Cohesive Poster Generator

Report of Mini project in Image Processing(EC-386)

Submitted By : **Anush** **(201EC108)**

Nakul Nayak B V **(201EC137)**

Project Guide : **Dr.Shyam Lal**

Department of Electronics and Communication Engineering

National Institute of Technology Karnataka, Surathkal

November 2021

Table of Contents

PROBLEM STATEMENT

Generate cohesive posters automatically according to the users needs, by creating variety of designs with different layouts, fonts and colours.

INTRODUCTION

Creativity is not something that everybody has and in today's world posters have become a very integral part of every event, business, organisation, students, etc. , at each point posters and needed. Making a poster is a very time consuming and intuitive-based process which not everyone can do. Poster making is a highly paid profession but not everyone would like to go to a professional for poster making nor spend time on an online poster makers. Free online post a makers have predefined templates which has to be later modified on the users requirements.

In this project we are designing a model / user interface which will generate posters on the given requirements of the user; example images, texts and theme. On the given requirements, it generates a dump of the poster, which is based on different permutation and combinations of layouts, templates, fonts and colour palettes. This automation will save a lot of time and will provide a user with variety of designs.

METHODOLOGY

Principle of Poster Making

In theory there is no such thing as principal of poster making, since poster making is completely an instinctive based process. But, here from the perspective of poster making we needed some kind of fundamental building block upon which we can work and design the model i.e.,

- Layout & Composition
- Typography
- Colours

Layout & Composition

The key to mastering layout and composition is to think like a designer. There are four basic principles that can help you transform your work and sharpen your eye for design.

1. Proximity – It's all about using visual space to show relationships in your content; all you have to do is group related items together, such as blocks of text or graphic elements. Groups that are unrelated to each other should be separated to emphasise their lack of a relationship visually. Overall, whether it's purely text or something more visual, this makes your work easier to understand at a glance.

   White space is negative space, like the spaces between your content, lines, and even the outer margins. There's no "one way" to use white space correctly, but it's good to understand its purpose. White space helps you define and separate different sections; it gives your content room to breath.

2. Alignment – Getting it right when aligning objects by yourself can be tricky, but the most important thing is to be consistent. It's that attention to detail that makes the composition easier to navigate. Without consistent alignment, your work could start to feel disorganized. It might help to imagine your content arranged inside of a grid, with equal-sized margins.
3. Contrast – It can help you catch the reader's eye, create emphasis, or call attention to something important. There are lots of strategies or creating contrast. You can use color, adjust the size, shape, or weight of an object, or use contrasting styles of text. High-level or important items are usually larger, bolder, or more eyecatching in some way. Establishing hierarchy is simple: just decide which elements you want the reader to notice first, then make them stand out. Contrast is closely tied to hierarchy, as it shows the viewer where to begin and where to go next, using different levels of emphasis.
4. Repetition – Every project should have a consistent look and feel. That means finding ways to reinforce your design by repeating or echoing certain elements. For instance, if you have a specific color palette, look for ways to carry it through. It's not just for aesthetic reasons—being consistent can also make your work easier to read.

   With a little attention to detail, you can create beautiful, professionallooking compositions. In many ways, layout and composition are the unsung heroes of design. It's easy to overlook their role, but they're part of everything you do.

Typography

It can also refer to the art of working with text—something you probably do all the time if you create documents or other projects for work, school, or yourself. We started to understand it by common types of fonts, and what is it that makes the difference.

For example, Serif and Sans serif fonts are considered more clean and modern than display fonts. Serif fonts have little strokes called serifs attached to the main part of the letter. Display fonts come in many different styles, like script, blackletter and all-caps. Because of their decorative nature, display fonts are best for small amounts of text.

Some fonts come with a extra baggage, like Comic Sans, Curlz, Papyrus, and many more. When deciding which fonts to use, less is more. If you need more contrast, try repeating one of your fonts in a different size, weight, or style. This trick is practically foolproof for creating interesting combinations.

