```
from keras.models import  Sequential
from keras.layers import Dense , Conv2D , MaxPooling2D, Flatten


model = Sequential()


num_classes = 4


model = Sequential()
model.add(Conv2D(64, (3,3), input_shape=(64,64,3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dense(num_classes, activation='softmax'))
```

```
/usr/local/lib/python3.11/dist-packages/keras/src/layers/convolutional/base_conv.py:107: UserWarning: Do not pass an `input_shape`/`inpu
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
```

```
model.add(Dense(4, activation='softmax'))


model.summary()
```

Model: "sequential_5"

Layer (type)	Output Shape	Param #
conv2d_2 (Conv2D)	(None, 62, 62, 64)	1,792
max_pooling2d_2 (MaxPooling2D)	(None, 31, 31, 64)	0
flatten_2 (Flatten)	(None, 61504)	0
dense_7 (Dense)	(None, 128)	7,872,640
dense_8 (Dense)	(None, 4)	516
dense_9 (Dense)	(None, 3)	15

Total params: 7,874,963 (30.04 MB)
Trainable params: 7,874,963 (30.04 MB)
Non-trainable params: 0 (0.00 B)

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
train_datagen = ImageDataGenerator(
    rescale=1./255,
    shear_range=0.2,
    zoom_range=0.2,
    width_shift_range=0.2,
    height_shift_range=0.2,
    fill_mode='nearest',
    vertical_flip=True,
    horizontal_flip=True)
test_datagen = ImageDataGenerator(rescale=1./255)
```

```
train_path = '/content/drive/MyDrive/test'
test_path = '/content/drive/MyDrive/test'
aim_path = '/content/drive/MyDrive/task/aim'
train_generator = train_datagen.flow_from_directory(
    train path,
```

```
train_generator.class_indices
```

```
class_mode='categorical')
```

```
{'test': 0, 'train': 1}
```

```
test_generator.class_indices
```

```
target_size=(64,64),
batch_size=16,
class_mode='categorical')
```

```
{'test': 0, 'train': 1}
```

```
aim_generator.class_indices
```

```
target_size=(64,64),
batch_size=16,
class_mode='categorical')
```

```
{'bikes': 0, 'cat': 1, 'cities': 2, 'dog': 3}
```

```
predictions = model.predict(test_generator)
```

```
Found 39 images belonging to 4 classes.
/usr/local/lib/python3.11/dist-packages/keras/src/trainers/data_adapters/py_dataset_adapter.py:121: UserWarning: Your `PyDataset` class
  self._warn_if_super_not_called()
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 249ms/step
```

```
model.add(Dense(4, activation='softmax'))
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
```

```
model.fit(train_generator, epochs=100, validation_data=test_generator)
```

```
Epoch 1/100
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-49-69693dbd5bfc> in <cell line: 0>()
----> 1 model.fit(train_generator, epochs=100, validation_data=test_generator)

 1 frames
/usr/local/lib/python3.11/dist-packages/keras/src/backend/tensorflow/nn.py in categorical_crossentropy(target, output, from_logits,
axis)
    658     for e1, e2 in zip(target.shape, output.shape):
    659         if e1 is not None and e2 is not None and e1 != e2:
--> 660             raise ValueError(
    661                 "Arguments `target` and `output` must have the same shape. "
    662                 "Received: "

ValueError: Arguments `target` and `output` must have the same shape. Received: target.shape=(None, 2), output.shape=(None, 4)
```