A NON-PHOTOREALISTIC RENDERING AND PIXELATION OF FACE IMAGES APPLICATION

Lee Min Xuan *Lim Wan Teng* *Tan Jia Qi* *Vickey Tan*

Faculty of Computing and Informatics (FCI), Multimedia University Cyberjaya

ABSTRACT

Given the rising public interest in photo editing applications and social media platforms, we are interested in extending the use of photo editing applications to real life. A non-photorealistic rendering and pixelation of face images application is currently built on PyWebIO framework and hosted locally. This application aims to provide basic photo editing functions, such as enhancement, filters, background changing and photo style conversion which also our application's main features, including cartoonization, oil painting, pencil sketching and watercolor effects, as well as pixelate effects to expand the use of output photos in real life situation.

In order to get valuable opinions on our application, we used Google Forms to conduct an online questionnaire as a social research to help understand user satisfaction and the usability of the application. The responses are collected confidentially and analyzed to find insights. The analysis results are displayed in the form of graphs and tables. In general, this application is usable and produces desirable output, although there is still room for improvement. We hope to enhance this application in the future by adding more features and implementing user suggestions.

Index Terms— Image processing, cartoonization, oil painting, pencil sketch, watercolour painting, pixilation, pixelization

1. INTRODUCTION

Looking at today's society, technology and computer vision are the leading trends. More and more people take and upload photos on various platforms such as Facebook, Instagram, Twitter, share things in life, and express themselves in their social circles. With the advent of smartphones and people's desire for beautiful photos, the invention of photo editing software has sprung up and the photo editing process has become so subtle and fast, anyone can easily edit and beautify photos as needed with just a few clicks on the photo editing application. There are all kinds of photos on the Internet today, and it is worth mentioning that there are a large number of selfies uploaded. However, these applications are still mainly limited to online use. Users usually use these applications to edit photos, and then only keep them for their own enjoyment and posting on social media. Therefore, we have created an application to extend the use of these image processing applications to real life, because this will make this type of application more meaningful and has practical value.

There are several different painting styles such as cartoon, sketch, watercolour and oil paint in the real world. These different painting forms are often used in various fields such as advertising, film, designing and photography, especially cartoons are widely employed in child-related fields. Currently, the production for artistic materials mainly relies on manual implementation. However, the creation of handmade crafts can be very toilsome, involving art, creativity, and innovative skills, which are just pure natural endowment or need to be acquired through hard work. For example, creating high-quality cartoonization of images requires the artist to carefully consider line, colours, texture and shadows, which means that it is a difficult and time-consuming job. The required skills are difficult to master, and it takes a long time to gain experience and self-understanding in artistic concepts to form one's own style in creation. Therefore, the technology of automatically converting real life photos into high quality cartoon style images in computer vision can help professional artists to have a rough idea on how specific image will look like with different drawing styles then they can make decision faster and save time in order to focus on more original creative works, while ordinary people without prior art knowledge can easily make handmade crafts and implement their own cartoonization of image style by referring to computer-generated images.

In recent years, the use of face transformation such as cartoonization and pixelation, has been a trend all over the world and highly associated with people's daily life. Hence, we decided to focus on creating an application that allows face images style transformation and customization in this project. For instance, selfie apps with face cartoonization features are very commonly used to edit and beautify faces as the profiles or posts of their social accounts. For example,

the Snapchat dog filter was a smash hit back in 2016 and then there are more AR filters being created. Moreover, making profile pictures into custom cartoons is a great way to catch attention in social media networks due to it being an easy, interesting, unique way to represent and embody who you are without actually showing the 100% real and original face. In addition, some people actually follow the trend just because of the herd mentality, and don't want to be considered outdated by others. Besides, face cartoonization allows people to have the opportunity to directly view their appearances, understand their possible and expected appearances in the two-dimensional space which are surreal or abstract, thereby satisfying people's imaginations of what they would have looked like in the cartoon world, disney, draw or animation.

In addition, pixelation which is one of the components in image processing that can be utilised in real life situations. Many people may not be able to free hand drawing or do art because they don't know where to start and don't understand the specific steps of drawing. For crafts, even after watching the tutorial, it's actually very difficult to step in, especially when creating new and exclusive patterns. Therefore, pixelation can help solve this problem and assist the beginner to start drawing or making handcrafts. In particular, colouring materials for kids, when the image is divided into grids labelled with different numbers, the kids can easily fill it with colour for the corresponding number. Therefore, it can help kids to raise and instill their interest in art from an early age. In the same way, pixelation can contribute to handcrafting industries such as traditional crafts, bead arts, embroidery and knitting. The reason lies behind it is they refer to pixelated images to form out a complete product step by step, so they can develop these activities as a hobby. Pixelation has the potential to be explored and incorporated to help people convert their favorite photos into easy-to-make products such as designing their own customized T-shirts, mobile phone cases, keychains, etc.

In view of the above factors, we decided to create an image processing application with higher practicality in real life by fusing different painting styles, cartoons and pixelated effects. This application is expected to consist of multiple functions and generate reasonable outputs that are suitable for fundamental usage. Thus, people are not required to spend extra time studying and learning different painting styles if they just want to create something basic. Through the pixelation function, the application is expected to produce pixelated output as the reference when producing or making handcrafts. Specifically, the pixelation function of the application can split the image into a flexible number of grids which can be clearly identified by the naked eye. It allows people to know the colour to be filled for each well-separated grid in order to get their desired output. Meanwhile, it prevents the scale distortion problem that might lead to the failure of a handcraft work due to the bad aesthetics or appearance.

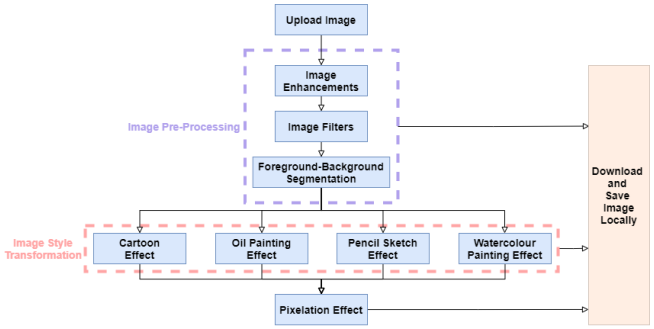

Figure 1: Flowchart of the Pixtono application.

In this project, we are proposing a non-photorealistic rendering and pixelation of face images application that comprises the functionalities such as image enhancement, filter, segmentation, face image transformation with different styles and pixelation. The overall plan of the application is illustrated in Figure 1 and described as follows. The application will be hosted on web browsers and will allow users to upload their own frontal facing images to upload. After the input image is uploaded successfully, the application will prompt the user to enhance the image. The users will be able to choose to rotate the image, equalize the image, smooth the image, and adjust the brightness and contrast of the image.

After enhancing the image, the user can choose one image filter to apply, the filter methods include sepia, lighting and Clarendon filters. Subsequently, the user can choose to remove the background of the image. The purpose of image background removal is to achieve a better output that can apply on beads art or other real world art works application. At this point, the image is pre-processed and ready to apply our main style transform features on the image. Users can choose to not perform pre-processing on the input image based on their own preferences. The main features to transform the image style are various painting effects such as cartoon, pencil sketch, watercolour and oil painting. Each feature has different style and parameters to let the user choose on their preferences. The cartoonization feature included three different types of style which are classic, comics and twilight cartoon style. For pencil sketch, the user is able to choose different colours to sketch and adjust the kernel size for the thickness of the lines. In watercolour effect, the adjustable parameter is sigma value, colour tone and segment scale. If the user chooses oil painting to apply, they are able to adjust parameters of brush size, brush colour, and style. Users can preview the output for each of the features and choose the desired effects to apply. After style transformation, the user will be prompted to proceed pixelation or no changes on the filtered image and the user can adjust the pixel size and pixel colours. For each stage of

the image application, the user is able to download the image as the output.

The real life usage of this application is to produce pixelated face images that can be used for making handmade crafts and goods. For instance, perler beads, toy brick puzzles, embroideries, knitted products, diamond paintings, and etc.

2. BACKGROUND STUDY

2.1. Image Enhancements

Image enhancement techniques have been widely used in many applications of image processing, where the subjective quality of the image is very important for human interpretation. It refers to the process of highlighting certain information of an image and weakening or removing any unnecessary information according to specific needs. For instance, removing noise, revealing fuzzy details and adjusting the level of contrast to highlight the characteristics of the image. Contract is an important factor in any subjective assessment of image quality and is caused by the difference in brightness reflected by two adjacent surfaces. Therefore, the histogram equalization in image processing aims to improve the distribution of intensity values in the image by equalizing or flattening the histogram as much as possible [1]. The difference between adaptive histogram equalization and ordinary histogram equalization is that the adaptive method calculates multiple histograms, each of which corresponds to a different part of the image, and uses them to redistribute the brightness value of the image. Another type of histogram equalization is the Contrast Limited Adaptive Histogram Equalization (CLAHE), which is within the contrast limit. In the case of CLAHE, the contrast limit procedure is applied to each neighbourhood, from which the transfer function is derived. CLAHE was developed to prevent excessive noise amplification that may be caused by adaptive histogram equalization. As shown in Figure 2, the image enhancement for histogram equalization is used for adjusting brightness and contrast.

