

The Superior University

Name: Fatima Nadeem	Roll No:076	Course: PAI Lab
Semester:4th	Section:4B	Department:SE
Submitted To: Sir Rasikh	Total Marks: 10	Date:

Lab Task 05

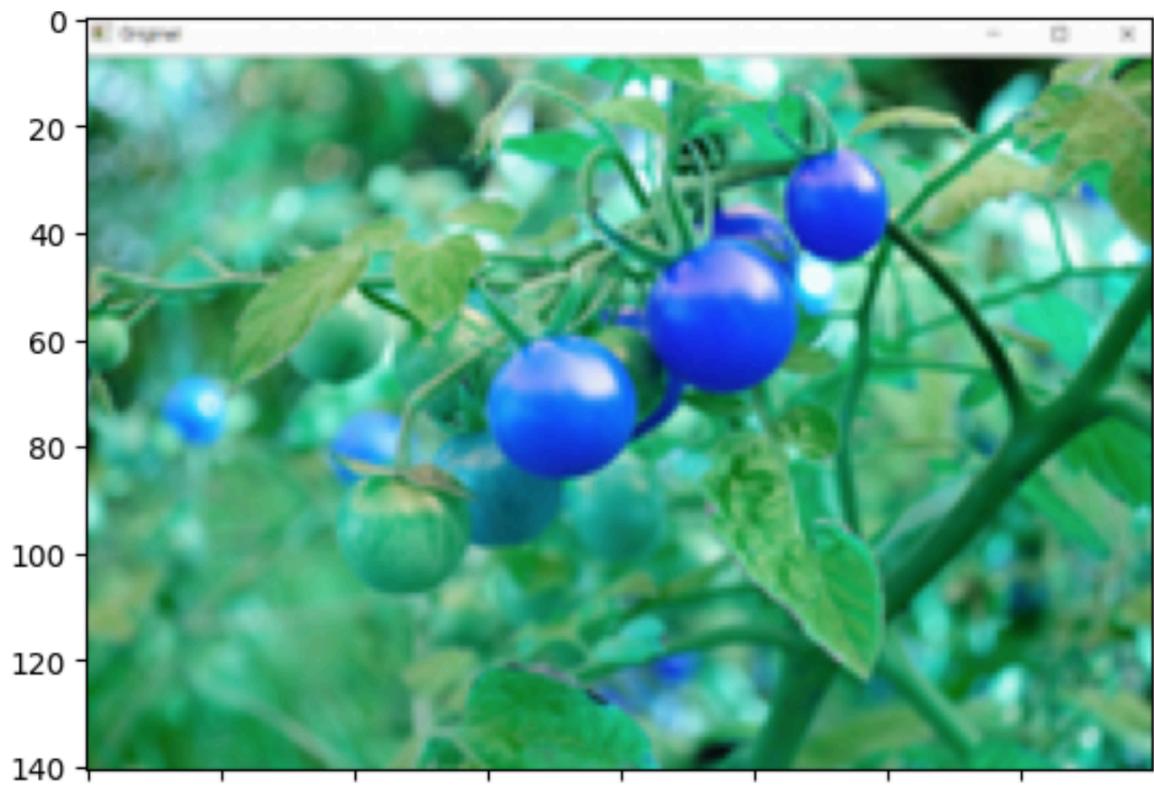
OPEN-CV

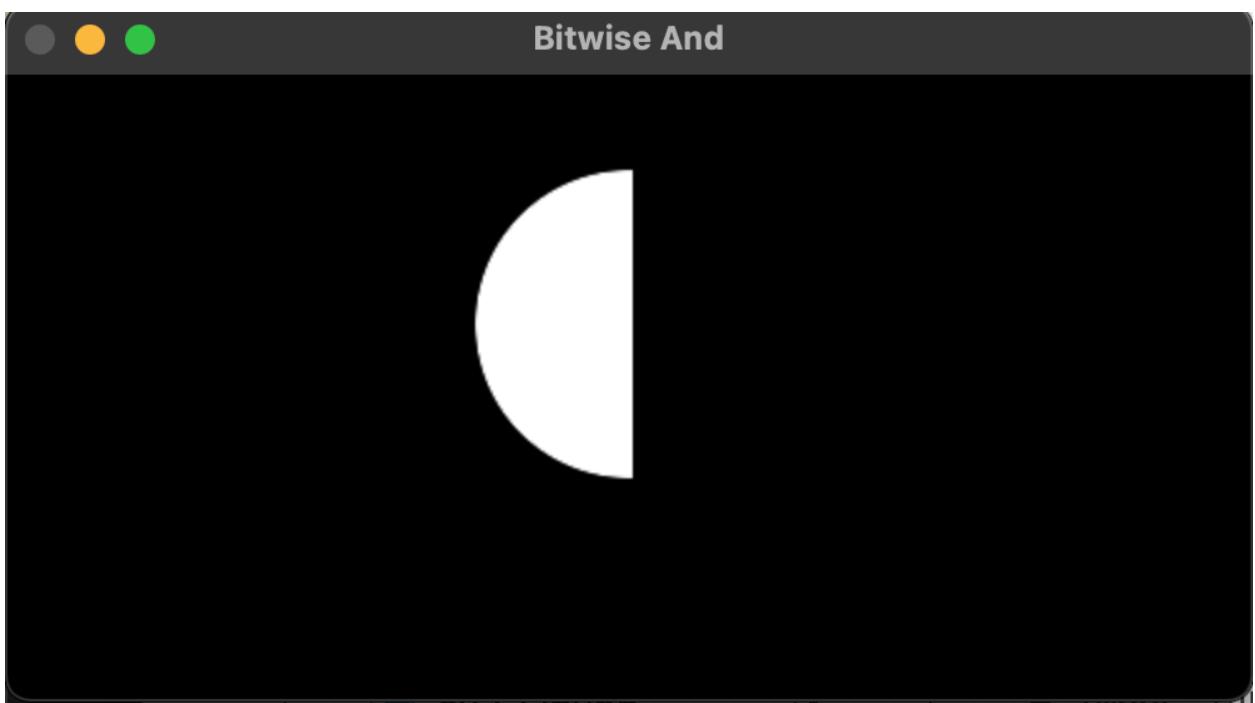
following are the open-cv tasks:

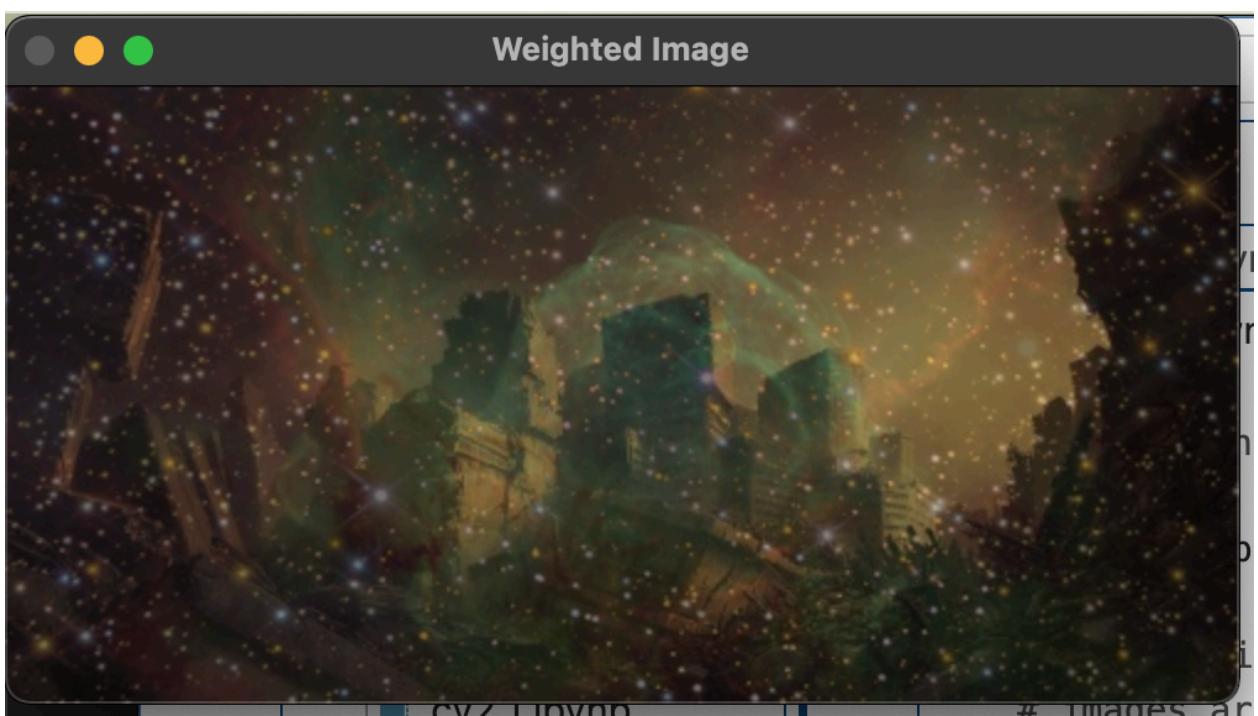
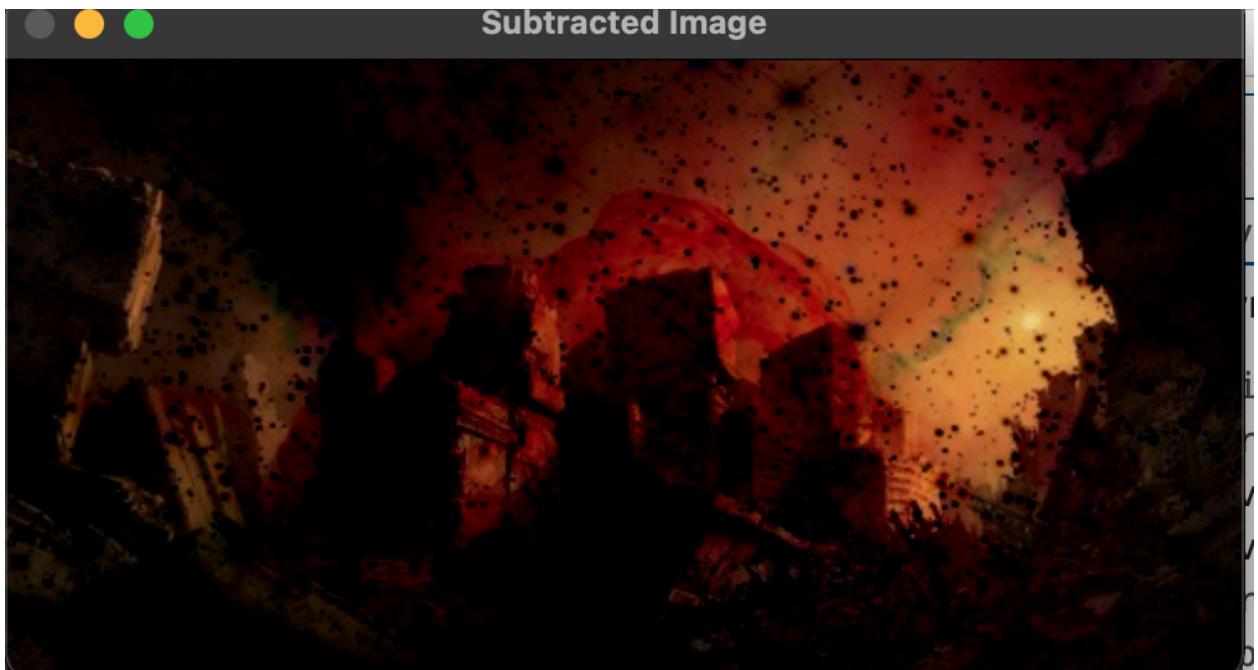
2.1

