

LONDON'S GLOBAL UNIVERSITY



# Estimating Biases in Facial Analysis Tools

Samuil Stoychev<sup>1</sup>

BSc Computer Science

Supervisor: Prof. Licia Capra

Submission date: Monday 27<sup>th</sup> April, 2020

<sup>1</sup>**Disclaimer:** This report is submitted as part requirement for the BSc Computer Science at UCL. It is substantially the result of my own work except where explicitly indicated in the text. The report may be freely copied and distributed provided the source is explicitly acknowledged

## **Abstract**

The recent advances in cognitive computing and image processing have facilitated the rapid development of facial analysis technologies. Those technologies have been employed to automate and enhance processes in business, medicine, security and other fields of high social importance. However, recent studies expose severe inconsistencies in the behaviour of facial analysis tools and suggest that their performance could be biased against certain demographic groups.

In this project, we examine popular facial analysis tools – Microsoft Face API, Amazon Rekognition, Face++ and Clarifai, and their behaviour when extracting gender, race, age and emotion from various datasets and demographics. We compare their individual performances and, through a combination of statistical analysis and regression modelling, identify potential biases present in the technologies.

We find that the two Big Tech companies significantly outperform their smaller competitors both in terms of accuracy and fairness. Additionally, we find ample evidence that facial analysis tools exhibit demographic biases, although the nature and the extent of those biases vary across different tools and classification tasks.

**Keywords:** facial analysis, algorithmic fairness, demographic bias, misclassification.

### **Acknowledgements**

I would like to extend my thanks to my supervisor – Prof. Licia Capra, for her invaluable guidance and advice throughout the project, and my parents for supporting my studies over the last three years.

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Motivation . . . . .	2
1.2	Project Aims . . . . .	3
1.3	Report Structure . . . . .	3
<b>2</b>	<b>Background</b>	<b>4</b>
2.1	Facial Analysis in the “Algorithm Economy” . . . . .	4
2.2	Algorithmic Fairness . . . . .	5
2.3	Related Work . . . . .	6
<b>3</b>	<b>Data Engineering</b>	<b>8</b>
3.1	Overview . . . . .	8
3.2	Preliminary Research . . . . .	9
3.2.1	Choice of Attributes . . . . .	10
3.2.2	Choice of APIs . . . . .	10
3.2.3	Choice of Datasets . . . . .	12
3.3	Data Preparation . . . . .	13
3.4	API Processing . . . . .	15
3.5	Crowdworking Experiment . . . . .	17
3.6	Data Standardisation . . . . .	18

3.7	Final Dataset Distribution . . . . .	19
<b>4</b>	<b>Methodology</b>	<b>21</b>
4.1	Research Questions . . . . .	21
4.2	Ethical Considerations . . . . .	22
4.3	Accuracy Comparison . . . . .	22
4.4	Correlation Analysis . . . . .	23
4.5	Logistic Regression Analysis . . . . .	24
<b>5</b>	<b>Results</b>	<b>26</b>
5.1	Accuracy Comparison . . . . .	26
5.1.1	Gender Classification . . . . .	26
5.1.2	Race Classification . . . . .	27
5.1.3	Age Classification . . . . .	28
5.1.4	Emotion Classification . . . . .	31
5.2	Correlation Analysis . . . . .	31
5.2.1	Gender Classification . . . . .	32
5.2.2	Race Classification . . . . .	33
5.2.3	Age Classification . . . . .	33
5.2.4	Emotion Classification . . . . .	34
5.3	Logistic Regression Analysis . . . . .	34
5.4	Summary of Results . . . . .	36
<b>6</b>	<b>Limitations and Future Work</b>	<b>38</b>
<b>7</b>	<b>Conclusion</b>	<b>40</b>
<b>A</b>	<b>Code Listing</b>	<b>43</b>
A.1	Guide into the File Structure . . . . .	43

A.2	NimStim JPEG Conversion . . . . .	44
A.3	AirBnb Random Sampler . . . . .	44
A.4	AI Dataset Generation . . . . .	45
A.5	Data Standardisation . . . . .	46
A.6	Definition of Age Misclassification . . . . .	47
<b>B</b>	<b>Crowdworking Experiment Design</b>	<b>49</b>
<b>C</b>	<b>Project Plan</b>	<b>52</b>
<b>D</b>	<b>Interim Report</b>	<b>54</b>

# Chapter 1

## Introduction

### 1.1 Motivation

Facial analysis (FA) refers to the automated processing of facial data in images. Early instances of the technology included facial detection (detecting the presence of a face in an image) and facial recognition (identifying a specific face) [1][2]. However, with the advance of machine vision and deep learning, facial analysis has evolved – the technology can now effortlessly detect and recognise human faces, but also deduce personal attributes such as gender and race. This last capability of facial analysis – to extract information about a subject<sup>1</sup> based on facial data, will be the main focus of our study.

To contextualise the importance of facial analysis technology, we need to consider the wide applicability it has nowadays in virtually every area of social life. In medicine, FA has been trained to detect genetic disorders, matching the accuracy of an expert clinician [3]. In marketing, the technology is being used for building detailed customer profiles [1] and subsequently targeted advertisement. It has found applications in crime and security for surveillance, identifying suspects, biometric identification (the so-called iris technology [4]). Facial analysis has also been adopted by researchers in fields like psychology, social sciences and human-computer interaction, to perform affect analysis [5] or extract demographic features[6][7] from large data sets, saving the time and resources associated with manual labelling.

The examples above serve to highlight the extent to which facial analysis has infiltrated almost every aspect of our lives. The technology is being used in processes of great social importance and it is therefore imperative that it is implemented correctly and ethically. However, recent studies have revealed inconsistencies between different facial analysis tools [6], and some have suggested that many tools exhibit algorithmic biases [7][8]. With this study, we aim to address these issues as we examine the behaviour of four popular facial analysis tools across different data sets and

---

<sup>1</sup>Throughout this report, we adopt the use of *subject* to refer to a person or an individual, in line with the terminology employed in most of the related literature.

demographics, comparing their performances and identifying potential biases.

## 1.2 Project Aims

With this project, we aim to address the issues identified above, setting two main goals. Firstly, we want to compare the accuracy of different facial analysis tools as they perform various classification tasks on different datasets. We would like to know whether there is indeed a gap of performance between different tools and, if so, which tools perform better and which ones perform worse. Is there a single “best” FA tool, or does none of the tools outperform the others across *all* classification tasks? And also, does the choice of dataset (or input) systematically affect the accuracy of the output – for example, do the tools tend to perform significantly better on images taken in a controlled environment with adjusted lighting and clearly visible faces?

Secondly, we want to investigate if the tools exhibit algorithmic biases when extracting demographic features. We are interested whether errors in the technologies are predisposed by certain demographic features and will try to quantify the extent and the significance of those biases. If we find sufficient evidence of algorithmic biases, we are interested to know whether those persist across all tools or are limited to a subset of them. The results of the study should provide clarity about the capabilities and fairness of the facial analysis technology.

## 1.3 Report Structure

In this study, we will be performing an “audit” on commercial facial analysis tools and their classification accuracy. This report aims to summarise all stages of the project – from preliminary research to data analysis and findings.

We start by providing a **literature review** and discussing related work in Chapter 2. We aim to contextualise facial analysis by discussing the social, economic and ethical dimensions of the technology. We introduce the novel problem of algorithmic fairness and see how it has been tackled by previous research.

The rest of the project is split into a data engineering stage and a data analysis stage. In the **data engineering** stage, we run a set of APIs on various data sets performing different types of feature extraction. Where metadata is missing, we create a ground truth via a crowdworking experiment.

In the **data analysis** stage, we subject the data obtained via API processing to statistical and regression analysis. We describe our methodology in Chapter 4 before we interpret the results and their significance in Chapter 5, and conclude the report with a brief summary of limitations and future work, and an overview of the project’s findings.



## Chapter 2

# Background

### 2.1 Facial Analysis in the “Algorithm Economy”

The term “*Algorithmic Economy*”<sup>1</sup> refers to the practice of monetization of algorithms by *vendors* (technology companies) who offer the right to use to *end users* (usually application developers). As facial analysis has become an integral element of many processes in medicine, business and security (as discussed in Section 1.1), technology companies have seized the opportunity to commercialise their own facial analysis technologies and offer it in line with the “Algorithmic Economy” business model.

As of now, the facial analysis market is dominated by *Big Tech* companies (Amazon, Google, Microsoft, etc.) and smaller challengers (such as Face++<sup>2</sup>, Clarifai<sup>3</sup>, Kairos<sup>4</sup>). Big Tech companies have the advantage of owning massive web infrastructures. Many of them even offer their own cloud services – Amazon, Google, Microsoft and IBM respectively own Amazon Web Services (AWS)<sup>5</sup>, Google Cloud<sup>6</sup>, Microsoft Azure<sup>7</sup> and IBM Cloud<sup>8</sup>. Cloud platforms are a huge advantage for algorithmic vendors as they provide them with resources such as elastic computing, key management and storage options. Those resources give Big Tech competitors a considerable boost in the “Algorithmic Economy” and, indeed, all of the cloud platforms listed above offer facial analysis services (as well as a variety of other cognitive services). Meanwhile, small companies and start-ups try to disrupt the market by offering niche functionalities or specialising in a specific service.

Facial analysis algorithms are normally offered over an API. End users send a request to the vendor

---

<sup>1</sup><https://blogs.gartner.com/smarterwithgartner/the-algorithm-economy-will-start-a-huge-wave-of-innovation/>

<sup>2</sup><https://www.faceplusplus.com/>

<sup>3</sup><https://www.clarifai.com/>

<sup>4</sup><https://www.kairos.com/>

<sup>5</sup><https://aws.amazon.com/>

<sup>6</sup><https://cloud.google.com/>

<sup>7</sup><https://azure.microsoft.com/en-gb/>

<sup>8</sup><https://www.ibm.com/uk-en/cloud/products>

over an Internet connection specifying the image to be analysed as well as optional preferences. The vendor then returns the output of the algorithm, whether it is the result of demographic feature extraction, detecting a human face or recognising a specific subject. Users authenticate their requests with a private API key and are charged either monthly (as a subscription) or depending on their usage of the algorithm (the more popular “*pay-as-you-go*” pricing model).

It is important to note that during this entire process, the internal proceedings of the algorithm remain perfectly hidden from the end users. To the developer or researcher using the API, the algorithm is nothing more than a “*black box*”, which receives input and produces output. This arrangement benefits algorithm vendors as it hides the implementation details of their solutions which is apparently in their best financial interest.

However, this model poses a security problem from the viewpoint of the user. As Ken Thompson said in his famous paper “*Reflections on Trusting Trust*” – “*You can’t trust code that you did not totally create yourself*” [9]. However, the business model of the “Algorithm Economy” means users have to blindly rely on the outputs of the proprietary algorithms as they have no access to the underlying design.

Furthermore, P. Barlas et al. argue that cognitive services in the “Algorithmic Economy” are an “*opaque technology*” not only to end users, but also to vendors [7]. That owes to the fact that cognitive services are overwhelmingly based on the neural network technology. While neural networks are particularly powerful at replicating certain sensory functions of the brain, they suffer from low explainability [10] – there is no straightforward way to interpret their results as they provide no justification for their outputs.

Thus, the “Algorithm Economy” can be seen as a phenomenon driving technological advance, but also as a threat to security. While the business model encourages competition and stimulates innovation by providing a financial incentive, it comes at the expense of transparency and explainability. The opaque nature of the “Algorithm Economy” means that facial analysis systems are often deployed without considerations of potential inaccuracies or biases. And the consequences of errors in those systems can be dire, as we will show in the next section.

## 2.2 Algorithmic Fairness

*Algorithmic fairness* is an emerging topic in machine learning and artificial intelligence. While there is no consensus in the scientific community about what exactly constitutes fairness [11], a *fair algorithm* or *unbiased algorithm* is broadly defined as one that does not discriminate against people of certain demographic or socio-economic groups. In contrast, an *unfair* algorithm or a *biased* algorithm is one whose output systematically harms or benefits a certain group. Examples of algorithmic biases have been receiving increasing media coverage as criminal risk assessment algorithms have been shown to discriminate against people of colour<sup>9</sup>, Apple Card has been labelled

---

<sup>9</sup><https://www.technologyreview.com/s/612775/algorithms-criminal-justice-ai/>

“sexist” by some users<sup>10</sup> and Amazon’s recruitment algorithm has been claimed to prefer male over female candidates<sup>11</sup>.

For the unprivileged social groups, algorithmic biases can have a detrimental effect on career prospect, financial risk assessment or even the right of a fair trial. Mechanisms and regulations are needed to ensure algorithmic fairness and prevent “*reinforcing the discriminatory practices in our society*” [7]. The need for fairer algorithms has given rise to a number of initiatives by non-government organisations and Big Tech. The Algorithmic Justice League has described itself as a “*movement towards equitable and accountable AI*”<sup>12</sup> and its members have contributed to some of the pioneering research in the field [8][12]. Google ML Fairness<sup>13</sup> aims to raise awareness about algorithmic fairness among developers, and IBM AI Fairness 360<sup>14</sup> offers a free open-source toolkit for identifying and mitigating biases in algorithms [13].

A lot of research has been aimed to devise strategies to mitigate algorithmic bias. The most naïve solution is to simply exclude *sensitive attributes* (such as race, gender, religion) from the training data. Gajane and Pechenizkiy refer to this as “*fairness through unawareness*” [11]. This does not eliminate the possibility of bias, though, since sensitive (or also *protected*) attributes can be inferred by algorithms via *non-protected* attributes [14] (such as nationality, occupation, address registration, etc.). Instead, a multitude of more complex bias mitigation approaches have been proposed including interventions at pre-processing level [15], introducing fairness-enhancing models [16][17] and reweighing strategies [13].

The problem of algorithmic bias extends to facial analysis as well. Back in 2015, Google made the news by misclassifying two Black people as gorillas<sup>15</sup>. More recently, a study titled “*Gender Shades*” [8] has demonstrated that popular facial analysis tools misclassify the gender of Black women up to 34.7% of the time, and White males in no more than 0.8% of the cases. The findings of the research were presented at a hearing at the American Committee on Oversight and Reform, which concluded that “*facial recognition technology misidentifies women and minorities at a much higher rate than white males, increasing the risk of racial and gender bias*”<sup>16</sup>.

## 2.3 Related Work

The growing awareness about the importance of algorithmic fairness has prompted a lot of research in the field. However, while algorithmic fairness is a fast-growing research area in computer science, biases in facial analysis tools, in particular, have not been thoroughly explored.

