Week 6 Progress Report

Presenter - Matt Behnke
Mentor - Dr. Guo

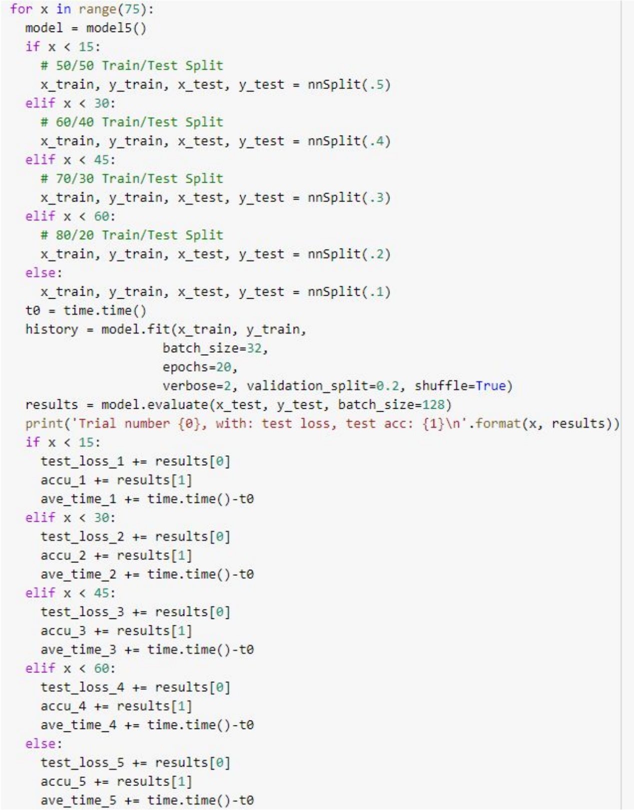
Table of Contents

1. Testing Neural Network Accuracy
2. Early Stopping for Epochs
3. Random Forest vs Early Stopping CNN
4. Using PCA to crop data
5. Using hottest pixel to crop data
6. Comparison of cropping methods

Testing Neural Network Accuracy

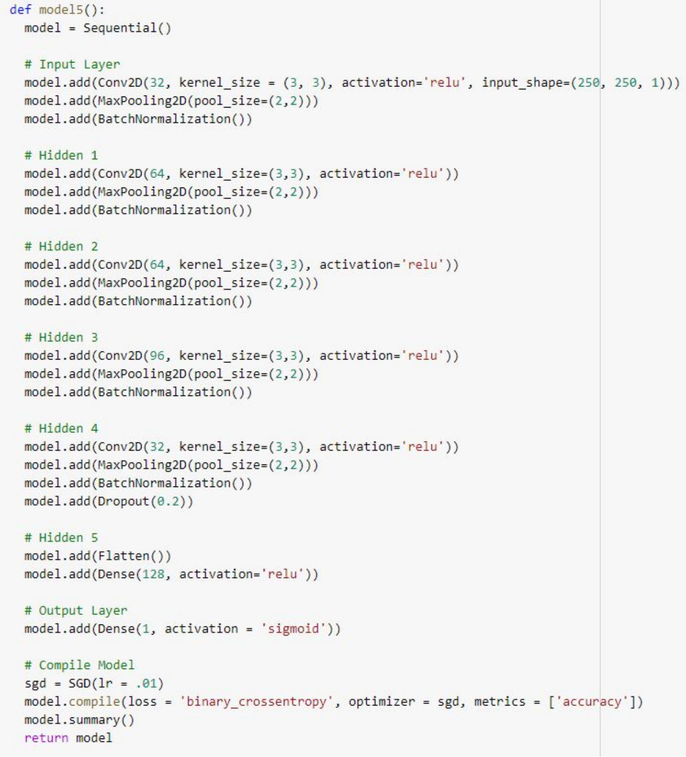
1. Test last weeks model over different splits and more trails
2. Results
3. Validation Accuracy Pattern Result

Testing the Model



```
for x in range(75):
  model = model5()
  if x < 15:
    # 50/50 Train/Test Split
    x_train, y_train, x_test, y_test = nnSplit(.5)
  elif x < 30:
    # 60/40 Train/Test Split
    x_train, y_train, x_test, y_test = nnSplit(.4)
  elif x < 45:
    # 70/30 Train/Test Split
    x_train, y_train, x_test, y_test = nnSplit(.3)
  elif x < 60:
    # 80/20 Train/Test Split
    x_train, y_train, x_test, y_test = nnSplit(.2)
  else:
    x_train, y_train, x_test, y_test = nnSplit(.1)
  t0 = time.time()
  history = model.fit(x_train, y_train,
                    batch_size=32,
                    epochs=20,
                    verbose=2, validation_split=0.2, shuffle=True)
  results = model.evaluate(x_test, y_test, batch_size=128)
  print('Trial number {0}, with: test loss, test acc: {1}\n'.format(x, results))
  if x < 15:
    test_loss_1 += results[0]
    accu_1 += results[1]
    ave_time_1 += time.time()-t0
  elif x < 30:
    test_loss_2 += results[0]
    accu_2 += results[1]
    ave_time_2 += time.time()-t0
  elif x < 45:
    test_loss_3 += results[0]
    accu_3 += results[1]
    ave_time_3 += time.time()-t0
  elif x < 60:
    test_loss_4 += results[0]
    accu_4 += results[1]
    ave_time_4 += time.time()-t0
  else:
    test_loss_5 += results[0]
    accu_5 += results[1]
    ave_time_5 += time.time()-t0
```

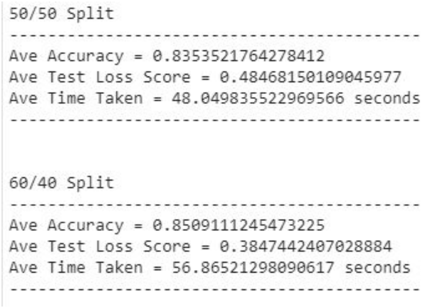
- 50 Trials Total
- 10 Trials for each (50/50, 60/40, 70/30, 80/20, 90/10 Train/Test Split)
- Recording
  - Average Test Accuracy
  - Average Test Loss
  - Average Time Taken for Training
- Each Train has
  - 20 Epochs
  - Batch Size of 32
- Every trial has a random split



