NOVEMBER 28, 2018

PIXELS

INTERACTIVE WEB APPLICATION FOR IMAGE PROCESSING

LOKESH BALANI, ABHISHEK TYAGI
PGSSP, IIITH

TABLE OF CONTENTS

Contents

REPORT

Introduction

PIXELS is the project undertaken as a mandatory requirement for the course “Digital Image Processing”. The aim of the course is to provide introduction to the field of digital image processing, with the focus on fundamentals which would aid students in leveraging them for more advanced courses as well as in their research.

PUPRPOSE OF THE DOCUMENT

In this document there is an overall description of the Web Application, PIXELS. It includes what we did. There is also description of the work experience gain from the project.

INTENDED AUDIENCE

The intended audience for this document include professors/teachers teaching image processing, students learning image processing, project members, the teaching assistants, the users willing to leverage the application and the students/professionals willing to contribute to the project.

PROBLEM STATEMENT

To create a Web-based interactive tool for demonstrating image processing techniques in real time to images provided by the user. The application should assist Professor in classroom and should also be modular and easily extensible.

PROJECT PIVOT

The objectives of this project are:

1. The Web Application should be interactive and to be able to be used as a companion for lectures
2. The Web Application should be modular and easily extendible
3. The Web Application should have minimalistic design
4. The Web Application should be easy to use
5. The Web Application should be able to accept user images as input and generate real time output

DELIVERABLES

The deliverables to be produced for this project include:

1. A web application
2. Project presentation
3. Github repository
4. Final project report

REPORT

GITHUB REPOSITORY

https://github.com/lokeshbalani/pixels

Tool Description

USER INTERFACE

The Web Application is very user friendly and intuitive and uses GUI Interface implemented in HTML, CSS & JavaScript on the Client Side and Django, Python on the Server Side.

Various features are self-explanatory and the modules can be easily added, updated and removed.

FEATURES

- Intuitive Interface
- Minimalistic Design
- Interactive GUI
- Support for Image Drag-n-Drop
- Support for Image Upload from Local Computer
- Downloadable Images
- Accessible Image and Module Info
- Required Configurable Option Settings for the Processing Techniques
- Modular with ease-of-extension
- Clean separation of various modules to facilitate easy modification and revision
- Python Library(s) support
- Segmentation into **Lecture Companion** and **Do It Yourself** Interface

ARCHITECTURE

- Django provides the basic platform. It starts a local web server, defines the URL patterns, binds the model and the view, and provides backend integration with python libraries
- The Image Processing Modules are written in Python using the libraries like OpenCV, etc. wherever necessary.
- The webpages have been created using the Django Templating Engine
- The Front End Code/Page View, Interface and Interaction has been coded using HTML/CSS/JavaScript/jQuery.

RUNNING THE WEBAPP (WITHOUT VIRTUALENV)

```
$git clone https://github.com/lokeshbalani/pixels.git
$cd pixels
$pip install -r requirements.txt
$python webapp/manage.py makemigrations
$python webapp/manage.py migrate
$sudo python webapp/manage.py runserver
```

RUNNING THE WEBAPP (ON VIRTUALENV)

```
$git clone https://github.com/lokeshbalani/pixels.git
$cd pixels
$virtualenv py_env
$source py_env/bin/activate
$pip install -r requirements.txt
$python webapp/manage.py makemigrations
$python webapp/manage.py migrate
$sudo python webapp/manage.py runserver
```

When done deactivate/terminate the virtual environment

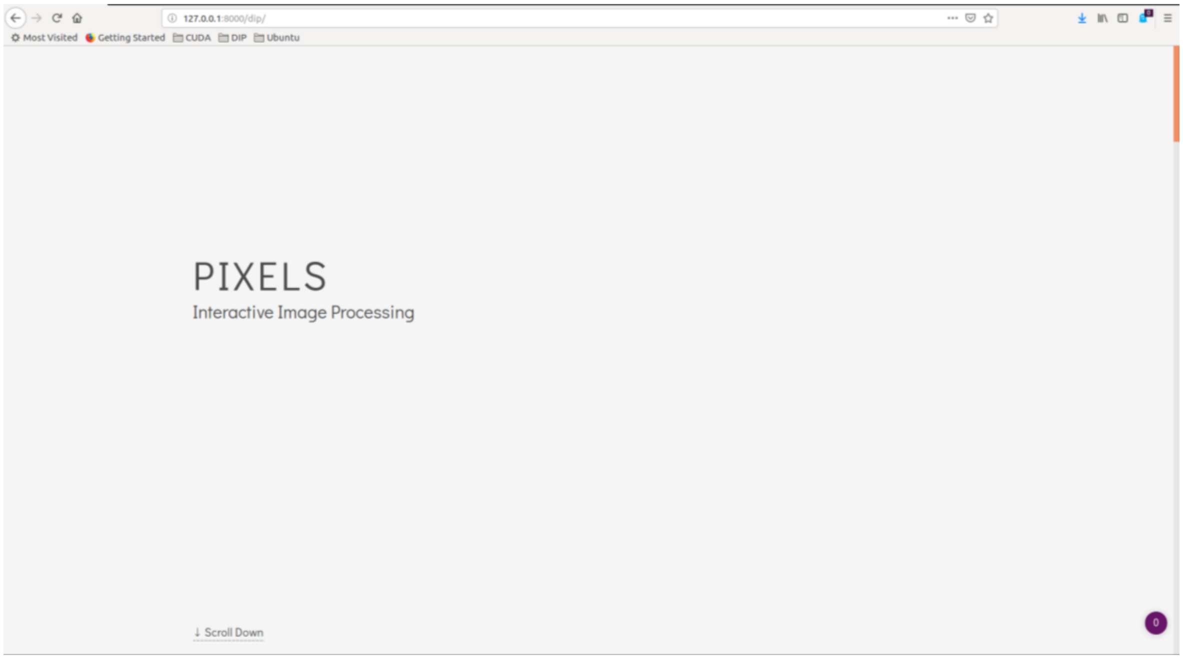
```
$deactivate
```

To install and setup python virtual environment refer to the below URL:
https://docs.python-guide.org/dev/virtualenvs/

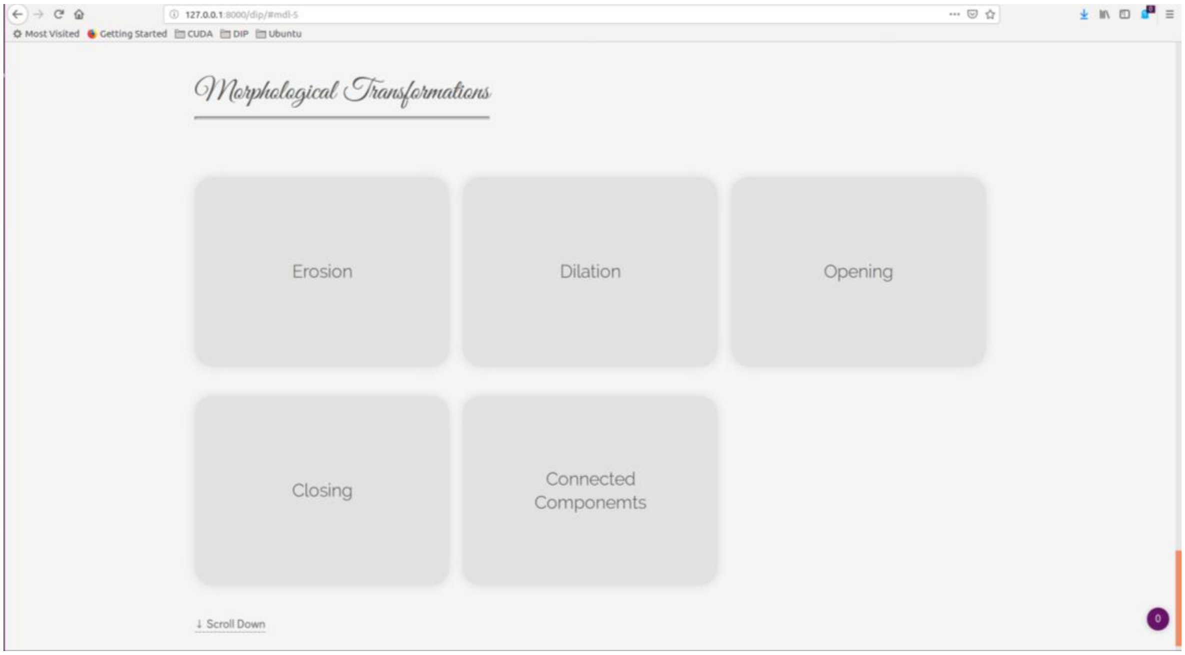
GRAPHICAL USER INTERFACE

The Web Application will start on the following IP Address: http://127.0.0.1:8000