Figure 2: Example of Image Enhancement using *Histogram Equalization* techniques

In addition, noise is a random change in image intensity, visible as part of particles in the image. In other words, noise means that the pixels in the image display different intensity values, rather than the actual pixel values obtained from the image. Hence, the noise removal algorithm is the process of reducing or removing the visibility of noise by smoothing the entire image [2]. The sample image of the denoising method as shown in Figure 3.

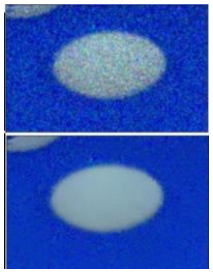

Figure 3: Example of Image Denoising

Furthermore, the captured image will be affected by the position of the camera relative to the scene, so the perspective of the scene geometry will be changed. Therefore, it is necessary to apply affine transformation by converting the measured values from ideal coordinates to actual coordinates to uniformly distort the image into the correct perspective distortion. Affine transformation is an important linear two-dimensional geometric transformation that maps variables to new variables by applying linear combinations of rotation, scaling, and/or shearing operations. All common affine transformation techniques are shown in Figure 4.

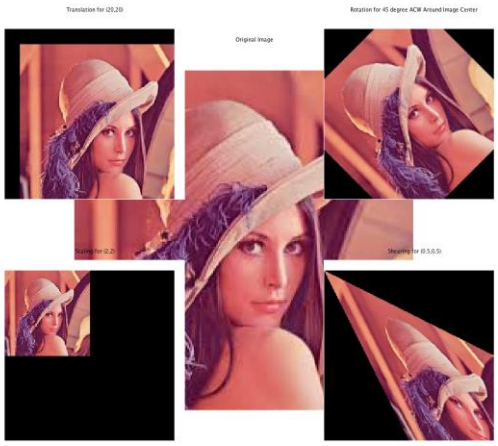

Figure 4: Example of Affine Transformation Techniques

Remini (https://remini.ai) is one of the existing applications that can enhance photos in real time, aiming to do this-all

photos and videos uploaded to the application will be processed for high quality results. In addition, Enhance It (https://www.reaimagineapps.com/) is an application that uses neural networks to enhance photos. It can restore old low-quality pictures, remove noise, enlarge low-resolution, and illuminate dark photos, enhance blurred images and make photos more colourful. For blurry or unclear images, these apps allow users to enhance their photos by adjusting brightness and contrast, removing noise, and rotating photos. The application we propose has the function of enhancing the image through image equalization, image denoising, contrast and brightness adjustment, and image rotation on the input images.

2.2. Image Filters

The image filter has many ways and different algorithms and arithmetic operations can be used to output a filtered image. By changing the pixel intensities to different values with several types of mathematical functions, the image might come out in different ways and are suitable to use as image filters. The fact that we usually did not realize is image arithmetic is used in many computer vision applications and image processing projects. In this project, the image filter is more towards instagram-like filters. Since many image processing techniques are ineffective in noisy environments, the image filter is used as a preprocessing module in our case. Cropped, resized, and altered into grey images are images taken using a digital camera for various experiments. To reduce noise and enhance the image for restoration, preprocessing filtering methods are often applied. The common image filters in image processing algorithms are averaging, median, wiener and non-local means (NLM) etc [3]. The filtering in our project is more towards increasing image information content which is a different concept compared to the filtering methods mentioned above. Our image filtering algorithm is inspired from the textured filters often appearing in image editing online tools or mobile applications like LunaPic, Tuxpi or PixEasy.

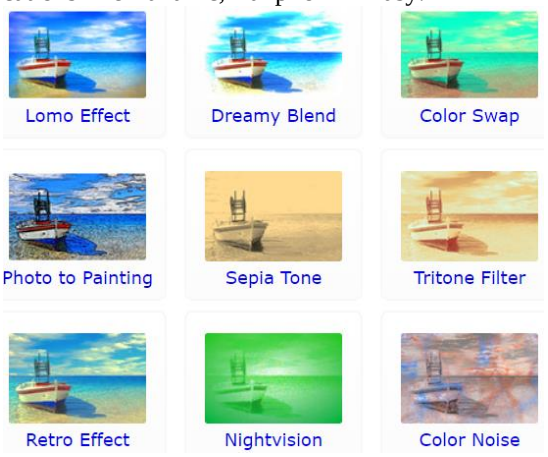

Figure 5: Example of filter in online image editing application Tuxpi [4]

2.3. Foreground-Background Segmentation

Foreground-background segmentation is a common task widely used in various situations, where the goal is to split the image into foreground and background. Usually, foreground is the main focus of an image where background is less important. Implementation of foreground-background segmentation in real life are such as the background blurring, focusing, depth of field functions of camera, asynchronous and synchronous background changing in images and videos, changes detection, etc. Most of the web image processing applications utilise foreground-background segmentation techniques to remove and change image background.

The foreground-background segmentation techniques are available in both traditional and modern ways. Traditionally, there are thresholding, contour detection, background subtraction, GrabCut algorithms with different background modelling and background maintenance techniques, where each algorithm or model is suitable for different situations [5]. These can be done in most of the programming languages with OpenCV library and other common libraries for array and matrix. Modernly, in order to obtain better results while considering various situations that may cause discrepancies in traditional foreground-background segmentation, people started to develop deep learning and instance segmentation models. For example, Mask R-CNN, U-Net. FgSegNet, DeepLab v3, etc. These algorithms generally produce better results than traditional algorithms and can fit for various situations depending on the training data given. The implementation of these models is usually fully automated from the user perspective but requires more computing resources, time and professional knowledge from the developers.

For instance, Clipping Magic (https://clippingmagic.com), cutout.pro (https://www.cutout.pro), PhotoScissors (https://photoscissors.com), removal.ai (https://removal.ai), and remove.bg (https://www.remove.bg) are some of existing applications that detects the background of the image uploaded by the user, thereby changing the background to transparent. Some of the applications provided extra functions of changing the background to a different solid colour, or sample background provided, or another image uploaded by the user, adding text and shadows on the image. With the help of semi-interactive settings, some applications allow users to mark on the processing image to segment the background more precisely by marking the pixels as “foreground” or “background” and the rest of the pixels will be defined based on the algorithm used. Our proposed application has the functionality to change image background to transparent, specific solid colours, patterned background, and another image uploaded by user, but not capable to indicate pixels to be classified or add things on

image. Figure 6 presented the screenshots of the background changing applications mentioned.

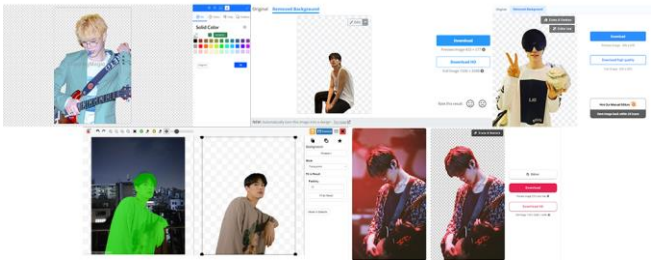

Figure 6: Screenshots of different background changing applications.

2.4. Cartoon Effect

A cartoon effect can transform human faces into more cartoon character-like faces. In brief, it is a cartoon face generator that allows people to see how their faces will appear in the cartoon world. The idea is inspired by Justin Pickney, he has published a face transformation app using deep learning techniques called Toonify (https://toonify.photos/), and it is a website that can instantly transform the image you uploaded into cartoon style. People can see what they look like in Computer-Generated Imagery (CGI) movies using Toonify. The architecture proposed by the author is using a deep learning network called StyleGAN2. StyleGAN2 is a style-based generator architecture for generative adversarial networks that uses StyleGAN as a backbone with additional improvements of the network architecture. In [6], Doron Adler fine-tuned the StyleGAN2 proposed by Justin Pinkney with Disney Pixar character faces and then integrated the tuned layers with the normal real face model (FFHQ) to perform Image-to-Image translation. As a result, the model can generate a disneyfied face image based on the real face image that was input in the beginning. However, considering our application cannot carry a heavy deep learning model. Another approach that is closer to cartoonization using image processing, the overall framework is to start from image processing then apply bilateral filter and luminance quantization. Additionally, the final output that is converted back to the RGB channel is added with an image with only edges detected, this will probably strengthen the effect of the cartoon [7]. In this project, the team plans to create an app that has a feature to transform human faces into cartoon faces using image processing techniques. Cartoonization feature has been popular in a lot of image editing apps, below shows the cartoon picture from different image editing apps which are PhotoCartoon.net and LunaPic.

