

Walid Demloj
Kjetil Grosberghaugen
Julian Nyland Skattum
Hans Petter Fauchald Taralrud

Face Image Quality Assessment

Bacheloroppgave i Dataingeniør
Veileder: Seyed Ali Amirshahi
Mai 2021

Walid Demloj
Kjetil Grosberghaugen
Julian Nyland Skattum
Hans Petter Fauchald Taralrud

Face Image Quality Assessment

Bacheloroppgave i Dataingeniør
Veileder: Seyed Ali Amirshahi
Mai 2021

Norges teknisk-naturvitenskapelige universitet
Fakultet for informasjonsteknologi og elektroteknikk
Institutt for datateknologi og informatikk



Kunnskap for en bedre verden

Abstract

The performance of face recognition systems are dependent on the quality of images used for their training and testing. To automatically evaluate the quality of facial images, Face Image Quality Metrics (FIQMs) are used. Such metrics provide an objective score which corresponds to how well the face is visible. In this project we introduce a web application which is able to calculate such objective scores for two different state-of-the-art FIQMs. To evaluate the accuracy of the mentioned FIQMs we collected subjective scores from expert and non-expert observers for three different available datasets of facial images. Further on we also collected a new facial image dataset which we believe is superior to current public datasets. This superiority is with regards to the number of images provided, the type of distortions the images are affected by, and also the nature of the images themselves (images with face covering by masks and images taken at an oblique angle). Our results show that the objective scores calculated by the two FIQMs have a low to moderate correlation with the subjective scores we have collected in our subjective experiments.

Sammen drag

Ytelsen på ansiktgjenkjenningssystemer er avhengig av kvaliteten på bildene for å kunne teste og trene disse systemene. For å automatisk vurdere kvaliteten på ansiktsbilder blir Face Image Quality Metrics (FIQMs) benyttet. Slike kvalitetsmetriker gir objektive resultater som korresponderer med hvor synlig ansiktet i et bilde er. I dette prosjektet introduserer vi en webapplikasjon som regner ut slike objektive resultater med to moderne FIQMs. For å vurdere nøyaktigheten på disse metrikkene, samlet vi inn subjektive data fra eksperter og ikke-eksperter på tre forskjellige dataset. Vi samlet også inn et nytt dataset med ansiktsbilder som vi mener er overlegent i forhold til andre nåværende dataset. Denne overlegenheten er med tanke på antall bilder, type forvregninger bildene er påvirket av og hvordan bildet er tatt (bilder med ansiktsmasker og bilder med skrå vinkler). Resultatene viser at de objektive resultatene regnet ut av de to kvalitetsmetrikkene har lav til moderat korrelasjon med de subjektive resultatene samlet inn via subjektive eksperimenter.

Contents

Abstract	iii
Sammendrag	v
Contents	vii
Figures	xi
Tables	xiii
Code Listings	xv
Acronyms	xvii
1 Introduction	1
1.1 Background	1
1.2 Subject Area	1
1.3 Task Description	2
1.4 Delimitation	2
1.5 Target Groups	3
1.5.1 Users of the Web Application	3
1.5.2 Thesis Readers	3
1.6 The Team	3
1.6.1 The Members	3
1.6.2 Academic Background	4
1.7 Why Did We Choose This Task?	5
1.7.1 Roles	5
1.7.2 Decision Making	6
1.8 Thesis Structure	6
2 Specification	9
2.1 Development Method	9
2.1.1 Considered Models	10
2.1.2 Our Model	11
2.1.3 Scrum Layout	12
2.1.4 Difficulties With Our Model	13
2.2 Risk Analysis	14
3 Face Quality Assessment	19
3.1 What Is a Good Image?	19
3.2 Face Image Quality Metrics	22
3.2.1 ISO Metrics	23
3.2.2 FaceQnet	23

4	Objective Assessment	25
4.1	Requirements	25
4.1.1	Type of Application	26
4.2	Use Case	27
4.3	Choice of Front- and Backend	29
4.3.1	Backend	29
4.3.2	Frontend	31
4.4	Design and Implementation	33
4.4.1	Backend	33
4.4.2	Frontend	35
4.4.3	Deployment	36
4.5	Sequence Diagrams	37
4.5.1	High Level - ISO Metrics	37
4.5.2	Low Level - FaceQnet	38
4.6	Testing	39
4.6.1	User Testing	39
5	Subjective Experiment	43
5.1	Why Collect Ground Truth Data?	43
5.2	Subjective Experiment Aspects	43
5.2.1	Experiment Method	44
5.2.2	Experiment Duration	44
5.2.3	Observers	44
5.3	Datasets	44
5.4	Our Subjective Experiment	46
5.4.1	Instruction Manual	46
5.4.2	Choice of Subjective Experiment Platform	47
5.5	COVID-19 Pandemic Implications	48
5.6	What Went Wrong?	48
5.7	Creating Our Own Dataset	49
5.7.1	Creation Process	49
5.7.2	Distortions	50
5.8	Second Subjective Experiment	52
6	Results and Discussion	55
6.1	Main Experiment	55
6.1.1	Results of the Subjective Evaluation	55
6.1.2	Evaluating the Performance of FIQMs	58
6.1.3	Significance of Our Results	62
6.2	Second Experiment	63
6.2.1	Results of the Subjective Evaluation	63
6.2.2	Results of the Objective Evaluation	64
6.2.3	Evaluating the Performance of FIQMs	66
7	Conclusion	71
7.1	Objective Assessment Evaluation	71
7.2	Subjective Experiment Evaluation	71

7.3	Evaluation of the Results	72
7.4	Group Work Evaluation	72
7.5	Learning Outcome	73
7.6	Future Work	73
	Bibliography	75
A	Project Assignment	79
B	Project Agreements	83
C	Project Plan	91
D	Survey Instructions	109
E	Results	121
F	Meetings With Project Supervisor	131
G	Meetings With Mobai	143
H	Toggl Reports	153

Figures

3.1	Sample images from a face quality dataset.	20
3.2	Examples of accepted and not accepted facial images in a facial recognition system.	21
3.3	Face Recognition system. Middle photo by Thomas Haugersveen, Statsministerens Kontor. Top left photo by Human-Etisk Forbund. Second and third from the top photos by Torbjørn Kjosvold, Forsvaret. Bottom left photo by Eirin Larsen, Statsministerens Kontor.	22
3.4	Typical FIQM process [12].	23
4.1	Use case diagram.	27
4.2	POST request with Postman.	34
4.3	ISO Metrics: high level sequence diagram.	37
4.4	FaceQnet: low level sequence diagram.	39
5.1	Facial images from Combined passport alike.	45
5.2	Facial images from Capture from photo.	45
5.3	Facial images from Selfie dataset.	46
5.4	A lineup from the experiment instruction manual and their quality ratings.	47
5.5	Different face masks usage.	51
5.6	Different oblique angle camera shots.	52
6.1	Histogram of MOS values calculated for the images in each dataset. There is a clear difference between the quality of the Selfie dataset and the two others.	56
6.2	Standard deviation values of the subjective scores given to each image in the three datasets. Sample images from each dataset are shown as examples.	57
6.3	2D scatter plots of the objective and subjective scores on the three datasets with objective scores along the x-axis and the subjective scores along the y-axis. The Spearman and Pearson correlation coefficients are shown above each plot.	59

6.4	2D scatter plots of the objective and subjective scores on the three datasets with objective scores along the x-axis and the subjective scores along the y-axis. The Spearman and Pearson correlation coefficients are shown above each plot.	60
6.5	2D scatter plots of the scores provided by using a weighted average of the FIQMs (with the same weight) on Combined passport alike and Capture from photo. The average FIQMs scores are displayed along the x-axis and subjective scores along the y-axis. The Spearman and Pearson correlation coefficients are presented above the plots.	61
6.6	Histogram of the subjective scores on the NFC dataset.	63
6.7	Histogram of the subjective scores of the original images and their corresponding distorted images in the NFC dataset.	64
6.8	Histogram of the objective scores of the original images and their corresponding distorted images in the NFC dataset.	65
6.9	2D scatter plots of FaceQnet scores on the original images along the x-axis and the distorted images along the y-axis. All correlation coefficients are given with a 95% confidence interval. RMSE values are included above the plots.	65
6.10	2D scatter plots of ISO Metrics scores on the original images along the x-axis and the distorted images along the y-axis. All correlation coefficients are given with a 95% confidence interval. RMSE values are included above the plots.	66
6.11	2D scatter plots of the scores on the NFC dataset with objective scores along the x-axis and subjective scores along the y-axis. The Spearman and Pearson correlation coefficients are shown above the plots.	66
6.12	Spider chart of the objective and subjective scores on different face mask images.	67
6.13	Spider chart of the objective and subjective scores on images with different oblique angles.	68
6.14	Spider chart of the objective and subjective scores on original and distorted facial images.	68
6.15	Spider chart of the objective and subjective scores on original and distorted facial images.	69

Tables

2.1	Consequence and likelihood color coding risk matrix.	15
2.2	Project risks.	16
2.3	Product risks.	17
2.4	Business risks.	18
4.1	Use case: Upload images.	28
4.2	Use case: Run FIQM.	28
4.3	Use case: Delete images.	28
4.4	High level use case for “Run all FIQMs”.	29
4.5	Issues and solutions following the user testing.	40
4.6	User testing tasks and feedback.	40
4.7	User testing questions and feedback.	41
5.1	Information about all datasets used in the project.	46
6.1	The calculated RMSE values of the FIQMs on the datasets relative to the subjective scores. The RMSE value was not calculated for the Selfie dataset with a weighted average of the FIQMs. The ‘X’ symbolises this.	62
6.2	All correlation coefficients on the three datasets between ISO Metrics, FaceQnet, FIQMs Weighted AVG and the subjective scores. The ‘X’-symbol indicated the correlation coefficients that had a p-value higher than 0.05 and were therefore ignored. The coefficients were not calculated where the ‘-’ symbol is placed.	63

Code Listings

4.1	API call implementation.	33
4.2	React hook implementation.	35
4.3	Axios HTTP request implementation.	36

Acronyms

The page number after an acronym refers to the first time it is used.

AI Artificial Intelligence. xv, 1

API Application Programming Interface. xv, 3

CIE International Commission on Illumination. xv, 48

CLI Command Line Interface. xv, 32

CNN Convolutional Neural Network. xv, 23

COVID-19 Coronavirus disease 2019. xv, 14

CVD Colour Vision Deficiency. xv, 48

DOM Document Object Model. xv, 32

FIQA Face Image Quality Assessment. xv, 2

FIQM Face Image Quality Metric. xv, 1

FR Face Recognition. xv, 22

GDPR General Data Protection Regulation. xv, 5

HTML HyperText Markup Language. xv, 32

HTTP Hypertext Transfer Protocol. xv, 35

ICAO International Civil Aviation Organization. xv, 19

IEC International Electrotechnical Commission. xv, 19

ISO International Organization for Standardization. xv, 5

JSON JavaScript Object Notation. xv, 26

- MOS** Mean Opinion Score. xv, 55
- MRTD** Machine Readable Travel Documents. xv, 19
- MTCNN** Multitask Cascaded Convolutional Networks. xv, 24
- MVC** Model-View-Controller. xv, 32
- REST** REpresentational State Transfer. xv, 25
- RMSE** Root Mean Square Error. xv, 61
- UI** User Interface. xv, 27
- XHTML** Extensible HyperText Markup Language. xv, 32
- XML** Extensible Markup Language. xv, 32
- XP** Extreme Programming. xv, 10

Chapter 1

Introduction

1.1 Background

Mobai is a spin off company from the research developed in the Norwegian Biometrics Laboratory at the Norwegian University of Science and Technology. They provide solutions for facial recognition, biometrical attack detection and face morph detection. To create models for detection of biometrical attributes, artificial intelligence (AI) and machine learning are essential tools for Mobai. In addition to the models, having access to appropriate datasets plays a crucial role in training and developing new models. An important focus of Mobai is using different Face Image Quality Metrics (FIQMs) to determine the quality of facial images in a dataset. In order to train models, quickly assess several datasets or create new datasets, it is important to know the quality of facial images. Therefore, Mobai now aims to develop an application that automates this process.

1.2 Subject Area

Digital image processing is a rapidly growing field within the world of engineering and computer science. A great amount of research has been done in this field of study, paving the way for multimedia systems to become one of the pillars of the modern information society. Digital image processing is used in a variety of technologies, including face detection and face recognition which itself could be categorized under the broader field of pattern recognition. Digitalization has drastically changed peoples everyday lives over the last decade, and with that change, biometrics has become more relevant and important than ever.

Solutions such as face recognition, presentation attack detection and face morph detection all heavily rely on the quality of the facial images used for machine learning training. The quality of facial images are dependent on several factors which FIQMs have learned trough artificial intelligence and machine learning. The performance of FIQMs can be measured by comparing the quality scores with human assessment. This thesis is mainly focused on Face Image Quality As-

assessment (FIQA) which plays a key role in improving the accuracy of face recognition systems.

1.3 Task Description

This bachelor project can be divided into two main parts, a programming part (mainly Chapter 4) and a research part (mainly Chapter 5).

Programming

The programming part involves the creation of a web application that uses two FIQMs provided by Mobai for evaluating the quality of facial images. The application will create a report which contains the FIQMs' calculations on a set of images. The key functionalities of the application expected from Mobai are:

- The user should be able to read/select images from a local machine.
- The user should be able to upload images to a directory.
- The user should be able to execute the two FIQMs on the uploaded images.
- The user should be able to display the results from the FIQMs.

Research

The research part consists of conducting a subjective experiment. The subjective experiment involves collecting ground truth data by having subjects evaluate the quality of facial images from a dataset based on certain criteria. During the experiment, observers will be shown different facial images of varying quality and asked to label the images into predefined categories. The subjective results will then be used to compare with the objective results from the FIQMs. By comparing the subjective results from the experiment with the objective measure calculated by FIQMs, we are then able to evaluate the performance of FIQMs.

1.4 Delimitation

In this project, our task was not solely made up of pure programming. Fortunately, the subject field we were working within allowed research which we found to be a great experience in the final step of our bachelor studies. As an example, facial images used for research in the field of face quality assessment and face recognition were usually taken from a straight angle with little to no tilting of the camera lens. After a literature review it was evident that studies addressing camera tilting was very limited. With the ongoing pandemic, wearing face masks in public has become a habit. Even though wearing a mask is a new aspect of our daily lives, the performance of FIQMs on images which show a subject wearing a face mask has not yet been studied. In that case, we wanted to assess the performance of the

FIQMs on face mask images. For the mentioned reasons, the bachelor group proposed the collection of a new dataset which Mobai supported the initiative. The mentioned dataset will not only be used by Mobai in their studies but can also be used by the bachelor group. Further information about the collected dataset is provided in Section 5.7. Finally we should point out that with the exception of our dataset, all FIQMs and datasets were provided to the bachelor group by Mobai.

1.5 Target Groups

In general, this project is targeted towards two groups, Mobai and the readers of the thesis.

1.5.1 Users of the Web Application

The web application will be used by employees at Mobai to evaluate the performance of the FIQMs on different datasets. Based on the Non Disclosure Agreement (NDA) signed by the bachelor group (Appendix B), because of possible competitive companies neither the source code or the application should be distributed to others except Mobai.

1.5.2 Thesis Readers

The target group for the bachelor thesis are people who need insight in how we did the project from a developer point of view as well as a scientific point of view. This includes but is not limited to our thesis supervisor, client and fellow students with similar background that can be interested in reading the thesis.

1.6 The Team

In this section we will introduce the members in our team. Here we look at their interests, responsibilities and their academic backgrounds.

1.6.1 The Members

Hans Petter

Hans Petter is a computer engineer who is interested in mathematics and artificial intelligence. He is interested in Python programming and was therefore one of the main developers of the backend. Hans Petter had the main responsibility for implementing APIs to the frontend. He also helped out the other team members whenever there were any API errors connecting Flask to React or general Python problems. In addition to the application task, Hans Petter collected the data from the subjective experiment to make it workable.

Walid

Walid has prior knowledge with artificial intelligence and finds topics that mix statistics and mathematics with artificial intelligence interesting. Throughout the project, Walid was mainly involved in the creation of the backend logic with Hans Petter. In particular, he developed API endpoints used by the frontend and was responsible for processing the objective and subjective data.

Julian

Julian is a computer engineer with an interest in programming, mathematics and artificial intelligence. His background consists of different subjects regarding programming, mathematics, software security and application development. Julian's main responsibility was creating the frontend of the application and handling the data received by the backend.

Kjetil

Kjetil is a developer with an interest in frontend web development, JavaScript and Python programming. Due to his background and interests, his main responsibility was to work with the frontend and help with the backend development.

1.6.2 Academic Background

All group members are studying for a bachelor's degree in computer engineering, hence our academic backgrounds are closely similar. However, during the fifth semester, we all chose different subjects. Hans Petter and Walid both chose *artificial intelligence*, *software design* and *calculus 3*. Julian had the elective subjects *application development*, *software security* and *calculus 3*, while Kjetil studied the subjects: *ergonomics in digital media*, *infrastructure as code* and *application development*.

Throughout our bachelor's program we have acquired a solid foundation when it comes to the software development process and scientific thinking by finishing courses like *software engineering*, *algorithmic methods*, *operating systems*, *scientific computing*, *calculus* and *physics*. The compulsory courses we finished helped us see different approaches to the same problem. There will always be several ways to approach a task and a bachelor's degree in computer engineering has cemented that statement. Our prior knowledge made it easier to understand and process the main concepts of the new topics we had to learn. During this semester, we have acquired more knowledge and built on our foundation. We had to perform research on specific topics in backend and frontend programming, as these were topics we were unfamiliar with. More specifically, the part of connecting the frontend to the backend by using API calls, was something we had never done before. In addition to that, a great amount of time was also spent on learning how to conduct and perform scientific experiments in a way that satisfies both

General Data Protection Regulation (GDPR) and the relevant ISO-standards.

1.7 Why Did We Choose This Task?

Although stationed in Gjøvik the group had never heard of Mobai or what they do. Their task description seemed interesting, and we were early to contact them. Already at the first meeting, before any bachelor tasks were selected, we got a very positive impression. The participants from Mobai were very curious and enthusiastic. They had clear ideas and suggestions on how we could approach the task. After this meeting, the bachelor thesis choice became an easy decision.

A reason why we chose this task was the field of work. All group members were interested in machine learning and artificial intelligence. Therefore, we saw the topic as a perfect opportunity to have a hands on experience working in this field. Another reason for our choice was the width of the project. We felt that we could use experiences and knowledge learned throughout the bachelor courses in this work. We had to combine math, statistics and programming with research, which given our bachelor's program suited us well. This kind of scope was complex, but manageable, and challenged us in several ways in terms of coding and research. We were motivated by the fact that Mobai expected us to create an application that they would utilize in addition to creating a subjective experiment.

1.7.1 Roles

Our project work areas were mainly divided into two categories: research and programming. While studying FIQMs, the subjective assessment and the creation of a subjective experiment which can be seen as research were done collectively to ensure equal professional foundation. When it came to the programming part, each group member were delegated responsibility more individually. However, due to the nature of the work, we had to closely collaborate with each other with backend and frontend because its dependency. During the project, the following roles were assigned to each group member:

- Kjetil Grosberghaugen was scrum master and developer. His main responsibility was frontend, written in React.js. As a scrum master he ensured the development process flew evenly and led sprint meetings.
- Julian Nyland Skattum was a developer with main responsibility in frontend. He was also responsible for programming language choices as well as the overall application appearance.
- Hans Petter Fauchald Taralrud was a developer with main responsibility in backend.
- Walid Demloj was a developer with main responsibility in backend.

Every member were participating in the full stack application despite their area of responsibility. The reason for having individual responsibilities was to easily maintain that the requirements were met. The group members had joint re-

sponsibility for the project plan (see Appendix C) and thesis as well as completing tasks according to the created requirements.

Seyed Ali Amirshahi was our supervisor. The project group had scheduled meetings every Monday at 12:00. Ali's role in the project was to guide us throughout the project. He would evaluate the project plan, thesis and answer professional questions. Summaries of the meetings with Ali can be found in Appendix F.

Kjartan Mikkelsen started as the product owner, but after about three months in the process, he left the company. Our new product owner and contact was Guoqiang Li. He expressed Mobai's vision and wishes during the project. We also got technical assistance from other employees at Mobai. Summaries of the meetings with Mobai can be found in Appendix G.

1.7.2 Decision Making

The group decided to go for a collaborative decision-making [1] approach when making choices throughout the project. Decisions like frontend and backend environment and experiment platform were decided collectively with knowledge and experience in mind. A touch of prior knowledge benefited our project, so we did not need to learn everything from scratch. Making choices in groups led to a comfortable team atmosphere with a good relationship between each team member. The decentralization of decision responsibilities made it easier to contribute in the decision making, because possible failures would be shared in the group. However, if a discussion had reached an impasse, the group leader had the final say. Choices related to project work were brought up in the daily scrum meetings, as they offered frequent changes as well as a cordial environment to give feedback and improvements. Choices related to project organization were arranged in the sprint retrospective meetings, every other week.

1.8 Thesis Structure

It is important to emphasize that this thesis is atypical. As almost every bachelor task include reporting the development process, our thesis adds an extra dimension in terms of research. The subjective part requires thorough research to be able to correctly compare with the results provided by the application. All this is reflected in the way we structured our thesis. We have chosen to structure the thesis as follows: in Chapter 2, we are discussing our choice of development methodology. We look at pros and cons for the methodology, address other possible development methods and elaborate the current development layout. Following the chosen development approach is an in-depth risk analysis. Chapter 3 is an important part of the thesis. Here we define some of the most essential concepts used in the project. We describe the definition of a good facial image in face recognition as well as introducing the FIQMs used in the application. Next, in Chapter 4 we will look at how the face image metric application has been implemented and the

reasoning behind our choices. In Chapter 5 we go through the process of conducting a subjective experiment to gather ground truth data. We also introduce the datasets used in the experiment and explain our own dataset creation. Both the objective assessment and subjective experiments are merged together in Chapter 6, where their results are compared. Following the results, we can conclude the FIQMs strengths and weaknesses based on their correlation with the subjective scores. In the conclusion Chapter 7 we conclude the results and introduce further work based on our research. The reasoning behind this project structure is mainly to provide the thesis readers a clear overview and a sequential story. Since the project's contents of the work are split, it is meaningful to describe the workload in separate chapters.

Chapter 2

Specification

In this chapter we will start in Section 2.1 by looking at our chosen development method for the project and obstacles that arose with it. To end the chapter of, an in-depth risk analysis will be presented in Section 2.2.

2.1 Development Method

Projects of this size are dependant on well-structured development methods to succeed as it affects the way teams are organized and tasks are delegated. The prioritization of workload is also affected by the choice of development method. This project was characterised by uncertainty, a limited amount of labor and a development phase with requirements that may change. These characteristics paved the way for us to pick an agile software development method which was suggested by our client Mobai.

1. Constant Feedback

Our group consists of four third year students with little to none experience with comprehensive projects of this size. Such lack of experience alone causes some uncertainty in regards to deadlines, subjective experiment work and development. Having regular meetings with both Mobai and our project supervisor as well as receiving regular feedback, will decrease the chance to drift off track. This also ensures that our final product is as close to Mobai's vision as possible. Agile software development methods provides the client with the chance to be far more involved in the project. This aspect itself was an important point to take into consideration.

2. Small Team

For a team like us, an agile method is more suited because small teams are better positioned to efficiently and effectively manage events like meetings [2]. The reduction of communication channels in a team decreases the possibility of misinterpretations among members. A small team is more likely to have an attentive communication where decisions are made quickly.

3. All Known Requirements Are Not final

In agile development methods it is expected that all the requirements are not finalized or even known in advance. Throughout the development process new requirements may be added and existing ones can be refined to better suit the upcoming application. This agility is not achieved by plan driven approaches because all requirements must be finalized in the planning phase. Bachelor projects rarely turn out exactly as planned, which is why an agile development method that handles changes would be beneficial.

2.1.1 Considered Models

Although the scale tipped towards an agile methodology, we also included plan-driven approaches in our considerations in case they were a better solution. The following is our reasoning behind considered models.

Waterfall Method

The fundamental process activities are broken down into linear sequential phases [3]. Each phase has to be completed before it cascades into the next. This allows for departmentalization and control where deadlines can be set for every phase, resulting in a clear workflow. However, this model would not be suitable for our development due to its lack of revision. Should the development phase run into issues that was not taken into consideration in the planning or design phase, the design phase would have to start over. This would be very time consuming. In addition, not having a working product until late in the development life cycle would be risky in regards to our programming uncertainty.

Extreme Programming

Extreme Programming (XP) emphasises five key values: communication, feedback, simplicity, courage and respect. It strives for high team efficiency and higher quality code by implementing 12 practises such as refactoring, pair programming, test first development, continuous integration and simple design. Programmers are expected to work closely together, which is encouraged by the values mentioned [4]. The main downside however, is its strict and difficult way of working. It requires great self-discipline to follow all the practises the method is built upon. XP mainly focuses on how software development should be approached, and since our project was not solely based on development, the model did not seem to fit our needs.

Kanban

Kanban provides solid structuring and flexibility. The methodology is all about limiting work in progress as well as maximizing the work efficiency. It is based on a continuous workflow structure with continuous deliveries with no set dates. This ensures the team is ready to adapt to change whenever priorities may change. The kanban team are reliant on each other to succeed. No predefined roles are set, which means it is a collective responsibility of the team to work together and finish tasks. Likewise, no set dates are placed for when certain functionalities should be released [5]. We felt that this “go with the flow” mentality of kanban would lead to more uncertainty regarding the bachelor groups progress. The roles and responsibilities structuring of kanban reinforced this mindset.

Scrum

Scrum is adjustable, flexible and a fast agile method that largely involves the client throughout the project cycle [6]. The requirements are never entirely determined which allows for updating the product. Small teams with defined roles are appropriate to provide different responsibility within the team. Having regular meetings ensures good communication as well as keeping team members up to date with the continuous progress.

2.1.2 Our Model

We chose to use a scrum model for this project. In addition, we wanted to implement the pair programming practise from XP since it would help to produce high-quality code as well as share experience in the field of programming. The pair programming was executed by screen sharing or with Visual Studio Code live-share. We also wanted to easily manage our working tasks. Therefore, we applied a kanban board to get a clear overview of the tasks that needed to be done. Trello [7] was used for tracking the tasks and the tasks were divided into three phases: To-Do, Doing and Done. Since Kanban was not our main method, no specific rules or limitations were set considering the amount of tasks in each phase. Due to the project’s nature, the adaptability and flexibility provided by scrum seemed to be a safe option which could also increase the chances of success in the project. The way the model organizes the workload into different sprints and facilitates to pre-defined member roles helped us keep up with the progression made throughout the project. Were we to finish Mobai’s desired application quickly, it was suggested by Mobai that additional functionality could be added to the solution if our time schedule allowed for it. Changes in requirements were something Mobai were open to discuss and scrum handles this well, which was another reason for our choice.

2.1.3 Scrum Layout

We had sprints with a two-week duration. This is because one-week sprints felt short and would lead to excessive meetings which would affect our time management. Longer sprints, however, tend to involve the client to a lesser degree, which would not be ideal in our case. The involvement of the client was not only to have continuous feedback to ensure that we satisfied their requirements, but was also what Mobai had emphasized at the start of the project. The sprints consisted of several meetings and scrum practises, such as Sprint planning meetings, Daily scrums, Sprint review meetings and Sprint retrospective meetings. These meetings were all followed to a certain extent.

Sprint Planning Meeting

The start of the sprints took place on Tuesdays, (with the first meeting on the 2nd of February) and were expected to finish by 11:00 every other Monday. The weekly scheduled meetings with our bachelor supervisor took place every Monday at 12:00, which gave us at least one hour to prepare both the next sprint and what we should discuss during the meeting. This is when all our sprint planning meetings were held and conversations about what features should be included in the upcoming sprint were planned. Planning Poker ¹ was used to estimate the time spent on different tasks.

Daily scrum

These short meetings were held at the start of every day at 9:00 and were included to assure the members were kept up to date about the progress of the project.

Sprint Review Meeting

The sprint review meetings with Mobai were arranged every other Tuesday, at the end of each sprint. Mobai was informed about our progress and provided us feedback. While these meetings were mainly attended by the product owner (Dr. Guoqiang Li) who provided feedback from the client's side, other employees of Mobai also attended these meetings.

Sprint Retrospective Meeting

At the end of each sprint right after our sprint review meetings with Mobai, time was delegated into the sprint retrospective meetings. During these meetings, we reflected about what went well and tried to improve these aspects for the next sprints. A crucial part of these meetings was to decide if our time estimates were on point.

¹<https://www.planningpoker.com/>

Definition of Done

We used the scrum-pattern Definition of Done [8] to collectively define what development-tasks were considered Done. In partnership with Mobai we came to an agreement upon different criteria that form the Definition of Done. It was important to set different criteria to different work tasks (there should be a difference between coding criteria and report criteria). This pattern gave us a familiar understanding of work quality and absoluteness. We also obtained good habits in our workflow using the Definition of Done as a checklist to correlate with the user stories. That way we prevented possible delays occurring in the development process. Here is an example of our report criteria:

1. The section is completed according to the member.
2. The member has analyzed the section's contents.
3. The section is checked for typos.
4. The whole group has read and approved the section.

2.1.4 Difficulties With Our Model

At the early stages of planning our bachelor project, our own development method seemed to fit all our requirements and needs with few disadvantages. However, once the development began, we quickly realised specific elements that were not desirable.

Our pair programming practise inspired by XP did not work out as we intended or predicted it would. As it in the initial start of the project was very helpful to share experience, later in the development process, we came to the realization that it was very time consuming and limiting. Originally, the pair programming was supposed to be done on Visual Studio Code's Live Share, which allows developers to collaborate in real-time. However, because of latency issues with the Live Share platform, we decided to achieve pair programming by screen-sharing instead. This method forced one member to write code while the other observed and gave input which we found out to be unproductive and a waste of our limited resources. Because of that, we ended up coding the product mostly individually. The quality of the code was likely affected by this, however the productivity of the team became increasingly higher. The progress was easily traceable given our everyday meetings on online platforms.