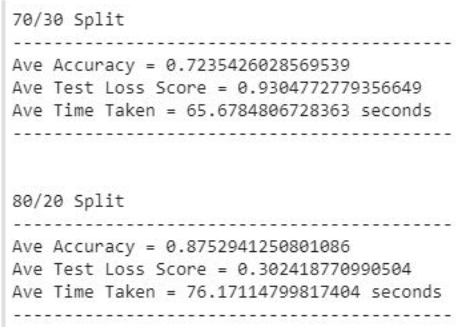
```
def model5():
  model = Sequential()

  # Input Layer
  model.add(Conv2D(32, kernel_size = (3, 3), activation='relu', input_shape=(250, 250, 1)))
  model.add(MaxPooling2D(pool_size=(2,2)))
  model.add(BatchNormalization())

  # Hidden 1
  model.add(Conv2D(64, kernel_size=(3,3), activation='relu'))
  model.add(MaxPooling2D(pool_size=(2,2)))
  model.add(BatchNormalization())

  # Hidden 2
  model.add(Conv2D(64, kernel_size=(3,3), activation='relu'))
  model.add(MaxPooling2D(pool_size=(2,2)))
  model.add(BatchNormalization())

  # Hidden 3
  model.add(Conv2D(96, kernel_size=(3,3), activation='relu'))
  model.add(MaxPooling2D(pool_size=(2,2)))
  model.add(BatchNormalization())

  # Hidden 4
  model.add(Conv2D(32, kernel_size=(3,3), activation='relu'))
  model.add(MaxPooling2D(pool_size=(2,2)))
  model.add(BatchNormalization())
  model.add(Dropout(0.2))

  # Hidden 5
  model.add(Flatten())
  model.add(Dense(128, activation='relu'))

  # Output Layer
  model.add(Dense(1, activation = 'sigmoid'))

  # Compile Model
  sgd = SGD(lr = .01)
  model.compile(loss = 'binary_crossentropy', optimizer = sgd, metrics = ['accuracy'])
  model.summary()
  return model
```

Test Results

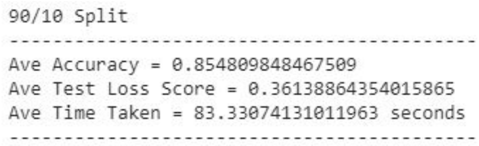
- Highest Accuracy was 87% with 80/20 split
- More Training size, the longer it took
- Not necessarily a increase in accuracy with more training data
- Not as promising as the first 5, 10% accuracy trials I had last week



```
50/50 Split
----------------------------------------
Ave Accuracy = 0.8353521764278412
Ave Test Loss Score = 0.48468150109045977
Ave Time Taken = 48.049835522969566 seconds
----------------------------------------


60/40 Split
----------------------------------------
Ave Accuracy = 0.8509111245473225
Ave Test Loss Score = 0.3847442407028884
Ave Time Taken = 56.86521298090617 seconds
----------------------------------------
```

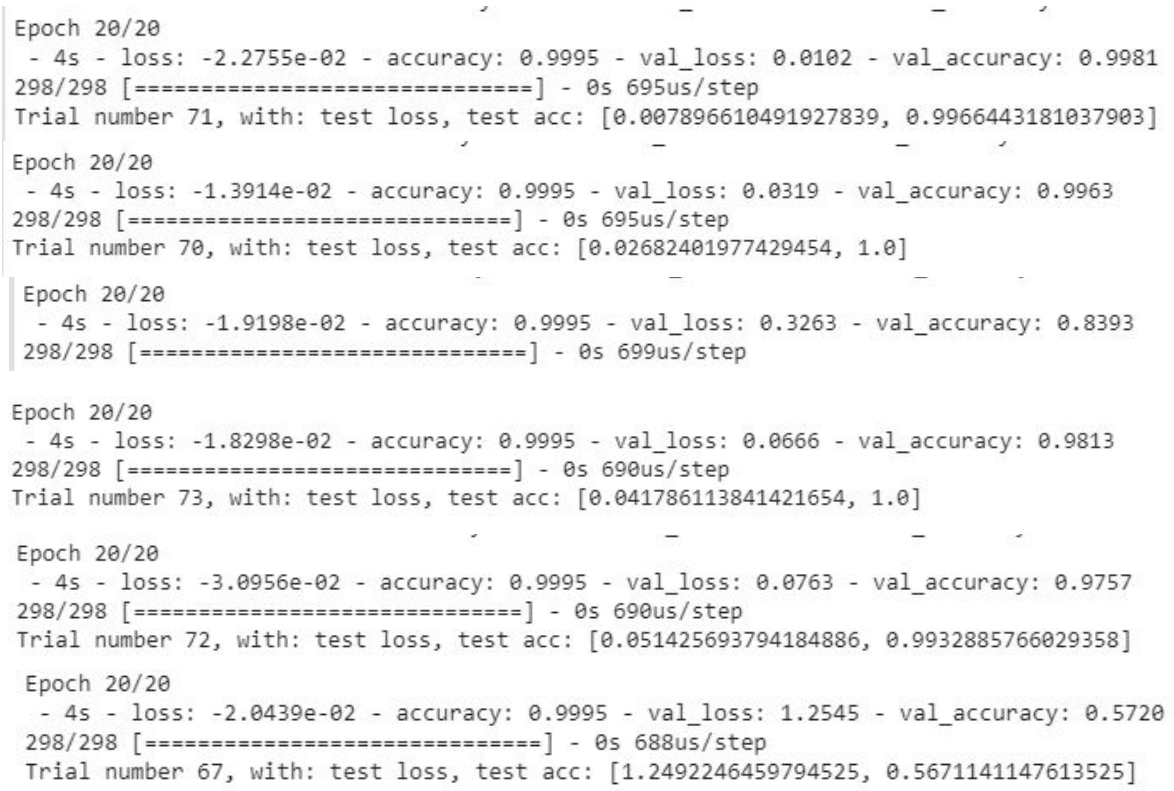


```
70/30 Split
----------------------------------------
Ave Accuracy = 0.7235426028569539
Ave Test Loss Score = 0.9304772779356649
Ave Time Taken = 65.6784806728363 seconds
----------------------------------------


80/20 Split
----------------------------------------
Ave Accuracy = 0.8752941250801086
Ave Test Loss Score = 0.302418770990504
Ave Time Taken = 76.17114799817404 seconds
----------------------------------------
```



```
90/10 Split
----------------------------------------
Ave Accuracy = 0.854809848467509
Ave Test Loss Score = 0.36138864354015865
Ave Time Taken = 83.33074131011963 seconds
----------------------------------------
```

Noticeable Trend during Data



```
Epoch 20/20
 - 4s - loss: -2.2755e-02 - accuracy: 0.9995 - val_loss: 0.0102 - val_accuracy: 0.9981
298/298 [==============================] - 0s 695us/step
Trial number 71, with: test loss, test acc: [0.007896610491927839, 0.9966443181037903]
```

```
Epoch 20/20
 - 4s - loss: -1.3914e-02 - accuracy: 0.9995 - val_loss: 0.0319 - val_accuracy: 0.9963
298/298 [==============================] - 0s 695us/step
Trial number 70, with: test loss, test acc: [0.02682401977429454, 1.0]
```

```
Epoch 20/20
 - 4s - loss: -1.9198e-02 - accuracy: 0.9995 - val_loss: 0.3263 - val_accuracy: 0.8393
298/298 [==============================] - 0s 699us/step
```

```
Epoch 20/20
 - 4s - loss: -1.8298e-02 - accuracy: 0.9995 - val_loss: 0.0666 - val_accuracy: 0.9813
298/298 [==============================] - 0s 690us/step
Trial number 73, with: test loss, test acc: [0.041786113841421654, 1.0]
```

```
Epoch 20/20
 - 4s - loss: -3.0956e-02 - accuracy: 0.9995 - val_loss: 0.0763 - val_accuracy: 0.9757
298/298 [==============================] - 0s 690us/step
Trial number 72, with: test loss, test acc: [0.051425693794184886, 0.9932885766029358]
```

```
Epoch 20/20
 - 4s - loss: -2.0439e-02 - accuracy: 0.9995 - val_loss: 1.2545 - val_accuracy: 0.5720
298/298 [==============================] - 0s 688us/step
Trial number 67, with: test loss, test acc: [1.2492246459794525, 0.5671141147613525]
```

