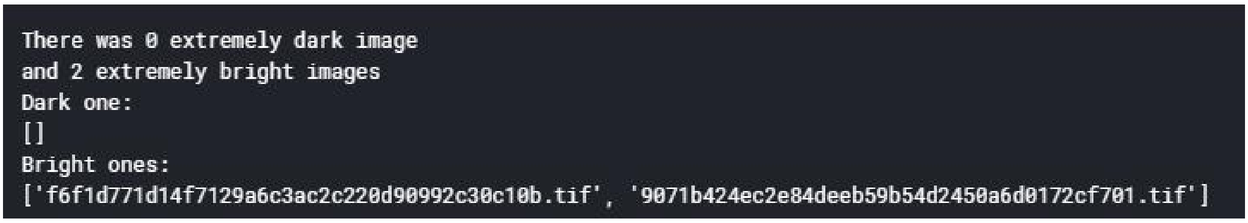
抽取10000张图片

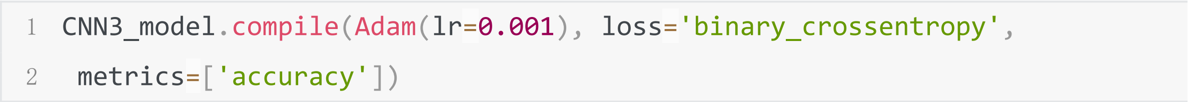
验证集与训练集：1：9

剔除黑和白的图片：



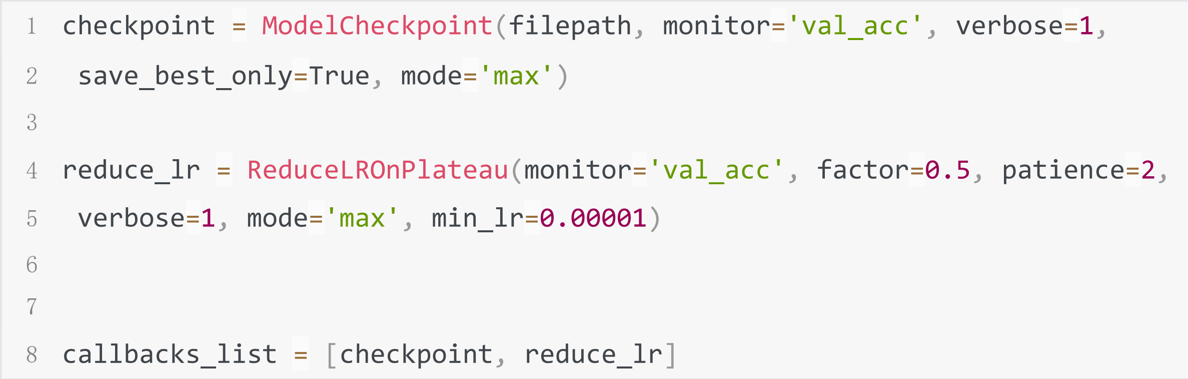
```
There was 0 extremely dark image
and 2 extremely bright images
Dark one:
[]
Bright ones:
['f6f1d771d14f7129a6c3ac2c220d90992c30c10b.tif', '9071b424ec2e84deeb59b54d2450a6d0172cf701.tif']
```

编译器



```
CNN3_model.compile(Adam(lr=0.001), loss='binary_crossentropy',
 metrics=['accuracy'])
```

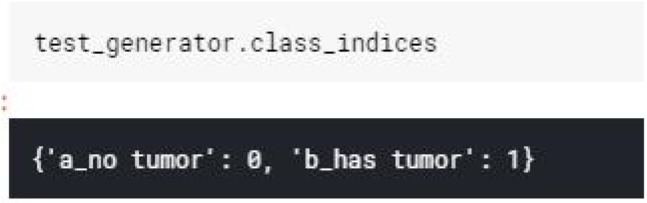
回调函数：



```
checkpoint = ModelCheckpoint(filepath, monitor='val_acc', verbose=1,
 save_best_only=True, mode='max')

reduce_lr = ReduceLROnPlateau(monitor='val_acc', factor=0.5, patience=2,
 verbose=1, mode='max', min_lr=0.00001)


callbacks_list = [checkpoint, reduce_lr]
```

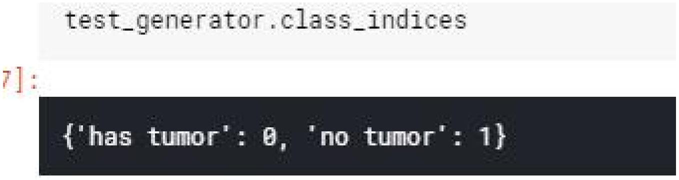
同样的训练集与测试集

为了让训练时的标签对应准确所以调整



```
test_generator.class_indices
```

```
:
{'a_no tumor': 0, 'b_has tumor': 1}
```

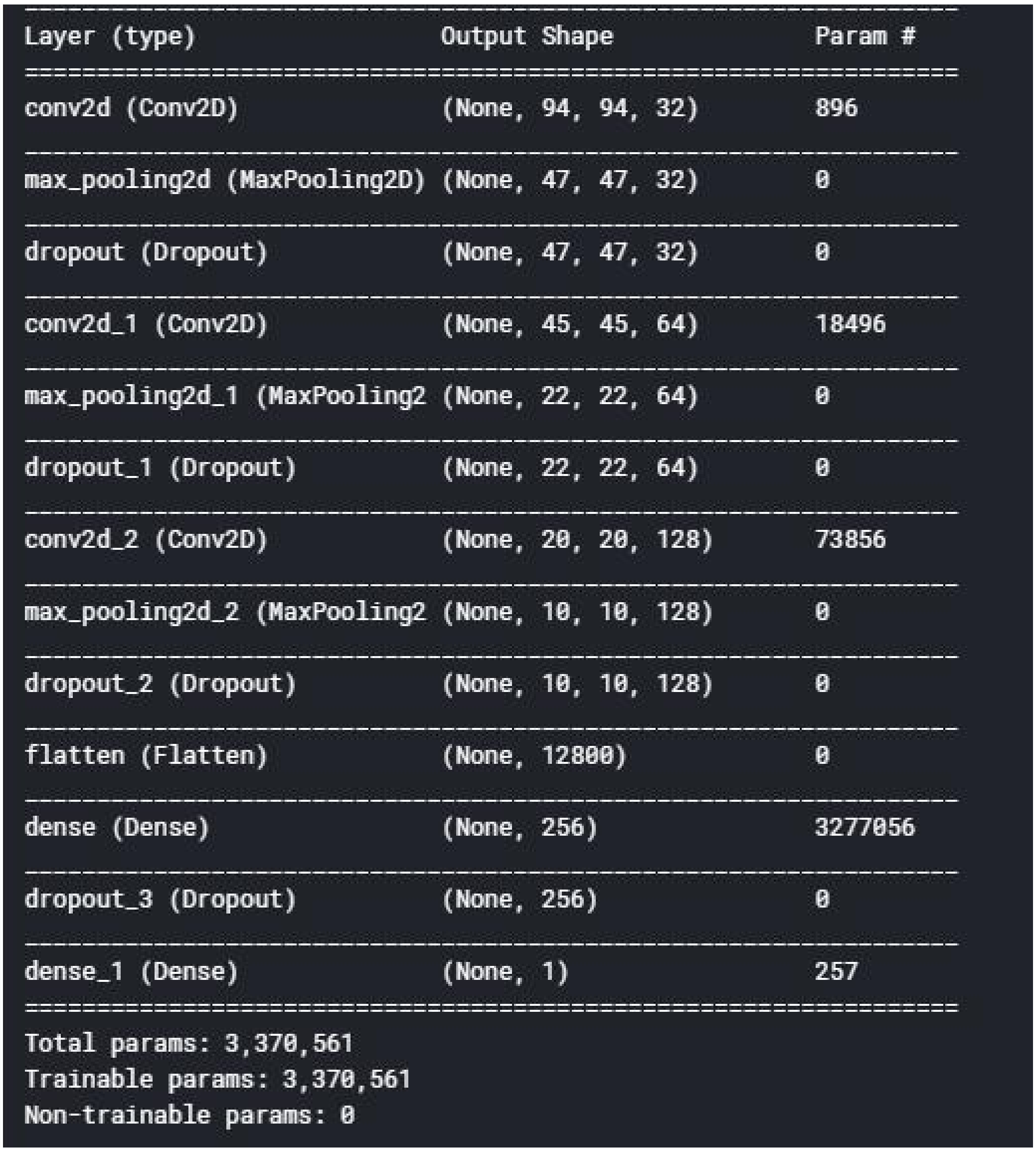


```
test_generator.class_indices
```

```
7]:
{'has tumor': 0, 'no tumor': 1}
```

