Helping the Blind See with Machine Learning:

Project Update

PHYS 453

Keaton, Christine, Roy
MCC Group

4/24/2020

Problem Statement

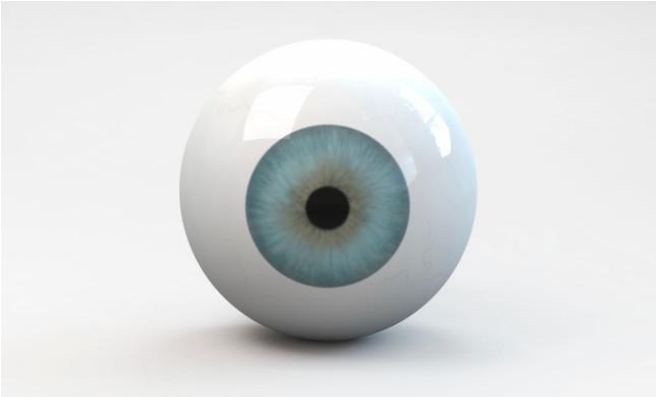
Research Question: Can we use machine learning to optimize image comprehension for the visually impaired?

Prevalence of Blindness

- 285 million people currently with visual impairments [1]
- 39 million blind [1]

Updated Question:

- What color is this?



New Features

Successful classification requires easily accessible features:

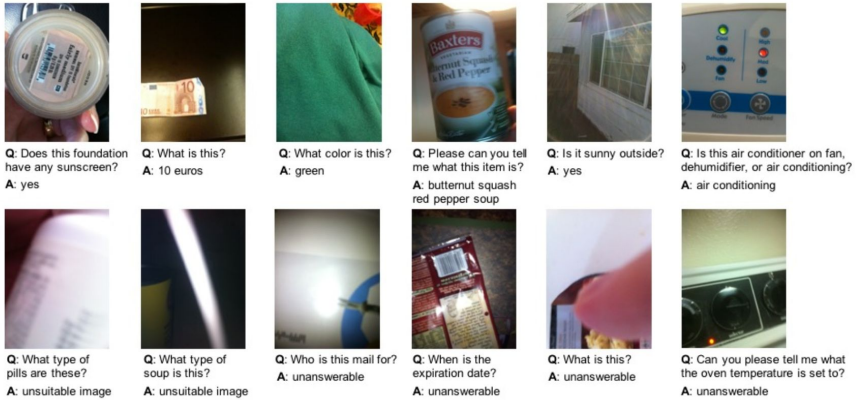
- Color distribution of main image
- Color distribution of subject

We will use these features to determine a normal name for the main colors in the image. For example, a computer can already tell us **#FF5733** or a specific name like **DARKSALMON,** but we are interested in what people actually recognize as a significant color in the image.

Training Data

Training data will come from Viz-Wiz due to large dataset with classified objects with description. However, for the scope of this project we are going to be looking at basic classification.

- Comes as a json file
- 20523 training images
- 205,230 training answers (w/ confidence level)



Basic question that we are looking to answer:

- What color is this?
- We have a parser that only keeps relevant data

Cool demo

Training Data Parser

Details about each visual question comes in the following format (.json file).

- Contains 10 answers to question
- Random pictures


```
"answerable": 0,
"image": "VizWiz_val_00028000.jpg",
"question": "What is this?"
"answer_type": "unanswerable",
"answers": [
    {"answer": "unanswerable", "answer_confidence": "yes"},
    {"answer": "chair", "answer_confidence": "yes"},
    {"answer": "unanswerable", "answer_confidence": "yes"},
    {"answer": "unanswerable", "answer_confidence": "no"},
    {"answer": "unanswerable", "answer_confidence": "yes"},
    {"answer": "text", "answer_confidence": "maybe"},
    {"answer": "unanswerable", "answer_confidence": "yes"},
    {"answer": "bottle", "answer_confidence": "yes"},
    {"answer": "unanswerable", "answer_confidence": "yes"},
    {"answer": "unanswerable", "answer_confidence": "yes"}

    ]
```


Image 17: `VizWiz_train_00000016.jpg`

Visual question: *How much money is on the table?*

Answers:

1. 80 cents
2. 80 cents
3. 1
4. 1 dollar
5. 80 cents
6. 80 cents
7. $1
8. 4 coins on table
9. 65 cents
10. $0.85

Training Data Parser

Three steps to this:

- Parser module (python)
- Parser interface (python)
- Script (shell)

All can be found in github repo

Data is handed in as map/dictionary
Parse through the maps in the .json file
Ask the user to enter color(s)
Check if the question is asking for color

- Looking for “color” or “color?”