All meetings associated with scrum presented at the end of the previous section were beyond doubt time consuming and required certain planning. The Planning Poker time estimating game was abandoned after a while because we felt it took valuable time away from the project. At the start it was a great tool that served us quite well, but as the project became clearer, we developed an understanding of time estimates. Because of that, we ended up estimating the tasks verbally, which saved us time and resources.

2.2 Risk Analysis

Almost every decision made throughout the project involved a risk. A risk was made up by the likelihood of something going wrong and the negative consequences entailed with that risk. Therefore, weighing up the risks before making a decision was essential. The better we understood the occurring risks, the more we were prepared to manage them. Carrying out a risk analysis was beneficial for us to make sure every decision were robust and well considered. Our risk identification was done by discussing possible events that could occur, as well as identifying their likelihood and consequences. Prior knowledge from earlier projects helped us estimate whether a risk was likely or unlikely to occur. Since we were working agile, not all risks were discovered in advance, but some were appended along the project process. The team categorized the risks according to the type of risk. The risk categories we used were: Project, Product and Business, which had been discussed in detail [9].

- **Project risks**

Risks that affect the available resources assigned to the project. These risks interfere with the schedule and may hinder reaching planned deadlines.

- **Product risks**

Risks that affect the overall performance of the system created. Risks in this category may affect the functionality which can reduce the overall quality.

- **Business risks**

These risks affect the organization developing a product. During this project, such risks will mainly be related to the team members as well as collaborators such as Mobai and our supervisor.

Each risk listed below in the section was evaluated and assessed based on its consequences and likelihood (Table 2.1). The numbers inside the Table reference what the risks were, and have nothing to do with the priority or severity.

- **Risk 1:** Group members leaving the project.
- **Risk 2:** Sprint deadlines are not met.
- **Risk 3:** Bachelor thesis is not finished in time.
- **Risk 4:** Mobai's expected involvement turn out to be minimal.
- **Risk 5:** A similar product launches.
- **Risk 6:** The software does not fulfill the minimum requirements.
- **Risk 7:** Inadequate planning and execution of subjective experiments.
- **Risk 8:** Loss of vital documents or source code.
- **Risk 9:** Sickness among group members, project supervisor or Mobai.
- **Risk 10:** Application breaking bugs.
- **Risk 11:** Planned tools for the development of the experiments and application prove to be unsatisfactory.
- **Risk 12:** COVID-19 restrictions interfere with the subjective experiment.

- **Risk 13:** Unable of connecting the frontend to the backend logic.
- **Risk 14:** Unrealistic demands are made along the way by the Product Owner.
- **Risk 15:** Additional help is required by other group members to overcome a specific problem.

Table 2.1: Consequence and likelihood color coding risk matrix.

		Consequence			
		Minor	Moderate	Major	Critical
Likelihood	Unlikely	R5		R1 & R4 & R6	R3 & R8 & R13
	Likely	R14	R11	R2	R10
	Quite likely	R15		R7 & R9	
	Very likely		R12		

Table 2.1 showcases a standard consequence/likelihood risk ranking matrix split into the likelihood of a risk occurring and its corresponding consequence (i.e severity). The consequences range from minor to critical and the likelihood from unlikely to very likely. The three colors represent different type of risk levels. The green color indicates low risk, yellow indicates moderate risk and red represent risks of great importance.

We chose a 4×4 matrix with three color codes because it covers and classifies a wide specter of risks. We could have settled for a 3×3 matrix, but we felt that risks would overlap and therefore a more severe risk could be mixed with a risk with lesser severity. A 5×5 matrix could also have been used, but since our project was limited in size relative to corporate projects, we felt that this type of granularity was not needed.

In the next pages, all the above-mentioned risks are placed into the three categories presented and discussed with countermeasures.

Table 2.2: Project risks.

ID	Description	Countermeasures
R3	Bachelor thesis is not finished in time.	Base our time estimates on worst-case scenarios, that way if tasks are finished before a set date, some extra time can be added to the end for polishing purposes.
R7	Inadequate planning and execution of subjective experiments.	Creating subjective experiments at an early stage and studying the pitfalls in conducting experiments will minimize the risk of designing lackluster and confusing subjective experiments. For more in depth and specific countermeasures, see Section 5.4.
R8	Loss of vital documents or source code.	Using backup-technologies when writing the report and coding the software prevents loss of important content. The thesis will be synced with a GitHub repository and changes to the Overleaf project will regularly be pushed to the repository.
R12	COVID-19 restrictions interfere with subjective experiments.	It is highly likely that we will not be able to conduct experiments in a controlled environment which may affect subjects' image evaluations. An instruction manual will be provided to all participants.
R2	Sprint deadlines are not met.	Having a consistent workflow and a regular meeting schedule makes it easier to meet the deadlines. If there are too many tasks in the upcoming sprint, a rescheduling may occur and the Product Owner will be informed.
R4	Mobai's expected involvement turn out to be minimal.	Doing extensive early research about our subject field is important to understand the basics, which we can build on ourselves. In addition, our supervisor should be utilized as best as possible.
R15	Additional help is required by other group members to overcome a specific problem.	Our research and planning should make us capable of knowing a little about several topics. Members should therefore be able to assist each other if needed.

Table 2.3: Product risks.

ID	Description	Countermeasures
R10	Application breaking bugs.	Analyze what critical bugs can occur when developing the product at an early stage. Applying continuous testing throughout the development phase should mitigate large faults in the application.
R13	Unable of connecting the frontend to the backend logic.	We should minimize the use of hard coding that may interfere with connecting React to Flask. Additionally, study materials which could help us in this issue should be studied by the members. Lastly, Mobai's development team may assist us if needed.
R6	The software does not fulfill the minimal requirements.	Keeping a consistent workflow ensures maximum efficiency. Should problems, which forces us to exceed deadlines by a great margin occur, a meeting with Mobai will be held where the requirements should be reconsidered.
R11	Planned tools for development of the experiments and application prove to be unsatisfactory.	All tools should be replaceable and carefully considered through research to minimize the use of inefficient tools. Most of our requirements should be met by the tool. If several tools match our needs, the most suitable and quickest to learn will be chosen.

Table 2.4: Business risks.

ID	Description	Countermeasures
R9	Sickness among group members, project supervisor or Mobai.	The team shall follow the regional and local COVID-19 measures to the greatest extent. Members becoming ill should be kept up to date about the progress of the project. Additionally, to prevent rescheduling, members should know what the rest of the group is working on. Daily meetings will be an effective countermeasure.
R1	Group members leaving the project.	A structured group with good planning and communication reduces the risk of members leaving the project. A safe team atmosphere where members are encouraged to bring fourth new ideas and input is essential. The team members should also be given the opportunity to work with something they find interesting and fun. Should however a member leave the team, a redistribution of the remaining work tasks shall be equally split amongst the remaining members.
R5	A similar product launches.	If a similar product launches, it is important to meet all of Mobai ´s specialized needs and avoid delays in the deployment.
R14	Unrealistic demands are made along the way by the Product Owner.	Our agile development method will mitigate large reschedules of the project if new demands are provided. However, our main task is written in the task description (1.3) and any additional demands are not required by us to complete.

Chapter 3

Face Quality Assessment

To start off the third chapter we introduce the reader to key definitions and concepts within the field of face quality in Section 3.1. In Section 3.2, we will throw light on face image quality with examples and elaborate how face recognition systems and the two FIQMs behave.

3.1 What Is a Good Image?

Throughout this thesis, we will mention the word quality several times. While some people already have a good understanding of what the word means, some confusion may arise. Within the standard image quality assessment field, the definition of quality is straight forward and is what most people think about when hearing the word. People working with images will for the most part think about an image with a high resolution and no distortion or noise as the most important characteristic that defines image quality. An excellent image of a face with this definition would have a high resolution and a sharp focus. However, we are working in the field of biometrics, where the definition differs from this. Our emphasis lies not with the image resolution itself, but with how well the face is visible. There are several aspects that heavily affect the quality of facial images with respect to the performance of biometric systems. These aspects are presented and discussed in two important standards:

- ISO/IEC TR 29794-5:2010: Information technology — Biometric sample quality — Part 5: Face image data [10].
- ICAO Doc 9303 Part 3: Specifications Common to all MRTDs [11].

The quality specific aspects of facial images presented in the two standards are what defines the quality in our work. These different quality factors are:

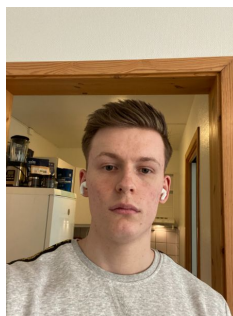
- Scenery characteristics such as lighting or background (Figure 3.2a).
- Complete or partial face covering.
 - Dark glasses fully covering the eyes (Figure 3.2b).
 - Any type of face coverings (Figure 3.2c).

- The behaviour of the subject.
 - Closed or open eyes (Figure 3.2d).
 - Closed or open mouth (Figure 3.2d).
 - Any kind of expression, e.g., smiling or neutral (Figure 3.2e).
 - Head pose, e.g. frontal or rotated in any direction (Figure 3.2f).
- Image properties like the size of the image or its resolution (Figure 3.2g).
- Image appearance characteristics like the exposure or noise (Figure 3.2h).
- Characteristics like the consistency between the skin colour on the image and the skin colour of the subject (Figure 3.2i).

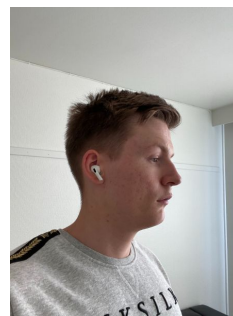
The above mentioned aspects all have an affect on the quality of the facial image to some degree. What we consider a high-quality facial image is similar to a standard passport photograph with the following characteristics:

- Open and visible eyes (Figure 3.2a, 3.2c, 3.2e, 3.2g, 3.2h, 3.2i).
- No dark tinted glasses (Figure 3.2a, 3.2c, 3.2d, 3.2e, 3.2f, 3.2g, 3.2h, 3.2i).
- Neutral or little to none facial expression (Figure 3.2a, 3.2b, 3.2c, 3.2e, 3.2h, 3.2i).
- Neutral face pose (Figure 3.2a, 3.2b, 3.2d, 3.2e, 3.2g, 3.2h, 3.2i).
- No garments covering the face (Figure 3.2a, 3.2d, 3.2e, 3.2f, 3.2g, 3.2h, 3.2i).
- Clean background (Figure 3.2).
- Neither too dark or too light background (Figure 3.2b, 3.2c, 3.2d, 3.2f, 3.2g, 3.2h).
- Photo taken neither too close or too far away (Figure 3.2).

Such characteristics will naturally have an affect on the image quality in a good or bad way. If all the bullet points above are followed, the quality of the facial image is near perfect. An image of a half-covered or missing face will negatively affect the quality, even though the image resolution itself may be impeccable.



(a) A facial image which has the requirements needed for a facial recognition system.



(b) A facial image with defects which does not have the requirements needed for a facial recognition system.

Figure 3.1: Sample images from a face quality dataset.

Image depicted in Figure 3.1a checks all bullet points in what defines a high quality facial image. The subjects' facial expression and head pose are neutral and the whole face is clearly visible. The second Figure 3.1b however, does not have high quality since the face is rotated 90 degrees sideways which only makes half of the face visible. The facial image in Figure 3.1b does have a neutral background with good lighting and image resolution, but those aspects do not weigh up against the head pose.

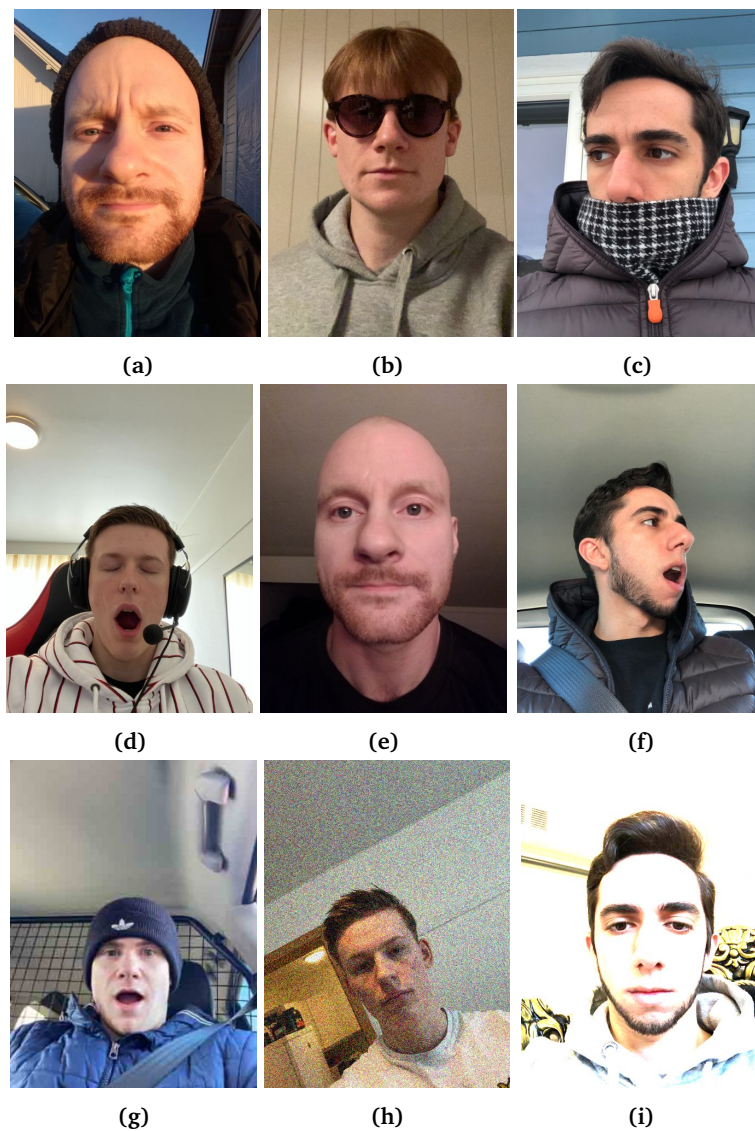


Figure 3.2: Examples of accepted and not accepted facial images in a facial recognition system.

3.2 Face Image Quality Metrics

With the inclusion of authentication to applications, Face Recognition systems are being utilized more than ever. These FR systems are reliant on a pre-captured reference image (a high quality image) which are then used as a reference to compare with the test image (also called probe image).

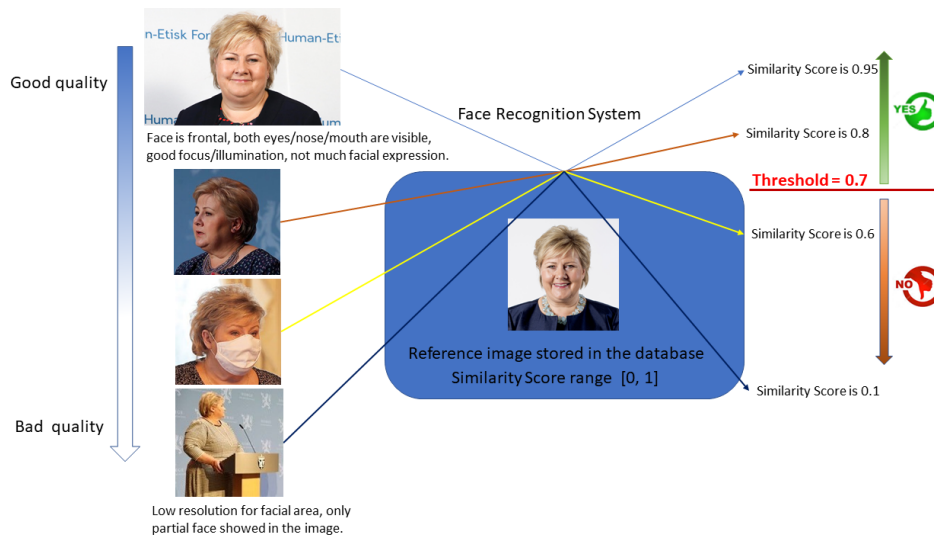


Figure 3.3: Face Recognition system. Middle photo by Thomas Haugersveen, Statsministerens Kontor. Top left photo by Human-Etisk Forbund. Second and third from the top photos by Torbjørn Kjosvold, Forsvaret. Bottom left photo by Eirin Larsen, Statsministerens Kontor.

Figure 3.3 visualizes how FR systems work. The quality of the probe images on the left side of the figure are assessed and compared with the reference image. A similarity score between the two images is calculated (in the case of the FR system depicted in Figure 3.3 the similarity scores are between zero and one). FR systems have a set threshold where probe images are rejected if the similarity score is too low.

When it comes to the performance of the system, the quality of the probe images plays a crucial role. If the probe images are of bad quality, the overall performance of the system will decrease. To keep the performance of FR systems, careful attention is paid to the quality of facial images so that only high-quality images are used in the system. For this and to evaluate the quality of facial images, FIQMs are used. FIQMs are automated algorithms that evaluate the quality of facial images and provide a score which represents the perceived quality of the given images. FIQMs can be based on different quality factors, such as subject-camera distance, inter-eye distance, pose, lighting and facial coverings.

Mobai chose FaceQnet and ISO Metrics to be used in our application. The reason for choosing these specific metrics was their differences in terms of evalu-

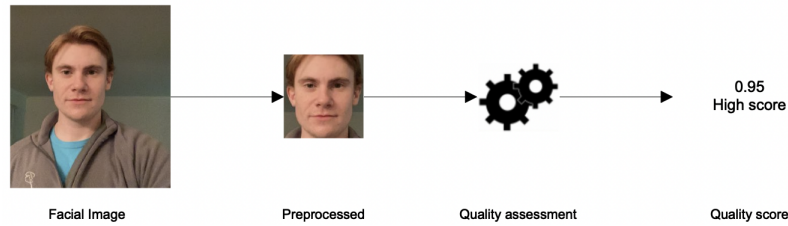


Figure 3.4: Typical FIQM process [12].

ating facial images. The two FIQMs are both no-reference approaches, which will be used to assist and provide feedback when an image is acquired for FR system enrollment.

3.2.1 ISO Metrics

ISO Metrics is a no-referenced FIQM. The metric is implemented based on ISO/IEC TR 29794-5:2010 Information technology — Biometric sample quality — Part 5: Face image data [10]. All factors described in the standard affecting the face image quality are implemented in the FIQM.

ISO Metrics calculate the inter-eye distance on the facial images. If this value is below a certain score, the metric will filter out these types of images. The inter-eye distance is related to the subject camera distance, because it indicates that the subject could be too close to or too far from the camera lens.

All image properties and characteristics described in [10] are taken into account when evaluating the quality of facial images in ISO Metrics. This includes the sharpness, contrast, blur, brightness, exposure, pose symmetry, light symmetry and illumination symmetry of the image. These factors are stored in an image properties array for each facial image.

To be able to calculate quality scores on the facial images, training data is needed. The metric uses random forest regression [13] with 214 estimators and 22 nodes of depth. The quality score for each facial image is computed by predicting the output score of the image properties array.

3.2.2 FaceQnet

FaceQnet [14] is an open source, no-reference FIQM using Convolutional Neural Networks (CNNs). FaceQnet has two versions implemented, FaceQnet v0 [15] and FaceQnet v1 [16]. In this project, we used the latest version (FaceQnet v1). Its quality measures are closely related to the ICAO standard [17] that provides strict guidelines for capturing images. Factors such as illumination, pose, resolution and focus are essential in regard to the final quality score.

A key part of the implementation of FaceQnet is data preprocessing (shown in Figure 3.4). Generally data preprocessing removes unnecessary data, which directly improves the quality of machine learning algorithms. The background of

images will affect the quality score which provides us with biased results. One way to avoid feature extraction from the background is to crop the input images to only include the face before using FIQMs. FaceQnet uses Multitask Cascaded Convolutional Networks (MTCNN) to detect and extract the coordinates of the face. In the next step, the facial image is cropped to an image with the size 224×224 and used as the input image in FaceQnet v1.

FaceQnet uses a subset of the VGGFace2 [18] dataset to create a pre-trained model to make its quality predictions. The subset consists of 300 subjects. The FIQM will first generate ground truth quality measures which are created by labeling the 300 subjects in the training dataset. The ground truth quality measures will then train the deep regression model in order to generate quality scores.

Chapter 4

Objective Assessment

This chapter contains the development process of the Face Image Quality Assessment application. To start we will look at functional and non-functional requirements in Section 4.1 followed by use cases in Section 4.2. Section 4.3 discusses our choice of front- and backend implementation, followed by Section 4.4 that shows off our design and implementation. We include sequence diagrams in Section 4.5 to show how some of the core functionality work. Finally, in Section 4.6 we conclude the chapter with an overview of the user testing process where we ensure the quality of the application.

4.1 Requirements

When approaching the requirements, we had to make decisions based on the task description as well as the meetings with Mobai. The description and Mobai were open in terms of setting requirements, but some core concepts of the desired product were simplicity, clarity, performance and modifiability. Based on these principles, we along with Mobai shaped non-functional and functional requirements. Also, as mentioned in Section 2.1.2, not all requirements were set at the start of the project, so they had to be fine-tuned and confirmed during the development process. While this did not result in any dramatic change in the software, as more functionality were suggested, requirements had to be adjusted to address these issues. The non-functional requirements describes how the software should perform. These are the following requirements:

- The application should be deployed in a container.
- The architecture should consist of a backend and a frontend.
- APIs should be implemented as REST APIs.
- It should be easy to add new FIQMs to the application.

The functional requirements defines what the software should do or not do. These are the following requirements:

- The application should run every available FIQM separately.

- The application should run all FIQMs together.
- The application should return a report with quality scores on every image based on the FIQMs as a JSON file.
- If only one image is evaluated, the image should be displayed.

4.1.1 Type of Application

The responsibility of choosing the application type was given to the bachelor group. Selecting either a web or desktop application was challenging, but we carefully evaluated the type of application that suited our needs the best.

Desktop Application

Since desktop applications are downloaded to your operating system, they are available independently whether you are connected to the internet or not. In that way it is possible to stay functional all the time [19]. The constant accessibility that desktop applications provide, automatically make them more secure. In a way, you never have to be connected to the internet when working on tasks which reduces the possibility of being susceptible to malware attacks. Having a fast computer would be beneficial for running a desktop application. The application uses memory and CPU which returns a good user experience given a resourceful computer. However, using an old computer would not be a problem either. The fact that desktop applications allows for running older versions of the program with all functional availability does not make a hardware upgrade necessary. Also, given that desktop applications do not coerce to update, the software makes it adaptable to choose a suitable version for the computer's specifications. It is also uncomplicated to store files from the application, as information are supported by the computer's hard drives.

Naturally, desktop applications have their downsides as well. The application may require multiple updates to enable full functionality which seems persistent and unnecessary. If the work is performed on different devices, each device need the same installation to synchronize the progress. This will also make it more challenging for multiple users to collaborate on desktop applications. Another disadvantage with a desktop application is its dependence on operating system requirements. To allow the latest functionality the application provides, specific requirements are required.

Web Application

A web application only require one installation step before the software is workable [19]. This is time saving because it provides the user access to the web application by typing the internet address into a web browser on any operating system. All updates are free and immediately available. The ease of availability for these applications make synchronizing with several devices a small matter. In addition to this, the simplicity of managing cooperation with different users is remarkable.

While some licence issues may arise, this solution still allows the users the chance to collaborate from their permanent working stations.

When it comes to the disadvantages of web applications, they require a constant internet connection. In addition, the application's capability depends on the internet connection and speed. A poor connection and speed could result in a bad workflow. As regular updates could be beneficial, there are less possibilities of using older versions of the software.

Our Application

Keeping in mind the mentioned advantages and disadvantages, we chose to build a web application. Given the intuitive user interface (UI) with a limited amount of functions, it was practical and more meaningful to develop a web application. We felt a desktop application would be more suitable if the application requirements increased and the programming tasks were more complex. Our web application would be time saving and comprehensible for Mobai employers since the application only need one installation before working on the software. When new FIQMs are added to the application, the update is easily done and users can instantly evaluate facial images with the new incoming metrics.

4.2 Use Case

We have created a use case diagram showcased in Figure 4.1 along with use cases in Table 4.1, 4.2, 4.3 and 4.4 to show the core functionality and activities within the application. The diagram was built from the perspective of the user (described in Section 1.5) which were employees at Mobai. The cases differ in complexity where running all metrics would be the most challenging. That is because it includes running every metric and providing scores for each image.

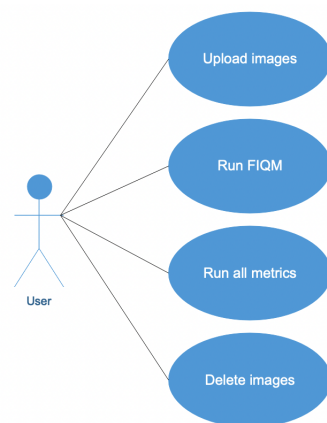


Figure 4.1: Use case diagram.

Table 4.1: Use case: Upload images.

Use case: Upload images.

Actor: User.

Goal: Upload selected images.

Description: The user can select what images he would like to upload to the application. Images will only be stored in the application in that session.

Table 4.2: Use case: Run FIQM.

Use case: Run FIQM.

Actor: User.

Goal: Evaluate images with a specific FIQM.

Description: After uploading selected images, the user would run a specific FIQM to assess the images. The application returns a report with quality scores for each facial image.

Table 4.3: Use case: Delete images.

Use case: Delete images.

Actor: User.

Goal: Remove images from the session.

Description: After running the FIQMs for the uploaded images, the user is able to remove all images used in that session.

Table 4.4: High level use case for “Run all FIQMs”.

Case	Run all metrics.
Description	The user runs all metrics in the application which returns scores for every image.
Basic Flow	<ol style="list-style-type: none"> 1. The user clicks the “Upload Images” button. 2. The user navigates to select wanted images to execute the metrics on. 3. After uploading images, the user clicks “Run all metrics” 4. The program returns a report with quality scores of the selected images.
Alternative	<p>The user runs all metrics without uploading any images.</p> <ol style="list-style-type: none"> 1. The application displays an error and asks the user to upload images. <p>The user presses reset before running the metrics.</p> <ol style="list-style-type: none"> 1. Uploading images again is necessary.
Pre-condition	The application is running and images are uploaded.
Post-condition	The FIQMs have predicted the perceived face quality on the images.

4.3 Choice of Front- and Backend

When choosing the development software, we had to take some considerations. First, both backend and frontend software should not be time consuming to master, given that the project consisted of coding and research. Second, it should be uncomplicated to add new FIQMs to the backend as well as creating an intelligible user interface.

4.3.1 Backend

The two FIQMs FaceQnet and ISO Metrics were written in Python. Since we were experienced with Python and operated with it in courses throughout the bachelor’s program, it was natural to choose a Python framework for the backend. With a dozen of possible web frameworks, we needed to select a framework which suited our needs in terms of scalability, performance and ease of use. In the end, the choices were between Django and Flask.

Django

Django is a fast framework, which quickens the development process for developers. It has a high level of scalability to the users. This feature is the reason many leading websites depend on Django to fulfill their high operational requirements. The framework includes several rebuilt development features such as user authentication, sitemaps and content administration. It has excellent security, preventing the users from several security issues. Django is very flexible as it can be used to create a wide specter of application types. Some of these are social networking sites like Instagram or content management systems such as Wagtail [20].

When it comes to the disadvantages, Django has a steep learning curve. Even though it is written in Python, it takes a long time for developers to get the hang of it. The framework is considered one of the hardest to master. Django is more suitable for large scale applications rather than smaller products with fewer features and requirements. The unique functionalities within Django can be confusing for developers working with a small project. Djangos' monolithic architecture has a small number of dependencies which make it challenging to use. It does not facilitate developers to utilize Python packages and tools, but focuses on code-oriented programming. Django can not provide fast development in terms of requests. Only one request at the time can be fulfilled, meaning it is unable to handle multiple requests concurrently [21].

Flask

For programmers with experience in Python, it is easy adaptable to work with Flask. This micro framework is simple to manage as there are few standards. Flask' modular nature let developers instantly create servers and applications, which are distributed across comprehensive networks with certain purposes. It is pliable, meaning that components within the framework are easy to modify, because it is simple to configure. Given that Flask is a micro framework, it has less abstraction layers between the users and the database, cache and requests. This design provide users high level of performance [20].

Flask also has some disadvantages. Many beginner web developers tend to use the Flask framework, resulting in low quality code and possibly a bad application. Flask has singular source, meaning that it handles requests in turns. With multiple requests, it could be time consuming to handle the requests. The use of modules in Flask raises security issues. It would be bad if a module contained spiteful data.

Our Choice

Initially, we started the backend development process using Django. This was mainly because the team working with the backend had learned about the framework in an earlier course of the bachelor's program. However, we eventually concluded not to use Django for the backend. Given some knowledge in the framework, we had not actually developed anything with it. Since Django did not provide

the usage of Python packages or tools, it would be more time consuming to code the application. The steep learning curve provided by the framework would make the development process even more time consuming. Our project differed in working tasks, which made the application smaller in terms of features and requirements. Django was more suitable for larger applications, making the decision of not using the framework clearer. We ended up using Flask for the backend. Immediately after installing the framework, the development productivity increased. Our Python experience was easily adaptable to Flask, which made it simple to make a deployable application and add FIQMs in the program.

4.3.2 Frontend

As mentioned in Section 4.1, the central ideas of how Mobai wanted the application to be built were simplicity, which was adaptable when choosing how to build the frontend. Mobai's vision was for it to be a demo to show some of the key concepts provided by the backend, like displaying results from different FIQMs on a small amount of images. It could also be used as a demonstration to Mobai's customers. Since we decided to build the frontend as a web application, we quickly understood that this would be coded in HTML, CSS and JavaScript.

