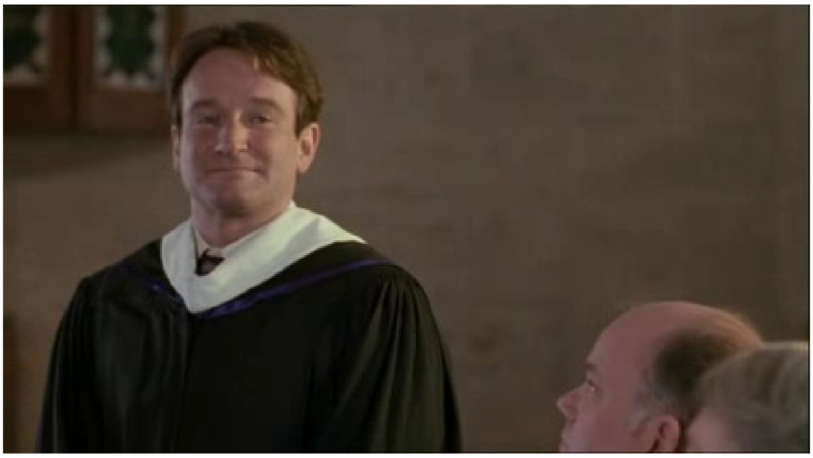


Figure 4.2: The image used before testing **A.1**.

The process creates an anonymized version of the extracted face and inpaint it on the original image to prodocue the results shown in **Figure 4.3** below.

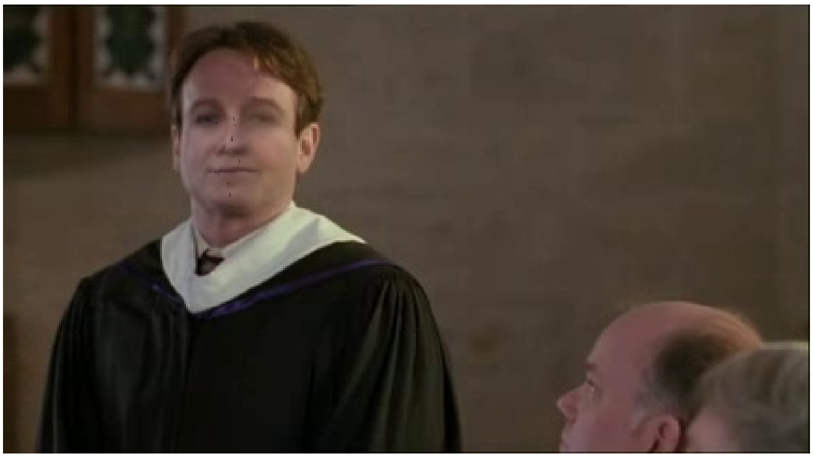


Figure 4.3: The same image after being anonymized and inpainted **A.2**.

Running both the original and anonymized images through the age and gender verification network *DeepFace* **(Subsection 2.1.2)** the following results are shown:

Attribute	**Original Image**	**Anonymized Image**
Age	33	27
Gender	Male	Male

Table 4.2: Age & Gender Prediction For The Experiment Above.

The age and gender verification results are 100% consistent in this situation by running the verification process 20 times on each image. The final result of the experiment shows the inpainting ability to produce good and semi-realistic results while preserving age and gender as closely as possible. In this situation, 100% gender and age within five years of the original age.

4.2 Optical Flow Traceability

The performance of optical flow changes depending on if the video has a single face or multiple. In the situation of a video input with a single human subject, with no regard to the environment, lighting or poses is. The video is split into frames, the frames go through the verification algorithm, and the face will be extracted and anonymised. When inpainting, all the faces of the subject in every frame will have the same anonymised face. The final results should be a video of a singular human subject where the anonymised face is preserved and consistent throughout the whole video. **Figure 4.4** below is the process of the experiment. The video is split into a frame, frames are verified, and the face is anonymised and kept track of. Then the inpainting process starts by masking the anonymised face on top of the original face in every frame while using *optical flow* to make the face movement and transitions smooth throughout the whole video.

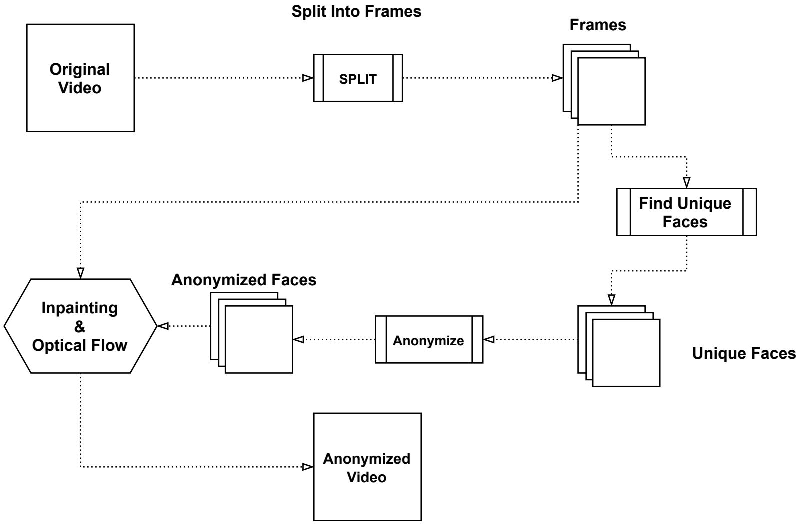


Figure 4.4: A video is split into frames where all faces are extracted and anonymised. Thereafter, the faces are inpainted on top of the original ones before applying the optical flow.

A video with a single subject is visible at all times is provided, for this video with direct face to camera contact was nessesary to tests the full tracibility and prediction of optical flow. One of the frames from the video are shown in **Figure 4.5** below.

