AI for Good Workshop

March 2019

Session 6

Artificial Intelligence for everyone

Practical Deep Learning for Professional Context (Computer Vision)

https://sites.google.com/view/AIforEveryone

+

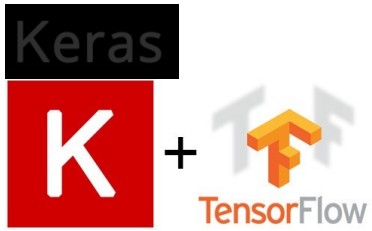
Topics in this video

Jump start your practical solution development!

1. Deep Learning based Image Classification (Computer Vision)

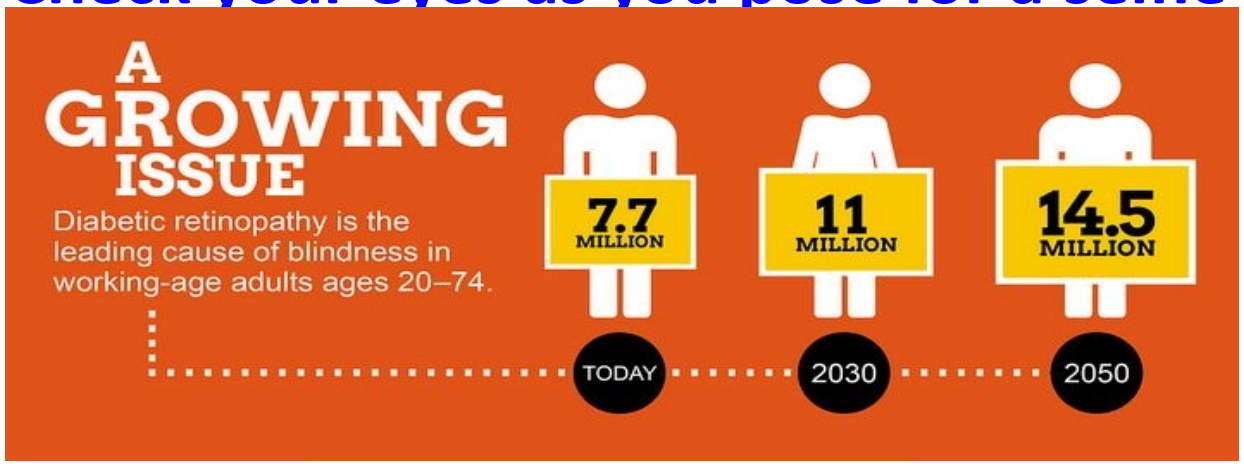
2. How to Create a professional solution ?

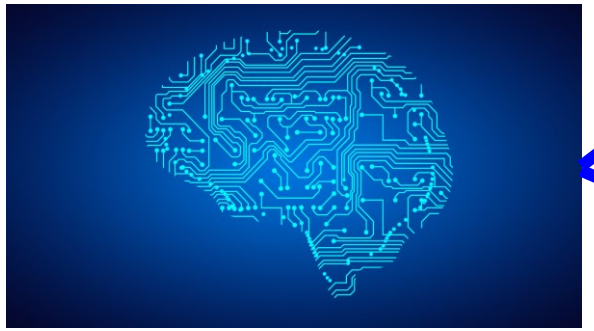
3. Hands-on : Create your 1st Deep Learning based Image classification solution

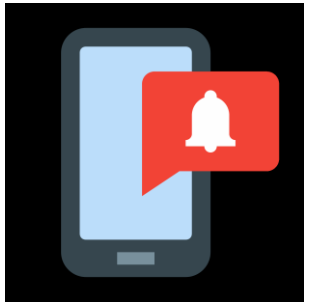


App idea: Selfie that checks your eye health

Idea: Check your eyes as you pose for a selfie !



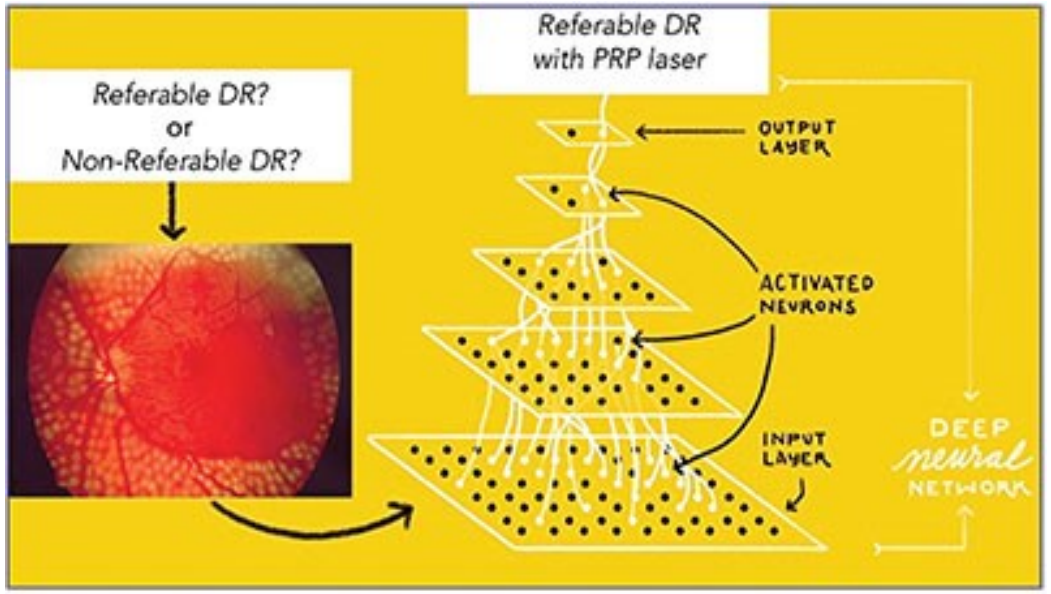




Your Eye is healthy

Better, Go for a eye check-up

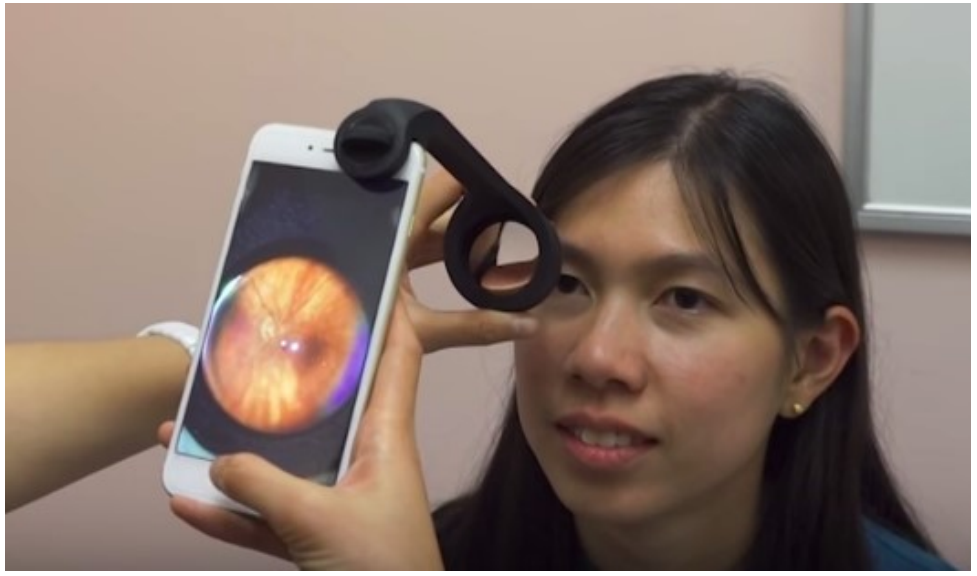
App idea: App that Screens your eyes as you pose for a selfie

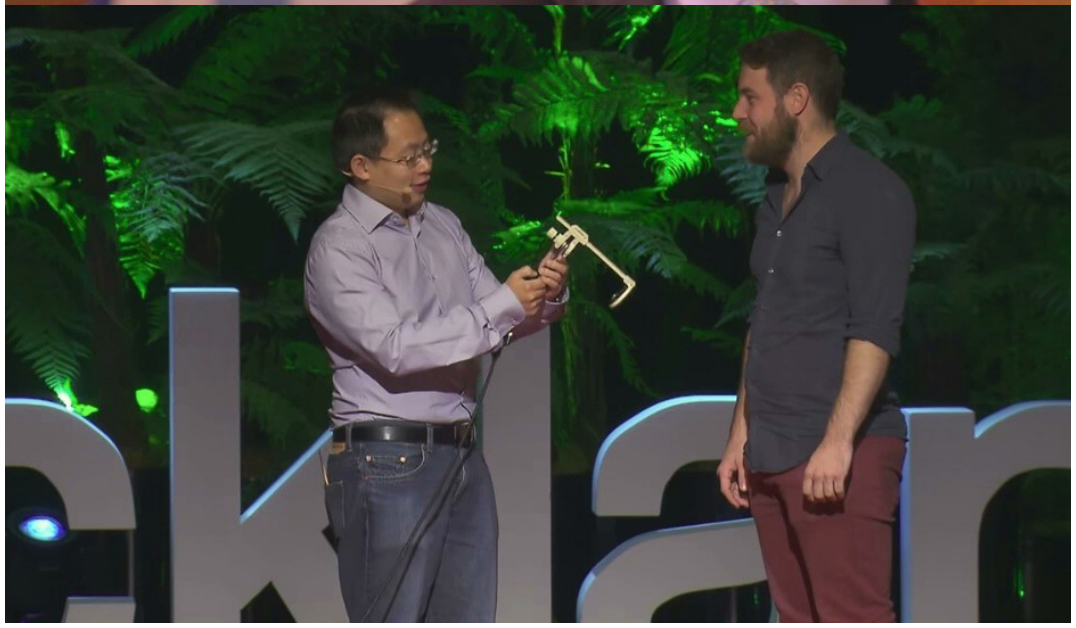


Screening Protocol

- Patients who need urgent referral
- Patients who need routine referral
- Patients who need regular screening and follow up annual

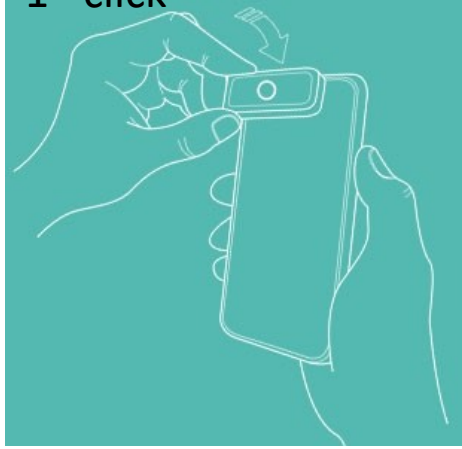
Fighting blindness with your Deep Learning smartphone app



