

```

import os
import h5py
import numpy as np
import json
import urllib.request
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
from IPython.display import Image, display, clear_output
from sklearn.metrics import classification_report, confusion_matrix

%matplotlib inline
sns.set_style('whitegrid')

```

In [2]:

```

from keras import optimizers
from keras.applications.vgg16 import VGG16
from keras.models import Sequential, load_model, Model
from keras.layers import Conv2D, MaxPooling2D, ZeroPadding2D, Activation,
Dropout, Flatten, Dense, Input
from keras.regularizers import l2, l1
from keras.utils.np_utils import to_categorical
from keras.preprocessing.image import ImageDataGenerator, array_to_img,
img_to_array, load_img
from keras.callbacks import ModelCheckpoint, History
from keras import backend as K
from keras.utils.data_utils import get_file
Using TensorFlow backend.

```

In [3]:

```

def plot_metrics(hist, stop=50):
    fig, axes = plt.subplots(nrows=1, ncols=2, figsize=(10,4))

    axes = axes.flatten()

    axes[0].plot(range(stop), hist['acc'], label='Training',
color='#FF533D')
    axes[0].plot(range(stop), hist['val_acc'], label='Validation',
color='#03507E')
    axes[0].set_title('Accuracy')
    axes[0].set_ylabel('Accuracy')
    axes[0].set_xlabel('Epoch')
    axes[0].legend(loc='lower right')

```

```

    axes[1].plot(range(stop), hist['loss'], label='Training',
color='#FF533D')
    axes[1].plot(range(stop), hist['val_loss'], label='Validation',
color='#03507E')
    axes[1].set_title('Loss')
    axes[1].set_ylabel('Loss')
    axes[1].set_xlabel('Epoch')
    axes[1].legend(loc='upper right')

plt.tight_layout();

print("Best Model:")
print_best_model_results(hist)

```

In [4]:

```

def plot_acc_metrics(hist1, hist2, stop=50):
    fig, axes = plt.subplots(nrows=2, ncols=1, figsize=(4.25, 6))

    axes = axes.flatten()

    axes[0].plot(range(stop), hist1['acc'], label='Training',
color='#FF533D')
    axes[0].plot(range(stop), hist1['val_acc'], label='Validation',
color='#03507E')
    axes[0].set_title('Training')
    axes[0].set_ylabel('Accuracy')
    axes[0].set_xlabel('Epoch')
    axes[0].legend(loc='lower right')

    axes[1].plot(range(stop), hist2['acc'], label='Training',
color='#FF533D')
    axes[1].plot(range(stop), hist2['val_acc'], label='Validation',
color='#03507E')
    axes[1].set_title('Fine-tuning')
    axes[1].set_ylabel('Accuracy')
    axes[1].set_xlabel('Epoch')
    axes[1].legend(loc='lower right')

plt.tight_layout();

```

In [5]:

```

def print_best_model_results(model_hist):
    best_epoch = np.argmax(model_hist['val_acc'])
    print('epoch:', best_epoch+1, \

```

```

', val_acc:', model_hist['val_acc'][best_epoch], \
', val_loss:', model_hist['val_loss'][best_epoch])

```

In [6]:

```

def save_bottleneck_features():
    datagen = ImageDataGenerator(rescale=1./255)

    model = VGG16(include_top=False, weights='imagenet')

    generator = datagen.flow_from_directory(train_data_dir,
target_size=(img_width, img_height), batch_size=batch_size,
class_mode=None, shuffle=False)
    bottleneck_features_train = model.predict_generator(generator,
nb_train_samples // batch_size)
    np.save(location+'/bottleneck_features_train.npy',
bottleneck_features_train)

    generator = datagen.flow_from_directory(validation_data_dir,
target_size=(img_width, img_height), batch_size=batch_size,
class_mode=None, shuffle=False)
    bottleneck_features_validation = model.predict_generator(generator,
nb_validation_samples // batch_size)
    np.save(location+'/bottleneck_features_validation.npy',
bottleneck_features_validation)

