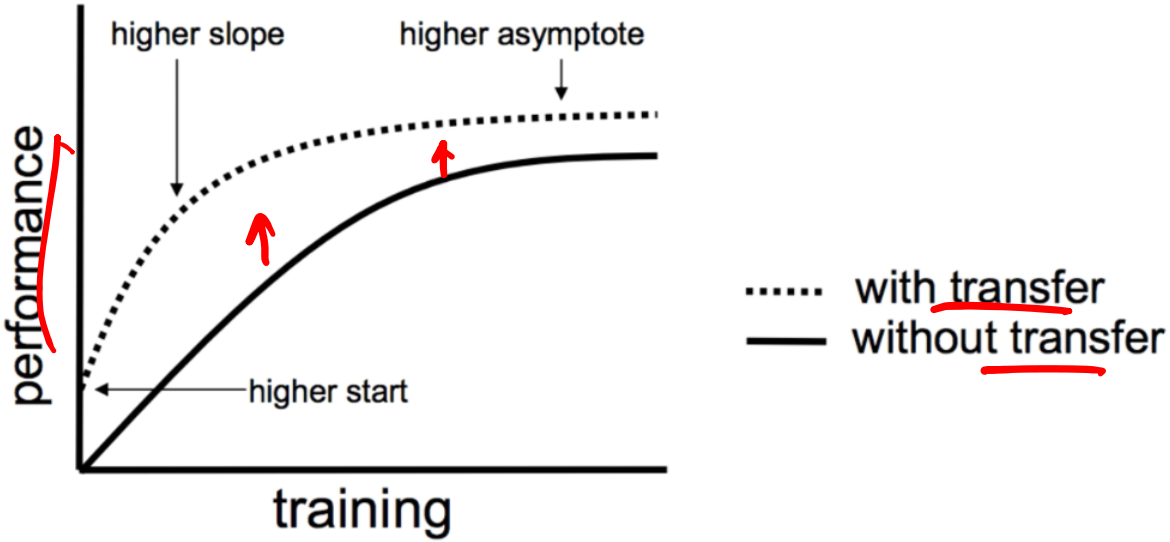
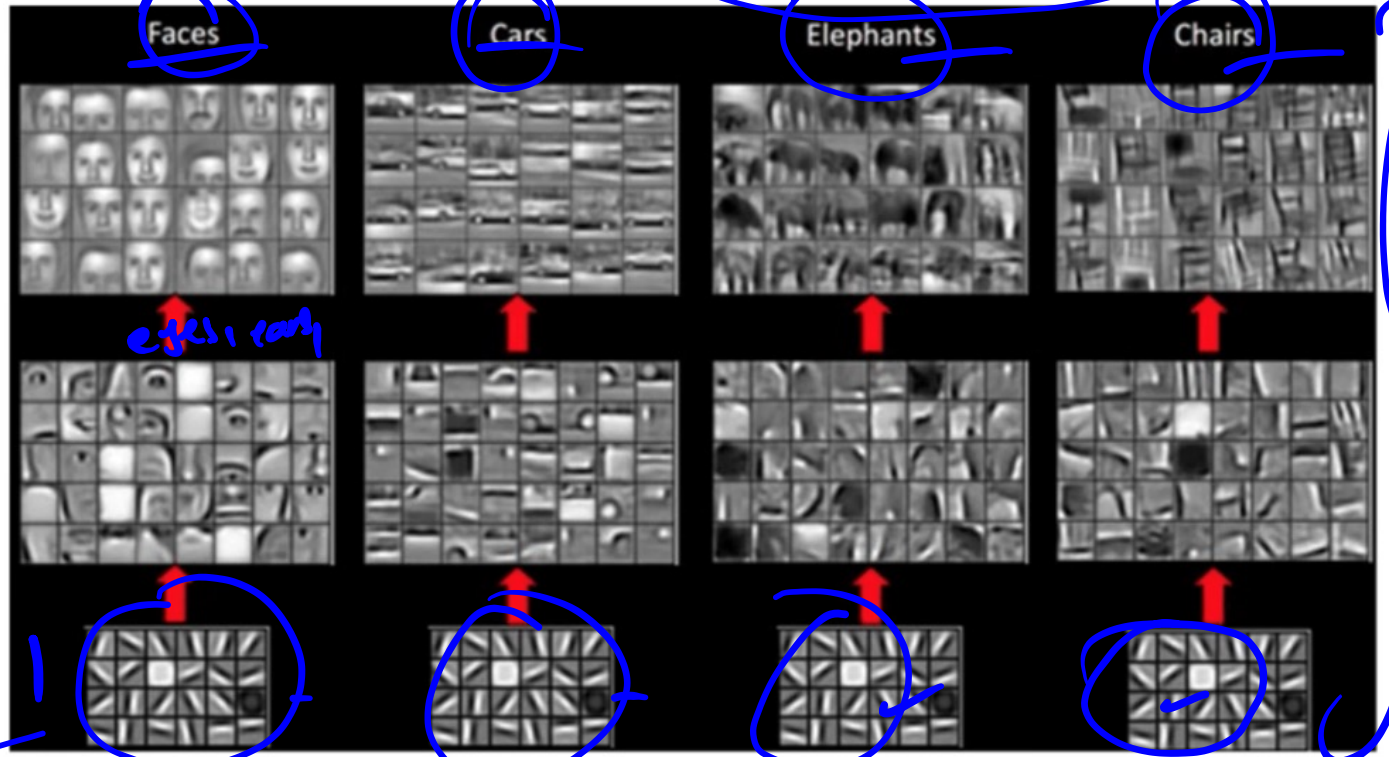
Transfer learning with TF-hub

Instructor: Dr. Darshan Ingle

- A very important topic in modern deep learning

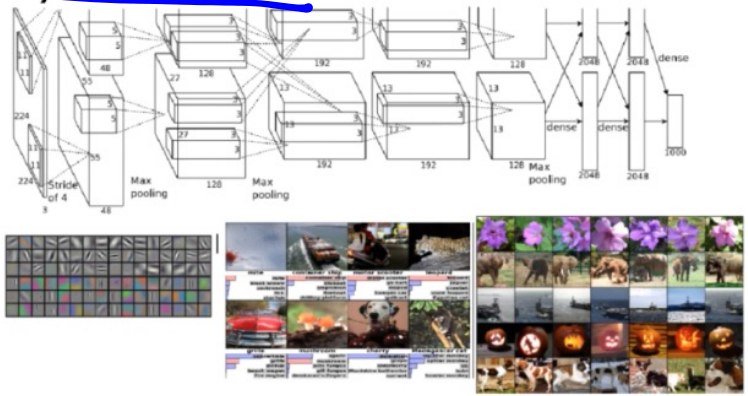


- Notice: *all* CNNs learn simple lines and strokes



- Main idea: the features I found from *one* task may be useful for *another* task
- Transfer Learning took off in the field of computer vision
- ImageNet - Large-scale image dataset (millions of images, 1k categories)
- Because the dataset is so *diverse*, weights trained on this dataset can be applied to a large number of vision tasks
    - Cats vs Dogs
    - Cars vs Trucks
    - Even microscope images / images never seen before!