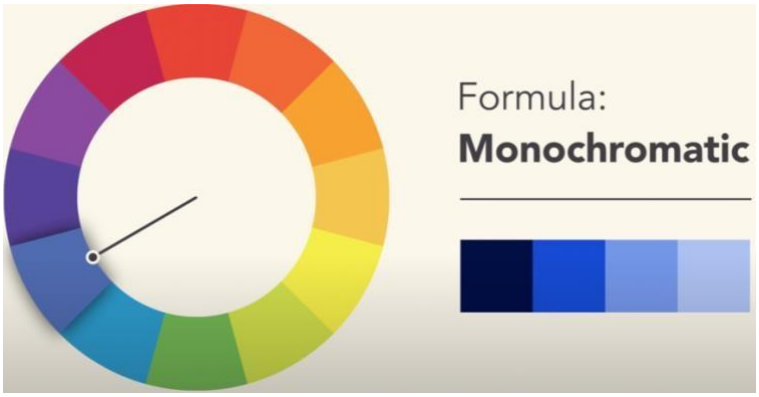
- **OPPOSITES** ATTRACT – combination of font styles that are different but complementary, like sans serif with serif... short with tall... or decorative with simple.

The space between lines of text, also known as line spacing. The goal is to make your text as comfortable to read as possible. Too much or too little spacing can make it unpleasant for the reader. Well-crafted text can mean the difference between an ordinary project... and an extraordinary project.

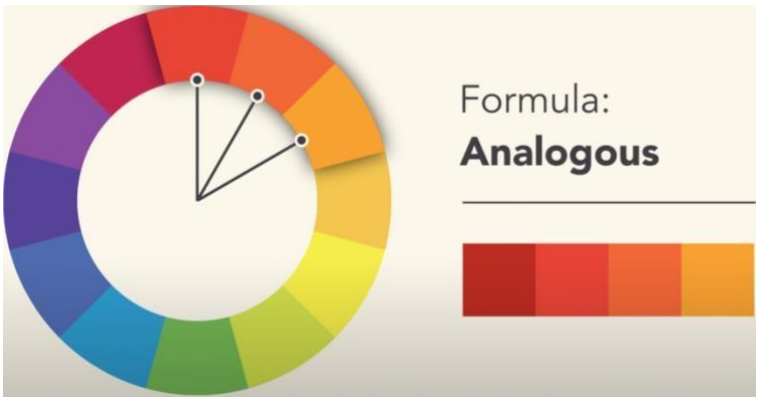
Colors

It plays a vital role in design. It can evoke a certain mood or emotion even communicate something important without using words at all. Artists and designers have follow colour theory, which makes up the colour wheel. Here in this project we take it one step further with hue, light and saturation (HLS colour mode). Hue, is basically just another word for "color". Light has to do with how dark or light the color is, ranging from black to white. Saturation refers to intensity—whether the color appears more subtle or more vibrant. when put all together it creates professional-looking color schemes, formulas based on something called color harmony.

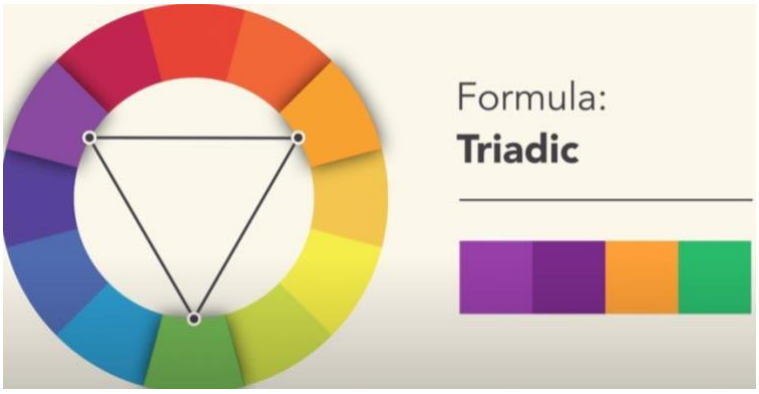
1. Monochromatic Color Scheme - it only uses one color or hue.