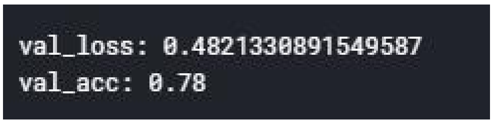
CNN3

模型结构

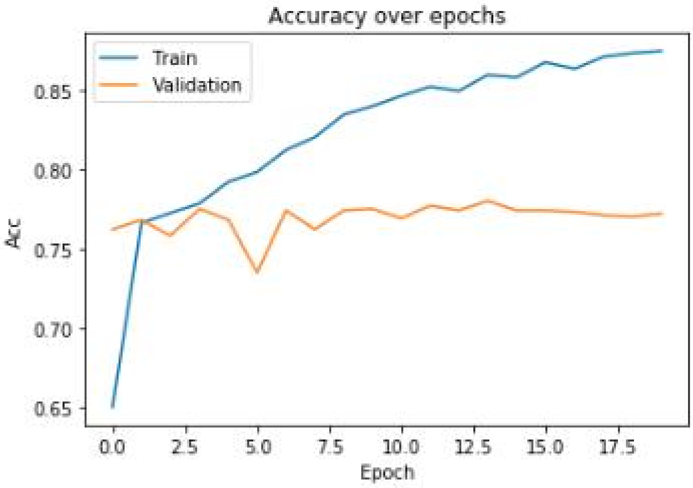


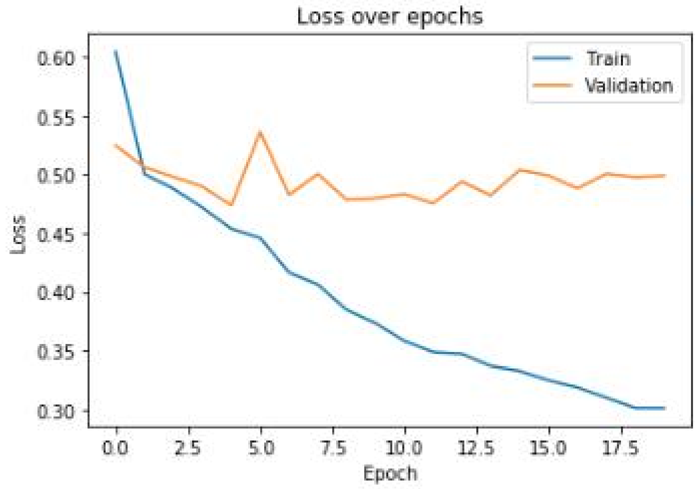
```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 94, 94, 32)        896
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 47, 47, 32)        0
_________________________________________________________________
dropout (Dropout)            (None, 47, 47, 32)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 45, 45, 64)        18496
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 22, 22, 64)        0
_________________________________________________________________
dropout_1 (Dropout)          (None, 22, 22, 64)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 20, 20, 128)       73856
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 10, 10, 128)       0
_________________________________________________________________
dropout_2 (Dropout)          (None, 10, 10, 128)       0
_________________________________________________________________
flatten (Flatten)            (None, 12800)             0
_________________________________________________________________
dense (Dense)                (None, 256)               3277056
_________________________________________________________________
dropout_3 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 257
=================================================================
Total params: 3,370,561
Trainable params: 3,370,561
Non-trainable params: 0
```

结果：

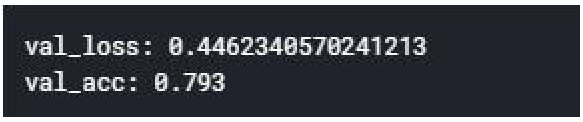


```
val_loss: 0.4821330891549587
val_acc: 0.78
```

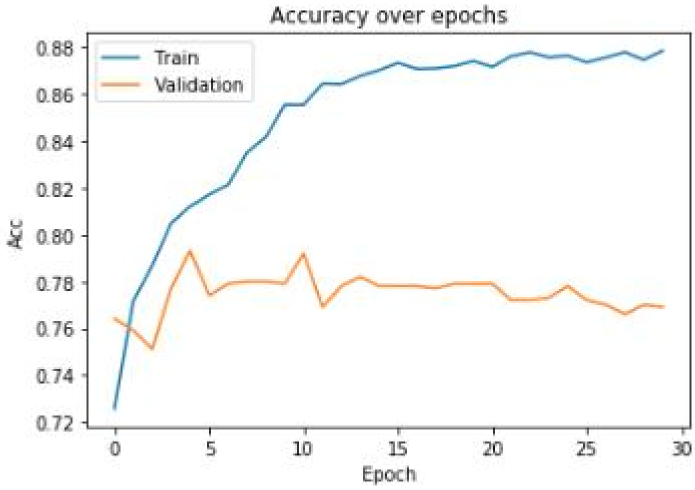


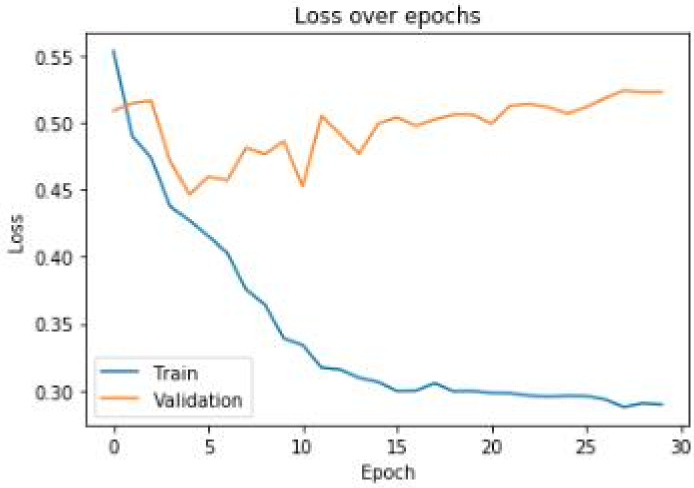


经过分析我们发现loss没有收敛，所以要加大epoch的数目，在这里我以为需要删除CNN3_model.h5然后再重新开始，会从最初的开始，但发现模型的loss值并非为最初的所以，故可以知道并没有从最初的状态开始，所以我明白了要重新走一遍流程CNN3的建立，然后loss值终于从0.7附近开始！验证成功！