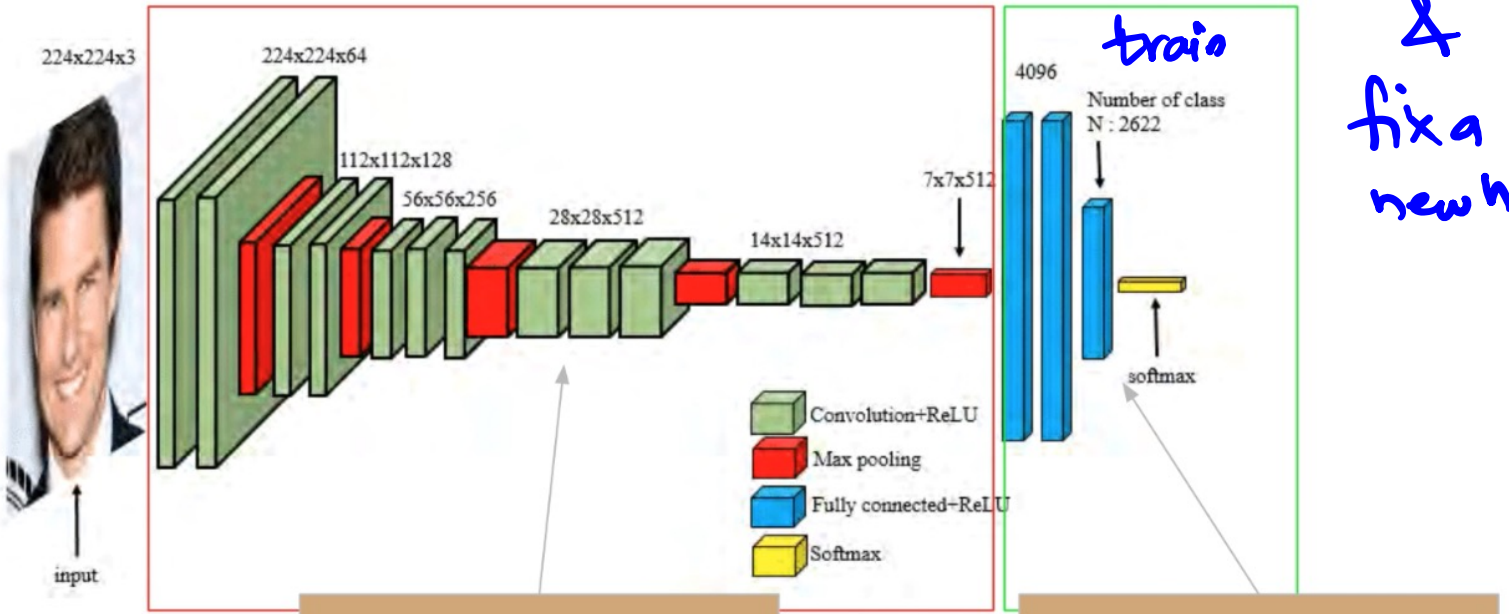
- It's not feasible for us to train on ImageNet ourselves
- Old days: training used to take days, weeks, even months!
- Now we can do it in minutes, using multi-GPU clusters, etc.
- The major CNNs which have won past ImageNet contests have publicly-released pre-trained weights
- And they are already included in Tensorflow/Keras!



Body

Head (Chop off the head) & fix a new head

train



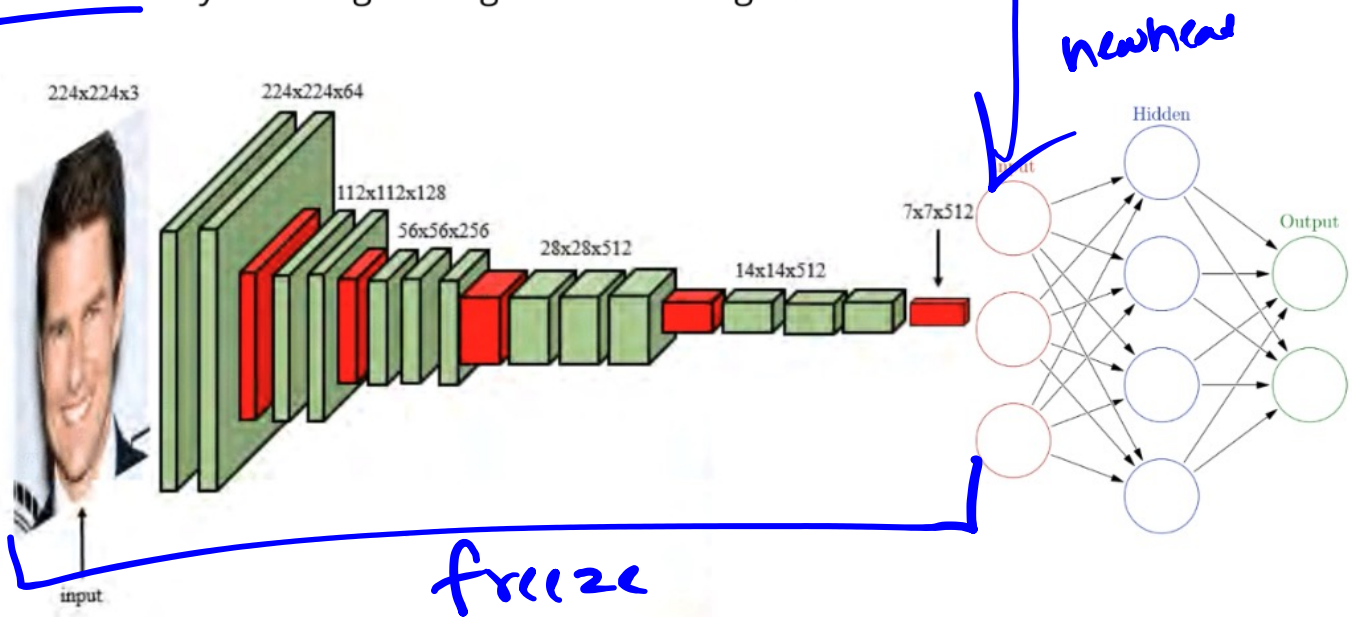
Feature transformer ("body")

ANN Classifier ("head")

Chop off the old “head”, add a new head!

