Group 1

Face Recognition Attack Group

Overview

1. Attacking Methods

- Images/Video
- Masks
- Physical print-outs
- Haar-Like Features

2. Analysis and Metrics

- Threshold values
- AUC/ROC

3. Prevention

- 3D camera
- Eye-blink detection
- Challenge-response method

4. What we Learned

- Deciding thresholds
- Artificial face creation

Attacking Methods

Images

Video Game Character Creator

- Popular Celebrities modeled using the Elden Ring character creator

Online Images

- Attacking systems with images of celebrities found online
- From file, iPad, print-outs

Masks

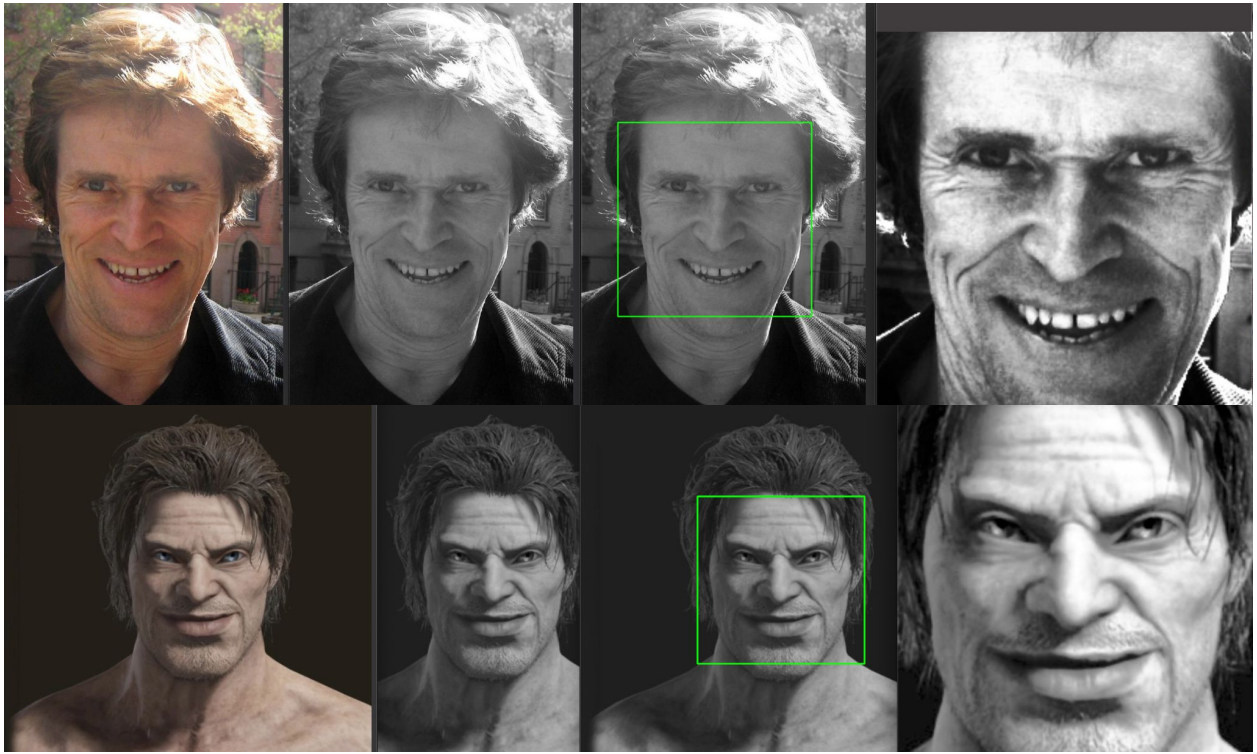
- Projecting face images onto an LED screen mask