```

In [7]:

```

def train_top_model():
    train_data = np.load(location+'/bottleneck_features_train.npy')
    train_labels = np.array([0] * (nb_train_samples // 2) + [1] *
(nb_train_samples // 2))

    validation_data =
np.load(location+'/bottleneck_features_validation.npy')
    validation_labels = np.array([0] * (nb_validation_samples // 2) + [1]
* (nb_validation_samples // 2))

    model = Sequential()
    model.add(Flatten(input_shape=train_data.shape[1:]))
    model.add(Dense(256, activation='relu'))
    model.add(Dropout(0.5))
    model.add(Dense(1, activation='sigmoid'))
    model.compile(optimizer='rmsprop', loss='binary_crossentropy',
metrics=['accuracy'])

    checkpoint = ModelCheckpoint(top_model_weights_path,

```

```
monitor='val_acc', verbose=1, save_best_only=True, save_weights_only=True,
mode='auto')
```

```
fit = model.fit(train_data, train_labels, epochs=epochs,
batch_size=batch_size, validation_data=(validation_data, validation_labels),
callbacks=[checkpoint])
```

```
with open(location+'top_history.txt', 'w') as f:
    json.dump(fit.history, f)
```

```
return model, fit.history
```

In [8]:

```
def finetune_binary_model():
    base_model = VGG16(weights='imagenet', include_top=False,
input_shape=(256, 256, 3))
    print("Model loaded.")

    top_model = Sequential()
    top_model.add(Flatten(input_shape=base_model.output_shape[1:]))
    top_model.add(Dense(256, activation='relu'))
    top_model.add(Dropout(0.5))
    top_model.add(Dense(1, activation='sigmoid'))

    top_model.load_weights(top_model_weights_path)

    model = Model(inputs=base_model.input,
outputs=top_model(base_model.output))

    for layer in model.layers[:25]:
        layer.trainable = False

    model.compile(loss='binary_crossentropy',
optimizer=optimizers.SGD(lr=1e-4, momentum=0.9), metrics=['accuracy'])

    train_datagen = ImageDataGenerator(rescale = 1./255, zoom_range=0.2,
shear_range=0.2, horizontal_flip=True)

    test_datagen = ImageDataGenerator(rescale=1./255)

    train_generator = train_datagen.flow_from_directory(train_data_dir,
target_size=(img_height, img_width), batch_size=batch_size,
class_mode='binary')
```

```

validation_generator =
test_datagen.flow_from_directory(validation_data_dir,
target_size=(img_height, img_width), batch_size=batch_size,
class_mode='binary')

checkpoint = ModelCheckpoint(fine_tuned_model_path, monitor='val_acc',
verbose=1, save_best_only=True, save_weights_only=False, mode='auto')

fit = model.fit_generator(train_generator,
steps_per_epoch=nb_train_samples//batch_size, epochs=epochs,
validation_data=validation_generator,
validation_steps=nb_validation_samples//batch_size, verbose=1,
callbacks=[checkpoint])

with open(location+'ft_history.txt', 'w') as f:
    json.dump(fit.history, f)

return model, fit.history

```

In [9]:

```

def evaluate_binary_model(model, directory, labels):
    datagen = ImageDataGenerator(rescale=1./255)

    generator = datagen.flow_from_directory(directory,
target_size=(img_height, img_width), batch_size=batch_size,
class_mode='binary', shuffle=False)

    predictions = model.predict_generator(generator, len(labels))

    pred_labels = [0 if i<0.5 else 1 for i in predictions]

    print('')
    print(classification_report(validation_labels, pred_labels))
    print('')
    cm = confusion_matrix(validation_labels, pred_labels)
    return cm

```

Defining input data

In [10]:

```

location = 'data2'
top_model_weights_path = location+'/top_model_weights.h5'
fine_tuned_model_path = location+'/ft_model.h5'

train_data_dir = location+'/training'

```

```

validation_data_dir = location+'/validation'
train_samples = [len(os.listdir(train_data_dir+'/'+i)) for i in
sorted(os.listdir(train_data_dir))]
nb_train_samples = 1824
validation_samples = [len(os.listdir(validation_data_dir+'/'+i)) for i in
sorted(os.listdir(validation_data_dir))]
nb_validation_samples = 448

```

```

img_width, img_height = 256,256
epochs = 50
batch_size = 16

```

In [ ]:

```

save_bottleneck_features()