Can be logistic regression or ANN

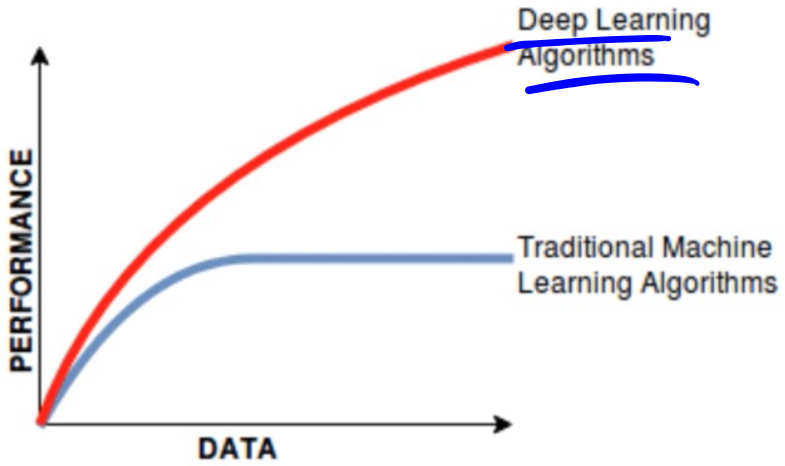
E.g. “cars vs trucks” may use a logistic regression with sigmoid



Don't need a lot of data to build a state-of-the-art model

With transfer learning, this work has been done for us - the earlier features were *already* trained on lots of data (ImageNet)

Small dataset + a lot less weights helps us train **fast**

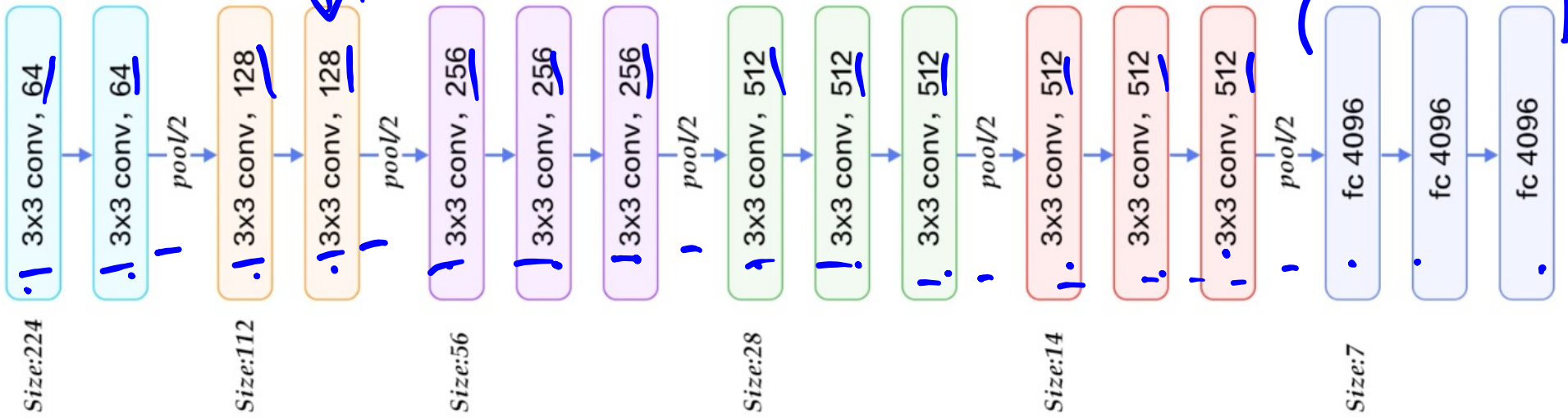


VGG

Named after the research group that created it - Visual Geometry Group

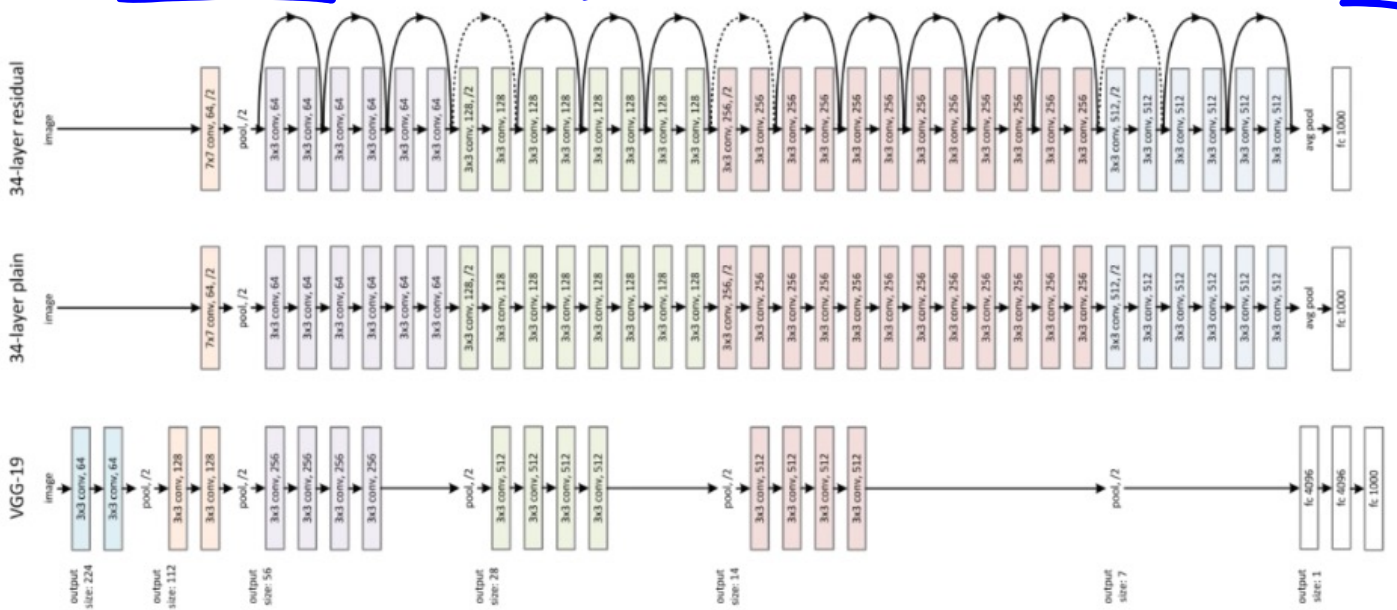
Not that different from CNNs we already know, just bigger