Video Game Character Creator

Samuel L Jackson

Angular Distance: 1.23

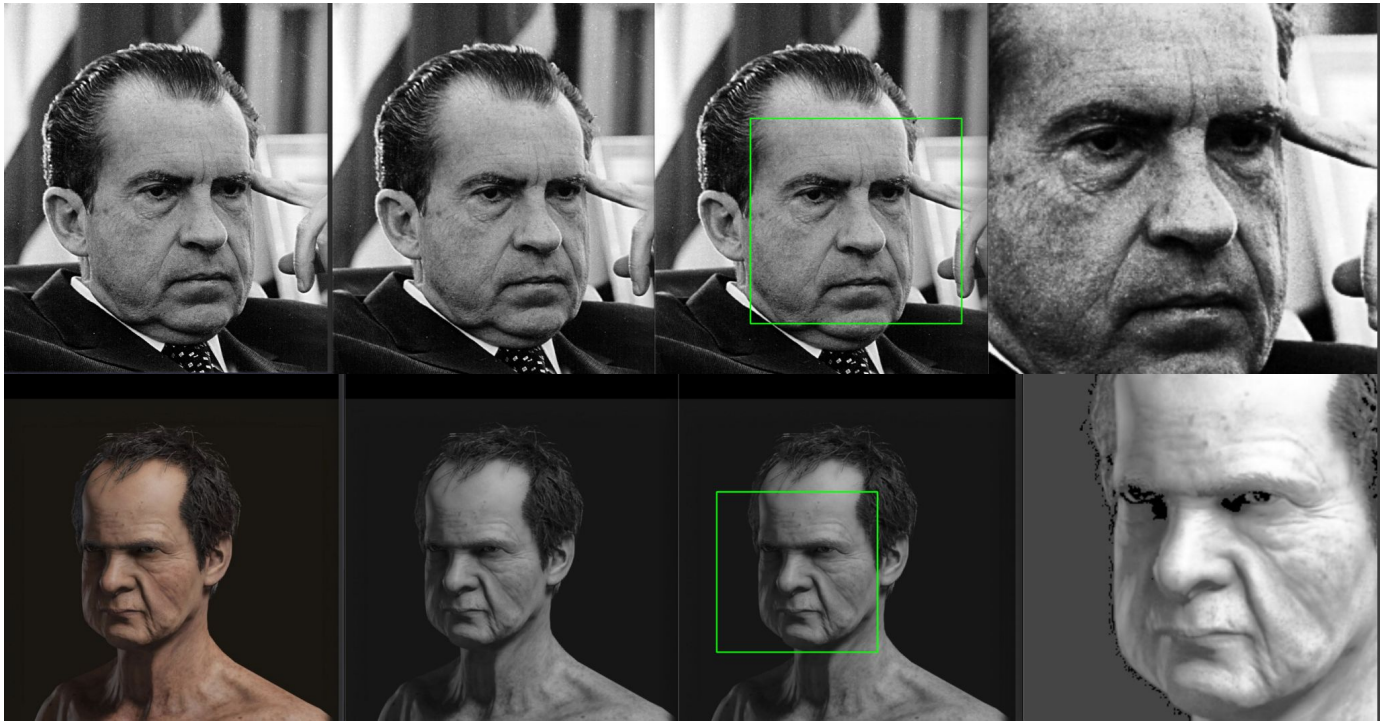
Video Game Character Creator



Willem Dafoe

Angular Distance: 1.38

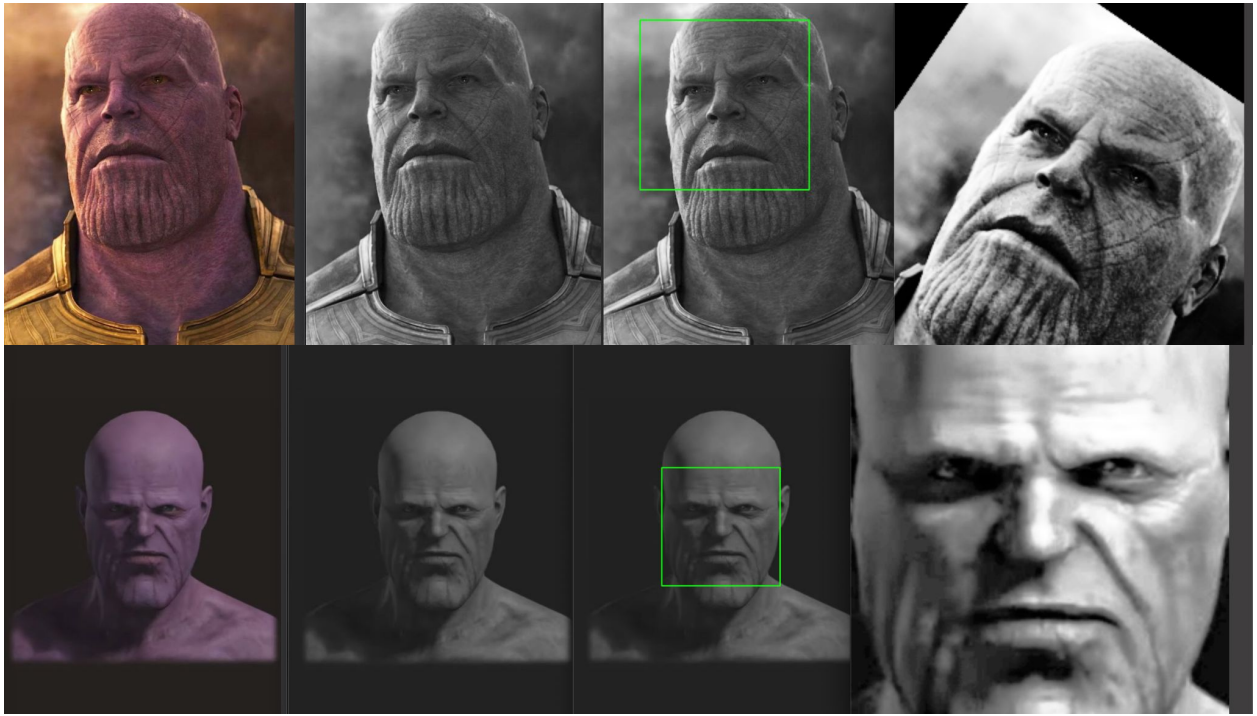
Video Game Character Creator



Richard Nixon

Angular Distance: 1.38

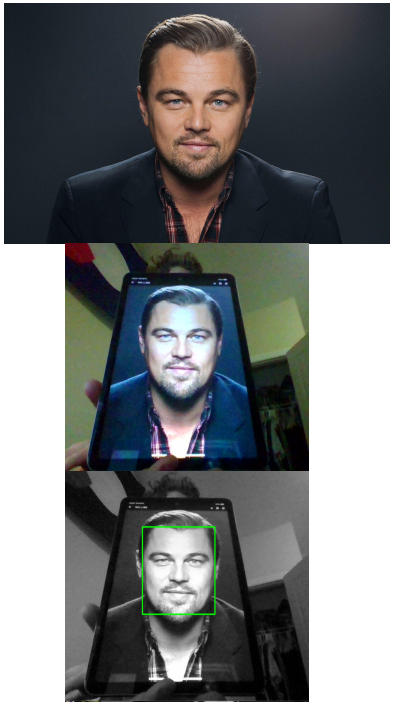
Video Game Character Creator



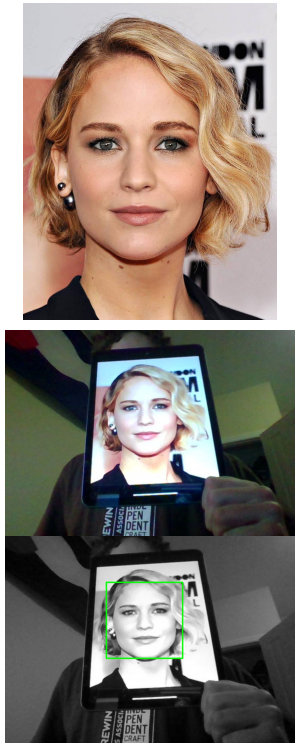
Thanos

Angular Distance: 1.57

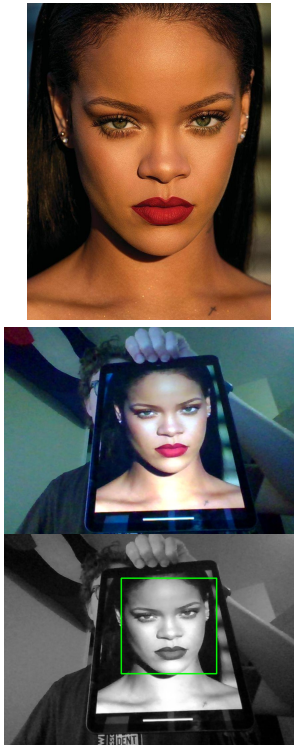
iPad



Distance: 0.435587

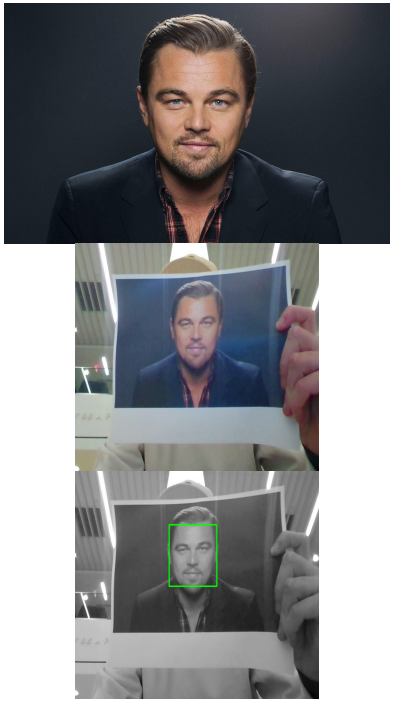


Distance: 0.660952



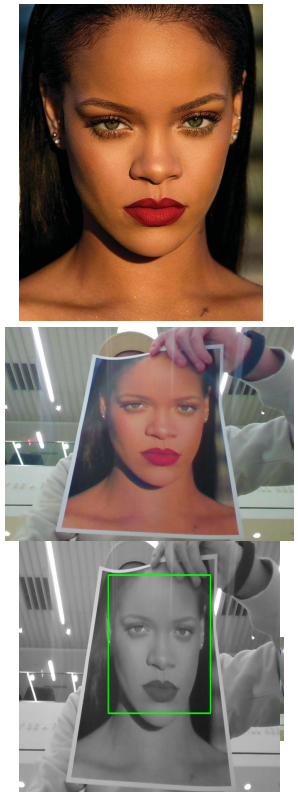
Distance: 0.444585

Color Print-Out



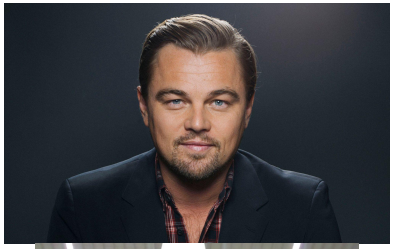
Distance: 0.217150

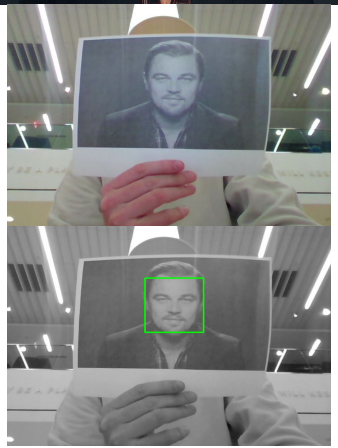
Distance: 0.466290