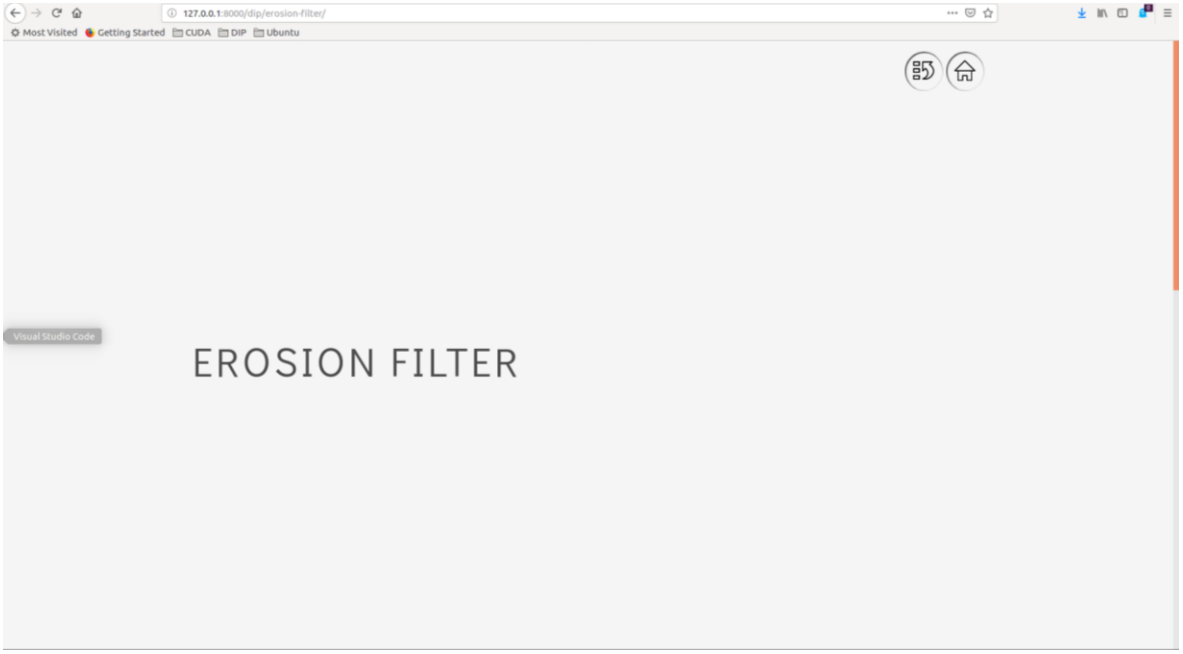
Web Application navigates to the main Landing Page of the application (see screenshots below)



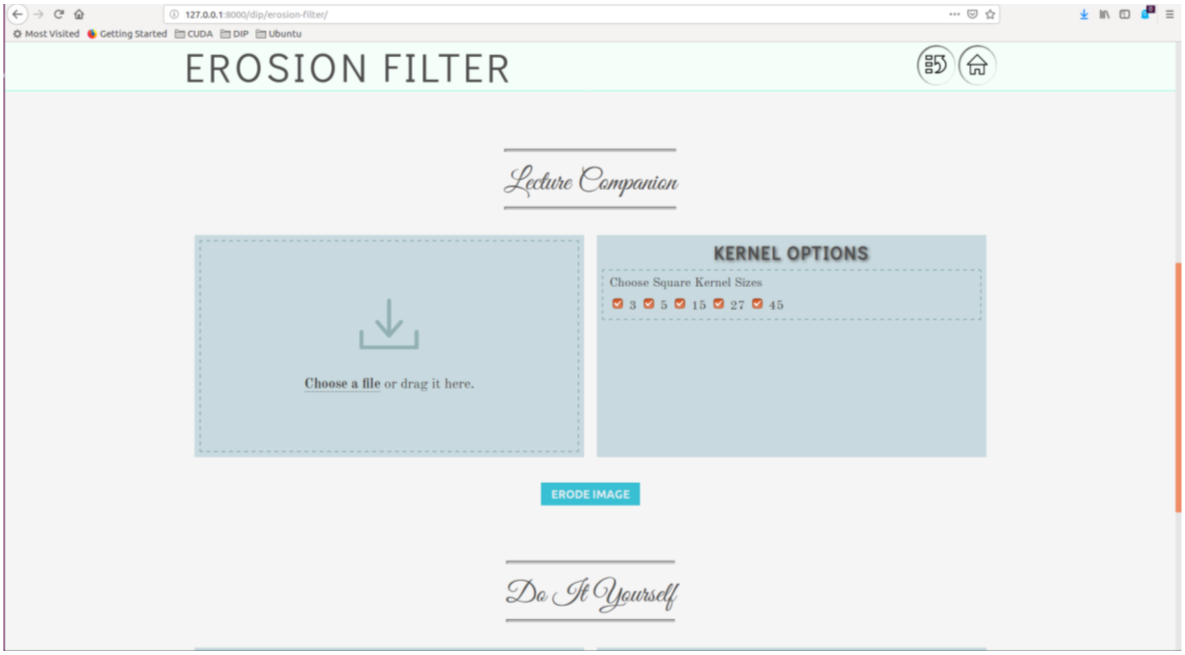
On Scrolling it displays the Modules implemented by the broad categorization (see below)



Clicking on the Individual Modules takes us to that Modules Page (see below)



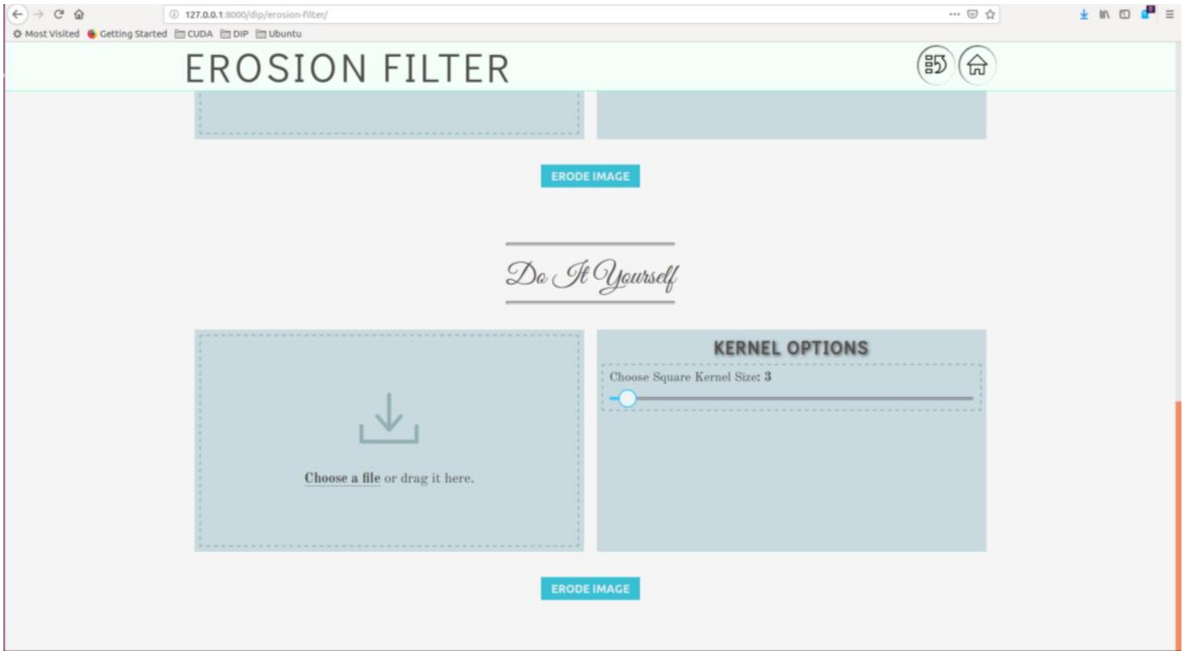
The Modules have two sections : Lecture Companion (for classroom demonstrations) and Do It Yourself



