An Analysis and Evaluation of Facial Recognition Software

Mairi McQueer - 1700231

CMP320 - Ethical Hacking 3 (Mini-Project)

BSc Ethical Hacking

19th May 2020

Abstract

This paper looks into some commonly used facial recognition software and compares their results against a series of photos which contain various levels of evasion techniques. First explaining the basics of this software, what neural networks are and how they are used in order to identify a person's face from an image or video. The three tools used in this report, Face++, AWS' Rekognition and OpenCV, are then described; including their origins, use cases and how the algorithms they implement work. Face++ and Rekognition are implemented, using the web API and Python respectively. OpenCV was attempted and the Python code included in this report but the author was unable to debug successfully and so no results could be gathered. The results revealed a clear disparity between the two algorithms, where Face++ was only able to identify 61% of the faces but Rekognition identified 88% and the latter proved to be much more accurate in determining both age and gender.

Contents

1 Introduction **5**

1.1 Background 5

Face++ 6

Rekognition 7

OpenCV 7

1.2 Aim 8

2 Procedure **9**

2.1 Overview 9

CV Dazzle Makeup 9

2.2 Face++ 9

Face Detection and Comparison 9

2.3 Rekognition 10

Detection 10

Comparison 11

2.4 OpenCV 12

3 Results **14**

3.1 Face++ 14

3.2 Rekognition 14

3.3 OpenCV 15

4 Discussion **16**

4.1 General Discussion 16

4.2 Countermeasures 16

4.3 Conclusions 17

4.4 Future Work 17

5 References **18**

6 Acronyms Used **19**

7 Appendices **19**

7.1 Face++ Male Results 19

7.2 Face++ Female Results 20

7.3 Face++ CV Dazzle Results 20

7.4 Rekognition Female Results 21
7.5 Rekognition Male Results 21
7.6 Rekognition CV Dazzle Results 22
7.7 Output from Rekognition Face Detection 22
7.8 Output from Rekognition Face Comparison 22
7.9 Output from Face++ Face Detection 23
7.10 Output from Face++ Face Comparing 26

1 Introduction

1.1 Background

Facial Recognition Software (FRS) is a broad term for software capable of identifying a face from a still image or video source, often used to then compare it to a facial database in order to verify the person's identity. *(Techopedia, 2019)*

Although there are a considerable number of different facial recognition algorithms the basic principle of how the software operates remains the same. These are broken into the following stages; First is detection where the software gathers information about the head's measurements and locates facial features. The second step is normalisation when the face is captured then scaled and rotated appropriately to allow the software a clearer look at the face. Then the image is converted into a value and stored within the software's own database, this is known as representation. Finally matching, the value is compared to all other values within the database in order to try and locate a match and determine the identity.*(Science ABC, 2019)*

Modern FRS makes use of machine learning and an Artifical Neural Network (ANN) in order to allow the algorithm to 'learn' and improve its accuracy with every image it is given. ANNs are designed to model a human brain with the nodes representing neurons and work by passing an input through a number of functions and at each stage calculating a weight of the outputs to then create a final result. The nodes that perform the functions are held within the hidden layer and these nodes can be fully connected, where each node in one layer connects to all of the others within the next layer. (Figure 1)

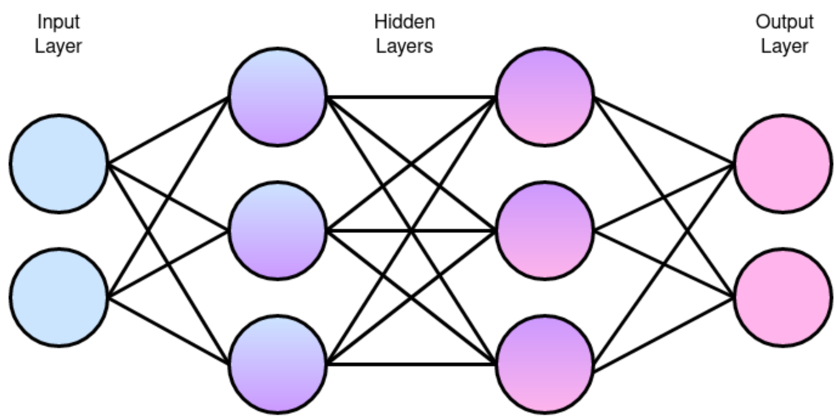

Figure 1: Diagram of a Simple Artificial Neural Network

Deep Neural Network (DNN) are a form of ANN that consists of multiple layers of inputs and outputs. One type of DNN are Convolutional Neural Network (CNN) which is designed for image recognition and language processing. The distinguishing features of this network type is the inclusion of a convolutional layer where the nodes are organised three-dimensionally. This layer consists of a series of filters which is applied to the image as a whole and the filtered image is passed onto the next node until a recognised pattern is detected. CNN also contain a pooling layer which is used to reduce the size of image to make it easier to discover features. *(Bengio, 2009)*

