Hacking Facial Recognition Systems

Dr. Rich Harang & Dr. Ethan Rudd

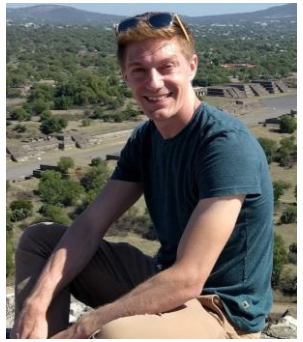
SOPHOS

Our backgrounds

- Rich
  - PhD in Statistics and Applied Probability (UCSB)
  - Five years at U.S. Army Research Laboratory on all things ML + network security
  - Three years (and counting) at Sophos – ML for endpoint security
- Ethan
  - PhD in Computer Science (University of Colorado)
  - Previous work w/ UCCS VAST Lab, IARPA/Janus, Securics Inc., Hewlett Packard, Google ATAP
  - Senior Data Scientist at Sophos since 2017

Data Science @ Sophos

Josh Saxe



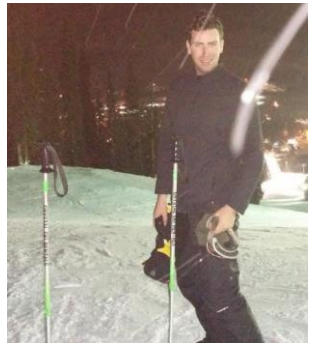
Kevin Hake

Rich Harang

Hillary Sanders

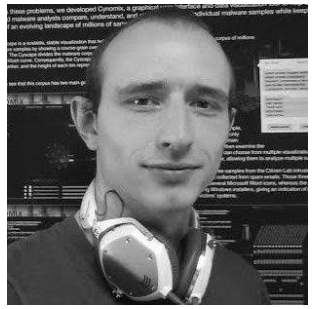
Adarsh Kyadige

Konstantin Berlin



Ethan Rudd

Felipe Ducau



Alex Long

Andrew Davis

William Lee

Matt Stec

Matt Burnett

You?

This talk

- Crash course on image classification
- Why facial recognition is different
- How facial recognition classifiers are trained
- The dystopia is already here
- A mad dash through some attacks
- An anecdotal look at transferability

Key takeaways

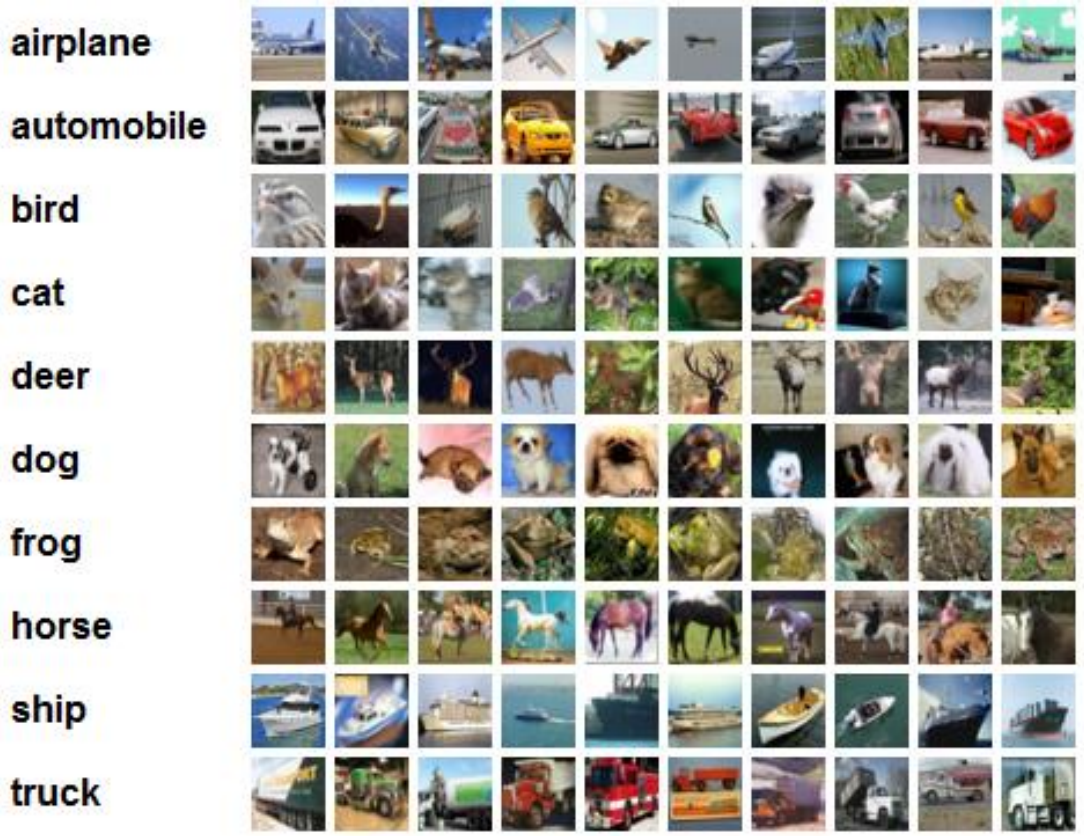
- Facial recognition systems are different from ‘general’ image classifiers
  - This creates openings for new attacks, makes some attacks less feasible
- Building facial recognition systems is easier than you might think
  - “Off-the-shelf” open source systems are already quite good
  - Cloud-based solutions are cheap and easy to use
- Reliable evasion of facial recognition is still very difficult
  - Best solution to date: hide your face

Background – a crash course in image classification

SOPHOS

Build a model to answer “What is this picture?”

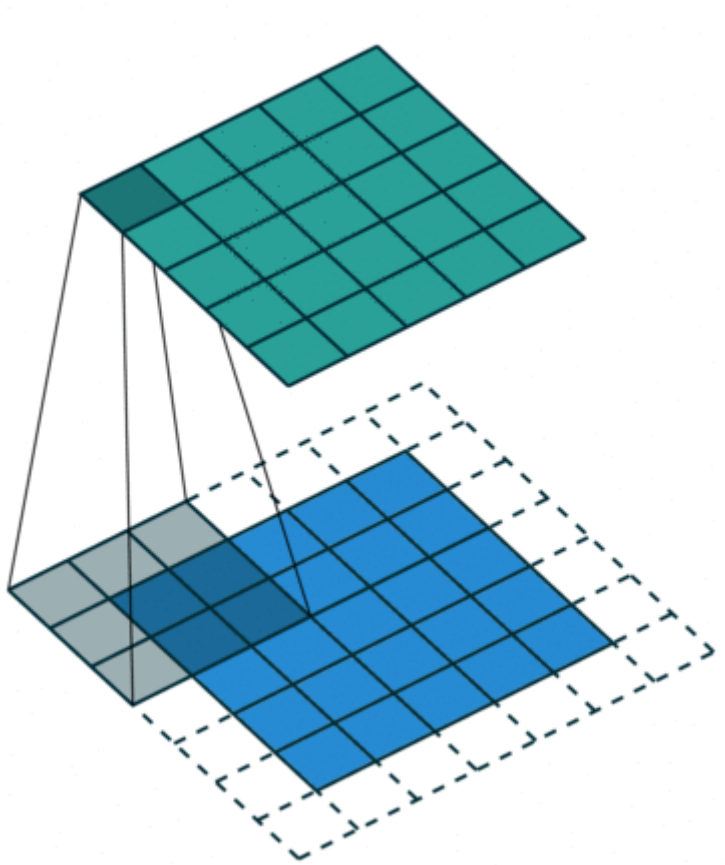
- Training data – examples of things you want to learn about
- Need multiple examples per class



http://karpathy.github.io/2011/04/27/manually-classifying-cifar10/

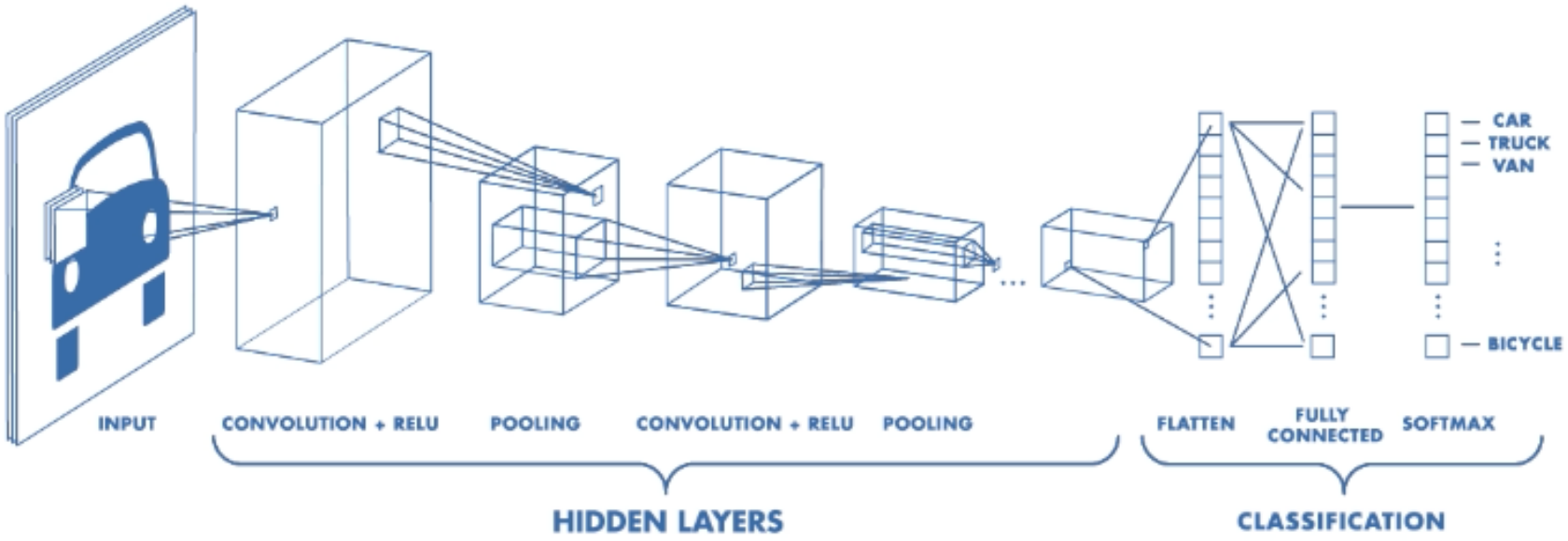
But how?

