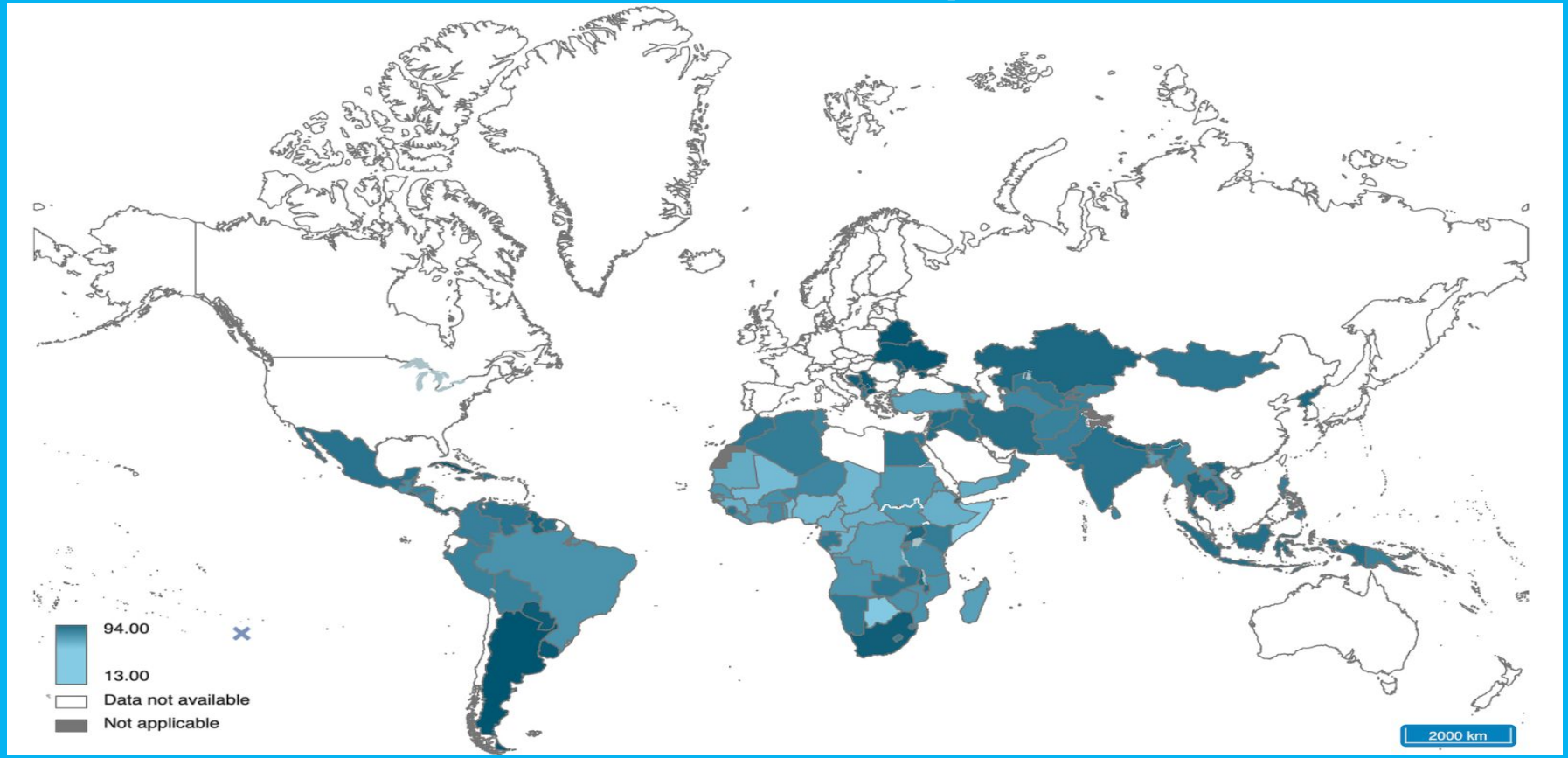
Diagnosing Pneumonia with AI

Children aged under 5 years with pneumonia symptoms taken to a healthcare provider (%)



https://www.who.int/data/gho/data/indicators/indicator-details/GHO/children-aged-5-years-with-pneumonia-symptoms-taken-to-a-healthcare-provider-(-)

Motivation

AI is a powerful tool. If we can improve healthcare services, we should.

