NUS-ISS
Pattern Recognition using Machine Learning System

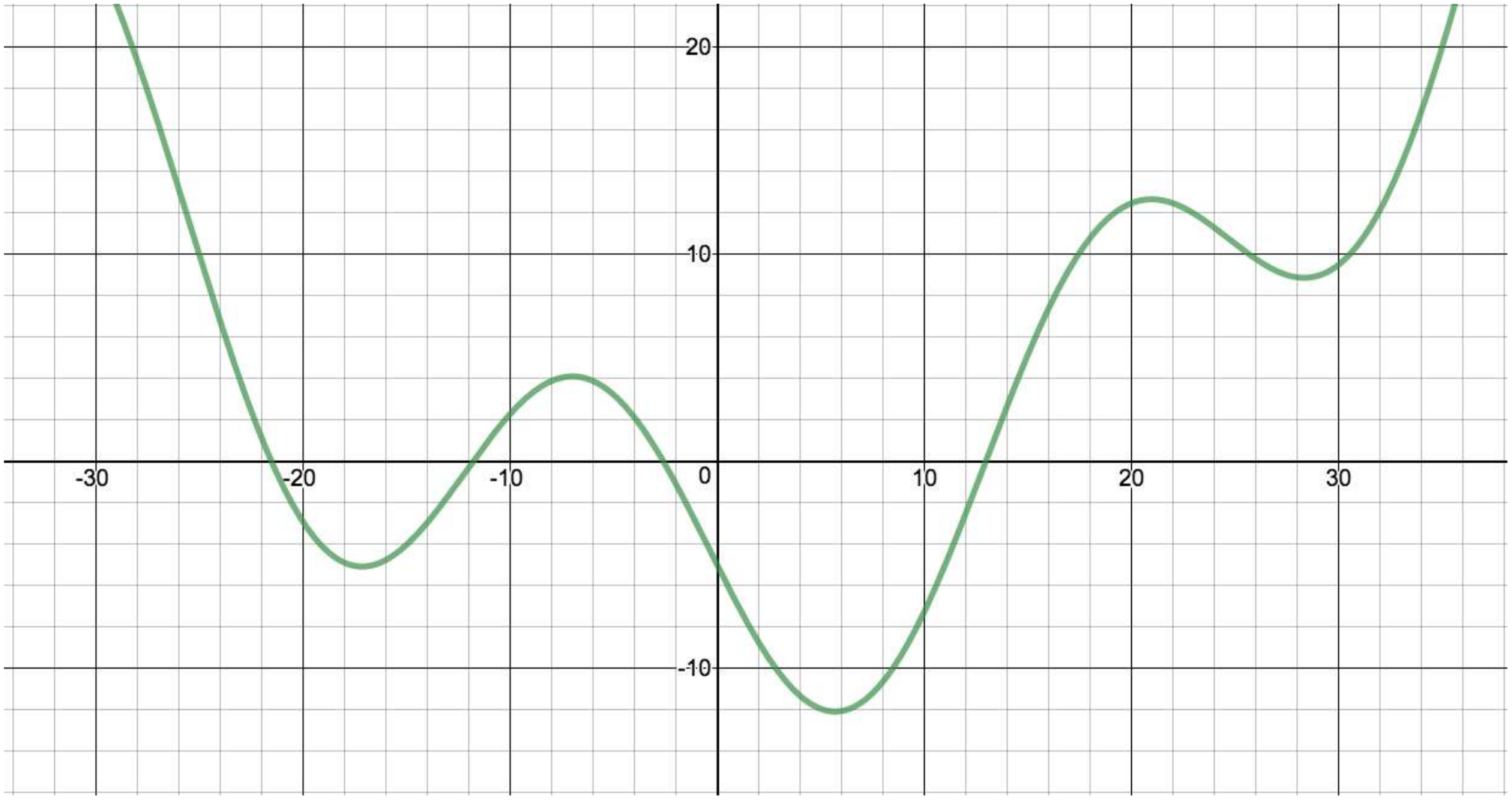
The need of a deeper network and functional APIs

by Dr. Tan Jen Hong

© 2020 National University of Singapore.
All Rights Reserved.

Representation

Do you think deep neural net can represent this function?



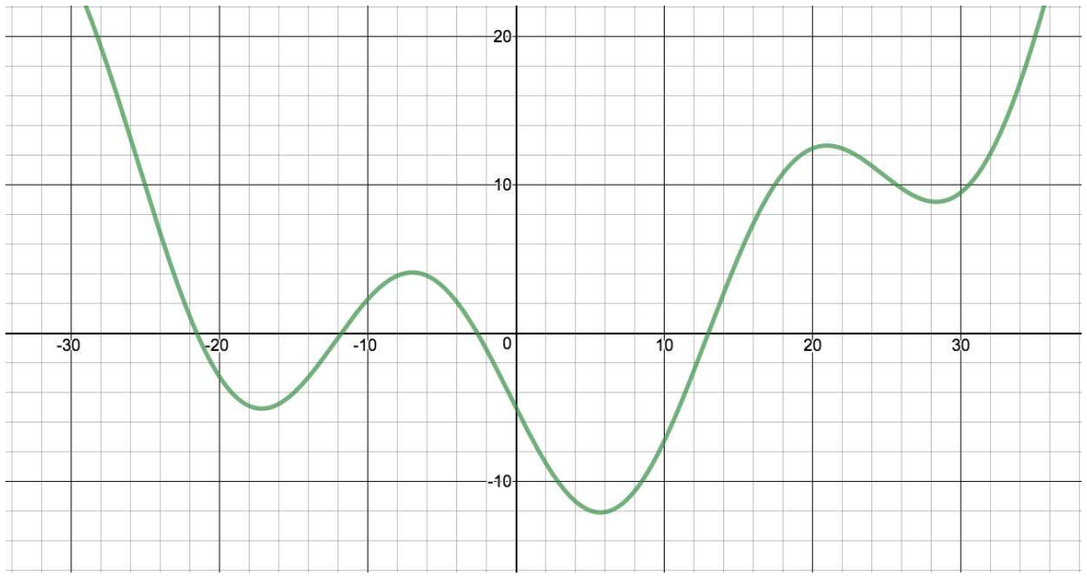
Source: By Brendan Fortuner

© 2020 National University of Singapore.
All Rights Reserved.

Universal approximation theorem

A theorem for all problem?

- A feedforward network with a single layer is enough to represent any function
- What is the implication?
- With the theorem, is deep neural network the panacea for all our challenges?



Source: By Brendan Fortuner

© 2020 National University of Singapore.
All Rights Reserved.

The extreme

An equation for all?

- Do you think it is possible to have one equation, with only a change in one parameter and generate the below four figure?

Source: By Laurent Boué

© 2020 National University of Singapore.
All Rights Reserved.

The extreme

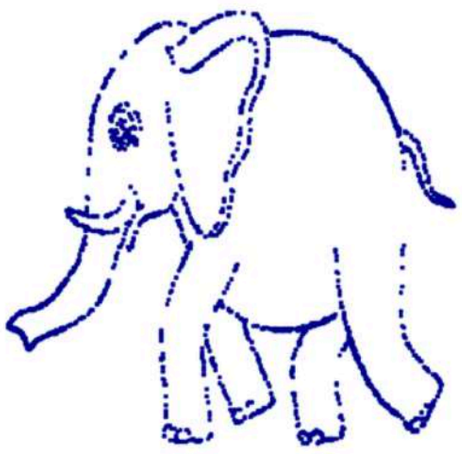
An equation for all?

- In fact, all the samples of any arbitrary dataset can be reproduced by (where each sample has the form (x, y))

$$y = f_\alpha(x) = \sin^2\left(2^{x\tau} \arcsin\sqrt{\alpha}\right)$$

- x must be a positive integer, τ is a constant controls the desired acuracy

$\alpha = 0.28495951\ \cdots$



$\alpha = 0.74933466\ \cdots$

$\alpha = 0.70704013\ \cdots$

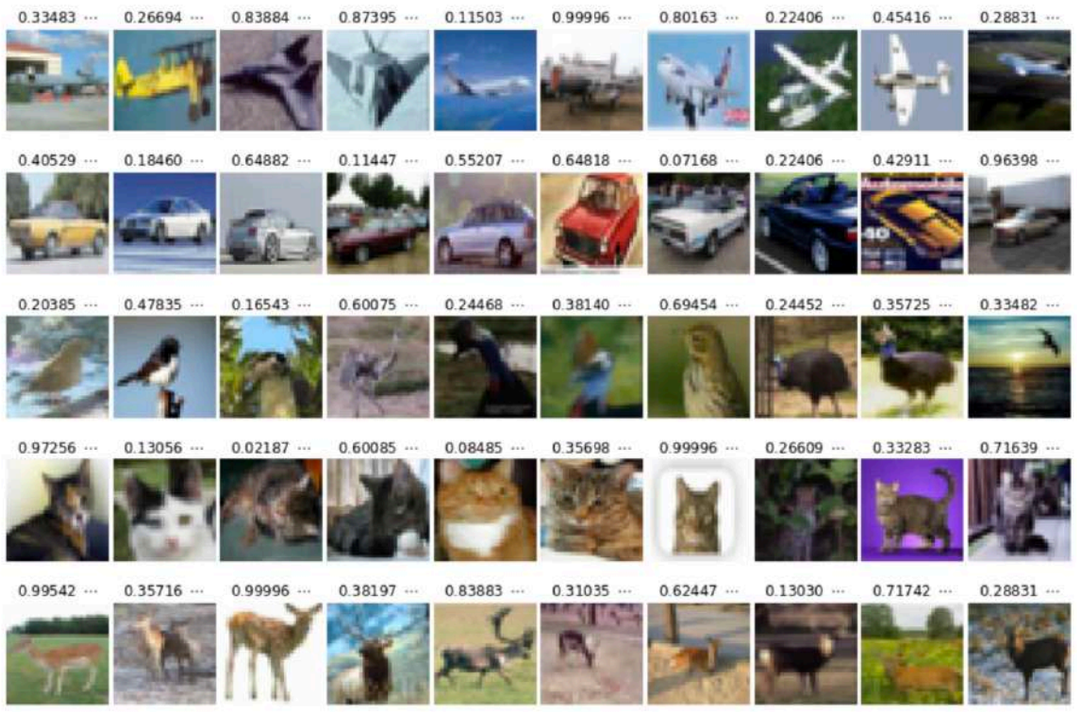
$\alpha = 0.46799746\ \cdots$

Source: https://arxiv.org/pdf/1904.12320.pdf

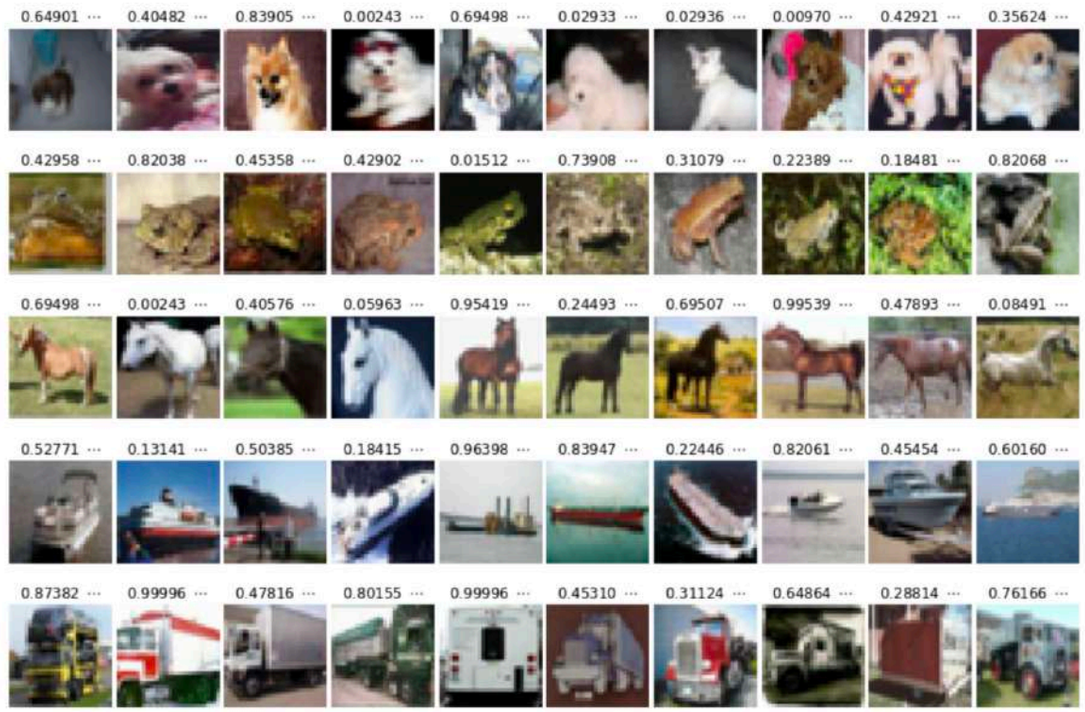
© 2020 National University of Singapore.
All Rights Reserved.