The backend framework Flask explained in Section 4.3.1 has a way of rendering simple HTML pages by itself. In the early stages of our development, we built a simple HTML page consisting of buttons and text and had Flask render the template. This was a good way of testing the backend functionality. Using this method would simplify the frontend and make the application run solely on the backend port 5000. The problem with this method was that the HTML pages rendered were quite static and had restrictions in form of design and functionality. This also meant that the frontend would be incorporated into the backend, and the backend had to be started in order for the frontend to work. After a discussion with Mobai, we found out that this was not the best solution.

We wanted the frontend to run independently from the backend on its own server. This way, the frontend would not be dependent on the backend to run, and we would get more flexibility when it came to design and implementations. We could still program the frontend using basic HTML and JavaScript, but we found that using a well-developed web framework would enhance the design and simplify for future development of the application. Angular and React are the two frameworks we considered to use.

React

React is a JavaScript library, but is often referred to as a whole framework because it is directly comparable to other web development frameworks. React is component based, which means the user interface can be rendered using different components. By doing this, the same component can be rendered multiple times without duplicating code and it will be easier to fix bugs because they are likely caused by one specific component. This is one of the key factors that makes

React flexible and dynamic. This also makes React applications easier to test and maintain. Considering it is a JavaScript library, learning is easier for any developer that has a JavaScript background, and also more lightweight than other frameworks. React uses its own virtual Document Object Model (DOM). This DOM is a cross-platform API which deals with HTML, XML or XHTML. The React virtual DOM exists entirely in memory and is a representation of the web browser's DOM. This means that when we want to render React, the virtual components will turn into the DOM, leading to smoother and faster performance [22].

React has a high pace of development, which can be both an advantage and a disadvantage. Since the environment constantly changes, some developers might not feel comfortable relearning the new ways of doing things regularly. It may be hard to adapt to new ways of doing things constantly and always be updated. Because of the constant updating, the technology is accelerating so fast that there might not be enough time to make proper documentation.

Angular

Angular is an open-source software engineering platform used to build UIs. The main benefit of using Angular is to turn static HTML-based documents into dynamic content using components. It was built on the Model-View-Controller architecture, where the framework is synchronized with the model and the view. This is called two-way binding. This allows developers that use Angular to reduce the development time as it will not require additional code to synchronize the model and the view. Angular uses something called dependency injection. Dependencies define how different pieces of code interacts with each other and how changing one component affects others. Using dependency injection, dependencies can be defined externally outside components, making the dependencies decoupled from the components. This makes the components more reusable, easier to manage and test [23].

In regard to the disadvantages, Angular is a framework that always has multiple ways of completing any task. Therefore the framework has a steep learning curve. Due to Angular's two-way binding, it may sometimes cause performance issues when rendering multiple components, especially on older devices [24]. Since Angular is a complete framework, it is created for building enterprise-scale applications, making the framework less flexible than other lightweight frameworks [23].

Our Choice

We ended up choosing React, considering its great performance, flexibility and its ability to create dynamic web pages. React has its own Command Line Interface (CLI) tool that makes building a React app easy. This command is called "create-react-app" and automatically builds up all files, packages and folders that are needed to run a React application [25]. The command also serves the React app

with a server using Node.js¹, which is an open source, cross platform, JavaScript runtime environment that gives the ability to run web pages on server side [26]. In order to run the CLI tool, Node.js would have to be installed first. Using this method, a React application could easily be installed and the web page be run on a local server on port 3000 using the Node.js command “npm start”. This way we have created a frontend independent of the backend running on port 3000.

4.4 Design and Implementation

In this section we will explain the design of our front- and backend applications, as well as show some key implementations from the requirements and use cases.

4.4.1 Backend

The backend’s implementation and functionality were developed using purely Python. The Flask framework explained in Section 4.3 is used to communicate with the frontend. When we start the backend, the Flask framework will start a server running on port 5000 on the local computer. From there, Flask will wait for HTTP requests in forms of GET or POST and execute the code based on what request was sent. This is what we call the REST API. Code listing 4.1 shows how we have implemented a REST API call for the use case “Run all metrics” showcased in the use case diagram in Figure 4.1.

Code listing 4.1: API call implementation.

```

1 @app.route("/allAlgorithms", methods=['POST', 'GET'])
2 def runAllAlgorithms():
3     path='uploads/'
4     csv_results = 'results.csv'
5     json_results = 'results.json'
6     if request.method == 'POST':
7         if (os.listdir(path)):
8             fun.runISO_Metrics();
9             fun.runFaceQNet();
10            if os.path.exists(csv_results):
11                os.remove(csv_results)
12            fun.save_results_to_file('ISO_metrics/results.csv', 'ISO')
13            fun.save_results_to_file('FaceQnet/scores_quality.csv', 'FaceQNet')
14            if os.path.exists(json_results):
15                os.remove(json_results)
16            fun.save_results_to_json(csv_results)
17            return send_file(json_results)

```

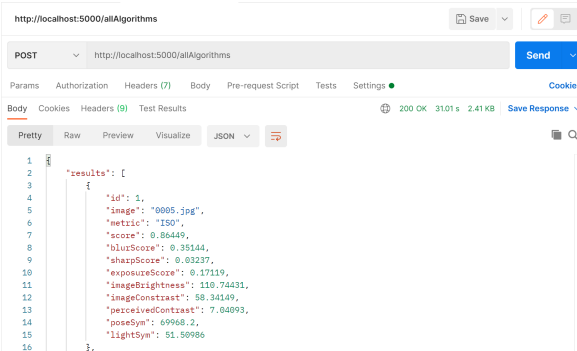
This code is run when the backend receives an API call with a route to “/allAlgorithms”. The code checks if there are images uploaded to the backend, and if there are, all FIQMs are executed on the uploaded images and the results will be written to one single file for all metrics and sent to the frontend for display. This function returns a JSON file that gets sent to the frontend. This is because of a

¹<https://nodejs.org/en/>

request Mobai had to save the results in a JSON format in case the data were going to be modified or used in the future. Our implementations for all use cases in our application follow the same structure as in Code listing 4.1, but with different routes and code.

For Mobai, the possibility to easily add more FIQMs into the backend was important, and one of the reasons why we chose to have an API call that can run all the metrics in one go. To simplify the process of adding more FIQMs, we have created independent functions in Python that do not rely on any FIQMs to work and can be used for multiple purposes in the future. Two of these functions are called “save_results_to_file” and “save_results_to_json” and are used in Code listing 4.1 to combine the results from both FIQMs into one single file, either in JSON or CSV format. Another step we took to simplify the adding of new FIQMs was to restructure the metrics code into Python functions. This way, we could run the different FIQMs by calling its specific function, and then save the results using the function “save_results_to_file”. These are options that make it easier for Mobai to add new metrics to the backed and reuse the code that we have already created.

Mobai also wanted the backend to work independently on the frontend, meaning that all functionality available in the frontend should be able to run in the backend alone. This includes running FIQMs and uploading images. Uploading images can be done by simply drag-and-dropping images into the “Uploads” folder in the backend, but running the FIQMs would require accessing the REST API without the use of the frontend. To solve this, we used another third party software to send HTTP requests to the backend. Postman² is such an application that can simulate a client and send POST and GET requests to the backend. By using Postman, we can send a POST request to the backend and get the returned results from a JSON file displayed. Figure 4.2 shows a POST request being sent to the backend using postman and then displaying the results from the JSON file. Consider the code in Code listing 4.1 to run when the API call gets sent.



```

1  "results": [
2    {
3      "id": 1,
4      "image": "0005.jpg",
5      "metric": "ISO",
6      "score": 0.86449,
7      "blusScore": 0.35144,
8      "shapScore": 0.03237,
9      "exposureScore": 0.17219,
10     "imageBrightness": 110.74431,
11     "imageContrast": 58.34149,
12     "perceivedContrast": 7.04093,
13     "passSym": 69968.2,
14     "lightSym": 51.50986
15   }
16 ]

```

Figure 4.2: POST request with Postman.

²<https://www.postman.com/>

4.4.2 Frontend

Due to React's popularity, multiple UI frameworks have been built to simplify the process of building web pages. One of these frameworks are called Material-UI³. Material-UI delivers already-built components, styles, icons and templates that support the React framework. Most of these components are free and can be used to easily get started with building a web page. When we discussed how to build the web page, we came to a conclusion that using the Material-UI framework would be beneficial to the simplicity, efficiency and design of our frontend. Different Material-UI libraries could be installed using the Node.js command “npm install”. We also chose to start the development process using a Material-UI pre-built pattern, called “Album”, that can be found on the Material-UI website [27].

To build dynamic web pages, React introduced a new addition in version 16.8, called “React hooks”. This addition provides a way of storing data as a state and the data gets updated automatically once the state is changed. This way we could change text fields, counters and data and display them automatically without refreshing the website, because they are stored in a state. We have used React hooks to store data that we know can be modified during the web applications lifetime. Code listing 4.2 is an example of how we have implemented a React hook to store images for the use case “Upload Images”, showcased in the use case diagram in Figure 4.1.

Code listing 4.2: React hook implementation.

```
1 const [ imagesSelected, setImagesSelected ] = useState([]);
```

The “imagesSelected” variable stores the state of what images have been selected from the user, and the “setImagesSelected” function updates the state every time a user selects images. If we wanted to display the images selected by the user, setting the variable name in a paragraph would suffice, and React would update the paragraph automatically once the state had been updated.

In order for the frontend functionality to work, the frontend would have to connect to the backend. This was done by sending a HTTP request in the form of a GET or POST command to the backend. The backend would then execute the specific code in the REST API based on what request was sent. React has different libraries that can send HTTP requests. We have chosen to use Axios to send HTTP request, because it is both compatible with the Node.js framework and can also be installed using the “npm install” command. By using Axios, we sent HTTP requests from the frontend to the backend. The backend would then return the desired results, either in the form of text or a file. The use case “Run all metrics” showcased in the use case diagram in Figure 4.1 was implemented using Axios. Code listing 4.3 shows our implementation.

³<https://material-ui.com/>

Code listing 4.3: Axios HTTP request implementation.

```
1 const runAllAlgorithmsHandler = () => {
2   setLoading(true);
3   axios({
4     url: 'http://localhost:5000/allAlgorithms',
5     method: 'POST'
6   }).then((response) => {
7     setFile(response.data);
8     setLoading(false);
9     console.log(response.data);
10  });
11  };
```

Using Axios, we send a POST request defining the port and the route to the backend REST API. The backend then executes all algorithms and returns the results in the form of a JSON file. React saves the results as a state and displays the state dynamically on the webpage.

4.4.3 Deployment

One of the main requirements given by Mobai was to deploy the application in containers. Containerization [28] provided us with several benefits:

- A container is independent of the operating system, making it portable between dissimilar platforms.
- Containers are efficient in terms of allowing applications to quickly be deployed, patched or scaled.
- Containers consume less system overhead than regular hardware or virtual environments owing to the fact that they do not include operating system images.
- Containers provide better application development, production cycles and testing as containers support an agile workflow.
- Having the application separated into different containers will improve security.

Once functionality for the application were mostly fulfilled, the task was to package it into two separate containers - one for the frontend and one for the backend. The basis of containers, images, were created using Dockerfiles. Information that automated the application running process was stored inside the Dockerfiles. This included information such as the base Docker image to run from (i.e., based on Alpine since the application was built in Python), location to project code, dependencies that the application had and which commands to run at start-up. When both Dockerfiles were written, the command “docker build” produced two disparate images. To enable both images to run together in a remote environment, we needed to use the Docker Compose tool. We created a Docker Compose file which included the definition of services that built the application. The command “docker-compose up” ran the application from our Docker Compose configuration file making the images become containers running on a Docker engine.

4.5 Sequence Diagrams

In this section, two sequence diagrams, one high level and one low level, will be showcased and discussed. Our web application is simple in design, but behind the scenes several entities are interacting with each other. To firmly understand how the application is constructed, it is essential to acknowledge the interaction between the components. Sequence diagrams are convenient, easy to read tools to visualize a given interaction within a time sequence. Providing both high and low level sequence diagrams will yield a complete picture of our system from a surface to an in-depth level.

4.5.1 High Level - ISO Metrics

Figure 4.3 is meant as a surface level visualization of what happens the first time the web application is started on a new computer and a metric is executed. After the two Docker images are built, the application is now ready to run.

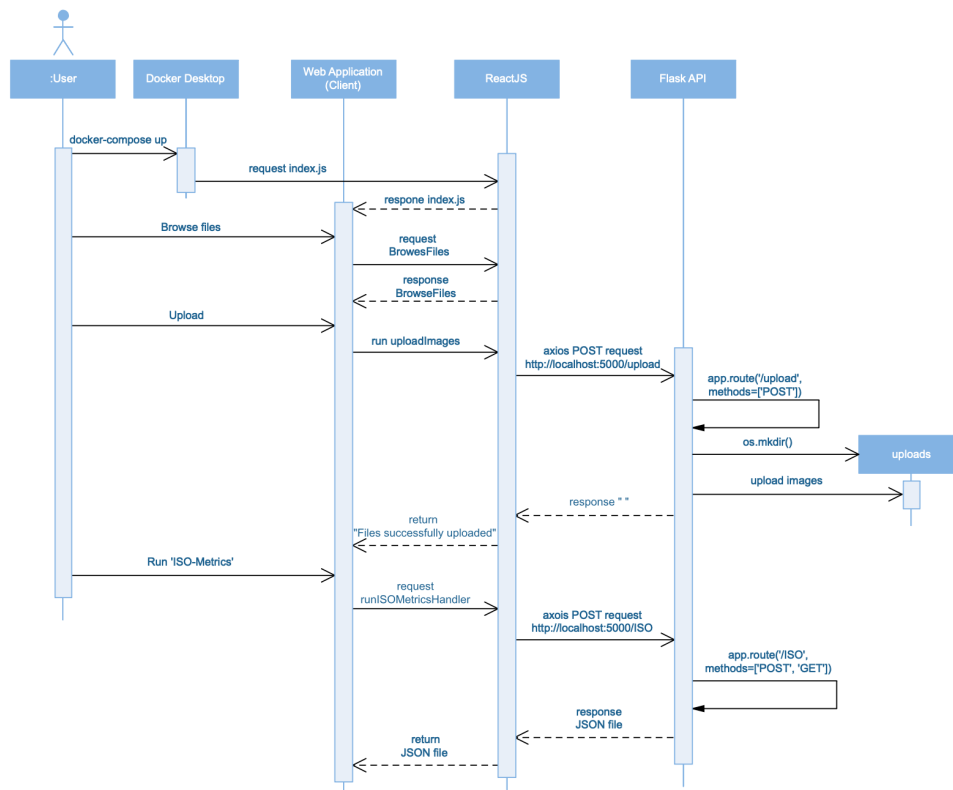


Figure 4.3: ISO Metrics: high level sequence diagram.

From the terminal window, the command: “docker-compose up”, will boot up both the React and Flask servers. An immediate request to render the index.js file in the web application is sent to the React server. After the home page is displayed

in the web browser, the user can press a button to browse files, which will send a request to the React server to open a window of your local files. Images are now ready to be selected. The selected images are stored in a state that waits for the “Upload” button to be pressed. Once the user presses the button, a function called “uploadImages” will be executed from the React server that in return redirects an Axios HTTP POST request to the external Flask backend API at port 5000. The POST request contains all the selected images. An API by the name of “upload” saves all selected images in a specified “uploads” folder in the backend before it is redirected back to the React server that displays a “Files successfully uploaded” message to the web application. Since this is the first time the application is executed, the “uploads” folder has not yet been created. The upload API therefore has to create it before uploading the images and returning to the web application.

Once the images are saved in the backend, the web application is now ready to execute any metric. The button “Run ISO-Metrics” on the web page is clicked which requests the function “runISOMetricsHandler” to be executed on the React server. This function will likewise redirect an Axios POST request to port 5000. An API by the name of “ISO” will receive the request which triggers the API to execute ISO Metrics. Once the metric has generated quality scores on the uploaded images, it returns a single JSON file that contains the scores. This JSON file is returned to the React server which in return displays the scores in the web page.

4.5.2 Low Level - FaceQnet

Unlike Figure 4.3, Figure 4.4 is a sequence diagram for how FaceQnet is executed the first time it is ran on a computer. Before the “Run FaceQnet” button is clicked, it is given that images have been selected and uploaded to the “uploads” folder which is shown in Figure 4.3. For that reason, the “uploads” folder already exists instead of being created along the way. The first few steps are quite similar to the sequence diagram depicted in Figure 4.3, however this time an API called “FaceQnet” is executed that in return starts the FaceQnet FIQM.

The cropping phase of FaceQnet begins, and since this is the first time the API has been called, two folders by the name of “uploads_cropped” and “uploads_cropping_failed” are created. The function then enters a for-loop that reads all the uploaded images from the original “uploads” folder, before they are cropped. Unfortunately, the cropping tool provided by MTCNN sometimes fails to detect the face in the images which results in an empty image. An alternative scenario then occurs, which sends these images to “uploads_cropping_failed”, and they will automatically be provided with the lowest score possible. If no exception happens during the face cropping, the cropped image is saved to the “uploads_cropped” folder. After having finished the cropping, the API starts the scores calculation, which first has to read all images from “uploads_cropped”, resize them and then calculate their scores. The metric likewise returns a JSON file with all scores back to the web page.

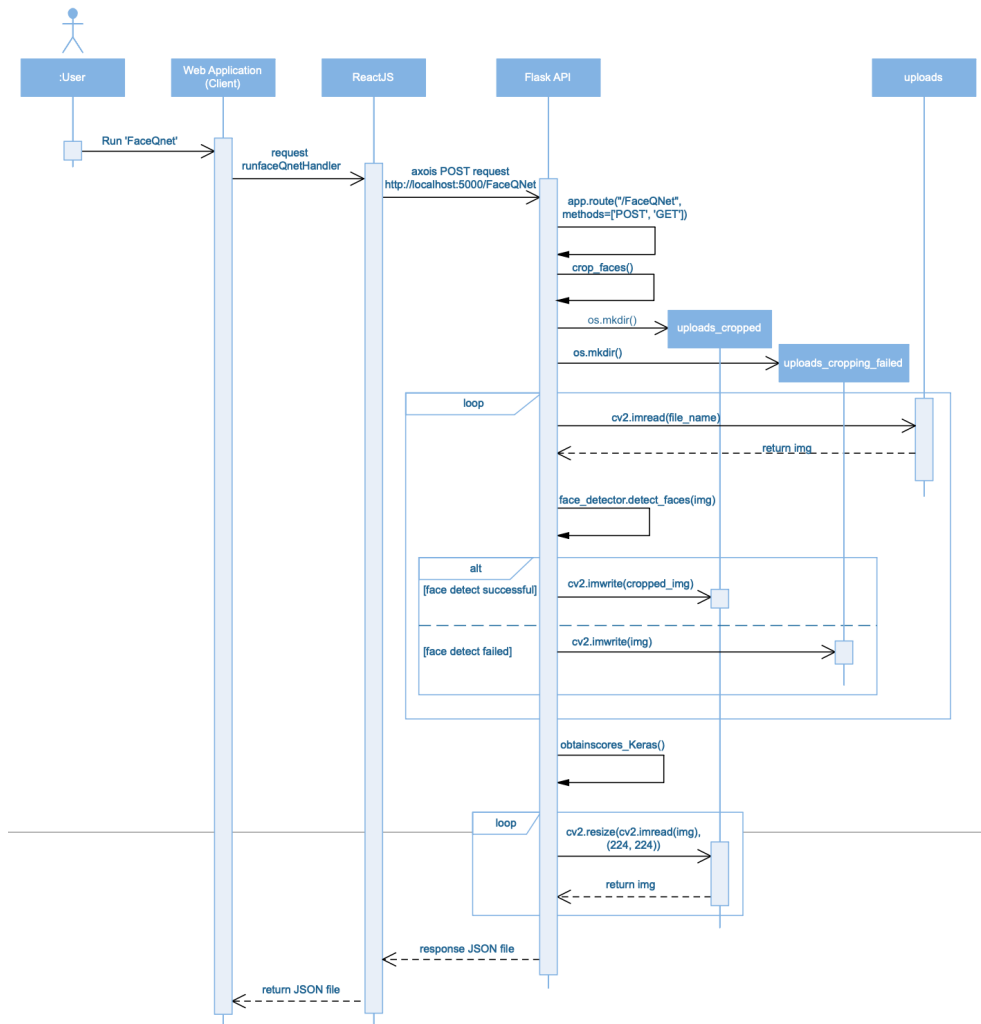


Figure 4.4: FaceQnet: low level sequence diagram.

4.6 Testing

Before we finalized the development of the web application, it was important to have the users and the product owner test the product and provide their feedback.

4.6.1 User Testing

We began the user testing at the end of the development process. The user testing was done by the product owner. While this might initially sound limited, we have to keep in mind that the application created is targeted for a small number of users at Mobai and the product owner will be a perfect representation of the users of our application. The testing began after the user received an email with a link to the Github repository containing the source code of the web application. The email

also included a list of tasks for the user to complete and a few questions about the user experience. The tasks and questions are listed in Table 4.6 and 4.7. The user testing was discussed with the product owner during our next meeting to expand upon the feedback we received. If there was no comment on the task or question, the feedback was set as “satisfactory”. The required changes were implemented as described in Table 4.5.

Table 4.5: Issues and solutions following the user testing.

Issue	Solution	Status
Headings in the web application has line breaks.	Edit the width of the React component from small to medium.	Done.
When the metrics are running in the backend, there is no text telling the user what is going on.	Add a text field next in the code next to the loading circle with the current status.	Done
The scores from the metrics have too many decimals. Change this to four decimals.	Add the round() function with four decimals to the metric scores.	Done.
The “Delete” button should be on the right side of the “Upload” button	Move the code for the “Delete” button to the correct position.	Done.
The attribute results from FaceQnet should not be set to zero.	Fix the code so that the attribute results from FaceQnet are set to -1. Add a text field explaining that -1 means that it does not return a score.	Done.

Table 4.6: User testing tasks and feedback.

Task	Feedback
Launch the frontend and the backend on the users’ computers using Docker.	Satisfactory.
Upload multiple images to the backend from the users’ computers using the frontend.	Satisfactory.
Run all metrics at the same time with the images uploaded in the second task.	Satisfactory.
Run each of the metrics separately with the images uploaded in the second task.	Satisfactory.
Delete the images uploaded to the backend.	Satisfactory.
Upload a single image to the backend from the users’ computers using the frontend.	Satisfactory
Run all metrics at the same time with the image uploaded in the sixth task.	Satisfactory
Run each of the metrics separately with the image uploaded in the sixth task.	Satisfactory.

Table 4.7: User testing questions and feedback.

Question	Feedback
How is the visual design?	The design looks good enough. Designers can update it later to make it better.
Is the user interface intuitive?	The user interface is simple and easy to understand.
Are the metric results displayed in a satisfying way?	Cut down the precision of the scores from the metrics down to four decimals. FaceQnet does not return any attribute scores and instead sets them to the number zero. This can create misunderstandings for the users. Change this to the text “None”. Another solution is to set the score to the number minus one and add the text “-1 indicates this approach does not return a score” to the frontend.
Any other feedback about the user experience?	A loading circle is shown when the metrics are working in the backend. Add the text “Processing is ongoing” to let the user know what is going on. The “Delete” button is located to the left of the “Browse Files” button. Move this to the right side of the “Upload” button. Some of the text in the headers of the web application is wrapping to a new line. Keep it on the same row.

Chapter 5

Subjective Experiment

This chapter contains the process of designing and conducting our subjective experiments. In Section 5.1 we explain how and why collecting ground truth data is important in this project. The next Section 5.2 addresses important aspects when creating a subjective experiment. These sections are followed by the introduction of datasets used in the experiment and application (Section 5.3) as well as an in-depth description of our subjective experiment in Section 5.4. At the end of this chapter we introduce the creation of our own dataset in Section 5.7 followed by our second subjective experiment in Section 5.8.

5.1 Why Collect Ground Truth Data?

Ground truth data is data based on human assessments. The ground truth data is collected by arranging subjective experiments where human observers are asked to evaluate a set of images according to given criteria [29]. In this project, observers graded facial images. Gathering ground truth data is an essential part of our work, because subjective data is needed to evaluate the performance of the objective assessments. Validation between subjective and objective assessments ensure whether the FIQMs correspond with how human observers rate facial images.

5.2 Subjective Experiment Aspects

As mentioned in Section 1.6.2, our lack of experience indicated that an introduction to subjective experiments was needed. The research about this topic made us realize the importance and difficulties with this task. There were several assessments to perform in terms of creating a comprehensive subjective experiment such as type of psychophysical experiment, duration of experiment and observer types.

5.2.1 Experiment Method

There were several psychophysical experiment methods to choose. We based our choice on the method that fitted our experiment the best. The options were between category judgment and rank order. Category judgment is a method where the observers rate facial images based on given criteria. These images are allocated to a category [29]. In rank order the observers are shown multiple images simultaneously and asked to rank them according to each other.

5.2.2 Experiment Duration

The experiment duration is an important factor when conducting a subjective experiment. On the one hand, it is essential to give the observers a high number of facial images to grade for data collecting purposes. On the other hand, having a too long experiment can cause observer fatigue such as eye fatigue or headache. These kinds of disorders will affect the subjective data negatively.

5.2.3 Observers

The subjective experiment was available for everyone to perform, but we invited a specific target group that works in this field. The observers were divided in two groups: experts and non-experts.

Experts

The experts are people who were specially invited to complete the experiment. These observers work either for Mobai with facial recognition systems or at NTNU in the Faculty of Information Technology and Electronics. A point of having experts in the field doing the subjective experiment was to quality assure our creation since they to a larger degree can distinguish between quality issues. In addition to that, we were to compare the results between the two groups.

Non-experts

Non-experts were people with no experience in the biometric field. They were our friends and family of different ages and educations.

5.3 Datasets

The usage of facial images in the application and subjective experiment raised GDPR issues. It would strike against the rules if we photographed people on the streets or collected random images on the web without consent from the subjects. The ground truth data in this project was collected on three different datasets provided by Mobai. Table 5.1 displays all datasets used in the project. Combined

passport alike was a high resolution dataset with images from the Tufts Face Database [30] and FEI Face Database [31], which has a similarity to passport or id photos, hence the title. The set consisted of 98 facial images of people with different poses, facial expressions and lighting. All images were taken using a Nikon D3100 with a dark blue or light grey background, as depicted in Figure 5.1. The camera distance was equal in every photo and the camera lens was in an even height with the faces.



Figure 5.1: Facial images from Combined passport alike.

Capture from photo was created by several employees at NTNU and combined with images from the CASIA Face Antispoofing Database [32]. All 70 images were selfies, taken with different phone cameras. Like Combined passport alike, the people had different poses and facial expressions. However, in this dataset, the images by the NTNU employees were photographed in multiple locations on campus, which made the background and lighting vary considerably. The CASIA images were captured from computer screens that would possibly affect the FIQM's evaluation scores. Due to GDPR issues, only images from CASIA are included in Figure 5.2.



Figure 5.2: Facial images from Capture from photo.

The Selfie dataset was a set of 126 images accessible on the web [33]. The images were selfies of different people at numerous locations: both indoors and outdoors. Given the selfies, the photos were taken with phone cameras and the camera to subject distance was short. A sample from the dataset is shown in Figure 5.3.

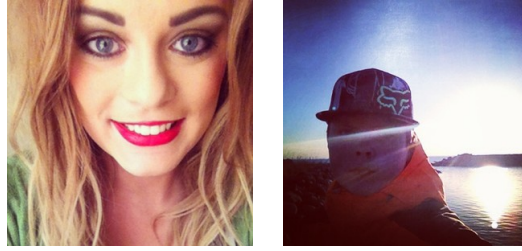


Figure 5.3: Facial images from Selfie dataset.

Table 5.1: Information about all datasets used in the project.

Dataset name	Number of images	Captured with	Distortions
Combined passport alike	98	Nikon D3100	Exposure
Captured from photo	70	Mobile phone	N/A
Selfie dataset	126	Mobile phone	N/A
Norwegian Facial Collection (Section 5.7)	450	Apple Iphone 8 Apple Iphone 11 Motorola Moto G5S Plus	Compression Blur Noise Face mask coverings Oblique angles

5.4 Our Subjective Experiment

In our subjective experiment we used category judgment. The observers were asked to grade facial images according to the criteria described in Section 5.4.1, and assign these images in five categories: poor, bad, fair, good and excellent. We did not chose the rank order system, because comparing multiple images would be challenging and time consuming for the observers. The experiment was separated into three sessions with a five-minute time break between each session. The duration of the subjective experiment was estimated to take approximately 45 minutes. Each session included images from the three datasets which were equally distributed among the sessions. For us, it was important to randomize the facial images to prevent observers getting biased towards one face. The subjective scores were collected from 15 observers - both experts and non-experts. Due to the GDPR, we only asked the observers' gender and if they were experts or not in this field. Out of the 15, 13 were men and two were women. Seven observers were experts, and eight were non-experts. Six experts were men, and one woman. Seven non-experts were men, and one non-expert was a woman.

5.4.1 Instruction Manual

When conducting the subjective experiment and collecting the ground truth data, we first needed to provide a detailed instruction to the observers. This was especially important in the case of non-experts who were not familiar with the task. Such an instruction manual will also allow us to provide a uniform understanding for the observers on how they should perform the subjective experiment. Thus, an

experiment instruction manual was created. The instruction manual is included in Appendix D

The first clarification we had to convey to the participants, was the meaning of face image quality. Both experts and non-experts would have different perceptions in terms of face image quality and we had to clarify all factors affecting the quality. The most clear and effective way to train the observers was to visualize examples of rated images like they would do in the subjective experiment. We created five example lineups with selfies of ourselves. Those lineups included most issues affecting the quality of an image.



