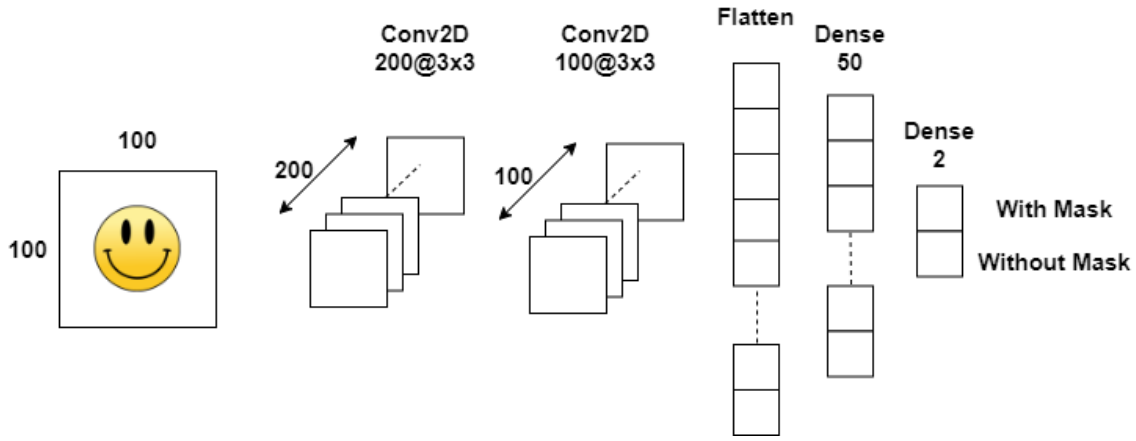
Convolutional Neural Network Architecture



In [17]:
```
import numpy as np

data=np.load('data.npy')
target=np.load('target.npy')

#loading the save numpy arrays in the previous code
```

In [18]:
```
from keras.models import Sequential
from keras.layers import Dense,Activation,Flatten,Dropout
from keras.layers import Conv2D,MaxPooling2D
from keras.callbacks import ModelCheckpoint

model=Sequential()

#The first CNN layer followed by Relu and MaxPooling layers
model.add(Conv2D(200,(3,3),input_shape=data.shape[1:]))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2,2)))


#The second convolution layer followed by Relu and MaxPooling layers
model.add(Conv2D(100,(3,3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2,2)))

#Flatten layer to stack the output convolutions from second convolution laye
model.add(Flatten())
model.add(Dropout(0.5))

#Dense layer of 64 neurons
model.add(Dense(50,activation='relu'))

#The Final layer with two outputs for two categories
model.add(Dense(2,activation='softmax'))
```

```
model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['acc
```

In [19]:
```
from sklearn.model_selection import train_test_split

train_data,test_data,train_target,test_target=train_test_split(data,target,t
```