The extreme

An equation for all?



- Or even for this



Source: https://arxiv.org/pdf/1904.12320.pdf

$$y = f_\alpha(x) = \sin^2\left(2^{x\tau} \arcsin\sqrt{\alpha}\right)$$

© 2020 National University of Singapore.
All Rights Reserved.

Back to the real world

The issue at hand

- Although by universal approximation theorem, possible to perform good classification on a single layer net, so far no learning algorithm can achieve that

- Key: a single layer net can approximate any function (proven), but the theorem does not provide any clue to achieve that

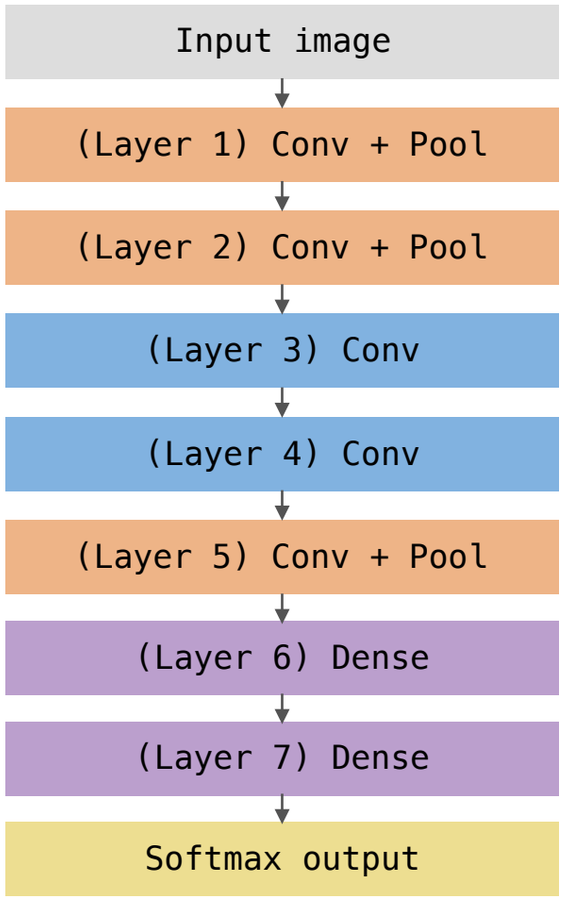
- On the contrary, the experiences from the past decade show that depth of a net is the key to great performance

© 2020 National University of Singapore.
All Rights Reserved.

The importance of depth

A study on Krizhevsky et al. model (2012)

- The model that won ILSVRC 2012
- 8 layers in total, 60 million parameters, 650,000 neurons
- Re-implementation gives 18.1% top-5 error



Source: Re-implementation by Rob Fergus

© 2020 National University of Singapore.
All Rights Reserved.

Top-5 error

How about top-1 error

Persian cat

Source: http://www.vetstreet.com/cats/persian#1_ugw20zmq

Top-1 error

```
Chihuahua (0.4)
Hyena (0.25)
Koala (0.15)
Persian cat (0.1)
Burmese cat (0.02)
```

Considered **incorrect**

Top-5 error

```
Chihuahua (0.4)
Hyena (0.25)
Koala (0.15)
Persian cat (0.1)
Burmese cat (0.02)
```

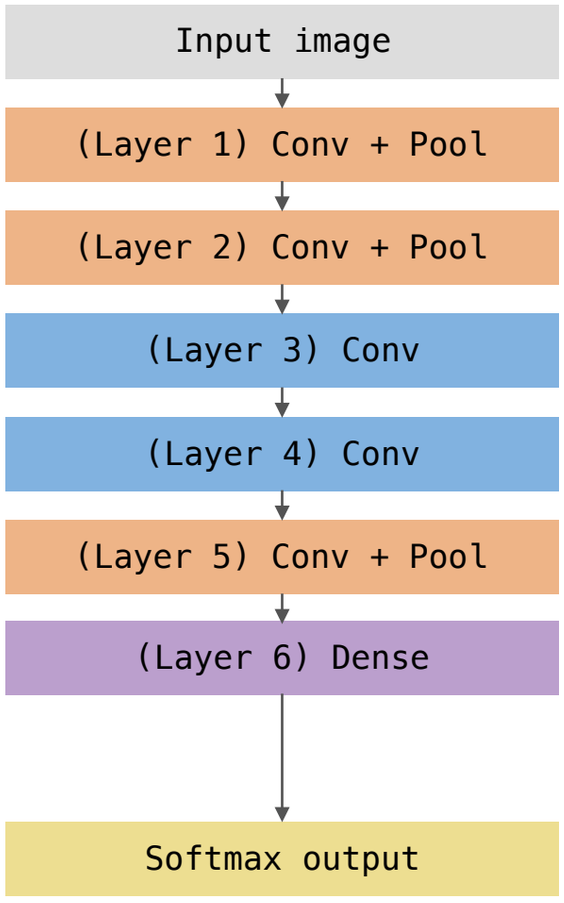
Considered **correct**

© 2020 National University of Singapore.
All Rights Reserved.

The importance of depth

A study on Krizhevsky et al. model (2012)

- Remove layer 7, the fully connected layer
- 16 million less parameters compared to the original model
- Only 1.1% drop in performance!



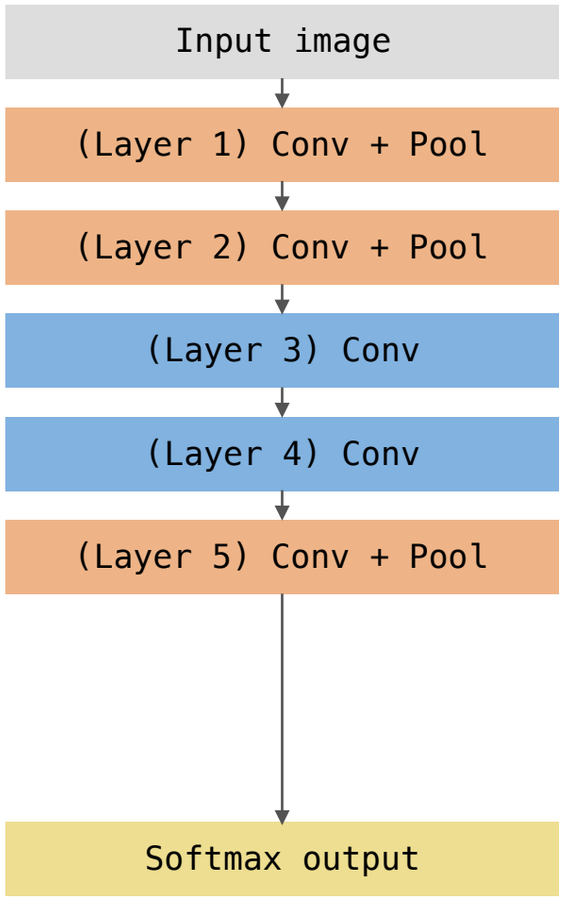
Source: Re-implementation by Rob Fergus

© 2020 National University of Singapore.
All Rights Reserved.

The importance of depth

A study on Krizhevsky et al. model (2012)

- Remove both layer 6 and layer 7, the two fully connected layers
- 50 million less parameters compared to the original model
- 5.7% drop in performance

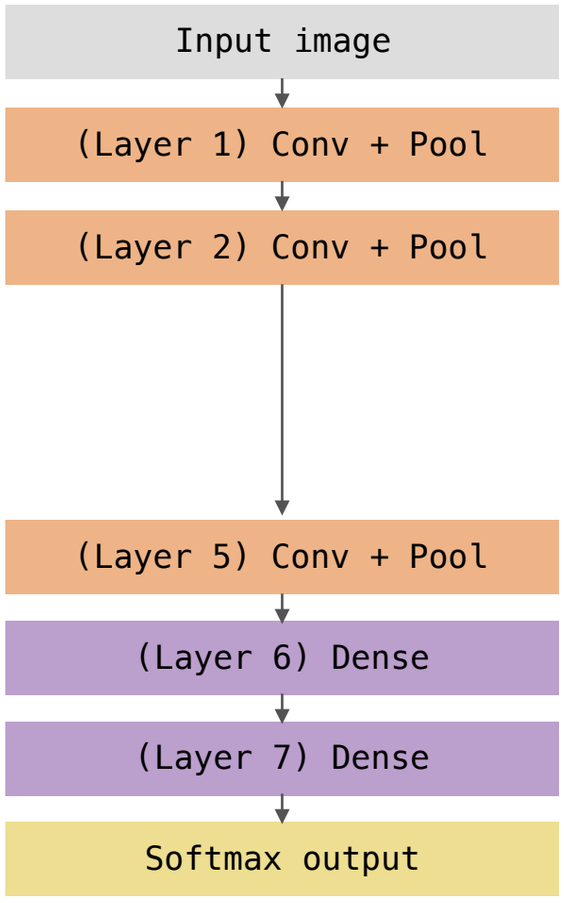


Source: Re-implementation by Rob Fergus

© 2020 National University of Singapore.
All Rights Reserved.

The importance of depth

A study on Krizhevsky et al. model (2012)



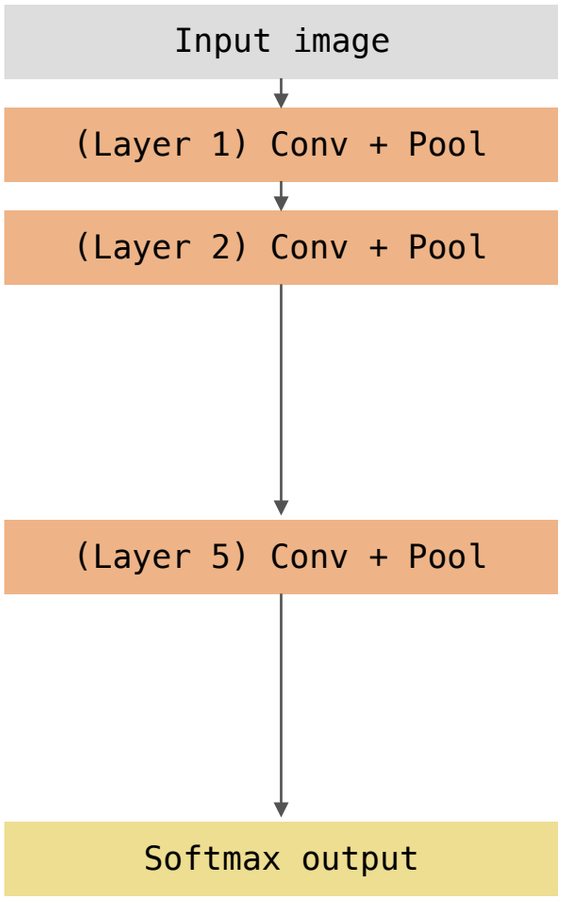
- Now try remove upper layers, the feature extractor. Remove layer 3 and 4
- 1 million less parameters compared to the original model
- 3.0% drop in performance

Source: Re-implementation by Rob Fergus

© 2020 National University of Singapore.
All Rights Reserved.

