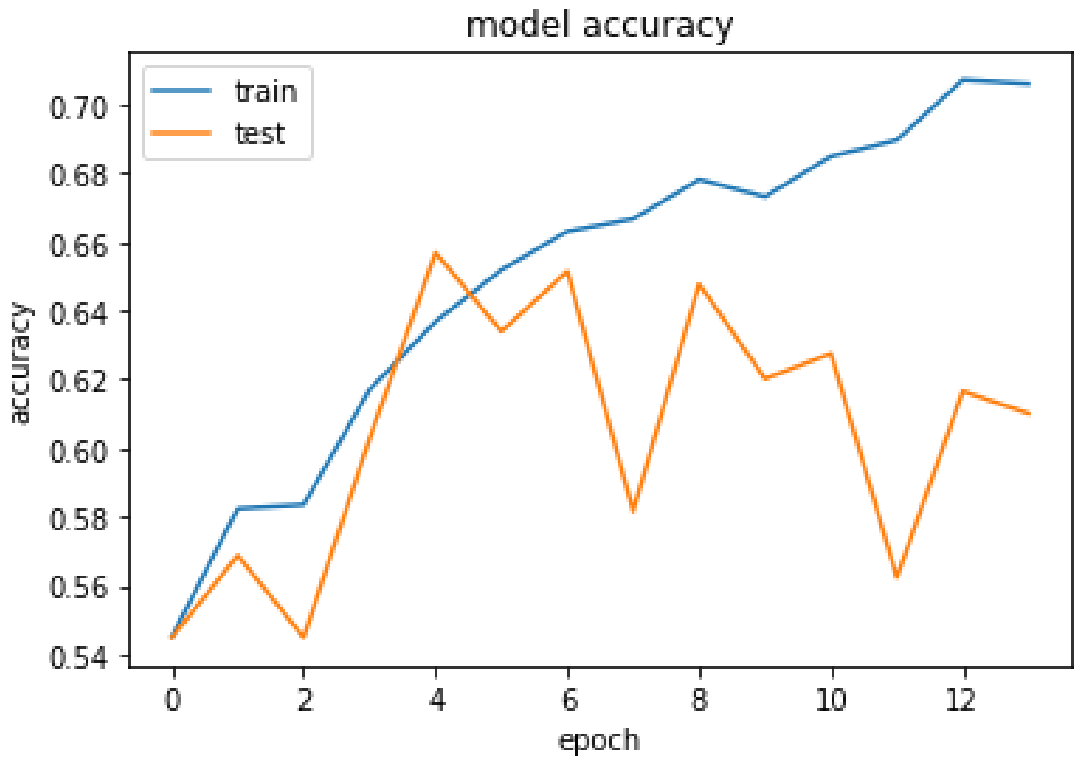
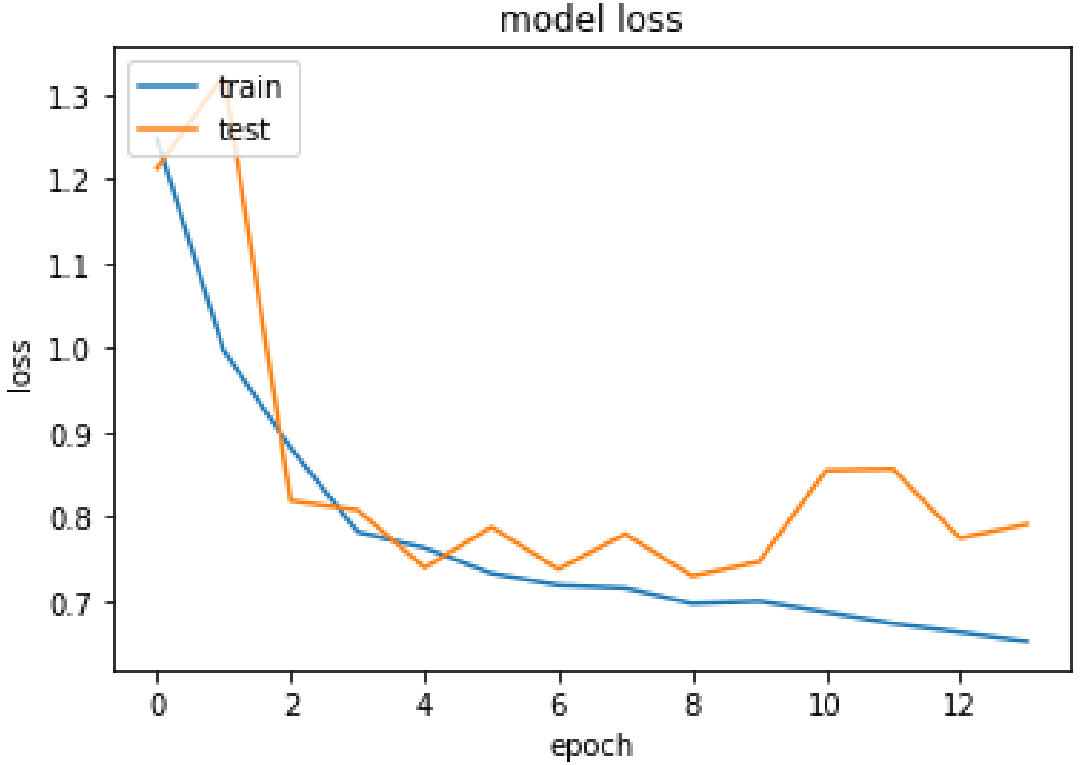
# Finding vs No Finding