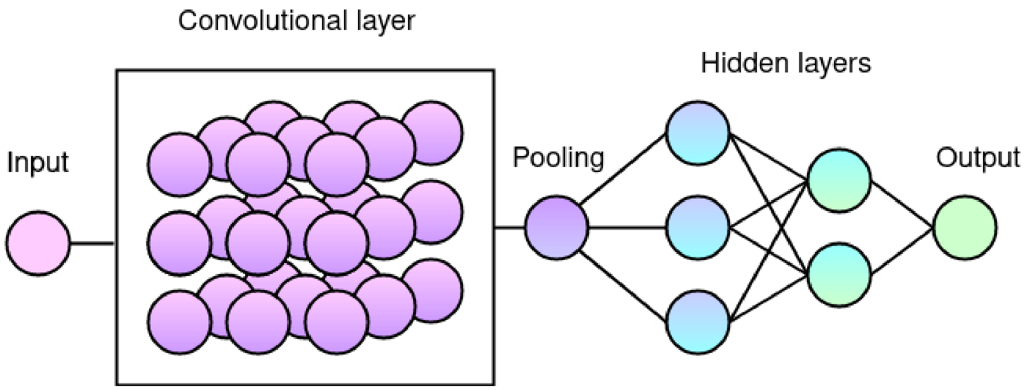

Figure 2: Diagram of a Simple Convolutional Neural Network

These algorithms need trained in order to determine the weighting of each output, this is done by providing a data set containing positive and negative data and informing the algorithm if it was successful in identifying the correct images. In the case of the software being discussed in this paper, weighting is the importance of a facial feature to determine whether the image contains a face or not and the output is an identification of a face within an image. There are an almost endless amount of use cases for this software but since it is not considered wholly reliable, as of writing, the number of adopters is low.

As previously mentioned there is a wide array of different facial recognition softwares available but due to the investigators limited budget, free applications were used exclusively for this paper. An introduction and overview of each will be detailed in the following sections.

Face++

The first piece of software that was analysed is Face++, created by the Chinese company Megvii Technology in 2012. There are eight different Application Programming Interface (API)s related to face detection offered by this organisation; Detection, Analyse, Compare, Search, Skinanalyze V1, Dense Facial Landmarks, 3DFace V1 and Beautify V2. Megvii also provides a cloud storage system for storing images to be used with Face++ called FaceSet. The cost of use for the face detection API is around ¥710 or £82 a day, there is also a free tier which is slightly limited but allows for use of three of the detect API's and their SDKs for twenty-four hours.

Face++ offers tools to use this software is the following languages; Python, Java, C++, Objective C, JavaScript and PHP. All of these tools and sample scripts are available on GitHub in both English and Chinese.

The detection API works by taking a JPG or PNG file and uses either 83 or 106 facial landmarks, depending on the version being used, to determine the following attributes:

- Gender
- Age
- Smiling
- Headpose
- Facequality
- Blur
- Eyestatus
- Emotion
- Ethnicity
- Beauty
- Mouthstatus
- Eyegaze
- Skinstatus

Most of the attributes are assigned an array of floating point numbers, the amount of which depends on the categories within each attribute, between zero and one hundred, representing the algorithms confidence of the attribute existing. Headpose is slightly different as it gives three values between 180 and -180 for the x, y and z angles of a person's head.

Compare takes two images and returns the following values confidence, thresholds and time_used. The first value, confidence, contains a single floating point number between zero and one hundred the higher the value the greater the algorithms confidence that the two faces belong to the same person. Thresholds contains an array of three floating point numbers that are placed in three categories, each depicting the confidence threshold of a different error rate. 1e-3; 0.1% , 1e-4; 0.01% and 1e-5 at 0.001%. If the faces return a value greater than the 1e-3 then they likely do not contain the same face contrariwise if the value is less than the strict 1e-5 threshold then it is highly likely to be two images of the same person. Time_used simply contains the time taken to complete the operation in milliseconds.

Face++ is currently very popular in Eastern Asia and some of it's current use cases are; Chinese law enforcement's analysis of CCTV, KFC's 'smile to pay' system, Didi Chuxing's ride sharing application and the mobile payment application Meitu. *(Forbes, 2019)*

Rekognition

Created by Amazon in 2016 and is currently one of eighteen machine learning services offered by Amazon Web Services (AWS), it is structured as a CNN. It is currently being used by law enforcement agencies in America such as ICE, Oregon Sheriff's Office and Homeland Security.

Rekognition offers a Software Development Kit (SDK) in C++, Go, Java, JavaScript, .NET, Node.Js, PHP, Python and Ruby. Alongside the SDKs AWS also has plugins for most of the popular IDEs, PyCharm, IntelliJ, Cloud9 and Visual Studio Code, as well as both Windows and Linux command line tools.

Due to the popularity of AWS there are vast amounts of official and unofficial documentation and support available online for this software; such as development guides, GitHub sample code, forums and blogs. The types of facial analysis Rekognition provides can be broken down into three main categories; detection, comparison and identification.

Detection, similarly to Face++'s version, returns confidence values based on the appearance of various features in the image. The main difference between this algorithm and Megvii's is that it also returns a confidence value on the image containing a face. The outputs are also presented in floating point numbers but the categories vary slightly, as shown below.

- Face
- Smile
- Gender
- Eyes Open
- Emotion
- Eyeglasses
- Beard
- Age Range
- Sunglasses
- Moustache
- Mouth Open

OpenCV

One of the most well known open source facial recognition libraries currently available is Open Source Computer Vision Library (OpenCV). Unlike the other two mentioned in this project is in entirely free to use and there are no paid tiers for commercial use. Initially released in 1999, OpenCV is designed for both image recognition and real time facial detection from videos. It uses a cascade classifier system, called Viola-Jones Haarcascade, which is used to identify the occurrence of specific facial features. This works by overlaying image with the features, shown below in figure 2, the black or grey areas carry a negative weight whilst the white ones hold a positive weight. Each classifier is weakly weighted but multiple ones are placed together to create a stronger classifier. For example a nose or a mouth would still carry a weak weighting and must be combined with multiple features.