- The validation accuracy score on the last epoch was also VERY close to the test accuracy
- Sometimes validation accuracy would drop off on the last epoch and therefore adversely affected test accuracy score
  - Ex: Epoch 19 had validation score of .99, then epoch 20 had validation score of .5, then test accuracy would be around .5 instead of .99
  - Pretty sure this was a result of overfitting

Early Stopping for Epochs

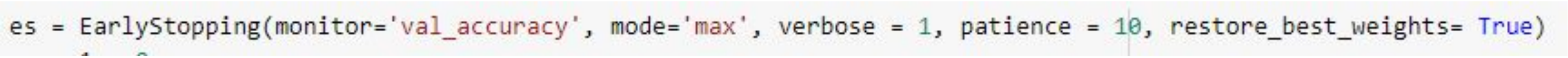
1. API Details and Parameter Choices
2. Results

Early Stopping API

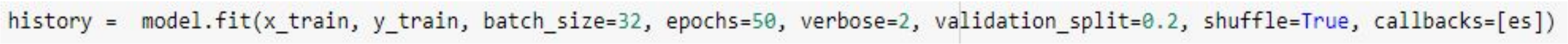
- Keras provides a callback API which can set a patience and restore best weights functionality to a neural network
  - Patience is a parameter defined as the number of epochs a model will keep training without an improvement in a specified metric
  - Restore Best Weights is a parameter which will restore the best epoch by a specified metric (as long as the model has not trained on its last epoch)

Using Early Stopping API

- Used validation accuracy as a metric to perform an early stop
- Used a patience of 10 and ramped up epochs to 50
  - By increasing the epochs, it gave a high probability that the last epoch would not end on a low validation accuracy
  - Patience of 10 gave a good chance of having a high validation accuracy

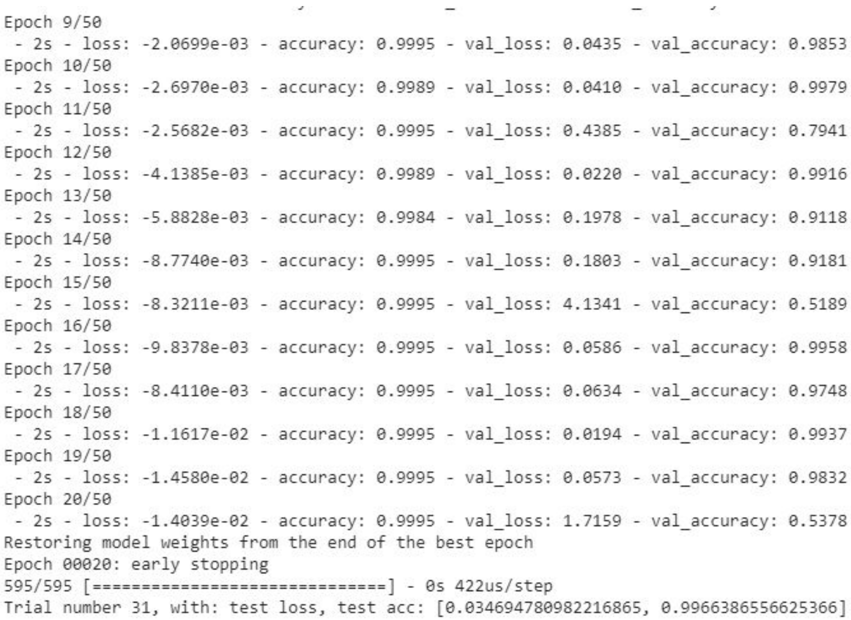


```
es = EarlyStopping(monitor='val_accuracy', mode='max', verbose = 1, patience = 10, restore_best_weights= True)
```



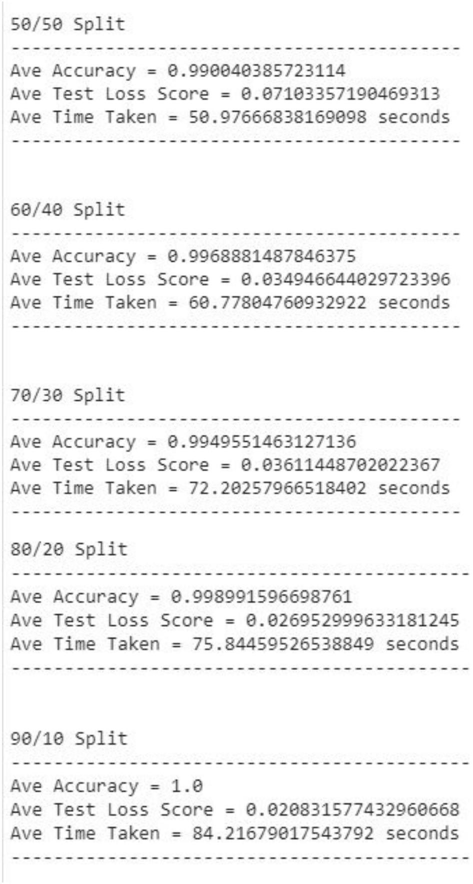
```
history =  model.fit(x_train, y_train, batch_size=32, epochs=50, verbose=2, validation_split=0.2, shuffle=True, callbacks=[es])
```

Early Stopping Example



```
Epoch 9/50
 - 2s - loss: -2.0699e-03 - accuracy: 0.9995 - val_loss: 0.0435 - val_accuracy: 0.9853
Epoch 10/50
 - 2s - loss: -2.6970e-03 - accuracy: 0.9989 - val_loss: 0.0410 - val_accuracy: 0.9979
Epoch 11/50
 - 2s - loss: -2.5682e-03 - accuracy: 0.9995 - val_loss: 0.4385 - val_accuracy: 0.7941
Epoch 12/50
 - 2s - loss: -4.1385e-03 - accuracy: 0.9989 - val_loss: 0.0220 - val_accuracy: 0.9916
Epoch 13/50
 - 2s - loss: -5.8828e-03 - accuracy: 0.9984 - val_loss: 0.1978 - val_accuracy: 0.9118
Epoch 14/50
 - 2s - loss: -8.7740e-03 - accuracy: 0.9995 - val_loss: 0.1803 - val_accuracy: 0.9181
Epoch 15/50
 - 2s - loss: -8.3211e-03 - accuracy: 0.9995 - val_loss: 4.1341 - val_accuracy: 0.5189
Epoch 16/50
 - 2s - loss: -9.8378e-03 - accuracy: 0.9995 - val_loss: 0.0586 - val_accuracy: 0.9958
Epoch 17/50
 - 2s - loss: -8.4110e-03 - accuracy: 0.9995 - val_loss: 0.0634 - val_accuracy: 0.9748
Epoch 18/50
 - 2s - loss: -1.1617e-02 - accuracy: 0.9995 - val_loss: 0.0194 - val_accuracy: 0.9937
Epoch 19/50
 - 2s - loss: -1.4580e-02 - accuracy: 0.9995 - val_loss: 0.0573 - val_accuracy: 0.9832
Epoch 20/50
 - 2s - loss: -1.4039e-02 - accuracy: 0.9995 - val_loss: 1.7159 - val_accuracy: 0.5378
Restoring model weights from the end of the best epoch
Epoch 00020: early stopping
595/595 [==============================] - 0s 422us/step
Trial number 31, with: test loss, test acc: [0.034694780982216865, 0.9966386556625366]
```