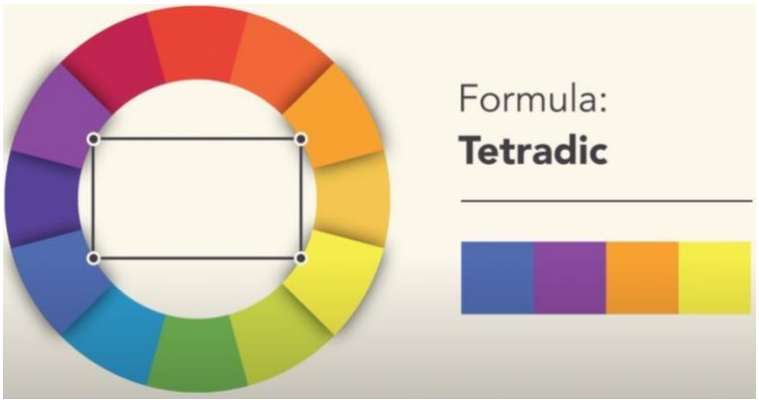
2. Analogous Color Scheme - It uses colors that are next to each other on the wheel.



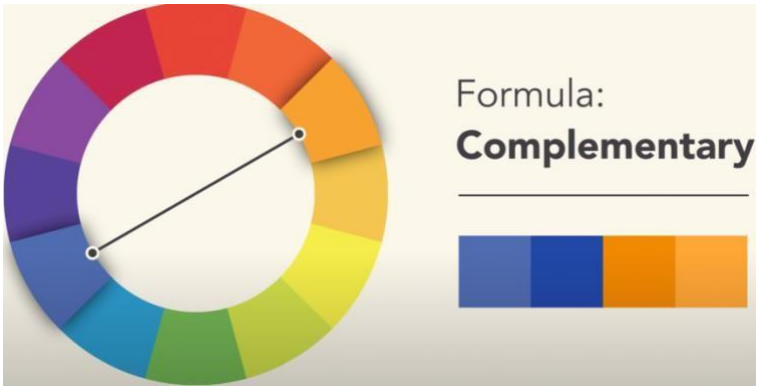
3. Triadic Color Scheme – This uses three colors that are evenly spaced, forming a perfect triangle



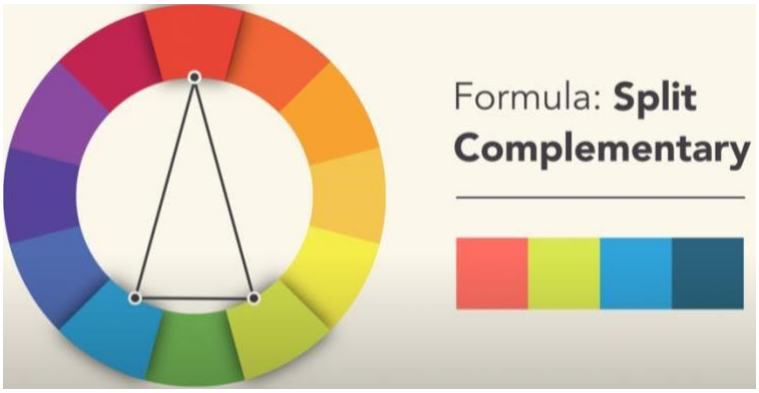
4. Tetradic Color Scheme – This forms a rectangle on the wheel, using not one but two complementary



5. Complementary Color Scheme – These are opposite each other on the wheel



6. Split-Complementary Color Scheme – It uses the colors on either side of the complementary color.



RESULTS AND DISCUSSION

Extraction of colour from Image

For this we have used a few libraries i.e., OpenCV, Numpy and Matplotlib. We have used Kmeans Algorithm, which takes the mean of the clusters of point in the RGB colour mode.