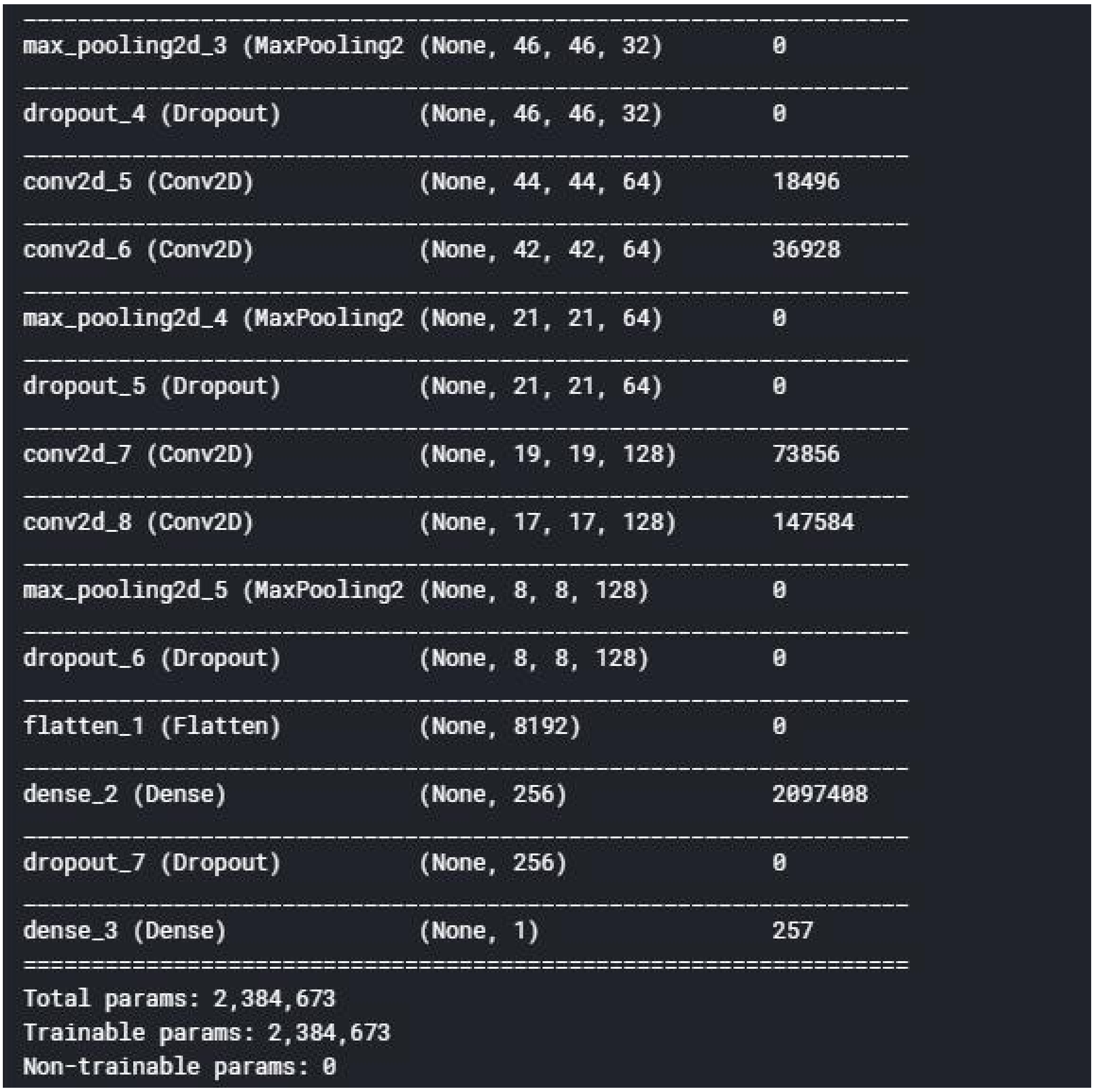
```
val_loss: 0.4462340570241213
val_acc: 0.793
```





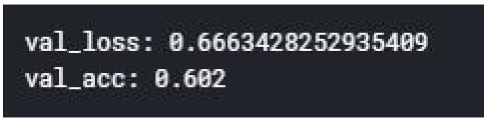
CNN6

模型结构

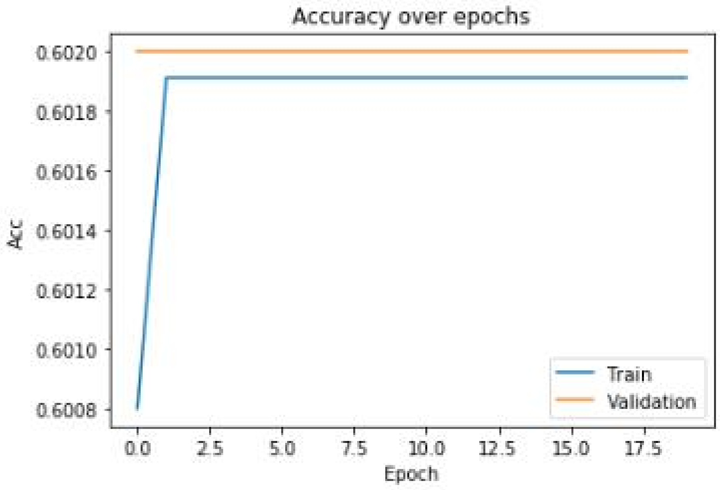


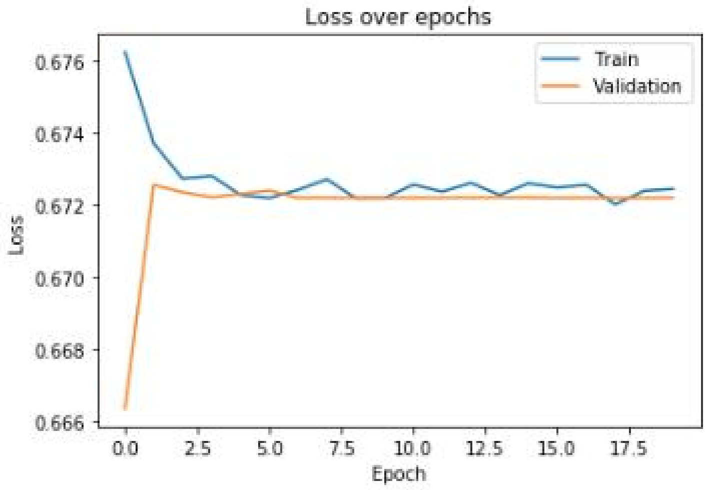
```
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 46, 46, 32)        0
_________________________________________________________________
dropout_4 (Dropout)          (None, 46, 46, 32)        0
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 44, 44, 64)        18496
_________________________________________________________________
conv2d_6 (Conv2D)            (None, 42, 42, 64)        36928
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 21, 21, 64)        0
_________________________________________________________________
dropout_5 (Dropout)          (None, 21, 21, 64)        0
_________________________________________________________________
conv2d_7 (Conv2D)            (None, 19, 19, 128)       73856
_________________________________________________________________
conv2d_8 (Conv2D)            (None, 17, 17, 128)       147584
_________________________________________________________________
max_pooling2d_5 (MaxPooling2 (None, 8, 8, 128)         0
_________________________________________________________________
dropout_6 (Dropout)          (None, 8, 8, 128)         0
_________________________________________________________________
flatten_1 (Flatten)          (None, 8192)              0
_________________________________________________________________
dense_2 (Dense)              (None, 256)               2097408
_________________________________________________________________
dropout_7 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_3 (Dense)              (None, 1)                 257
=================================================================
Total params: 2,384,673
Trainable params: 2,384,673
Non-trainable params: 0
```

