Fooling Neural Network face recognition

GROUP MEMBERS

Ludger Heid :	**2820166935**
Kiiza Juma :	**2820160055**
Wei Jiayi :	**2620160022**
Ernesto O A Castellanos :	**2820160029**
Wadeh J Wisner:	**2820160073**
Jiazhi:	**2620160016**
Mogendi Enoch:	**3820120105**

Outline

- Introduction to Neural Networks basics
  - Convolution Neural Networks
  - Particle Swarm Optimization
- Basics of face detection
- Research problem
- Methodology
- implementation
- Conclusion

Introduction

- **The ability to recognizing an animal, describing a view, differentiating among visible objects accurately are really simple tasks for humans where as to impart this ability to a computer with reasonable accuracy is a progressive research.**

- **Object detection is considered to be the most basic application of computer vision.**

- **key challenges involved when we try to design systems similar to our eye:**

1. Variations in Viewpoint

same object different positions and angles in an image depending on the relative position of the object and the observer for

Easy for humans to recognize that these are the same object, not very easy to teach this to a computer

2. Difference in Illumination

Different images can have different light conditions

3. Background Clutter

a man in this image. As simple as it looks, it's an uphill task for a computer to learn.

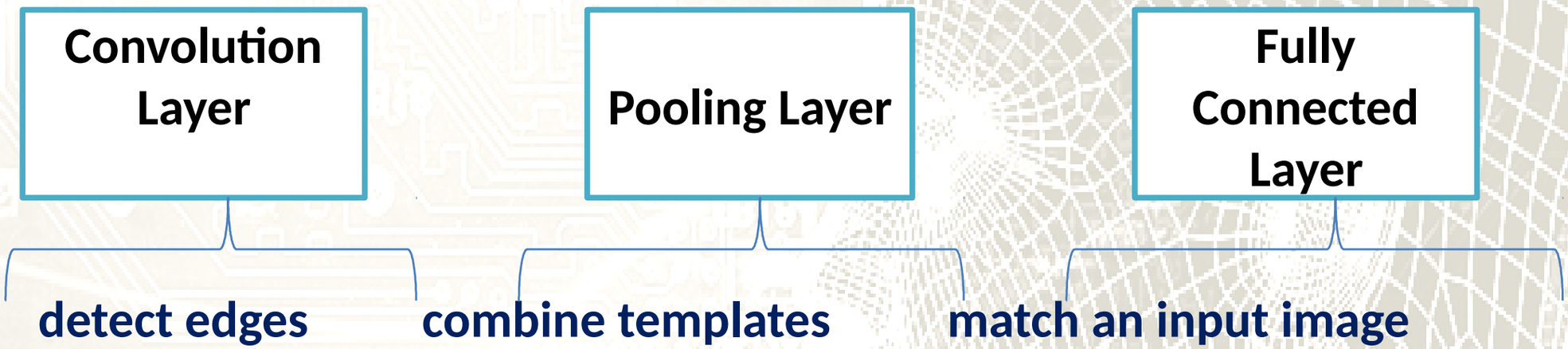
C N N

- Those are some of the challenges that can make us appreciate the complexity of the tasks our eyes and brain does with ease.

- Breaking up all these challenges and solving individually is still possible in the area of Convolution Neural Networks (CNNs)

Deep CNNs work by .....

consecutively modeling small pieces of information and combining them deeper in network.



So, deep CNNs are able to model complex variations and behaviour giving highly accurate predictions.

Particle Swarm Optimization PSO

(PSO) is a method, whose basic principle states that:

Over a number of iterations, a group of variables have their values adjusted closer to the member whose value is closest to the target at any given moment

