VISUAL RECOGNITION

ANUSH SANKARAN
IBM RESEARCH

Most of the slide content are borrowed from CS231n. Thanks to the original authors

EASY LOGISTICS: GITHUB

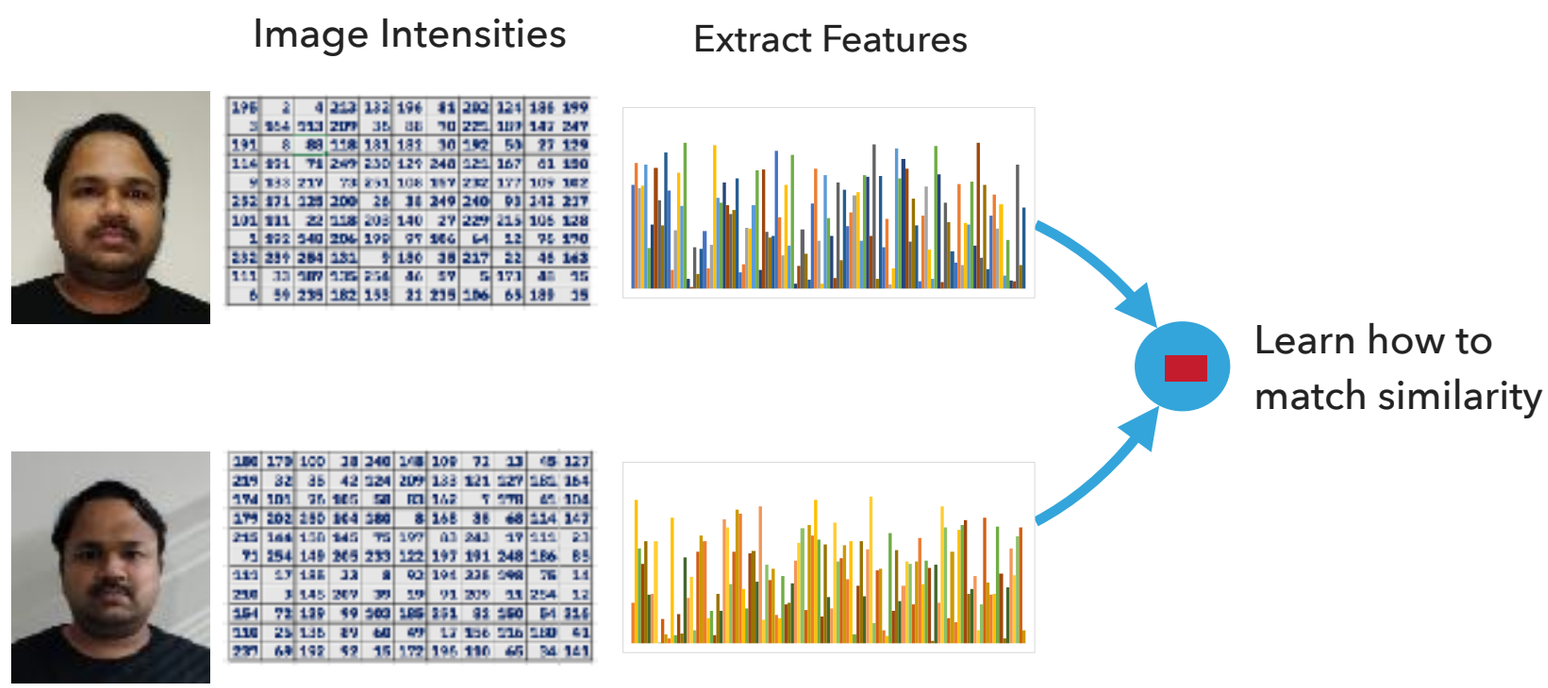
- Github Repo: https://github.com/goodboyanush/iiit-bangalore-march-april-2019
- Lectures slides, Hands-on notebooks, Assignment solutions
- Have any doubt in my lectures or assignments?
  - Go ahead and create an issue in the repo!
  - I will try to answer them asap!
  - Everyone will be benefitted by the questions asked by one student

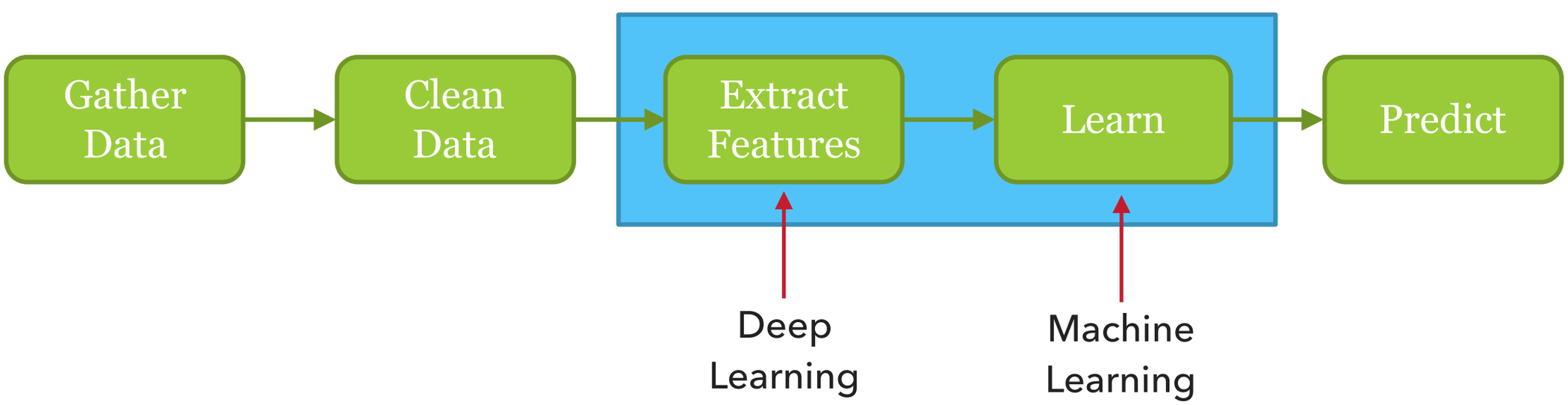
RECAP: WHAT IS THE PROBLEM?

- You are a genius …
- Only if a machine or an algorithm could do it …

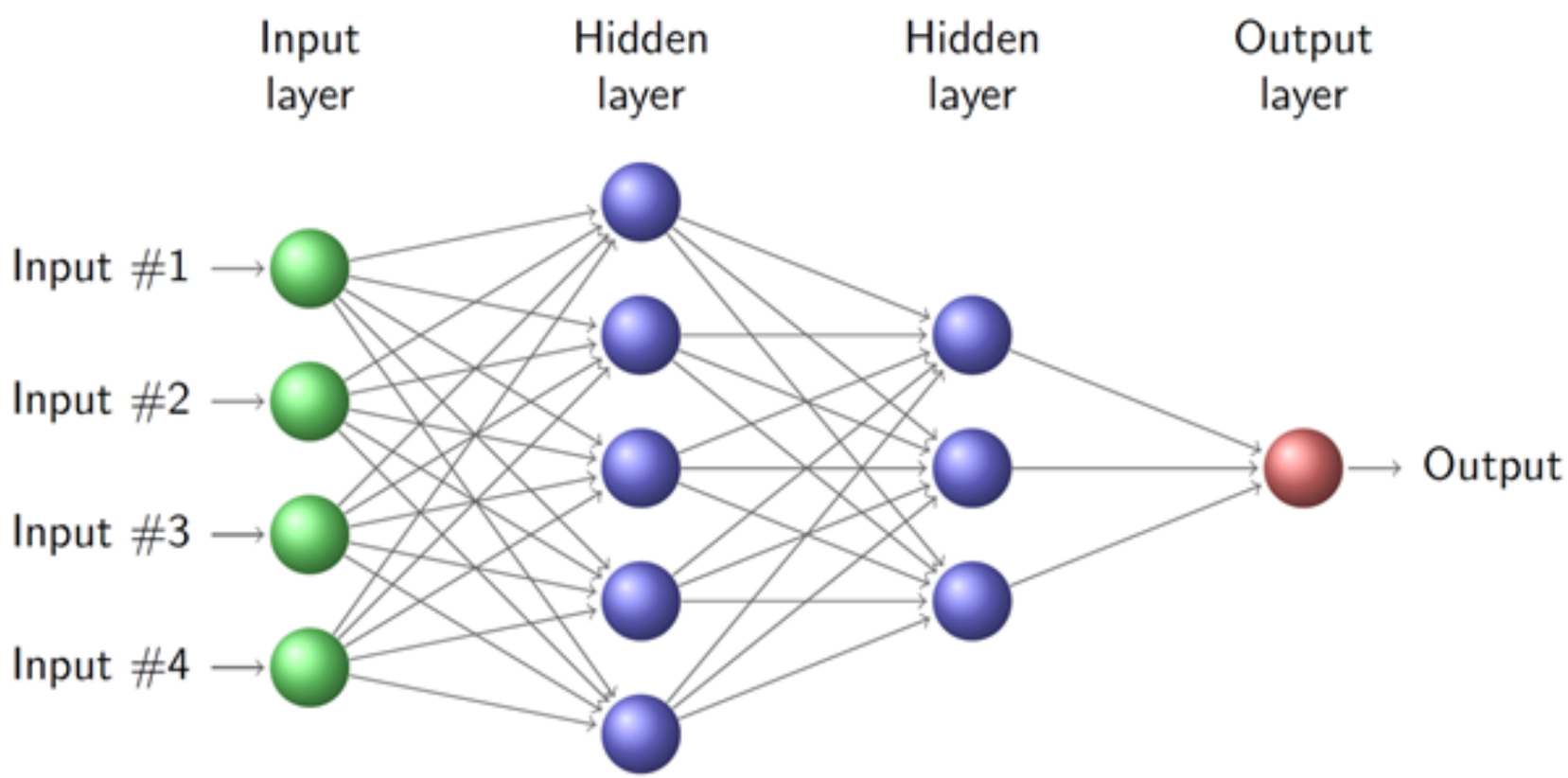
- Make a computer understand the image …
- Understand the background & environment …
- Classify various objects in the image …

RECAP: WHAT IS THE SOLUTION PIPELINE

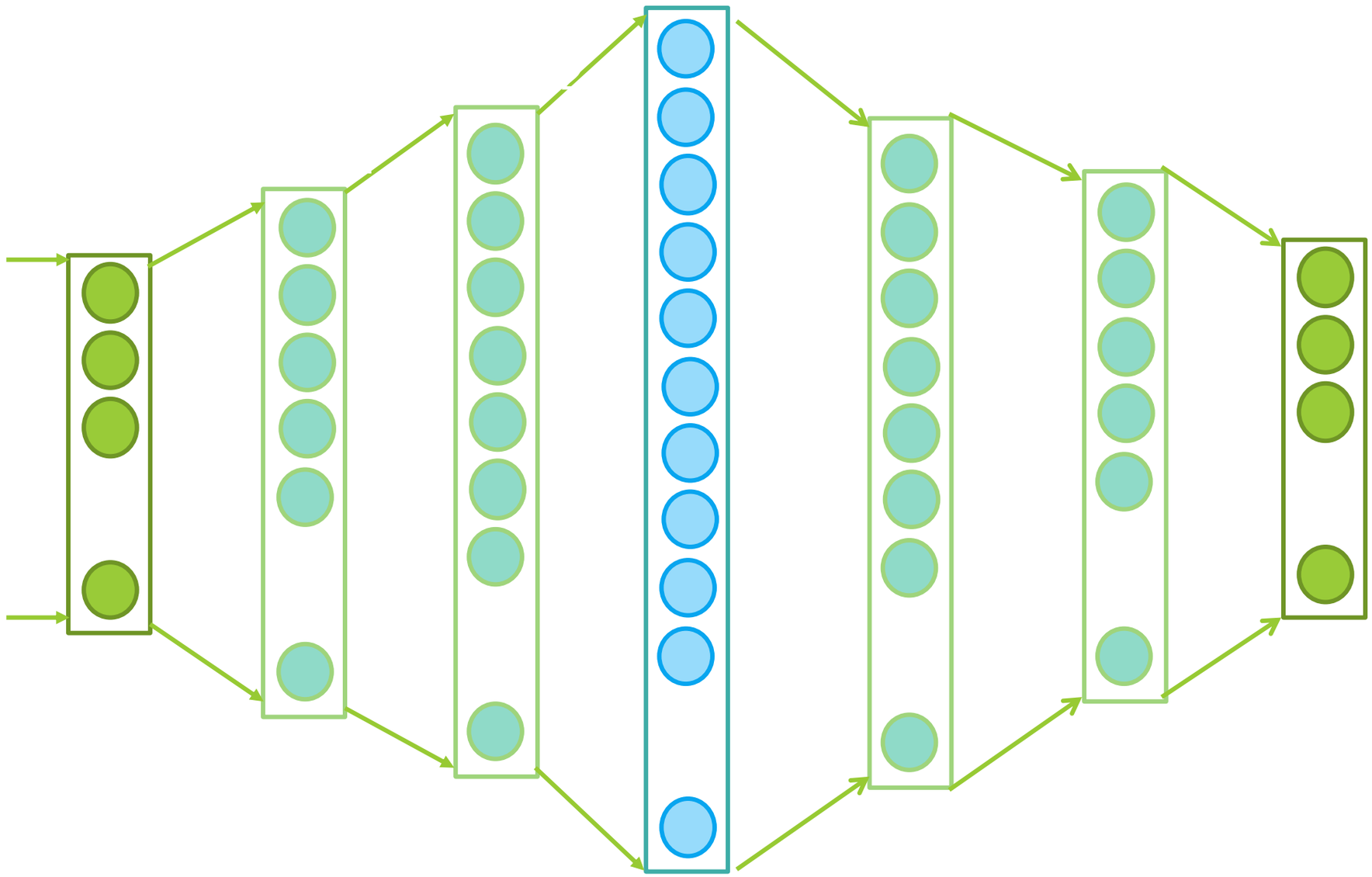




RECAP: MULTI LAYER PERECEPTRON



RECAP: STACKED AUTOENCODER



RECAP: DIFFERENT TYPES OF DEEP LEARNING

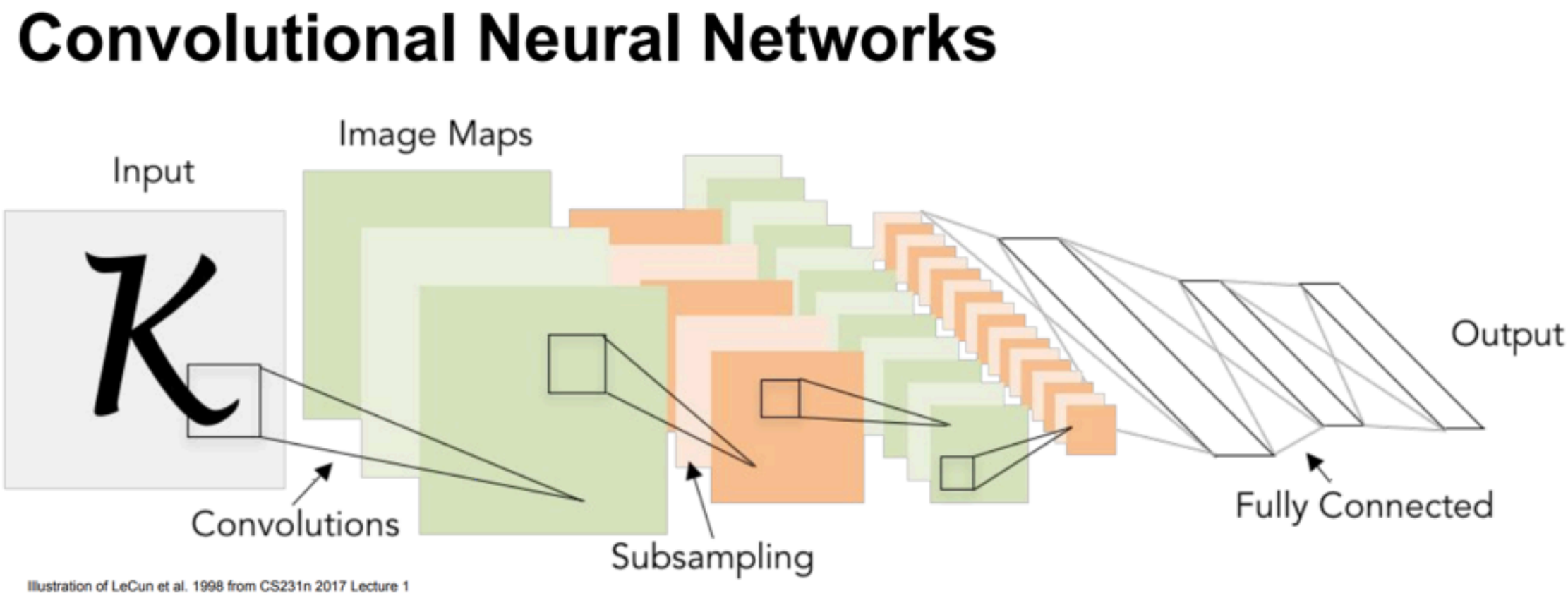
- Input data is a vector Stacked
  - Autoencoder
  - Restricted Boltzmann Machines
  - Deep Belief Networks

- Input data is multi-dimensional
  - Convolutional Neural Network
  - Example: Image analysis, Object recognition

- Input data is time-series
  - Recurrent Neural Network
  - Example: Language Modelling, Speech recognition

RECAP: CONVOLUTIONAL NEURAL NETWORK

Where we are now...