The standard building block for image recognition is the *convolutional layer*



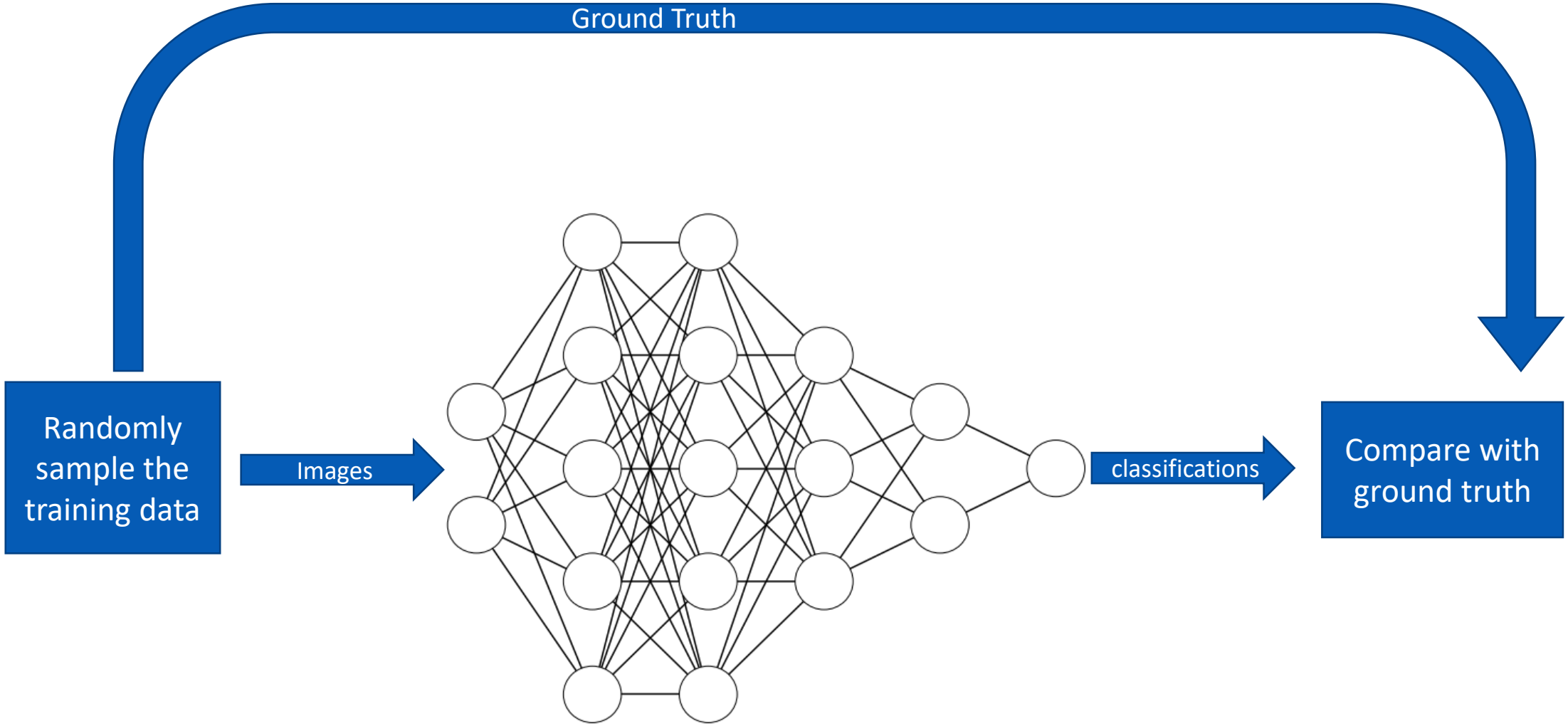
https://github.com/vdumoulin/conv_arithmetic

Stack to ‘learn features’ – add a classification network to the end

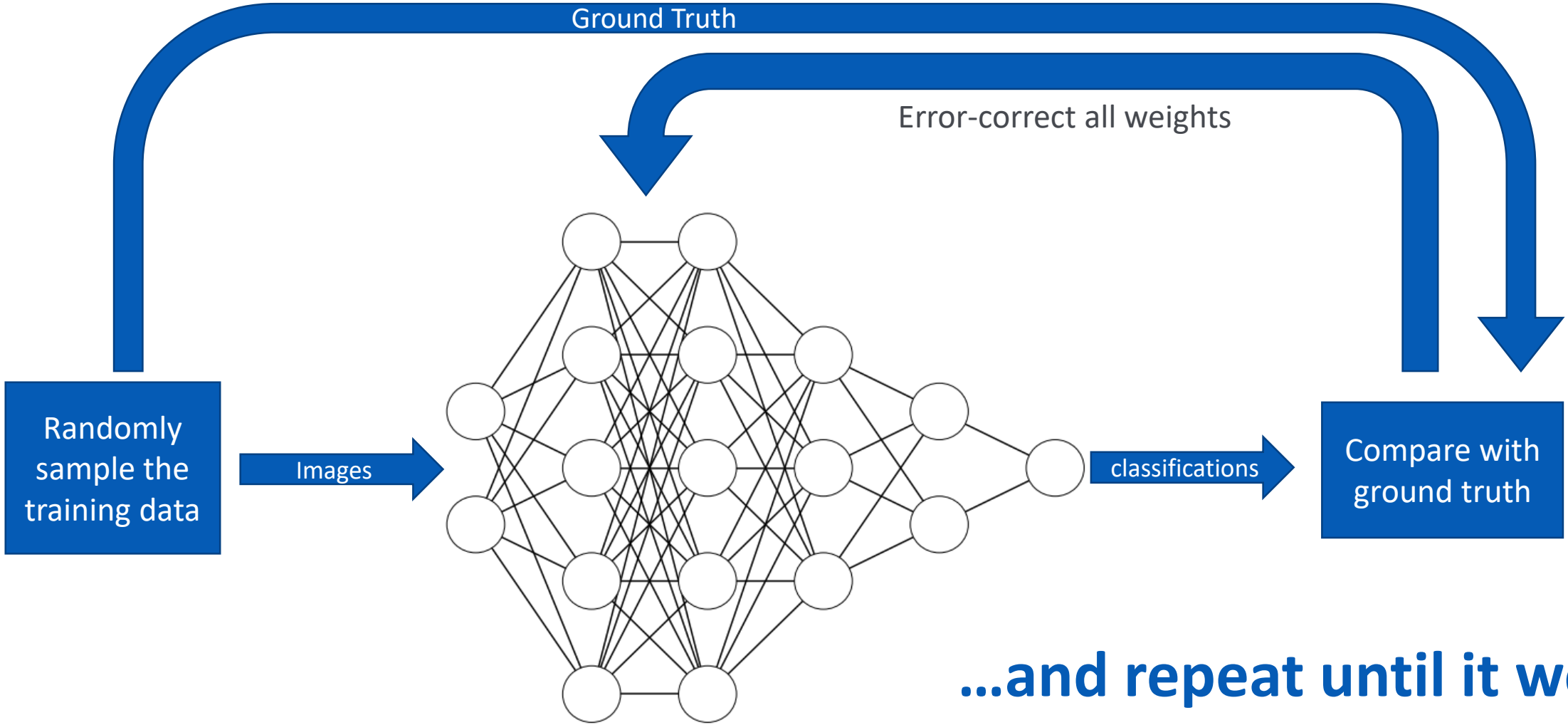


https://www.freecodecamp.org/news/an-intuitive-guide-to-convolutional-neural-networks-260c2de0a050/

Training


Ground Truth
Randomly sample the training data
Images
classifications
Compare with ground truth

Training



…and repeat until it works

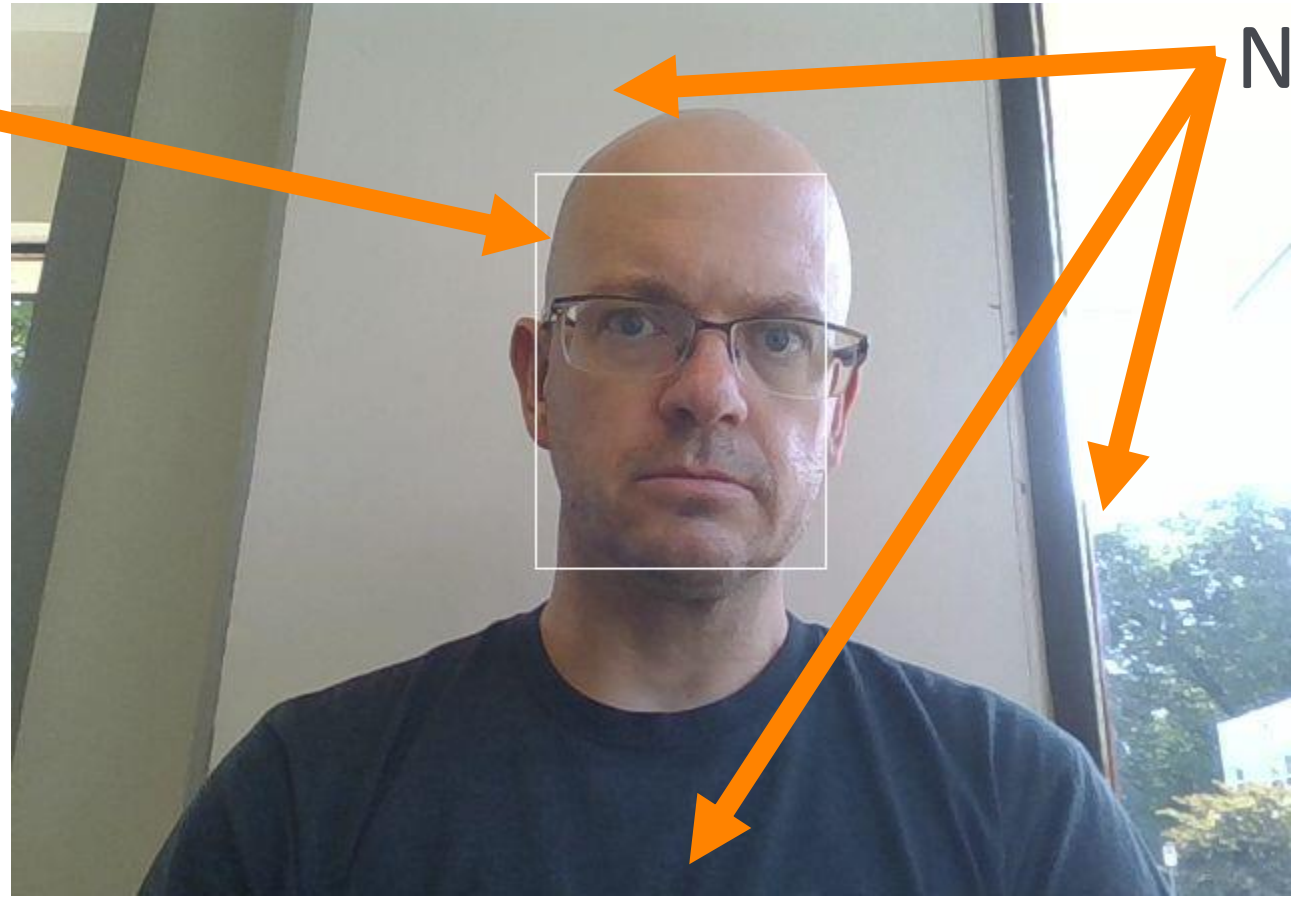
Now do it for faces!

Wait a second… data?