KERNEL OPTIONS

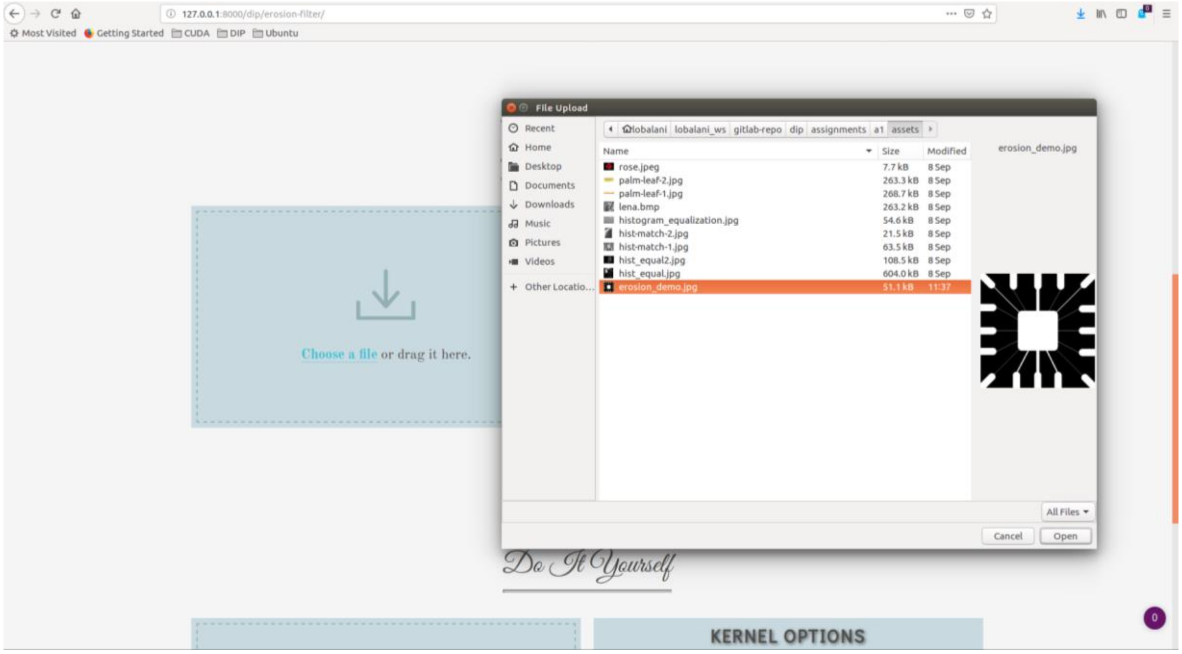
Choose Square Kernel Sizes

☑ 3 ☑ 5 ☑ 15 ☑ 27 ☑ 45



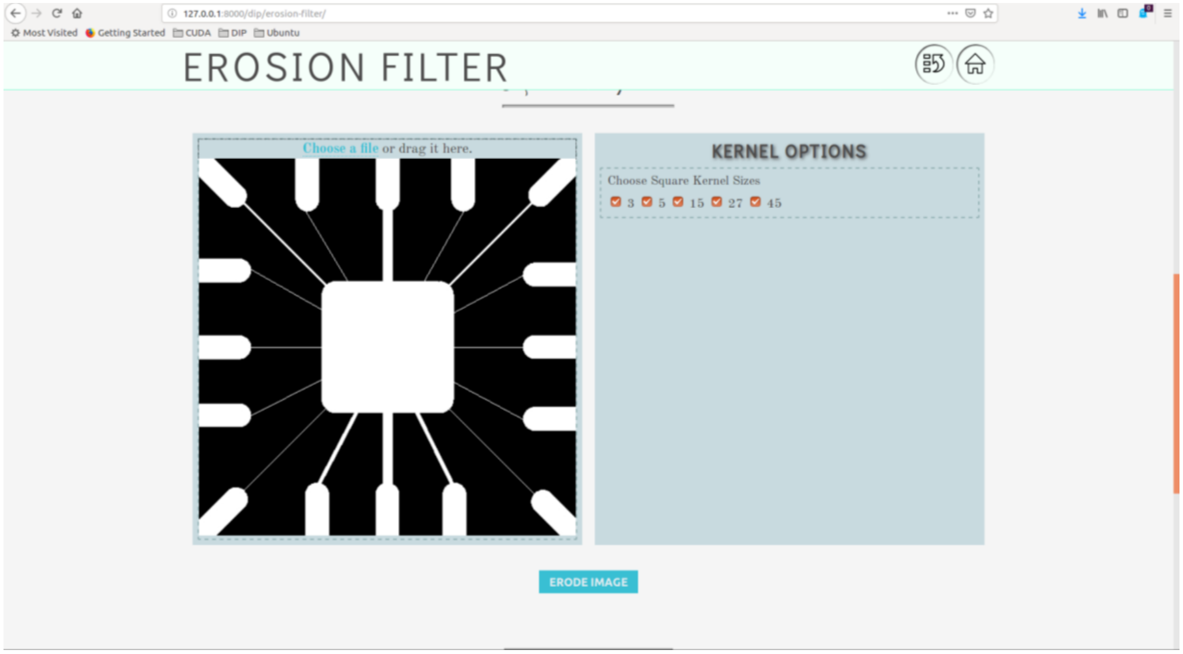
The right section shown above contains the parameters that can be set by user for the respective technique.

Click on **Choose File** option to select an Image from Local Computer, or Drag-n-Drop Image to the upload section on the left (see below)



Name	Size	Modified
rose.jpeg	7.7 kB	8 Sep
palm-leaf-2.jpg	263.3 kB	8 Sep
palm-leaf-1.jpg	268.7 kB	8 Sep
lena.bmp	263.2 kB	8 Sep
histogram_equalization.jpg	54.6 kB	8 Sep
hist-match-2.jpg	21.5 kB	8 Sep
hist-match-1.jpg	63.5 kB	8 Sep
hist_equal2.jpg	108.5 kB	8 Sep
hist_equal.jpg	604.0 kB	8 Sep
erosion_demo.jpg	51.1 kB	11:37

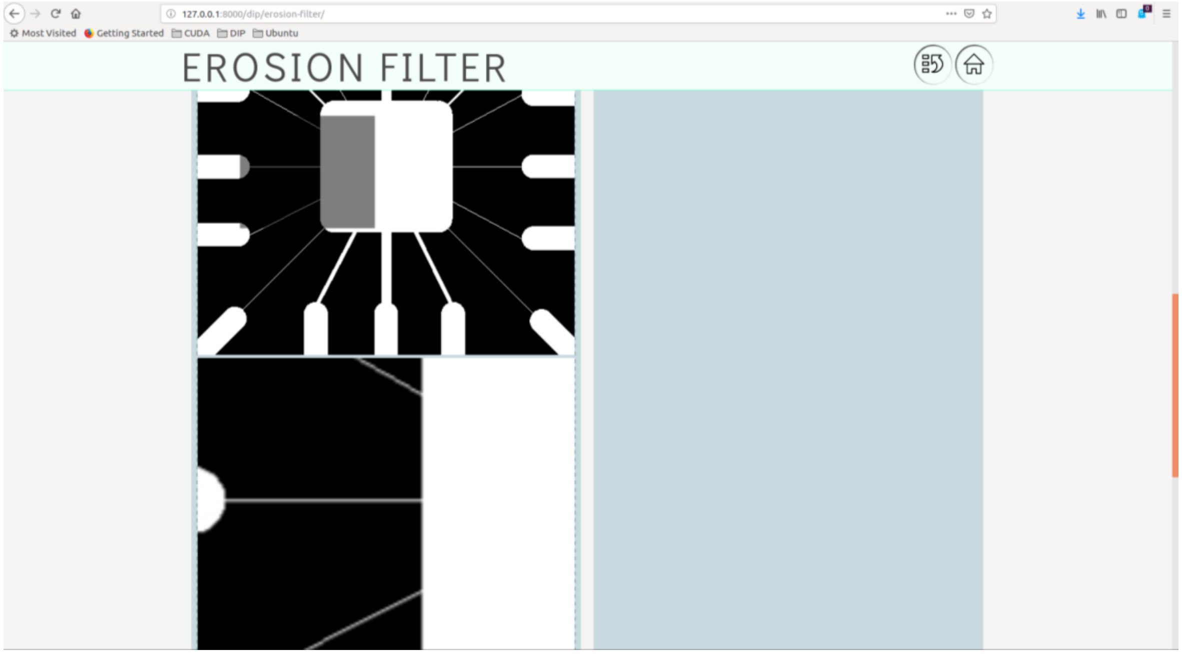
On selecting an input image, the preview of the image is rendered. The rendered image has the option to select a window to zoom to a particular part of image as shown below:



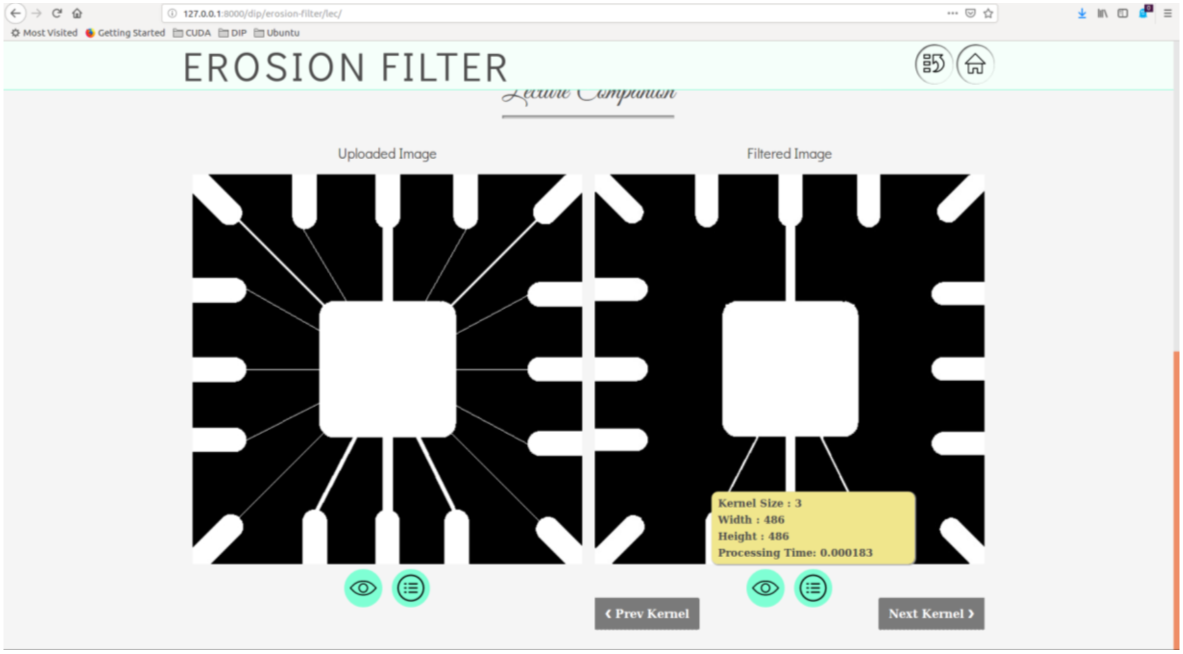
KERNEL OPTIONS

Choose Square Kernel Sizes

☑ 3 ☑ 5 ☑ 15 ☑ 27 ☑ 45



Clicking on the ERODE IMAGE button generates the output eroded image along with the image and processing info as shown below