RECAP: WHAT HYPER-PARAMETERS TO PLAY WITH?

- Network architecture
  - How many layers?
  - What sequence of layers?
  - What is the depth of the model?
- Learning rate
  - Decay rate
  - Update strategy
- Regularization
  - L2 / Dropout strength
- Which loss function?
  - hinge/ L2/ Cross-entropy
- Which optimization?
  - SGD/ adam/ RMSPROP

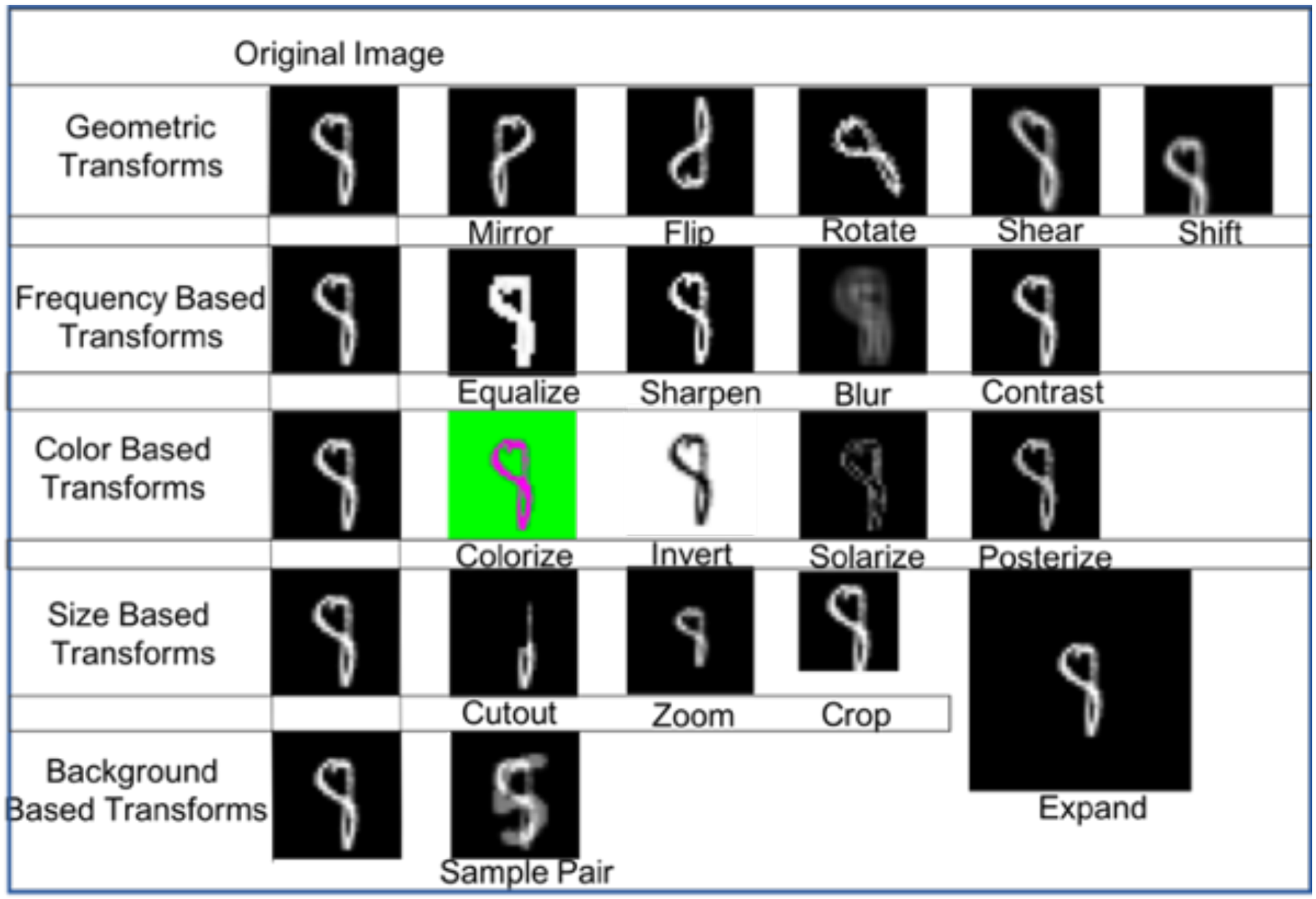
RECAP: MOST IMPORTANT CHALLENGE

CNN Model has lot of weight parameters to learn…

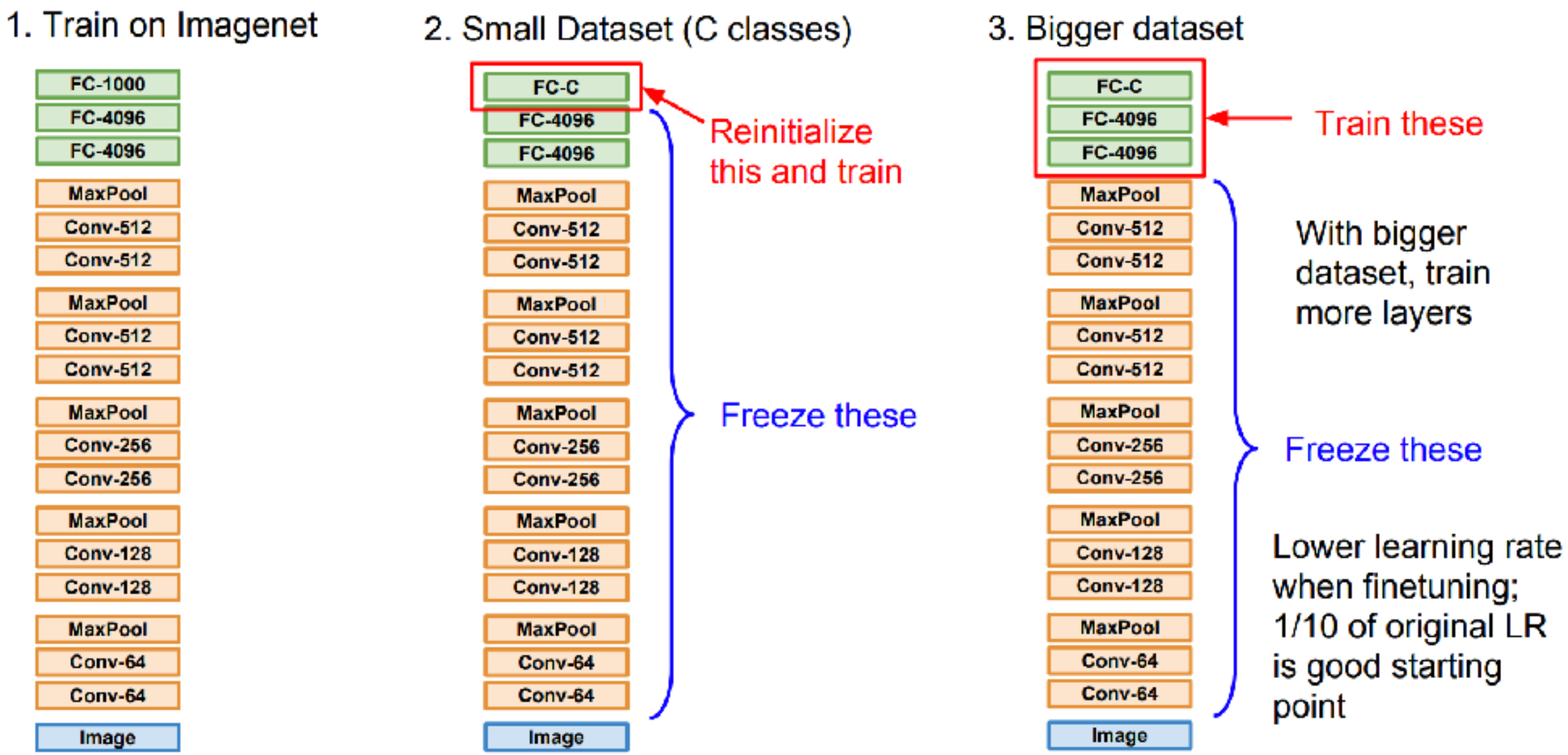
"You need lots of data to train/use a CNN model"

1. Data Augmentation
2. Transfer Learning

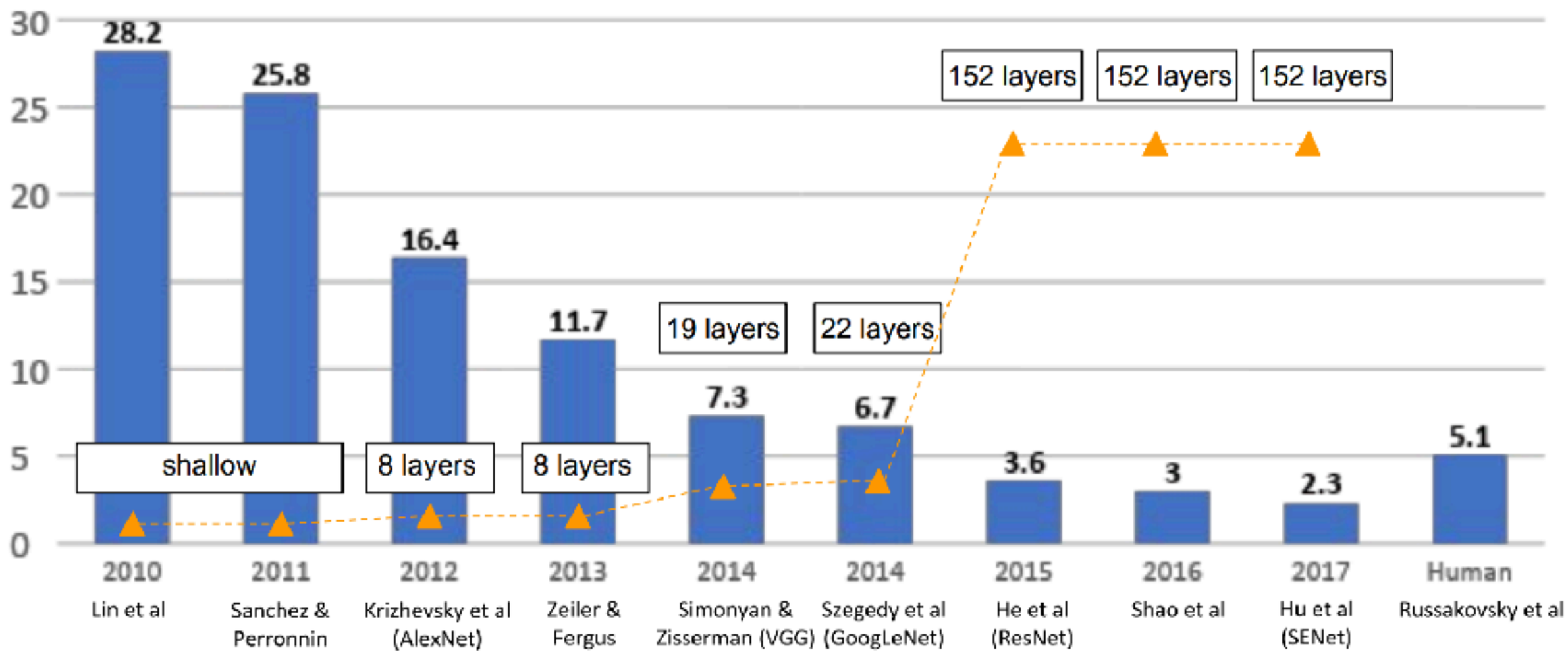
RECAP: DATA AUGMENTATION



RECAP: TRANSFER LEARNING



RECAP: IMAGENET CHALLENGE WINNERS



RECAP: CNN FOR UNSUPERVISED LEARNING – GENERATIVE MODELLING

Given training data, generate new samples from same distribution

Training data ~ $p_{data}(x)$

Generated samples ~ $p_{model}(x)$

Want to learn $p_{model}(x)$ similar to $p_{data}(x)$

Addresses density estimation, a core problem in unsupervised learning

Several flavors:

- Explicit density estimation: explicitly define and solve for $p_{model}(x)$
- Implicit density estimation: learn model that can sample from $p_{model}(x)$ w/o explicitly defining it

RECAP: TAXONOMY OF GENERATIVE MODELS

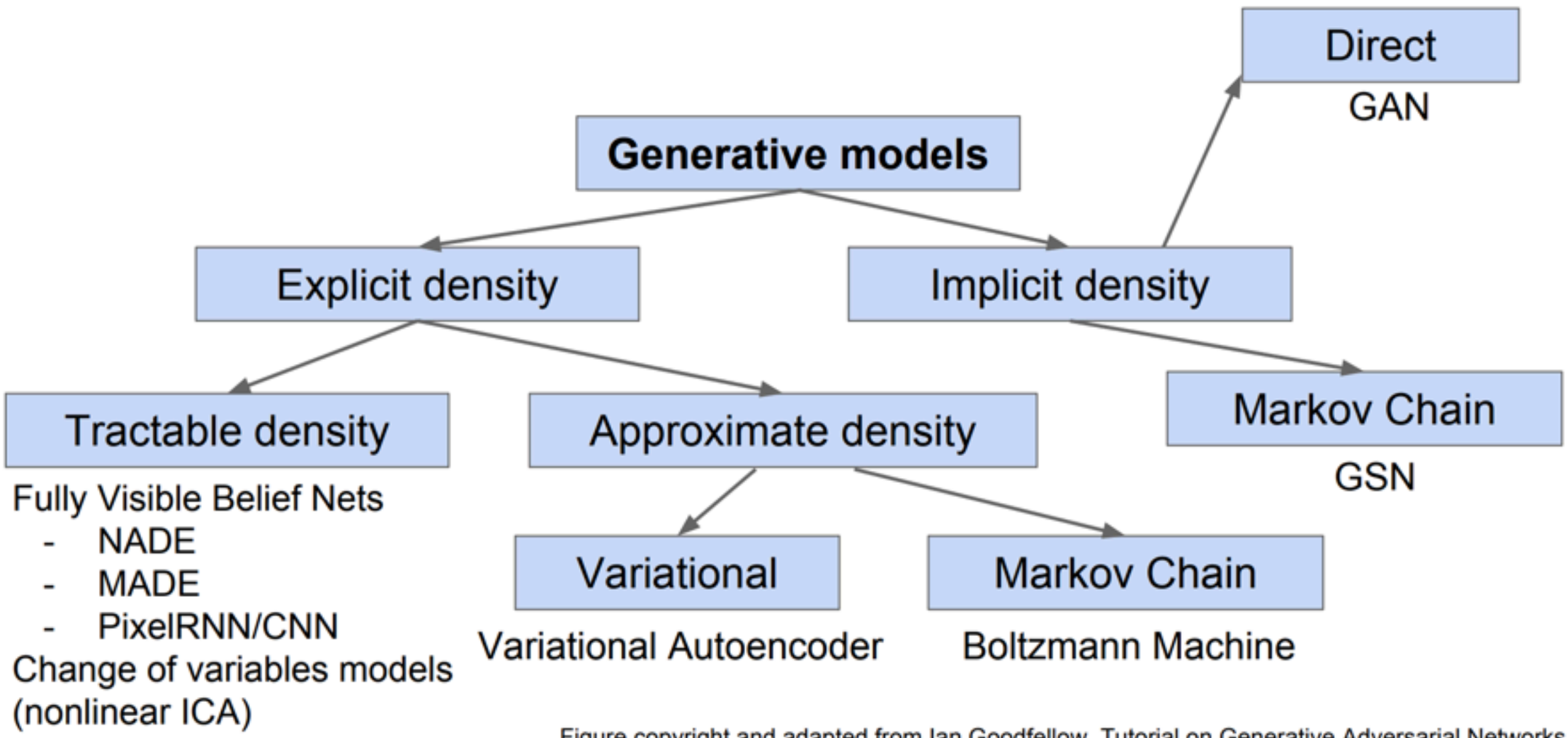


Figure copyright and adapted from Ian Goodfellow, Tutorial on Generative Adversarial Networks, 2017.