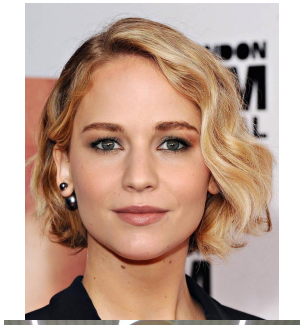
Distance: 0.965169

B&W Print-Out

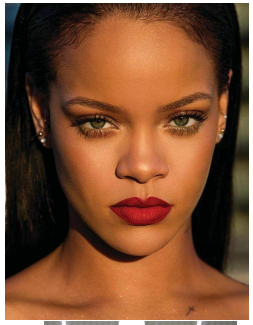


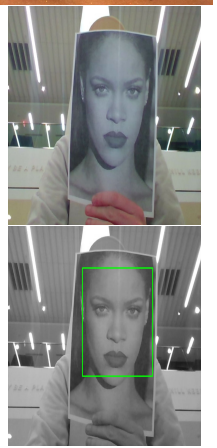


Distance: 0.209230



Distance: 0.858021





Distance: 0.520563

Presentation Attack Results

Subject	iPad	Color Print-Out	Black & White Print-Out
Leonardo DiCaprio	0.435587	0.217150	0.209230
Jennifer Lawrence	0.660952	0.466290	0.858021
Rihanna	0.444585	0.965169	0.520563

Key Findings

Orient face with camera angle

Remove all possible glare sources

Color vs Black & White does not matter as much as camera quality

LED Mask Attack

Genuine Attack

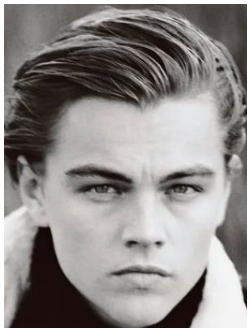
- Angular Distance: 1.79

Imposter Attack

- Angular Distance: 1.83

Online Images

Testing on Leonardo Dicaprio



Angular Distances:

0.69

0.26

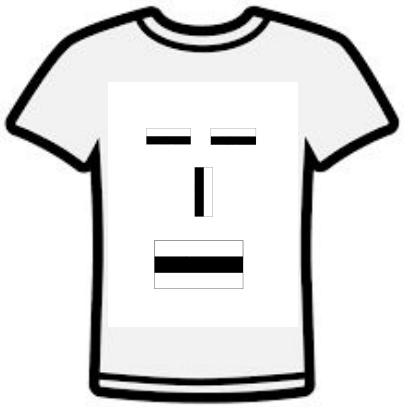
0.24

1.23

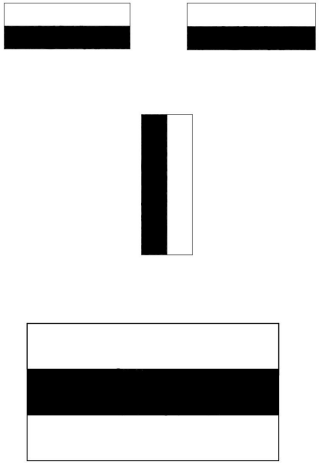
1.21

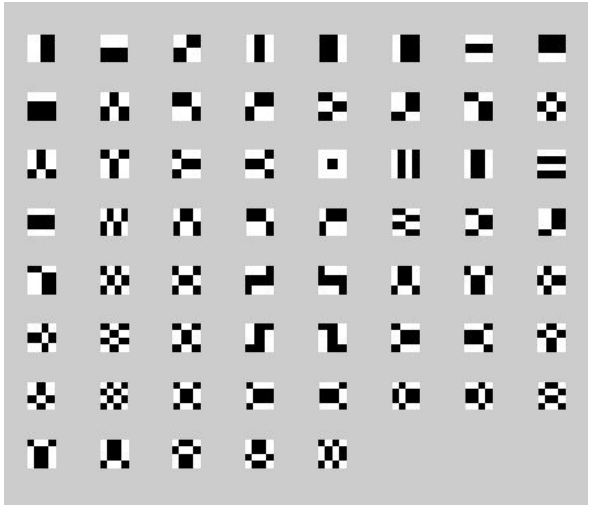
Haar-Like Features

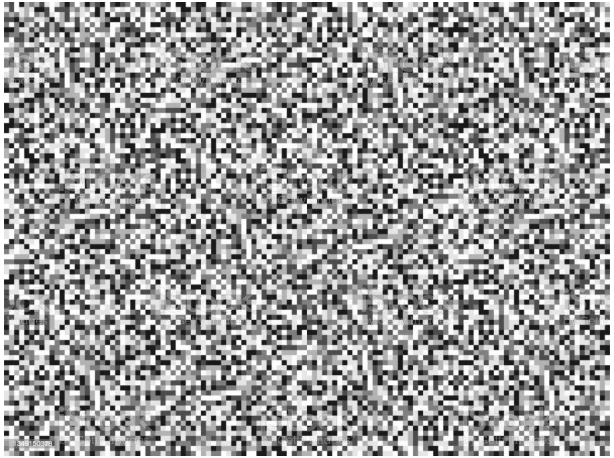
- Motivation
  - Sometimes you don’t want to be enrolled in facial recognition system.
  - e.g Hong Kong
- Idea
  - Generate features that Viola Jones will recognize as a face, so it won’t focus on your actual face.
- Tried using rectangular white noise, as well as custom Haar-Like Features to try and trick the system
- If it works, one can imagine a piece of clothing or accessories to try and trick these types of systems.



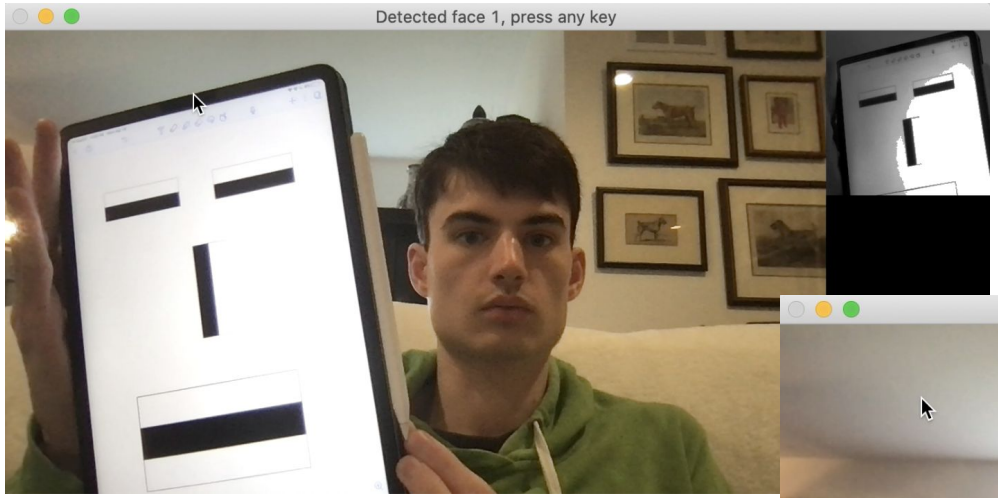
Haar-Like Features cont.





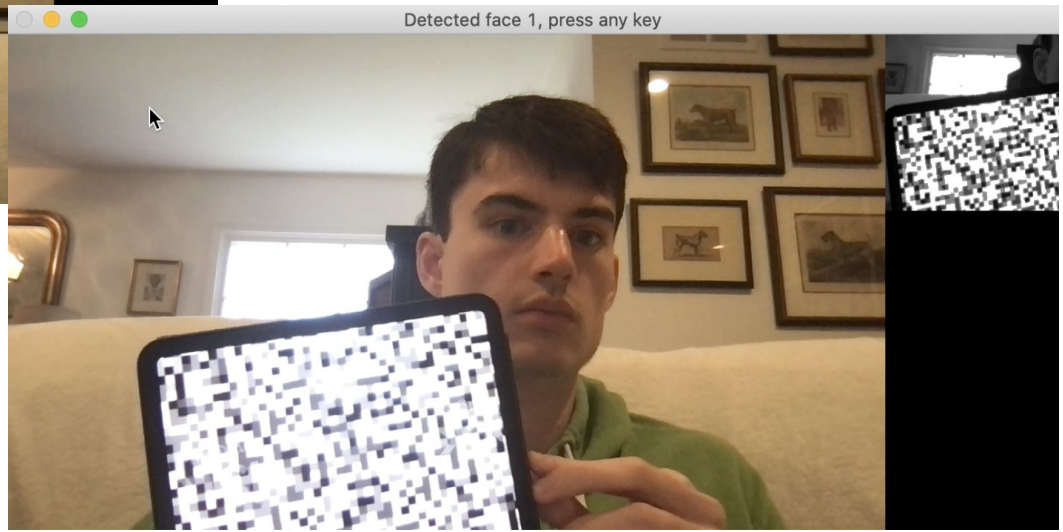


Haar-Like Features cont.

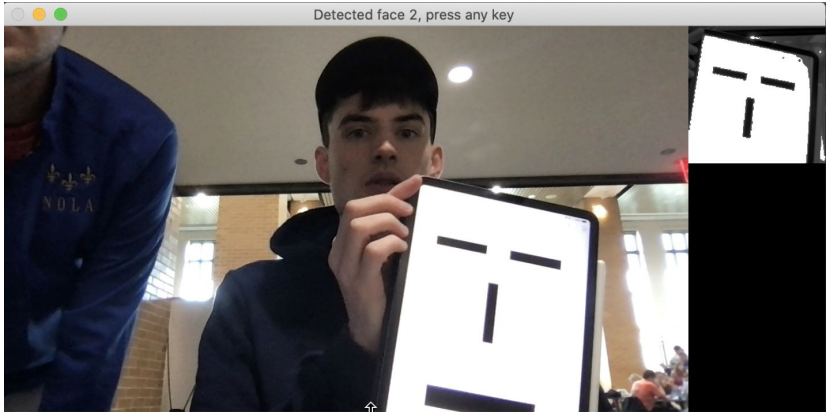


Worked much better, more consistent recognition

Not as consistent, would flip between my face and the noise.



Haar Like Features cont.