PSO algorithm

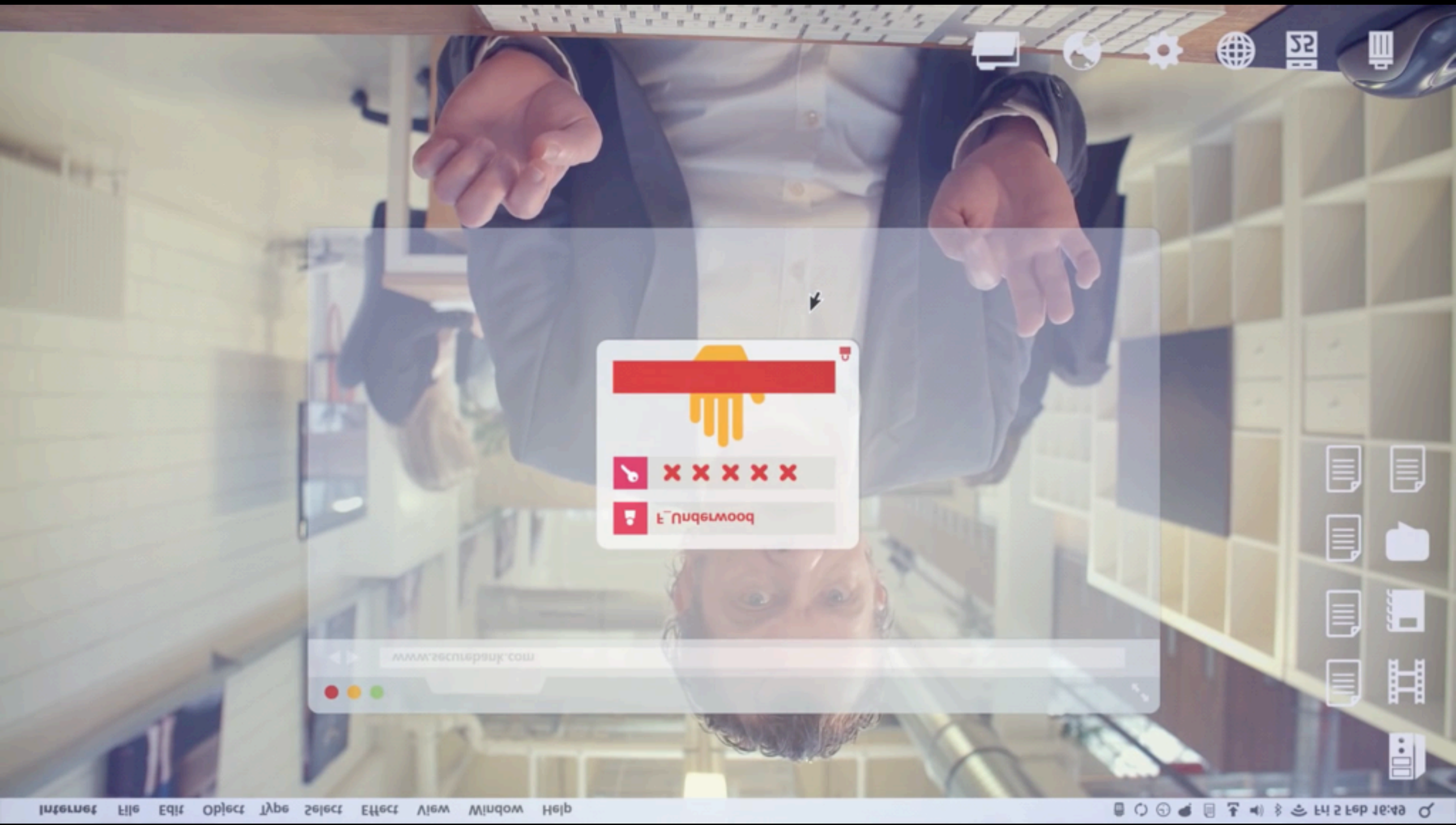
- **It's an algorithm that keeps track of three global variables:**
- **Target value or condition**
- **Global best (gBest) value indicating which particle's data is currently closest to the Target**
- **Stopping value indicating when the algorithm should stop if the Target isn't found**

- Each particle consists of:
- Data representing a possible solution
- A Velocity value indicating how much the Data can be changed
- A personal best (pBest) value indicating the closest the particle's Data has ever come to the Target

