



# **UAS-DTU**

## **Unmanned Aerial Systems - Delhi Technological University**

### **ROUND 2: Technical Round**

#### **Software Department**

## **Background**

Autonomous image segmentation and feature detection and classification is an important aspect of image processing. Image segmentation is the process of “partitioning a digital image into multiple segments”. The goal of segmentation is to simplify and change the representation of an image into something that is more meaningful and easier to analyse. Feature detection includes finding areas of interests such as edges, corners and simple shapes. These features are then classified into various categories based on their shape, colour or other inherent features. These concepts are widely used in military and civilian UAV missions to gather information about areas out of human reach, such as disaster-stricken or mountainous areas.

# Task

The theme for this task is Search and Rescue. A fire has broken out in a civilian area and your job is to gather information about the location of houses and buildings in the area. Your UAV is collecting images of the search area that look like the sample image given below.

## Information about the input image

- The brown area is burnt grass
- The green area is unburnt (green) grass
- The blue and red triangles are houses
- Each house colour has an associated priority
- Blue house have a priority of 2
- Red houses have a priority of 1



*Sample Image*

**The task for you is to return the following information from the image**

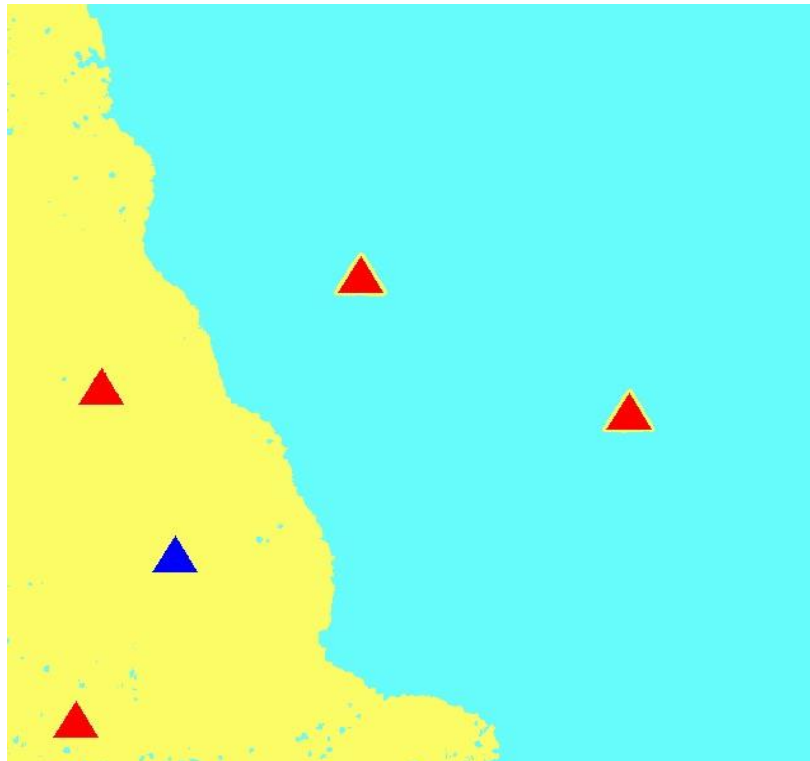
## Input

A list of 10 images, similar to the sample image provided above

## Expected Output

1. An output image, for each input image, that clearly shows the difference between the burnt grass and green grass, by overlaying 2 unique colors on top of each. The expected output for the given sample input is given below
2. The number of houses on the burnt grass ( $H_b$ ) and the number of houses on the green grass ( $H_g$ ), saved in a list
3. The total priority of houses on the burnt grass ( $P_b$ ) and the total priority of houses on the green grass ( $P_g$ ), saved in a list
4. A rescue ratio of priority  $P_r$  where  $P_r = P_b / P_g$ , saved in a list
5. A list of the names of the input images , arranges in descending order of their rescue ratio ( $P_r$ )

The expected output for the given sample image is given below:

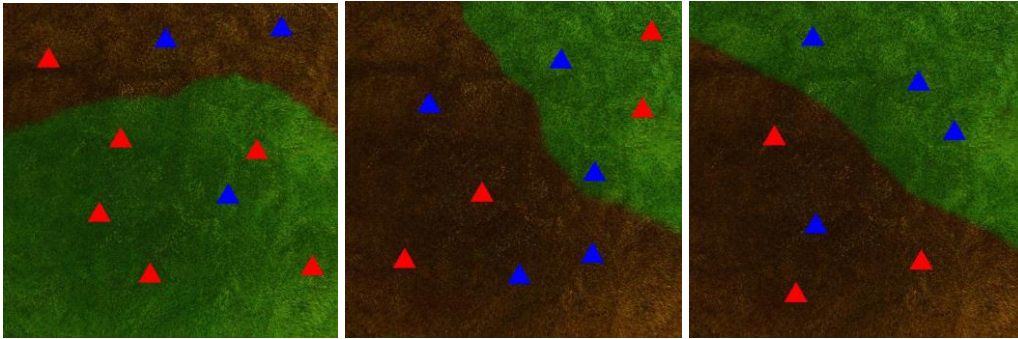


*Sample output*

2.  $[[3,2]]$  (There are 3 house on the burnt grass and 2 houses on the green grass)
3.  $[[4,2]]$  (Priority of houses on burnt grass =  $2 \times 1 + 1 \times 2 = 4$  (two red houses and one blue house) and priority of houses on green grass =  $2 \times 1 = 2$  (two red houses))
4.  $[2]$  (Ratio of priority is  $4/2 = 2$ )
5.  $[image1, image3, image4..... etc]$  (this is based on the priority ratio of the various images given as input) **(not related to the given sample image)**

