In [ ]:

```
https://www.tensorflow.org/tutorials/images/classification
```

In [2]:

```
import matplotlib.pyplot as plt
import numpy as np
import os
import PIL
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
from tensorflow.keras.models import Sequential
```

In [3]:

```
import pathlib
dataset_url="https://storage.googleapis.com/download.tensorflow.org/example_images/flower_photos.tgz"
data_dir = tf.keras.utils.get_file('flower_photos',origin=dataset_url, untar=True)
data_dir = pathlib.Path(data_dir)
```

In [4]:

```
length=len(list(data_dir.glob('*/*.jpg')))
print(length)
```

```
3670
```

In [5]:

```
sunflower=list(data_dir.glob('sunflowers/*'))
PIL.Image.open(str(sunflower[5]))
```

Out[5]:

In [6]:

```
roses=list(data_dir.glob('roses/*'))
PIL.Image.open(str(roses[5]))
```

Out[6]:

In [7]:

```
tulips=list(data_dir.glob('tulips/*'))
PIL.Image.open(str(tulips[0]))
```

Out[7]:

In [8]:

```
batch_size=32
img_height=180
img_width=180
```

In [9]:

```
train_ds = tf.keras.preprocessing.image_dataset_from_directory(
  data_dir,
  validation_split=0.2,
  subset="training",
  seed=123,
  image_size=(img_height, img_width),
  batch_size=batch_size)
```

```
Found 3670 files belonging to 5 classes.
Using 2936 files for training.
```

In [10]:

```
val_ds=tf.keras.preprocessing.image_dataset_from_directory(
       data_dir,
       validation_split=0.2,
       subset="validation",
       seed=123,
       image_size=(img_height,img_width),
       batch_size=batch_size)
```

```
Found 3670 files belonging to 5 classes.
Using 734 files for validation.
```

In [11]:

```
class_names=train_ds.class_names
print(class_names)
```

```
['daisy', 'dandelion', 'roses', 'sunflowers', 'tulips']
```

In [12]:

```
import matplotlib.pyplot as plt
plt.figure(figsize=(10,10))
for images,labels in train_ds.take(1):
    for i in range(9):
        ax=plt.subplot(3,3,i+1)
        plt.imshow(images[i].numpy().astype("uint8"))
        plt.title(class_names[labels[i]])
        plt.axis("off")
```

In [13]:

```
for image_batch,label_batch in train_ds:
    print(image_batch.shape)
    print(label_batch.shape)
    break
```

```
(32, 180, 180, 3)
(32,)
```

In [14]:

```
normaliztion_layer=tf.keras.layers.experimental.preprocessing.Rescaling(1./255)
```

In [15]:

```
num_classes=5
model=tf.keras.Sequential([layers.experimental.preprocessing.Rescaling(1./255),
                          layers.Conv2D(32,3,activation='relu'),
                          layers.MaxPooling2D(),
                          layers.Conv2D(32,3,activation='relu'),
                          layers.MaxPooling2D(),
                          layers.Conv2D(32,3,activation='relu'),
                          layers.MaxPool2D(),
                          layers.Flatten(),
                          layers.Dense(128,activation='relu'),
                          layers.Dense(num_classes)
                          ])
```

In [16]:

```
model.compile(optimizer='adam',loss=tf.losses.SparseCategoricalCrossentropy(from_logits=True),
              metrics=['accuracy'])
```

In [17]:

```
AUTOTUNE = tf.data.experimental.AUTOTUNE

train_ds = train_ds.cache().prefetch(buffer_size=AUTOTUNE)
val_ds = val_ds.cache().prefetch(buffer_size=AUTOTUNE)
```

In [18]:

```
model.fit(train_ds,validation_data=val_ds,epochs=3)
```

```
Epoch 1/3
92/92 [==============================] - 331s 4s/step - loss: 1.4052 - accuracy: 0.3851 - val_loss:
1.0659 - val_accuracy: 0.5967
Epoch 2/3
92/92 [==============================] - 305s 3s/step - loss: 0.9707 - accuracy: 0.6064 - val_loss:
1.0386 - val_accuracy: 0.5995
Epoch 3/3
92/92 [==============================] - 224s 2s/step - loss: 0.8135 - accuracy: 0.6792 - val_loss:
0.9901 - val_accuracy: 0.6158
```

Out[18]:

```
<tensorflow.python.keras.callbacks.History at 0x7f248c344b90>
```

In [ ]:

```

```