EROSION FILTER

Uploaded Image

Filtered Image

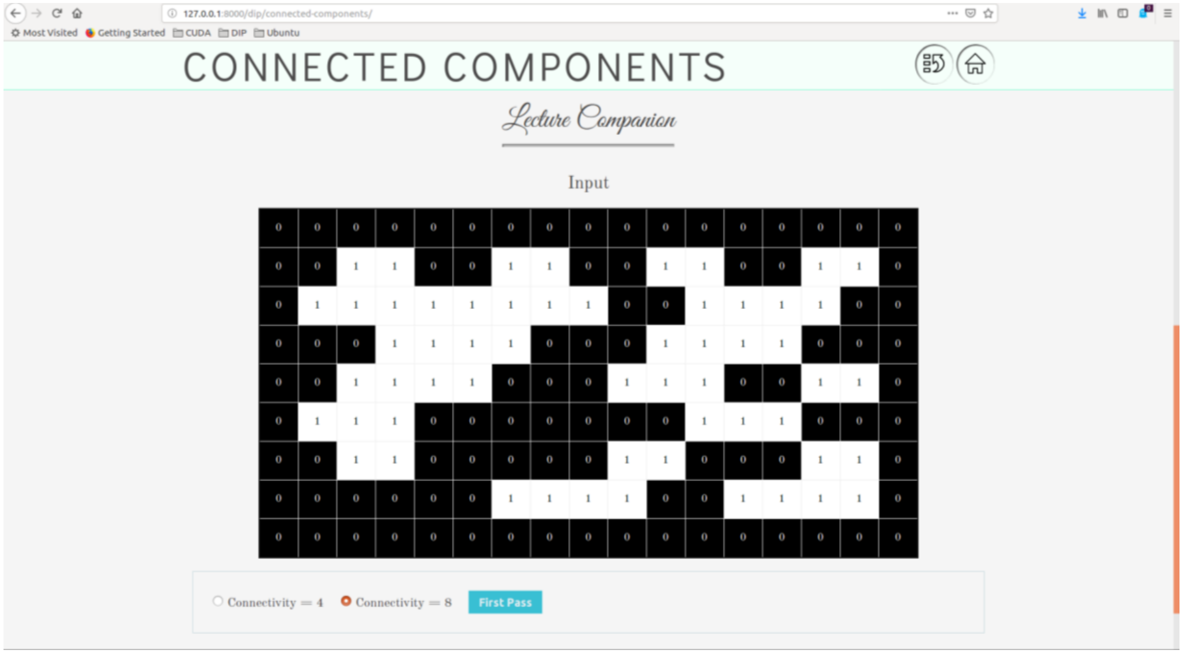
Kernel Size : 3
Width : 486
Height : 486
Processing Time: 0.000183

By Clicking on the preview Icon the Image can be downloaded.

Clicking on the Next Kernel or Previous Kernel Option generates the Image Output for the next kernel defined in the settings list.

Select Zoom functionality has been implemented for the Output Images as well.

The Modules can also be implemented so that the output can be shown in steps as it has been implemented for **Connected Components** (as shown below)



CONNECTED COMPONENTS

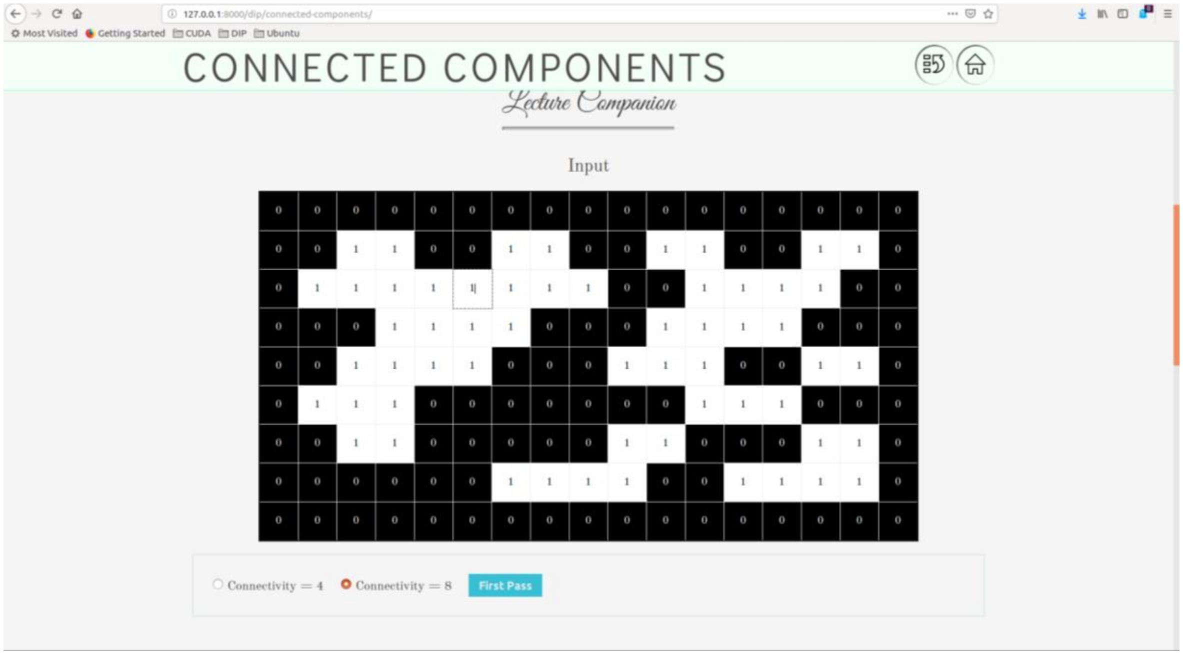
Lecture Companion

Input

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0
0	1	1	1	1	1	1	1	1	0	0	1	1	1	1	0	0
0	0	0	1	1	1	1	0	0	0	1	1	1	1	0	0	0
0	0	1	1	1	1	0	0	0	1	1	1	0	0	1	1	0
0	1	1	1	0	0	0	0	0	0	0	1	1	1	0	0	0
0	0	1	1	0	0	0	0	0	1	1	0	0	0	1	1	0
0	0	0	0	0	0	1	1	1	1	0	0	1	1	1	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

○ Connectivity = 4 ● Connectivity = 8 First Pass

The Cells in the Input matrix are editable (see below)



CONNECTED COMPONENTS

Lecture Companion

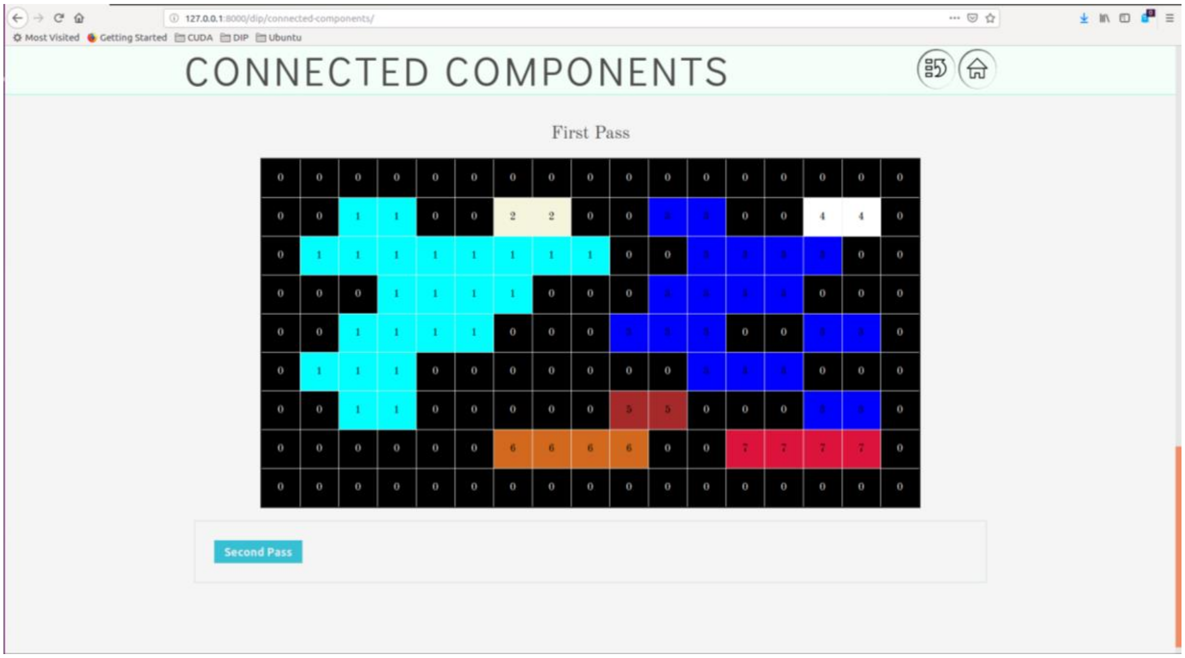
Input

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0
0	1	1	1	1	1	1	1	1	0	0	1	1	1	1	0	0
0	0	0	1	1	1	1	0	0	0	1	1	1	1	0	0	0
0	0	1	1	1	1	0	0	0	1	1	1	0	0	1	1	0
0	1	1	1	0	0	0	0	0	0	0	1	1	1	0	0	0
0	0	1	1	0	0	0	0	0	1	1	0	0	0	1	1	0
0	0	0	0	0	0	1	1	1	1	0	0	1	1	1	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