Options: VGG16, VGG19



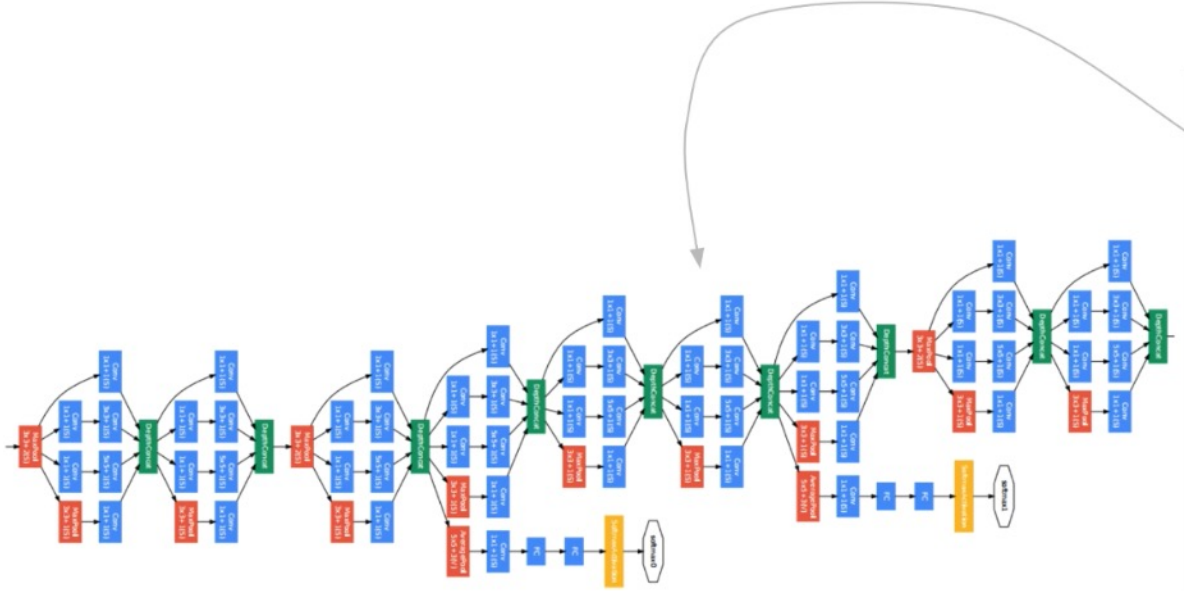
A CNN with *branches* (one branch is the identity function, so the other learns the *residual*)

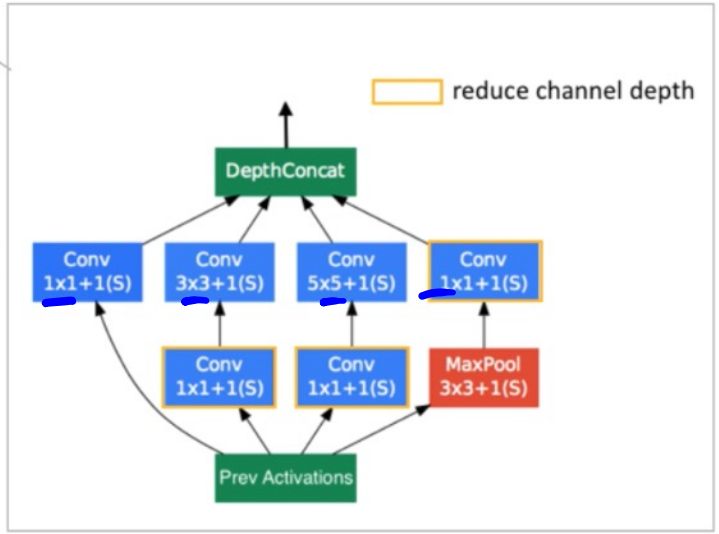
Variations: ResNet50, ResNet101, ResNet152, ResNet_v2, ResNeXt



Multiple convolutions in parallel branches

Instead of trying to choose different filter sizes (1x1, 3x3, 5x5, etc.) just try them all!

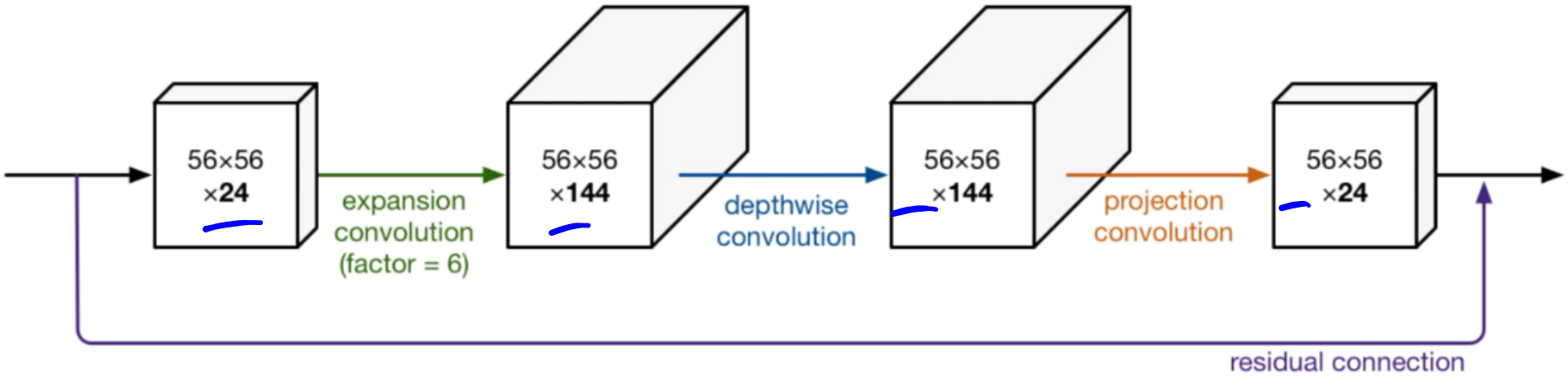




MobileNet

Lightweight: makes a tradeoff between speed and accuracy

Meant for less powerful machines (mobile, embedded)