# L2 regularization + Adam opt



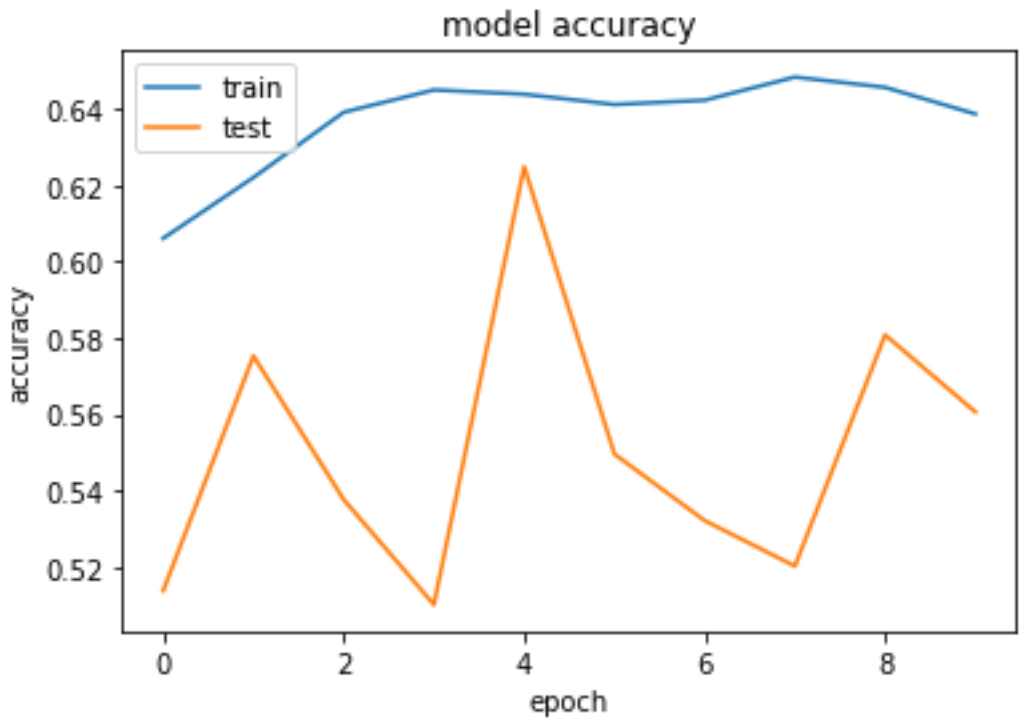



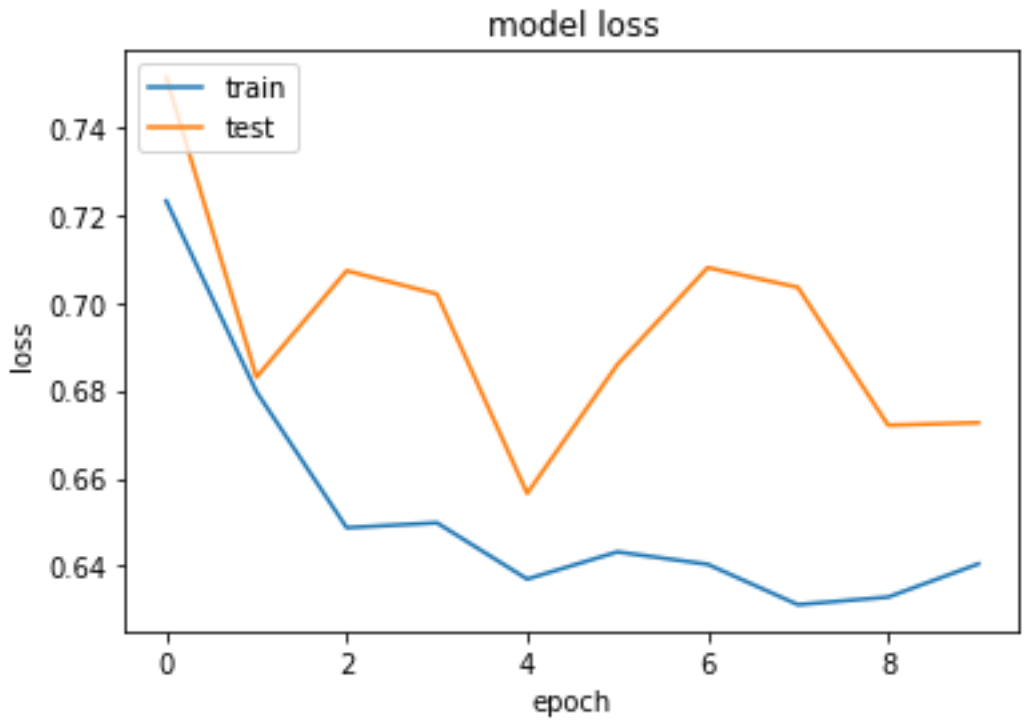

66% validation accuracy

Below is the table that shows image size, weights size, top-1 accuracy, top-5 accuracy, no.of.parameters and depth of each deep neural net architecture available in Keras.

| Model | Image size | Weights size | Top-1 accuracy | Top-5 accuracy | Parameters | Depth |
|---|---|---|---|---|---|---|
| Xception | 299 x 299 | 88 MB | 0.790 | 0.945 | 22,910,480 | 126 |
| VGG16 | 224 x 224 | 528 MB | 0.715 | 0.901 | 138,357,544 | 23 |
| VGG19 | 224 x 224 | 549 MB | 0.727 | 0.910 | 143,667,240 | 26 |
| ResNet50 | 224 x 224 | 99 MB | 0.759 | 0.929 | 25,636,712 | 168 |
| InceptionV3 | 299 x 299 | 92 MB | 0.788 | 0.944 | 23,851,784 | 159 |
| Inception ResNetV2 | 299 x 299 | 215 MB | 0.804 | 0.953 | 55,873,736 | 572 |
| MobileNet | 224 x 224 | 17 MB | 0.665 | 0.871 | 4,253,864 | 88 |