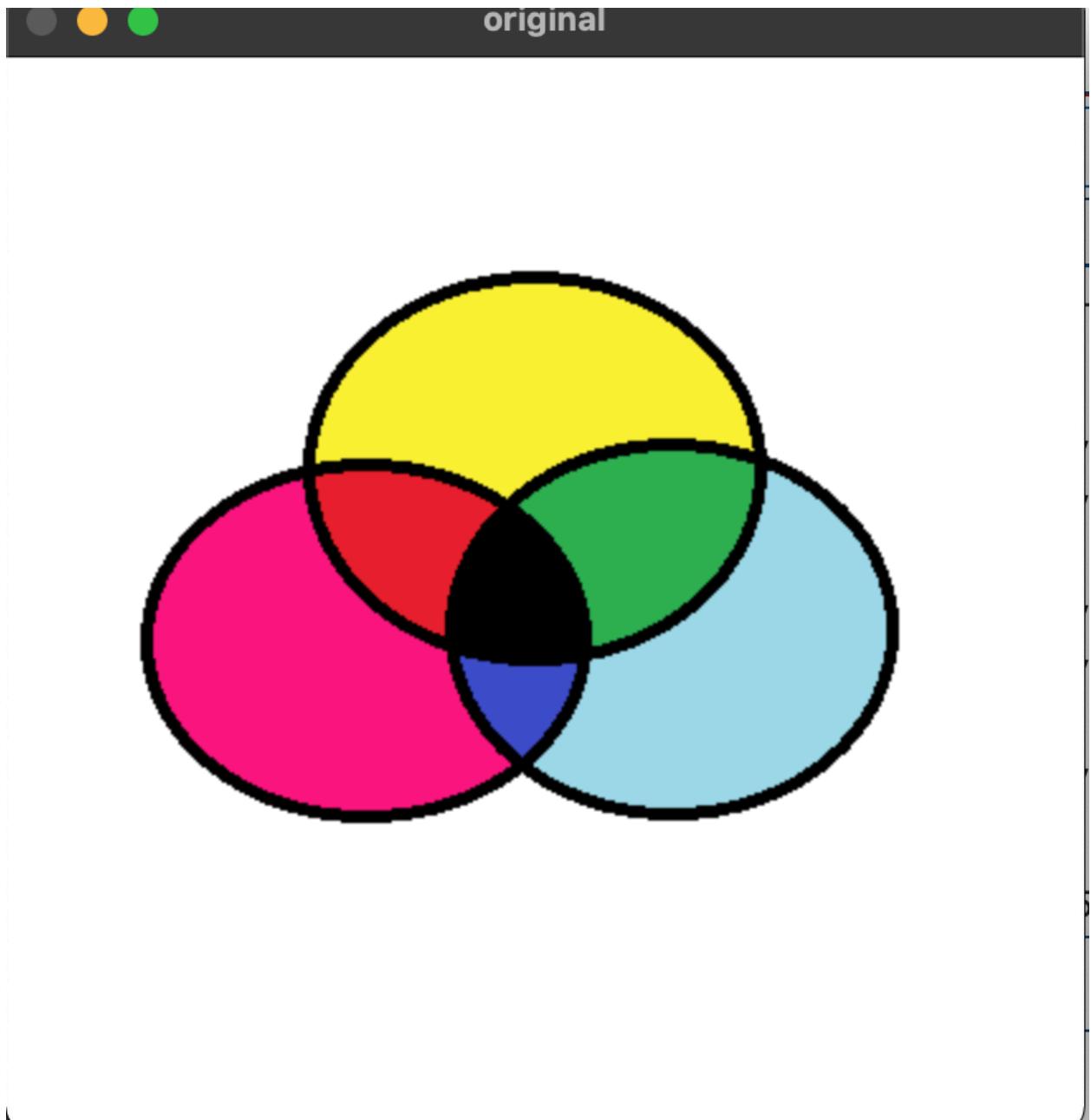
- Reading an image in OpenCV using Python
- Display an image in OpenCV using Python
- Writing an image in OpenCV using Python
- OpenCV | Saving an Image
- Color Spaces
- Arithmetic operations on Images
- Bitwise Operations on Binary Images