Since we are using pre-built CNNs, our data must be formatted just like the original

We usually work with RGB pixel values in [0, 1] or [-1, +1]

VGG (for example) uses BGR with pixel values *centered* but not *scaled*

Just import the `preprocess_input` function from the same module as your pretrained network

```
from keras.layers import Input, Lambda, Dense, Flatten
from keras.models import Model
from keras.applications.resnet50 import ResNet50, preprocess_input
# from keras.applications.inception_v3 import InceptionV3, preprocess_input
from keras.preprocessing import image
from keras.preprocessing.image import ImageDataGenerator
```

When you are first learning about deep learning, it is convenient to have MNIST, CIFAR-10, SVHN, ... as a CSV / Numpy array

In the real world, images are not CSVs, images are images!

i.e. image **files** (JPEG, PNG, ...)

Real images are much larger than MNIST (28x28) and CIFAR-10 (32x32)

VGG and ResNet are trained on ImageNet images resized to 224x224

ImageNet

How much space does it take to store 1 million images of size 224 x 224?

1 million images x (224 x 224 x 3) bytes / image

~150 billion bytes

~140 GB

Would not fit in RAM on a standard machine

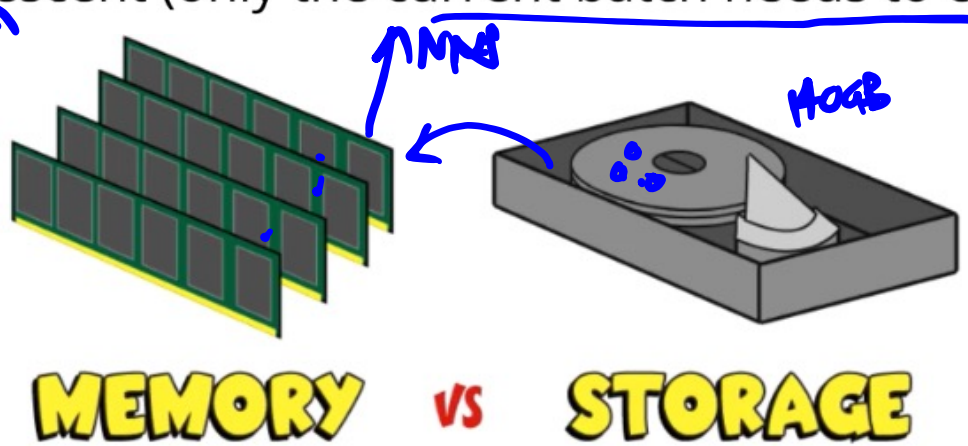
Approach the problem like an engineer, not that of a simple API user copying code off the Internet

Recognize the difference between disk and memory

Disk: slow, memory: fast

Model reads data from memory; image files live on disk

We do *batch* gradient descent (only the current batch needs to exist in memory)



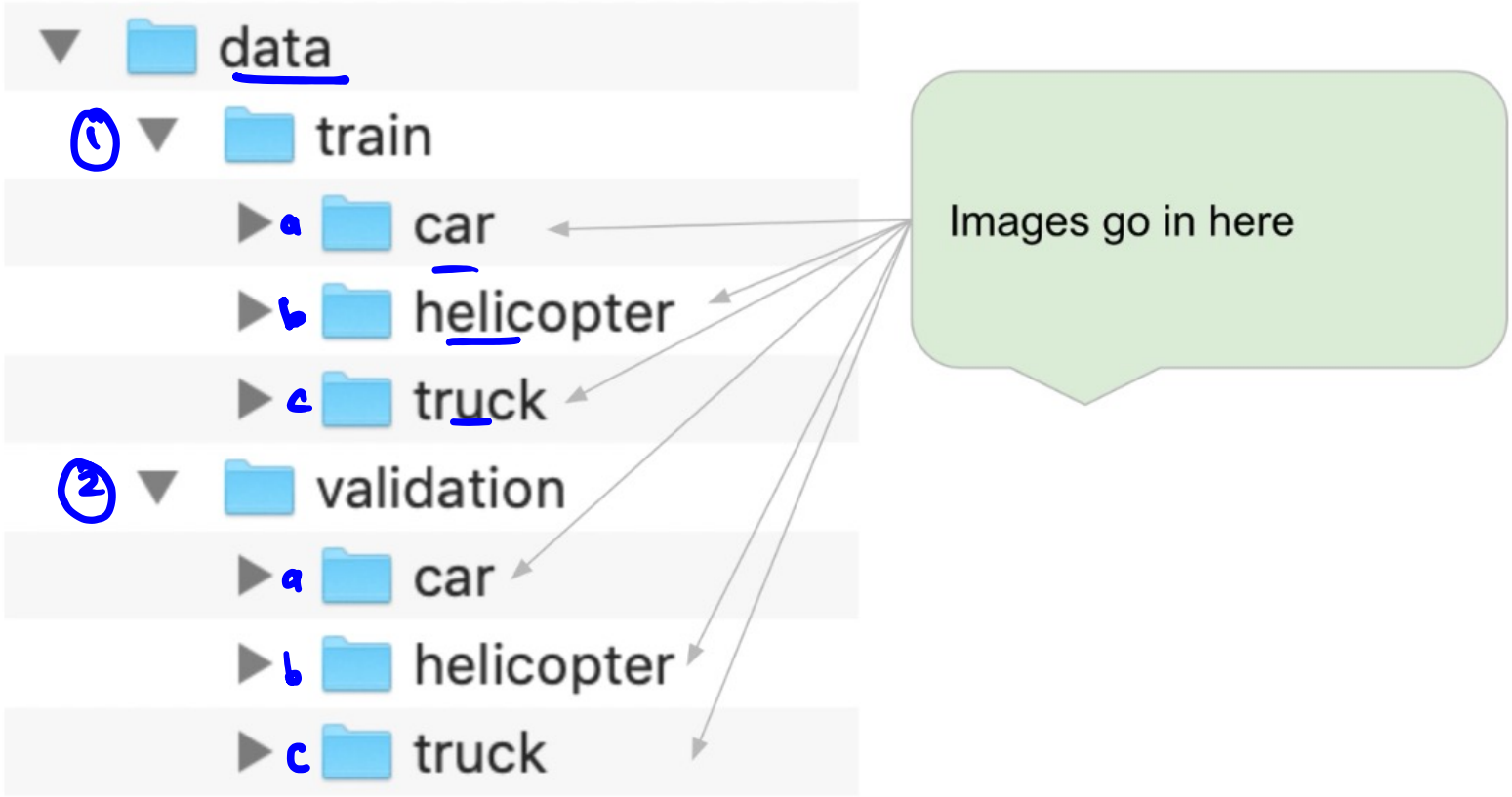
- Assume we have 2 arrays/lists: list of filenames, list of labels
- Suppose batch_size = 32

