

Documentation

How to run

1. Clone the repository from the github. Make sure to install all the libraries mentioned below. Depending upon the operating system, the installation procedure may change. We have written the installation procedure for Ubuntu 20.04. Make sure python3 is installed and of version 3.6.9 or plus. Make sure pip3 is installed and updated.
2. To run the GUI, run the `main.py` file present in `src/GUI` folder. Make sure to be present in project directory and run the following command to invoke the GUI.

```
python3 ./src/GUI/main.py
```

3. In the GUI, there would be a option to select an image file `open file`(please make sure to select an valid format image file) at the top center of GUI, and an option to choose one among the recommended images(which are displayed at the bottom of GUI). The choosen image would be displayed on screen under the title input image.
4. The occluded part (part of image that needed to be removed) has be drawn/traced on the input image. We can draw on the input image using cursor. Please make sure to draw bit slowly on the input image, so that it would be easy to trace the boundary drawn by user. It would be recommended to draw a closed shape on input image to better visualize the results.
5. After drawing a boundary on the image, click on `inpaint` button, to run the algorithm. The output would be shown under the tile output image. The algorithm may take long time to execute(5minutes - 10 minutes), please be patient until the results appears. To track the progress, we would be printing the stage of execution of the algorithm in console(terminal).

Libraries Used :

1. numpy :

To install numpy libraries, run :

```
pip3 install numpy
```

2. tkinter :

To install tkinter library, run :

On Linux:

```
sudo apt-get install python3-tk
```

3. os:

os library would be installed/present along with the python.

4. PIL:

To install PIL library , run :

```
sudo apt-get install python3-pil python3-pil.imagetk
```

5. sys

The sys module comes packaged with Python.

6. matplotlib

To install matplotlib, run :

```
pip3 install matplotlib
```

7. cv2

To install cv2, run :

```
pip3 install opencv-python
```

8. functools

The functools module comes packaged with Python.

Links

Input Images folder :

<https://drive.google.com/drive/folders/1aRwHv7Htnn3KoxRn3FZ8ijzrxOve6kZs?usp=sharing>

Output Images folder :

https://drive.google.com/drive/folders/1wkzILOEARgR90bcAcMb4x8gC2A4XVz_n?usp=sharing

Presentation slides :

https://drive.google.com/drive/folders/1ep6zo36RXSEoXO3RxAmb6_VBySFE8Q2X?usp=sharing

Navigation Through code

ANN

```
./src/ANNwithPatchMatch.py
```

- Implemented Patch Match algorithm for approximate nearest neighbour.
- Outputs the shiftmap which maps occluded pixels to some pixel in unoccluded region
- shiftmap is useful in other parts of the algorithm including reconstruction of image etc

Reconstruction of image

```
./src/ReconstructingTheImage.py
```

- Will assign the intensities of the occluded region using the shift map which is calculated using previous algorithm.
- Returns reconstructed image with modified intensities in occluded region.

Onion peel initialization

```
./src/onionPeel.py
```

- We initialize the regions of occluded region layerwise. To understand this better we first initialize the outermost layer in occluded region using already known pixel intensity and texture values.
- But in the next go when we initialize deeper layers we use the previously calculated layers and known pixels to compute the current layers values.
- We take the coarser level image as input and this function will output an onion peel initialized image.

Creating pyramid levels

```
./src/pyramids.py
```

- We subsample the image to get images of lower dimension. Using lower dimension makes it easier for initializing the intensities, it also helps us to get out of suboptimal solutions.
- We pass the details of the lower dimension image to higher dimension image by upsampling the mapping we used in the lower dimension.
- This function will give l levels of images, which are sub sampled with a factor of $1/2$ and at each level blurred using gaussian
- We used coarse level images to construct higher dimensional finer level images.

Inpainting

```
./src/GUI/inpaintingGUI.py
```

- This is the file where we used the above defined functions to get the over-all desired image.
- We used onion peel initialization, pyramids, ANN patch match and reconstruction to reconstruct the occluded region as given in the pseudo code (which is provided in research paper).

GUI

```
./src/GUI/main.py
```

- Declared functions to maintain the functionality and display of the GUI.