Figure 7. Cartoon effect of original image (left) on Photocartoon.net (middle) [8] and LunaPic (right) [9]

2.5. Oil Painting Effect

Oil painting has been one of the most famous artistic creation techniques since ancient times. This is because oil paints are elastic and have different colour depths. Oil painting is a medium composed of pigments suspended in dry oil. The excellent facilities for realizing tonal or colour fusion make it unique among fluid painting media; at the same time, it is easy to obtain satisfactory linear processing and crisp effects. In addition, it can be applied in many different ways, from thin glazes diluted with turpentine to thick coatings. Due to the drying time and the materials used, the production cost of oil painting artwork will be very high. Consequently, the demand for oil painting effect algorithms is getting higher and higher.

Nevertheless, there is still some research on oil painting effect algorithms to explore more possibilities and improve artistic skills in computer vision. In the research of Luft and Deussen (2016) [10], an artistic rendering of oil painting method was proposed to automatically convert images into oil painting style. The proposed Kuwabara filter introduces saliency control, which controls the size of brush strokes, produces rich stroke scales, describes objects from different levels of detail, and is closer to the artist's painting skills.

PhotoFunia (https://photofunia.com/) is one of the existing free web applications, which can convert images into oil painting effect images as shown in Figure 8. However, the oil painting effect of PhotoFunia is not obvious and cannot be adjusted. Therefore, the team plans to create an application that uses image processing technology to convert a human face into an oil painting effect image with adjustable parameters.

Figure 8: Example of Oil Painting using *PhotoFunia*

2.6. Pencil Sketch Effect

Pencil sketch has been a famous form of art creation since a long time ago. It is because pencils are the most basic tool that can be used to express the abstract understanding of a natural scene by drawing on a paper. The light and portable properties of pencils allow people to draw at anytime and anywhere. Pencil sketch is very essential as it acts as the fundamental for all kinds of art forms such as oil painting. However, it is challenging to create a good quality pencil sketch as it requires one to have professional skills like good contouring ability and fabulous shading drawing skills. Some examples of sketching techniques are stippling, scribbling, circling, smooth shading and blending as well as hatching and cross-hatching. It takes a lot of time and effort for people to master those important techniques and then apply them to produce pencil sketches. Consequently, the demand for pencil sketch rendering algorithms is getting higher day by day.

The current existing pencil sketch algorithms are based on non-photorealistic rendering technology. The methods can be divided into 3D model-based rendering and 2D image-based rendering. A 3D model that can perform contour detection and imitate human contour drawing is presented [11]. The oriented textures are mapped onto the surface of objects to denote the shading. However, 3D models can only provide satisfactory results when the structure and lighting conditions are available. With the rapid development of digital camera technology, 2D images are getting more attention than 3D models as it is easy to obtain high quality 2D images rather than constructing a complex 3D model. The 2D image based pencil rendering focuses on two main concepts which are pencil strokes generation and pencil tonal texture drawing. A study suggested a sketch filter that uses a maximum filter to do feature extraction of hand-drawn style edge and texture [12]. The sketch filter can calculate the local area complexity and then output an image that illustrates the edge and texture feature that is exactly the same as a pencil sketch drawn by artists using pencils. Besides, another study merged pencil tone drawing and line drawing with pencil strokes to produce pencil sketches [13]. The former is applied to display the shape, shading, shadow and tone whereas the latter is associated with convolution framework to focus more on the scene structure. In addition, open source OpenCV libraries have provided a pencil sketch function under non-photorealistic rendering. It can directly be used for transforming the image into black and white pencil sketch.

Sketchmypic (http://sketchmypic.com/) is one of existing free web applications that can only turn images into black and white pencil drawings. Besides, PhotoFunia (https://photofunia.com/) is a more advanced web application that provides both black & white and colour pencil sketches based on the colour of original images. It enables the user to choose the types of paper for the sketches to draw on such as texture paper, colour paper and white paper as well as fading the edges. LunaPic (https://www12.lunapic.com/editor/) also provides pencil sketch effects with colour variation. They also have some different styles of pencil sketching such as texture, scribble, escher, pen and ink. The user can adjust the level of effect to be applied from 0%-100%. The pencil sketch effect provided in our proposed application allows the user to control the thickness of pencil edges. Additionally, the colour feature is provided for the user to produce colour sketches. But the colour sketch is slightly different from the aforementioned applications as the user is only allowed to choose a single colour of pencil edges to produce a single colour pencil sketch. It is not based on the original image colour like what PhotoFunia and LunaPic did. Figure 9 displays the output of pencil sketch applications mentioned.

Figure 9: Outputs of different sketch applications.

2.7. Watercolour Painting Effect

Watercolour painting is a form of art that is based on two main materials: water and colour pigments. Watercolour painting gives a sense of unique charm with its graceful and flexible visual effects. Applying watercolour repeatedly to increase the saturation of painting is not applicable instead it will cause damages to the artwork. Due to its extraordinary

properties, the artists need a high level of professionalism in creating watercolour paintings. They must control the amount of water and the use of painting media or tools. Knowledge on colour mixing is also indispensable for creation of watercolour paintings.

For the existing methods, there is a study that proposed an algorithm to endow real 3D scenes with a watercolour appearance [14]. They mock the watercolour effect by focusing on processing of image space which includes edge darkening and pigment granulation. Therefore, there are two watercolour paintings that can be cooperated which are the wet-on-wet and the wet-on-dry painting. To achieve more realistic watercolour effects, another study implemented a more complex model based on the Kubelka-Munk model [15]. They improved it by adopting a shallow water simulation, more reliable rendering and optical coloured layers. As a result, many watercolour effects can be created such as edge-darkening, backruns, glazing, etc. For the open source library, OpenCV has a watercolour function under non-photorealistic rendering to add the effect on the image. By simply calling the function, watercolour effects would be applied on the input.

BeFunky (https://www.befunky.com/create/photo-to-art/) and PhotoMania (https://photomania.net/editor) are examples of online applications that can transform images into watercolour images. Both of the applications provide different kinds of watercolour effects for applying on the images. They also allow the adjustment of smoothness and intensity. Furthermore, they provide an interesting feature which is the brush. The brush can be used as an eraser to brush the watercolour effect on the images. Our proposed application has the functionality to apply a watercolour effect on the images by controlling the smoothness and the watercolour-like segmentation scale. There are 7 colour tones for the users to transform the colour style of their images. Our watercolour outputs are more soft and harmonious without the obvious edges. Figure 10 illustrates the output of watercolour applications mentioned.

Figure 10: Outputs of different watercolour applications.

2.6. Pixelation Effect

Pixelation is actually adding blurriness and fuzziness to an image, but it can be regarded as a form of pixel art. In fact, computers view each image as an array or matrix which is different from how humans view images. Therefore, it is possible to create a pixelation effect with some basic functions in most of the programming languages. In particular, Python libraries such as Matplotlib, OpenCV, Pillow, and a project library specially designed for pixelation, pixelate, are all based on the process of resizing the image with different sizes and shapes. Nonetheless, there are still studies on image pixelation and pixel art in mind to explore more possibilities and improve pixelation techniques. For instance, [16] focused on two algorithms, Simple Linear Iter-ative Clustering (SLIC) and Fuzzy Iterative Self-Organizing Data Analysis (FISODATA), thus create an automatic pixelation algorithm for portrait images, [17] presented a novel animation framework tailored to pixel art images based on conventional key-frame animation framework and state-of-the-art image warping techniques, [18] concentrated on abstracting high resolution images into very low resolution outputs with reduced colour palettes in the style of pixel art.

Some existing platforms that perform basic pixelation on images are PineTools (https://pinetools.com/pixelate-effect-image), Pixel It (https://giventofly.github.io/pixelit/), and PEKO STEP (https://www.peko-step.com/en/tool/pixelate.html) whereas Pixel-Stitch (https://www.pixel-stitch.net/), Pixel-Beads (https://www.pixel-beads.net), as well as Pic2Pat (https://www.pic2pat.com/index.en.php) able to perform pixelation with purposes of making crafts through stitching, knitting, making beads art, etc. These applications can produce pixelated images with different pixel sizes, some applications can specify pixel shape and choose colour palettes or show tutorials. Our proposed application has the functionality to pixelate images by specifying the pixel size and number of colours in the output, the colours chosen are the closest to actual colours, but not capable of producing step-by-step tutorials for stitching, making beads arts and setting specific colour palettes. Figure 11 showed the outputs of the pixelation applications mentioned.

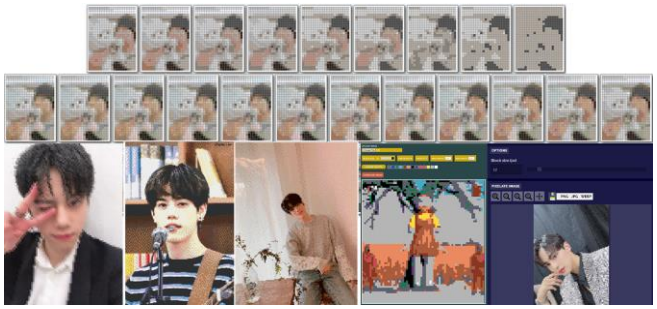

Figure 11: Outputs of different pixelation applications.