```

In [11]:

```

d2_model1, d2_history1 = train_top_model()
WARNING:tensorflow:From C:\Anaconda3\envs\envdlcv\lib\site-
packages\tensorflow\python\framework\op_def_library.py:263: colocate_with
(from tensorflow.python.framework.ops) is deprecated and will be removed in
a future version.
Instructions for updating:
Colocations handled automatically by placer.
WARNING:tensorflow:From C:\Anaconda3\envs\envdlcv\lib\site-
packages\keras\backend\tensorflow_backend.py:3445: calling dropout (from
tensorflow.python.ops.nn_ops) with keep_prob is deprecated and will be
removed in a future version.
Instructions for updating:
Please use `rate` instead of `keep_prob`. Rate should be set to `rate = 1
- keep_prob`.
WARNING:tensorflow:From C:\Anaconda3\envs\envdlcv\lib\site-
packages\tensorflow\python\ops\math_ops.py:3066: to_int32 (from
tensorflow.python.ops.math_ops) is deprecated and will be removed in a
future version.
Instructions for updating:
Use tf.cast instead.
Train on 1824 samples, validate on 448 samples
Epoch 1/50
1824/1824 [=====] - 19s 10ms/step - loss: 7.9614
- acc: 0.5016 - val_loss: 8.0590 - val_acc: 0.5000

Epoch 00001: val_acc improved from -inf to 0.50000, saving model to
data2/top_model_weights.h5
Epoch 2/50
1824/1824 [=====] - 19s 10ms/step - loss: 8.0590

```

- acc: 0.5000 - val\_loss: 8.0590 - val\_acc: 0.5000

Epoch 00002: val\_acc did not improve from 0.50000

Epoch 3/50

1824/1824 [=====] - 18s 10ms/step - loss: 8.0590

- acc: 0.5000 - val\_loss: 8.0590 - val\_acc: 0.5000

Epoch 00003: val\_acc did not improve from 0.50000

Epoch 4/50

1824/1824 [=====] - 18s 10ms/step - loss: 8.0590

- acc: 0.5000 - val\_loss: 8.0590 - val\_acc: 0.5000

Epoch 00004: val\_acc did not improve from 0.50000

Epoch 5/50

1824/1824 [=====] - 20s 11ms/step - loss: 4.0526

- acc: 0.6552 - val\_loss: 0.5081 - val\_acc: 0.8036

Epoch 00005: val\_acc improved from 0.50000 to 0.80357, saving model to  
data2/top\_model\_weights.h5

Epoch 6/50

1824/1824 [=====] - 19s 11ms/step - loss: 0.7258

- acc: 0.8026 - val\_loss: 0.4214 - val\_acc: 0.8549

Epoch 00006: val\_acc improved from 0.80357 to 0.85491, saving model to  
data2/top\_model\_weights.h5

Epoch 7/50

1824/1824 [=====] - 19s 10ms/step - loss: 0.4354

- acc: 0.8520 - val\_loss: 0.2513 - val\_acc: 0.9174

Epoch 00007: val\_acc improved from 0.85491 to 0.91741, saving model to  
data2/top\_model\_weights.h5

Epoch 8/50

1824/1824 [=====] - 20s 11ms/step - loss: 0.4085

- acc: 0.8739 - val\_loss: 0.5095 - val\_acc: 0.8460

Epoch 00008: val\_acc did not improve from 0.91741

Epoch 9/50

1824/1824 [=====] - 20s 11ms/step - loss: 0.2958

- acc: 0.8964 - val\_loss: 0.2074 - val\_acc: 0.9375

Epoch 00009: val\_acc improved from 0.91741 to 0.93750, saving model to  
data2/top\_model\_weights.h5

Epoch 10/50

1824/1824 [=====] - 21s 12ms/step - loss: 0.2781  
- acc: 0.9052 - val\_loss: 0.2311 - val\_acc: 0.9286

Epoch 00010: val\_acc did not improve from 0.93750

Epoch 11/50

1824/1824 [=====] - 21s 12ms/step - loss: 0.2184  
- acc: 0.9216 - val\_loss: 0.2545 - val\_acc: 0.9286

Epoch 00011: val\_acc did not improve from 0.93750

Epoch 12/50

1824/1824 [=====] - 20s 11ms/step - loss: 0.2304  
- acc: 0.9189 - val\_loss: 0.4140 - val\_acc: 0.8728