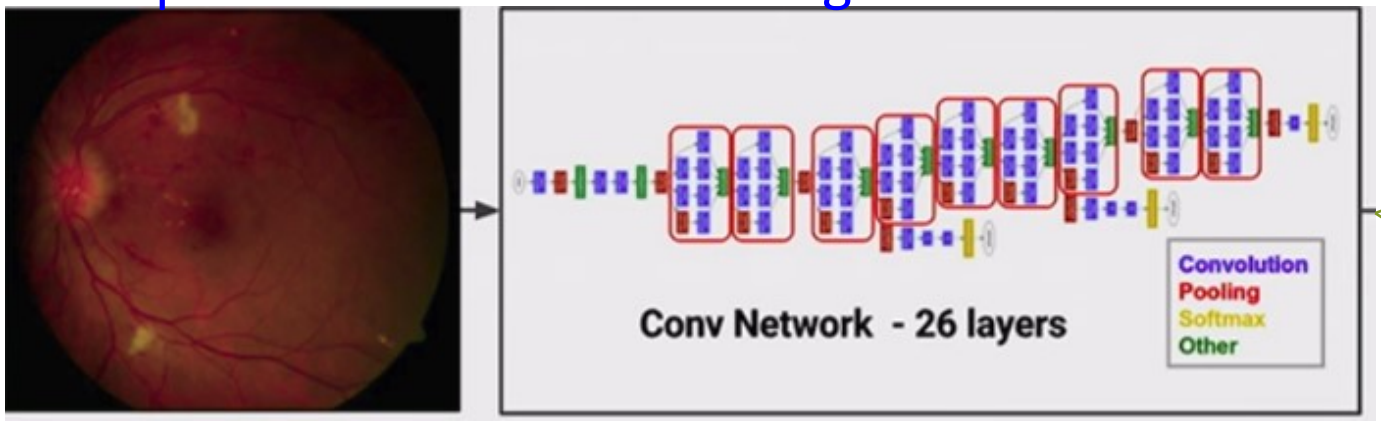


Fighting blindness with your Deep Learning smartphone app

1st click

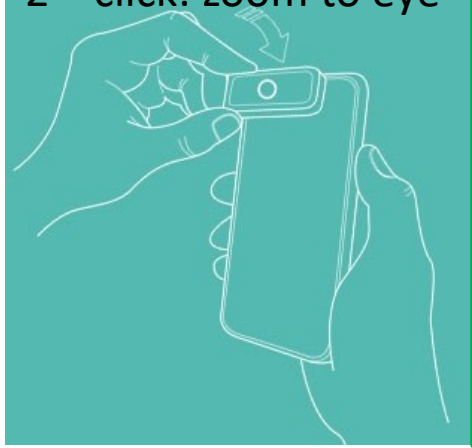


Sequential network consisting of CNN & Dense

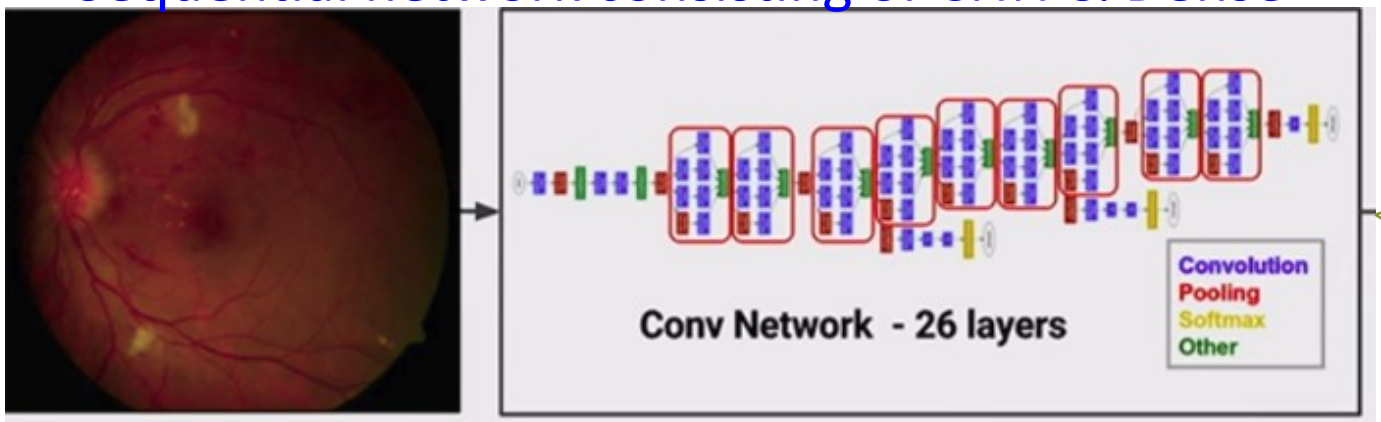


Ensemble of classifiers

2nd click: zoom to eye



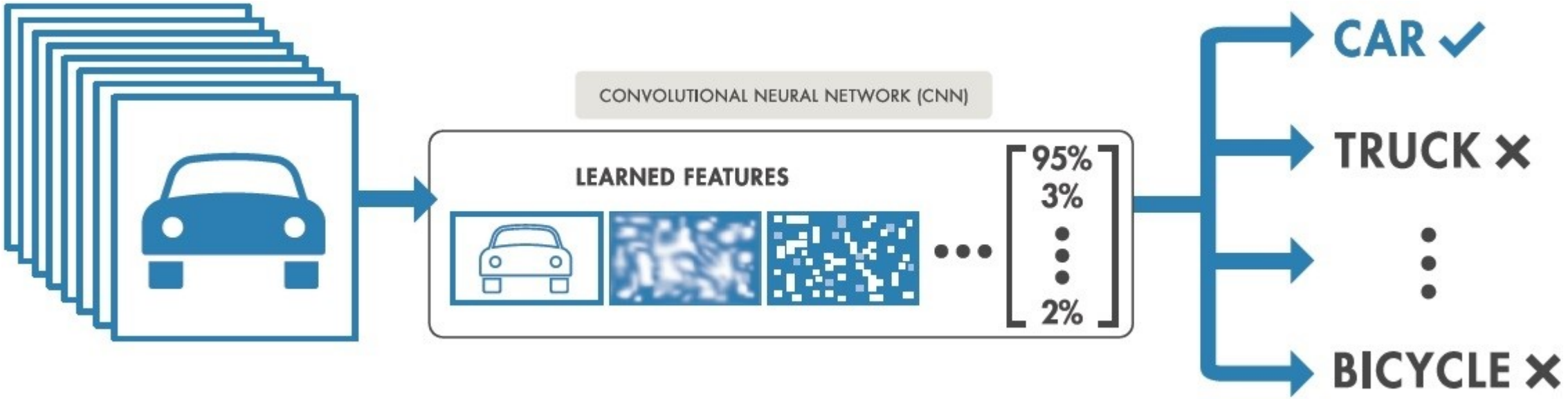
Sequential network consisting of CNN & Dense



Merge

What type of problem?

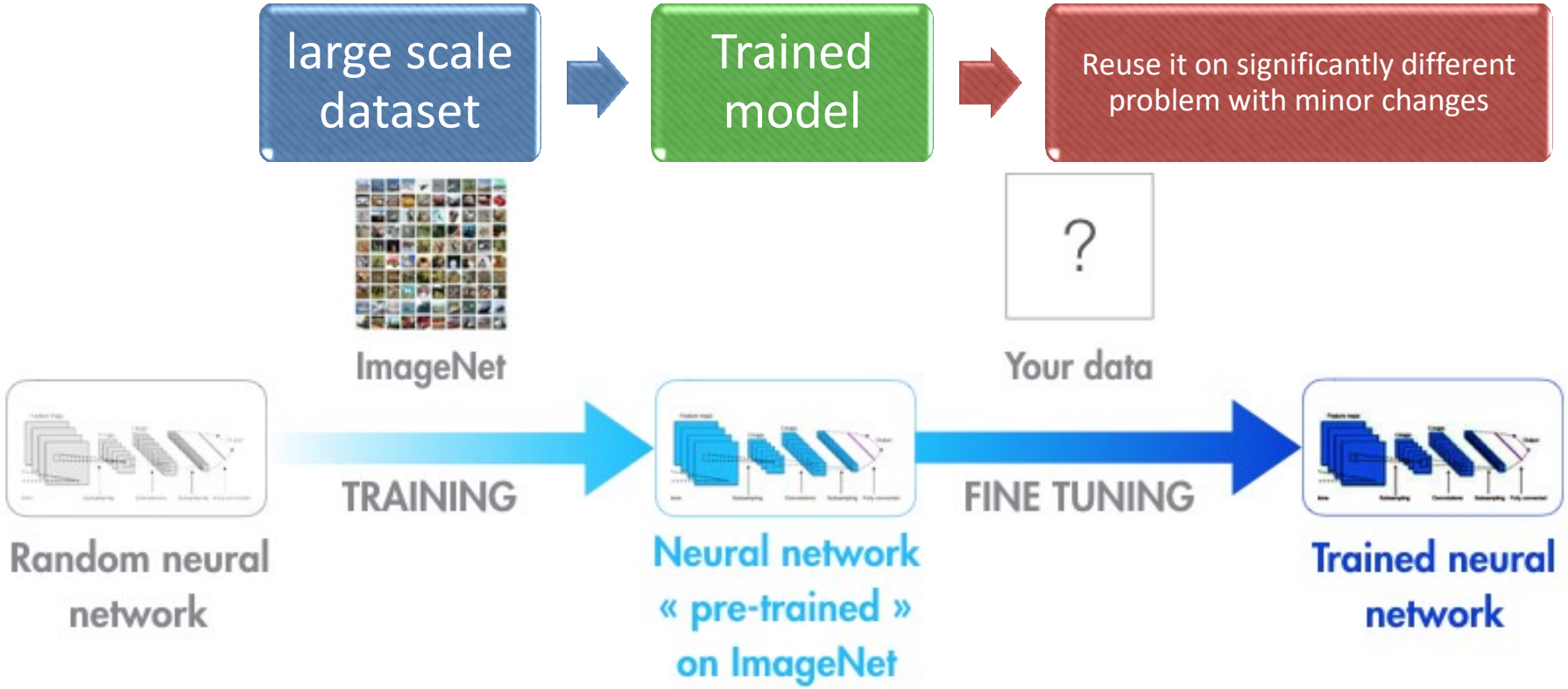
Practical problems: Image classification for small datasets



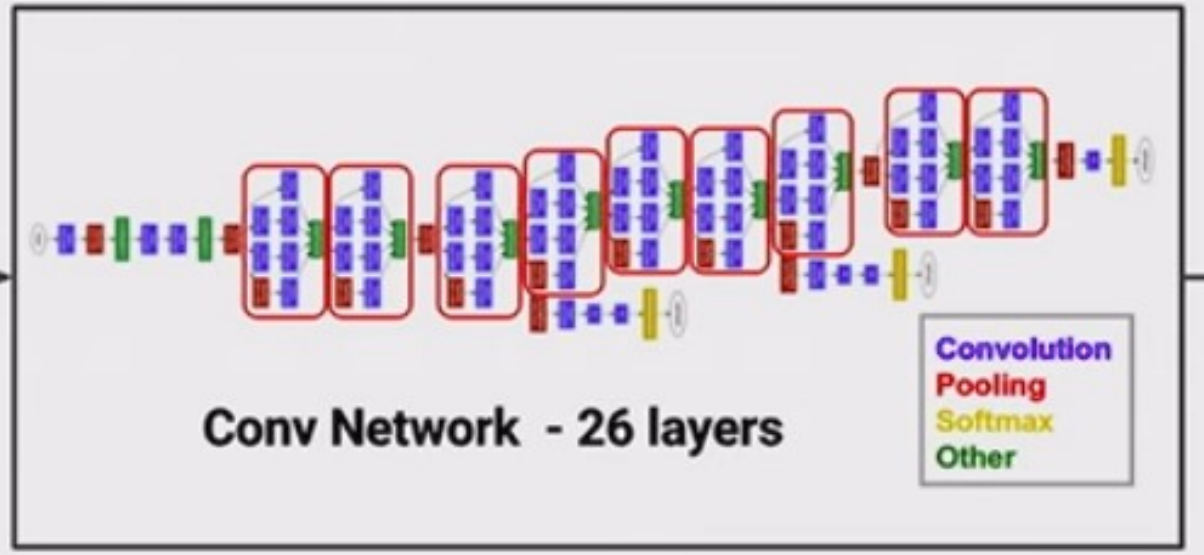
Have only very little data for training?

Deep learning models are re-purposable!

Having to train a image classification model using only very little data is a common situation.