○ Connectivity = 4 ● Connectivity = 8 First Pass

User can choose between Connectivity 8 or 4 and then clicking on First Pass generates the First Pass (see below)

REPORT

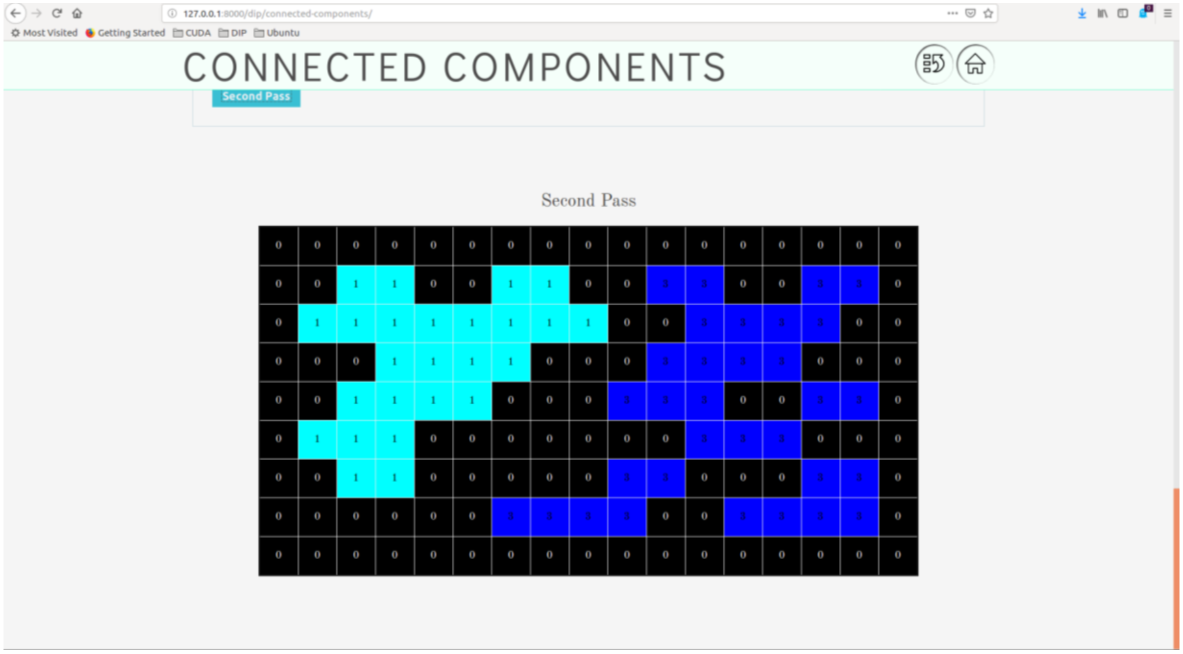


CONNECTED COMPONENTS

First Pass

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	2	2	0	0	3	3	0	0	4	4	0
0	1	1	1	1	1	1	1	1	0	0	3	3	3	3	0	0
0	0	0	1	1	1	1	0	0	0	3	3	3	3	0	0	0
0	0	1	1	1	1	0	0	0	3	3	3	0	0	3	3	0
0	1	1	1	0	0	0	0	0	0	0	3	3	3	0	0	0
0	0	1	1	0	0	0	0	0	5	5	0	0	0	3	3	0
0	0	0	0	0	0	6	6	6	6	0	0	7	7	7	7	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Clicking on the Second Pass Button Generates the Second Pass (see below)



CONNECTED COMPONENTS

Second Pass

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	1	1	0	0	3	3	0	0	3	3	0
0	1	1	1	1	1	1	1	1	0	0	3	3	3	3	0	0
0	0	0	1	1	1	1	0	0	0	3	3	3	3	0	0	0
0	0	1	1	1	1	0	0	0	3	3	3	0	0	3	3	0
0	1	1	1	0	0	0	0	0	0	0	3	3	3	0	0	0
0	0	1	1	0	0	0	0	0	3	3	0	0	0	3	3	0
0	0	0	0	0	0	3	3	3	3	0	0	3	3	3	3	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

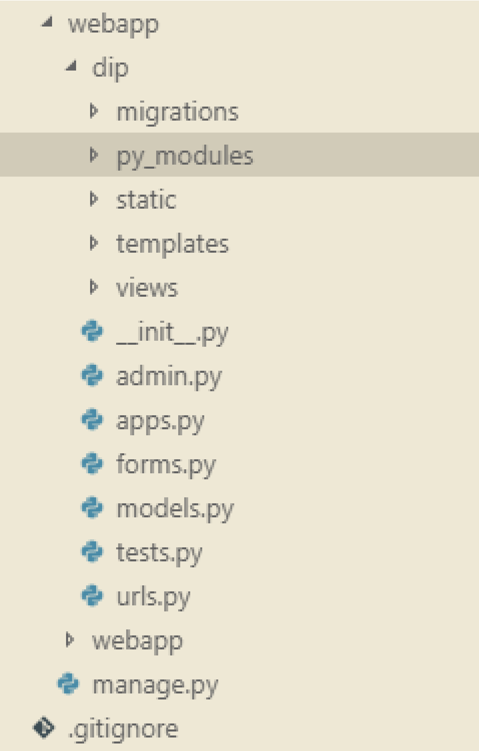
IMAGE PROCESSING TECHNIQUES INVENTORY

- Histograms
  - Histogram Plot
  - Histogram Equalization
- Image Thresholding
  - Global Thresholding
  - Adaptive Thresholding
  - Otsu’s Binarization
- Sharpening Spatial Filters
  - Sobel Derivatives
  - Laplacian Derivatives
- Smoothing Spatial Filters
  - Averaging Filter
  - Gaussian Filter
  - Median Filter
  - Bilateral Filter
- Morphological Transformations
  - Connected Components
  - Erosion
  - Dilation
  - Opening
  - Closing

Application Architecture and Development

FOLDER STRUCTURE

The folder structure for the project is as follows:



```
◢ webapp
  ◢ dip
    ▷ migrations
    ▷ py_modules
    ▷ static
    ▷ templates
    ▷ views
      __init__.py
      admin.py
      apps.py
      forms.py
      models.py
      tests.py
      urls.py
  ▷ webapp
    manage.py
  .gitignore
```

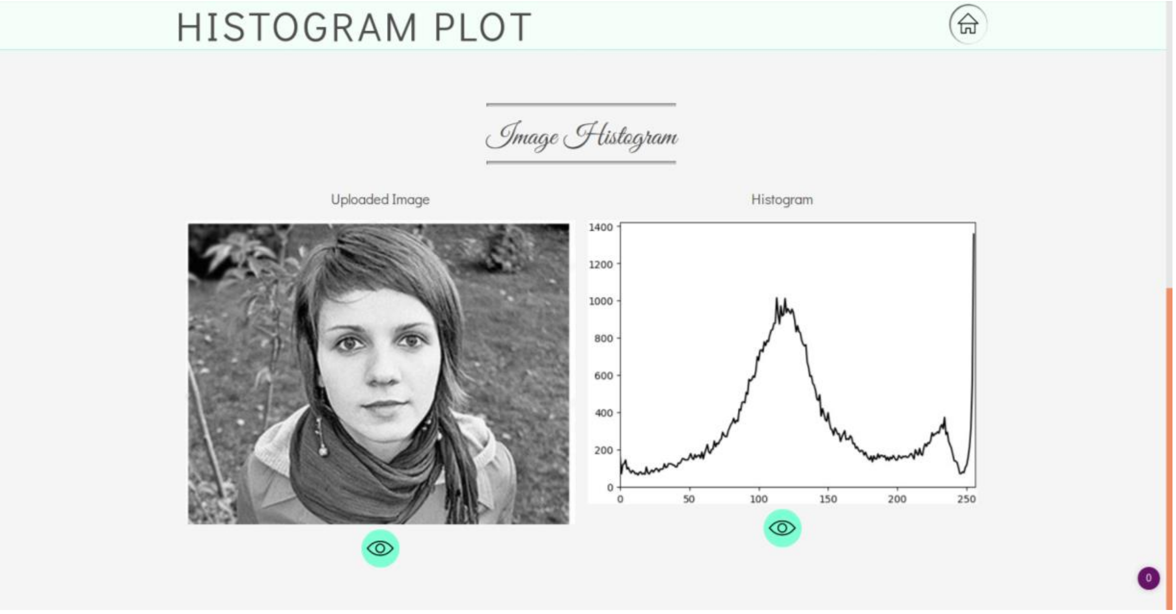
- **/webapp/dip/py_modules** : Contains the Python Implementation of Image Processing Modules
- **/webapp/dip/static**: Contains the static assets like CSS, JS for the website. It is further subdivided in to **css/** and **js/** and **fonts/** folders
- **/webapp/dip/templates**: Contains the Web Page Django Templates for the Image Processing Modules
- **/webapp/dip/views**: Contains the view controller logic python scripts for the Image Processing Modules
- **Urls.py :** This file contains the URL settings and view bindings for the Modules

Above mentioned are the folders and files that the developer needs to develop for if the application needs to be extended.