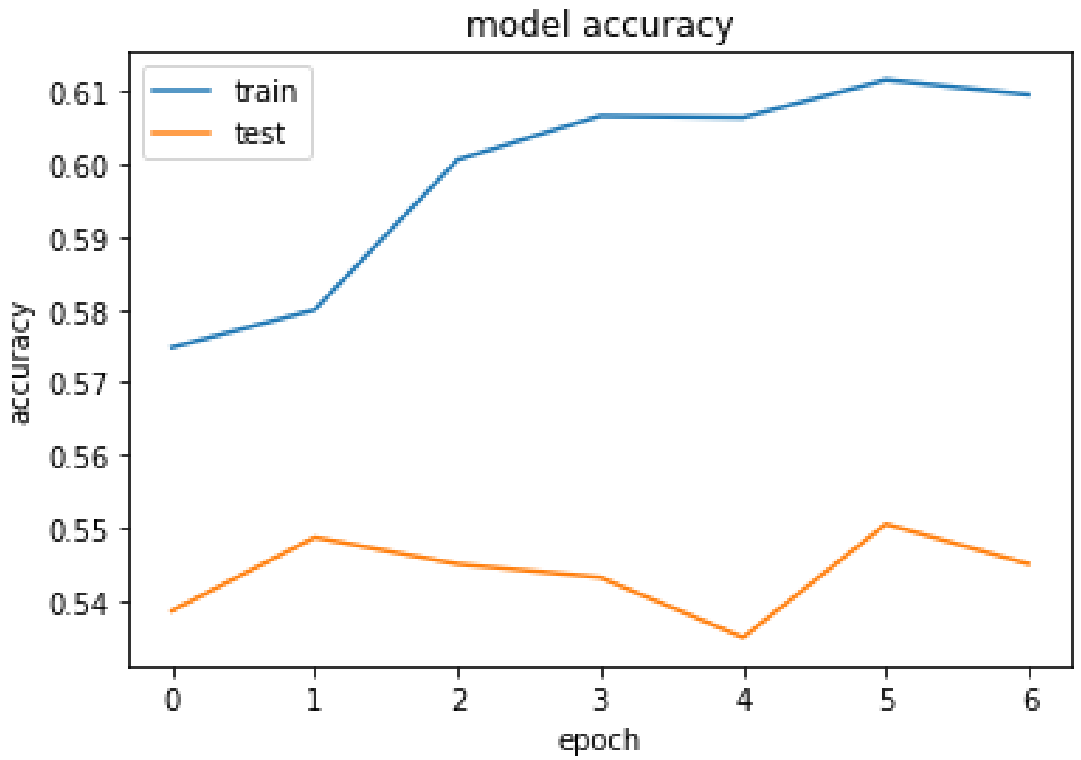
# Pretrained Model: VGG16

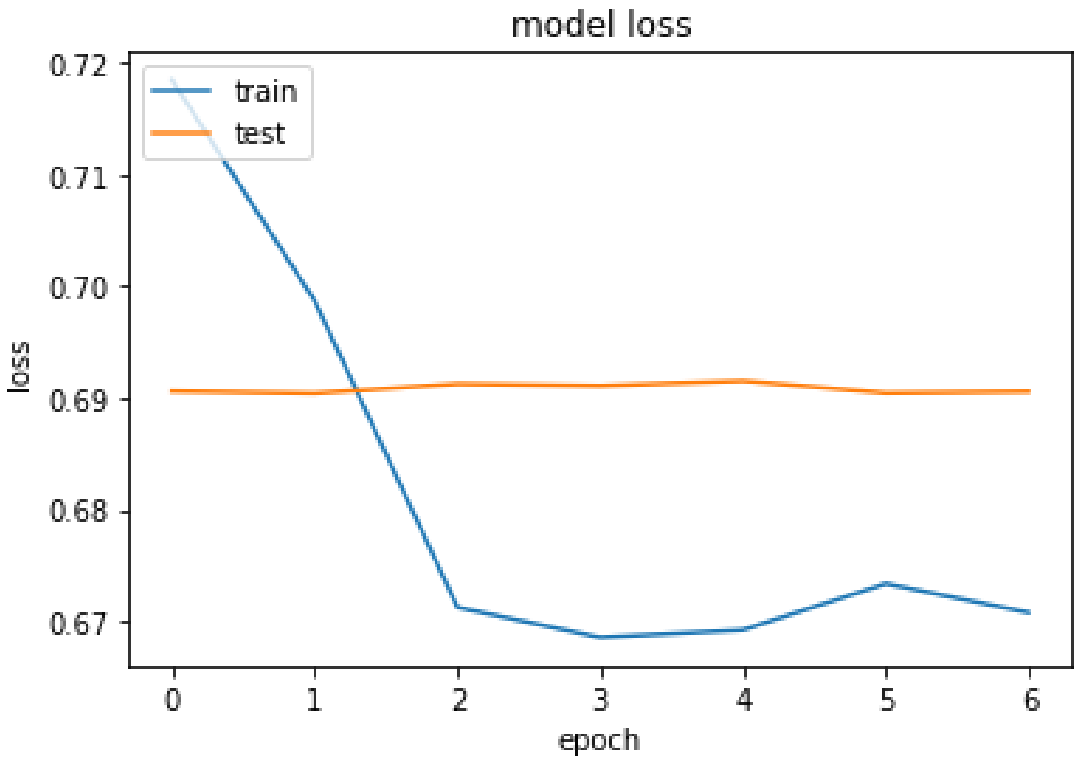



62% validation accuracy

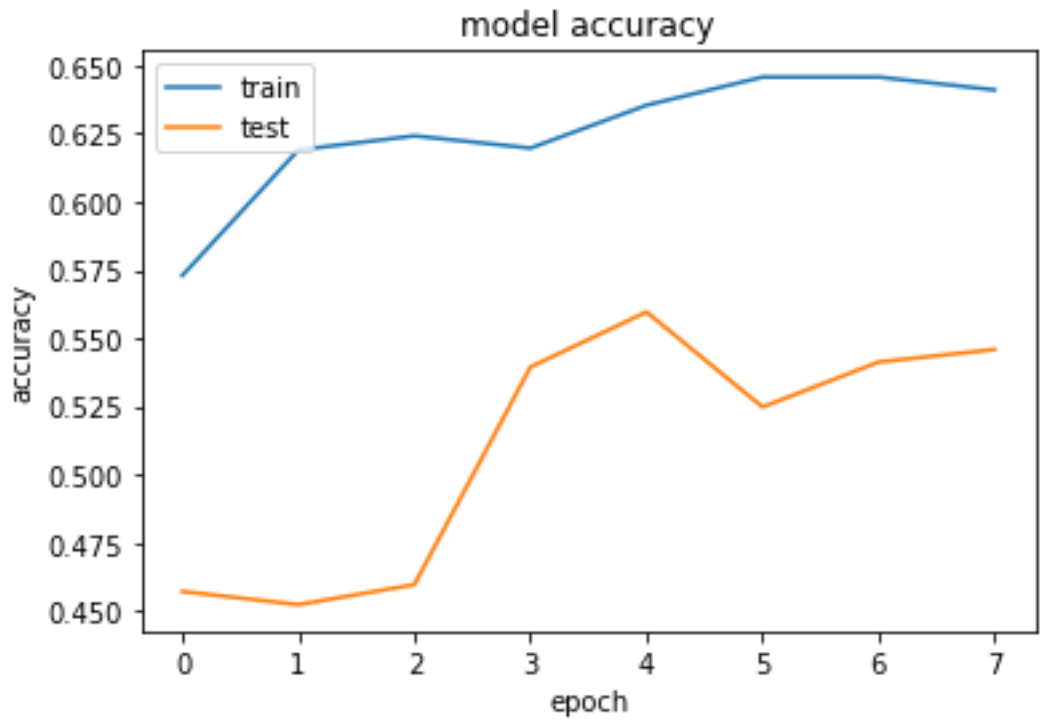
# Pretrained Model: Xception

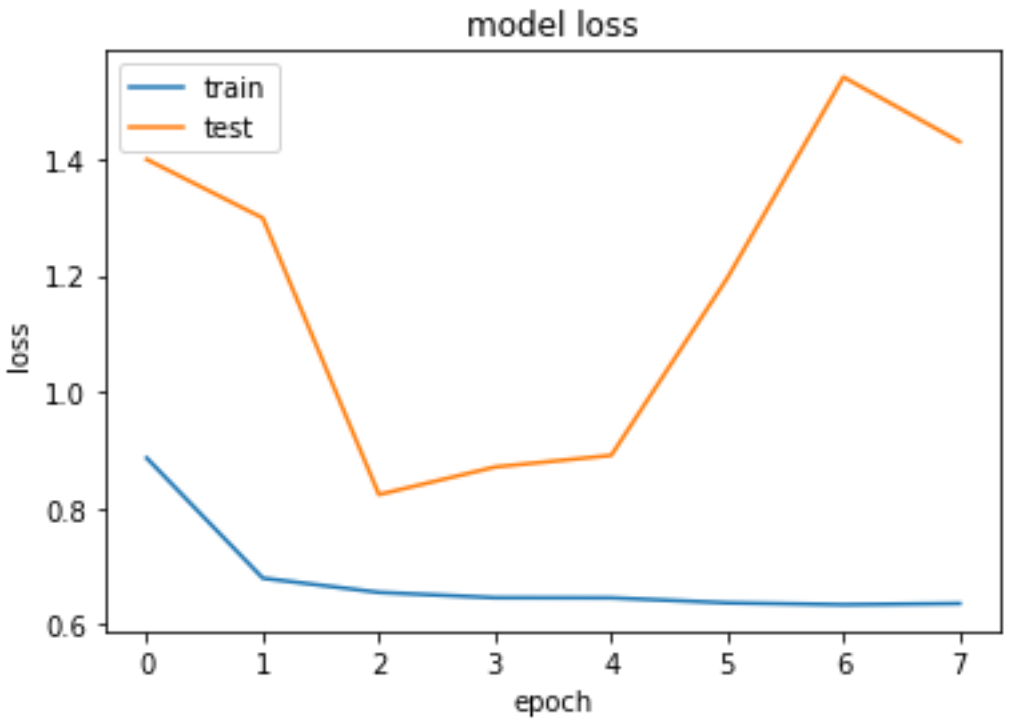



Overfitting

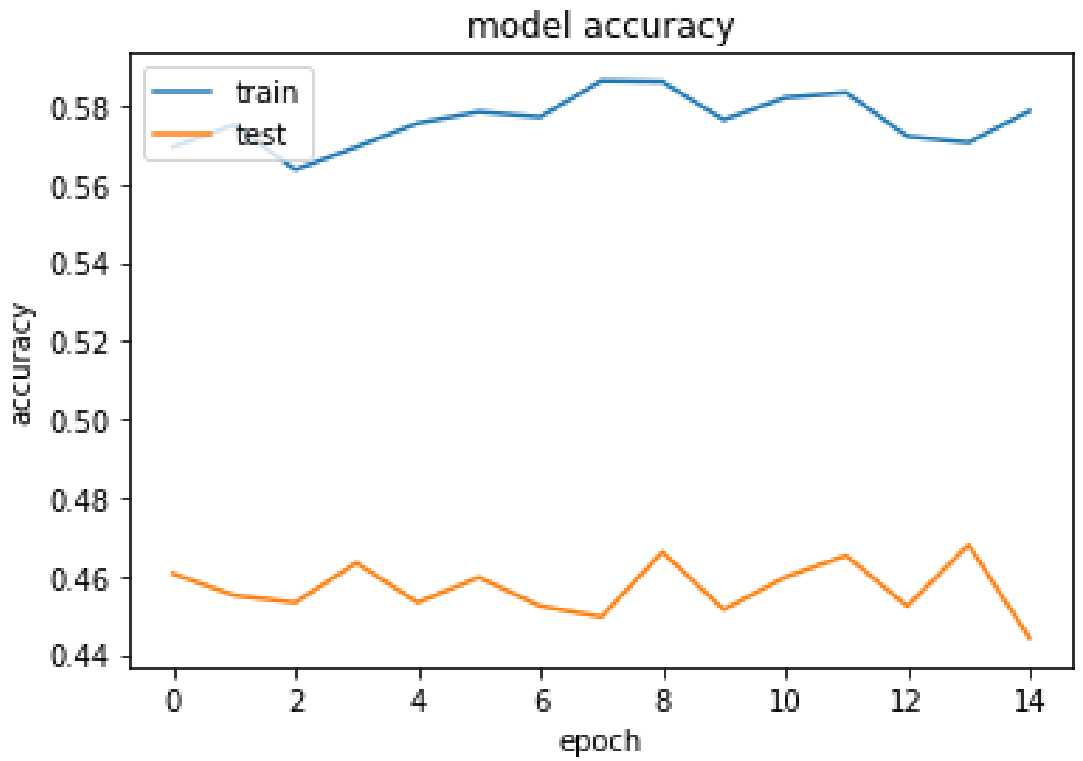
# Pretrained Model: ResNet50

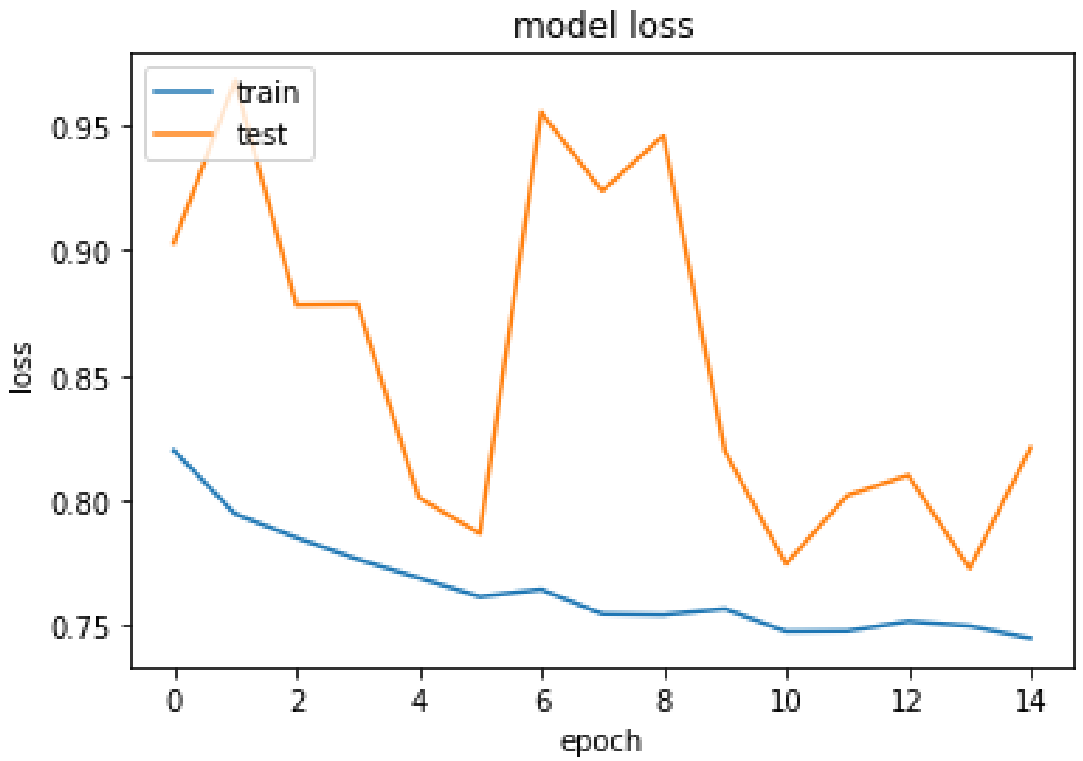



Overfitting

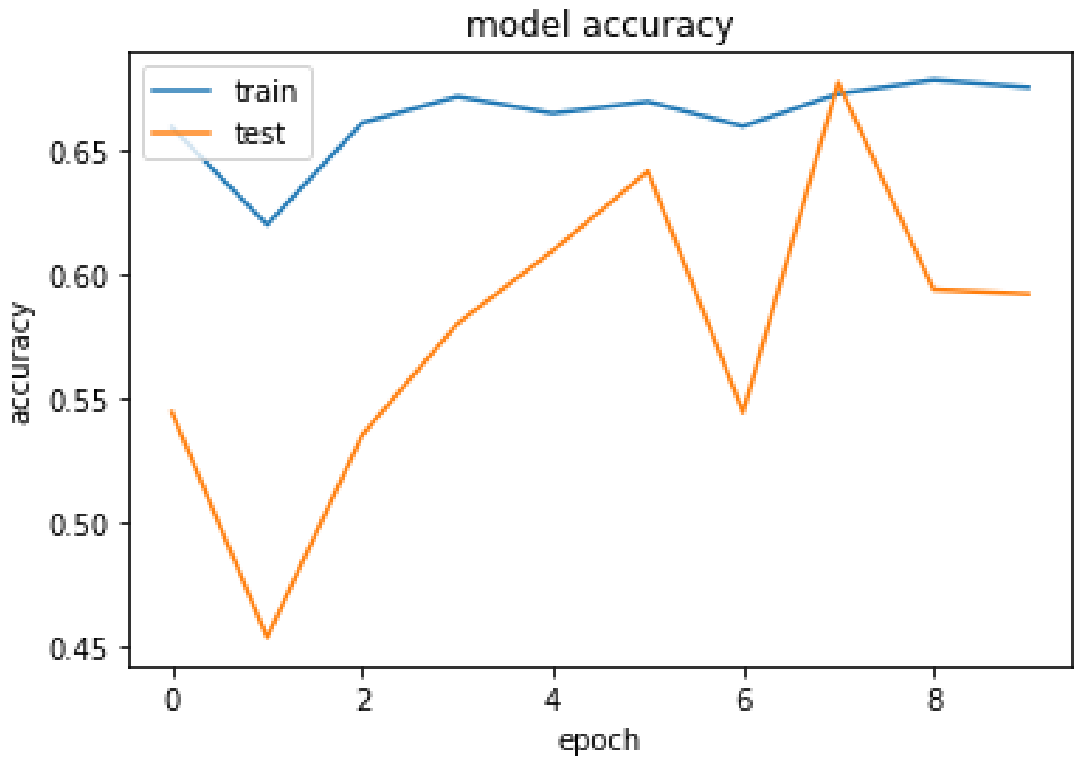
# Pretrained Model: InceptionV3

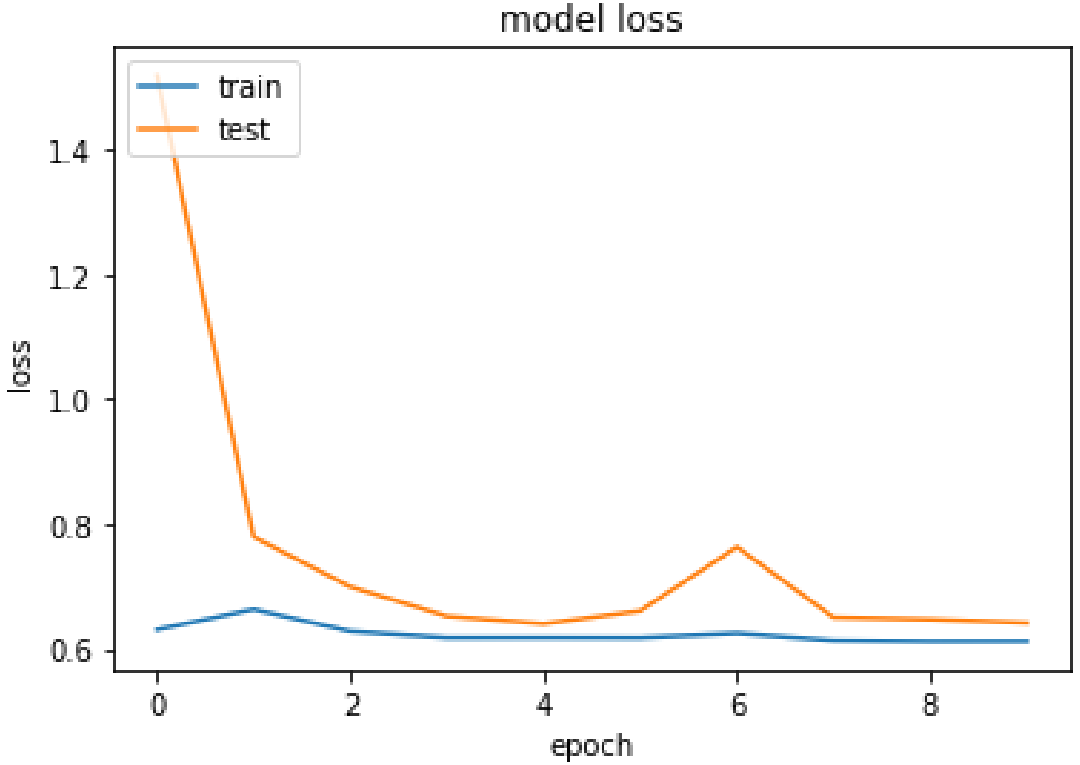



Overfitting

# Pretrained Model: VGG16 + Fine Tuning



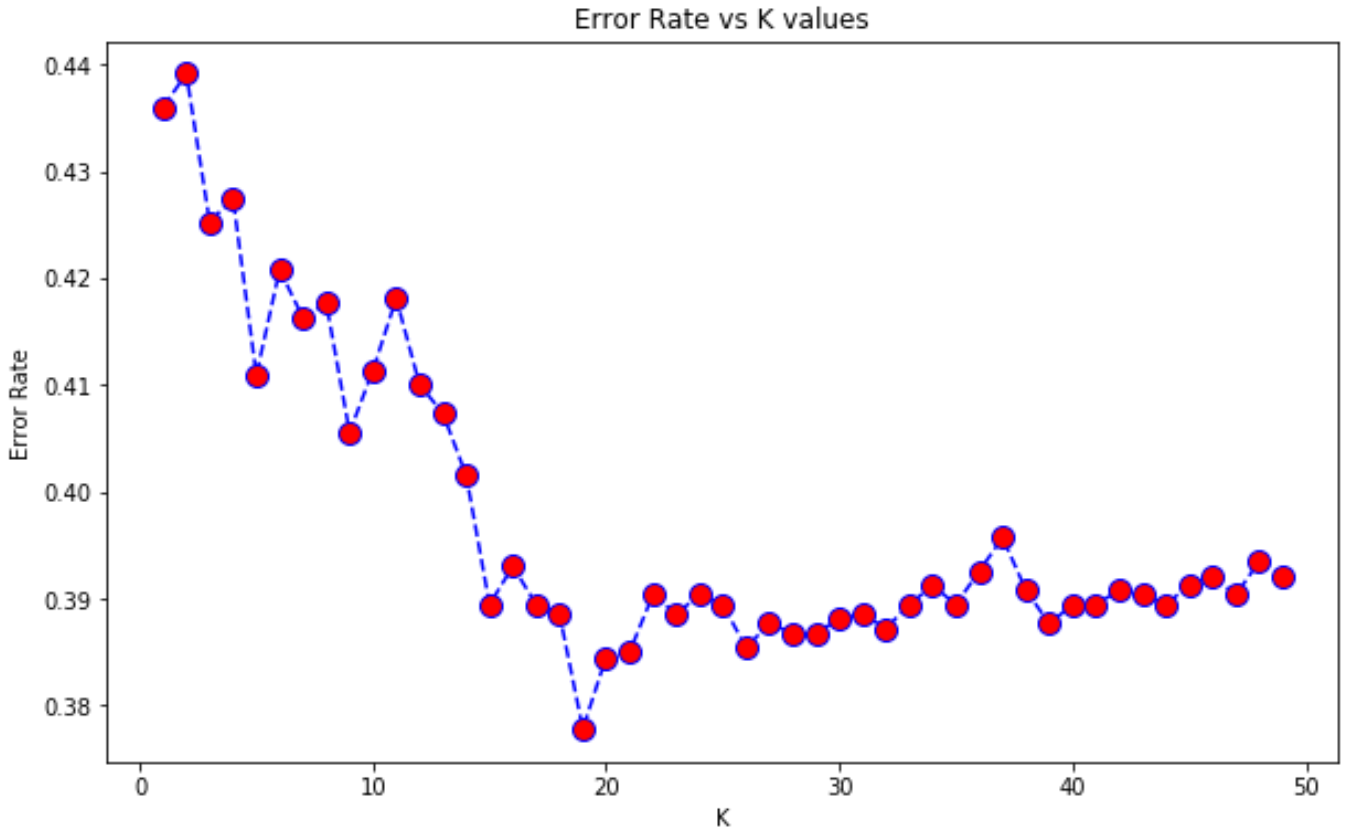

67% validation accuracy

# Support Vector Machine (SVM)

```
#SVC
from sklearn.svm import SVC
print("\n")
print("[INFO] evaluating raw pixel accuracy...")
model = SVC(max_iter=2000,class_weight='balanced',C = 5, gamma = 0.1)
model.fit(X_train, y_train)
acc = model.score(X_test, y_test)
print("[INFO] SVM-SVC raw pixel accuracy: {:.2f}%".format(acc * 100))

#from sklearn.grid_search import GridSearchCV
#param_grid = {'C':[0.1,1,10,100],'gamma':[1,0.1,0.01,0.001]}
#grid = GridSearchCV(SVC(),param_grid,verbose=1)
#grid.fit(X_train, y_train)
```

```
[INFO] evaluating raw pixel accuracy...

/opt/conda/lib/python3.6/site-packages/sklearn/svm/base.py:244: ConvergenceWarning: Solver ter
minated early (max_iter=2000).  Consider pre-processing your data with StandardScaler or MinMa
xScaler.
  % self.max_iter, ConvergenceWarning)

[INFO] SVM-SVC raw pixel accuracy: 55.86%
```

