In [77]:
```
import numpy as np
import pandas as pd
import tensorflow as tf
```

In [78]:
```
import os
train_data=os.path.join('C:/Users/Administrator/Desktop/saqlain python/CNN/train'
validation_data=os.path.join('C:/Users/Administrator/Desktop/saqlain python/CNN/v
```

In [79]:
```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
train_datagen = ImageDataGenerator(rescale=1./255)
train_generator = train_datagen.flow_from_directory(train_data,target_size=(150,

validation_datagen = ImageDataGenerator(rescale=1./255)
validation_generator = validation_datagen.flow_from_directory(validation_data,tar
```

```
Found 299 images belonging to 2 classes.
Found 49 images belonging to 2 classes.
```

In [80]:
```
from tensorflow.keras import layers
from tensorflow.keras import models
```

In [81]:
```
from tensorflow.keras import optimizers
model = models.Sequential()
model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['acc'])
```

In [82]:
```
model.add(layers.Conv2D(32, (3, 3), activation='relu',input_shape=(150, 150, 3)))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(128, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(128, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Flatten())
model.add(layers.Dense(512, activation='relu'))
model.add(layers.Dense(2, activation='softmax'))
```

In [83]:
```
result = model.fit_generator(train_generator,epochs=20,validation_data=validatior
```

```
Epoch 1/20
10/10 [==============================] - 9s 888ms/step - loss: 0.9175 - acc: 0.
5719 - val_loss: 0.5172 - val_acc: 0.9184
Epoch 2/20
10/10 [==============================] - 6s 624ms/step - loss: 0.4016 - acc: 0.
8395 - val_loss: 0.3923 - val_acc: 0.8776
Epoch 3/20
10/10 [==============================] - 6s 622ms/step - loss: 0.3088 - acc: 0.
8696 - val_loss: 0.2301 - val_acc: 0.9388
Epoch 4/20
10/10 [==============================] - 6s 626ms/step - loss: 0.2177 - acc: 0.
9231 - val_loss: 0.1866 - val_acc: 0.9592
Epoch 5/20
10/10 [==============================] - 6s 626ms/step - loss: 0.1614 - acc: 0.
9498 - val_loss: 0.1630 - val_acc: 0.9796
Epoch 6/20
10/10 [==============================] - 6s 628ms/step - loss: 0.1468 - acc: 0.
9498 - val_loss: 0.1850 - val_acc: 0.9796
Epoch 7/20
10/10 [==============================] - 7s 660ms/step - loss: 0.1350 - acc: 0.
9532 - val_loss: 0.1788 - val_acc: 0.9592
Epoch 8/20
10/10 [==============================] - 6s 632ms/step - loss: 0.1857 - acc: 0.
9331 - val_loss: 0.1689 - val_acc: 0.9388
Epoch 9/20
10/10 [==============================] - 7s 664ms/step - loss: 0.1238 - acc: 0.
9331 - val_loss: 0.1449 - val_acc: 0.9796
Epoch 10/20
10/10 [==============================] - 7s 679ms/step - loss: 0.0894 - acc: 0.
9666 - val_loss: 0.0984 - val_acc: 0.9796
Epoch 11/20
10/10 [==============================] - 7s 664ms/step - loss: 0.0658 - acc: 0.
9866 - val_loss: 0.0875 - val_acc: 0.9796
Epoch 12/20
10/10 [==============================] - 7s 672ms/step - loss: 0.0442 - acc: 0.
9833 - val_loss: 0.1636 - val_acc: 0.9796
Epoch 13/20
10/10 [==============================] - 7s 674ms/step - loss: 0.0902 - acc: 0.
9699 - val_loss: 0.0586 - val_acc: 0.9796
Epoch 14/20
10/10 [==============================] - 7s 694ms/step - loss: 0.0658 - acc: 0.
9866 - val_loss: 0.0788 - val_acc: 0.9796
Epoch 15/20
10/10 [==============================] - 7s 664ms/step - loss: 0.0327 - acc: 0.
9967 - val_loss: 0.0241 - val_acc: 0.9796
Epoch 16/20
10/10 [==============================] - 6s 640ms/step - loss: 0.0083 - acc: 1.
0000 - val_loss: 0.0020 - val_acc: 1.0000
Epoch 17/20
10/10 [==============================] - 6s 631ms/step - loss: 0.0047 - acc: 1.
0000 - val_loss: 0.0355 - val_acc: 0.9796
Epoch 18/20
10/10 [==============================] - 6s 628ms/step - loss: 0.0154 - acc: 0.
9967 - val_loss: 0.0068 - val_acc: 1.0000
Epoch 19/20
```

```
10/10 [==============================] - 6s 625ms/step - loss: 0.0116 - acc: 1.
0000 - val_loss: 0.0178 - val_acc: 0.9796
Epoch 20/20
10/10 [==============================] - 6s 630ms/step - loss: 0.0131 - acc: 0.
9967 - val_loss: 0.1250 - val_acc: 0.9796
```

In [47]:
```
train_generator.class_indices
```

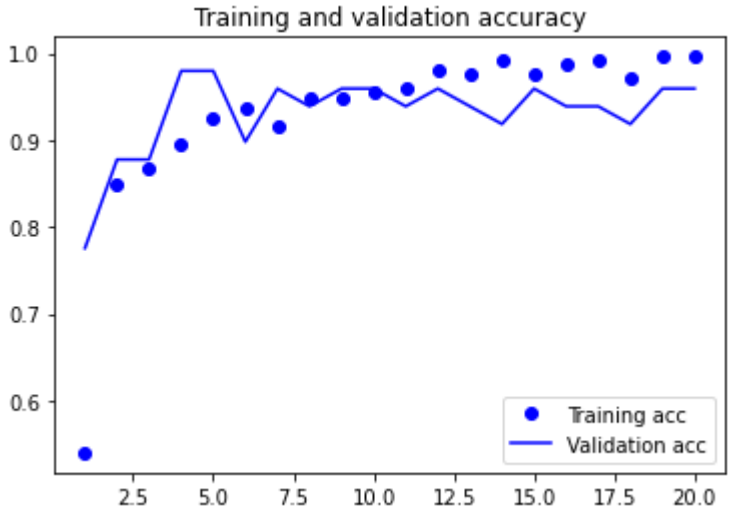
Out[47]:
```
{'sunflower': 0, 'tulip': 1}
```

In [46]:
```
train_generator.labels
```

Out[46]:
```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1])
```

In [72]:
```
import matplotlib.pyplot as plt
acc = history.history['acc']
val_acc = history.history['val_acc']
loss = history.history['loss']
val_loss = history.history['val_loss']
epochs = range(1, len(acc) + 1)
plt.plot(epochs, acc, 'bo', label='Training acc')
plt.plot(epochs, val_acc, 'b', label='Validation acc')
plt.title('Training and validation accuracy')
plt.legend()
plt.figure()
# plt.plot(epochs, loss, 'bo', label='Training loss')
# plt.plot(epochs, val_loss, 'b', label='Validation loss')
# plt.title('Training and validation loss')
# plt.legend()
# plt.show()
```

Out[72]:
```
<Figure size 432x288 with 0 Axes>
```




```
<Figure size 432x288 with 0 Axes>
```

In [97]:
```
test_image=cv2.imread('test_image2.jpg')
test_image=cv2.resize(test_image,(150,150))/255
test_image=np.expand_dims(test_image,axis=0)
```

In [98]:
```
model.predict(test_image)
model.predict_classes(test_image)
```

Out[98]:
```
array([1], dtype=int64)
```