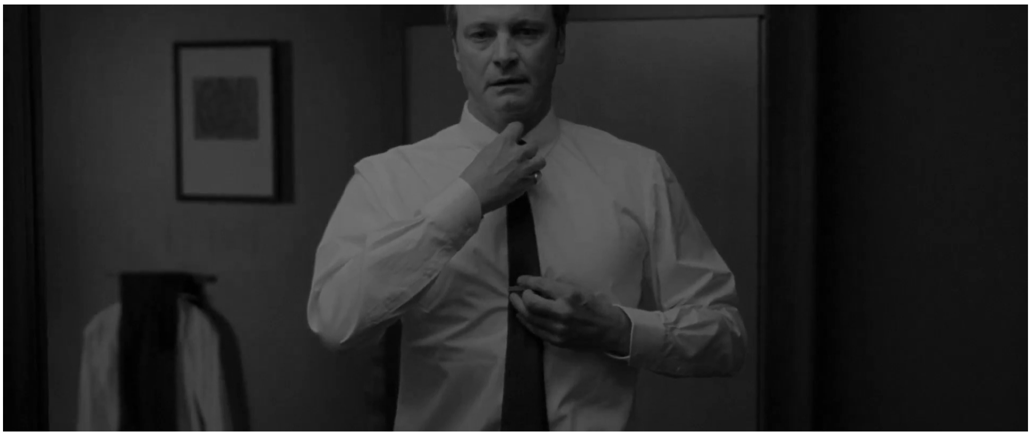


Figure 4.5: A frame extracted from a video [1].

After running the frames through the verification system, it finds only one unique face, which is a given considering there is only one person in the video's entirety. The extracted face is then anonymised with *DeepPrivacy* and given the same file name as the unique face but saved in a separate folder. The unique face, its corresponding anonymised face and the original frames are taken by the inpainting framework to produce a fully anonymised & inpainted video with semi-realistic and smooth movement and transition. The frames inpainted are shown in **Figure 4.6** below.

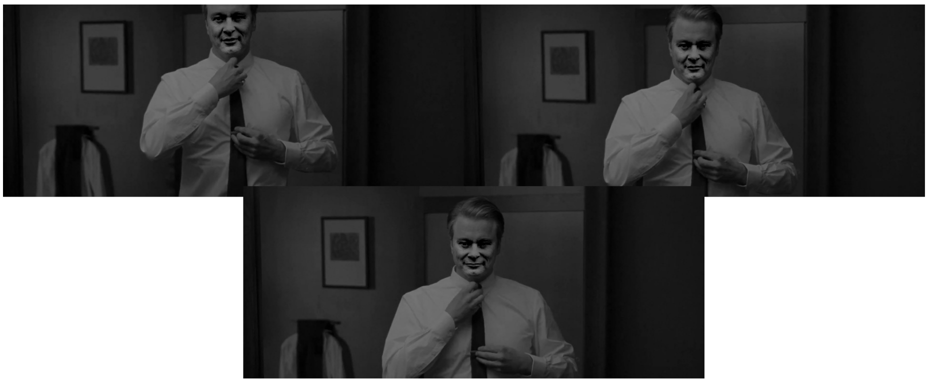


Figure 4.6: The anonymized frames after the original video are inpainted with the anonymized face.

Taking the experiment one step further and using a video where multiple subjects are shown, not necessarily in the same frame but in different frames to see how optical flow manages to handle the transition of inpainting smoothly. The verification process extracted two unique faces shown in **Figure 4.7** below.

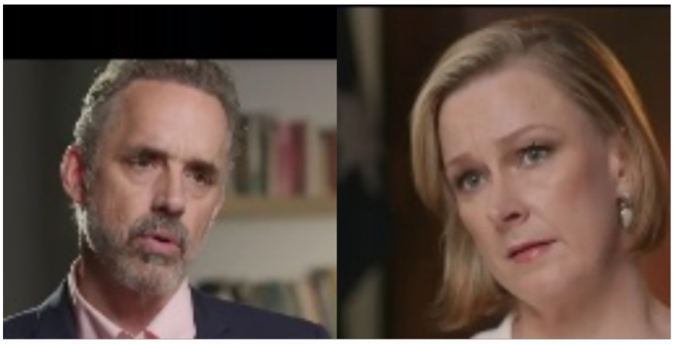


Figure 4.7: The unique faces extracted from the original video [19].

These unique faces are take by the anonymization network and produce the two anonymized faces shown in **Figure 4.8** below.

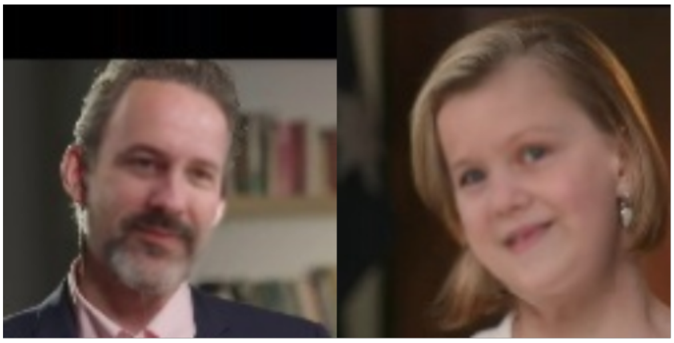


Figure 4.8: The anonymized version of unique faces.

Finally, the unique faces and the anonymized faces are verified and inpainted on top of the original video to produce a video with two anonymized people with smooth movement and transition while keeping an anonymized unique face for each of them without mixing the faces.

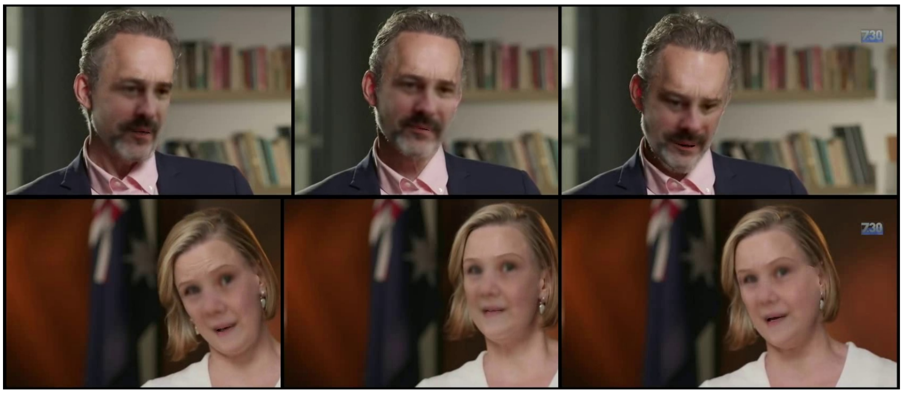


Figure 4.9: The final results of the inpainting **A.2**.

Figure 4.9 above shows what the final results look like after the inpainting process is done.