Image recognition is just one area of AI but is being used in analysis of x-rays and CAT scans with great success.[1,2]

[1] McKinney, S.M., Sieniek, M., Godbole, V. *et al.* International evaluation of an AI system for breast cancer screening. *Nature* 577, 89–94 (2020) doi:10.1038/s41586-019-1799-6
[2] B. Parmadean et al. Transfer Learning from Chest X-Ray Pre-trained Convolutional Neural Network for Learning Mammogram Data, (Procedia Computer Science Volume 135, 2018, Pages 400-407)
https://doi.org/10.1016/j.procs.2018.08.190

808,694

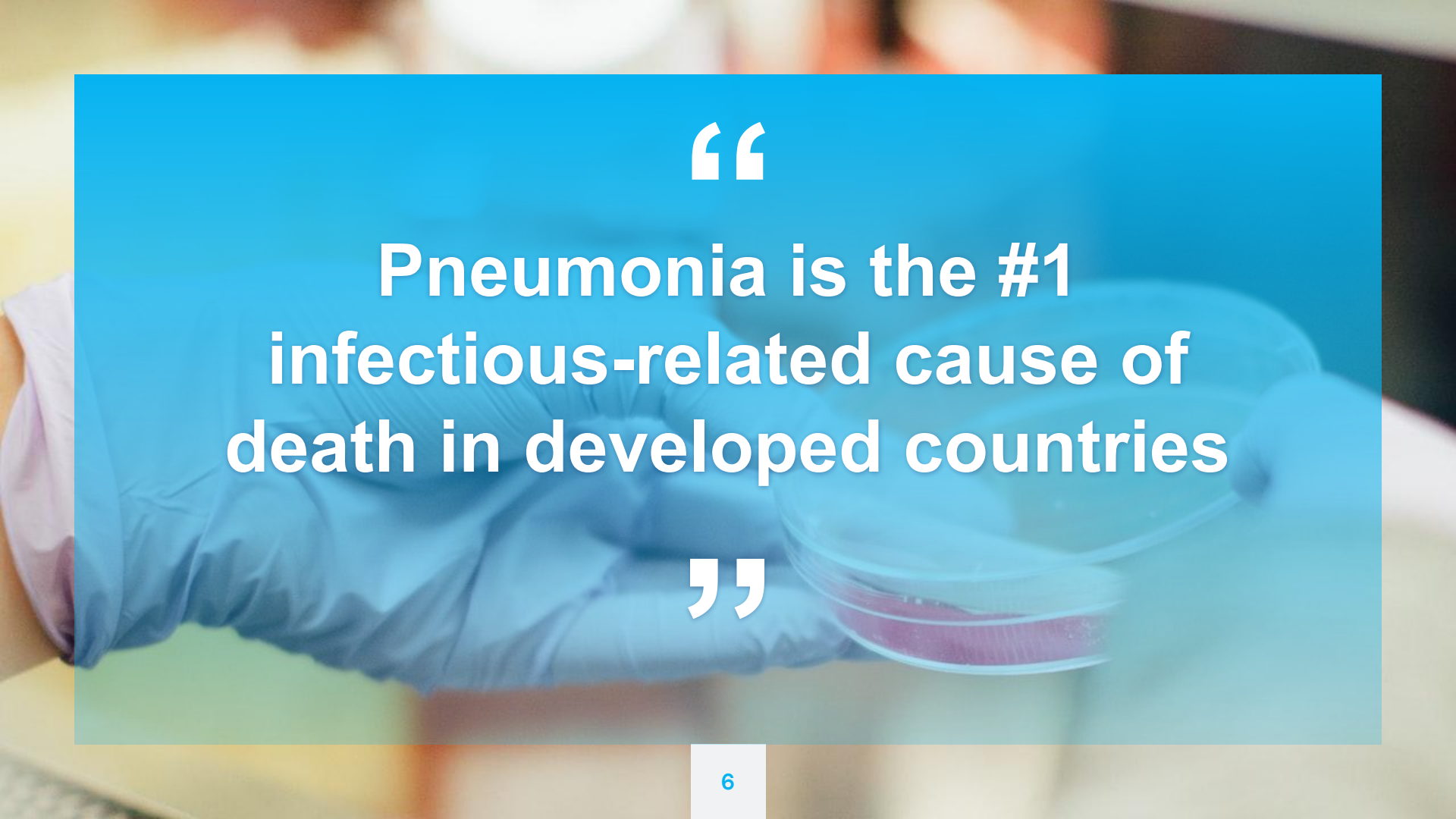
Number of deaths worldwide in children aged under 5 years (2017)

What is Pneumonia?

