

In []:

```

In [ ]: def encoder(inputs, num_filters):
    x = tf.keras.layers.Conv2D(num_filters,3,padding = 'valid')(inputs)
    x = tf.keras.layers.Activation('relu')(x)

    x = tf.keras.layers.Conv2D(num_filters,3,padding = 'valid')(x)
    x = tf.keras.layers.Activation('relu')(x)

    x = tf.keras.layers.MaxPool2D(pool_size = (2,3), strides = 2)(x)

    return x

In [ ]: def decoder(inputs, skip_features, num_filters):
    x = tf.keras.layers.Conv2DTranspose(num_filters,(2,3),strides = 2,padding = 'va

    skip_features = tf.image.resize(skip_features,size = (x.shape[1],x.shape[2]))

    x = tf.keras.layers.Concatenate()([x,skip_features])

    x = tf.keras.layers.Conv2D(num_filters,3,padding = 'valid')(x)
    x = tf.keras.layers.Activation('relu')(x)

    x = tf.keras.layers.Conv2D(num_filters,3,padding = 'valid')(x)
    x = tf.keras.layers.Activation('relu')(x)

    return x

In [ ]: import tensorflow as tf

def unet_model(input_shape = (256,256,3), num_classes = 1):
    inputs = tf.keras.layers.Input(input_shape)

    s1 = encoder(inputs,64)
    s2 = encoder(s1,128)
    s3 = encoder(s2,256)
    s4 = encoder(s3,512)

    b1 = tf.keras.layers.Conv2D(2014,3,padding = 'valid')(s4)
    b1 = tf.keras.layers.Activation('relu')(b1)
    b1 = tf.keras.layers.Conv2D(1024,3, padding = 'valid')(b1)
    b1 = tf.keras.layers.Activation('relu')(b1)

    s5 = decoder(b1,s4,512)
    s6 = decoder(s5,s3,256)
    s7 = decoder(s6,s2,128)
    s8 = decoder(s7,s1,64)

    outputs = tf.keras.layers.Conv2D(num_classes,1,padding = 'valid', activation =

    model = tf.keras.models.Model(inputs = inputs,outputs = outputs , name = 'U-Net

    return model

```

```
In [ ]: if __name__ == '__main__':  
        model = unet_model(input_shape=(572,572,3), num_classes = 2)  
        model.summary()
```

Model: "U-Net"

Layer (type)	Output Shape	Param #	Connected to
input_10 (InputLayer)	[(None, 572, 572, 3)]	0	[]
conv2d_75 (Conv2D)	(None, 570, 570, 64)	1792	['input_10[0]
activation_75 (Activation)	(None, 570, 570, 64)	0	['conv2d_75[0]
conv2d_76 (Conv2D)	(None, 568, 568, 64)	36928	['activation_75
conv2d_77 (Conv2D)	(None, 282, 281, 128)	73856	['max_pooling2d_
activation_77 (Activation)	(None, 282, 281, 128)	0	['conv2d_77[0]
conv2d_78 (Conv2D)	(None, 280, 279, 128)	147584	['activation_77
activation_78 (Activation)	(None, 280, 279, 128)	0	['conv2d_78[0]
max_pooling2d_32 (MaxPooling2D)	(None, 140, 139, 128)	0	['activation_78

conv2d_79 (Conv2D)	(None, 138, 137, 256)	295168	['max_pooling2d_32[0][0]']
activation_79 (Activation)	(None, 138, 137, 256)	0	['conv2d_79[0][0]']
conv2d_80 (Conv2D)	(None, 136, 135, 256)	590080	['activation_79[0][0]']
activation_80 (Activation)	(None, 136, 135, 256)	0	['conv2d_80[0][0]']
max_pooling2d_33 (MaxPooling2D)	(None, 68, 67, 256)	0	['activation_80[0][0]']
conv2d_81 (Conv2D)	(None, 66, 65, 512)	1180160	['max_pooling2d_33[0][0]']
activation_81 (Activation)	(None, 66, 65, 512)	0	['conv2d_81[0][0]']
conv2d_82 (Conv2D)	(None, 64, 63, 512)	2359808	['activation_81[0][0]']
activation_82 (Activation)	(None, 64, 63, 512)	0	['conv2d_82[0][0]']
max_pooling2d_34 (MaxPooling2D)	(None, 32, 31, 512)	0	['activation_82[0][0]']
conv2d_83 (Conv2D)	(None, 30, 29, 2014)	9282526	['max_pooling2d_34[0][0]']
activation_83 (Activation)	(None, 30, 29, 2014)	0	['conv2d_83[0][0]']
conv2d_84 (Conv2D)	(None, 28, 27, 1024)	18562048	['activation_83[0][0]']
activation_84 (Activation)	(None, 28, 27, 1024)	0	['conv2d_84[0][0]']
conv2d_transpose_5 (Conv2DTranspose)	(None, 56, 55, 512)	3146240	['activation_84[0][0]']
tf.image.resize_3 (TFOPLambda)	(None, 56, 55, 512)	0	['max_pooling2d_34[0][0]']
concatenate_2 (Concatenate)	(None, 56, 55, 1024)	0	['conv2d_transpose_5[0][0]', 'tf.image.resize_3[0][0]']