Figure 5.4: A lineup from the experiment instruction manual and their quality ratings.

The images in Figure 5.4 is a sample lineup provided to the observers to guide them on how to rate the images. In the experiment however, some participants found it harder to rate the experiment images, because they included several image properties to take into account.

5.4.2 Choice of Subjective Experiment Platform

A decision we thought would be easy, but ended up being challenging and time consuming was selecting the platform in which we would run our subjective experiment. There were a number of possible platforms for conducting the subjective experiment. However, we had to bare in mind the factors described in Section 5.2 in our selection. One experiment platform we found interesting was Survey Monkey. The main reason we abstained from Survey Monkey was their demand of creating a yearly subscription to use functions such as displaying questions in random order. We even asked for a four month subscription without luck. Another platform we found appropriate was Survey Legend, a subjective experiment platform perfectly suitable for images. It included many functions we needed, except enabling questions to be ordered randomly for each observer. Therefore, we unfortunately had to abort Survey Legend as well.

After consulting with our supervisor, we ended up arranging the subjective experiment in QuickEval [34] - an experiment platform developed by NTNU. The platform provided all functionality we needed such as category judgment and random order of questions. Each image was positioned in the center of the screen on

a neutral background, we also took care that every image in the datasets were similar in size. QuickEval displayed statistics in the application gradually as people responded to the experiment. The statistics data was also downloadable as a CSV, HTML or Excel file. Using QuickEval as a experiment platform ended up being a good choice because it was created by NTNU and we had a close relationship with the developers. When an issue occurred, we submitted that issue and communicated directly with a developer, resulting in the issue to be fixed quickly.

5.5 COVID-19 Pandemic Implications

Due to the ongoing COVID-19 pandemic, limitations with the subjective experiment occurred. For example, we did not have the opportunity to create a controlled environment and invite the participants to complete the subjective experiment there. Ideally, such a controlled environment would be a well-lit room with good air quality, timed breaks between the sessions and a computer with specific screen settings for all participants to complete the experiment on. The Norwegian Colour and Visual Computing Laboratory in NTNU Gjøvik would be a perfect environment that could be standardized according to International Commission on Illumination (CIE) publications, but unfortunately this was not achievable given the COVID-19 restrictions. For that reason, all the participants completed the subjective experiments in an uncontrolled environment using their own computers with their custom screen resolution and lighting conditions. While factors such as the monitor brightness and color temperature and even the participants distance from the monitor as well as the illuminance of the room, plays an important role. We believe that due to the nature of our experiment (collecting ground truth data) the results collected in our work would be reliable and scientifically accurate. We should point out that while when it comes to evaluating the general quality of the image parameters such as the viewing condition and/or emotions of the observers play an important role this is normally not the case in the type of subjective data we were collecting.

5.6 What Went Wrong?

Despite conducting the subjective experiment in uncontrolled environments, the accomplishment of the experiment was successful. Due to the targeted invitation we sent to observers, a reasonable number of participant provided us with their subjective evaluation in a short time which resulted in us having more time analyzing the data. Since it was our first time creating this kind of experiment, some experiment aspects were left out of the consideration that could be important. The main flaw with this experiment was the lack of information about observes' characteristics, especially color blindness. Research show that Colour Vision Deficiency (CVD) affects eight percent of men and 0.5 percent of women in the world [35]. Given that 87 percent of the observers participating in our experiment were

men, there was a possibility that color blindness existed among the observers. People with a type of color blindness have the ability to discern colors to a lesser degree than people with normal color vision. Therefore, this will affect the way they assess the facial images, which is the reason why they are not contemplated as optimal observers [29]. A good solution would be to test all participants with standardized color vision tests like an Ishihara test [36] or a Cambridge color test [37].

Another experimental aspect we systematically excluded was to provide the observers with a question during the experiment. We had initially planned for the experiment to consist of a precise question to minimize any doubt the observers could have. A criterion we had in mind was: “How well is the face recognizable?”. We felt this question could confuse the observers given that the word recognize often refers to knowing a person in advance. Since most participants would not know any of the subjects in the datasets in advance, we dropped that question. However, since we created an instruction manual with aspects that affected the face quality, we figured the observers would have a clear understanding of what was to be rated. But looking back, a question like: “How visible is the face according to the criteria described in the instruction manual?”, could have been beneficial to clear up any misunderstandings.

5.7 Creating Our Own Dataset

When working with machine learning and AI in the context of face recognition, the probability of using one of the famous pre-curated Labeled Faces in the Wild (LFW) datasets are highly likely. Conditions such as poor lighting, extreme poses and face coverings are somewhat lacking in LFW and these are all important aspects for Mobai’s face recognition system. This provided us with the opportunity to collect a specialized dataset designed to fit Mobai’s needs which led to the proposal of a new dataset. This initiative was positively received by Mobai.

The idea to collect a new dataset came about when the team was discussing flaws with the Selfie dataset. Mobai’s definition of face quality is, as mentioned in Section 3.1, originally based on ISO 29794-5 and ICAO Doc 9303 Part 3. Based on those definitions, one could argue that the images in the Selfie dataset did not fulfill the criteria. A significant part of the images in the Selfie dataset are of faces that either are way off-centered, too zoomed in or a combination of both. We figured it could be valuable for Mobai to have a new dataset aligned with certain criteria they valued.

5.7.1 Creation Process

We had initially taken a few images of ourselves which were used for the instruction manual, illustrated in Figure 5.4, to the subjective experiment. These were included when we started collecting images for the dataset. We came to the conclusion that each day, starting 1. March and ending 15. April, we would capture

at least five selfies of ourselves every day. This would result in a total of 250 images of each member. We ended up capturing 1172 images. With this amount of images, there were a lot of repetitive facial images. Therefore, we selected 250 images from the collection. In addition, we selected 50 of these images and added different distortions (described in detail in Section 5.8) on them. The whole dataset ended up consisting of 450 images. Our dataset is relatively large in size in comparison to the datasets used in our experiment. Some of the images are similar to the Combined passport alike dataset in regards to pose, but our dataset includes several varieties and specialized conditions. We chose to name our dataset Norwegian Facial Collection (NFC).

All images were captured using both the front and back camera of our mobile phones. This led to our dataset containing images of varying resolution, given that our phones were not of the same type. The phones used for the dataset collection were:

- Apple iPhone 8
- Apple iPhone 11 (2 units)
- Motorola Moto G5S Plus

These had different camera specifications with the front cameras ranging from seven to 12 megapixels and the back cameras ranging from 12 to 13 megapixels. While the difference in image resolution was very minor making it barely noticeable to the user, it still is able to simulate what FIQMs will need to address in real life applications.

5.7.2 Distortions

Our dataset was inspired by the three datasets introduced in Section 5.3, but was unique because elements like oblique angled images and face masks were represented. During our image collection, we gathered examples of our faces with:

- Different lighting conditions.
- Different facial expressions and head poses.
- Different face and head coverings.
- Different camera angles and camera tilts.
- Different backgrounds.
- Different distortions such as compression, blur and noise.

A crucial consideration was to gather a high number of images that cover a wide range of varying qualities. Facial images of bad quality were equally as important as excellent quality facial images, because machine learning algorithms such as FIQMs rely on diversified images to learn.

Face Masks

One of the two significant aspects of our dataset was the inclusion of face mask images. The usage of face masks has drastically changed people's everyday lives

across the globe. This is especially true in western society, since face masks were almost non-existent in public areas before the corona pandemic. Seeing that face masks became a normal part of peoples lives, we figured Mobai’s face recognition system would have limited images with face masks to test their FIQMs. To achieve a variety of face mask images, we altered the coverage of the face masks. The common way to wear a face mask is for it to cover your mouth and nose. Some of the images were captured with those aspects taken into consideration, but a significant number were captured with the masks covering less of the face, as shown in Figure 5.5. These images were expected to receive noteworthy different scores by ISO Metrics and FaceQnet because the visibility of the faces were dissimilar.



Figure 5.5: Different face masks usage.

Camera Angles

The other important contribution to our dataset, was that we experimented with different camera angles, also known as “oblique angles” or “Dutch angle”. These types of images involves tilting the camera at an oblique angle on its roll axis, which produces images where the viewpoint is similar to tilting one’s head to the side. Images like these create a form of disorientation because the camera has been rotated relative to the horizon of an image. This type of disorientation can be perceived by humans, but whether the FIQMs react differently towards these type of images is yet to be seen. For that reason, we wanted to look into oblique angle facial images and see if the FIQMs produce significantly different scores solely based on the camera angle.

As briefly mentioned in Section 1.4, finding datasets that addressed and included facial images taken from oblique angles were hard to come by. The closest datasets we came by, only included facial images where the subjects had their heads tilted and not the camera which does not correspond to the same. Not only are the inclusion of the oblique angle facial image quite new, but our dataset furthermore mixed these images with face masks, which created an even more distinct dataset.

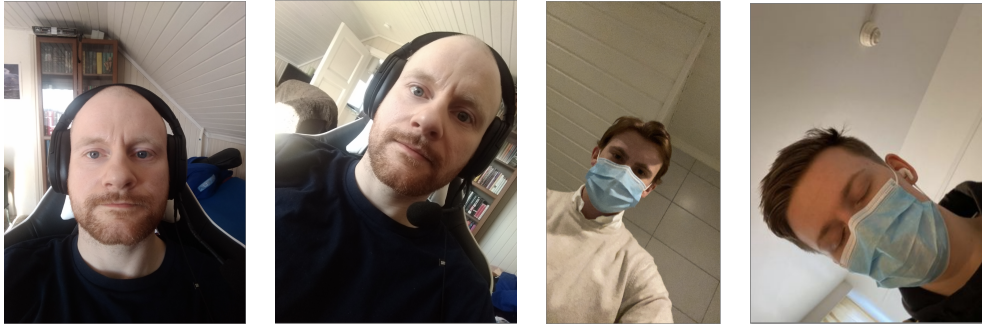


Figure 5.6: Different oblique angle camera shots.

5.8 Second Subjective Experiment

A key part we wanted to achieve while conducting the second subjective experiment was to make it as straightforward and understandable as possible. One should not have to be trained once more in order to understand and perform the subjective experiment. Therefore, we designed the second experiment as similar to our first one, and invited no new participants. The ground truth data was collected by running the experiment with six participants who were the bachelor team, our supervisor and the product owner. The key elements of the experiment were:

- Randomizing the order of the images.
- Splitting the experiment into two phases.
- Using a category judgement.

The second experiment consisted of two phases. The first phase included the 250 undistorted images from NFC dataset without splitting it into several connected experiment sessions as we did in our first experiment. Experience from our first experiment showed that the average duration to finish a session with 100 images was shorter than expected. We had initially planned for 100 images to be included in each session with an estimated duration time of around 15 minutes, but feedback from our participants showed that in most cases the experiments was finished in six to eight minutes. Based on those results, a subjective experiment with 250 images should not take longer than 25 minutes which is below the maximum limit of 30 minutes suggested by International Telecommunication Union [38]. Although the experiment was 2.5 times larger than the sessions of the first experiment, only people directly involved with the project were expected to participate and so creating several sessions felt like unnecessary overhead.

Even though our subjective experiment had new ways and aspects of testing the two FIQMs, we wanted to test them even further in phase two. The FIQMs' ways of giving quality scores are based on how well they perceive faces, explained in Chapter 3, but the facial images' overall quality have an impact. We wanted to see how the two FIQMs would assess if we added distortions to the images. Based on this, we decided to extend our dataset by introducing different distortions to a

number of images in our own dataset. Based on the results from the 250 images used in phase one of our second subjective experiment, we selected 10 images from each quality category and added distortions to them. This gave us 50 images to add each distortion on. We chose to add four different distortions, giving us 200 distorted images. From there, we conducted another subjective experiment, identical in design from phase one and with the same number of subjects which we referred to as the second phase of the subjective experiments. This subjective experiment consisted of the 200 distorted images, as well as the 50 original images selected from each quality category to be used as control images. In total, giving us 250 images to evaluate.

The type of distortions we added had to be reproducible, that way future research could add the same distortions and expect similar results. We added the following distortions:

- Compression with Telegram messaging application ¹: The compression was done by messaging the images, resulting in change of the images' resolution and size.
- Compression with Adobe Photoshop (version 22.4): All images were compressed on compression level one.
- Noise with Adobe Photoshop: The images were added with noise of seven percent.
- Blur with Adobe Photoshop: The images were added a blur level of two.

Having two phases in the second subjective experiment, gave us a considerable amount of data to test the two FIQMs on. Based on phase one and phase two, we could compare the subjective results from both phases and see how the subject's scores differ with distortions added to the images. Ideally, both the FIQMs and the subjective results should not differ in the two phases as long as the faces are visible in the images.

¹<https://telegram.org/>

Chapter 6

Results and Discussion

In this chapter we provide the reader with a detailed evaluation of the FIQMs and their correlation with the subjective scores collected in our subjective experiments. Since a large part of our bachelor project consisted of face image quality research, this chapter is of great importance. After conducting the experiments in Chapter 5, we gathered a large amount of data, which we had to process (Appendix E). This data will be presented in the form of statistical calculations, such as standard deviation graphs, correlation graphs, histograms and spider charts. The main goal of the chapter is to tie together the results we achieved in the objective and subjective face quality assessment, discuss them and present an in-depth look into the performance of the FIQMs relative to our collected ground truth data. The results we achieved are split into two sections. Section 6.1 covers the results from the first subjective experiment based on the three datasets provided by Mobai, while Section 6.2 covers the results from our own dataset.

6.1 Main Experiment

In this first section the results of our first subjective experiment and the objective assessment will be presented and discussed. Both assessment categories are based on the three datasets provided by Mobai, which were described in Chapter 5.

6.1.1 Results of the Subjective Evaluation

As a first step in analyzing the subjective scores collected in our experiment, we calculated the mean opinion score (MOS) for the facial images in each dataset. Figure 6.1 shows the histogram of the MOS results in each dataset. The score distribution for Combined passport alike and Capture from photo were quite similar which was expected given that the images in the datasets were similar in context. This was not the case in the Selfie dataset where unlike the other two datasets, images were selfies taken by the people depicted in the image. The standard deviation was calculated for every image, but also for each dataset and all datasets together. The total average standard deviation ($\bar{\sigma}$) for both experts and non-experts

on all three datasets together was calculated to $\bar{\sigma} = 0.7488$. This confirmed that the participants had an equal understanding of what defined a good or bad facial image. The instruction manual likely affected the quality perception of the participants as anticipated. Given that we were collecting ground truth data, it was expected that the deviation should be low.

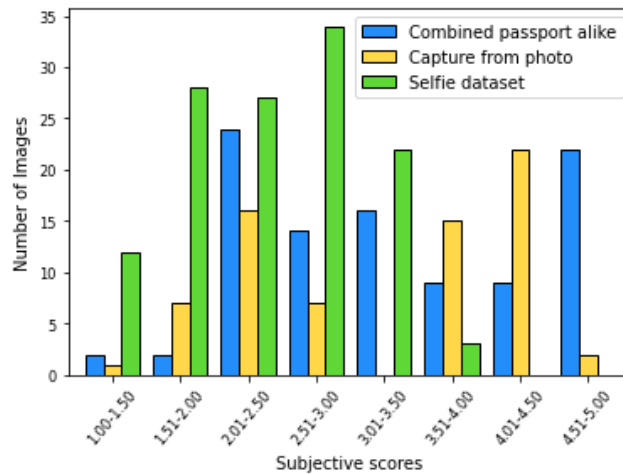
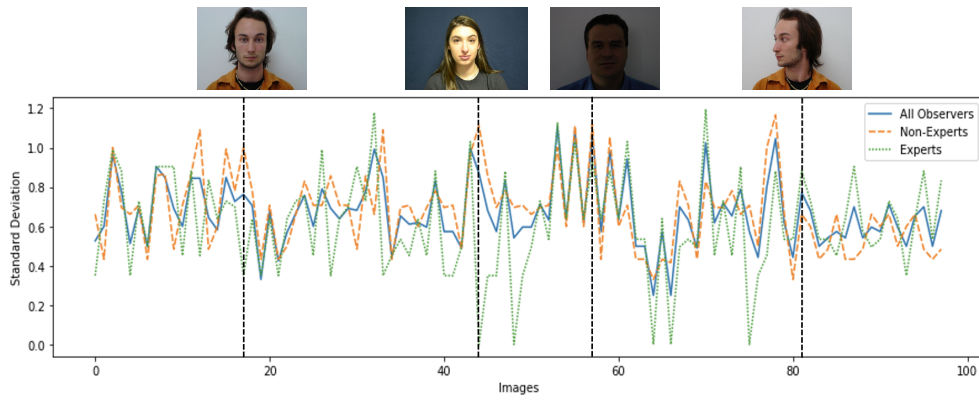


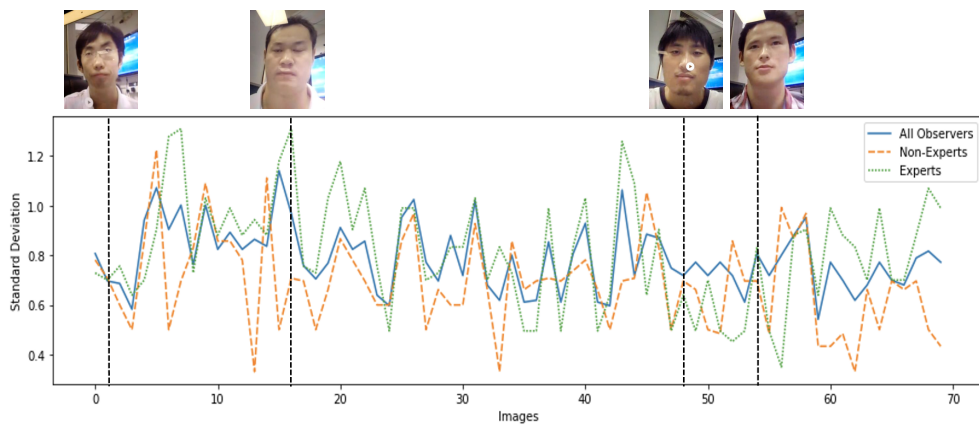
Figure 6.1: Histogram of MOS values calculated for the images in each dataset. There is a clear difference between the quality of the Selfie dataset and the two others.

Figure 6.2 depicts three line plots of the standard deviation of the subjective scores on all images in each dataset. The plots show the subjective score distribution and they take into consideration the total standard deviation and the deviation of each participant group. Whenever the three lines have a low spread between each other it indicates that the deviation of the participant groups were similar. The lower the deviation the more the majority of the participants agreed upon a similar score. This was especially apparent in the Combined passport alike dataset (Figure 6.2a) which also generated the lowest total standard deviation of $\sigma = 0.6707$. The two remaining datasets (Figure 6.2b and 6.2c) received a similar total standard deviation of $\sigma = 0.7891$ and $\sigma = 0.7872$ respectively. Some images had greater variety than others. The Selfie dataset contained some of those, but all in all the general participants rated images alike. Even between the experts and non-experts, the evaluation was more or less equal. In fact non-experts had a slightly better standard deviation of $\sigma = 0.6908$, than the experts of $\sigma = 0.7608$ on the three datasets.

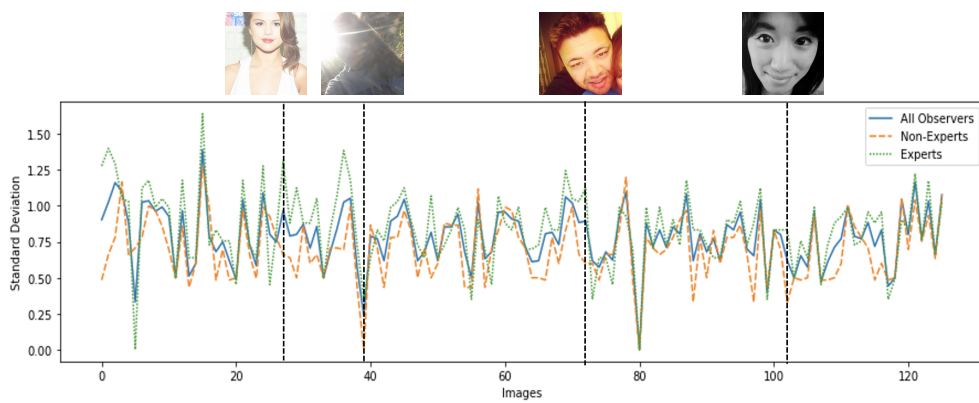
Four images were chosen and previewed from each dataset. The first plot, Figure 6.2a, which showcases Combined passport alike, the first and second images would be considered excellent facial images. The experts rated these images very similarly which is apparent by the low deviation scores. The spread amongst non-experts were greater with deviations around 1.0. The two remaining images were of lower face quality, but the deviations were not affected by that.



(a) Combined passport alike dataset



(b) Capture from photo dataset



(c) Selfie dataset

Figure 6.2: Standard deviation values of the subjective scores given to each image in the three datasets. Sample images from each dataset are shown as examples.

The four next facial images of Capture from photo in Figure 6.2b shows an overall equal standard deviation between the non-experts and experts. However, the second image had notable differences in the subjective scores between non-experts and experts. The standard deviation amongst experts were close to 1.2, which was the highest of the dataset. The great distribution of scores was likely due to the nature of the facial image. On one side the face is visible, but on the other side the image is taken from a computer screen which has heavily reduced the appearance of the face. This also applies to the last two images, however the standard deviation was lower and more even between the experts and non-experts.

The last four previewed facial images belong to the Selfie dataset (Figure 6.2c). The first image had an total deviation of 0.90. It was highly likely that the deviation of 0.90 was because the face was off-centered and not fully shown, which was the case for a large portion of the Selfie dataset images. However the second image had considerably lower standard deviation values. The face was barely visible which both participant groups assessed similarly. The scores of the experts on the last two images were more spread than the non-experts.

6.1.2 Evaluating the Performance of FIQMs

The FIQMs presented in Chapter 3 use different approaches to predict the perceived face quality. Their predicted perception of quality is supposed to correlate with human assessment, which is the whole purpose of FIQMs. The objective predictions produced by the FIQMs were measured against human assessment to evaluate the accuracy of the proposed approaches. This evaluation was carried out by calculating the correlation coefficients. The sample Pearson correlation coefficient (r) was calculated as:

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \quad (6.1)$$

where n is the sample size, x_i and y_i denotes the individual sample scores from the objective and the subjective assessment respectively, while $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$ and $\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$ represents the sample means from the objective and the subjective scores. Another type of correlation coefficient, the Spearman rank correlation coefficient (ρ) [39] was also calculated because it does not assume that both variables are normally distributed.

The subjective scores were normalized in order to achieve the same ranking scale when being compared with the objective scores. Our subjective experiment initially had a categorical judgment with a scale from one to five, which was converted to scores between zero and one. It is easy to think that the MOS for each image should be divided by the number of score alternatives in the subjective experiment, which in this case was five. However it is worth noting that had we only divided the subjective scores by five, the lowest subjective scores achievable would be 0.2. This means that the FIQMs could provide scores from zero to one,

while the subjective scores were from 0.2 to one. Dividing by five would therefore provide us with an uneven scoring scale where a score of five equals 100% while the lowest score of one equals 20% instead of 0%. We needed a five-point scale that would increment the score alternatives with 25% starting from 0%. To properly convert a five-point scale to percentages we used the following equation (6.2) on every image:

$$MOS_{Normalized} = \frac{MOS - 1}{4} \quad (6.2)$$

ISO Metrics

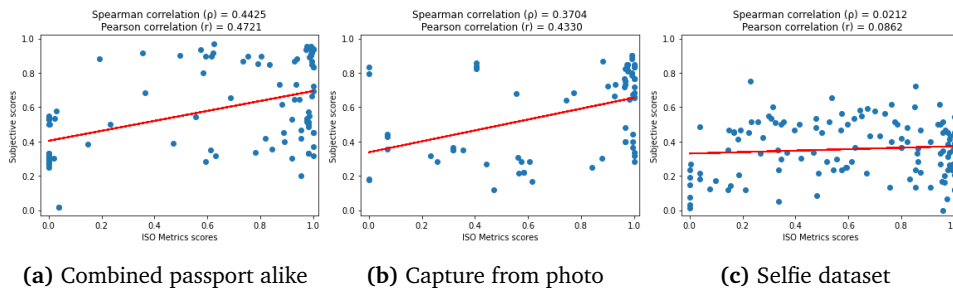


Figure 6.3: 2D scatter plots of the objective and subjective scores on the three datasets with objective scores along the x-axis and the subjective scores along the y-axis. The Spearman and Pearson correlation coefficients are shown above each plot.

As a first step in analyzing the performance of FIQMs, we calculate the correlation between the subjective scores and the objective scores calculated by the metrics. Figure 6.3 provides different plots of subjective against objective scores for the three introduced datasets. The correlation coefficients on Combined passport alike and Capture from photo were just shy of 0.5 which indicated a low to moderate association, whereas the correlation on the Selfie dataset was non-existent. The performance of ISO Metrics was clearly worse on the Selfie dataset relative to the two others. The FIQM was challenged by the dataset which was not surprising given that the images were of mediocre face quality which ISO Metrics often tends to over evaluate. In other words ISO Metrics was not suited for the Selfie dataset.

FaceQnet

The correlation between FaceQnet and the subjective scores regressed with each dataset the same way as ISO Metrics and the subjective scores did, shown in Figure 6.4. The performance of the two FIQMs was comparable on all the datasets. Combined passport alike achieved the highest correlation of the datasets, with Spearman and Pearson values similar to the ones in Figure 6.3a. Like ISO Metrics, the performance of FaceQnet on Capture from photo was worse than Combined

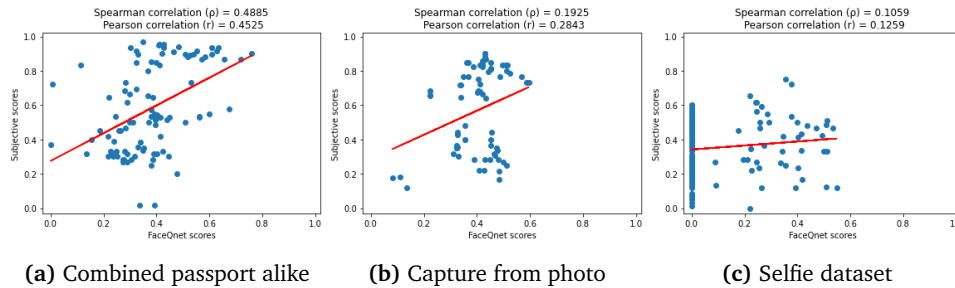


Figure 6.4: 2D scatter plots of the objective and subjective scores on the three datasets with objective scores along the x-axis and the subjective scores along the y-axis. The Spearman and Pearson correlation coefficients are shown above each plot.

passport alike, but this time FaceQnet performed slightly worse than ISO Metrics. The performance on the Selfie dataset was even worse with a non-existing negligible correlation. A large part of the Selfie dataset images were rated zero. The facial images rated zero were mostly because the cropping part of FaceQnet failed. This happened when the cropping part failed to detect the face of the images. The images were therefore manually provided a score of zero. In other words, FaceQnet is not a suitable FIQM for the Selfie dataset.

Could We Use Different FIQMs Simultaneously?

An interesting idea the team came up with, was if the correlation between the FIQMs and the subjective scores could be improved if a weighted average of the two FIQMs were used as the final FIQM. As a first step in such an approach, we simply gave the same weight to both FIQMs and tested the idea on the Combined passport alike and Capture from photo datasets. Since FaceQnet did not work for over 50% of the images in the Selfie dataset, we found no reason to test this approach on that dataset.

Figure 6.5 shows the correlation coefficients and the linear regression line calculated on the two datasets. The plot depicted in Figure 6.5a shows the highest correlation coefficients we ever achieved during our experiments. The r -value of 0.6045 and the ρ -value of 0.5942 were considerably higher than the correlation coefficients of the FIQMs individually. A value around 0.6 would indicate a moderate to strong correlation between the two data types. Even Capture from photo showed an increase in the correlation value. Initially, ISO Metrics performed better on the dataset than FaceQnet, but after the weighted average approach, the combined scores performed slightly better than ISO Metrics. Nevertheless, the correlation was still considered low to moderate.

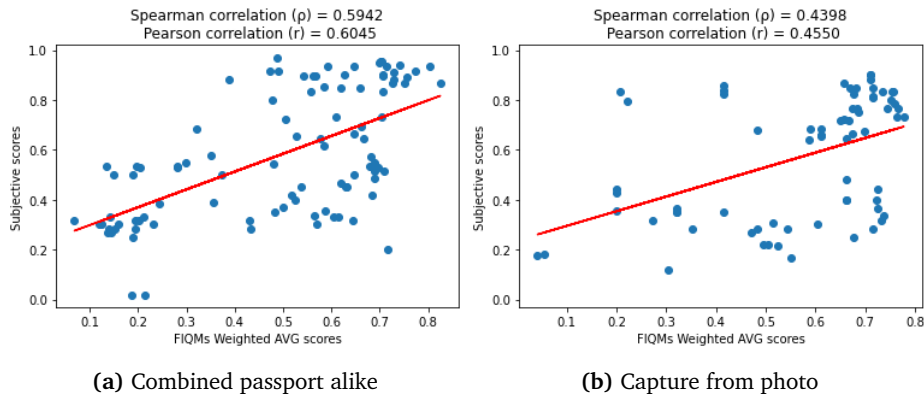


Figure 6.5: 2D scatter plots of the scores provided by using a weighted average of the FIQMs (with the same weight) on Combined passport alike and Capture from photo. The average FIQMs scores are displayed along the x-axis and subjective scores along the y-axis. The Spearman and Pearson correlation coefficients are presented above the plots.