The “Gender Shades” study [8] which we already introduced in the last section, was the founda-

---

<sup>10</sup><https://www.nytimes.com/2019/11/10/business/Apple-credit-card-investigation.html>

<sup>11</sup><https://www.reuters.com/article/us-amazon-com-jobs-automation-insight/amazon-scraps-secret-ai-recruiting-tool-that-showed-bias-against-women-idUSKCN1MK08G>

<sup>12</sup><https://www.ajlunited.org/about>

<sup>13</sup><https://developers.google.com/machine-learning/fairness-overview>

<sup>14</sup><https://aif360.mybluemix.net/>

<sup>15</sup><https://www.theverge.com/2015/7/1/8880363/google-apologizes-photos-app-tags-two-black-people-gorillas>

<sup>16</sup><https://oversight.house.gov/legislation/hearings/facial-recognition-technology-part-1-its-impact-on-our-civil-rights-and>

tional work, which revealed the presence of biases in facial analysis tools. The paper tests the gender classification performance of three commercial facial analysis APIs (Microsoft, Face++ and IBM) and analyses their misclassification rates on lighter and darker-skinned males and females. As mentioned previously, the paper concludes that the algorithms tend to misclassify women of colour up to 30 times more often than White males.

“Gender Shades” also suggests that algorithmic bias in facial analysis is rooted in the usage of highly unbalanced training sets. The study shows that two of the most popular benchmarks used for evaluating facial analysis performance – *Adience*<sup>17</sup> and *IJB-A*<sup>18</sup>, consisted of a predominantly White, male population. For example, women of colour constitute only 4.4% of IJB-A, while the White male population makes up as much as 59.4% of the same dataset. Because of this, biases against minorities can easily remain unnoticed since misclassifying the tiny proportion of minorities in the dataset only has a negligible effect on the overall quoted accuracy.

A subsequent paper called “*Actionable Auditing*”[12] was produced shortly after “Gender Shades”. The follow-up study found that biases in the three APIs “*audited*” by “Gender Shades” have been considerably mitigated just seven months after the publication of the original paper. The study makes a case for regular investigations into algorithmic performance, which could stimulate API providers to enhance the fairness of their algorithms.

In line with the general strategy of “Gender Shades” and “Actionable Auditing”, we will be performing an “algorithmic audit” on popular facial analysis tools. We build on the work of those studies by examining a wider set of feature extraction tasks and datasets, as well as conducting a more detailed statistical analysis on the results.

---

<sup>17</sup><https://talhassner.github.io/home/projects/Adience/Adience-data.html>

<sup>18</sup><https://www.nist.gov/itl/iad/image-group/ijb-dataset-request-form>

## Chapter 3

# Data Engineering

### 3.1 Overview

During the data engineering stage of the project, we run a selection of facial analysis APIs on a number of datasets to obtain their outputs across different feature extraction tasks. Prior to the data engineering experiment itself, we conduct **preliminary research** (Section 3.2) where we make our selection of APIs, datasets, as well as attributes of interest.

The data engineering experiment itself is illustrated by Figure 3.1 below. After undergoing initial **data preparation** (Section 3.3), the datasets have been cleaned and uploaded to AWS as S3 buckets<sup>1</sup>. Each dataset is presented as an input to each of the APIs (1) which process the image data and store their outputs as logs - JSON or .txt files (2). The relevant information from the logs is then extracted and stored as multiple .csv files – one for each dataset-API pair (3). This concludes the **API processing** component of the experiment.

In order to benchmark algorithmic accuracy and bias, we need ground truth relating to the attributes of the examined subjects. While for some datasets ground truth is provided by the owner, it is missing or incomplete for others. That is why we design a **crowdworking experiment** where we present the images from those incomplete datasets to crowdworkers (4) using the Figure Eight crowdworking platform. The platform provides us with the output of the crowdworkers (5), which we use to complete our ground truth.

Finally, the multiple files obtained from API processing and the crowdworking experiment are standardised and combined into a single .csv file (6), (7) containing complete ground truth about each entry in our datasets, as well as the associated data extracted by the APIs.

---

<sup>1</sup>S3 (Simple Storage Service) is Amazon Web Services' storage service: <https://aws.amazon.com/s3/>. S3 storage is organised into "*buckets*".

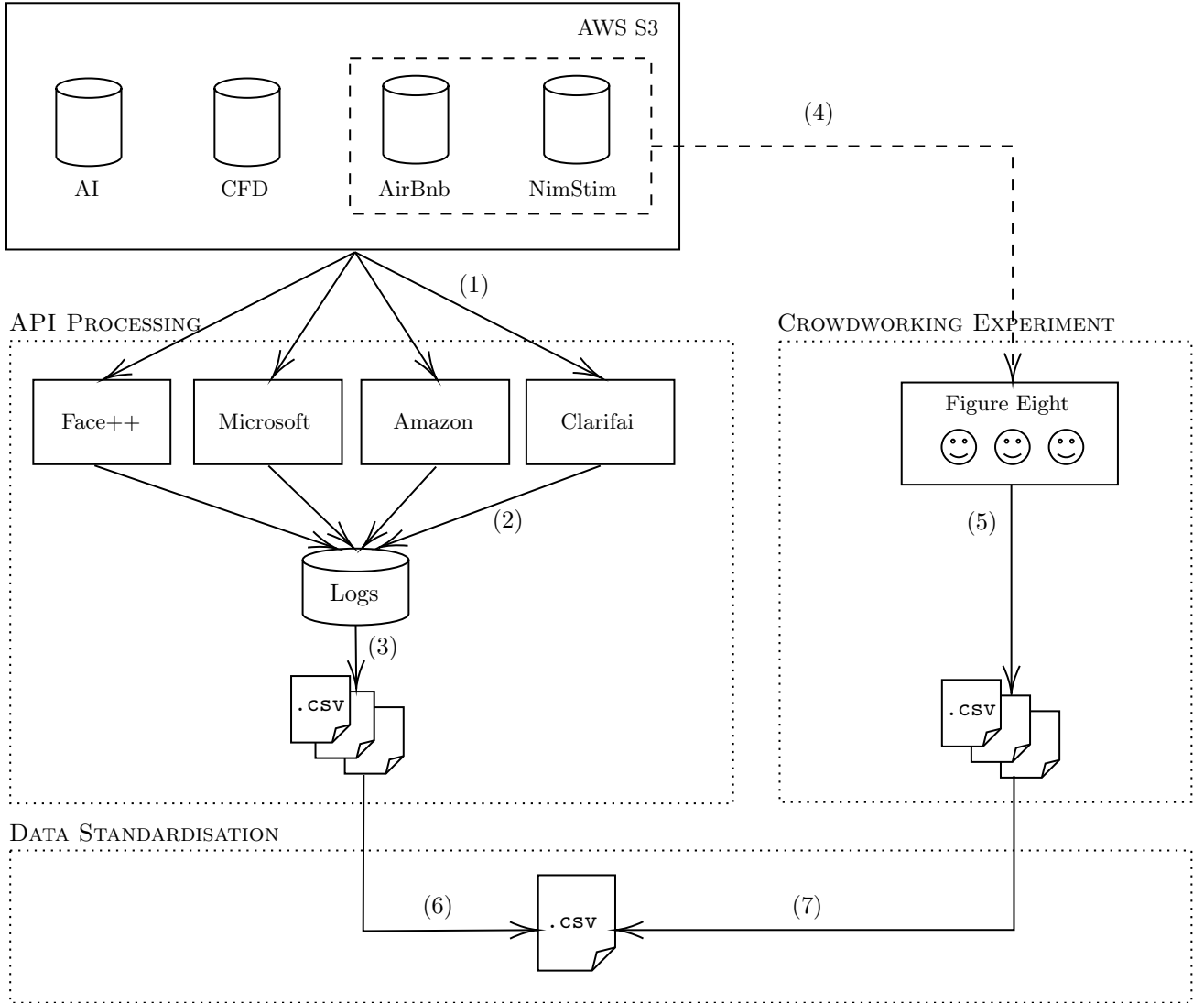


Figure 3.1: High-level overview of the data engineering process.

## 3.2 Preliminary Research

Prior to the data engineering process itself, preliminary research was needed to identify a few important design decisions, namely:

- **Choice of attributes** – what set of attributes will we be seeking to extract? Depending on the tools we select, there might be limitations on that since every tool performs a different set of classification tasks.
- **Choice of APIs** – which APIs or algorithms are we going to examine? What criteria do we pose on the APIs we decide to test and do they collectively provide the features we would like to extract?

- **Choice of Datasets** – what datasets will we be processing through the APIs? Are the datasets standardised or unstandardised, labelled or unlabelled?

Those choices are described and justified in the next three subsections.

### 3.2.1 Choice of Attributes

When selecting what attributes to extract via the APIs, we based our choice on two main criteria. Firstly, we wanted to focus on demographic features since there seems to be a growing sensitivity about this category of features, especially with the introduction of new data privacy regulations. And secondly, the attributes should preferably be classified by multiple APIs so that we can provide a comparison between tools. Taking those into account, we opted to consider four attributes: **gender**, **race**, **age** and **emotion**.

**Gender** and **age** are key demographic attributes, and the extraction of those features is fundamental to a lot of cognitive systems in security, surveillance and marketing. While biases in gender classification have been previously explored [1][8], there has been no study into potential biases of age extraction to the best of our knowledge. Both of those attributes are handled by most commercial facial analysis APIs.

**Race** extraction is only offered by a small set of facial analysis tools. However, it falls into GDPR’s category of *sensitive personal data*<sup>2</sup>, which is why it will be particularly interesting to analyse whether it has an effect on misclassification. As for **emotion** – while it is not a demographic or a sensitive feature, automated emotion analysis (also affect analysis) is becoming an increasingly important capability of facial analysis [5]. Just like age, the extraction of neither of those attributes seems to have been studied before in the context of algorithmic bias. We hope that by introducing a wider range of attributes, the project will be able to expand previous work on the topic.

### 3.2.2 Choice of APIs

For our choice of APIs, we considered seven commercial facial analysis tools – Amazon Rekognition<sup>3</sup>, Google Cloud Vision<sup>4</sup>, Microsoft Face API<sup>5</sup>, IBM Watson Visual<sup>6</sup>, Face++<sup>7</sup>, Kairos<sup>8</sup> and Clarifai<sup>9</sup>. Upon inspection, not all of those were relevant to our study – for example, Google Cloud Vision does not extract any demographic features, which is the majority of the features we would like to explore. For our project, we decided to study two Big Tech APIs – **Amazon Rekognition** and **Microsoft Face API**, and two of the smaller API providers – **Face++** and **Clarifai**. We

<sup>2</sup><https://ico.org.uk/for-organisations/guide-to-data-protection/guide-to-the-general-data-protection-regulation-gdpr/lawful-basis-for-processing/special-category-data/>

<sup>3</sup><https://aws.amazon.com/rekognition/>

<sup>4</sup><https://cloud.google.com/vision>

<sup>5</sup><https://azure.microsoft.com/en-gb/services/cognitive-services/face/>

<sup>6</sup><https://cloud.ibm.com/apidocs/visual-recognition/visual-recognition-v3>

<sup>7</sup><https://console.faceplusplus.com/documents/5679127>

<sup>8</sup><https://www.kairos.com/docs/api/>

<sup>9</sup><https://www.clarifai.com/models/demographics-image-recognition-model-c0c0ac362b03416da06ab3fa36fb58e3>

API	Age	Race	Gender	Emotion
Face++	Integer	N/A	$\in \{\text{'male'}, \text{'female'}\}$	$\in \{\text{'sadness'}, \text{'neutral'}, \text{'disgust'}, \text{'anger'}, \text{'surprise'}, \text{'fear'}, \text{'happiness'}\}$
Clarifai	List of tuples (Integer, Probability)	List of tuples (Race, Probability)	List of tuples ( $\{\text{'masculine'}, \text{'feminine'}\}$ , Probability)	N/A
Microsoft	Integer	N/A	$\in \{\text{'male'}, \text{'female'}\}$	$\in \{\text{'anger'}, \text{'contempt'}, \text{'disgust'}, \text{'fear'}, \text{'happiness'}, \text{'neutral'}, \text{'sadness'}, \text{'surprise'}\}$
Amazon	Low and high boundary - both Integers.	N/A	$\in \{\text{'male'}, \text{'female'}\}$	$\in \{\text{'happy'}, \text{'sad'}, \text{'angry'}, \text{'confused'}, \text{'disgusted'}, \text{'surprised'}, \text{'calm'}, \text{'unknown'}, \text{'fear'}\}$

Table 3.1: Output of facial analysis tools. Probability denotes a float in the range [0,1].

picked those on the basis of popularity (judging by previous studies of facial analysis [7][8][12] and Internet articles), and the variety of extracted features.

Amazon Rekognition and Microsoft Face API are products of the well-known tech giants Amazon and Microsoft. They are offered as part of their respective cloud platforms Amazon Web Services (AWS) and Microsoft Azure. Both companies promote the high performance of their products with Microsoft recently claiming to have mitigated racial biases in its API<sup>10</sup>.

Face++ is a product of the China-based AI company Megvii<sup>11</sup>, which is thought to be the world’s “biggest provider of third-party authentication software”<sup>12</sup>. Lately, Face++ has been surrounded by controversy regarding its connections with the Chinese government<sup>13</sup> and the use of its software in surveillance systems<sup>14</sup>. Just like Face++, Clarifai is another challenger in the dynamic facial analysis market. The New-York startup has been involved in military projects with the American Department of Defense<sup>15</sup> and describes itself as an “accurate and unbiased technology”<sup>16</sup>.

We explored the API references of the four tools and ran quick experiments with them to verify that their output matches the official documentation. Table 3.1 shows the set of attributes provided by each API as well as the signature of the output. Notice that Clarifai is the only tool offering race extraction. At the time of selection of APIs, Face++ also used to return race, but then disabled this functionality<sup>17</sup> shortly after the start of the project.

<sup>10</sup><https://blogs.microsoft.com/ai/gender-skin-tone-facial-recognition-improvement/>

<sup>11</sup><https://megvii.com/en>

<sup>12</sup><https://www.scmp.com/tech/article/3012103/rising-chinese-ai-star-megvii-gets-caught-us-china-tech-war>

<sup>13</sup><https://www.bloomberg.com/news/articles/2019-05-24/trump-s-latest-china-target-includes-a-rising-star-in-ai>

<sup>14</sup><https://www.scmp.com/tech/start-ups/article/3013229/ai-unicorn-megvii-not-behind-app-used-surveillance-xinjiang-says>

<sup>15</sup><https://www.clarifai.com/blog/why-were-part-of-project-maven>

<sup>16</sup><https://www.clarifai.com/blog/how-clarifai-builds-accurate-and-unbiased-ai-technology>

<sup>17</sup><https://console.faceplusplus.com/documents/5679127>



### 3.2.3 Choice of Datasets

For the choice of datasets, we need our selection to fulfil two key criteria. *Firstly*, the datasets should provide a mixture between *standardised* and *unstandardised* data. Standardised images are taken in a controlled environment – they consist of uniform-sized portrait images with one individual facing the camera in each entry. Lighting is usually adjusted, and the background is neutral. In contrast, unstandardised images can depict any number of people, with faces possibly not visible or obfuscated. This way, we can explore the effect of different image types on performance (which is a goal we outline in Section 1.2). And *secondly*, our datasets need to provide us with a reasonably balanced and diverse distribution of the attributes we are about to explore (age, gender, race and emotion).