Figure.a – Test image Used

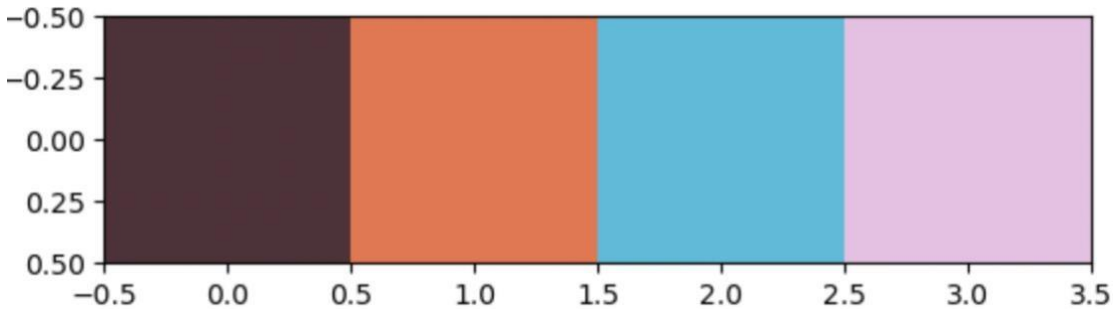


Figure.b – Colours extractes form the test image

To extract different Colour schemes form Figure.b, we first convert the RGB colour mode to HLS colour mode then by varying the angle and value of the axises. Then it is converted back to RGB colour mode and printed/used.

Figure c. – Extracted Colours, Complementary Color Scheme, Analogous Color Scheme, Triadic Color Scheme, Tetradic Color Scheme (Left to right)

Generate image with text and shape

For this we have used OpenCV library. Here we defined function using the commands in OpenCV library to create shape, then control the opacity of shape and then add text on it. Here our intension was to create a dump of images with all the permutation and combinations of placement of shape, its colour, its opacity and the different fonts, it colour and variation in thickness.

But we due to error in the defined function we are currently stuck and creating images with different coloured texts.

Figure d. - Test image with shape and text

Creation of Command-Line Interface

For the command line interface, we used PyInquirer Library, which provides an up/down arrow UI mechanism, for the command line interface. With the Help of the OS library, we listed the current Images in the library and created the option to choose them with absolute paths.

The main menu was created using PyInquirer to switch between different options like image - pre-processors ,poster generator, exit. We used Pillow Library to give the preview of selected images with its details and metadata.

Figure e(1). Command Line Interface - Open page

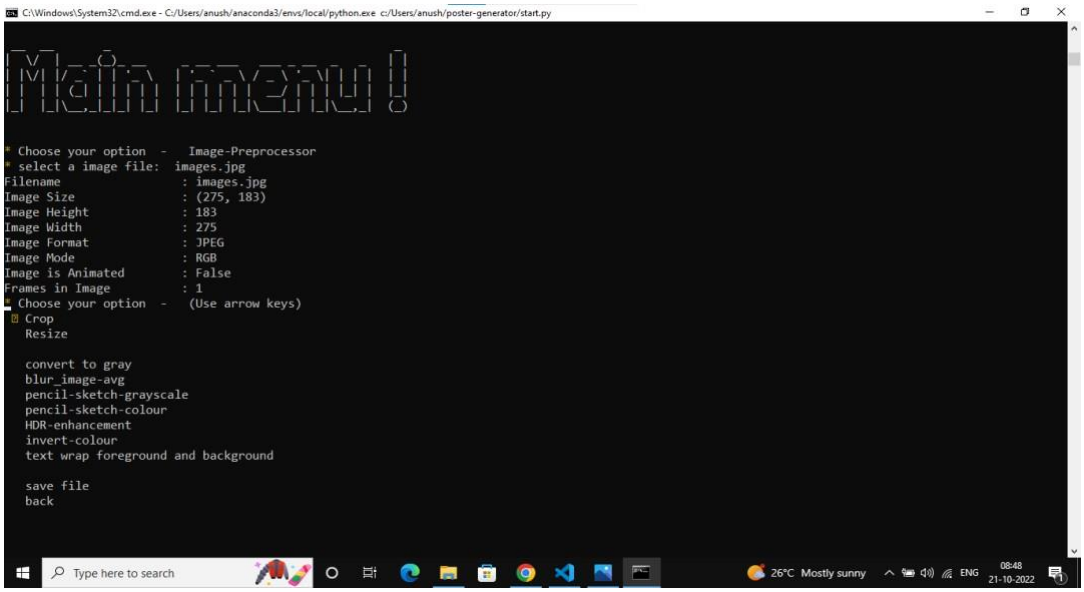


Figure e(2). Command Line Interface – Options page

Image - preprocessor options were implemented using OPEN-CV and numpy using image processing algorithms such as kernel,which we give to the Users in CLI using PyInquirer. In Image pre-processor we have multiple image editing and image effect tools such as crop, resize, grayscale, HDR Enhancement, pencil sketch- grey & colour ,blur, invert colour.