结果



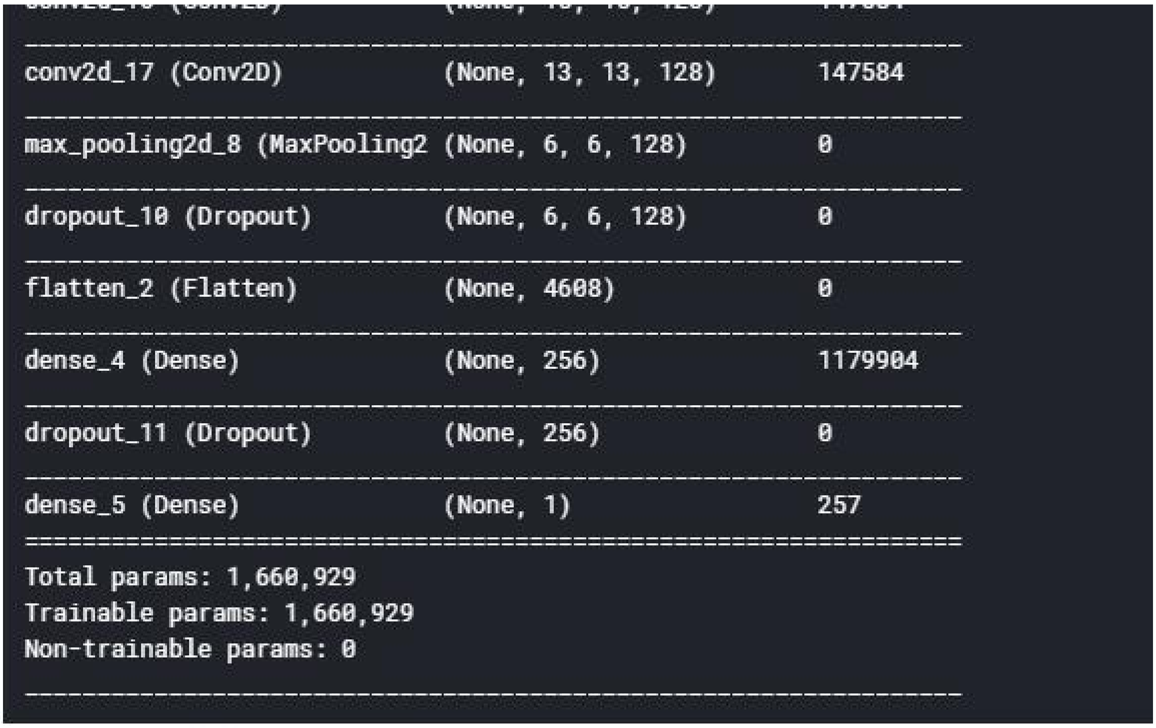
```
val_loss: 0.6663428252935409
val_acc: 0.602
```





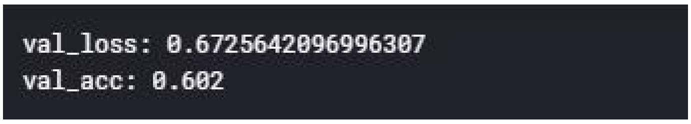
CNN9

网络结构：

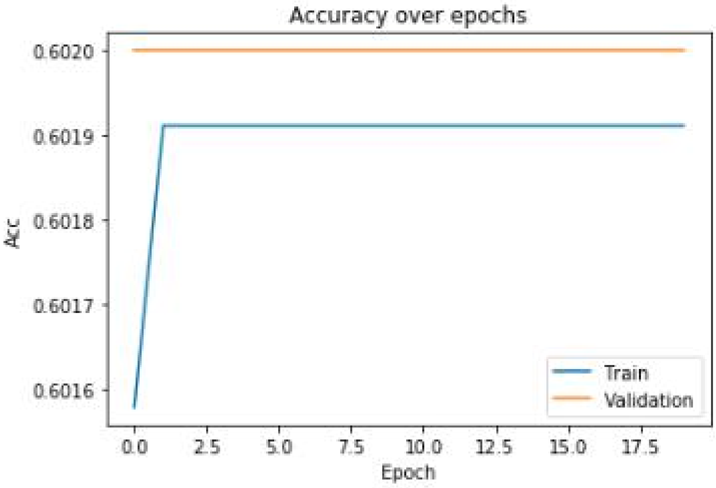


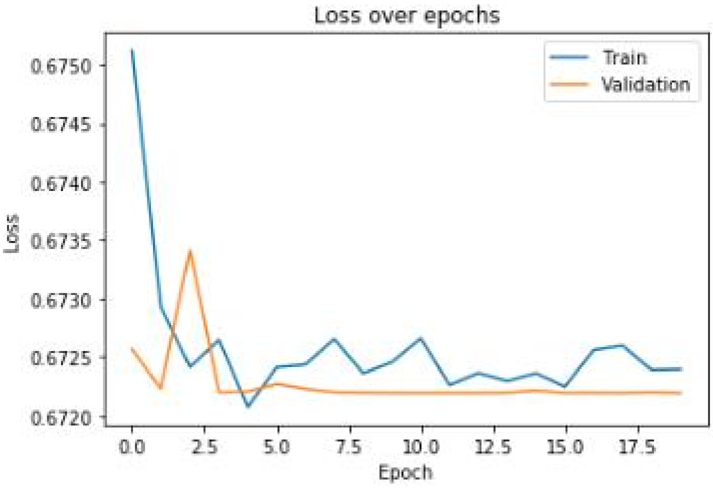
```
_________________________________________________________________
conv2d_17 (Conv2D)           (None, 13, 13, 128)       147584    
_________________________________________________________________
max_pooling2d_8 (MaxPooling2 (None, 6, 6, 128)         0         
_________________________________________________________________
dropout_10 (Dropout)         (None, 6, 6, 128)         0         
_________________________________________________________________
flatten_2 (Flatten)          (None, 4608)              0         
_________________________________________________________________
dense_4 (Dense)              (None, 256)               1179904   
_________________________________________________________________
dropout_11 (Dropout)         (None, 256)               0         
_________________________________________________________________
dense_5 (Dense)              (None, 1)                 257       
=================================================================
Total params: 1,660,929
Trainable params: 1,660,929
Non-trainable params: 0
_________________________________________________________________
```

结果：

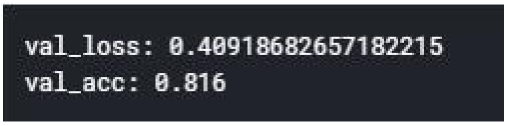


```
val_loss: 0.6725642096996307
val_acc: 0.602
```

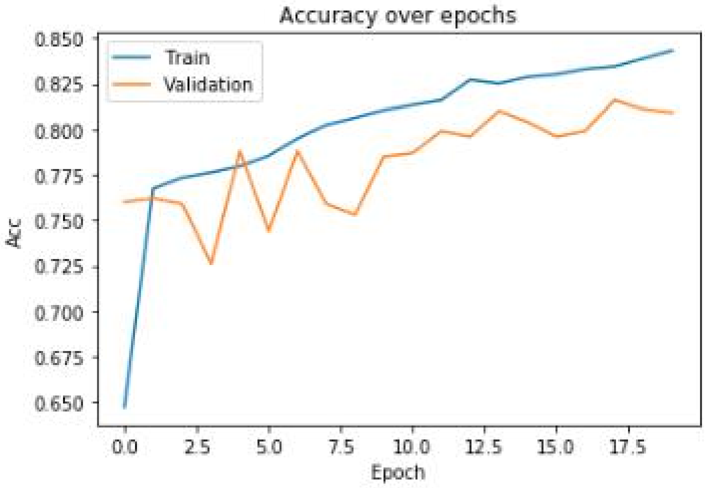


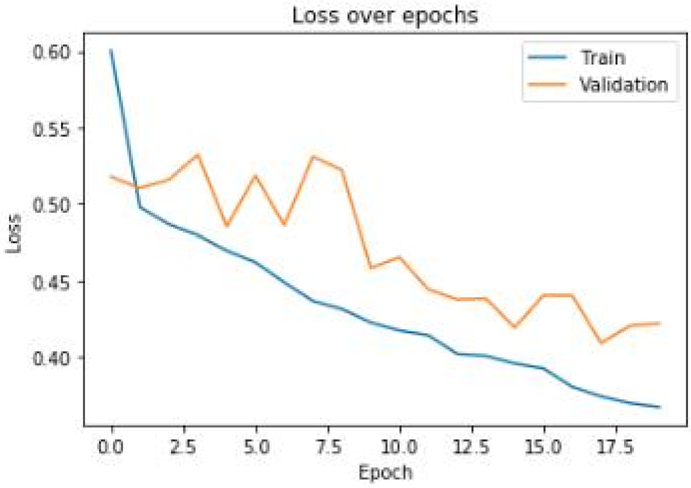


调整了学习率为0.0001，结果：

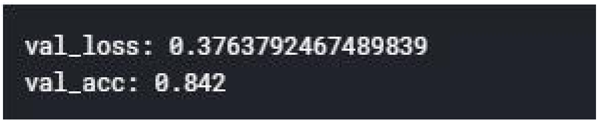


```
val_loss: 0.40918682657182215
val_acc: 0.816
```