RECAP: GENERATIVE MODELS

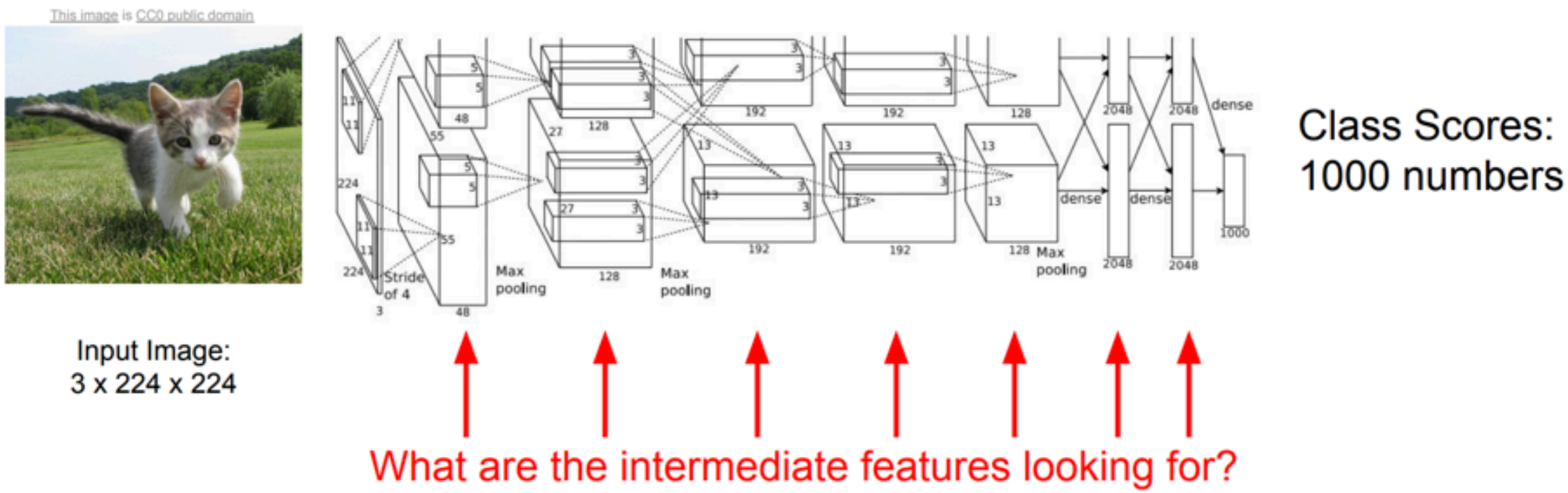
Generative Models

- PixelRNN and PixelCNN — Explicit density model, optimizes exact likelihood, good samples. But inefficient sequential generation.
- Variational Autoencoders (VAE) — Optimize variational lower bound on likelihood. Useful latent representation, inference queries. But current sample quality not the best.
- Generative Adversarial Networks (GANs) — Game-theoretic approach, best samples! But can be tricky and unstable to train, no inference queries.

Also recent work in combinations of these types of models! E.g. Adversarial Autoencoders (Makhanzi 2015) and PixelVAE (Gulrajani 2016)

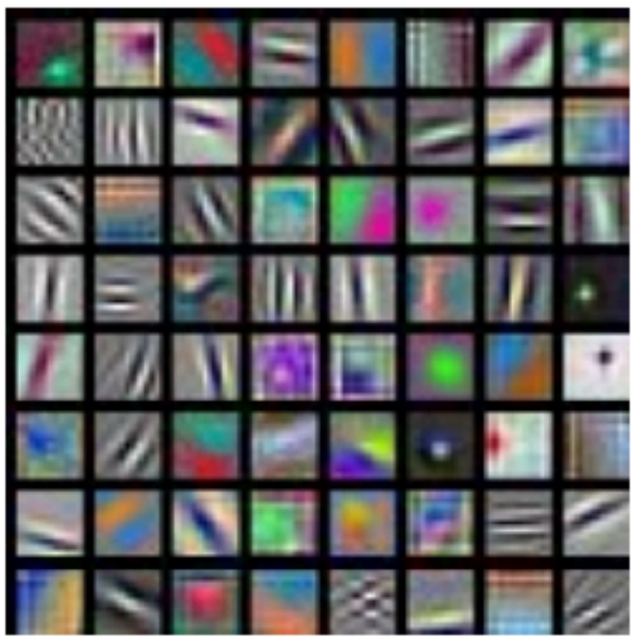
CNN VISUALIZATION

WHAT IS THE CNN LEARNING?



- Visualize Filters/ Kernels
- Visualize Activation Maps

FIRST LAYER FILTER VISUALIZATION



AlexNet:
64 x 3 x 11 x 11

ResNet-18:
64 x 3 x 7 x 7

ResNet-101:
64 x 3 x 7 x 7

DenseNet-121:
64 x 3 x 7 x 7

VISUALIZE ALL THE LAYERS

Visualize the filters/kernels (raw weights)

We can visualize filters at higher layers, but not that interesting

(these are taken from ConvNetJS CIFAR-10 demo)

layer 1 weights

16 x 3 x 7 x 7

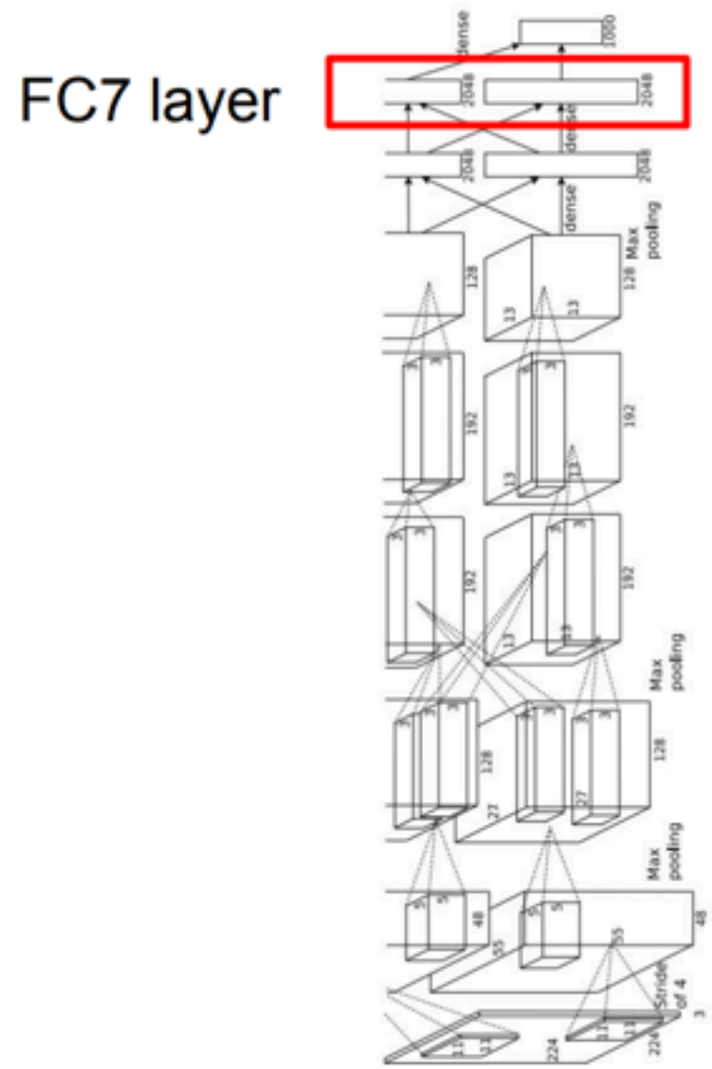
layer 2 weights

20 x 16 x 7 x 7

layer 3 weights

20 x 20 x 7 x 7

FULLY CONNECTED LAYER VISUALIZATION



4096-dimensional feature vector for an image (layer immediately before the classifier)

Run the network on many images, collect the feature vectors

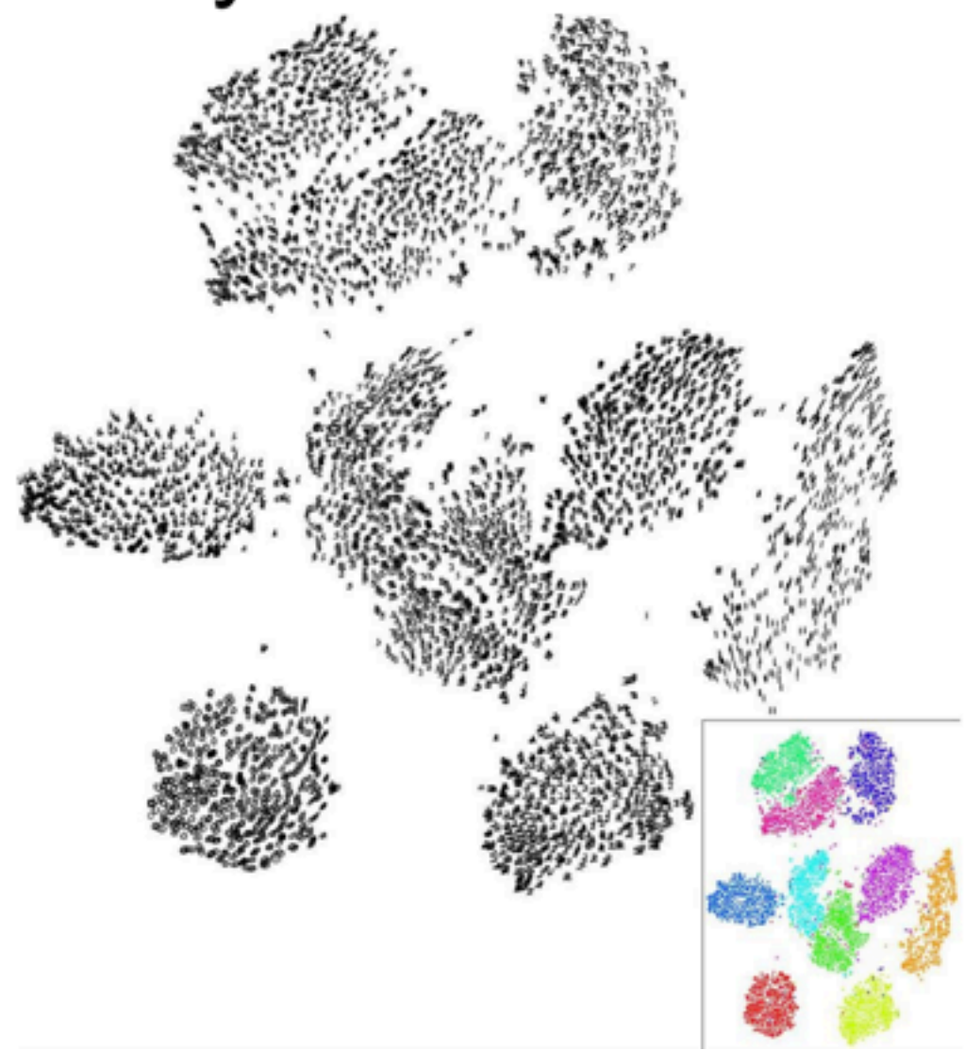
FULLY CONNECTED LAYER VISUALIZATION

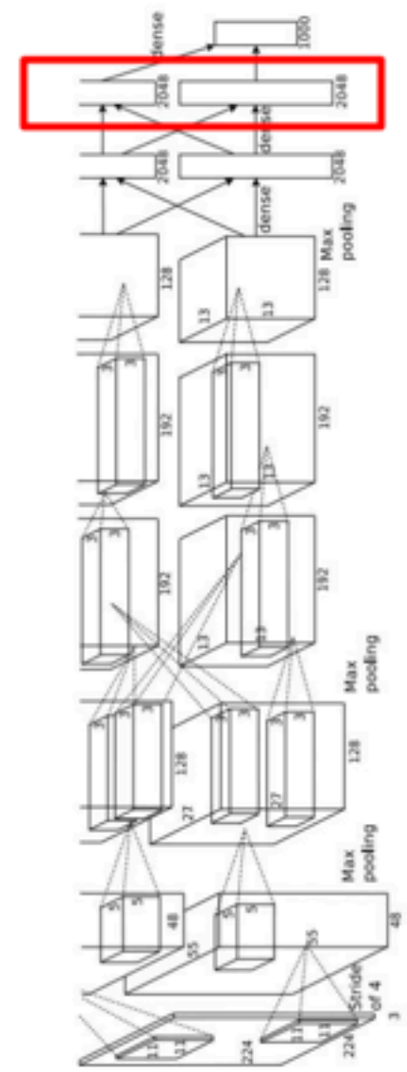
Last Layer: Dimensionality Reduction

Visualize the “space” of FC7 feature vectors by reducing dimensionality of vectors from 4096 to 2 dimensions

Simple algorithm: Principal Component Analysis (PCA)

More complex: **t-SNE**

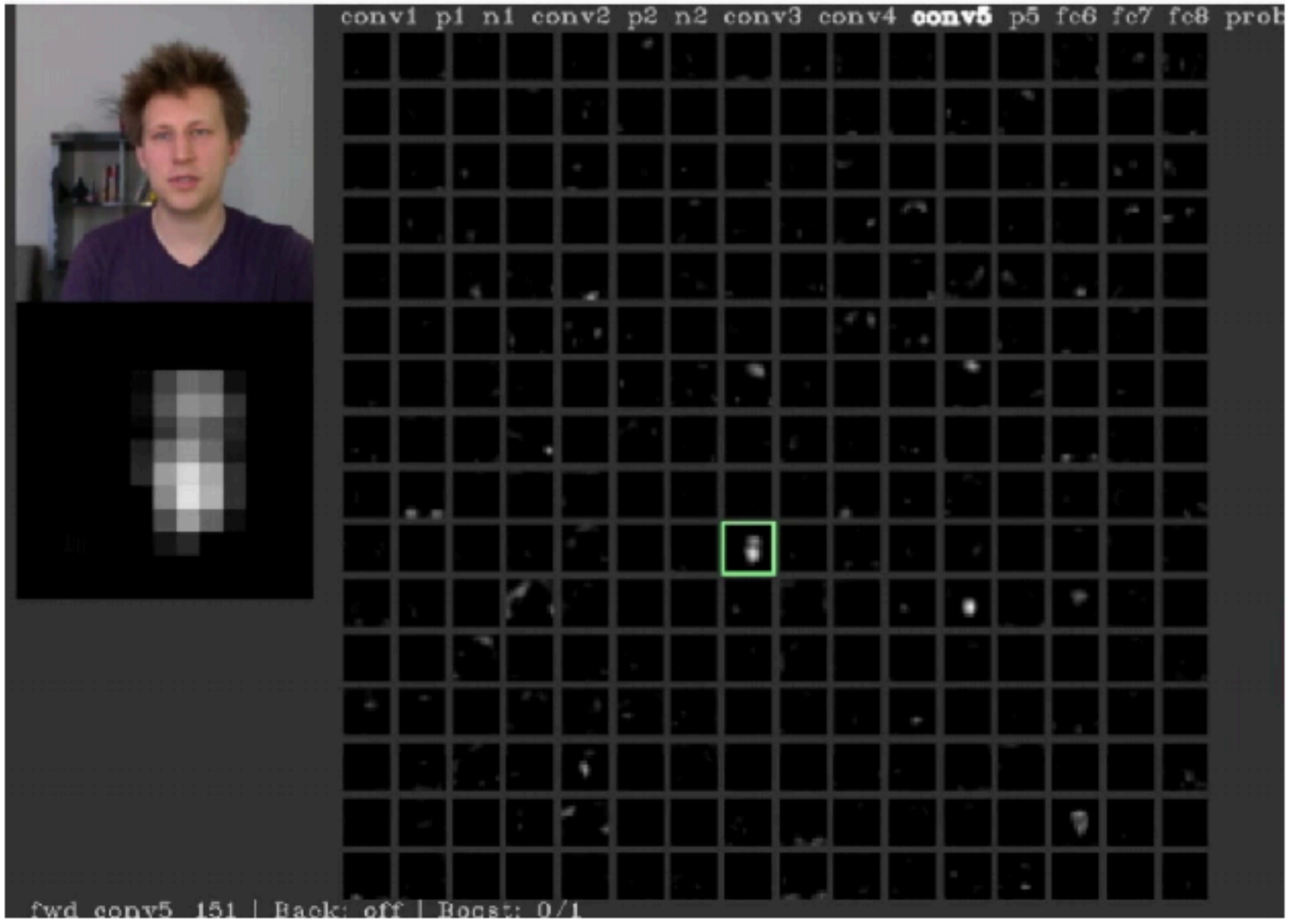




Van der Maaten and Hinton, “Visualizing Data using t-SNE”, JMLR 2008
Figure copyright Laurens van der Maaten and Geoff Hinton, 2008. Reproduced with permission.