Throughout all of the tests, one thing remains consistent: the preservation of the newly generated faces. Following the framework's architecture, the triangular inpainting for a face occurs only when an anonymised face is tagged as the corresponding face for the detected face. If there is no anonymised face at all, then the operation for that specific face is skipped. However, anonymising and tagging faces occurs before the inpainting process even starts. Thus, if triangular inpainting is not performed on a face, the process beforehand failed to detect the said face appearing in the frame, which resulted in it not having an anonymised version. As mentioned earlier, the framework uses a *dlib* facial detector as the primary detection method. The *dlib* detector is exemplary, but it is not the best, and on many occasions, it has skipped some faces when performing the detection process. That is why we performed tests using different detectors, mainly *CNN* detector and *Retina* detectors.

File Name	Dlib	CNN	Retina
American Beauty-01500.avi	1512	1710	1674
As Good As It Gets-01400.avi	1192	1349	1320
Big Fish - 00060.avi	126	142	139
Forrest Gump-01805.avi	235	260	266
Big Fish-00674.avi	1911	2161	2115

Table 4.3: Number Of Faces Detected By Different Detection Methods.

Table 4.3 shows an estimate of how many appearing faces in each video are detected by the different detectors. The number represents the number of faces and not the number of people. For example, if a video is 10 seconds long with a framerate of 25 fps and a single person showing throughout the entirity of the video then the number of detected faces would be 250 faces. From the table it is easy to tell that as a detector, *CNN* is the more superior detector. However, the framework uses *dlib* detector instead because of how much time the *CNN* detection process takes especially since the framework requires performing the detection mutiple times to ensure verify the existense of three versions of the face. The *Retina* detector is much faster than the *CNN* detector, however, it was not designed to output the same object type that the framework requires which render it automatically useless unless the framework was redisgned to accomodate for the *Retina* detector.

4.3 Preserving Age and Gender in Inpainted Videos

One of the hypotheses presented in this thesis is *Using triangular inpainting ensures the preservation of age and gender without the need for different neural networks or preservation models.* The experiment is divided into two stages, both of which use the DeepFace network described in **Section 2.1.2**. The stages of this experiment occur once before the inpainting process and once after the inpainting process. In a singular-image situation and after the unique faces from the image are extracted, the faces go through the *DeepFace* algorithm, where the algorithm will predict the age and gender of the people's faces. The data is collected and compared to the same process data that takes the anonymised and inpainted version of the faces. When it comes to the videos, the process is the same. The difference would be that each person in every frame needs to be checked before and after anonymisation and inpainting. The final results should be a table showing the age and gender prediction of the people after the inpainting and before the inpainting. The comparison between the two will show how much of the original age and gender the triangular inpainting preserves.

The *DeepFace* algorithm when used on the second video provided in the previous in **Subsection 4.1** gives the following results for subject one which is a male in his 50s:

Attribute	**Original Face**	**Anonymized Face**	**Inpainted Image**
Age	94% within 5 years	72% within 5 years	89% within 5 years
Gender	100%	60%	90%

Table 4.4: Age & Gender Prediction For The Anonymized Male.

Subject two on the other hand is a female in her late 40s, and the results for that are in the figure above.

Attribute	**Original Face**	**Anonymized Face**	**Inpainted Image**
Age	90% within 5 years	50% within 5 years	85% within 5 years
Gender	96%	69%	83%

Table 4.5: Age & Gender Prediction For The Anonymised Female.

As shown in the previous table, the gender prediction for the second subject is less accurate for both the original and the inpainted versions. that is simply because subject 1 had a beard, making it easier for the network to assume that the subject was male.

These results show that anonymisation networks like *CIAGAN* and *DeepPrivacy* do not consider age and gender too much when performing their work as their main concern is the generation of new faces. On the other hand, triangular inpainting concerns itself with reshaping faces in the image of other faces, which can help preserve crucial information like age and gender. It needs to be said that on the anonymisation of *CIAGAN* which takes less notice of age and gender compared to *DeepPrivacy* which boasts up to 90% age accuracy in some situations and up to 93% accuracy in gender preservation. The biggest problem with it is that its accuracy changes drastically based on the age group of the anonymised subject.

Furthermore, to test this hypothesis even further, we take a large number of images from the *CelebA* dataset and anonymise them using CIAGAN and *DeepPrivacy* separately and see how the in-built masking system in these frameworks works compared to an inpainted version of the anonymised faces. The first comparison is a gender comparison.

Method	Male	Female	Total Estimation
CIAGAN	73%	66%	69%
DeepPrivacy	80%	74%	77%
Triangular Inpainting	86%	79%	82%

Table 4.6: Gender Estimation Comparison Between *CIAGAN*, *DeepPrivacy*, and Triangular Inpainting.

From the first table we can assume that based on the use of the *CelebA* dataset, Triangular inpainting is more superior in preserving the gender of subjects when applying the anonymized face on top of the original subject.

Method	Male	Female	Total Estimation
CIAGAN	40% within 5 years	46% within 5 years	43% within 5 years
DeepPrivacy	44% within 5 years	48% within 5 years	46% within 5 years
Triangular Inpainting	59% within 5 years	63% within 5 years	61% within 5 years

Table 4.7: Age Gender For The Anonymized Female.

The age estimation table above also shows that triangular inpainting improves the estimation quality for age. However, the estimation differs based on the dataset as the triangular inpainting is not a trainable network that can have absolute results. Furthermore, the estimation was based on a five-year cap since it is difficult for all age prediction networks to estimate the exact age. Thus, the five-year cap's choice helps show what method preserves age as close as possible.

4.4 Discussion

The results from the previous sections indicate that the system can take any image containing one or several people and correctly anonymise the faces and inpaint them. The system is also in line with the hypothesis of preserving anonymised faces throughout a complete video. This is thanks to the verification system that qualifies faces appearing in a video as unique and non-unique faces, which helps keep track of what anonymised face belongs to what original human subject. Furthermore, the data from the previous section suggests that using the triangular inpainting method helps preserve the original age and gender of the subjects appearing in videos. These data assure us that there is no real need to implement a system with the sole purpose of preserving age and gender, which would also slow the process of the system.

