Vishion Technical Planning

Devansh Desai | Shamika Kulkarni | Shweta Patil | Nate Wilson

Current Status:

“Vishion is the only mobile app that uses a product’s true color data, fashion trends and user preferences to help shoppers find the ideal decor.” -Vishion.co

Vishion is still working on their color identification tool although it appears to be able to pick out approximate colors of a scene. To get better representation of the color, Vishion has tasked us with accurately identifying the color variation due to lighting. From what we understand, Vishion currently is using a Java backend that is able to segment the image to determine what object needs the color extracted. Unfortunately we are uncertain of the specifics as we have yet to be connected with Vishion’s CTO so we will be making hypotheses based on the latest information we have. We will be contacting Vishion’s programming department as a first step in clarifying our technical plan.

Goal:

Building on Vishion’s backend, we plan on taking two inputs, the image and the segmented object that needs color matching. As they haven’t tackled color identification of complex patterns, we will only be using single matte color objects as our input. Because we do not currently have access to their object-selection algorithm, we will be creating a bounding-box region of interest on the object and consider it the segmented portion. The output should be the exact color regardless of any deviations caused by light, shadow or bounce illumination. This will be in HSL format.

We currently are planning on working in Python due to our familiarity with the libraries and to avoid any large overhead that Java can create. Ideally though, we should be able to port the trained network into java to allow integration with their current system.

Plan:

Data collection

The majority of work seems to be leaning towards the massive data collection. Given possible variations in lighting, hard shadows and color make up of a scene, we need to provide a wide dataset to train on. We will need to capture images for multiple situations: different temperatures of light sources that would alter the object’s hues, diffuse and hard lights would create a wide variety of falloffs that the algorithm would have to account for, and target objects

will need to share the picture with a variety of color compositions from even color distribution to monochromatic. We would then need to augment the pictures with a region of interest bounding box to replicate the segmentation step. Finally we need the exact color of the object in HSL space. We will use the ColorReader 2.0 that was provided to us by Vishion for this step.

Data storage

The data will be stored as jpgs with sequential numbers as the name and have a matching set of region-of-interest coordinates in a csv file. They HSL color from the color reader will also be stored in a csv file.

Data Manipulation

We plan to do a little bit of data manipulation before we feed it into the network: the pictures need to be converted to HSL and a region-of-interest needs to be cropped out for input. Of course we need to normalize our images as well.

Network Structure

We are most likely using a convolutional neural network as it can closely mirror the representation of color channels.

We will have two inputs made up of the image and the cropped image (RoI). Our idea is that by specifying the pixels values that need to be converted to the precise color as well as provide more data to determine the color shift needed.

Output will be three values of HSL.

Team:

To make best use of this final month, we will need to split the most time consuming portion, the data collection. We plan on distributing the photo captures so we can get 4x the photos in the same time period. We will also be splitting the workload of tagging the region of interest and then finally pooling our data into a single location. As for the neural network, we will attempt to use our free AWS credits to leverage their greater processing power.