We only care about faces…

We only care about faces…

Interesting*

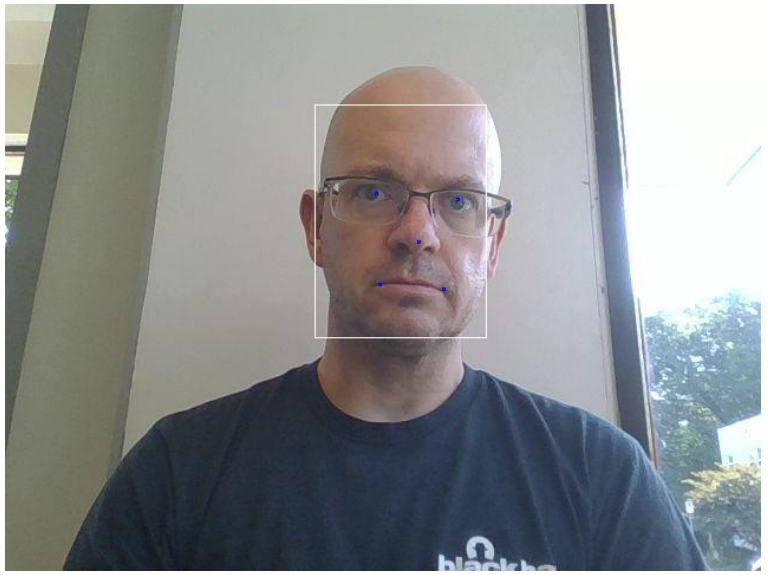


Not interesting

* For certain values of “interesting”

Step zero: bounding boxes, landmarks, alignment

- Isolate faces
- Identify landmarks
- Crop
- Use landmarks to align to a standard orientation*



Lots of tools available to do this

- (MTCNN shown)

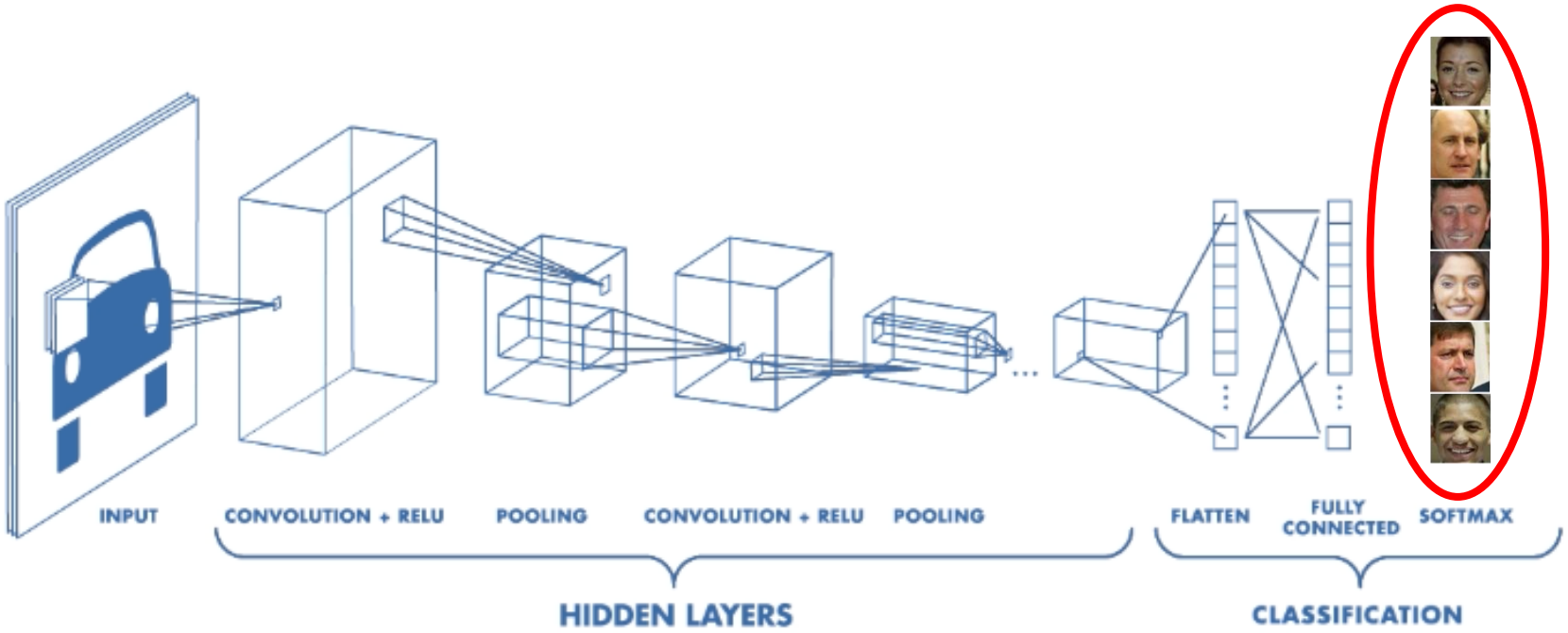
*Precise alignment not needed for some models

OK! *Now* do it for faces!

There’s a problem…

This is it: the problem

The structure of the model assumes a *fixed number of classes* and a *closed world*.



https://www.freecodecamp.org/news/an-intuitive-guide-to-convolutional-neural-networks-260c2de0a050/

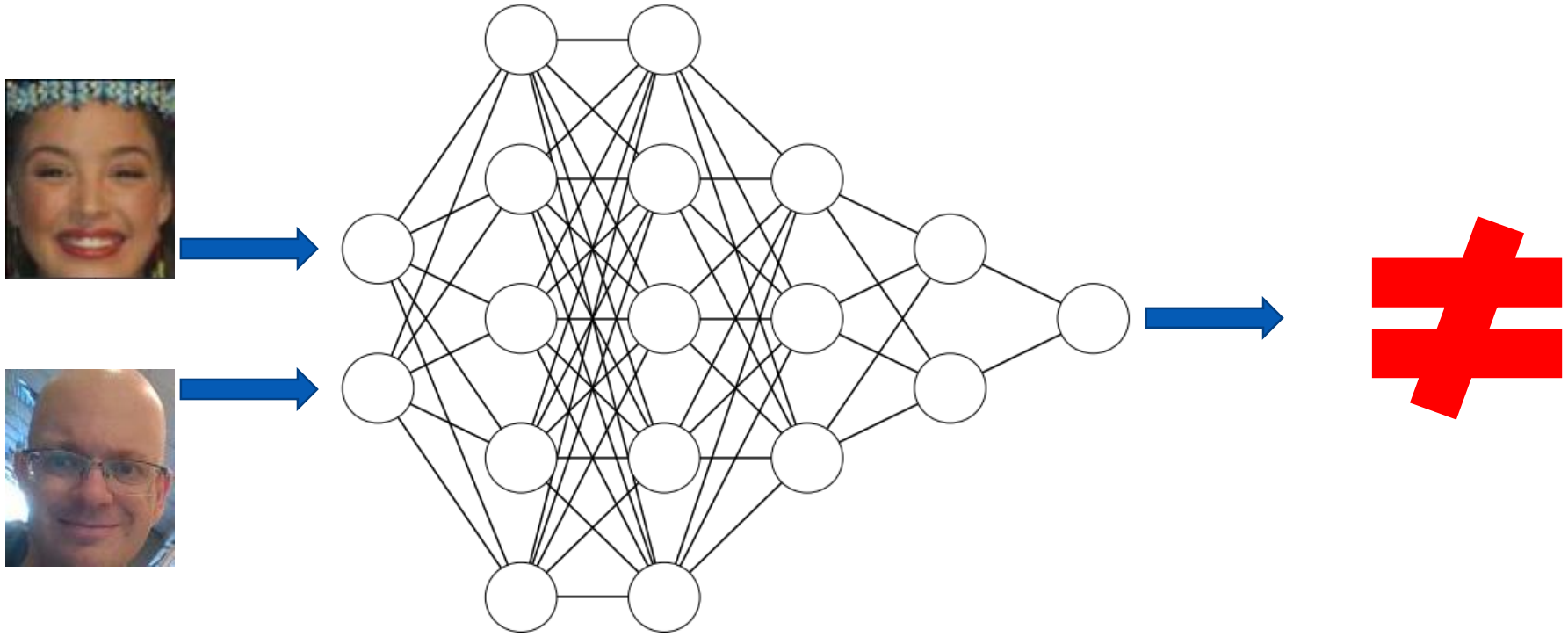
What if…

- We present it a face not in the training set?
  - A softmax classifier will *always* return a result from the training set
- We only care about a few people?
  - Limited training data
  - Can add more people to produce more training data… but then we might ID people we don’t care about!
- We later decide we have more people we want to identify?
  - We have to retrain the network!

First fix

Dealing with a large/unknown number of classes

- Focus on matching faces that have been extracted/aligned.
- Separates training data from data for people we want to track (“registered” faces).



Good news/bad news

Good news:

- Super easy to register new faces: just add them to the set of faces you compare against

Bad news:

- Every single registered face has to be run through the network every time we want to identify someone
- What if we have thousands of faces in our “registered” database?

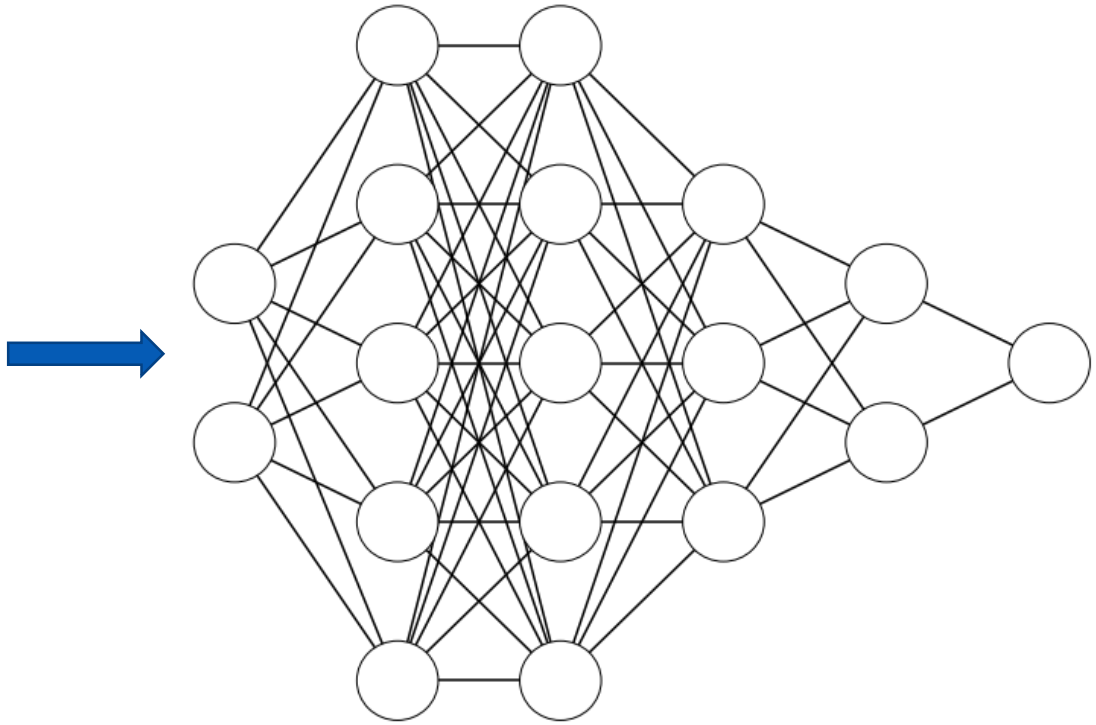
Too slow to be useful beyond toy problems