Early Stopping NN Results



```
50/50 Split
-------------------------------------------
Ave Accuracy = 0.990040385723114
Ave Test Loss Score = 0.07103357190469313
Ave Time Taken = 50.97666838169098 seconds
-------------------------------------------


60/40 Split
-------------------------------------------
Ave Accuracy = 0.9968881487846375
Ave Test Loss Score = 0.034946644029723396
Ave Time Taken = 60.77804760932922 seconds
-------------------------------------------


70/30 Split
-------------------------------------------
Ave Accuracy = 0.9949551463127136
Ave Test Loss Score = 0.03611448702022367
Ave Time Taken = 72.20257966518402 seconds
-------------------------------------------

80/20 Split
--------------------------------------------
Ave Accuracy = 0.998991596698761
Ave Test Loss Score = 0.026952999633181245
Ave Time Taken = 75.84459526538849 seconds
--------------------------------------------


90/10 Split
--------------------------------------------
Ave Accuracy = 1.0
Ave Test Loss Score = 0.020831577432960668
Ave Time Taken = 84.21679017543792 seconds
--------------------------------------------
```

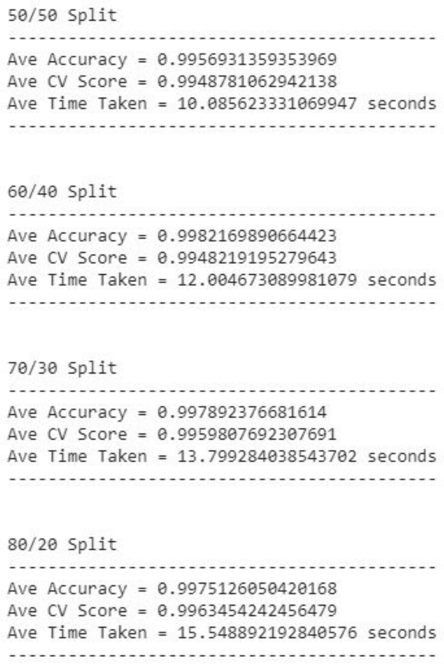
- Tested the same as the before
  - 10 Trials per train/test split
  - Every individual trail has a random split
- All test splits had accuracy of 99% and above
- 90/10 Split had a perfect accuracy
- Does not take a short amount of time per train/test

Random Forest vs Early Stopping CNN

1. Comparing the two models
2. Metrics/Results

Random Forest vs Early Stopping NN

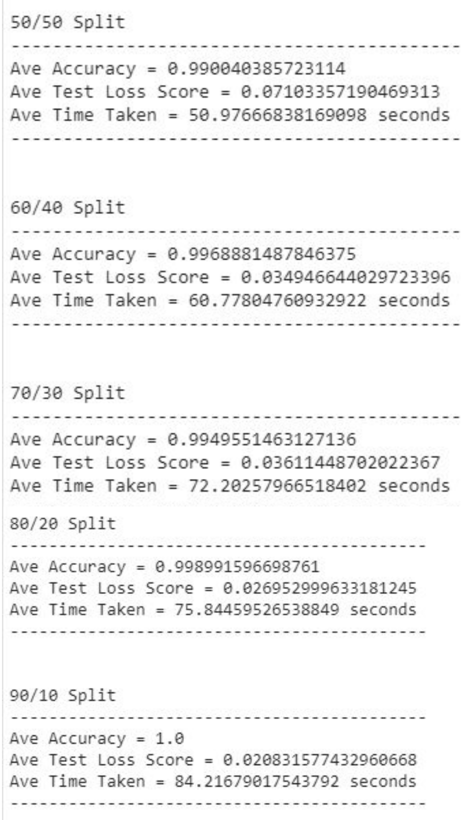
Random Forest



```
50/50 Split
------------------------------------------
Ave Accuracy = 0.9956931359353969
Ave CV Score = 0.9948781062942138
Ave Time Taken = 10.085623331069947 seconds
------------------------------------------


60/40 Split
------------------------------------------
Ave Accuracy = 0.9982169890664423
Ave CV Score = 0.9948219195279643
Ave Time Taken = 12.004673089981079 seconds
------------------------------------------


70/30 Split
------------------------------------------
Ave Accuracy = 0.997892376681614
Ave CV Score = 0.9959807692307691
Ave Time Taken = 13.799284038543702 seconds
------------------------------------------


80/20 Split
------------------------------------------
Ave Accuracy = 0.9975126050420168
Ave CV Score = 0.9963454242456479
Ave Time Taken = 15.548892192840576 seconds
------------------------------------------
```

- Random forest had 25 trials for each split
- Early Stopping NN has 10 trials per each split
- Very similar accuracy scored
- Both had test accuracy scores for all splits over 99%
- Random Forest is significantly shorter

Early Stopping NN



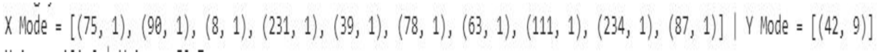
```
50/50 Split
------------------------------------------
Ave Accuracy = 0.990040385723114
Ave Test Loss Score = 0.07103357190469313
Ave Time Taken = 50.97666838169098 seconds
------------------------------------------


60/40 Split
------------------------------------------
Ave Accuracy = 0.9968881487846375
Ave Test Loss Score = 0.034946644029723396
Ave Time Taken = 60.77804760932922 seconds
------------------------------------------


70/30 Split
------------------------------------------
Ave Accuracy = 0.9949551463127136
Ave Test Loss Score = 0.03611448702022367
Ave Time Taken = 72.20257966518402 seconds
80/20 Split
----------------------------------------
Ave Accuracy = 0.998991596698761
Ave Test Loss Score = 0.026952999633181245
Ave Time Taken = 75.84459526538849 seconds
----------------------------------------


90/10 Split
----------------------------------------
Ave Accuracy = 1.0
Ave Test Loss Score = 0.020831577432960668
Ave Time Taken = 84.21679017543792 seconds
----------------------------------------
```

Using PCA to Crop Data Files

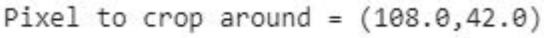
1. How is PCA being used to crop data files
2. Visualizing new Crop Files
3. Testing new files on current Models
    a. Results from Random Forest
    b. Results from Early Stopping NN

FInding an Area to Crop

- Continuing off last week where 40 components was necessary to achieve 95 % variance
- From the 40 components and the top 10 most important pixels from each component
  - Use the mode of the top 10 most important pixel, if no mode, use the average of the 10 top most important pixels
  - Find the average of all 40 of the x and y modes (or average, if applicable)
  - Use that x,y combo to crop around



```
X Mode = [(75, 1), (90, 1), (8, 1), (231, 1), (39, 1), (78, 1), (63, 1), (111, 1), (234, 1), (87, 1)] | Y Mode = [(42, 9)]
```