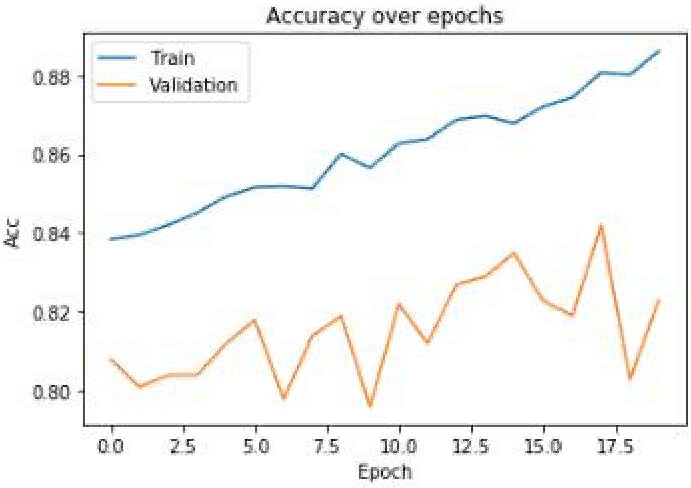


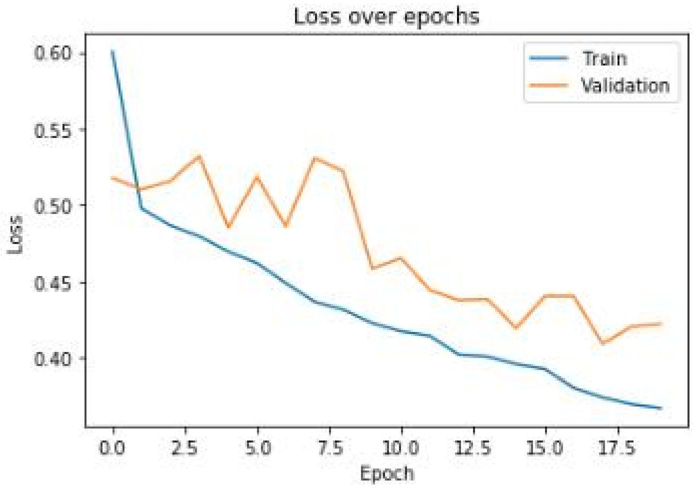


没有收敛，加大epoch



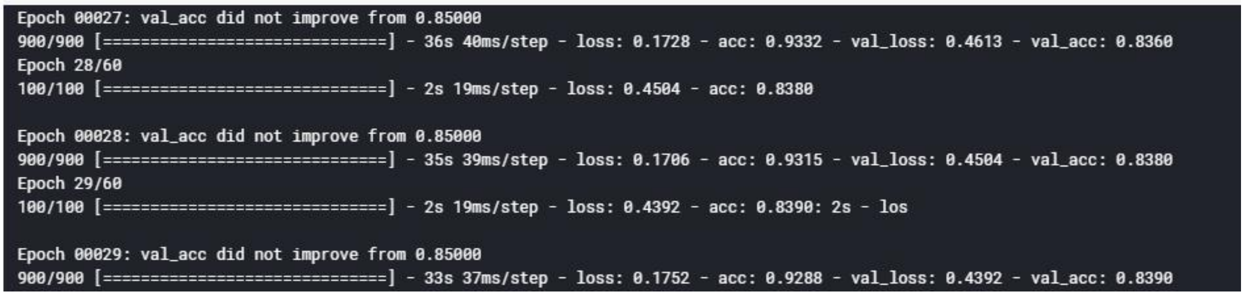
```
val_loss: 0.3763792467489839
val_acc: 0.842
```





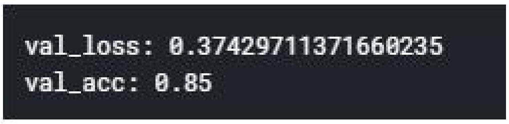
还没收敛，调整min_lr=0.00001，增大epoch继续训练

卡了，但是已经收敛了

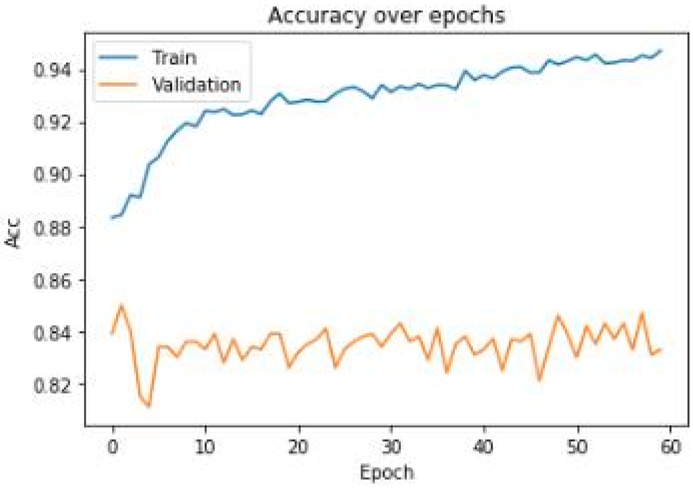


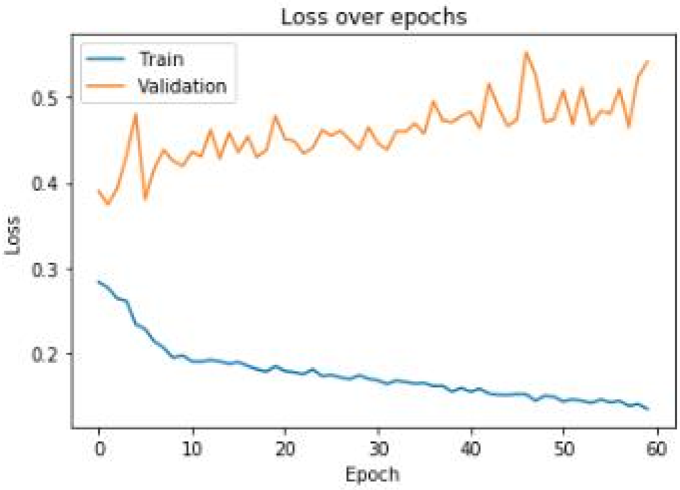
```
Epoch 00027: val_acc did not improve from 0.85000
900/900 [==============================] - 36s 40ms/step - loss: 0.1728 - acc: 0.9332 - val_loss: 0.4613 - val_acc: 0.8360
Epoch 28/60
100/100 [==============================] - 2s 19ms/step - loss: 0.4504 - acc: 0.8380

Epoch 00028: val_acc did not improve from 0.85000
900/900 [==============================] - 35s 39ms/step - loss: 0.1706 - acc: 0.9315 - val_loss: 0.4504 - val_acc: 0.8380
Epoch 29/60
100/100 [==============================] - 2s 19ms/step - loss: 0.4392 - acc: 0.8390: 2s - los

Epoch 00029: val_acc did not improve from 0.85000
900/900 [==============================] - 33s 37ms/step - loss: 0.1752 - acc: 0.9288 - val_loss: 0.4392 - val_acc: 0.8390
```