However, there are some limitations to the system. When some of the tests are performed on a video where several people appear in the same frame, the faces inpainted become unhinged where they start to move a lot and randomly. That occurs because of how the optical flow system works, as it can only work on one face at a time. Having multiple faces appear at once makes optical flow unable to keep track of what face it should apply itself on, which results in it shutting itself down. On the other hand, even if a single face appears in the video. If the face moves in a very random pattern, then optical flow fails to follow the face as the velocity of the facial movement between frames is less consistent and more random.

Nevertheless, the system is able and, as expected and planned to detect faces in videos, anonymise these faces, inpaint them and output a result which contains the subjects anonymised with preserved and consistent anonymised faces as well as the anonymised faces preserving the correct age and gender of the original subjects.

More results can be seen in **Section B**.

Chapter 5

Conclusion

5.1 Conclusion

In this thesis, triangular inpainting was introduced in combination with the optical flow as a method to achieve the goal described in **Section 1.2**. We found that triangular inpainting can help preserve age and gender in a much easier way than using a separate neural network for preserving age and gender. Furthermore, triangular inpainting ensures that the generated faces used to mask the original faces are always consistent. This finding considerably increased the quality of our results as it made the anonymised subjects more realistic and smooth without the extra work.

We introduced a verification system that extracted all the faces appearing in a video and anonymised them separately before inpainting them. This verification system connects every unique face that appears in a video with its corresponding anonymised face, which helps the inpainting system by giving it the correct anonymised face to mask over the original whenever the inpainting system needs it. Triangular inpainting divides both the original and anonymised faces into triangles and reshapes the triangles of the anonymised face into the shape of the corresponding triangles of the original face. After the process of triangular inpainting, the final results show that the faces that are inpainten are 100% consistent all the time. That means that the people appearing in the video are not inpainted with a different face every frame. Furthermore, the gender is preserved and is predicted correct **79% - 86%** of the time, while age is preserved and is predicted within five years of the correct age **59% - 63%** of the time.

Overall, through experiments and testing, we show that our approach, which is a combination of multiple *DNNs*, can solve the problems presented in the thesis and thus reach satisfying results in correlation with the goal presented at the start of this thesis.

5.2 Future Work

Even after achieving the goals set for this thesis, there is still much more to do to create a system able to correctly and realistically anonymise subjects in videos. One of the things that need to be worked on is the verification system. The current system takes video frames first, then extract the unique faces from them and anonymises these faces before the inpainting process starts. A good structure could be implementing a database system where the unique faces and their corresponding anonymised versions are stored. This would expand the system's horizon, making the verifying process faster. If a face exists in the database, there would be no need to extract it and confirm its uniqueness. On the other hand, the system could anonymise specific people. The system could be fed a database containing people and their anonymised faces and force the system only to detect the existence of these people in any video or image and inpaint the corresponding anonymised face. An example would be a school using the system only to anonymise students appearing in their CCTV footage.

Furthermore, another thing that could be worked on is the detection system. Sometimes the inpainted face is placed incorrectly on the original one, which would be the detection system. Different poses make the detection weaker sometimes where it fails to detect all the points of the face, resulting in the anonymised face not knowing all the edges or regions of the face.

Increasing the quality of the inpainted faces is a goal that should be achieved in the future. The current process of extracting faces and performing several operations on them before inpainting them results in the image quality of the inpainted face is lower than the rest of the image. A solution that could be tested is using a resolution enhancing neural network that can be applied to the anonymised faces right before they get inpainted. Of course, this would result in slower performance, so it should be worked on in a balancing manner to ensure good results in both speed and quality.

A final significant implementation that could be added is a layered system that can ensure that some of the problems discussed in **Section 4.4** could be solved. The problem would be the optical flow problem that appears when a scene has several people in each frame. Optical flow work in a way that detects a face appearing in a video and ensures the smooth movement and transition of said face. The problem is that when there are several faces, optical flow fails to focus on one, which makes it unable to perform its duties. The solution would then be a layered system that divided a frame into several layers where each layer contains one and only one face. This would ensure that optical flow can only apply itself to a single face at a time.

Bibliography

[1] *A Single Man (2009) - 'Becoming George' scene [1080] - YouTube*. https://www.youtube.com/watch?v=SicnnTl9IW4&ab_channel=ScreenThemes. (Accessed on 05/29/2022).

[2] *A World With a Billion Cameras Watching You Is Just Around the Corner - WSJ*. https://www.wsj.com/articles/a-billion-surveillance-cameras-forecast-to-be-watching-within-two-years-11575565402. (Accessed on 03/20/2022).

[3] Athira Babu, Shruti Nair, and K Sreekumar. "Driver's Drowsiness Detection System Using Dlib HOG." In: *Ubiquitous Intelligent Systems*. Springer, 2022, pp. 219–229.

[4] John L Barron, David J Fleet, and Steven S Beauchemin. "Performance of optical flow techniques." In: *International journal of computer vision* 12.1 (1994), pp. 43–77.

[5] Steven S Beauchemin and John L Barron. "The computation of optical flow." In: *ACM computing surveys (CSUR)* 27.3 (1995), pp. 433–466.

[6] Marcelo Bertalmio et al. "Image inpainting." In: *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. 2000, pp. 417–424.

[7] Samarth Brahmbhatt. "Affine and Perspective Transformations and Their Applications to Image Panoramas." In: *Practical OpenCV*. Springer, 2013, pp. 155–172.

[8] *CelebA Dataset*. https://mmlab.ie.cuhk.edu.hk/projects/CelebA.html. (Accessed on 11/17/2021).

[9] Paolo Cignoni et al. "Parallel 3d delaunay triangulation." In: *Computer Graphics Forum*. Vol. 12. 3. Wiley Online Library. 1993, pp. 129–142.

[10] *Face Detection: What Is It and How Does This Tech Work? — RecFaces*. https://recfaces.com/articles/what-is-face-detection. (Accessed on 03/31/2022).

[11] *GitHub - YuvalNirkin/fsgan: FSGAN - Official PyTorch Implementation*. https://github.com/YuvalNirkin/fsgan. (Accessed on 04/29/2022).

[12] Berthold KP Horn and Brian G Schunck. “Determining optical flow.” In: *Artificial intelligence* 17.1-3 (1981), pp. 185–203.

[13] *How do you split a video into frames and merge frames to make a video*. https://quick-adviser.com/how-do-you-split-a-video-into-frames-in-python/. (Accessed on 04/26/2022).