The second requirement means it would not be appropriate to use popular facial analysis benchmarks such as Adience and IJB-A which (as we discussed in Section 2.3) have been shown by the “Gender Shades” study to be severely imbalanced in favour of lighter and male subjects. The systematic imbalance in facial analysis benchmarks is an obstacle to training fair algorithms and attempts have been made to produce demographically balanced sets of human faces. Back in 2018, IBM announced the development of IBM Diversity in Faces Dataset – a fully annotated dataset of 1 million images “*equally distributed across skin tones, genders, and ages*”<sup>18</sup>. We submitted a request for access to the dataset but received no response. The project’s dedicated webpage now redirects to another page, and IBM has not commented on it since 2019, which leads us to believe that the project has unfortunately been abandoned.

Since most publicly available datasets are *individually* not diverse enough, for our study we consider a combination of datasets which *collectively* should provide us with ample diversity across age, gender, race and emotion. Having laid out the considerations above, the data we use throughout the project consists of the four datasets that we introduce below:

- **Chicago Face Database (CFD)** [19] is a set of standardised images originally aimed for psychological research into face stimuli. Since its original release in 2015, CFD has been updated over several iterations, with the most recent version containing 1207 images of 597 unique subjects (mostly young adults) expressing various emotions. The dataset has been created with diversity in mind and contains a balanced representation of White, Black, Asian and Latino individuals as well as males and females. The images in the set are of uniform size and high resolution, and subjects have been photographed wearing identical clothing and against a neutral white background. CFD provides metadata about the individuals’ race, gender and emotion, as well as comprehensive norming data including estimates about age extracted from tens of raters.
- **NimStim** [20] is another standardised dataset originally designed for psychological research. It consists of 673 entries of 43 unique subjects each of which expressing emotions such as happy, sad, fearful, etc. While the dataset consists of relatively few individuals (most of which are young actors in their 20s), the set is partially annotated, diverse in terms of both

---

<sup>18</sup><https://www.ibm.com/blogs/research/2018/06/ai-facial-analytics/>

race and gender, and provides a wide spectrum of emotions, which will allow us to test emotion extraction across tools better.

- The **AirBnb** dataset is a set of profile pictures of hosts in AirBnb’s<sup>19</sup> rental platform. Images in the dataset have been scraped by previous research into the platform’s pairing dynamics [6]. Including the AirBnb dataset into the project will allow us to study instances of unstandardised images.
- The last dataset we include is a set of **AI-generated faces** (which we will be referring to as the AI dataset). These types of images are increasingly popular in media and advertisement<sup>20</sup> and one provider of such images is Generated Photos<sup>21</sup>. The AI-generated faces used to be provided freely (as a huge folder of images) by Generated Photos at the beginning of the project. However, Generated Photos changed their service model shortly after that and now generates images on request via a REST API<sup>22</sup>, which means that we had to *generate* our own AI dataset by making a series of requests (we describe the API and the generation of the dataset in Section 3.3 Data Preparation). We include AI-generated faces in our project as it allows us to study the behaviour of the APIs on young children and teenagers (who would normally fall into a protected category) and elderly people (which are underrepresented in all of the previous three datasets).

### 3.3 Data Preparation

Before launching the data engineering experiment, some amount of data preparation was necessary. Some of the data preparation involved manual cleaning and inspection of the data. For example, a few of the images in the CFD dataset exceeded Face++’s restriction of 2MB per image<sup>23</sup> and those needed to be resized. Meanwhile, some of NimStim’s files contained typos in their names (such as an extra dot, or a missing underscore) – those needed to be manually corrected as the file names would later be automatically parsed to obtain the metadata that they contain. Apart from that, the NimStim dataset also consisted of several file extensions including `.BMP` and `.TIFF` files, which are not supported by all of the APIs. All NimStim images were converted to `.JPEG` by the `convert_to_jpeg.py` script (Listing A.1).

AirBnb’s dataset was too large (especially compared to the sizes of NimStim and CFD), so we decided to only use a random sample of it. We selected the samples from the data collected in Hong Kong and Chicago since those provided the highest racial diversity according to V. Koh [6]. The sampling script in A.2 randomly selects 500 images from each of the two cities and copies them locally to produce our sample.

As for the AI dataset, we already mentioned that Generated Photos does not provide the AI-

---

<sup>19</sup><https://www.airbnb.co.uk/>

<sup>20</sup><https://www.theverge.com/tldr/2019/2/15/18226005/ai-generated-fake-people-portraits-thispersondoesnotexist-stylegan>

<sup>21</sup><https://generated.photos>

<sup>22</sup><https://generated.photos/api>

<sup>23</sup><https://console.faceplusplus.com/documents/5679127>

generated faces at a single repository, so we had to generate the set by making multiple requests using their API's free-tier. The functionality of the API is illustrated in Figure 3.2: The user provides a request for AI-generated faces providing (optionally) the required race, gender and age of the subjects. The API's response is a JSON file which contains the URLs of the faces generated by the API.

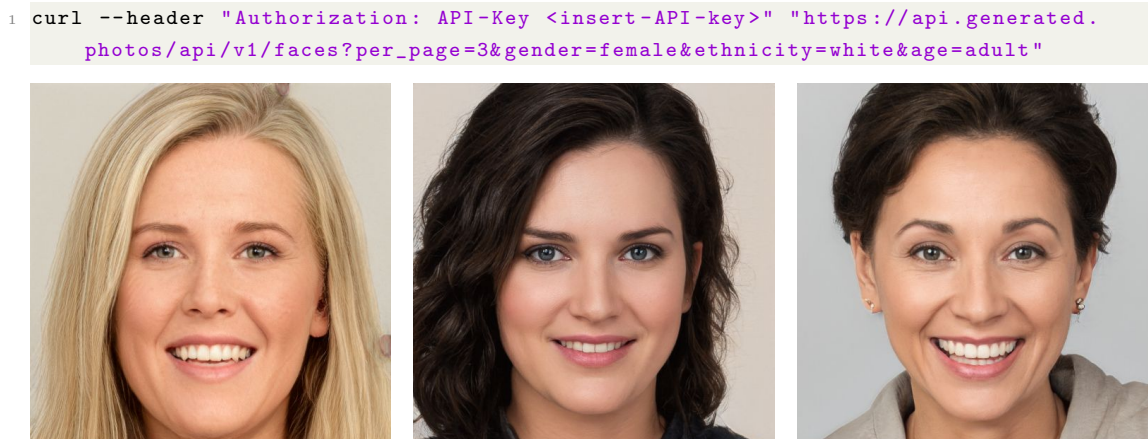


Figure 3.2: Demonstration of the AI face-generation API. The response to the request above returns URLs pointing to the three images below.

The AI-generation API provides 4 categories for race (White, Latino, Asian and Black), 5 for age (infant, child, young-adult, adult and elderly) and 2 for gender (male, female). To keep the resulting AI set racially diverse, we aimed to extract as many subjects as possible from each of the 40 intersectional demographic categories. For that purpose, the `generate_urls.py` script provided as Listing A.3 makes requests for each category and logs the responses as JSON files. As the API was at its early days at the beginning of the project, we only managed to generate 248 unique faces. However, this is still comparable to the number of unique individuals in other datasets, and the set is quite diverse across all demographic categories as can be seen in Table 3.2. Subsequently, the URLs were extracted from the JSON into a .csv table by the `extract_urls.py` script (A.4) and finally downloaded locally by `download_urls.py` (A.5).

This completes the data preparation stage. Table 3.3 summarises the datasets we will be processing in the next stage. All datasets were uploaded to AWS as S3 buckets. This is needed since most APIs require images to be provided as public URLs pointing to the corresponding images. The choice of AWS S3 as a hosting platform is also convenient since Amazon Rekognition accepts image input only as a reference to an S3 object or a byte array<sup>24</sup>.

<sup>24</sup><https://aws.amazon.com/rekognition/faqs/>

Attribute	Category	Number of entries
Gender	Male	122
	Female	126
Race	Latino	60
	Asian	46
	White	100
	Black	42
Age	Infant	20
	Child	62
	Young-adult	80
	Adult	66
	Elderly	20

Table 3.2: Distribution of demographic categories in the AI dataset. Numbers obtained by the `find_distribution.py` script available in the project’s Git repository.

Dataset	Number of entries	Unique individuals	Type of images	Annotated
CFD	1207	597	Standardised	Yes
NimStim	673	43	Standardised	Partially
AirBnb	1000	1000	Unstandardised	No
AI	248	248	Standardised	Yes

Table 3.3: Summary of the four datasets. Note that not all 1000 entries from AirBnb are used in the analysis since a lot of them fail to be detected by the APIs (*i.e. the APIs failed to identify a human face in the image*).

### 3.4 API Processing

During the API processing stage, we use our four cleaned datasets as inputs for the facial analysis APIs. All of the four facial analysis tools provide programming interfaces in the form of Python modules<sup>25 26 27 28</sup> and return the response as a JSON file similar to the Face++ example provided in Figure 3.3 – tools receive input data in the form of an image URL, and extract various demographic features. Each API is called in a different way and requires a different setup, set of arguments or other adjustments. That is why, to automate processing, a separate set of scripts<sup>29</sup> is provided for each of the four APIs. Those scripts automatically process all images across the datasets through the APIs and dump the responses as logs.

<sup>25</sup><https://sdk.clarifai.com/python/docs/latest/tutorial.html>

<sup>26</sup><https://pypi.org/project/python-facepp/>

<sup>27</sup><https://boto3.amazonaws.com/v1/documentation/api/latest/reference/services/rekognition.html>

<sup>28</sup><https://docs.microsoft.com/en-us/dotnet/api/microsoft.azure.cognitiveservices.vision.face.faceclient>

<sup>29</sup>All API processing scripts are available in the project’s Git repository – their location is specified in the code listing section of the Appendix

```

1 "attributes":{
2   "gender":{
3     "value":"Female"
4   },
5   "age":{
6     "value":28
7   },
8   "smile":{
9     "value":100.0,
10    "threshold":50.0
11  },
12  "emotion":{
13    "anger":0.0,
14    "disgust":0.0,
15    "fear":0.0,
16    "happiness":100.0,
17    "neutral":0.0,
18    "sadness":0.0,
19    "surprise":0.0
20  },
21  "beauty":{
22    "male_score":45.807,
23    "female_score":50.939
24  }
25 }

```



Figure 3.3: An entry from the AI dataset (*right*) and part of the corresponding Face++ JSON response (*left*).

However, only parts of the original JSON responses are useful to us. Responses normally contain a lot of redundant metadata and information that we do not need for the purposes of this project. Besides, every API adopts a different convention of constructing the JSON response. That is why, after responses have been logged, another set of scripts extracts the relevant information from the logs, as well as the corresponding metadata about the original image entry, and saves those as .csv tables – one for each dataset-API pair, which completes the API processing stage.

It is worth noting that while facial analysis tools successfully detect faces in all standardised images, they sometimes fail to do so on the unstandardised data - i.e. the AirBnb dataset. Detecting faces in the AirBnb dataset is not always straightforward since many of the profile pictures in it contain obfuscated faces, people not facing the camera or, in many cases, pictures of pets. While face detection performance is not the subject of this study, Table 3.4 shows that the face detection success rates for AirBnb varies across APIs.

API	Microsoft	Face++	Clarifai	Amazon
Successful	764	849	820	894
Failed	236	151	180	106

Table 3.4: Number of successfully and unsuccessfully processed AirBnb entries across APIs. Failed processing means no face was detected in the image.

### 3.5 Crowdsourcing Experiment

Crowdsourcing (also known as crowdsourcing) platforms provide a mechanism of having manual tasks executed by humans (often referred to as crowdworkers, raters or contributors). Tasks are usually hard to automate – for example labelling objects in an image, classifying a customer’s review as a negative or positive, etc. Crowdworkers perform the required tasks for which they get paid depending on the amount of work completed or on their performance (the so-called “*Performance-Based Payments*”) [21].

We make use of crowdsourcing to complete our ground truth about the AirBnb and NimStim datasets. We ask crowdworkers to estimate the demographic features and emotion of the subjects in those sets, which should provide us with sufficient metadata to conduct the data analysis. For our crowdsourcing experiment, we considered the services of two popular commercial crowdsourcing platforms – Amazon Mechanical Turk<sup>30</sup> and Figure Eight<sup>31</sup> (formerly known as CrowdFlower). We decided to use Figure Eight, as it seems to provide more control over the visual representation of the tasks, as well as a more detailed breakdown of results<sup>32</sup>.

We estimated the financial cost of the experiment<sup>33</sup>, and made sure our data is formatted into the appropriate layout required by Figure Eight<sup>34</sup>. We then designed two crowdsourcing experiments – one for NimStim, and one for AirBnb.

The NimStim dataset lacked attributes for gender and age. We asked crowdworkers to classify each of the subjects in the images as “*Male*”, “*Female*” or “*I am not sure*”, and to provide an estimate of the subject’s age in years. For AirBnb, there is the additional complication that images might contain multiple faces. To work around this, we first ask raters whether the image contains more than one face. If they think it does not, then they are asked to provide their judgement about the subject’s gender, age, race and emotion. A minimum of three raters labelled each image. Where the majority of raters determined the image contains multiple faces, the image was subsequently discarded from the dataset. The complete design of the experiment is provided in Appendix B.

To track the quality of crowdworkers’ work, we place what are known as “*gold*” [22] entries or “*attention checks*” [7]. “Gold” entries are images that we have manually labelled, and that we

<sup>30</sup><https://www.mturk.com/>

<sup>31</sup><https://www.figure-eight.com/>

<sup>32</sup><https://www.figure-eight.com/platform/how-figure-eight-works/>

<sup>33</sup><https://success.figure-eight.com/hc/en-us/articles/202703165-Job-Costs-FAQ>

<sup>34</sup><https://success.figure-eight.com/hc/en-us/articles/202703175>

believe should be straightforward to be classified correctly by a concentrated crowdworker. Those include unambiguous questions such as answering “Yes” to “*Does the picture contain multiple faces?*” (where the picture clearly depicts more than one face). Figure Eight’s policy dictates that contributors who perform poorly on “gold” data are automatically removed<sup>35</sup>. The final experiment report provided by Figure Eight shows that crowdworkers have performed well on “gold” data and that they are located in a wide range of countries. The latter is illustrated by Figure 3.4 and is particularly important for our study – it means that crowdworkers are likely to be of different cultural and racial backgrounds, minimising the potential biases in their judgements.

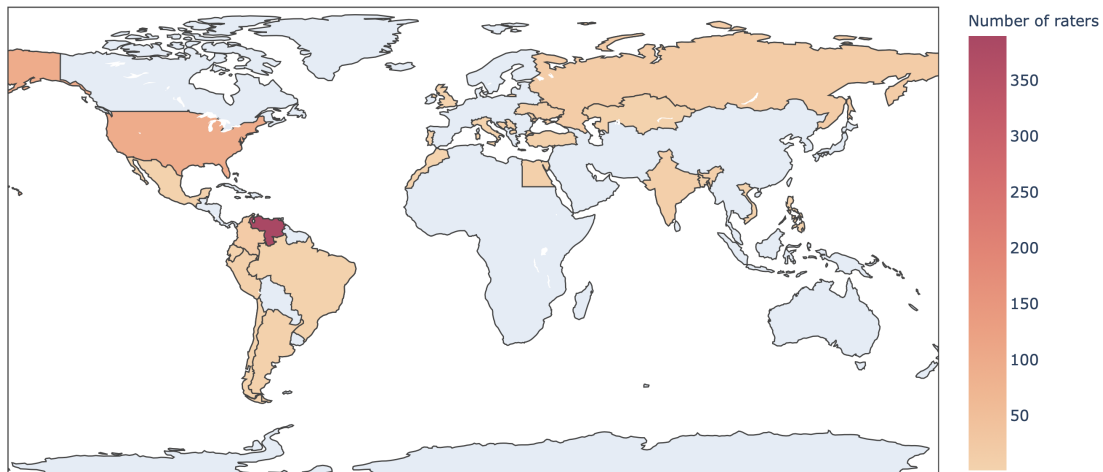


Figure 3.4: Geographical distribution of Figure Eight crowdworkers in both of our experiments.

### 3.6 Data Standardisation

During the data standardisation stage, we subject the data collected from APIs and crowdworkers to a round of transformations aimed to make it ready for analysis. This requires combining the different data sources obtained from API processing and the crowdworking experiment into a single data matrix containing the ground truth and the processing results for each entry across the four datasets.

We start by combining the results from the API processing into a single table. We then proceed by cleaning the results from the Figure Eight experiment – we correct erroneous inputs by raters (such as “29YEARS” or “1998” for age) and dismiss entries which include multiple faces according to raters. Images are labelled by multiple crowdworkers and their judgements need to be reduced to a single value per attribute. For the cases of gender, race and emotion, we select the category selected by the majority of crowdworkers. Where there is no clear majority, we discard the entry from the analysis. For the case of age, we take an average of all the estimations. Finally, we merge the crowdworking results with the combined API processing table to complete the missing ground truth.

<sup>35</sup><https://success.figure-eight.com/hc/en-us/articles/202702985-How-to-Create-Test-Questions>

The resulting table contains both ground truth and API outputs. However, the representation of attributes is not uniform (some APIs denote male as “M”, and others as “male” or “masculine”) and is not numerical – values are represented as strings, which is not feasible for data analysis. This requires us to define appropriate numerical mappings for attributes.

For example, in the case of emotion, we saw that different APIs define different sets of possible emotions. We need to “cluster” those and find some correspondence between them. Table 3.5 shows the equivalences of emotions we identified in Face++, Microsoft and Amazon. Some emotions (such as “contempt” in Microsoft) have no counterparts in the other APIs, but for most of them, there is a clear analogue. Once we identify the corresponding categories in the metadata, we can assign each one a numerical value.

<b>Face++</b>	anger	N/A	disgust	fear	happiness	neutral	sadness	surprise	N/A
<b>Microsoft</b>	anger	contempt	disgust	fear	happiness	neutral	sadness	surprise	N/A
<b>Amazon</b>	angry	N/A	disgusted	fear	happy	calm	sad	surprised	confused

Table 3.5: Mapping of emotion categories of different APIs.

We can follow a similar strategy for gender and race. For age, the values are already numerical with the exception of the AI dataset where it comes in five categories - ‘infant’, ‘child’, ‘young-adult’, ‘adult’ and ‘elderly’. We assign a numerical age (number of years) to each category based on observing the images in the AI dataset and taking a cue from Erikson’s concept of “*stages of psychological development*” [23]. We define an infant to be 1-year-old, child – 10, young adult – 25, adult – 35, and elderly – 60-year-old. Subjective mappings like this one are inevitable in the data standardisation process, but we account for those throughout the research process and make sure they do not affect the conclusions of our analysis (this is discussed further in Chapter 6 Limitations and Future Work).

### 3.7 Final Dataset Distribution

To conclude the data engineering stage, we inspect the demographic distribution of the resulting dataset, to verify that we have produced a balanced set of human faces, that is diverse enough to let us examine algorithmic bias.

The set is almost perfectly balanced in terms of **gender** representation – the combined set includes 1319 images of male subjects 1357 images of female subjects, making up respectively 49.3% and 50.7% of the dataset.

Table 3.6 provides a breakdown of the distribution of the **race** attribute. While White subjects still dominate the dataset, the set still depicts a multitude of races and is reasonably balanced – especially compared to the Adience and IJB-A benchmarks where lighter-skinned faces contribute for respectively 86.2% and 79.6% of the population [8].



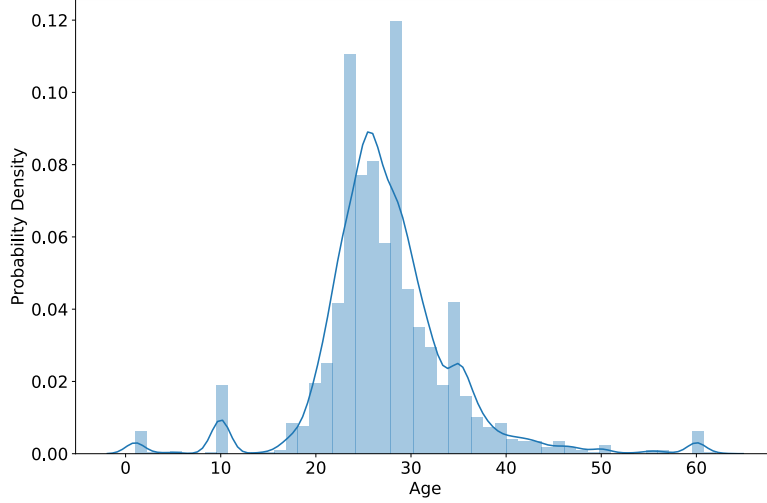


Figure 3.5: Age distribution in the final dataset.

Race	Count	Percentage
Asian	466	17.4%
Black	740	27.6%
Latino	211	7.9%
White	1256	46.9%
Others	4	0.1%

Table 3.6: Distribution of race in the final dataset.

Emotion	Count	Percentage
angry	241	9.9%
calm/neutral	954	39.3%
disgusted	82	3.4%
fearful	232	9.6%
happy	779	32.0%
sad	86	3.5%
surprised	51	2.1%
other	4	0.2%

Table 3.7: Distribution of emotion in the final dataset (excluding entries from the AI set, where no emotion is available).

As for **age**, the set’s distribution (Figure 3.5) seems to be dominated by young adults in their 20s and 30s. Still, because of the inclusion of the AI-generated images, we have an ample number of instances of infant and elderly faces.

Lastly, Table 3.7 illustrates the distribution of **emotions** in the dataset. As could be expected, most of the subjects have posed as happy or neutral in the photographs, but there is also an abundance of instances of other emotions such as fearful or angry.

## Chapter 4

# Methodology

### 4.1 Research Questions

In the data analysis stage of the project, we aim to extract insights from the data collected during data engineering. In this chapter we outline the methodology behind our analysis and we begin by formalising our project aims (listed back in Section 1.2) by wording them as *research questions* (RQs) which our study should address:

- **RQ1:** *“Does the performance of facial analysis vary across tools and datasets?”* – With this first research questions we address predictive performance rather than bias. We want to know if all facial analysis tools exhibit comparable classification accuracy, or if some tools are “better” than others (and if so, at which tasks). Additionally, we would be interested to know whether the choice of a dataset systematically affects performance - do tools tend to perform better on standardised or unstandardised data?
- **RQ2:** *“Is misclassification biased towards certain demographic groups?”* – Here we shift our focus to the topic of algorithmic bias. We want to know if the demographic profile of an individual (their gender, race and age) can make them more or less likely to be misclassified by facial analysis tools. By identifying biases, our approach should provide a sense of the direction (which groups are advantaged, and which disadvantaged) and significance (how strong is the evidence) of the bias.
- **RQ3:** *“Can misclassification be predicted?”* – We want to know whether the combined effect of a subject’s attributes can be used to make probabilistic statements about misclassification. That is, can a person’s demographic profile and emotion be used to predict if they are going to be misclassified or not?

## 4.2 Ethical Considerations

Our project requires the need of ethical assessment since it involves the processing of facial data and demographic attributes of individuals. We have concluded that our study does not pose a risk for the individuals whose images are analysed by the study. We do not share the contents of our datasets, and the statistics we quote in this report are only aggregate – no personal or identifiable data is disclosed. All the subjects involved in the study have approved of their images being used for scientific work and research. After the completion of the project, images will be deleted from all local and remote storage.

To comply with UCL’s ethical research policy, we have applied for an ethical approval through UCL’s Research Ethics Committee which granted us permission to conduct our study (Project ID 6725/001). In addition, all members involved in the project have passed the Ethics Training provided by the Computer Science department.

## 4.3 Accuracy Comparison

To answer **RQ1**, we compare the accuracy rates of the tools across the four datasets. We examine accuracy in the context of the *misclassification rate*, which we define intuitively as:

$$\frac{\text{number of misclassified entries}}{\text{total number of entries}} \quad (4.1)$$

However, to formalise this, we need to define what constitutes misclassification – this will have a different meaning in different classification (or feature extraction) tasks. For the case of extracting categorical features – race, gender and emotion, defining misclassification is relatively straightforward. We define the output of an API as misclassification if it does not match the value of the ground truth. To provide additional tolerance towards the tools, we do not consider the output as misclassification whenever the ground truth or the output itself is an undefined value – corresponding to “*unsure*” or “*other*”.

In the case of age classification, though, we identify two viable strategies to measure misclassification. One would be to compare the ground truth and the API output, and to define misclassification only in the case where the difference between the two exceeds some tolerance interval. Another one would be to measure the absolute difference between ground truth and API output. We decide to adopt the first strategy, since it allows us to express misclassification as a binary value, which will be useful for conducting the next stages in our analysis.

We thus need to define an appropriate threshold that will allow us to determine which differences between ground truth and algorithmic output are significant – i.e. constitute a misclassification, and which differences are acceptable – that is, the output is still a reasonable estimate of the actual age despite not matching the ground truth precisely. A simple policy would be to assign a constant threshold of, say, 10 years so that whenever the output predicts more than 10 years above

or below the actual value, we declare misclassification. However, there is a problem with applying a constant threshold to all ages. For example, classifying a 40-year-old individual as 50-year-old might be totally acceptable, while classifying a 3-year-old baby as a 13-year-old teenager is a clear algorithmic error which should count as misclassification.

Because individuals change very rapidly at a young age and not as much as they grow older, it makes sense to assign a smaller tolerance interval for young subjects, and a wider one for older subjects. We have therefore defined a “dynamic” tolerance interval for age classification that grows with age (shown in Listing A.7). For children under the age of 10, we set a tolerance of 5 years. For teenagers and young-adults (up to the age of 25), we allow for 10 years of difference in the classification output. And for subjects of over 25 years of age, we define a tolerance interval of 15 years.

The definition of the dynamic tolerance interval is designed to avoid penalising algorithms for insignificant or permissible errors and to be fair in grading their performance. We acknowledge that in this case, the choice of the tolerance interval is inevitably arbitrary – there is no way of objectively determining what constitutes misclassification and what not. However, we have repeated the experiment with multiple variations of our tolerance interval definition, and this has not changed the conclusions of our results.

With this definition of misclassification, we can proceed to obtaining and comparing the misclassification rates for different tools on different datasets across all feature extraction tasks. We present the results of the accuracy comparison in the next chapter.

## 4.4 Correlation Analysis

As we discussed back in Section 2.2 the computer science literature offers multiple definitions of algorithmic fairness. To identify biases in facial analysis tools and address **RQ2**, we need to define the notions of fairness and bias. For the purposes of our study, we will interpret biases as instances of *disparate mistreatment* - Zafar et al. define disparate mistreatment to be present when “*the misclassification rates differ for groups of people having different values of [a certain] sensitive attribute*” [24]. The idea is very similar to the concept of predictive parity or predictive rate parity [25] [26].

Therefore, we consider fairness in the context of equal misclassification rates for different demographics. If a particular subject has a demographic attribute  $D$  that can take on any value in a domain  $V$ , a target feature<sup>1</sup>  $Y$ , and the algorithm classifies the target feature as  $\hat{Y}$ , then we require that:

$$P(Y \neq \hat{Y} | D = d_i) = P(Y \neq \hat{Y} | D = d_j) \quad \forall d_i, d_j \in V \quad (4.2)$$

In this way, we require the misclassification rate (the probability  $P(Y \neq \hat{Y})$ ) to be the same, no matter the underlying demographic feature. Therefore, if our definition of fairness holds, there

---

<sup>1</sup>The target feature is the one we aim to obtain in the feature extraction task

should be no dependence between the subject’s demographic attribute and misclassification. To test the independence between those two variables, we employ statistical correlation tests. Some of the most popular correlation tests include Pearson, Spearman and Chi-Square. In our study, we make use of the *Spearman correlation test* since it makes no assumptions about the underlying distribution and captures any monotonic (and not just linear) correlation [27].

We conduct one-to-one correlation tests where we test the independence between each of the three demographic features (gender, age and race) and the misclassification<sup>2</sup> in each of the four feature extraction tasks. Note that we do not consider the effect of emotion on misclassification since, as specified in RQ2, we are only interested in biases associated with *demographic* attributes. We interpret the correlation coefficients as well as the p-values of the tests at 95% confidence level to determine the direction and the significance of potential biases.

## 4.5 Logistic Regression Analysis

While the correlation tests give us insights into the influence of individual demographic attributes on different types of misclassification, we still have not made a statement about the combined effect of those variables. In particular, we would like to know whether this combined effect is significant enough that it can be used to predict misclassification of an individual based on their demographic profile and emotion (**RQ3**).

For that purpose, we will train a logistic regression model on the four attributes contained in our ground truth (gender, race, age and emotion). For a given feature extraction task, we define an entry has been misclassified if *any* of the APIs has misclassified the entry. We create one model per classification task.

Evaluating the model, however, requires choosing a suitable metric. Accuracy is not relevant in this case since the dataset is highly imbalanced – misclassified entries usually make up about 15% of the dataset, so even a dummy classifier would achieve 85% accuracy by simply predicting the zero-class (zero indicating no misclassification).