App Goal: Selfie Binary classifier to alert you for eye check-up



No DR

Mild DR

Moderate DR

Severe DR

Proliferative DR

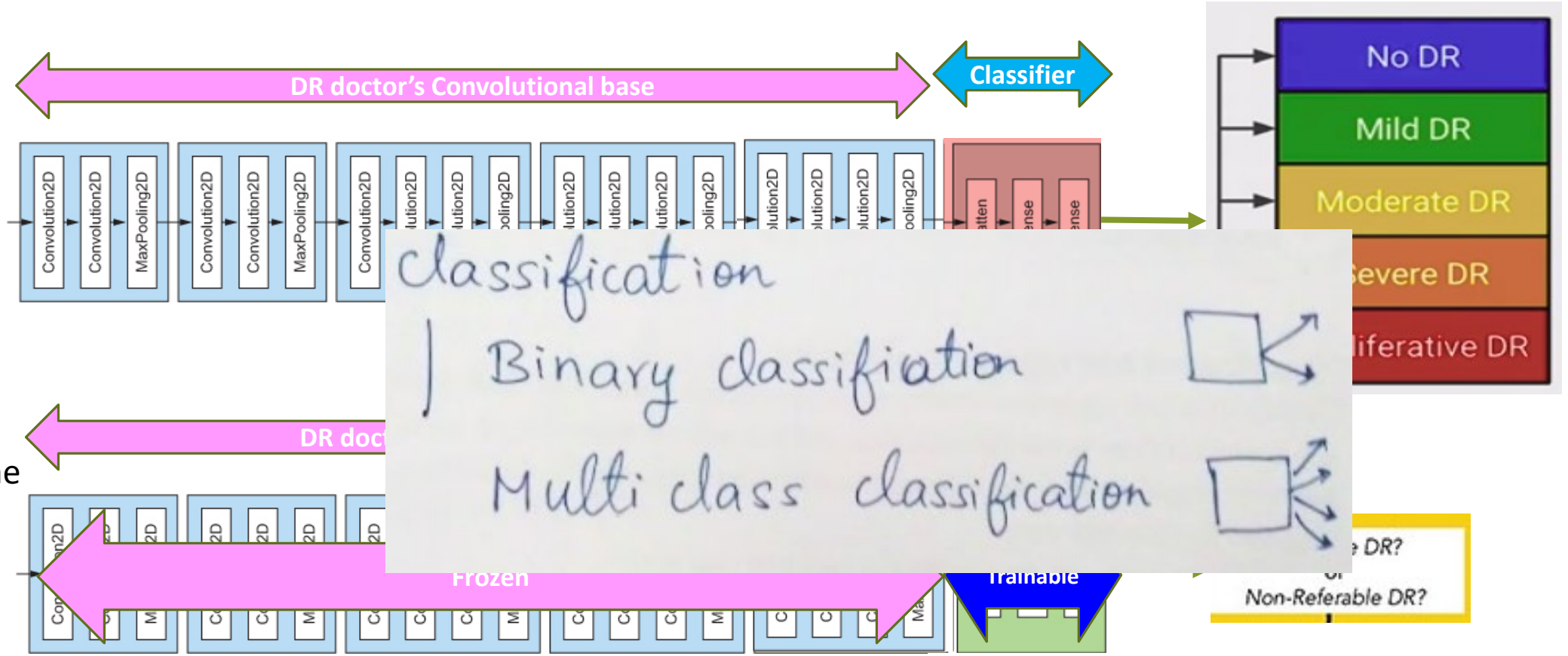
Image Quality

L/R eye

Field of View

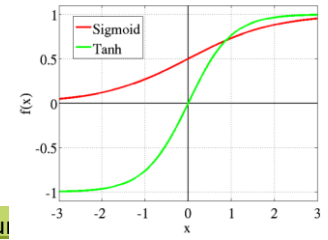
Multi class classifier → Binary classifier

Purpose:
Eye camp



Purpose:
App / Home

classification
Binary classification
Multi class classification



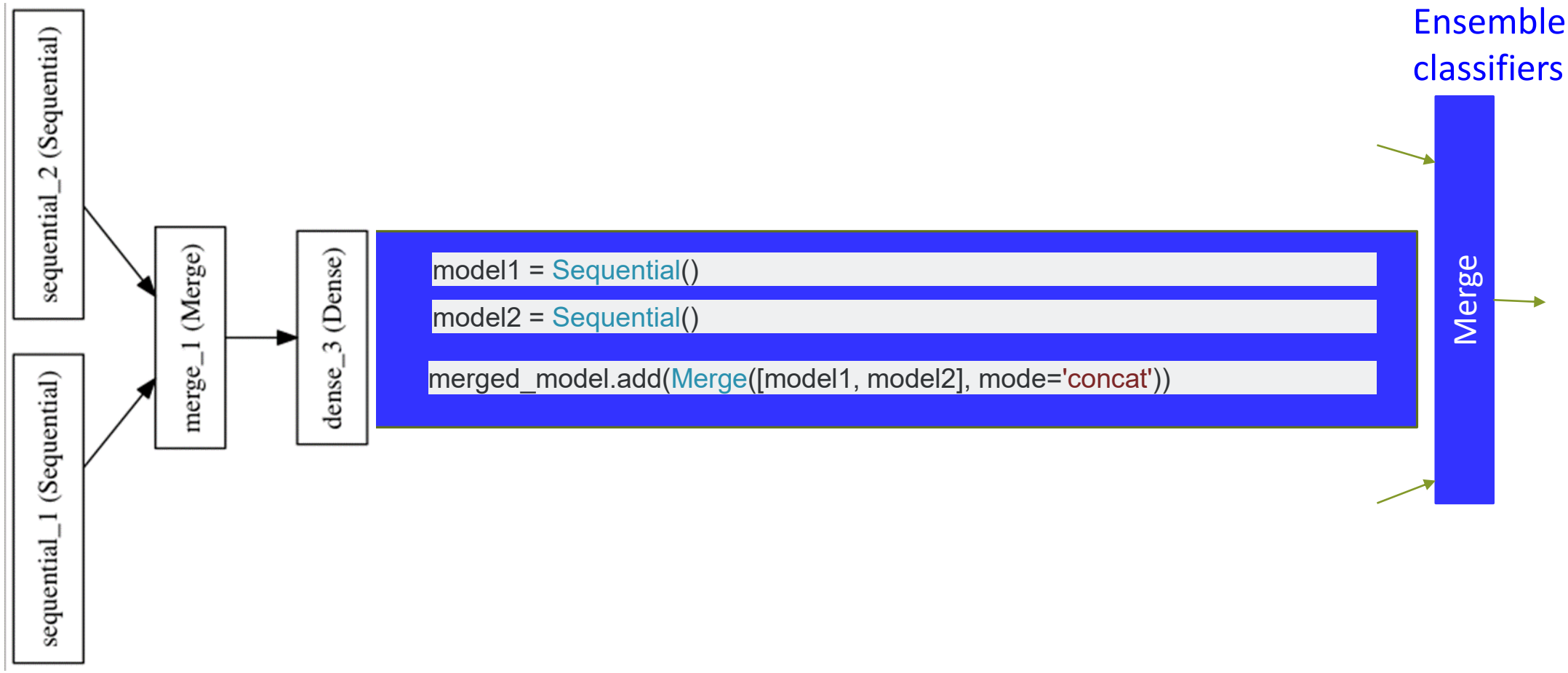
Choose wisely objective function to minimize false Non-Refereable DR

Cheat sheet : objective function

Choosing the right last-layer activation and loss function for your model

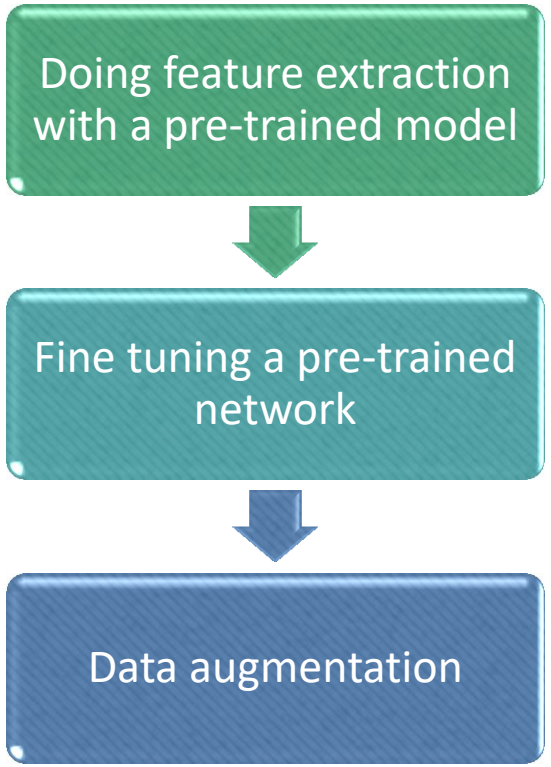
Problem type	Last-layer activation	Loss function
Binary classification	sigmoid	binary_crossentropy
Multiclass, single-label classification	softmax	categorical_crossentropy
Multiclass, multilabel classification	sigmoid	binary_crossentropy
Regression to arbitrary values	None	mse
Regression to values between 0 and 1	sigmoid	mse or binary_crossentropy

Merge 2 networks



How to develop professional solution

Toolbox to develop professional solution



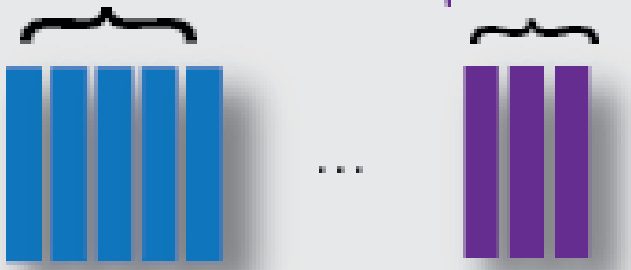
1. Doing feature extraction with a pre-trained model
2. Fine tuning a pre-trained network
3. Data augmentation

How to develop professional solution

Load pretrained network

Early layers that learned low-level features (edges, blobs, colors)

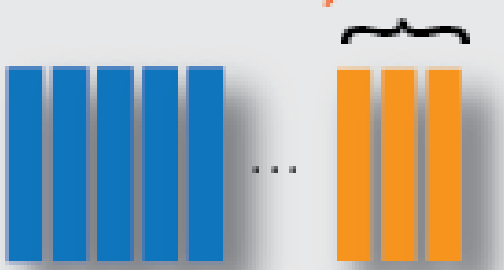
Last layers that learned task specific features



