Elements: Software for Image Editing Over Voice Commands

Akash Johnny Kunnath, Albin Saji, Benjamin G Nechicattu,
Done Maria James, Suma R
St. Joseph's College of Engineering and Technology, Palai

Abstract:- Image editing is simply the processes of manipulating images, in whatever format they are, for instance, digital photographs, old photo-chemical photographs, or illustrations. Traditional analogue image manipulation involves photo retouching, using mechanical tools like an airbrush to modify photographs else by editing illustrations with any traditional art techniques. Graphic software programs can be broadly categorized into vector graphics editors, raster graphics editors, and 3D modellers. This software is the primary tool with which a user may manipulate, enhance, and transform images. Many image editing programs are also used to render or create computer art from scratch over voice commands and direct interaction.

I. INTRODUCTION

Image editing from scratch has become a time-consuming process for non-professionals as well as for upgrading professionals. Learning chunks of shortcuts and completely accessing editing tools via mouse and keyboard has become difficult, time-consuming and particularly overhead for at least a few. Here, we introduce an image editing interface that comprises of vocal command recognizer, image editing is difficult to perform with voice alone. For flexible and easy editing-control we use both voice and manual editing interaction, using mouse and keyboard. Selecting an object or a layer within the workspace has become easier. The editing panel is a grid fashion workspace and x-y axes (rulers) are scaled for selection of points at the workspace where an image is to be edited. This application contains an image storage directory linked to the desktop so that importing images becomes easy which is already in the application storage. There are combinations of filters that provide a professional touch to the images. Elements like adding texts and formatting, colour-comb are add-on features. The functions with varying values can be adjusted in percentage/values by saying it while specifying the arguments. Voice interface makes complex tasks easier and accessible as they Allow users to simply state goals without learning an interface.

Fig 1:- Info Screen

II. INTENDED AUDIENCE

This software concentrates on people who are involved in image editing. Professionals, freelancers, tutors, students and even people who love to make images better and look good. Moreover, it especially concerns people who are differently-abled who are involved in image editing, with good craftsmanship, may be physically challenged.

'Elements' a user-friendly voice-enabled editing platform uses Voice commands to alter images. Out of numerous editing software that are available currently, none of them provide voice capability for editing which in turn makes these editors much more difficult. One of the major add on benefits is that this can be used by the differently-abled persons too mainly those who have some physical disability which makes '**Elements**' adaptable to a large range of users. All you have to do is just say what you have to do. This makes it easier to use and simply you don't require by heart commands anymore. Image editing software like Photoshop, Fotos, GIMP, etc. have a wide variety of codes and commands which are a bit difficult to learn, and without these commands by hearted no one can work on any of these platforms with increased efficiency and full accuracy. By using 'elements' the same Performance accuracy can be achieved without even by hearting a single piece of command. You can do what you need to do just by saying what is to be done.

III. TECHNOLOGIES USED

➢ *Python*

Core Programing is based on Python Programing language which is more convenient flexible and fast. Python is more understandable as well readable. Execution and complexity of the program is comparatively easier and less respectively. Python is an interpreter language which helps in sequential execution if the program.

➢ *Visual Studio*

We used visual studio programing platform for the development of the project. WPF along with C# framework makes it easier to integrate all the functions and make the software

➢ *Tkinter Python GUI*

Python have Tkinter GUI which makes combining the scripts together. This makes it executable on any machines that have python with in thus making the program cross platform.

➢ *Pillow Library*

The Python Imaging Library[iii] (PIL/pillow) adds image processing capabilities to the Python interpreter. Basically every operations on image can be done using this pillow library. This gives wide file format support, an efficient internal representation, and fairly powerful image processing capabilities. PhotoImage and BitmapImage interfaces helps to show the image. Pillow library supports image resizing, rotation and arbitrary affine transforms.