Error Method

Plotting the results of the FIQMs against the subjective scores was not enough to assess the performance of the FIQMs on the datasets. Although the correlation coefficients on the datasets mostly were below 0.5 and indicated low to moderate correlation, error methods such as Root Mean Square Error (RMSE) can be used to track both the efficiency and accuracy of our FIQMs. The RMSE values were calculated by the following general formula:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (Predicted_i - Actual_i)^2} \quad (6.3)$$

In Equation 6.3 N corresponds to the number of samples, and in our case these were images. The predicted values denoted the objective scores by the FIQMs while the actual values were the corresponding subjective scores. This calculation gave us the standard deviation of the prediction errors. In other words, RMSE measured how far from the linear regression line the predicted values were. The closer to zero the scores were, the better the FIQMs predicted the right scores.

The highest correlation coefficient, without using the weighted average approach on the FIQMs, was achieved on the Combined passport alike dataset with ISO Metrics as shown in Figure 6.3a. Although the correlation was around 0.5, the RMSE value was around 0.36, shown in Table 6.1. This meant that on average, the perceived quality predicted by ISO Metrics was off by ± 0.36 points compared to the actual subjective scores. Since our scale was from zero to one, 0.36 points off equaled an error of $\pm 36\%$ which was remarkably high. Even worse was the RMSE value on the Selfie dataset with ISO Metrics with a staggering error of 43%. Despite having a lower correlation with the subjective scores on average,

Table 6.1: The calculated RMSE values of the FIQMs on the datasets relative to the subjective scores. The RMSE value was not calculated for the Selfie dataset with a weighted average of the FIQMs. The 'X' symbolises this.

RMSE values	ISO Metrics	FaceQnet	FIQMs Weighted AVG
Combined passport alike	0.3675	0.3031	0.2262
Capture from photo	0.3551	0.2887	0.2309
Selfie dataset	0.4342	0.3253	X

FaceQnet outperformed ISO Metrics on all datasets with its RMSE values. On average, FaceQnet's errors were 0.8 points lower than ISO Metrics which equaled an error decrease of 20%.

The interesting results were those of the weighted average FIQMs. They achieved considerably better values than the FIQMs individually. Although an error of ± 0.22 and ± 0.23 in our case would be considered mediocre, the decrease could not be overlooked. Weighted average FIQMs had a decreased error rate of ± 0.14 points on Combined passport alike and ± 0.08 , relative to ISO Metrics and FaceQnet. This equaled a decrease of 38.5% and 25.4% respectively. The accuracy of weighted average FIQMs was similar on the Capture from photo dataset.

6.1.3 Significance of Our Results

Our results without any kind of validation were of little value. A validation had to be done in order to conclude whether our results were statistical significant. We had to be confident that our results could be replicated and that they did not occur by coincidence.

The plots depicted in Figure 6.3 and Figure 6.4 showed for the most part, except for the Selfie dataset, a low to moderate correlation between the objective and subjective scores. The correlation coefficients were calculated with a 95% confidence interval, which gave us a significance level of 0.05. The p-values of the correlation coefficients were calculated and the lower the p-value the more statistical significant our results were. Anything above a value of $p = 0.05$ was considered insignificant, because it meant that the likelihood of an uncorrelated system providing those exact coefficients randomly would be 5%.

Table 6.2 show only the correlation coefficients that were considered statistically significant, i.e. we were 95% confident the correlation coefficients did not occur by coincidence. The coefficients were replaced by an 'X' symbol where the p-values were greater than the significance level and should therefore be taken with a grain of salt. All results regarding whether the FIQMs were correlated with the subjective scores on the Selfie dataset were confirmed to be uncorrelated with the high p-values. With a combination of high p-values and low correlation coefficients, it is highly likely that the objective and subjective scores were uncorrelated. The results gathered from the remaining datasets were within the significance level and were considered accurate.

Table 6.2: All correlation coefficients on the three datasets between ISO Metrics, FaceQnet, FIQMs Weighted AVG and the subjective scores. The ‘X’-symbol indicated the correlation coefficients that had a p-value higher than 0.05 and were therefore ignored. The coefficients were not calculated where the ‘-’ symbol is placed.

	ISO Metrics		FaceQnet		FIQMs Weighted AVG	
	Spearman ρ	Pearson r	Spearman ρ	Pearson r	Spearman ρ	Person r
Combined passport alike	0.4425	0.4721	0.4885	0.4525	0.5942	0.6045
Capture from photo	0.3704	0.4330	X	0.2843	0.4398	0.4550
Selfie dataset	X	X	X	X	-	-

6.2 Second Experiment

This section is about the second experiment we conducted based on our own NFC dataset of 450 images, and the results we achieved. A key goal the team considered was having the NFC dataset consist of varying face quality to achieve a balanced dataset. A histogram of the MOS values calculated in the second subjective experiment on the NFC dataset is depicted in Figure 6.6. One can see how the different quality categories are represented to an acceptable level. Images of specific quality were not over represented relative to the other quality categories and there exist a nice distribution in the dataset with images of varying quality which was the goal of the group.

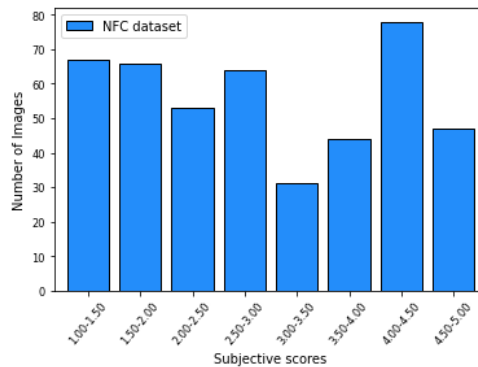


Figure 6.6: Histogram of the subjective scores on the NFC dataset.

6.2.1 Results of the Subjective Evaluation

The MOS of each image was calculated as we did in the first subjective experiment in Section 6.1.1. The subjective scores of the NFC dataset provided by the participants correlated closely. The average standard deviation of the subjective scores was calculated to be $\bar{\sigma} = 0.53$ which was substantially lower than those of the datasets in the first subjective experiment mentioned in Section 6.1.1. This time the experiment was not split into several sessions to prevent fatigue even though

it consisted of several times the facial images of the first subjective experiment. With that in mind the average standard deviation was lower, which goes to show how the subjective experiment aspects mentioned in Section 5.2 mattered to a lesser degree since we were collecting ground truth data. Provided that six subjects participated in the subjective experiment and only two of those were experts we found no reason to compare the standard deviations between those types of participants.

6.2.2 Results of the Objective Evaluation

An important metric we voluntarily wanted to measure was how the FIQMs performed on distorted images relative to their corresponding undistorted ones. This was tested on a subset of the NFC dataset which consisted of the 200 distorted images and their 50 corresponding original images. In order to assess how the FIQMs performed we first had to analyze how the subjective scores were distributed. A histogram of the MOS values for different types of distortions are shown in Figure 6.7. From Figure 6.7 it is clear that the distortions we added to the images did not have a considerable affect on the subjective scores. It is interesting to investigate how such distortions can affect the performance of FIQMs.

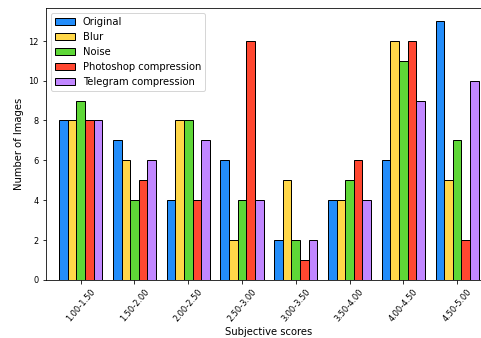


Figure 6.7: Histogram of the subjective scores of the original images and their corresponding distorted images in the NFC dataset.

Figure 6.8 showcases how the FIQMs performed on the original and distorted images. Even though the scores provided vastly differed from the subjective scores, one can see how the objective scores highly correlated between the different distortions. Among the FIQMs, FaceQnet had an overall better performance on the types of distortions we added relative to ISO Metrics. Its scores correlated close to perfect which is shown in Figure 6.9. Three out of four distortions had correlation coefficients above 0.8, which is considered a very strong positive correlation. Only the Telegram compression on FaceQnet performed slightly worse with a moderate to high correlation. ISO Metrics performed likewise, however the scores were more spread as shown in Figure 6.8a. The scores of facial images with noise differed noticeably from its original ones. Facial images with noise also had a significantly lower correlation than the other distortions, shown

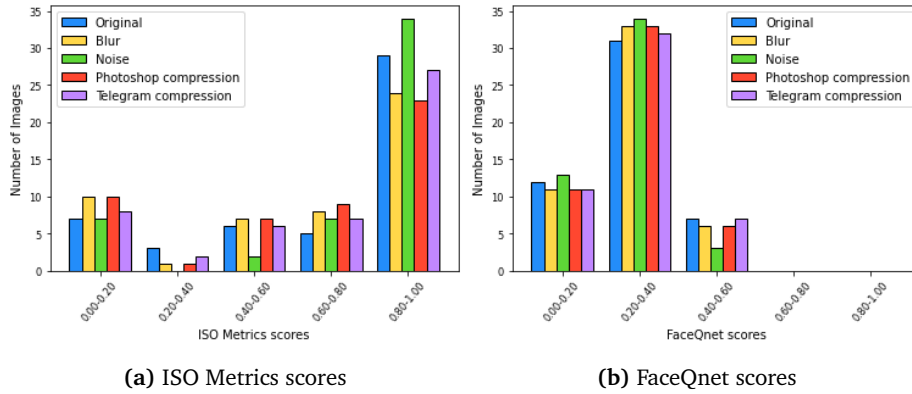


Figure 6.8: Histogram of the objective scores of the original images and their corresponding distorted images in the NFC dataset.

in Figure 6.10. The FIQM struggled with facial images with noise which is clear by the moderate correlation coefficients close to 0.5, relative to the other distortions which are considered strongly correlated. Not only did the correlation coefficients confirm that FaceQnet performed better on the distortions, but the FIQM had considerably lower RMSE values on $\frac{3}{4}$ distortions than ISO Metrics. Only by looking at the RMSE values mentioned above each plot depicted in Figure 6.9 and Figure 6.10, it is clear that ISO Metrics is less consistent than FaceQnet when assessing distorted images.

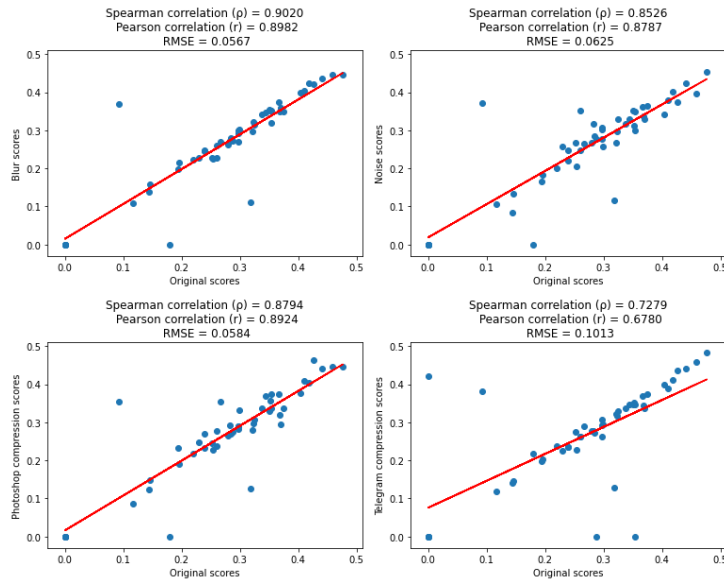


Figure 6.9: 2D scatter plots of FaceQnet scores on the original images along the x-axis and the distorted images along the y-axis. All correlation coefficients are given with a 95% confidence interval. RMSE values are included above the plots.

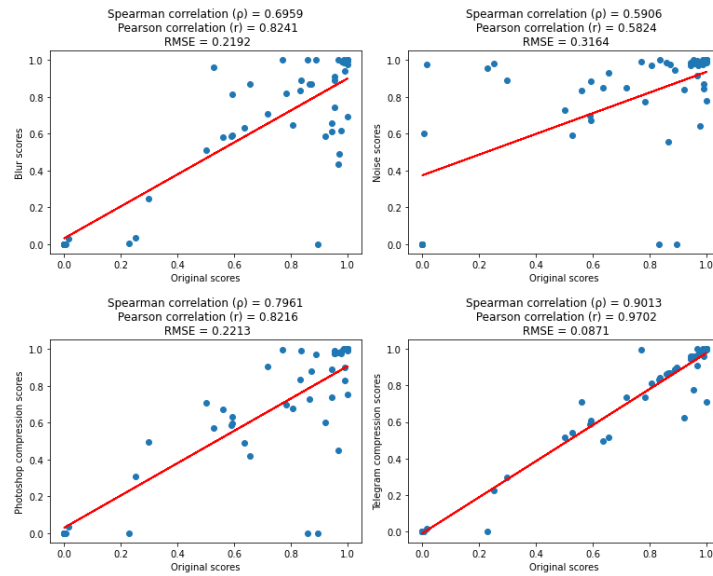


Figure 6.10: 2D scatter plots of ISO Metrics scores on the original images along the x-axis and the distorted images along the y-axis. All correlation coefficients are given with a 95% confidence interval. RMSE values are included above the plots.

6.2.3 Evaluating the Performance of FIQMs

The correlation between the FIQMs and the collected subjective scores was calculated and plotted in Figure 6.11. A 0.05 significance level was used when calculating the correlation coefficients. The coefficients ranged between 0.32 to 0.42 which meant there was a weak association between the objective and subjective scores. Using a weighted average of the two FIQMs did not show a significant difference. There were no clear difference between the performance of either FIQM on our dataset. Even though the correlation was weak, the result were not entirely negative. Our dataset was supposed to challenge the FIQMs by introducing new measures they had not been exposed to or created to assess, which was the case.

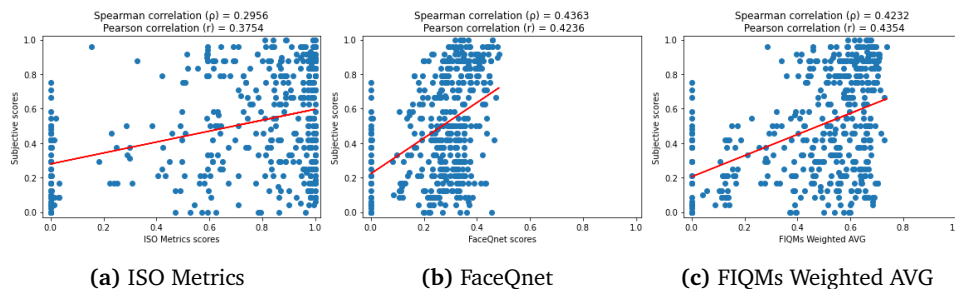
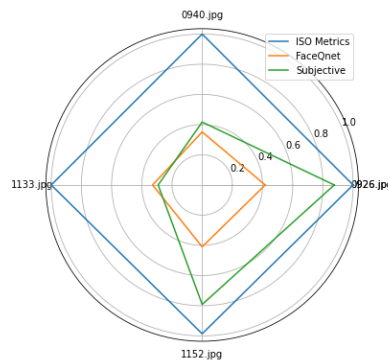


Figure 6.11: 2D scatter plots of the scores on the NFC dataset with objective scores along the x-axis and subjective scores along the y-axis. The Spearman and Pearson correlation coefficients are shown above the plots.

Score Distribution on Distorted Images

The spider chart in Figure 6.12 takes a closer look into how the FIQMs reacted to the different face mask usages. The subjective scores for 0940.jpg and 1133.jpg were lower than the two other face mask images because the face is more covered and therefore less visible. ISO Metrics did not react to that whatsoever, but FaceQnet did provide slightly lower scores, but the change was very minor. Regarding the spider plot of the oblique angled images depicted in Figure 6.13, both FIQMs provided lower scores than the subjective, but the ISO Metrics scores were consistent whereas FaceQnet had larger differences between the facial images.

The two spider plots depicted in Figure 6.14 and Figure 6.15 are used to show the score distribution of the previewed distorted facial images. On image 0329.jpg in Figure 6.14 we can see how ISO Metrics predicted equal scores, but the facial image with noise had a major effect on the score. This is consistent with the histogram in Figure 6.8a. Even the Photoshop compression received double the score of its original image. FaceQnet had an overall more consistent quality perception close to the subjective scores. Image 0718.jpg in Figure 6.15 again showed how FaceQnet was not affected by adding any of our distortions. The plot reinforces the claim that the distribution of objective scores on distorted images were negligible with FaceQnet, which is shown in Figure 6.8b. The FaceQnet scores were quality wise closely related to the subjective scores, whereas the quality perception by ISO Metrics was off. This time, the FIQM gave much more consistent scores with no large spikes, which goes to show its inconsistency.

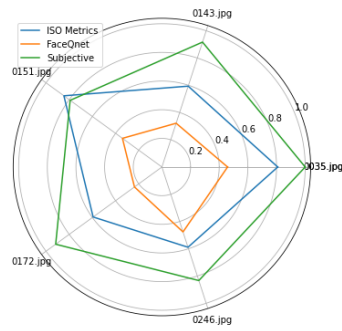


(a) Different use of face masks.

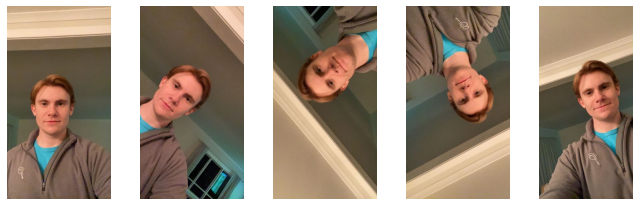


(b) 0926.jpg (c) 0940.jpg (d) 1133.jpg (e) 1152.jpg

Figure 6.12: Spider chart of the objective and subjective scores on different face mask images.

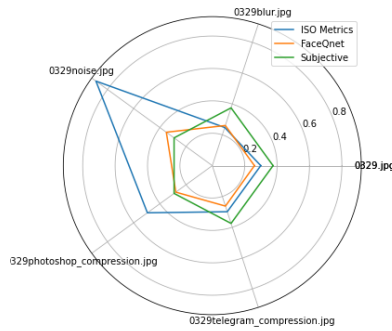


(a) Different oblique angles.

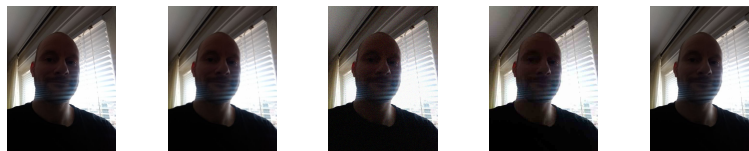


(b) 0035.jpg (c) 0143.jpg (d) 0151.jpg (e) 0151.jpg (f) 0246.jpg

Figure 6.13: Spider chart of the objective and subjective scores on images with different oblique angles.

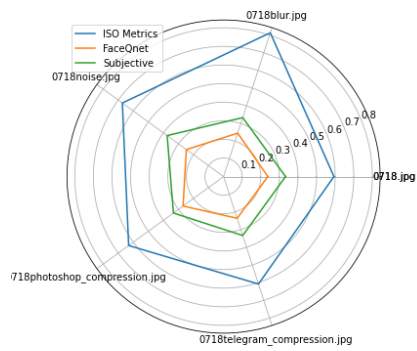


(a) 0329.jpg distorted.



(b) Original. (c) Blur. (d) Noise. (e) Photoshop compression. (f) Telegram compression.

Figure 6.14: Spider chart of the objective and subjective scores on original and distorted facial images.



(a) 0718.jpg distorted.

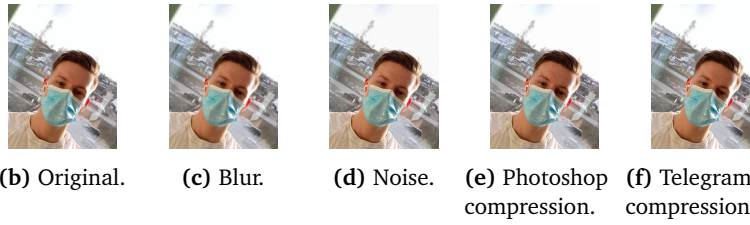


Figure 6.15: Spider chart of the objective and subjective scores on original and distorted facial images.

Chapter 7

Conclusion

Now that we have looked at the various aspects that went into the development of the web application and the subjective experiment, it is time to reflect on the journey our group has gone through the last months. In Section 7.1 we discuss how the application turned out and how the user test supported that the requirements were fulfilled. Further in Section 7.2, we reflect upon our subjective experiments. In Section 7.3 we summarize our results. To end the chapter, we evaluate the group work (Section 7.4), learning outcome (Section 7.5) and discuss future work (Section 7.6).

7.1 Objective Assessment Evaluation

To evaluate our work done in the objective assessment, we have looked at how well we managed to complete the task Mobai gave us. Based on the requirements stated in Section 4.1 and the use cases showcased in Figure 4.1, we can see that we have managed to build and complete all the main functionality and requirements that Mobai had given us in the objective assessment. Given the feedback from the user testing showcased in Table 4.6, all functionality worked and was set to “Satisfactory” by the Product Owner. From the user testing questions in Table 4.7, we got some feedback regarding doing some minor design changes to the frontend. Overall we have done what Mobai asked us and according to their feedback, they were satisfied.

7.2 Subjective Experiment Evaluation

As introduced in Section 5.2, there were several aspects to take into account when conducting a subjective experiment. We spent much time researching and analyzing different aspects and platforms when deciding the structure of the experiment. Collecting observers was more difficult with the ongoing COVID-19 restrictions. Ideally we wanted to invite observers to perform the subjective experiment in a controlled environment which was not possible. However, conducting a subject-

ive experiment on the web ended up being surprisingly less complicated than we thought.

7.3 Evaluation of the Results

When it comes to the two FIQMs, ISO Metrics and FaceQnet, we can conclude that they performed similar on the Combined passport alike dataset with a low to moderate correlation shy of 0.5 with the subjective scores, as shown in Figure 6.3 and Figure 6.4. On the Capture from photo dataset, ISO Metrics outperformed FaceQnet by having a low to moderate correlation, while FaceQnet's correlation coefficients were considered weak. The dataset in which the FIQMs performed crucially poor, were the Selfie dataset. Both FIQMs had a non-existent association with the subjective scores. With regards to FaceQnet, the FIQM did not work for that particular dataset, because the faces in the images were not detected and therefore the cropping phase became defective. ISO Metrics on the other hand, struggled with estimating the correct quality because the facial images of the Selfie dataset greatly differed in quality relative to the two other datasets. The facial images were either off-centered, too zoomed in or a combination of both, which ISO Metrics did not react accordingly to. The strongest association achieved was done by using a weighted average of the scores from ISO Metrics and FaceQnet on Combined passport alike. With Pearson and Spearman values around 0.6, we can conclude that using a weighted average approach of the FIQMs can be a reliable choice for datasets where the facial images checks several of the bullet points in what defines a good facial image listed in Section 3.1.

The performance of the FIQMs with regards to face masks were subpar. The perceived quality by ISO Metrics did not change with different face masks coverings, whereas FaceQnet had very minor differences. The two FIQMs can not be considered reliable when assessing face masks. The same can not be said for oblique angled facial images. The FIQMs showed minor differences when assessing those types of facial images.

Assessing the quality of distorted facial images did affect the perceived quality for ISO Metrics, but not for FaceQnet. The last mentioned FIQM perceived the quality consistently with close to perfect association with the original images shown in Figure 6.8b and Figure 6.9. The FIQM should be considered accurate when predicting the perceived quality on the distortions we added, which was not entirely the case for ISO Metrics. The FIQM had trouble assessing facial images with noise and was less accurate than FaceQnet when assessing blur and compression done in Photoshop shown in Figure 6.10.

7.4 Group Work Evaluation

Although only working from home and communicating on digital platforms, the collaboration between the group members has been great. We have mostly worked

together and all agreements and deadlines were met. The workflow throughout the project was consistent which ended up in great quality work. Toggl (Appendix H) was a great tool for following each members working tasks. The group members have helped each other with reviewing their work. The thesis writing started early and we are satisfied with the outcome.

7.5 Learning Outcome

By working closely together the past months, we have learned the importance of having a strict schedule and expectations among the members to ensure everyone keep their morale and the good work throughout the project. We experienced that not having clear deadlines resulted in incompleteness of working tasks. These types of lacks required more time spending on reviewing the code and report. On the programming side, the group has learned new programming languages, tools, libraries and how these interact. We may not have come up with the best approach, however the requirements were met and the web application is working. No one in the team had developed a web application before, resulting in a new experience for all members. Coding wise, it was manageable to create the backend in Python and a frontend UI in React. However, no one had experience in making them interact with each other, so the APIs could be developed in a more well-designed way. In terms of research, the members have developed new abilities throughout the project by retrieving information and using it in our work. This project has also improved our English and how to present our work in a more scientific way.

7.6 Future Work

We managed to include everything specified in Chapter 4. Since one of the requirements was to easy facilitate new FIQMs to be added into the web application, using the application in further work was a reality. Mobai has stated that this application will be a part of their face recognition system, where the web application works as a sieve to filter out low-quality images. Chapter 4 can be used as documentation to describe the different technologies used to develop the web application.

The two FIQMs utilized in this project, can be studied further. Whereas both provide quality scores to facial images, they act differently. To give a better evaluation of facial images, studying the weighted average between the two FIQMs can provide improved accuracy of the quality scores. Proposing a new weighted approach where the FIQMs are weighted differently may be the next step in achieving improved performance of the metrics.

Our dataset, described in Section 5.7, was outperforming the datasets provided by Mobai. It consisted of more facial images, distortions and new aspects. The new aspects consisting of camera angels and wearing face masks tested the FIQMs in a new way. Mobai is able to use the NFC dataset in their further work of testing

new FIQMs and face recognition systems.

Bibliography

- [1] T. Barnett, *Group decision making [internet]*, Avdameg Inc [cited 20.04.2021], Available from: <https://www.referenceforbusiness.com/management/Gr-Int/Group-Decision-Making.html>.
- [2] A. Simmons, *Why small is the key to big agile results [internet]*, assemble; 13.06.2016 [cited 24.04.2021], Available from: <https://assemblesystems.com/blog/why-small-is-the-key-to-big-agile-results/>.
- [3] I. Sommerville, *Software Engineering 10th edition*. Pearson, 2016, pp. 47–49.
- [4] A. Alliance, *Extreme programming [internet]*, Agile Alliance; 2016 [cited 21.01.2021], Available from: <https://www.agilealliance.org/glossary/xp/>.
- [5] M. Rehkopf, *Kanban vs. scrum: Which agile are you? [internet]*, Atlassian; 2021 [cited 27.04.2021], Available from: <https://www.atlassian.com/agile/kanban/kanban-vs-scrum>.
- [6] *What is scrum? [internet]*, digite [cited 27.04.2021], Available from: <https://www.digite.com/agile/scrums-methodology/>.
- [7] *Trello*, <https://trello.com/>.
- [8] T. S. P Group, *Definition of done [internet]*, Raleigh: The Scrum Patterns Group; 2019 [cited 25.01.2021], Available from: <http://scrumbook.org/>.
- [9] I. Sommerville, *Software Engineering 10th edition*. Pearson, 2016, pp. 648–650.
- [10] *Iso/iec tr 29794-5:2010 - information technology — biometric sample quality — part 5: Face image data*, Available from: <https://www.iso.org/standard/50912.html>, ISO [cited 27.04.2021], Apr. 2010.
- [11] *Machine readable travel documents [internet]*, Seventh edition, Available from: https://www.icao.int/publications/Documents/9303_p3_cons_en.pdf, ICAO [cited 27.04.2021], 2015.
- [12] T. Schlett, C. Rathgeb, O. Henniger, J. Galbally, J. Fierrez and C. Busch, *Face image quality assessment: A literature survey*, Cornell University [cited 27.04.2021], Available from: <https://arxiv.org/abs/2009.01103>, Sep. 2020.

- [13] *Random forest regressor [internet]*, scikit-learn [cited 27.04.2021], Available from: <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html/>.
- [14] *Faceqnet [internet]*, Github [cited 27.04.2021], Available from: <https://github.com/uam-biometrics/FaceQnet/>.
- [15] J. Hernandez-Ortega, J. Galbally, J. Fierrez, R. Haraksim and L. Beslay, *Faceqnet: Quality assessment for face recognition based on deep learning*, Available from: <https://arxiv.org/abs/1904.01740>, 2019.
- [16] J. Hernandez-Ortega, J. Galbally, J. Fierrez and L. Beslay, *Biometric quality: Review and application to face recognition with faceqnet*, Available from: <https://arxiv.org/abs/2006.03298>, 2021.
- [17] *Portrait quality (reference facial images for mrted) [internet]*, Version 1.0, Available from: <https://www.icao.int/Security/FAL/TRIP/Documents/TR%20-%20Portrait%20Quality%20v1.0.pdf>, ICAO [cited 27.04.2021], Apr. 2018.
- [18] Q. Cao, L. Shen, X. W., O. Parkhi and A. Zisserman, *Vggface2: A dataset for recognising faces across pose and age*, Cornell University [cited 27.04.2021], Available from: <https://arxiv.org/abs/1710.08092>, May 2018.
- [19] B. Gaille, *15 desktop vs web application pros and cons [internet]*, BRANDON GAILLE; 06.03.2019 [cited 29.04.2021], Available from: <https://brandongaille.com/15-desktop-vs-web-application-pros-and-cons/>.
- [20] *Top 10 backend frameworks*, Back4App [cited 21.04.2021], Available from: <https://blog.back4app.com/backend-frameworks/>.
- [21] S. Bhatt, *Pros and cons of django framework for app development*, DZone; 31.08.2020 [cited 22.04.2021], Available from: <https://dzone.com/articles/pros-and-cons-of-django-framework-for-app-developm>.
- [22] *Pros and cons of reactjs*, JavaTPoint; [cited 22.04.2021], Available from: <https://www.javatpoint.com/pros-and-cons-of-react>.
- [23] *The good and the bad of angular development*, altexsoft; 25.03.2020 [cited 06.05.2021], Available from: <https://www.altexsoft.com/blog/engineering/the-good-and-the-bad-of-angular-development/>.
- [24] *Understanding the pros and cons of angular development*, greyCampus; 30.05.2019 [cited 06.05.2021], Available from: <https://www.greycampus.com/blog/programming/good-and-bad-of-angular-development>.
- [25] *Getting started*, React; [cited 06.05.2021], Available from: <https://reactjs.org/docs/getting-started.html>.
- [26] *Node.js*, Wikipedia; 11.1.2019 [cited 06.05.2021], Available from: <https://no.wikipedia.org/wiki/Node.js>.
- [27] *Album layout*, Material-UI; [cited 12.05.2021], Available from: <https://material-ui.com/getting-started/templates/album/>.