The importance of depth

A study on Krizhevsky et al. model (2012)



- Now try remove upper layers, the feature extractor. Remove layer 3, 4, 6, and 7
- Only 4 layers left
- **33.5%** drop in performance!
- Depth is the key

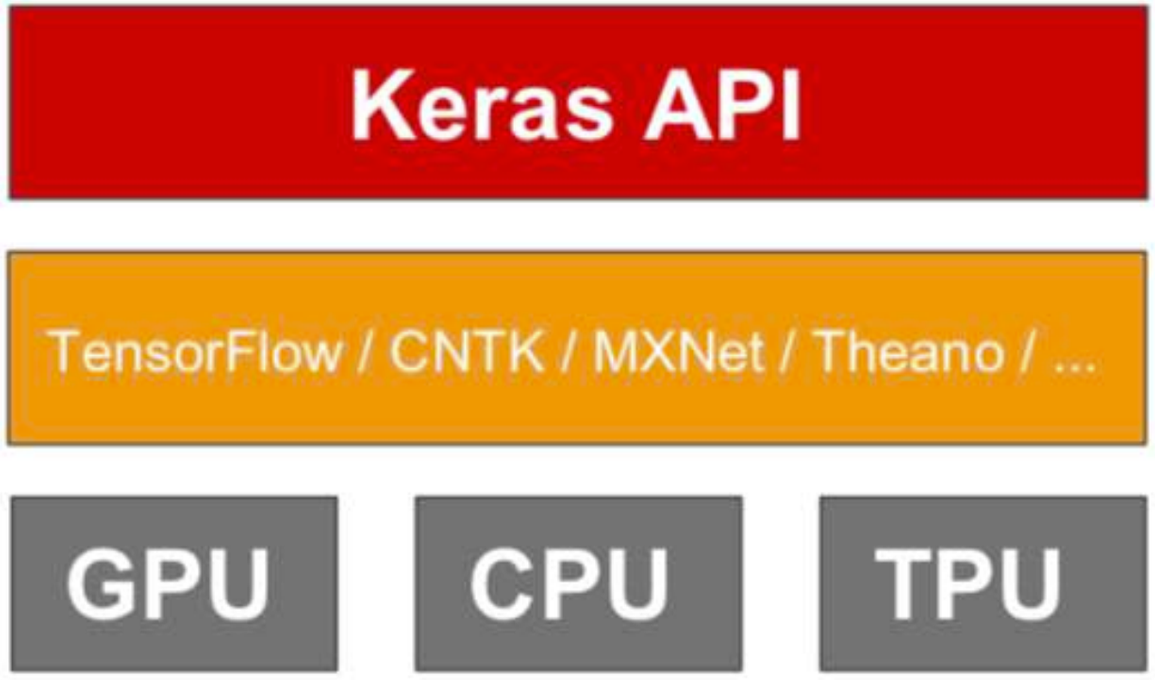
Source: Re-implementation by Rob Fergus

© 2020 National University of Singapore.
All Rights Reserved.

Keras

What is Keras

The basic architecture

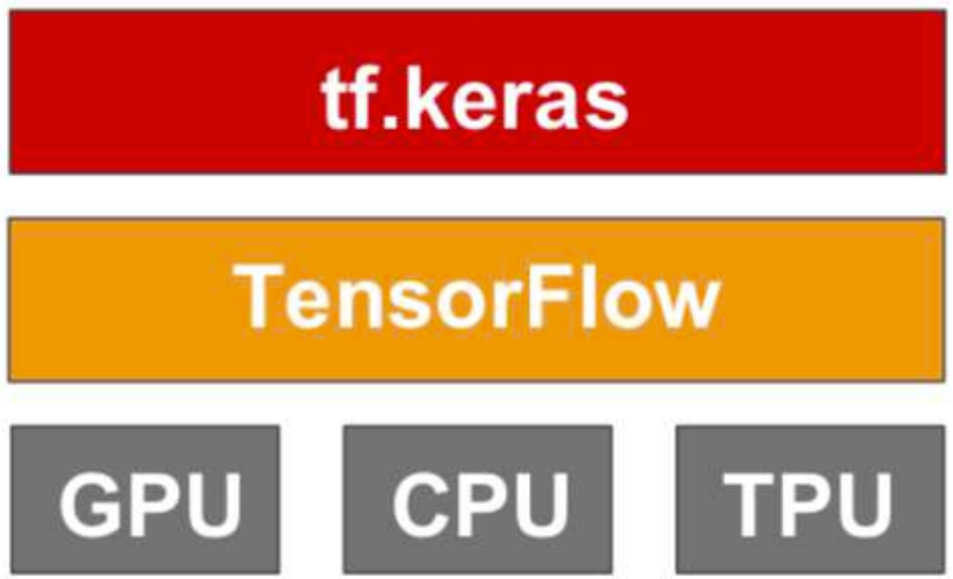


Source: Francois Chollet

© 2020 National University of Singapore.
All Rights Reserved.

Keras + Tensorflow

The official high-level API of Tensorflow



Source: Francois Chollet

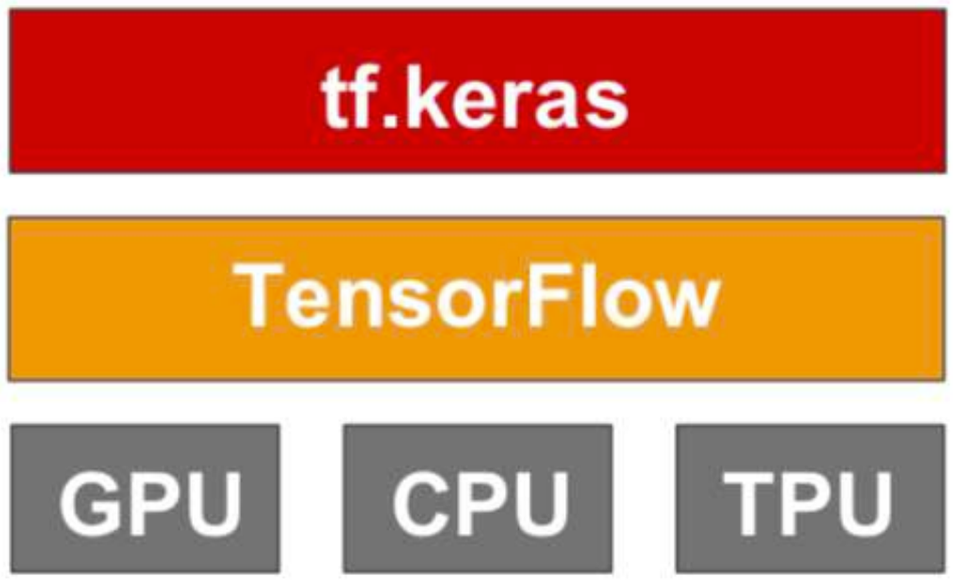
- Possible to build deep net solely using Tensorflow, but toooo many repetition
- Need to define every detail (weight, bias, initialization and etc.)
- Since Tensorflow v1.4, Keras is part of the core
- In Tensorflow v1.13, we have tensorflow.keras module
- From Tensorflow v2.0, Keras API is **THE preferred way** of building neural network

© 2020 National University of Singapore.
All Rights Reserved.

Keras + Tensorflow

The official high-level API of Tensorflow

- In tensorflow.keras, we have full Keras API
- Better optimized for TF
- Better integration with TF-specific features
- Estimator API, Eager execution, tf.Data



Source: Francois Chollet

© 2020 National University of Singapore.
All Rights Reserved.

Keras + Plaidml

For those who do not have NVIDIA GPU

- Currently deep learning mostly runs on Nvidia GPU, as Nvidia has CUDA, performed better than OpenCL
- Still, there is a need / request to run deep learning on non-Nvidia GPU
- With Plaidml as the backend (instead of Tensorflow or Theano), now it is possible
- Vertex.AI, the startup that build Plaidml, was bought by Intel in 2018

© 2020 National University of Singapore.
All Rights Reserved.

Keras

The user experience

- Follow best practices for reducing cognitive load
- Offer consistent and simple APIs
- Minimize number of user actions required for common use cases
- Provide clear and actional feedback on user error

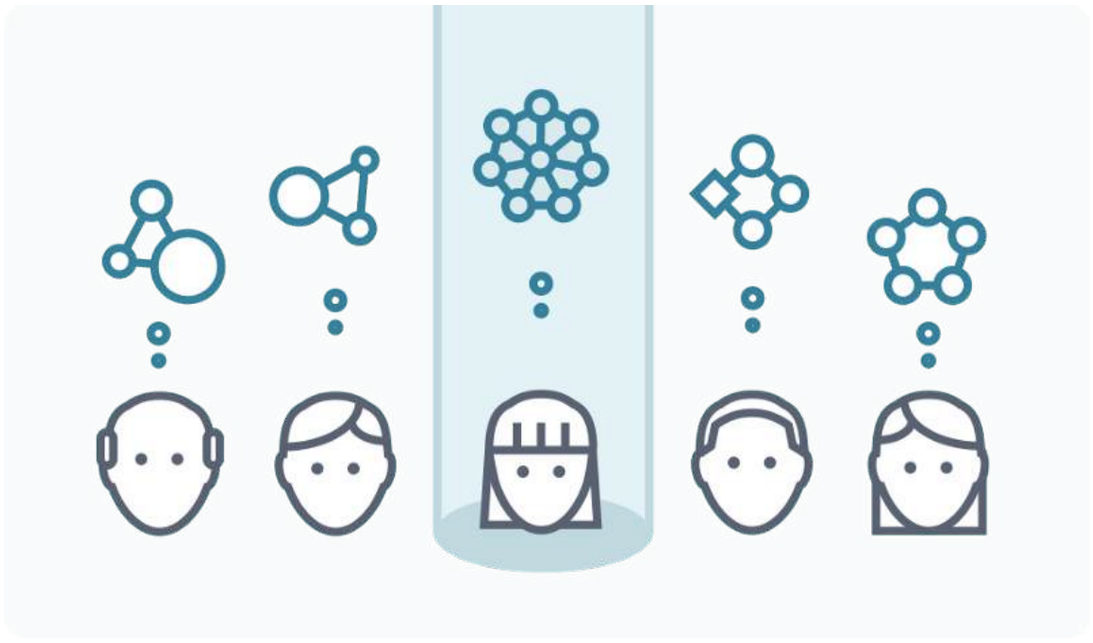
Source: Francois Chollet

© 2020 National University of Singapore.
All Rights Reserved.

Keras

A higher chance of success

- Since the APIs are easy to use, developers are more productive, have more time to try more ideas
- They key to win competitions and further explorations!
- Ease of use does not come at the cost of reduced flexibility; Keras easily integrates with Tensorflow with tensorflow.keras



Source: Francois Chollet
Image: https://medium.com/implodinggradients/should-you-kaggle-5b8dbdef442f

© 2020 National University of Singapore.
All Rights Reserved.