结果最高准确率不变



```
val_loss: 0.37429711371660235
val_acc: 0.85
```

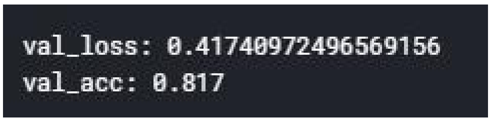




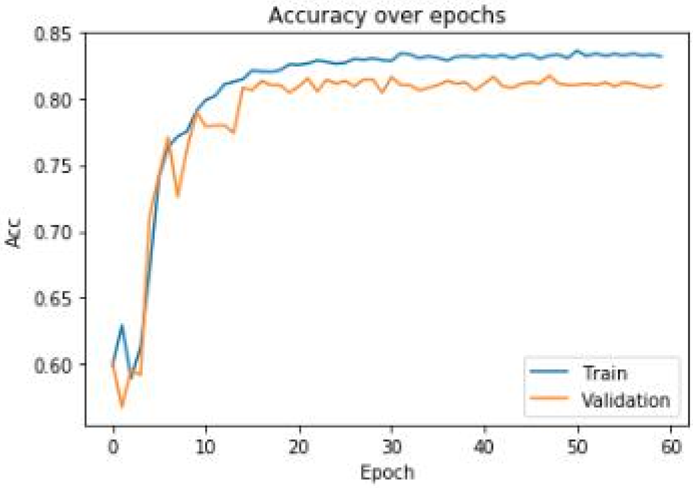
研究最好的结果是否和优化器相关

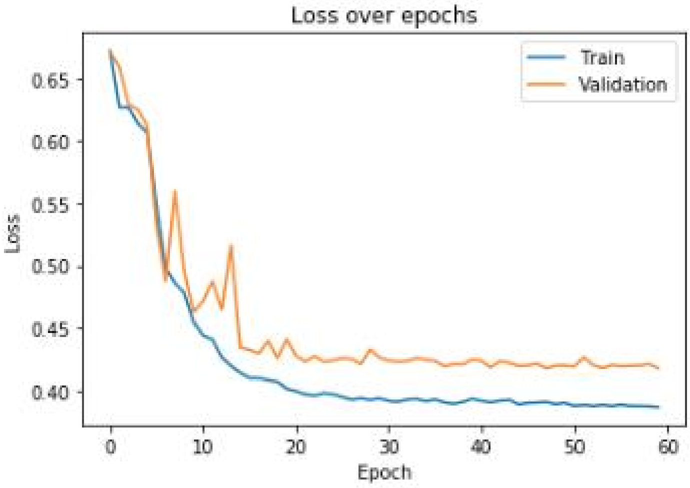
在这里改变优化器：

```
sgd = keras.optimizers.SGD(lr=0.001, decay=1e-6, momentum=0.9, nesterov=True)
CNN9_model.compile(loss='binary_crossentropy', optimizer=sgd, metrics=['accuracy'])
```

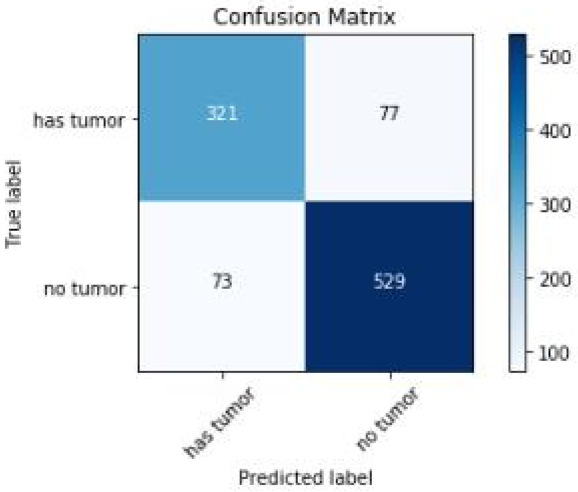


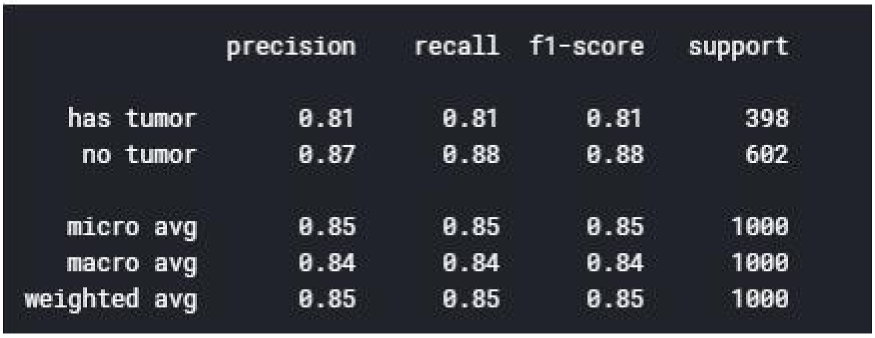
```
val_loss: 0.41740972496569156
val_acc: 0.817
```





几种解释一是落在鞍点，一是收敛的慢



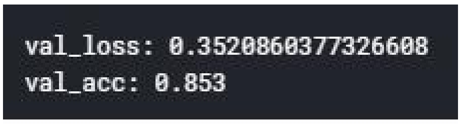


```
              precision    recall  f1-score   support

   has tumor       0.81      0.81      0.81       398
    no tumor       0.87      0.88      0.88       602

   micro avg       0.85      0.85      0.85      1000
   macro avg       0.84      0.84      0.84      1000
weighted avg       0.85      0.85      0.85      1000
```

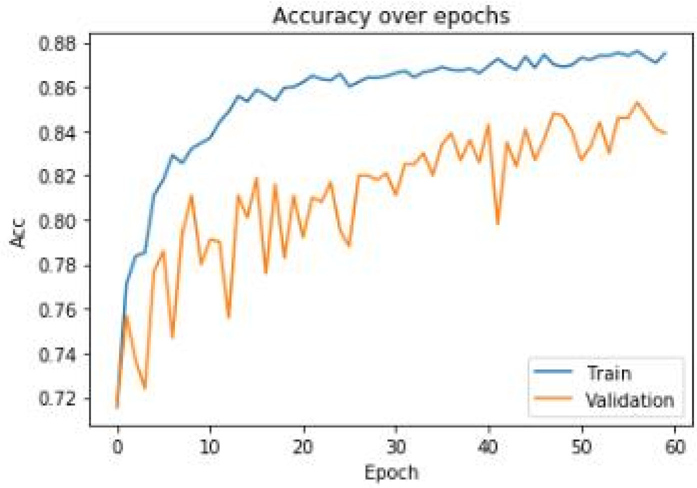
得出一个结论优化器确实会影响最后的结果!

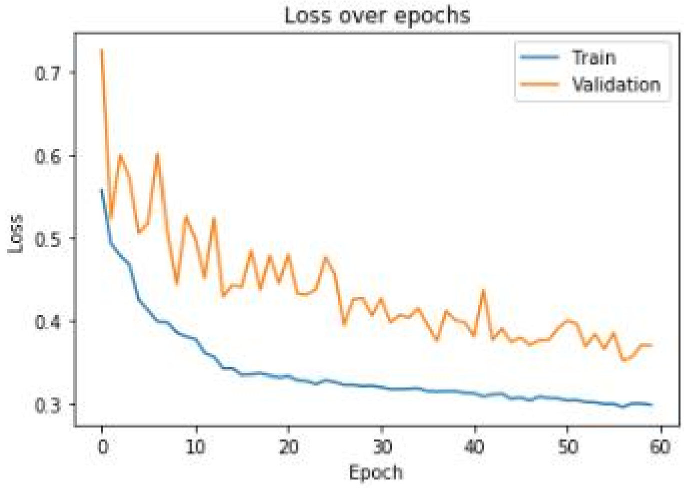
添加图片增强

```
train_datagen = ImageDataGenerator(
 rescale=1. / 255,
 horizontal_flip=True,
 vertical_flip=True,
 rotation_range=40,
# zoom_range=0.2,
# width_shift_range=0.1,
# height_shift_range=0.1
)