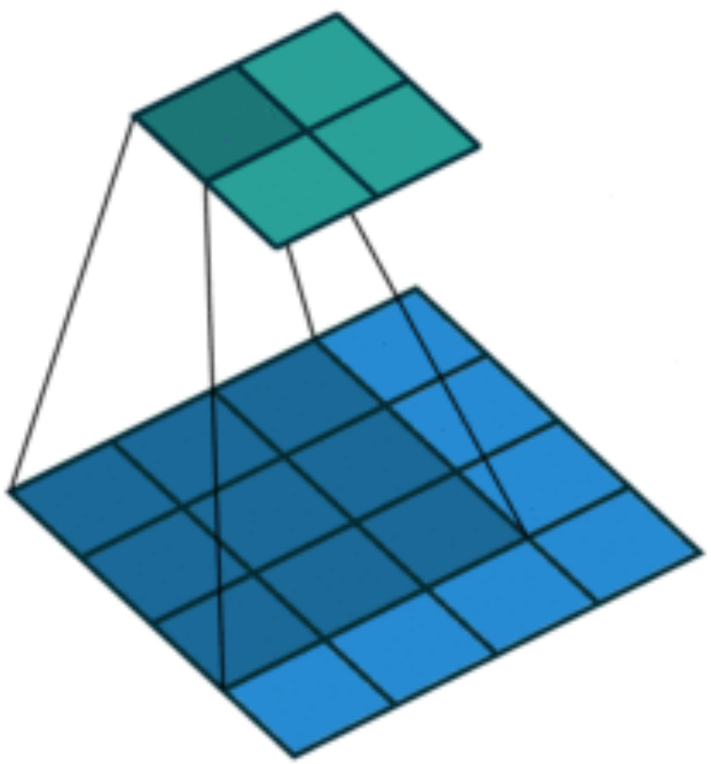
VISUALIZING ACTIVATIONS

conv5 feature map is 128x13x13; visualize as 128 13x13 grayscale images

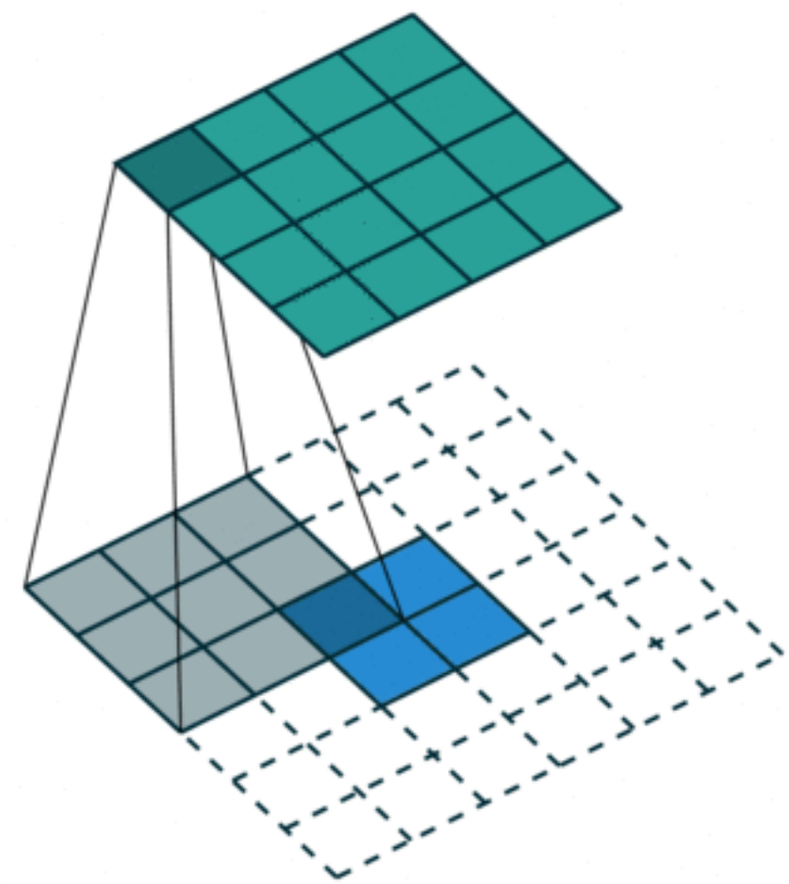


DECONVOLUTION LAYERS

Convolutional Layer

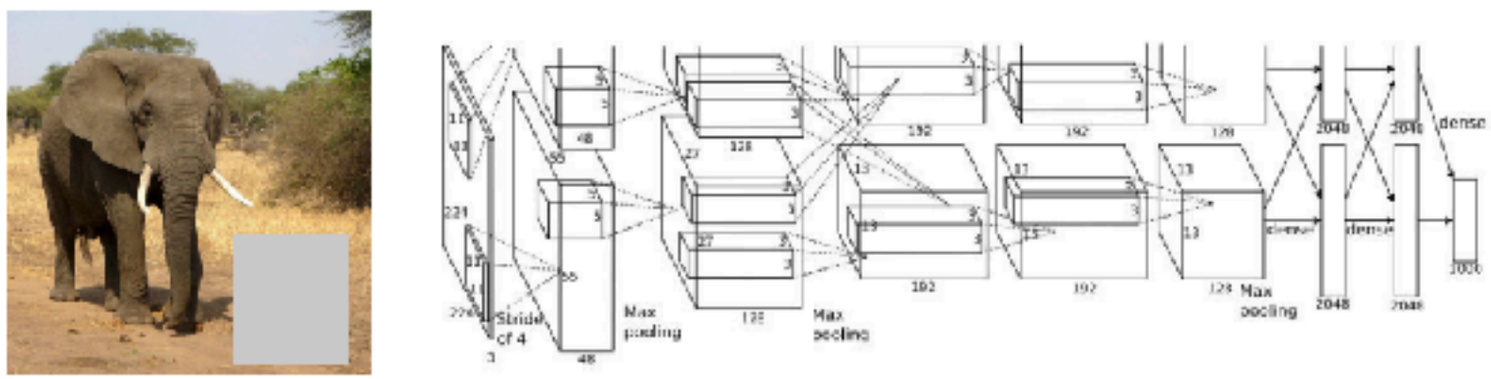


Deconvolutional / Transposed Convolution Layer

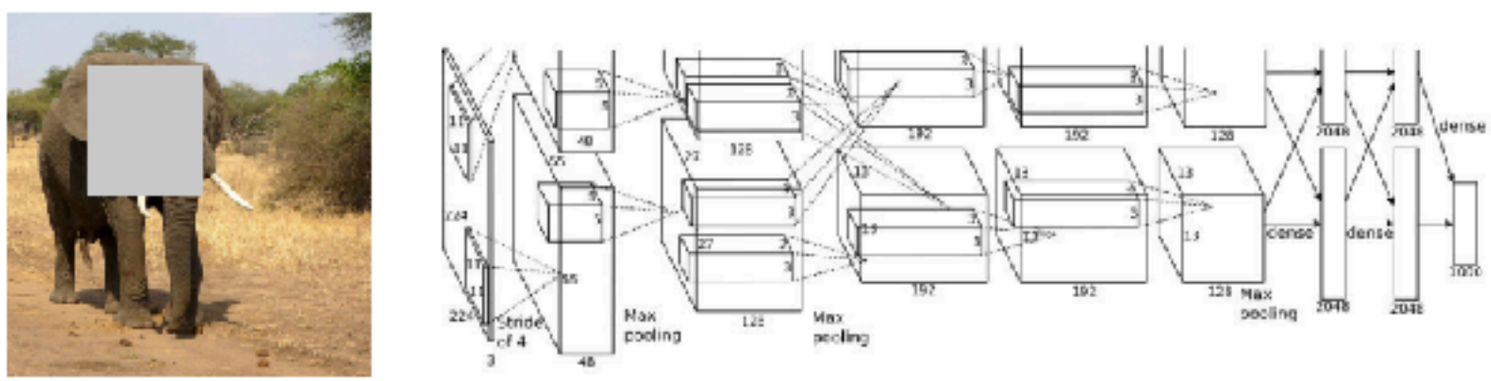


WHICH PIXELS MATTER: SALIENCY

Mask part of the image before feeding to CNN,
check how much predicted probabilities change



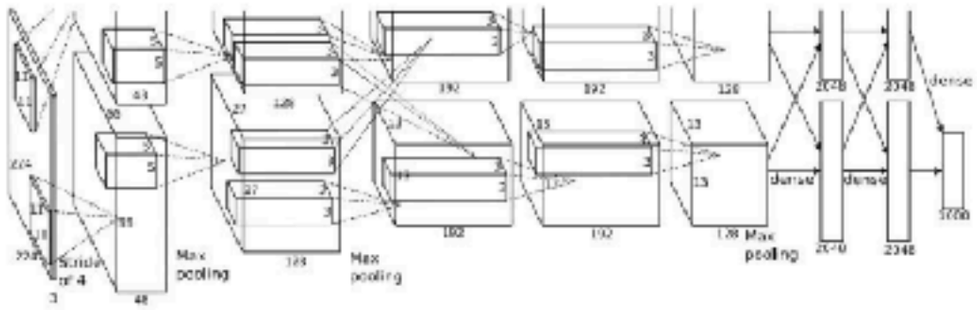
P(elephant) = 0.95



P(elephant) = 0.75

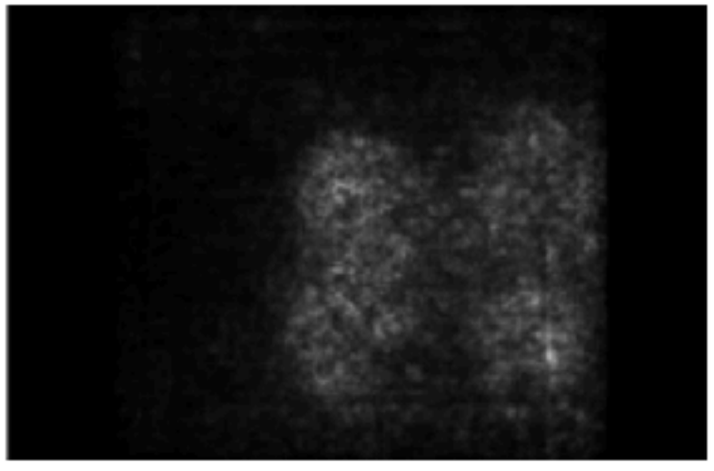
WHICH PIXELS MATTER: SALIENCY VIA BACKPROP

Forward pass: Compute probabilities



Dog

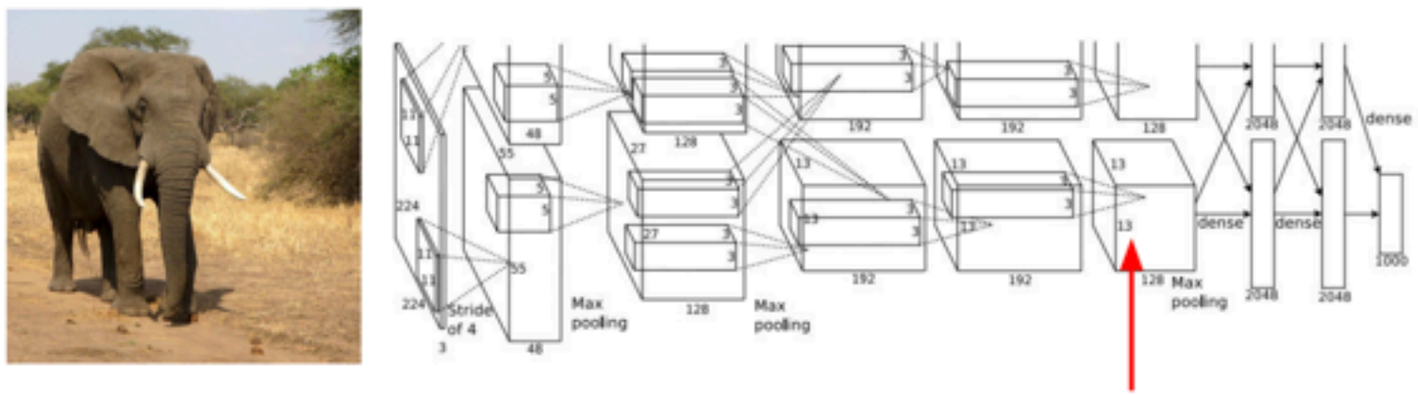
Compute gradient of (unnormalized) class score with respect to image pixels, take absolute value and max over RGB channels



Simonyan, Vedaldi, and Zisserman, "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps", ICLR Workshop 2014.

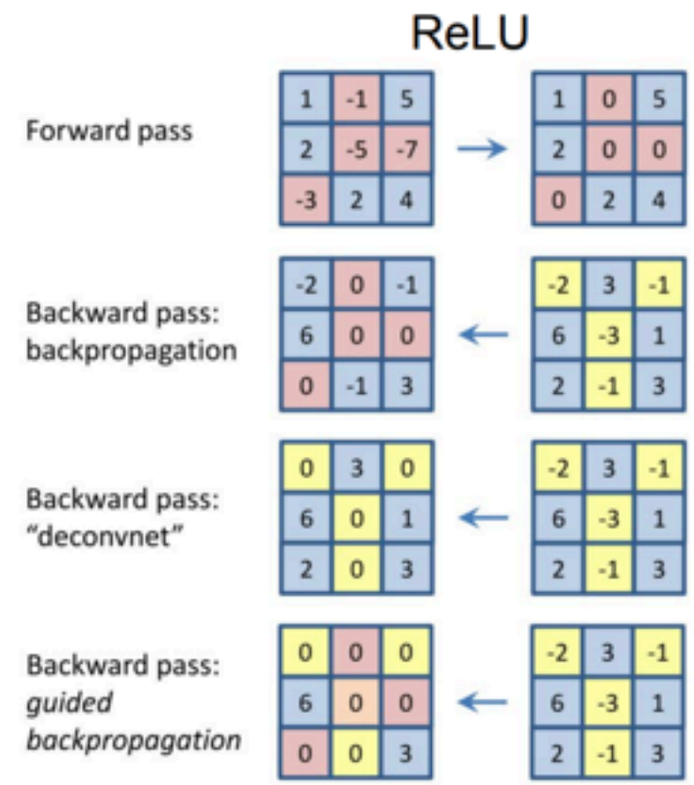
Figures copyright Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman, 2014; reproduced with permission.

GUIDED BACKPROP



Pick a single intermediate neuron, e.g. one value in 128 x 13 x 13 conv5 feature map

Compute gradient of neuron value with respect to image pixels

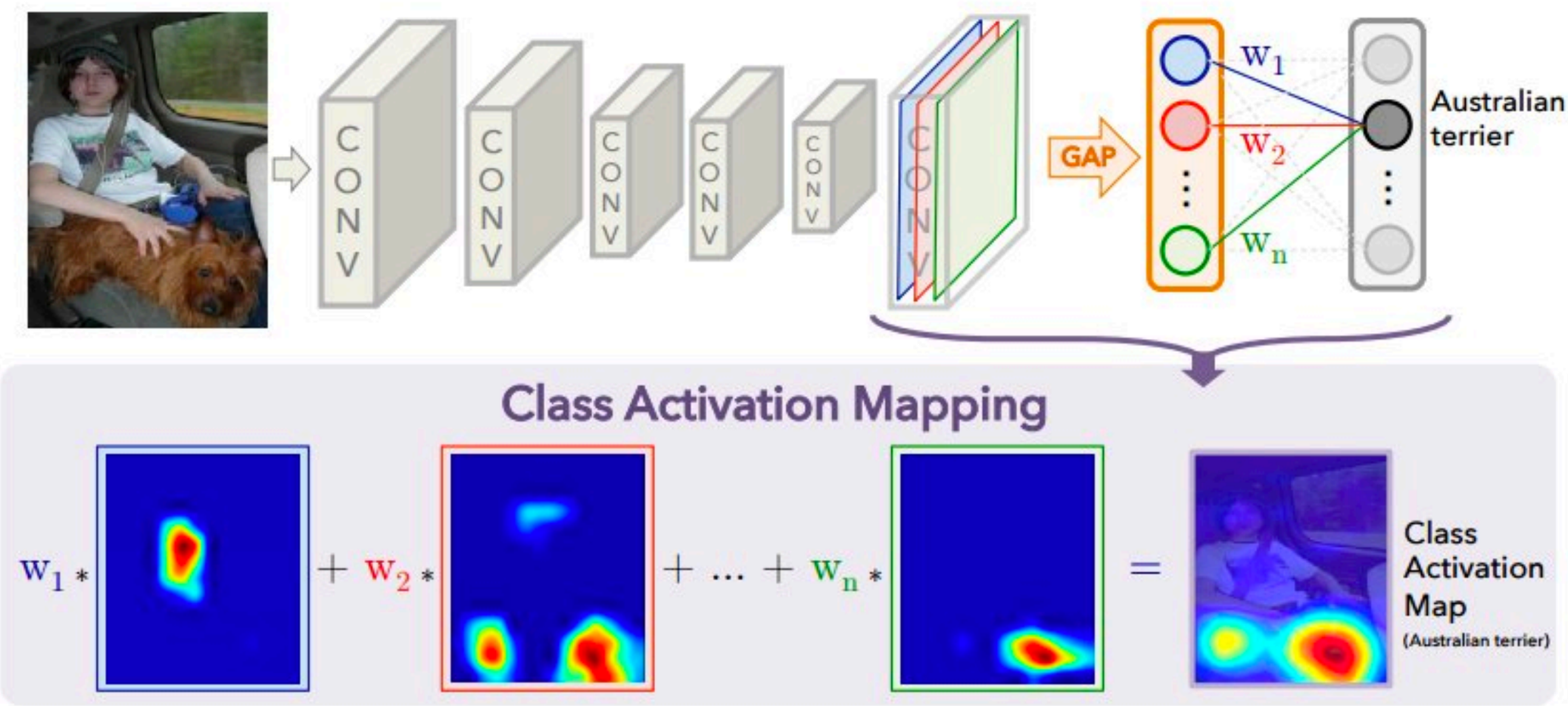


Images come out nicer if you only backprop positive gradients through each ReLU (guided backprop)

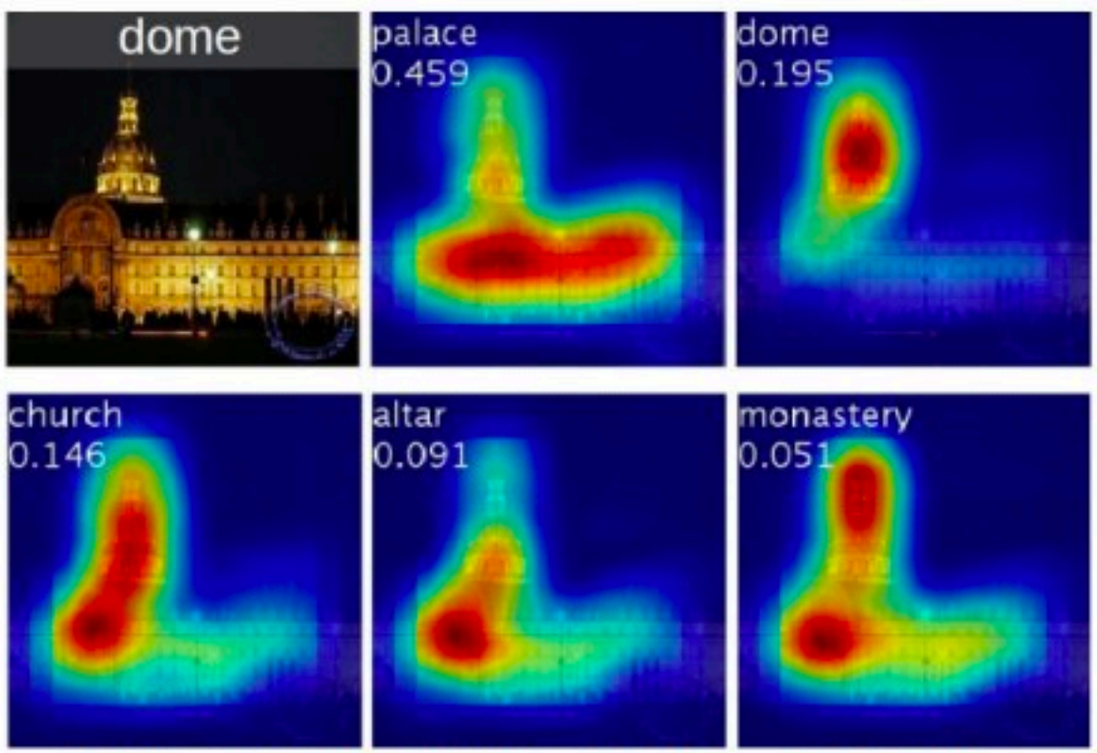
Zeiler and Fergus, "Visualizing and Understanding Convolutional Networks", ECCV 2014
Springenberg et al, "Striving for Simplicity: The All Convolutional Net", ICLR Workshop 2015

Figure copyright Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, Martin Riedmiller, 2015; reproduced with permission.