Keras

Largest array of options for productizing models

- TF-Serving
- Within web browser, with GPU acceleration (WebKeras, Keras.js, WebDNN ...)
- Android (TF, TF Lite), iPhone (native CoreML support)
- Raspberry Pi
- JVM

Source: Francois Chollet

© 2020 National University of Singapore.
All Rights Reserved.

Keras

Three API styles

- Sequential model
  Super simple
  Only for single-input, single-output sequential layer stacks
  Good for 70+% of use cases

- Functional API
- Works like playing Lego bricks
- Multi-input, multi-output, arbitrary static graph topologies
- Good for 95% of use cases

- Model subclassing
  Maximum flexibility
  Larger potential error surface

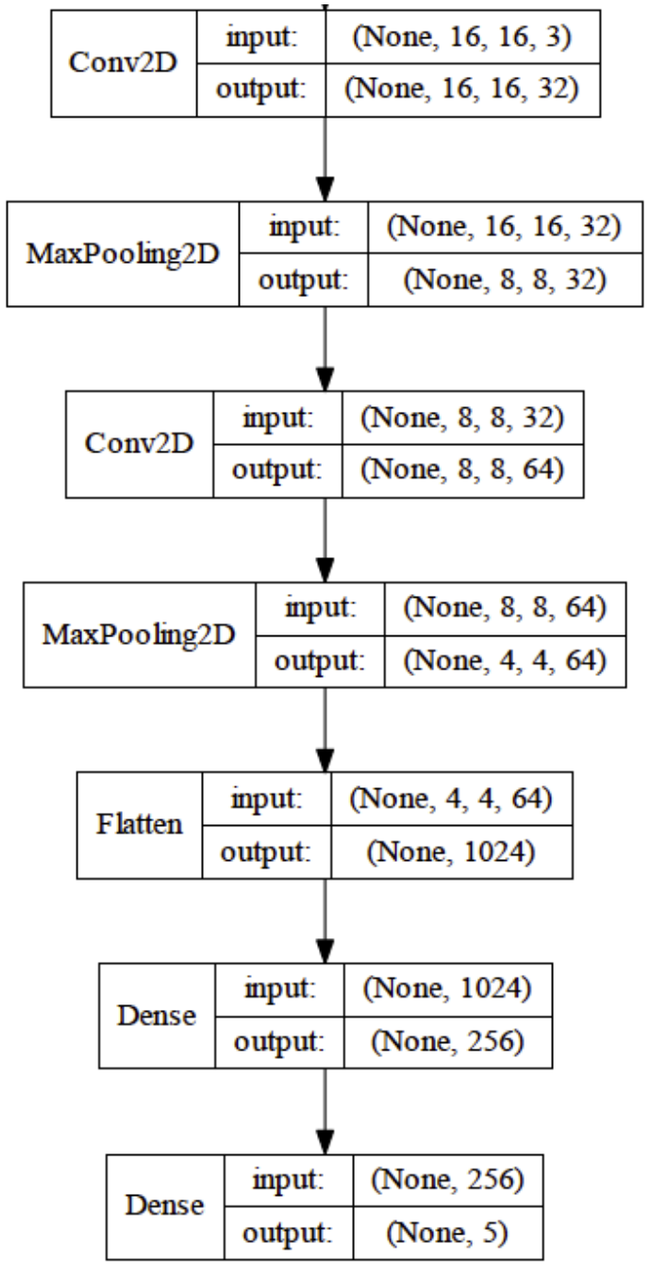
Source: Francois Chollet

© 2020 National University of Singapore.
All Rights Reserved.

Keras

Sequential model

- Adding layers by sequence (layer by layer)
- `input_shape` must be present in the first layer

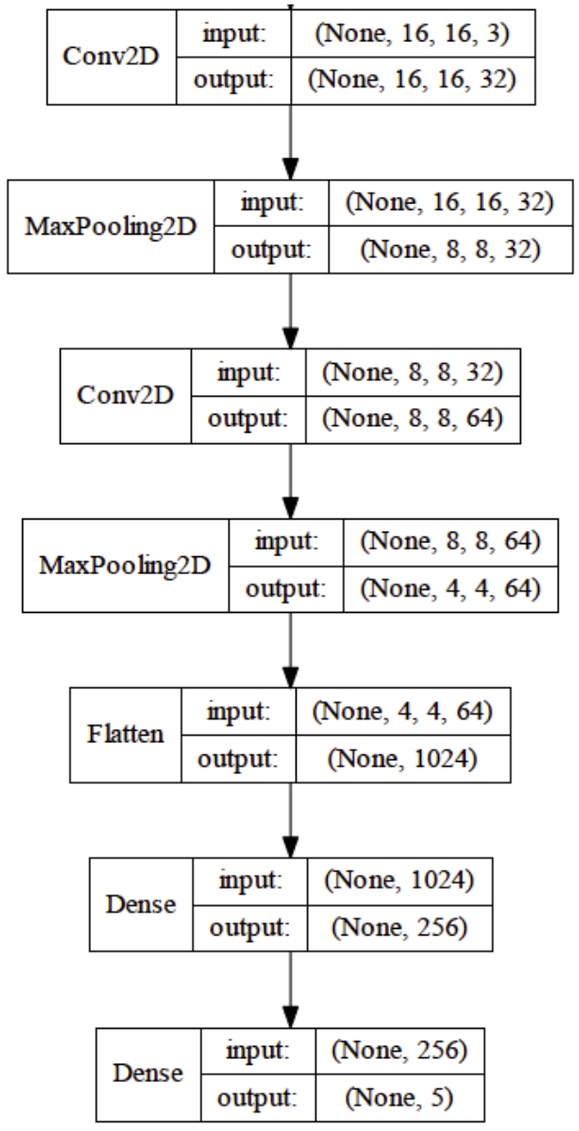


© 2020 National University of Singapore.
All Rights Reserved.

Keras

Sequential model

```
> def createSeqModel():
      model   = Sequential()
      model.add(Conv2D(32,(3,3),
                input_shape=(16,16,3),
                padding='same',
                activation='relu'))
      model.add(MaxPooling2D(pool_size=(2,2)))
      model.add(Conv2D(64,(3,3),
                padding='same',
                activation='relu'))
  model.add(MaxPooling2D(pool_size=(2,2)))

  model.add(Flatten())
  model.add(Dense(256,activation='relu'))
  model.add(Dense(5,activation='softmax'))

  model.compile(loss='categorical_crossentropy',
                optimizer='rmsprop',
                metrics=['accuracy'])
  return model
```



© 2020 National University of Singapore.
All Rights Reserved.

Keras

Sequential model

- Create the model and show the model summary

```
> modelSeq    = createSeqModel()
> modelSeq.summary()
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 16, 16, 32)        896
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 8, 8, 32)          0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 8, 8, 64)          18496
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 4, 4, 64)          0
_________________________________________________________________
flatten (Flatten)            (None, 1024)              0
_________________________________________________________________
dense (Dense)                (None, 256)               262400
_________________________________________________________________
dense_1 (Dense)              (None, 5)                 1285
=================================================================
Total params: 283,077
Trainable params: 283,077
Non-trainable params: 0
_________________________________________________________________
```


© 2020 National University of Singapore.
All Rights Reserved.

Keras

Sequential model

- The problem

  Not possible for multiple input
  Not possible for multiple output
  A single direction of flow of tensors, no branching, no merging
  Not possible to reuse layer; shared layer is not possible

```
> def createSeqModel():
      model   = Sequential()
      model.add(Conv2D(32,(3,3),
                input_shape=(16,16,3),
                padding='same',
                activation='relu'))
  model.add(MaxPooling2D(pool_size=(2,2)))
  model.add(Conv2D(64,(3,3),
                padding='same',
                activation='relu'))
  model.add(MaxPooling2D(pool_size=(2,2)))

  model.add(Flatten())
  model.add(Dense(256,activation='relu'))
  model.add(Dense(5,activation='softmax'))

  model.compile(loss='categorical_crossentropy',
                optimizer='rmsprop',
                metrics=['accuracy'])
  return model
```


© 2020 National University of Singapore.
All Rights Reserved.

Multiple inputs?

An video and a question

- "What is the person doing?"
  "Tidying"

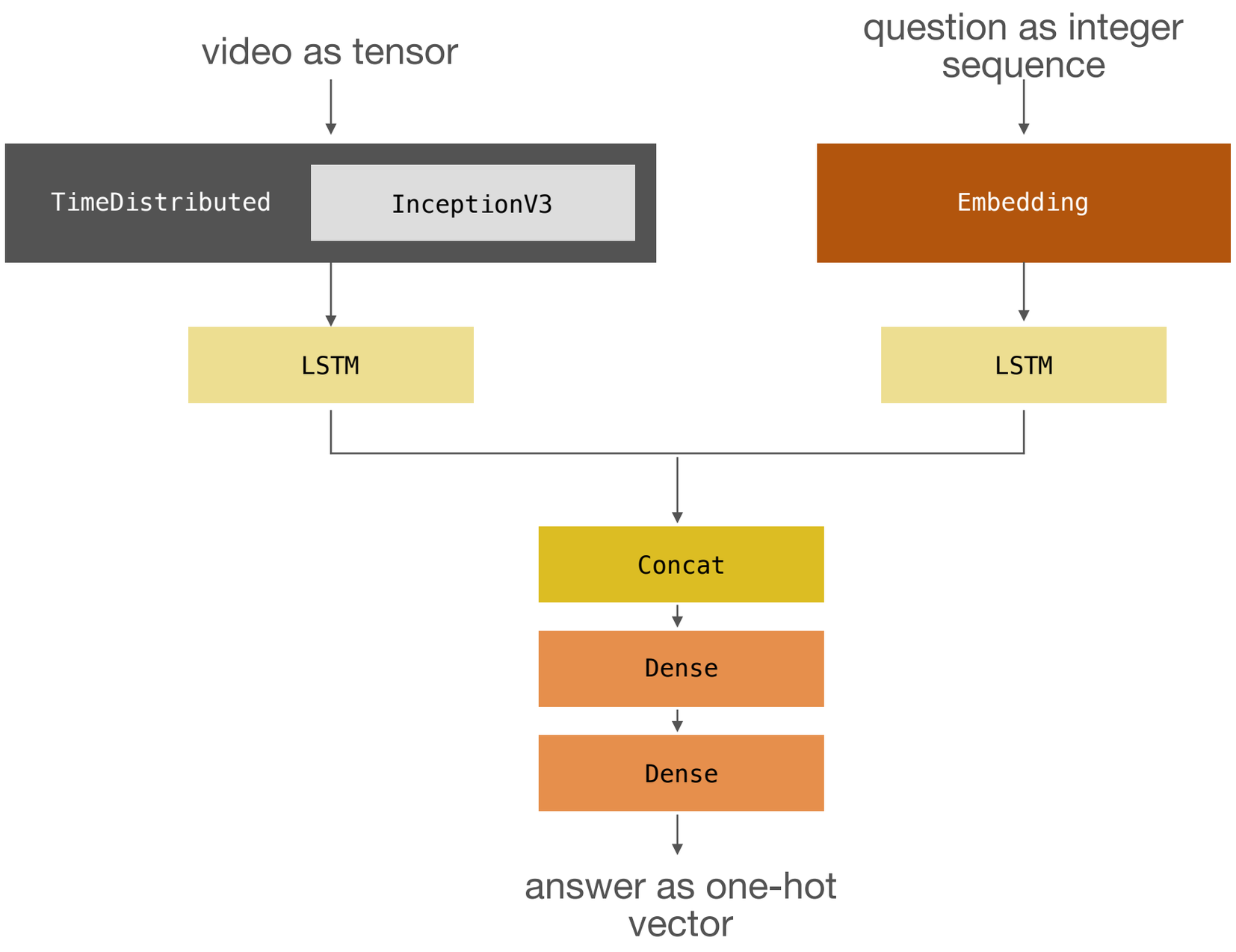
- "Who is the person?"
  "Kondo"

Source: people.com

© 2020 National University of Singapore.
All Rights Reserved.