1 million images
1000s classes

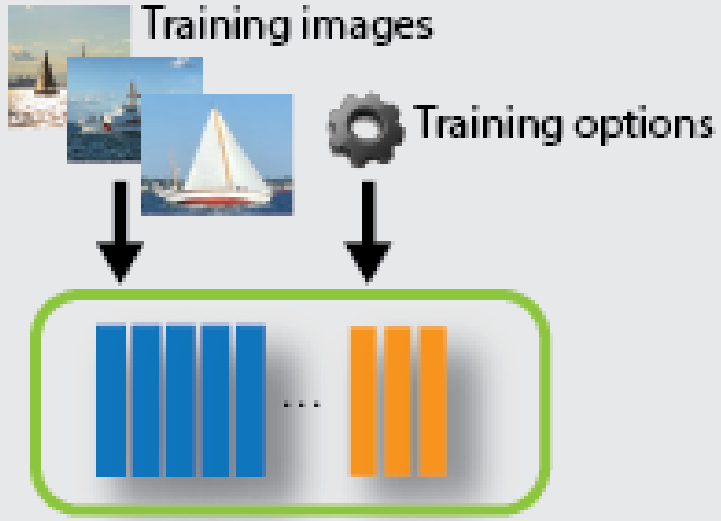
Replace final layers

New layers to learn features specific to your data set



Fewer classes
Learn faster

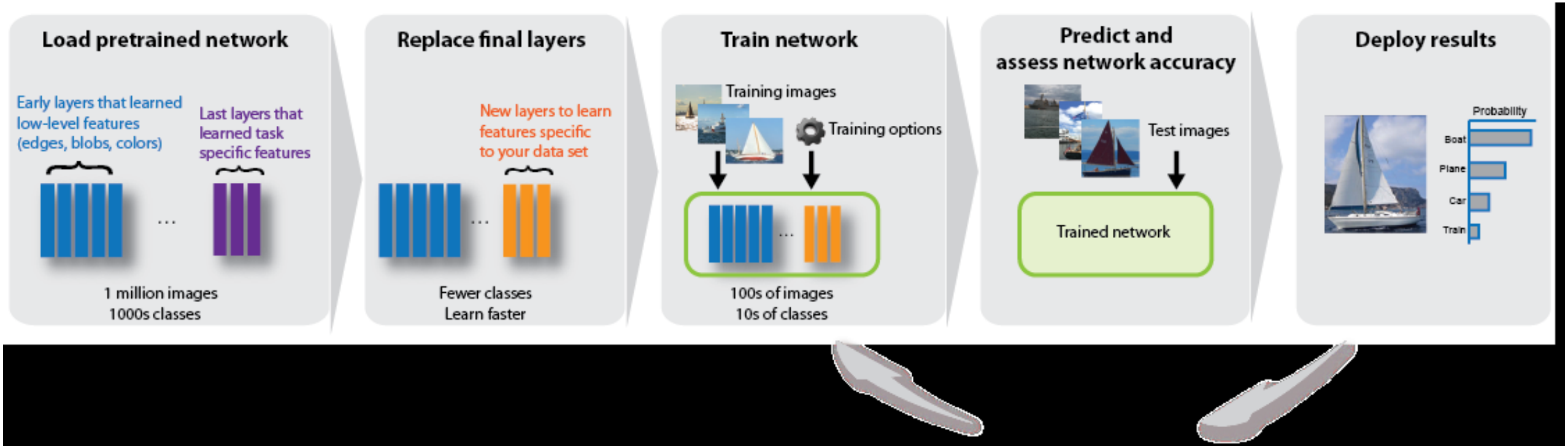
Train network



100s of images
10s of classes

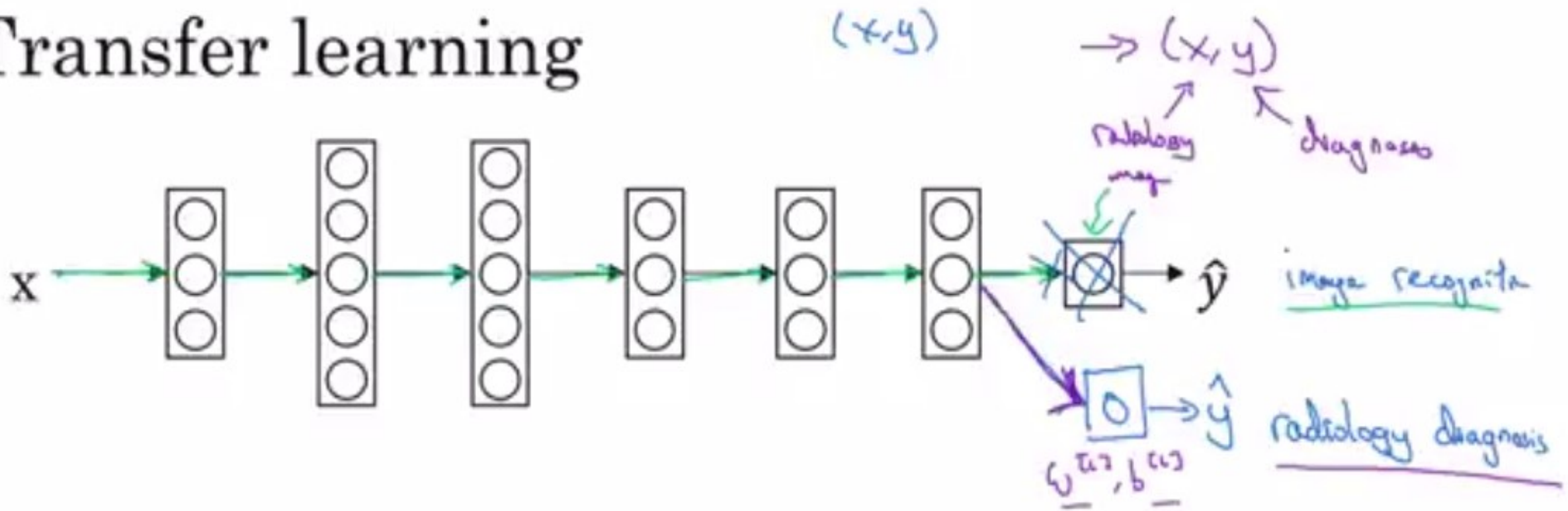
Acknowledgments & Credits are mentioned to inspirational resources presented in last slides. For more like this, https://sites.google.com/view/AIforEveryone

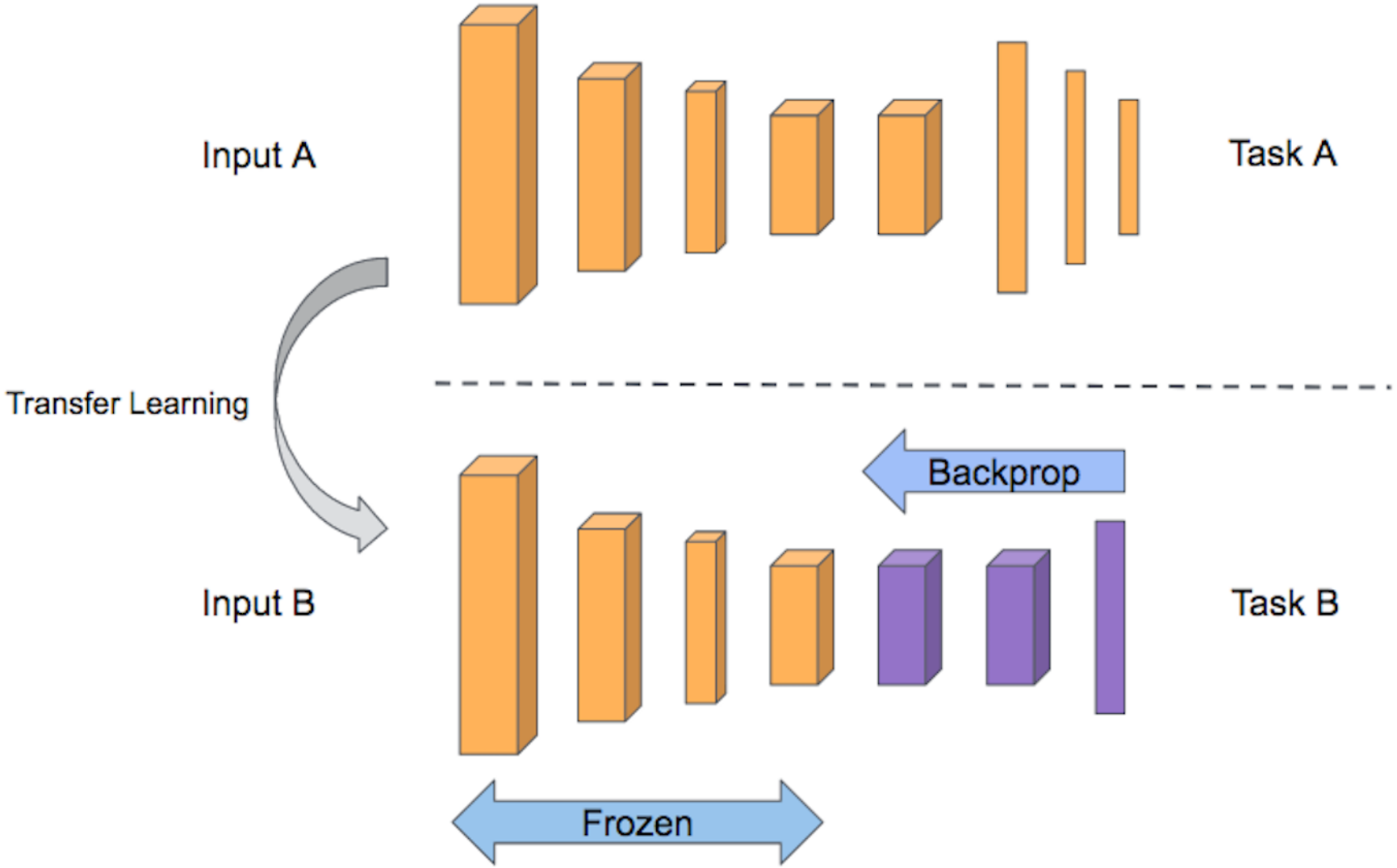
How to develop professional solution



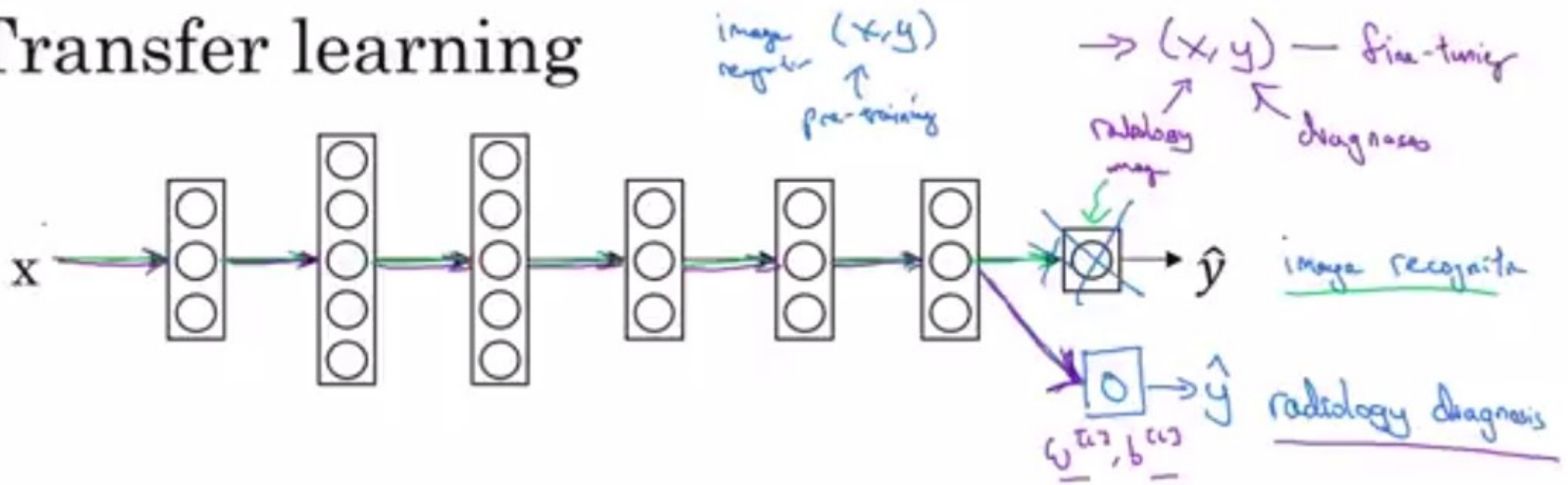
Acknowledgments & Credits are mentioned to inspirational resources presented in last slides. For more like this, https://sites.google.com/view/AIforEveryone