3. APPLICATION IMPLEMENTATION AND USABILITY

3.1. Image Enhancements

The image enhancement in our application has four parts which are image equalization, image denoising, image rotation and brightness and contrast adjustment. When it comes to image equalization, CLAHE will be applied on the perceptual lightness L* of CIELAB photos. The contrast limit and title size of the CLAHE function are 2.0 and 8 x 8, respectively. This image-based pre-processing technique improves the quality of face images. As shown in Figure 12, CLAHE is applied to the face image, so the contour of the face is more visual.

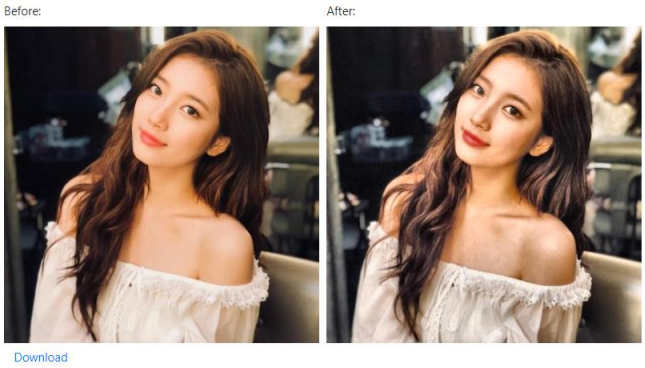

Figure 12: Outputs of image equalization

Next, the image denoising in our application will be used to reduce or remove the noise in the original image, so subsequent images will not be affected by noise. In the image denoising method, the function of *fastNlMeansDenosingcoloured()* has been used with the deciding filter strength, h and *hForcolourComponents* of 10, template window size of 7 and search window size of 21. The parameter is a standard denoising method, which can remove the noise in the colour image. The denoised image has been smoothed from the original image, as shown in Figure 13.

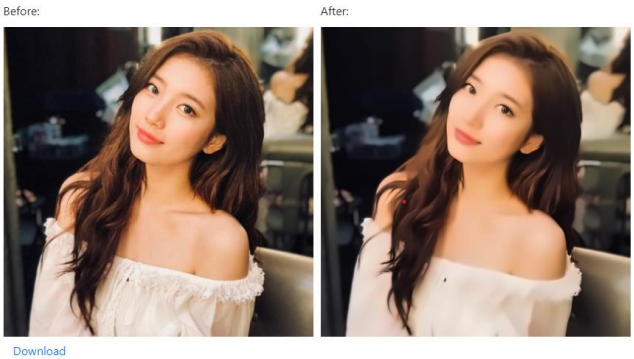

Figure 13: Outputs of image denoising

In addition, the image rotation will be used to customize the image angle required by the user. The rotation angle specified by the user is -90°, clockwise rotation and 90°, counter clockwise rotation. First, get the center of the image to use for rotation matrix function. Next, the rotation matrix is obtained from the *getRotationMatrix2D()* function with the image center, specific angle and scale of 1.0. Besides, apply *warpAffine()* function to the original image, with parameters of the input image, transformation/rotation matrix, preserving the original image size, linear interpolation, and copied boundary mode. Finally, by applying the rotation method, the rotation image with input image and specified angle is produced. A sample image with rotation of 45° is shown in Figure 14.

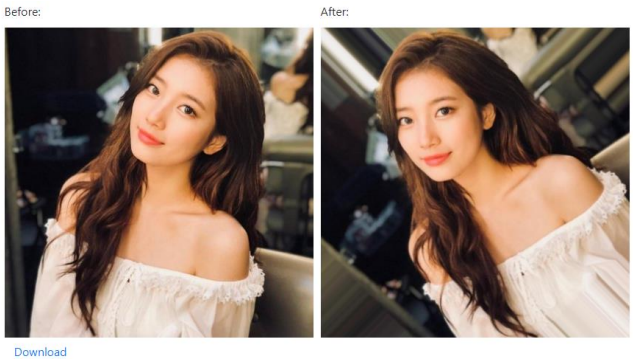

Figure 14: Outputs of image rotation

The last parts of the image enhancement process is the brightness and contrast adjustment. The user-specified contrast and brightness values are between -127 and 127. Brightness is the simplest function to measure the linear conversion of image intensity to display intensity, and contrast is the ratio of the brightest point to the darkest point in the image. Therefore, the user specifies the values of brightness and contrast and applies the function to obtain the output image. As shown in Figure 15, the sample image is applied with the brightness of 30 and contrast of 20.

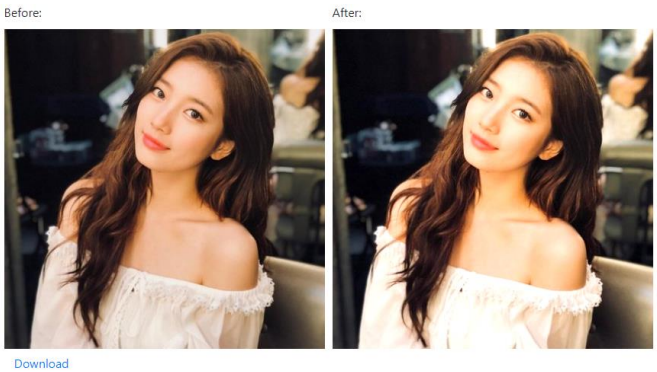

Figure 15: Outputs of brightness and contrast adjustment

3.2. Image Filters

The image filters in our application have three which are sepia, lighting and Clarendon. When it comes to image editing, the sepia filter is one of the most popular. The sepia effect gives the images a warm brown tone. The photographs are given a warm touch by using a sepia filter, which has a very calming effect. The photograph has an antique feel to it due to the sepia tone. Sepia toning was a procedure used in film photography to give black and white photographs a warm brown tone during the printing process. In digital photography, a sepia tone filter can be used to get the same effect. Most image editing apps and software use sepia filters as one of the features. The sepia filter is implemented using *cv2.transform()* and multiplies the pixel values with a special sepia matrix, then normalizes the pixel value bigger than 255 to 255.

The image lighting effect refers to the presence of a halo effect similar to light in the image, and the enhancement of the pixel value of the image in a circular range around the center point of the light. The Python implementation code is mainly to traverse each pixel of the image through a double-layer loop, find the center point of the image, and then calculate the distance from the current point to the center of illumination (the distance between two points in the plane coordinate system) to determine the relationship between this distance and the radius of the center circle of the image. The gray value of the image within the center circle range is enhanced, and the gray value of the image outside the range is retained, and the final lighting effect is generated by combining the boundary range judgment.

Filters are mainly used to realize various special effects of images, and they have a very magical effect in Photoshop. Filters usually need to be used in combination with channels, layers, etc. to achieve the best artistic results. This section will describe a filter processing method based on the Look up Table shown in Figure 16, which obtains a new colour by converting each original colour. For example, a certain pixel of the original image is red (R-255, G-0, B-0), and it turns green (R-0, G-255, B-0) after conversion. After that, all places that are red will be automatically converted to green, and the colour lookup table is to convert all colours once (matrix), many filter functions provide such a conversion matrix, and colour conversion is carried out on the basis of the original colour. Suppose there is now a new filter colour lookup table, as shown in the figure, it is a 512×512 size image containing the colour distribution of each pixel. The following picture is saved locally and can be directly used for image filter processing. It uses the custom *getbgr()* function to obtain the mapped filter colours in the colour lookup table, and then cycle through each pixel in turn.

The example output of the sepia filter, lighting filter and Clarendon filter are shown in Figure 17.