- [28] E. Sarpotdar, *Why should you containerize applications?* [internet], CloudHedge; 09.08.2019 [cited 23.04.2021], Available from: <https://cloudhedge.io/why-should-you-containerize-applications/>.
- [29] M. Pedersen, 'Image quality metrics for the evaluation of printing workflows,' Ph.D. dissertation, University of Oslo, 2011.
- [30] K. Panetta, Q. Wan, S. Agaian, S. Rajeev, S. Kamath, R. Rajendran, S. Rao and et al., 'A comprehensive database for benchmarking imaging systems,' 2018, Available from: <http://tdface.ece.tufts.edu/>.
- [31] C. E. Thomaz and G. A. Giraldi, 'A new ranking method for principal components analysis and its application to face image analysis,' *Image and Vision Computing*, vol. 28, no. 6, pp. 902–913, 2010, Available from: <https://www.sciencedirect.com/science/article/abs/pii/S0262885609002613?via%3Dihub/>.
- [32] Z. Zhang, J. Yan, S. Liu, Z. Lei, D. Yi and S. Z. L. A, 'A face antispoofing database with diverse attacks,' 2012.
- [33] M. M. Kalayeh, M. Seifu, W. LaLanne and M. Shah, 'How to take a good selfie?' *Proceedings of ACM Multimedia Conference 2015 (ACMMM 2015)*, 2015, Available from: <https://www.crcv.ucf.edu/data/Selfie/>.
- [34] K. V. Ngo, J. J. Storvik, C. A. Dokkeberg, I. Farup and M. Pedersen, 'Quick-eval: A web application for psychometric scaling experiments,' *Image Quality and System Performance XII*, vol. 9396, pp. 9396–24, 2015.
- [35] *Colour blindness* [internet], Colour Blind Awareness; [cited 12.05.21], Available from: <https://www.colourblindawareness.org/colour-blindness/>.
- [36] D. S. Ishihara, *The Series of Plates Designed as a Test for Colour-Blindness*, 24 Plates Edition. Kanehara Shuppan CO., LTD, 1972.
- [37] *Cambridge colour test* [internet], Cambridge Research Systems [cited 16.05.21], Available from: <https://www.crsLtd.com/tools-for-vision-science/measuring-visual-functions/cambridge-colour-test/>.
- [38] *Recommendation itu-r bt.500-13: Methodology for the subjective assessment of the quality of television pictures* [internet], Thirteenth edition, Available from: https://www.itu.int/dms_pubrec/itu-r/rec/bt/R-REC-BT.500-13-201201-I!!PDF-E.pdf, International Telecommunication Union/ITU Radiocommunication Sector [cited 27.04.2021], 2012.
- [39] W. contributors, *Spearman's rank correlation coefficient* [internet], Wikipedia; [cited 28.04.2021], Available from: https://en.wikipedia.org/w/index.php?title=Spearman%27s_rank_correlation_coefficient&oldid=1019524247.

Appendix A

Project Assignment

Oppdragsgiver:

Oppdragsgiver og kontaktinformasjon – Mobai AS:

- Kjartan Mikkelsen, +47 489 90 013, km@mobai.bio
- Brage Strand, +47 40490411, brage@mobai.bio
- Besøksadresse: Raufossvegen 40, 2821 Gjøvik.
- Postadresse: Postboks 104, 2801 Gjøvik

Bakgrunn: Bildekvalitet på ansiktsbilder

Ytelsen på ansiktsgjenkjenningssystemer er avhengig av kvaliteten på bildene som brukes til trening og analyse. Kvalitet er et begrep som er avhengig av flere faktorer, men det finnes algoritmer som automatisk kan beregne kvalitet på en slik måte at det kan brukes til å filtrere bilder for bruk i biometriske systemer. (Referanse: <https://arxiv.org/pdf/2009.01103.pdf>)

Oppgave: Face quality metrics application

Mobai er en start-up som arbeider med teknologi utviklet ved Norsk Biometrilab på Gjøvik. Mobai leverer løsningen for ansiktsgjenkjenning, angrepsdeteksjon mot biometriske systemer (presentation attack deteksjon, som f.eks masker eller bilde-angrep), og deteksjon av Face Morphs (en facemorph er et bilde som er en sammensmelting av flere ansikt, og de lages for å lure ansiktsgjenkjenningssystemer).

Et viktig verktøy for Mobai er kunstig intelligens og maskinlæring, hvor vi lager modeller for gjenkjenning av biometriske attributter (f.eks ansikt) og oppdager angrep mot slike løsninger. Gode modeller er avhengig av hensiktsmessige datasett for trening av modellene.

For å trene gode modeller, gi en rask vurdering av en kundes datasett, eller bygge egne og bra datasett er det viktig å vite kvaliteten på dataene. Dette er en oppgave som ønskes automatisert.

Oppgaven går ut på å byggen en applikasjon som bruker to algoritmer som Mobai har for å gjøre kvalitetsvurdering av bilder. Applikasjonen skal lese et bildedatasett fra en database, kjøre dem gjennom algoritmene og presentere resultatet som en rapport over datasettet. Det skal i tillegg være mulig å gjøre menneskelige vurderinger av ansiktsbilder i databasen for og korrelere med kvalitetsmetrikken gitt av algoritmene. Det skal også være mulig å vurdere statistikken og resultatet fra algoritmene ved å måle ansiktsgjenkjenningssystemer ved bruk av Mobai algoritmer for ansiktsgjenkjenning.

Veiledning fra forskere og utviklere med lang erfaring fra SW bransjen.

Appendix B

Project Agreements

Prosjektavtale

mellom NTNU Fakultet for informasjonsteknologi og elektroteknikk (IE) på Gjøvik (utdanningsinstitusjon), og Mobai AS (oppdragsgiver), og Walid Demloj, Kjetil Grosberghaugen, Julian Nyland Skattum og Hans Petter Fauchald Taralrud (student(er))

Avtalen angir avtalepartenes plikter vedrørende gjennomføring av prosjektet og rettigheter til anvendelse av de resultater som prosjektet frembringer:

1. Studenten(e) skal gjennomføre prosjektet i perioden fra 11.01.2021 til 08.06.2021 .

Studentene skal i denne perioden følge en oppsatt fremdriftsplan der NTNU IE på Gjøvik yter veiledning. Oppdragsgiver yter avtalt prosjektbistand til fastsatte tider. Oppdragsgiver stiller til rådighet kunnskap og materiale som er nødvendig for å få gjennomført prosjektet. Det forutsettes at de gitte problemstillinger det arbeides med er aktuelle og på et nivå tilpasset studentenes faglige kunnskaper. Oppdragsgiver plikter på forespørsel fra NTNU å gi en vurdering av prosjektet vederlagsfritt.

2. Kostnadene ved gjennomføringen av prosjektet dekkes på følgende måte:
 - Oppdragsgiver dekker selv gjennomføring av prosjektet når det gjelder f.eks. materiell, telefon, reiser og nødvendig overnatting på steder langt fra NTNU i Gjøvik. Studentene dekker utgifter for ferdigstillelse av prosjektmateriell.
 - Eiendomsretten til eventuell prototyp tilfaller den som har betalt komponenter og materiell mv. som er brukt til prototypen. Dersom det er nødvendig med større og/eller spesielle investeringer for å få gjennomført prosjektet, må det gjøres en egen avtale mellom partene om eventuell kostnadsfordeling og eiendomsrett.
3. NTNU IE på Gjøvik står ikke som garantist for at det oppdragsgiver har bestilt fungerer etter hensikten, ei heller at prosjektet blir fullført. Prosjektet må anses som en eksamensrelatert oppgave som blir bedømt av intern og ekstern sensor. Likevel er det en forpliktelse for utøverne av prosjektet å fullføre dette til avtalte spesifikasjoner, funksjonsnivå og tider.
4. Alle beståtte bacheloroppgaver som ikke er klausulert og hvor forfatteren(e) har gitt sitt samtykke til publisering, kan gjøres tilgjengelig via NTNUs institusjonelle arkiv NTNU Open.

Tilgjengeliggjøring i det åpne arkivet forutsetter avtale om delvis overdragelse av opphavsrett, se «avtale om publisering» (jfr Lov om opphavsrett). Oppdragsgiver og veileder godtar slik offentliggjøring når de signerer denne prosjektavtalen, og må evt. gi skriftlig melding til studenter og instituttleder/fagenhetsleder om de i løpet av prosjektet endrer syn på slik offentliggjøring.

Den totale besvarelsen med tegninger, modeller og apparatur så vel som programlisting, kildekode mv. som inngår som del av eller vedlegg til besvarelsen, kan vederlagsfritt benyttes til undervisnings- og forskningsformål. Besvarelsen, eller vedlegg til den, må ikke nyttes av NTNU til andre formål, og ikke overlates til utenforstående uten etter avtale med de øvrige parter i denne avtalen. Dette gjelder også firmaer hvor ansatte ved NTNU og/eller studenter har interesser.

5. Besvarelsens spesifikasjoner og resultat kan anvendes i oppdragsgivers egen virksomhet. Gjør studenten(e) i sin besvarelse, eller under arbeidet med den, en patentbar oppfinnelse, gjelder i forholdet mellom oppdragsgiver og student(er) bestemmelsene i Lov om retten til oppfinnelser av 17. april 1970, §§ 4-10.
6. Ut over den offentliggjøring som er nevnt i punkt 4 har studenten(e) ikke rett til å publisere sin besvarelse, det være seg helt eller delvis eller som del i annet arbeide, uten samtykke fra oppdragsgiver. Tilsvarende samtykke må foreligge i forholdet mellom student(er) og faglærer/veileder for det materialet som faglærer/veileder stiller til disposisjon.
7. Studenten(e) leverer oppgavebesvarelsen med vedlegg (pdf) i NTNUs elektroniske eksamenssystem. I tillegg leveres ett eksemplar til oppdragsgiver.
8. Denne avtalen utferdiges med ett eksemplar til hver av partene. På vegne av NTNU, IE er det instituttleder/faggruppetleder som godkjenner avtalen.
9. I det enkelte tilfelle kan det inngås egen avtale mellom oppdragsgiver, student(er) og NTNU som regulerer nærmere forhold vedrørende bl.a. eiendomsrett, videre bruk, konfidensialitet, kostnadsdekning og økonomisk utnyttelse av resultatene. Dersom oppdragsgiver og student(er) ønsker en videre eller ny avtale med oppdragsgiver, skjer dette uten NTNU som partner.
10. Når NTNU også opptreer som oppdragsgiver, trer NTNU inn i kontrakten både som utdanningsinstitusjon og som oppdragsgiver.
11. Eventuell uenighet vedrørende forståelse av denne avtale løses ved forhandlinger avtalepartene imellom. Dersom det ikke oppnås enighet, er partene enige om at tvisten løses av voldgift, etter bestemmelsene i tvistemålsloven av 13.8.1915 nr. 6, kapittel 32.
12. Deltakende personer ved prosjektgjennomføringen:

NTNUs veileder (navn): Seyed Ali Amirshahi

Oppdragsgivers kontaktperson (navn): Kjartan Mikkelsen

Student(er) (signatur): Wahid Demloj dato 26.01.21

Kjehil Grøsbekhammen dato 26.01.21

Felton Nyland S Bakken dato 26.01.21

Hans Petter Fauchald Tarabrad dato 26.01.21

Oppdragsgiver (signatur): Brage Strand dato 27.01.21

Signert avtale leveres digitalt i Blackboard, rom for bacheloroppgaven.

Godkjennes digitalt av instituttleder/faggruppeleder.

Om papirversjon med signatur er ønskelig, må papirversjon leveres til instituttet i tillegg.

Plass for evt sign:

Instituttleder/faggruppeleder (signatur): _____ dato _____

Bachelor Agreement

Mobai, hereafter addressed as The Company, is an Norwegian SME which is a Spin-off from The Norwegian Biometrics Lab at The Norwegian University of Science and Technology. Mobai work with multiple biometric technologies to provide secure identity verification and authentication, as well as to prevent and detect fraud scenarios.

Walid Demloj, Julian Nyland Skattum, Kjetil Grosberghaugen and Hans Petter Fauchald Taralrud, here after addresses as The Students, are students attending NTNU Gjøvik and are currently working on their bachelor thesis.

This document is the agreement between The Company and The Students for regarding the joint collaboration on a bachelor thesis projects.

Bachelor topics and goals

The Students will work on a bachelor thesis topic provided by The Company. The project goals, scope and timeline is described in the project plan. Important goals for the Company is to: (1) Produce workable applications that can be used in ensuring face quality, and (2) Secure the legal framework for The Company to patent any potential innovations.

Legal obligations

The Company will provide topics and challenges for The Students in project, as well as internal resources in the Company that The Company can spare to assist and help in conducting the thesis activities. The Student is responsible for planning and producing the bachelor report and related applications on its own. The Bachelor project is not an employment and The Company is not paying The Students for any of their work on the thesis.

The Students shall keep all information relating to the Company confidential, such as for instance business strategy and technical source code and road maps. The confidentiality obligation remains in full force and effect after the bachelor project.

So long the Students is conducting the bachelor project, they shall not engage in any activity that, directly or indirectly, competes with the business and affairs of the Company.

All results produced by the Students is the sole property of the Company, and if the Students discovers, develops, invents or otherwise becomes aware of intellectual property rights related to the Company, the Company's business, plans and objectives, the Students confirm that all rights to such intellectual property rights shall be exclusively held, owned and further developed by the Company. This include that all materials and intellectual property rights (source code, presentations, data sets, etc) produced by the Students is owned by the Company.

The Students is obligated to coordinate with the Company regarding potential publications of the Bachelor Thesis Report. The Students is obligated to not make the bachelor thesis content available to the public or publish reports, presentation or source code without the consent of The Company. The reason for this is that The Company is considering patenting innovations that this bachelor project will work on.

The Company is willing to be a reference use case for The Students.



Brage Strand, CEO, Mobai AS

Gjøvik, 27.01.2020

Walid Demloj
Julian Nyland Skattum
Kjetil Grosberghaugen
Hans Petter Fauchald Taralrud

09.02.2021

Walid Demloj

Julian Nyland Skattum

Kjetil Grosberghaugen

Hans Petter Fauchald Taralrud

Appendix C

Project Plan

Project plan
Face quality metrics application

Walid Demloj
Kjetil Grosberghaugen
Julian Nyland Skattum
Hans Petter Fauchald Taralrud

January 2021

Contents

1	Goals and Frameworks	3
1.1	Background	3
1.2	Project goals	3
1.2.1	Main goal	3
1.2.2	Long term goals	3
1.3	Framework	4
2	Scope	4
2.1	Subject Area	4
2.2	Delimitation	4
2.3	Task Description	4
3	Project Organization	5
3.1	Responsibilities and roles	5
3.2	Routines and rules in the group	6
4	Planning, Follow-Up and Reporting	6
4.1	Main division of the project	6
4.1.1	Choice of software development method	6
4.1.2	Scrum layout	7
4.2	Plan for status meetings and decision points	8
5	Organization of Quality Assurance	9
5.1	Documentation, standard usage and source code	9
5.2	Configuration Management	9
5.2.1	Development workflow	9
5.3	Tools	10
5.4	Risk analysis	10
5.4.1	Identifying the risks	10
5.4.2	Risk analysis	11
5.4.3	Risk management and countermeasures	12
6	Plan of Execution	13
6.1	Gantt diagram	13
6.2	Milestones and decision points	15

1 Goals and Frameworks

1.1 Background

Mobai is a technology firm and system that works primarily with facial recognition. They are located at Norsk Biometrilab in Gjøvik and delivers Systems for facial recognition, attack detection against biometrical systems and face morph detection.

An important tool for Mobai is artificial intelligence and machine learning, which they use to create models for detection of biometric attributes like faces, and detect attacks against these solutions. Good models for these solutions are dependent on relevant datasets and training of the models. In order to train good models, it is important to have good quality on the data. This is a process that Mobai would like to have automatized.

Currently Mobai is using different Image Quality Metrics (IQMs) to determine the quality of the images in a dataset. In order to make this process go smoother, Mobai now wishes to create an application that will take a dataset, and display the quality of the images as a report using two IQMs Mobai already has to determine the quality. Mobai wants to include human assessments to evaluate the accuracy of the IQMs. This will make it easier for Mobai to determine whether a specific dataset is usable in training good models, or give a quick assessment of a customers dataset.

1.2 Project goals

1.2.1 Main goal

The main goal is separated into three essential parts:

- Provide Mobai a working web application that showcases face image quality scores for images in a dataset, using two IQMs given by the company.
- Create and conduct a subjective experiment to obtain data from human appraisals.
- Evaluate the IQMs accuracy by correlating the subjective and objective face image quality scores.

1.2.2 Long term goals

Building reliable models are an essential part of artificial intelligence and machine learning. Therefore knowing the quality of images plays an important part. An application would automate the process of training models, building satisfying and clean datasets, and evaluating customers datasets.

The main goals Mobai wishes to fulfill in the long term are:

- Speed up the whole process of evaluating customers datasets
- Achieve better models for training

- Create new datasets for further research

1.3 Framework

- The application will be packed and delivered as a container solution or as a set of containers.
- The client side of the application will have a graphical user interface.

2 Scope

2.1 Subject Area

For Mobai, artificial intelligence and machine learning are essential subjects for their work. For this report and creation of the application, it is important to understand what is a good image, and how different IQMs work in separate ways to generate a quality score.

Our task will cover different technologies and subjects within programming and image quality assessment, these include:

- Different face and image quality assessment papers, like [1] and [2]
- Face image quality metrics in python
- Docker desktop and containers for ease of deployment
- Subjective assessment survey to collect data
- Unit testing
- Front-end web-developed user interface using HTML and JavaScript
- Back-end development using mainly python

2.2 Delimitation

We as developers are not responsible for creation of the IQMs or relevant data sets. All IQMs for face image quality and relevant data sets are given to us from Mobai.

2.3 Task Description

The task is separated into two parts, one objective and one subjective Face Image Quality evaluation. The objective part consists of creating an application that uses two IQMs from Mobai that generates a quality score on each individual image within a dataset. The application will create a report on the whole dataset and output relevant data about the quality of the dataset, including face image quality score on the images.

The subjective assessment consists of having subjects evaluate facial images from a dataset based on what they think is good or bad quality. They will be shown different images of varying quality and asked to rate the quality of the images ranging from very low to very high. This way, we get multiple quality scores on the images in the dataset. The subjective results will then be used to compare with the objective results and presented in the report. The higher the correlation, the more accurate the IQM will be.

3 Project Organization

3.1 Responsibilities and roles

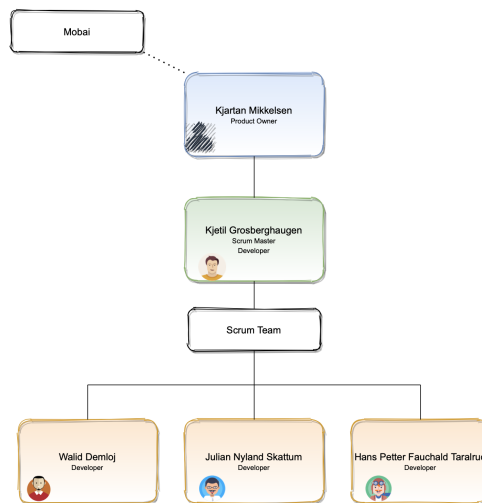


Figure 1: Organization chart

By using a Scrum development model to structure the project, the group members have different responsibilities according to their roles within this agile methodology.

Kjartan Mikkelsen is the Product Owner and our main contact from Mobai. He will participate in all sprint planning meetings/sprint review meetings. Mikkelsen is accompanied by two scientist that work for Mobai. They are specialized in our topic and will join our scheduled meetings when intricate questions occur.

Kjetil Grosberghaugen is the Scrum Master. Beside being a developer, his main tasks are to ensure that the development process flows evenly, arrange necessary meetings as well as acting like a connector between the developers and the Product Owner.

Walid Demloj, Julian Nyland Skattum and Hans Petter Fauchald Taralrud are the project developers. Throughout the project lifetime they will be working

on different tasks. However, their main tasks are to perform assigned work and to ensure that they have something to do.

3.2 Routines and rules in the group

Some essential points from our group rules scheme:

- It is expected that each group member works daily with this project, and the workload should be about 30-40 hours per week.
- All group members are expected to be available to work together as a group for at least 4 hours every working day, unless an acceptable reason has been given in advance.
- All group members must be active in the project. If there are things you don't understand, it's better to ask the other members for help than do nothing at all.
- All working hours shall be logged for each group member and be presented at status meetings.
- Status meetings will be held once a week.

4 Planning, Follow-Up and Reporting

4.1 Main division of the project

The project will consist of a web application with both an objective assessment and a subjective assessment for face image quality.

4.1.1 Choice of software development method

This project is characterised by uncertainty, a limited amount of labor, several meetings, a survey that may differ in complexity and a development phase with requirements that can change. These characteristics paved the way for us to pick the agile software development method Scrum, as was suggested by Mobai.

Our group consists of four inexperienced students with little to none experience with comprehensive projects of this size. Our inexperience alone causes some uncertainty in regards to deadlines, survey work and development. Having regular meetings with both Mobai and our project supervisor as well as receiving regular feedback, will decrease the chance to drift of track. This also ensures that our final product is as close to Mobai's vision as possible. The agile software development methods involves the client to a far more degree than the plan-driven approaches, which was an important point to take into consideration.

Since our team is rather small in size with no experts, the agile development methods seemed natural. Should we have decided to use one of the plan-driven methods, like the Waterfall-method, the chance to not succeed seemed greater.

A key principle in the Waterfall-method is its strict way of dividing a project into phases. The model emphasises finishing one phase before the next one starts. This would not be suitable for our project because of the very reason that it's time consuming. Should the development phase run into trouble that was not taken into consideration in the planning/design phase, the design phase would have to start over. It's likely that problems will occur in our project and having the flexibility provided by Scrum assures that progress is achieved as much as possible.

Should we finish Mobai's desired application quickly it was suggested by Mobai that additional functionality could be added to the solution if our time schedule allowed for it. Changes in the requirements were something Mobai were open to discuss and Scrum handles this well, which was another reason for our choice.

Within the agile development methods both Kanban and eXtreme Programming (XP) were taken into consideration. Both of these are reliable methods that provide solid structuring and flexibility [3]. However Kanban usually does not incorporate the element of predefined roles and therefore it suits our project to a lesser degree. The chapter "Responsibilities and roles" (3.1) describes what each group member is in charge of. Although all of us are expected to participate in all the tasks, some delegation of responsibility were decided to achieve greater structure of the tasks. In addition to this the end dates for all the tasks are rather tough to determine, which is why a sprint with several tasks will be more reliable.

With that said, Kanban is excellent for getting an overview of the tasks needed to be done, which is why we also decided to incorporate this method to a lesser degree. *Trello*[4] is used for tracking the tasks and the tasks are divided into three phases: *To-Do*, *Doing* and *Done*. Since Kanban is not our main method, no specific rules are set considering the amount of tasks in each phase.

XP was briefly considered, but we quickly realised that it was difficult to follow all 12 practices all the time. However we decided to implement the *Pair Programming* practice with a little tweak: Due to the uncertainty that the pandemic causes, the pair programming will be done by screen-sharing on laptop etc. Pair programming will improve the code quality which is the reason behind our choice [5].

4.1.2 Scrum layout

We will have sprints with a two-week duration. One-week sprints are rather short and leads to excessive meetings which affects our time management. On the other hand longer sprints tends to involve the client to a lesser degree, which is not ideal in our case. We are dependant on input from Mobai to satisfy their requirements.

The start of the sprints takes place on Tuesdays, starting 2. February. These are expected to finish before 11:00 every other Monday. The weekly scheduled meetings with our bachelor supervisor takes place every Monday at 12:00, which

gives us at least one hour to prepare both the next sprint and what we should discuss during the meeting. The following Tuesday a meeting with Mobai will be held, where we will go through what the group achieved during the sprint.

We use the Scrum-pattern *Definition of done* to collectively define what development-tasks are considered done [6]. In partnership with Mobai we will come to an agreement upon different criteria that form the definition of done. It is important to set different criteria to different work tasks (it should be a difference between coding-criteria and report-criteria). This pattern gives us a familiar understanding of work quality and absoluteness. We also obtain good habits in our workflow using the definition of done as a checklist to correlate with the user stories. In that way we prevent possible delays occurring in the development process. Here is an example of our report-criteria:

1. The section is completed according to the member
2. The member has analyzed the section's contents
3. The section is checked for typos
4. The whole group has read and approved the section

4.2 Plan for status meetings and decision points

The group will have weekly status meetings on Mondays at 11:00. On these meetings we will discuss and evaluate our current progress regarding the project, and discuss other subjects of importance.

We will have regular meetings with our NTNU project supervisor every Monday at 12:00. If we think an extraordinary meeting is needed we can contact our supervisor and try to arrange an extra meeting that week, preferably on Thursday or early on Friday.

During the initial phase of the project we will have regular status meetings with Mobai every Tuesday at 12:00. The frequency of these meetings can be adjusted as needed during the project.

5 Organization of Quality Assurance

5.1 Documentation, standard usage and source code

Documents and storage

- The bachelor-thesis is written in \LaTeX using Overleaf online compiler.
- \LaTeX source code is synchronized between Overleaf and our Github-repository to maintain backup protection.
- Summaries from meetings with Mobai and our supervisor are stored in a shared folder in Google disk.

Testing and code quality For coding in python, we will be using the pep-8 standard for project structure and style. Pylint[7] will be used to confirm that the standard is used correctly. We will use unit tests for testing the code, and SonarQube[8] for a whole analysis of the code quality.

5.2 Configuration Management

When multiple developers are working on this project at the same time, it is important to have a management system. This is to ensure that all developers have the latest version of the program, and also that the developers don't interfere with each others work. The commits to the project will have to be coordinated, to ensure that no developer is doing something that already has been done.

We will choose to use Github as our configuration management system. Commits will be done evenly to ensure that all developers always have access to the latest version of the program, with descriptive commit messages that shortly explains what has been done. The commit history can be used if we need to rollback to a previous version of the code. If two developers are working on the same component, a merge conflict will occur and have to be fixed before it gets pushed out to Github. This will keep the developers from overwriting each others code. Bugs, features and improvements can be tracked with the Github issue tracker.

5.2.1 Development workflow

Issues tracking For tracking issues during coding we will use Github's integrated Issues tracker. Discovered bugs, new features, suggested improvements and other potential issues will be added to Issues.

Smart commits We will use smart commits during the development to track which issues the commits are connected to. Git commits and merge requests will contain a keyword for what the commit is doing with the issue (e.g. fix, close, implement), the issue number the developer is trying to solve and a clear commit description.

5.3 Tools

Table 1: Table of tools

Name	Description	Application
draw.io	Online flowchart and diagram software	Creating charts and diagrams
Docker	Software containerization platform	Containers for final product
GanttProject	Project management software	Creating Gantt charts
Git	Distributed version-control system for tracking software changes	Version control
GitHub	Internet hosting for software development and version control	Version control
GitLab	Internet hosting for software development and version control	Version control
Google Drive	File storage and synchronization service	File storage and synchronization
Overleaf	Cloud-based editor used for writing and editing LaTeX documents	Project report
Planning Poker	Online sprint planning game	Sprint planning
Pylint	Source-code, bug and quality checker for Python	Code quality inspection
SonarQube	Platform for code quality inspection	Code quality inspection
Toggl	Application and online tool for time tracking	Time tracking
Trello	Online KanBan board for task tracking	Project management
Visual Studio Code	Development IDE	Application development

5.4 Risk analysis

5.4.1 Identifying the risks

A project with this range of scope is clearly exposed for several risks. It is important to identify these hazards, and to initiate countermeasures if needed. The list below shows different risks that possibly could occur during the project. The risks are numbered independently of their severity.

1. Group members leaving the project
2. Deadlines are not met
3. Bachelor thesis is not delivered in time
4. Mobai cancels the project
5. A similar project launches
6. The software does not fulfill the requirements
7. Inadequate planning and execution of subjective experiment
8. Loss of documents and source code
9. Sickness among group members, project supervisor or Mobai
10. Application breaking bugs

5.4.2 Risk analysis

The figure (2) analyzes the different risks that could happen during the project. These risks are split into two different factors: the probability for an action to occur and the consequences of that action. The colors indicates levels of risks where green illustrates low risk, yellow illustrates moderate risk and red illustrates high risk. Each number in the table are adopted from the risk identification described in the previous section.

Consequence / Probability	Little consequence	Moderate consequence	Serious consequence	Very serious consequence
Unlikely	5		1, 6	3, 4, 8
Likely			2	10
Very likely			7, 9	
Vastly likely				

Figure 2: Overview of the consequences and probability

5.4.3 Risk management and countermeasures

There are discussed measures to avoid risk events to occur. The countermeasures lower the negativity of the consequences or could reduce the probability of an action to take place at all.