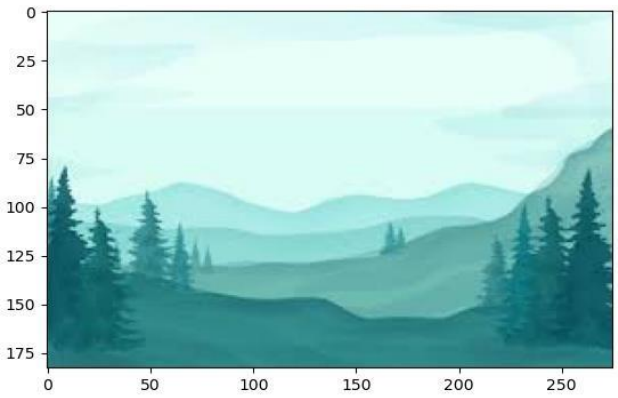


Figure f(1) – Original image

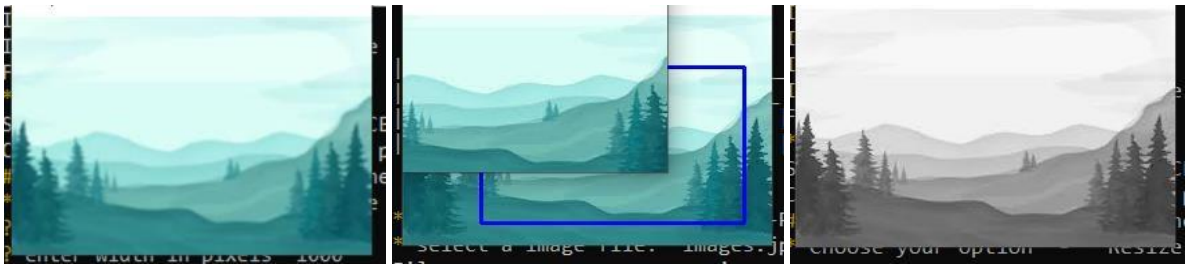


Figure f(2) - Blurred Image, Cropped Image, Grey Image (left to right)

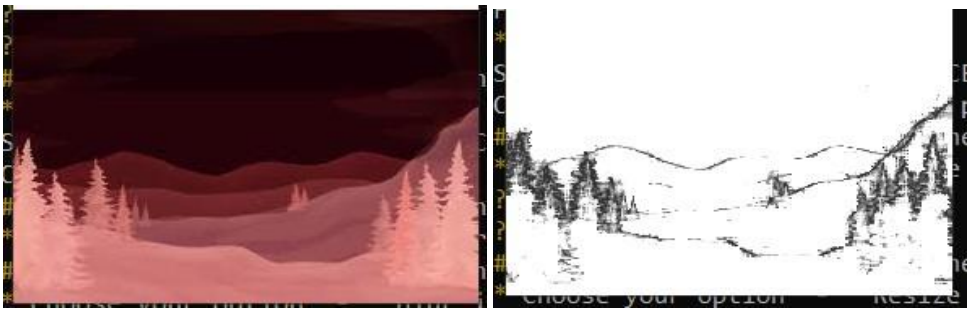


Figure f(3) - Inverted Image, Pencil Image (left to right)

Every Change made was given as preview to Users so that they can keep or discard it and continue. Save file options was given using write function at Every point of preprocessing so that Users can store their latest changes made.

Colour segmentation and Edge Detection

For finding Background colour segmentation We used linear Iterative clustering algorithm using skimage for clustering different segments. From colour segmented background, we got Text wrap masks using Edge detection algorithms such Roberts and Sobel using skimage library.