fooling neural network face recognition

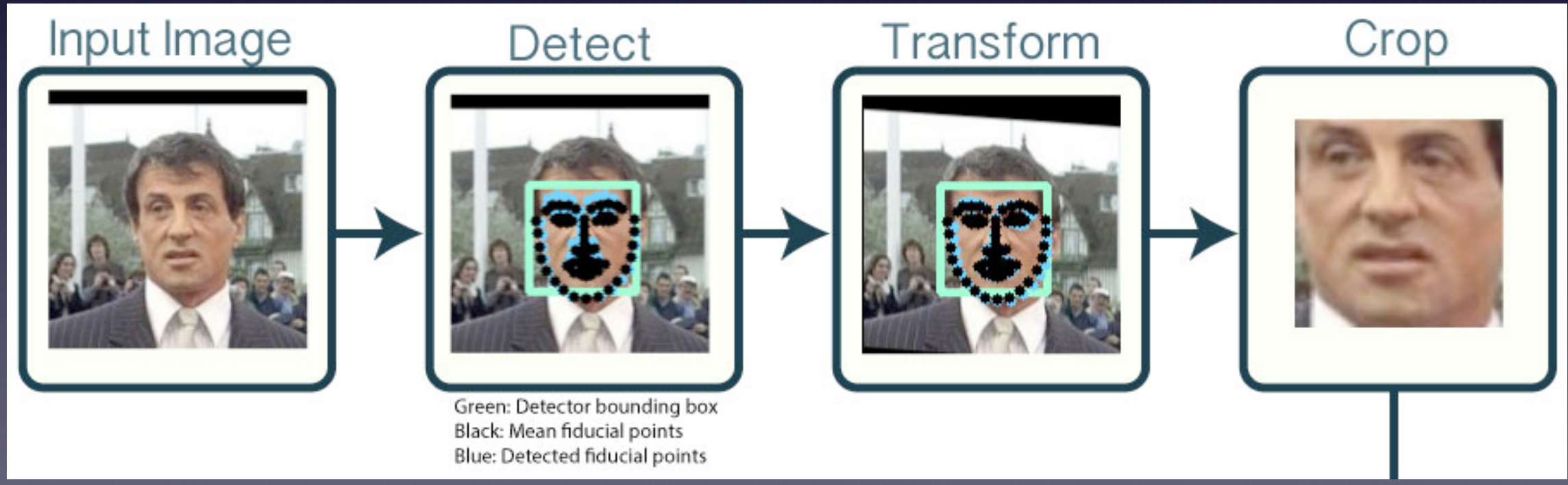
research paper & our results

Ludger Heide	2820166935
Kiiza Juma	2820160055
Wei Jiayi	2620160022
Ernesto O A Castellanos	2820160029
Wadeh J Wisner	2820160073
Jiazhi	2620160016
Mogendi Enoch	3820120105


E_Underwood
Internet File Edit Object Type Select Effect View Window Help
Fri 2 Feb 12:48

alignment

- first: *face detection* and *face alignment*
- not discussed further



source: https://cmusatyalab.github.io/openface/

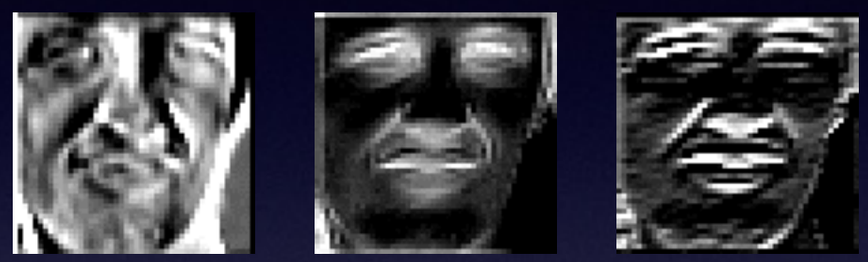
basics

- neural network as series of filters on higher semantic levels

- output: vector describing the face

- difference between vectors describes *distance* between faces

3



7

13

21

basics

- neural network as series of filters on higher semantic levels

- output: vector describing the face

- difference between vectors describes *distance* between faces

```
[-0.03731764 -0.01561464  0.01871112 -0.1896884   0.0141813   0.11998807
 -0.1775018   0.11719089 -0.03584165 -0.05334716  0.01933085  0.00068819
 -0.00438189  0.01604373  0.13468499  0.05427272 -0.02201736 -0.12284043
  0.07369097 -0.08206913  0.02801489  0.14109327  0.14309002 -0.03021246
  0.02277708  0.14497924  0.24213907 -0.06582481  0.00859915 -0.14085065
  0.12641153 -0.10929709  0.11195789  0.11451519 -0.06058023  0.09117635
  0.09233699  0.06618129  0.07209234  0.16133237  0.04103077 -0.02225384
 -0.01103241  0.04949003 -0.05075597 -0.03860864  0.08009686 -0.07201371
 -0.07325712  0.05659901 -0.04616901 -0.04319548 -0.05779455  0.08067982
 -0.02417592  0.06572749  0.07561753  0.08042991 -0.09355874 -0.02977388
 -0.12101793 -0.09106496  0.07635678 -0.08613656  0.07799065 -0.05007986
 -0.00889494  0.12500684  0.08848483  0.17552792 -0.07404196  0.02500297
 -0.1411252  -0.02654243  0.14151649 -0.06760675 -0.05882904  0.08380576
 -0.05317562  0.02084809 -0.07514285  0.04593345  0.1303952   0.05175722
 -0.09221626 -0.04649157  0.11642307  0.06461141 -0.0159792  -0.01741427
  0.05841637 -0.0620751  -0.06009264  0.0657089   0.11441118  0.04963372
 -0.0427677   0.12926039 -0.06114963  0.01086538 -0.16900277 -0.08736809
 -0.10464593  0.0792935   0.02797886  0.00219243 -0.12738679 -0.07894275
 -0.04251058 -0.1815545  -0.04598103  0.08486018 -0.00588039  0.05039927
  0.10944649  0.00051715  0.0305217  -0.17299615  0.06423193  0.13510123
  0.18512736  0.02827185  0.050085    0.0298596  -0.02292899 -0.05598702
                          0.0607643   0.1213531 ]
```