Epoch 00012: val\_acc did not improve from 0.93750

Epoch 13/50

1824/1824 [=====] - 20s 11ms/step - loss: 0.1787  
- acc: 0.9430 - val\_loss: 0.3403 - val\_acc: 0.9107

Epoch 00013: val\_acc did not improve from 0.93750

Epoch 14/50

1824/1824 [=====] - 19s 11ms/step - loss: 0.1734  
- acc: 0.9419 - val\_loss: 0.2575 - val\_acc: 0.9286

Epoch 00014: val\_acc did not improve from 0.93750

Epoch 15/50

1824/1824 [=====] - 19s 10ms/step - loss: 0.1523  
- acc: 0.9501 - val\_loss: 0.2354 - val\_acc: 0.9330

Epoch 00015: val\_acc did not improve from 0.93750

Epoch 16/50

1824/1824 [=====] - 19s 10ms/step - loss: 0.0997  
- acc: 0.9649 - val\_loss: 0.7065 - val\_acc: 0.8616

Epoch 00016: val\_acc did not improve from 0.93750

Epoch 17/50

1824/1824 [=====] - 18s 10ms/step - loss: 0.1160  
- acc: 0.9644 - val\_loss: 0.3953 - val\_acc: 0.9263

Epoch 00017: val\_acc did not improve from 0.93750

Epoch 18/50

1824/1824 [=====] - 18s 10ms/step - loss: 0.1124  
- acc: 0.9660 - val\_loss: 0.3622 - val\_acc: 0.9286



Epoch 00018: val\_acc did not improve from 0.93750  
Epoch 19/50  
1824/1824 [=====] - 18s 10ms/step - loss: 0.0781  
- acc: 0.9770 - val\_loss: 0.3651 - val\_acc: 0.9263

Epoch 00019: val\_acc did not improve from 0.93750  
Epoch 20/50  
1824/1824 [=====] - 18s 10ms/step - loss: 0.0896  
- acc: 0.9731 - val\_loss: 0.7346 - val\_acc: 0.8795

Epoch 00020: val\_acc did not improve from 0.93750  
Epoch 21/50  
1824/1824 [=====] - 19s 10ms/step - loss: 0.0980  
- acc: 0.9742 - val\_loss: 0.6882 - val\_acc: 0.8906

Epoch 00021: val\_acc did not improve from 0.93750  
Epoch 22/50  
1824/1824 [=====] - 20s 11ms/step - loss: 0.0843  
- acc: 0.9775 - val\_loss: 0.4760 - val\_acc: 0.9196

Epoch 00022: val\_acc did not improve from 0.93750  
Epoch 23/50  
1824/1824 [=====] - 21s 12ms/step - loss: 0.0810  
- acc: 0.9825 - val\_loss: 0.4074 - val\_acc: 0.9375

Epoch 00023: val\_acc did not improve from 0.93750  
Epoch 24/50  
1824/1824 [=====] - 21s 12ms/step - loss: 0.0680  
- acc: 0.9819 - val\_loss: 0.6060 - val\_acc: 0.9241

Epoch 00024: val\_acc did not improve from 0.93750  
Epoch 25/50  
1824/1824 [=====] - 20s 11ms/step - loss: 0.0743  
- acc: 0.9825 - val\_loss: 0.4872 - val\_acc: 0.9330

Epoch 00025: val\_acc did not improve from 0.93750  
Epoch 26/50  
1824/1824 [=====] - 19s 10ms/step - loss: 0.0469  
- acc: 0.9836 - val\_loss: 0.6003 - val\_acc: 0.9152

Epoch 00026: val\_acc did not improve from 0.93750  
Epoch 27/50  
1824/1824 [=====] - 19s 10ms/step - loss: 0.0808