CLASS ACTIVATION MAPS (CAM)



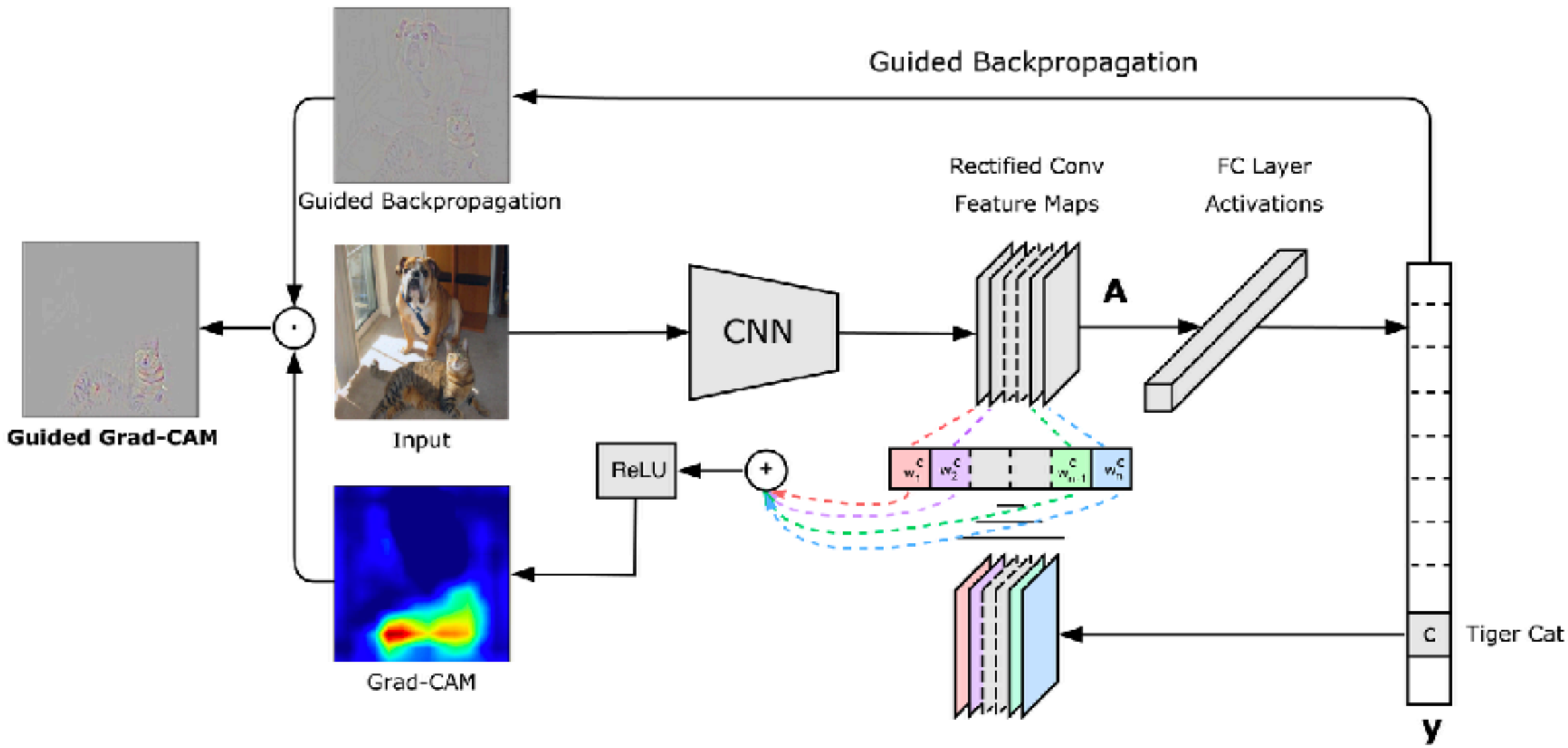
CLASS ACTIVATION MAPS (CAM)



Class activation maps of top 5 predictions

Class activation maps for one object class

GRAD-CAM


Guided Backpropagation
Guided Backpropagation
Rectified Conv
Feature Maps
FC Layer
Activations
A
CNN
Guided Grad-CAM
Input
ReLU
Grad-CAM
c
Tiger Cat
y

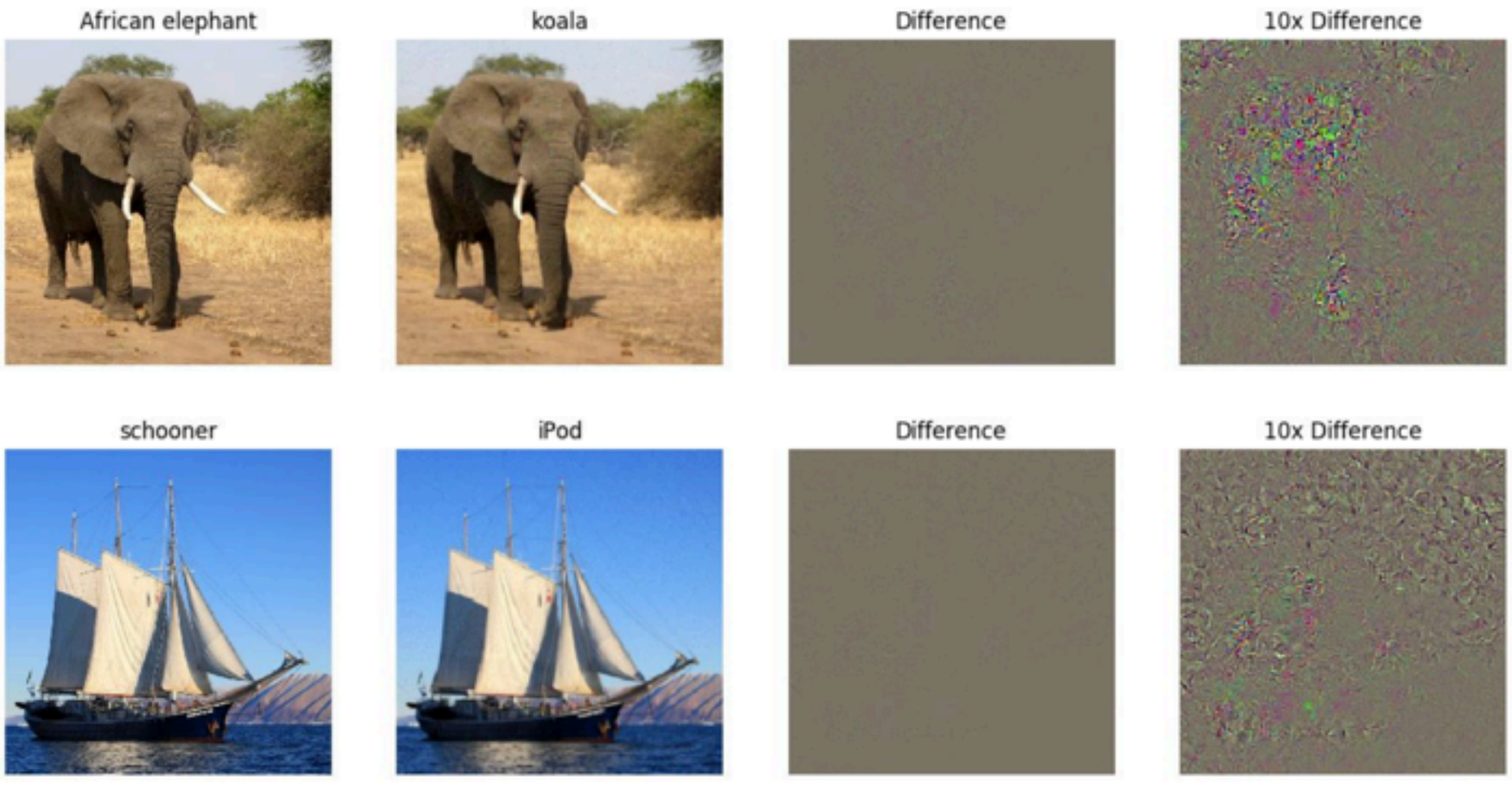
IBM RESEARCH AI

VISUALIZATION IN DEEP LEARNING

		WHY				WHO			WHAT					HOW						WHEN		WHERE
Author	Year	Interpretability & Explainability	Debugging & Improving Models	Comparing & Selecting Models	Education	Model Developers & Builders	Model Users	Non-experts	Computational Graph & Network Architecture	Learned Model Parameters	Individual Computational Units	Neurons in High-dimensional Space	Aggregated Information	Node-link Diagrams for Network Architecture	Dimensionality Reduction & Scatter Plots	Line Charts for Temporal Metrics	Instance-based Analysis & Exploration	Interactive Experimentation	Algorithms for Attribution & Feature Visualization	During Training	After Training	Publication Venue
Abadi, et al.	2016	■	■	■		■	■						■			■				■	■	arXiv
Bau, et al.	2017	■		■		■					■						■		■		■	CVPR
Bilal, et al.	2017	■	■			■					■		■				■		■	■		TVCG
Bojarski, et al.	2016	■	■			■				■			■				■		■	■	■	arXiv
Bruckner	2014	■	■			■			■	■				■			■		■	■		MS Thesis
Carter, et al.	2016	■			■	■	■	■			■	■	■				■	■			■	Distill
Cashman, et al.	2017	■	■			■	■			■	■						■				■	VADL
Chae, et al.	2017	■	■			■					■		■			■	■			■		VADL
Chung, et al.	2016	■	■			■			■	■	■	■		■	■	■	■			■		FILM
Goyal, et al.	2016	■						■		■							■	■	■		■	arXiv
Harley	2015	■			■			■	■	■	■			■			■	■			■	ISVC
Hohman, et al.	2017	■		■	■			■					■				■	■	■		■	CHI
Kahng, et al.	2018	■	■			■	■		■		■	■	■	■	■		■				■	TVCG
Karpathy, et al.	2015	■				■	■				■	■	■		■		■				■	arXiv
Li, et al.	2015	■				■	■				■	■	■		■		■				■	arXiv
Liu, et al.	2017	■	■			■			■	■	■		■	■			■				■	TVCG
Liu, et al.	2018	■	■			■			■	■	■		■	■		■	■			■		TVCG
Ming, et al.	2017	■		■		■					■		■				■				■	VAST
Norton & Qi	2017	■	■		■	■	■	■									■	■			■	VizSec
Olah	2014	■			■			■				■			■		■	■			■	Web
Olah, et al.	2018	■			■	■	■	■	■		■	■	■				■	■	■		■	Distill
Pezzotti, et al.	2017	■	■			■					■	■	■		■	■	■			■		TVCG
Rauber, et al.	2017	■	■	■		■					■	■	■		■		■			■	■	TVCG
Robinson, et al.	2017	■				■	■				■	■	■				■				■	GeoHum.
Rong, et al.	2016	■	■			■	■				■		■				■				■	ICML VIS
Smilkov, et al.	2016	■					■				■	■	■		■		■				■	NIPS Workshop
Smilkov, et al.	2017	■	■		■			■	■	■	■			■		■		■		■	■	ICML VIS
Strobelt, et al.	2017	■	■			■	■				■	■	■		■		■				■	TVCG
Tzeng & Ma	2005	■				■			■	■			■	■		■					■	VIS
Wang, et al.	2018	■	■	■		■				■	■		■		■	■	■			■		TVCG
Webster, et al.	2017				■			■									■	■		■	■	Web
Wongsuphasawat, et al.	2018		■			■			■				■	■								TVCG
Yosinski, et al.	2015	■			■		■	■	■	■	■						■	■	■		■	ICML DL
Zahavy, et al.	2016	■	■			■					■	■	■		■		■				■	ICML
Zeiler, et al.	2014	■	■			■				■	■								■		■	ECCV
Zeng, et al.	2017	■		■		■			■		■						■			■		VADL
Zhong, et al.	2017	■	■			■					■	■	■		■	■	■		■	■		ICML VIS
Zhu, et al.	2016	■				■	■	■					■				■	■	■		■	ECCV

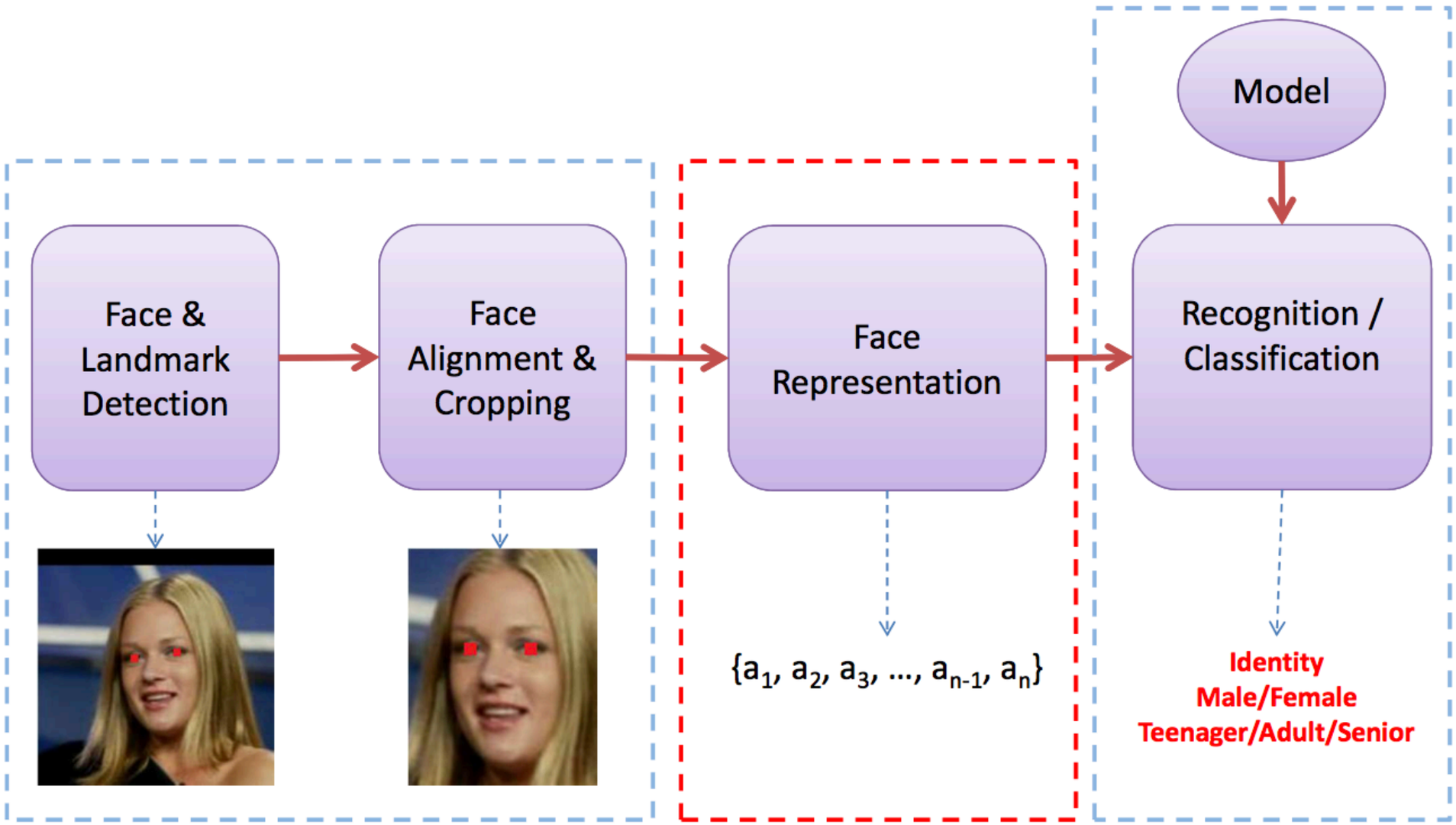
https://fredhohman.com/visual-analytics-in-deep-learning/

FOOLING CNN/ ADVERSARIAL NETWORKS

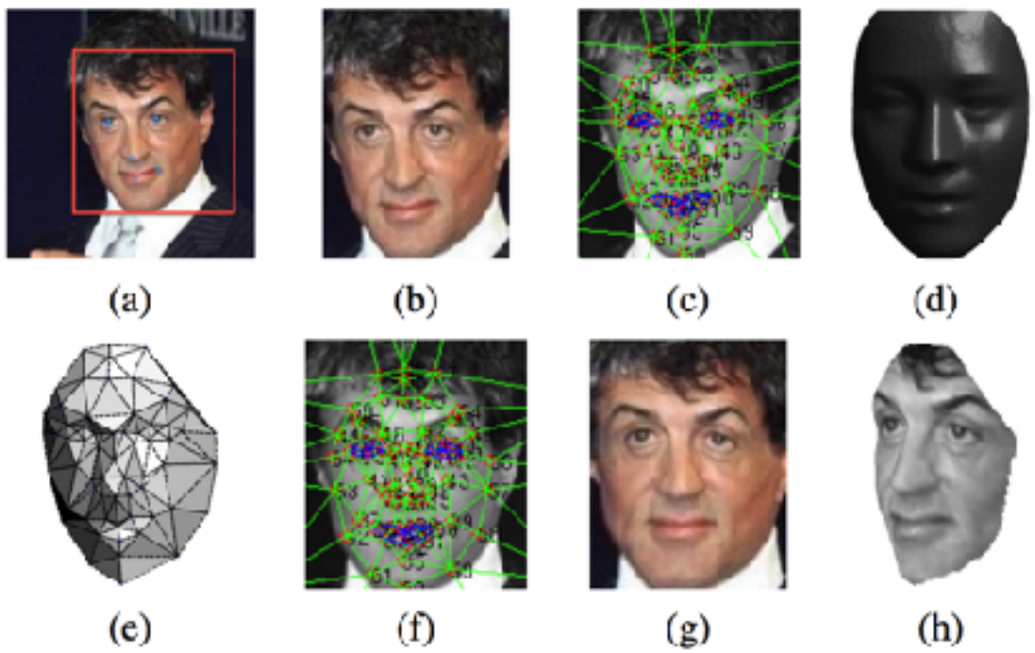


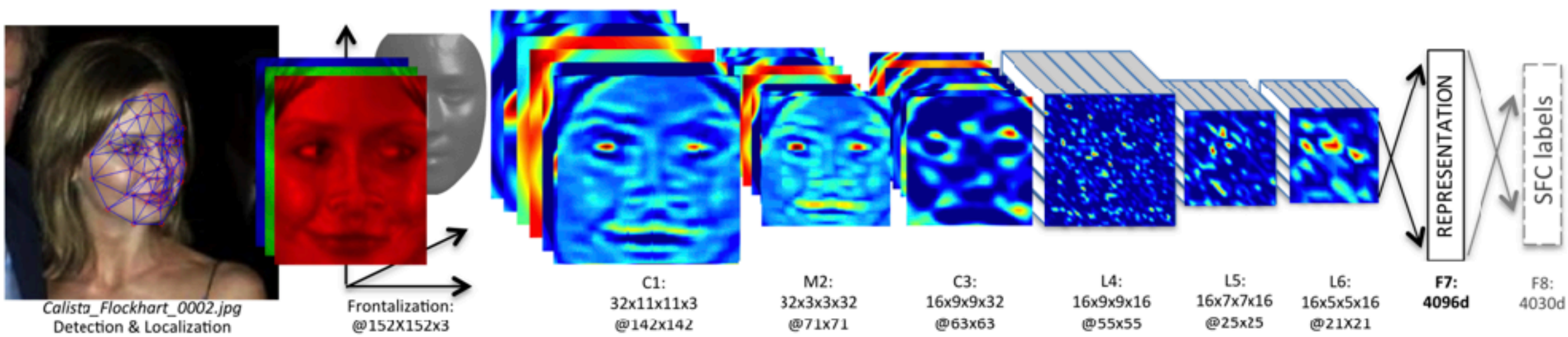
FACE RECOGNITION

FACE RECOGNITION: PIPELINE

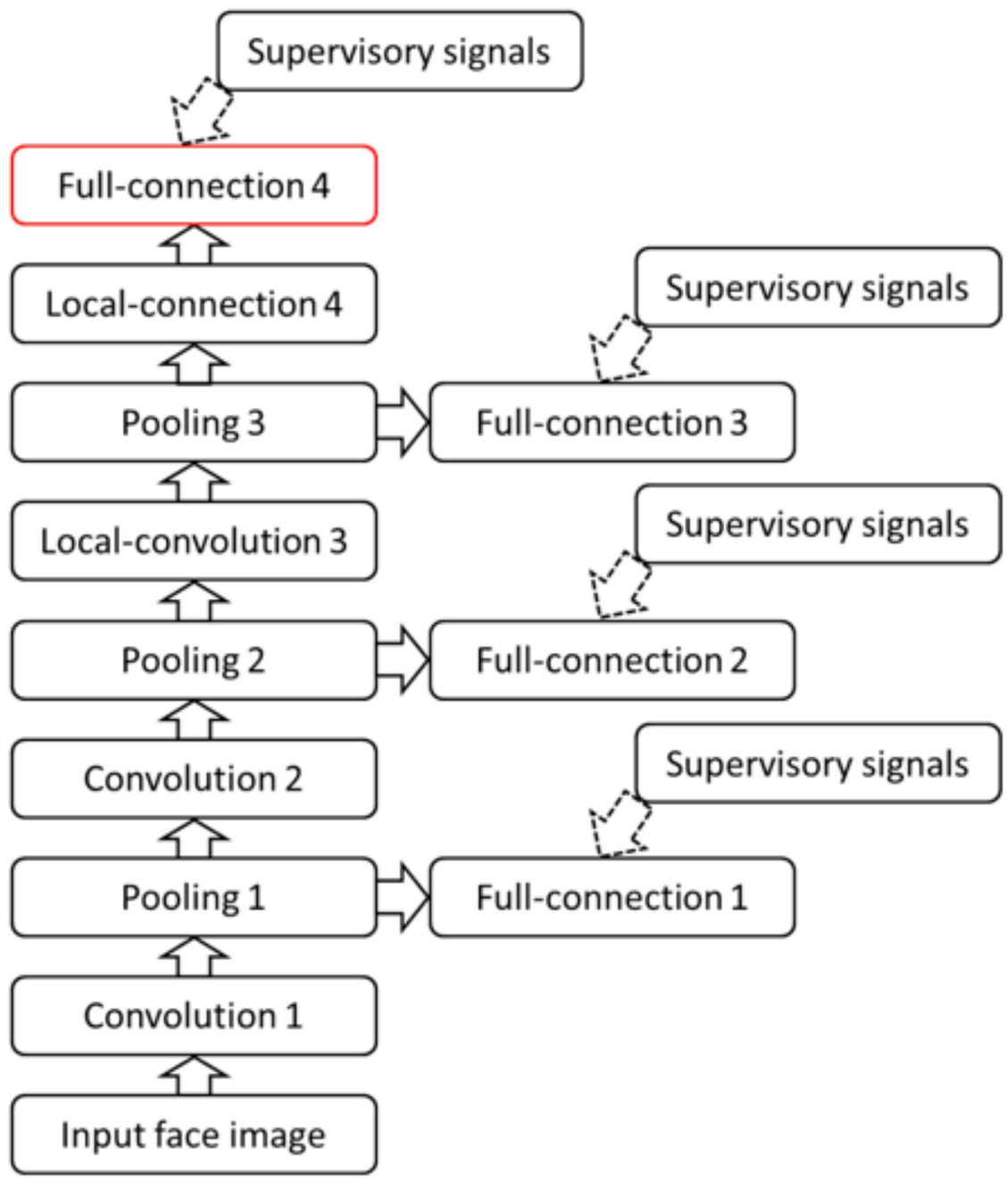


DEEPFACE: HUMAN LEVEL FACE VERIFICATION

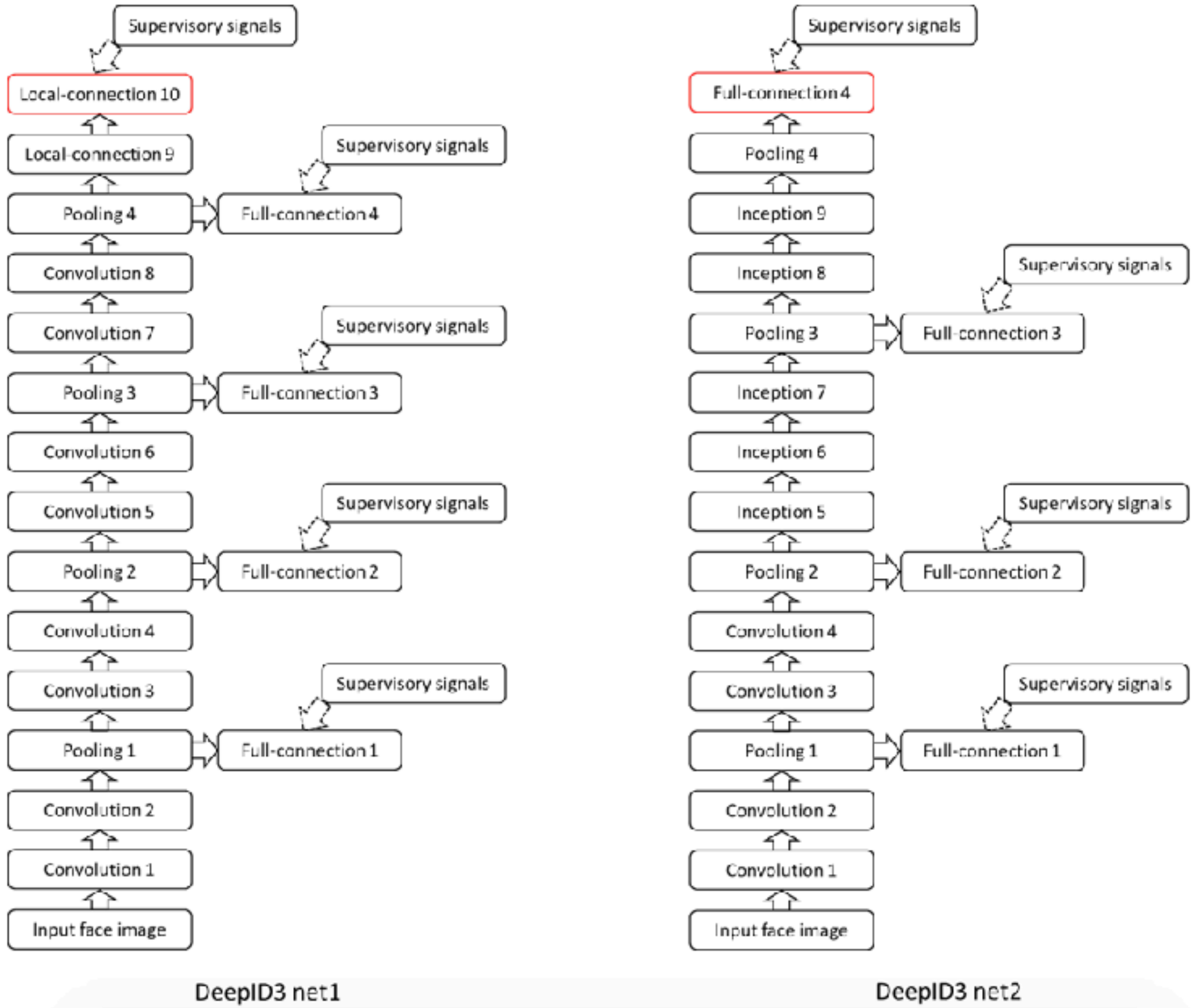




DEEPID2


Supervisory signals
Full-connection 4
Local-connection 4
Supervisory signals
Pooling 3
Full-connection 3
Local-convolution 3
Supervisory signals
Pooling 2
Full-connection 2
Convolution 2
Supervisory signals
Pooling 1
Full-connection 1
Convolution 1
Input face image

