# Test 4

**95 % accuracy**

In [81]:

*# Importing all necessary libraries*

**from** keras.preprocessing.image **import** ImageDataGenerator

**from** keras.models **import** Sequential

**from** matplotlib **import** pyplot

**import** matplotlib.pyplot **as** plt

**from** keras.layers **import** Conv2D, MaxPooling2D

**from** sklearn.metrics **import** classification\_report, confusion\_matrix

**from** keras.layers **import** Activation, Dropout, Flatten, Dense

**from** keras **import** backend **as** K

**import** numpy **as** np

**from** keras.models **import** load\_model #1 load\_model to load the model

**import** keras

**from** keras.callbacks **import** ModelCheckpoint # 2 It record the best accuracy

*#Image Size*

img\_width, img\_height **=** 224, 224

*# Reading training data from the directory*

train\_data\_dir **=** '/content/drive/MyDrive/Car Vs Plan Data Sets/train'

*# Reading validation data from the directory*

validation\_data\_dir **=** '/content/drive/MyDrive/Car Vs Plan Data Sets/test'

*# Total train instances*

nb\_train\_samples **=**400

*# Total Test instances*

nb\_validation\_samples **=** 100

batch\_size **=** 8 #change batch size to so it can learn better

*# To check if the shape of input image is correct*

**if** K**.**image\_data\_format() **==** 'channels\_first': input\_shape **=** (3, img\_width, img\_height)

**else**:

input\_shape **=** (img\_width, img\_height, 3)

*# Defining Sequential CNN model*

model **=** Sequential()

model**.**add(Conv2D(32, (3,3), input\_shape**=**input\_shape)) model**.**add(Activation('relu'))

model**.**add(MaxPooling2D(pool\_size**=**(2, 2)))

model**.**add(Conv2D(32, (3, 3))) model**.**add(Activation('relu'))

model**.**add(MaxPooling2D(pool\_size**=**(2, 2)))

model**.**add(Conv2D(64, (3, 3))) model**.**add(Activation('relu'))

model**.**add(MaxPooling2D(pool\_size**=**(2, 2))) model**.**add(Flatten())

model**.**add(Dense(64))

model**.**add(Activation('relu')) model**.**add(Dropout(0.5))

model**.**add(Dense(1))

model**.**add(Activation('sigmoid'))

*# compile model*

model**.**compile(loss**=**'binary\_crossentropy', optimizer**=**'rmsprop',metrics**=**['accuracy'])

checkpoint **=** ModelCheckpoint("best\_model.hdf5", monitor**=**'val\_accuracy', verbose**=**1, save\_best\_only**=True**, mode**=**'auto', period**=**1)

*#cearting virtual datastore for training data set*

#Better Data Augmentation with roataion range shift range and parameter tuning

train\_datagen **=** ImageDataGenerator( rotation\_range**=**40,

width\_shift\_range**=**0.2, height\_shift\_range**=**0.2, rescale**=**1. **/** 255,

shear\_range**=**0.2, zoom\_range**=**0.3,

horizontal\_flip**=True**, fill\_mode**=**'nearest')

*#cearting virtual datastore for testing data set*

test\_datagen **=** ImageDataGenerator(rescale**=**1. **/** 255)

*# Fetching training data from folder*

train\_generator **=** train\_datagen**.**flow\_from\_directory( train\_data\_dir,

target\_size**=**(img\_width, img\_height), batch\_size**=**batch\_size,

class\_mode**=**'binary')

*# Fetching testing data from folder*

validation\_generator **=** test\_datagen**.**flow\_from\_directory( validation\_data\_dir,

target\_size**=**(img\_width, img\_height), batch\_size**=**batch\_size,

class\_mode**=**'binary')

*# Fitting model*

history **=** model**.**fit\_generator(train\_generator,steps\_per\_epoch**=**len(train\_generator),

validation\_data**=**validation\_generator,validation\_steps**=**len(validation\_generator), epochs model**.**save\_weights('first\_try.h5') *# 5 always save your weights after training or during*

*# list all data in history*

print(history**.**history**.**keys())

WARNING:tensorflow:`period` argument is deprecated. Please use `save\_freq` to specify th e frequency in number of batches seen.

Found 400 images belonging to 2 classes. Found 100 images belonging to 2 classes.

/usr/local/lib/python3.6/dist-packages/tensorflow/python/keras/engine/training.py:1844: UserWarning: `Model.fit\_generator` is deprecated and will be removed in a future versio

n. Please use `Model.fit`, which supports generators.

warnings.warn('`Model.fit\_generator` is deprecated and ' Epoch 1/50

Thank you for using [www.freepdfconvert.com](http://www.freepdfconvert.com/) service!

Only two pages are converted. Please Sign Up to convert all pages. <https://www.freepdfconvert.com/membership>