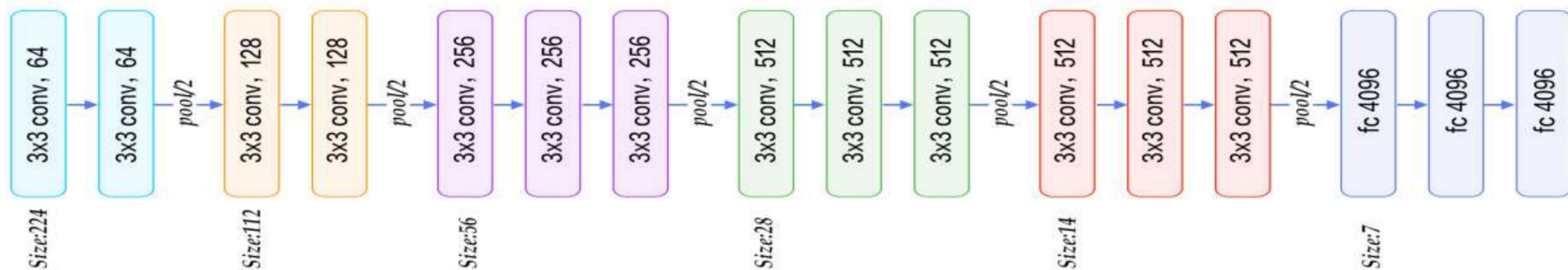
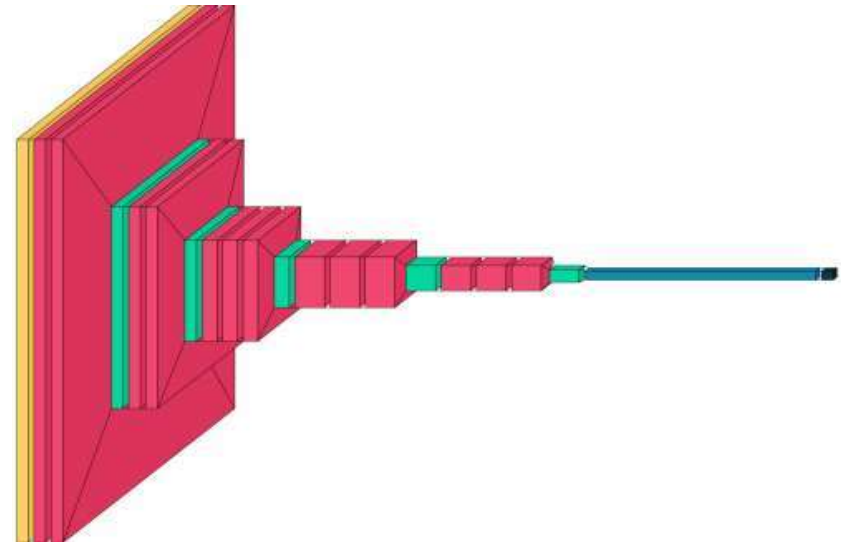
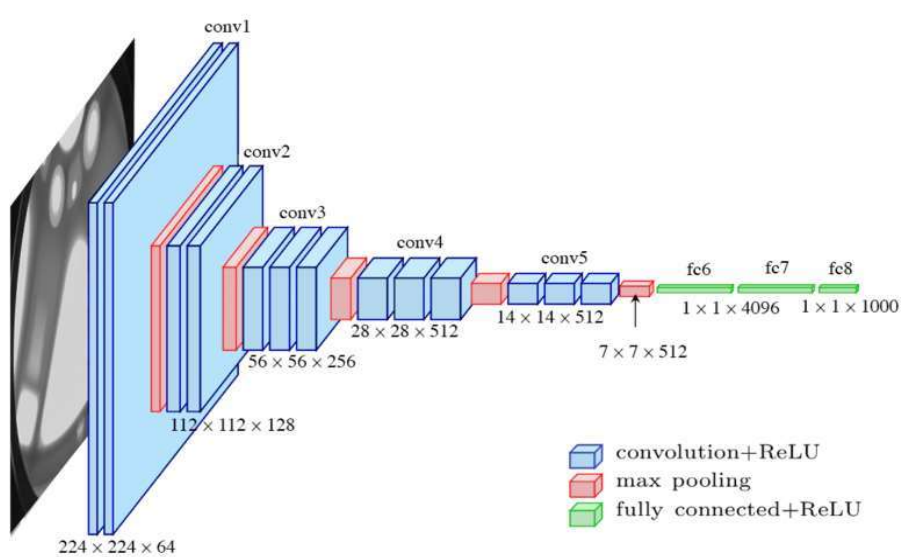


Covid-19 detection

USING CNN WITH VGG16 IMAGENET

VGG16 Architecture



•VGG16 architecture

Convolutional Layers = 13

Pooling Layers = 5

Dense Layers = 3

- **Input:** Image of dimensions (224, 224, 3).
- **Convolution Layer Conv1:**
 - Conv1-1: 64 filters
 - Conv1-2: 64 filters and Max Pooling
 - Image dimensions: (224, 224)
- **Convolution layer Conv2:** Now, we increase the filters to 128
 - Input Image dimensions: (112,112)
 - Conv2-1: 128 filters
 - Conv2-2: 128 filters and Max Pooling
- **Convolution Layer Conv3:** Again, double the filters to 256, and now add another convolution layer
 - Input Image dimensions: (56,56)
 - Conv3-1: 256 filters
 - Conv3-2: 256 filters
 - Conv3-3: 256 filters and Max Pooling

•VGG16 architecture

Convolutional Layers = 13

Pooling Layers = 5

Dense Layers = 3

- **Convolution Layer Conv4:** Similar to Conv3, but now with 512 filters
 - Input Image dimensions: (28, 28)
 - Conv4-1: 512 filters
 - Conv4-2: 512 filters
 - Conv4-3: 512 filters and Max Pooling
- **Convolution Layer Conv5:** Same as Conv4
 - Input Image dimensions: (14, 14)
 - Conv5-1: 512 filters
 - Conv5-2: 512 filters
 - Conv5-3: 512 filters and Max Pooling
 - The output dimensions here are (7, 7). At this point, we flatten the output of this layer to generate a feature vector
- **Fully Connected/Dense FC1:** 4096 nodes, generating a feature vector of size(1, 4096)
- **Fully Connected/Dense FC2:** 4096 nodes generating a feature vector of size(1, 4096)
- **Fully Connected /Dense FC3:** 4096 nodes, generating 1000 channels for 1000 classes. This is then passed on to a **Sigmoid** activation function
- **Output layer**

Activation and Dropout

We used Relu and Sigmoid as activation functions:

Relu: is the most used activation nowadays since it **doesn't activate** all layers every time which makes the **backward propagation** more **efficient** because weights will be updated as needed and not all the time

We didn't use any dropout techniques:

We didn't use any **dropout techniques** but avoided **overfitting** by making all layers **untrainable**