➢ *Natural Language toolkit*

Natural Language toolkit[iv] [vi] is used in order to get the speech and convert it to a machine understandable form so that the machine can make meaning from it. Every commands that is given to the system is tokenized by the NLTK and this enables the system to find out what operation is to be done on the image taken.

➢ *Google Speech Recognition Engine*

Google speech recognition engine[x] coverts the speech that is captured to corresponding text[i]. This text is then used by the Natural Language toolkit (NLTK). The Speech is recorded by the system and acquires Google API for speech recognition and uploads the speech to generate the corresponding text[ii].

IV. PLATFORM

The software works on Windows PC. Tkinter GUI[vii] can be run on any devices that supports python. Thus can be installed on Windows and Linux Machines and mac OS too. Thus making it a cross platform program.

V. DESIGN AND IMPLEMENTATION CONSTRAINTS

It makes use of Google Speech recognizer. Google Speech Recogniser works over a fast internet connection. Hence the application requires a fast internet connection. Access to mic and recording is a must.

VI. ASSUMPTIONS AND DEPENDENCIES

It makes use of Google Speech recognizer to take in voice commands from the user. Hence Speech recognition library is needed to be installed along as a dependency. Pythons' PyAudio package is needed to be taken into consideration since the software involves the usage of the system mic. Python's Pillow library is a powerful image processing support. Image manipulation and Enhancement can easily be done using the Pillow package. Hence Python Pillow Package has to be considered as a major dependency. Python GUI Tkinter[vii][viiii]is used for the development of the project.

VII. SYSTEM FEATURES

An image editing project requires a 4 GB recommended Memory. So the system must provide the recommended space. The recommended processor speed of an average 2.0 GHz is suggested. All since it involves a continuous rendering of the image. A graphics card is recommended, if not available, process.

VIII. USER INTERFACES

The user interface is made simpler so that everyone can use it even without any prior knowledge of how the software works. Got a clean interface than that of any other UI provided by similar products.
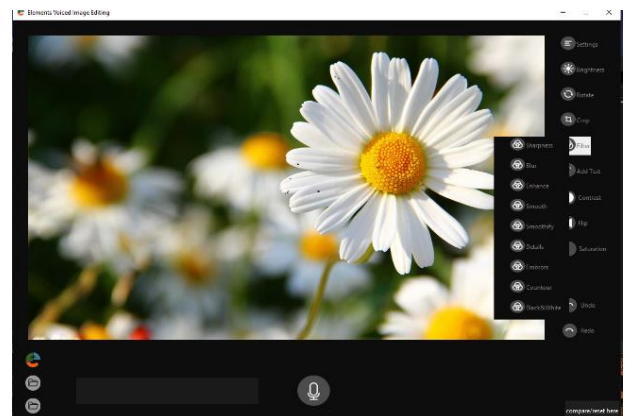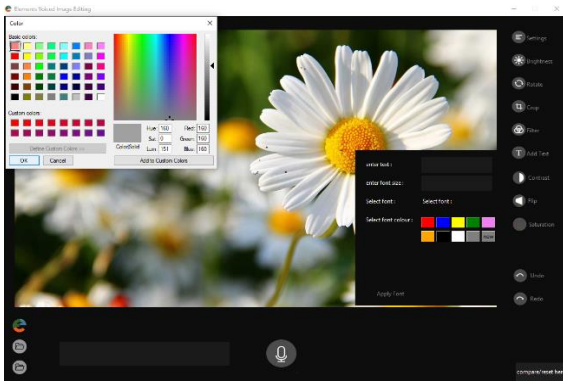

Fig 2:- Screenshot of the Workspace with filters

Fig 3:- Add Text Option

An example of the UI is shown above (fig 2). Tools are arranged on one side which makes it easier to access. The workspace[viii] is equipped with Suggestions and fine-tuning sliders (fig 3 text options) for precision.