Multiple inputs

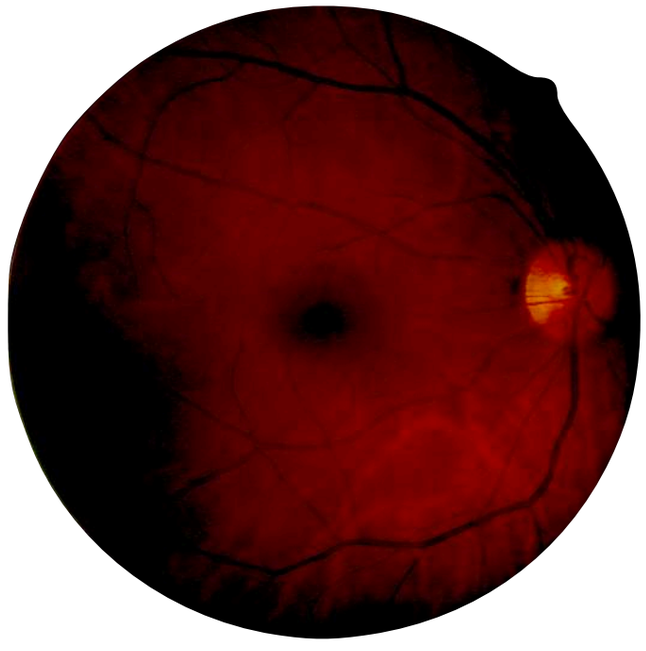
An video and a question



© 2020 National University of Singapore.
All Rights Reserved.

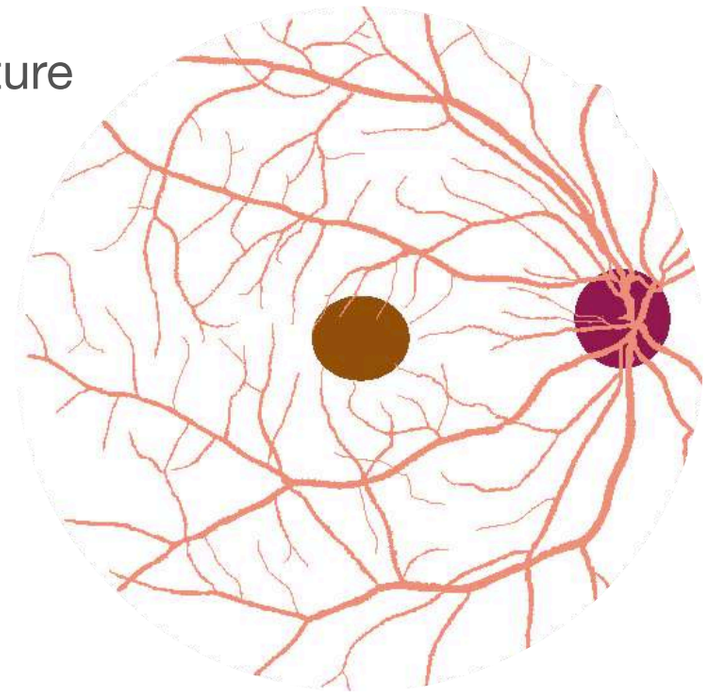
Multiple outputs?

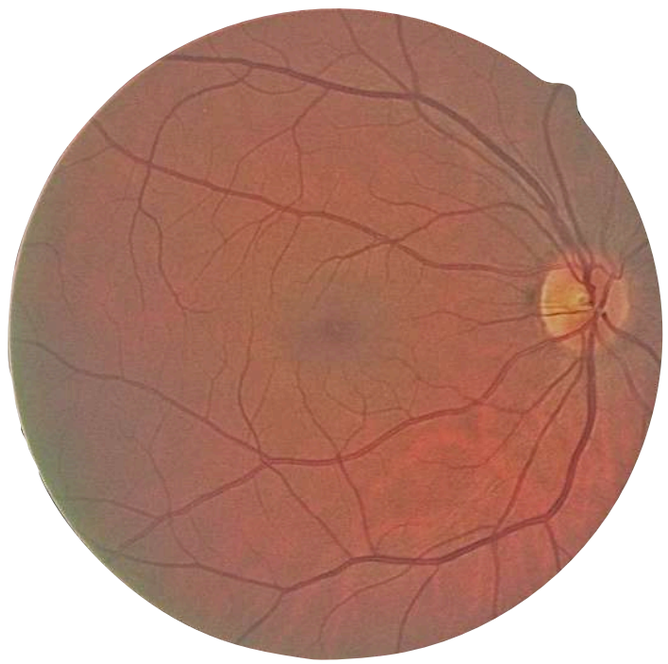
An adjusted image and a segmentation



input image

anatomy feature segmented



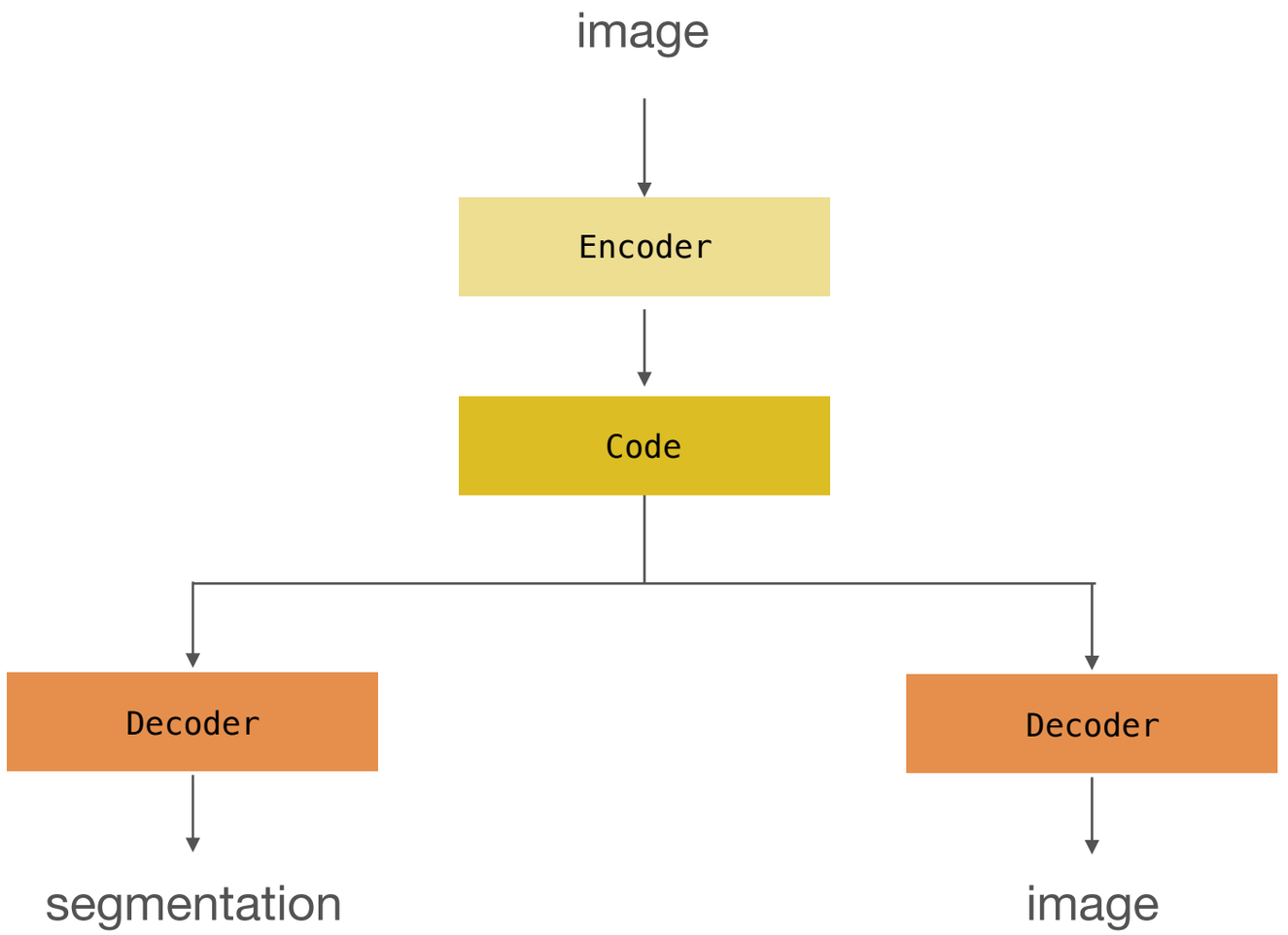


adjusted output image

© 2020 National University of Singapore.
All Rights Reserved.

Multiple outputs

An adjusted image and a segmentation



© 2020 National University of Singapore.
All Rights Reserved.

Shared layers?