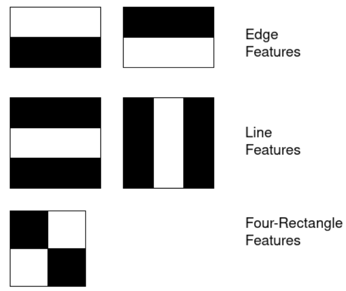

Figure 3: Haar Cascade Features

OpenCV also implements the cascade of classifiers technique which systematically increases the number of features it is searching for and if any stage fails then the image is not classified as containing a face. Unlike the other two used in this paper OpenCV does not offer comparison functionality in it's original state and must be heavily modified to fit this purpose and as a result this will not be investigated here.

1.2 Aim

The aim of this project is to look into some commonly used open source software; how the algorithms work, where they are being implemented and how effective various evasion techniques are against them. The guess of the author is that female faces will have a lower identification accuracy and that CV Dazzle makeup will prove to be entirely ineffective against all but OpenCV.

2 Procedure

2.1 Overview

In order to compare fairly each software was given the same group of images which contained the following categories:

- Control: no obstructions and hair out of face
- Minor obstructions: hair in face, hands in front of face and or wearing glasses
- Medium obstructions: facemask or sunglasses
- Major obstructions: face mask and sunglasses
- CV Dazzle

The forty-three images used in this report were collected from both volunteers and the model images from the CV Dazzle look book. There are 14 female and 18 male photos and the breakdown of them is as follows; 4 female and 5 male control, 4 female and 6 male minor obstructions, 3 female and 4 male medium obstructions and 3 female and 3 male major obstructions. This breakdown is not inclusive of the eleven images of CV Dazzle models, described below.

CV Dazzle Makeup

The final category for obstruction techniques being used was invented by Adam Harvey in 2010. Inspired heavily from World War One naval camouflage, this technique uses inverted contouring and colour blocking in order to confuse FRS by making it more difficult to distinguish facial features.*(CV Dazzle, 2020)*

2.2 Face++

Due to the time limit on access to the SDK the author chose to use the web interface version of this software as opposed to building it in a more local environment. Another, very time consuming, method that had to be implemented was the manual including of images one at a time to be processed by the software, as opposed to uploading all of them to FaceSet and automating the process.

Face Detection and Comparison

For the purpose of comparing this software to the others in this paper only two of the API's were used, as previously introduced in background. These are; Face Detection which determines whether a face is present in an image and Face Comparison that looks at two images in order to check if they contain the same person.

The web interface requires the user to upload images, one for detection and two for comparison, and it will then run them through the corresponding algorithms to produce a result and display it on screen.

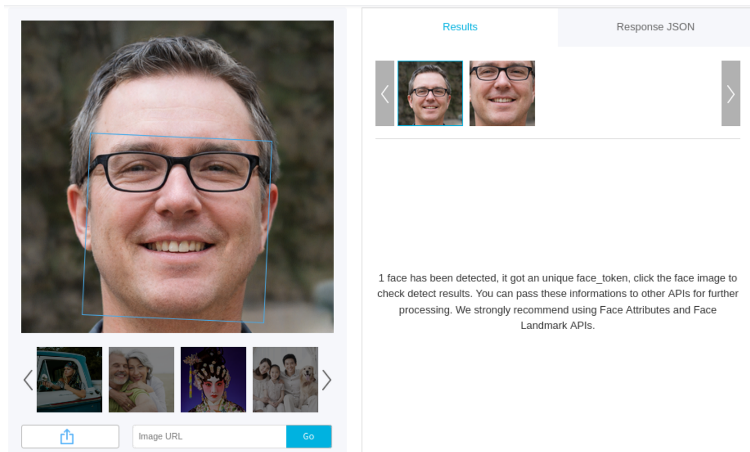

Figure 4: Face++ Detect Face Web Interface

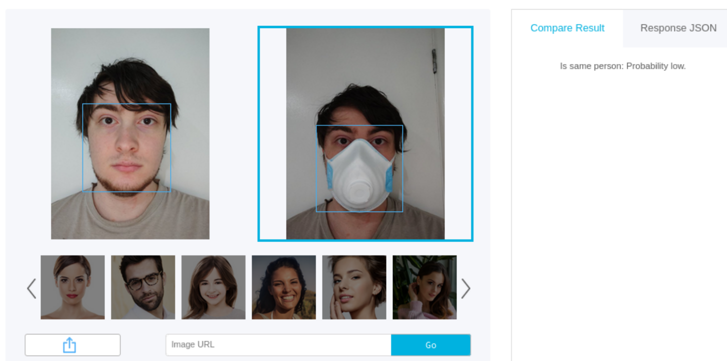

Figure 5: Face++ Face Comparing Web Interface

2.3 Rekognition

For this FRS Python 3 was chosen to develop the API in as the author was the most comfortable and had the most experience in this language. First the author created an Identity and Access Managment (IAM) user account with full access to the Rekognition API and downloaded the credentials to a local csv file. Then the next step was to install the corresponding Python package using the following command; `sudo pip boto3`.

Detection

The credentials for the IAM user must then be placed into the Python script, by reading from the csv file rather than hardcoding the values in the author did not have to censor any part of their code for this report. The images were iterated through using pythons os library, so that all 320 did not have to be referenced individually in the script. Next the credentials from the csv file are used to access the Rekognition Detect Face API and write the results to a text file.

```python
import csv
import boto3
import os

output = open('rekognitionResults.txt', 'w')

def OutputDetails(details):
    output.write(Face)
    for emotion in details['Emotions']:
        output.write("  {Type} : {Confidence}%".format(**emotion))

    for quality, value in details['Quality'].iteritems():
                output.write("  {quality} : {value}".format(quality=quality, value=value))

        for feature, data in details.iteritems():
                output.write("  {feature}({data[Value]}) : {data[Confidence]}%".format(feature=feature, data=data))

with open ('credentials.csv', 'r') as input:
    next(input)
    reader = csv.reader(input)
    for line in reader:
        keyID = line[2]
        secretKey = line[3]

client = boto3.client('rekognition', aws_access_key_id = keyID, aws_secret_access_key = secretKey)

for photo in os.listdir('../../Pictures/FRS'):
    with open(photo, 'rb') as sourcePic:
        picBytes = sourcePic.read()
        response = client.detect_faces(Image = {'Bytes': picBytes}, Attributes = ['ALL'])
        output.write(response['FaceDetails'])

output.close()
```