32000

n_bat = #Batch = 1000

```
for i in range(n_batches):
  x = load_images(filenames[i: i + 32])
  y = labels[i: i + 32]
  model.train_on_batch(x, y)
```

- The main ingredients:
- 1 **`gen = ImageDataGenerator()`**
  - Automatically generates data in batches
  - Data augmentation (shifting, rotation, flipping, ...)
  - Preprocessing (via preprocess_input)
- 2 **`generator = gen.flow_from_directory()`**
  - Specify target image size (e.g. resize all images to be 224 x 224), batch size
- 3 **`model.fit_generator(generator)`**
  - Used in place of `model.fit(X, Y)`


data
1
train
a
car
b
helicopter
c
truck
2
validation
a
car
b
helicopter
c
truck
Images go in here

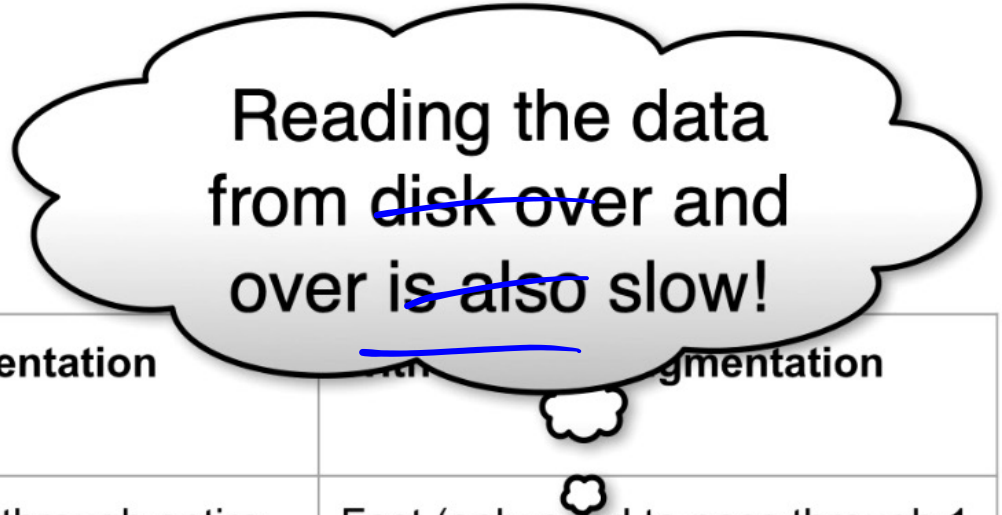
Approach #1 - use data augmentation with ImageDatagenerator

- Entire CNN computation must be inside loop

Approach #2 - precompute Z without data augmentation

- Only need to train a logistic regression on (Z, Y)

	With data augmentation	Without data augmentation
Speed	Slow (must pass through entire CNN)	Fast (only need to pass through 1 dense layer)
Generalization / Accuracy	Possibly better for generalization	Possibly worse for generalization



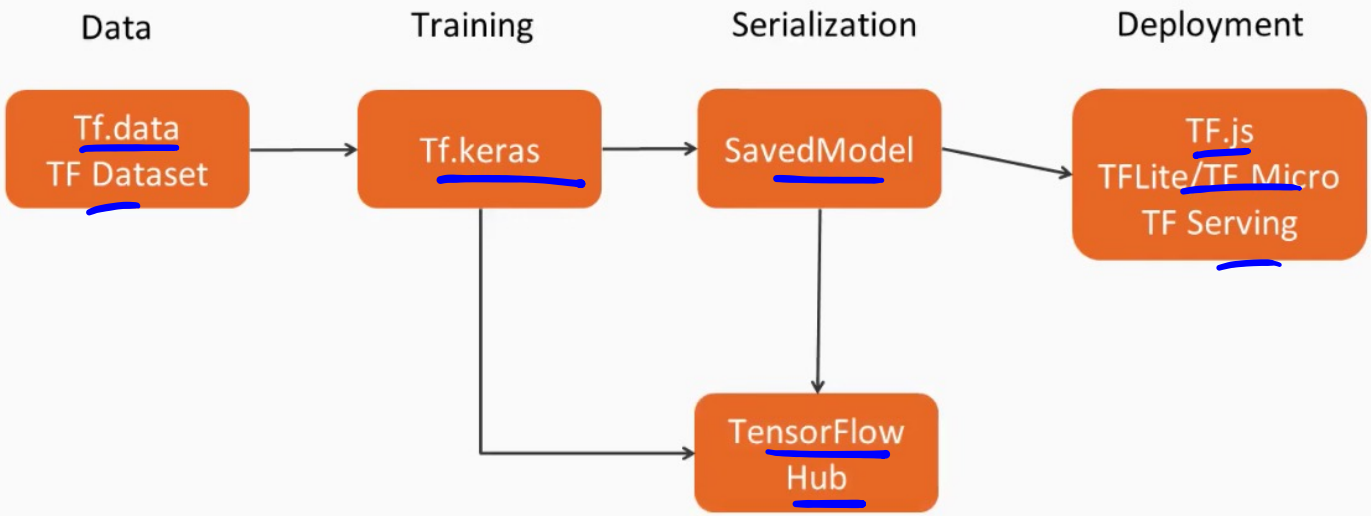
	With data augmentation	**gmentation**
Speed	Slow (must pass through entire CNN)	Fast (only need to pass through 1 dense layer)
Generalization / Accuracy	Possibly better for generalization	Possibly worse for generalization

What Is TensorFlow Hub?