e_3[0][0]']

conv2d_85 (Conv2D) [0][0]']	(None, 54, 53, 512)	4719104	['concatenate_2
activation_85 (Activation) [0]']	(None, 54, 53, 512)	0	['conv2d_85[0]
conv2d_86 (Conv2D) [0][0]']	(None, 52, 51, 512)	2359808	['activation_85
activation_86 (Activation) [0]']	(None, 52, 51, 512)	0	['conv2d_86[0]
conv2d_transpose_6 (Conv2D Transpose) [0][0]']	(None, 104, 103, 256)	786688	['activation_86
tf.image.resize_4 (TFOPLa 33[0][0]'] bda)	(None, 104, 103, 256)	0	['max_pooling2d_
concatenate_3 (Concatenate se_6[0][0]',) e_4[0][0]']	(None, 104, 103, 512)	0	['conv2d_transpo 'tf.image.resiz
conv2d_87 (Conv2D) [0][0]']	(None, 102, 101, 256)	1179904	['concatenate_3
activation_87 (Activation) [0]']	(None, 102, 101, 256)	0	['conv2d_87[0]
conv2d_88 (Conv2D) [0][0]']	(None, 100, 99, 256)	590080	['activation_87
activation_88 (Activation) [0]']	(None, 100, 99, 256)	0	['conv2d_88[0]
conv2d_transpose_7 (Conv2D Transpose) [0][0]']	(None, 200, 199, 128)	196736	['activation_88
tf.image.resize_5 (TFOPLa 32[0][0]'] bda)	(None, 200, 199, 128)	0	['max_pooling2d_
concatenate_4 (Concatenate se_7[0][0]',) e_5[0][0]']	(None, 200, 199, 256)	0	['conv2d_transpo 'tf.image.resiz
conv2d_89 (Conv2D) [0][0]']	(None, 198, 197, 128)	295040	['concatenate_4
activation_89 (Activation)	(None, 198, 197, 128)	0	['conv2d_89[0]

[0]']

conv2d_90 (Conv2D)	(None, 196, 195, 128)	147584	['activation_89 [0][0]']
activation_90 (Activation)	(None, 196, 195, 128)	0	['conv2d_90[0] [0]']
conv2d_transpose_8 (Conv2D Transpose)	(None, 392, 391, 64)	49216	['activation_90 [0][0]']
tf.image.resize_6 (TFOPLam 31[0][0]') bda)	(None, 392, 391, 64)	0	['max_pooling2d_ 31[0][0]']
concatenate_5 (Concatenate se_8[0][0]'),) e_6[0][0]']	(None, 392, 391, 128)	0	['conv2d_transpo se_8[0][0]'], 'tf.image.resiz e_6[0][0]']
conv2d_91 (Conv2D)	(None, 390, 389, 64)	73792	['concatenate_5 [0][0]']
activation_91 (Activation)	(None, 390, 389, 64)	0	['conv2d_91[0] [0]']
conv2d_92 (Conv2D)	(None, 388, 387, 64)	36928	['activation_91 [0][0]']
activation_92 (Activation)	(None, 388, 387, 64)	0	['conv2d_92[0] [0]']
conv2d_93 (Conv2D)	(None, 388, 387, 2)	130	['activation_92 [0][0]']