For that reason, we will be evaluating the model on two main metrics – *balanced accuracy* and *recall*. Balanced accuracy is defined as:

$$\frac{\frac{TP}{P} + \frac{TN}{N}}{2} \quad (4.3)$$

where  $TP$  stands for *True Positives*,  $TN$  for *True Negatives*,  $P$  for total number of positives and  $N$  for total number of negatives. And recall is defined as:

$$\frac{TP}{TP + FN} \quad (4.4)$$

---

<sup>2</sup>During the technical implementation, misclassification is treated as a binary variable indicating correct classification (0) or misclassification (1).

where *FN* stands for *False Negatives*. We use recall in our model evaluation, since it provides us with a measure of what proportion of the actual misclassifications we have predicted. This is useful, since it is the prediction of misclassification for high-risk groups that we are interested in – predicting correct classification is not of prime importance. However, recall is hard to optimise against, since it is inversely related to precision [28] – minimising false negatives would likely lead to an unreasonable surge of false positives. That is why, we decide to optimise with respect to balanced accuracy, and we only use recall to track the performance of our model.

We implement the model using Python’s `scikit-learn` library. We start by splitting the overall data into 85% training data and 15% test data (the latter remains unseen until the final evaluation of the model). *One-hot encoding*<sup>3</sup> is applied to the categorical attributes of race and emotion. The one-hot representation ensures the attributes are interpreted by the model as independent categories and not as ordered values. We apply *feature scaling*<sup>4</sup> to all variables – this eases convergence and gives us a more sensible weight vector after fitting.

The model then undergoes *hyperparameter tuning* via *grid search*<sup>5</sup> on the *class weights*<sup>6</sup> – those determine the “penalty” for false positives and false negatives. During the entire process, validation is performed using 5-fold cross-validation with balanced accuracy as the scoring function<sup>7</sup>. Cross-validation prevents the model from overfitting and ensures low generalisation error [29].

Finally, the tuned model is trained on the training data. We analyse the weight vector obtained during training, and then run the model on the unseen test data. We compare the prediction it has produced with the actual misclassification values, calculate the relevant metrics and repeat the process for each type of feature extraction. The entire training and evaluation process is documented in the `stage_3` Jupyter Notebook in the project’s Git repository.

---

<sup>3</sup><https://machinelearningmastery.com/why-one-hot-encode-data-in-machine-learning/>

<sup>4</sup><https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html>

<sup>5</sup>[https://scikit-learn.org/stable/modules/generated/sklearn.linear\\_model.LogisticRegression.html](https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html)

<sup>6</sup>[https://scikit-learn.org/stable/modules/generated/sklearn.linear\\_model.LogisticRegression.html](https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html)

<sup>7</sup>[https://scikit-learn.org/stable/modules/model\\_evaluation.html#scoring-parameter](https://scikit-learn.org/stable/modules/model_evaluation.html#scoring-parameter)

# Chapter 5

## Results

### 5.1 Accuracy Comparison

Below we list the results of the accuracy comparison of tools. We consider each type of classification task separately and describe performance in terms of misclassification rate.

#### 5.1.1 Gender Classification

All of the facial analysis tools in our study provide gender classification, so for this feature extraction task we can provide a performance comparison between all of them. The number of misclassifications produced by each tool are described in Table 5.1 and the corresponding misclassification rates can be found in Tables 5.2.

We can notice a couple of interesting facts about the data. Firstly, the gap between the best gender classifier (Microsoft) and the worst one (Clarifai) is huge. This gap in performance becomes even more pronounced if we exclude the AI dataset from the calculations. Outside of the AI set, Microsoft has committed only 9 errors compared to the whopping 289 by Clarifai.

In fact, all APIs have their worst misclassification rates on the AI set. An inspection of the errors on the AI set shows that the vast majority of errors there have been made for individuals in the

	Clarifai	Microsoft	Amazon	Face++
CFD	152	3	36	82
NimStim	88	0	6	31
AirBnb	45	6	18	20
AI	49	38	39	35
<b>Total</b>	334	47	99	168

Table 5.1: Number of gender misclassifications per tool. *Total* indicates total number of errors.

	Clarifai	Microsoft	Amazon	Face++
CFD	12.5%	0.2%	2.9%	6.7%
NimStim	13.0%	0.0%	0.8%	4.6%
AirBnb	8.1%	1.0%	3.2%	3.6%
AI	19.7%	15.3%	15.7%	14.1%
<b>Total</b>	12.4%	1.7%	3.6%	6.2%

Table 5.2: Gender misclassification rates (in percentages) per tool. *Total* indicates the misclassification rate on the entire dataset.

“infant” or “child” age group. This makes sense since young children do not exhibit secondary sex characteristics such as facial hair, heavier skull for men or gender-specific face proportions. This makes determining the gender of children a considerably harder task and looking at some of the misclassified young subjects in the AI dataset (Figure 5.1), we can see that the gender of those faces can be challenging to classify even for humans, effectively pushing the APIs into arbitrary choices. For that reason, we will be ignoring the results on the AI set when analysing gender extraction performance and we exclude it from the correlation tests for gender classification.



Figure 5.1: Examples of artificially generated faces from the AI dataset, whose gender has been misclassified from the algorithms. All three subjects are male according to the face-generation API.

No matter whether we consider the AI data, though, we can see that the two Big Tech companies significantly outperform their smaller competitors Clarifai and Face++. As expected, each tool seems to achieve similar misclassification rates on each of the two standardised data sets (CFD and NimStim). Surprisingly though, Face++ and Clarifai actually perform better on the unstandardised AirBnb data despite faces in it being occasionally obfuscated or badly lighted. However, this might owe to the different distribution of demographics within AirBnb.

### 5.1.2 Race Classification

Of all the APIs we are analysing, Clarifai is the only one that provides race classification. Its performance is summarised in Table 5.3. Interestingly, we can see that the overall race misclassification rate for Clarifai is 14.3% which is only marginally higher than its misclassification rate for gender extraction (12.4%). This comes as a surprise since race classification is supposed to be



	CFD	NimStim	AirBnb	AI	Total
<b>Number of misclassifications</b>	168	52	99	65	384
<b>Misclassification rate</b>	13.9%	7.7%	18.0%	26.2%	14.3%

Table 5.3: Race extraction performance for Clarifai.

a significantly harder problem – while gender classification is a binary decision problem, race can be classified by Clarifai into as many as 7 racial categories. Clarifai therefore seems to be much better at race classification – its “specialism” since few API providers still offer race classification after the introduction of new data regulations defining race as protected data.

If we observe Clarifai’s performance by dataset, we can notice that while Clarifai performed gender classification better on unstandardised data, the API deals better on standardised data when it comes to race extraction. It is possible that obstacles associated with unstandardised data, such as bad lighting or low image resolution, could have a more severe impact on race classification, which requires, for example, detecting the skin tone of the subject.

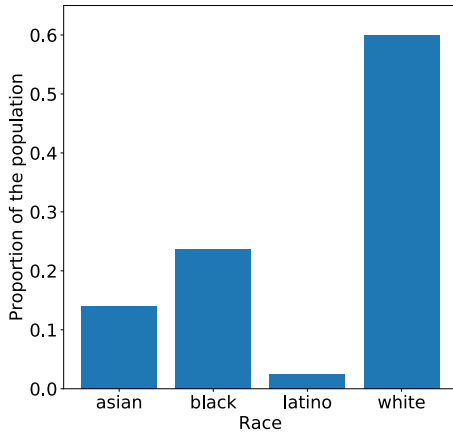


Figure 5.2: Race distribution in NimStim.

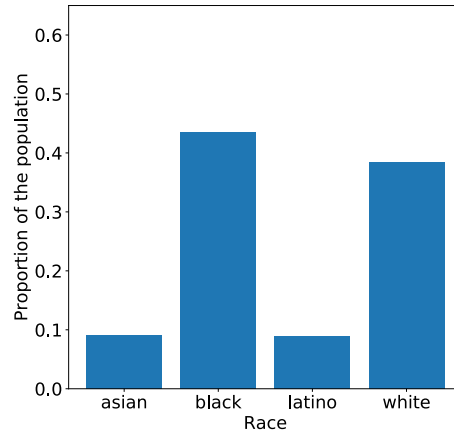


Figure 5.3: Race distribution in CFD.

Additionally, Clarifai seems to perform worse on NimStim compared to CFD although both are instances of standardised data. This could be attributed to the different race balance encountered in the two datasets – NimStim’s population (Figure 5.2) is less racially diverse than CFD (Figure 5.3) with White subjects dominating the set. The impact of race on race misclassification falls into the category of algorithmic bias, which we analyse through correlation testing as outlined in Section 4.4.

### 5.1.3 Age Classification

Similar to gender classification, all four APIs provide an age classification (or rather, age *estimation*) service. As discussed in Section 4.3, the results shown here are dependent on our tolerance

	Clarifai	Microsoft	Amazon	Face++
CFD	368	4	69	234
NimStim	137	16	77	212
AirBnb	98	5	25	96
AI	58	34	28	70
<b>Total</b>	661	59	199	612

Table 5.4: Number of age misclassifications per tool.

	Clarifai	Microsoft	Amazon	Face++
CFD	30.4%	0.3%	5.7%	19.3%
NimStim	20.3%	2.3%	11.4%	31.5%
AirBnb	17.8%	0.9%	4.5%	17.4%
AI	23.3%	13.7%	11.2%	28.2%
<b>Total</b>	24.7%	2.2%	7.4%	22.9%

Table 5.5: Age misclassification rates of different tools.

threshold definition which is, inevitably, subjective. However, we have experimented with a wide range of tolerance intervals and the “*ranking*” of the APIs does not change. The numbers of misclassifications and the misclassification rates are provided in Tables 5.4 and 5.5.

As with the results for gender classification, we can see the best performing tool is Microsoft’s Face API. Microsoft’s algorithm has misclassified the age (that is, it has provided an unreasonable estimate of age) of only 59 faces, the majority of which are artificially generated. Amazon Rekognition follows closely with Face++ and Clarifai lagging behind.

In Section 4.3 we also mentioned that another feasible measure of accuracy would be considering the absolute difference between a subject’s true age (according to the ground truth), and the prediction of the algorithm or:

$$\frac{1}{N} \times \sum_{i=1}^N |age_{predicted}^{(i)} - age_{true}^{(i)}| \quad (5.1)$$

Calculating the expression above for all algorithms, we obtain the following values:

**Clarifai** - 8.93  
**Microsoft** - 3.82  
**Amazon** - 5.37  
**Face++** - 8.60

In fact, using this metric still produces the same results as our custom threshold definition of misclassification. Microsoft remains the best performer with an average offset of age of just 3.82 years. Amazon comes second with Face++ and Clarifai seriously behind, estimating 8 years off the target value on average.

What we just calculated in (5.1) was the average *absolute* difference. However, the average *signed*

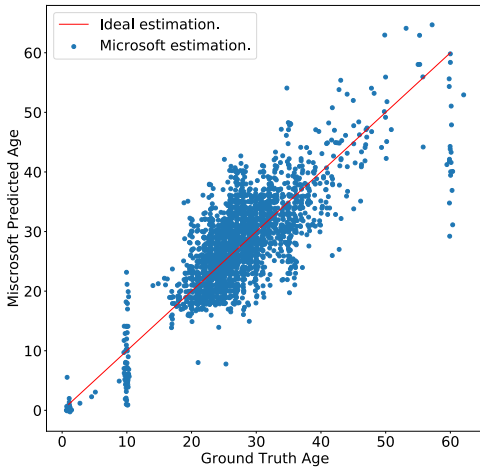
difference, which can be defined as:

$$\frac{1}{N} \times \sum_{i=1}^N (age_{predicted}^{(i)} - age_{true}^{(i)}) \quad (5.2)$$

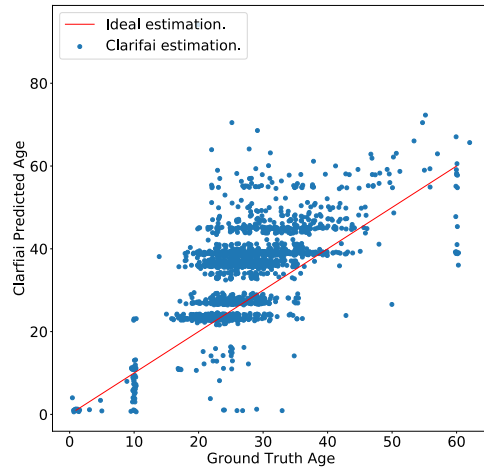
also yields interesting results, especially in the context of *unbiased estimators*. In statistics, an estimator  $\hat{\theta}$  for an estimand  $\theta$  is called unbiased if and only if  $E(\hat{\theta}) = \theta$  [30]. Notice that the notion of an unbiased estimator does not have anything to do with the concept of algorithmic bias – it simply means that *on average* the estimator guesses the true value correctly. If we assume that the four APIs we are considering are unbiased estimators of an individual’s age, then we would expect the expression in (5.2) to approach zero. However, we see that is not true for all APIs - if we calculate the value of (5.2) for each of them, we obtain:

**Clarifai** - 7.72  
**Microsoft** - 0.35  
**Amazon** - 2.24  
**Face++** - 7.37

We can see that while the estimators of Microsoft and, to some extent, Amazon fit the definition of an unbiased estimator, Clarifai and Face++ overestimate age with more than 7 years on average. Figures 5.4a and 5.4b illustrate that – we can observe that the “noise” in Microsoft’s model seems to be centered around zero with an equal amount of predictions falling below and above the true value. In contrast, Clarifai’s output is systematically above the true value and it also produces more significant outliers.



(a) Age estimation by Microsoft.



(b) Age estimation by Clarifai.

Figure 5.4: Comparing bias in the age estimators of Microsoft and Clarifai. The red line indicates  $f(x) = x$  or the output of an *ideal* estimator. For visualisation purposes, scattered points were “jittered” [31] by adding normal noise to avoid overlapping.

#### 5.1.4 Emotion Classification

Emotion classification is provided by three of the APIs in our study – Microsoft, Amazon and Face++. Also, the emotion attribute is not available in the AI dataset, so we will be quoting performance on the other three datasets.

	Microsoft	Amazon	Face++
CFD	239	191	310
NimStim	143	172	263
AirBnb	63	77	123
<b>Total</b>	445	440	696

Table 5.6: Number of emotion misclassifications per tool.

	Microsoft	Amazon	Face++
CFD	19.8%	15.8%	25.6%
NimStim	21.2%	25.5%	39.0%
AirBnb	11.4%	14.0%	22.4%
<b>Total</b>	18.3%	18.1%	28.6%

Table 5.7: Emotion misclassification rates of different tools.