An open-source repository of ready-to-use pre-trained models

- Announced at TF Dev Summit 2018
- Use the models directly

- More than just SavedModels on TensorFlow Hub
- TensorFlow.js
- TensorFlow Lite + Metadata
- Coral Edge TPU



Image

- Classification
- Object detection
- Style Transfer
- Generator, and so on

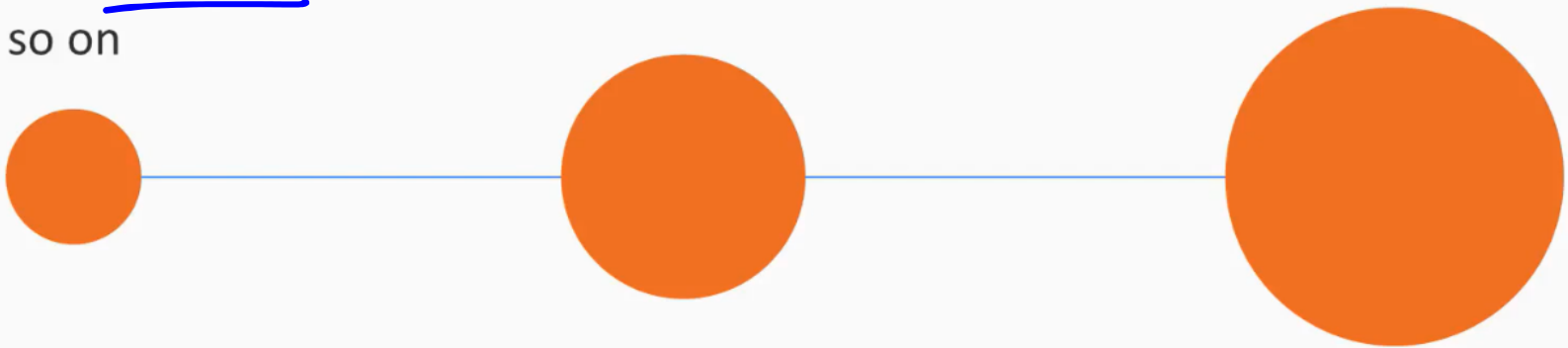
Text

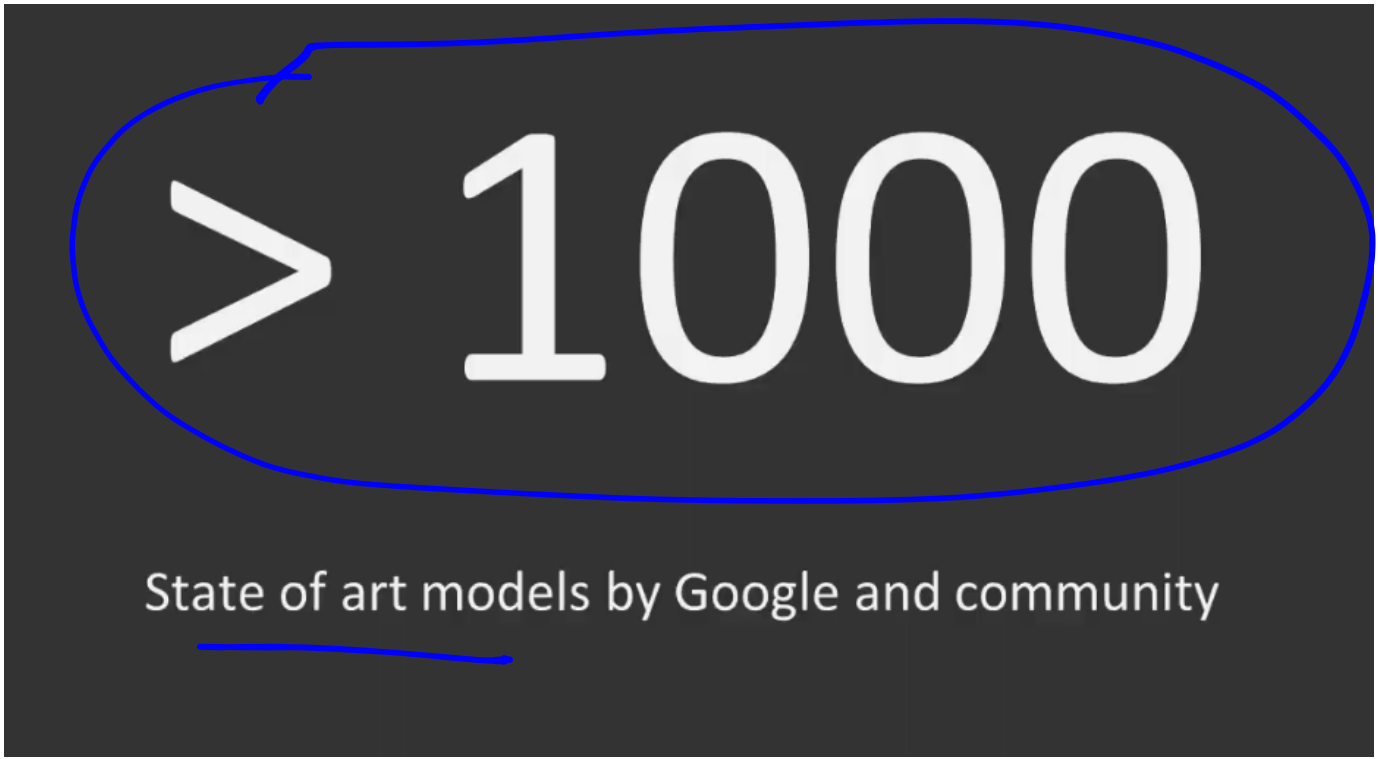
- Classification
- BERT
- Embedding

Video

- Classification
- Generator

Speech

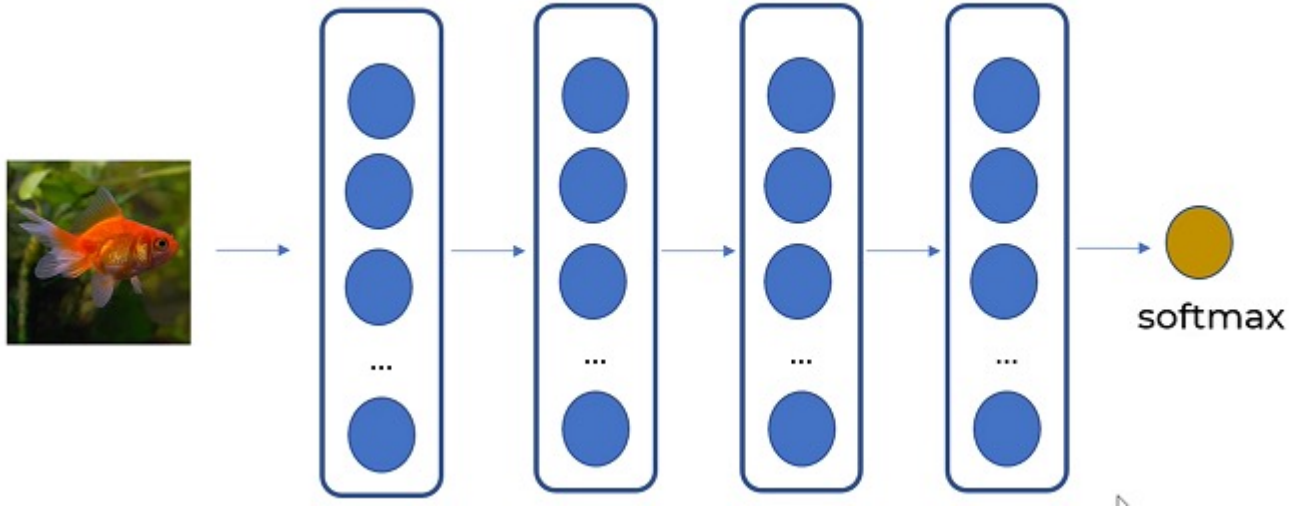