Tools are probably less accessed since the project works over voice commands.
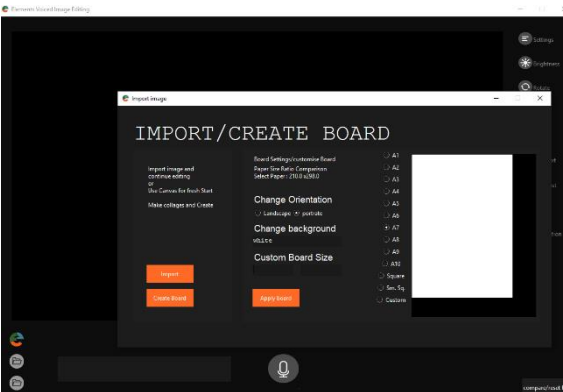

Fig 4:- Create board

The in-app file browser makes it easier to browse through the image files in your system over voice commands. You can either start with an image or with a board (fig 4). Board feature requires more customisation yet to be done to work with voice which is not available in this version. In later versions board will be as easy as drawing on paper with voice.

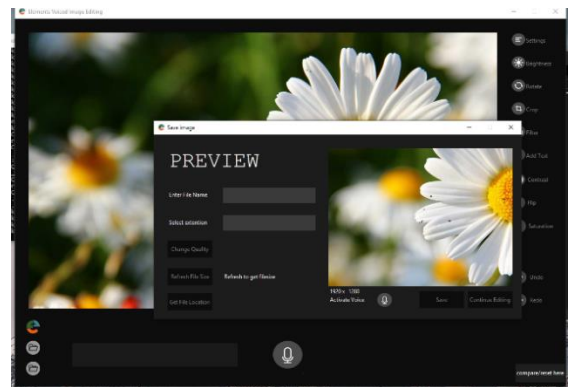Workspace imports images to work with. Fig 5 shows the voice activated saving screen.

Fig 5:- Save options

With great precision, images can be edited in the workspace.

IX. HARDWARE/COMMUNICATION INTERFACES

Hardware interface involves a microphone which comes inbuilt with most of the Personal Computers. If not available, an external mic can be plugged in. A microphone is a must, since we need to gather the voice commands.
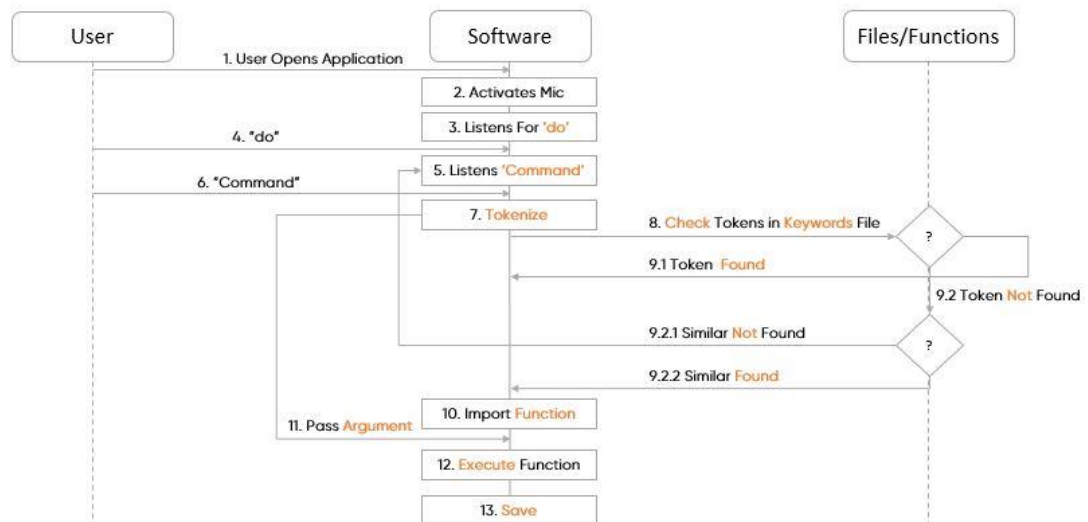
X. PROGRAM EXECUTION FLOW

Fig 6:- Sequence diagram

The Program begins when the application is open. The application welcomes the user with a splash screen. Soon after the application files and libraries are loaded it checks for the internet access. If an internet access is available the mic gets activated and listen for a 'do' command. After a 'do' audio fingerprint is detected you can say any command to be performed in the image.

