

Coin Year Classification

Holden Gordon
Derek Halliwell

ELEN 520: Spring 2019



Problem Statement

- We will attempt to identify coin years from images of U.S. pennies
- H_0 : Image pixel values are not correlated with the year of the coin
- H_1 : Image pixel values are correlated with the year of the coin



Background

- Worth of coin is determined by year and condition
- Everyone has loose change. Could be valuable
- Ongoing debate about phasing out pennies

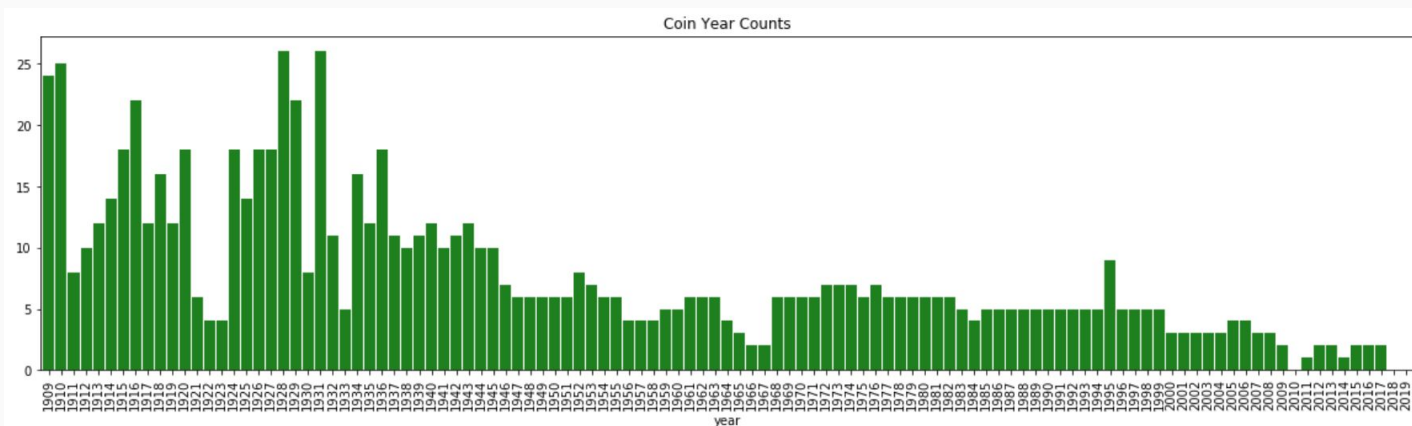
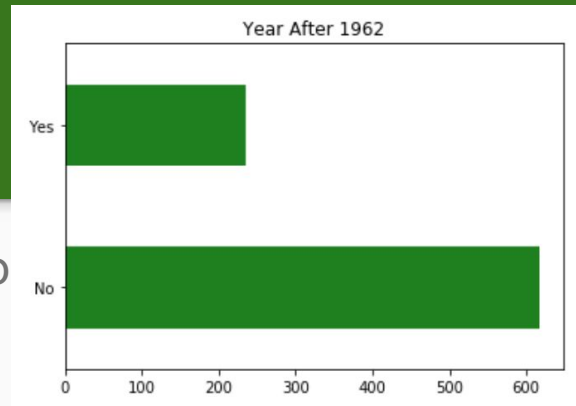
Ultra-Rare Penny Sells for \$204,000 at Auction

FORTUNE Chris Morris
Fortune January 11, 2019, 6:21 PM UTC



Target Variable

- We chose to predict whether a coin is made before or after 1963.
- The alternative is a target with 111 classes



Data

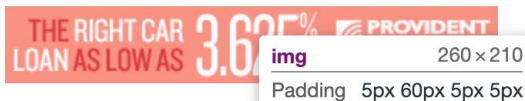
- Training data: We found 845 labeled penny images online
- Training data was scraped with the Selenium python package
 - Image
 - Year
- Test data was collected with camera connected to PYNQ board



Scraping Images with Python

1963 Lincoln Penny

CoinTrackers.com has estimated the 1963 Lincoln Penny value at an average of 1 cent, certified mint state (MS+) could be worth \$15. ([see details](#))...