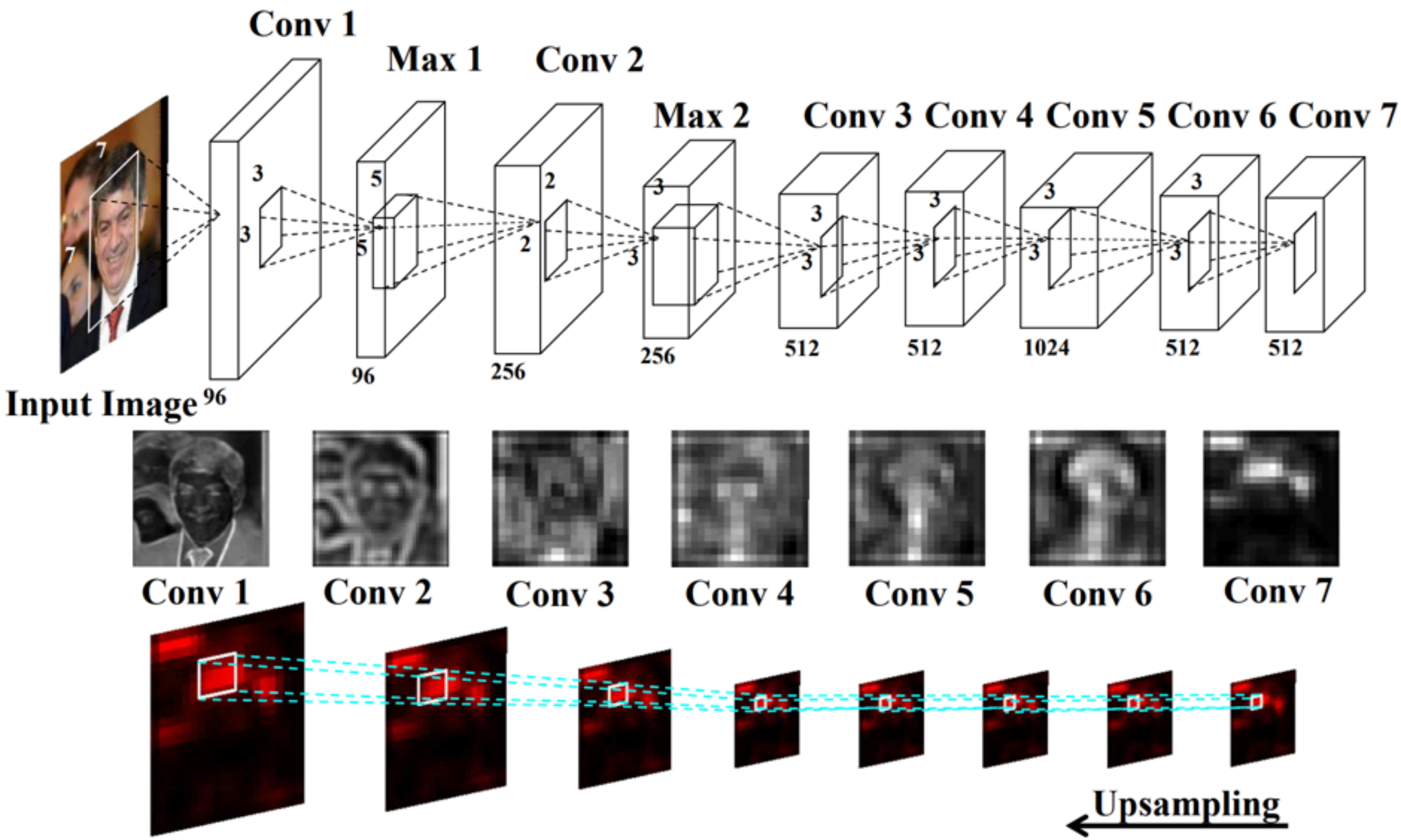
DEEPID3



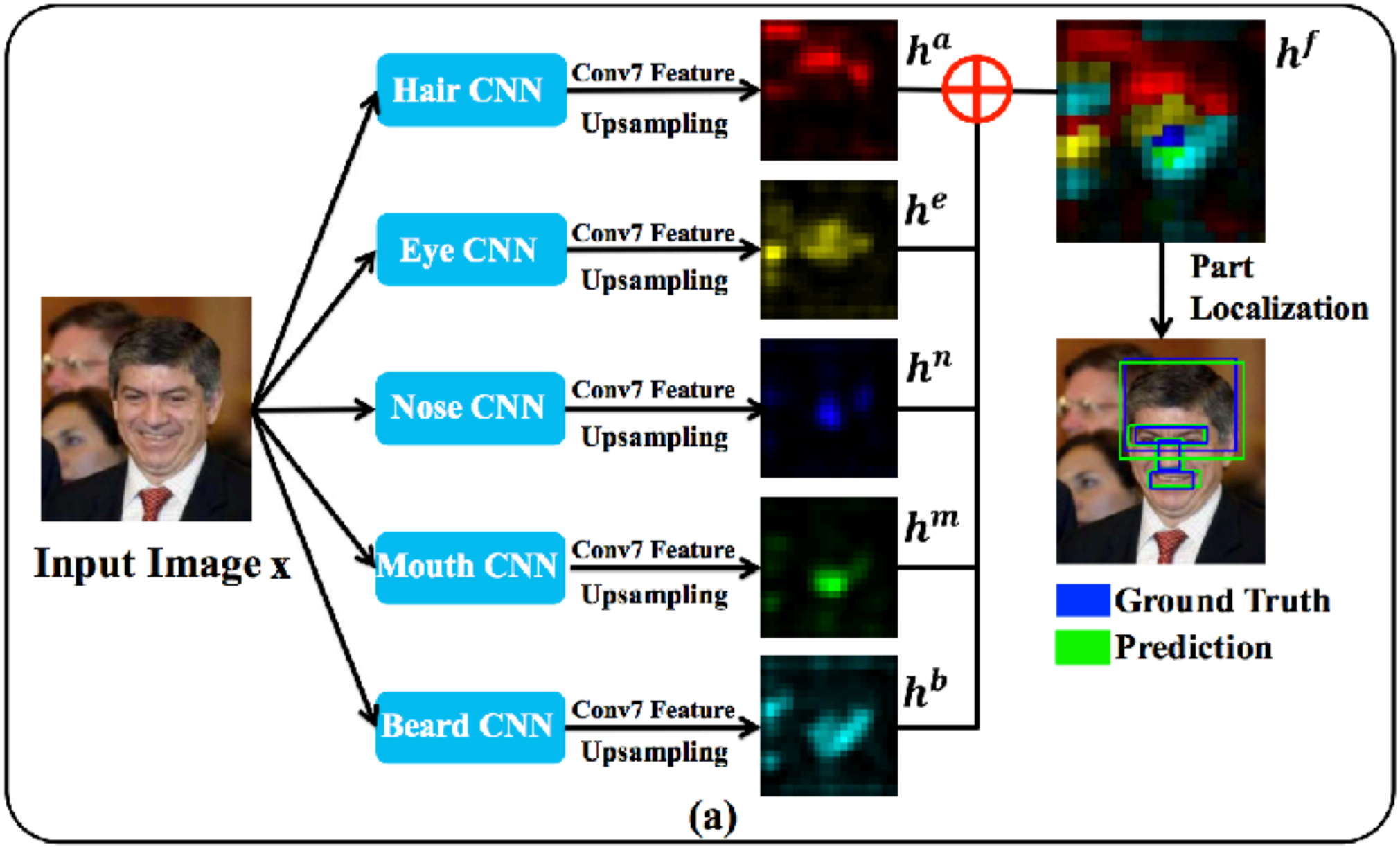
DeepID3 net1 DeepID3 net2

FACE DETECTION

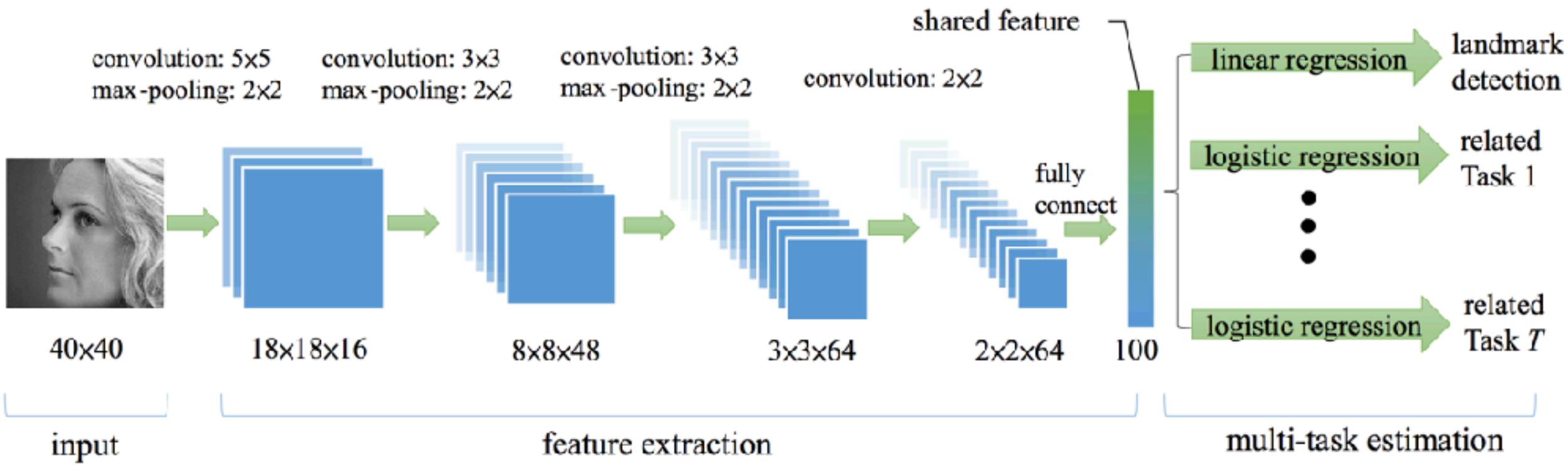
FACE DETECTION



FACE PART DETECTION



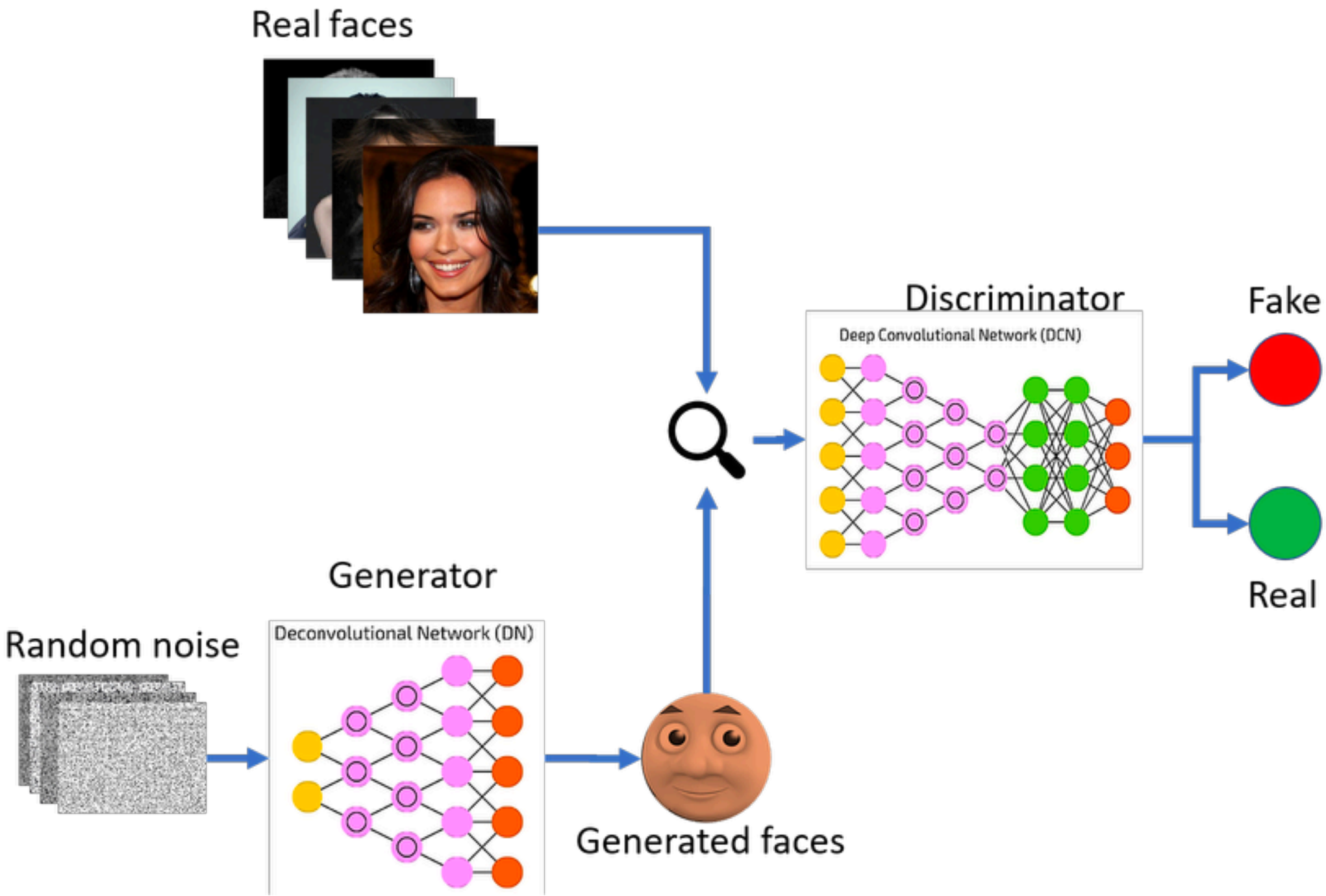
TASK CONSTRAINED DCN: MORE THAN ONE TASK