```



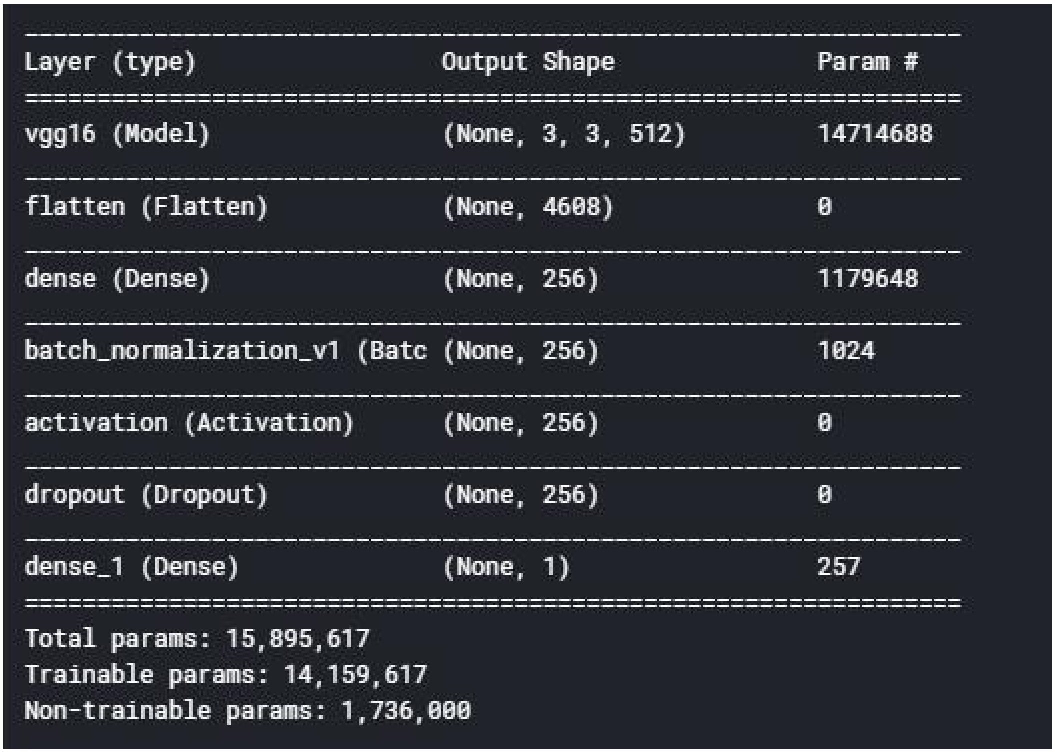
```
val_loss: 0.3520860377326608
val_acc: 0.853
```



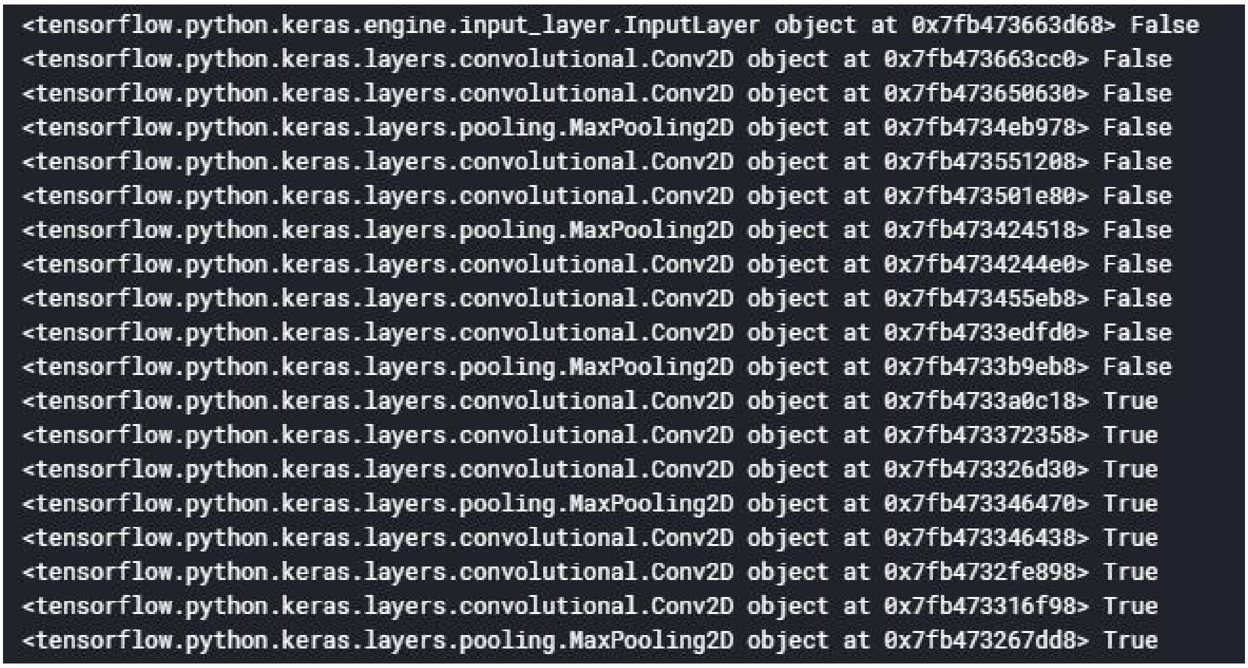


VGG16

网络结构

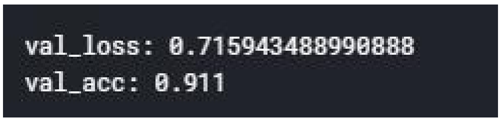


```
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
vgg16 (Model)                (None, 3, 3, 512)         14714688  
_________________________________________________________________
flatten (Flatten)            (None, 4608)              0         
_________________________________________________________________
dense (Dense)                (None, 256)               1179648   
_________________________________________________________________
batch_normalization_v1 (Batc (None, 256)               1024      
_________________________________________________________________
activation (Activation)      (None, 256)               0         
_________________________________________________________________
dropout (Dropout)            (None, 256)               0         
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 257       
=================================================================
Total params: 15,895,617
Trainable params: 14,159,617
Non-trainable params: 1,736,000
```