# KNN: Finding the optimal k




```
[INFO] evaluating raw pixel accuracy...
[INFO] k-NN classifier: k=1000
[INFO] raw pixel accuracy: 61.68%
```

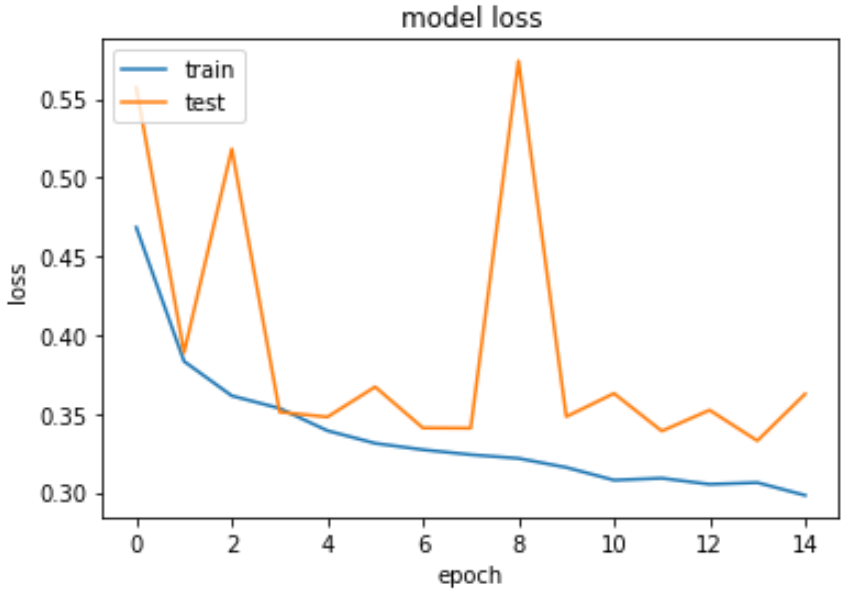
# Random Forest

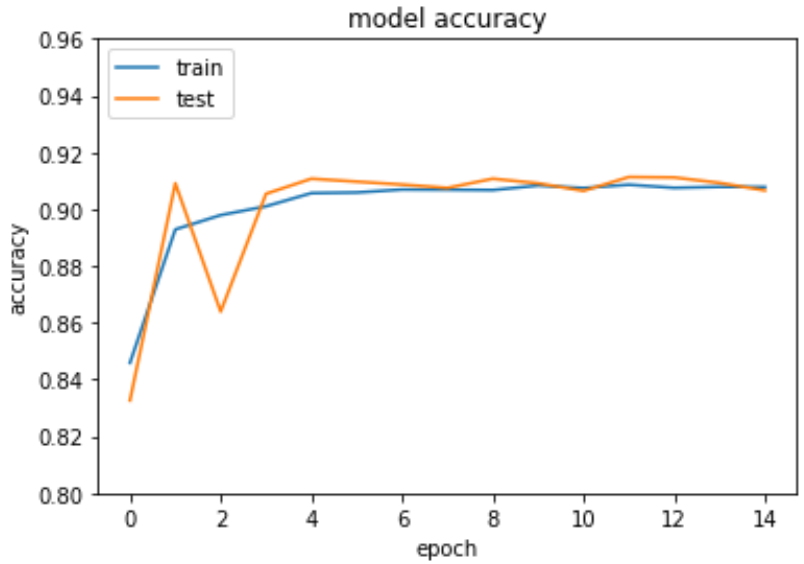
```python
##Random Forest
from sklearn.ensemble import RandomForestClassifier
print("\n")
print("[INFO] evaluating raw pixel accuracy...")
model = RandomForestClassifier(n_estimators=200)
model.fit(X_train, y_train)
acc = model.score(X_test, y_test)
print("[INFO] Random Forest raw pixel accuracy: {:.2f}%".format(acc * 100))
y_predict = model.predict(X_test)
print(confusion_matrix(y_test, y_predict))
print('\n')
print(classification_report(y_test,y_predict))
```

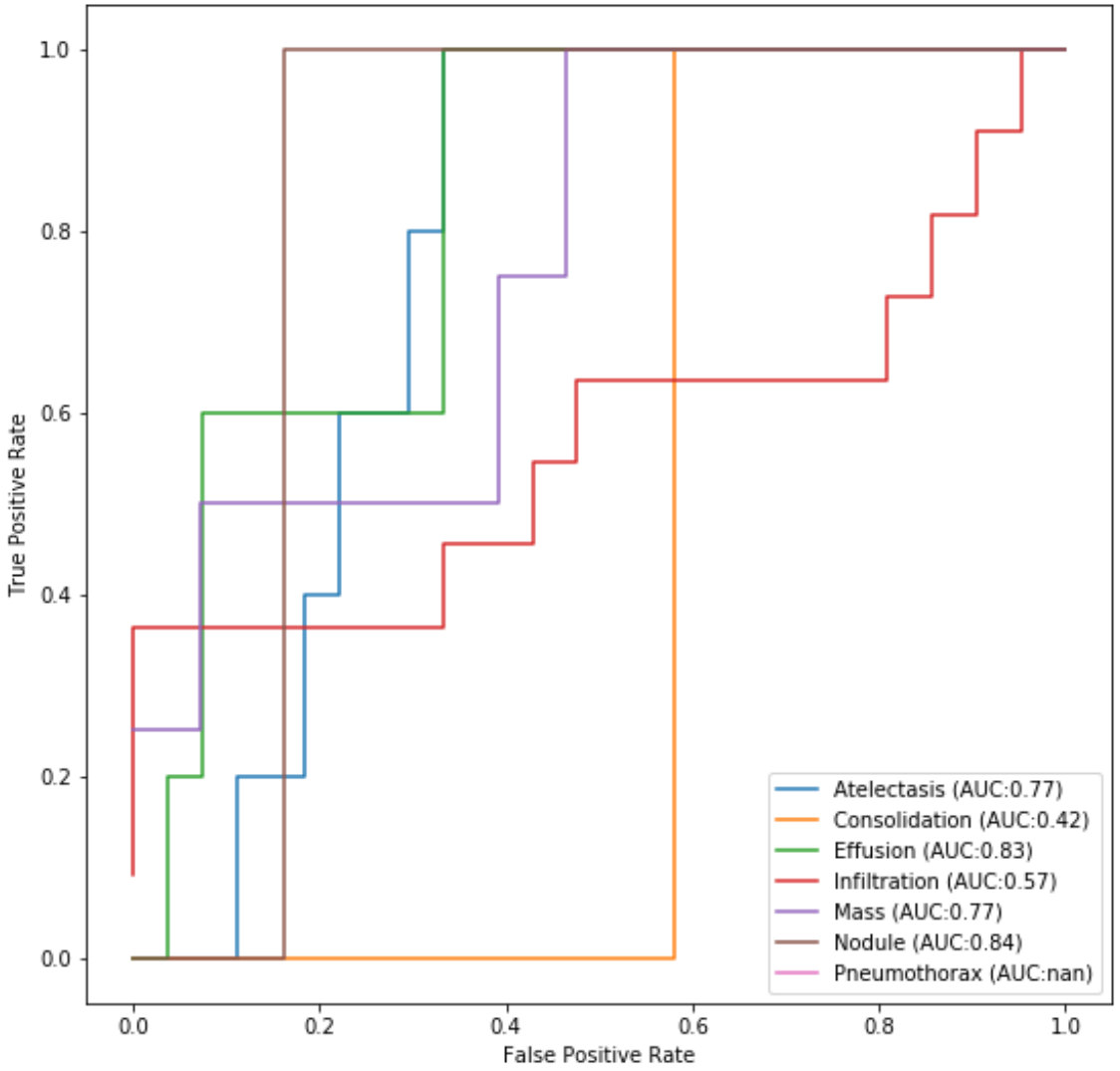
```
[INFO] evaluating raw pixel accuracy...
[INFO] Random Forest raw pixel accuracy: 64.82%
[[457 162]
 [231 267]]


              precision    recall  f1-score   support

           0       0.66      0.74      0.70       619
           1       0.62      0.54      0.58       498

   micro avg       0.65      0.65      0.65      1117
   macro avg       0.64      0.64      0.64      1117
weighted avg       0.65      0.65      0.64      1117
```

