

**CSCI - 8980**  
**Project Status Report**  
Hue Chef - A color recommender  
[Use ML driven color suggestions for scenes]

[Team Members]  
Anick Saha <[saha0077@umn.edu](mailto:saha0077@umn.edu)>  
Ojas Bhavani Narayanann <[bhava006@umn.edu](mailto:bhava006@umn.edu)>

★ **The problem we are addressing and solving:**

We are trying to solve the problem of choosing harmonious color combinations using machine learning and a frequentist approach.

Some of the many use cases of this system are -

- Ability to choose the article with matching color while buying personal accessories both online and offline.
- Assisting users with color blindness to effectively make choice of accessory purchase decisions based on the application.
- Enabling users to make smart decisions by recommending them the list of colors that is essential for them based on their tagging of existing wardrobe colors.

*Future work:* The high level approach can be further extended to the Augmented Reality world, by showing the changed scenes at real time to the user.

★ **The current status: what has been done and what remains to be done.**

- The HLD and LLD of the system has been done.
- The core algorithm for the approach has been finalized.
- The training dataset required for the GAN has been identified.

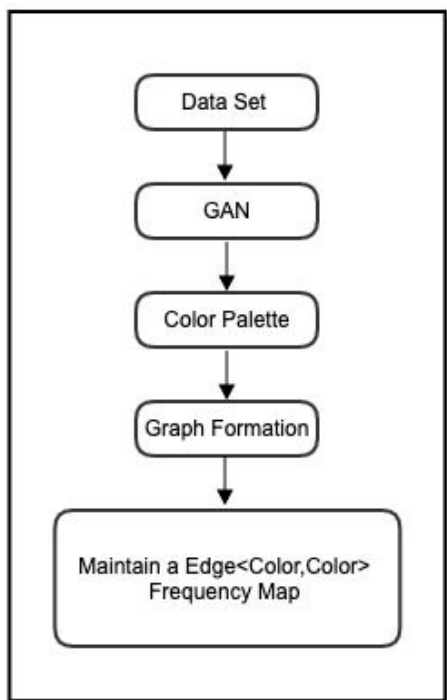
The approach is (at a single node) ~

Each node will maintain it's local Map <ColorEdge to Frequency>.  
Let's call this local node map as Node\_Map;

- 1) We will be extracting good color combination pairs from an image.
- 2) First, the image will be passed through a GAN(Generative adversarial network) block to generate a palette of colors for the image.
- 3) The palette colors are the representational colors of the image.

- 4) The palette colors (RGB) will act as nodes to form a graph out of the image.
- 5) The nodes will have a relaxation on the RGB values they represent.  
For example, a node with RGB (10,10,10) will be representing RGB values in range { [9-11], [9-11], [9-11] }.  
Here  $\delta = 1$ .  
This relaxation factor  $\delta$  needs to be refined.
- 6) Now, how do we form the graph? We will do a pixel by pixel Breadth First Search of the image.
- 7) A pixel at (i,j) will have 8 neighbours.
- 8) If two neighbours C1 and C2 can be mapped to two different Palette RGB nodes (P1, P2), then we increase the frequency of edge <P1,P2> by one. (At a map which is specific to the image)
- 9) Now after the image is traversed and processed, we will traverse the map and if the value is greater than a particular threshold <T> then we will increase the Node\_Map [ ColorEdge ] by one.
- 10) The threshold 'T' needs to be defined.

A simple workflow at a **single node** can be explained by the following image:



- We will be running more numbers of such Node workflows in a Map-Reduce fashion, where this part is the Mapper Job. [ToDo]

- The reducer job will do a union of all the local Maps < Color Edge TO frequency > at its end.
  - Once the reducer job is done, we will be using its output Map to recommend colors based on the frequency.
- 

We are done with the workflow at a single node level. We need to do an exhaustive testing though to see if this approach works well.

*Immediate Plan of Action:* Build an end-to-end flow and run the workflow for 1k images, and check the recommendations.

★ **Any deviation from the original plan:**

It might be the case that we will be doing a color suggestion API using our approach. The further extension to this will be to enhance images by prioritizing objects and changing/enhancing those, and then subsequently move to the next prior object.

★ **Plan for the remaining semester:**

- Target 100000 images to be trained in the GAN to generate palettes and store it in a NoSQL database like MongoDB.
- Identify the service to implement the MapReduce job from AWS, Google Cloud and Azure.
- Implement Distributed Hash Table (Chord Protocol) for faster access to color suggestions.
- Create a basic UI to capture the image and choose a color from the image for the API.
- Compare accuracy against other color recommenders.