Second fix

Avoiding running the network over and over and over and…

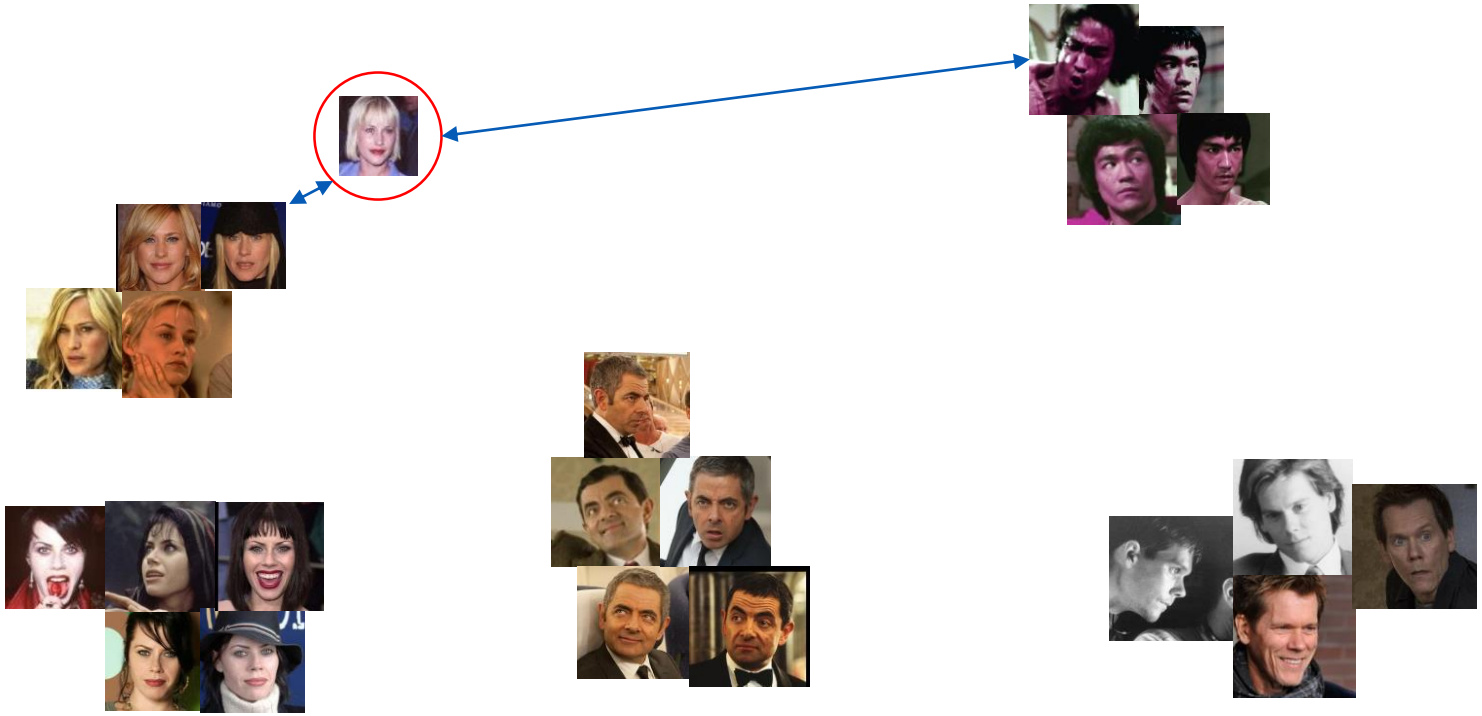
- The network's job is to learn an *embedding* for faces
  - Two faces from the same person should have a similar embedding
  - Two faces from different people should have *dissimilar* embeddings

Registration:



```
+0.574 +0.305 -0.915 ...
-0.173 +2.964 +1.516 ...
+2.693 +0.508 +1.733 ...
+1.744 +0.030 +0.931 ...
-1.091 +1.555 +0.991 ...
-0.237 -0.389 +0.193 ...
-1.783 +0.980 +1.171 ...
+0.500 +0.595 -0.619 ...
-0.348 +1.871 +0.858 ...
-2.227 -0.823 +0.428 ...
```

Embeddings produce clusters: can define a *distance*



With an *embedding* and a *distance*..

- Can exploit fast algorithms to find nearest neighbor

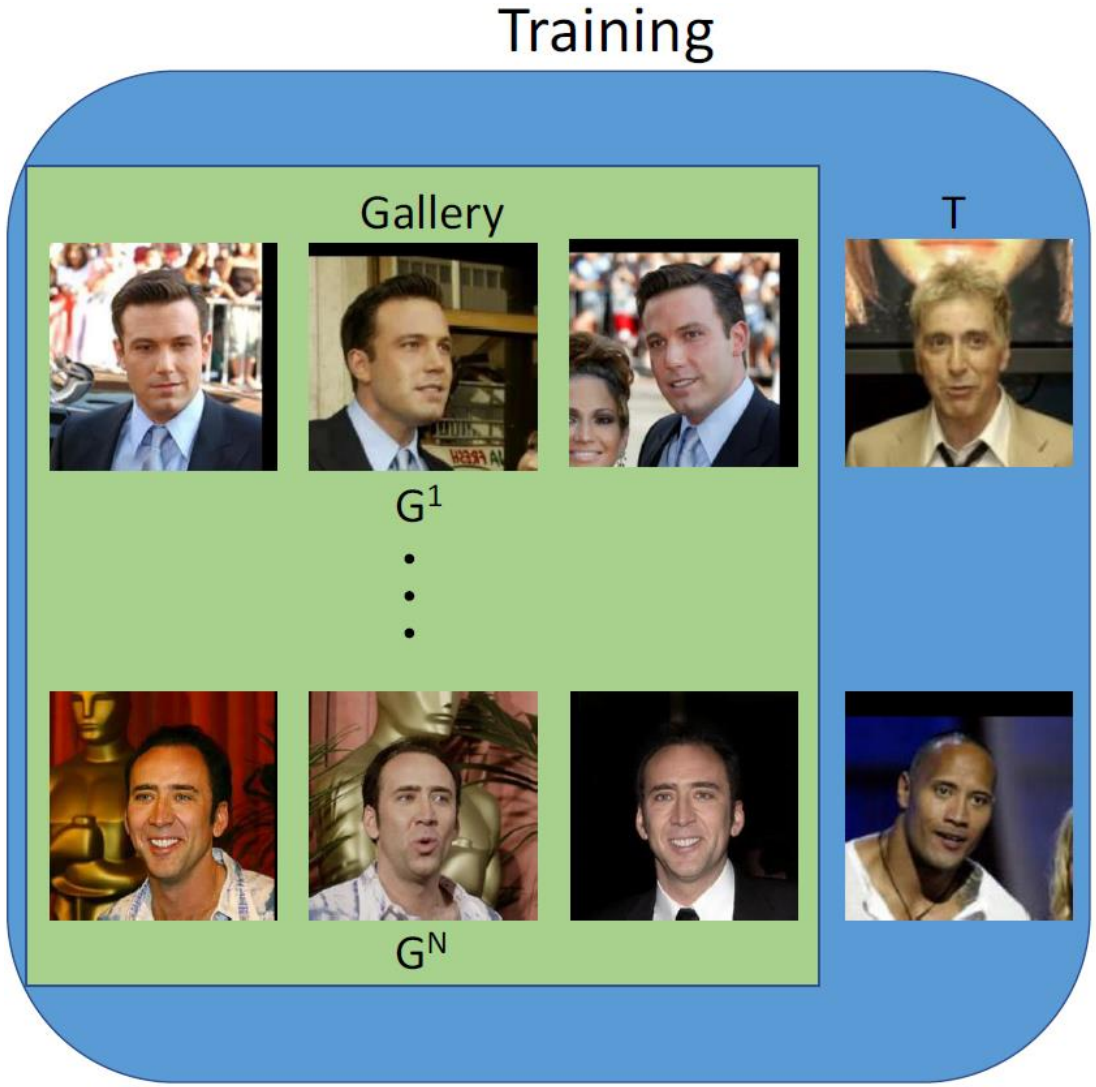
1. Get an image
2. Isolate and align faces
3. Use network to embed
4. For each face, find the nearest neighbor (or neighbors) among registered faces, return those as the result

A key difference on data

- You don’t need to know who a person is to train on their face or enroll them
- You just need
  - a unique identifier and
  - multiple images of them

- This is **much easier** than you might think:
  - Social media
  - Video frames
  - Tracking by device, gait, etc.

Different Operational Scenarios



https://arxiv.org/abs/1705.01567

Recap

- Two models:
  - Face detection
  - Face embedding

} These are trained with whatever face data you have available

- Faces you want to detect don't have to be available when training the models
  - "Enroll" faces by generating their embedding
  - Recognize faces by detecting/cropping the unknown face, finding the embedding, and then finding nearby matches

Facial recognition pipeline: the short, short version

Find, crop, maybe align face with localization model

Get embedding from embedding model

“Registration”: save labeled embedding; optionally form “templates”	“Recognition”: Find nearest embedding (or template) to current unlabeled embedding

“The future is already here – it's just not evenly distributed.”

– William Gibson

SOPHOS

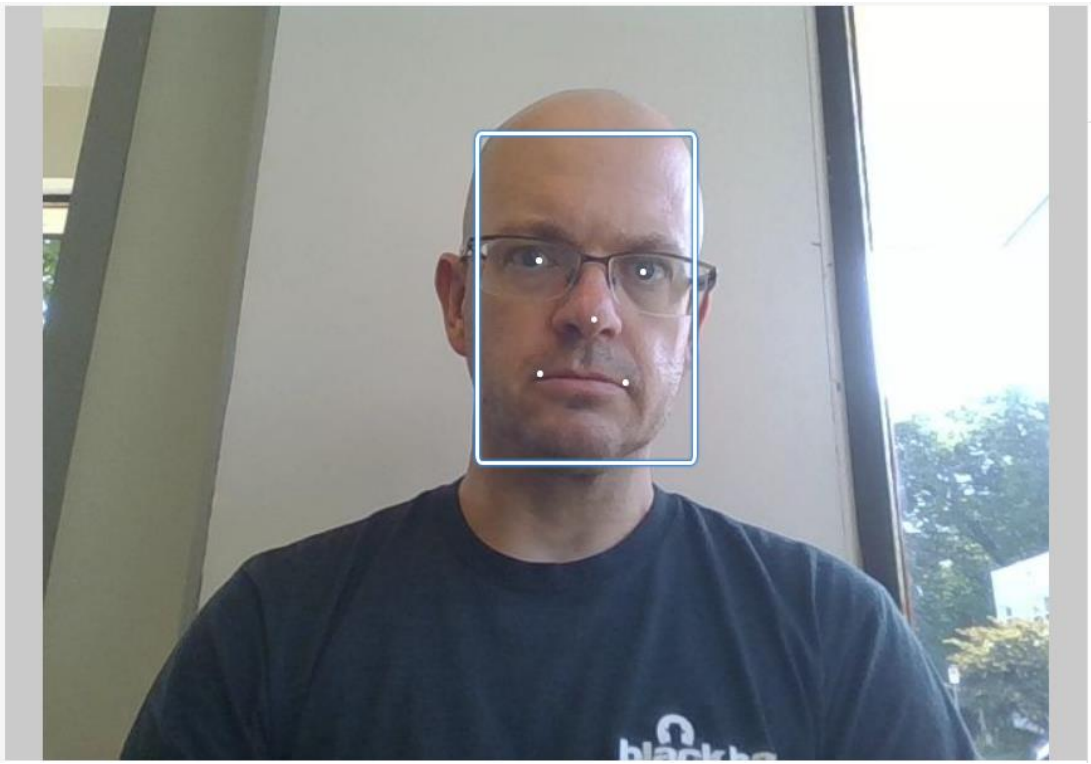
Need models? Got models. Need data? Plenty of data.