Person re-identification

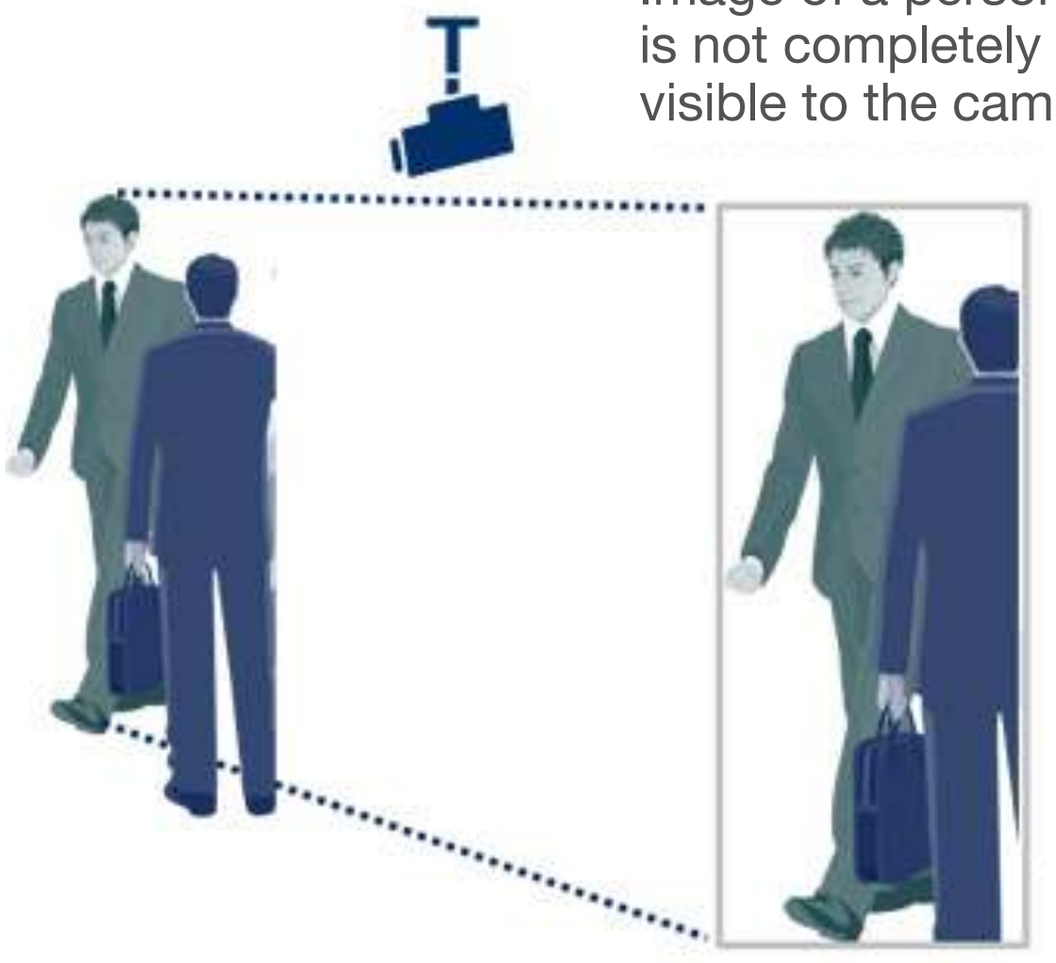


Image of a person that is not completely visible to the camera

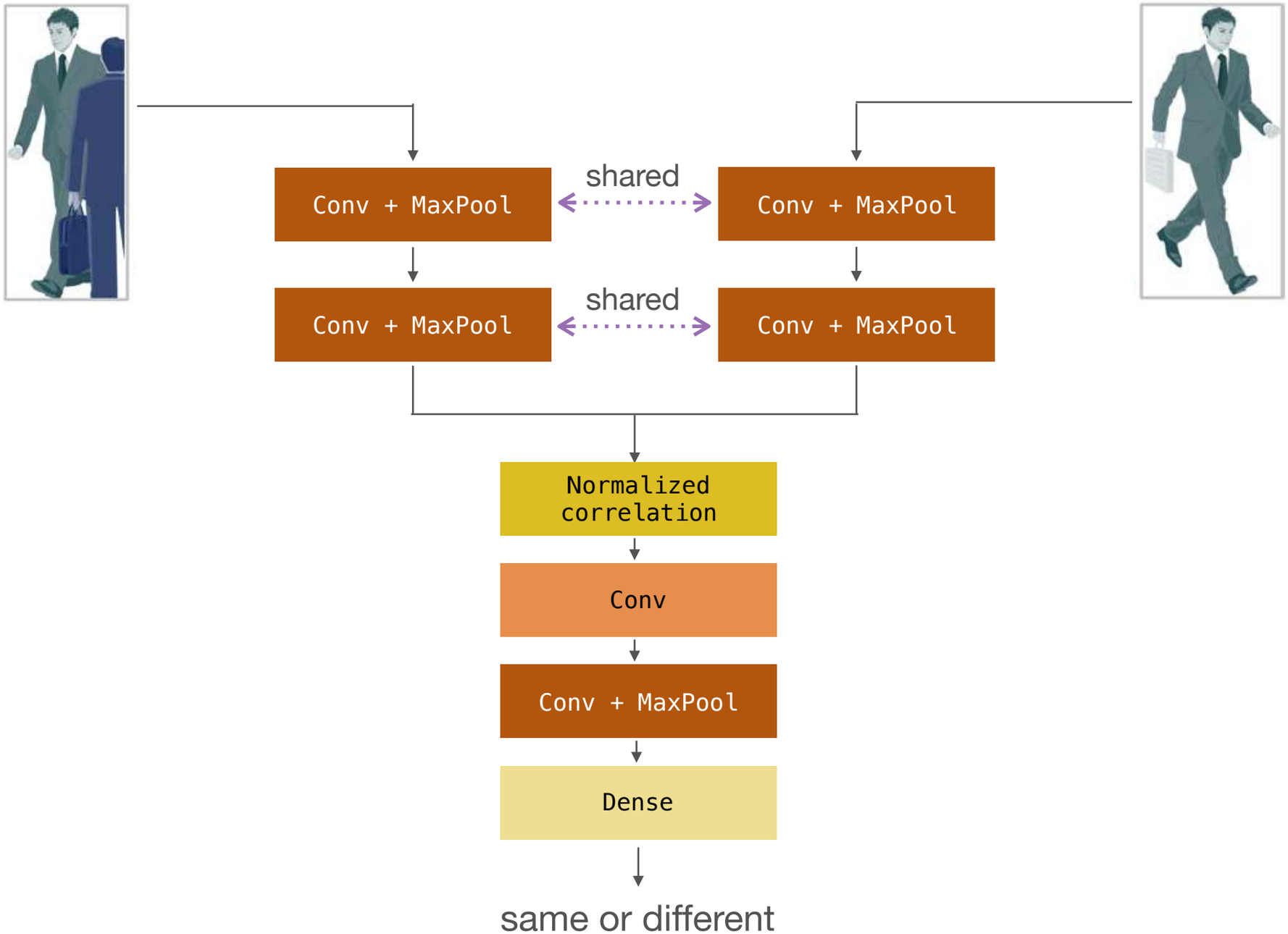
Image of the person to be checked

Source: https://yellrobot.com/nec-person-reidentification-technology-can-identify-body-shape/

© 2020 National University of Singapore.
All Rights Reserved.

Shared layers

Person re-identification

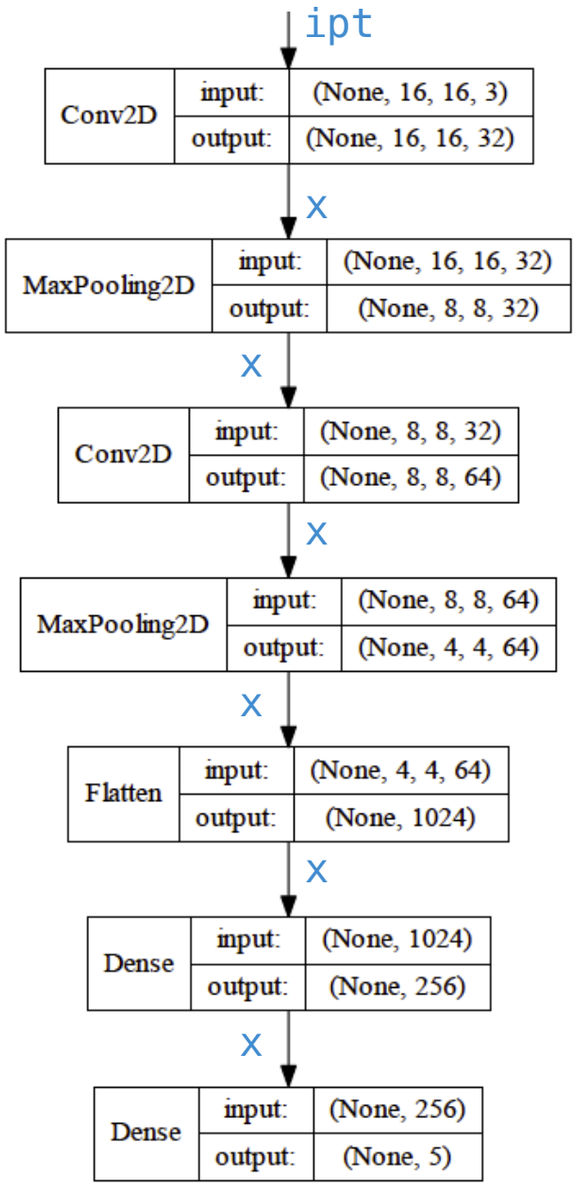


© 2020 National University of Singapore.
All Rights Reserved.

Keras

functional APIs

```
> def createFuncModel():
    ipt     = Input(shape=(16,16,3))
    x       = Conv2D(32,(3,3),
                     padding='same',
                     activation='relu')(ipt)
    x       = MaxPooling2D(pool_size=(2,2))(x)
    x       = Conv2D(64,(3,3),
                     padding='same',
                     activation='relu')(x)
    x       = MaxPooling2D(pool_size=(2,2))(x)
    x       = Flatten()(x)
    x       = Dense(256,activation='relu')(x)
    x       = Dense(5,activation='relu')(x)

    model   = Model(inputs=ipt,outputs=x)

    model.compile(loss='categorical_crossentropy',
                  optimizer='rmsprop',
                  metrics=['accuracy'])
    return model
```



© 2020 National University of Singapore.
All Rights Reserved.

Keras

Comparison

functional APIs

```
> def createFuncModel():
    ipt     = Input(shape=(16,16,3))
    x       = Conv2D(32,(3,3),
                     padding='same',
                     activation='relu')(ipt)
    x       = MaxPooling2D(pool_size=(2,2))(x)
    x       = Conv2D(64,(3,3),
                     padding='same',
                     activation='relu')(x)
    x       = MaxPooling2D(pool_size=(2,2))(x)
    x       = Flatten()(x)
    x       = Dense(256,activation='relu')(x)
    x       = Dense(5,activation='relu')(x)

    model   = Model(inputs=ipt,outputs=x)

    model.compile(loss='categorical_crossentropy',
                  optimizer='rmsprop',
                  metrics=['accuracy'])
    return model
```

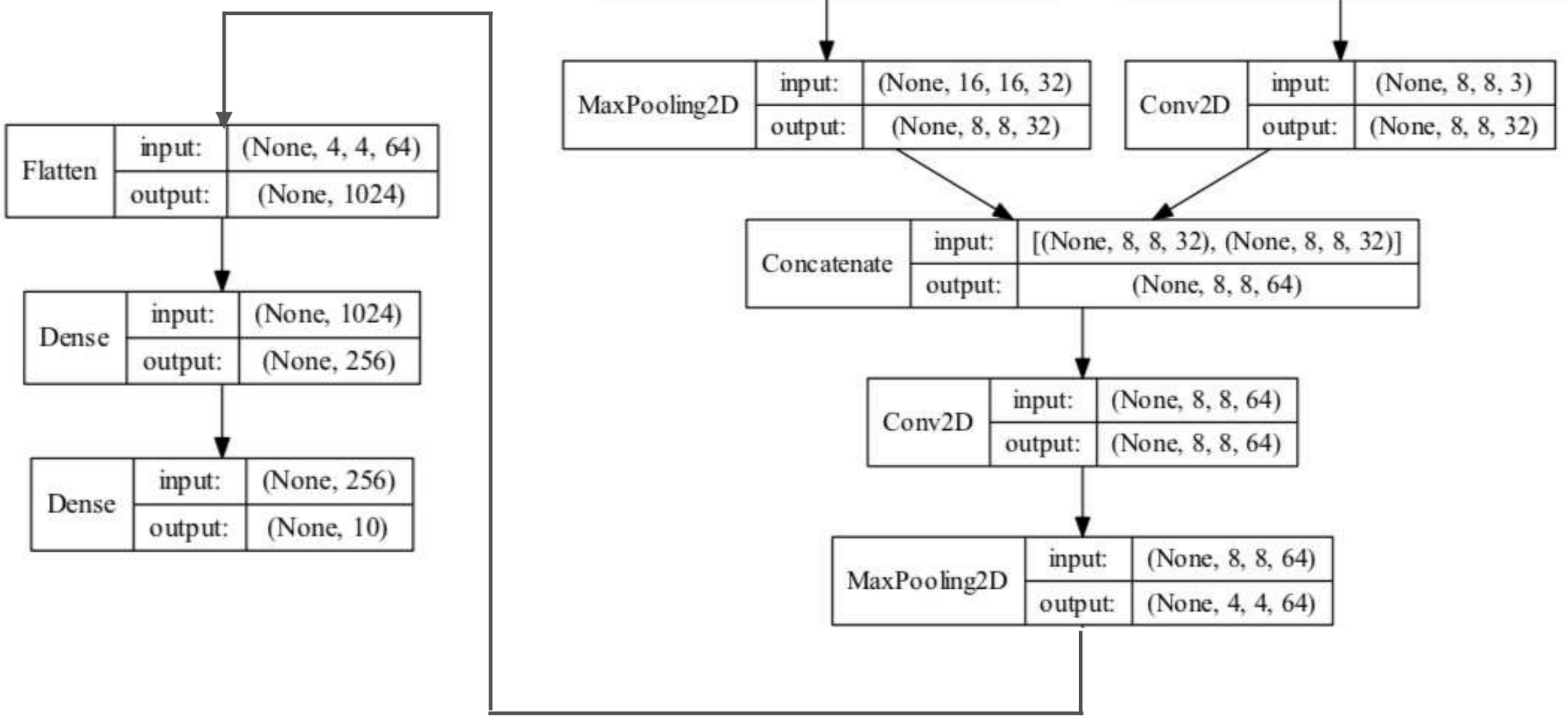
Sequential model

```
> def createSeqModel():
    model   = Sequential()
    model.add(Conv2D(32,(3,3),
                     input_shape=(16,16,3),
                     padding='same',
                     activation='relu'))
    model.add(MaxPooling2D(pool_size=(2,2)))
    model.add(Conv2D(64,(3,3),
                     padding='same',
                     activation='relu'))
    model.add(MaxPooling2D(pool_size=(2,2)))

    model.add(Flatten())
    model.add(Dense(256,activation='relu'))
    model.add(Dense(5,activation='softmax'))

    model.compile(loss='categorical_crossentropy',
                  optimizer='rmsprop',
                  metrics=['accuracy'])
    return model
```


© 2020 National University of Singapore.
All Rights Reserved.