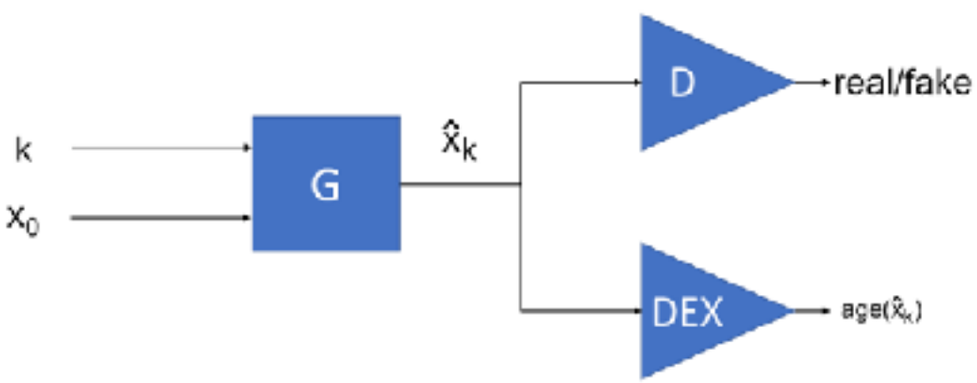
FACE GENERATION

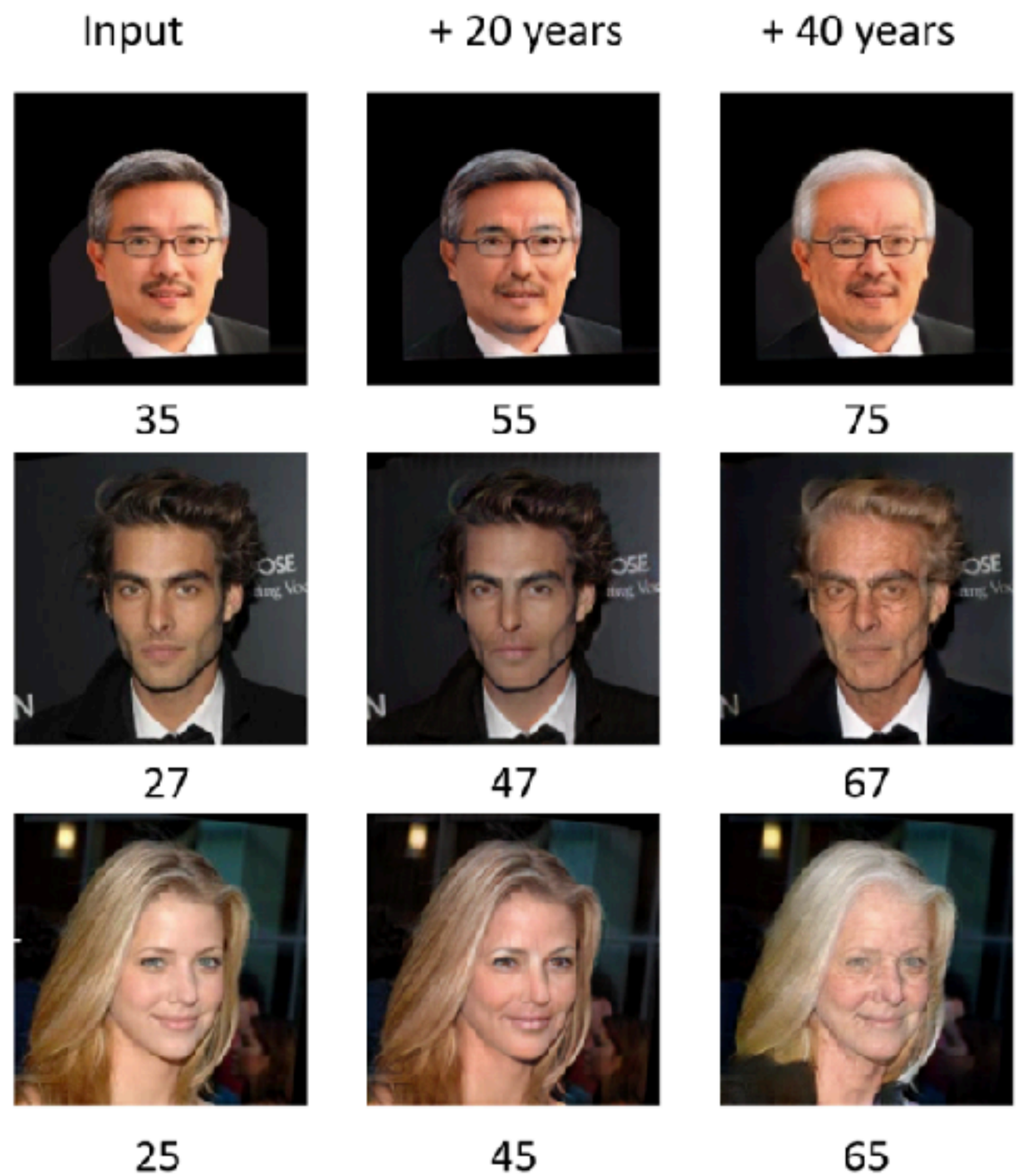
https://thispersondoesnotexist.com/

FACE GENERATION

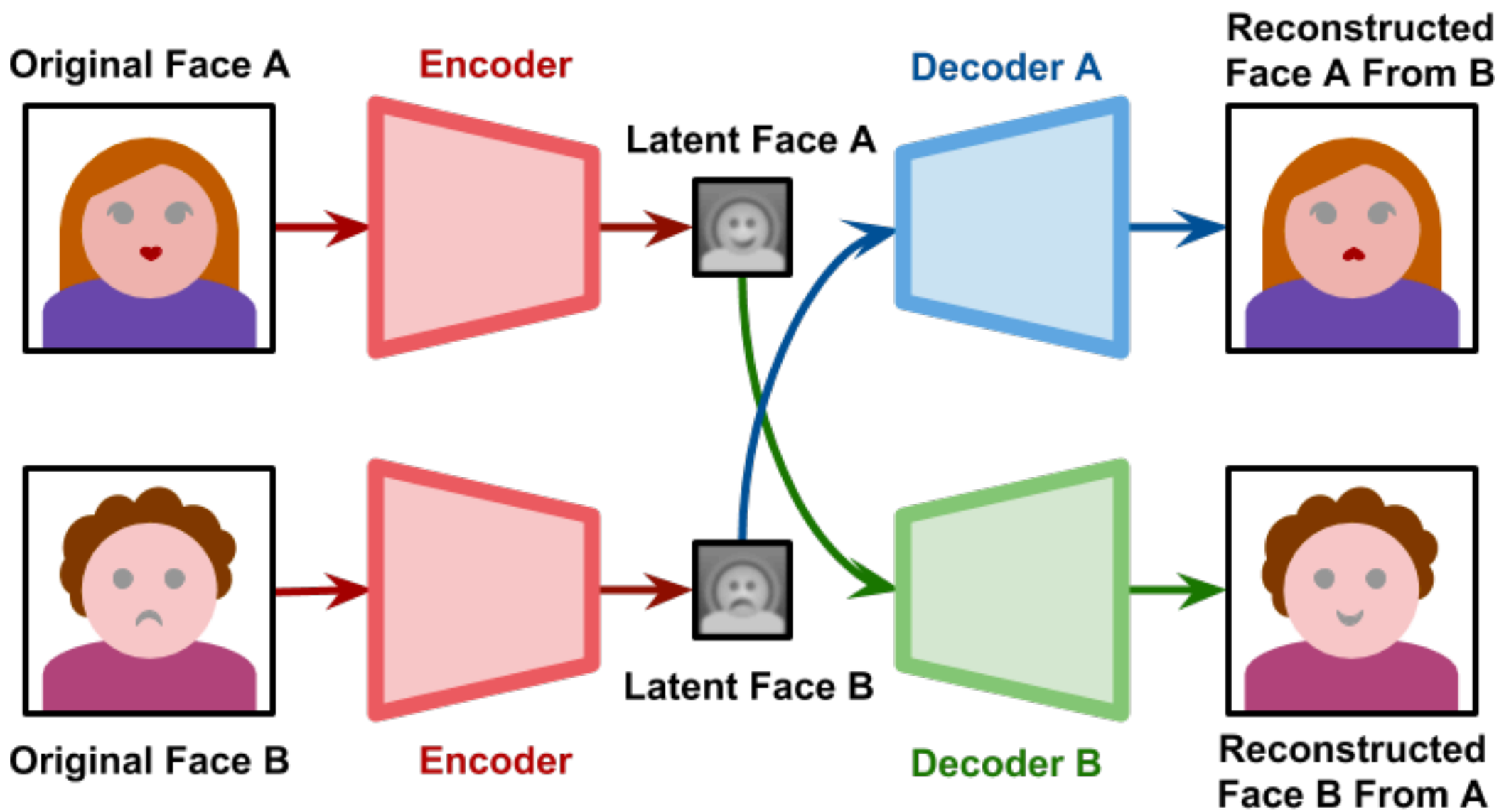


FACE STYLE TRANSFER

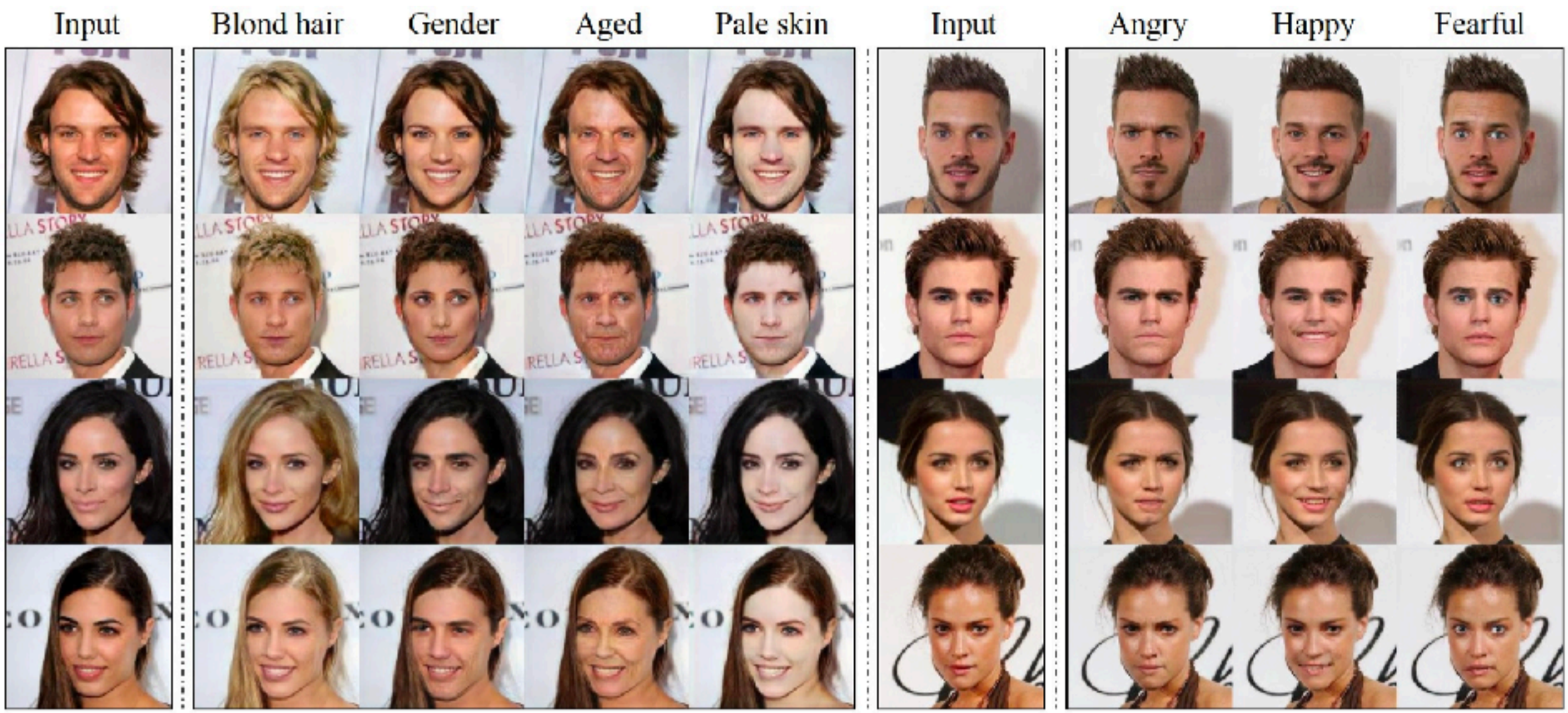




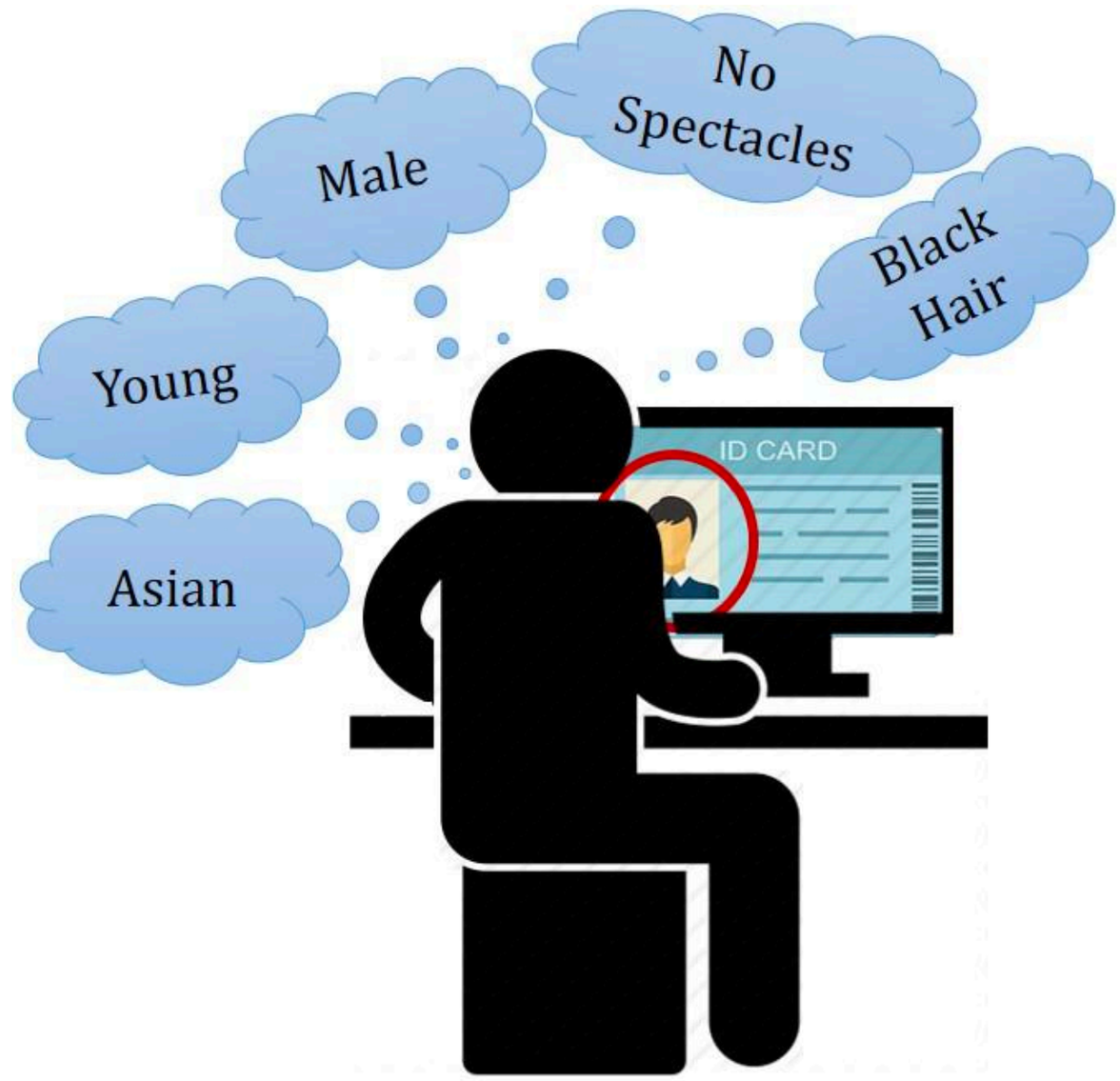
DEEPFAKE


Original Face A
Encoder
Latent Face A
Decoder A
Reconstructed Face A From B
Original Face B
Encoder
Latent Face B
Decoder B
Reconstructed Face B From A

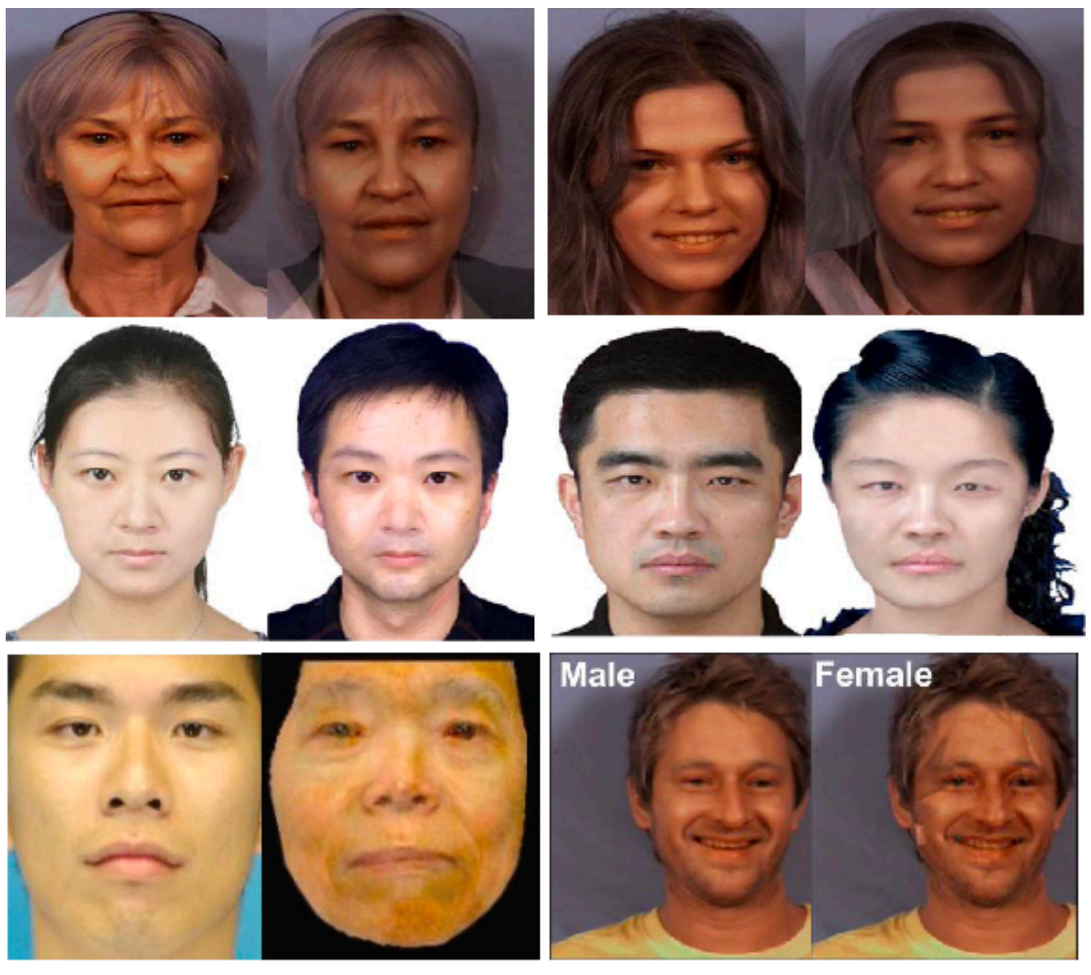
DEEPFAKE


Input
Blond hair
Gender
Aged
Pale skin
Input
Angry
Happy
Fearful

PRIVACY PROTECTION

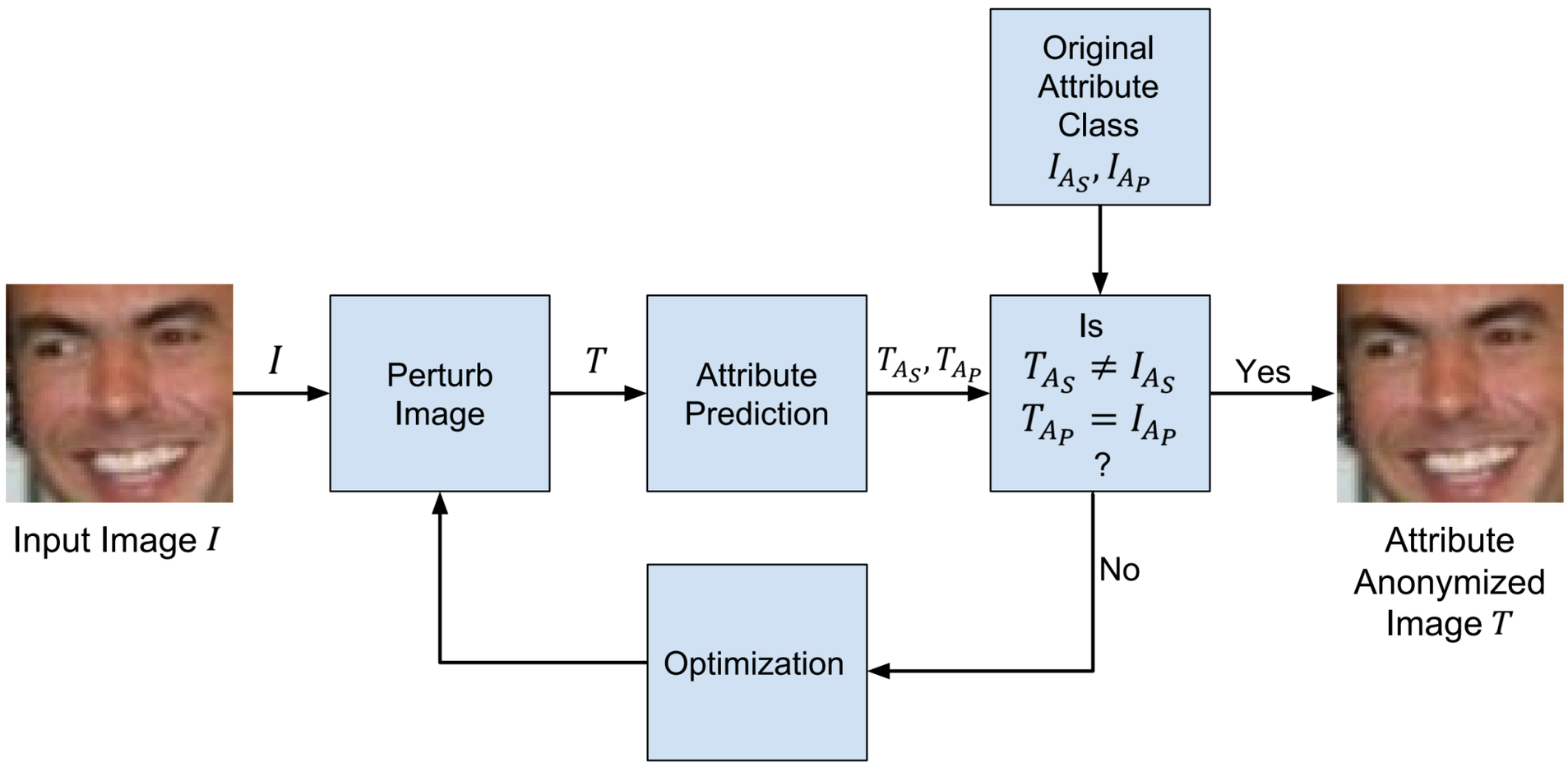


FACE PRIVACY PROTECTION



Attributes are preserved while identity is lost …

ADVERSARIAL LEARNING



PRESERVE MORE THAN ONE AT

ASSIGNMENT

- Consider the IIIT Bangalore Face dataset (the one that you used for your previous assignment!)
- Part1: Use a Deep CNN for doing face verification (Binary Classification). Two input face images will be given to the model, and the model will classify them as “belonging to the same person” or “belonging to different person”. Use any CNN model and perform any face preprocessing as deemed necessary
- Part2: Use the images in the IIIT Bangalore Face dataset to generate a new image that does not exist in that dataset (or any other dataset in the whole wide world)
  - Bonus marks: Can you perform style transfer on images. Given a face image, change the hair style of the same face image.

THAT'S ALL FOLKS!