Type: [Lincoln Penny](#)

Year: [1963](#)

Mint Mark: No mint mark

Face Value: 0.01 USD

Total Produced: 754,110,000 [?]

Silver Content: 0%

Numismatic Value: 1 cent to \$15.00

Value: As a rough estimate of this coins value you can assume this coin in average condition will be valued at somewhere around **1 cent**, while one in certified mint state (MS+) condition could bring as much as **\$15** at auction. This price does not reference any standard [coin grading](#) scale. So when we say average, we mean in a similar condition to other coins issued in



```
><head>...</head>
▼<body>
  ▼<div class="container">
    ▶<div class="header">...</div>
    ▶<div class="navigation">...</div>
    ▼<div class="main">
      ▼<div class="content">
        ▶<div id="breadcrumbs">...</div>
        <h1>1963 Lincoln Penny</h1>
        ▶<p>...</p>
        ▶<p align="center">...</p>
        ▼<div class="right">
          ...
           == $0
          <br>
```

Extract Image URL

```
image_xpath = '//*[@div[1]/div[3]/div[1]/div[2]/img'
```

```
image = driver.find_element(By.XPATH, image_xpath)
```

```
image_url = image.get_attribute("src")
```

Save Image

```
destination = 'ct/' + name + '.png'
```

```
try:
```

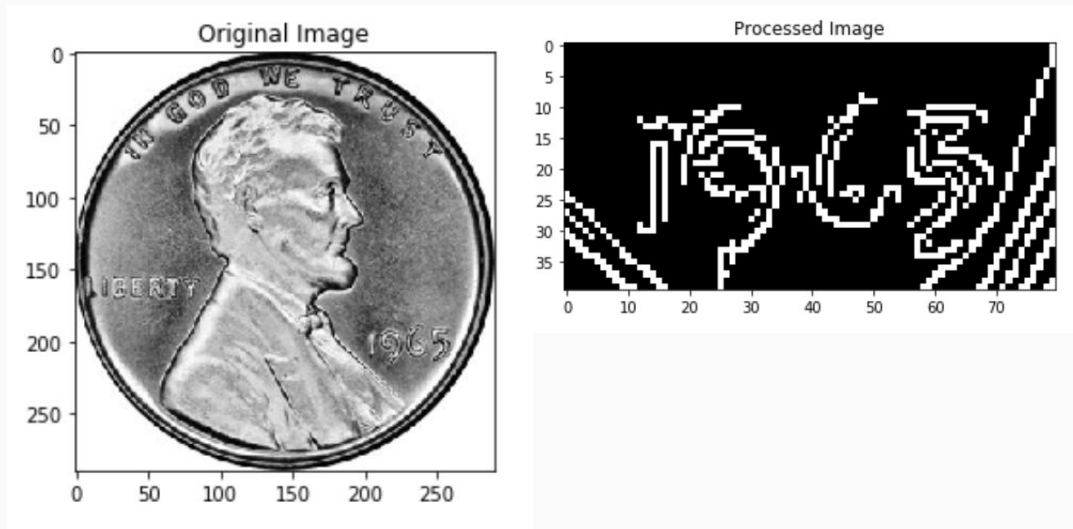
```
    urllib.request.urlretrieve(image_url, destination)
```

```
except:
```

```
    pass
```

