Institute of System Science, National University of Singapore

#### 

**Intelligent Software Systems**

**Project Report**

**Photo Face Sorter & Anonymizer**

Team members

Ankeit Taksh (A0213496B)

Sindhuja Kumaran (A0213537J)

Zhu Weiwei (A0213545L)

# CONTENTS

[**CONTENTS**](#_7lkdtmf2c14j) **2**

[**1. Executive Summary**](#_yfr1yrdmozal) **3**

[**2. Problem Description**](#_hoqd8uhleqwg) **3**

[2.1 Project Objective](#_p4s2moak8d8e) 3

[**3. Proposed Solution**](#_xy2kzpfgme35) **4**

[3.1 System Design](#_h3148fu7p8bj) 4

[3.2 Data Model](#_8f605h8egl7u) 5

[3.3 System Implementation](#_ba6lx8l9527a) 6

[3.3.1 Online system](#_9o9ig9mjjga8) 6

[3.3.2 Clustering](#_fp5m2bhxbnok) 6

[3.3.3 Face Recognition](#_v2m3pmlgis70) 6

[3.4 Technology and Usage](#_talpbtiwwdzt) 7

[4. Dataset](#_o9y5kvy1p5k6) **7**

[**5. Results**](#_ii0ft6ljm94b) **7**

[**6. Limitations and Future Enhancements**](#_m4ihg1p8rcad) **11**

[**7. Deployment**](#_c2lnyseg6lag) **11**

[**8. Appendix**](#_7c8oky1ir7vz) **12**

[8.1 Appendix A: System Functions Mapping against ISAs](#_w3pfp9jirynf) 12

[8.2 Appendix B: Installation Guide](#_2a1oocfjg9yv) 12

[8.2.1 Prerequisites](#_t4ywwskez6ie) 12

[8.2.2 Application Preparing:](#_ojqkwvyjczhh) 12

[8.2.3 Environment Setup:](#_ec6tfltcursj) 12

[8.2.4 Application Running:](#_e201ixjej51) 13

[8.3 Appendix C: Team Contribution](#_rswlmwnkdnmc) 14

[8.4 Appendix D: Individual Reflection](#_tkqapno098hw) 15

[8.4.1 Ankeit Taksh](#_55i9fak3y4y4) 15

[7.4.2 Sindhuja Kumaran](#_99afnpiodgwi) 16

[7.4.3 Zhu Weiwei](#_chiab01aevo6) 17

[**8. References**](#_ppnlsk627gzo) **19**

# 

# 1. Executive Summary

In the current digital era, where photographs are shared digitally, there is growing concern for sharing them publicly. Nowadays, there is more potential to misuse someone’s face with deepfakes and face spoofing. Hence in many institutions like kindergartens and schools, they address it with most care to share students’ pictures with concerned parties only.

Presently, the institutions make an effort to not share their student’s' photographs with everyone. They share photos of kids to their respective parents only. However, the parents are still able to view other kids, who are present in those photos, together with their own.

# 2. Problem Description

In institutions like kindergartens, the number of pictures the teachers click every day on each student’s activities is too many. If it’s an individual activity, they make sure to capture only one student in each photo. But with activities involving multiple children, they don’t have the time to capture a single child in each picture. So even though they share the picture of each child with their parents, they end up sharing some pictures of group activities with many children, which ends up as a privacy concern.

The current approach of teachers manually sorting the pictures of each child in the bunch of photos taken every month is very time consuming. The sheer number of pictures taken makes it difficult too. Hence there is always potential for human error resulting in sharing of photographs to unintended recipients and failure to share with the intended recipient.

If this process could be automated, then the teachers or staff doing this repetitive task could use their time in other tasks that require their attention like curriculum planning or paying more attention to needy children.