Before saving image:

```
['pic8.png', '2.1.png', 'pic1.png', 'water_jug_problem.py', 'pic3.png', 'pic2.png', '2.
```

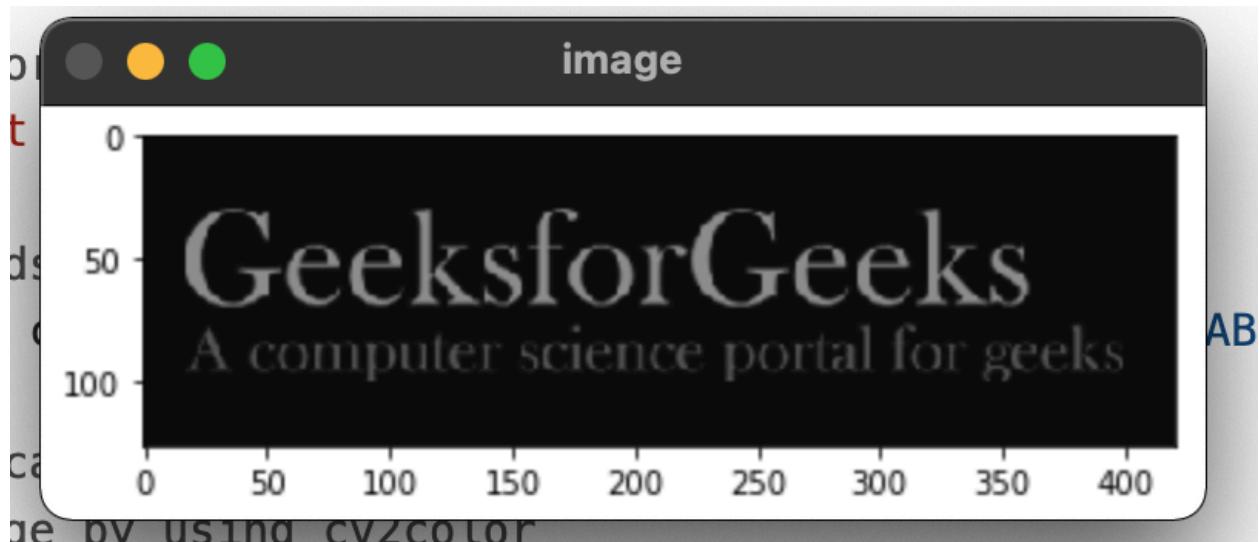
After saving image:

```
['pic8.png', '2.1.png', 'pic1.png', 'water_jug_problem.py', 'pic3.png', 'pic2.png', '2.
```