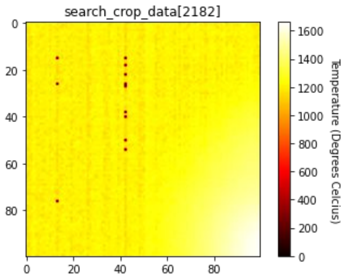


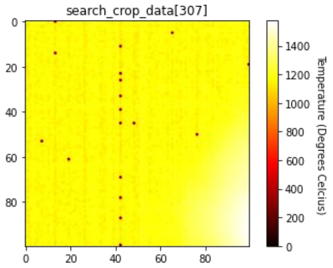
```
Pixel to crop around = (108.0,42.0)
```

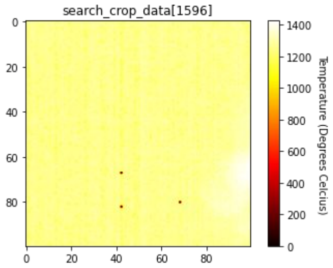
Creating a smaller crop from already cropped data

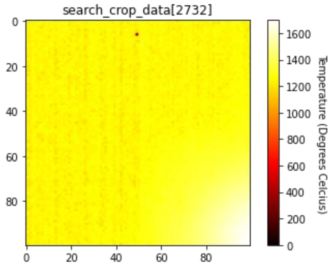
- Chose to make an even smaller crop, because PCA on the original data took an astronomical amount of time/space
- See if the files can be minimized even further beyond the original crop of the melt pool
- Chose to do a 100x100 pixel crop

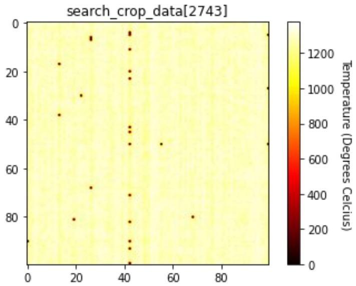
Visualizations of PCA Crop

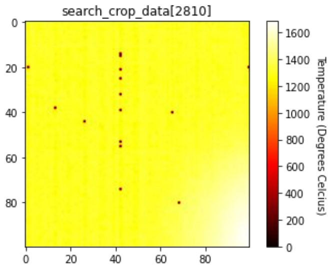

search_crop_data[2182]

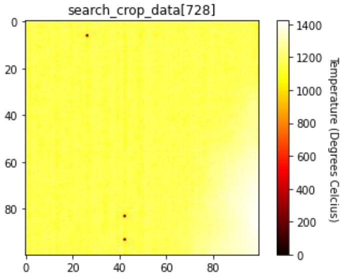

search_crop_data[307]

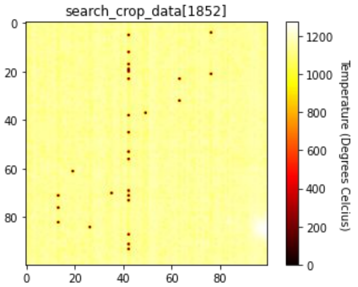

search_crop_data[1596]


search_crop_data[2732]

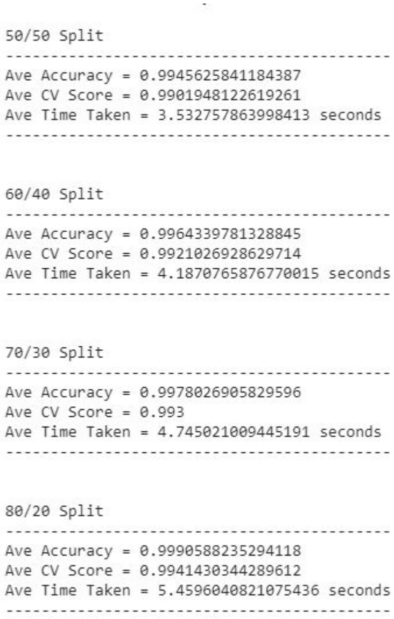

search_crop_data[2743]


search_crop_data[2810]


search_crop_data[728]


search_crop_data[1852]

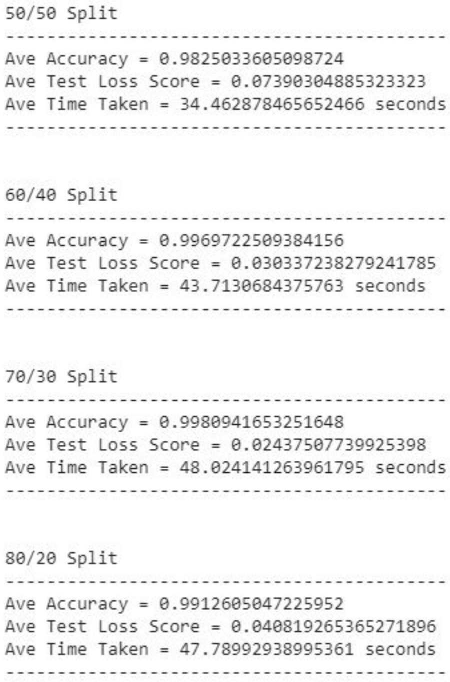
Results with Random Forest



```
50/50 Split
-------------------------------------------
Ave Accuracy = 0.9945625841184387
Ave CV Score = 0.9901948122619261
Ave Time Taken = 3.532757863998413 seconds
-------------------------------------------


60/40 Split
-------------------------------------------
Ave Accuracy = 0.9964339781328845
Ave CV Score = 0.9921026928629714
Ave Time Taken = 4.1870765876770015 seconds
-------------------------------------------


70/30 Split
-------------------------------------------
Ave Accuracy = 0.9978026905829596
Ave CV Score = 0.993
Ave Time Taken = 4.745021009445191 seconds
-------------------------------------------


80/20 Split
-------------------------------------------
Ave Accuracy = 0.9990588235294118
Ave CV Score = 0.9941430344289612
Ave Time Taken = 5.4596040821075436 seconds
-------------------------------------------
```

- 100 Trials
- 25 for each 50/50, 60/40, 70/30, 80/20 train/test split
  - Random split train/test split for each individual trial
- All had greater than 99% test accuracy
- Took a very short time

Results with Early Stopping NN



```
50/50 Split
-------------------------------------------
Ave Accuracy = 0.9825033605098724
Ave Test Loss Score = 0.07390304885323323
Ave Time Taken = 34.462878465652466 seconds
-------------------------------------------


60/40 Split
-------------------------------------------
Ave Accuracy = 0.9969722509384156
Ave Test Loss Score = 0.030337238279241785
Ave Time Taken = 43.7130684375763 seconds
-------------------------------------------


70/30 Split
-------------------------------------------
Ave Accuracy = 0.9980941653251648
Ave Test Loss Score = 0.02437507739925398
Ave Time Taken = 48.024141263961795 seconds
-------------------------------------------


80/20 Split
-------------------------------------------
Ave Accuracy = 0.9912605047225952
Ave Test Loss Score = 0.040819265365271896
Ave Time Taken = 47.78992938995361 seconds
-------------------------------------------
```

- 10 trials each with 50/50, 60/40, 70/30, 80/20 training/test split
- Each got an average accuracy over 98%
    - 50/50 split the lowest with ~98%