'Y' shape architecture

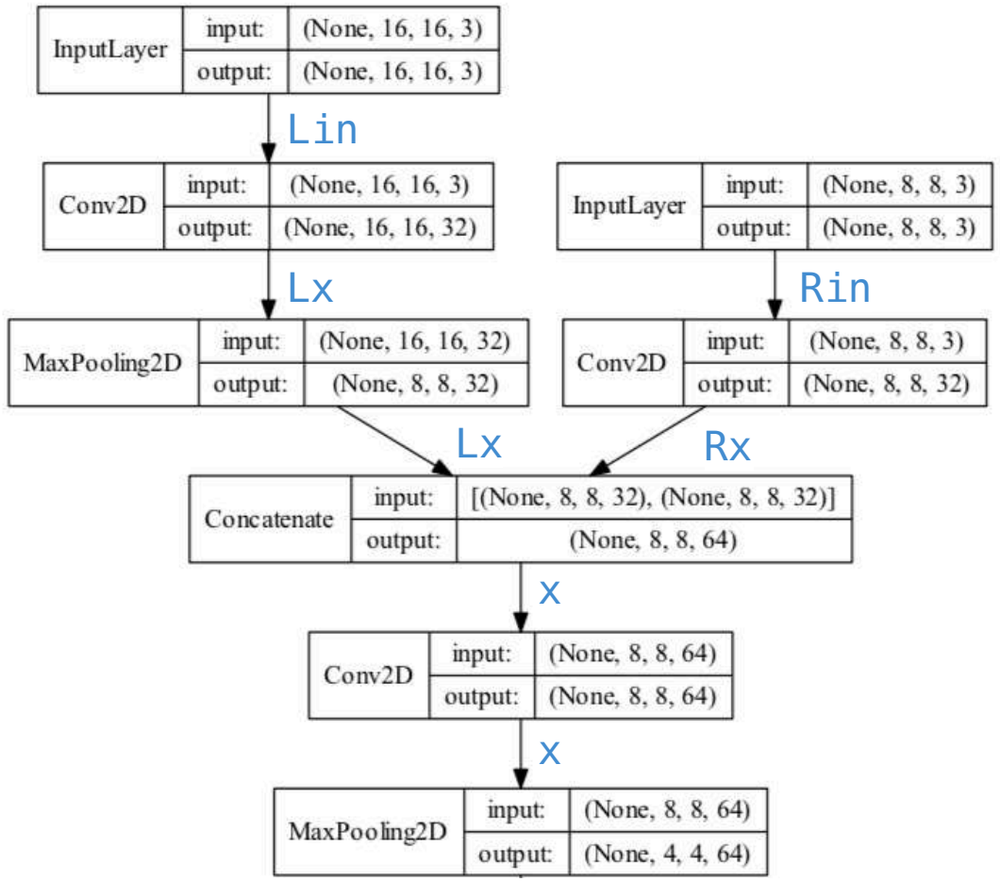
how to create?



© 2020 National University of Singapore.
All Rights Reserved.

'Y' shape architecture

the code

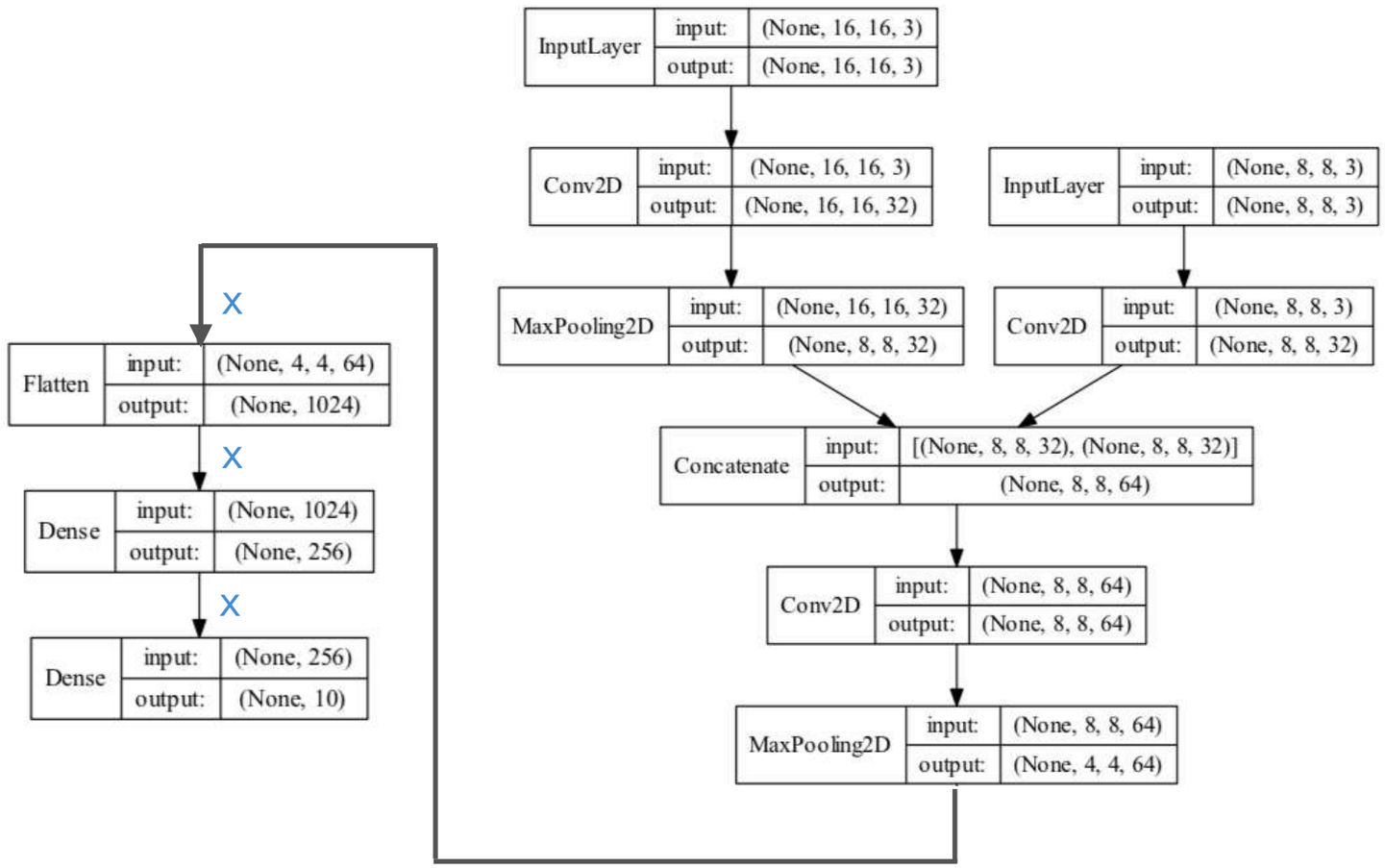



```
> def createDualInputModel():
    Lin     = Input(shape=(16,16,3))
    Lx      = Conv2D(32,(3,3),padding='same',activation='relu')(Lin)
    Lx      = MaxPooling2D(pool_size=(2,2))(Lx)

    Rin     = Input(shape=(8,8,3))
    Rx      = Conv2D(32,(3,3),padding='same',activation='relu')(Rin)

    x       = concatenate([Lx,Rx],axis=-1)
    x       = Conv2D(64,(3,3),padding='same',activation='relu')(x)
    x       = MaxPooling2D(pool_size=(2,2))(x)
    ...
```


© 2020 National University of Singapore.
All Rights Reserved.

'Y' shape architecture

the code




```
> def createDualInputModel():
    ....
    x       = Flatten()(x)
    x       = Dense(128,activation='relu')(x)
    x       = Dense(3,activation='softmax')(x)

    model   = Model(inputs=[Lin,Rin],outputs=x)
    model.compile(loss='categorical_crossentropy',
                  optimizer='rmsprop',
                  metrics=['accuracy'])
    return model
```


© 2020 National University of Singapore.
All Rights Reserved.

'Y' shape architecture

Putting all together ....

- Remember to include the below before the function

```
from tensorflow.keras.layers import concatenate
```

```
> def createDualInputModel():
    Lin     = Input(shape=(16,16,3))
    Lx      = Conv2D(32,(3,3),padding='same',activation='relu')(Lin)
    Lx      = MaxPooling2D(pool_size=(2,2))(Lx)

    Rin     = Input(shape=(8,8,3))
    Rx      = Conv2D(32,(3,3),padding='same',activation='relu')(Rin)

    x       = concatenate([Lx,Rx],axis=-1)
    x       = Conv2D(64,(3,3),padding='same',activation='relu')(x)
    x       = MaxPooling2D(pool_size=(2,2))(x)

    x       = Flatten()(x)
    x       = Dense(128,activation='relu')(x)
    x       = Dense(3,activation='softmax')(x)

    model   = Model(inputs=[Lin,Rin],outputs=x)
    model.compile(loss='categorical_crossentropy',
                  optimizer='rmsprop',
                  metrics=['accuracy'])
    return model
```


© 2020 National University of Singapore.
All Rights Reserved.

Multiple inputs

the code

- Create the model and show the model summary

```
> modelDual    = createDualInputModel()
> modelDual.summary()
```

```
_________________________________________________________________________________________________
Layer (type)                 Output Shape              Param #     Connected to
=================================================================================================
input_1 (InputLayer)         (None, 16, 16, 3)         0
_________________________________________________________________________________________________
conv2d (Conv2D)              (None, 16, 16, 32)        896         input_1[0][0]
_________________________________________________________________________________________________
input_2 (InputLayer)         (None, 8, 8, 3)           0
_________________________________________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 8, 8, 32)          0           conv2d[0][0]
_________________________________________________________________________________________________
conv2d_1 (Conv2D)            (None, 8, 8, 32)          896         input_2[0][0]
_________________________________________________________________________________________________
concatenate (Concatenate)    (None, 8, 8, 64)          0           max_pooling2d[0][0]
                                                                   conv2d_1[0][0]
_________________________________________________________________________________________________
conv2d_2 (Conv2D)            (None, 8, 8, 64)          36928       concatenate[0][0]
_________________________________________________________________________________________________
max_pooling2d_1 (MaxPooling2D) (None, 4, 4, 64)        0           conv2d_2[0][0]
_________________________________________________________________________________________________
flatten (Flatten)            (None, 1024)              0           max_pooling2d_1[0][0]
_________________________________________________________________________________________________
dense (Dense)                (None, 128)               131200      flatten[0][0]
_________________________________________________________________________________________________
dense_1 (Dense)              (None, 3)                 387         dense[0][0]
=================================================================================================
Total params: 170,307
Trainable params: 170,307
Non-trainable params: 0
_________________________________________________________________________________________________
```


© 2020 National University of Singapore.
All Rights Reserved.