If so, investigate the answers

- If there is more than 3 mentions of our color, we save the image name and move on

Transferring to the RPI

Currently in progress,

Execution procedure

- Ask user to enter color
- Have parser (modules) to parse through the .json file to find instances of where that color is mentioned
- Save to a .txt file
- Run color_average python (still to be fully implemented) to find prominent colors in the image
- Save the RGB distribution and percentage of each color as features
- Use the name of the color we searched for as the target value
    - Will consist of the RGB value of the color we are looking for
    - Consist of the name of the color

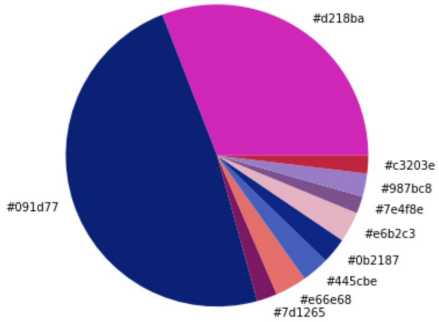
Splitting Image and background

- As this program is not 100% effective, we will keep information on the subject and entire image, so that we avoid losing too much information

Splitting Image and background

Color Distributions

We can break a picture into the pixels with a certain color. Here is a screenshot of a colorwheel showing the most color out of the following picture. Using contours, we are able to determine the top three colors of the overall image and the top three colors of the subject.



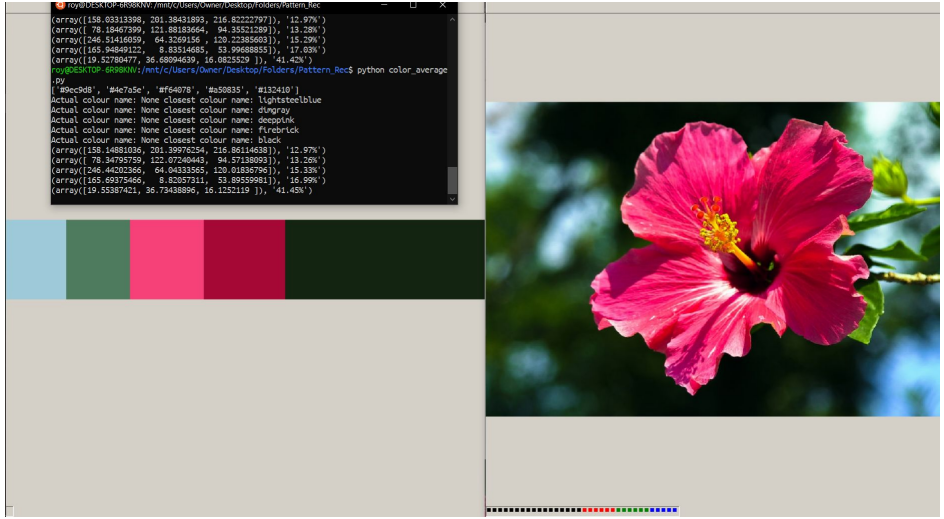
Color Features

Additionally, we are also exploring the ability to find the dist. of color, hex value and name.

Still has to be completely transferred to the RPI

Will implement a modification of both methods seen to optimize the result

Both use KMeans to find the color



['#9ec9d8', '#4e7a5e', '#f64078', '#a50835', '#132410']
Actual colour name: None closest colour name: lightsteelblue
Actual colour name: None closest colour name: dimgray
Actual colour name: None closest colour name: deeppink
Actual colour name: None closest colour name: firebrick
Actual colour name: None closest colour name: black
(array([158.14881036, 201.39976254, 216.86114638]), '12.97%')
(array([78.34795759, 122.07240443, 94.57138093]), '13.26%')
(array([246.44202366, 64.04333565, 120.01836796]), '15.33%')
(array([165.69375466,   8.82057311, 53.89559981]), '16.99%')
(array([19.55387421, 36.73438896, 16.1252119]), '41.45%')

How much data do we need?

We will use all available data for the colors we choose. As mentioned, the parser finds all images which are associated with an answer of color. So, for all the colors we use, we will use the full dataset.

Techniques to be Used

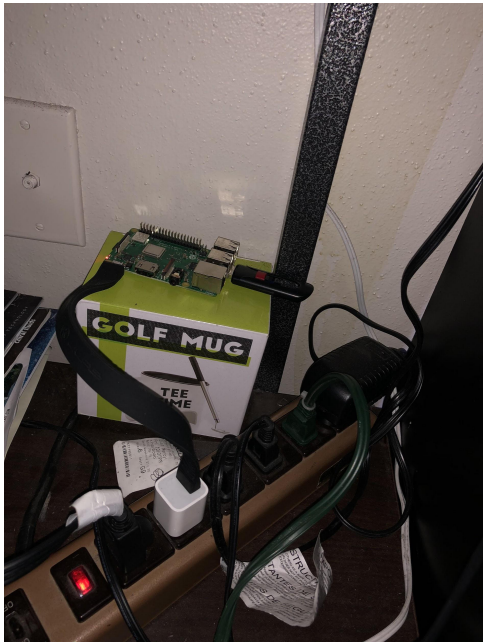
Methods used would have to be able to compute “objects” without burning our computers

- Support Vector Machines
  - Supervised learning
  - Due to the having the ability to control our support vectors
  - The hyperplane can be optimized
- Neural networks
  - Supervised learning
  - Has weights for each neuron
  - Able to modify with different hidden layers, activation function, number of neurons, etc.
- Trees
  - Fast classifications
  - Good with discrete options (name of color, hex color value)
  - Training on large data set should be faster

Design

- Design details will depend on the next steps we decide to take but will follow general outline
  - Use RPI to extract images
    - Have online server to ssh into
  - Will extract data from images
  - The design will include running this on local machines and having a repository to post updates (github?)
  - Will be designed around python and abilities (modules) that python has
  - Will develop further in next couple of days
  - Will definitely have to be considerate of everyone’s schedule

RPI Set Up



- Hooked up under desk with 32 Gb USB
- Sits there patiently

SSH

- The purpose of SSH is to get rid of having one machine with all of the code and data
- Allows us to view progress and each other’s code in scope of our goal
- Getting SSH abilities from ACU network required:
    - VPN Tunneling service due to ACU network security

Server

We currently have a server where we can ssh into a RPI and access our data and codes. Ultimately , we would like to have the server have images placed into it and the processed to return a color (and eventually) size, shape, and brightness. (Running Apache Server)

One can access the web page where the server is hosted only if you are on the ACU network. This is poised to provide a user friendly interface to use the trained classifier.

Analysis of Results

$$\text{accuracy} = \min\left(\frac{\#\text{ humans that provided that answer}}{3}, 1\right)$$

Evaluation metric is the minimum between 1 and the number of people who provided the answer minus 1.

- We will plot the accuracies from different classifiers and different parameter variations
- If possible, it would be cool to have the computer draw a red box around parts of the image it used to describe it. For example, if it used text, it would be a useful visual aid for the program to throw a box around the text for us to see.

Timeline

Friday 4/12/20	Download Dataset using RPI (Roy, Christine, Keaton)✔️ Outline approach using pseudocode (Roy) ✔️ First Presentation ✔️
Friday 4/17/20	Extract features from picture data (Christine, Keaton, Roy) ✔️ Begin breaking down the training set into features (Roy,Christine, Keaton) ✔️ Justify Features (Christine, Keaton, Roy) ✔️ Informal Report 1 ✔️
Friday 4/24/20	Try classifications with different types of classifiers (Keaton, Roy)🕐 Preliminary Error analysis report (Keaton)🕐 Second Presentation
Friday 5/01/20	Analyze results from optimized data and classifier (Keaton) Informal Report 2
Friday 5/05/20	Polish, consider or draft further applications Final Presentation
Friday 5/06/20	Make zillions of dollars ✔️

Summary / Future Work

- All parts of code are currently functional
- Need to work to combine different parts and train our classifier
- Ideally, we will put the trained classifier on our web server, and allow people to put in their own images

Questions?

Sources

[1] World Health Organization, *Global Data On Visual Impairments,* 2010, https://www.who.int/blindness/GLOBALDATAFINALforweb.pdf

[2] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

[3] Jake Vanderplas, *Python Data Science Handbook: Essential Tools for Working with Data*, https://jakevdp.github.io/PythonDataScienceHandbook/

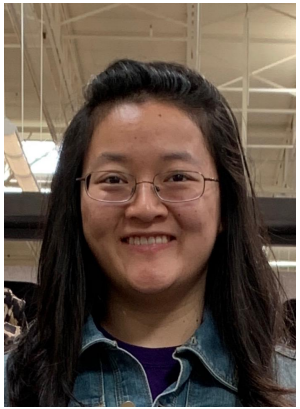
[4] D. Gurari, Y. Zhao, M. Zhang, N. Bhattacharya, *Captioning Images Taken by People Who are Blind,* University of Texas at Austin. arXiv 20 Feb 2020.

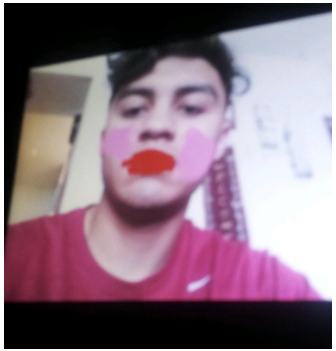
[5] Viz-Wiz, V*isual Question Answering*, https://vizwiz.org/tasks-and-datasets/vqa/

[6] Danna Gurari, Qing Li, Abigale J. Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P. Bigham. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

Roles

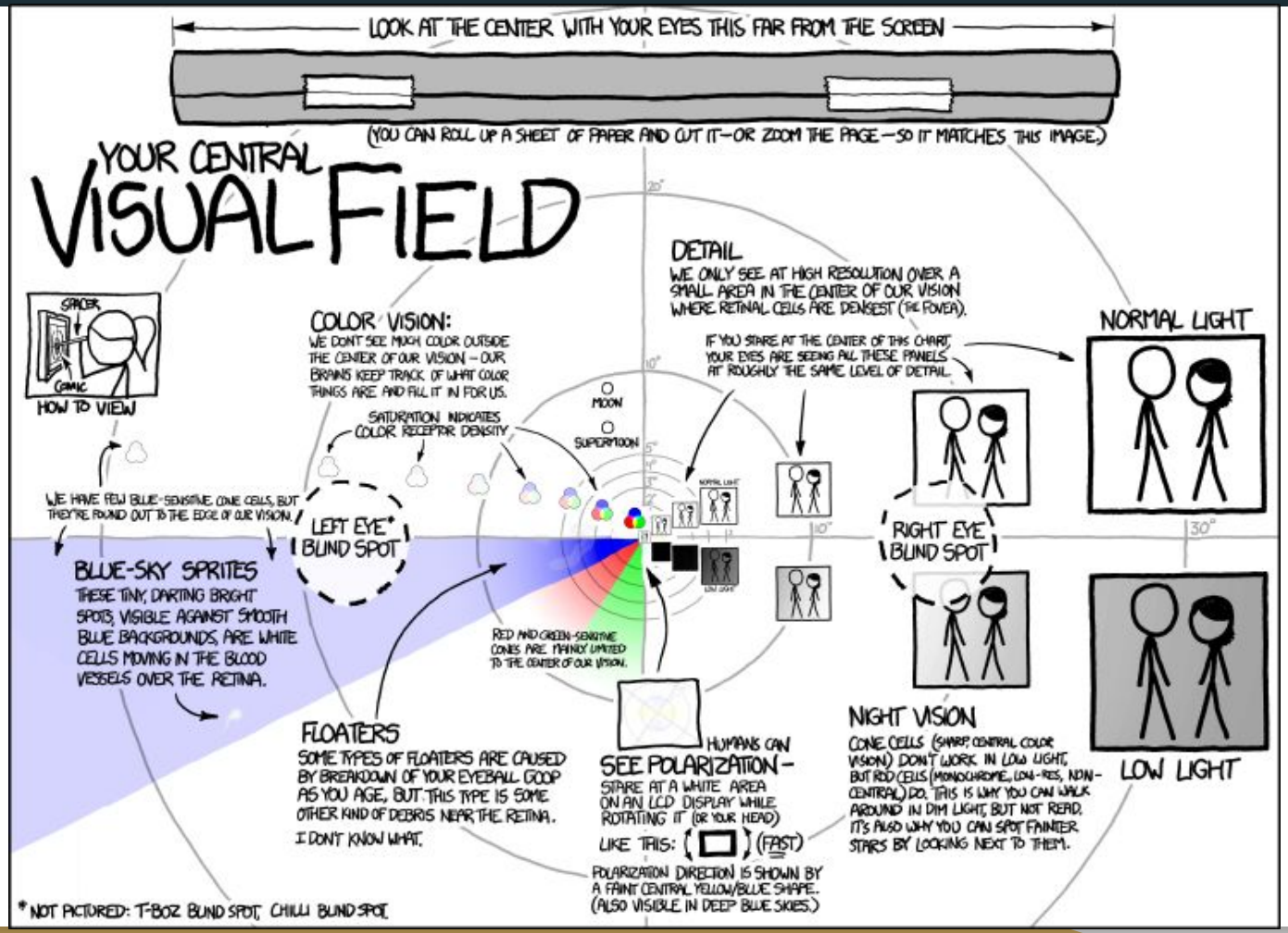
Keaton Brewster	Christine Case	Roy Salinas
Learn about problem	Learn about problem	Learn about problem
Classifier comparison	Breakdown Training Set into Features	Classifier comparison
Error analysis	Feature Justification	Problem approach
Identification of future applications	Feature extraction from raw data	Autonomous data extraction





PC: Youngest sister


I LEARNED TO READ BRAILLE A WHILE BACK, AND I'VE NOTICED THAT THE MESSAGES ON SIGNS DON'T ALWAYS MATCH THE REGULAR TEXT.
THIRD FLOOR OFFICE
S-I-G-H-T-E-D-P-E-O-P-L-E-S-U-C-K ... HEY!


LOOK AT THE CENTER WITH YOUR EYES THIS FAR FROM THE SCREEN
(YOU CAN ROLL UP A SHEET OF PAPER AND CUT IT—OR ZOOM THE PAGE—SO IT MATCHES THIS IMAGE.)
YOUR CENTRAL VISUAL FIELD
SPACER
COMIC
HOW TO VIEW
COLOR VISION:
WE DON'T SEE MUCH COLOR OUTSIDE THE CENTER OF OUR VISION — OUR BRAINS KEEP TRACK OF WHAT COLOR THINGS ARE AND FILL IT IN FOR US.
SATURATION INDICATES COLOR RECEPTOR DENSITY
MOON
SUPERMOON
WE HAVE FEW BLUE-SENSITIVE CONE CELLS, BUT THEY'RE FOUND OUT TO THE EDGE OF OUR VISION.
LEFT EYE* BLIND SPOT
BLUE-SKY SPRITES
THESE TINY, DARTING BRIGHT SPOTS, VISIBLE AGAINST SMOOTH BLUE BACKGROUNDS, ARE WHITE CELLS MOVING IN THE BLOOD VESSELS OVER THE RETINA.
RED AND GREEN-SENSITIVE CONES ARE MAINLY LIMITED TO THE CENTER OF OUR VISION.
FLOATERS
SOME TYPES OF FLOATERS ARE CAUSED BY BREAKDOWN OF YOUR EYEBALL GOOP AS YOU AGE, BUT THIS TYPE IS SOME OTHER KIND OF DEBRIS NEAR THE RETINA.
I DON'T KNOW WHAT.
HUMANS CAN SEE POLARIZATION—
STARE AT A WHITE AREA ON AN LCD DISPLAY WHILE ROTATING IT (OR YOUR HEAD)
LIKE THIS: (FAST)
POLARIZATION DIRECTION IS SHOWN BY A FAINT CENTRAL YELLOW/BLUE SHAPE. (ALSO VISIBLE IN DEEP BLUE SKIES.)
* NOT PICTURED: T-BOZ BLIND SPOT, CHILLI BLIND SPOT
DETAIL
WE ONLY SEE AT HIGH RESOLUTION OVER A SMALL AREA IN THE CENTER OF OUR VISION WHERE RETINAL CELLS ARE DENSEST (THE FOVEA).
IF YOU STARE AT THE CENTER OF THIS CHART, YOUR EYES ARE SEEING ALL THESE PANELS AT ROUGHLY THE SAME LEVEL OF DETAIL.
NORMAL LIGHT
RIGHT EYE* BLIND SPOT
NIGHT VISION
CONE CELLS (SHARP, CENTRAL COLOR VISION) DON'T WORK IN LOW LIGHT, BUT ROD CELLS (MONOCHROME, LOW-RES, NON-CENTRAL) DO. THIS IS WHY YOU CAN WALK AROUND IN DIM LIGHT, BUT NOT READ. IT'S ALSO WHY YOU CAN SPOT FAINTER STARS BY LOOKING NEXT TO THEM.
LOW LIGHT

Classification Categories

Size	Shape	Color	Brightness
Will help us determine how big it is	What shape it is	Give a description of color	This category is not for the blind's own use
Expect large and small sizes (for an introductory level of classifying)	Expect a variety of shapes that will fall under: round, square, rectangular, narrow, etc.	Will utilize "group" pixels to determine color. Use individual pixel RGB channels.	Brightness will be used to aid color classification and/or physical description