Image Preprocessing

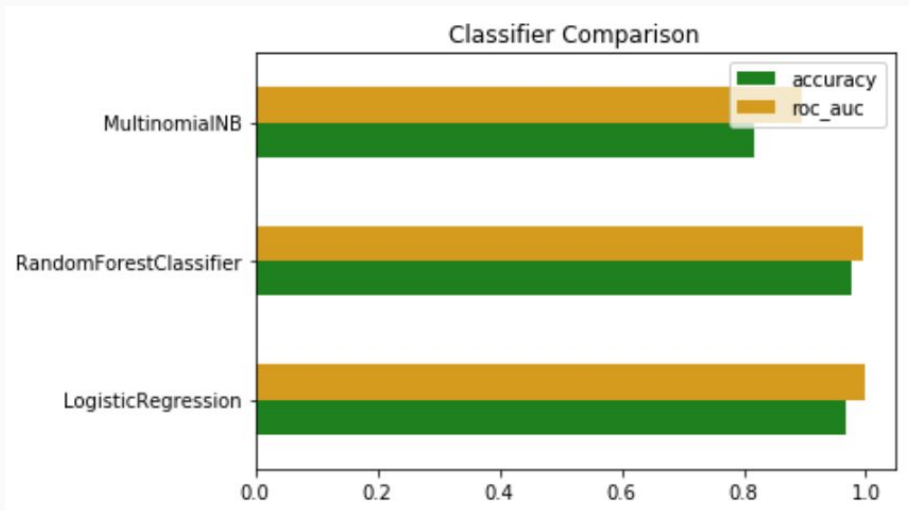
- All operations performed by OpenCV
 - Apply gaussian blur
 - Canny edge detection
 - Slice matrix to include year location
- OpenCV is an open source package that allows complex image processing operations with only a few lines of code
- Size reduced from 50 x 150 to 40 x 80



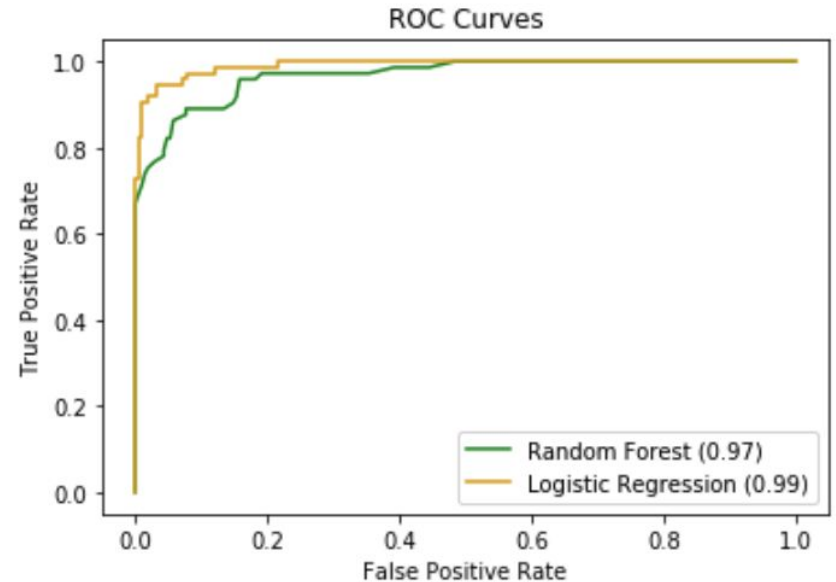
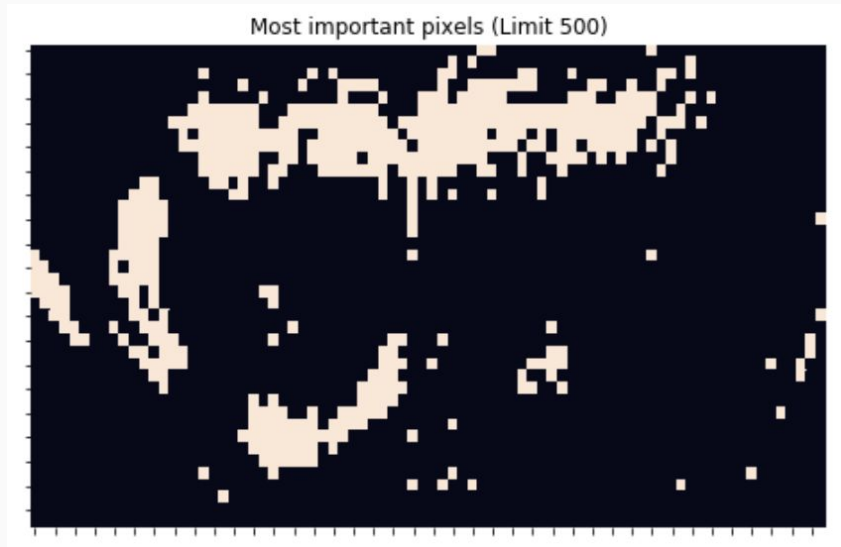
Classifier Comparison - Training Data