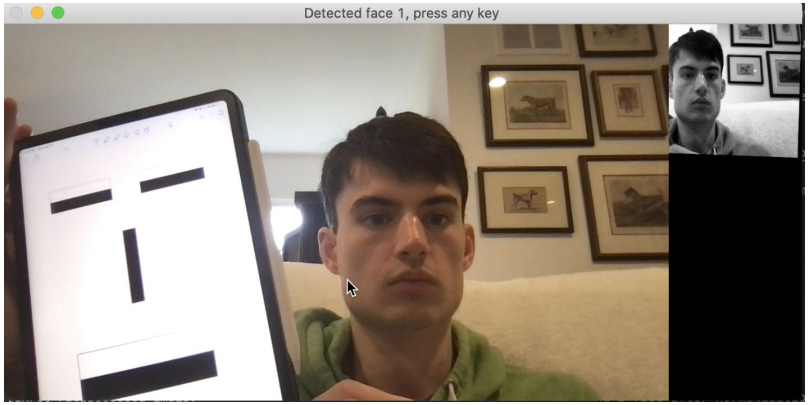


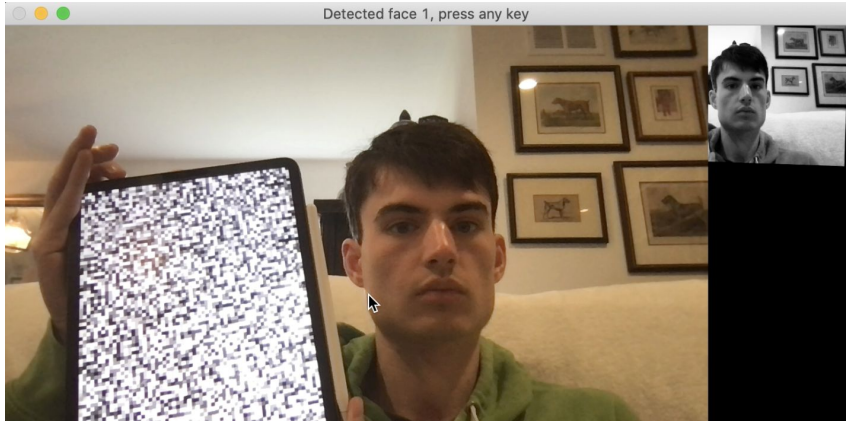
Tried using Haar Like features to trick system into thinking me and my friend are the same person.

Distance = .227, way below threshold.

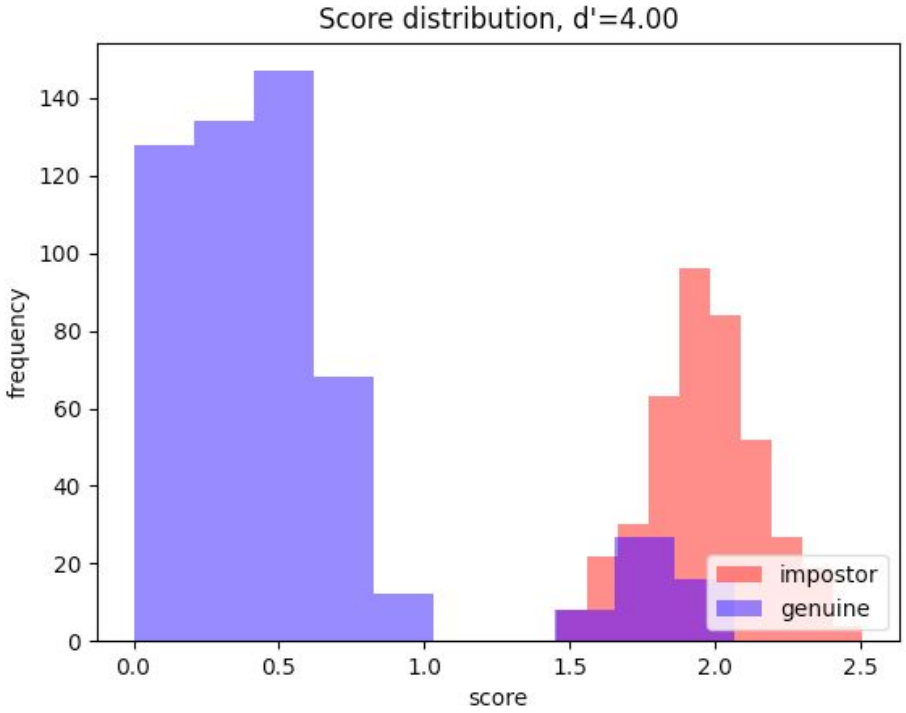
Haar-Like Features Prevention

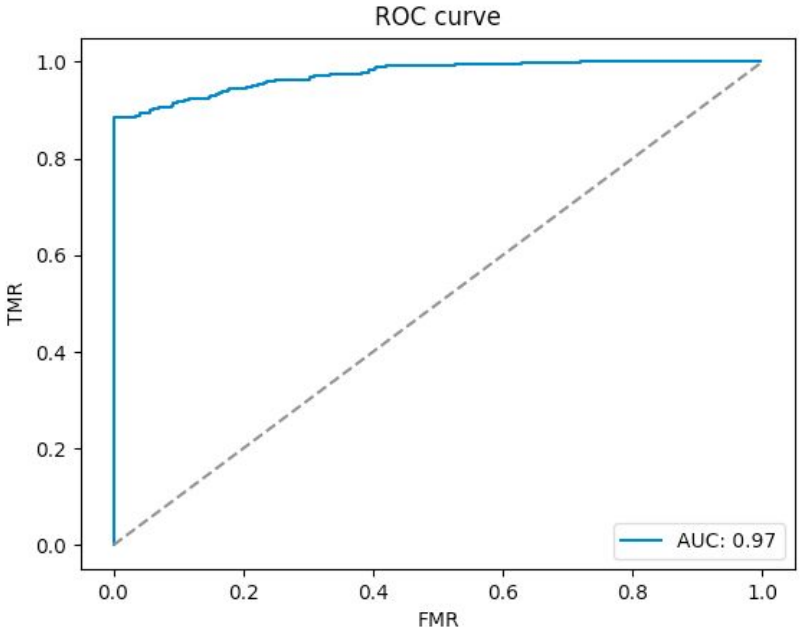
- While Viola-Jones detection is fast, we can prevent this type of attack by using the DLIB CNN face and landmark detector.
- Much slower, but seems robust to these types of attacks.





