In [2]:

```python
import numpy as np
import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

In [5]:

```python
train_datagen = ImageDataGenerator(
        rescale=1./255,
        shear_range=0.2,
        zoom_range=0.2,
        horizontal_flip=True)
training_set = train_datagen.flow_from_directory(
        'training_set',
        target_size=(64, 64),
        batch_size=32,
        class_mode='categorical')
```

```
Found 3459 images belonging to 5 classes.
```

In [6]:

```python
test_datagen = ImageDataGenerator(rescale=1./255)
test_set = test_datagen.flow_from_directory(
        'test_set',
        target_size=(64, 64),
        batch_size=32,
        class_mode='categorical')
```

```
Found 858 images belonging to 5 classes.
```

In [7]:

```python
cnn = tf.keras.models.Sequential()
```

In [8]:

```python
cnn.add(tf.keras.layers.Conv2D(filters=64 , kernel_size=3 , activation='relu' , input_shape=[64,64,3]))
cnn.add(tf.keras.layers.MaxPool2D(pool_size=2,strides=2))
```

```
C:\anaconda\Lib\site-packages\keras\src\layers\convolutional\base_conv.py:107: UserWarning: Do not pass an `input_shape`/`input_dim` argument to a layer. When using Sequential mode
ls, prefer using an `Input(shape)` object as the first layer in the model instead.
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
```

In [9]:

```python
cnn.add(tf.keras.layers.Conv2D(filters=64 , kernel_size=3 , activation='relu' ))
cnn.add(tf.keras.layers.MaxPool2D(pool_size=2 , strides=2))
```

In [10]:

```python
cnn.add(tf.keras.layers.Dropout(0.5))
```

In [11]:

```python
cnn.add(tf.keras.layers.Flatten())
```

In [12]:

```python
cnn.add(tf.keras.layers.Dense(units=128, activation='relu'))
```

In [13]:

```python
cnn.add(tf.keras.layers.Dense(units=5 , activation='softmax'))
```

In [14]:

```python
cnn.compile(optimizer = 'rmsprop' , loss = 'categorical_crossentropy' , metrics = ['accuracy'])
```

In [15]:

```python
cnn.fit(x = training_set , validation_data = test_set , epochs = 30)
```