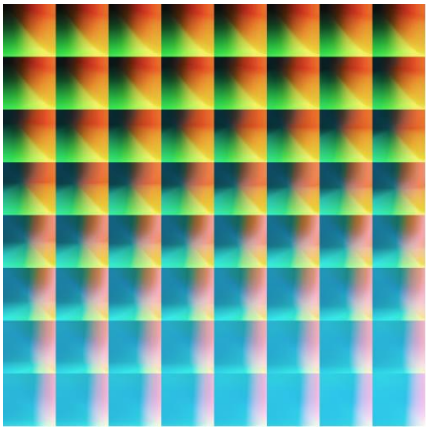

Figure 16. Colour look up table

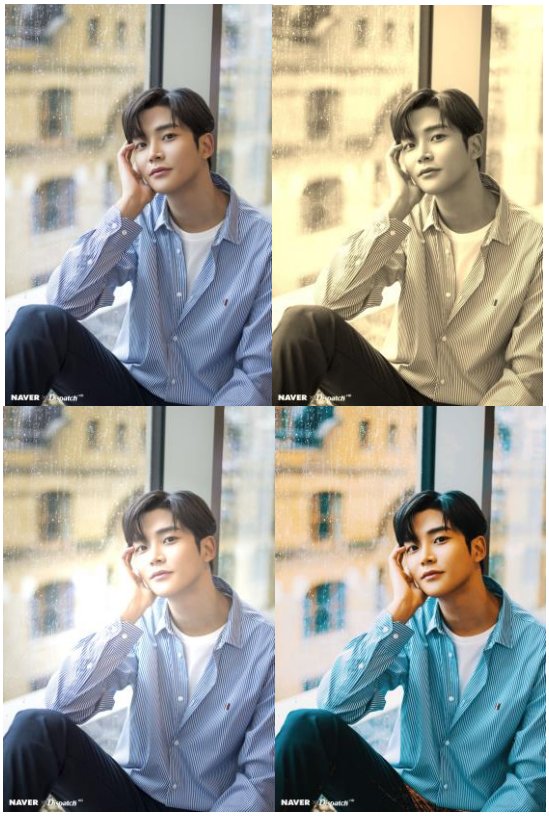

Figure 17: Original image and examples of applying sepia filter, lighting filter and Clarendon filter

3.3. Foreground-Background Segmentation

The foreground-background segmentation feature is intended to remove the original background and highlight the main focus of the image, so the image will be more suitable for doing handmade crafts. This is because the images uploaded by users could be having messy, disorganized backgrounds that might affect the pixelated

image looking less appealing. In order to reduce the burden of the application, we decided to use a ready-use pre-trained deep learning model for background removal to produce a better and clearer output compared to the traditional methods. The Python library, rembg with U2-Net as the basenet, is employed in our application. The background removal process is fully automated without the need to set coordinates of foreground or background manually by using the function *remove(imgfile)*, then continues with the background changing process through Pillow in the Python library. With the Pillow image pasting function (*background.paste(foreground, mask=foreground)*), we are able to replace the background with different settings according to the user preference. The application allows the user to keep the background as transparent only too. The detected foreground will placed in the center of new background after calculating the width and height of new background using *.width* and *.height* functions

The application shows a pop-up window about the preview of the patterned background and solid colour choices for the user to have an idea on the possible upcoming outputs when he chooses to change the background. The user will be provided with four choices during the background removal, which are transparent background, solid colour background, own customized background and pattern background. There are 20 colours prepared for solid colour options and 15 pattern backgrounds prepared in three sizes, users are allowed to upload their own image as the background or keep the original background. This process does not require users to crop out the foreground and background, as the output will be generated automatically. Thus, the outputs produced can be downloaded in png format, as png format helps to retain the transparency and give higher image quality. Figure 18, 19, 20 show the example of background changing.

Figure 18: Example of solid colour background changing.

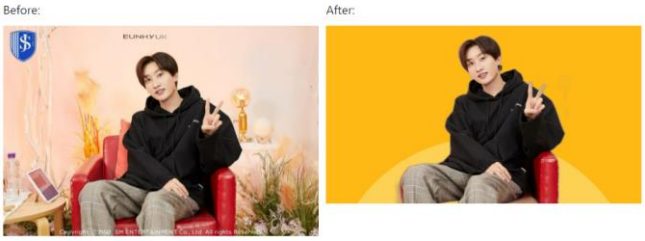

Figure 19: Example of patterned background changing.

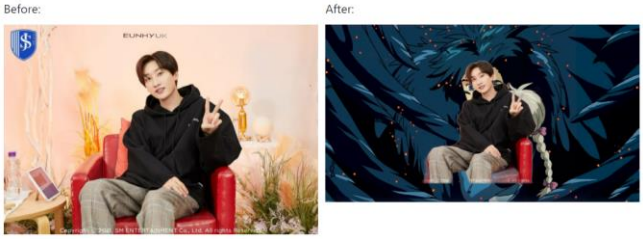

Figure 20: Example of customized background changing.

3.4. Cartoon Effect

There are three types of cartoon effects in our application to allow users to choose based on their preferences. The effects are named classic style, comics style and twilight style. A typical cartoon character avatar can see that the contours of the face are very distinct. The cheeks, chins, etc. are clearly distinguished from the background and hair, but the colour inside the face is relatively uniform. When converted into image processing, it means that the edge is obvious and the surface is smooth. So to get an image cartoonized, in the comics style, we use contourf function to draw contour lines and fill contours with inferno colour map. For twilight style, after reading the input image, The edge is detected using the *cv2.adaptiveThreshold()* function. We transform the image to grayscale before moving on to the edges. After that, the *cv2.medianBlur* function is used to minimise the grayscale image's noise. The line width and blur value are both set to 7. A bigger line size is used to emphasise the thicker edges. When compared to an HD shot, a cartoon sketch will have considerably fewer colours. As a result, a technique known as colour quantization is applied. The number of colours in the shot will be reduced. The procedure is carried out using the K Means clustering technique. The "k" value is changed according to the number of colours we require. After the image has been colour quantized, the noise in the image is reduced. For this task, a bilateral filter can be utilised. The image will be slightly blurred and the sharpness will be lowered as a result. Finally, output the completed cartoonized image with the bitwise and function. The reason that we name it twilight style is because it has a fade and romantic feel when seeing the result. For classic style, we just added the contours function to get the contours. In twilight style we use medianblur and adaptive threshold to get edges mask to add on the quantized and blurred image

while for classic style, we use canny edges to find contours. After adding the contours, the resulting image looks more solid. The example after applying the cartoon effects are shown in Figure 21 below.

Figure 21: Original image and examples of applying cartoonization in comics, classic, and twilight styles

3.5. Oil Painting Effect

There are four types of oil painting effects in our application to allow users to choose based on their preferences. The brush effects are based on the edge detectors of Prewitt, Roberts, Scharr, Sobel. Next, in order to make the picture look like a painting, we applied *MedianBlur* to the picture. Besides, we disrupted the order of colouring and randomly decided which stroke to draw; and in this step, we determined the density of the strokes by skipping some strokes to avoid repeated colouring and speed up. If the parameter of the palette is set to 0, it means that the palette is not used, and the colour at the center of the ellipse is used as the stroke colour; if the palette parameter is given, the colour is quantized according to the parameter. For example, set to 30, then the entire picture can only be coloured with 30 colours per channel; each stroke is coloured with the colour closest to these colours. However, this setting will make the entire colour block become the same colour (for example, the sky is blue); therefore, when we determine the closest colour, we add Random to make the colour block not too monotonous.

When the user continues to make the oil painting effect, the application will pop up a window about the oil painting effect preview, which contains the same palette value and brush size of different oil painting styles, so that the user can understand the possible upcoming output. After that, they need to provide two inputs, namely palette and brush size. We limit the brush size to the range of 1-10 because we think that after trying different brush size values, they are the most suitable values to produce a proper oil painting effect. In addition, we also limit the palette values to the range of 1-30 because we think they are the most appropriate colour stroke values. Besides, they also need to choose the oil painting style to obtain a specific edge effect. Then, the output is automatically generated and can be downloaded with higher image quality in png format. As shown in Figure 22, different brush strokes are used for different oil painting effects on the same image.

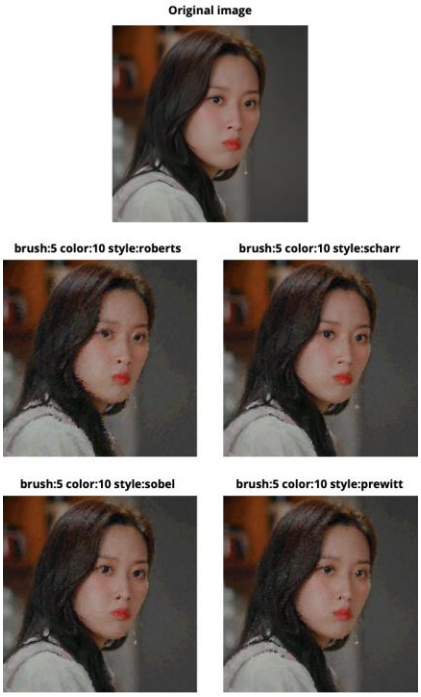

Figure 22: Different oil painting styles with same brush size and colour values

3.6. Pencil Sketch Effect

To achieve pencil sketch effect, we have utilised several functions from the open source OpenCV library. Firstly, we convert the image into grayscale image using the *cvtcolour* function of OpenCV. The grayscale image is then inverted using the OpenCV *bitwise_not* function. Next, the Gaussian blur will be applied to the inverted image. The second argument of the *GaussianBlur* function refers to the kernel size which is preferably set to odd as they are symmetric around the origin. The kernel size determines the level of thickness of pencil edges where the larger the kernel size, the thicker the edges would be. The next step is to invert the blurred image again using the same *bitwise_not* function. Lastly, we will use the divide function of OpenCV to perform element wise division between the gray scale image and the inverted blurred image. The division will output the desired pencil sketch image in black and white colour. To provide colour sketch features, we have created our own

function by looping the black and white output and checking each pixel value. The function then replaces the non-white pixels with the colour chosen by the user. Hence, we have predefined the RGB value of 6 colours which are red, orange, yellow, green, blue and purple for the replacement. Due to the replacement condition constraint (only replace non-pure-white pixels), the colour sketch will work better if the image can replace the background colour with white by using the foreground-background segmentation feature of our application.