- Pre-trained models and tools widely available, see (e.g.):
  - https://github.com/ageitgey/face_recognition
  - https://www.docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec_tutorial.html#face-recognition
  - https://github.com/clcarwin/sphereface_pytorch
  - https://github.com/ipazc/mtcnn
  - https://github.com/davidsandberg/facenet
  - …and many others

- Data also widely available (but see e.g. MS pulling Celeb-A data offline)
  - http://face-rec.org/databases/
  - http://vis-www.cs.umass.edu/lfw/ (Labeled faces in the wild)
  - http://www.robots.ox.ac.uk/~vgg/data/vgg_face2/ (VGG-Face 2)
  - http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html (alternate celebA data)
  - https://sites.google.com/view/sof-dataset (specs on faces dataset)
  - http://iab-rubric.org/resources/dfw.html (Disguised faces in the wild)*
  - http://iab-rubric.org/resources/facedisguise.html (Recognizing Disguised Faces)*
  - http://www.antitza.com/makeup-datasets.html (faces with and without makeup)*
  - Googling for CAISA-webface will eventually turn up a recent download link
  - … et cetera

* Requires some form of agreement/application to download

Amazon’s “Rekognition”



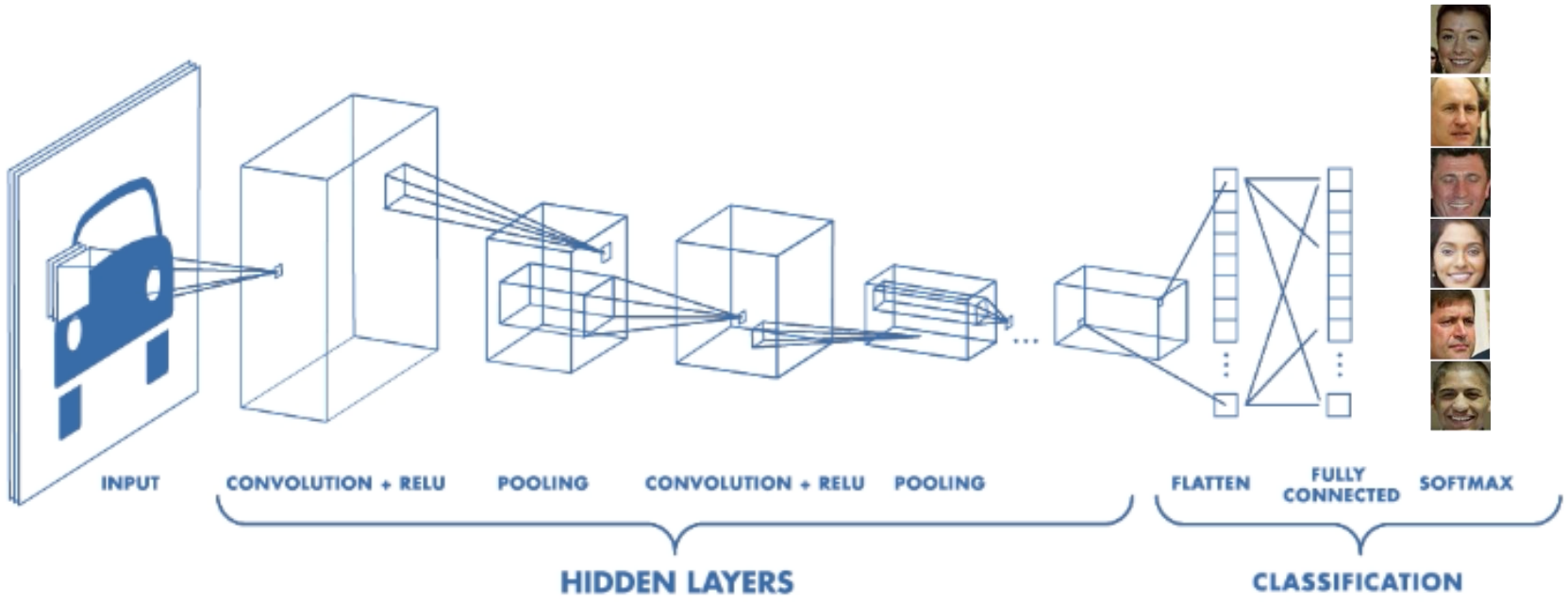
looks like a face	99.9 %
appears to be male	99.9 %
age range	38 - 57 years old
not smiling	96.7 %
appears to be calm	83.3 %
wearing glasses	94.4 %
not wearing sunglasses	98.2 %
eyes are open	99.9 %
mouth is closed	98 %
does not have a mustache	98.2 %
does not have a beard	57.9 %

Attacking it

SOPHOS

First: a word of warning

- Some attacks against facial recognition in the scientific literature attack this kind of model, not embedding-based systems; check the model first!



Plan your attack

What are my capabilities?

- Can I attack model training? Data poisoning!
- Can I get a copy of the models? White box/offline attacks!
- Can I query the system repeatedly? Black box/online attacks!

New in facial recognition:

- Can I enroll new faces…
  - Try data poisoning (two flavors!)
  - Try resource exhaustion!
- Can I alter data in between the two models? Face-only attack!
- Can I read data in between the two models? Detection attack!
- Can I get a score from the system? Black-box attack!

Plan your attack

What are my capabilities?

- Can I attack model training? Data poisoning!
- Can I get a copy of the models? White box/offline attacks!
- Can I query the system repeatedly? Black box/online attacks!

New in facial recognition:

Pretty straightforward – not going to cover

- Can I enroll new faces…
  - …with labels? Data poisoning (two flavors!)
  - …without labels? Resource exhaustion!
- Can I alter data in between the two models? Face-only attack!
- Can I read data in between the two models? Detection attack!
- Can I get a score from the system? Black-box attack!

Plan your attack

What are my capabilities?

- Can I attack model training? Data poisoning!
- Can I get a copy of the models? White box/offline attacks!
- Can I query the system repeatedly? Black box/online attacks!

New in facial recognition:

- Can I enroll new faces…
  - …with labels? Data poisoning (two flavors!)
  - …without labels? Resource exhaustion!
- Can I alter data in between the two models? Face-only attack!
- Can I read data in between the two models? Detection attack!
- Can I get a score from the system? Black-box attack!

Easy mode

Plan your attack

What are my capabilities?

- Can I attack model training? Data poisoning!
- Can I get a copy of the models? White box/offline attacks!
- Can I query the system repeatedly? Black box/online attacks!

New in facial recognition:

- Can I enroll new faces…
  - …with labels? Data poisoning (two flavors!)
  - …without labels? Resource exhaustion!
- Can I alter data in between the two models? Face-only attack!
- Can I read data in between the two models? Detection attack!
- Can I get a score from the system? Black-box attack!

Surprisingly hard

Plan your attack

What are my capabilities?

- Can I attack model training? Data poisoning!
- Can I get a copy of the models? White box/offline attacks!
- Can I query the system repeatedly? Black box/online attacks!

New in facial recognition:

- Can I enroll new faces…
  - …with labels? Data poisoning (two flavors!)
  - …without labels? Resource exhaustion!
- Can I alter data in between the two models? Face-only attack!
- Can I read data in between the two models? Detection attack!
- Can I get a score from the system? Black-box attack!

But! Field is moving fast! See "Hiding Faces in Plain Sight" by Yuezun Li, Xin Yang, Baoyuan Wu and Siwei Lyu

Surprisingly hard

Plan your attack

What are my capabilities?

- Can I attack model training? Data poisoning!
- Can I get a copy of the models? White box/offline attacks!
- Can I query the system repeatedly? Black box/online attacks!

New in facial recognition:

- Can I enroll new faces…
  - …with labels? Data poisoning (two flavors!)
  - …without labels? Resource exhaustion!
- Can I alter data in between the two models? Face-only attack!
- Can I read data in between the two models? Detection attack!
- Can I get a score from the system? Black-box attack!

Unsurprisingly hard

The most realistic scenario:

You won't…

a) have any idea if your photo is being analyzed, or

b) know any detail of how that system works, or

c) get any feedback on the effectiveness of any attack you might try.