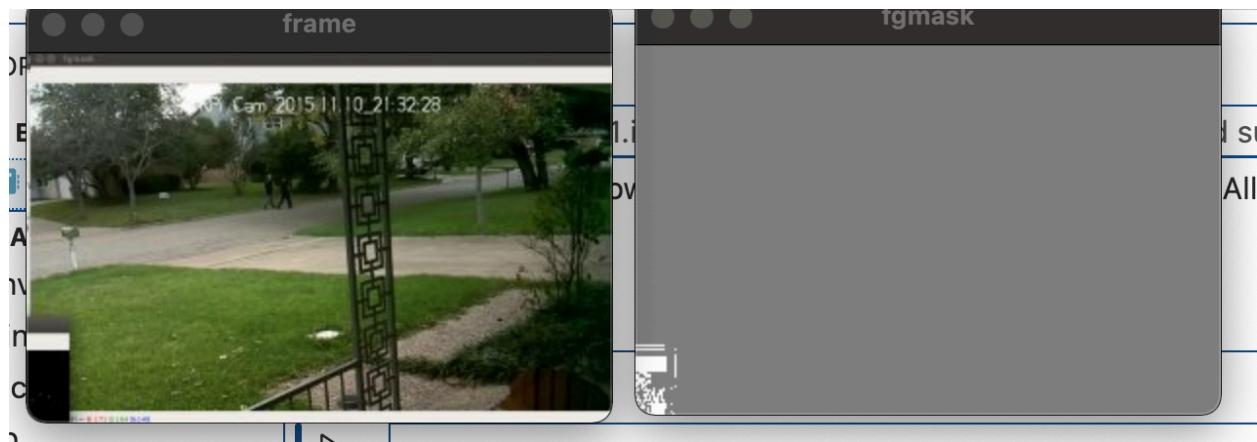
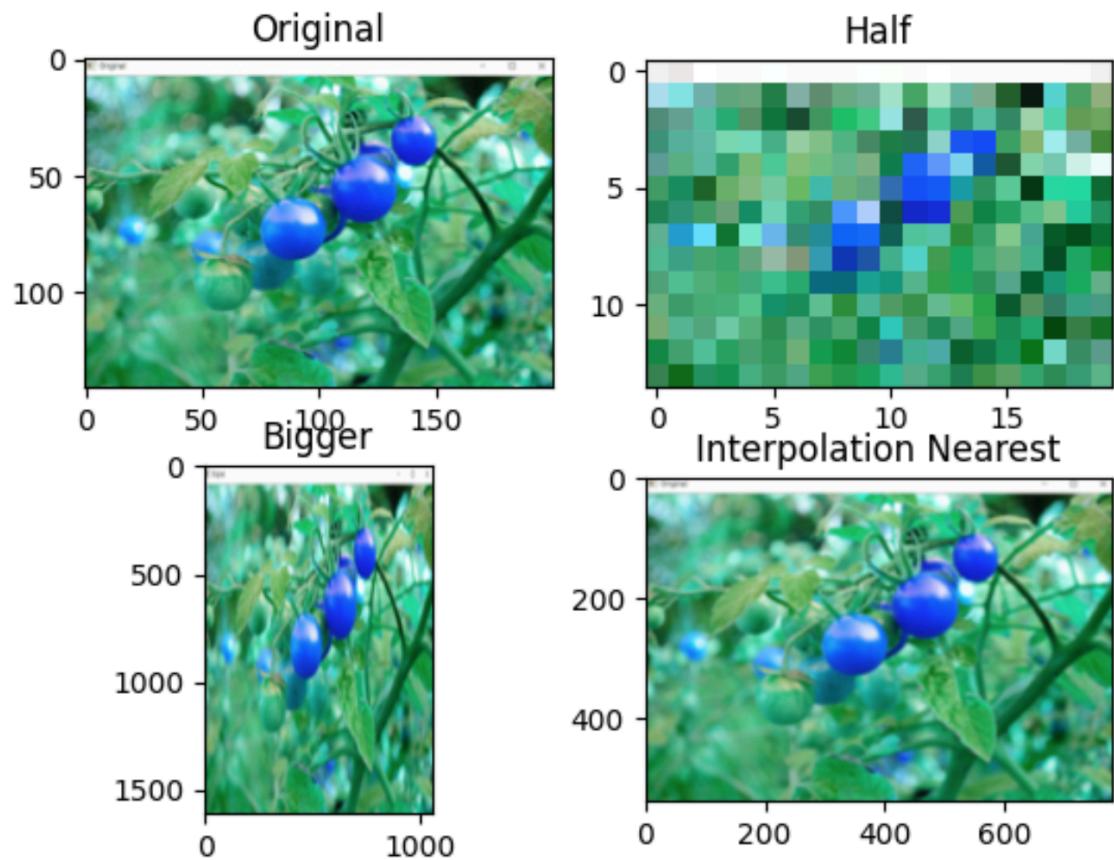
Successfully saved