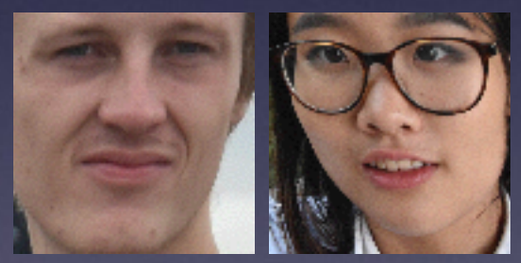
basics

- neural network as series of filters on higher semantic levels

diff() = 0.0

- output: vector describing the face

- difference between vectors describes *distance* between faces

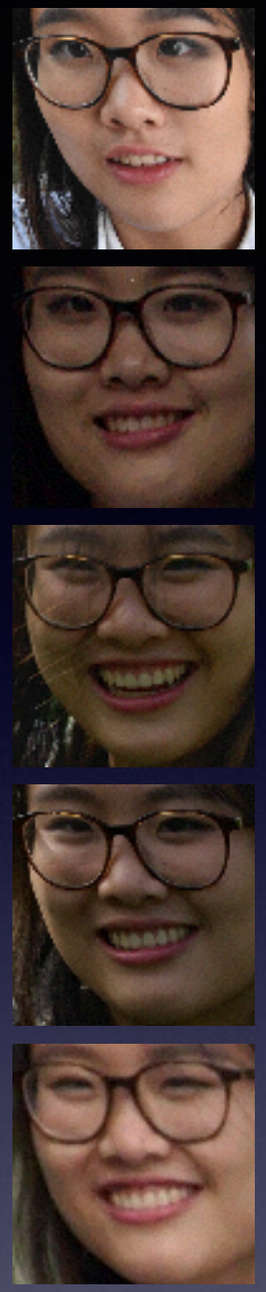
diff() = 2.0

basics

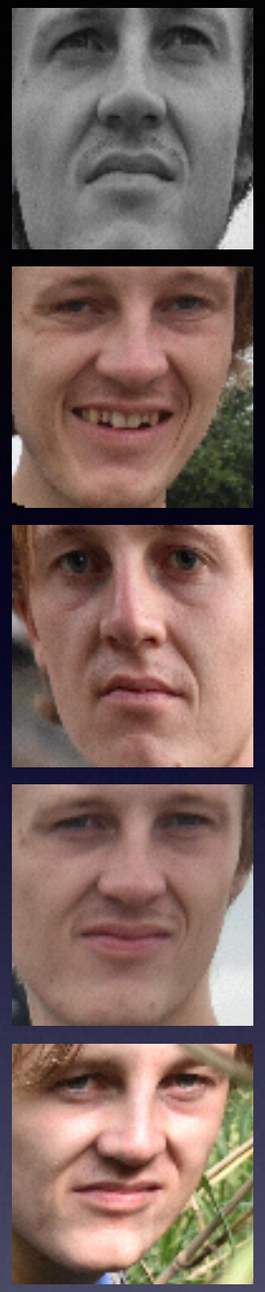
- neural network as series of filters on higher semantic levels

- output: vector describing the face

- difference between vectors describes *distance* between faces



average between different faces 2.04



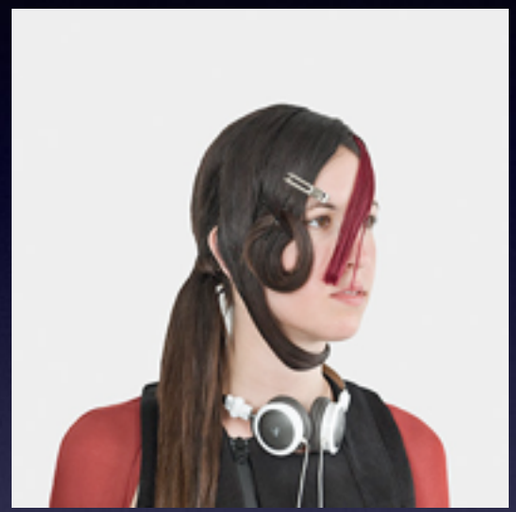
average distance 0.26

average distance 0.35

(all standard deviations ~0.2)

