Notebook here:

<https://colab.research.google.com/drive/1y-YOqvIOdIzW8y0Hfi4mOZNITveRqqyw?usp=sharing>

## from tensorflow.keras.applications.vgg16 import VGG16

* Import the VGG16 model

pre\_trained\_model = VGG16(input\_shape=(600,600,3), include\_top = False)

for layer in pre\_trained\_model.layers:

layer.trainable = False

* Set the input\_shape to the shape you want
* Set include\_top to be false so you can train your own classification layer
* Freeze all the weights of the VGG16 Model so you won’t retrain them again

pre\_output = pre\_trained\_model.get\_layer('conv5\_block3\_1\_conv').output

x = Flatten()(pre\_output)

x = Dense(512, activation='relu')(x)

x = Dense(5, activation='softmax')(x)

model = Model(pre\_trained\_model.input, x)

* Get the output from VGG16 Model’s ‘conv5\_block3\_1\_conv’ layer and pass it as input to your first custom layer
* Add custom layers to the pre\_trained model (These layers are the ones specific to your problem)
* Combine the pre\_trained\_model and the custom layers and combine them, using the input of the VGG16 model and the output of the custom model.

Note: ***You only have to add on dense layers (For custom classification). No need to add your own Conv layers since they are already implemented in the pretrained\_model***