DEEP LEARNING WITH KERAS

IMAGE RECOGNITION

Themistoklis Diamantopoulos

Image Recognition

- Cats vs Dogs
- Binary Classification problem

Source: https://www.kaggle.com/c/dogs-vs-cats/data

Feature Representation

- 150 x 150 x 3
- 3 channels for RGB

Blue channel

Green channel

Red channel

171	200	19	6	...	26	
24	56	230	1	...	8	89
120	67	89	107	...	13	18
12	216	145	26	...	181	81
0	16	4	45	...	44	56
0	78	90	167	...	25	...
...	...	...	...	...	...	12
12	67	82	141	...	12	

8	
71	
...	
7	

Source: http://www.subsubroutine.com/sub-subroutine/2016/9/30/cats-and-dogs-and-convolutional-neural-networks

Data Augmentation

- What to do when data are few?
- Augment them!

Source: https://blog.keras.io/building-powerful-image-classification-models-using-very-little-data.html

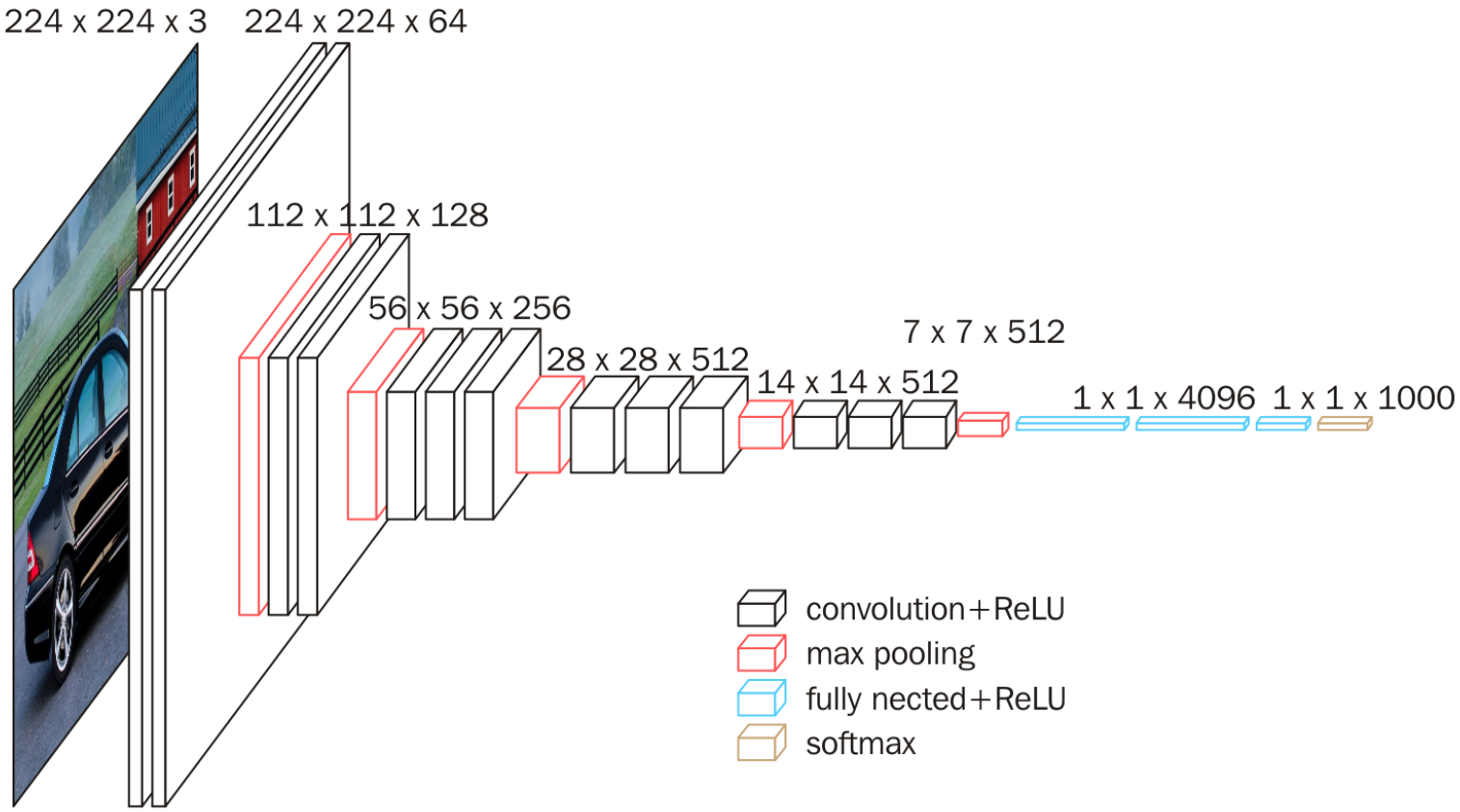
Solution using CNN

- 3-layer convolutional
- 3-layer dense

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 148, 148, 32)	896
activation_1 (Activation)	(None, 148, 148, 32)	0
max_pooling2d_1 (MaxPooling2	(None, 74, 74, 32)	0