Analysis and Metrics





441 Genuine pairs and 405 imposter pairs

EER at angular distance threshold of 1.7. With FMR = FNMR = 0.091

AUC = 0.97

Prevention

source: https://hcis-journal.springeropen.com/articles/10.1186/s13673-018-0157-2

3D Camera

acquisition

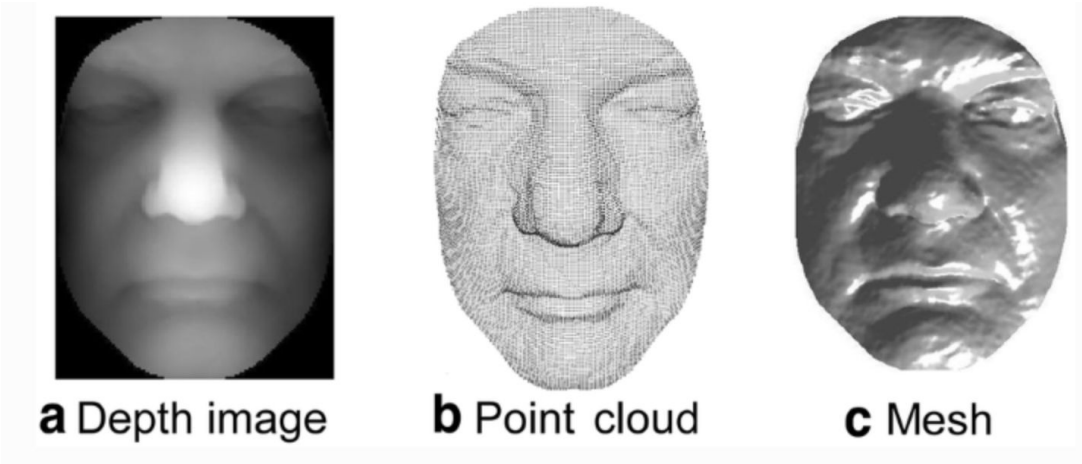
- triangulation
  - scanner measures the emitting and receiving angles of infrared beams and create a facial map using the "reflection points" of these beams
  - slow but precise
- structured light
  - emit light pattern or grid to target face
  - measure this light pattern's projection to calculate the surface's shape
  - faster but less accurate (more popular on consumer level)

source: https://hcis-journal.springeropen.com/articles/10.1186/s13673-018-0157-2

3D Camera

preprocessing

- geometric transformations to "center" face on camera's axis
- then use facial features like nose and eyes in order to isolate facial features, instead of features like hats, jewelry, and eye-glasses
- then interpret in 3 different formats:



source: https://hcis-journal.springeropen.com/articles/10.1186/s13673-018-0157-2

3D Camera

feature extraction, database, and matching

- global approach
  - entire face as feature vector that is stored in database and matched to other complete faces
- component-based approach
  - store and match individual parts like eyes, nose, and mouth individually
- hybrid combines these two
  - most accurate, but most costly

source: https://towardsdatascience.com/anti-spoofing-techniques-for-face-recognition-solutions-4257c5b1dfc9

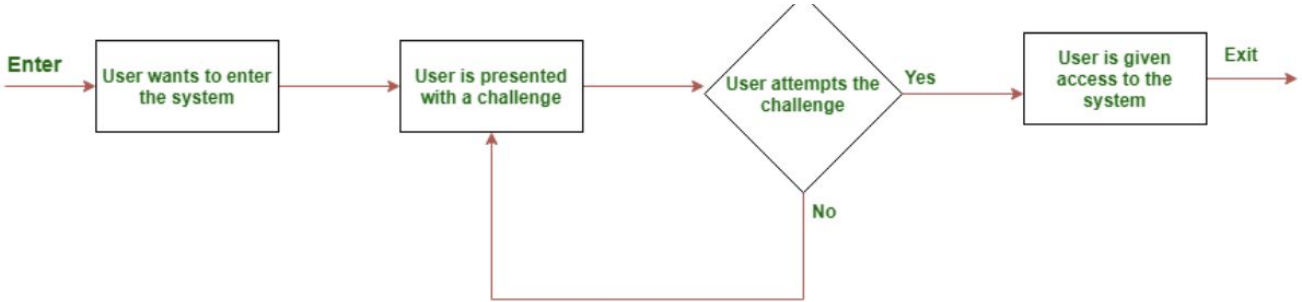
Eye Blink Detection

- simple method to determine liveliness of a face:
  - avg. human: 15-30 blinks/min
  - eyes shut for ~250 ms/blink
  - while using video, find frames with closed eyes → count consecutive frames with closed eyes → compare to average numbers
  - can do so using open source libraries similar to what we use

source: https://www.techtarget.com/searchsecurity/definition/challenge-response-system

Challenge-Response

- Facial expressions
  - demand users to make a facial expression or action like smile, frown, etc.
  - depending on complexity of software & hardware, possible to detect mask-like tendencies and actions
- Password authentication accompaniment
  - attacker can be required to supplement facial match with a password, code, or any other form of authentication



What We Learned

What We Learned

- Equilibrium threshold is not necessarily best for facial recognition
  - Can reduce the threshold significantly to far decrease FMR without a significant impact to FNMR
  - Constraining acquisition conditions (i.e. lighting, glasses, etc) will create a far more robust system as the overlap between genuine and impostor scores will be far smaller
- It is easy to make a fake face, it is hard to make a really good fake face
  - Our system has no liveness detection and is not particularly robust but still handled some of the attacks very well, introducing small hardware upgrades would vastly improve the system
- AI based face id is very good but not perfect
  - Locked out due to haircut

Questions?