Lungs are made up of small air filled sacs, called alveoli.

Pneumonia causes these alveoli in the lungs to fill up with fluid.

This causes them to become enflamed.


“
Pneumonia is the #1 infectious-related cause of death in developed countries
”

Diagnosis & Treatment

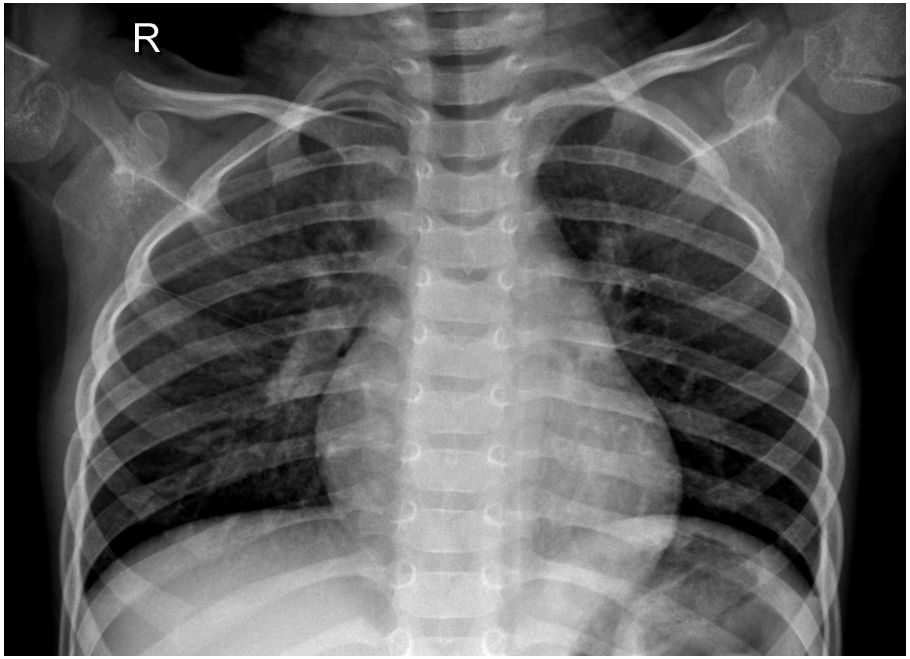
Main types of Pneumonia:
Bacterial, Viral.

Differential diagnoses only discovered through failure of treatment.