- Oversampled minority class to balance our target variable
- Classifiers generally performed well on training set
- Logistic regression showed the best accuracy and area under ROC
- Expected ensemble methods to perform better

name	accuracy	roc_auc
LogisticRegression	0.967071	0.997560
RandomForestClassifier	0.976761	0.996773
MultinomialNB	0.821548	0.898377

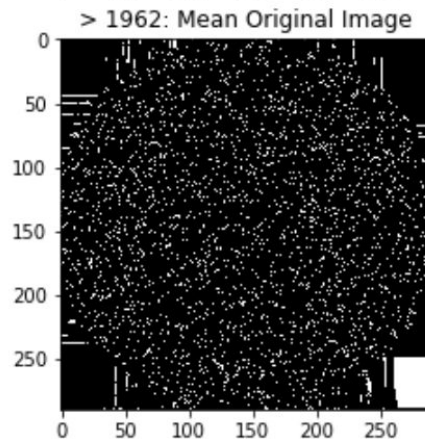
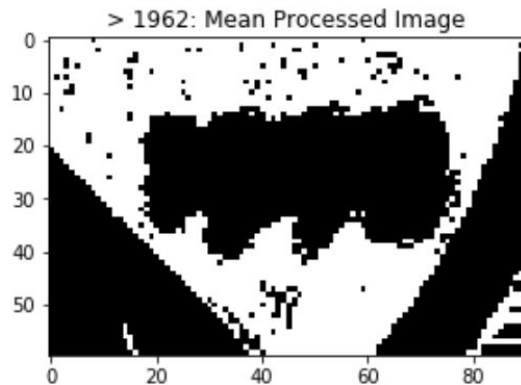
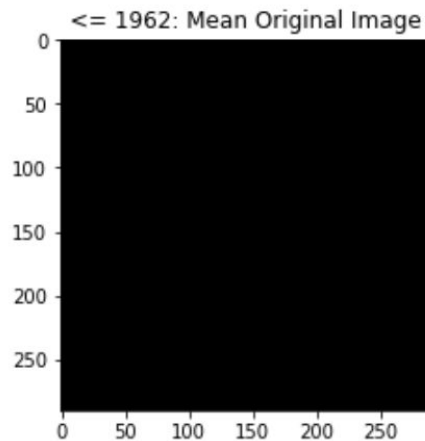
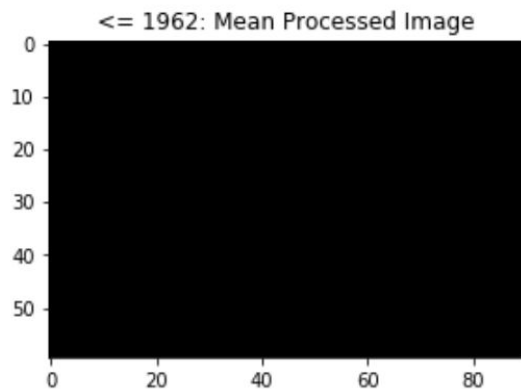