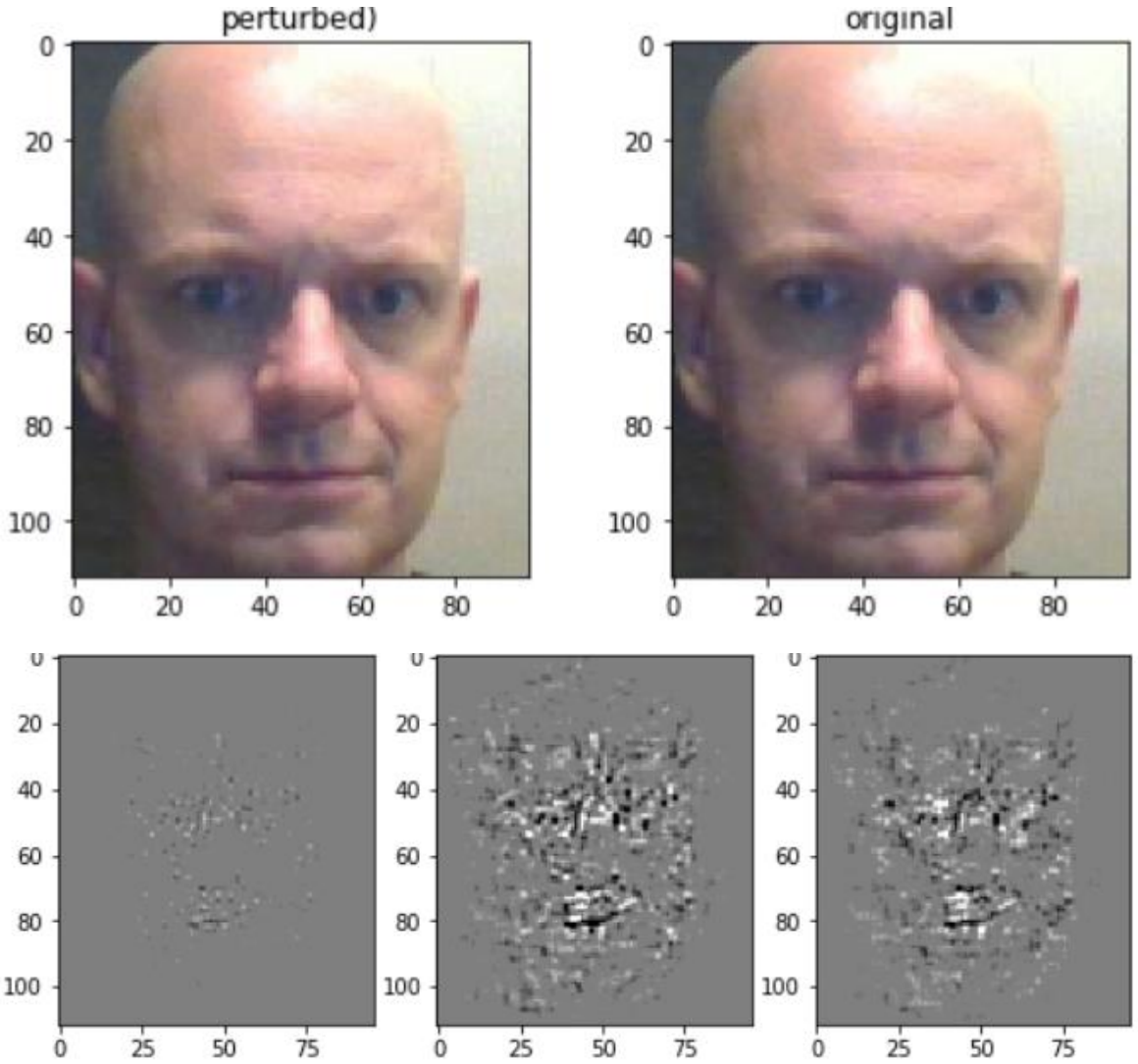
Outline of a white-box attack

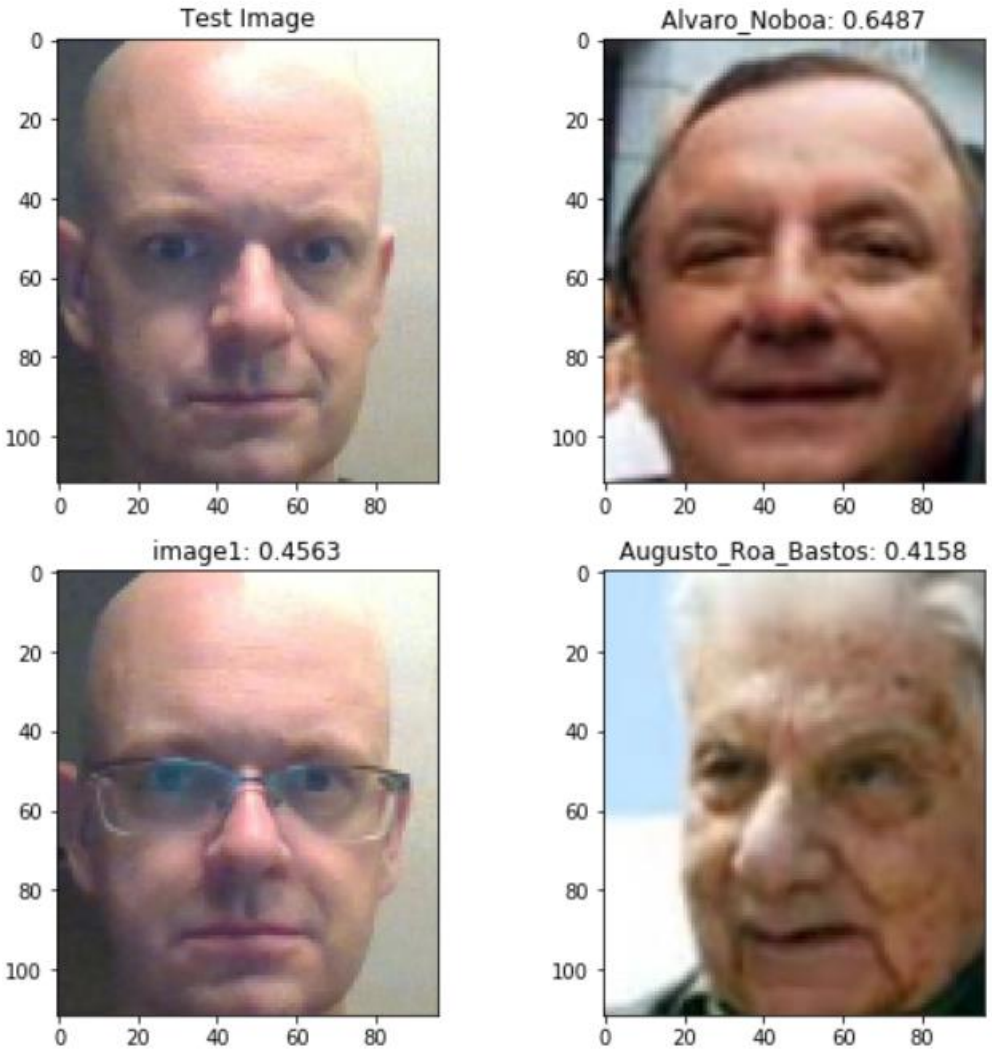
- Gradient-based methods
  - Fast Gradient Sign Method: https://arxiv.org/pdf/1608.04644.pdf
  - Saliency: https://arxiv.org/pdf/1511.07528.pdf
- Basic idea:
  - Use gradients to find the most sensitive input features; tweak those features to evade

My Lazy Implementation:

- Treat the perturbation as a PyTorch Variable
- Roll L1/L2 penalty into loss
- Fix target “prediction gap”; give no additional loss reduction below that point
- SGD to Glory