Comparison

The code for this part of the procedure is very similar but instead of one image, two are being passed through and the author instead called the comparison function. Due to the small sample size and need for the first, control, image to remain the same the image names were hard coded in and changed every time the script was ran. This use of hardcoded images and considerably smaller output removed the need for the `OutputDetails` class included in the script above.

```python
import csv
import boto3
import os, sys

output = open('rekognitionResults.txt', 'w')

with open ('credentials.csv', 'r') as input:
    next(input)
    reader = csv.reader(input)
    for line in reader:
        keyID = line[2]
        secretKey = line[3]

client = boto3.client('rekognition', aws_access_key_id = keyID, aws_secret_access_key = secretKey)

for photo in os.path.dirname(os.path.abspath('./FRS')):
    with open(photo, 'rb') as sourcePic:
        picBytes = sourcePic.read()
        response = client.detect_faces(Image = {'Bytes': picBytes}, Attributes = ['ALL'])
        output.write(response['FaceDetails'])

def OutputDetails(details):
    output.write(Face)
    for emotion in details['Emotions']:
        output.write("  {Type} : {Confidence}%".format(**emotion))

    for quality, value in details['Quality'].iteritems():
                output.write("  {quality} : {value}".format(quality=quality, value=value))

        for feature, data in details.iteritems():
                output.write("  {feature}({data[Value]}) : {data[Confidence]}%".format(feature=feature, data=data))


output.close()
```

2.4 OpenCV

As with Rekognition the tester decided to use the programming language Python. This program however required far more set-up before the script could be ran; as the `opencv-contrib-python`, `numpy` and `scikit-image` libraries had to be installed before being imported into the program. The latest OpenCV files was also downloaded onto the authors machine to allow access to the Haarcascade classifiers held on the `haarcascade\_frontalface\_default.xml file.`

In the python script itself the author first defines the classifier file to be used and passes through an iteration of the folder containing all of the images. Next the image is converted into greyscale in order for OpenCV to find features more easily. When the program has discovered a face on an image it places a rectangle around the face or faces and then appends a text file with the caption 'Face found in: photo name'.

```python
import cv2
import os, sys
import numpy as np
from skimage import io

output = open('openCVResults.txt', 'w')

def DetectFace(photo):
    faceCascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
    image = io.imread(photo, pilmode='RGB')
    grey = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

    faces = faceCascade.detectMultiScale(
        grey,
        scaleFactor=1.1,
        minNeighbors=5,
        minSize=(30, 30),
        flags = cv2.cv.CV_HAAR_SCALE_IMAGE
    )

    for (x, y, w, h) in faces:
        cv2.rectangle(image, (x, y), (x+w, y+h), (0, 255, 0), 2)
        output.write('Face found in:', photo)
        cv2.waitKey(0)


for photo in os.path.dirname(os.path.abspath('./FRS')):
    try:
        DetectFace(photo)
    except(ValueError, SyntaxError) as e:
        print(photo, "Error")

output.close()
```

3 Results

Rather than including all 32 of the image results only the first from each part has been included in the appendices. *(Appendices 4-8)* Again for brevity, the results included in this section of the report are not broken down to individual results but these can be found in Appendices 1-4. Overall all of the FRSs did very well and had a high accuracy rate even with facial coverings and other obfuscation techniques.

3.1 Face++

Face++ struggled with some of the images containing faces in both the 'medium obstructions' and 'major obstructions' categories, especially those of women. The software correctly identified 76% of both female and male faces, although it misgendered women 28% and incorrectly guessed their ages 12% more than men. With the CV Dazzle makeup Face++ struggled with identifying faces belonging to women much more than identifying those of men, even when fed considerably more female presenting images. An interesting result is that with one image the software returned that it contained two faces, one being the actual face and the other allegedly on the nose of the person in question. This image had the man's mouth and nose partially covered which suggests that this caused the erroneous result.

Gender	Obstruction	Guessed Age	Guessed Gender	Face found	Matches control
Female	None	80%	100%	100%	n/a
	Minor	60%	80%	100%	100%
	Medium	33%	66%	66%	66%
	Major	0%	25%	50%	50%
	CV Dazzle	n/a	12%	62%	n/a
Total		47%	48%	76%	75%
Male	None	100%	100%	100%	n/a
	Minor	57%	85%	71%	85%
	Medium	80%	100%	100%	80%
	Major	0%	40%	20%	0%
	CV dazzle	n/a	33%	66%	n/a
Total		59%	76%	76%	63%

3.2 Rekognition

Rekognition was able to detect that almost all of the images contained faces, although it did not succeed with one of the male faces containing a major obstruction. This obstruction was both sunglasses and a scarf which obscured both the mouth and nose of the wearer. There is still a great disparity between male and female when it comes to guessing of the age and gender, with the age gap being 10% higher than that of Megvii's software. Rekogition had very few problems with CV Dazzle makeup although it did prove to reduce the software's ability to determine the gender of the person.