Multiple inputs

training

- Assume RDat and LDat are the training input, TLbl is the label for the training

- Furthermore, vRDat and vLDat is the validation input, and vLbl is the label

- The training is done by

```
> model.fit([RDat,LDat],
            TLbl,
            validation_data=([vRDat,vLDat], vLbl),
            epochs=100,
            batch_size=128,
            shuffle=True,
            callbacks=callbacks_list)
```


© 2020 National University of Singapore.
All Rights Reserved.

Multiple inputs

Getting the model plot

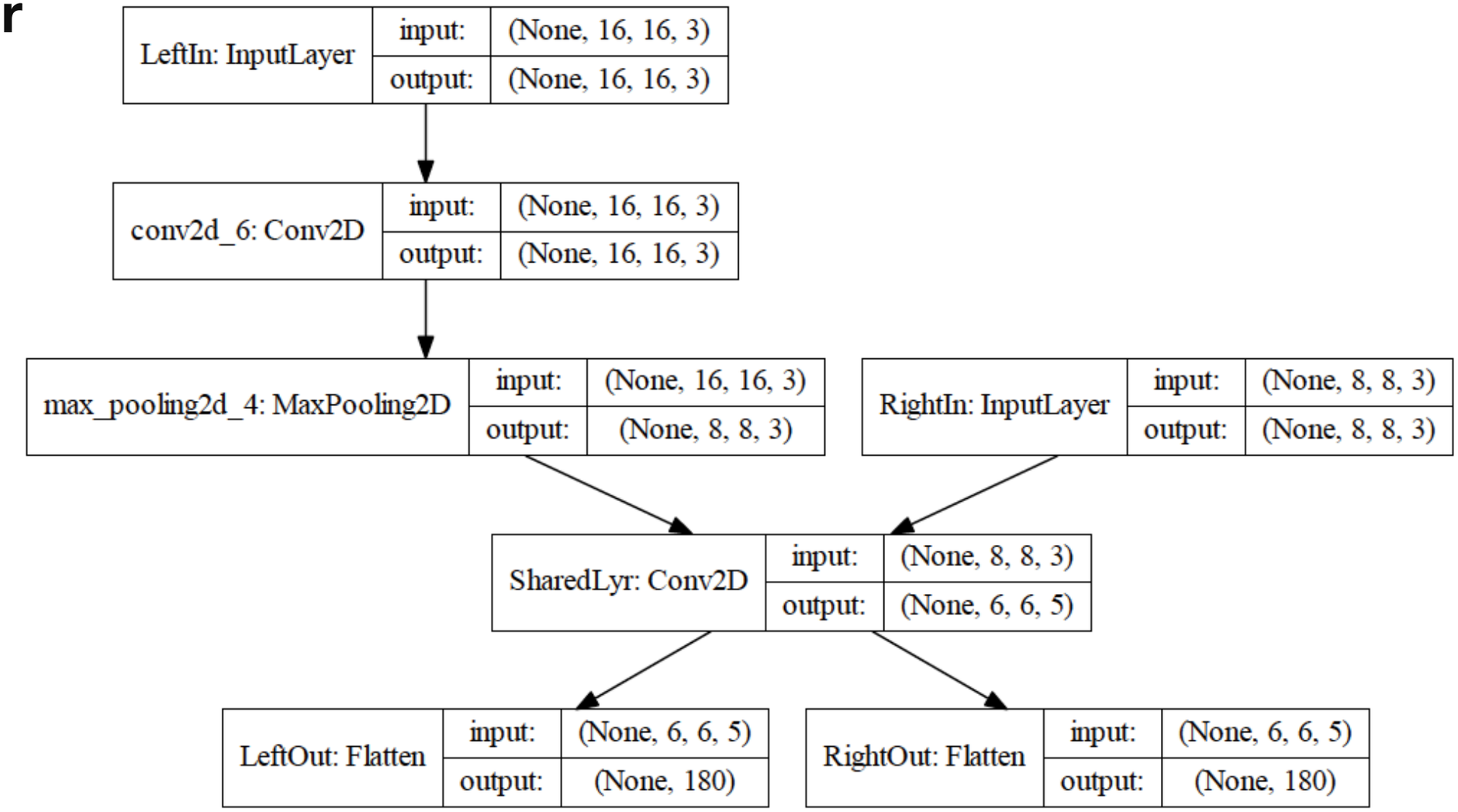
- To plot model, it requires pydot and graphviz
- On Mac, there is a need for additional installation for graphviz

```
> from tensorflow.keras.utils import plot_model

> plot_model(modelDual,
             to_file='Dual_model.pdf',
             show_shapes=True,
             show_layer_names=False,
             rankdir='TB')
```


© 2020 National University of Singapore.
All Rights Reserved.

Shared layer

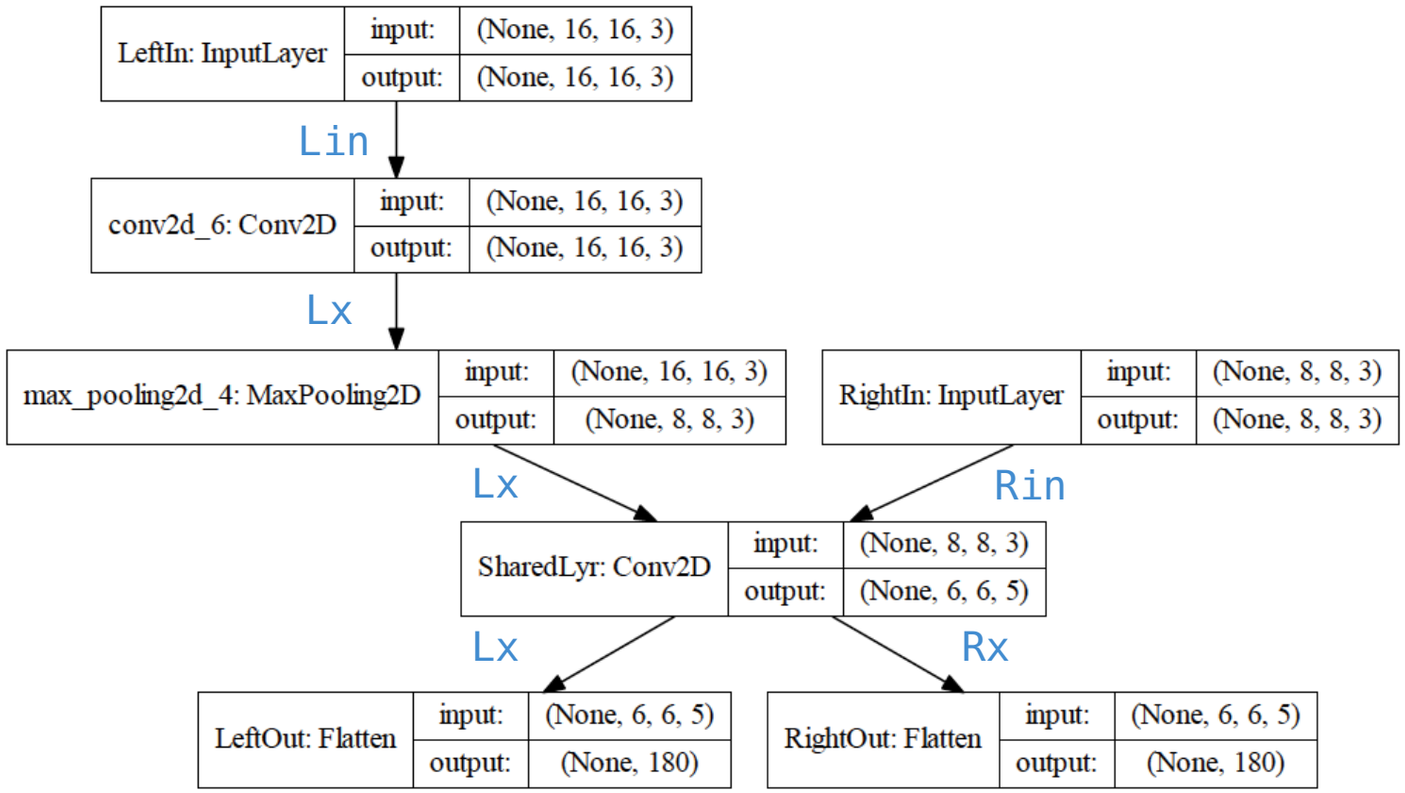
how to create?



© 2020 National University of Singapore.
All Rights Reserved.

Shared layer

the code




```
> def createSharedModel():
    shared  = Conv2D(5,(3,3),activation='relu',name='SharedLyr')

    Lin     = Input(shape=(16,16,3),name='LeftIn')
    Lx      = Conv2D(3,(3,3),padding='same',activation='relu')(Lin)
    Lx      = MaxPooling2D(pool_size=(2,2))(Lx)
    Lx      = shared(Lx)
    Lx      = Flatten(name='LeftOut')(Lx)

    Rin     = Input(shape=(8,8,3),name='RightIn')
    Rx      = shared(Rx)
    Rx      = Flatten(name='RightOut')(Rx)
    model   = Model(inputs=[Lin,Rin],outputs=[Lx,Rx])
    ...
```


© 2020 National University of Singapore.
All Rights Reserved.

Shared layer

the code

- For model with more than one output, pass in a dictionary that specifies the loss function for each output

```
> def createSharedModel():
    shared  = Conv2D(5,(3,3),activation='relu',name='SharedLyr')

    Lin     = Input(shape=(16,16,3),name='LeftIn')
    Lx      = Conv2D(3,(3,3),padding='same',activation='relu')(Lin)
    Lx      = MaxPooling2D(pool_size=(2,2))(Lx)
    Lx      = shared(Lx)
    Lx      = Flatten(name='LeftOut')(Lx)

    Rin     = Input(shape=(8,8,3),name='RightIn')
    Rx      = shared(Rx)
    Rx      = Flatten(name='RightOut')(Rx)

    model   = Model(inputs=[Lin,Rin],outputs=[Lx,Rx])
    model.compile(loss={'LeftOut':'categorical_crossentropy',
                        'RightOut':'mean_squared_error'},
                  optimizer='rmsprop',
                  metrics=['accuracy'])
    return model
```


© 2020 National University of Singapore.
All Rights Reserved.

Keras

Merge layer

- Other possible layers to fuse multiple flows of tensors in Keras

```
tensorflow.keras.layers.add
tensorflow.keras.layers.subtract
tensorflow.keras.layers.multiply
tensorflow.keras.layers.average
tensorflow.keras.layers.maximum
tensorflow.keras.layers.minimum
```

- For the above, the tensor inputs to the layer should have the same size

- Good enough for most use cases, otherwise write your own lambda layer to perform the necessary task

© 2020 National University of Singapore.
All Rights Reserved.

Keras
Multiple GPU

- Training using multiple GPUs is easy in Keras
- In the example, each GPU will process 64 samples in a single batch
- Why a need for multiple GPUs?

```
> from tensorflow.keras.utils import multi_gpu_model

> parallel    = multi_gpu_model(model, gpus=4)
> parallel.compile(loss='categorical_crossentropy',
                   optimizer='rmsprop')

> parallel.fit(x, y, epochs=50, batch_size=256)
```


© 2020 National University of Singapore.
All Rights Reserved.