[14] Joseph Howse. *OpenCV computer vision with python*. Packt Publishing Birmingham, 2013.

[15] Håkon Hukkelås, Rudolf Mester, and Frank Lindseth. “Deepprivacy: A generative adversarial network for face anonymization.” In: *International Symposium on Visual Computing*. Springer. 2019, pp. 565–578.

[16] Emilija Jasinskaite et al. “DP-ATT: Combining DeepPrivacy with AttGAN to preserve gender and age in de-identified CCTV footage.” In: *Proceedings of the 35th International Conference on Industrial, Engineering & Other Applications of Applied Intelligent Systems*. 2022.

[17] *Jimmy answering questions | Taken by Beatrice Murch (blmurch. . . | Flickr*. https://www.flickr.com/photos/41749772@N06/3857644058. (Accessed on 03/31/2022).

[18] Youngjoo Jo and Jongyoul Park. “Sc-fegan: Face editing generative adversarial network with user’s sketch and color.” In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 1745–1753.

[19] *Jordan Peterson: Free Speech & the Right to Offend - YouTube*. https://www.youtube.com/watch?v=44pERGAaKHw&t=98s&ab_channel=ABCLibrarySales. (Accessed on 05/20/2022).

[20] Ashu Kumar, Amandeep Kaur, and Munish Kumar. “Face detection techniques: a review.” In: *Artificial Intelligence Review* 52.2 (2019), pp. 927–948.

[21] Ivan Laptev et al. “Learning Realistic Human Actions from Movies.” In: *IEEE Conference on Computer Vision & Pattern Recognition*. 2008.

[22] Maxim Maximov, Ismail Elezi, and Laura Leal-Taixé. “Ciagan: Conditional identity anonymization generative adversarial networks.” In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 5447–5456.

[23] Yuval Nirkin, Yosi Keller, and Tal Hassner. “Fsgan: Subject agnostic face swapping and reenactment.” In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 7184–7193.

[24] *Optical flow - Wikipedia*. https://en.m.wikipedia.org/wiki/Optical_flow. (Accessed on 05/04/2022).

[25] *Peeling back the layers: Painting of murdered Renaissance princess revealed beneath layers of paint added centuries later to make her face conform to Victorian beauty ideals | Daily Mail Online*. https://www.dailymail.co.uk/news/article-2671679/Peeling-layers-Painting-murdered-Renaissance-princess-revealed-beneath-layers-paint-added-centuries-later-make-face-conform-Victorian-beauty-ideals.html. (Accessed on 04/16/2022).

[26] Patrick Pérez, Michel Gangnet, and Andrew Blake. "Poisson image editing." In: *ACM SIGGRAPH 2003 Papers*. 2003, pp. 313–318.

[27] Zhongzheng Ren, Yong Jae Lee, and Michael S Ryoo. "Learning to anonymize faces for privacy preserving action detection." In: *Proceedings of the european conference on computer vision (ECCV)*. 2018, pp. 620–636.

[28] Christos Sagonas et al. "300 faces in-the-wild challenge: The first facial landmark localization challenge." In: *Proceedings of the IEEE international conference on computer vision workshops*. 2013, pp. 397–403.

[29] Sefik Ilkin Serengil and Alper Ozpinar. "HyperExtended LightFace: A Facial Attribute Analysis Framework." In: *2021 International Conference on Engineering and Emerging Technologies (ICEET)*. IEEE. 2021, pp. 1–4.

[30] Sefik Ilkin Serengil and Alper Ozpinar. "Lightface: A hybrid deep face recognition framework." In: *2020 Innovations in Intelligent Systems and Applications Conference (ASYU)*. IEEE. 2020, pp. 1–5.

[31] *What is Face Detection and How Does It Work?* https://www.techtarget.com/searchenterpriseai/definition/face-detection. (Accessed on 03/31/2022).

[32] Xizhou Zhu et al. "Deep feature flow for video recognition." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 2349–2358.

Appendix A

Data

A.1 Video Data

Input Video 1

Input Video 2

A.2 Video Results

Results Video 1

Results Video 2

A.3 Code

Source Code

Appendix B

Extra Results

Below are more results of the work. The figures show four versions of frames taken from the dataset described in **Section 3.1.1**. The versions from left to right represent the original, *CIAGAN* inpainting, *DeepPrivacy* inpainting, and Triangular Inpainting. The results show how lacking *CIAGAN* is in the realistic look, age, gender, and skin colour in every figure presented. Furthermore, Both DeepPrivacy and triangular inpainting do great work as they inpaint correctly and try their best to preserve the age and gender of the subject. However, the thesis focuses on the preservation of faces in videos. So, in this situation, the faces *DeepPrivacy* shows in the results are not kept by the network when anonymising the same person in the next frame. On the other hand, triangular inpainting always makes sure to use the same anonymised face for the subject in every single frame.

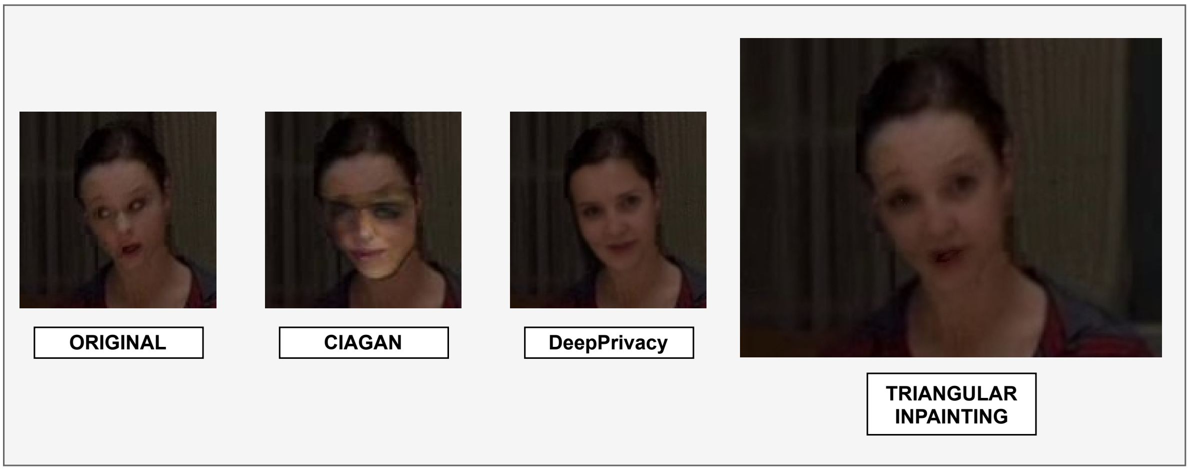


Figure B.1: Comparing Triangular Inpainting Results To CIAGAN and DeepPrivacy.