- acc: 0.9836 - val\_loss: 0.3693 - val\_acc: 0.9241

Epoch 00027: val\_acc did not improve from 0.93750

Epoch 28/50

1824/1824 [=====] - 21s 11ms/step - loss: 0.0319

- acc: 0.9907 - val\_loss: 0.6494 - val\_acc: 0.9040

Epoch 00028: val\_acc did not improve from 0.93750

Epoch 29/50

1824/1824 [=====] - 20s 11ms/step - loss: 0.0501

- acc: 0.9857 - val\_loss: 0.4839 - val\_acc: 0.9308

Epoch 00029: val\_acc did not improve from 0.93750

Epoch 30/50

1824/1824 [=====] - 18s 10ms/step - loss: 0.0702

- acc: 0.9846 - val\_loss: 0.6352 - val\_acc: 0.9263

Epoch 00030: val\_acc did not improve from 0.93750

Epoch 31/50

1824/1824 [=====] - 19s 10ms/step - loss: 0.0568

- acc: 0.9868 - val\_loss: 0.4939 - val\_acc: 0.9330

Epoch 00031: val\_acc did not improve from 0.93750

Epoch 32/50

1824/1824 [=====] - 20s 11ms/step - loss: 0.0333

- acc: 0.9901 - val\_loss: 0.5689 - val\_acc: 0.9286

Epoch 00032: val\_acc did not improve from 0.93750

Epoch 33/50

1824/1824 [=====] - 20s 11ms/step - loss: 0.0477

- acc: 0.9890 - val\_loss: 0.6067 - val\_acc: 0.9308

Epoch 00033: val\_acc did not improve from 0.93750

Epoch 34/50

1824/1824 [=====] - 19s 10ms/step - loss: 0.0297

- acc: 0.9901 - val\_loss: 0.5569 - val\_acc: 0.9241

Epoch 00034: val\_acc did not improve from 0.93750

Epoch 35/50

1824/1824 [=====] - 19s 10ms/step - loss: 0.0405

- acc: 0.9940 - val\_loss: 0.5417 - val\_acc: 0.9241

Epoch 00035: val\_acc did not improve from 0.93750

Epoch 36/50  
1824/1824 [=====] - 19s 10ms/step - loss: 0.0387  
- acc: 0.9907 - val\_loss: 0.5860 - val\_acc: 0.9085

Epoch 00036: val\_acc did not improve from 0.93750

Epoch 37/50  
1824/1824 [=====] - 19s 10ms/step - loss: 0.0485  
- acc: 0.9901 - val\_loss: 0.5715 - val\_acc: 0.9286

Epoch 00037: val\_acc did not improve from 0.93750

Epoch 38/50  
1824/1824 [=====] - 18s 10ms/step - loss: 0.0366  
- acc: 0.9890 - val\_loss: 0.6733 - val\_acc: 0.9129

Epoch 00038: val\_acc did not improve from 0.93750

Epoch 39/50  
1824/1824 [=====] - 18s 10ms/step - loss: 0.0337  
- acc: 0.9890 - val\_loss: 0.5863 - val\_acc: 0.9219

Epoch 00039: val\_acc did not improve from 0.93750

Epoch 40/50  
1824/1824 [=====] - 19s 10ms/step - loss: 0.0256  
- acc: 0.9951 - val\_loss: 0.7194 - val\_acc: 0.9219

Epoch 00040: val\_acc did not improve from 0.93750

Epoch 41/50  
1824/1824 [=====] - 19s 10ms/step - loss: 0.0151  
- acc: 0.9956 - val\_loss: 0.6697 - val\_acc: 0.9174

Epoch 00041: val\_acc did not improve from 0.93750

Epoch 42/50  
1824/1824 [=====] - 18s 10ms/step - loss: 0.0786  
- acc: 0.9857 - val\_loss: 0.5607 - val\_acc: 0.9174

Epoch 00042: val\_acc did not improve from 0.93750

Epoch 43/50  
1824/1824 [=====] - 18s 10ms/step - loss: 0.0083  
- acc: 0.9973 - val\_loss: 0.6423 - val\_acc: 0.9286

Epoch 00043: val\_acc did not improve from 0.93750

Epoch 44/50  
1824/1824 [=====] - 21s 11ms/step - loss: 0.0325  
- acc: 0.9901 - val\_loss: 0.6561 - val\_acc: 0.9174

Epoch 00044: val\_acc did not improve from 0.93750  
Epoch 45/50  
1824/1824 [=====] - 21s 11ms/step - loss: 0.0154  
- acc: 0.9940 - val\_loss: 0.6484 - val\_acc: 0.9241

Epoch 00045: val\_acc did not improve from 0.93750  
Epoch 46/50  
1824/1824 [=====] - 21s 11ms/step - loss: 0.0235  
- acc: 0.9956 - val\_loss: 0.6600 - val\_acc: 0.9241