Classifier Comparison - Training Data



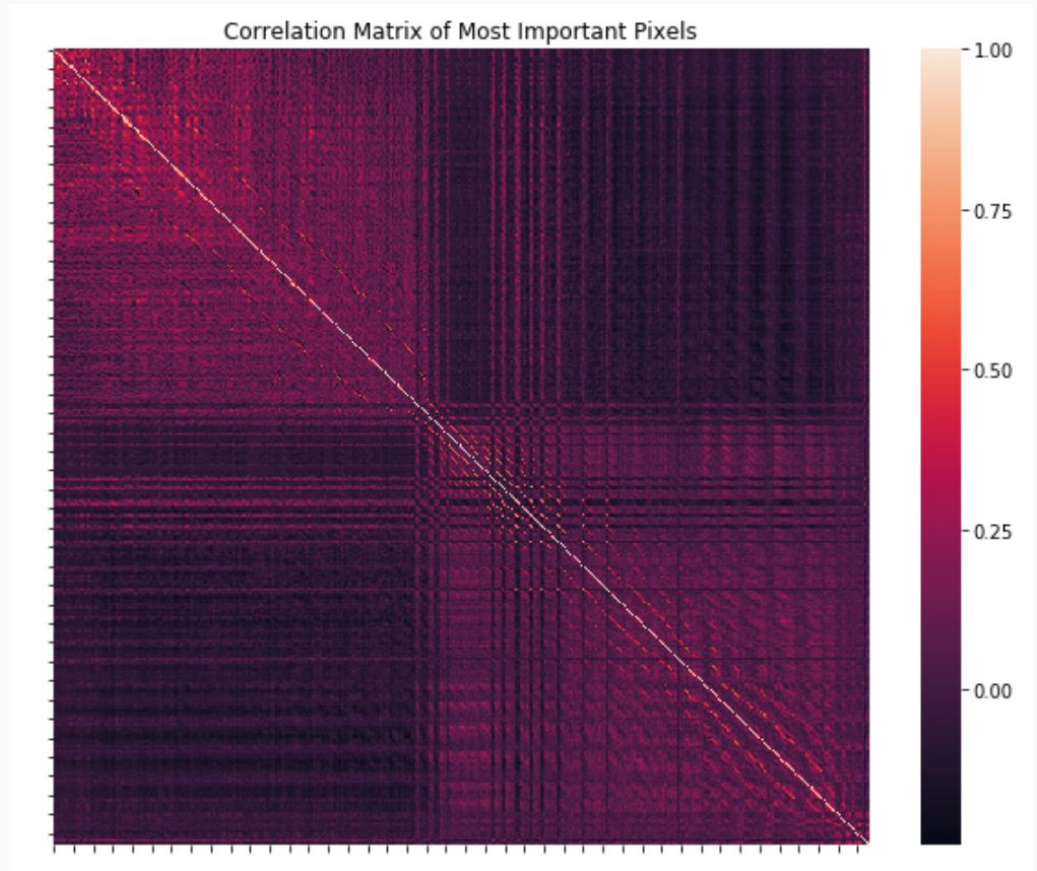
Target Variable

- Coin years after 1962 are brighter than before 1962
- This effect is preserved through preprocessing



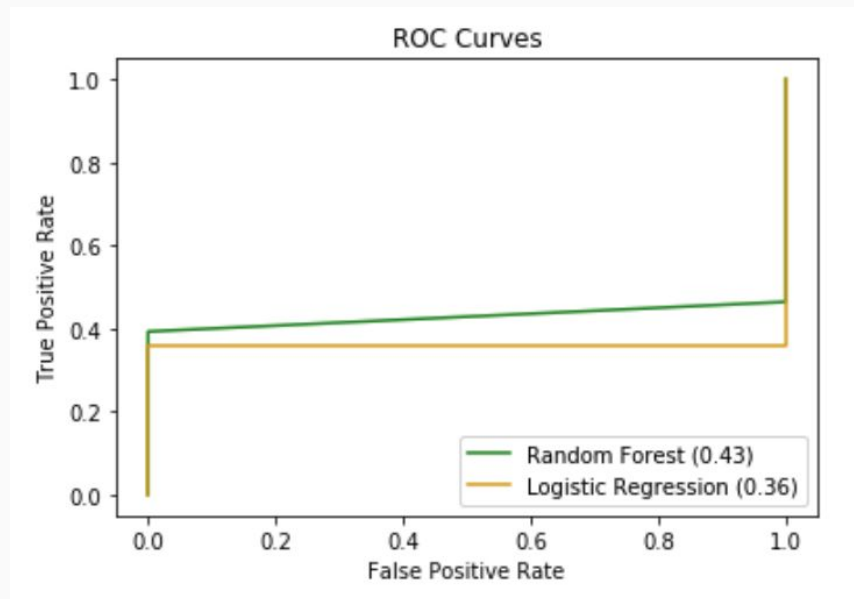
Collinearity

- Looking at the 500 most important pixels, we see some patterns.
- Most of the important pixels are on the border of the year on the coin



Classifier Comparison - Test Data

- Significant drop in performance when applying to test data
- Suspected causes:
 - Small amount of test data
 - Image preprocessing could be more robust



Next Steps

- Look at individual digits
- Better image preprocessing
- Collect more data and attempt to classify all years
- Predict coin condition (good, fair, excellent, etc.)
- Explore Deep Learning algorithms