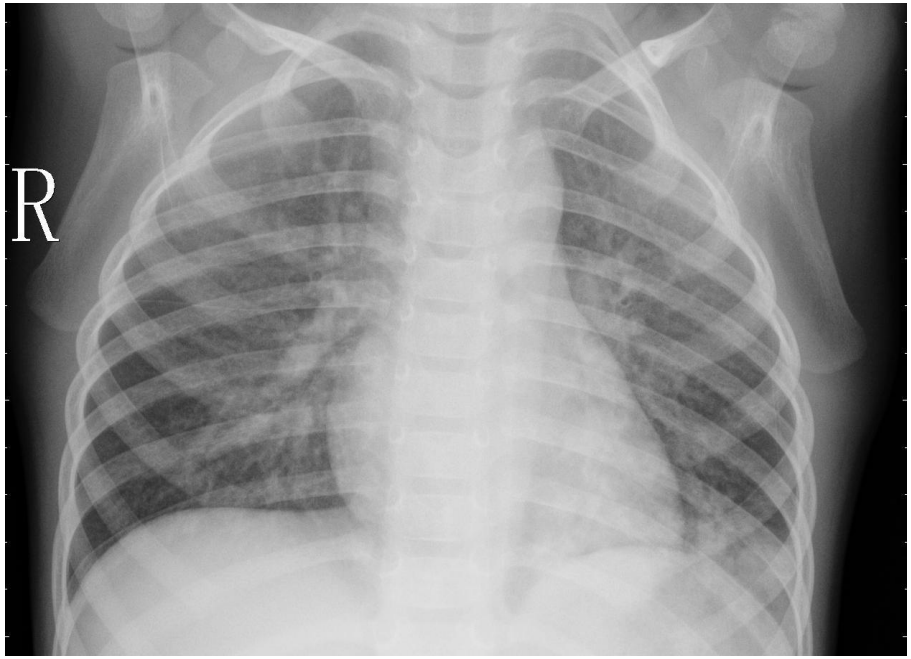
↑ Diagnoses Time = ↑ Mortality Rate

AI for Image Recognition

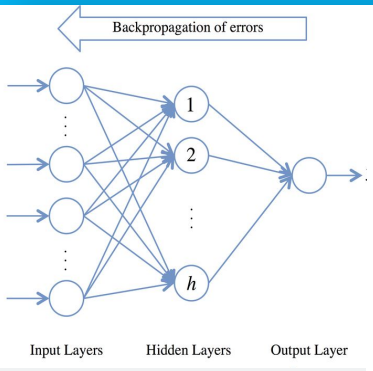
Can you tell the difference?



Healthy



Pneumonia



Neural Networks Can

98% Sick Patients
Correctly Diagnosed

86% Healthy Patients
Correctly Diagnosed

93%
Overall Accuracy

Benefits

- Saves time and resources.
- Higher overall accuracy than traditional diagnosis methods.
- Potential for diagnosing other diseases.

We can improve diagnoses speed and reduce mortality rates.

Thanks!

Any questions?

Credits

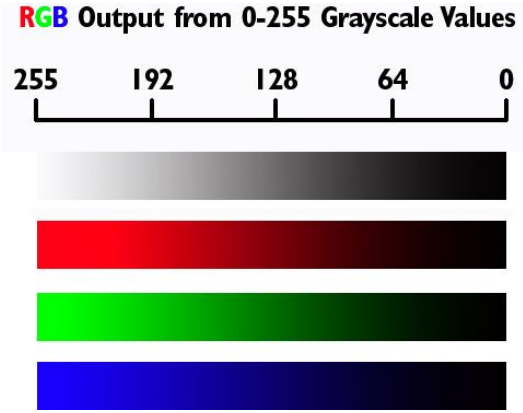
Special thanks to all the people who made and released these awesome resources for free:

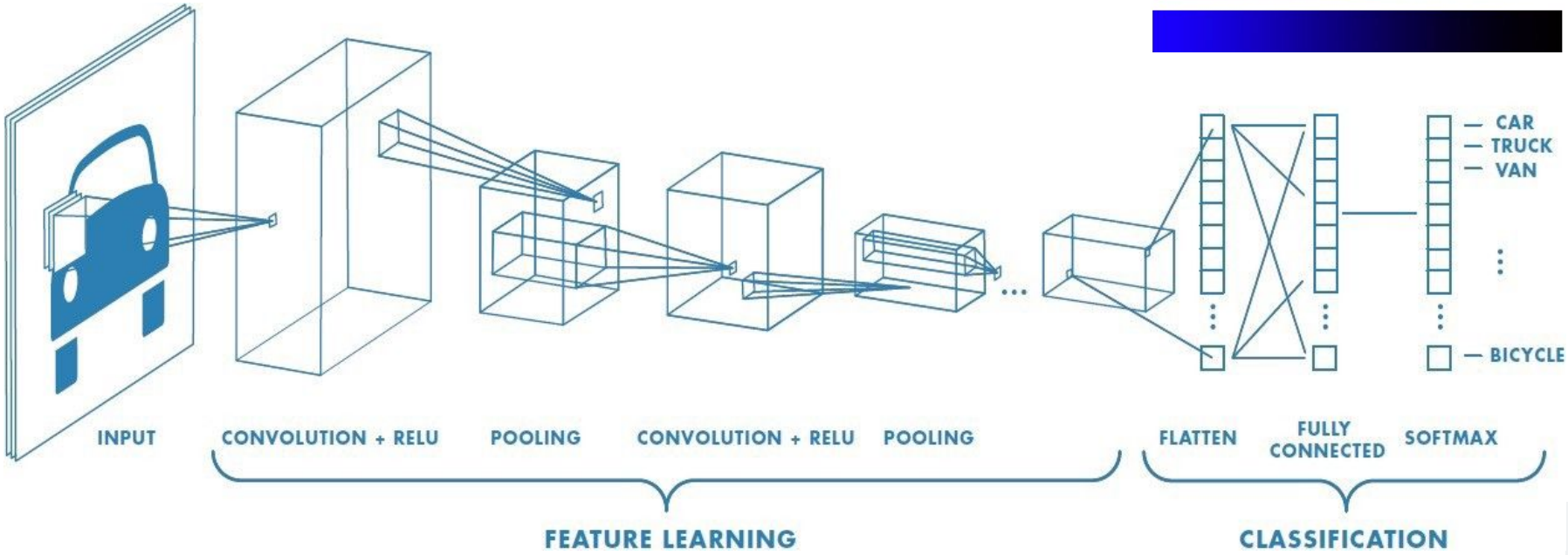
- Presentation template by SlidesCarnival
- Photographs by Unsplash