conv2d_2 (Conv2D)	(None, 72, 72, 32)	9248
activation_2 (Activation)	(None, 72, 72, 32)	0
max_pooling2d_2 (MaxPooling2	(None, 36, 36, 32)	0
conv2d_3 (Conv2D)	(None, 34, 34, 64)	18496
activation_3 (Activation)	(None, 34, 34, 64)	0
max_pooling2d_3 (MaxPooling2	(None, 17, 17, 64)	0
flatten_1 (Flatten)	(None, 18496)	0
dense_1 (Dense)	(None, 64)	1183808
activation_4 (Activation)	(None, 64)	0
dropout_1 (Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 1)	65
activation_5 (Activation)	(None, 1)	0

Use pretrained model

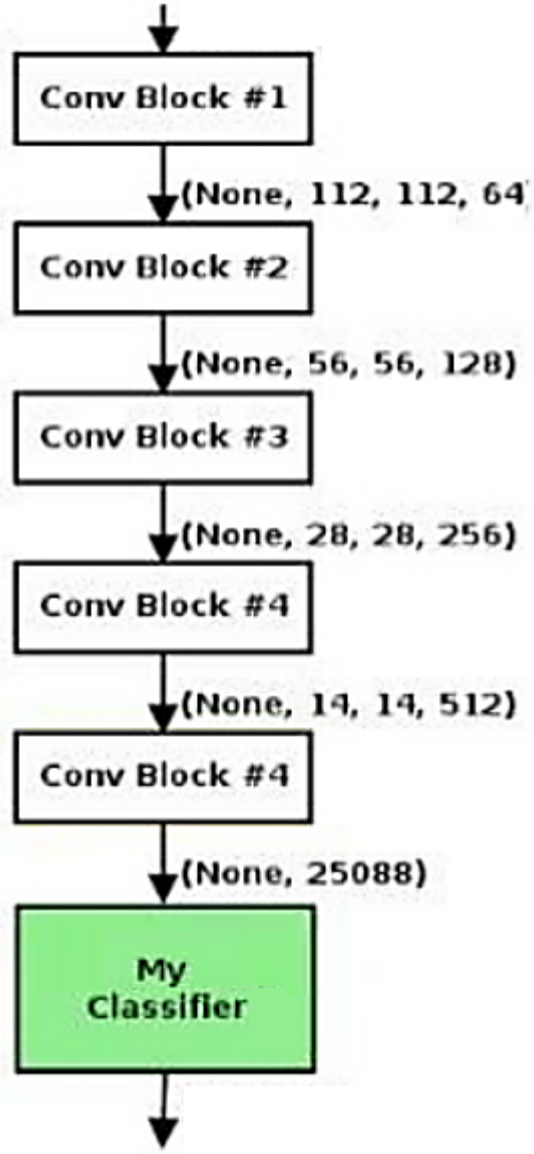
- VGG16 architecture



Source: https://www.researchgate.net/figure/VGG16-architecture-16_fig2_321829624

Solution using VGG16

- Pretrained on Imagenet
- Extract bottleneck features
- Attach own classifier at the bottom (a fully connected MLP)



Layer (type)	Output Shape	Param #
flatten_1 (Flatten)	(None, 8192)	0
dense_1 (Dense)	(None, 256)	2097408
dropout_1 (Dropout)	(None, 256)	0
dense_2 (Dense)	(None, 1)	257

Source: https://www.slideshare.net/LalitJain29/object-classification-using-cnn-vgg16-model-keras-and-tensorflow