Selecting the image is much easier with an in app file browser which shows the images with in the PC. All you have to do is to say the name of the image or select the image manually. Selected image is brought to the workspace window. Where you can perform image editing. Now we need to say what operation had to be performed on the image. It is a command. The command is then converted to its corresponding text via Googles Speech Recogniser API. API returns the corresponding text.

The command is now tokenised to tokens. For every token, compares to a keyword in the keyword file. If token found, calls the corresponding function and performs the action. Else if no token is found in the keyword file, token is compared with similar file, to avoid miss predictions. If a similar keyword is found. Then the corresponding function to the 'similar keyword' is called and then performs the action on to the image. For some functions, arguments are needed to be passed. For instance, say angle for rotation. When a rotation function is called, an argument has to be passed, angle. Now it's turn for the argument to be listened and is passed to the function. We can perform enough actions on the image until a save or quit command appears. Save command confirms the edited image and saves the image in new name and a new extension as the user prefer. Quit command quits the image editing window without saving the changes.

XI. IMAGE MANIPULATION OPERATIONS

The usage of pillow library brings up a large space for image editing[v]. Among them few are loaded in to the application.

➢ **Rotation.** Rotation operation rotates the image in the workspace window. The user can specify the angle to which the image has to be rotated. Say rotate to activate command over voice, keyboard press r, or select rotate.

➢ **Brightness.** User can now change the brightness by saying change brightness. Brightness too needs a parameter. A floating point number is passed as the argument for brightness which in turn increase the brightness by that much. Say Brightness to activate command over voice, keyboard B, or select brightness.

➢ **Contrast.** Contrast increases the contrast of the image by a floating point value. Say contrast to activate command over voice, keyboard c, or select contrast.

➢ **Saturation.** This operation Increase the saturation of the image or reduces the saturation. Say saturation to activate command over voice, keyboard S, or select saturation.

➢ **Flip.** Flips the image left to right or up to down. This makes the image looks like mirrored or tilted upside down. Say flip right/ flip up to activate command over voice, keyboard F/f, or select flip.

➢ **Warmth.** Increases the red in every pixel and makes the image warmer. Say warmth to activate command over voice, keyboard w, or select warmth.

➢ **Text.** Now we can add text in to an image. When the text function is called, it asks for the sentence to be added in the image. Varity of fonts are also incorporated along. Font type has to be specified by specifying the font name. Position of the text in the image has to be specified by passing the x, y co-ordinates or by saying how much to right or down. Say add text to activate command over voice, keyboard T, or select text.

➢ **Black and white filter.** Now a days black and white images brings nostalgic feel to the image. This can be brought to the image by a call and specify the strength of the filter. Say black and white to activate command over voice, or select black and white.

➢ **Sharpness.** Increases the sharpness of the image and makes the image crispier and less blurred. Say sharpness to activate command over voice, or select sharpness.

➢ **Detail.** Increases the details and the structure of the objects in the image. Say details to activate command over voice, keyboard d, or select details.

➢ **Crop.** Crop the image by passing four arguments corresponding to left, upper, right and lower of the image. Say crop to activate command over voice, keyboard C, or select crop.

➢ **Blur.** Blurs the image or reduces the sharpness of the image which gives a spread appearance to the image. Say blur to activate command over voice, or select blur.

➢ **Contour.** This can select the pixels with same intensity and find out the edges in an image. Say contour to activate command over voice, or select contour.