Gradient-based (white box) attack – face chip only





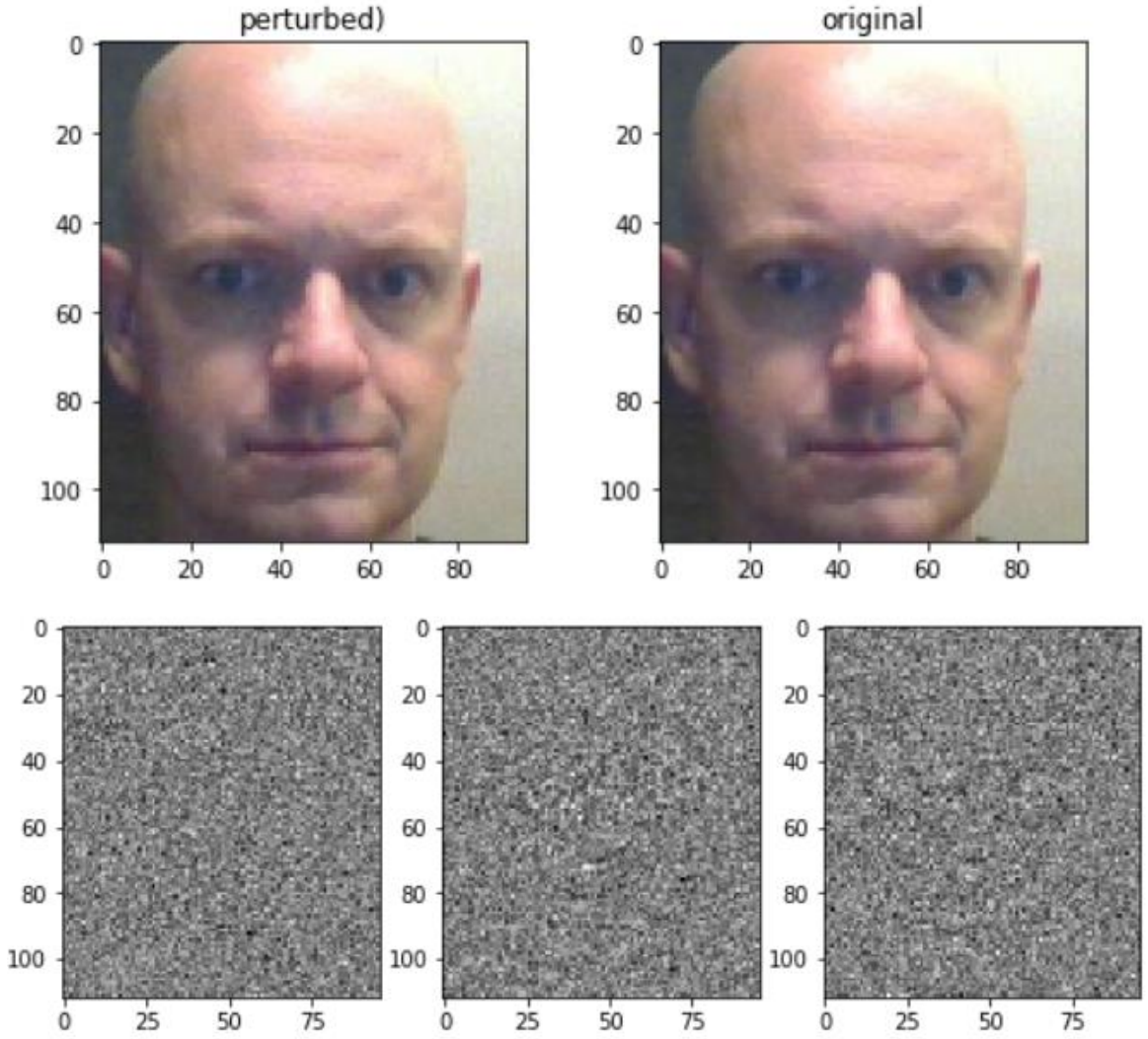
Outline of a black-box attack

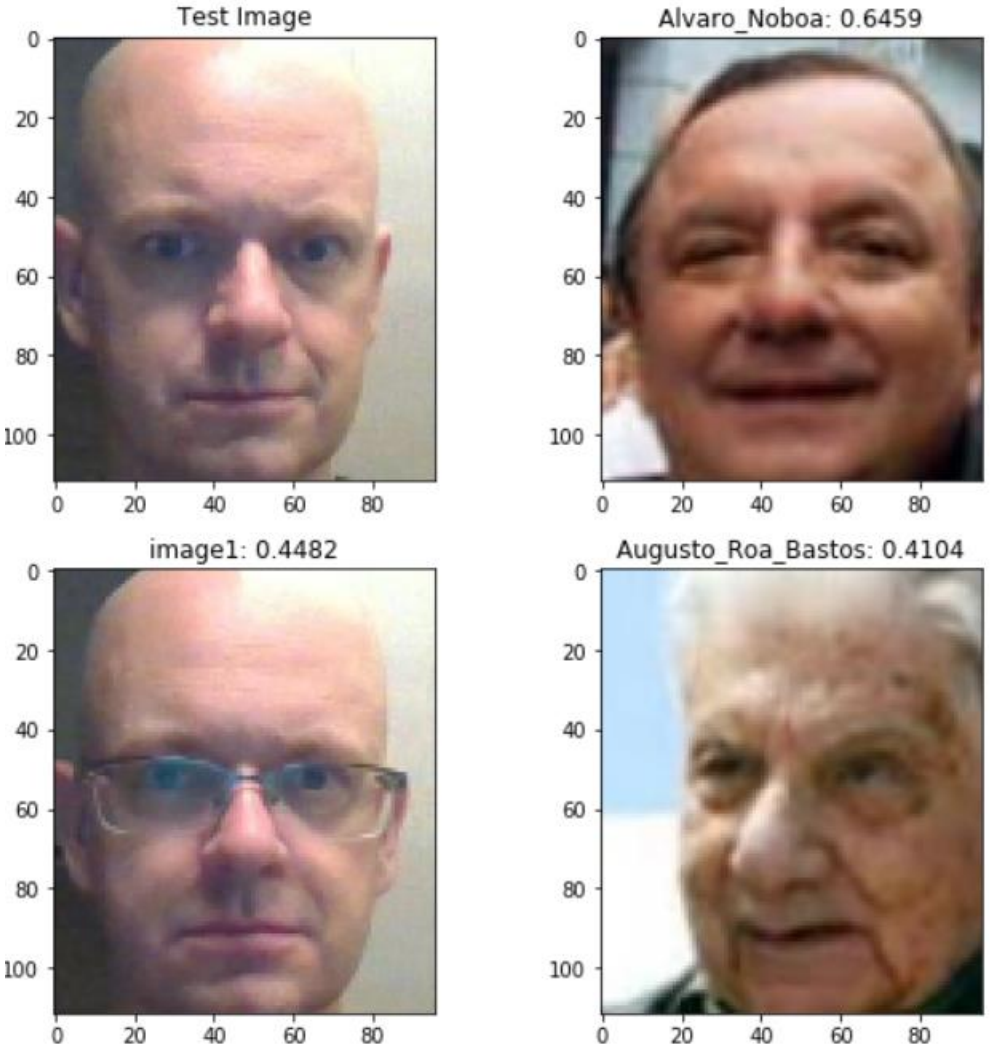
- In the literature:
  - Proxy model attacks (e.g. https://arxiv.org/pdf/1602.02697.pdf)
  - Problem: Two models; might not know what data we need to train proxy model
- Our approach
  - Black-box optimization: genetic algorithm based on model input/output pairs

My Lazy Implementation:

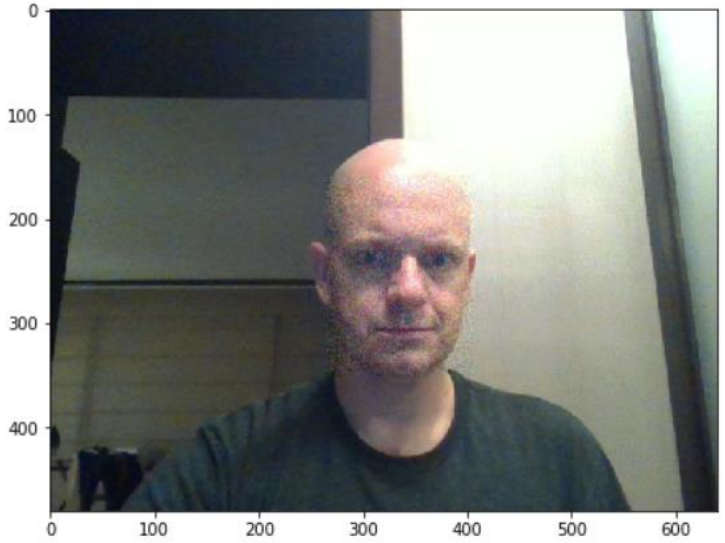
- Treat the perturbation as a numpy array
- Roll L1/L2 penalty into loss
- Fix target “prediction gap”; give no additional loss reduction below that point
- Genetic Algorithm to Glory

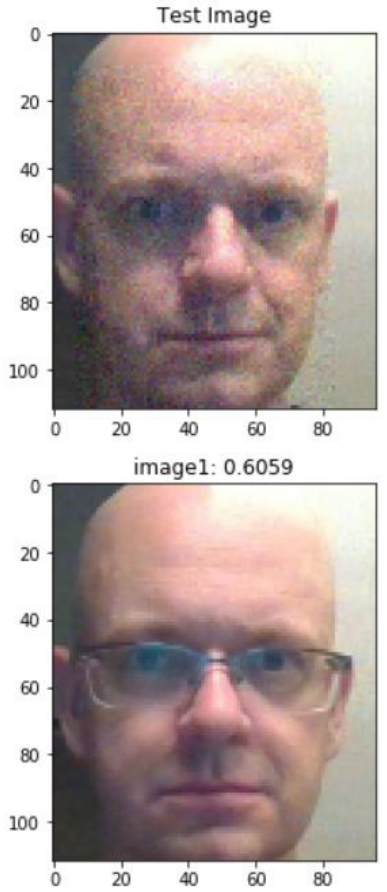
Gradient-free (black box) attack – face chip only

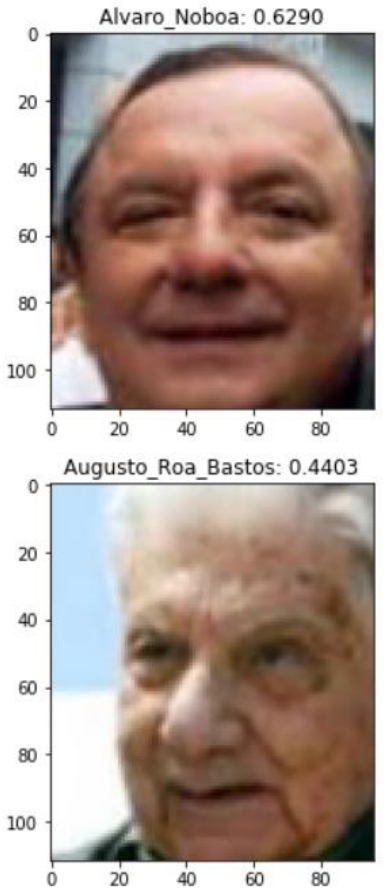




End-to-end; $\mu + 1$ genetic algorithm







Points to consider

- White box – if you can do it – is fast and easy
- Black box requires lots of calls to the model API with nearly-identical images
  - Someone's probably going to notice

What about transferability?

SOPHOS

Example

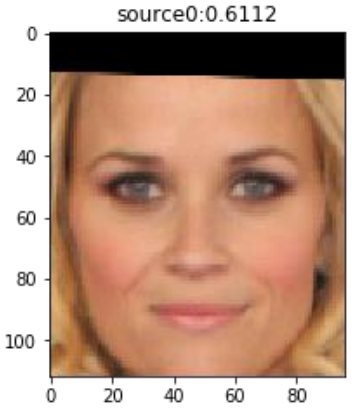
- “Accessorize to a Crime” -- http://users.ece.cmu.edu/~mahmoods/publications/ccs16-adv-ml.pdf
  - Uses restricted number of classes, limited region, smoothness criteria on perturbations, build physical glasses that provide targeted evasions against (now older) VGG-face model

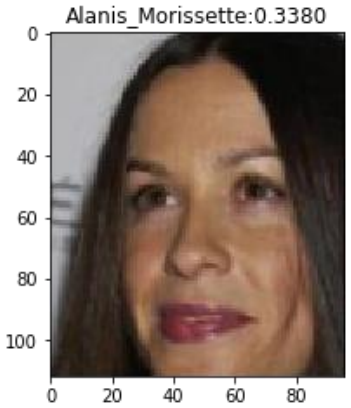
Who would win?

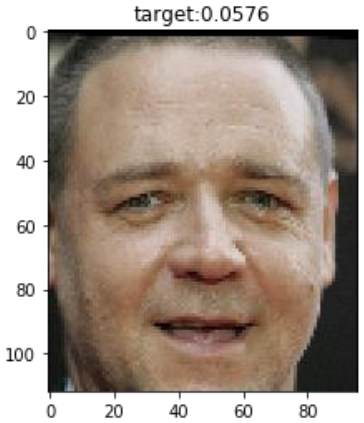
- Some really smart academics -- “Accessorize to a Crime”:
  - Use 2d affine alignment based on facial feature location (Zisserman 2009)
  - Used reduced-class VGG-face for classification
    - O. M. Parkhi, A. Vedaldi, A. Zisserman; “Deep Face Recognition”; British Machine Vision Conference, 2015 (http://www.robots.ox.ac.uk/~vgg/software/vgg_face/)

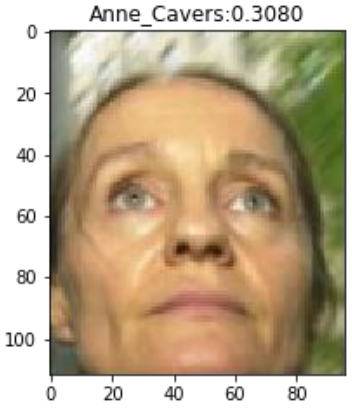
- Me YOLOing stuff together from pre-trained models on github
  - Use 2d affine alignment based on MTCNN facial feature localization
  - Use SphereFace for classification: registered one face from each class in LFW-A (432 faces), as well as the “Accessorize” source and target faces copied from PDF
    - Liu, Wen, Yu, Li, Raj, and Song, 2017. Sphereface: Deep hypersphere embedding for face recognition. IEEE-CCVPR. (https://arxiv.org/abs/1704.08063.pdf)

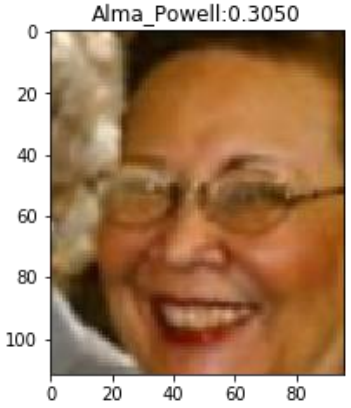
Does it transfer?

