EE604: Assignment 1

Prof. Tushar Sandhan
sandhan@iitk.ac.in

Semester-I, 2022

Due date: 28^{th} August, 2022 Weight: 18%
Due time: 23:45PM Submission: MookIT

Introduction

In this work you will be learning basic image manipulation, usage of peculiar properties of human visual system and color image representations. You are allowed to use python libraries of your choice, but only for system input-output, reading images to get matrices and writing images in JPG files. You may require the usage of image representation in a matrix form, 2D as well as 3D matrix manipulations and your prefrontal cortex in order to solve significant real-world problems using just image representations.

1 Sea Marks: `dotmatrix` [5%]

Sea markers are crucial traffic-control and safety indications along seashores. Dot matrix panels, like those used at traffic signals on the road, can be built with high intensity LEDs for both long-distance and quick visibility. The LED sea marker at Coachin Shipyard, where several ships keep approaching at each hour for docking, malfunctioned one fateful night. If they are not instantly routed to their docking arena, it will take several weeks to clear the ensuing traffic jam.

Engineers have chosen to quickly transform a standard LCD advertising television panel into a dot matrix display in order to use it as a sea marker. Each dot in the dotmatrix should be a circular disc of white or black colour with a radius of 25 pixels, taking into account the Mach band effect and the spatial frequency response properties of human visual systems. 15 discs make up each number (row x col - 5×3). Height and width of dotmatrix image should be 300×500 and it should be grayscale (unsigned int) having any pixel value to be either 0 or 255 (e.g. Fig. 1).

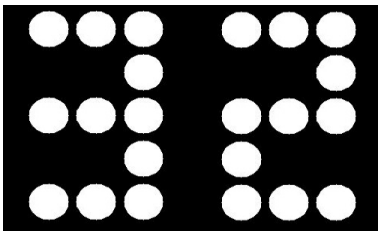

Figure 1: `dotmatrix.jpg` file for the below command `$python dotmatrix.py 32`

Write a python program file `dotmatrix.py`, which accepts command line docking station number $n_d \in \{00, 01, ..., 99\}$ and directly stores the `dotmatrix.jpg` file (e.g. Fig. 1) at current location (Note: No need to display any image, just store image file).

2 Jigsaw Puzzle: `jigsolver` [6%]

In the input `jigsaw.jpg` image, as shown in Fig. 2, some portions and sections are misaligned or jumbled. You need to perform rearrangements of the sub-matrices (i.e. pieces of the jigsaw puzzle), to solve the puzzle. Write a python program file `jigsolver.py`, which accepts command line file path for input `jigsaw.jpg` and directly stores the `jigsolved.jpg` file at current location where the command `{$python jigsolver.py ./jigsaw.jpg}` is being executed.

Figure 2: Input `jigsaw.jpg`

3 Cyclone: `cycloneanalyzer` [7%]

Cyclone generates an environment so tumultuous that it is nearly impossible to analyze it using physical sensors, sea water ships or undersea submarine observations, or even aircraft flying nearby to collect data for predicting its route or intensity. The only method that can aid in this weather forecasting is the processing of satellite images. It aids in the automatic identification of cyclone cores, gauging the feeding and dispersing materials, estimating rotational and translational velocities, and future forecasts. For the sake of simplicity of the assignment, cyclone cores are assumed to be given (red) and that their rotating speed and injecting-dispersing stuff remain constant. In contrast to ocean regions, man-made infrastructures over land restricts open wind flows, therefore as cyclones approach land, their inertia increases and they continue along the same route.

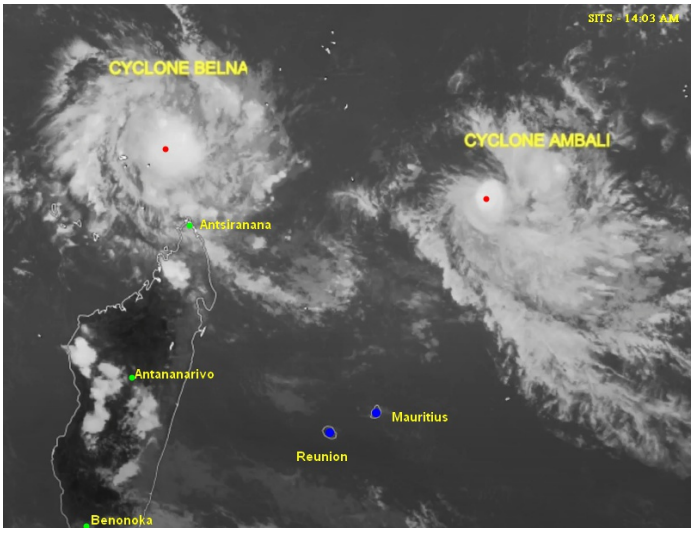

Figure 3: Satellite image capturing the twin cyclones and corresponding geo-locations in the Indian ocean

Madagascar is located on an island in the Indian Ocean. It stretches 1500 kilometres from north Antsiranana (green) to south Benonoka (green) as shown in Fig. 3. It has the capital city of Antananarivo in the centre (green). The islands of Reunion and Mauritius (blue) are separated by 250 kilometres to its east. The twin cyclones Belna and Ambali are approaching Madagascar. The geographic place of their future trajectory intersection is known as apocalyptic juncture (apojuncture). Predict the apojuncture for Belna-Ambali and forecast: How far (in Km) the apojuncture will be from the capital city? and When will it occur (how many hours after the last satellite image capture)? and Will they actually collide in reality?

You have to write python notebook file `cycloneanalyzer.ipynb` for doing all your analysis including reading, manipulating images, intermittent results and final answers justifying your analysis (use various cells of python notebook: code, section, heading, text etc.).

Submission

For each question if requires then use the original images (JPG files) that were provided to you separately apart from this document. The only purpose of the pictures and figures here is to help with the proper explanations.

Before the submission deadline, compress only the files listed below into a single file named `rollno_A1.zip` (eg.`191234_A1.zip`) and upload it to mooKIT.

If your program does not produce output and throws errors, the TA will not debug it and you will receive a score of 0 for that question. If more than 80% student's program execute correctly on TA's evaluation environment then remaining students can not claim that 'program was working on their PC, but not on TA's evaluating environment'. You alone are accountable for the submitted program's proper operation. Kindly make sure it executes and outputs just the things which are being asked for.

Depending on the output quality, partial to full credit will be awarded if the program executes correctly. For plagiarized responses, even if your program does not execute or unexpectedly produces correct answer, in all cases you will be awarded full marks with an honorarium prefix as a negative sign.

Files in `rollno_A1.zip` should be:

- `dotmatrix.py`
- `jigsolver.py`
- `cycloneanalyzer.ipynb`

■