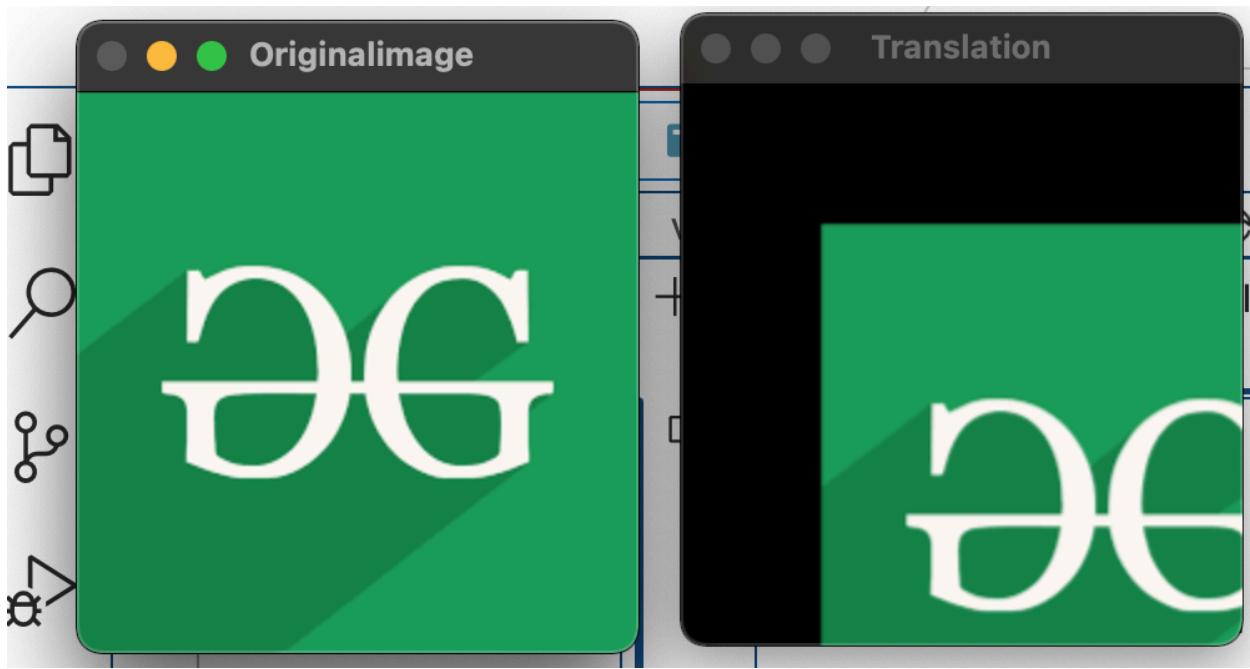
2.2

only important ones.









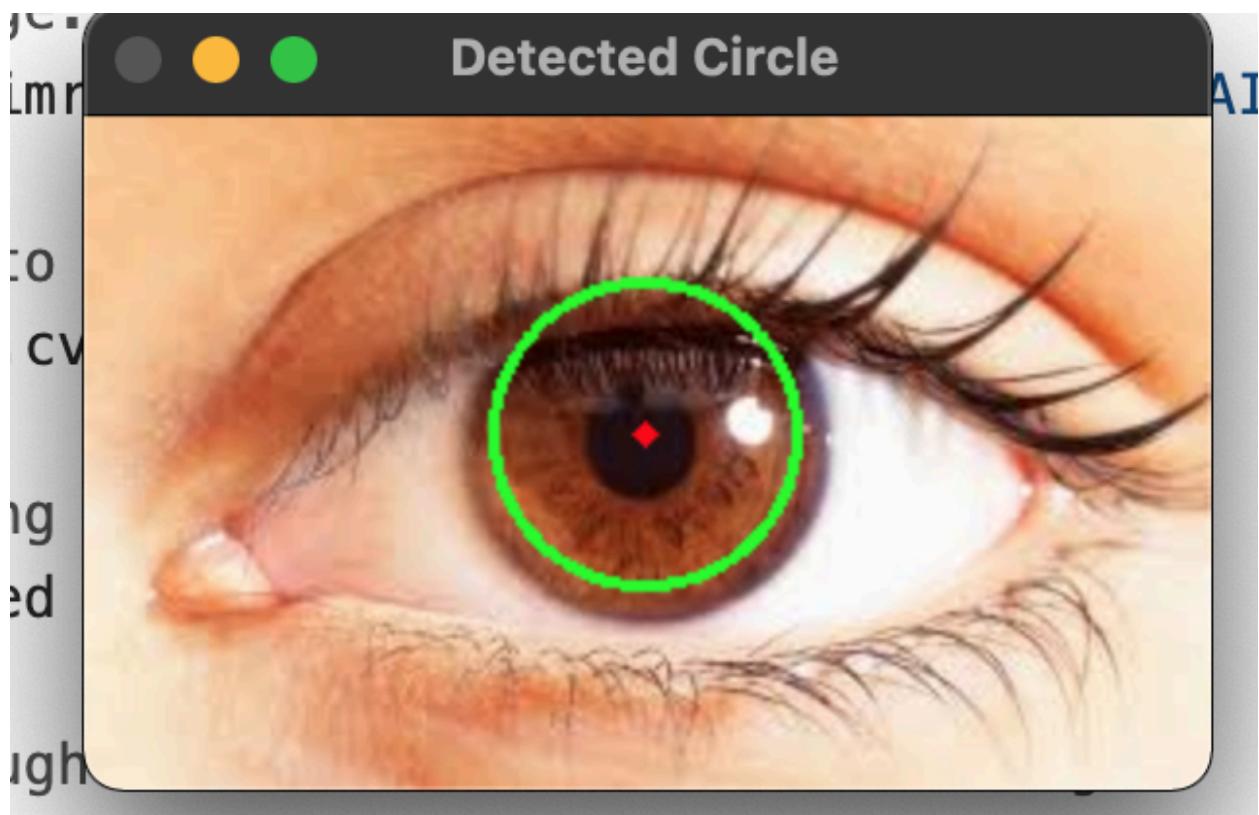


2.3

Feature Detection and Description

- [Circle Detection](#)
- [Detect corner of an image](#)
- [Corner Detection with Shi-Tomasi method](#)
- [Corner detection with Harris Corner Detection](#)
- [Find Circles and Ellipses in an Image](#)



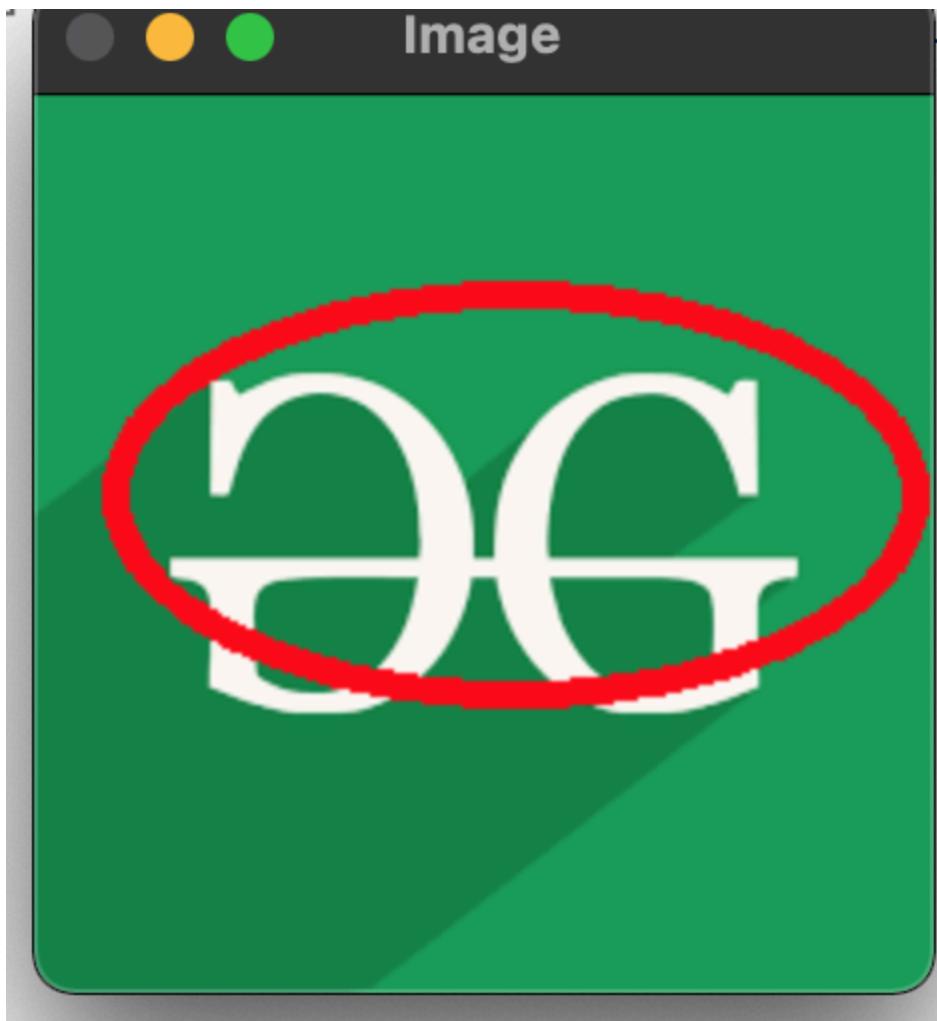


2.4

Drawing Functions

- [Draw a line](#)
- [Draw arrow segment](#)
- [Draw an ellipse](#)
- [Draw a text string](#)
- [Find and Draw Contours](#)
- [Draw a triangle with centroid](#)