Epoch 00046: val\_acc did not improve from 0.93750  
Epoch 47/50  
1824/1824 [=====] - 19s 11ms/step - loss: 0.0320  
- acc: 0.9934 - val\_loss: 0.7059 - val\_acc: 0.9219

Epoch 00047: val\_acc did not improve from 0.93750  
Epoch 48/50  
1824/1824 [=====] - 18s 10ms/step - loss: 0.0259  
- acc: 0.9951 - val\_loss: 0.9661 - val\_acc: 0.8973

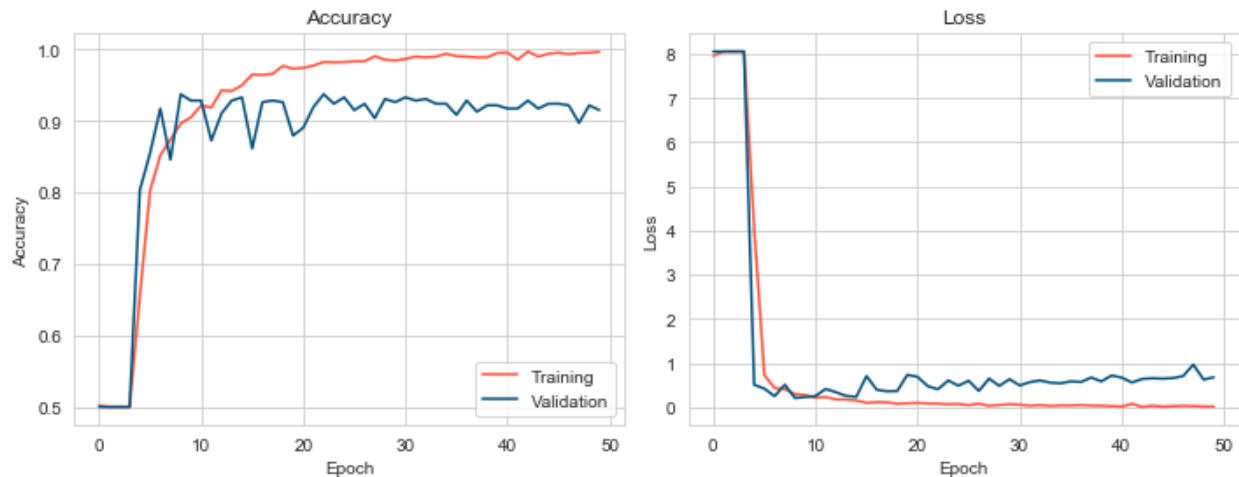
Epoch 00048: val\_acc did not improve from 0.93750  
Epoch 49/50  
1824/1824 [=====] - 18s 10ms/step - loss: 0.0169  
- acc: 0.9956 - val\_loss: 0.6273 - val\_acc: 0.9219

Epoch 00049: val\_acc did not improve from 0.93750  
Epoch 50/50  
1824/1824 [=====] - 18s 10ms/step - loss: 0.0154  
- acc: 0.9967 - val\_loss: 0.6788 - val\_acc: 0.9152

Epoch 00050: val\_acc did not improve from 0.93750

In [12]:

```
plot_metrics(d2_history1)
Best Model:
epoch: 9 , val_acc: 0.9375 , val_loss: 0.2073782096683447
```