Transfer learning

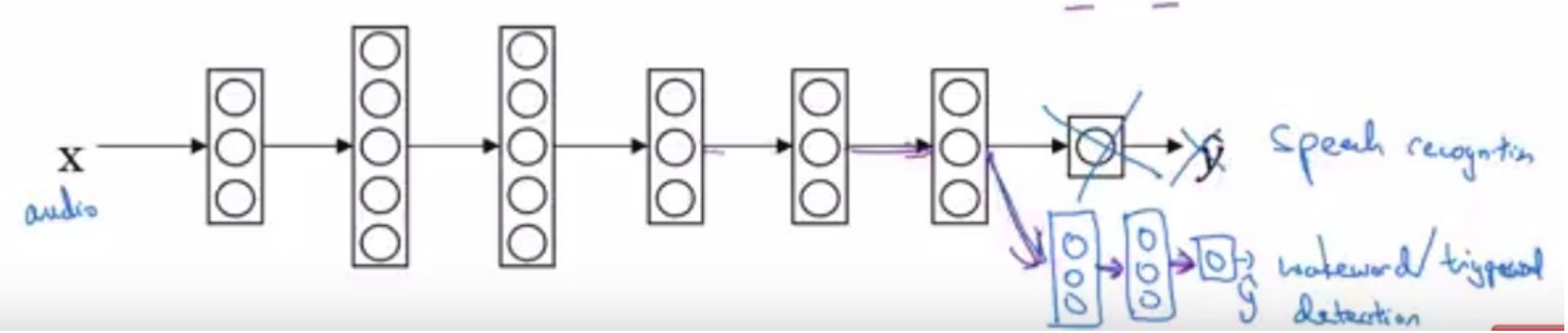



Input A
Task A
Transfer Learning
Backprop
Input B
Task B
Frozen

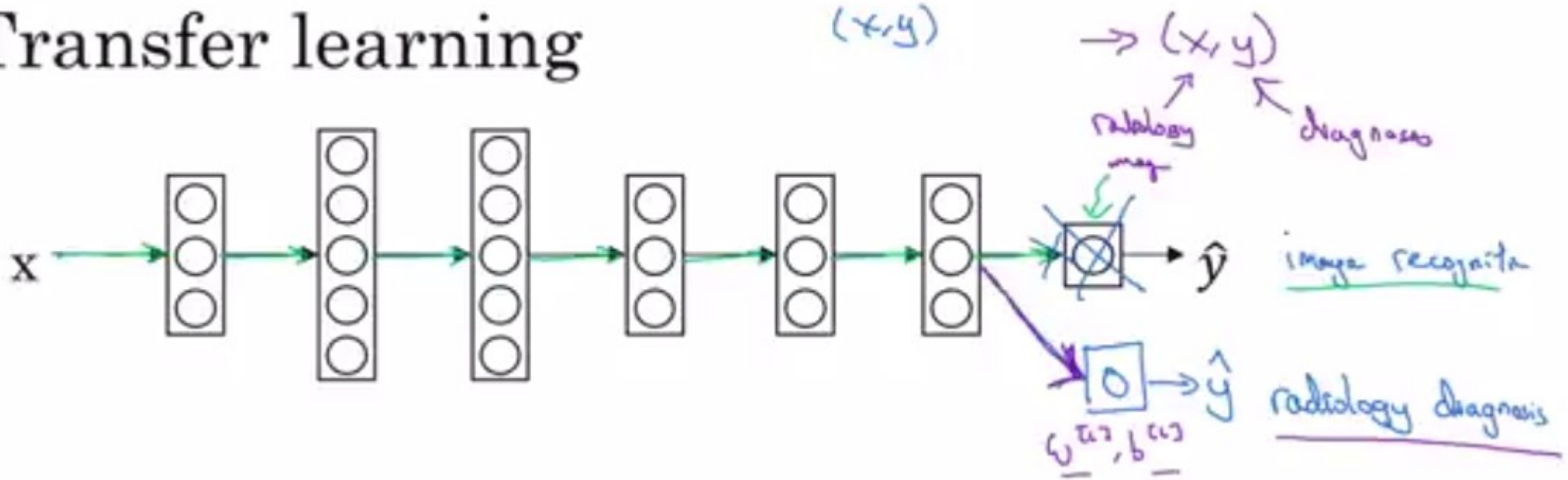
Transfer learning

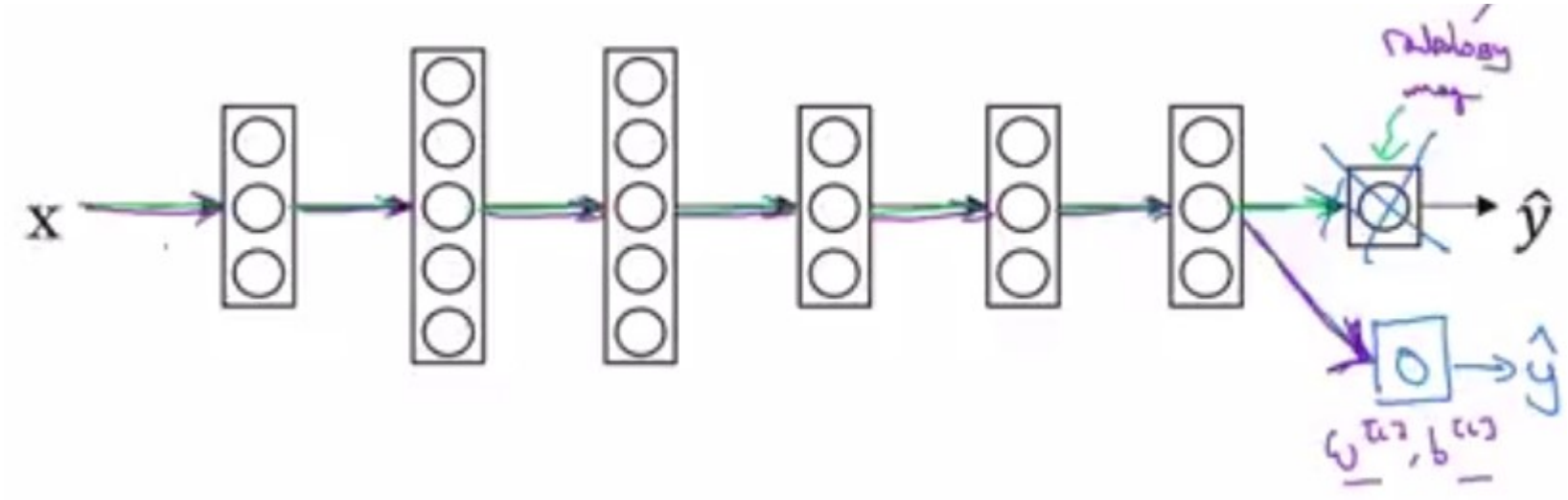


“Hey Karunya”

