import tensorflow as tf

def HybridizeLUActivationFunctionInCNNForSteganalysis(model):

activation\_functions = ['elu', 'relu', 'leakyrelu']

modified\_layers = []

for i in range(5):

for layer in model.layers:

if isinstance(layer, tf.keras.layers.Activation):

activation\_name = layer.get\_config()['activation']

activation\_output = layer(layer.input)

if tf.reduce\_sum(activation\_output) < 0:

new\_activation = tf.keras.layers.ELU()

elif tf.reduce\_sum(activation\_output) > 0:

new\_activation = tf.keras.layers.ReLU()

else:

new\_activation = tf.keras.layers.LeakyReLU()

modified\_layers.append(new\_activation)

else:

modified\_layers.append(layer)

model = tf.keras.models.Sequential(modified\_layers)

modified\_layers = []

return model

# Create a sample model

model = tf.keras.Sequential([

tf.keras.layers.Conv2D(32, (3, 3), activation='relu', input\_shape=(28, 28, 1)),

tf.keras.layers.MaxPooling2D((2, 2)),

tf.keras.layers.Flatten(),

tf.keras.layers.Dense(64, activation='relu'),

tf.keras.layers.Dense(10, activation='softmax')

])

# Apply the function to modify the model

modified\_model = HybridizeLUActivationFunctionInCNNForSteganalysis(model)

# Print the modified model summary

modified\_model.summary()