Event	Severity	Countermeasure
Group members leaving the project	1	A structured group with good planning prevents members leaving the project. If a member is forced to leave by lack of participation, a new project orchestration will be necessary.
Deadlines are not met	2	Having a consistent workflow and a regular meeting schedule makes it easier to meet the deadlines. If there are too many tasks in the upcoming sprint, a rescheduling may occur and Product Owner will be informed.
The software does not fulfill the requirements	6	Performing research at an early stage in the development process clarifies what is possible to develop technologically. Being able to communicate with the Product Owner during the software development process enables regular requirements updates.
Inadequate planning and execution of subjective experiment	7	Creating the survey at an early development stage and ensuring enough participants to perform the survey.
Loss of documents and source code	8	Using backup-technologies when writing the report and coding the software prevents loss of important content.
Sickness among group members or Mobai	9	Following the Covid-19 restrictions and arranging online-meetings baffle sickness to the greatest extent.
Application breaking bugs	10	At an early stage in the coding process analyze what bugs that can result being application breaking. Applying regular testing to our workflow enable the bugs to be fixed quickly.

Figure 3: Risk management

6 Plan of Execution

6.1 Gantt diagram

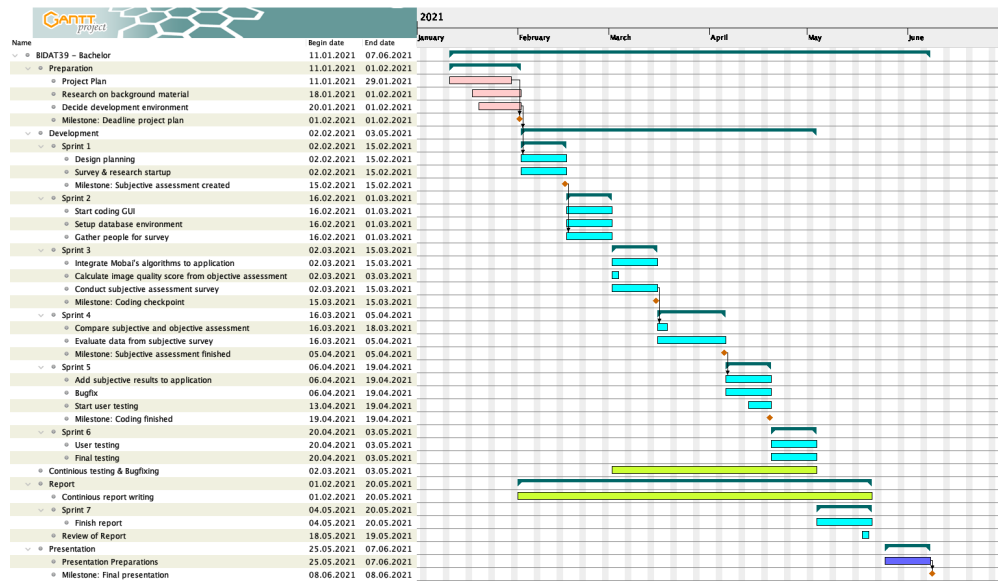


Figure 4: Gantt diagram

BIDAT39 - Bachelor	11.01.2021	07.06.2021
Preparation	11.01.2021	01.02.2021
Project Plan	11.01.2021	29.01.2021
Research on background material	18.01.2021	01.02.2021
Decide development environment	20.01.2021	01.02.2021
Milestone: Deadline project plan	01.02.2021	01.02.2021
Development	02.02.2021	03.05.2021
Sprint 1	02.02.2021	15.02.2021
Design planning	02.02.2021	15.02.2021
Survey & research startup	02.02.2021	15.02.2021
Milestone: Subjective assessment created	15.02.2021	15.02.2021
Sprint 2	16.02.2021	01.03.2021
Start coding GUI	16.02.2021	01.03.2021
Setup database environment	16.02.2021	01.03.2021
Gather people for survey	16.02.2021	01.03.2021
Sprint 3	02.03.2021	15.03.2021
Integrate Mobai's algorithms to application	02.03.2021	15.03.2021
Calculate image quality score from objective assessment	02.03.2021	03.03.2021
Conduct subjective assessment survey	02.03.2021	15.03.2021
Milestone: Coding checkpoint	15.03.2021	15.03.2021
Sprint 4	16.03.2021	05.04.2021
Compare subjective and objective assessment	16.03.2021	18.03.2021
Evaluate data from subjective survey	16.03.2021	05.04.2021
Milestone: Subjective assessment finished	05.04.2021	05.04.2021
Sprint 5	06.04.2021	19.04.2021
Add subjective results to application	06.04.2021	19.04.2021
Bugfix	06.04.2021	19.04.2021
Start user testing	13.04.2021	19.04.2021
Milestone: Coding finished	19.04.2021	19.04.2021
Sprint 6	20.04.2021	03.05.2021
User testing	20.04.2021	03.05.2021
Final testing	20.04.2021	03.05.2021
Continious testing & Bugfixing	02.03.2021	03.05.2021
Report	01.02.2021	20.05.2021
Continious report writing	01.02.2021	20.05.2021
Sprint 7	04.05.2021	20.05.2021
Finish report	04.05.2021	20.05.2021
Review of Report	18.05.2021	19.05.2021
Presentation	25.05.2021	07.06.2021
Presentation Preparations	25.05.2021	07.06.2021
Milestone: Final presentation	08.06.2021	08.06.2021

Figure 5: Tasks

6.2 Milestones and decision points

In order to keep up with the time schedule, we have made some general milestones for when different activities should be finished. This is to have a greater assurance that we are in route with the project.

- **Milestone 1, 1.February:** Deadline project plan, start coding
- **Milestone 2, 15.February:** Subjective experiment created.
- **Milestone 3, 15.March:** Coding checkpoint.
 - Database should be working.
 - Data sets should be read from database and run through IQMs. Initialized docker desktop.
- **Milestone 4, 30.March:** Subjective experiment finished.
 - Gathered data from the subjective experiment and ready to implement it into the objective results and solution.
- **Milestone 5, 19.April:** Coding finished.

References

- [1] Torsten Shlett, Christian Rathgeb, Olaf Henniger, Javier Galbally, Julian Fierrez, and Christoph Busch. Face image quality assessment: A literature survey. 22(1):1–17, 2020.
- [2] Javier Hernandez-Ortega, Javier Galbally, Julian Fierrez, Rudolf Haraksim, and Laurent Beslay. Faceqnet: Quality assessment for face recognition based on deep learning. 22(2):1–9, 2020.
- [3] Rachaelle Lynn. *Kanban vs. Scrum: What are the Differences?* [Internet]. Planview; 2021 [hentet 21.01.2021]. Available from: <https://www.planview.com/no/resources/guide/introduction-to-kanban/kanban-vs-scrum/>.
- [4] Trello. <https://trello.com/>.
- [5] Agile Alliance. *Extreme Programming [Internet]*. Agile Alliance; 2016 [hentet 21.01.2021]. Available from: <https://www.agilealliance.org/glossary/xp/>.
- [6] The Scrum Patterns Group. *Definition of done [Internet]*. Raleigh: The Scrum Patterns Group; 2019 [hentet 25.01.2021]. Available from: <http://scrumbook.org/>.
- [7] Pylint. <https://www.pylint.org/>.
- [8] Sonarqube. <https://www.sonarqube.org/>.

Appendix D

Survey Instructions

Face Image Quality Survey

- Instruction manual

Description

In this survey you are going to evaluate the quality of images of faces based on several factors.

The survey consists of ranking face images from a scale of Poor (1) to Excellent (5).

The following slides are prepared to better introduce different issues that affect the quality of a face image.

The criteria used to assess face images are based on ISO/IEC 29794-5. Familiarity with the ISO-standard is not needed.

What issues can affect the quality of a face image?

- Image properties like the resolution.
- Image appearance characteristics like the exposure or noise.
- Scenery characteristics like lightning and background.
- Complete or partial face covering.
- The behavior of the subject.
 - Closed or open mouth.
 - Any kind of expression, e.g. smiling or neutral.
 - Head pose, e.g. frontal or rotated in any direction.

The next slides are examples of rated images of faces. These lineups are guidelines for how you can assess the images in the survey.



1/5 - Poor



2/5 - Bad



3/5 - Fair



4/5 - Good



5/5 - Excellent



1/5 - Poor



2/5 - Bad



3/5 - Fair



4/5 - Good



5/5 - Excellent



1/5 - Poor



2/5 - Bad



3/5 - Fair



4/5 - Good



5/5 - Excellent



1/5 - Poor



2/5 - Bad



3/5 - Fair



4/5 - Good



5/5 - Excellent



1/5 - Poor



2/5 - Bad



3/5 - Fair



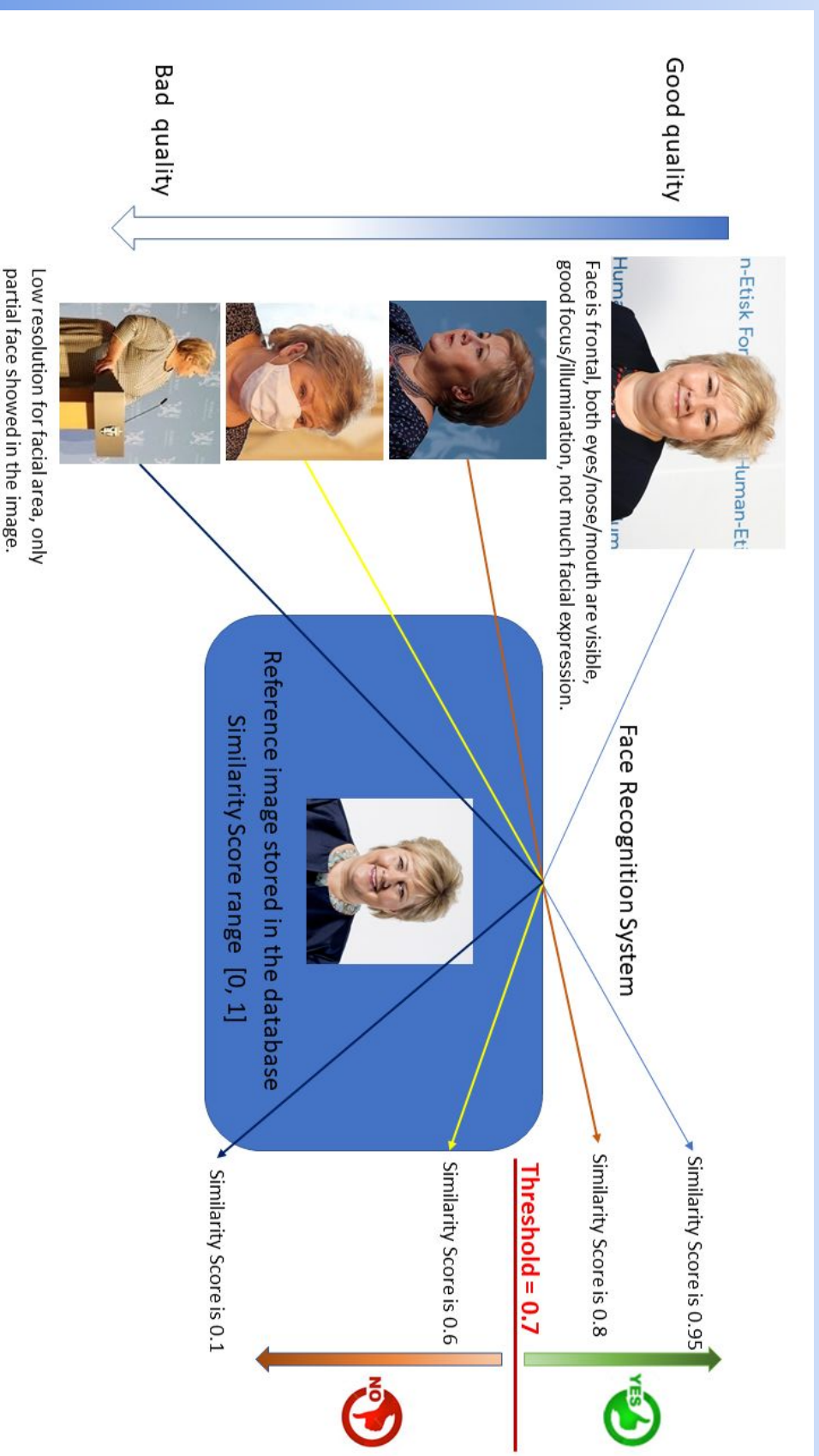
4/5 - Good



5/5 - Excellent

How will the rated images be used?

- Low-quality face images will be rejected by the face recognition system, see an example below.



You are now ready to perform the survey.

Thank you for participating!

Bachelor group

Appendix E

Results

Image	ISO Metric	FaceQNet	Average Subjective score	STD	AVG STD
001.jpg	0,9907	0,4111	4,8125	0,5266	0,6707
003.jpg	0,6875	0,3686	3,6250	0,5995	
006.jpg	1,0000	0,0084	3,8824	0,9630	
009.jpg	1,0000	0,0000	2,4706	0,7759	
015.jpg	0,9766	0,4286	4,8235	0,5128	
018.jpg	0,9299	0,2217	3,5882	0,6910	
021.jpg	0,9860	0,5271	4,5625	0,4961	
024.jpg	0,7804	0,3476	2,3529	0,9037	
030.jpg	0,9953	0,6548	4,4706	0,8484	
033.jpg	0,9486	0,3855	3,5882	0,6910	
035.jpg	0,8037	0,6104	4,5882	0,5999	
036.jpg	0,8458	0,3294	2,4118	0,8443	
037.jpg	0,7897	0,3802	4,4118	0,8443	
038.jpg	1,0000	0,3246	3,7647	0,6444	
039.jpg	0,0000	0,4095	3,1176	0,5823	
040.jpg	0,9299	0,5729	4,4706	0,8484	
041.jpg	0,9860	0,3932	3,1875	0,7262	
042.jpg	0,9907	0,4658	4,6471	0,7624	
043.jpg	0,5561	0,4019	3,1765	0,7059	
044.jpg	0,6262	0,3480	4,8750	0,3307	
045.jpg	0,9766	0,4161	3,1176	0,6758	
046.jpg	1,0000	0,4822	4,7647	0,4242	
047.jpg	0,9813	0,3808	3,2941	0,5703	
048.jpg	0,9953	0,5513	4,6667	0,6667	
049.jpg	0,9206	0,4507	3,1176	0,7579	
050.jpg	0,6168	0,5092	4,5882	0,5999	
051.jpg	0,2336	0,5099	3,0000	0,7906	
052.jpg	0,5935	0,5409	4,5882	0,6910	
053.jpg	0,9813	0,3946	2,9412	0,6390	
054.jpg	0,7523	0,3309	4,5882	0,6910	
055.jpg	0,0280	0,6737	3,3125	0,6818	
056.jpg	0,7336	0,7174	4,4706	0,7759	
057.jpg	0,0000	0,5610	3,1250	0,9922	
058.jpg	0,1916	0,5849	4,5294	0,8484	
099.jpg	1,0000	0,4284	4,7333	0,4422	

100.jpg	0,0000	0,5990	3,2000	0,6532
103.jpg	0,4953	0,7604	4,6000	0,6110
106.jpg	0,0000	0,5608	3,1333	0,6182
109.jpg	0,3551	0,6265	4,6667	0,5963
112.jpg	1,0000	0,2652	2,8000	0,8327
128.jpg	0,9907	0,3013	4,7333	0,5735
129.jpg	0,9346	0,2827	3,9333	0,5735
130.jpg	0,8364	0,4003	4,4000	0,4899
131.jpg	0,4720	0,2389	2,5625	0,9980
132.jpg	0,6215	0,3243	4,6667	0,8692
133.jpg	0,3645	0,2784	3,7333	0,6799
134.jpg	0,9720	0,6336	4,7333	0,5735
135.jpg	0,8879	0,1859	2,8000	0,8327
136.jpg	0,9907	0,4071	4,8000	0,5416
137.jpg	0,9953	0,3001	3,6667	0,5963
138.jpg	1,0000	0,4111	4,3333	0,5963
139.jpg	0,8925	0,1546	2,6000	0,7118
140.jpg	0,0047	0,3715	3,0000	0,6325
141.jpg	0,0187	0,4456	2,2000	1,1075
142.jpg	0,0000	0,2996	3,0000	0,6325
143.jpg	0,1495	0,3395	2,5333	1,0873
144.jpg	0,0234	0,2438	3,1333	0,6182
145.jpg	1,0000	0,2885	2,2667	1,0625
146.jpg	0,9813	0,3951	3,0667	0,5735
147.jpg	0,9533	0,4782	1,8000	0,9798
148.jpg	0,9813	0,3116	3,0000	0,6325
149.jpg	0,9533	0,4144	2,6667	0,9428
150.jpg	0,9766	0,3789	3,1333	0,4989
152.jpg	0,0000	0,3915	3,1333	0,4989
153.jpg	0,0374	0,3373	1,0667	0,2494
154.jpg	0,9766	0,4390	3,0667	0,5735
155.jpg	0,0374	0,3915	1,0667	0,2494
212.jpg	1,0000	0,1135	4,3333	0,6992
221.jpg	0,0000	0,3792	2,0000	0,6325
222.jpg	0,9953	0,3232	4,4000	0,4899
223.jpg	0,9533	0,2843	2,8667	1,0242

224.jpg	0,9393	0,5193	4,5333	0,6182
225.jpg	0,8832	0,2879	3,4667	0,7180
226.jpg	0,5841	0,3684	4,2000	0,6532
227.jpg	0,8178	0,2165	2,6667	0,7888
228.jpg	0,5748	0,6104	4,7333	0,5735
229.jpg	0,8738	0,5319	3,9333	0,4422
230.jpg	0,9907	0,4218	4,6000	0,8000
231.jpg	1,0000	0,2574	2,8000	1,0456
232.jpg	0,0000	0,2767	2,1333	0,6182
233.jpg	0,0000	0,2899	2,0667	0,4422
234.jpg	0,6308	0,2310	2,2667	0,7717
235.jpg	0,0000	0,1362	2,2667	0,6799
236.jpg	0,5935	0,2729	2,1333	0,4989
237.jpg	0,9206	0,2160	2,2000	0,5416
238.jpg	0,0000	0,4020	2,2667	0,5735
239.jpg	0,0000	0,3188	2,2000	0,5416
240.jpg	0,9813	0,2251	2,3333	0,6992
241.jpg	0,0000	0,2506	2,2000	0,5416
242.jpg	0,9813	0,2491	2,3333	0,5963
243.jpg	0,0000	0,2750	2,0667	0,5735
244.jpg	0,6121	0,3527	2,4000	0,7118
245.jpg	0,0000	0,2813	2,3333	0,5963
246.jpg	0,0000	0,3038	2,1333	0,4989
247.jpg	0,0000	0,2375	2,2000	0,6532
248.jpg	0,0000	0,4231	2,3333	0,6992
249.jpg	0,0000	0,3863	2,1333	0,4989
250.jpg	0,0000	0,3899	2,2667	0,6799

Image	ISO Metric	FaceQNet	Average Subjective score	STD	AVG STD
002.jpg	0,9953	0,4544	2,7647	0,8065	0,7891
004.jpg	0,5841	0,4260	1,8750	0,6960	
005.jpg	0,8411	0,5113	2,0000	0,6860	
007.jpg	0,5841	0,4070	1,8824	0,5823	
008.jpg	0,5607	0,4693	2,2353	0,9412	
010.jpg	0,0701	0,3271	2,7059	1,0718	
011.jpg	1,0000	0,4761	2,3529	0,9037	
012.jpg	0,9766	0,4539	4,2353	1,0017	
013.jpg	0,4065	0,4222	4,3529	0,7624	
014.jpg	0,0701	0,3271	2,7647	1,0017	
016.jpg	0,0000	0,0816	1,7059	0,8235	
017.jpg	0,5561	0,4115	3,7059	0,8921	
019.jpg	0,4065	0,4222	4,2941	0,8235	
020.jpg	0,7430	0,4320	3,5625	0,8638	
022.jpg	1,0000	0,2225	3,6471	0,8360	
023.jpg	0,0701	0,3271	2,4118	1,1406	
025.jpg	0,9766	0,4539	4,2500	0,9682	
026.jpg	0,9019	0,4124	3,8824	0,7579	
027.jpg	0,4065	0,4222	4,4375	0,7043	
028.jpg	0,9953	0,4028	3,6875	0,7680	
029.jpg	0,9813	0,3405	3,5882	0,9113	
031.jpg	0,9673	0,3864	4,2941	0,8235	
032.jpg	0,0000	0,4444	4,1765	0,8565	
034.jpg	1,0000	0,4878	4,0588	0,6390	
101.jpg	0,6168	0,4848	1,6667	0,5963	
102.jpg	0,3551	0,4758	2,4000	0,9522	
104.jpg	0,9953	0,4520	2,4667	1,0242	
105.jpg	1,0000	0,4640	2,2667	0,7717	
107.jpg	1,0000	0,2225	3,6250	0,6960	
108.jpg	0,9953	0,4508	2,6000	0,8794	
110.jpg	0,5654	0,4825	1,8667	0,7180	
111.jpg	0,9673	0,3551	2,6000	1,0198	
113.jpg	1,0000	0,2225	3,7333	0,6799	
114.jpg	1,0000	0,3352	3,8667	0,6182	
115.jpg	0,9766	0,4539	4,4000	0,8000	

116.jpg	1,0000	0,3621	4,4000	0,6110
117.jpg	0,8832	0,4323	4,4667	0,6182
118.jpg	0,9673	0,5878	3,9333	0,8537
119.jpg	0,9907	0,4292	4,6000	0,6110
120.jpg	0,3178	0,3251	2,4667	0,8055
121.jpg	0,9673	0,3551	2,9333	0,9286
122.jpg	0,9720	0,3677	4,4000	0,6110
123.jpg	0,0000	0,4118	4,3333	0,5963
124.jpg	0,9634	0,5676	4,0667	1,0625
125.jpg	0,4720	0,1357	1,4667	0,7180
126.jpg	0,9907	0,5254	4,1333	0,8844
127.jpg	0,9252	0,4211	3,6667	0,8692
197.jpg	0,8785	0,3290	2,2000	0,7483
198.jpg	1,0000	0,4296	2,1333	0,7180
199.jpg	0,4439	0,4980	2,0667	0,7717
200.jpg	0,6075	0,4792	2,1333	0,7180
201.jpg	0,2336	0,3111	2,2667	0,7717
202.jpg	0,9907	0,4292	4,5333	0,7180
203.jpg	0,3178	0,3251	2,4000	0,6110
204.jpg	0,2570	0,4481	2,1333	0,8055
205.jpg	0,5748	0,3886	2,1333	0,7180
206.jpg	0,9907	0,4292	4,6000	0,8000
207.jpg	0,3178	0,3251	2,4000	0,8794
208.jpg	0,9673	0,3551	2,6000	0,9522
209.jpg	0,9907	0,5130	4,2000	0,5416
210.jpg	1,0000	0,3486	4,0667	0,7717
211.jpg	1,0000	0,5137	4,3333	0,6992
213.jpg	0,9626	0,3379	3,8667	0,6182
214.jpg	0,0000	0,1100	1,7333	0,6799
215.jpg	0,9299	0,5951	3,9333	0,7717
216.jpg	1,0000	0,5087	4,3333	0,6992
217.jpg	0,7710	0,4063	3,7333	0,6799
218.jpg	0,9720	0,4969	4,3333	0,7888
219.jpg	0,9673	0,4064	4,0000	0,8165
220.jpg	1,0000	0,3679	4,0667	0,7717

Image	ISO Metric	FaceQNet	Average subjective score	STD	AVG STD
059.jpg	0,1822	0	2,6471	0,9037	0,7872
060.jpg	0,4673	0	2,3529	1,0256	
061.jpg	0,0374	0,5076	2,9412	1,1617	
062.jpg	0,2757	0,4917	2,6875	1,1022	
063.jpg	0,5561	0	2,1875	0,8817	
064.jpg	0	0	1,125	0,3307	
065.jpg	0,9813	0,4149	2,3529	1,0256	
066.jpg	0,9439	0,2412	3,4706	1,0357	
067.jpg	0,8551	0,3757	3,8824	0,963	
068.jpg	0,1495	0	2,875	0,9922	
069.jpg	0,9579	0,2228	2,375	0,927	
070.jpg	0,0748	0,5094	1,5	0,5	
071.jpg	0,3692	0	2,3529	0,9666	
072.jpg	1	0	1,8235	0,5128	
073.jpg	0,8738	0,2261	1,875	0,5995	
074.jpg	0,257	0	3,0588	1,392	
075.jpg	0,8645	0	2,4375	0,7881	
076.jpg	0,9626	0	1,6471	0,6809	
077.jpg	0,9766	0	2,25	0,75	
078.jpg	0,1869	0	1,8235	0,6169	
079.jpg	0	0	1,5882	0,4922	
080.jpg	0,7009	0	3,3125	1,044	
081.jpg	0	0	1,7647	0,73	
082.jpg	0,0981	0	1,6875	0,583	
083.jpg	0,7617	0,2452	3,4706	1,091	
084.jpg	0,9907	0	1,9412	0,8022	
085.jpg	0	0	1,9375	0,7474	
086.jpg	0,5748	0	3,25	0,9682	
087.jpg	0,5981	0,2931	3	0,7906	
088.jpg	0,7757	0	3,0588	0,8022	
089.jpg	0,9813	0,3358	2,0588	0,8725	
090.jpg	1	0	3,1765	0,7059	
091.jpg	0,6729	0,2629	3,375	0,857	
092.jpg	1	0,549	1,4706	0,4991	
093.jpg	0,9766	0	1,9333	0,6799	
094.jpg	0,3458	0	3	0,8402	

095.jpg	0,9813	0,4028	2,6471	1,0256
096.jpg	0,5374	0,2208	3,625	1,0533
097.jpg	0	0	1,2941	0,5703
098.jpg	0	0	1,0588	0,2353
151.jpg	0,8598	0,3441	2,3333	0,7888
156.jpg	0,3364	0,2542	1,9333	0,7717
157.jpg	0,8645	0,1951	2,1333	0,6182
158.jpg	0,9486	0,3999	3	0,8944
159.jpg	0,3551	0	3,0667	0,9286
160.jpg	0,1682	0	2,8	1,0456
161.jpg	0,7991	0,3449	2,6667	0,8692
162.jpg	0,757	0	1,5333	0,6182
163.jpg	0,9486	0,088	2,0667	0,6799
164.jpg	0,785	0	3	0,8165
165.jpg	0,0374	0	1,8667	0,6182
166.jpg	0,986	0	2,7333	0,8537
167.jpg	0,8224	0	2,9333	0,8537
168.jpg	0,6449	0	3,3333	0,9428
169.jpg	0,9953	0,2443	2,0667	0,6799
170.jpg	1	0,264	1,4667	0,4989
171.jpg	1	0	2,4667	1,0242
172.jpg	0,5935	0,3558	2	0,6325
173.jpg	0,0047	0	2,0667	0,6799
174.jpg	0,6215	0	3,1333	0,9568
175.jpg	0,9766	0	2,5333	0,9568
176.jpg	0,4206	0	2,2	0,9092
177.jpg	0,9393	0	3	0,8944
178.jpg	0,486	0	1,8667	0,718
179.jpg	0,1449	0	2,4	0,611
180.jpg	0,9112	0	1,5333	0,6182
181.jpg	0,3084	0	3,1333	0,8055
182.jpg	0,5888	0	2	0,8165
183.jpg	0,9907	0	2	0,7303
184.jpg	0,9579	0	3,0667	1,0625
185.jpg	0,3131	0	3,4	1,0198
186.jpg	0,6262	0,2745	2,4667	0,8844
187.jpg	0,4159	0,2571	3	0,8944

188.jpg	0,8551	0,0908	1,5333	0,6182
189.jpg	0,8318	0	1,7333	0,5735
190.jpg	0,5421	0,4019	1,9333	0,6799
191.jpg	0,1542	0	1,5714	0,6227
192.jpg	0,528	0	3,0667	0,9286
193.jpg	0,229	0,3558	4	1,0954
194.jpg	0,9533	0,4177	1,6667	0,5963
195.jpg	0,9579	0,2203	1	0
196.jpg	0,9673	0,5343	2,8667	0,8844
251.jpg	0,8738	0,2593	2,8667	0,718
252.jpg	0,2991	0	3,2	0,8327
253.jpg	0,9112	0	2,6	0,7118
254.jpg	0,7196	0,4148	2,7333	0,8537
255.jpg	0,7991	0	2,4667	0,8055
256.jpg	0,8505	0	3,4	1,0832
257.jpg	0,6682	0	1,8667	0,6182
258.jpg	0,6121	0,2121	2,1333	0,8055
259.jpg	0,6822	0	2,7333	0,6799
260.jpg	0,4766	0,4381	2,9333	0,7717
261.jpg	0,5187	0	2,1333	0,6182
262.jpg	0,9486	0,5003	2,3333	0,8692
263.jpg	0,5	0,1772	2,8	0,8327
264.jpg	0,4766	0,3786	3,1333	0,9568
265.jpg	0,2664	0,5111	2,3333	0,6992
266.jpg	0,9579	0	2,8	0,6532
267.jpg	0,6495	0,2851	3,2	1,0456
268.jpg	0,3364	0	1,2	0,4
269.jpg	1	0	2,8	0,8327
270.jpg	0,3318	0	2,4	0,8
271.jpg	0,3972	0,4724	2,8667	0,6182
272.jpg	0,1449	0	1,4667	0,4989
273.jpg	0,9626	0	1,8	0,6532
274.jpg	0,5748	0	1,9333	0,5735
275.jpg	0,5701	0	2,8667	0,9568
276.jpg	0,4813	0	1,3333	0,4714
277.jpg	0,8224	0	2,6	0,611
278.jpg	0,7617	0	2,6	0,7118

279.jpg	1	0	2,0667	0,7717
280.jpg	0,2336	0	3,0667	0,9978
281.jpg	1	0	2,5625	0,7881
282.jpg	0,8551	0	1,9333	0,7717
283.jpg	0,1916	0	2,8667	0,8844
284.jpg	0,7991	0	2,4667	0,718
285.jpg	0,2243	0	2,8	0,8327
286.jpg	0,972	0	1,2667	0,4422
287.jpg	0,2103	0,3917	1,4667	0,4989
288.jpg	0,9766	0	2,8	1,0456
289.jpg	0,9953	0	2,4	0,8
control-im	0,3224	0,5104	3,0488	1,1677
control-im	0,3458	0	2,1951	0,7721
control-im	0,1682	0	2,8293	1,0336
control-im	0,0374	0	1,7317	0,6635
control-im	0,7383	0,2444	3,2439	1,077

Appendix F

Meetings With Project Supervisor

We had regular meetings with our project supervisor Ali on Mondays at 12:00. There were also some extraordinary meetings during the project.