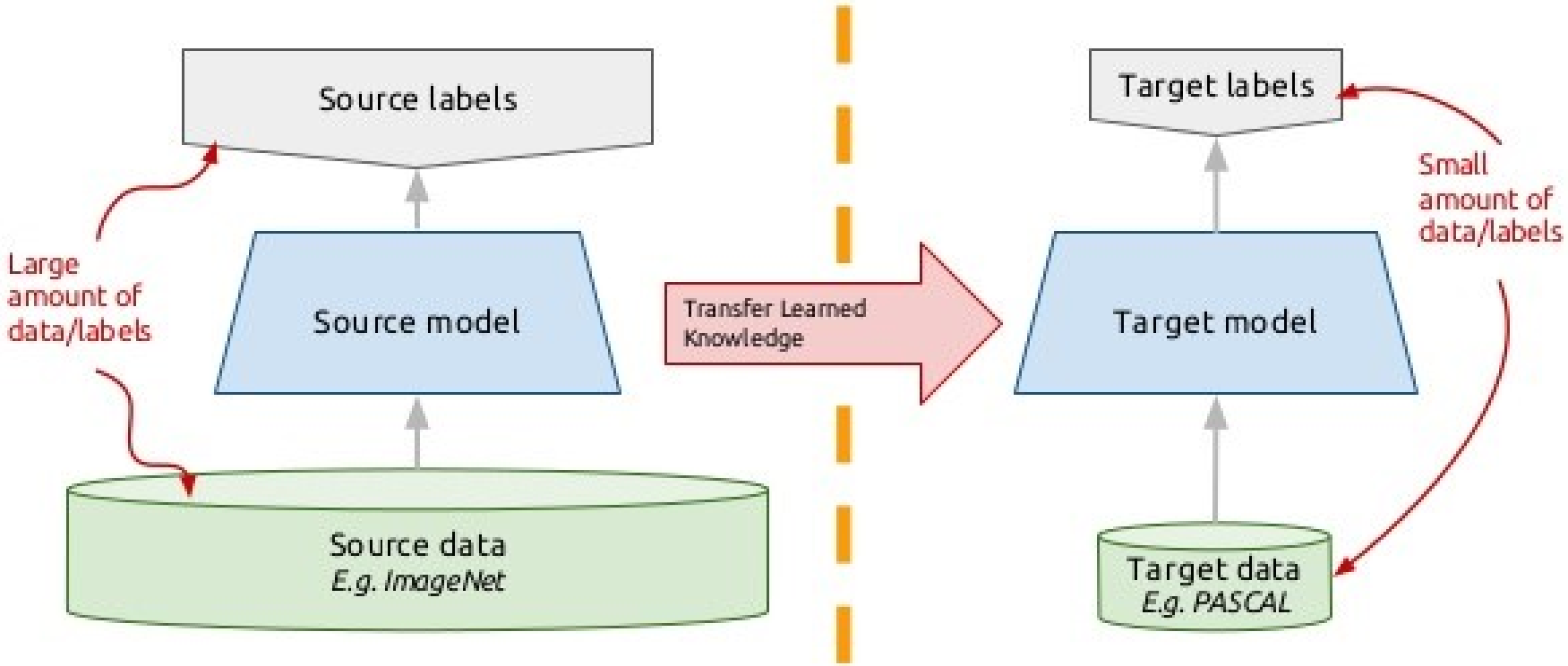


Transfer learning

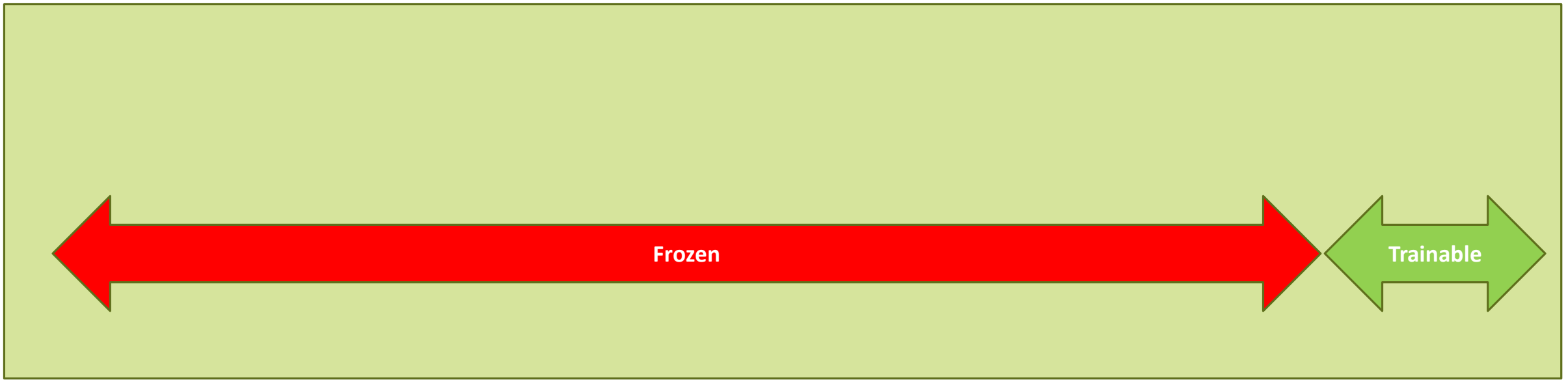


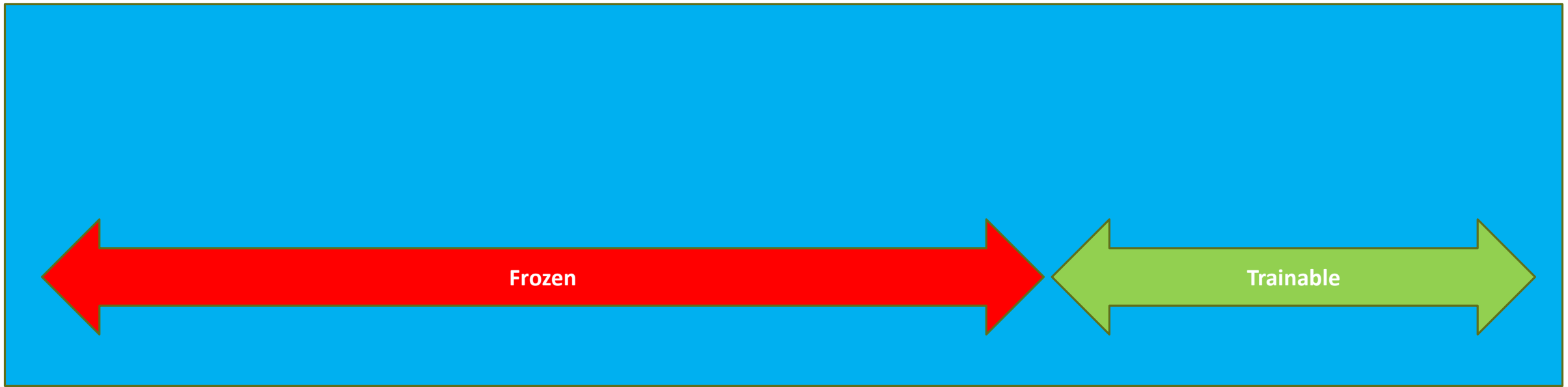


Transfer learning: idea



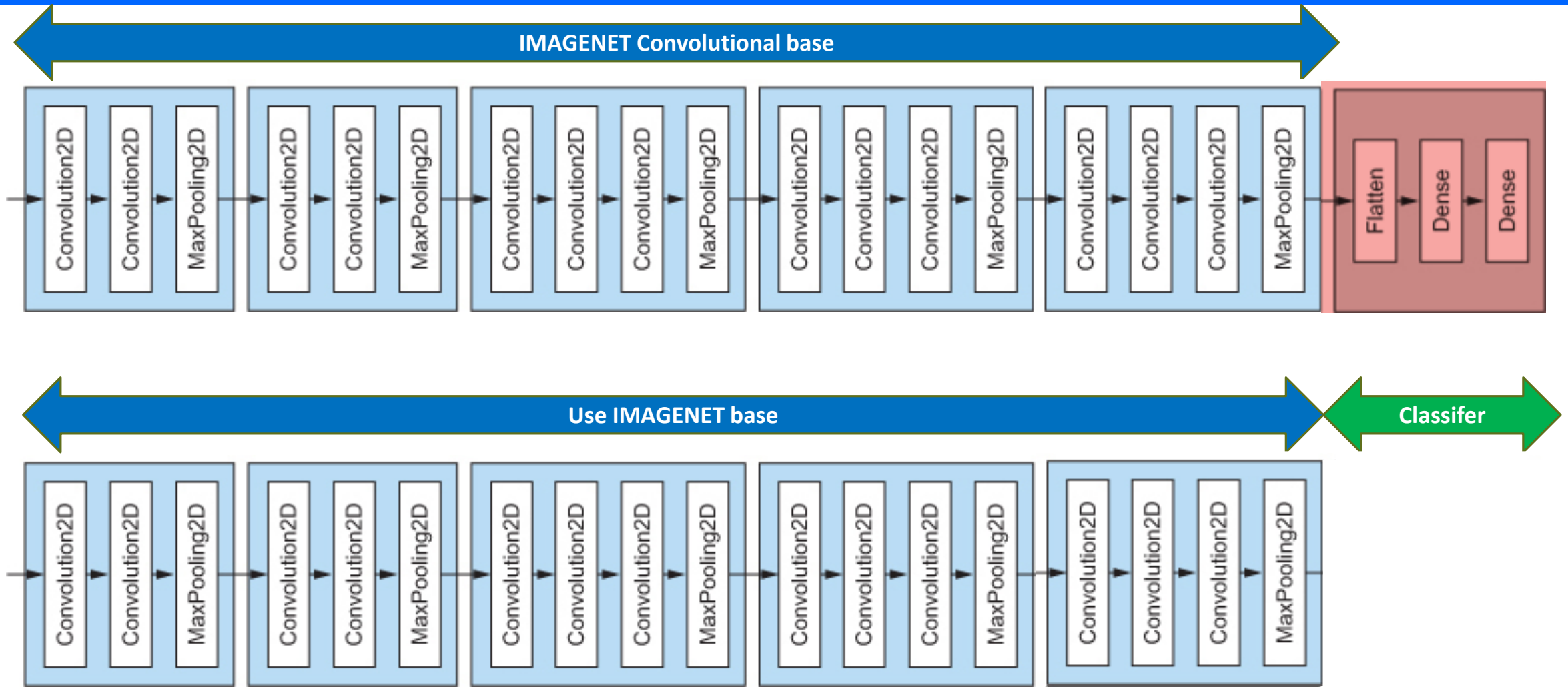
Network 1 or Network 2





Network 1

Doing feature extraction with a pre-trained model



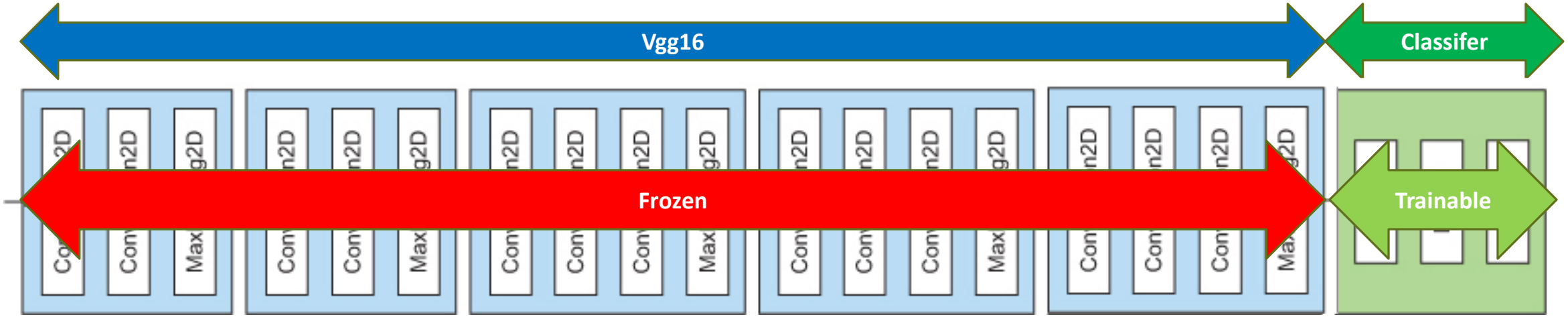
Acknowledgments & Credits are mentioned to inspirational resources presented in last slides. For more like this, https://sites.google.com/view/AIforEveryone

Network 1

Doing feature extraction with a pre-trained model

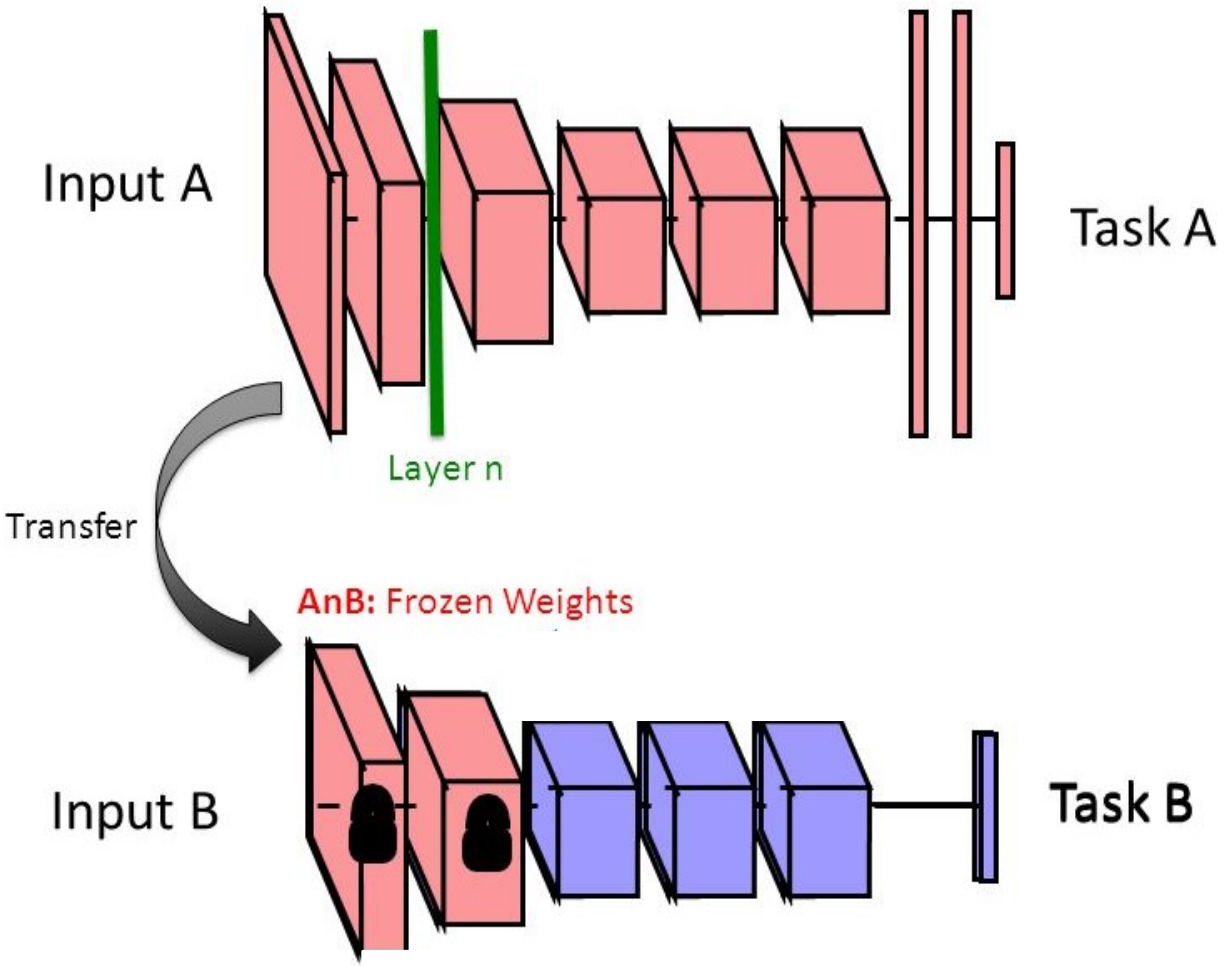
LAYER	TRAINABLE
vgg16	False
flatten_1	True
dense_1	True
dense_2	True

convbase.trainable = **False**



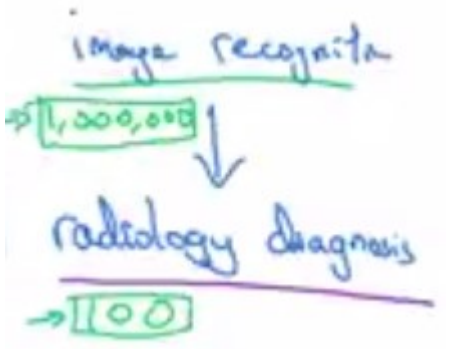
Acknowledgments & Credits are mentioned to inspirational resources presented in last slides. For more like this, https://sites.google.com/view/AIforEveryone