Color segmentation (combining of pixels in the image plane based on their color similarity and proximity) was one using Linear Iterative Clustering Algorithm which then was used for

Layout detection. Simple Linear Iterative Clustering algorithm performs a local clustering of pixels in 5-D space defined by the L, a, b values of the CIELAB colorspace and x, y coordinates of the pixels. It has a different distance measurement which enables compactness and regularity in the superpixel shapes, and can be used on grayscale images as well as color images.

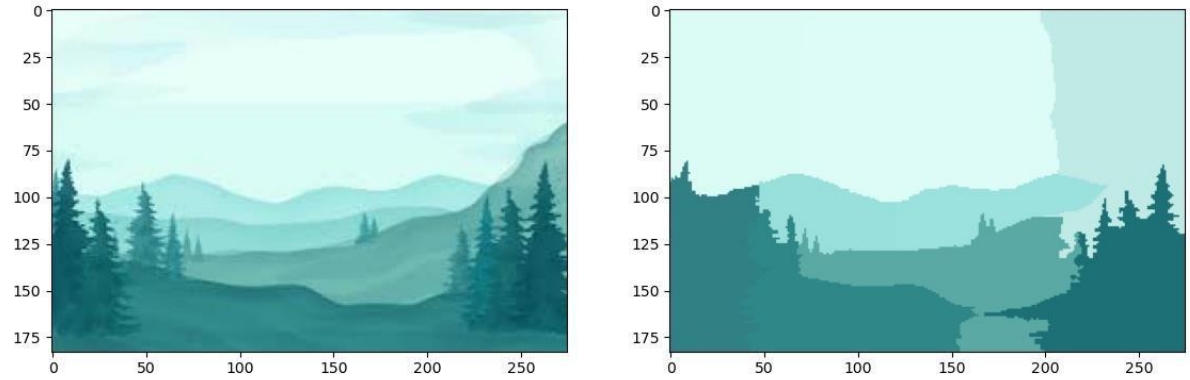


Figure g.(1) - Original Image, Color Segmented Image (left to right)

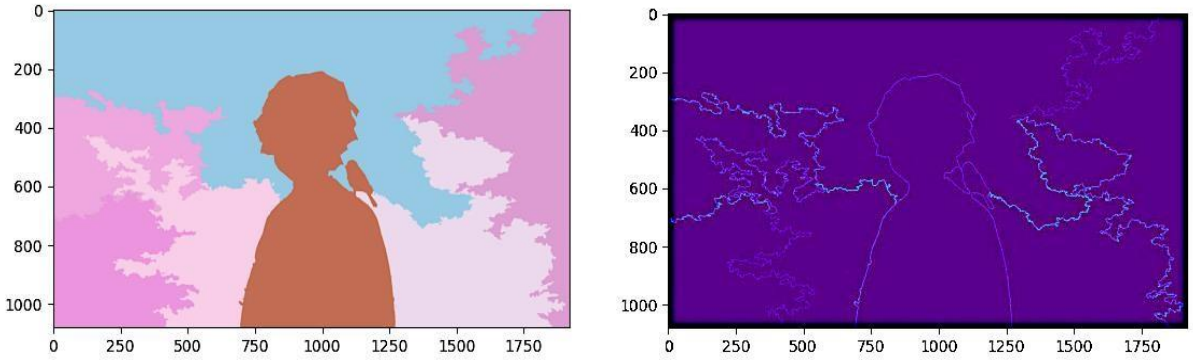


Figure g.(2) - Colour Segmented Image of the test image , Edge mask detection of test image (left to right)

Layout Detection

Numpy and Skimage were used to process labelled data from colour segmentation to obtain layout characteristics such as outer boundary coordinates, average colour, centroid, major axis length, orientation angle, colour frequency, colour percentage, percentage of area covered by colour of layout area, and so on. All information was combined into a Pandas data frame for ease of access, and the Layout data frame can be used to further process images for poster creation.