Gender	Obstruction	Guessed Age	Guessed Gender	Face found	Matches control
Female	None	60%	100%	100%	n/a
	Minor	75%	100%	100%	100%
	Medium	66%	100%	100%	67%
	Major	50%	50%	100%	25%
	CV Dazzle	n/a	38%	100%	n/a
Total		59%	60%	100%	67%
Male	None	66%	100%	100%	n/a
	Minor	86%	86%	100%	100%
	Medium	100%	100%	100%	100%
	Major	60%	40%	80%	40%
	CV Dazzle	n/a	33%	100%	n/a
Total		81%	80%	96%	86%

3.3 OpenCV

Unfortunately the author was unable to complete their work on OpenCV and their python script did not manage to process any of the images provided. As a result of this there are no results for this software and it can only be compared to the other two algorithms on a theoretical basis.

4 Discussion

4.1 General Discussion

The author noticed that in all of the FRS's the accuracy for identifying female presenting faces was lower, Face++ having a difference of 24% for detecting faces. This includes Amazon's software misidentifying the genders of 15% of the images containing women as apposed to only misgendering 10% of male photos. Even if Rekognition was able to identify almost every image with a high degree of accuracy the almost 20% for identifying both age and gender are hard to ignore. On the surface this appears to be both a trivial matter and a very slight failure rate but many studies have demonstrated similar results. *(Georgetown Law Center on Privacy and Technology, 2016)* This includes a study by MIT researchers which concluded that the, at the time, most popular facial recognition algorithms performed significantly better on white men and performed worst on dark skinned female faces. *(Buolamwini and Gebru, 2018)* Another point on the matter is that a popular use of FRS is in law enforcement and if the algorithms cannot accurately detect female presenting or non-white persons then that could impact the utility of the software overall.

The increased use of FRS by large corporations and governments has been the source of much debate in recent years. Some have argued that widespread adoption of these softwares will limit the privacy of individuals and that it will eventually lead to mass surveillance. The concern here is about governmental censorship and a 'big brother' style scenario where no citizen has the right to privacy. On the other side of the debate is that FRS provides more benefits than the potential drawbacks. These benefits can include ease of travel as people may no longer require physical identification cards or passports and their face could be used instead. Corporations could also use this software to better predict a persons spending habits based on their age and emotion, which would allow for more targeted advertising.

By making the software, at least partially, free to use or open source it allows for a continuous peer review system as people will always be working on improving and testing the software for free. It also allows users to adapt the software easily for their needs and encourages more documentation and guides on how to use or alter the code. This in a large part is the reason for the popularity of OpenCV as it is very well documented and does not cost a thing to use.

4.2 Countermeasures

The most effective technique appeared to be covering the lower half of the face, especially the nose. Before the year 2020 it was uncommon to see a person, outside of Eastern Asia, covering that part of their face so in many ways it would make them stand out more if they were trying to evade facial recognition. Now due to the global pandemic it is encouraged that people in all countries should wear face masks and cover their mouth and nose which makes it considerably difficult for software to detect their faces. However a benefit of this is that more training data involving face coverings may soon become easily accessible. As a result this may make covering the face as an evasion technique ineffective soon.

Another potential countermeasure would be to 'poison' the training data so that the software will give inaccurate results. Poisoning the training data is when an attacker feeds the algorithm a series of data and teaches it that those images are what it is looking for so that when it receives an actual valid input it returns incorrect values. This is not limited to software that is currently being fed training data by it's developers as this sort of attack can even be done on pre-trained algorithms.

4.3 Conclusions

In conclusion, Rekognition appears to be the more accurate FRS in every single category investigated in this paper. However the sample size was small and the comparison was only between two, rather than the originally envisioned three, thus the results may not accurately depict the accuracy of these programs. The author did note that of the faces undetectable by both Face++ and Rekognition almost all of them involved the mouth and nose being covered in some shape or form, including Dazzle makeup. This may be indicative of a potential evasion technique that could work against popular facial recognition software, however as noted above this may soon become obsolete as training databases adapt to the modern climate and fashion trends.

4.4 Future Work

Due to the outbreak of COVID-19 the researcher found it difficult to get participants for the practical element of this study, so had to resort to using images gathered from volunteers and the models from Adam Harvey's original study into CV Dazzle makeup. This resulted in the images being of predominantly white men under 30. In the future they would like to repeat the study with a larger group of ethnicities, ages and genders in order to determine if the current results would match those of a more inclusive sample size.

There are other evasion techniques that have been developed that would be interesting to test in order to examine their effusiveness. Examples include Adam Harvey's Hyperface Glamoflage pattern and Shariff et al's patterned glasses, as shown in the figures below. *(AH Projects, 2018)* and *(Shariff et al, 2016)*

Figure 6: Hyperface Pattern

Figure 7: Shariff et al Glasses

Finally since OpenCV was not implemented successfully the author would like to continue working with the C2 python library and attempt to get it working successfully. The entire procedure of this paper could then be reattempted with all three softwares and larger much more varied data, thus allowing for more accurate results.

5 References

AH Projects (2018). *HyperFace* [online]. Available at: https://ahprojects.com/hyperface/ (Accessed 17 March)

Buolamwini.J and Buolamwini.T (2018). *Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification* [PDF]. Massachusetts: Massachusetts Institute of Technology. (Accessed 16 May)

CV Dazzle (2020). *CV Dazzle* [online]. Available at: https://cvdazzle.com/ (Accessed 17 March)

Forbes(2019). *The Amazing Ways Chinese Face Recognition Company Megvii (Face++) Uses AI And Machine Vision* [online]. Available at: https://www.forbes.com/sites/bernardmarr/2019/05/24/the-amazing-ways-chinese-face-recognition-company-megvii-face-uses-ai-and-machine-vision/ (Accessed 16 March)

FaceX (2020). *The Best Face Recognition SDKs & APIs for you* [online]. Available at: https://facex.io/ (Accessed 12 May)