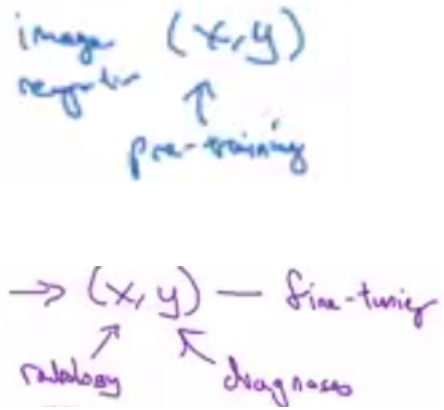
Re-purposing neural nets



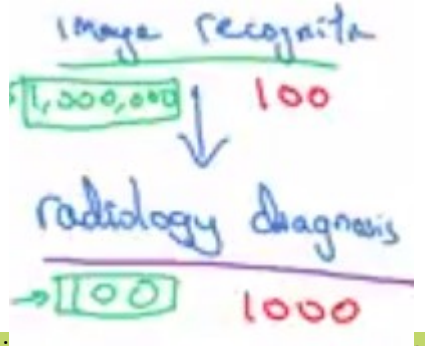
When does transfer transfer makes says

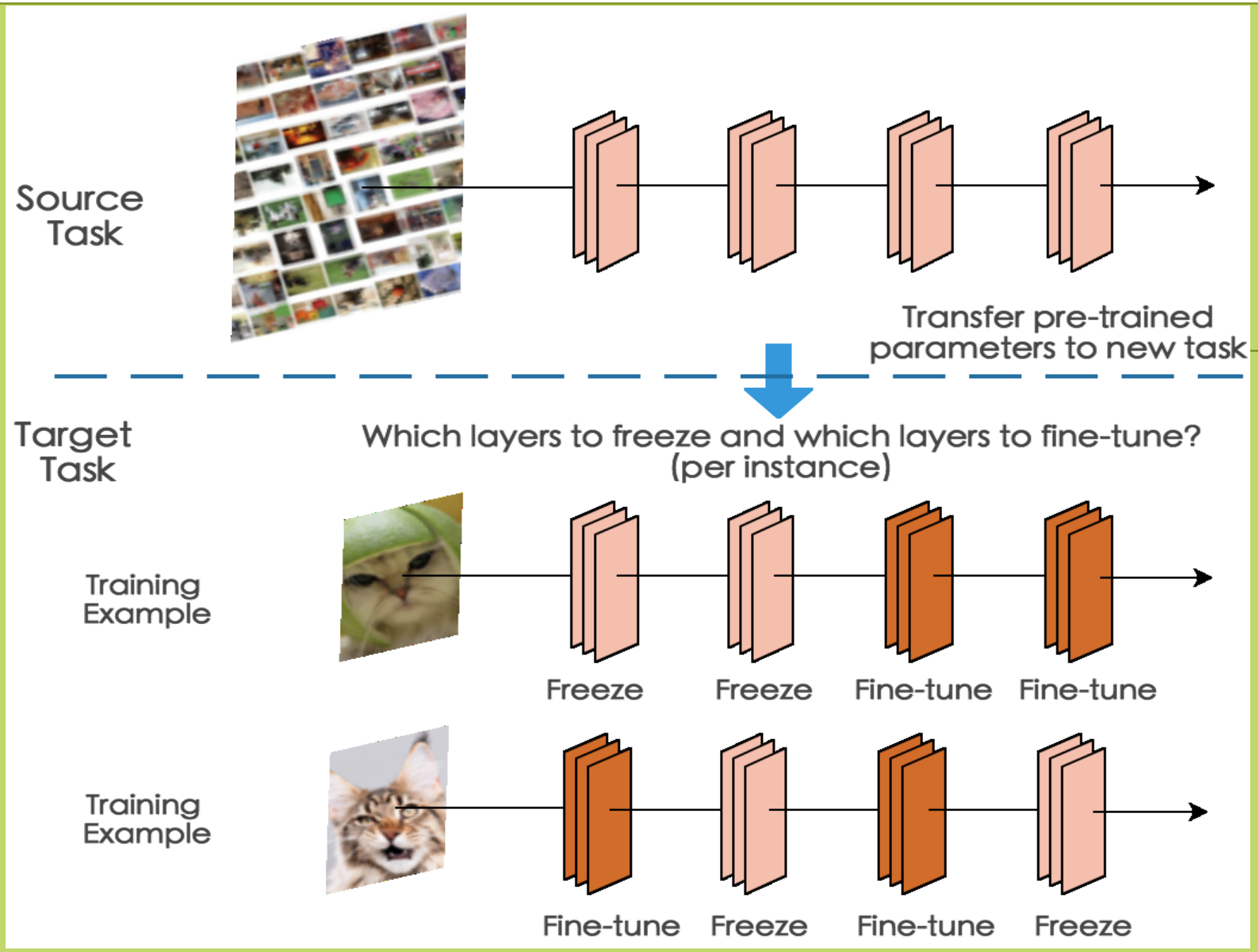
- Image recognition - Million examples
- Radiology recognition – 100 examples





- Speech – 10, 000 hr data
- Wake word – 1 hr data

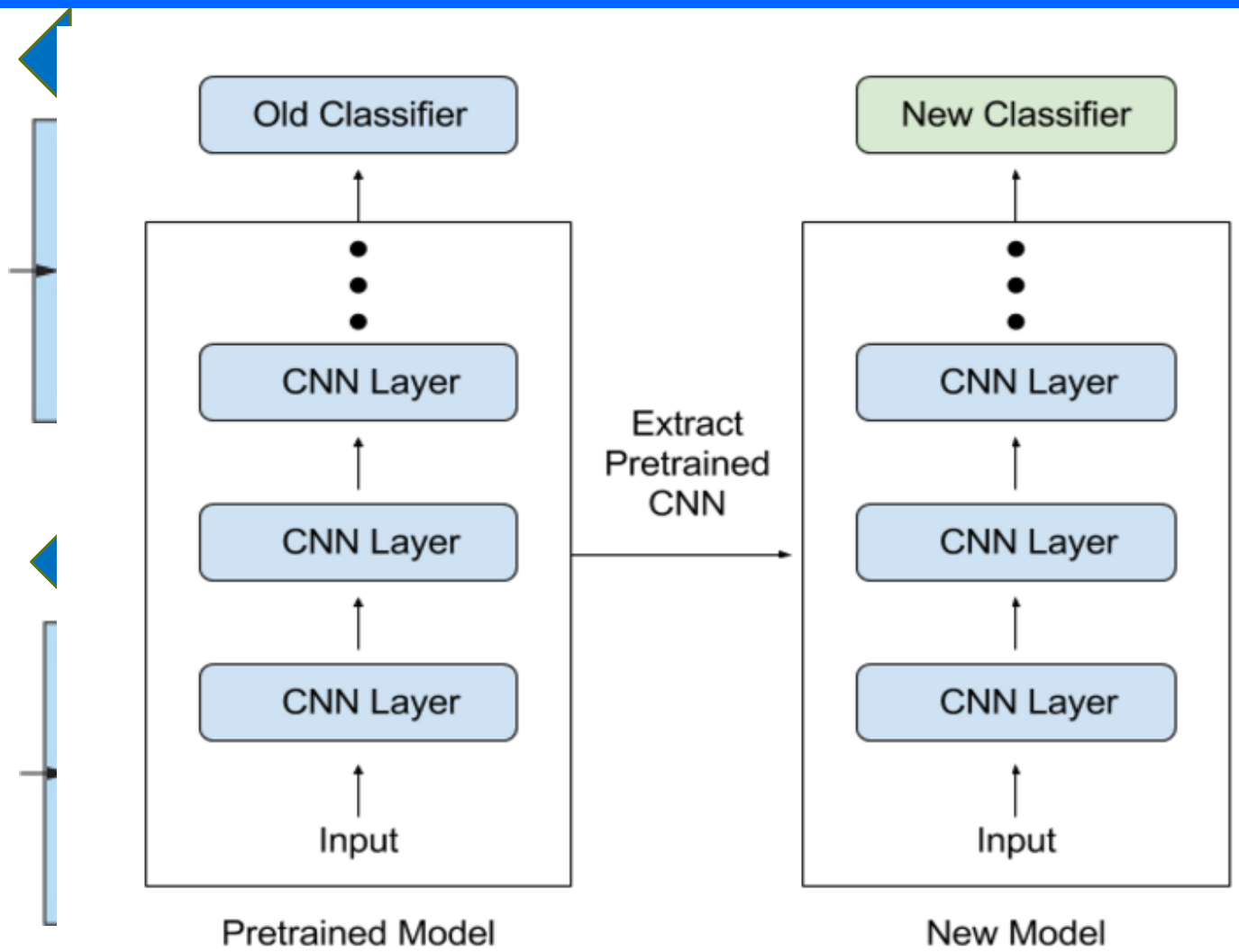


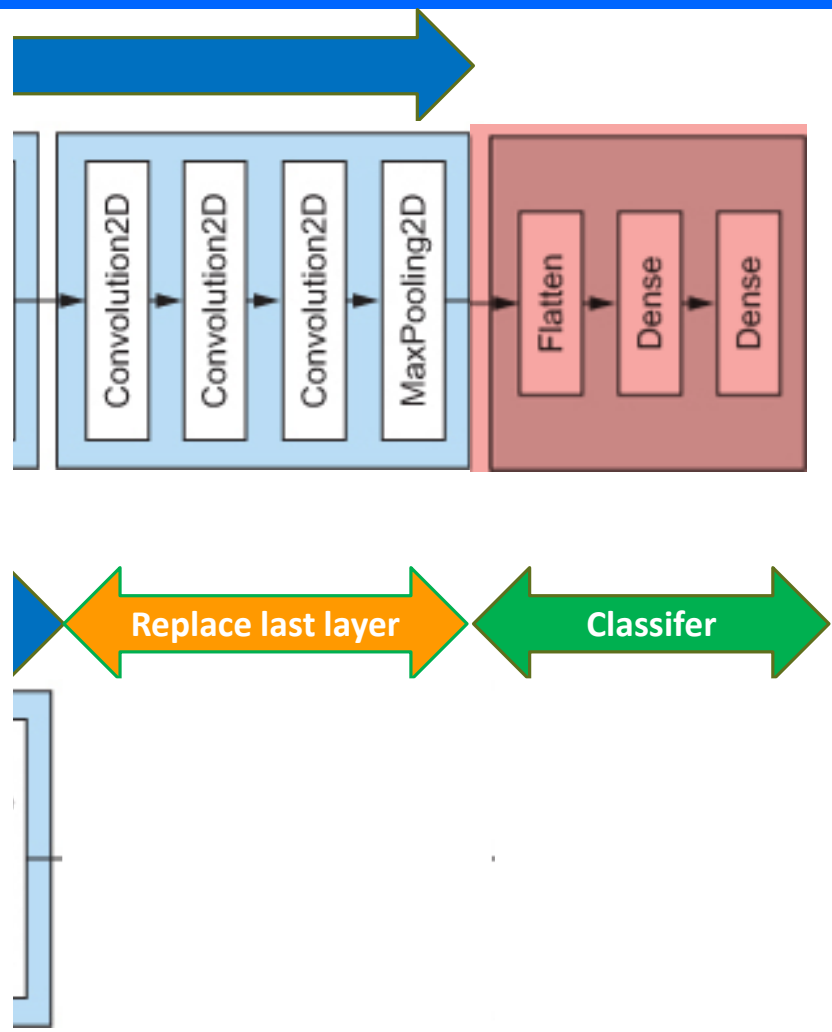


Network 2

Doing feature extraction with a pre-trained model

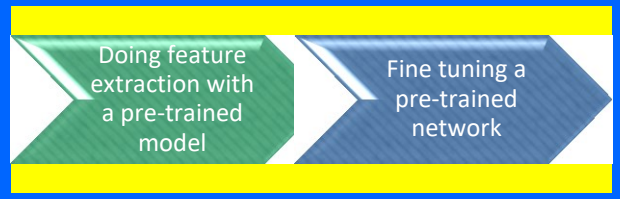
Fine tuning a pre-trained network





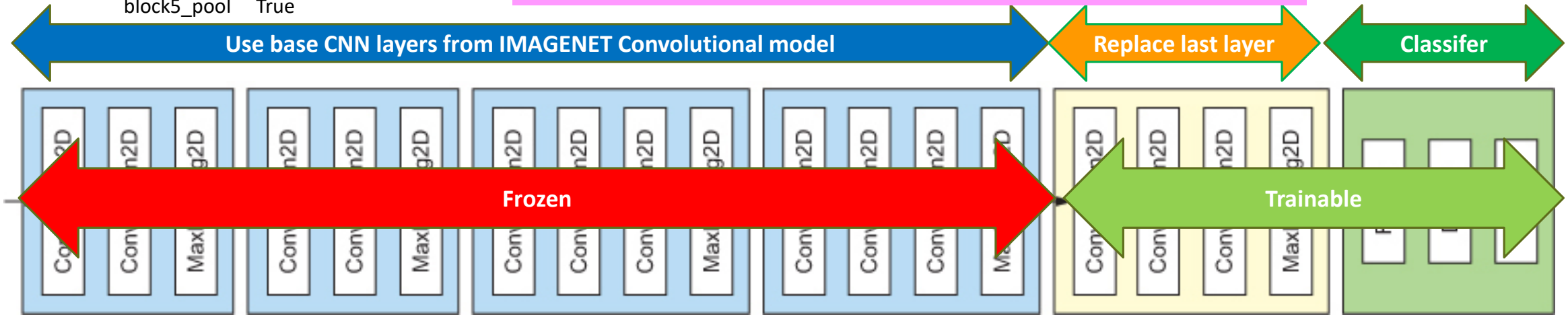
Visualization of transfer learning

Network 2




```
block1_conv1  False
block1_conv2  False
block1_pool   False
block2_conv1  False
block2_conv2  False
block2_pool   False
block3_conv1  False
block3_conv2  False
block3_conv3  False
block3_pool   False
block4_conv1  False
block4_conv2  False
block4_conv3  False
block4_pool   False
block5_conv1  True
block5_conv2  True
block5_conv3  True
block5_pool   True
```

```
convbase.trainable = True
isblock5conv1crossed = False;
for ainsidelayer in convbase.layers:
    layername = ainsidelayer.name
    if layername == 'block5_conv1':
        isblock5conv1crossed = True;
    if isblock5conv1crossed:
        ainsidelayer.trainable = True
        print("Setting trainable for " + str(layername))
    else:
        ainsidelayer.trainable = False
```