# Multi-label classification

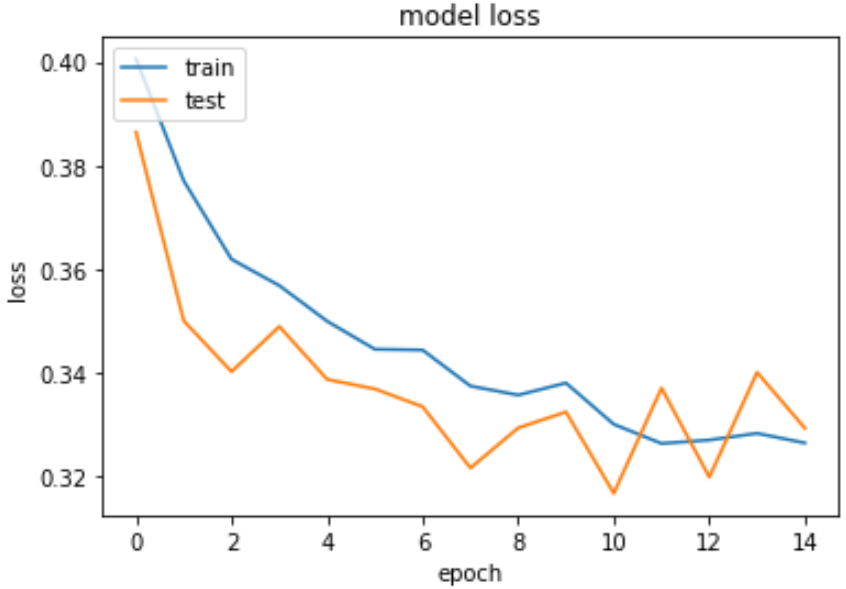
L2 regularization + Adam opt

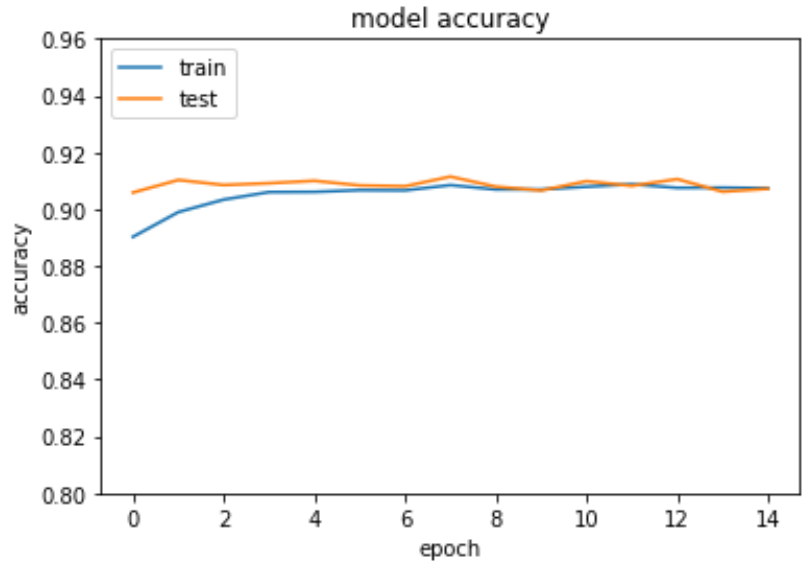


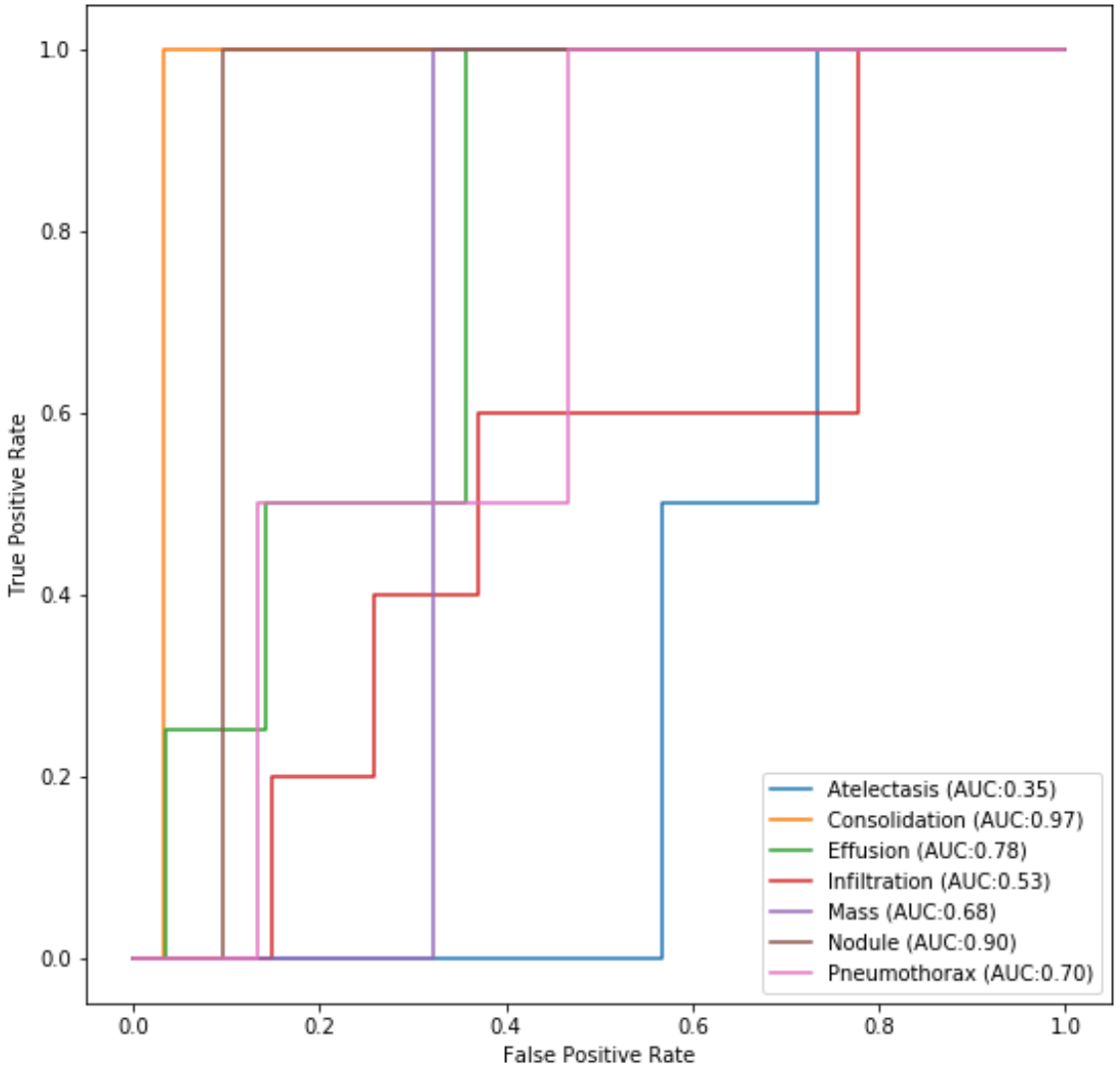



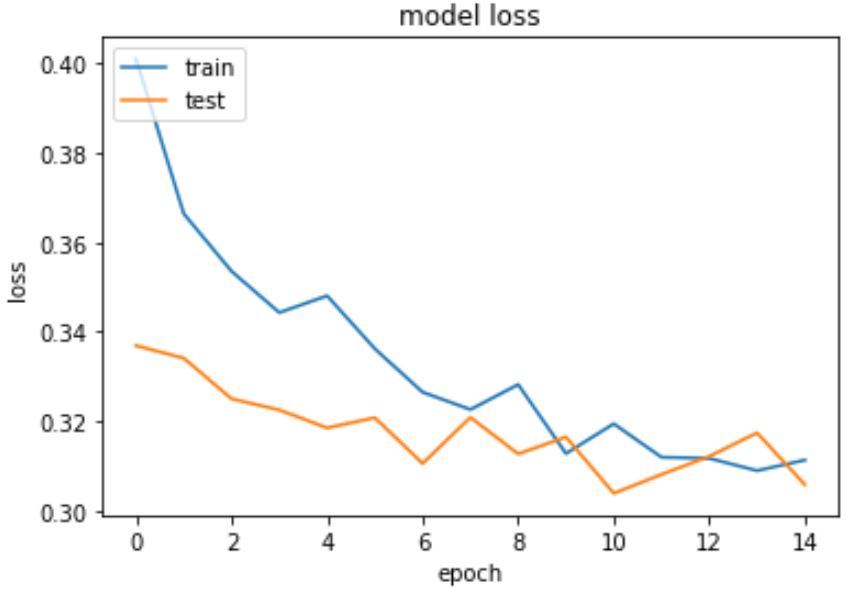
# Image Augmentation

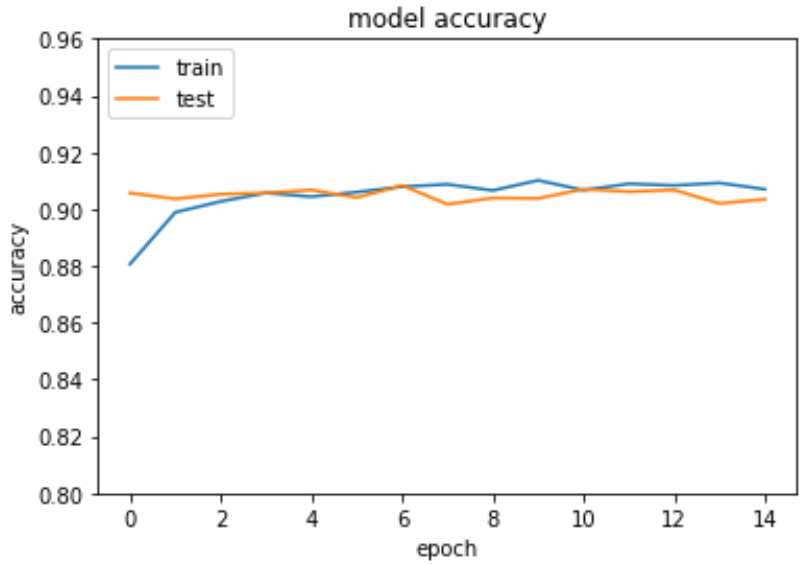


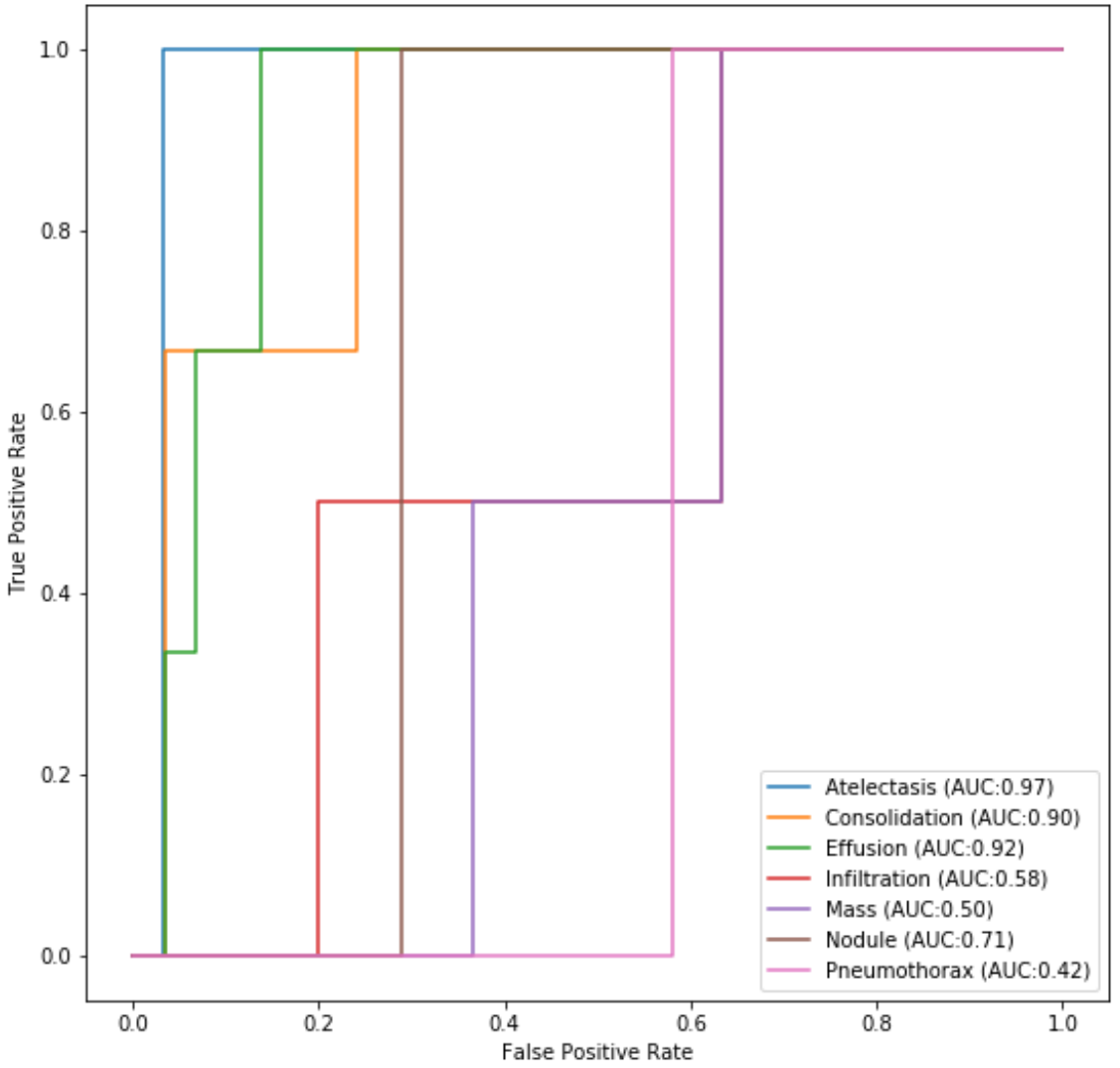



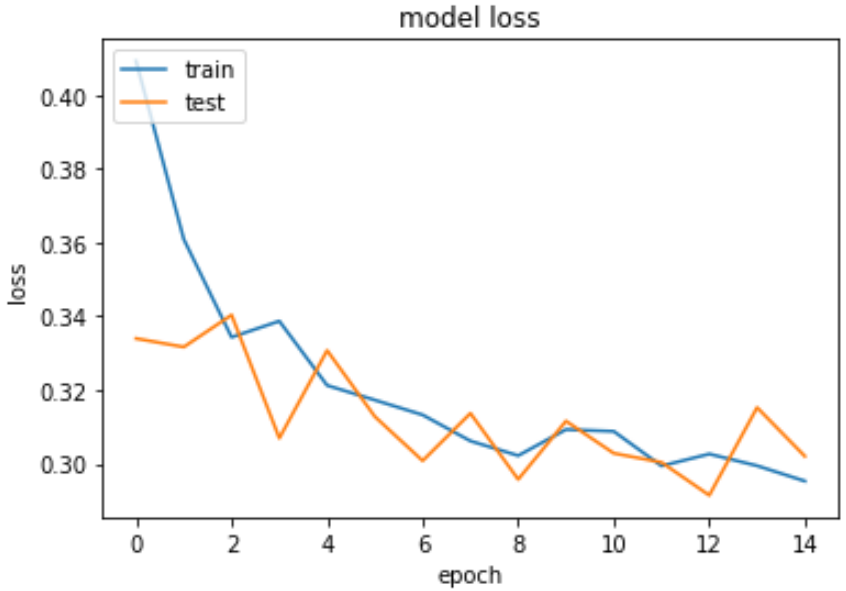
# Pretrained Model

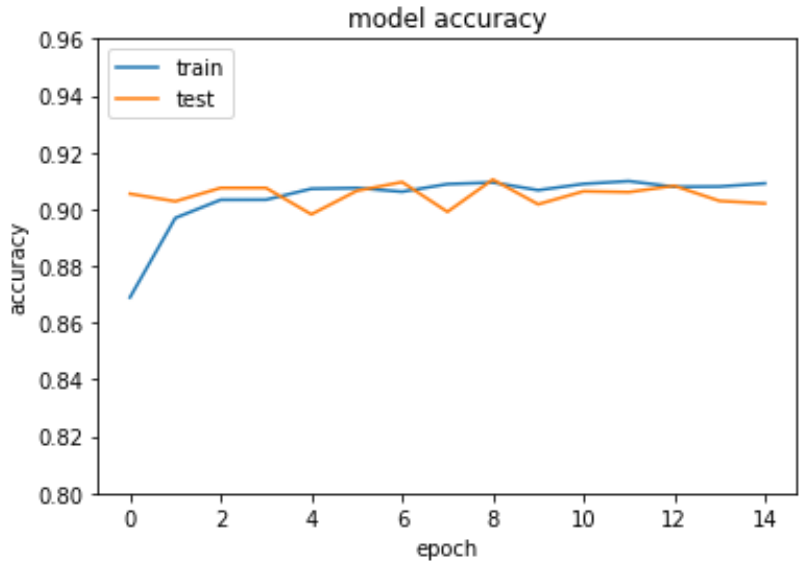


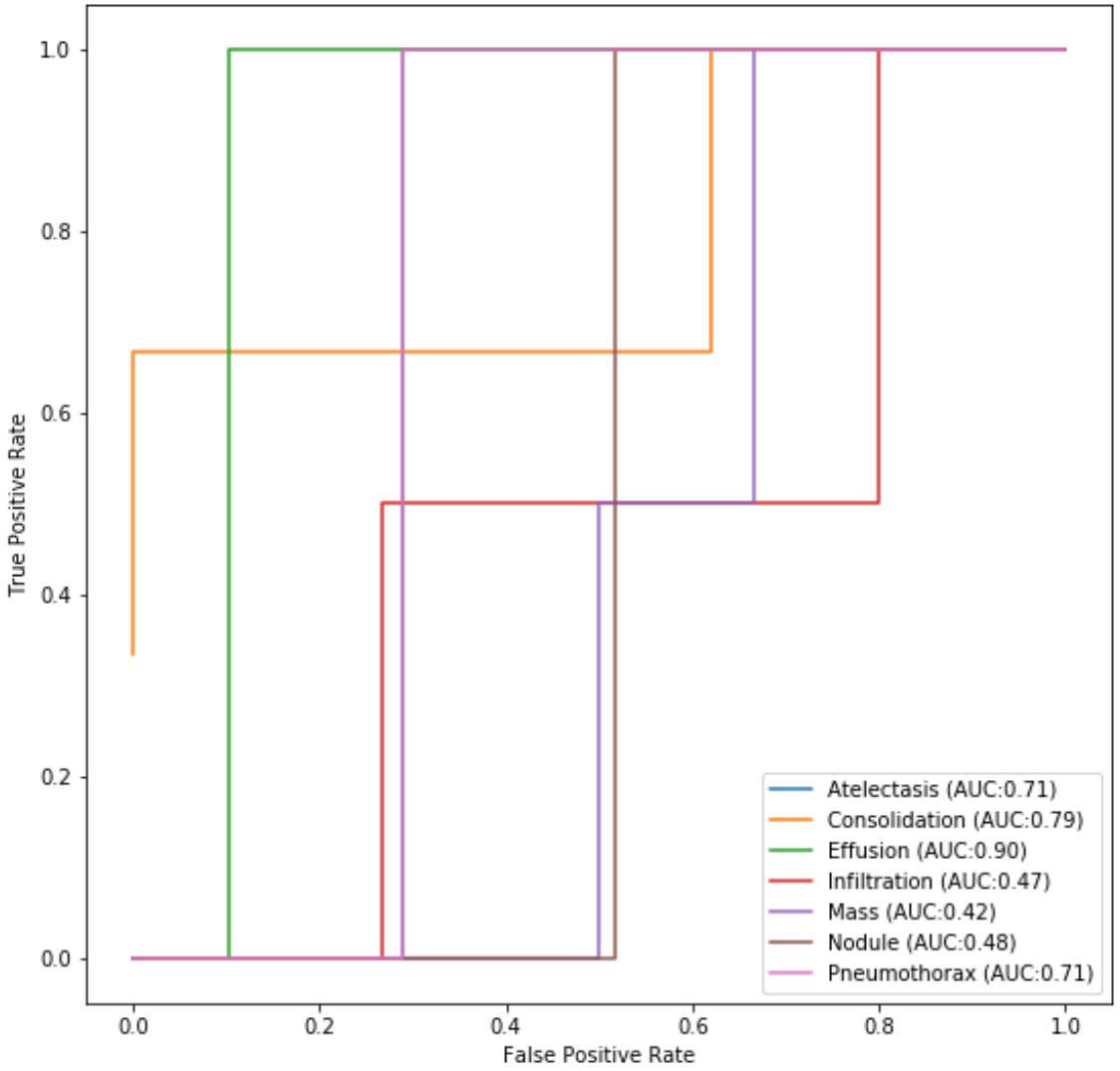



# Pretrained Model + Fine tuning

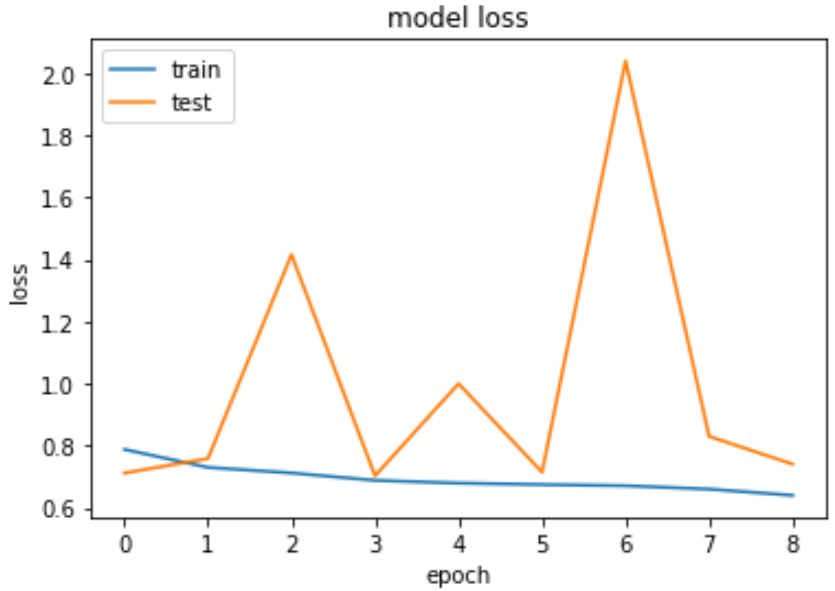