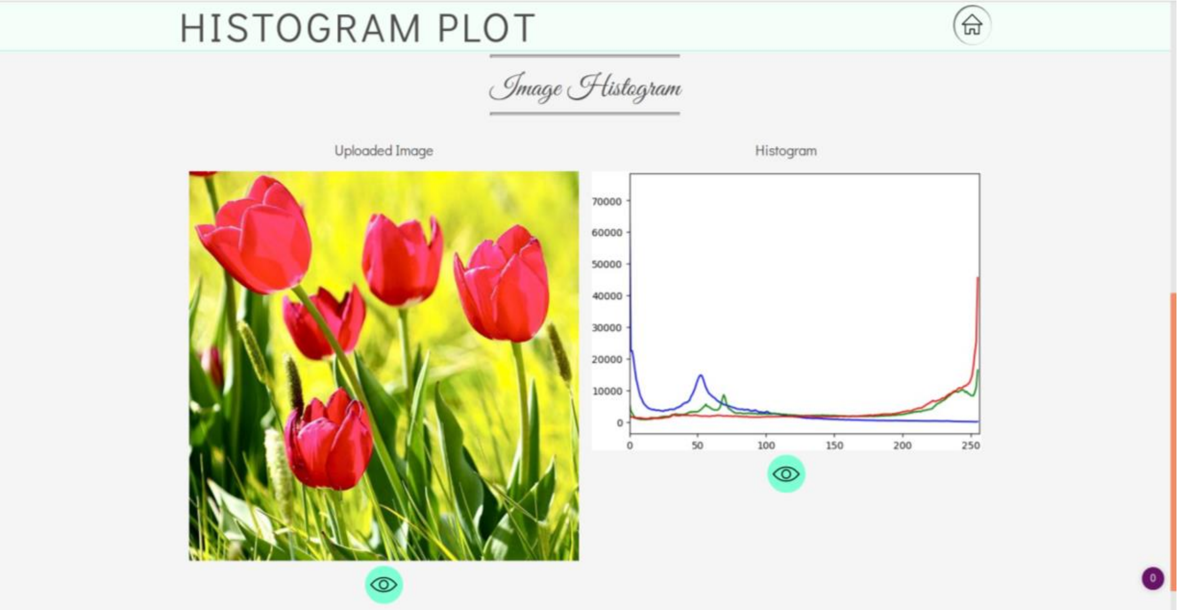
Results

HISTOGRAM PLOT

GreyScale

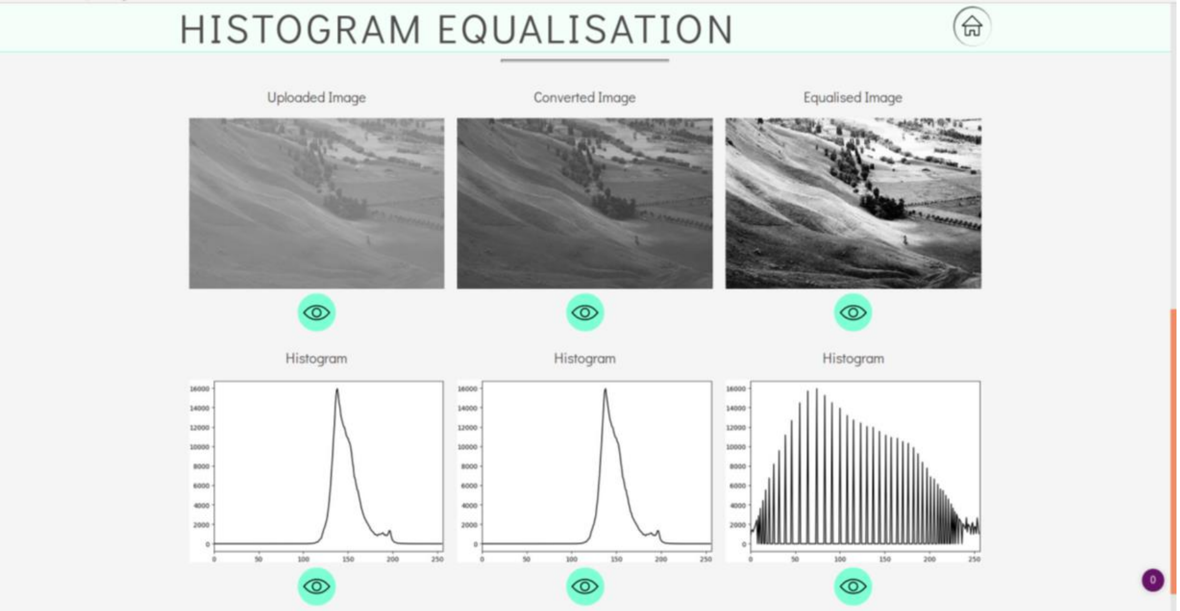


Color

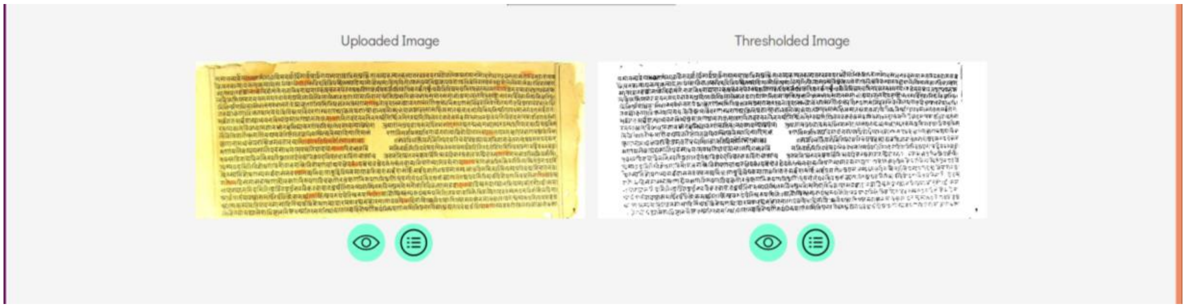


REPORT

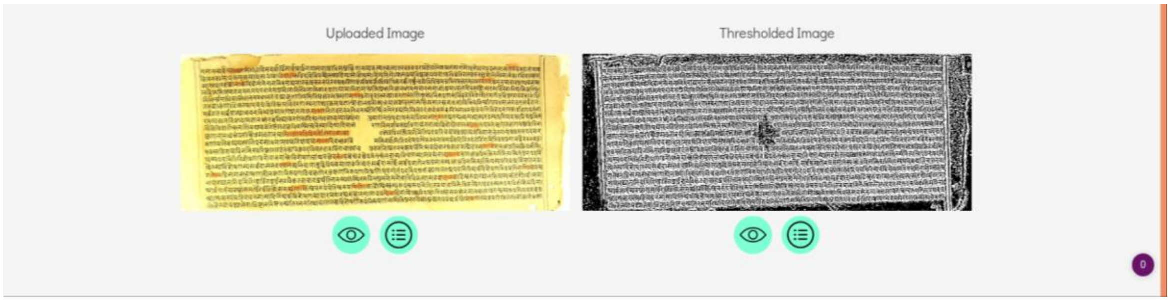
HISTOGRAM EQUALISATION



GLOBAL THRESHOLDING

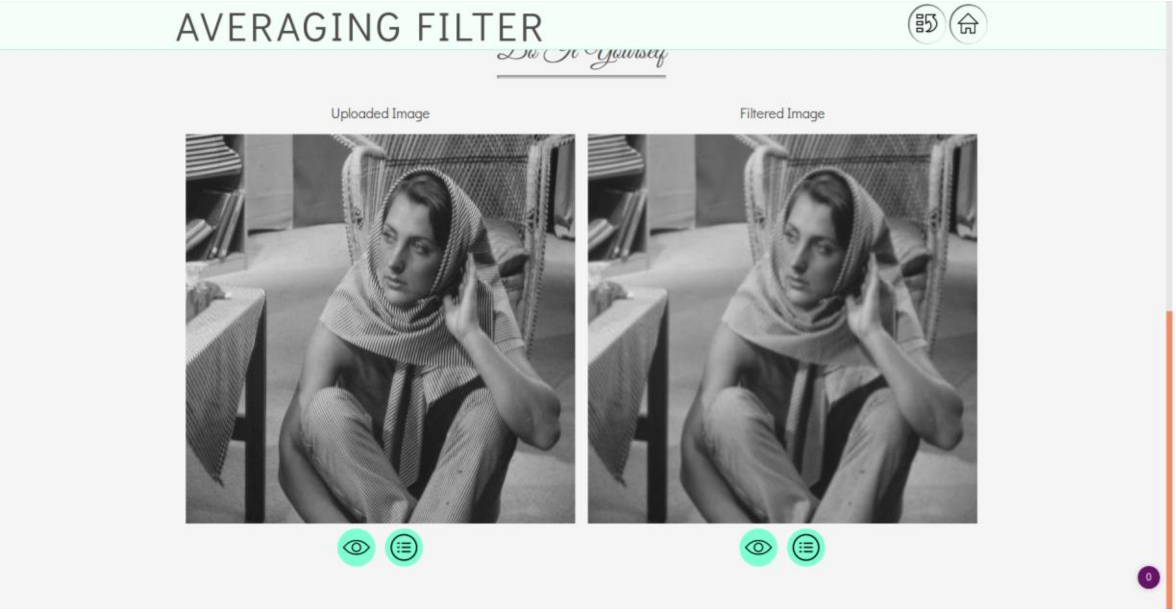


ADAPTIVE THRESHOLDING