Acknowledgments & Credits are mentioned to inspirational resources presented in last slides. For more like this, https://sites.google.com/view/AIforEveryone

Data Augmentation

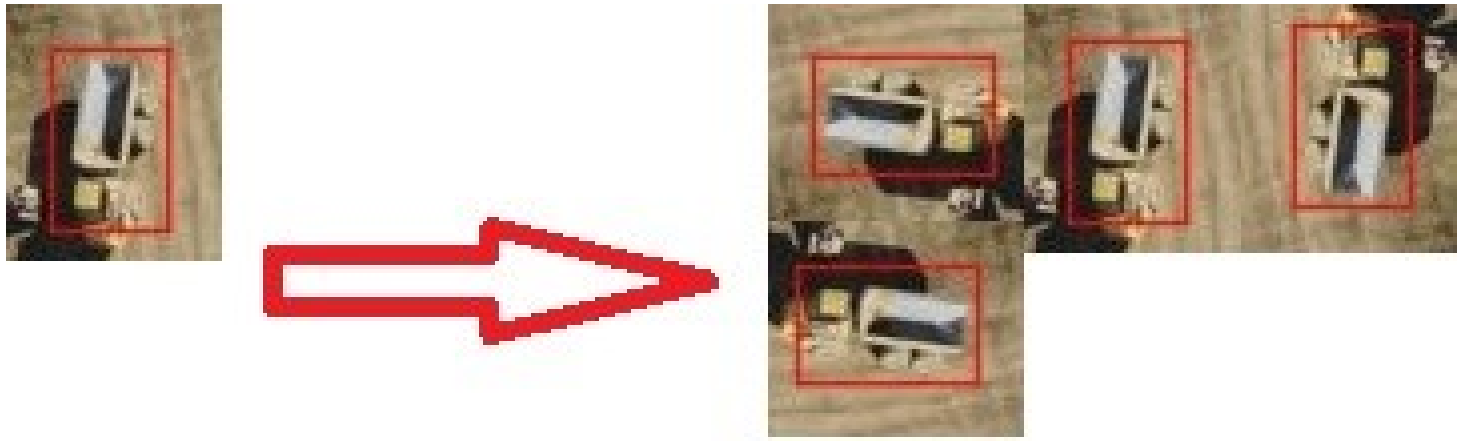
```
imageGeneratordataaugumented =
ImageDataGenerator(rescale=1.0/255,
rotation_range=0.01,
width_shift_range=0.01,height_shift_range=0.01,
shear_range=0.01,zoom_range=0.01,
horizontal_flip=True, fill_mode='nearest')
```

Training with Generator

The science behind how computers see

Visual world	What do CNN learn?
The visual world is fundamentally translation invariant	CNN learn patterns that are translation invariant
The visual world is fundamentally spatially hierarchical	CNN learn spatial hierarchies of patterns

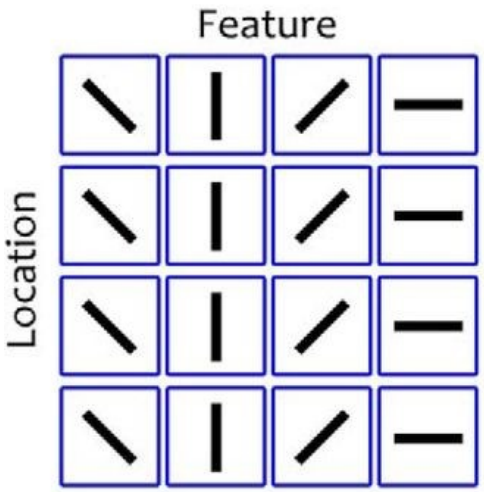
The visual world is fundamentally translation invariant

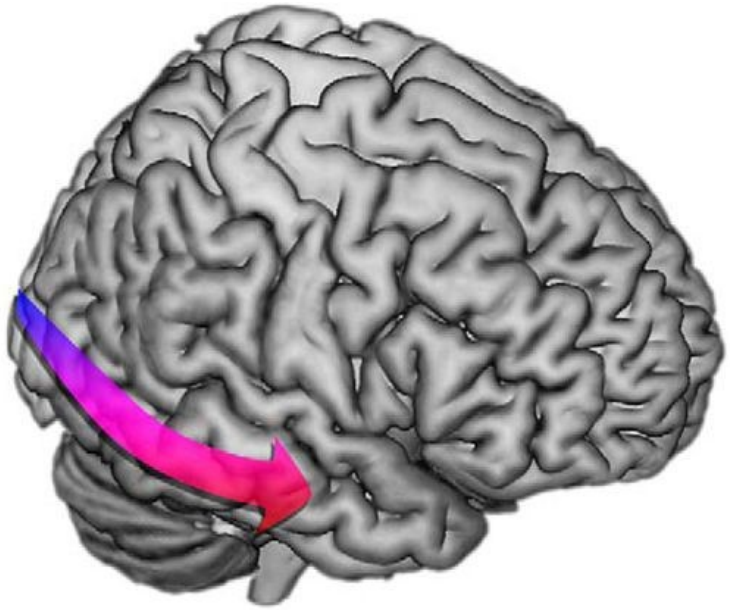


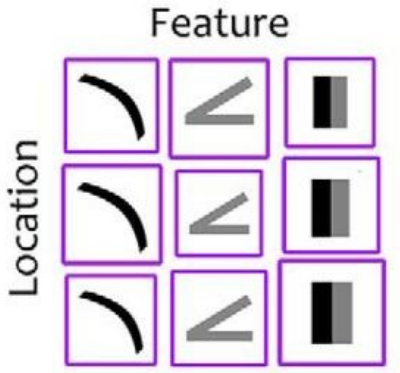
train

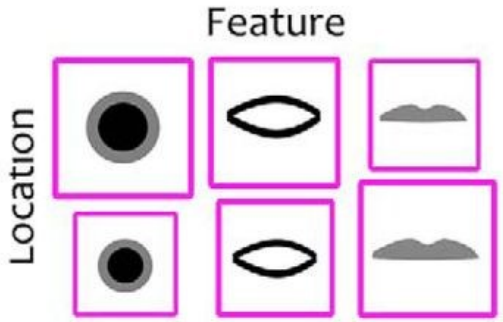
detect

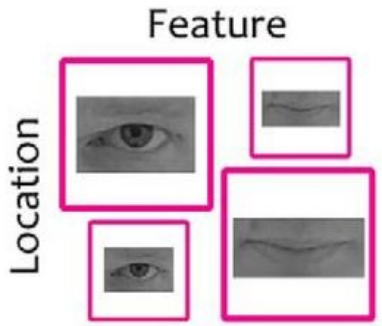
The visual world is fundamentally spatially hierarchical

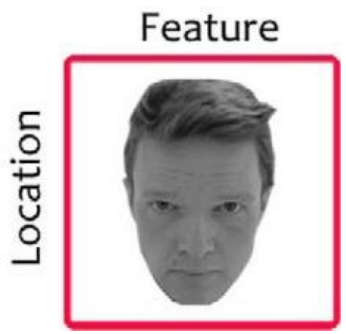










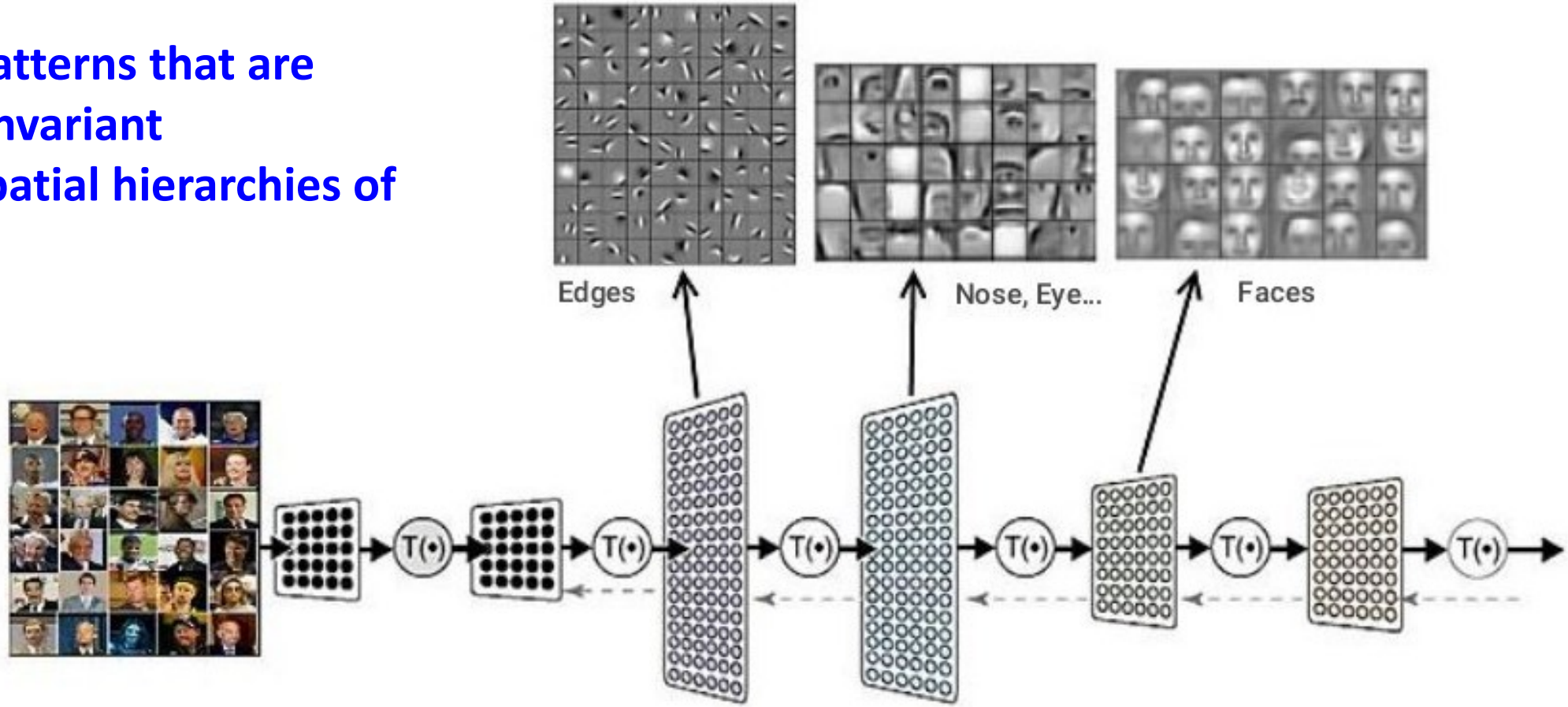


Acknowledgments & Credits are mentioned to inspirational resources presented in last slides. For more like this, https://sites.google.com/view/AIforEveryone

What CNN learns?

A deep neural network consists of a hierarchy of layers, whereby each layer transforms the input data into more abstract representations (e.g. edge -> nose -> face). The output layer combines those features to make predictions.

- CNN learn patterns that are translation invariant
- CNN learn spatial hierarchies of patterns



Acknowledgments & Credits are mentioned to inspirational resources presented in last slides. For more like this, https://sites.google.com/view/AIforEveryone

Hands-on

- How to use Transfer learning
  - 2_2_image_classification__cats_part2_using_Pretrained_model.ipynb

- Transfer learning in medical images
  - 4_Medical_image_classification_.ipynb