When the users proceed to do the sketch effect, the application shows a pop-up window about the preview of the pencil sketch effect with different colours and values of kernel size for the user to have an idea on the possible upcoming outputs when he chooses to apply the pencil sketch effect. After that, they are required to provide two inputs which are sketch colour and kernel size. We have limited the kernel size within a range of 3-199 as we think they are the most suitable value to produce an appropriate pencil sketch after experimenting with different values of kernel size. Then, the outputs are generated automatically and available for downloading in higher image quality using png format. Figure 23 displays the result of applying black and white sketch with kernel size of 111 while Figure 24 shows the blue colour sketch output with kernel size of 11 by using the image that has been pre-processed with foreground-background segmentation.

Figure 23: The black and white sketch output with kernel size = 111.

Figure 24: The blue colour sketch output with kernel size = 11 after applying foreground-background segmentation.

3.7. Watercolour Painting Effect

We have made use of functions in the OpenCV library to implement the watercolour effect step by step. There are four steps which are edge preserving filtering, colour adjustment, mean shift filtering, and image segmentation with mean value filling. First of all, we apply the *edgePreservingFilter* with *RECURS_FILTER* which will perform a normalized convolution filtering to smoothen the image. . Next, colour adjustment is performed to allow the user to change the colour tone of the image. Based on the colour chosen, we will read in another corresponding colour tone image from our sources. The smoothened image also would be converted to CIE Lab format. Then, the mean and standard deviation of those two images are calculated. The colour tone is then adjusted by several operations between two images such as subtraction, multiplication and addition by using the obtained mean and standard deviation. The resultant image will be clipped to prevent overflow and converted back to RGB format. After that, we apply the *pyrMeanShiftFiltering* function from OpenCV to prepare the image for the segmentation. Subsequently, the segmentation using *felzenszwalb* function is performed. The mean value of segmented image is used to fill the resultant image from the colour adjustment just now. Now, the output will look like a watercolour painting.

When using the watercolour effect of the application, the application shows a pop-up window about the preview of the watercolour effect with different values of sigma, colour tone and segmentation scale for the user to have an idea on the possible upcoming outputs when he chooses to apply the watercolour effect. After that, they are required to provide inputs such as sigma_r, sigma_s, colour tone and segmentation scale. The range for sigma_s and sigma_r should be in the range of 0-200 and 0-1 respectively. Seven

colour tones which are blue, brown, gray, green, pink, purple and yellow are available for users to choose. Whereas the range of segmentation scales are not limited. If the scale is higher, it means the watercolour-like segmentation clusters size would be larger. Then, the outputs are generated automatically and available for downloading in png format that can retain higher image quality. Figure 25 illustrates a sample output of the image after applying the watercolour effect.

Figure 25: The watercolour output using sigma_r = 20, sigma_s = 0.4, colour tone = purple and segmentation scale = 40.

3.8. Pixelation Effect

Instead of importing the Python library, pixelate, we have created our own functions for pixelation from scratch with the resize function in Pillow library and K-means clustering method in sklearn library for smoothing, reducing noises and controlling number of colours in the pixelated image. The pixelation function allows users to set the size of pixels to their own preferences by using Pillow's *.resize()* to resize the image according to the pixel size chosen with calculation of the original image size (*.size*) and selection of nearest pixel (*Image.NEAREST*) from input image. Users will be able to select the number of colours to appear in the final output. After defining the number of colours, our application is able to find the most dominant colours and assign each pixel to its corresponding closest colours available. The image will be segmented into different group of pixels using K-means algorithm according to the number of colours chosen, then the pixels will be replaced with its nearest and suitable colours. By specifying the number of colours, users can make the output more suitable to their craft purposes, tools, and materials available. When assigning pixels to generated colours, the pixelated images are being smoothed to create less photorealistic output that is suitable for making crafts.

The application shows a pop-up window about the preview of the pixelation effect with different values of pixel size and number of colours for the user to have an idea on the possible upcoming outputs when he chooses to apply the pixelation effect. The user will need to input two parameters values, where both pixel size and number of colours will affect the details and sharpness of the output. The minimum value and maximum value for both parameters are 2 and 128 respectively. We think the range from 2 to 128 is best suitable for the application after we experimented with different values. Then, the outputs are generated automatically and available for downloading in png format, as png format retains higher image quality. Figure 26 depicts a sample output of the image after applying the pixelation effect.

Figure 26: The pixelation output using pixel size = 32 and number of colours = 6.

3.9. Web Application

The framework we used to build this application is PyWebIO. By importing the pywebio package, we are able to apply the PyWebIO functions such as *put_text()*, *put_image()*, *put_file()*, etc. to build up the application. As mentioned above, a number of libraries which are listed in requirements.txt are applied to deliver those features. Those features are written in readable separated functions. Combining the features and pywebio functions, the website is accomplished to show the instructions, run each feature and provide download functionality.

To use the web application, the user is required to host locally. We have prepared a README.md to guide the user in running the web application. The following commands is required to type in the command prompt:

1. conda env create -f conda-env.yml
2. conda activate pixtono
3. python main.py

After running the main.py script, the user now can access the application via the web browser at url: localhost:80/. An example of how the webpage will look like is shown in Figure 27.

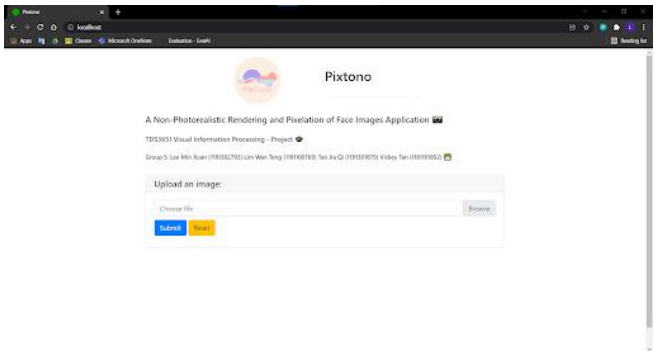

Figure 27: Pixtono web application interface.

4. EVALUATION

In order to evaluate our application, we have conducted a survey by sending out forms, both close-ended and open-ended questions. The survey is separated into three sections: user interface, main features, overall satisfaction. For our survey, we record a video to present the application testing to users. In the video, we present the application flow including uploading image, preprocessing process, four main features of application which are cartoon effect, oil paint effect, pencil sketch effect and watercolour effect, and pixelation effect. Next, we send out the video with the google form to the users to make the evaluation on the application. To make the analysis on our application, we have collected 30 responses to our survey form about user satisfaction on our app.

4.1. Application User Interface

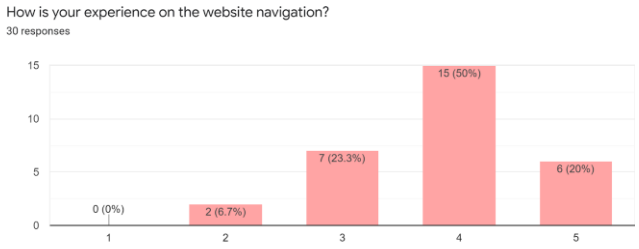

Figure 28: Experience of website navigation.

Figure 28 shows how users respond to the website navigation. Overall, in the video demonstration, many respondents felt it was a good experience to browse the website. A total of 15 respondents had a good experience with website navigation, which means that 50% of respondents rated this question as 4. In addition, only 7 respondents, 23.3% of the respondents rated the website navigation experience as 3, which means normal feeling. Only 5 respondents felt the best experience during website navigation, which means they have no problems with how to navigate the website. On the other hand, only 2 respondents felt the worst experience during the navigation of the website, which means that they had difficulty browsing the website.

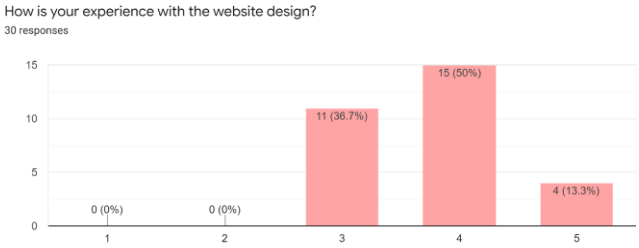

Figure 29: Experience of website design.

Figure 29 illustrates how users respond to the website design. Overall, in the video presentation, many respondents felt it was a good experience in website design. A total of 15 respondents had a good experience with website navigation, which means that 50% of respondents rated this question as 4. In addition, only 11 respondents, 23.3% of them rated the website navigation experience as 3, indicating that they feel normal. Only 4 respondents, 13.3% of them felt the best experience, which means they were satisfied with the design of the website. On the other hand, no respondents felt the worst or bad experience during the website design, which means that none of them felt uncomfortable with the website design.