Using Hottest Pixel to Crop Data File

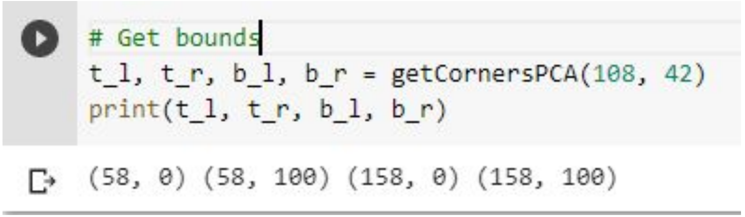
1. Algorithm/Procedure on cropping the data file

2. Visualizing new Crop Files

3. Testing new files on current Models
   a. Results from Random Forest
   b. Results from Early Stopping NN

Search for Hottest Pixel + Crop Algorithm

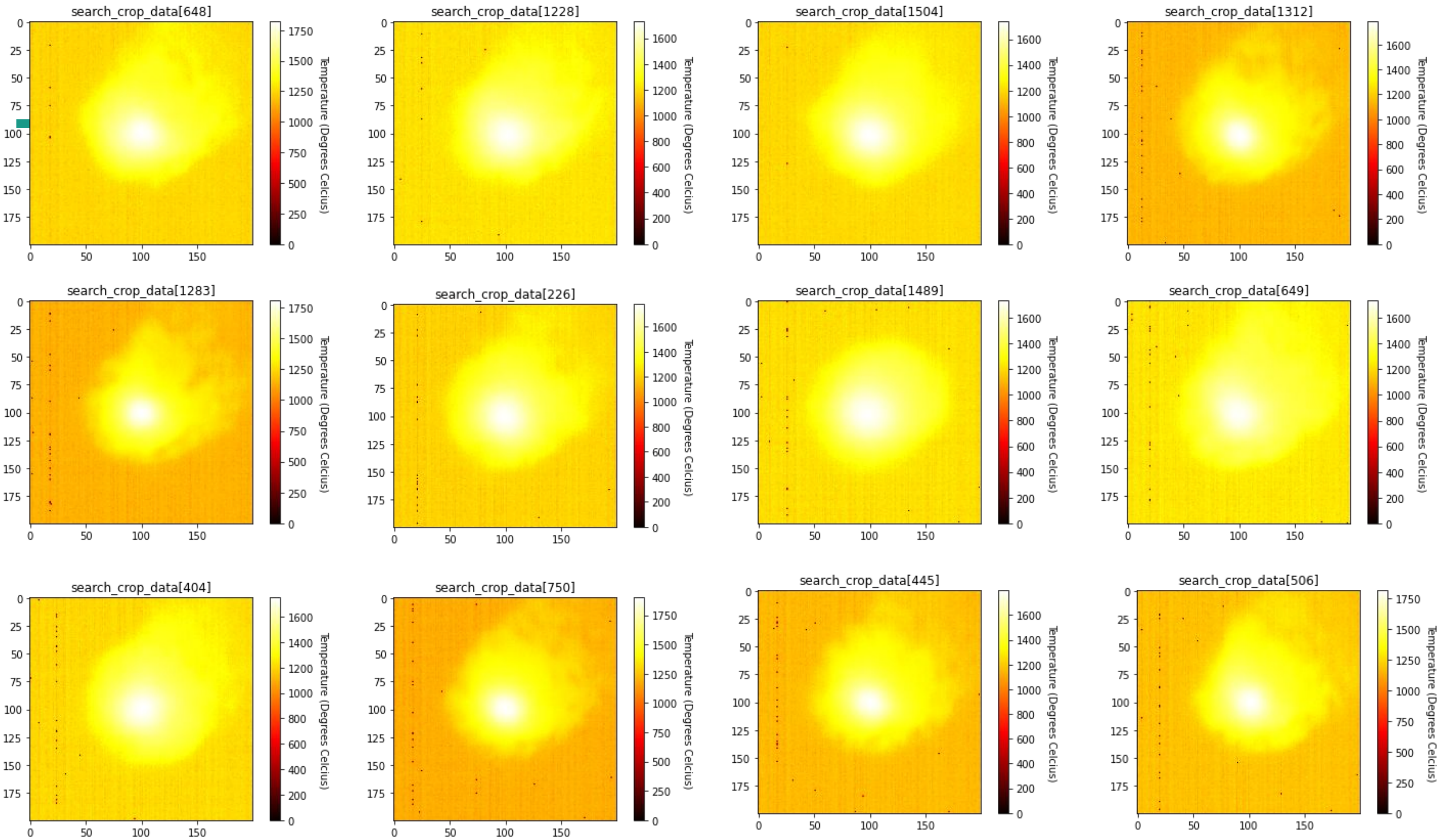
- Locate hottest pixel
  - If there are is a tie between multiple, take the average of them
- Create a ‘box’ crop around it
  - Box is 50 pixels right, above, left, below it
  - If the pixel is closer than 50 pixels to an edge, then crop to the edge and make other side make up for it
- Use those in the Random Forest and NN Model



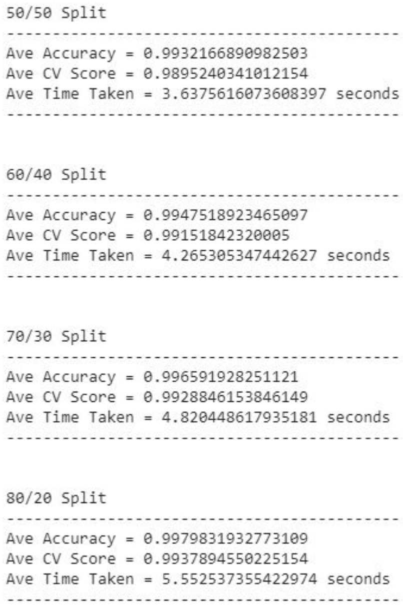
```
# Get bounds
t_l, t_r, b_l, b_r = getCornersPCA(108, 42)
print(t_l, t_r, b_l, b_r)
```

```
(58, 0) (58, 100) (158, 0) (158, 100)
```

Visualizations of Crops



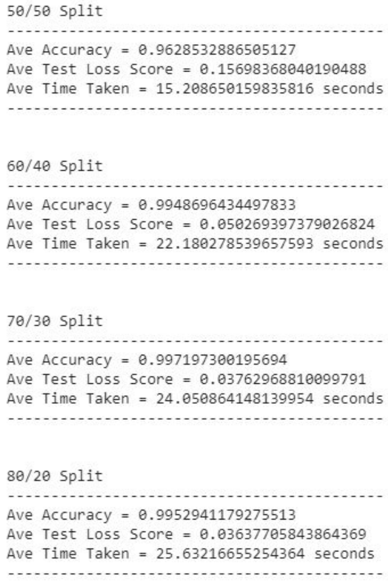
Results with Random Forest



```
50/50 Split
-------------------------------------------
Ave Accuracy = 0.9932166890982503
Ave CV Score = 0.9895240341012154
Ave Time Taken = 3.6375616073608397 seconds
-------------------------------------------


60/40 Split
-------------------------------------------
Ave Accuracy = 0.9947518923465097
Ave CV Score = 0.99151842320005
Ave Time Taken = 4.265305347442627 seconds
-------------------------------------------


70/30 Split
-------------------------------------------
Ave Accuracy = 0.996591928251121
Ave CV Score = 0.9928846153846149
Ave Time Taken = 4.820448617935181 seconds
-------------------------------------------


80/20 Split
-------------------------------------------
Ave Accuracy = 0.9979831932773109
Ave CV Score = 0.9937894550225154
Ave Time Taken = 5.552537355422974 seconds
-------------------------------------------
```