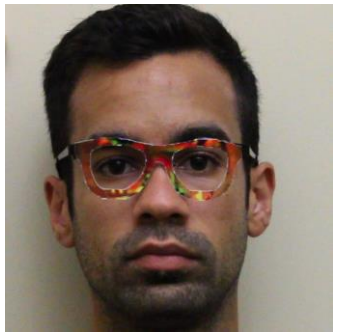


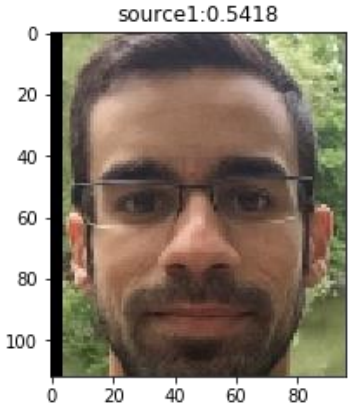


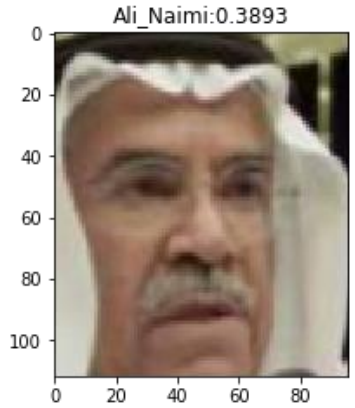


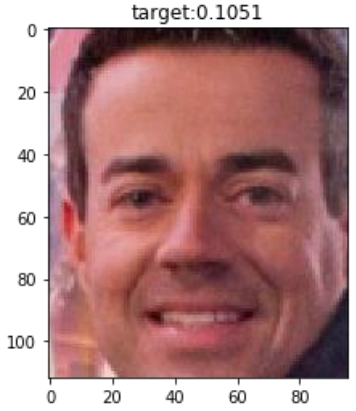


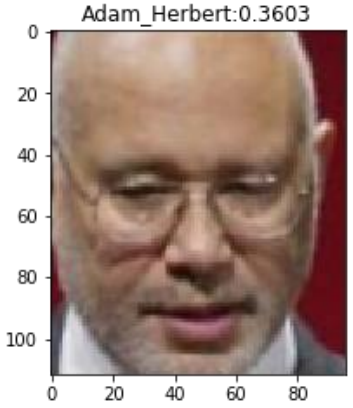




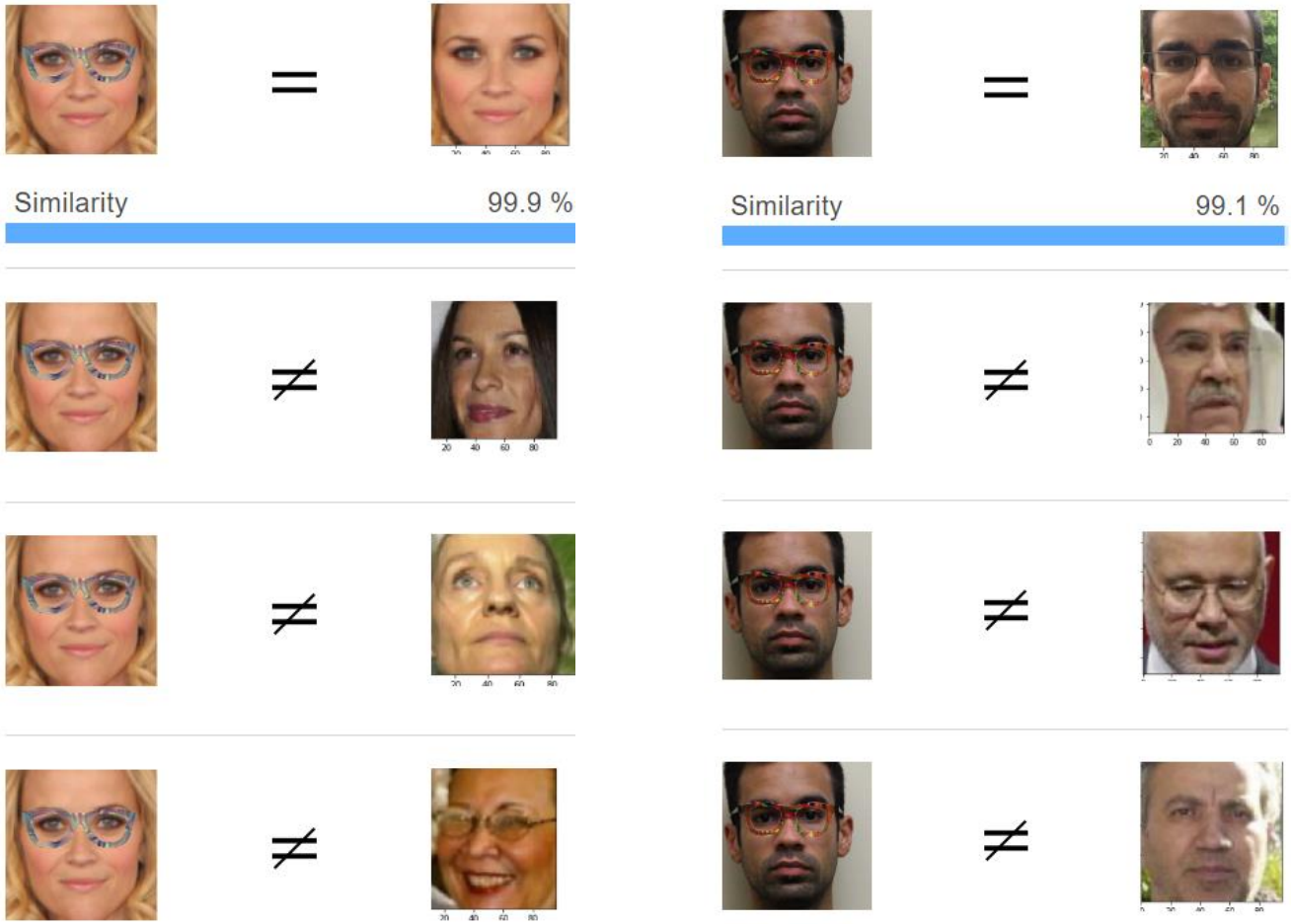




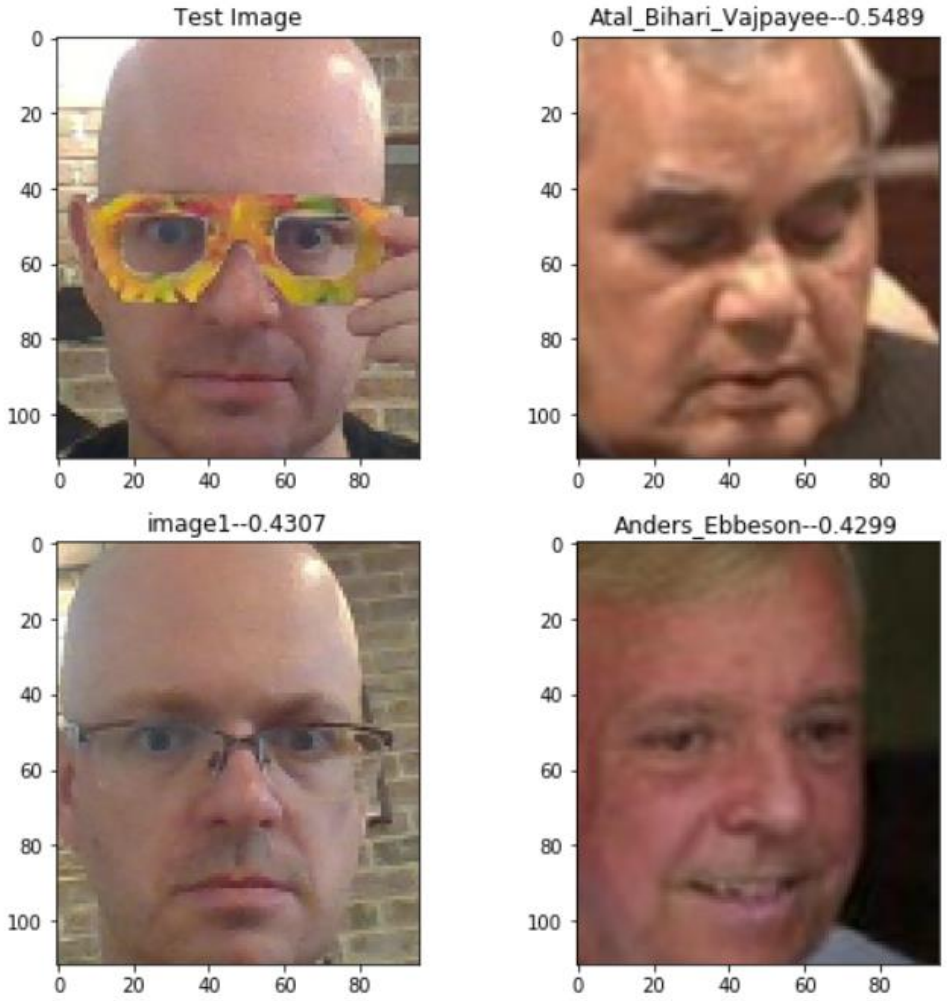


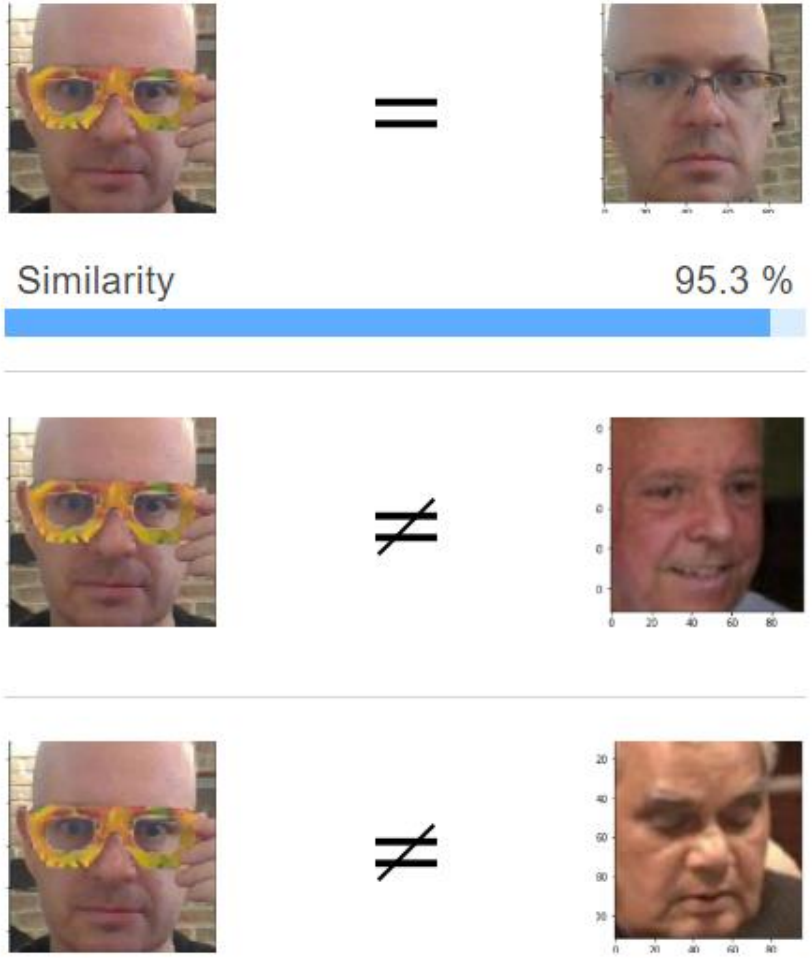


Using Amazon's "Rekognition" demo



One last one


Test Image
Atal_Bihari_Vajpayee--0.5489
image1--0.4307
Anders_Ebbeson--0.4299


Similarity
95.3 %

Wrap-up

- Facial recognition is different from ‘normal’ image classification
- These tools are already widely available and easy to use
- Differences make some attacks harder, but also open up new potential attack surfaces…
  - Gradient-based attacks in particular harder to use “end-to-end” – try black-box optimization like Genetic Algorithms (but noisy)
  - Exhaustion attacks are a new possibility
  - Tampering with embeddings – if possible – is direct and extremely powerful

Wrap-up

- Play around yourself: ready-to-rock pre-trained system available at:
  - https://colab.research.google.com/drive/1GZP4sqdtdiaAl7gsZgGpnsAbwPvvKx8v
  - TinyURL: **https://preview.tinyurl.com/yyb4jve9**
  - (or https://tinyurl.com/yyb4jve9)

Google account required:

1. Open link
2. Select “File -> Open in playground mode”
3. Save your own copy to preserve changes you make to it.

SOPHOS

Cybersecurity evolved.