Meeting with project supervisor 11.01.2021

Planned agenda

- Getting started.

Summaries

- Think about how we will run the user tests?
- Where will we get face images? From Mobai?
- About the project report: potential for a publication?

Agenda for the next week

- Prepare a time plan.
- Should we write everything in English?

Meeting with project supervisor 18.01.2021

Planned agenda

- Explain the project further to the supervisor.
- Show the time plan to the supervisor.
 - The subjective experiment will take more than a month. Probably one and a half months.
- Discuss further work
 - The final product: local application or web application?

- Time requirements for the subjective assessment (manual observations). How long will it take? Probably many weeks.
- Show and discuss the group rules.

Summaries

- How should we conduct the subjective experiment?
 - Evaluate which platform we want to use and why we decide to use it. We can use QuickEval, SurveyMonkey, LimeSurvey, or some other platform.
 - The images should be shown in a random order.
 - What can we include in our dataset/subjective experiments that are unique?
- Create a structure of the thesis, with names of the chapters and sections. Include the number of pages.

Agenda for the next week

- Send research papers and group rules to Ali.
- Write the first draft for the structure of the thesis. Write down the chapter names and number of pages.

Meeting with project supervisor 27.01.2021

Planned agenda

- Show our project plan
- Show our draft of the project structure with chapter headings and number of pages.
- Subjective assessment:
 - Which score range is the best one to use? 1-5, 1-20?
 - How should we do the survey? Score one and one image? Put multiple images in order of quality score?
- In the report, can we use different tenses (past, present, future) in the same paragraph?

Summaries

- Draft of the project structure:
 - Add numbers to the chapters.
 - Add the number of pages (and required time) to sections.
 - The sections we have written down look good.

- Maybe move the subjective assessment to a separate chapter (if it's long and makes sense to put it in a separate section).
- Excessive theory that is not used in the report will affect the grading of the report.
- Creating the subjective assessment:
 - Check a few datasets that are out there, see what they do and how they do it.
 - Display the images in random order.
- About the thesis writing.
 - Don't jump between tenses. Use Past tense.
 - In the results, use past or present tense.
 - Do a shift to present in the end.

Agenda for the next week

- Send the project plan to Ali.
- Look into one or two data sets.
- Finalize our time plan.

Meeting with project supervisor 01.02.2021

Planned agenda

- Go through our project plan and the comments from Ali.
- Show the supplementary agreement from Mobai. Any thoughts?
- What is a data set? A collection of images?
- Our goal with the application. How should it work?

Summaries

- We went through the project plan and wrote down the comments from Ali.
- Speak to Tom about the supplementary agreement.
- Speak with Mobai about how the application should work.
- Dataset:
 - It's called a dataset or datasets (one word).
 - Dataset: a collection of images.
 - We evaluate the quality of an image.
 - The quality of the dataset: how we present and run the test decides the quality of the dataset. The higher the quality, the more reliability.

Agenda for the next week

- Create a pilot test for the subjective experiment. Be prepared for comments and changes. Have a finished test by the 15th of February.

Meeting with project supervisor 08.02.2021

Planned agenda

- Show our pilot of the subjective experiment.
- Mobai's and Ali's definition of a quality image are two different things?
- We will use SurveyMonkey/QuickEval for the subjective experiment, with random image order.
- Our dataset consists of 280 images. Mobai wants all participants in the subjective experiment to rate all of the 280 images in the dataset.
- How do we perform the subjective experiment?
- How do we pick the best 100 images from the "Faces in the wild" dataset?
- Invite Ali to join a meeting with Mobai.

Summaries

- Use the same rating system for the experiment.
- Can we split the images in the subjective experiment into three sets? Ask Mobai.
- How do we perform the subjective experiment?
 - Suggestion for the subjective experiment: split the experiment into 3 sessions. One session for each image folder in the dataset. Should the sessions be split over multiple days or the same day with breaks between sessions?
 - Remember: if we split into multiple sessions we need to track the sessions of the participants so we can connect the different sessions to the same participant.
 - Have a training set first to train the participants for the experiment?
 - Don't mention "quality" (or, define clearly what we mean by quality.
 - We want to test the face recognition quality.
 - Define who we want to ask to take the subjective experiment.
- How do we pick the best 100 images from the "Faces in the wild" dataset?
 - Variant image quality matters. I.e., pick a variety of images where the faces are covered or less visible in some way, e.g. with sunglasses, different poses, different perspectives and so on.
 - Also pick images where the faces are clearly visible ("passport-like").
- There are three kinds of metrics:
 - Full access metrics

- Reduced reference
- No reference
- Ali will join our next meeting with Mobai on Tuesday.

Agenda for the next week

- Create clear instructions for the users of the subjective experiment. What do we want the users to evaluate?
- Test the datasets with the IQMs and check out the results.

Meeting with project supervisor 11.02.2021

Planned agenda

- Talk about our meeting with Mobai on Tuesday.
- Use face image quality score from 1-10?

Summaries

- Find the smallest of all of the images in the datasets. Downsize all the images to the same size (and keep the aspect ratio).
- Use the face image quality scores 1-5, or maybe 1-7?

Meeting with project supervisor 15.02.2021

Planned agenda

- Go through the comments on the survey instructions
- Should all of the images be of the same size?
- Dataset 3, “Faces in the wild” is not good enough.

Summaries

- We went through the survey instructions and made the necessary adjustments:
 - We can create a control dataset. We can use images of ourselves.
 - We will create image lineups from the control dataset.
 - Have Mobai rate the images in the survey instruction.

Meeting with project supervisor 22.02.2021

Planned agenda

- Update on our current status:

- Currently working on the frontend and backend for the web application.

Summaries

- We will continue working on the web application.

Meeting with project supervisor 01.03.2021

Planned agenda

- Show experiment instructions:
 - At the beginning of the instructions, include “a very brief intro about the face recognition system”?
 - Delete the slides at the end?

Summaries

- Survey instructions:
 - Send 15 images each to Mobai and have them rate all of them. Ask Guoqiang to have them rated by Wednesday.
 - In the image lineups: image with the lowest score on the left.
 - Random is good. In a lineup, we include different pictures of each of the members. Have a 1-5 rating for each member.
- Write about our dataset in the thesis.
- Before sending out the subjective experiment:
 - Test the experiment ourselves first.
 - Get approval of the experiment by sending it to Ali and Guoqiang.
 - Then send it out to a few people, not all of them.
- Start writing about things in the thesis that will not change.

Agenda for the next week

- Send 15 images each to Mobai and have them rate all of them. Ask Guoqiang to have them rated by Wednesday.
- Ask mobai if they want us to create our own dataset.

Meeting with project supervisor 08.03.2021

Planned agenda

- About the subjective experiment on QuickEval.

Summaries

- About sending out the subjective experiment to the participants:
 - Do not ask the users for any personal information in the survey.
 - Add “Mobai” in the title of the email. The title of the email is important. We need to engage people.
 - Use Blind carbon copy (BCC) when sending the email to survey participants.

Agenda for the next week

- Send the mail to Ali and Guoqiang first.
- Send the mail to the participants.

Meeting with project supervisor 15.03.2021

Planned agenda

- About the dataset we are creating. How do we create a “wow factor”?

Summaries

- Suggestions for the dataset we are creating:
 - Make it a bit challenging.
 - Have other people in the background.
 - Painting of someone in the background.
 - Partial faces (like the selfie dataset).
 - Test images with the metrics.
- We discussed different ways to display the data from the experiments:
 - Calculate the different correlation values. Which metric is performing better?
 - 3D-plot, Spider chart, Rose chart.
 - How we present it is important. It will stick in the mind of the reviewer.
 - What do we want to present to the user?

Meeting with project supervisor 22.03.2021

Planned agenda

- Is Ali gone next Monday?
- Delimitation (avgrensning?)

Summaries

- There will be a meeting next Monday as usual.
- We talked about the structure of the thesis:
 - Why we made the decisions we did.
 - Write about why we went for a web platform. Pros and cons.
 - Write the skeleton for the whole thesis.
 - Write the sections so we easily can move them around.
 - Add flow between the sections.

Meeting with project supervisor 29.03.2021

Planned agenda

- Thesis structure.

Summaries

- We discussed how we should structure the thesis (details not included).

Agenda for the next week

- Send thesis chapter 1 to Ali.

Meeting with project supervisor 06.04.2021

Planned agenda

- Talk about the thesis.

Summaries

- Ali will look at the final draft of our chapters:
 - Send him the final draft of a chapter when we finish it.
 - He will go through the whole report in the end.
- Some other people will join our meeting on the 19th.
- Find a name for our dataset.

Meeting with project supervisor 12.04.2021

Planned agenda

- Review thesis chapter 1.
- Will Ali join our meeting with Mobai tomorrow?

Summaries

- We reviewed chapter 1:
 - Be more strict about using paragraphs. The language is good, just present it in a better way.
- Run the matrices on our own dataset.

Meeting with project supervisor 19.04.2021

Planned agenda

- Discuss thesis chapter 2.
- Creating our own dataset is completed.

Summaries

- We discussed thesis chapter 2 and the comments from Ali. (Comments are not added).
- We will create a second subjective experiment with our own dataset:
 - Select 250 images from the dataset.
 - Add effects like blurring, noise etc. to the images and test them with the metrics. Will the scores change?

Meeting with project supervisor 26.04.2021

Planned agenda

- Discuss user testing.
- Split the second subjective experiment into two parts? 125 images in each part.
- Can Ali edit 50 of the images? How should we edit the images? Blurring, noise, exposure.
- Talk about the results from the subjective experiment.
- Plotting the results from the experiments.

Summaries

- User tests should be done by people at Mobai.
- We discussed the second subjective experiment.
- We will create a third subjective experiment based on the second subjective experiment:
 - Select 50 images from the second subjective experiment. 10 images from each rating category (10 images rated 1 - poor, 10 images rated 2 - bad, and so on).

- Edit the images in different ways and create the new experiment with these images.
- We can add “untypical” distortions.
- We talked about the thesis and the metrics. (Notes not included).
- Plotting the results:
 - Box plot. Shows the distribution of the scores.
 - Show experts, non-experts, and both of them together.
 - Subjective and objective scores should be separate.
- Our dataset:
 - All images in the dataset should be portrait. Fix landscape images to portrait images.

Meeting with project supervisor 03.05.2021

Planned agenda

- All invited participants have completed the second subjective experiment.
- How should we distort the images for the final experiment?
- Discuss thesis chapter 3.

Summaries

- We discussed thesis chapter 3. (Notes not included).
- Create the third and final subjective experiment:
 - Select 50 images, 10 images from each rating category.
 - The image should be as diverse as possible.
 - Add four distortions to each image:
 - Two types of compression, e.g., Facebook Messenger.
 - Two types of image edits, e.g., noise, blur or exposure.
 - The distortions should be reproducible, i.e., something that other people can do. Use e.g., Photoshop or IrfanView.
 - Include the unedited version of the images.
 - Summary: 200 (50 x 4) distorted + 50 non-distorted. In total, 250 images.
 - Use these 250 images and create the final experiment.

Meeting with project supervisor 06.05.2021

Planned agenda

- Talk about the third subjective experiment.

Summaries

- We discussed the third subjective experiment:
 - The dataset should be challenging for the metrics.
 - The changes should not change the subjective scores.
 - Use Telegram compression. Telegram changes the images.
 - The editing we have done is too dramatic. Tune it down.
 - When we add noise or blur to the images, find a value that works for about 30-35 of the images.

Meeting with project supervisor 10.05.2021

Planned agenda

- Discuss the thesis and the future bachelor presentation.

Summaries

- We discussed finishing the thesis.
- We talked about the future bachelor presentation.
- Send the chapters we discussed to Ali by Thursday.
- We will have a meeting Friday afternoon the 14th of May.
- We will have a meeting on Tuesday at 11:00 the 18th of May.

Meeting with project supervisor 14.05.2021

Planned agenda

- Discuss chapter 4 of the thesis.
- Discuss chapter 5 of the thesis.
- Discuss the conclusion of the thesis.
- Discuss the abstract.
- Should we keep the list of acronyms?
- Should we include a glossary?
- How do we properly include photo credits for the images used in the figures?
- Prepare a final schedule for the project.

Summaries

- We will edit chapter 4 and 5, based on the feedback from Ali.
- We are currently finishing the work on chapters 6 and 7. We will send the draft to Ali.
- We will keep the list of acronyms.
- We will not include a glossary.
- We will send Ali our thesis on the 16th of May.

- We have prepared a final schedule for the project.

Meeting with project supervisor 18.05.2021

Planned agenda

- Discuss the feedback on the thesis.
- What to include as appendices?
- What is a fitting name for the thesis?

Summaries

- We will review the thesis based on the feedback from Ali.
- We will further discuss what to include in the appendices.
- "Face Image Quality Assessment" can be a fitting name for the thesis.

Meeting with project supervisor 19.05.2021

Planned agenda

- Discuss our review of chapter 6.
- Discuss the abstract.

Summaries

- We will do some more changes to chapter 6.
- We will have a few meetings with Ali again before the project presentation.

Appendix G

Meetings With Mobai

We had weekly meetings with Mobai every Tuesday at 12:00. This later changed to bi-weekly meetings every other Tuesday at the same time.

Unless stated otherwise, the planned agenda is written by the bachelor group.

Meeting with Mobai 12.01.2021

Planned agenda

- Mobai's planned topics:
 - What do the students need to get going?
 - How do we structure the project?
 - I'd like to structure this a bit like a normal agile project, what do you think? Scrum?
 - Anything else?

Summaries

- The group members got more specific details about the bachelor assignment:
 - Mobai will send us the algorithms (metrics).
 - The people from Mobai explained how the algorithms work.
 - Each algorithm returns a quality score for each single image.
 - Mobai will send us datasets that we can use.
- About the subjective assessment:
 - We use the same datasets for the subjective and objective assessment.
 - Ask about 10 people to participate in the subjective experiment.
 - Have the participants rate the images with a score from 1 to 100.
 - Get the subjects opinions about the images.
 - Calculate the average scores.

- Calculate the correlation between the subjective and objective assessments.
- Display the results as graphs.
- About the objective assessment:
 - The programming language is mostly up to us.
 - Put the application in containers. Using Docker desktop is a good idea.

Future meetings with Mobai:

- Every Tuesday at 12:00. We will keep the meetings short, about 30 min.

Agenda for the next week

- Create a project plan.
- Read face quality assessments and other research papers.

Meeting with Mobai 26.01.2021

(The planned meeting on 19.01.2021 was canceled due to unforeseen circumstances.)

Planned agenda

- We have tested the algorithms and they are running.
- The repository contains three algorithms, which should we use?
- Talk more detailed about what actually are going to do in the project.
- We need datasets.
- Signing a supplementary agreement with Mobai.
- Use containers for the database and the application?
- Is this project a research project?
- What are Mobai's "effect goals" for this project?
- About the subjective assessment: how many images should the participants rate.
- We're planning to do the Scrum development method.

Summaries

- Focus on including the ISOMETRICS and FaceQNET algorithms. Include the third if there is time.
- What we want to determine in this project (in the short term and in the long term):
 - We want to see which one of these open source algorithms that is the best, based on the results from the subjective assessment.

- The police do not have algorithms that check the quality of passport images, they do this manually. Automating this would be very valuable.
- Specialists are creating the ICAO and ISO standards, which defines good or bad quality. Mobai wants to create a better algorithm based on these two algorithms and the subjective assessments.
- Create a web application. Platform independent.
- Use docker containers to easily integrate into their system, because they also use Docker containers as well. Use two containers: one for the frontend and one for the backend.
- When it comes to the application, do not think about security or authentication, Mobai doesn't care about that. Think about the front- and backend.
- A dataset is a collection of images. The dataset will be sent to us.
- The report will be a research report.
- Participants of the subjective assessment should look at the same images:
 - Get an average from the participants.
 - Rating the images for about 30-60 min.
 - Do a quick evaluation of each photo.
- About Scrum: 2 week Scrum. Use these meetings as demos and planning.
- We are allowed to come up with new ideas during the project. No need to get permission for everything.

Agenda for the next week

- Finish the project plan.
- Start with a UI shell for the application.
- By Monday, decide if we want a meeting next Tuesday.
- Read and understand more about face quality metrics.

Meeting with Mobai 09.02.2021

Planned agenda

- Our supervisor will join us during this meeting and discuss different topics.
- What defines a good face image (resolution, pixels, how much of the face is shown)?
- Do we call a good face image "face quality", "face recognition", "face identification" or something else?
- Create instructions on the subjective experiment? What should we ask the subjects?
- Discuss if 300 images is too much in one go for the subjective experiment and if we should split it into three groups with a mix of all three folders (datasets) in each group.

Summaries

- About the subjective experiment:
 - Include images from low to high quality.
 - We want to collect ground truth data.
 - Get 15 participants to join.
 - Look at the average score from the participants.
 - Splitting the experiment into three sessions is fine.
 - Give the observer some training before running the test. Create instructions.
 - No specific background is required for the participants (except for the basic training test).
 - Advice: stick with the experts.
 - We (the bachelor group and our supervisor) can do the test ourselves. Mobai can get us participants.
 - Still yet to decide which range of scores should we use for the experiment. Maybe 1-10?

Agenda for the next week

- Create the first draft of the instructions for the subjective experiment.

Meeting with Mobai 16.02.2021

Planned agenda

- Show the current website design.
- Show the pilot survey in QuickEval. (Note that we had to downscale the images.)
- Show the instruction manual for the subjective experiment.
- Create a training dataset to be used in the instructions:
 - Create some (4-5?) image lineups with the same person in each lineup, with different attributes and poses.
 - Use images from the datasets or our own images?
 - Have Mobai rate each image in the lineup. (Get ground truth data from Mobai)
- Do Mobai have an example of rating from 1 to 5 (except for the previous research papers)?
- Have Mobai rate the images in the survey instructions for us.

Summaries

- We will have a meeting with Brage next week and brainstorm more about the web application and its features.

- We can use Python and Flask for the backend.
- Examples of high/low res images in the instructions manual should be moved to the last slide. This to make it clear for the users that the image resolution is not the most important thing.

Agenda for the next week

- Send Guoqiang a link to the pilot survey on QuickEval.
- Send Guoqiang images in the survey instructions so he can rate them from 1-5.

Meeting with Mobai 19.02.2021

Planned agenda

- Brainstorm around the face quality applications with Brage.
- How should the GUI of the application look?
- Any recommended technologies we should use?
- Design and functionality of the app

Summaries

- The frontend will be used mostly for demo purposes.
- Prioritize the development of the backend:
 - Create a backend API service. REST API.
 - The API should receive an image and return a score.
 - Create the backend so it's easy to add new algorithms.
 - Python based backends are generally good. We can use Django or Flask.
- Put it all into Docker containers.
- Create a website where you can upload one or more images.
- Contact Martin of Mobai to get help with the design and functionality of the web application.

Agenda for the next week

- Do research on REST API.

Meeting with Mobai 23.02.2021

Planned agenda

- We have to resize the images in the datasets. Should the subjects evaluate the resized or the original images?

- The images in the dataset “Faces in the wild” are not working because of the small distance between the eyes.
- More talk about the frontend and the backend for the web application:
 - We have started using Flask for the backend and React.js for the frontend.

Summaries

- The subjects should evaluate the resized images.
- The metric results must also be from the same resized images.
- Guoqiang will send us a new dataset this week to replace the “Faces in the wild” dataset.
- Using Flask and React.js for the web application is fine.
- We can create a separate Docker container for each metric, for easier maintenance? Start by putting everything in one container.
- We will now have bi-weekly meetings instead. Next meeting 2. March, then bi-weekly.
- Get in touch on Teams or email if we need some input before the meeting.

Meeting with Mobai 02.03.2021

Planned agenda

- At the beginning of the survey instructions, we want a very brief intro about the face recognition system.
- We’re thinking about creating our own dataset. Do you think that is a good idea?
- Do you want us to create a specific dataset that you can experiment with?
- Include tilted images in the subjective and object experiments.
- We need ratings for the images we’re going to include in the instruction manual.
- About the web application

Summaries

- Guoqiang will create a slide about the face recognition system that we can include in the instructions, between slide two and three. He will send it to us tomorrow.
- Mobai thinks it’s a great idea to create our own dataset. We will create a dataset of ourselves (selfies) with different properties: Indoor, outdoor, different places (e.g. at the store), at different times during the day, with different lighting, and so on.
- We will include tilted images and write about them in the thesis.

- Guoqiang will rate the images we are going to use in the survey instructions and send them to us today.
- How the demo should work: select local images (“browse files”), send the images to the backend, get the results back from the backend, then save the results as a file.
- Being able to add more algorithms to the web application is more important than processing multiple images.

Meeting with Mobai 16.03.2021

Planned agenda

Summaries

- Ask for technical help with the frontend and backend.
- We will invite the rest of the people in our list to join the subjective experiment.

Summaries

- Create API for uploading the file.
- Create API for running the metrics.
- Make Flask render the React frontend.
- Have everything in one Docker container. Split into two containers if possible.
- Add Martin to the Github repository of the project.

Meeting with Mobai 30.03.2021

Planned agenda

- How to deploy the web application?
- Should the web application be public? Public for NTNU people?
- How should the API (backend) work?
- Should the web application be able to process the scores from the subjective experiment?

Summaries

- The web application will be used for different demo purposes.
- About the backend:
 - The backend is the most important part.
 - Do a request to the backend.
 - Set the return format as a JSON object.

- The JSON object should contain all the data, like scores from the metrics, brightness score, and so on.
- Return the data as a JSON file. This makes it system independent.
- Mobai will handle the returned file.
- About the frontend:
 - Display/preview the image (if one image is selected) in the frontend.
 - Display the scores for all of the images (including brightness and so on) returned from the backend.
- The web application does not need to handle the scores from the subjective experiment. We will do the correlation calculations and so on manually.
- Forward technical questions to Martin.
- Add Guoqiang to the Github repository of the project.

Meeting with Mobai 13.04.2021

Planned agenda

- Can we publish the report?
- We need two more from Mobai to complete the experiment:
 - We need at least a total of 15 replies.
 - Five people from Mobai have replied, plus two more non-Mobai experts.
 - The rest of the replies are from non-experts.
- What else do we need to complete?
 - Have the API working as desired
 - Decide on the correct JSON format
 - Display the results from the JSON format in the web application
 - Put in containers
- Include C++ code in the project? (No time).
- Ask about the API:
 - Without using the frontend, how exactly do you expect to do an API call?
 - Is it enough if the backend returns a JSON file with the data, as it is doing now?
 - We can add image attributes from ISO metrics to the JSON file.
 - Have Mobai looked at how to put it in Docker containers?
- Can Mobai help us with setting up the containers? Or, send us an example of how it's done.
- Give a name to the web application?
- Shall we deploy (create a deployment build) the web application? We have only tested the production build.

- How should we format the JSON file returned from the backend?
- Missing features in the frontend, and are they needed?
 - Prevent the user from uploading other files than images (sort of works).
 - Display a loading bar/screen when the metrics are running.
 - If one single image is uploaded and run through the metrics, display the metric scores and image attributes.
- Many images are failing when running FaceQnet.

Summaries

- Publication is OK with Mobai.
- Martin will do the subjective experiment.
- Returning an array of JSON objects works fine. Display the results in the frontend. If Mobai need to do something else with the JSON file/array, they will handle that themselves.
- We do not need to create a deployment build for the web application. We can submit the development build.
- Martin will take a look at how to put the web application into Docker containers this week and let us know.
- The current name (“Face Image Quality Assessment”) of the web application is fine.
- Add a loading screen/bar to the frontend if it doesn’t take too much effort. (Maybe use some premade module?).
- Try to add one more metric (FaceImageQuality).
- Share the results with Mobai when we have something to show.
- If the images fail while running the metrics, set the score to 0.
- Test all the images and send the images that fail to Guoqiang.

Meeting with Mobai 27.04.2021

Planned agenda

- Cropping the faces on the images fails on many images. Can we run the FaceQnet metric without cropping the images?
- Any updates regarding Docker?

Summaries

- Cropping is needed for the FaceQnet metric to work properly. We can get some weird results if we include the background of the images.
 - Add new code for cropping the faces in the images. Guoqiang will send us the code. If an image still fails (the code doesn’t detect the face), set the score to 0.

- Some images are “filtered out” by the ISO metrics. Too narrow eye distance (lower than the ISO standard) can be one of the reasons. Give these images a score of 0.
- Mobai will update us on Docker soon.

Meeting with Mobai 11.05.2021

Planned agenda

- Will this thesis be a noncommercial project? Some of the images in the thesis are only allowed to be included in noncommercial projects.
- Can Mobai read through the report and approve it before we publish it on NTNU Open?
- Are all IQMs no-reference? Do any of them use some other image as reference?
- Feedback from the user testing.

Summaries

- The thesis will not be used in a commercial setting.
- The IQMs do not use other images as a reference when they rate images. Face quality is different from face comparison.
- Guoqiang will send us an email with the user testing feedback.
- Remember to include citations to FaceQnet and ISO metrics in the thesis.

Agenda for the next week

- Send the finished thesis to Gouqiang. Have him approve the thesis before we publish it on NTNU Open.
- Send Guoqiang the data from the subjective experiment after we have finished the report. Include the raw data.
- Send Guoqiang a file with all of the images that we want to use in our thesis. Get approval and references in order.

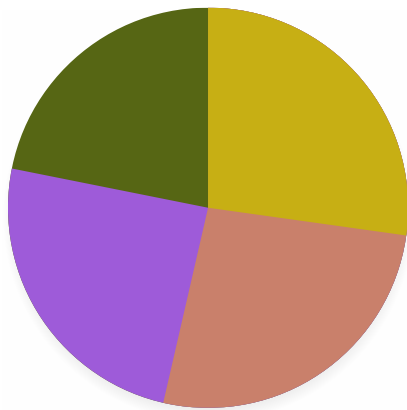
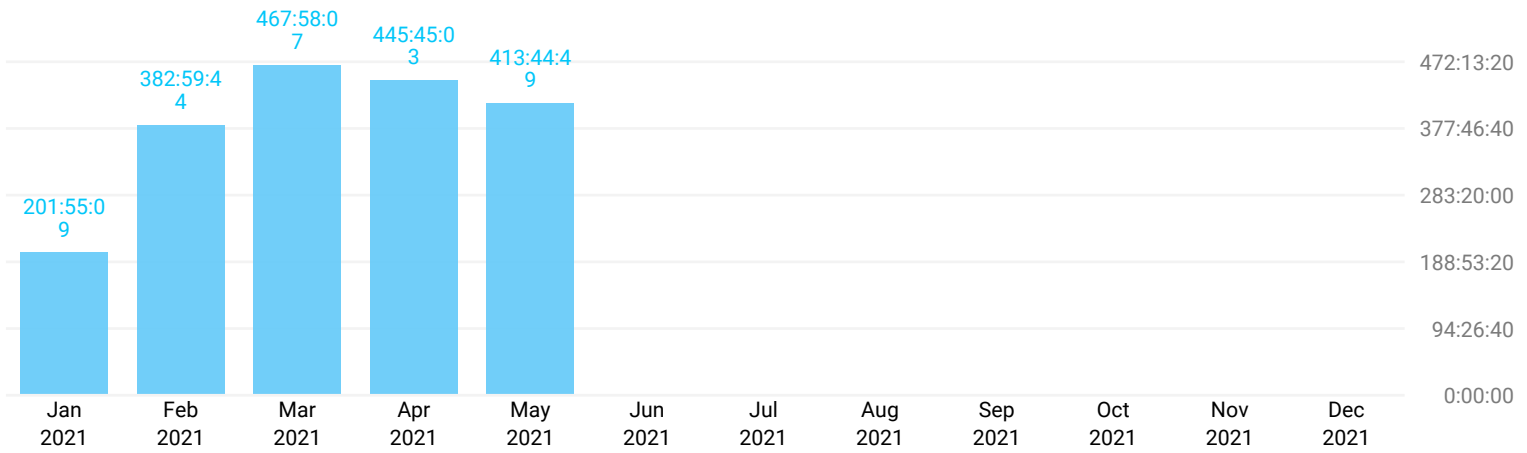
Appendix H

Toggl Reports

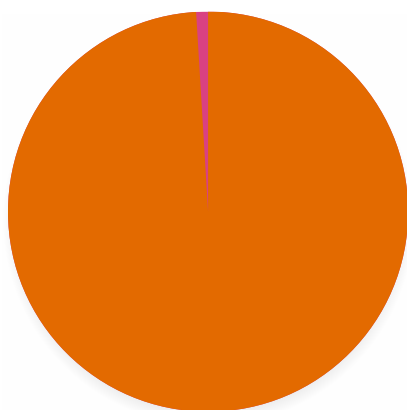
Summary Report

01/01/2021 – 12/31/2021









TOTAL HOURS: 1912:22:52



USER	DURATION
HP Hans Petter	519:18:00
WD Walid Demloj	504:23:09
KG Kjetil Grosberghaugen	469:29:13
JS Julian Skattum	419:12:30



PROJECT	DURATION
BachelorOppg	1894:23:05
Without project	17:59:47

USER - PROJECT	DURATION
 Hans Petter	519:18:00
 BachelorOppg	519:18:00
 Julian Skattum	419:12:30
 BachelorOppg	401:12:43
Without project	17:59:47
 Kjetil Grosberghaugen	469:29:13
 BachelorOppg	469:29:13
 Walid Demloj	504:23:09
 BachelorOppg	504:23:09