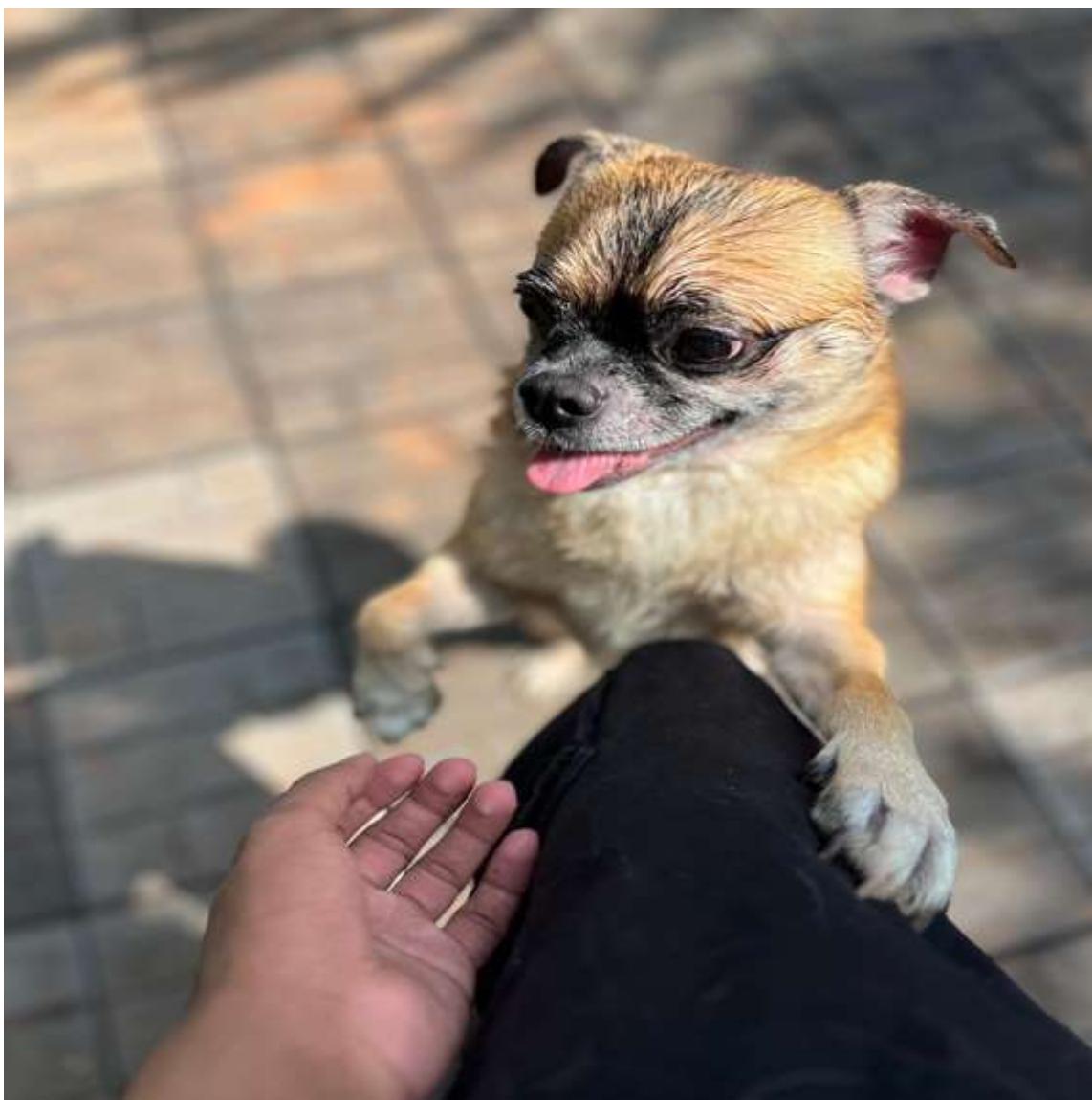
```
=====
=====
```

```
Total params: 46111200 (175.90 MB)
Trainable params: 46111200 (175.90 MB)
Non-trainable params: 0 (0.00 Byte)
```

```
In [ ]: import numpy as np
        from PIL import Image
        from tensorflow.keras.preprocessing import image
```

```
In [ ]: img
```

Out[]:



```

In [ ]: img = Image.open('images/dog.jpg')

img = img.resize((572,572))
img_array = image.img_to_array(img)
img_array = np.expand_dims(img_array[:,:,:3], axis = 0)
img_array = img_array / 255.

model = unet_model(input_shape = (572,572,3), num_classes = 2)

predictions = model.predict(img_array)

predictions = np.squeeze(predictions,axis = 0)
predictions = np.argmax(predictions,axis = -1)
predictions = Image.fromarray(np.uint8(predictions*255))
predictions = predictions.resize((img.width,img.height))

predictions.save('images/outputs/predicted_image.jpg')
predictions

```

1/1 [=====] - 2s 2s/step

Out[]:

