

Mike Warner

Dara Behjat

12/17/14

Compo Robo

Final Project Write Up

Machine Learning

What was the goal of your project?

At the foundation of the project we wanted to explore the concept of applying color to object recognition using machine learning. Many object recognition algorithms ignored color as a feature. Color is a difficult feature to effectively compare across images due to the potential for a wide variability in color due to variations in lighting. We focused our project on identifying human faces, where skin tones dominate the frame of the image. A color global descriptor lacks in versatility, but is relatively fast. We used scikit-learn for all of our machine learning algorithms, for its ease of use and power. We saw this as an area for us to do a project where we could incorporate the disciplines we have learned in class and past projects.

How did you solve the problem? (Note: this doesn't have to be super-detailed, you should try to explain what you did at a high-level so that others in the class could reasonably understand what you did).

We wanted a way to use color as a feature in object recognition without sacrificing speed. We found that our use of the SIFT algorithm in the Neato projects generated a noticeable amount of

lag that slowed down performance for recognition. Using color as a global descriptor, we could avoid generating keypoints and calculating SIFT on those points, and saving much computation time. Our approach was to then generate a color dictionary from a database of images, downweight the most common and uninformative colors, and then use a nearest neighbor machine learning algorithm to match similar images. Using color as a global descriptor meant that we had to convert our images to 16 x 16 color bags to make the computation lighter. These images contributed to creating a color dictionary from these large libraries of images. The most recurring color values were down weighted using term frequency - inverse document frequency to differentiate the identifying colors. The images were then clustered according to the k - nearest neighbor algorithm. We then went and scraped our own images from different human centric databases across the web. We used face databases in dynamic environments to see if our algorithm is able to analyze through many pictures and learn to differentiate between humans and “noise” or different objects

We also explored OpponentSIFT to incorporate color as a local descriptor. We used the SIFT algorithm from Paul’s github at first as the base of our project. We decided to incorporate different filters to see if we can find something more efficient. We implemented a color filtering algorithm that looked through images and modified them from images to a bag of colors, which were imported from open cv.

Describe a design decision you had to make when working on your project and what you ultimately did (and why)? These design decisions could be particular choices for how you

implemented some part of an algorithm or perhaps a decision regarding which of two external packages to use in your project.

A design decision we made was to add the TF - IDF weighting as the way to improve the quality of our algorithm. We found that by downweighting the most common colors allowed us to account for background noise and eliminate non-descriptive colors. For example, if many of our images had a white background, TF-IDF allowed us to focus less on this color and on a more descriptive color for our image. TF-IDF is an algorithm used much in word processing that allows downweighting of non-descriptive words that occur frequently, such as “the” and “a.” The same concept can be applied to color features.

How did you structure your code?

Our code was structured in three different files. Because much of the database processing was computationally intensive, we strove to reduce the amount of time repeating tasks that took a lot of time. Our first file generated the color dictionary, which was the most computationally intensive part of our algorithm due to the k-means clustering. The second file generating our cache and calculated the tf-idf portion. The third file trained our k-nearest neighbor algorithm and produced results.

What if any challenges did you face along the way?

There are many different capabilities that scikit has as it is a very powerful Library. Our biggest challenge was filtering through all of its capabilities and making sure we picked the right specific

tools and not getting lost in the capabilities that scikit has. This project is also very algorithm intensive and less task intensive so deciding when we are finished and being realistic with our expectations was difficult. We also ran into some difficulty finding the right databases and image sets for our program to run through, but also incorporating them into our code once we found the right image databases was also a challenge.

What would you do to improve your project if you had more time?

We would have liked to incorporate more filters. The possibilities are endless in terms of how many different filters and algorithms we could have added to make the program more robust. Beside color and SIFT we had done research on spatial pyramid matching which would have added more detail to facial features of the pictures for example. It would have also been interesting to find some real world applications to apply the project to. We had discussed possibilities like search and rescue missions or even security implementations which would have been fun to continue working on. We could have started looking for databases of natural disasters or even started to analyze video with more time to take the project to the next step.

Did you learn any interesting lessons for future robotic programming projects? These could relate to working on robotics projects in teams, working on more open-ended (and longer term) problems, or any other relevant topic.

If you are planning on using different algorithms to incorporate into your own you can't assume that it will work fluidly together. Extensive research is necessary to make sure the algorithm is entirely applicable and also time must be allotted to make it incorporate seamlessly. Also we

have learned especially with very open ended projects it is extremely easy to lose track of tangible work you are doing and the end goal. It is helpful for partners to keep tabs on each other to make sure that the goal is kept in scope and different perspectives are added.