➢ **Edge Enhance.** Edge enhance enhances the edges of objects in an image. For more edge enhance use more edge enhance. Say edge to activate command over voice, or select edge enhance.

➢ **Emboss.** Creates an emboss effect to the given input image. Say emboss to activate command over voice, or select emboss.

- ➢ **Smooth.** Just like blurring, this function smoothens the objects in the image. For more smooth use more Smooth. Say smooth to activate command over voice, or select smooth.

- ➢ **Resize.** This function reduce the image size, resolution thus reducing the memory space occupied by the image

while saving. Say resize to activate command over voice, or select resize in save.

- ➢ **Save.** Save function saves the image in a specific name and extension format as the user specify in a user space. Say save to activate command over voice.
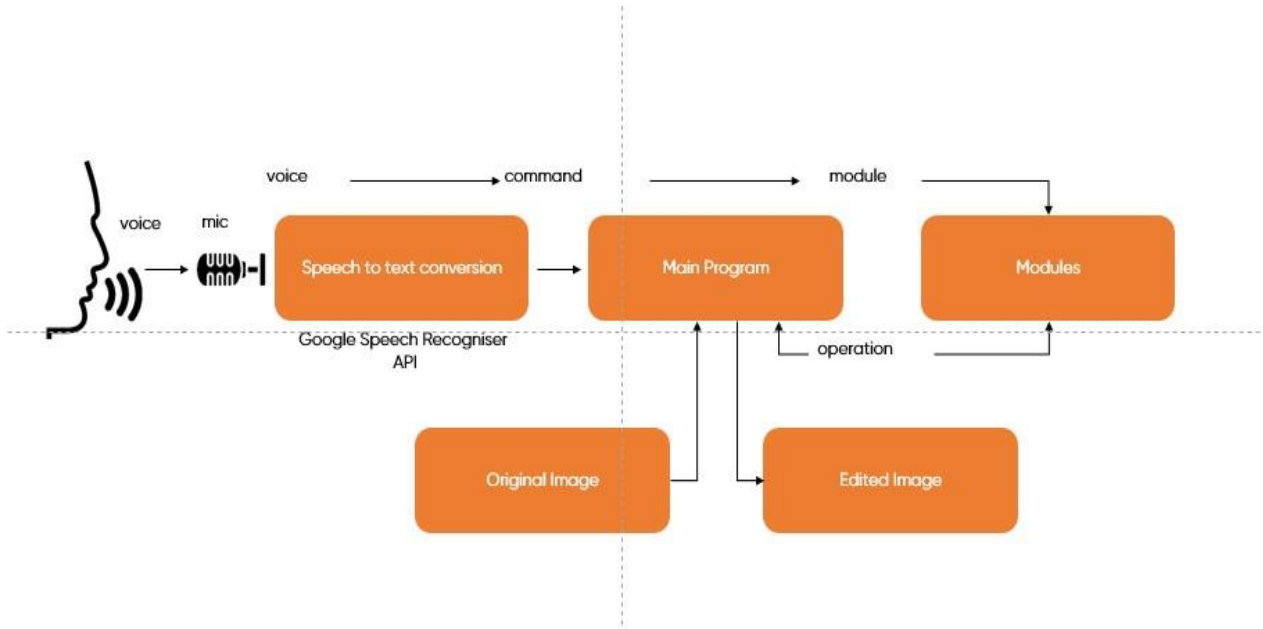
XII. BLOCK REPRESENTATION

Fig 7:- Block Diagram

XIII. RESULT

We designed our evaluation to solve few tasks. How does our proposed multimodal interface compare with a traditional image editing interface? Success rate for both interfaces were identical. Even though the multimodal interface slightly shows more attraction. Figures fig 8 is a test image. Following figures (Fig 9–23) are the output of individual operations on image.