- 25 Trials for each split
- All over 99% average accuracy
- Very short times (due to less data)

Results with Early Stopping NN



```
50/50 Split
-------------------------------------------
Ave Accuracy = 0.9628532886505127
Ave Test Loss Score = 0.15698368040190488
Ave Time Taken = 15.208650159835816 seconds
-------------------------------------------


60/40 Split
-------------------------------------------
Ave Accuracy = 0.9948696434497833
Ave Test Loss Score = 0.050269397379026824
Ave Time Taken = 22.180278539657593 seconds
-------------------------------------------


70/30 Split
-------------------------------------------
Ave Accuracy = 0.997197300195694
Ave Test Loss Score = 0.03762968810099791
Ave Time Taken = 24.050864148139954 seconds
-------------------------------------------


80/20 Split
-------------------------------------------
Ave Accuracy = 0.9952941179275513
Ave Test Loss Score = 0.03637705843864369
Ave Time Taken = 25.63216655254364 seconds
-------------------------------------------
```

- Lowest result of an average of ~96% test accuracy
  - 50/50 train/test split
- All other train/test splits had an average accuracy over 99%
- Much shorter time due to less data because of crop

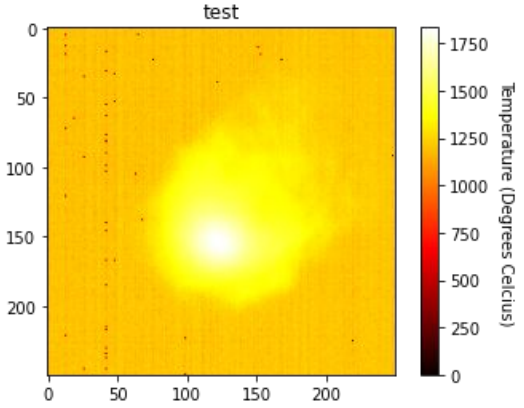
Comparing Cropping Methods

1. Visualization of Each

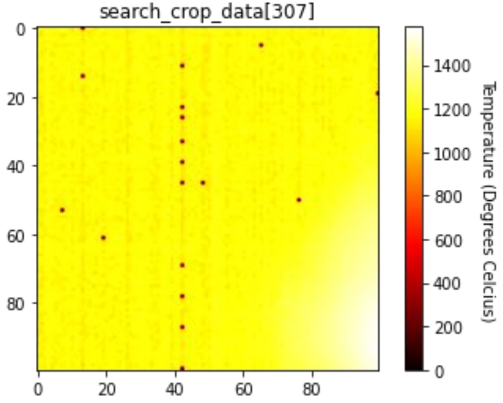
2. Results
   a. Compare all with Random Forest
   b. Compare all with CNN

Visualization of Each Crop Style

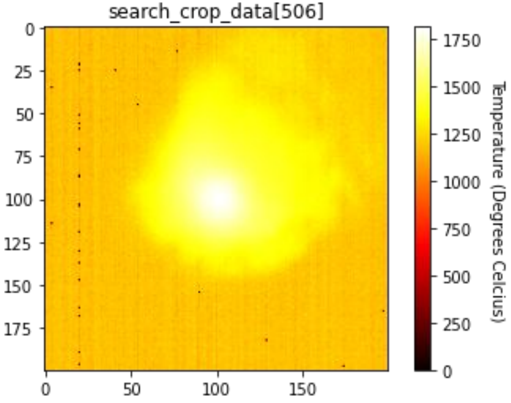
Manual Visual Crop



PCA Crop

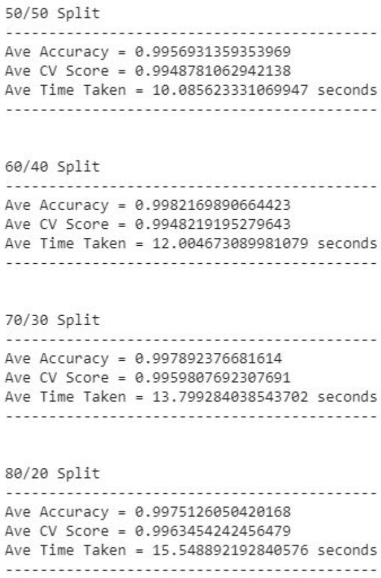


Hottest Pixel Crop



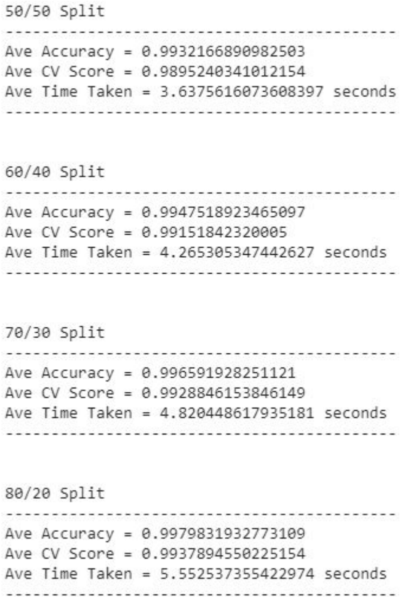
Comparing Crop Results (Random Forest)

Manual



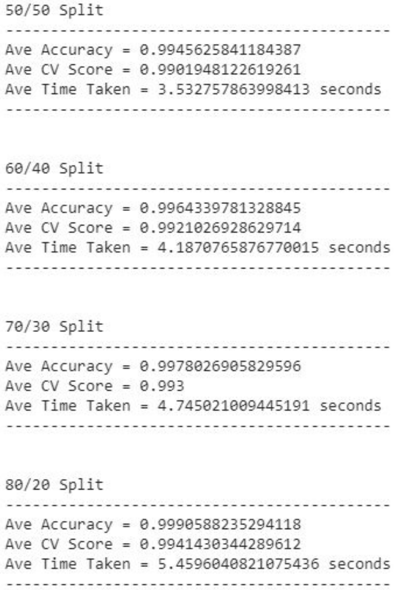
```
50/50 Split
-------------------------------------------
Ave Accuracy = 0.9956931359353969
Ave CV Score = 0.9948781062942138
Ave Time Taken = 10.085623331069947 seconds
-------------------------------------------


60/40 Split
-------------------------------------------
Ave Accuracy = 0.9982169890664423
Ave CV Score = 0.9948219195279643
Ave Time Taken = 12.004673089981079 seconds
-------------------------------------------


70/30 Split
-------------------------------------------
Ave Accuracy = 0.997892376681614
Ave CV Score = 0.9959807692307691
Ave Time Taken = 13.799284038543702 seconds
-------------------------------------------


80/20 Split
-------------------------------------------
Ave Accuracy = 0.9975126050420168
Ave CV Score = 0.9963454242456479
Ave Time Taken = 15.548892192840576 seconds
-------------------------------------------
```