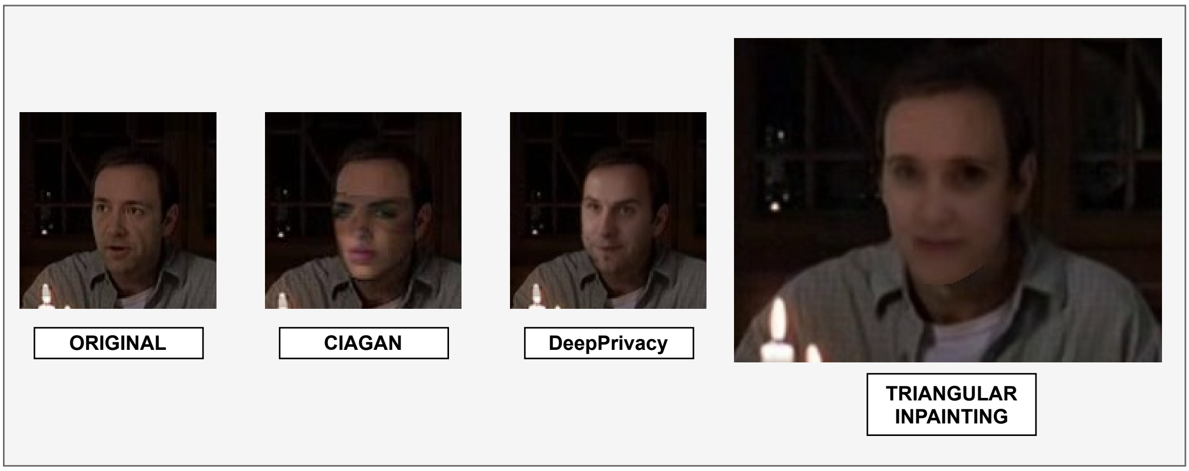


Figure B.2: Comparing Triangular Inpainting Results To CIAGAN and DeepPrivacy.

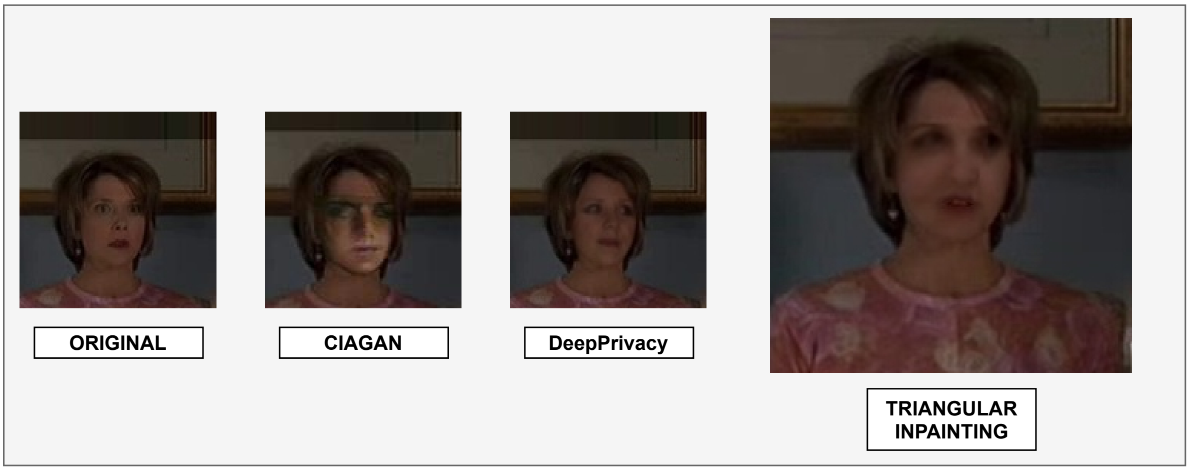


Figure B.3: Comparing Triangular Inpainting Results To CIAGAN and DeepPrivacy.

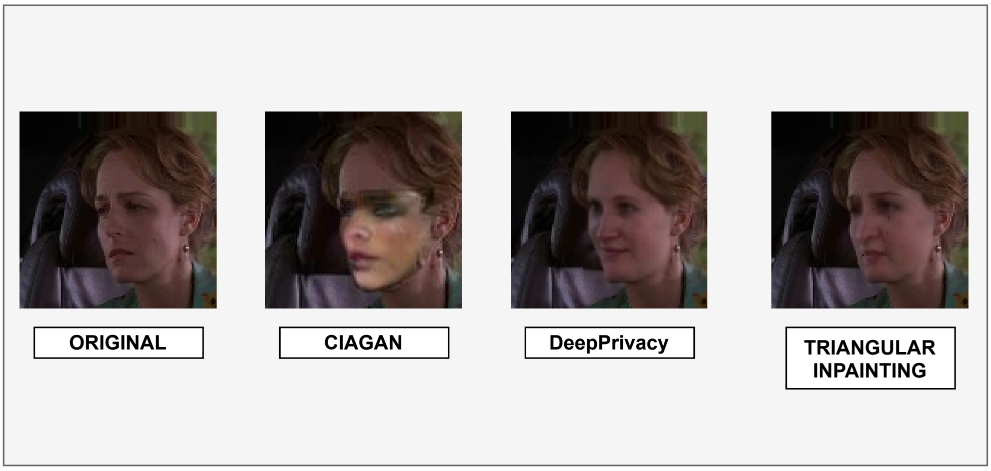


Figure B.4: Comparing Triangular Inpainting Results To CIAGAN and DeepPrivacy.