Find the notebook at: https://github.com/Greg-S-12/pneumonia_cnn

Appendix

Convolutional Neural Networks





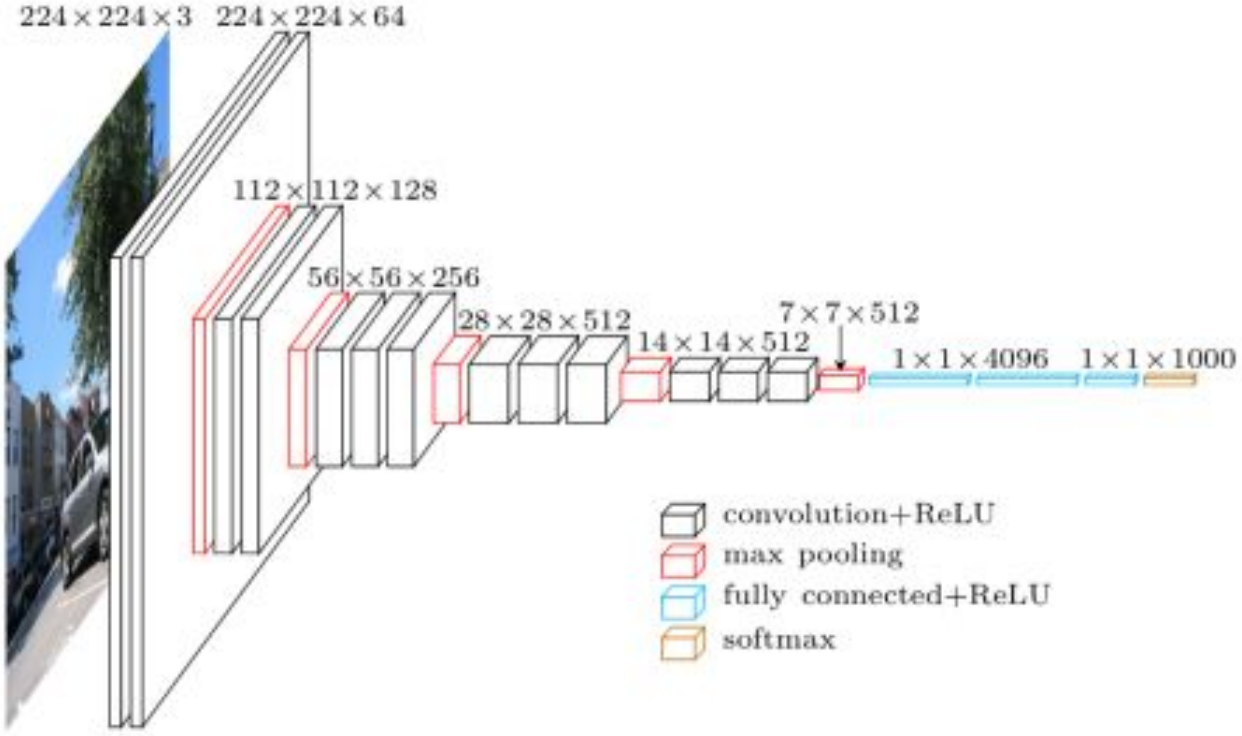
Appendix

The Model:

```
model_2 = Sequential()

model_2.add(VGG16(include_top=False, input_shape=(224,224,3)).layers[0])
model_2.add(VGG16(include_top=False, input_shape=(224,224,3)).layers[1])
model_2.add(VGG16(include_top=False, input_shape=(224,224,3)).layers[2])
model_2.add(VGG16(include_top=False, input_shape=(224,224,3)).layers[3])

model_2.add(Conv2D(filters=64, kernel_size=(3,3), padding="same", activation="swish", name='Conv1_1'))
model_2.add(Conv2D(filters=64, kernel_size=(3,3), padding="same", activation="swish", name='Conv1_2'))
model_2.add(MaxPool2D(pool_size=(2,2),strides=(2,2), name='Pool1'))

model_2.add(Conv2D(filters=128, kernel_size=(3,3), padding="same", activation="swish", name='Conv2_1'))
model_2.add(Conv2D(filters=128, kernel_size=(3,3), padding="same", activation="swish", name='Conv2_2'))
model_2.add(MaxPool2D(pool_size=(2,2),strides=(2,2), name='Pool2'))

model_2.add(Conv2D(filters=128, kernel_size=(3,3), padding="same", activation="swish", name='Conv3_1'))
model_2.add(Conv2D(filters=128, kernel_size=(3,3), padding="same", activation="swish", name='Conv3_2'))
model_2.add(MaxPool2D(pool_size=(2,2),strides=(2,2), name='Pool3'))

model_2.add(Conv2D(filters=256, kernel_size=(3,3), padding="same", activation="swish", name='Conv4_1'))
model_2.add(BatchNormalization())
model_2.add(Conv2D(filters=256, kernel_size=(3,3), padding="same", activation="swish", name='Conv4_2'))