PCA



```
50/50 Split
-------------------------------------------
Ave Accuracy = 0.9932166890982503
Ave CV Score = 0.9895240341012154
Ave Time Taken = 3.6375616073608397 seconds
-------------------------------------------


60/40 Split
-------------------------------------------
Ave Accuracy = 0.9947518923465097
Ave CV Score = 0.99151842320005
Ave Time Taken = 4.265305347442627 seconds
-------------------------------------------


70/30 Split
-------------------------------------------
Ave Accuracy = 0.996591928251121
Ave CV Score = 0.9928846153846149
Ave Time Taken = 4.820448617935181 seconds
-------------------------------------------


80/20 Split
-------------------------------------------
Ave Accuracy = 0.9979831932773109
Ave CV Score = 0.9937894550225154
Ave Time Taken = 5.552537355422974 seconds
-------------------------------------------
```

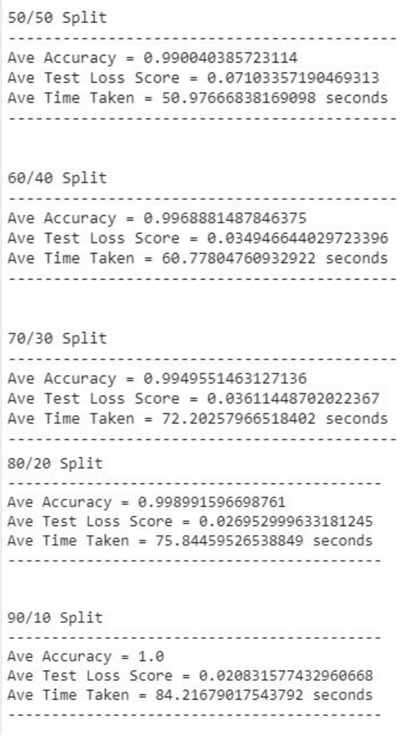
Hottest Pixel



```
50/50 Split
-------------------------------------------
Ave Accuracy = 0.9945625841184387
Ave CV Score = 0.9901948122619261
Ave Time Taken = 3.532757863998413 seconds
-------------------------------------------


60/40 Split
-------------------------------------------
Ave Accuracy = 0.9964339781328845
Ave CV Score = 0.9921026928629714
Ave Time Taken = 4.1870765876770015 seconds
-------------------------------------------


70/30 Split
-------------------------------------------
Ave Accuracy = 0.9978026905829596
Ave CV Score = 0.993
Ave Time Taken = 4.745021009445191 seconds
-------------------------------------------


80/20 Split
-------------------------------------------
Ave Accuracy = 0.9990588235294118
Ave CV Score = 0.9941430344289612
Ave Time Taken = 5.4596040821075436 seconds
-------------------------------------------
```

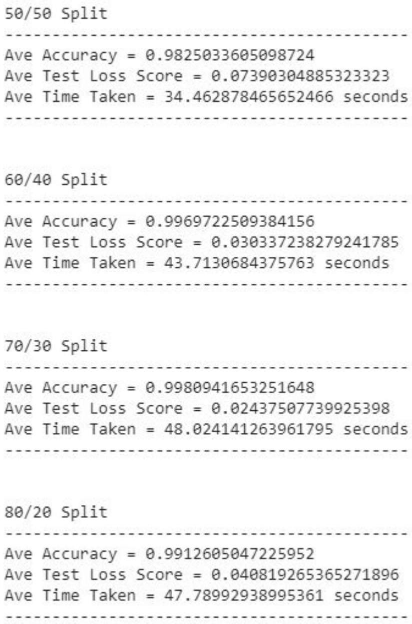
Comparing Crop Results (Early Stopping NN)

Manual



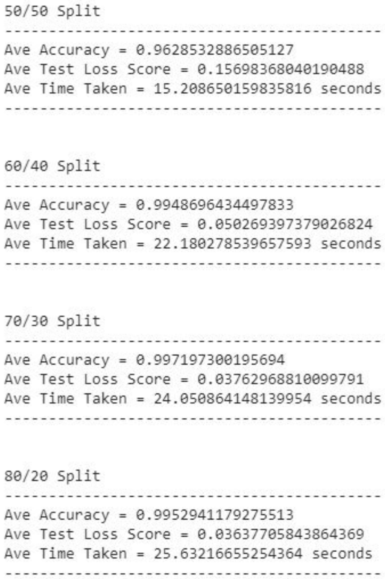
```
50/50 Split
--------------------------------------------
Ave Accuracy = 0.990040385723114
Ave Test Loss Score = 0.07103357190469313
Ave Time Taken = 50.97666838169098 seconds
--------------------------------------------


60/40 Split
--------------------------------------------
Ave Accuracy = 0.9968881487846375
Ave Test Loss Score = 0.034946644029723396
Ave Time Taken = 60.77804760932922 seconds
--------------------------------------------


70/30 Split
--------------------------------------------
Ave Accuracy = 0.9949551463127136
Ave Test Loss Score = 0.03611448702022367
Ave Time Taken = 72.20257966518402 seconds
--------------------------------------------
80/20 Split
------------------------------------------
Ave Accuracy = 0.998991596698761
Ave Test Loss Score = 0.026952999633181245
Ave Time Taken = 75.84459526538849 seconds
------------------------------------------


90/10 Split
------------------------------------------
Ave Accuracy = 1.0
Ave Test Loss Score = 0.020831577432960668
Ave Time Taken = 84.21679017543792 seconds
------------------------------------------
```

PCA



```
50/50 Split
--------------------------------------------
Ave Accuracy = 0.9825033605098724
Ave Test Loss Score = 0.07390304885323323
Ave Time Taken = 34.462878465652466 seconds
--------------------------------------------


60/40 Split
--------------------------------------------
Ave Accuracy = 0.9969722509384156
Ave Test Loss Score = 0.030337238279241785
Ave Time Taken = 43.7130684375763 seconds
--------------------------------------------


70/30 Split
--------------------------------------------
Ave Accuracy = 0.9980941653251648
Ave Test Loss Score = 0.02437507739925398
Ave Time Taken = 48.024141263961795 seconds
--------------------------------------------


80/20 Split
--------------------------------------------
Ave Accuracy = 0.9912605047225952
Ave Test Loss Score = 0.040819265365271896
Ave Time Taken = 47.78992938995361 seconds
--------------------------------------------
```

Hottest Pixel



```
50/50 Split
--------------------------------------------
Ave Accuracy = 0.9628532886505127
Ave Test Loss Score = 0.15698368040190488
Ave Time Taken = 15.208650159835816 seconds
--------------------------------------------


60/40 Split
--------------------------------------------
Ave Accuracy = 0.9948696434497833
Ave Test Loss Score = 0.050269397379026824
Ave Time Taken = 22.180278539657593 seconds
--------------------------------------------


70/30 Split
--------------------------------------------
Ave Accuracy = 0.997197300195694
Ave Test Loss Score = 0.03762968810099791
Ave Time Taken = 24.050864148139954 seconds
--------------------------------------------


80/20 Split
--------------------------------------------
Ave Accuracy = 0.9952941179275513
Ave Test Loss Score = 0.03637705843864369
Ave Time Taken = 25.63216655254364 seconds
--------------------------------------------
```

Future Goals / Next Steps

- Combining a hottest pixel search + pca after?
- Trying lower train and higher testing ratios?
- Finding which datafiles are being labeled incorrectly?
- Suggestions?