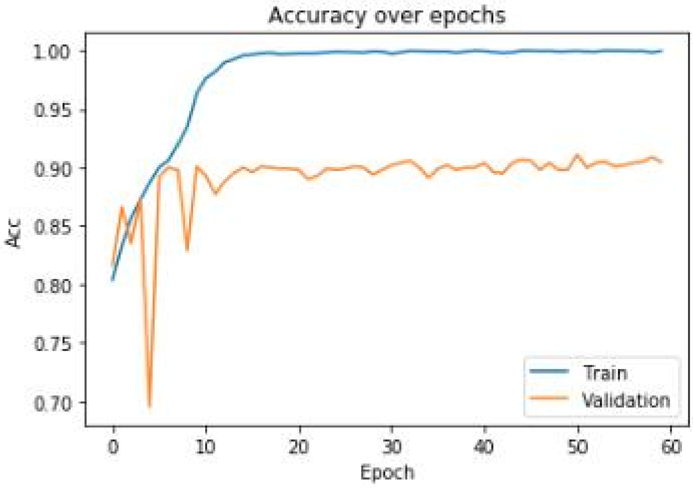


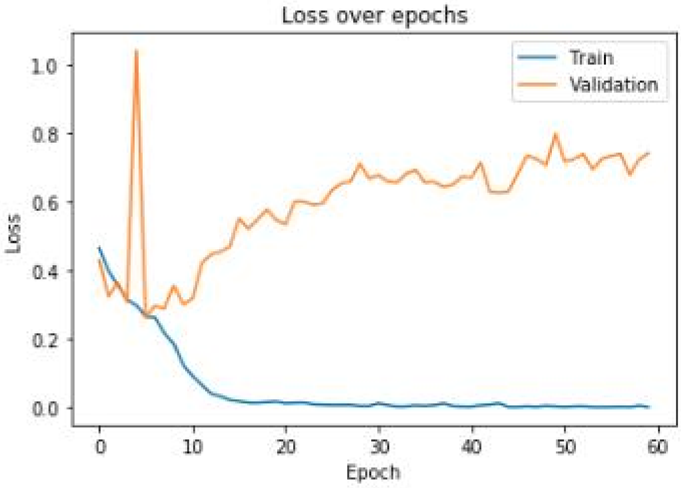
```
<tensorflow.python.keras.engine.input_layer.InputLayer object at 0x7fb473663d68> False
<tensorflow.python.keras.layers.convolutional.Conv2D object at 0x7fb473663cc0> False
<tensorflow.python.keras.layers.convolutional.Conv2D object at 0x7fb473650630> False
<tensorflow.python.keras.layers.pooling.MaxPooling2D object at 0x7fb4734eb978> False
<tensorflow.python.keras.layers.convolutional.Conv2D object at 0x7fb473551208> False
<tensorflow.python.keras.layers.convolutional.Conv2D object at 0x7fb473501e80> False
<tensorflow.python.keras.layers.pooling.MaxPooling2D object at 0x7fb473424518> False
<tensorflow.python.keras.layers.convolutional.Conv2D object at 0x7fb4734244e0> False
<tensorflow.python.keras.layers.convolutional.Conv2D object at 0x7fb473455eb8> False
<tensorflow.python.keras.layers.convolutional.Conv2D object at 0x7fb4733edfd0> False
<tensorflow.python.keras.layers.pooling.MaxPooling2D object at 0x7fb4733b9eb8> False
<tensorflow.python.keras.layers.convolutional.Conv2D object at 0x7fb4733a0c18> True
<tensorflow.python.keras.layers.convolutional.Conv2D object at 0x7fb473372358> True
<tensorflow.python.keras.layers.convolutional.Conv2D object at 0x7fb473326d30> True
<tensorflow.python.keras.layers.pooling.MaxPooling2D object at 0x7fb473346470> True
<tensorflow.python.keras.layers.convolutional.Conv2D object at 0x7fb473346438> True
<tensorflow.python.keras.layers.convolutional.Conv2D object at 0x7fb4732fe898> True
<tensorflow.python.keras.layers.convolutional.Conv2D object at 0x7fb473316f98> True
<tensorflow.python.keras.layers.pooling.MaxPooling2D object at 0x7fb473267dd8> True
```

结果：



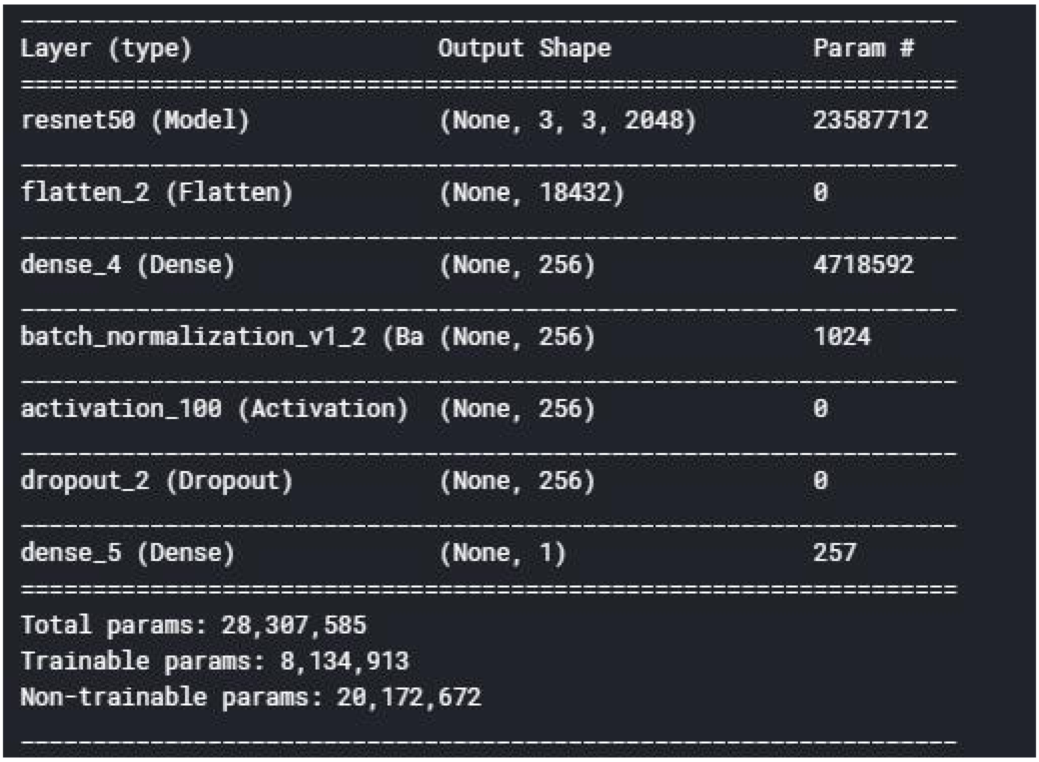
```
val_loss: 0.715943488990888
val_acc: 0.911
```





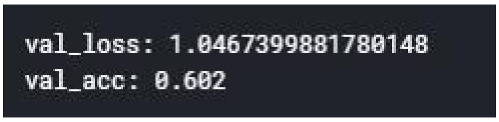
RESNET50

网络结构

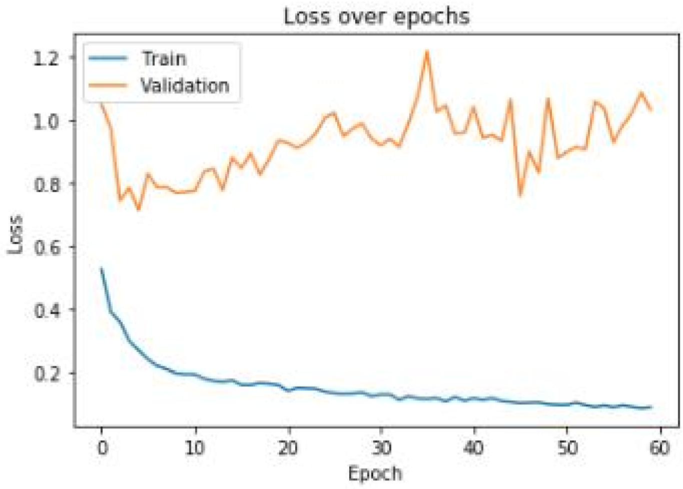


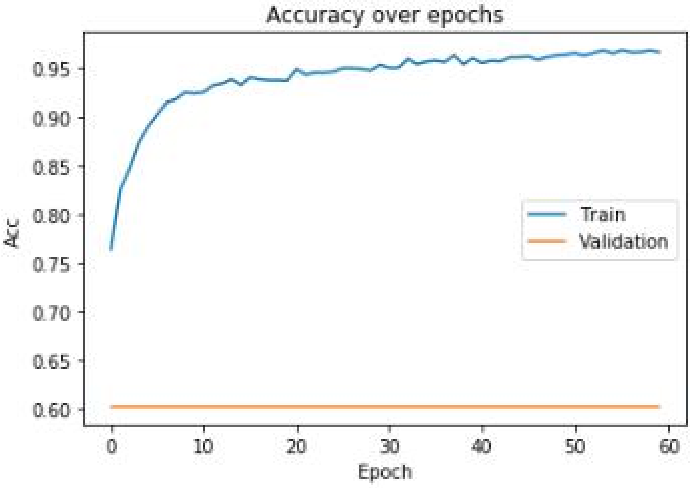
```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
resnet50 (Model)             (None, 3, 3, 2048)        23587712
_________________________________________________________________
flatten_2 (Flatten)          (None, 18432)             0
_________________________________________________________________
dense_4 (Dense)              (None, 256)               4718592
_________________________________________________________________
batch_normalization_v1_2 (Ba (None, 256)               1024
_________________________________________________________________
activation_100 (Activation)  (None, 256)               0
_________________________________________________________________
dropout_2 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_5 (Dense)              (None, 1)                 257
=================================================================
Total params: 28,307,585
Trainable params: 8,134,913
Non-trainable params: 20,172,672
_________________________________________________________________
```

最后结果



```
val_loss: 1.0467399881780148
val_acc: 0.602
```




Accuracy over epochs
Acc
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0.60
0
10
20
30
40
50
60
Epoch
Train
Validation