Bilkent University

Department of Computer Engineering

CS 491: Senior Design Project

Fakenstein

Project Specifications Report

Group Members:

Yusuf Ardahan Doğru
Atakan Dönmez
Öykü Irmak Hatipoğlu
Elif Kurtay
Cansu Moran

Supervisor: Dr. Selim Aksoy
Innovation Expert: Adnan Erdursun
Jury Members: Dr. Shervin Arashloo and Dr. Hamdi Dibeklioğlu

Progress Report October 11, 2021

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Senior Design Project course CS491/2.

Contents

1. Introduction

The concerns about the violation of privacy have become more and more prominent in human lives with the ever growing internet. Every day, people are photographed without their consent and appear in the photographs that are uploaded online. On average, an American is caught on camera 75 times a day without being aware of it [1]. This poses a security threat for the vulnerable, for example children. Additionally, in social media celebrities are starting to put emojis where their children's faces are in the posts they share in order to protect the child's privacy. In social events when a picture is going to be taken, there are always a few people who do not want to appear in the photograph. Therefore, before taking a photograph, permission must be sought from all participants. With more widespread use of the internet from day to day, personal data protection laws are becoming stricter. When taking pictures, people have to be more sensitive about the privacy of the data holders. For example, from Fig. 1, it can be seen that Google Street View blurs the faces of people appearing in photographs in order to protect their privacy [2]. Apart from blurring the

Fig. 1: People with blurred faces in the Google Street View of Bilkent.

images, another method to protect the privacy rights of the individuals that appear in the photograph is to remove them using external tools such as Adobe Photoshop. However, removing people and filling the remaining space with background is a task that requires advanced knowledge in Adobe Photoshop and would take a significant amount of time and effort. A more user-friendly alternative to remove unwanted people from the photograph is an image manipulation tool named Inpaint designed

for the purpose of image restoration [3]. However, this tool requires the user to carefully paint each object to be removed. If the painting is not precise enough or if the photograph has a complicated background, then the Inpaint tool outputs unrealistic photographs. A complicated background is considered to be a background that does not have consistent lines or has different patterns in close areas.

Our proposed phone application, Fakenstein, aims to realistically replace the faces of unknown people with artificially generated faces in order to protect their anonymity. Unlike Inpaint or Adobe Photoshop, Fakenstein intends to keep user interaction as little as possible while outputting a realistic looking photograph. Today, the style-based GAN algorithms can produce extremely realistic artificially generated faces that do not exist. The examples of such faces can be examined from the website where each time the page is refreshed, the website will display a generated face [4]. For the task of replacing faces in a picture, the guidance of the infamous "deep fakes" can be used. Mostly originated from the GitHub repository "DeepFaceLab", deep fakes are used to replace faces in videos to generate realistic fake videos, for example, it is possible to replace Iron Man's face with Tom Cruise's and produce a video that looks like Tom Cruise is the actual actor playing Iron Man. Using a similar method we aim to obtain seamlessly replaced faces.

This report will start with the description of our project, Fakenstein. Then, the project's constraints will be discussed under various sections such as economic, political, and ethical. After that, professional and ethical issues regarding this project will be examined. Later, the requirements of this project will be listed under functional and non-functional requirements.

1.1. Description

Fakenstein aims to produce realistic photos by replacing people's faces in the background with artificially generated faces in order to preserve people's privacy. Fakenstein will be developed as a cross-platform application and will be available as an Android and desktop application. Our application can be used by uploading a photograph from the gallery. Our algorithm will perform a face detection using Machine Learning models and find the identifiable faces on the background of the photograph. Our main goal is to make the application as automatic as possible. However, if the face detection algorithm misses some faces or wrongly guesses the main people on the picture as background, the user will be able to manually edit which faces to be replaced. After the faces are selected, artificially generated faces will replace real human faces using the deepfake method to perform realistic relocation. In order to preserve the nature of the original photograph, the genders, skin colors, and the ages of people will be taken into consideration. This will be

possible by performing classification on the detected faces in terms of their sex (man, woman), skin color (fair or dark skin), and age group (children, adult, elderly). After that, the corresponding generated faces will be used according to the classification results. Again, if the face generated will not seem to be suitable, the user will have the capability to change the artificial face with another one by specifying its qualities. Additionally, another point to increase the realisticity is to determine the orientation of the face by using a head pose estimation algorithm and utilizing an appropriately generated face.

Our main target audience is anyone who uses social media and publishes photographs online. From event photography to news and blog posts, instead of trying to blur, cover, or remove people from photographs, the publishers can easily use Fakenstein to obtain realistic looking photographs that do not violate the privacy of individuals.

1.2. Constraints

1.2.1. Economic Constraints

- Our project will use IDEs that are available for free through educational packages and a GitHub hosted website, which is also free.
- The libraries to be used in computer vision and deep learning tasks are open-source.
- Deep learning requires powerful GPUs. Google Colab provides free GPUs for cloud computing for a limited time.
- Open-source datasets will be used for training the models.
- One of the main goals of this app is to help prevent invasion of privacy, therefore the application will not utilise customer data as a source of income.

1.2.2. Social Constraints

- The application will include artificially generated faces that represent all visibly detectable characteristics such as sexes, age groups, and skin colors and replace the faces in the background accordingly.
- The terms and conditions should handle the fact that these generated photos may be persevered as an altered reality.
- Even though the replaced faces are fake, the user may choose to not include people from certain skin colors or sexes. Again, terms and conditions should clearly state that this application should not be used for bigoted purposes.
- In the terms and conditions it should also be stated that the generated fake faces should not be used to deceive people by making them think that the fake faces belong to a real person, for example catfishing.

1.2.3. Political Constraints

- Our application aims to protect Turkish Personal Data Protection Law [6] by using artificially generated faces.
- Our application will be developed regarding the European Union data protection rules [7].

1.2.4. Ethical Constraints

- The application will request access to the user's media on their phone to be able to receive, edit and save photos to their gallery.
- Our application will not use the user's information including the uploaded photographs for any commercial purposes. The information will strictly be used for editing purposes.
- The uploaded photographs will not be stored to an external database or shared with third party applications, they will be stored locally on the user's device.
- As the main purpose of this application is to protect the anonymity of people in the background by using artificially generated faces, the generated faces must be able to mask people in photos completely.

1.2.5. Sustainability Constraints

- A feedback system will be used to collect recommendations and bug reports.
- The application will be maintained regularly for bug fixes.

1.2.6. Implementation Constraints

- Fakenstein will be an Android application for personal use and a desktop application for professional use.
- GitHub will be used as a version control and issue tracking tool. The website of this project will be hosted by GitHub.
- Android Studio IDE will be used to create an Android application, using Java.
- The desktop application will also be written in Java so that it should run at every platform having JVM.
- Desktop application will be developed in Intellij IDE.
- Python will be used to implement computer vision algorithms.
- Pytorch framework will be used for deep learning tasks.
- The website will be created using HTML, CSS, SCSS, and JavaScript.
- Object Oriented Programming (OOP) paradigm will be adopted throughout the project.

1.2.7. Language Constraints

- The application will be written in English as it is a global language.

1.3. Professional and Ethical Issues

1.3.1. Professional Issues

- There will be at least one meeting per week regarding the project's development.
- All decisions will be made by voting among the teammates.
- The workload of the project will be distributed equally regarding the skills of each individual in the team.

1.3.2. Ethical Issues

- Regarding the privacy of the application user, no photos or other forms of personal data will be collected by the application, and the necessary information will be stored on the user's device locally.
- The classification results will be based on the trained model and the dataset. No personal judgement is made while determining the sex, skin color, or age of the faces.

2. Requirements

2.1. Functional Requirements

2.1.1. User Functionality

The user should be able to:

- download the application via Google Play Store to their mobile phone or via the website of the application to their desktop.
- agree or disagree to the terms and conditions.
- open the gallery on his/her device to upload pictures.
- choose either the application's face detections or manually choose any mislabeled faces.
- edit the features of the generated faces if the program does not create suitable faces.
- save the new picture.

- restart the whole process.
- provide feedback for the application in terms of feature requests or bug reports.

2.1.2. System Functionality

The system should be able to:

- open the gallery on the user's device.
- open pictures from the gallery.
- detect all the faces in the picture.
- label the detected faces as the main subject or background.
- enable the user to add or remove faces to be replaced with generated faces.
- classify the detected face’s sex, skin color, and age group.
- determine the pose of the faces.
- generate artificial faces that are compatible with the detected features of the original faces.
- replace original faces with artificial ones as seamlessly as possible.
- save the final image to the user’s device.
- send user feedback to developers.

2.2. Non-Functional Requirements

2.2.1. Usability

- The application should have a simple and self-explanatory interface.
- The language used in the application should be clear and easy to understand.
- The application should enable the user to upload and get their image.

2.2.2. User Friendliness

- The application should not be hard for the user to understand and use.
- The application should include a help or tutorial section to help users adjust to the application.
- The application should be optimized in order not to bore users with long loading and processing time.
- The user should not be required to know about machine learning algorithms that will be used in the application.
- The user interface design may be updated annually according to user feedback to adapt to changes in popularity and to better user experience.

2.2.3. Maintainability

- Any open source or third party software used will be required to have long term support for future maintenance.
- The application should have periodic version updates to use the latest versions of the external modules and third party applications.

2.2.4. Extensibility

- The application should be extensible so that in the future it can be used as a plug-in or be integrated by other applications.
- The application should be open to any upgrades considering there could be new functional/non-functional requirements or user interface changes.

2.2.5. Reliability

- The application will be tested for various cases in terms of features.
- The application should not crash for reasons other than operating system based version differences. The reason for the crash must be logged and the user should be informed about it.
- The application should work for differing sizes, brightness, image types, and image formats such as BMP, JPEG, PNG, WebP, and HEIF.

2.2.6. Accessibility

- The Android version of the application should be made available on the Google Play Store to reach a vast amount of users.
- The desktop version of the application should be available for download on the website of the application.

2.2.7. Portability

- The desktop application will be able to run independently from the operating system via Java Virtual Machine.
- The Android application will be able to run independently from the brand or model of the mobile phone. The only constraint should be the version of Android installed.

2.2.8. Efficiency

- The application should be able to run smoothly in older and newer generations of Android smartphones with Android Marshmallow 6.0 or higher installed.
- The picture upload should not take more than approximately 10 seconds (subject to change).
- The application should provide the user with the output image under approximately 20 seconds (subject to change).

2.2.9. Scalability

- The application is made for daily use, so available platforms and users may change as there are new developments.
- The service we are providing will not be affected by user/customer number as the application will work locally once downloaded.

3. References

[1] R. Eveleth, “How Many Photographs of You Are Out There In the World?,” *The Atlantic*, 03-Nov-2015. [Online]. Available: https://www.theatlantic.com/technology/archive/2015/11/how-many-photographs-of-you-are-out-there-in-the-world/413389/. [Accessed: 08-Oct-2021].

[2] “Contributed street view imagery policy,” *Google*. [Online]. Available: https://www.google.com/intl/en_uk/streetview/policy/. [Accessed: 09-Oct-2021].

[3] “Remove people from photo: The easy way,” *Inpaint*. [Online]. Available: https://theinpaint.com/tutorials/pc/how-to-remove-unwanted-people-from-photo. [Accessed: 09-Oct-2021].

[4] “This person does not exist,” *This Person Does Not Exist*. [Online]. Available: https://thispersondoesnotexist.com/. [Accessed: 09-Oct-2021].

[5] Iperov, “Iperov/deepfacelab: Deepfacelab is the leading software for creating deepfakes.,” *GitHub*. [Online]. Available: https://github.com/iperov/DeepFaceLab. [Accessed: 09-Oct-2021].

[6] “Turkish Personal Data Protection Law No. 6698 • Kişisel Verilerin korunması mevzuatı • KVKP,” *Kişisel Verilerin Korunması Platformu (KVKP)*. [Online]. Available: https://www.kisiselverilerinkorunmasi.org/kanunu-ingilizce-ceviri/. [Accessed: 10-Oct-2021].

[7] “Data Protection and online privacy,” *Your Europe*. [Online]. Available: https://europa.eu/youreurope/citizens/consumers/internet-telecoms/data-protection-online-privacy/index_en.htm. [Accessed: 11-Oct-2021].