Face++ (2020). *Doc Center* [online]. Available at: https://console.faceplusplus.com/documents/5679127 (Accessed 20 March)

Georgetown Law Center on Privacy and Technology (2016) *Unregulated Police Face Recognition in America* [online]. Available at: https://www.perpetuallineup.org/ (Accessed 16 May)

Science ABC (2019). *How Does Facial Recognition Work?* [online]. Available at: https://www.scienceabc.com/innovation/facial-recognition-works.html (Accessed 16 March)

Shariff et al (2016). *Accessorize to a Crime: Real and Stealthy Attacks on State-of-the-Art Face Recognition* [PDF]. Pittsburgh:Carnegie Mellon University. Available at: https://www.cs.cmu.edu/ sbhagava/papers/face-rec-ccs16.pdf (Accessed 17 March)

Techopedia (2019). *Facial Recognition* [online]. Available at:https://www.techopedia.com/definition/32071/facial-recognition (Accessed 16 March)

Towards Data Science (2019). *Poisoning attacks on Machine Learning* [online]. Available at: https://towardsdatascience.com/poisoning-attacks-on-machine-learning-1ff247c254db (Accessed 17 May)

Y. Bengio (2009). *Learning Deep Architectures for AI* [PDF]. Canada:Université de Montréal. Available at: https://www.iro.umontreal.ca/ lisa/pointeurs/TR1312.pdf (Accessed 17 March)

6 Acronyms Used

Facial Recognition Software (FRS)
Artifical Neural Network (ANN)
Deep Neural Network (DNN)
Application Programming Interface (API)
Software Development Kit (SDK)
Convolutional Neural Network (CNN)
Identity and Access Managment (IAM)
Open Source Computer Vision Library (OpenCV)

7 Appendices

7.1 Face++ Male Results

	Obstruction	**Guessed Age**	**Guessed Gender**	**Face found**	**Matches control**
Male1	Control	yes	yes	yes	n/a
	Minor1	yes	yes	yes	yes
	Minor2	yes	no	yes	yes
	Medium	yes	yes	yes	no
	Major	n/a	n/a	no	n/a
Male2	Control	yes	yes	yes	n/a
	Minor	yes	yes	yes	yes
	Medium	yes	yes	yes	yes
Male3	Control	yes	yes	yes	n/a
	Minor	no	yes	yes	yes
	Medium	yes	yes	yes	yes
	Major	n/a	n/a	no	n/a
Male4	Control	yes	yes	yes	n/a
	Minor	yes	yes	yes	yes
	Major	no	yes	yes (two faces)	no
Male5	Control	yes	yes	yes	n/a
	Minor	no	yes	yes	yes
	Medium	no	yes	yes	yes
Male6	Control	yes	yes	yes	n/a
	Minor	n/a	n/a	no	n/a
	Medium	yes	yes	yes	yes
	Major1	n/a	n/a	no	n/a
	Major2	n/a	n/a	no	n/a

7.2 Face++ Female Results

	Obstruction	Guessed Age	Guessed Gender	Face found	Matches control
Female1	Control	yes	yes	yes	n/a
	Minor	no	yes	yes	yes
	Medium	n/a	n/a	no	n/a
Female2	Control	yes	yes	yes	n/a
	Minor	yes	no	yes	yes
	Medium	yes	yes	yes	yes
Female3	Control	yes	yes	yes	n/a
	Minor	yes	yes	yes	yes
	Medium	no	yes	yes	yes
	Major	n/a	n/a	no	n/a
Female4	Control	yes	yes	yes	n/a
	Minor	yes	yes	yes	yes
	Major1	n/a	n/a	no	n/a
	Major2	no	no	yes	yes
Female5	Control	no	yes	yes	n/a
	Minor	no	yes	yes	yes
	Major	no	yes	yes	yes

7.3 Face++ CV Dazzle Results

	Guessed Gender	Face found
Female1	n/a	no
Female2	n/a	no
Female3	no	yes
Female4	n/a	no
Female5	yes	yes
Female6	no	yes
Female7	no	yes
Female8	no	yes
Male1	no	yes
Male2	n/a	no
Male3	yes	yes

7.4 Rekognition Female Results

	Obstruction	Guessed Age	Guessed Gender	Face found	Matches control
Female1	Control	yes	yes	yes	n/a
	Minor	no	yes	yes	yes
	Medium	no	yes	yes	no
Female2	Control	yes	yes	yes	n/a
	Minor	yes	yes	yes	yes
	Medium	yes	yes	yes	yes
Female3	Control	no	yes	yes	n/a
	Minor	yes	yes	yes	yes
	Medium	yes	yes	yes	yes
	Major	yes	yes	yes	no
Female4	Control	yes	yes	yes	n/a
	Minor	no	yes	yes	yes
	Major1	no	no	yes	no
	Major2	no	no	yes	no
Female5	Control	no	yes	yes	n/a
	Minor	yes	yes	yes	yes
	Major	yes	yes	yes	yes

7.5 Rekognition Male Results

	Obstruction	Guessed Age	Guessed Gender	Face found	Matches control
Male1	Control	yes	yes	yes	n/a
	Minor1	yes	yes	yes	yes
	Minor2	yes	yes	yes	yes
	Medium	yes	yes	yes	yes
	Major	no	no	yes	no
Male2	Control	yes	yes	yes	n/a
	Minor	yes	yes	yes	yes
	Medium	yes	yes	yes	yes
Male3	Control	no	yes	yes	n/a
	Minor	no	yes	yes	yes
	Medium	yes	yes	yes	yes
	Major	yes	yes	yes	yes
Male4	Control	yes	yes	yes	n/a
	Minor	yes	yes	yes	yes
	Major	yes	no	yes	yes
Male5	Control	yes	yes	yes	n/a
	Minor	yes	yes	yes	yes
	Medium	yes	yes	yes	yes
Male6	Control	no	yes	yes	n/a
	Minor	yes	no	yes	yes
	Medium	yes	yes	yes	yes
	Major1	yes	yes	yes	no
	Major2	n/a	n/a	no	n/a