The full results are provided in Tables 5.6 and 5.7. Once again, Big Tech algorithms are the most accurate performers and this time Amazon’s algorithm is slightly better than Microsoft’s one with a margin of just 5 misclassifications in total. Face++ still commits about 50% more errors than the other two algorithms, but this is a much smaller performance gap than the one observed in gender and age extraction.

We notice that all the APIs perform best on the AirBnb dataset, which is dominated by neutral and happy emotions. Meanwhile, NimStim is the “hardest” dataset. This could be expected since NimStim contains a considerably wider range of emotions.

## 5.2 Correlation Analysis

In this section of the report, we present the results of our correlation analysis, which aims to address the question posed by **RQ2**: *“Is misclassification biased towards certain demographic groups?”*. As outlined in our methodology, we will define bias in the cases where correlation has been established between a demographic input variable and the algorithmic output. We use Spearman’s Correlation testing with 95% confidence, so we consider p-values below 0.05 to be statistically significant.

We run the tests on all three demographic features extraction tasks (gender, race and age). For the

reasons described in Section 5.1.1, we exclude the AI set from the tests on gender classification. For each type of feature extraction task, we analyse the effect of each of the demographic attributes. We provide a summary of the more important results, but a detailed report of the correlation testing can be found in the `stage_2.ipynb` Jupyter Notebook available in the project’s Git repository.

### 5.2.1 Gender Classification

As mentioned back in Section 3.7, our dataset consists of 49.3% males and 50.7% females. Taking that into account we would expect that errors would be relatively balanced across males and females. However, Table 5.8 shows that for gender extraction this is not the case for all APIs. Misclassification in Face++ and Clarifai is clearly biased towards women – Clarifai has misclassified the gender of 22% of the female subjects and just above 1% of the male ones. The statistical significance of the bias is confirmed by the Spearman test which reports p-values well below 0.05 for Face++ and Clarifai. Meanwhile, Amazon and Microsoft show no indication of bias in that respect.

	<b>Clarifai</b>	<b>Microsoft</b>	<b>Amazon</b>	<b>Face++</b>
Females misclassified	271 (22.01%)	4 (0.32%)	32 (2.59%)	116 (9.42%)
Males misclassified	14 (1.16%)	5 (0.41%)	28 (2.33%)	17 (1.42%)
Correlation coefficient	-0.323	0.007	-0.008	-0.174
p-value	$3.582 \times 10^{-60}$	0.705	0.684	$2.975 \times 10^{-18}$

Table 5.8: The effect of gender on gender misclassification. Significant p-values produced by the Spearman test are highlighted in red.

Next, we want to test the correlation between race and gender misclassification. However, race is represented as an ordinal variable taking values from 0 to 5. This representation is not appropriate for correlation testing since it implies there are different distances between different race values. That is why we hypothesise that Black people are misclassified more often (as suggested by previous studies [8]) and consider an `is_black` variable, defined as 1 only if the subject is Black and 0 otherwise. Indeed, the tests suggest there is a correlation between being Black and gender misclassification across all APIs apart from Microsoft (which has only misclassified gender 9 times overall). However, the correlation is relatively weak for Amazon and Clarifai (reporting p-values of 0.011 and 0.005 and a correlation coefficients of about 0.05), and most significant for Face++ which has misidentified the gender of about 10% of Black subjects, and only 3% of non-Black ones.

Lastly, we find that age also affects gender classification. We discretise age into 5 age intervals as shown in Figure 5.5. For all four APIs, misclassification rates decrease with the increase of age. This correlation is validated by the Spearman tests which indicate significance for all four algorithms.

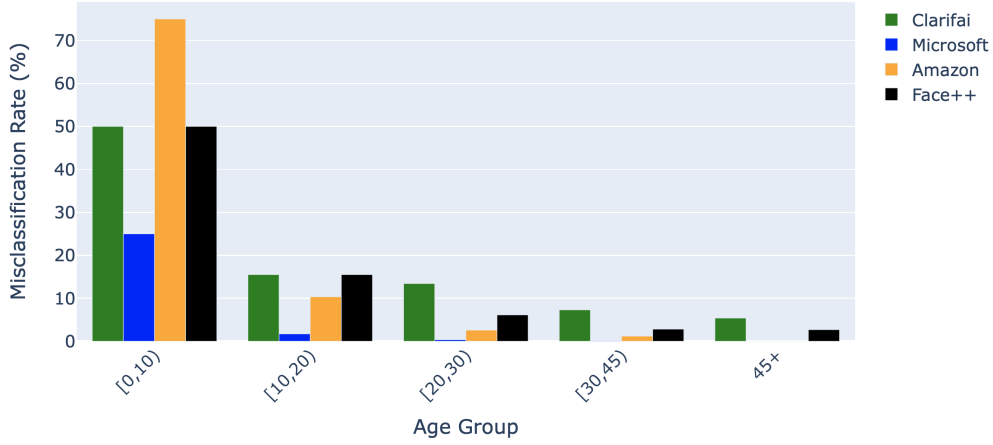


Figure 5.5: Gender misclassification rates for different age groups.

### 5.2.2 Race Classification

For race classification, we perform correlation tests on the only API that offers race extraction - Clarifai. If we examine the distribution of errors across genders, we observe that Clarifai has misclassified the race of 214 males (or 16.2% of all males) and 170 females (12.5%). Spearman’s test returns a low correlation coefficient of 0.0539 but a significant p-value of 0.005, indicating that misclassification is slightly biased towards men.

Repeating a similar analysis for the race attribute, we find that Clarifai misidentifies certain racial groups more often than others. The race of White and Latino individuals is misidentified respectively 16% and 49.7% of the cases, while Black and Asian subjects are misclassified in 7.5% and 4.7% of the time. If we conduct a correlation test between a custom variable `is_white_or_latino` and race misclassification, we obtain a significant p-value of  $1.14 \times 10^{-25}$ . This might owe to the fact that Latino and White subjects share some similar facial features and do not always differ significantly in terms of skin tone. Indeed, most of the misclassified White subjects have been identified as Latino by Clarifai (Figure 5.6b). Strangely, though, Latinos are most often misclassified as Asians (Figure 5.6a).

Interestingly, while Clarifai misclassifies the gender of females and Black individuals more often, race misclassification is actually biased towards males and lighter-skinned individuals. As far as age is concerned, the Spearman test indicated no evidence of bias towards certain age groups.

### 5.2.3 Age Classification

The tests show that age misclassification is also correlated with demographic attributes. Similar to the results for gender misclassification, we find that females’ age is misclassified more often by Clarifai and Face++ and this is confirmed by the results of the Spearman test. Meanwhile, age misclassification remains balanced for Amazon and Microsoft. The most significant bias is

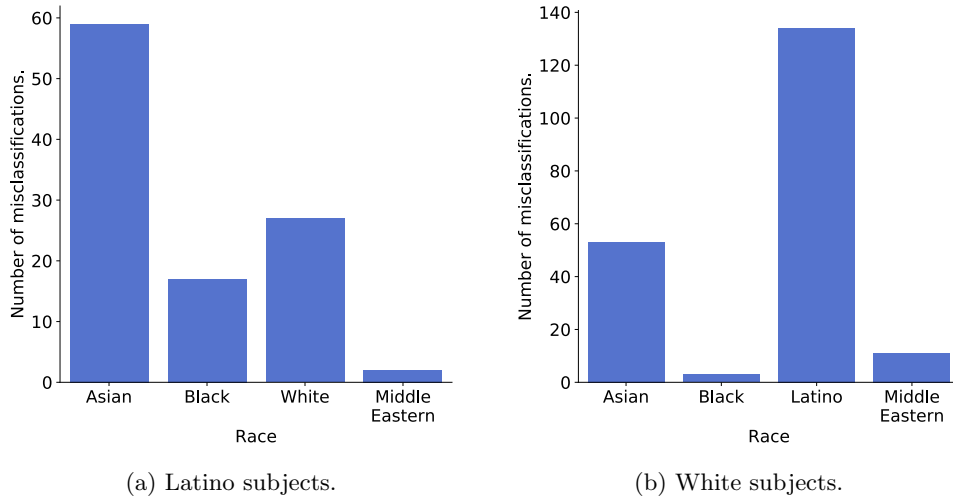


Figure 5.6: Distribution of Clarifai’s output when misclassifying Latino (a) and White (b) subjects.

detected in Clarifai which misclassifies the age of 29.1% of the female subjects and 20% of males.

For the effect of the race attribute, we conduct two correlation tests – one to test the correlation between being Black and age misclassification, and the other one to test the correlation between being White and misclassification. The results indicate that being Black is significantly *positively* correlated with age misclassification for Amazon and Clarifai, and significantly *negatively* correlated for Face++. That is, while Amazon and Clarifai tend to misclassify age more often when the subject is Black, the inverse relation holds for Face++. Additionally, Face++ is the only API that tests positive for correlation between being White and age misclassification. The algorithm misclassifies White people in 27.7% of the cases and non-Whites 18.5% of the time.

#### 5.2.4 Emotion Classification

In the case of emotion misclassification, we find no significant evidence of algorithmic bias. While all three APIs that offer the service (Microsoft, Amazon and Face++) misclassify Black and White individuals more often than Latino and Asian ones, this is likely due to the fact that Blacks and Whites constitute over 80% of the CFD and NimStim datasets. In those datasets, subjects exhibit a wide range of emotions, which makes emotion extraction a harder task.

### 5.3 Logistic Regression Analysis

The results of the regression analysis should help us measure the combined effect of an individual’s attributes on misclassification and find if this effect is significant enough to actually predict (with reasonable accuracy) whether a subject will get misclassified or not.

We start by implementing the *gender classification* model. After the initial training of the model,

we observe that the accuracy of the classifier is high at 84% but accuracy, in this case, is misleading since the dataset we are dealing with is highly imbalanced in favour of the zero-class. Indeed, a further check shows that the balanced accuracy (which is the metric we are really interested in) of this classifier is at a mere 50% as it acts much like a “*dummy classifier*” predicting the zero-class (standing for correct classification) in nearly all cases.

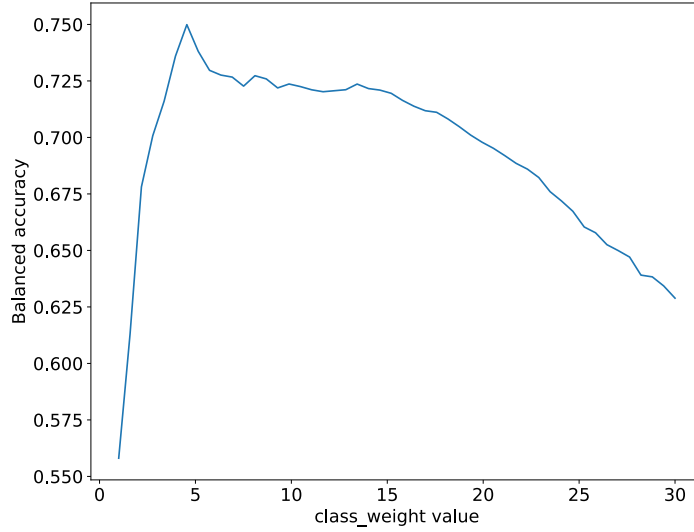


Figure 5.7: Hyperparameter tuning for `class_weight` with respect to balanced accuracy.

However, after hyperparameter tuning with respect to balanced accuracy (Figure 5.7), we manage to raise balanced accuracy to 74.35% (average balanced accuracy obtained from 5-fold cross-validation) by increasing the `class_weight` parameter, effectively increasing the penalty for false negatives. We can then analyse the weight vector of the resulting tuned classifier:

$$\vec{w} = \begin{bmatrix} \text{'Age'} = -0.37 \\ \text{'Gender'} = -1.13 \\ \text{'is.black'} = 0.15 \\ \text{'is.white'} = 0.09 \\ \vdots \\ \text{'is.surprised'} = -0.03 \\ \text{'is.disgusted'} = -0.20 \end{bmatrix} \quad (5.3)$$

The weight coefficients should be treated with caution since they have been obtained from optimising with respect to a nontraditional metric. Still, the weight vector does provide us with a few insights into the behaviour of the model. For a start, the most influential attribute in the input data (the one having a weight coefficient with the highest absolute value) is ‘Gender’. It has a very low negative value and since our model represents ‘male’ with a higher numerical value than ‘female’, it means the model interprets that male subjects are less likely to be misclassified, which



Model	Gender Classification	Race Classification	Age Classification	Emotion Classification
<b>Balanced Accuracy</b>	0.74	0.64	0.65	0.72
<b>Recall</b>	0.83	0.53	0.60	0.56
<b>Weighted Accuracy</b>	0.68	0.77	0.65	0.78

Table 5.9: Performances of the three misclassification prediction models. Results were obtained on the test data.

is exactly what our correlation analysis indicated for the case of gender classification. ‘Age’ has also been assigned a very low negative weight, which again means that higher age is associated by the model with lower probability of misclassification, which is yet another confirmation of our results from the correlation tests.

What is of greater interest to us though, is how well this model can utilise the obtained weights to predict unseen test data. Since we made use of cross-validation for evaluating our classifier, we expect to get similar results on the test data. Indeed, Table 5.9 shows that the classifier reports a balanced accuracy of 74% on the test data and a reasonable weighted accuracy of 68%, while achieving high recall of 83%.

Table 5.9 also summarises the test data results of the other three models, which have been obtained following an analogous training and evaluation procedure as the gender classification predictor that we just described. It can be seen that the performance of the other models is not as high, especially when it comes to recall. This leads us to believe that those types of misclassifications are harder to anticipate (at least by the logistic regression model which we have adopted).

## 5.4 Summary of Results

The results obtained from the data analysis stage of the project allow us to address the research questions we posed earlier:

**RQ1:** *“Does the performance of facial analysis vary across tools and datasets?”*

A key finding of our study is that facial analysis performance can vary dramatically across tools. Our results from Section 5.1 indicate that there is a significant performance gap between Big Tech companies and their smaller competitors. Microsoft seems to be the best “all-round” performer although Amazon reports narrowly better results for age classification. Meanwhile, Face++ and Clarifai lag behind in classification performance and, in the case of age estimation, tend to severely overestimate the true age of a subject.

Performance did vary on different datasets as well, but those variations were not systematic and, in most cases, could be explained by differences in the distribution of demographics, rather than the type of images (standardised or unstandardised) that they contained.

**RQ2:** *“Is misclassification biased towards certain demographic groups?”*

Through the use of statistical correlation testing, we managed to identify many areas of facial analysis where algorithmic bias is present. The extent and the significance of this bias, though, are not constant across tools. The tools that exhibited the best accuracy – Microsoft and Amazon, are also the two least biased APIs. On the other hand, the performance of Face++ and Clarifai suggests the presence of problematic biases embedded in their technologies.

