#VGG16

import tensorflow as tf

from tensorflow.keras.applications import VGG16

from tensorflow.keras import layers, Model

# Layers of vgg 16

vgg16.summary()

# Load VGG16 pre-trained on ImageNet data

base\_model = VGG16(weights='imagenet', include\_top=False, input\_shape=(150, 150, 3))

# Freeze the convolutional base

base\_model.trainable = False

# Add custom classification head

x = base\_model.output

x = layers.GlobalAveragePooling2D()(x)

x = layers.Dense(256, activation='relu')(x)

predictions = layers.Dense(6, activation='softmax')(x)  # 7 classes, change according to your dataset

# Combine base model and custom head

model = Model(inputs=base\_model.input, outputs=predictions)

# Compile the model

model.compile(optimizer='adam',

              loss='sparse\_categorical\_crossentropy',  # Use sparse categorical crossentropy since your labels are integers

              metrics=['accuracy'])

# Print model summary

model.summary()