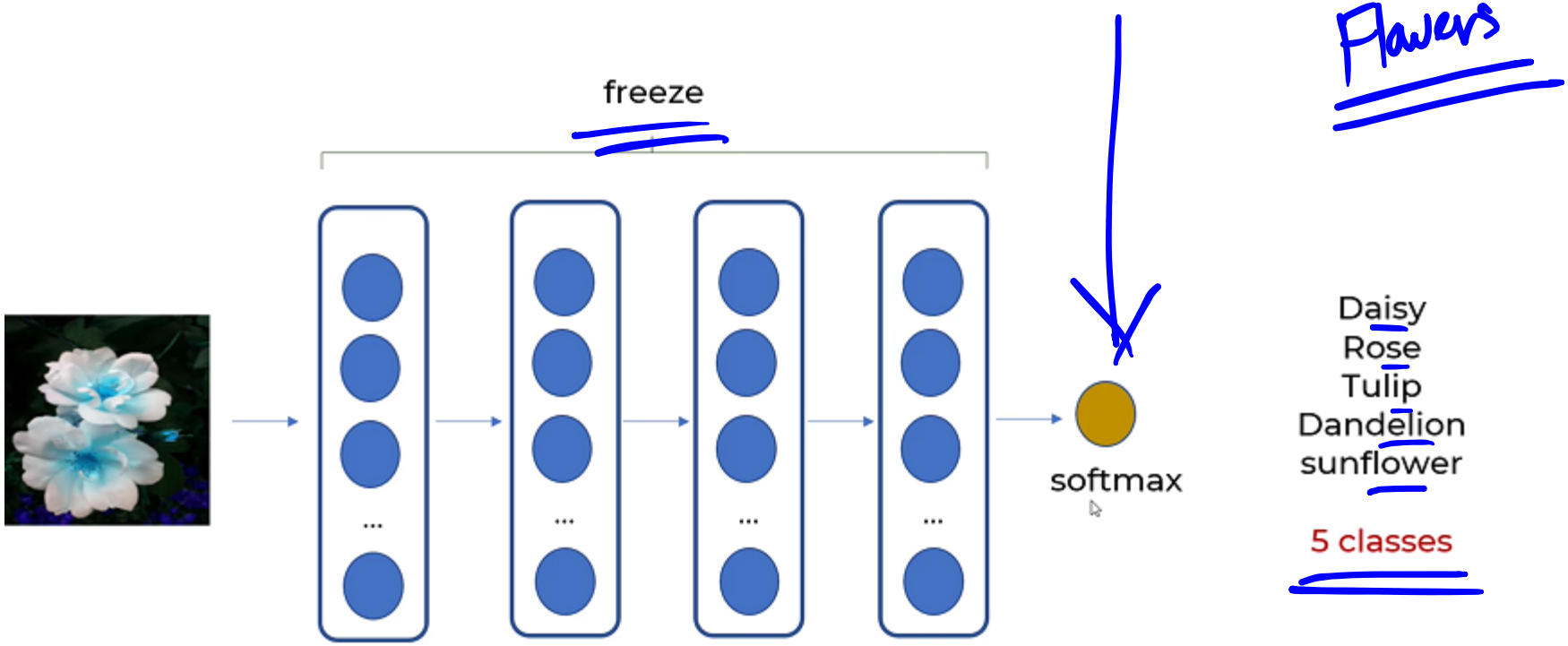

> 1000
State of art models by Google and community

- Google
- DeepMind
- MediaPipe
- Kaggle

- NVIDIA
- Microsoft AI for earth
- The metropolitan Museum of Art
- iNaturalist

https://github.com/tensorflow/hub/tree/master/tfhub_dev


softmax
Goldfish
great white shark
tiger Shark
Hammerhead
electric ray
Hen
...
1000 classes


freeze
softmax
Flowers
Daisy
Rose
Tulip
Dandelion
sunflower
5 classes

#LifeKoKaroLift

Thank You!

Connect me: https://www.linkedin.com/in/dr-darshan-ingle-corporate-trainer/