Group 2

Face(mask) Recognition

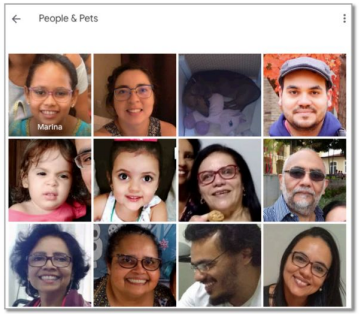
Rafael Mendizabal
Calvin Kusek

Motivation

- Facial recognition has become a common part of our lives and our interactions with technology, which can be seen through facial recognition software used by phones and things like Google Photos
- Many of us (iPhone Users) had issues with unlocking our phones during the pandemic because of masks, as masks block key features of our face
- We wanted to explore solutions to using facial recognition technology with masks

Personal devices

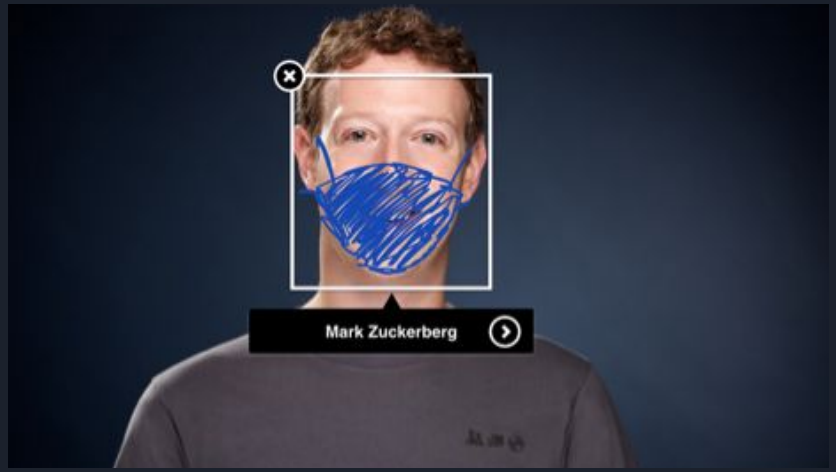
Facebook



Google Photos

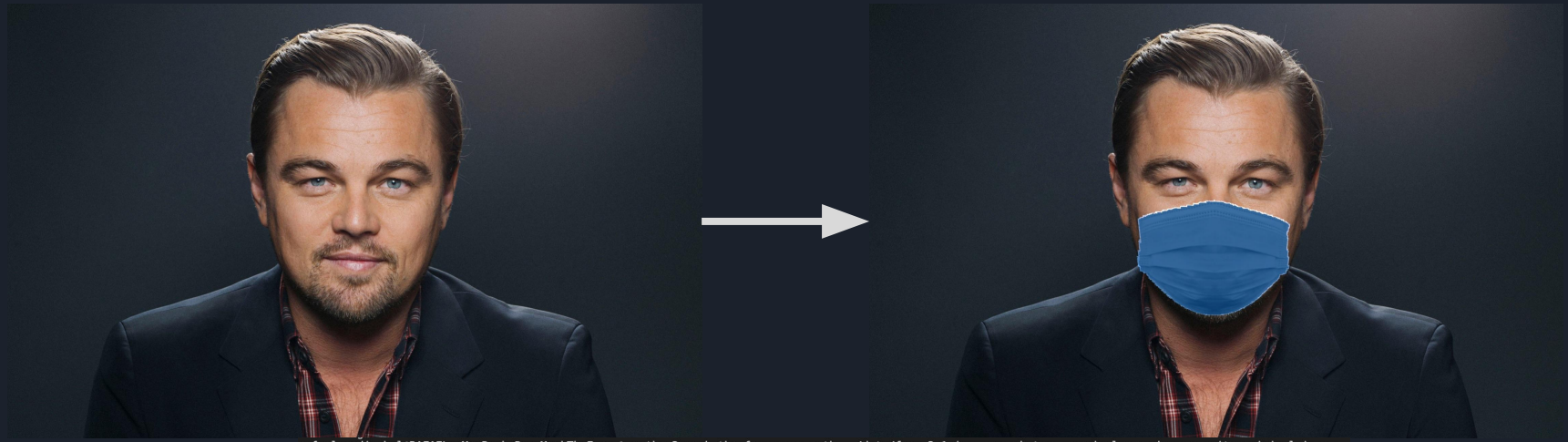
Goals

- Generate dataset of people with and without masks
- Explore face detection with and without masks
- Explore approaches for authenticating people when wearing masks



Generating the dataset

- We used MaskTheFace
  - An open source project that digitally inserts masks into faces
- Took sample of images present dataset for Assignment 4

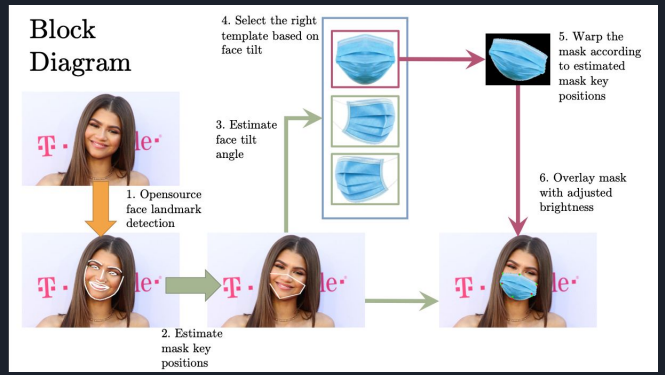


```
rafaelmendizabal@RAFAELs-MacBook-Pro MaskTheFace % python3 mask_the_face.py --path ../data/face_2_1.jpg  --mask_type surgical --verbose --write_original_image
```

```
Masking image file
Faces found:  1
Processing Done
```

How Mask The Face works

- Uses a dlib based face landmarks detector to identify the face tilt.
- Based on the face tilt, the corresponding mask template is selected from the library of the mask.
- The template mask is then transformed based on the six key features to fit perfectly on the face.
    - MaskTheFace provides several masks to select from.
- Identifies all the faces within an image and applies the user-selected masks to them
- A single image or entire directory of images can be used as input to code.



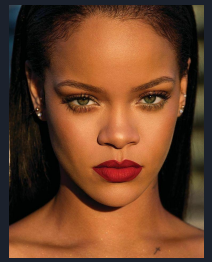
Generating our data

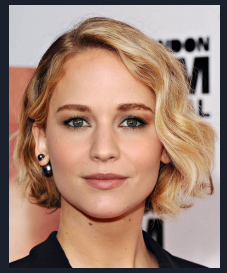
- To generate the pictures, we used the script provided in the cloned MaskTheFace repo https://github.com/aqeelanwar/MaskTheFace.git

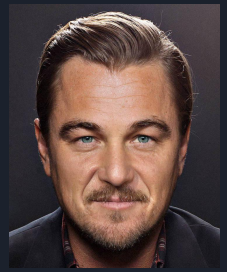
```
python3 mask_the_face.py --path 'data/office.jpg' --mask_type 'N95'
--verbose --write_original_image
```

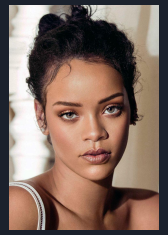
- Ran a script to use this command for a dataset
- We used the surgical mask for our dataset
- Used the images that were provided in Assignment 4 for testing.