Which demographic group is affected by algorithmic biases depends on the specific classification task. Gender classification errors seem to be prevalent among female and Black individuals, while race classification errors are more common with Whites and males. Even within the same classification task, different tools can exhibit different biases – for example, age misclassification is biased towards Blacks in the case of Amazon and Clarifai, but biased towards Whites for Face++.

**RQ3:** *“Can misclassification be predicted?”*

The logistic regression models we implemented in Section 5.3 demonstrated that indeed the combined effect of demographic variables and emotion is strong enough to predict misclassifications. The logistic regressor seems to deliver best results for predicting gender misclassifications where it achieves high balanced accuracy and recall. However, the tasks of predicting race, age or emotion misclassification are not as susceptible to this kind of modelling and for them, the predictive performance is lower.

## Chapter 6

# Limitations and Future Work

### Limitations

The authenticity of the data analysis process and therefore the findings of our study are dependent on the quality of the underlying data. That is why, when discussing the limitations of our study, we should start with the restrictions of our data engineering approach. The most obvious limitation in our study is the use of crowdworkers for producing ground truth data. While necessary for our project, crowdworking has implications for the quality of the data. Crowdworkers are humans and it is possible that they commit errors or even exhibit biases on their own. We believe that we have mitigated this issue through the design of the experiment, the cultural diversity of our crowdworkers and the data cleaning process. Still, crowdworking is a potential limitation of our study that has to be acknowledged.

Another limitation of the study is that it does not include “confidence values” or “probabilities” that some API providers attach to their outputs. Amazon Rekognition, for example, provides confidence values for almost all of the features it returns and recommends using a 99% confidence threshold when misclassification can be harmful<sup>1</sup>. However, not all APIs make use of confidence intervals, which is why, to keep the comparison of performance uniform and fair, we decided not to consider those values in the study.

Lastly, to provide a measure of quantifying biases, we have occasionally made use of custom metrics and definitions which could be seen as subjective. One example is the threshold for age estimation which we discuss in Stage 4.3. While those methods can indeed be imperfect, we have aimed to justify their use and made sure that arbitrary decisions in the analysis process have not affected the core results and conclusions of the study.

### Future Work

Algorithmic fairness is a young and dynamic field of computer science and plenty of pressing problems remain unaddressed. For instance, at the time of writing, we cannot identify a single publicly

---

<sup>1</sup><https://docs.aws.amazon.com/rekognition/latest/dg/guidance-face-attributes.html>

available set of human faces that is large enough and diverse enough to provide a reliable performance benchmark for facial analysis. Most related studies (including ours) tackle the problem by generating their own datasets, which is time-consuming and prone to errors. The creation of such a dataset would therefore be highly beneficial to the research community, but it would also pose a significant challenge in the light of new restrictions posed by data privacy regulations.

And while bias in facial analysis technologies has been the center of a lot of research lately, there are multiple other areas of technology of great social importance where biases to this day remain undetected or unexplored. Future studies can apply the concept of “algorithmic audit” to speech and voice recognition, chatbots, automated decision systems and many more.

## Chapter 7

# Conclusion

This concludes our study into biases in facial analysis. Throughout the project we assembled a dataset of images of human faces and processed it through a selection of popular commercial APIs. The subsequent data analysis produced some interesting findings:

- We found that performance can vary greatly across facial analysis tools both with regards to accuracy and bias as **Big Tech companies dominate their smaller challengers**.
- Depending on the feature extraction task, certain **demographic groups can be significantly impacted by algorithmic biases**. We identify a new source of algorithmic bias – age, and show that gender and race affect classification beyond the well-studied task of gender extraction.
- **The demographic profile and emotion of a subject can be used to predict certain algorithmic errors** and target individuals in high risk of misclassification.

In the words of Raju and Buolamwini, algorithmic audits exert “*external pressure*” on algorithmic vendors which “*remains a necessary approach to increase transparency and address harmful model bias*” [12]. We hope that the results of our “algorithmic audit” will lead to better public understanding of the capabilities and limitations of facial analysis, and the real and present threat of algorithmic bias.

# Bibliography

- [1] M. K. Scheuerman, J. M. Paul, and J. R. Brubaker, “*How Computers See Gender: An Evaluation of Gender Classification in Commercial Facial Analysis and Image Labeling Services*”, Proc. ACM Hum.-Comput. Interact (2019).
- [2] J. A. Coan and J. B. Allen, “*Handbook of Emotion Elicitation and Assessment*” (2007)
- [3] Y. Gurovich, “*Identifying facial phenotypes of genetic disorders using deep learning*”, Nature Medicine (2019).
- [4] R. P. Wildes, “*Iris recognition: an emerging biometric technology*”, Proceedings of the IEEE (1997).
- [5] W. Mou, H. Gunes and J. Patras, “*Your Fellows Matter: Affect Analysis across Subjects in Group Videos*”, 14th IEEE International Conference on Automatic Face & Gesture Recognition (2019).
- [6] V. Koh, W. Li, G. Livan and L. Capra, “*Offline Biases in Online Platforms: a Study of Diversity and Homophily in Airbnb*”, EPJ Data Science (2019).
- [7] P. Barlas et al., “*Social B(eye)as: Human and Machine Descriptions of People Images*”, Proceedings of the Thirteenth International AAAI Conference on Web and Social Media (2019).
- [8] J. Buolamwini and T. Gebru, “*Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification*”, Conference on Fairness, Accountability, and Transparency (2018).
- [9] K. Thompson, “*Reflections on Trusting Trust*”, Turin Award Lecture (1984).
- [10] Z. Yang, A. Zhang and A. Sudjianto, “*Enhancing Explainability of Neural Networks through Architecture Constraints*” (2019).
- [11] P. Gajane and M. Pechenizkiy, “*On Formalizing Fairness in Prediction with Machine Learning*” (2018).
- [12] I. D. Raji and J. Buolamwini, “*Actionable Auditing: Investigating the Impact of Publicly Naming Biased Performance Results of Commercial AI Products*”, Association for the Advancement of Artificial Intelligence (2019).
- [13] R. K. E. Bellamy et al., “*AI Fairness 360: An Extensible Toolkit for Detecting, Understanding, and Mitigating Unwanted Algorithmic Bias*” (2018).

- [14] M. Hardt, E. Price and N. Srebro, “*Equality of Opportunity in Supervised Learning*” (2016).
- [15] S. A. Friedler et al., “*A comparative study of fairness-enhancing interventions in machine learning*” (2018).
- [16] J. Zhao et al., “*Men Also Like Shopping: Reducing Gender Bias Amplification using Corpus-level Constraints*” (2017).
- [17] K. Burns et al., “*Women also Snowboard: Overcoming Bias in Captioning Models*” (2019).
- [18] M. Merler et al., “*Diversity in Faces*” (2019).
- [19] D. S. Ma, J. Correll and B. Wittenbrink, “*The Chicago face database: A free stimulus set of faces and norming data*”, Behaviour Research Method (2015).
- [20] N. Tottenham et al., “*The NimStim set of facial expressions: Judgments from untrained research participants*”, Psychiatry Research (2009).
- [21] C. Ho, A. Slivkins, S. Suri and J. W. Vaughan, “*Incentivizing High Quality Crowdwork*”, WWW ’15: Proceedings of the 24th International Conference on World Wide Web (2015).
- [22] P. Ipeirotis, F. Provost and J. Wang, “*Quality Management on Amazon Mechanical Turk*”, Conference on Human Computation & Crowdsourcing (2010).
- [23] R. Sacco, “*Re-Envisaging the Eight Developmental Stages of Erik Erikson: The Fibonacci Life-Chart Method (FLCM)*”, Journal of Educational and Developmental Psychology (2013).
- [24] M. Zafar, I. Valera, M. Rodriguez and K. Gummadi, “*Fairness Beyond Disparate Treatment Disparate Impact: Learning Classification without Disparate Mistreatment*”, 2017 International World Wide Web Conference Committee (2017).
- [25] D. Hellman, “*Measuring Algorithmic Fairness*”, University of Virginia Law School: Public Law and Legal Theory Paper Series (2019).
- [26] S. Verma and J. Rubin, “*Fairness Definitions Explained*”, 2018 ACM/IEEE International Workshop on Software Fairness (2018).
- [27] S. Yue, P. Pilon and G. Cavadias, “*Power of the Mann–Kendall and Spearman’s rho tests for detecting monotonic trends in hydrological series*”, Journal of Hidrology (2002).
- [28] C. Cleverdon, “*On the Inverse Relationship of Recall and Precision*”, Journal of Documentation (1972).
- [29] P. Domingos, “*A Few Useful Things to Know about Machine Learning*”, Communications of the ACM (2012).
- [30] V. Voinov and M. Nikulin, “*Unbiased Estimators and Their Applications*”, Volume 1: Univariate Case (1993), pp. 19-21.
- [31] A. Gelman and J. Hill, “*Data Analysis Using Regression and Multilevel/Hierarchical Models*” (2007), pp. 32.

# Appendix A

## Code Listing

Below is a listing of some of the relevant code produced during the project. The entire code, together with logs and corresponding data can be found at:

[https://github.com/samuil1998/researchproject\\_public](https://github.com/samuil1998/researchproject_public)

Because of data privacy regulations, the contents of the CFD, NimStim and AirBnb datasets have not been included in this public repository. Also, for security purposes, details of the code have been omitted (such as strings containing API keys, AWS credentials or S3 bucket names).

### A.1 Guide into the File Structure

The root directory of the Git repository is split into five folders:

- **analysis** – contains the Jupyter notebooks for the data analysis. **analysis.ipynb** is used for visualisation. The **stage\_1**, **stage\_2** and **stage\_3** notebooks implement the accuracy performance, the correlation testing and the logistic regression analysis respectively.
- **api\_processing** – contains the scripts for the API processing (in the **/Code** subdirectory) and the datasets (**/Data**). In the **/Code** subdirectory, each API has a folder assigned, which in turn contains one folder per dataset. In each of those folders, you can find one processing script named **<API\_name>\_processing.py**, the resulting logs and a script transforming the logs into a **.csv** file.
- **experiment** – contains data and scripts for the crowdworking experiment.
- **preprocessing** – includes the data standardisation scripts.
- **tables** – includes the **.csv** tables which contain the API processing results.



## A.2 NimStim JPEG Conversion

```
1 from PIL import Image
2 import os
3
4 NIMSTIM_PATH = '/Users/samuilstoychev/Desktop/researchproject/api_processing/
  Datasets/NimStim/Crop-White Background/'
5
6 def create_jpg(name):
7     img = Image.open(NIMSTIM_PATH + name)
8     img = img.convert('RGB')
9     name = name.split(".")[0] + ".jpeg"
10    img.save("./converted/" + name, "jpeg")
11
12 for filename in os.listdir(NIMSTIM_PATH):
13     # Filter images (the folder might contain OS files such as .DS_Store)
14     extension = filename.split(".")[1]
15
16     if extension == "BMP":
17         print(1)
18         create_jpg(filename)
```

Listing A.1: convert\_to\_jpeg.py

## A.3 AirBnb Random Sampler

```
1 import os
2 import shutil
3 import random
4
5 HK_PATH = "/Users/samuilstoychev/AirBnb/hk/"
6 CH_PATH = "/Users/samuilstoychev/AirBnb/chicago/kevin/"
7 DEST = "/Users/samuilstoychev/Desktop/researchproject/api_processing/Datasets/
  AirBnb/random_sample"
8
9 hk_pictures = os.listdir(HK_PATH)
10 ch_pictures = os.listdir(CH_PATH)
11
12 hk_random_500 = random.sample(range(1, len(hk_pictures)), 500)
13 ch_random_500 = random.sample(range(1, len(ch_pictures)), 500)
14
15 count = 0
16
17 for index in hk_random_500:
18     file_name = hk_pictures[index]
19     if file_name.split(".")[1] == "jpg":
20         file_path = HK_PATH + file_name
21         shutil.copy(file_path, DEST)
22         count += 1
23
24 for index in ch_random_500:
25     file_name = ch_pictures[index]
26     if file_name.split(".")[1] == "jpg":
27         file_path = CH_PATH + file_name
```

```

27     shutil.copy(file_path, DEST)
28     count += 1
29
30 print("Sampled " + str(count) + " random images. ")

```

Listing A.2: random\_sample.py

## A.4 AI Dataset Generation

The three scripts below - request\_urls.py, export\_urls.py and download\_images.py generate the AI dataset by respectively making the request to the Generates Photos API, transforming the JSON logs into a .csv file, and finally downloading the images locally.

```

1
2 import requests
3 import json
4 from flask import jsonify
5 import itertools
6
7 API_KEY = "API-Key <insert-API-Key-here>."
8 genders = ["male", "female"]
9 ethnicities = ["white", "latino", "asian", "black"]
10 ages = ["infant", "child", "young-adult", "adult", "elderly"]
11
12 def generate_uri(gender, ethnicity, age):
13     return "https://api.generated.photos/api/v1/faces?per_page=100&gender={}&ethnicity={}&age={}".format(gender, ethnicity, age)
14
15 if __name__ == "__main__":
16
17     for x in itertools.product(genders, ethnicities, ages):
18         gender = x[0]
19         ethnicity = x[1]
20         age = x[2]
21
22         uri = generate_uri(gender, ethnicity, age)
23         response = requests.get(uri, headers={'Authorization': API_KEY})
24         output_file_name = "{}_{}_{}.json".format(gender, ethnicity, age)
25
26         with open(output_file_name, 'w') as output_file:
27             response_json = json.loads(response.text)
28             json.dump(response_json, output_file)

```

Listing A.3: request\_urls.py

```

1 import os
2 import json
3 import csv
4 faces = os.listdir("./faces")
5
6 rows = []
7
8 for file in faces:

```

```

9     face_tags = file.split(".")[0].split("_")
10
11     gender = face_tags[0]
12     ethnicity = face_tags[1]
13     age = face_tags[2]
14
15     with open("./faces/" + file) as json_file:
16         python_dict = json.load(json_file)
17         for obj in python_dict["faces"]:
18             id = obj["id"]
19             url = obj["urls"][4]['512']
20             rows.append((id, gender, ethnicity, age, url))
21
22 # Open file and write the information
23 with open("faces_data.csv", "w", newline='') as output_file:
24     csvwriter = csv.writer(output_file, quoting=csv.QUOTE_MINIMAL)
25     csvwriter.writerow(("ID", "Gender", "Ethnicity", "Age", "URL"))
26     for row in rows:
27         csvwriter.writerow(row)

```

Listing A.4: export\_urls.py

```

1 import csv
2 import urllib.request
3
4 CSV_LOCATION = './faces_data.csv'
5
6 def get_input_data():
7
8     with open(CSV_LOCATION, newline='') as csvfile:
9         csvreader = csv.reader(csvfile, delimiter=',')
10        return list(csvreader)[1:]
11
12 inputs = get_input_data()
13 ids = [x[0] for x in inputs]
14
15 for input in inputs:
16     id = input[0]
17     gender = input[1]
18     race = input[2]
19     age = input[3]
20     url = input[4]
21     output_name = "_".join([id, gender, race, age]) + ".jpeg"
22     urllib.request.urlretrieve(url, "./images/" + output_name)