7.6 Rekognition CV Dazzle Results

	Guessed Gender	Face found
Female1	no	yes
Female2	no	yes
Female3	yes	yes
Female4	yes	yes
Female5	no	yes
Female6	yes	yes
Female7	no	yes
Female8	no	yes
Male1	no	yes
Male2	yes	yes
Male3	no	yes

7.7 Output from Rekognition Face Detection

```
Face (99.99979400634766%)
CONFUSED :  0.2623405456542969%
ANGRY : 0.27865070104599%
HAPPY : 0.011549402959644794%
CALM : 97.8419189453125%
SAD : 1.416614055633545%
FEAR : 0.024414271116256714%
SURPRISED : 0.10043074190616608%
DISGUSTED : 0.064079150557518%
AgeRange(Low) : 14,
AgeRange(High) : 26
Smile(False): 99.94913482666016%
Eyeglasses(False) : 99.02244567871094%
Sunglasses(False) : 99.75312042236328%
Gender(Male) : 99.0409927368164%
Beard(False) : 79.76945495605469%
Mustache(False) : 99.21092987060547%
EyesOpen(True) : 99.62873840332031%
MouthOpen(False) : 99.88142395019531%
```

7.8 Output from Rekognition Face Comparison

```
Source Face (98.951602417%)
Target Face (94.991434896%)
Similarity : 96.9%
```