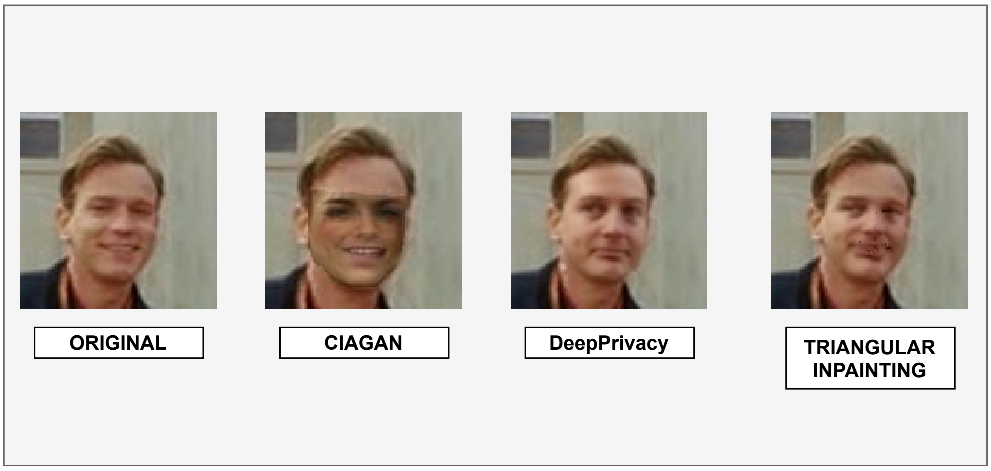


Figure B.5: Comparing Triangular Inpainting Results To CIAGAN and DeepPrivacy.

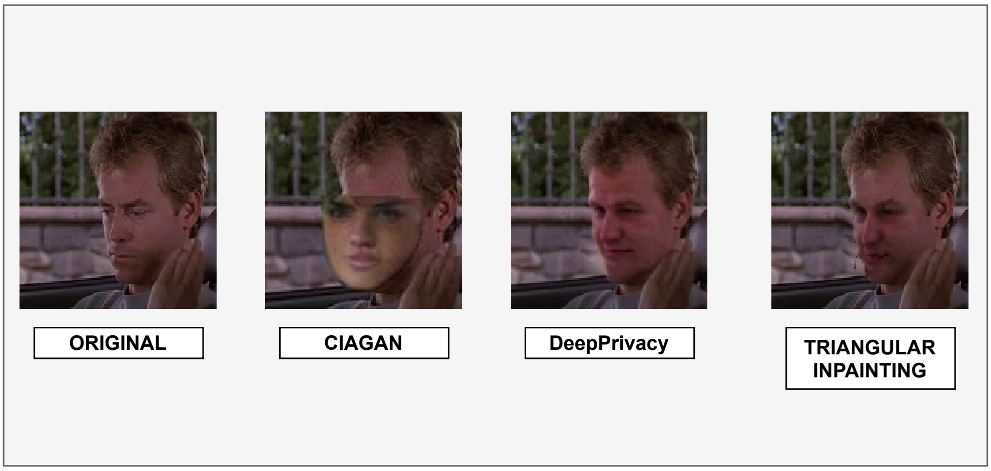


Figure B.6: Comparing Triangular Inpainting Results To CIAGAN and DeepPrivacy.

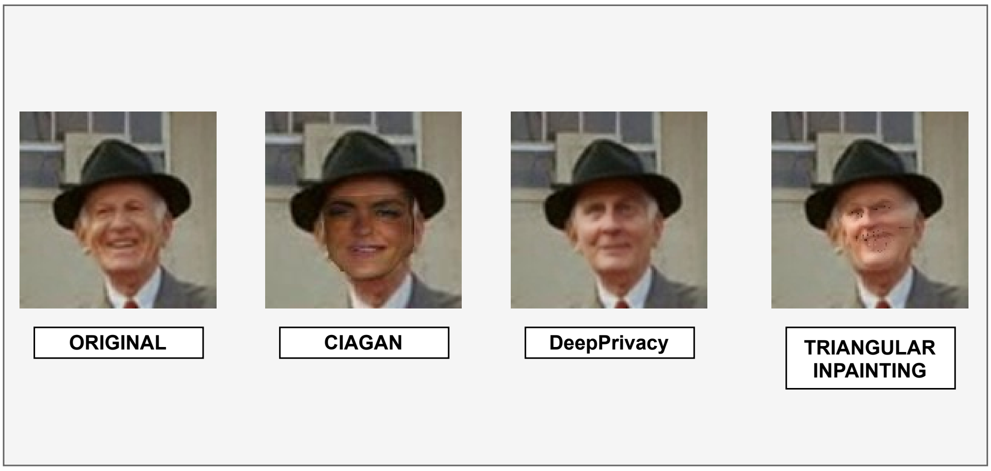


Figure B.7: Comparing Triangular Inpainting Results To CIAGAN and DeepPrivacy.

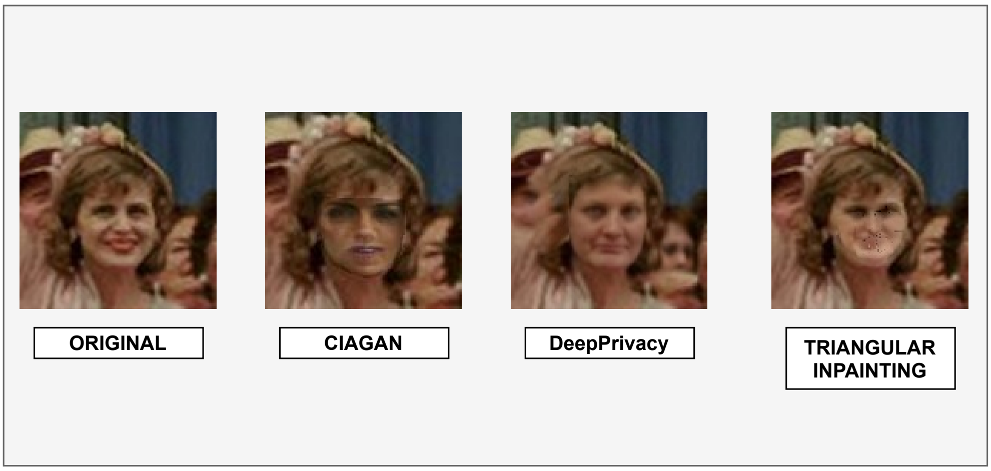


Figure B.8: Comparing Triangular Inpainting Results To CIAGAN and DeepPrivacy.