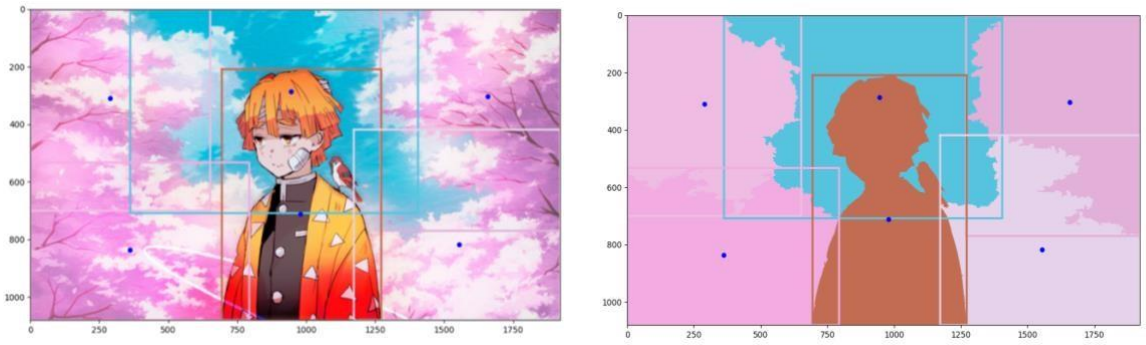


Figure h.(1) - Layout Segmented Image of the original image, Layout Segmented Image on segmented image (left to right)

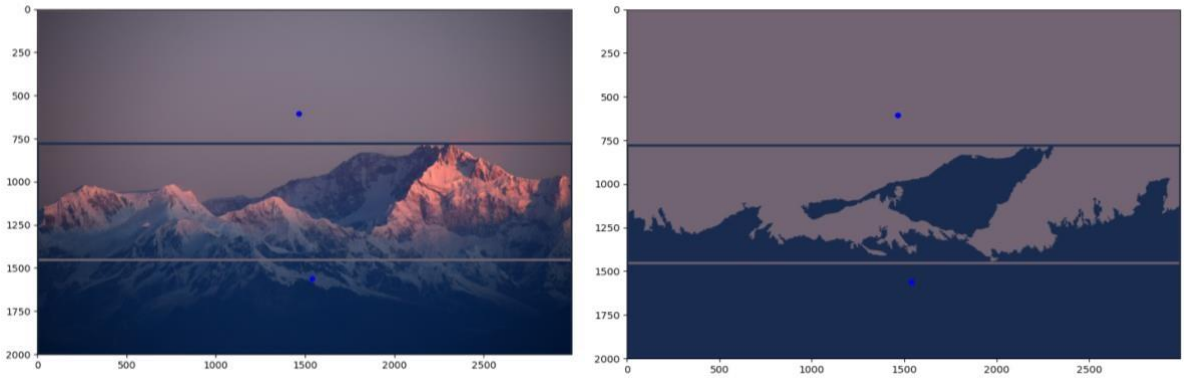


Figure h.(2) - Layout Segmented Image of the original image, Layout Segmented Image on segmented image (left to right)

Layout Implementation

Poster creation with a detected layout data frame and exported custom fonts.

Dual colour segmentation was employed in the preceding example, and layout data was used to determine text position and font position, as well as colour from extracted layout, in a complementary approach in the poster generated.

Figure i. – Text implemented on the segmented layouts.

General Prototype

Here there are 3 parts which has been implemented,

1. Placement of Logo - The user will have option either right side or left side. The given logo will be resized to 12% of dimensions of the poster and its doesn't matter if the given logo is rectangle shape or square.

2. Placement of Text - For the heading, words will be spit into two parts and placed accordingly and we have pre defined the ratio of two parts as 1:2. The text will be placed 45% of the poster y-dimensions form top and 15% of the poster x-dimensions form left, but the text will be placed dynamically based on the size of the text in the centre.

   For the bottom content, the words will be placed 90% of the poster y-dimensions form top and 15% of the poster x-dimensions form left, but the text will be placed dynamically based on the size of the text in the centre.

3. Colouring of Text – The text colour is dynamic ,i.e. it changes according the background color. Here we used the same method we used for color extraction.

