US Coin Classifier and Counter

By Brazos Fitch and Yves Ngenzi

Problem Statement

Given an assortment of mostly U.S. coins, can we create a Python script which can recognize the objects and sort the currency into different classes, finally returning a total amount.

=

$15.69

Classification Categories

These are the classes we expect to see:

- Quarter
- Dime
- Nickel
- Penny
- Unknown

The Unknown category is intended to collect all objects that are not a U.S. coin

What features will be used?

The most obvious features:

- Size (diameter)
- Color (shade)
- Pixels (in case of an image)

One classifier could use size and color whilst another compares images

This of course would require new categories for the heads and tails sides of the coins

Design of software

1. Object Recognition - masking
    a. Here we use software such as OpenCV for object recognition, and to crop and copy an image of a single coin
2. Image Processing - segmentation
    a. Resize the image and perform other processes to achieve best possible results. (i.e. rotation, color scale, contrast)
3. Classification
    a. Then use a classifier to classify the coin in the new image

Training Data

Training data can be made from using photos of individual coins found around the house or even exchanged at a bank

Training data (images) can even be found on Google and could undergo image preprocessing steps before being used to train the classifier

1 going to google and download bunch of images like a 1000 images and use these image for deep learning.

Process:

gooling the coin image that we want, and download the images

challenge :

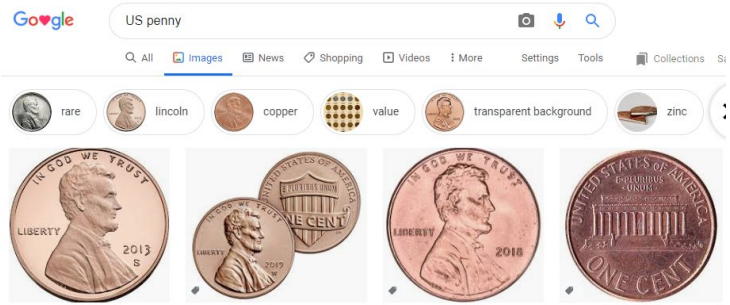
Unsure how we can use these images to create training pixels for other classifiers like Bayes, nearest neighbor, or decision tree.

Training data from coin images

Training data can be made from using photos of individual coins found around the house or even exchanged at a bank

Training data (images) can even be found on Google and could undergo image preprocessing steps before being used to train the classifier

A large dataset can be made with the y dimension being items and the x dimension being features and coin type



Pattern recognition techniques and why.

Deep learning is our number one go if we use google images because it can recognize images and their shapes easily

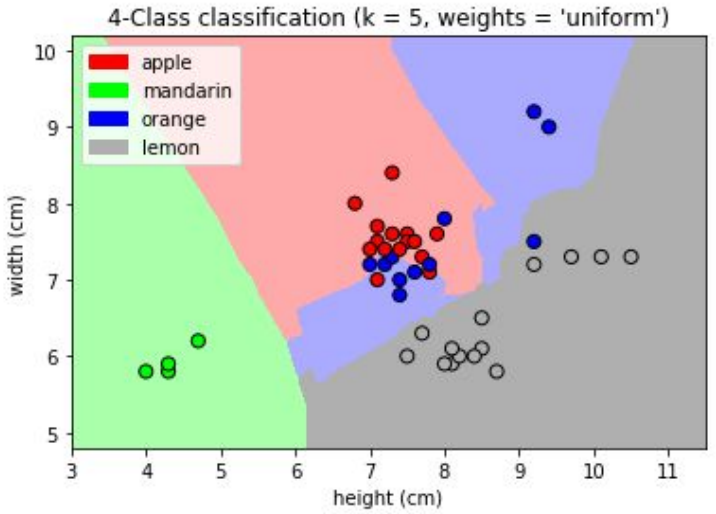
We can also use Bayes Naive classifier which is depend on conditional probability. For example, if a coin size is 17.91mm then classify that as a dime.

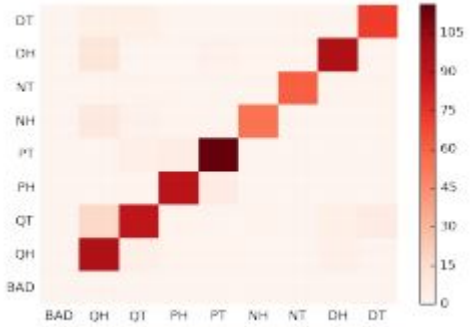
Design choice we have to make for both classifier is adjusting the pixels while we create training data.

Another plan is to find a way to recognize name of coins written on them and then classify those words. However, the words are at different places.

Results: what to know and see

- What percentage of what coin was classified correctly?
- Which classification was most difficult?
- How much money was there?
- What was the decision boundary?
- What was the decision boundary between the two most similar coins?





Schedule/Plan

Week 1

- Create training dataset
- Work with object recognition and OpenCV

Week 2

- Begin classifying
- Optimize and find best technique

Week 3

- Polish the code
- Provide a write-up which displays useful results

Tasks

Brazos - I would like to work on object recognition and image preprocessing.

Yves - I would like to work on finding the online data and train the coins over the already made data and compare with Brazos own data, and if I do not find that I would like to work on creating my own data and share it with Brazos so we can test our coins on it.

References

https://towardsdatascience.com/solving-a-simple-classification-problem-with-python-fruits-lovers-edition-d20ab6b071d2

https://andrewbfang.com/resources/coin.pdf

https://github.com/chen-yumin/euro-coin-classifier

https://github.com/j05t/coin_detector

https://tremaineconsultinggroup.com/opencv-coin-detection-project/