## Fine Tuning

In [ ]:

```
ft_model, ft_history = finetune_binary_model()
Model loaded.
Found 1824 images belonging to 2 classes.
Found 448 images belonging to 2 classes.
Epoch 1/50
114/114 [=====] - 1571s 14s/step - loss: 0.3264 -
acc: 0.8799 - val_loss: 0.2074 - val_acc: 0.9375

Epoch 00001: val_acc improved from -inf to 0.93750, saving model to
data2/ft_model.h5
Epoch 2/50
114/114 [=====] - 1573s 14s/step - loss: 0.3228 -
acc: 0.8871 - val_loss: 0.2074 - val_acc: 0.9375

Epoch 00002: val_acc did not improve from 0.93750
Epoch 3/50
114/114 [=====] - 24835s 218s/step - loss: 0.3095
- acc: 0.8860 - val_loss: 0.2074 - val_acc: 0.9375

Epoch 00003: val_acc did not improve from 0.93750
Epoch 4/50
114/114 [=====] - 38145s 335s/step - loss: 0.3335
- acc: 0.8854 - val_loss: 0.2074 - val_acc: 0.9375

Epoch 00004: val_acc did not improve from 0.93750
Epoch 5/50
114/114 [=====] - 1602s 14s/step - loss: 0.3127 -
acc: 0.8942 - val_loss: 0.2074 - val_acc: 0.9375
```

```
Epoch 00005: val_acc did not improve from 0.93750
Epoch 6/50
114/114 [=====] - 1624s 14s/step - loss: 0.2912 -
acc: 0.8964 - val_loss: 0.2074 - val_acc: 0.9375

Epoch 00006: val_acc did not improve from 0.93750
Epoch 7/50
114/114 [=====] - 1649s 14s/step - loss: 0.3236 -
acc: 0.8843 - val_loss: 0.2074 - val_acc: 0.9375

Epoch 00007: val_acc did not improve from 0.93750
Epoch 8/50
114/114 [=====] - 1711s 15s/step - loss: 0.3301 -
acc: 0.8887 - val_loss: 0.2074 - val_acc: 0.9375

Epoch 00008: val_acc did not improve from 0.93750
Epoch 9/50
110/114 [=====>..] - ETA: 1:06 - loss: 0.3238 - acc:
0.8841
```

In [ ]:

```
plot_metrics(ft_history)
```

**Load Model**

In [17]:

```
ft_model = load_model(location+'ft_model.h5')
WARNING:tensorflow:From C:\Anaconda3\envs\envdlcv\lib\site-
packages\tensorflow\python\ops\math_ops.py:3066: to_int32 (from
tensorflow.python.ops.math_ops) is deprecated and will be removed in a
future version.
Instructions for updating:
Use tf.cast instead.
C:\Anaconda3\envs\envdlcv\lib\site-packages\keras\engine\saving.py:327:
UserWarning: Error in loading the saved optimizer state. As a result, your
model is starting with a freshly initialized optimizer.
  warnings.warn('Error in loading the saved optimizer '
```

In [ ]:

```
with open('dataala/top_history.txt') as f:
    top_history = json.load(f)
```

In [ ]:

```
with open('dataala/ft_history.txt') as f:
    ft_history = json.load(f)
```

In [ ]:

```
plot_acc_metrics(top_history, ft_history)
```

In [22]:

```
validation_labels = np.array([0] * (nb_validation_samples // 2) + [1] *  
(nb_validation_samples // 2))
```

In [51]:

```
cm = evaluate_binary_model(ft_model, validation_data_dir,  
validation_labels)
```

Found 448 images belonging to 2 classes.

```
-----  
-
```

**KeyboardInterrupt**

Traceback (most recent call  
last)

<ipython-input-51-bf52512d511d> in <module>

```
----> 1 cm = evaluate_binary_model(ft_model, validation_data_dir,  
validation_labels)
```

<ipython-input-27-304db6f68ef2> in evaluate\_binary\_model(model, directory,  
labels)

```
4 generator = datagen.flow_from_directory(directory,  
target_size=(img_height, img_width), batch_size=batch_size,  
class_mode='binary', shuffle=False)
```

```
5  
----> 6 predictions = model.predict_generator(generator, len(labels))  
7
```

```
8 pred_labels = [0 if i<0.5 else 1 for i in predictions]
```

C:\Anaconda3\envs\envdlcv\lib\site-packages\keras\legacy\interfaces.py in  
wrapper(\*args, \*\*kwargs)

```
89 warnings.warn('Update your ``' + object_name + ``  
call to the ' +
```

```
90 'Keras 2 API: ' + signature,  
stacklevel=2)
```

```
----> 91 return func(*args, **kwargs)  
92 wrapper._original_function = func  
93 return wrapper
```

C:\Anaconda3\envs\envdlcv\lib\site-packages\keras\engine\training.py in  
predict\_generator(self, generator, steps, max\_queue\_size, workers,  
use\_multiprocessing, verbose)

```
1520 workers=workers,  
1521 use_multiprocessing=use_multiprocessing,  
-> 1522 verbose=verbose)
```

```

C:\Anaconda3\envs\envdlcv\lib\site-
packages\keras\engine\training_generator.py in predict_generator(model,
generator, steps, max_queue_size, workers, use_multiprocessing, verbose)
    451             x = generator_output
    452
--> 453             outs = model.predict_on_batch(x)
    454             outs = to_list(outs)
    455