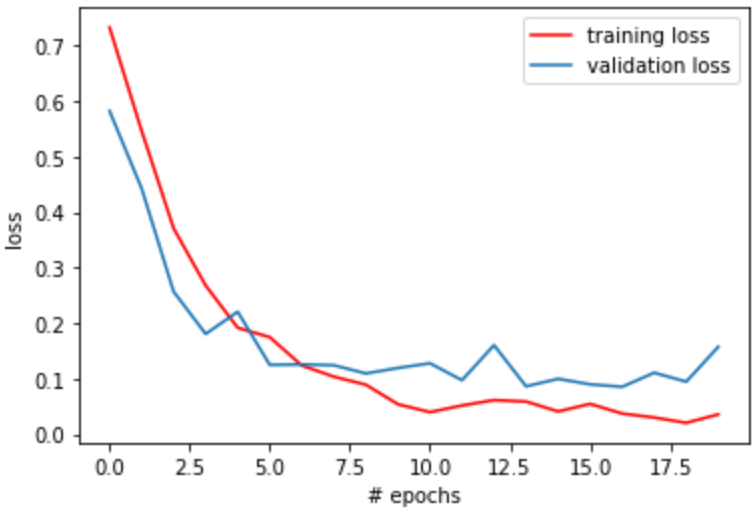
In [20]:
```
checkpoint = ModelCheckpoint('model-{epoch:03d}.model',monitor='val_loss',ve
history=model.fit(train_data,train_target,epochs=20,callbacks=[checkpoint],v
```

```
Train on 990 samples, validate on 248 samples
Epoch 1/20
990/990 [==============================] - 93s 94ms/step - loss: 0.7326 - ac
curacy: 0.5626 - val_loss: 0.5822 - val_accuracy: 0.6290
Epoch 2/20
990/990 [==============================] - 93s 94ms/step - loss: 0.5465 - ac
curacy: 0.7253 - val_loss: 0.4429 - val_accuracy: 0.8185
Epoch 3/20
990/990 [==============================] - 93s 94ms/step - loss: 0.3708 - ac
curacy: 0.8354 - val_loss: 0.2568 - val_accuracy: 0.9032
Epoch 4/20
990/990 [==============================] - 95s 96ms/step - loss: 0.2679 - ac
curacy: 0.8970 - val_loss: 0.1807 - val_accuracy: 0.9476
Epoch 5/20
990/990 [==============================] - 93s 94ms/step - loss: 0.1917 - ac
curacy: 0.9303 - val_loss: 0.2207 - val_accuracy: 0.9315
Epoch 6/20
990/990 [==============================] - 93s 94ms/step - loss: 0.1749 - ac
curacy: 0.9343 - val_loss: 0.1249 - val_accuracy: 0.9597
Epoch 7/20
990/990 [==============================] - 95s 96ms/step - loss: 0.1238 - ac
curacy: 0.9576 - val_loss: 0.1258 - val_accuracy: 0.9637
Epoch 8/20
990/990 [==============================] - 94s 95ms/step - loss: 0.1037 - ac
curacy: 0.9616 - val_loss: 0.1243 - val_accuracy: 0.9516
Epoch 9/20
990/990 [==============================] - 94s 95ms/step - loss: 0.0893 - ac
curacy: 0.9687 - val_loss: 0.1095 - val_accuracy: 0.9556
Epoch 10/20
990/990 [==============================] - 94s 95ms/step - loss: 0.0540 - ac
curacy: 0.9828 - val_loss: 0.1193 - val_accuracy: 0.9597
Epoch 11/20
990/990 [==============================] - 92s 93ms/step - loss: 0.0399 - ac
curacy: 0.9899 - val_loss: 0.1278 - val_accuracy: 0.9677
Epoch 12/20
990/990 [==============================] - 93s 94ms/step - loss: 0.0518 - ac
curacy: 0.9818 - val_loss: 0.0974 - val_accuracy: 0.9718
Epoch 13/20
990/990 [==============================] - 96s 97ms/step - loss: 0.0615 - ac
curacy: 0.9778 - val_loss: 0.1604 - val_accuracy: 0.9274
Epoch 14/20
990/990 [==============================] - 97s 98ms/step - loss: 0.0589 - ac
curacy: 0.9828 - val_loss: 0.0863 - val_accuracy: 0.9597
Epoch 15/20
990/990 [==============================] - 94s 95ms/step - loss: 0.0411 - ac
curacy: 0.9808 - val_loss: 0.0998 - val_accuracy: 0.9677
Epoch 16/20
990/990 [==============================] - 81s 82ms/step - loss: 0.0547 - ac
curacy: 0.9747 - val_loss: 0.0899 - val_accuracy: 0.9556
Epoch 17/20
990/990 [==============================] - 79s 80ms/step - loss: 0.0372 - ac
curacy: 0.9889 - val_loss: 0.0855 - val_accuracy: 0.9637
Epoch 18/20
990/990 [==============================] - 78s 79ms/step - loss: 0.0301 - ac
curacy: 0.9879 - val_loss: 0.1107 - val_accuracy: 0.9556
Epoch 19/20
```

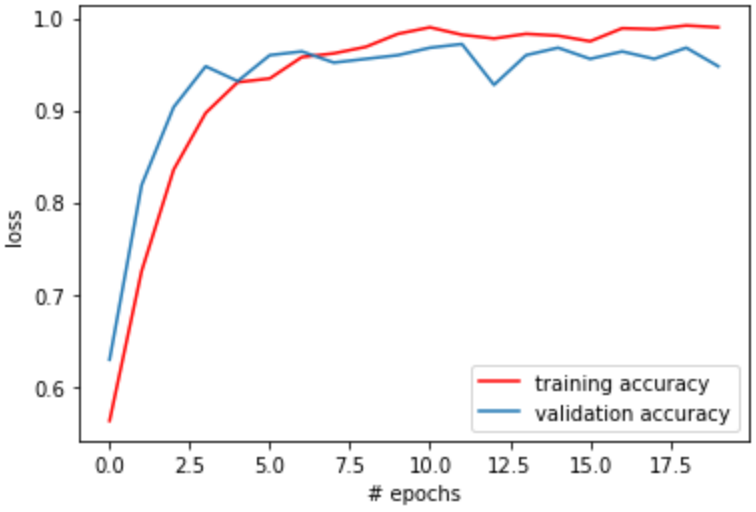
```
990/990 [==============================] - 89s 90ms/step - loss: 0.0206 - ac
curacy: 0.9919 - val_loss: 0.0947 - val_accuracy: 0.9677
Epoch 20/20
990/990 [==============================] - 94s 95ms/step - loss: 0.0358 - ac
curacy: 0.9899 - val_loss: 0.1575 - val_accuracy: 0.9476
```

In [21]:
```
from matplotlib import pyplot as plt

plt.plot(history.history['loss'],'r',label='training loss')
plt.plot(history.history['val_loss'],label='validation loss')
plt.xlabel('# epochs')
plt.ylabel('loss')
plt.legend()
plt.show()
```



In [22]:
```
plt.plot(history.history['accuracy'],'r',label='training accuracy')
plt.plot(history.history['val_accuracy'],label='validation accuracy')
plt.xlabel('# epochs')
plt.ylabel('loss')
plt.legend()
plt.show()
```



In [23]:
```
print(model.evaluate(test_data,test_target))
```

```
138/138 [==============================] - 6s 44ms/step
[0.14019694376358952, 0.9637681245803833]
```