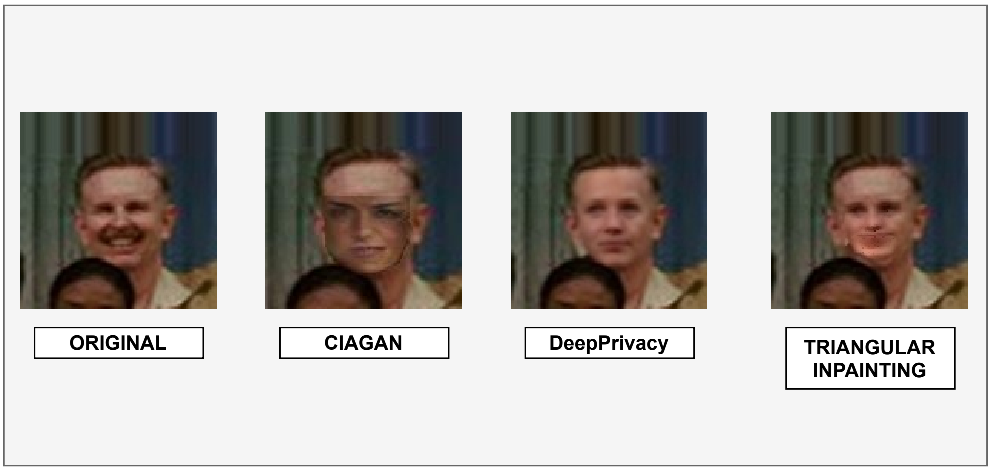


Figure B.9: Comparing Triangular Inpainting Results To CIAGAN and DeepPrivacy.

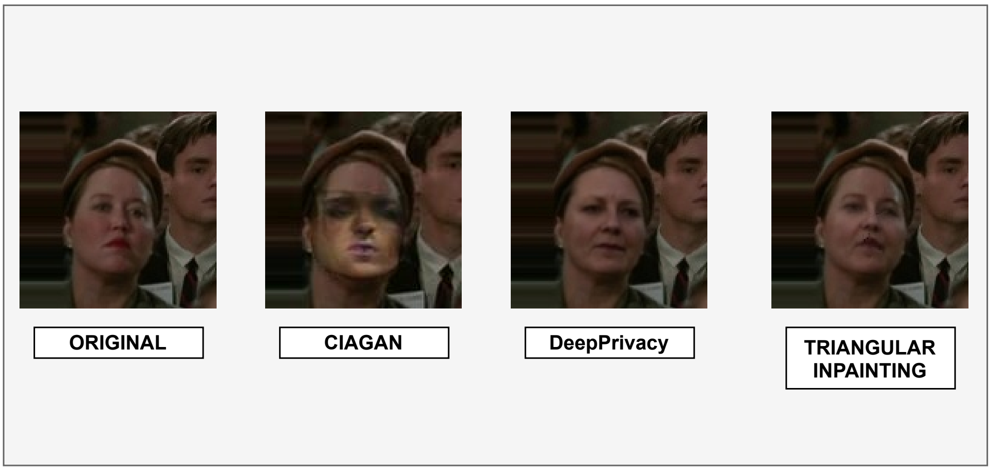


Figure B.10: Comparing Triangular Inpainting Results To CIAGAN and DeepPrivacy.

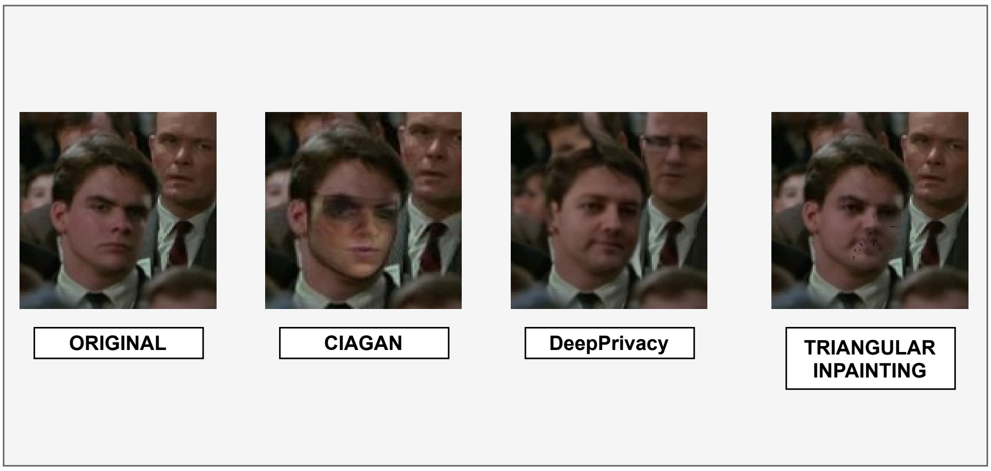


Figure B.11: Comparing Triangular Inpainting Results To CIAGAN and DeepPrivacy.

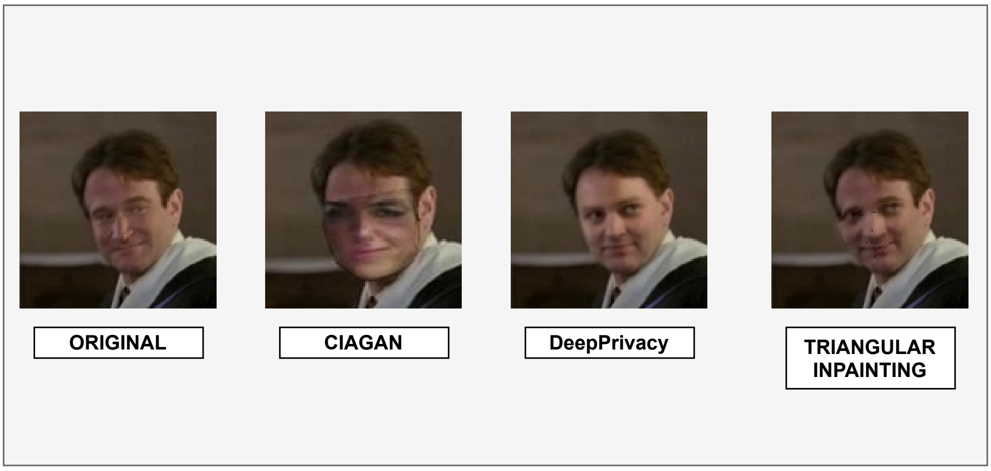


Figure B.12: Comparing Triangular Inpainting Results To CIAGAN and DeepPrivacy.