```

```

C:\Anaconda3\envs\envdlcv\lib\site-packages\keras\engine\training.py in
predict_on_batch(self, x)
    1272         ins = x
    1273         self._make_predict_function()
-> 1274         outputs = self.predict_function(ins)
    1275         return unpack_singleton(outputs)
    1276

```

```

C:\Anaconda3\envs\envdlcv\lib\site-
packages\keras\backend\tensorflow_backend.py in __call__(self, inputs)
    2713         return self._legacy_call(inputs)
    2714
-> 2715         return self._call(inputs)
    2716     else:
    2717         if py_any(is_tensor(x) for x in inputs):

```

```

C:\Anaconda3\envs\envdlcv\lib\site-
packages\keras\backend\tensorflow_backend.py in _call(self, inputs)
    2673         fetched = self._callable_fn(*array_vals,
run_metadata=self.run_metadata)
    2674     else:
-> 2675         fetched = self._callable_fn(*array_vals)
    2676         return fetched[:len(self.outputs)]
    2677

```

```

C:\Anaconda3\envs\envdlcv\lib\site-
packages\tensorflow\python\client\session.py in __call__(self, *args,
**kwargs)
    1437         ret = tf_session.TF_SessionRunCallable(
    1438             self._session._session, self._handle, args, status,
-> 1439             run_metadata_ptr)
    1440         if run_metadata:
    1441             proto_data = tf_session.TF_GetBuffer(run_metadata_ptr)

```



**KeyboardInterrupt:**

In [ ]:

```
heatmap_laebels = ['Damaged', 'Whole']
```

In [ ]:

```
sns.heatmap(cm, annot=True, annot_kws={"size":16}, fmt='g', cmap='OrRd',  
xticklabels=heatmap_labels, yticklabels=heatmap_labels)
```

In [ ]:

```
sns.heatmap(cm, annot=Ture, annot_kws={"size":16}, fmt='g', cmap='Blues',  
xticklabels=heatmap_labels, yticklabels=heatmap_labels)
```

Pipe2

In [11]:

```
def pipe2(image_path, model):  
    urllib.request.urlretrieve(image_path, 'save.jpg')  
    img = load_img('save.jpg', target_size=(256,256))  
    x = img_to_array(img)  
    x = x.reshape((1,) + x.shape)/255  
    pred = model.predict(x)  
    print("Validating that damage exists....")  
    print(pred)  
    if (pred[0][0]<=0.5):  
        print("Validation complete - proceed to location and severity  
determination")  
    else:  
        print ("Are you sure that your car is damaged? Please submit  
another picture of the damage.")  
        print ("Hint: Try zooming in/out, using a different angle or  
different lighting")
```

In [12]:

```
Image('http://3.bp.blogspot.com/-  
PrRY9XxCqYQ/UDNutnMI7LI/AAAAAAAABdw/UGygghh-hRA/s1600/Bumper+scuff.JPG')
```

Out[12]:



In [18]:

```
pipe2('http://3.bp.blogspot.com/-  
PrRY9XxCqYQ/UDNutnMI7LI/AAAAAAAAABdw/UGygghh-hRA/s1600/Bumper+scuff.JPG',  
ft_model)  
Validating that damage exists....  
[[0.0002488]]  
Validation complete - proceed to location and severity determination
```

In [40]:

```
Image('https://i.ytimg.com/vi/4oV1klVPogY/maxresdefault.jpg')
```

Out[40]:



## Scratch at both the doors

In [41]:

```
pipe2('https://i.ytimg.com/vi/4oVlklVPogY/maxresdefault.jpg', ft_model)
Validating that damage exists....
[[0.01300194]]
Validation complete - proceed to location and severity determination
```

In [47]:

```
Image('http://blog.automart.co.za/wp-
content/uploads/2014/09/Accident_Damaged_Car.png')
```

Out[47]:



In [46]:

```
pipe2('http://blog.automart.co.za/wp-  
content/uploads/2014/09/Accident_Damaged_Car.png', ft_model)  
Validating that damage exists....  
[[0.11757535]]  
Validation complete - proceed to location and severity determination
```