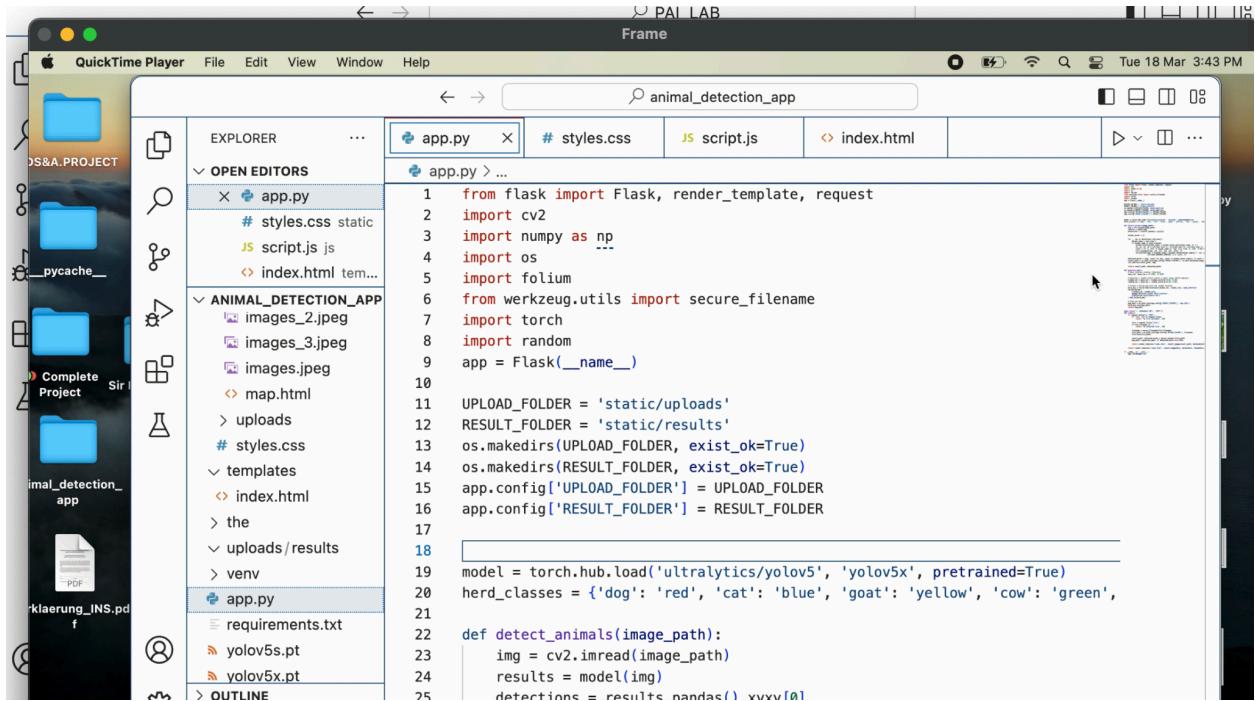






3.1

- [Play a video using OpenCV](#)



```
from flask import Flask, render_template, request
import cv2
import numpy as np
import os
import folium
from werkzeug.utils import secure_filename
import torch
import random
app = Flask(__name__)
UPLOAD_FOLDER = 'static/uploads'
RESULT_FOLDER = 'static/results'
os.makedirs(UPLOAD_FOLDER, exist_ok=True)
os.makedirs(RESULT_FOLDER, exist_ok=True)
app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER
app.config['RESULT_FOLDER'] = RESULT_FOLDER
model = torch.hub.load('ultralytics/yolov5', 'yolov5x', pretrained=True)
herd_classes = {'dog': 'red', 'cat': 'blue', 'goat': 'yellow', 'cow': 'green'}
def detect_animals(image_path):
    img = cv2.imread(image_path)
    results = model(img)
    detections = results.pandas().xyxy[0]
```

3.2

Video Processing

- [Create video using multiple images](#)
- [Extract images from video](#)

Creating.... /data/frame12.jpg

Creating.... /data/frame13.jpg

Creating.... /data/frame14.jpg

Creating.... /data/frame15.jpg

Creating.... /data/frame16.jpg

Creating.... /data/frame17.jpg

Creating.... /data/frame18.jpg

Creating.... /data/frame19.jpg

Creating.... /data/frame20.jpg

Creating.... /data/frame21.jpg

Creating.... /data/frame22.jpg

Creating.... /data/frame23.jpg

Creating.... /data/frame24.jpg

...

Creating.... /data/frame4072.jpg

Creating.... /data/frame4073.jpg

Creating.... /data/frame4074.jpg

Creating.... /data/frame4075.jpg

Video generated successfully!

coin-detection.jpg is resized

8.jpeg is resized

6.jpg is resized

Number of Images: 3