model_2.add(BatchNormalization())
model_2.add(MaxPool2D(pool_size=(2,2),strides=(2,2), name='Pool4'))

model_2.add(Conv2D(filters=256, kernel_size=(3,3), padding="same", activation="swish", name='Conv5_1'))
model_2.add(BatchNormalization())
model_2.add(Conv2D(filters=256, kernel_size=(3,3), padding="same", activation="swish", name='Conv5_2'))
model_2.add(BatchNormalization())
model_2.add(MaxPool2D(pool_size=(2,2),strides=(2,2), name='Pool5'))

model_2.add(Conv2D(filters=512, kernel_size=(3,3), padding="same", activation="swish", name='Conv6_1'))
model_2.add(BatchNormalization())
model_2.add(Conv2D(filters=512, kernel_size=(3,3), padding="same", activation="swish", name='Conv6_2'))
model_2.add(BatchNormalization())
model_2.add(MaxPool2D(pool_size=(2,2),strides=(2,2), name='Pool6'))

model_2.add(Flatten(name="Flatten"))
model_2.add(Dense(units=1024,activation="swish", name='Dense1'))
model_2.add(Dense(units=512,activation="swish", name='Dense2'))
model_2.add(Dense(units=2, activation="softmax", name='Result'))
```

Appendix

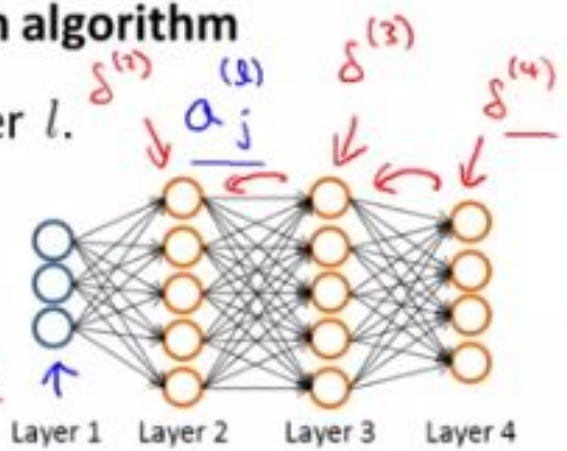
Transfer Learning: VGG16



Appendix

Gradient computation: Backpropagation algorithm

Intuition: $\delta_j^{(l)}$ = "error" of node j in layer l.



For each output unit (layer L = 4)

$$\delta_j^{(4)} = a_j^{(4)} - y_j$$

$$\delta^{(3)} = (\Theta^{(3)})^T \delta^{(4)} .* g'(z^{(3)})$$

$$\delta^{(2)} = (\Theta^{(2)})^T \delta^{(3)} .* g'(z^{(2)})$$

(No $\delta^{(1)}$)

Courtesy of Andrew Ng (Coursera)