```
Epoch 1/30
C:\anaconda\Lib\site-packages\keras\src\trainers\data_adapters\py_dataset_adapter.py:121: UserWarning: Your `PyDataset` class should call `super().__init__(**kwargs)` in its constr
uctor. `**kwargs` can include `workers`, `use_multiprocessing`, `max_queue_size`. Do not pass these arguments to `fit()`, as they will be ignored.
  self._warn_if_super_not_called()
109/109 ━━━━━━━━━━━━━━━━━━━━ 63s 552ms/step - accuracy: 0.3058 - loss: 1.5520 - val_accuracy: 0.5361 - val_loss: 1.0949
Epoch 2/30
109/109 ━━━━━━━━━━━━━━━━━━━━ 16s 142ms/step - accuracy: 0.5196 - loss: 1.1580 - val_accuracy: 0.6131 - val_loss: 0.9763
Epoch 3/30
109/109 ━━━━━━━━━━━━━━━━━━━━ 16s 141ms/step - accuracy: 0.5952 - loss: 1.0259 - val_accuracy: 0.6212 - val_loss: 0.9542
Epoch 4/30
109/109 ━━━━━━━━━━━━━━━━━━━━ 16s 139ms/step - accuracy: 0.6158 - loss: 0.9672 - val_accuracy: 0.6375 - val_loss: 0.9167
Epoch 5/30
109/109 ━━━━━━━━━━━━━━━━━━━━ 16s 143ms/step - accuracy: 0.6609 - loss: 0.8785 - val_accuracy: 0.5862 - val_loss: 1.1033
Epoch 6/30
109/109 ━━━━━━━━━━━━━━━━━━━━ 17s 150ms/step - accuracy: 0.6665 - loss: 0.8694 - val_accuracy: 0.6142 - val_loss: 0.9769
Epoch 7/30
109/109 ━━━━━━━━━━━━━━━━━━━━ 17s 146ms/step - accuracy: 0.7035 - loss: 0.8001 - val_accuracy: 0.6200 - val_loss: 1.0681
Epoch 8/30
109/109 ━━━━━━━━━━━━━━━━━━━━ 16s 139ms/step - accuracy: 0.7059 - loss: 0.7737 - val_accuracy: 0.6375 - val_loss: 0.9107
Epoch 9/30
109/109 ━━━━━━━━━━━━━━━━━━━━ 16s 144ms/step - accuracy: 0.7072 - loss: 0.7496 - val_accuracy: 0.6037 - val_loss: 1.1726
Epoch 10/30
109/109 ━━━━━━━━━━━━━━━━━━━━ 22s 199ms/step - accuracy: 0.7232 - loss: 0.7204 - val_accuracy: 0.6946 - val_loss: 0.8329
Epoch 11/30
109/109 ━━━━━━━━━━━━━━━━━━━━ 57s 509ms/step - accuracy: 0.7411 - loss: 0.6801 - val_accuracy: 0.6970 - val_loss: 0.8214
Epoch 12/30
109/109 ━━━━━━━━━━━━━━━━━━━━ 75s 448ms/step - accuracy: 0.7469 - loss: 0.6450 - val_accuracy: 0.6935 - val_loss: 0.8562
Epoch 13/30
109/109 ━━━━━━━━━━━━━━━━━━━━ 49s 432ms/step - accuracy: 0.7547 - loss: 0.6451 - val_accuracy: 0.6620 - val_loss: 0.9396
Epoch 14/30
109/109 ━━━━━━━━━━━━━━━━━━━━ 51s 447ms/step - accuracy: 0.7866 - loss: 0.5850 - val_accuracy: 0.6853 - val_loss: 0.8712
Epoch 15/30
109/109 ━━━━━━━━━━━━━━━━━━━━ 52s 462ms/step - accuracy: 0.7660 - loss: 0.6184 - val_accuracy: 0.6702 - val_loss: 0.8784
Epoch 16/30
109/109 ━━━━━━━━━━━━━━━━━━━━ 52s 463ms/step - accuracy: 0.7883 - loss: 0.5676 - val_accuracy: 0.6993 - val_loss: 0.8656
Epoch 17/30
109/109 ━━━━━━━━━━━━━━━━━━━━ 54s 486ms/step - accuracy: 0.7976 - loss: 0.5298 - val_accuracy: 0.6503 - val_loss: 0.9462
Epoch 18/30
109/109 ━━━━━━━━━━━━━━━━━━━━ 53s 473ms/step - accuracy: 0.7996 - loss: 0.5105 - val_accuracy: 0.6865 - val_loss: 0.8822
Epoch 19/30
109/109 ━━━━━━━━━━━━━━━━━━━━ 53s 475ms/step - accuracy: 0.8081 - loss: 0.5186 - val_accuracy: 0.7249 - val_loss: 0.8601
Epoch 20/30
109/109 ━━━━━━━━━━━━━━━━━━━━ 60s 537ms/step - accuracy: 0.8093 - loss: 0.4854 - val_accuracy: 0.6713 - val_loss: 1.0125
Epoch 21/30
109/109 ━━━━━━━━━━━━━━━━━━━━ 50s 446ms/step - accuracy: 0.8149 - loss: 0.4976 - val_accuracy: 0.7168 - val_loss: 0.8898
Epoch 22/30
109/109 ━━━━━━━━━━━━━━━━━━━━ 54s 486ms/step - accuracy: 0.8317 - loss: 0.4575 - val_accuracy: 0.6725 - val_loss: 1.0386
Epoch 23/30
109/109 ━━━━━━━━━━━━━━━━━━━━ 52s 460ms/step - accuracy: 0.8313 - loss: 0.4273 - val_accuracy: 0.7040 - val_loss: 0.9411
Epoch 24/30
109/109 ━━━━━━━━━━━━━━━━━━━━ 52s 459ms/step - accuracy: 0.8443 - loss: 0.4176 - val_accuracy: 0.6760 - val_loss: 1.0296
Epoch 25/30
109/109 ━━━━━━━━━━━━━━━━━━━━ 57s 512ms/step - accuracy: 0.8493 - loss: 0.4151 - val_accuracy: 0.6678 - val_loss: 1.0436
Epoch 26/30
109/109 ━━━━━━━━━━━━━━━━━━━━ 53s 470ms/step - accuracy: 0.8552 - loss: 0.3910 - val_accuracy: 0.6888 - val_loss: 0.9671
Epoch 27/30
109/109 ━━━━━━━━━━━━━━━━━━━━ 54s 475ms/step - accuracy: 0.8471 - loss: 0.4197 - val_accuracy: 0.6970 - val_loss: 0.9602
Epoch 28/30
109/109 ━━━━━━━━━━━━━━━━━━━━ 68s 609ms/step - accuracy: 0.8607 - loss: 0.3625 - val_accuracy: 0.7016 - val_loss: 0.9873
Epoch 29/30
109/109 ━━━━━━━━━━━━━━━━━━━━ 56s 499ms/step - accuracy: 0.8840 - loss: 0.3466 - val_accuracy: 0.6888 - val_loss: 1.1544
Epoch 30/30
109/109 ━━━━━━━━━━━━━━━━━━━━ 57s 508ms/step - accuracy: 0.8637 - loss: 0.3555 - val_accuracy: 0.7098 - val_loss: 0.9345
```

Out[15]:

```
<keras.src.callbacks.history.History at 0x15aa6b056d0>
```

In [34]:

```python
from keras.preprocessing import image
test_image = image.load_img('Prediction/tu1.jpeg',target_size=(64,64))
test_image = image.img_to_array(test_image)
test_image = np.expand_dims(test_image,axis=0)
result = cnn.predict(test_image)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step
```

In [35]:

```python
training_set.class_indices
```

Out[35]:

```
{'daisy': 0, 'dandelion': 1, 'rose': 2, 'sunflower': 3, 'tulip': 4}
```

In [36]:

```python
if result[0][0]==1:
    print('Daisy')
elif result[0][1]==1:
    print('Dandelion')
elif result[0][2]==1:
    print('Rose')
elif result[0][3]==1:
    print('SunFlower')
elif result[0][4]==1:
    print("Tulip")
```

```
Tulip
```

In [37]:

```python
print(result)
```

```
[[0. 0. 0. 0. 1.]]
```

In [ ]:

```python
```