## 2.1 Project Objective

Our team aims to create an efficient intelligent system that automates the tedious, time-consuming task of sharing particular photographs with concerned parties. In our System, for the sake of prototyping, we have limited the number of faces an end user can upload in his profile. Non-admin users can upload 2 face images to the system. Later our system will try to match and retrieve photos with these 2 faces. Our solution aims to fulfill these objectives:

* Search and retrieve all photographs that contain a child’s face in it, given a bunch of photographs with many different children present in them.
* Mask other children’s faces, who are present in the same photograph along with searched child
* Share the sorted photographs with respective parents.

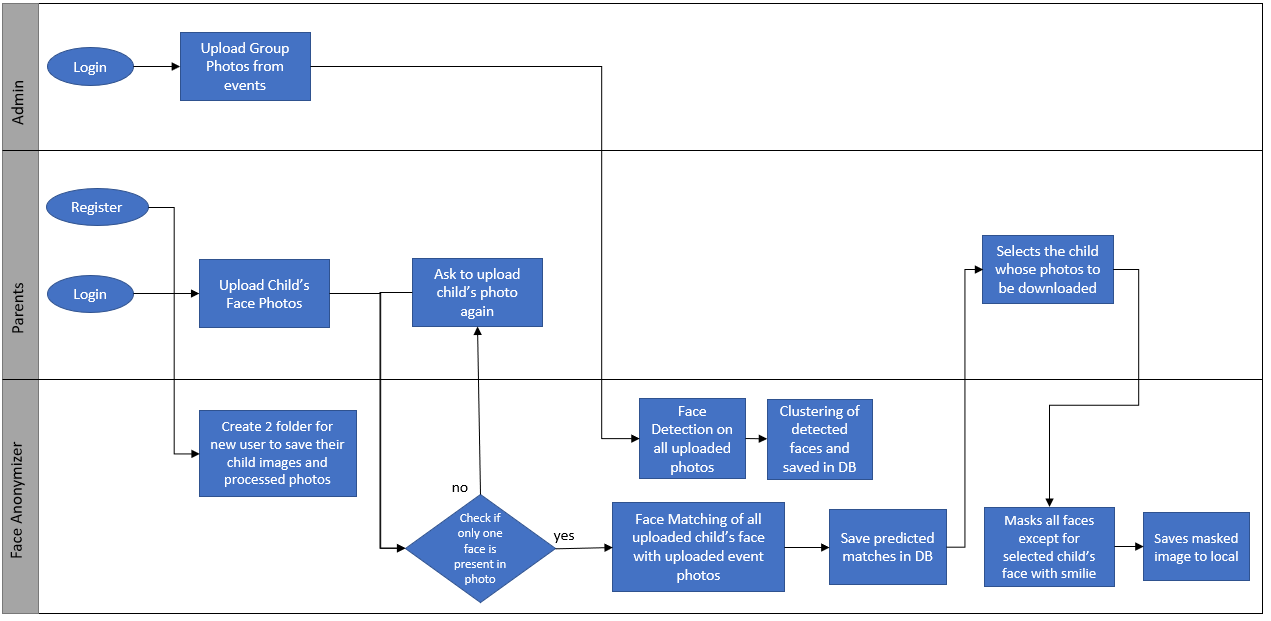
# 3. Proposed Solution

## 3.1 System Design

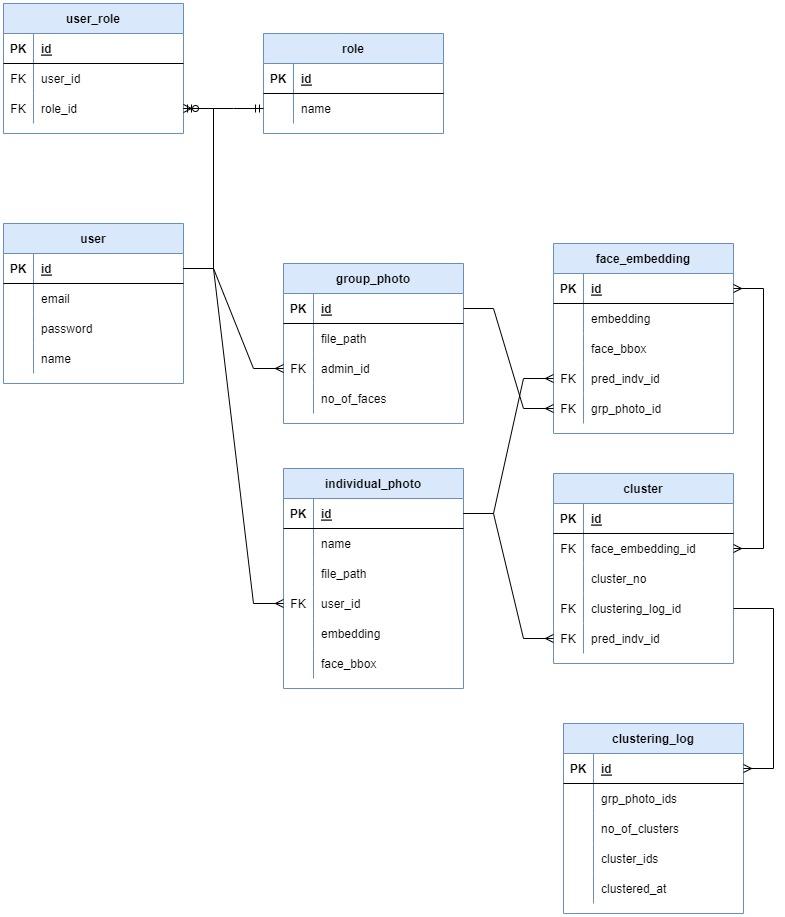
The project consists of two main components.

1. Admin Portal: allow admin to upload the photos in batch.
2. User Portal: allow users to update personal profiles, upload individual photos and mark photos based on choosing individual photos.

Below shows the system design workflow.



## 3.2 Data Model



## 3.3 System Implementation

### 3.3.1 Online system

Online system refers to the application that is deployed at client for production user. It is a web-based application built in python with Flask and JavaScript.

There are two main portals: *admin portal* and *user portal*. In the admin portal, system admin could upload a bunch of group photos. System will utilize a clustering algorithm to group all faces detected in these photos then match against those individual photos. All matched faces will be updated with the matched person id. In the user portal, users need to register an account before using this system. Users cannot upload more than two people's photos. Application will raise an alert and block this user from uploading the same photo If there is no face found. Users can select one or both the faces to filter group photos uploaded by system admin and mark all unmatched faces with a smile image.

### 3.3.2 Clustering

Clustering is an unsupervised technique to deal with large scale data. Since admin could upload lots of images which brings great challenges to classify how many groups are there in those uploaded photos. DBSCAN is a popular density-based data clustering algorithm. Unlike the K-Means algorithm, we don’t need to provide the number of clusters required prior.

We assume that the system admin has uploaded lots of photos, the application will use DBSCAN to group all faces and match face embedding with stored value based on the distance.

### 3.3.3 Face Recognition

Face recognition is a popular way of identifying and confirming individual people using their face. We explored the Face Recognition library, created by Adam Geitgey, which is simple and easy to integrate into our system. We use it to do face recognition, face detection and face makeup. Whenever an admin or a user uploads an image to the system, faces are detected and embedded by local AI using Face Recognition library. Then the bounding boxes and face embeddings are stored into the database.

## 

## 3.4 Technology and Usage

| Technology | Description |
| --- | --- |
| DBSCAN | Density-Based Spatial Clustering of Applications with Noise. Finds core samples of high density and expands clusters from them. Good for data which contains clusters of similar density |
| Face Recognition | Recognize and manipulate faces from Python or from the command line with  the world’s simplest face recognition library. |
| Flask | Flask is used as the web framework to construct the user interface as well as the web API handler for the system. |
| Selenium, Beautifulsoup | Web scraping of Images to collect data |

# 4. Dataset

In this project, we used multiple sources of dataset available publicly on the web from Kaggle [10].

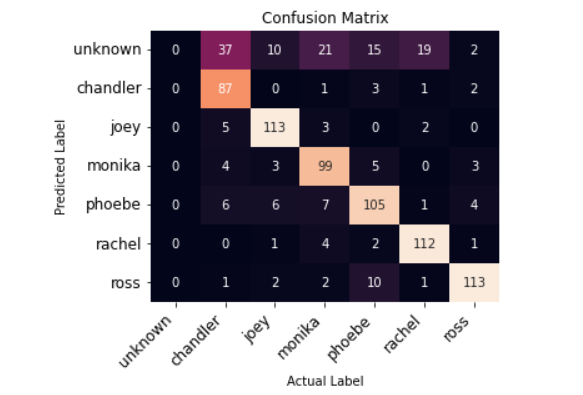
We also created our own dataset by scraping images from Google. We scraped about 500 images of Friends western TV series using *Selenium*. These images mostly contain a group of people in the frame. The scraped images were annotated manually using VoTT and used to test the masking for selected people.

# 5. Results

Although our main Business case of Photo Face Sorter and Anonymizer system is a child-care center, due to privacy concerns, we tested our system using a self-collected dataset of images from the western TV series ‘Friends’. We uploaded these photos to our admin system. Then used the photos of characters from the drama series as the faces photos for matching. From the testing, we noted that the system matched the person even when he has aged. Here are some of the sample predictions from our testing.

| **Chandler** |  |
| --- | --- |
| **Joey** |  |
| **Monika** |  |
| **Phoebe** |  |
| **Rachel** |  |
| **Ross** |  |

The confusion matrix of testing results is below



As we can see, the False positives for each class is lower. So this will help to avoid sharing photos with unintended parties.

# 6. Limitations and Future Enhancements

Ensuring the highest quality of the system is always the mission of the team. However, there are many improvements the team wishes to implement within this project. One observation is that the performance of clustering in the current system is not as good as we expected. Future improvement can work on this area to improve the model to classify faces fast with high accuracy.

The face detection model is not able to detect the faces in profile view or when it is partially occluded.

Face Recognition model can also be further enhanced to identify the face by utilizing the history of face photos uploaded by the users. We would also like to improve the system by helping it identify only when the matched person's face is clearly visible, when he is centered in the picture or when he is not present in a corner.

Application runs in a local virtual environment and could migrate it into a cloud-based instance to improve performance in the future. Functions also can extend to fetch images from any website using TagUI and apply the face marking method to cover faces in need.

# 7. Deployment

We could deploy the system into a cloud-based GPU instance, compared to the CPU instance in our local environment. It will run significantly faster on a GPU enabled machine. But the cost per hour for the GPU instance will be much too high,

Thus we deployed and tested the system all in our local instance, it also achieved a good result.

# 

# 8. Appendix

## 8.1 Appendix A: System Functions Mapping against ISAs

| Main Function | ISS-ISA Module | Skill Learnt |
| --- | --- | --- |
| Face detection, Face recognition and Face Mark | Intelligent Process Automation | Local AI |
| Data Collection | RPA and IPA - Strategy & Management | Image scraping |
| Overall | Software Robots and Best Practices | MVP |

## 8.2 Appendix B: Installation Guide

### 8.2.1 Prerequisites

Web Browsers

* Google Chrome.
* Firefox
* Edge

Software & Program Language

* Python
* Anaconda

### 8.2.2 Application Preparing:

* Download and unzip the file. or
* Clone from repo: <https://github.com/Yunxiai88/facemask.git>

### 8.2.3 Environment Setup:

**We strongly recommend you use Anaconda for python and its dependencies.**

1. Virtual Environment:

i. Create a virtual environment using Anaconda Prompt:

***conda create -n facemask python=3.7***

ii. Activate newly created environment:

***conda activate facemask***

iii. In the virtual environment, go to the project root folder and run below command to install packages:

***pip install -r requirements.txt***

1. If any packages fail to install, try installing individually  
    If there are any errors, try to do this one more time to avoid packages being missed out.
2. Create Tables (optional)

***python create\_table.py***

### 8.2.4 Application Running:

* Execute the command below in the project root folder to start application:

***python main.py --ip 127.0.0.1 --port 8000***

* Open URL: [http://127.0.0.1:8000](http://127.0.0.1:8000/):

(You can also specify your customized IP address and port number by)

***python main.py --ip=\*\*\*\* --port=\*\*\****

## 

## 8.3 Appendix C: Team Contribution

| Official Name | Student ID | Contribution |
| --- | --- | --- |
| Ankeit Taksh | A0213496B | Project Proposal, Data Acquisition, Modeling, System integration Test, Models Performance assessment, Report Writing. |
| Sindhuja Kumaran | A0213537J | System Architecture Design, Data Modeling, Project development, Data Collection, Model Performance assessment, Report Writing, Video Creation. |
| Zhu Weiwei | A0213545L | System design, Knowledge discovery, Project development, Feature implementation, Report Writing, Video Creation. |

## 

## 8.4 Appendix D: Individual Reflection

### 8.4.1 Ankeit Taksh

**Student ID:A021396B**

The project is a brainchild of **Professor Sam (Gu Zhan)** for a social cause with special credit to **Sindhu** and **Weiwei** from my team for advancing the idea and hence we collaborated to ensure that a solution can be achieved with multiple skills working together.

1. Contribution to the project

My contribution was on research work was on initial conceptualization and further blueprinting via towardsdatascience, medium articles, official tensorflow, and github projects websites. Though the project as a whole cannot be broken down to subsections which elaborates the specific portions for individuals but on my part I was more focussed on the backend python model testing ,deployment and testing part with the different Algorithms as a part of CI/CD pipeline mode.

Aside from product implementation , it coordinated with the team in testing images with different models like Retina face , Face masking , Facial Recognition using CNN and analysis of the outcome was the most important part where we tested various ML techniques .

My team had worked on many existing codes to reuse the modules mostly from Kaggle and github and how we can utilize the existing modules to leverage on our project.

I had assisted the team for image identification with better accuracy using HAAR functions so the team can focus more on the frontend. We tested on keras, tensorflow and analyzed the quality of data output using OpenCV and pre-trained weights.

As well worked with our core project planner **Sindhu** and frontend developer **Weiwei** to provide guidance from devops side to build the solutions in a way where it can be scaled for real production.

The project application thus enabled us to explore various methods of not just encoding and generation with one algorithm but to cross verify results across other algorithms.

This can be implemented in various sections of machine learning.

2. What you have learnt from the project

The project application thus enabled us to explore various methods of not just encoding and generation with one algorithm but to cross verify results across other algorithms. This can be implemented in various sections of machine learning.

Instead of exploiting our existing knowledge we used exploration methods and searched for more ways for bringing RPA knowledge for social welfare.

A very good learning for me was to analyze that use case and its social benefits is more important than using complicated algorithms

3. How you can apply this in future work-related projects

The Project gives us ideas on how to create products which can come in handy for Realtime benefits. Team coordination and social objectives were other great learning. To be specific we can apply the learnt methods of RPA in face recognition based identity verification for business purposes.

The similar methods for RPA and Machine learning face recognition has great application in entrance security of offices and hence I am planning to design the similar use case as an advancement of this project but with retina face in practice.

**Contribution description**

### 7.4.2 Sindhuja Kumaran

**ID: A0213537J**

1. Personal contribution:

For the Intelligent Software Systems practice module, our team built a Photo Face Sorter and Annonymizer to sort the photographs based on the faces in the photograph, retrieve the photographs belonging to a particular person and anonymize all other faces in those pictures.

In this project, my contribution is mainly on

* System design and implementation of face Detection
* Testing feasibility of using rpa to create folders in gdrive
* Data Modeling and Data Collection
* Annotating collected images for face recognition
* Evaluating the performance of the System

2. What you have learnt from the project.

I learnt to utilize the knowledge of developing AI model, to develop and integrate a full stack AI application. I also learnt how we could use our rpa bots to login on behalf of us, into gdrive and make changes in it. Also learnt how I could automate a bot to scrape images from the web and save them in local folders.

Moreover, I learnt how to narrow down features for MVP as near term goals, mid term goals and future goals.