research problem

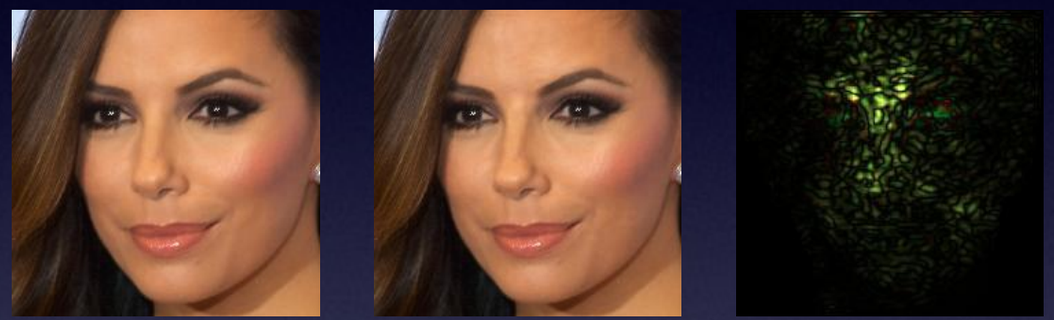
- can those algorithms be fooled?
- not discussed here:
  fooling *face detection*
- fooling *face recognition* by
  (subtly) changing input image
  - dodging
    not being identifiable
  - impersonation
    appearing as a different person



source: https://cvdazzle.com/#looks

research problem

- can those algorithms be fooled?
- not discussed here:
  fooling *face detection*
- fooling *face recognition* by
  (subtly) changing input image
  - dodging
    not being identifiable
  - impersonation
    appearing as a different person

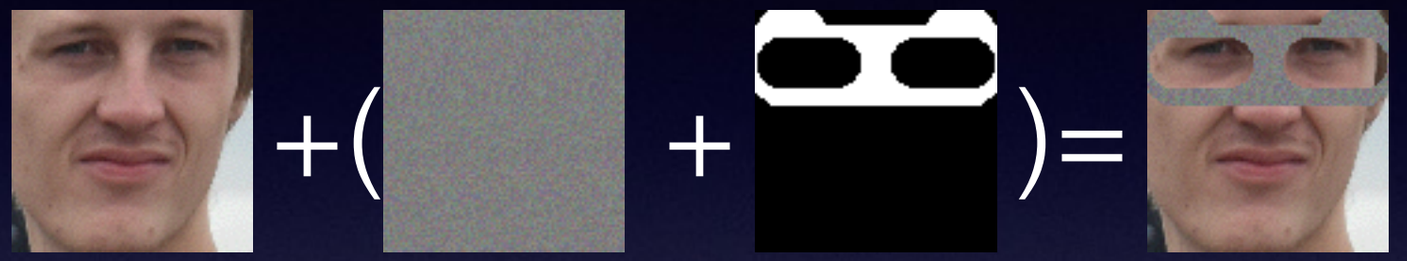


Dodging by subtle manipulation. source: [1]

Impersonation by using glasses. source: [1]

methodology

- overlay pattern on source image



- calculate distance to target image

diff()

- optimize pattern to minimize difference

min(diff(*params*))

  - what pattern to use?
  - what optimization method?

methodology (2)

- 10 lines
  - *start, end, color, width*
    2 + 2 + 3 + 1
    = 8 parameters per line



  - + blurring at the end

  ➡ **room for improvement**

- particle swarm optimization
  - good results for problems without clear dependencies

implementation

- openFace provides python API for faceNet from Google
- fooling pattern generated using opencv
- optimization using the pyswarm package
- runtime ~2.5 minutes (don't have a Titan X 😕)
    - quadratic growth for optimizing *n* against *n* images

conclusions

- *offline, black-box* impersonation attacks on known neural networks are feasible
- for single-image impersonation acceptable results are easily reached
  - ~0.3-0.4
- for multi-image fooling results worsen to ~0.6-0.7
  - ~0.3-0.4 achievable with bigger mask (or better pattern?)

Thank you for your attention!

Sources

- our project:
  https://github.com/ludgerheide/neural-network-PSO

- [1] paper reproduced:
  Sharif, Mahmood, et al. „*Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition.*“ Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2016.

- [2] neural network used:
  Schroff, Florian, Dmitry Kalenichenko, and James Philbin. "*Facenet: A unified embedding for face recognition and clustering.*" Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015.

- [3] face recognition API:
  B. Amos, B. Ludwiczuk, M. Satyanarayanan, „*Openface: A general-purpose face recognition library with mobile applications*“, CMU-CS-16-118, CMU School of Computer Science, Tech. Rep., 2016.