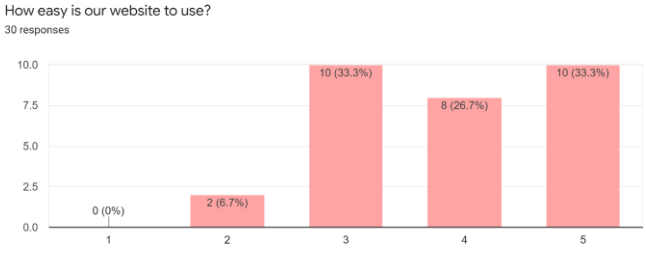

Figure 30: Ease of use on the website

Based on Figure 30, a total of 10 respondents felt it was very easy to use the website, which means that 33.3% rated this question as 5, and the same number of respondents thought the website felt average experience on the ease of use. In addition, only 8 respondents, 26.7% of them rated the ease of use as 4, indicating that they feel easy to use the website. On the other hand, only 2 respondents felt the difficulty to use the website during the video demonstration.

4.2. Application Main Features

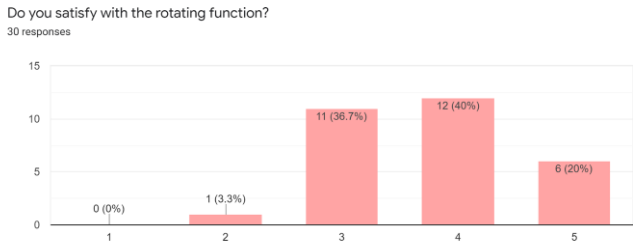

Figure 31: Experience of website navigation.

Figure 31 illustrates the level of user satisfaction towards the rotation function of the application. Overall, the respondents are satisfied with the rotation function as 60% of the respondents rated their satisfaction above 4. There are 36.7% of the respondents rated their satisfaction with 3, indicating that their satisfaction level is average only. However, there is 1 respondent gave a rate of 2, which means that the rotation features are not good and need to be further improved.

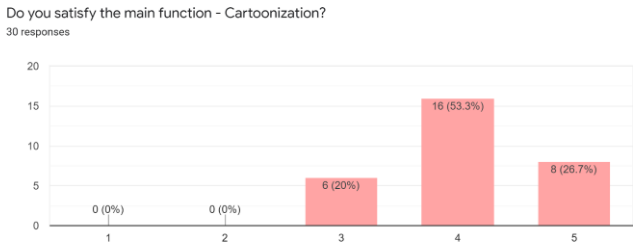

Figure 32: User satisfaction with the cartoonization function.

Figure 32 illustrates the level of user satisfaction towards the cartoonization function of the application. Overall, the respondents are very satisfied with the cartoonization function as 80% of the respondents rated their satisfaction above 4. However, there are still 20% of the respondents rated their satisfaction with 3, indicating that their satisfaction level is average only. Additionally, no one gives the rate below 2. We might say that the cartoonization effect is very acceptable for all the respondents.

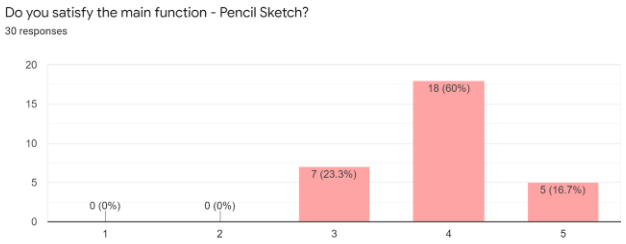

Figure 33: User satisfaction with the pencil sketch function.

Figure 33 illustrates the level of user satisfaction towards the pencil sketch function of the application. Overall, the respondents are very satisfied with the pencil sketch function as 76.7% of the respondents rated their satisfaction above 4. However, there are still 23.3% of the respondents rated their satisfaction with 3, indicating that their satisfaction level is average only. Additionally, no one gives the rate below 2. We might say that the pencil sketch effect is able to provide acceptable output to the users.

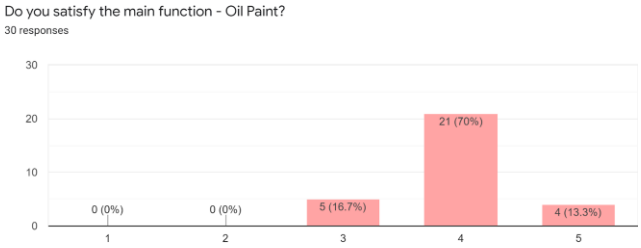

Figure 34: User satisfaction with the oil paint function.

Figure 34 illustrates the level of user satisfaction towards the oil paint function of the application. Overall, the respondents are very satisfied with the oil paint function as 83.3% of the respondents rated their satisfaction above 4. However, there are still 16.7% of the respondents rated their satisfaction with 3, indicating that their satisfaction level is average only. Additionally, no one gives the rate below 2. We might say that the oil paint effect is very acceptable for all the respondents.

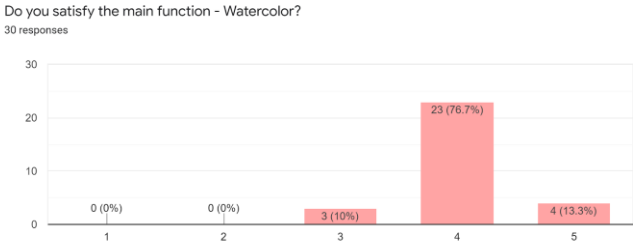

Figure 35: User satisfaction with the watercolor function.

Figure 35 illustrates the level of user satisfaction towards the watercolour function of the application. Overall, the respondents are very satisfied with the watercolor function as 90% of the respondents rated their satisfaction above 4. However, there are still 10% of the respondents rated their satisfaction with 3, indicating that their satisfaction level is average only. Additionally, no one gives the rate below 2. We might say that the watercolour effect is able to provide good output to the users.

4.3. Application Overall Satisfaction

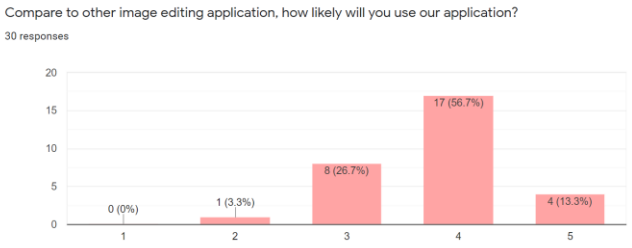

Figure 36: Experience of website navigation.

Figure 36 shows a bar chart on how our users respond to how likely they will use our application. Overall, around 70% of the users which is a total of 21 respondents are likely and desire to use our application. 8 respondents, which also 26.7% of the respondents rated 3 which means they might use our application. There is only 1 respondent which is 3.3% of the respondents less likely to use our application. Compared to other image editing applications, our application is still in the favor of users as 96.7% of the respondents respond neutrally and positively.

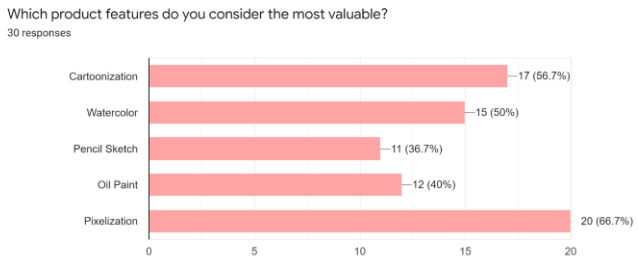

Figure 37: Preference of the users on the application features

Based on Figure 37, we have obtained a result on which pixelization is the most preferred feature with the highest rate with 20 respondents which is 66.7% of the respondents. This is probably due to the pixelization seldom appearing in image editing apps, therefore the users think it is interesting and valuable for real world applications like beads art. The second highest rating feature is cartoonization which occupies a total of 17 respondents which is also 56.7% of the survey respondents. Watercolor feature is also considered a valuable feature as it was rated by half of the 30 respondents which is 50% of the respondents. Minority of our respondents think that pencil sketch and oil painting are valuable to them, which occupies 36.7% and 40% of the respondents, respectively.

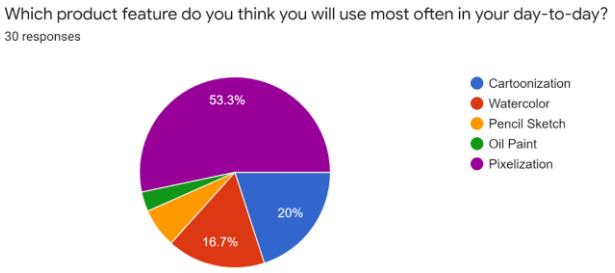

Figure 38: Product features that will use often in daily.

Figure 38 is a pie chart that illustrates which product features that the users think they will use frequently in daily. Majority of respondents think that pixelization is the most frequently used daily which accounts for 53.3% of the respondents. Then, followed by cartoonization and watercolor features which occupies 20% and 16.7% of the total respondents. The minority of the respondents think that pencil sketch and oil painting are the features they will often use in daily life. The responses of respondents support the statement that pixelization, cartoonization and watercolor are the most useful features in our web application.