**Example:**

**Input images**

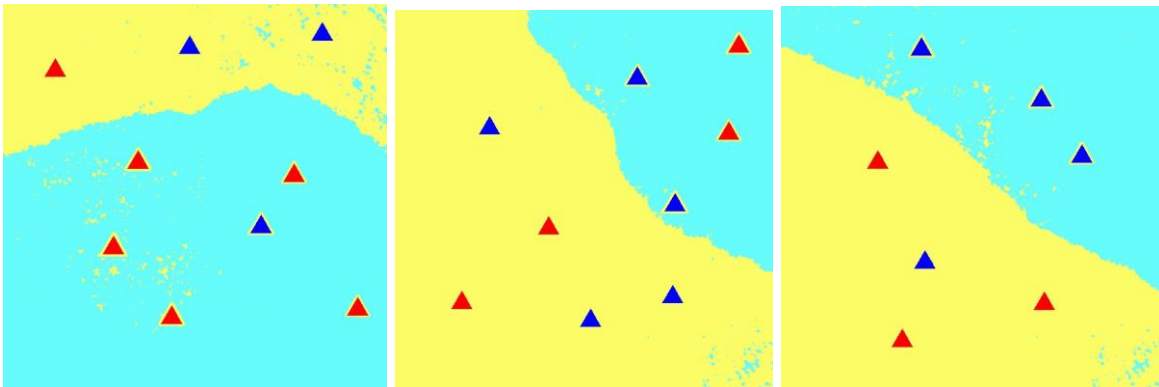


**image1**

**image2**

**image3**

**Expected Output**



**image1**

**image2**

**image3**

```
n_houses = [[3,6],[5,4],[4,3]]
```

```
priority_houses = [[5,7],[8,6],[5,6]]
```

```
priority_ratio = [0.71,1.33, 0.83]
```

```
image_by_rescue_ratio = [image2, image3, image1]
```

**To simplify the given task, we've given a step-by-step approach to learn various concepts and libraries that are required to complete the task**

# Step By Step

1. [Optional] Install Ubuntu in your disk partition, refer to this link for details on how to do this. Note that this task often takes time but we highly encourage you to use Ubuntu once recruited so doing this might give a head start. But note that this is purely optional and you will not be evaluated on the basis of this. (<https://www.youtube.com/watch?v=-iSAyiicyQY>)
2. Learning and picking up new things is the key to work we do. Download and learn Python3, get comfortable with basic data structures used in Python3 (lists, dict, strings, etc). If you know CPP this might not be a steep learning curve. Refer to the beautifully written documentation python has to offer <https://www.python.org/downloads/release/python-378/>.
3. Learn the basics of NumPy. It is the fundamental package for scientific computing with Python. NumPy brings the computational power of languages like C and Fortran to Python, a language much easier to learn and use. With this power comes simplicity and speed. You can refer to these tutorials: <https://www.codecademy.com/learn/intro-statistics-numpy/modules/dspath-intro-numpy>.  
  
If you want to go deeper you can read the official documentation of NumPy on <https://numpy.org/> or YouTube videos.
4. Learn the basics of OpenCV (Open Source Computer Vision Library), It is an open-source computer vision and machine learning software library which is available in multiple programming languages. You can refer to this video lecture series to learn the basics quickly: <https://pythonprogramming.net/loading-images-python-opencv-tutorial/>. You can also refer to the official documentation of OpenCV on <https://opencv.org/> or YouTube videos.
5. Maintain a logbook or write a short report of details showing on a google doc. Share this with your mentor on his/her email address.
  - a. What did you do each day?
  - b. What changes did you make to your code to improve its performance?
    - i. Error Analysis
    - ii. Identify areas of improvement
    - iii. Make changes
    - iv. Write it down

**The task should be compulsorily done on GitHub and should have a comprehensive readme.**

### Evaluation Criteria:

1. Your overall approach to understanding the basics of python
2. Understanding of basic NumPy and OpenCV
3. Code writing skills (is the code clean, well commented)
4. Skills in understanding and usage of new tools which are integral to the work that we are doing here.
5. Documentation.
6. Ability to think analytically and critically.
7. Ability to do the error analysis appropriately.
8. Most importantly grit and commitment! **Task Deadline : 8th Oct 2024 (8:00 PM)**

### Relevant Links:

- Dual boot Ubuntu 20 and Windows 10: <https://www.youtube.com/watch?v=-iSAyiicyQY>
- Python 3.7: <https://www.python.org/downloads/release/python-378/>
- Learn Python 3: <https://automatetheboringstuff.com/> (First 6 chapters are sufficient)
- Learn OpenCV: [https://docs.opencv.org/4.x/d6/d00/tutorial\\_py\\_root.html](https://docs.opencv.org/4.x/d6/d00/tutorial_py_root.html) [Video Lecture]
- Learn GitHub : <https://docs.github.com/en/get-started/quickstart/hello-world>

### End Note:

For those of you with a background in programming and knowledge of python we assume this task won't be very difficult for you. Similarly if someone's just starting off with python this may seem overwhelming and impossible. We need you to know **that's okay and your previous knowledge will not play a role in our selection.** We will make sure it's a level playing field for everyone, **so in case you're just starting out we don't expect you to complete this task 100% but we expect 100% follow through and dedication from all.** If you are able to complete this task earlier than stipulated time we will assign more things to you, the purpose of this task is to see your adaptability to new environments so we encourage you to ask doubts search the internet and find solutions and most importantly enjoy (you will most likely have a good-looking project by the end of this recruitment)

We wish you all the best, and hope to work with you soon!

### Mentor Division

Please find your specific mentor and ask questions only to that mentor, it really helps our management if you stick to the person assigned. Please refrain from calling unnecessarily. Drop a text message or an email.