```

Listing A.5: download\_images.py

## A.5 Data Standardisation

```

1 gender_mapping = {
2     1: ('F', 'Female', 'f', 'female', 'feminine'),
3     2: ('M', 'Male', 'male', 'masculine'),

```

```

4     0: ('unsure', )
5 }
6
7 race_mapping = {
8     1: ('A', 'asian'),
9     2: ('B', 'black', 'black or african american'),
10    3: ('L', 'latino', 'hispanic, latino, or spanish origin'),
11    4: ('W', 'white'),
12    5: ('middle_eastern', 'middle eastern or north african'),
13    0: ('other', 'american indian or alaska native', 'native hawaiian or pacific
14        islander'),
15 }
16
17 emotion_mapping = {
18     # Angry
19     1: ('A', 'AN', 'ANGRY', 'an', 'anger', 'angry'),
20     # Calm / Neutral
21     2: ('CA', 'CALM', 'N', 'NE', 'ca', 'neutral'),
22     # Disgusted
23     3: ('DI', 'DISGUSTED', 'di', 'disgust'),
24     # Fearful
25     4: ('F', 'FE', 'FEAR', 'fear', 'scared'),
26     # Happy
27     5: ('HA', 'HAPPY', 'HC', 'HO', 'happiness', 'happy'),
28     # Sad
29     6: ('SA', 'SAD', 'sad', 'sadness'),
30     # Surprised
31     7: ('SURPRISED', 'surprise', 'surprised', 'SP'),
32     # Other
33     0: ('CONFUSED', 'contempt', 'other'),
34     # None!
35     -1: (None, )
36 }
37
38 # Only for AI (need to make the discrete categories continuous)
39 age_mapping = {
40     'adult': 35,
41     'child': 10,
42     'elderly': 60,
43     'infant': 1,
44     'young-adult': 25
45 }

```

Listing A.6: Numerical mappings of attributes. Full standardisation code available in the `data_preprocessing` Jupyter Notebook

## A.6 Definition of Age Misclassification

```

1 def age_misclassification(gt, prediction):
2     """Given a vector of ground truth values 'gt' and a vector of prediction values
3         'gt', return a vector containing 1 if there has been age misclassification and
4         0 otherwise. """
5     n = len(gt)

```

```

4     assert(len(prediction) == n)
5     result = np.zeros(n)
6
7     for i in range(n):
8         tolerance = 0
9         # If the subject is a child, tolerance interval equals 5
10        if gt[i] <= 10:
11            tolerance = 5
12        # For teenagers and pre-young-adults, tolerance is 10
13        elif gt[i] <= 25:
14            tolerance = 10
15        # For the rest of the population, tolerance is 15
16        else:
17            tolerance = 15
18
19        if gt[i] - tolerance <= prediction[i] <= gt[i] + tolerance:
20            result[i] = 0
21        else:
22            result[i] = 1
23    return result

```

Listing A.7: Defining age misclassification through a tolerance interval. Part of the `stage_1` Jupyter Notebook

## Appendix B

# Crowdworking Experiment Design

The figures below are screenshots from the Figure Eight crowdworking experiment. Because of data privacy considerations, the corresponding images are not shown in this report. You can see the full design of both the NimStim and the AirBnb experiment (including instructions, examples given to the contributors and sample tasks) in the project's Git repository - at `/experiment/NimStim/layout.htm` and `/experiment/AirBnb/layout.htm` respectively.

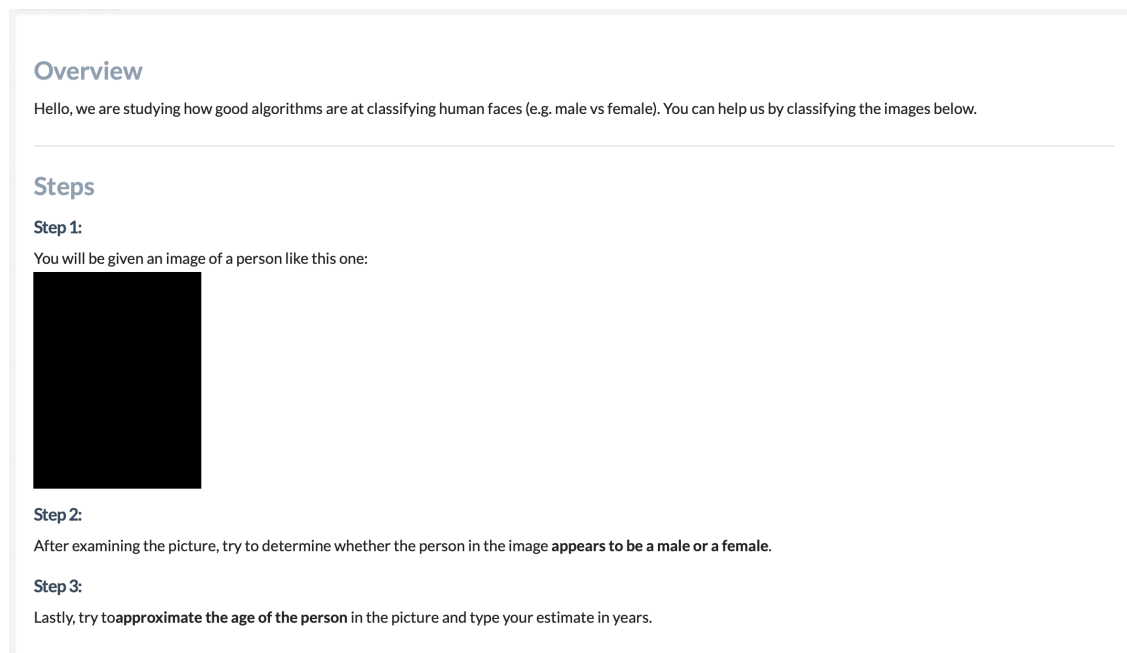
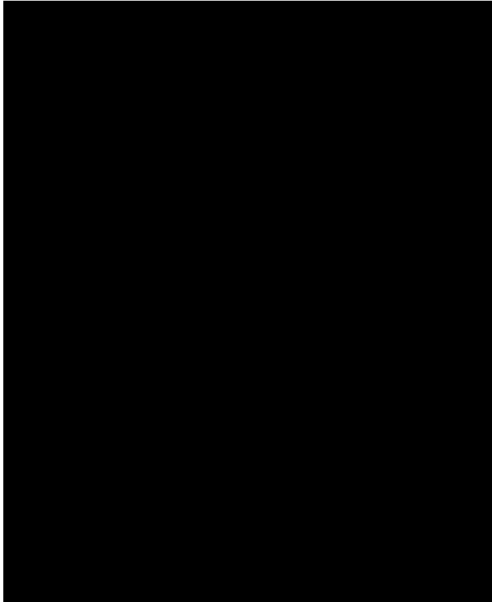


Figure B.1: NimStim experiment - instructions for contributors.



Is the person in the image male or female? (required)

☒ Male  
☐ Female  
☐ I am not sure.

Age (in years): (required)

27

Figure B.2: NimStim experiment - task layout.

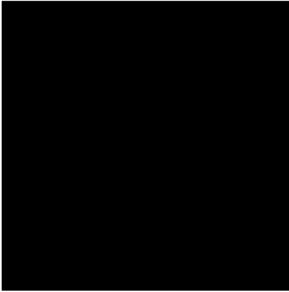
## Overview

Hello, we are studying how good algorithms are at classifying human faces (e.g. male vs female). You can help us by classifying the images below.

---

## Steps

**Step 1:**  
You will be presented with a picture like this one:

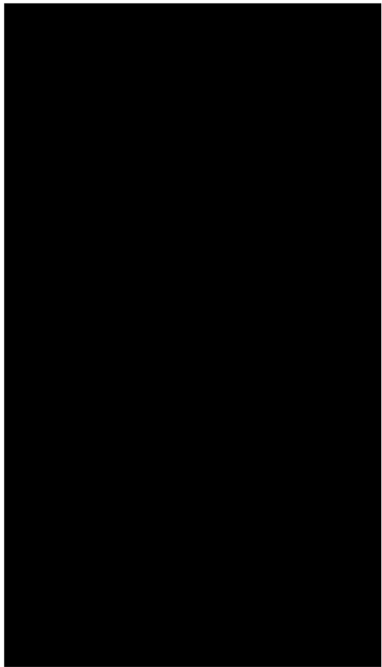


**Step 2:**  
The picture can contain one or more human faces. Your first task will be to **determine whether or not the image contains more than one face**. (Ignore faces seen in the background).

**Step 3:**  
If the picture only contains one face, you will be asked to determine the appearance of the person in terms of **gender, ethnicity and emotion**.

**Step 4:**  
Finally, you need to provide an estimate of the person's age as a whole number.

Figure B.3: AirBnb experiment - instructions for contributors.



**Can you clearly see more than one face in this picture? (required)**

☐ Yes

☒ No

**Is the person in the picture male or female? (required)**

☐ Male

☒ Female

☐ I am not sure.

**What is the ethnicity of the person? (required)**

☐ White

☐ Black

☐ Asian

☒ Latino

☐ Middle Eastern

☐ Other

**What emotion does the person express? (required)**

☐ Neutral or Calm

☒ Happy

☐ Sad

☐ Angry

☐ Scared

☐ Surprised

☐ Disgusted

☐ Other

**How old do you think the person is? (in years): (required)**

Figure B.4: AirBnb experiment - task layout.



# Appendix C

## Project Plan

**Student:** Samuil Stoychev

**Project Title:** “Quantifying Biases in Facial Analysis Tools”

**Supervisor:** Professor Licia Capra

### 1. AIMS & OBJECTIVES

**Aim of the project:** Face recognition is a branch of computer science with a growing importance in medicine, security, social media, etc. However, agreement between facial analysis tools is often low, especially when extracting demographic features, such as race, age and gender. This project aims to address this problem, by analysing and estimating the biases of popular image recognition tools (Face++, Amazon Rekognition, Clarifai, Azure Cognitive Services).

#### **Objectives:**

- Review recent research on the problem, social and technological implications, resources in terms of tools and data.
- Research facial analysis and tools and their suitability for the task.
- Annotate the input data via the Amazon Mechanical Turk crowdsourcing service and build a ground-truth data set.
- Process images from different data sets through the facial analysis tools. Extract demographic features and cleanse and validate the collected data to prepare it for processing.
- Analyse the processed data and check for patterns in inaccuracies. Check against the ground truth and quantify the bias of the technologies.
- Make conclusions on the basis of the analysis. How (in)accurate are facial analysis tools? Are the technologies less accurate under certain conditions (pictures of low quality, bad lighting, images of people of a specific race and/or gender, etc.)? Do the biases reflect the social biases indicated by crowd workers?

- Provide a measure of tool accuracy and benchmark tools against each other and under varying conditions.

## 2. DELIVERABLES

Deliverables for this project include:

- A literature review on the existing research in facial analysis tools.
- A comparison of face recognition technologies.
- An annotated data set of tool-processed data (potentially useful for future research).
- Detailed analysis on the collected data.
- The scripts used for the collection, processing and analysis of data.

## 3. WORK PLAN

The work plan for the project is as follows:

- Project start to end of October: Literature review and selection of facial analysis tools and image data sets.
- November: Estimation of the cost for the processing of data. Gaining ethics approval. Processing the data through the tools and Amazon Mechanical Turk.
- December to end of January: Cleansing the data and performing data analysis on it.
- 1st – 15th February: Validating and drawing conclusions from the results.
- Mid-February to End-March: Writing and editing the final report.

# Appendix D

## Interim Report

**Name:** Samuil Stoychev

**Project Title:** “Quantifying Biases in Facial Analysis Tools” (remains unchanged)

**Supervisor:** Professor Licia Capra

### 1. CURRENT PROGRESS

#### 1.1. LITERATURE REVIEW

The first couple of weeks after the beginning of the project in September have been dedicated to a literature review. We have analysed papers dealing with the issues of algorithmic fairness and image recognition. This provided an insight into the current challenges in the field and related work.

The literature review helped identify common approaches to dealing with the problems with algorithmic fairness. It also shed light on the underlying issues related to image recognition tools and algorithmic bias.

#### 1.2. SELECTION OF TOOLS AND DATA SETS

The first major decision during the course of the project was the selection of facial analysis tools that we wanted to examine as well as the data sets we wanted to run them on. We selected four image recognition APIs – Clarifai, Face++, Amazon Rekognition and Microsoft Azure Cognitive Services. We examined the capabilities and the restrictions of those technologies through their respective online documentations.

As for data sets, we decided to explore a wide variety of images to get a better approximation of the tools’ performance under different inputs. We selected two data sets with standardised images – Chicago Face Dataset and NimStim. Those sets contain labelled data that would be useful for obtaining a reliable ground-truth for the data analysis stage. On top of those, we are using Airbnb’s data set of hosts’ profile pictures (as an instance of non-standardised imagery) and we

have also created a set of AI-generated faces using the API provided by “Generated Photos”.

Finally, we went through UCL’s ethical approval procedures and justified the use of image recognition tools and the data sets.

### 1.3. DATA ENGINEERING

We processed the images of all four data sets through each of the four APIs. We have extracted gender, ethnicity, age and emotion (where the API provides those features). The data was cleaned and exported in .csv format to facilitate the data analysis later.

As a number of the APIs require providing images as publicly accessible URLs (Amazon Rekognition even requires S3 objects), images have been stored in AWS S3 buckets. The buckets have been made private with listing options turned off to ensure data is stored securely. As per GDPR requirements, the images will be removed from the AWS platform at the end of the project.

All image processing has been done using the free tiers the different APIs provide and the corresponding Python scripts have been stored in a private GitHub repository, which will be made available at the end of the project.

Lastly, we have designed and conducted a crowdworking experiment to create a ground truth for the data sets where labelled data was missing or incomplete. We used Figure Eight (formerly known as Crowdfunder) for the task.

## 2. FUTURE WORK

Future work will deal mainly with cleaning and normalizing the data collected from image processing and crowdworkers to prepare it for analysis. In particular, the crowdworkers data should be filtered against invalid inputs and bad contributors, and the features obtained from the API processing need to be standardised.

The last major task in the project would be conducting data analysis on the results. We will begin by devising a strategy to tackle the problem comparing existing approaches, metrics for identifying bias as well as statistical techniques. Ideally, the results should provide us with insights into facial analysis (and maybe human) biases and a benchmark for comparing API algorithmic fairness performance.

The data analysis stage should be finished by the end of February, which would leave about a month for writing and editing the final report.