7.9 Output from Face++ Face Detection


```
{"request_id": "1589574847,03030217-c13e-4464-85b1-a00a7494be77",
  "time_used": 66,
  "faces": [
    {
      "face_token": "4242a49034992bdbec61d7712c6d5ec0",
      "face_rectangle": {
        "top": 244,
        "left": 104,
        "width": 280,
        "height": 280
      },
      "landmark": {
        "contour_chin": {
          "x": 246,
          "y": 524
        },
        "contour_left1": {
          "x": 119,
          "y": 304
        },
        "contour_left2": {
          "x": 118,
          "y": 335
        },
        "contour_left3": {
          "x": 119,
          "y": 366
        },
        "contour_left4": {
          "x": 123,
          "y": 397
        },
        "contour_left5": {
          "x": 128,
          "y": 429
        },
        "contour_left6": {
          "x": 139,
          "y": 458
        },
        "contour_left7": {
          "x": 157,
          "y": 483
        },
        "contour_left8": {
          "x": 181,
          "y": 504
        },
        "contour_left9": {
          "x": 211,
          "y": 518
        },
        "contour_right1": {
          "x": 372,
          "y": 308
        },
        "contour_right2": {
          "x": 372,
          "y": 338
        },
        "contour_right3": {
          "x": 370,
          "y": 368
        },
        "contour_right4": {
          "x": 366,
          "y": 399
        },
        "contour_right5": {
          "x": 361,
          "y": 430
        },
        "contour_right6": {
          "x": 350,
          "y": 459
        },
        "contour_right7": {
          "x": 332,
          "y": 483
        },
        "contour_right8": {
          "x": 309,
          "y": 503
        },
        "contour_right9": {
          "x": 281,
          "y": 518
        },
        "left_eye_bottom": {
          "x": 183,
          "y": 310
        },
        "left_eye_center": {
          "x": 185,
          "y": 300
        },
        "left_eye_left_corner": {
          "x": 159,
          "y": 299
        },
        "left_eye_lower_left_quarter": {
          "x": 169,
          "y": 306
        },
        "left_eye_lower_right_quarter": {
          "x": 199,
          "y": 308
        },
        "left_eye_pupil": {
          "x": 187,
          "y": 297
        },
        "left_eye_right_corner": {
          "x": 212,
          "y": 305
        },
        "left_eye_top": {
          "x": 185,
          "y": 287
        },
        "left_eye_upper_left_quarter": {
          "x": 170,
          "y": 290
        },
        "left_eye_upper_right_quarter": {
          "x": 201,
          "y": 292
        },
```

```
"left_eyebrow_left_corner": {
  "x": 134,
  "y": 270
},
"left_eyebrow_lower_left_quarter": {
  "x": 153,
  "y": 260
},
"left_eyebrow_lower_middle": {
  "x": 174,
  "y": 259
},
"left_eyebrow_lower_right_quarter": {
  "x": 196,
  "y": 266
},
"left_eyebrow_right_corner": {
  "x": 219,
  "y": 269
},
"left_eyebrow_upper_left_quarter": {
  "x": 149,
  "y": 248
},
"left_eyebrow_upper_middle": {
  "x": 175,
  "y": 243
},
"left_eyebrow_upper_right_quarter": {
  "x": 202,
  "y": 249
},
"mouth_left_corner": {
  "x": 195,
  "y": 442
},
"mouth_lower_lip_bottom": {
  "x": 244,
  "y": 464
},
"mouth_lower_lip_left_contour1": {
  "x": 220,
  "y": 443
},
"mouth_lower_lip_left_contour2": {
  "x": 208,
  "y": 452
},
"mouth_lower_lip_left_contour3": {
  "x": 224,
  "y": 460
},
"mouth_lower_lip_right_contour1": {
  "x": 269,
  "y": 443
},
"mouth_lower_lip_right_contour2": {
  "x": 280,
  "y": 453
},
"mouth_lower_lip_right_contour3": {
  "x": 264,
  "y": 461
},
"mouth_lower_lip_top": {
  "x": 244,
  "y": 444
},
"mouth_right_corner": {
  "x": 295,
  "y": 443
},
"mouth_upper_lip_bottom": {
  "x": 244,
  "y": 444
},
"mouth_upper_lip_left_contour1": {
  "x": 232,
  "y": 428
},
"mouth_upper_lip_left_contour2": {
  "x": 213,
  "y": 434
},
"mouth_upper_lip_left_contour3": {
  "x": 220,
  "y": 442
},
"mouth_upper_lip_right_contour1": {
  "x": 257,
  "y": 429
},
"mouth_upper_lip_right_contour2": {
  "x": 276,
  "y": 435
},
"mouth_upper_lip_right_contour3": {
  "x": 269,
  "y": 443
},
"mouth_upper_lip_top": {
  "x": 245,
  "y": 432
},
"nose_contour_left1": {
  "x": 225,
  "y": 305
},
"nose_contour_left2": {
  "x": 215,
  "y": 360
},
"nose_contour_left3": {
  "x": 221,
  "y": 392
},
"nose_contour_lower_middle": {
  "x": 240,
  "y": 399
},
"nose_contour_right1": {
  "x": 263,
  "y": 307
},
"nose_contour_right2": {
  "x": 272,
  "y": 362
},
```

```
        "nose_contour_right3": {
          "x": 261,
          "y": 394
        },
        "nose_left": {
          "x": 206,
          "y": 382
        },
        "nose_right": {
          "x": 280,
          "y": 385
        },
        "nose_tip": {
          "x": 240,
          "y": 375
        },
        "right_eye_bottom": {
          "x": 312,
          "y": 312
        },
        "right_eye_center": {
          "x": 310,
          "y": 303
        },
        "right_eye_left_corner": {
          "x": 281,
          "y": 307
        },
        "right_eye_lower_left_quarter": {
          "x": 296,
          "y": 310
        },
        "right_eye_lower_right_quarter": {
          "x": 327,
          "y": 309
        },
        "right_eye_pupil": {
          "x": 311,
          "y": 299
        },
        "right_eye_right_corner": {
          "x": 338,
          "y": 301
        },
        "right_eye_top": {
          "x": 310,
          "y": 290
        },
        "right_eye_upper_left_quarter": {
          "x": 294,
          "y": 295
        },
        "right_eye_upper_right_quarter": {
          "x": 326,
          "y": 293
        },
        "right_eyebrow_left_corner": {
          "x": 272,
          "y": 267
        },
        "right_eyebrow_lower_left_quarter": {
          "x": 294,
          "y": 266
```

```
        },
        "right_eyebrow_lower_middle": {
          "x": 316,
          "y": 264
        },
        "right_eyebrow_lower_right_quarter": {
          "x": 337,
          "y": 267
        },
        "right_eyebrow_right_corner": {
          "x": 358,
          "y": 273
        },
        "right_eyebrow_upper_left_quarter": {
          "x": 290,
          "y": 251
        },
        "right_eyebrow_upper_middle": {
          "x": 316,
          "y": 247
        },
        "right_eyebrow_upper_right_quarter": {
          "x": 341,
          "y": 254
        }
      },
      "attributes": {
        "gender": {
          "value": "Male"
        },
        "age": {
          "value": 22
        },
        "smile": {
          "value": 0.174,
          "threshold": 50
        },
        "headpose": {
          "pitch_angle": 4.2951937,
          "roll_angle": 1.0409329,
          "yaw_angle": 3.9633303
        },
        "blur": {
          "blurness": {
            "value": 0.41,
            "threshold": 50
          },
          "motionblur": {
            "value": 0.41,
            "threshold": 50
          },
          "gaussianblur": {
            "value": 0.41,
            "threshold": 50
          }
        },
        "eyestatus": {
          "left_eye_status": {
            "no_glass_eye_open": 99.999,
            "no_glass_eye_close": 0,
            "normal_glass_eye_open": 0.001,
            "normal_glass_eye_close": 0,
            "dark_glasses": 0,
            "occlusion": 0
```

```
        },
        "right_eye_status": {
          "no_glass_eye_open": 99.997,
          "no_glass_eye_close": 0,
          "normal_glass_eye_open": 0.003,
          "normal_glass_eye_close": 0,
          "dark_glasses": 0,
          "occlusion": 0
        }
      },
      "facequality": {
        "value": 91.443,
        "threshold": 70.1
      },
      "glass": {
        "value": "None"
      }
    }
  }
 ],
 "image_id": "LobtMdbKtjrf+MgeDjPUXA==",
 "face_num": 1
}
```


7.10 Output from Face++ Face Comparing


```
{"faces1": [
    {
      "face_rectangle": {
        "width": 280,
        "top": 244,
        "left": 104,
        "height": 280
      },
      "face_token": "7658e086e26efdd16fbb557de645aab5"
    }
  ],
  "faces2": [
    {
      "face_rectangle": {
        "width": 276,
        "top": 313,
        "left": 98,
        "height": 276
      },
      "face_token": "47388363c91c4e0e0ea8639abe6d56de"
    }
  ],
  "time_used": 427,
  "thresholds": {
    "1e-3": 62.327,
    "1e-5": 73.975,
    "1e-4": 69.101
  },
  "confidence": 61.954,
  "image_id2": "RtpYvmLPIUVr0VOombn8wA==",
  "image_id1": "LobtMdbKtjrf+MgeDjPUXA==",
  "request_id": "1589565764,cda67fce-cd93-4e65-93df-de50bd1aa1ea"}
```