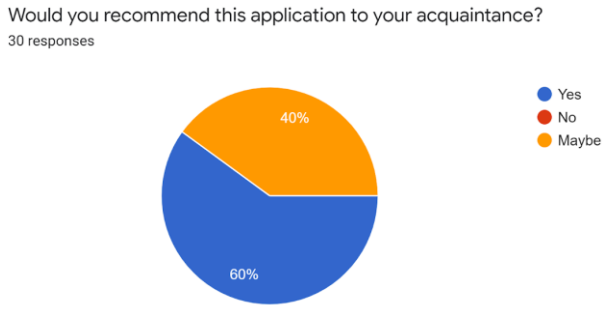

Figure 39: The willingness of respondents in recommending the application to their acquaintance.

Based on Figure 39, it can be observed that all the respondents responded naturally and positively to recommend our application to their acquaintance. 60% of the respondents want to make the recommendation towards their acquaintance. While 40% of the respondents put the recommendation of this application in their consideration only.

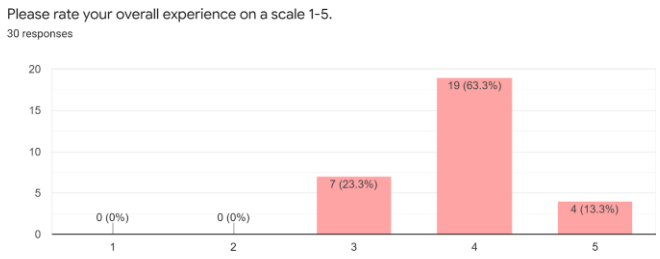

Figure 40: The overall experience of the respondents in using the application

Figure 40 shows the overall experience of the respondents in using the application. No respondents rate their experience as 1 or 2, inferring that our application is not worse. Around 23.3% of the respondents felt that our application is normal and might give satisfactory results to them on those provided features. More than half of the respondents, 76.6% of them think that the application brought them very good experience when using the application. Comprehensively, it might be concluded that our application has potential in the eyes of the users. The performance of the web application is above average and approved by them.

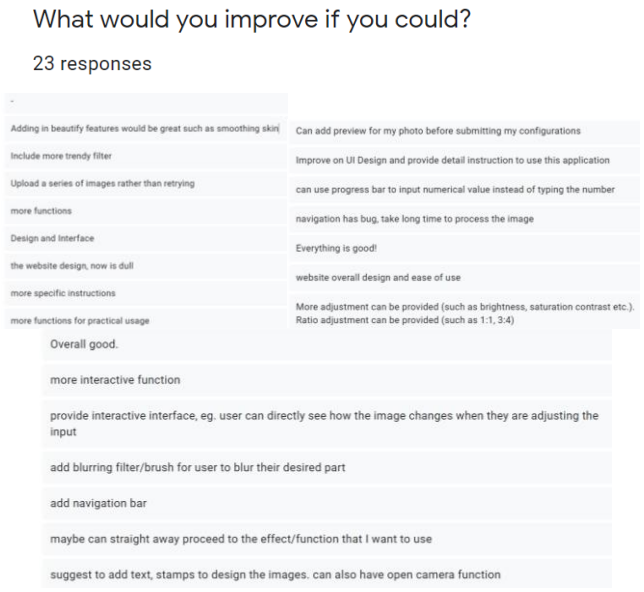

Figure 41: The improvements suggested by respondents.

The final question gathered opinions of respondents on the improvement of the Pixtono website. Among the 30 responses only 22 of them answer the questions with some useful suggestions. We are able to categorize the responses into 2 aspects: features and website design. They pointed out that the website is required to have more functions or features for users to customize the images. For instance, adding beautify filters, adjustment on brightness and contrast, image ratio adjustment and blurring filter. One of the most valuable suggested improvements for website main features is adding text, stamps to design the images as well as implementing a camera opening function to take pictures in real time. In addition, the most valuable suggestion for improvement in website design is to provide an interactive interface, such as adding a navigation bar, using a progress bar to enter values instead of numbers, and previewing the changes in the image when the user adjusts the input.

5. CONCLUSION

In conclusion, we have implemented a web hosted application using PyWebIO and several different methods using Python OpenCV that can perform non-photorealistic rendering and pixelation of face images. In the web application, we perform preprocessing processes including image enhancement, image filter and foreground-background segmentation on the input image. The main features of our application are cartoon effect, oil painting effect, pencil sketch effect and watercolour effect. Last but not least, our application can pixelate the input image based on user preferences. Our application aims to help in real world art works applications. Based on the survey results obtained from users, pixelation, cartoonization and watercolor features are acknowledged as the most useful features in our application. All features of our application eventually meet our users' expectations and satisfaction. There is still room for improvement for our application and our end users have suggested a lot of useful functions that can be future implemented in our application. In short, our application can be improved by adding additional application features, for instance, adding text, stamps or stickers on the output and allowing users to free-hand drawings. An open camera function can be also implemented to allow users to take selfies and use our application in real time. Moreover, we also obtained practical suggested improvements on web interface design by giving more detailed instructions, interactive output and ways to adjust the parameters of each feature. We greatly appreciate our users providing us so much of the helpful and functional suggestions. Our team has planned to use a more scalable and versatile web or even mobile application framework so that our developer can design a more attractive user interface, provide clearer instruction and useful features. Besides, we intend to improve the practicability of the application by providing step-by-step tutorial of different handcrafts and colour themes based on different merchants of specific crafts materials. We hope that we can deliver a higher quality and advantageous product to our valuable users in the future.

6. REFERENCES

[1] Cheng, H. D., & Shi, X. J. (2004). A simple and effective histogram equalization approach to image enhancement. Digital signal processing, 14(2), 158-170.

[2] Celebi, M. E., Kingravi, H. A., & Aslandogan, Y. A. (2007). Nonlinear vector filtering for impulsive noise removal from colour images. Journal of Electronic Imaging, 16(3), 033008.

[3] C M Maheshan, C M Maheshan & Kumar, H.. (2020). Performance of image pre-processing filters for noise removal in transformer oil images at different temperatures. SN Applied Sciences. 2. 67.

[4] *Online Photo Editor - Tuxpi.com.* (2013). https://Www.Tuxpi.Com/. https://www.tuxpi.com/

[5] Bouwmans, Thierry. (2014). Traditional and Recent Approaches in Background Modeling for Foreground Detection: An Overview. Computer Science Review. 11. 10.1016/j.cosrev.2014.04.001.

[6] Cartoon - Pinkney, J. N., & Adler, D. (2020). Resolution Dependent GAN Interpolation for Controllable Image Synthesis Between Domains. arXiv preprint arXiv:2010.05334.

[7] Z. Shahcheraghi and J. See, "On the effects of pre- and postprocessing in video cartoonization with bilateral filters," 2013 IEEE International Conference on Signal and Image Processing Applications, 2013, pp. 37-42, doi: 10.1109/ICSIPA.2013.6707974.

[8] P. (n.d.-b). PhotoCartoon.net | FREE Photo to Cartoon Converter (Online). Https://Photocartoon.Net/. https://photocartoon.net/

[9] LunaPic.com Photo Editor tool. (n.d.). Lunapic. https://www10.lunapic.com/editor/

[10] Luft, T., & Deussen, O. (2006). Real-time watercolour for animation. Journal of Computer Science and Technology, 21(2), 159-165.

[11] Lee, H., Kwon, S., & Lee, S. (2006, June). Real-time pencil rendering. In Proceedings of the 4th international symposium on Non-photorealistic animation and rendering (pp. 37-45).

[12] Kim, G., Kim, T., Lim, D. H., & Yim, C. (2013). Sketch filter for feature extraction and rendering applications. Electronics letters, 49(13), 805-806.

[13] Lu, C., Xu, L., & Jia, J. (2012, June). Combining sketch and tone for pencil drawing production. In Proceedings of the symposium on non-photorealistic animation and rendering (pp. 65-73).

[14] Luft, T., & Deussen, O. (2006). Real-time watercolour for animation. Journal of Computer Science and Technology, 21(2), 159-165.

[15] Curtis, C. J., Anderson, S. E., Seims, J. E., Fleischer, K. W., & Salesin, D. H. (1997, August). Computer-generated watercolor. In Proceedings of the 24th annual conference on Computer graphics and interactive techniques (pp. 421-430).

[16] Yunyi Shang, Hon-Cheng Wong. (2021). Automatic portrait image pixelization. Computers & Graphics Forum. 95. 47-59. 10.1016/j.cag.2021.01.008.

[17] Kuo, Ming-Hsun & Yang, Yong-Liang & Chu, Hung-Kuo. (2016). Feature-Aware Pixel Art Animation. Computer Graphics Forum. 35. 411-420. 10.1111/cgf.13038.

[18] Gerstner, Timothy & DeCarlo, Doug & Alexa, Marc & Finkelstein, Adam & Gingold, Yotam & Nealen, Andrew. (2012). Pixelated image abstraction. 29-36.