3. How you can apply this in future work-related project

RPA and IPA are very useful tools that I can use in my day-to-day life. They are very useful for repetitive, boring and time-consuming tasks. I will also use them for data collection in my future projects.

Having learnt strategy and management of RPA & IPA, now I know better which processes to automate using RPA.

**Contribution description**

### 7.4.3 Zhu Weiwei

**ID: A0213545L**

1. Contribution to the project

At the very beginning we came up with an idea which was mentioned by the professor in the class, and we decided to use new knowledge to achieve it. After that I engaged in the below phases.

* System design to analyze feasibility of functions.
* Explore available libraries and cloud services which could be used in the project.
* UI design and Data preparation.
* System integration and testing.

In the last phase, my teammates and I work together to finish our report and re-test all functions to make sure all of them work. This project is an achievement of teamwork. All of us spent a lot of time and effort on it.

1. What you have learnt from the project

In phase one, we came out with an MVP and listed out all functions with a score. And pick out top N of those functions.

In phase two, every team member assigned a task to study feasibility of existing services in the market which can be used in the project. Our team works well and decided to use face recognition and clustering techniques to build our project. Thereafter, we tested the accuracy and performance of these services, and found that it can meet our requirements.

We encountered a lot of problems when building this project.

* Data preparation: We spent a lot of time downloading data (using TagUI to download images with certain keywords in search engines like google chrome) and avoiding data privacy policy.
* System integration: Each individual module could work well, but it performs badly when integrated together (more process step needs before feed data into downstream module).

1. How you can apply this in future work-related projects

RPA is a quite useful tool to download resources to our local PC without much manual effort. It also can replace repetitive tasks and allow us to put more effort on valuable work.

Intelligent service can process larger data like images in this project with high accuracy and allow us to optimize it with fine tune parameters. We also learned how to use existing AI cloud services like sentiment analysis, voice recognition and chatbot to enhance project functions.

We also learned project management matters; we can apply MVP knowledge to build startup projects (basic functions and build quickly and avoid wasting money and use agile methodology to add more features).

In the future, we can apply this knowledge in any situation at that time.

# 8. References

1. Ageitgey, (2022 Mar 10). Face Recognition. Retrieved from <https://github.com/ageitgey/face_recognition>

# 

2. Cardano-max, (2022 Apr 04). Find faces in pictures. Retrieved from<https://deepnote.com/@cardano-max/Face-Recognition-1c4ca171-5e58-4d2c-b6a9-7904402a65c0>

3. Mohammed Maheer, (2022 Apr 04). Face Recognition Using ‘face\_recognition’ API. Retrieved from https://medium.com/analytics-vidhya/face-recognition-using-face-recognition-api-e7fa4dcabbc3

4. Vimsky.com, (2022 Apr 04). Python face\_recognition.compare\_faces. Retrieved from https://vimsky.com/zh-tw/examples/detail/python-method-face\_recognition.compare\_faces.html

5. Facile Code, (2022 Apr 04). How to blur faces on images with OpenCV. Retrieved from <https://facilecode.com/how-to-blur-faces/>

6. Geeksforgeeks, (2022 Apr 04). How to Blur Faces in Images using OpenCV in Python. Retrieved fromhttps://www.geeksforgeeks.org/how-to-blur-faces-in-images-using-opencv-in-python/

7. Geeksforgeeks, (2022 Apr 12). ML | Unsupervised Face Clustering Pipeline. Retrieved from <https://www.geeksforgeeks.org/ml-unsupervised-face-clustering-pipeline/>

8. [Adrian Rosebrock](https://pyimagesearch.com/author/adrian/), (July 9, 2018). Face clustering with Python. Retrieved from <https://www.geeksforgeeks.org/ml-unsupervised-face-clustering-pipeline/>

9. VoTT (Visual Object Tagging Tool). Retrieved from <https://github.com/microsoft/VoTT>

10. Kaggle Dataset, (2022 Mar 10), Trump Photos, Retrieved from <https://www.kaggle.com/datasets/mbkinaci/trump-photos>