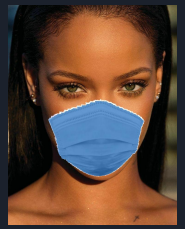
Comparing Data

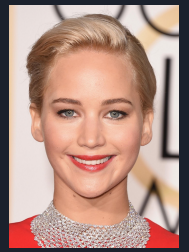


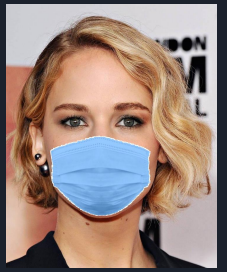










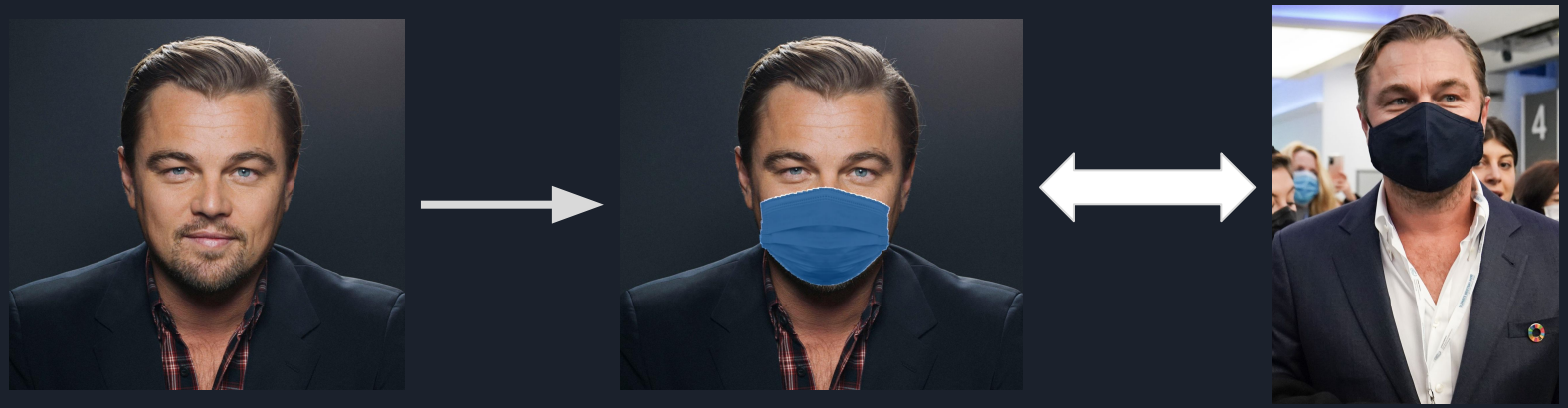


0.7914776 0.79415166 1.036541 1.1860108 0.4787405 0.6337838

For comparison: Non-pairs had distances between 1.44 and 2.14

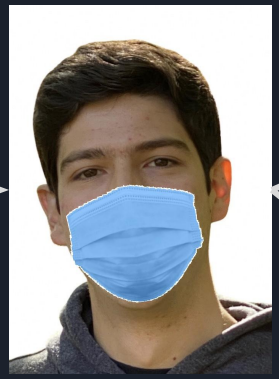
Possible Strategy #1

- For matching images of an individual with and without a mask, one strategy would be to use Mask The Face to add a fake mask to the image of the individual without the mask and then compare it with the real image of the individual with the mask
- This will then “level the playing field” with regards to facial features present in both images for the recognition software, as both images will then have features of the individual’s face covered up by a mask



Real World Example

Without masking step, distance was 1.115109

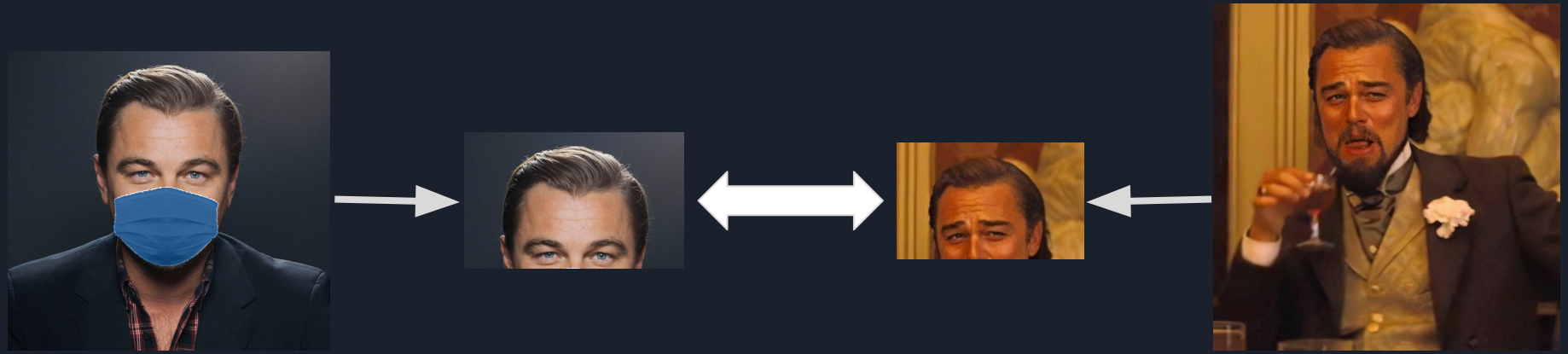


0.45599937

1.6731946

Possible Strategy #2

- Since masks already cover up over half of the individual's face in a picture, another possible strategy we came up with would be to just crop the images and cut out the mask and everything below it
- This would then allow for focusing on the individual's upper half of their face, with their eyes being especially important
- We would then run the same software on the cropped images and see if similar results come from the tests. This removes the possibility for the software to mistake the mask as a feature of the individual's face



Other Possible Strategies

- Similar approach to the cropping strategy that involves zooming in on and focusing on the individual's eyes. Would then rely heavily on iris recognition software for results
- Add in a second form of verification. A real world example of this is Apple with iPhone facial recognition paired with the Apple Watch. If the user is wearing a mask and the iPhone isn't 100% able to verify their face, it will check to see if the user is wearing their verified Apple Watch as well
- Add more layers of biometrics, such as voice or fingerprint recognition

Thank You!