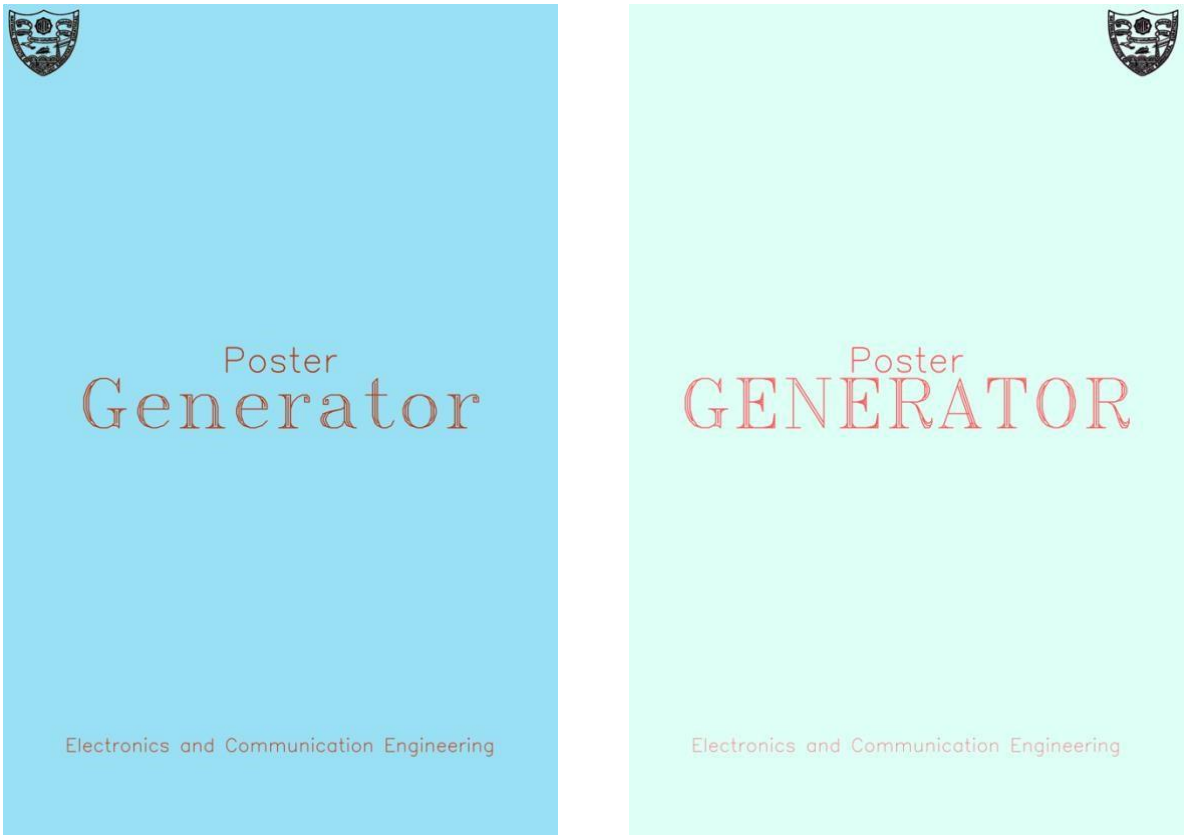


Figure j. – General Prototype for Future integration.

FUTUTRE WORK

- Integrate layout with Prototype.
- Making the model work Flexible between libraries.
- Write more Policies to refine the model.
- Integrate the Model into User Interface.
- Add more image pre-processing tools to smoothen User Experience.

References

[1] “scikit-image: Image processing in Python — scikit-image,” *scikit-image.org*. https://scikit-image.org/

[2] “Image Module — Pillow (PIL Fork) 6.2.1 documentation,” *Readthedocs.io*, 2011. https://pillow.readthedocs.io/en/stable/reference/Image.html

[3] “opencv-python,” *PyPI*, Nov. 21, 2019. https://pypi.org/project/opencv-python/

[4] Google, “Google Fonts,” *Google Fonts*, 2019. https://fonts.google.com/