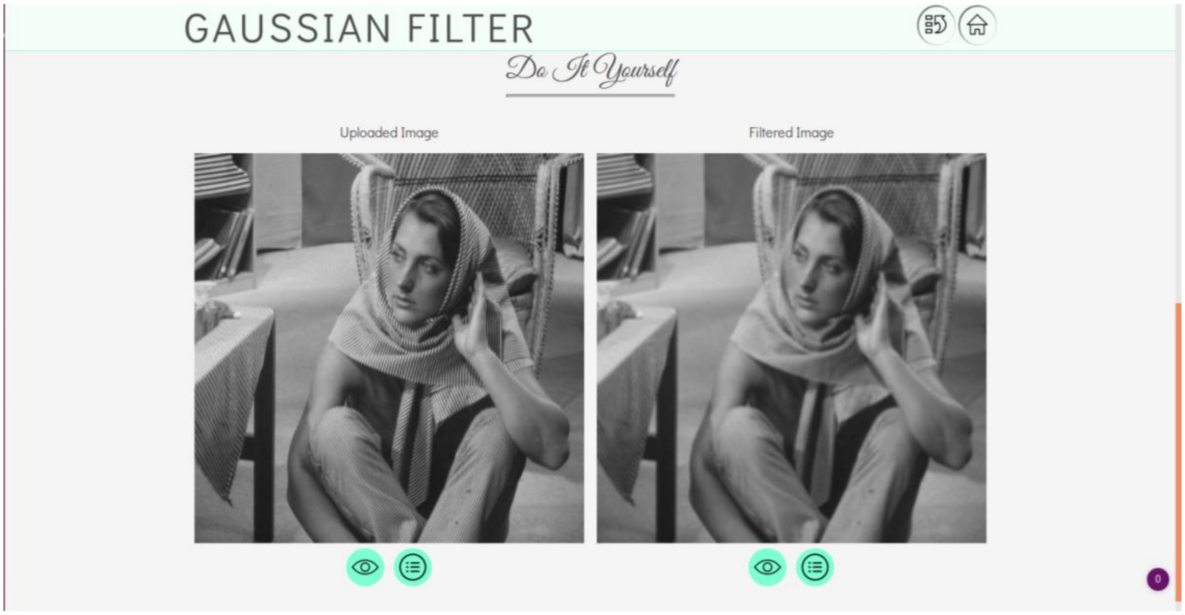


REPORT

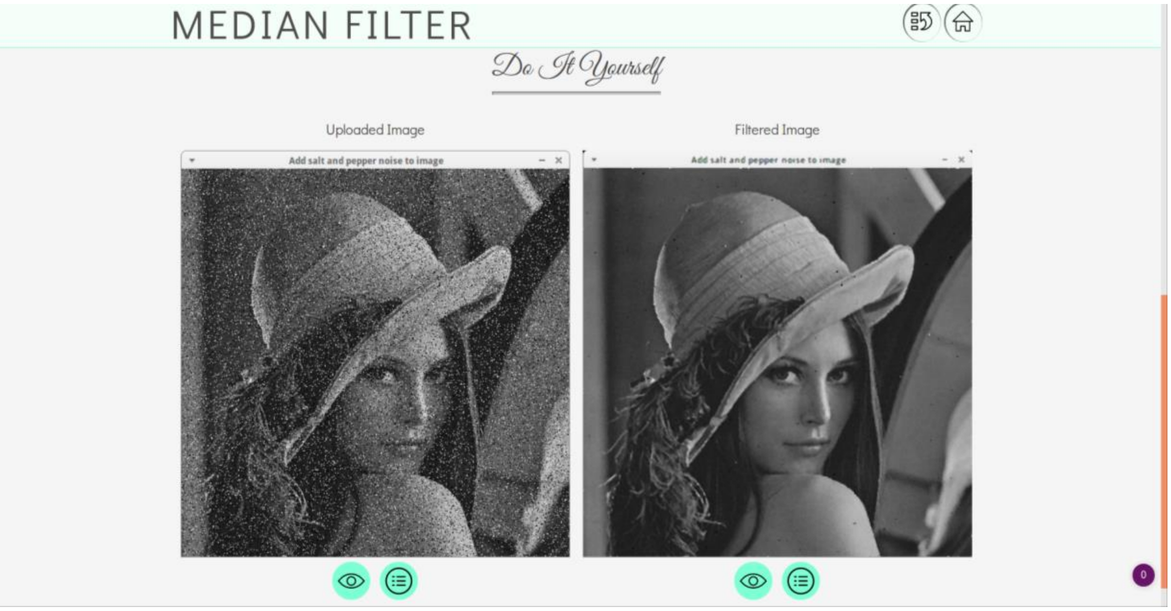
AVERAGE FILTER



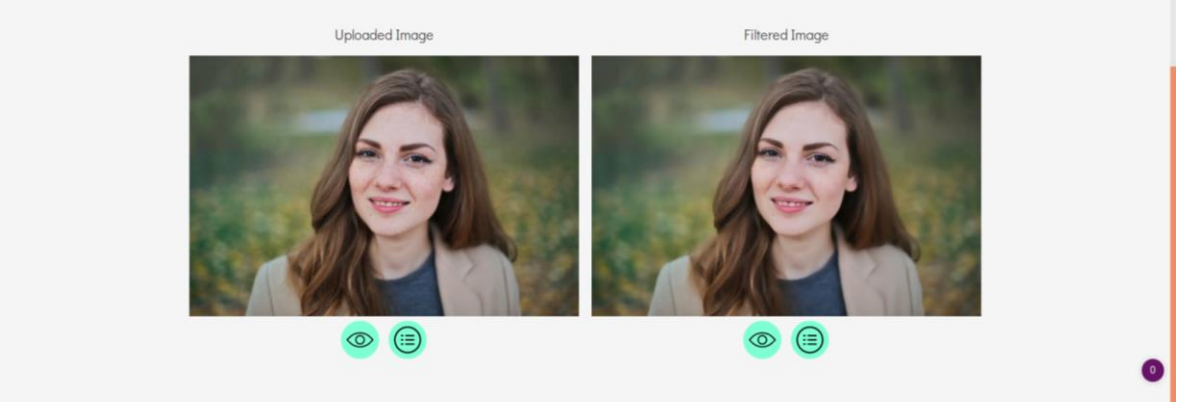
GAUSSIAN FILTER



MEDIAN FILTER

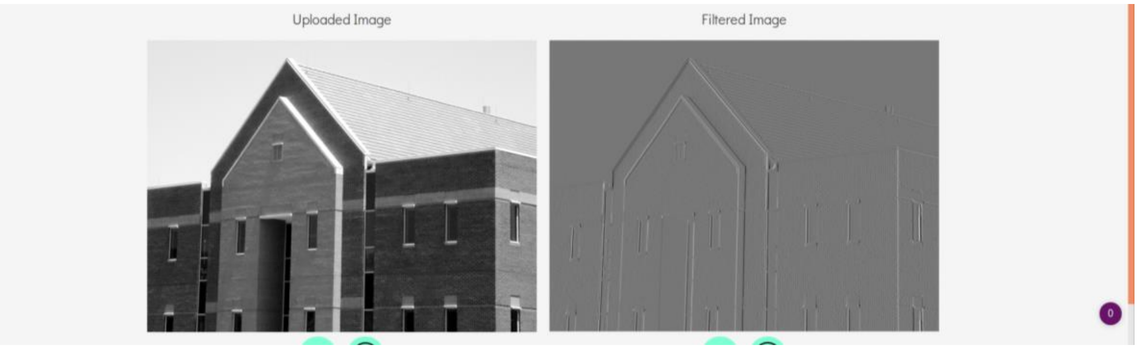

MEDIAN FILTER
Do It Yourself
Uploaded Image
Filtered Image
Add salt and pepper noise to image
Add salt and pepper noise to image
0