Fig 8:- Test Image

➢ *Output*

Fig 9:- Rotated Image

Fig 10:- brightened Image

Fig 11:- contrast varied Image

Fig 12:- saturated Image
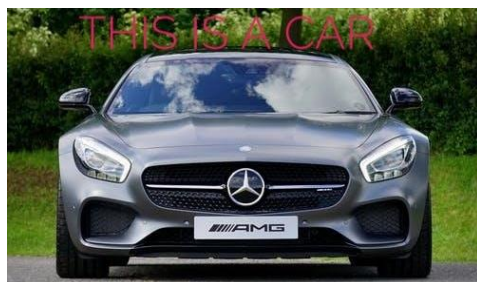

Fig 13:- flip vertical

Fig 14:- flip horizontal

Fig 15:- Adding text

Fig 16:- Black and white

Fig 17:- sharpenned Image

Fig 18 : detailed Image

Fig 19 : cropped Image

Fig 20 : blured Image
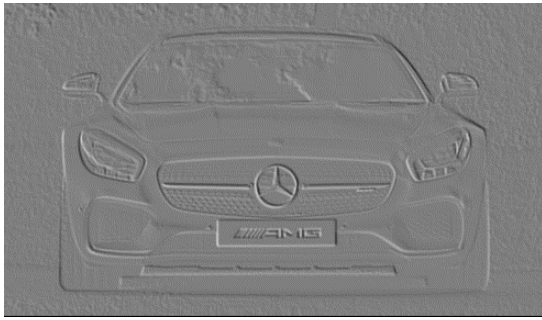

Fig 21 : edge enhanced Image

Fig 22:- embossed Image

Fig 23:- smoothed Image

XIV. CONCLUSION

Here we introduced, ELEMENTS, A multimodal interface system to enhance image editing tasks through voice and conventional direct manipulation. Other than editing functionalities "Elements" is enabled with browsing of an image as well as saving an image after editing. We can browse our file manager or even internet by using appropriate voice commands. After editing procedure is complete user can save image using the "save" command and we can specify appropriate location as well as name in which image is to be saved. Thereby implementing each functionalities with voice. Coming to the editing functionalities we have implemented all the features that are essential for an editing tool. Features include brightness, Contrast, crop, rotate, a total of 9 filters etc., and all these using voice commands. "Elements" have an add on functionality of image compression. Image that we select for editing maybe of larger size and we can compress them after according to our requirement, compression ratio is on a scale of 0-100. The key feature that makes "Elements" unique from other editing tools is that it is voice enabled, as it is voice controlled it can be used by the "differently abled people". With the board facilities in later versions makes it more advanced. Voice commands are less complex than shortcuts and is has a user-friendly UI which all makes it easy to use. So now editing is no more a complex task just tell what to do and it's done.

REFERENCES

[1]. *Research on Speech Recognition Technology and Its Application,* *Youhao Yo* international conference on computer science and Electronics engineering. 2012

[2]. *Speech Recognition System : A Review Sandheep Sharma, Nithin Washani ,* International Journal of computer Applications. April 2015

[3]. *Pillow 7.1.2,* https://pypi.org/project/Pillow/ , 2020

[4]. *NLTK 3.5 documentation*, https://www.nltk.org/ , 2020

[5]. *Python | Working with the Image Data Type in pillow*, geeks for geeks https://www.geeksforgeeks.org/python-working-with-the-image-data-type-in-pillow/ , 2020

[6]. *Natural_Language_Processing-Python,*_tutorials_point https://www.tutorialspoint.com/natural_language_processing/natural_language_processing_python.htm, 2020

[7]. *Python-GUI_Programming_(Tkinter),*_tutorials_point https://www.tutorialspoint.com/python/python_gui_programming.htm, 2020

[8]. *tk-tools 0.12.0*, https://pypi.org/project/tk-tools/, 2020

[9]. *pocketsphinx,* https://github.com/cmusphinx/pocketsphinx , 2020

[10]. *SpeechRecognition3.8.1*,_https://pypi.org/project/SpeechRecognition/ ,2020