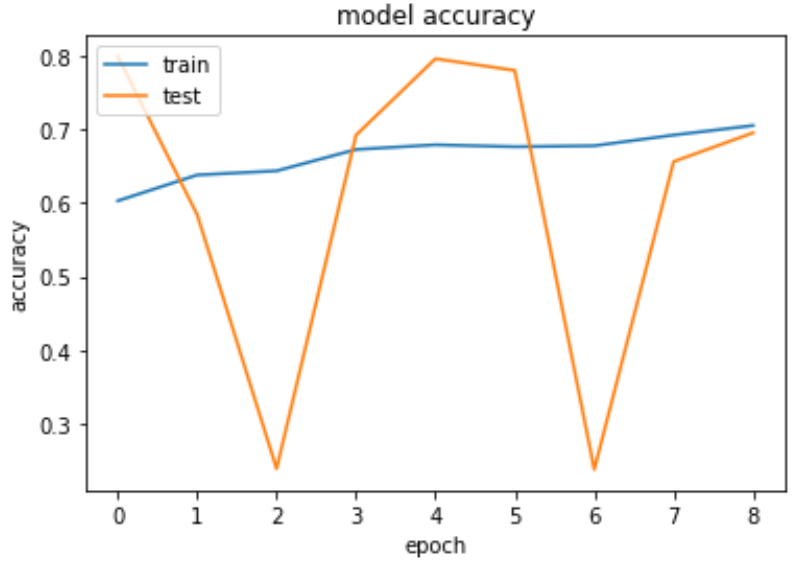



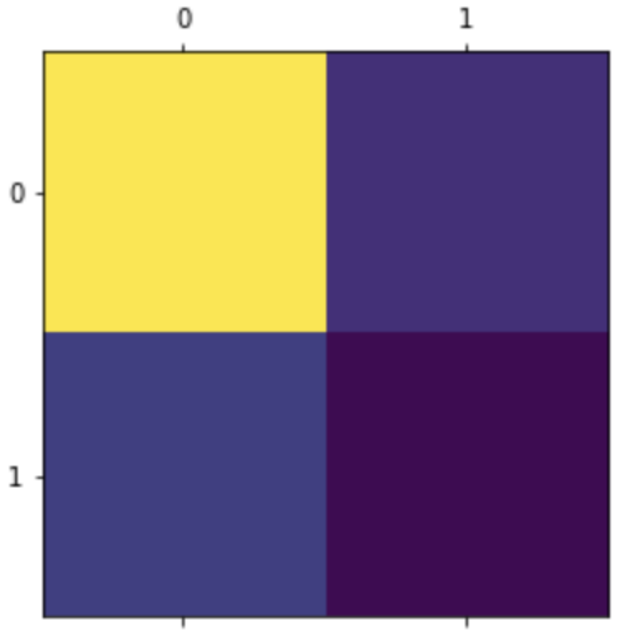
# Infiltration classification

# Pretrained Model + Fine tuning




| | precision | recall | f1-score | support |
|---|---|---|---|---|
| Else | 0.81 | 0.84 | 0.82 | 203 |
| Infiltration | 0.27 | 0.23 | 0.25 | 53 |
| avg / total | 0.70 | 0.71 | 0.70 | 256 |



# Conclusions and Outlooks

1. The low quality of the training may be because of the data set and the labeling quality.

2. Pretrained model doesn’t show obvious help here in guiding better training. More tests on other data sets could be explored to see if those pretrained model is good for medical image processing.

3. The image augmentation procedure needs to carefully tuned. Different types of modification would introduce different effects that may worsen the training.

4. More advanced technique is worth exploring such as attention matrix used in NLP. Here it may be hard for machine to detect those disease features with much stronger other features like bones.