BILATERAL FILTER

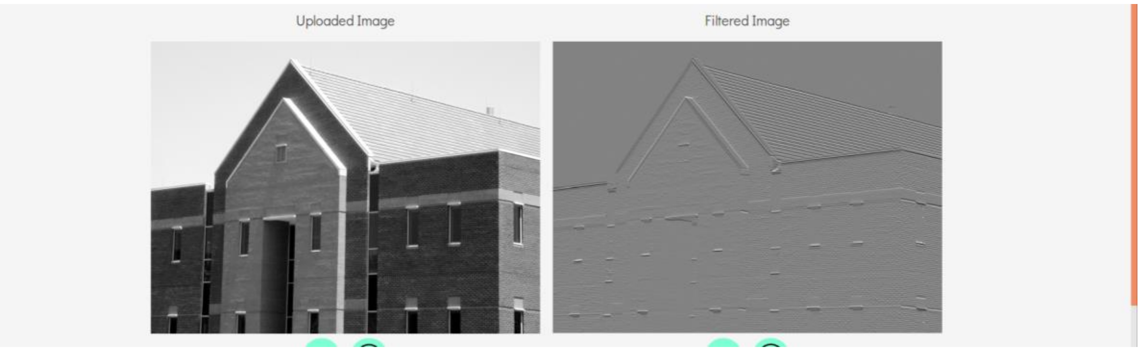


SOBEL FILTER

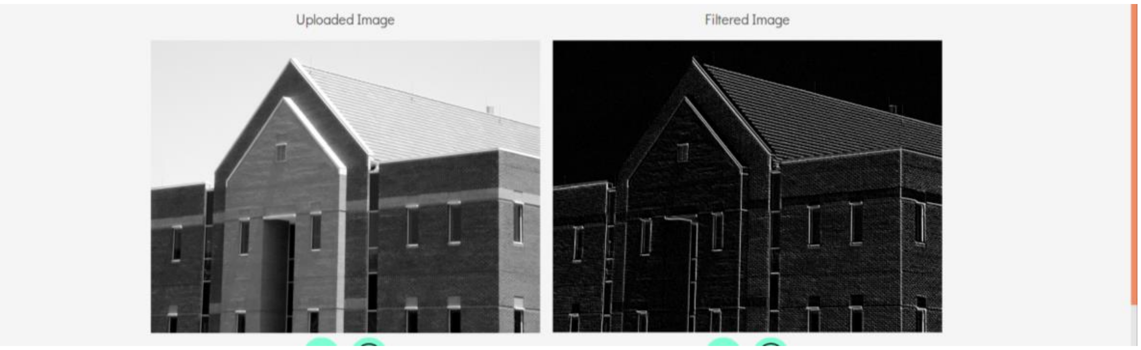
0 Degrees



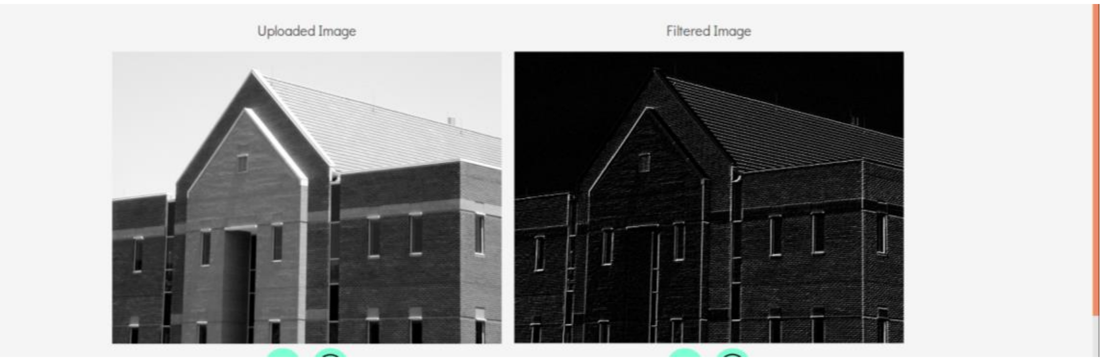
+90 Degree



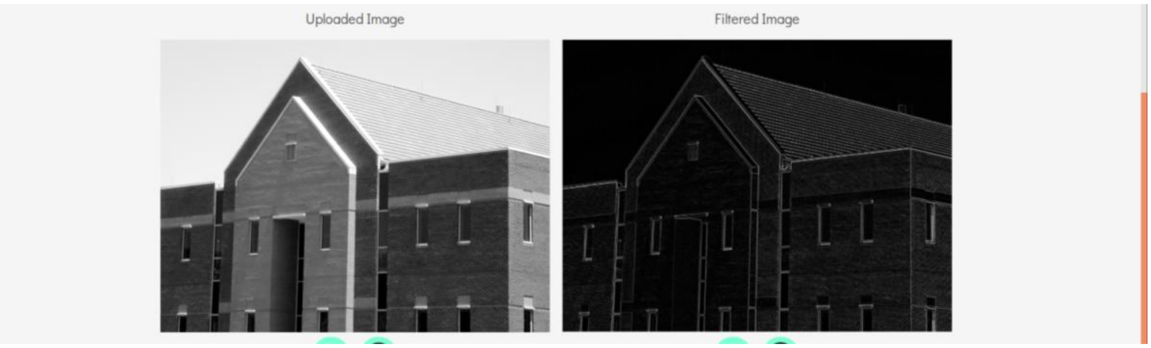
-45 Degree



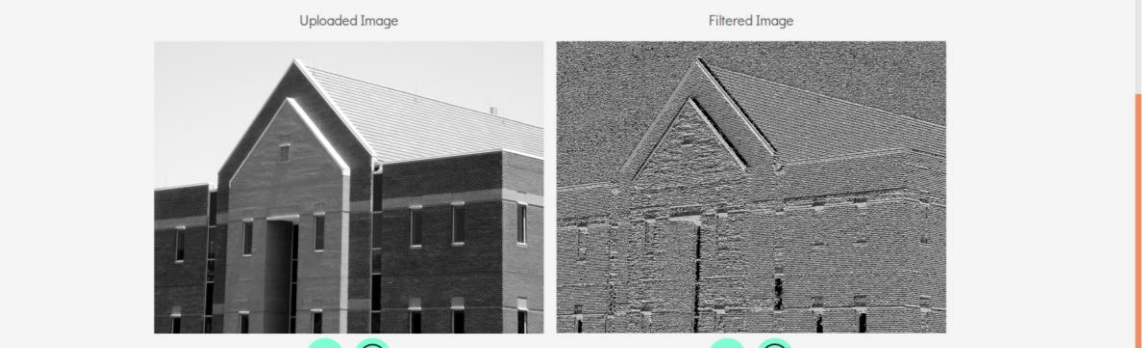
+45 Degree



Magnitude



Angle



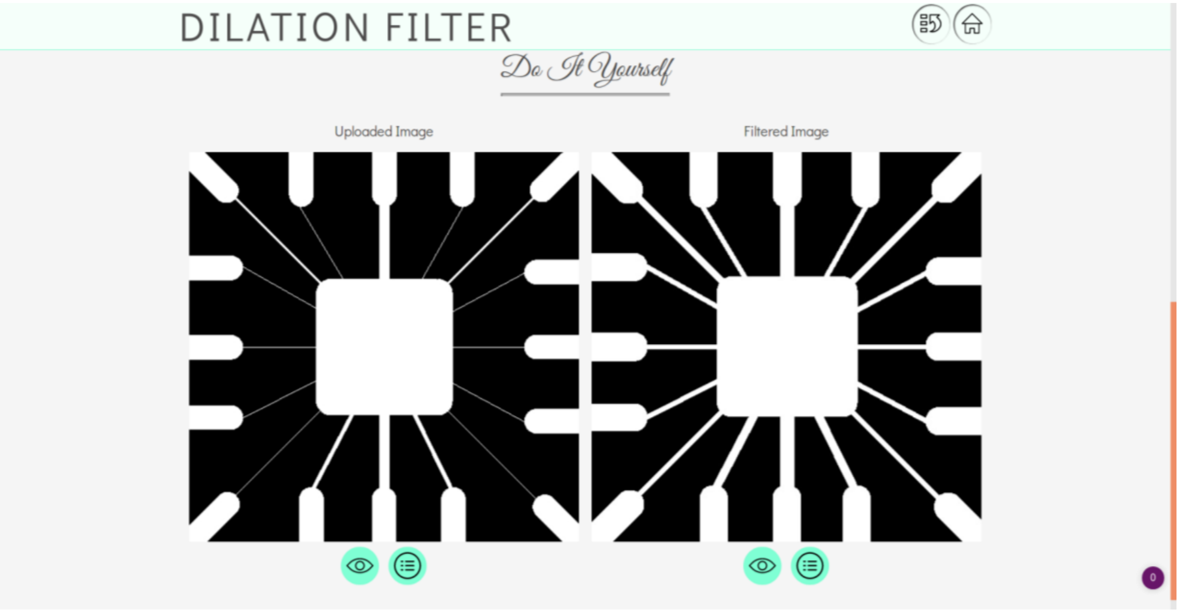
CONNECTED COMPONENTS

See before

EROSION

See before

DILATION


DILATION FILTER
Do It Yourself
Uploaded Image
Filtered Image
0

Team Members

LOKESH BALANI 20173073 PGSSP, IIITH	ABHISHEK TYAGI 20173067 PGSSP, IIITH

Project Milestones and Contribution

PROJECT MILESTONES

- Basic Django Platform Setup
- UX & Design for the Web Application
- Developing Front End (CSS/JS)
- Image Drag & Drop, Select Zoom Support
- Back End Support for Python Libraries
- Implementing Image Processing Techniques
- Support for Configurable Options
- Division of View
    - Instructors' Lecture Companion
    - DIY (Do It Yourself)

CONTRIBUTION - LOKESH

- Basic Django Platform Setup
- UX & Design for the Web Application
- Developing Front End (CSS/JS)
- Image Drag & Drop, Select Zoom Support
- Support for Configurable Options
- Implementing Image Processing Techniques
    - Histogram
- Connected Components, First Pass & Second Pass with Union Find - JavaScript
- Division of View
    - Instructors' Lecture Companion
    - DIY (Do It Yourself)

REPORT

CONTRIBUTION - ABHISHEK

- UX & Design for the Web Application
- Back End Support for Python Libraries
- Implementing Image Processing Techniques
    - Morphological
    - Smoothing Spatial Filters
    - Sharpening Spatial Filters (Laplacian & Sobel)
    - Image Thresholding
    - Connected Components, First Pass & Second Pass with Union Find - Python