23 Jan 2026 – Image Processing Fundamentals

Topics Covered:

- Image resizing techniques and impact on resolution
- Understanding image resolution and pixel representation
- Basics of NumPy for image manipulation
- OpenCV color spaces (RGB, HSV, grayscale)
- Edge detection techniques in OpenCV

Outcome:
Developed a foundational understanding of how images are represented and processed, which is essential for UAV-based vision tasks such as terrain analysis and object detection.

24 Jan 2026 – Image Segmentation & Feature Extraction

Topics Covered:

- Image segmentation concepts using OpenCV
- Contour detection and analysis
- Basic contour operations:
    - Shape approximation
    - Distance calculation between detected features
- Segmentation using masks and overlay techniques

Outcome:
Implemented segmentation and contour-based processing pipelines, enabling separation of regions of interest—an important step toward identifying land, water, or potential survivor regions in aerial imagery.

25 Jan 2026 – Version Control & Project Documentation

9Topics Covered:

- Fundamentals of Git and GitHub for version control
- Repository creation and structure planning
- Best practices for continuous progress commits
- Documentation of project progress

Outcome:
Established a structured GitHub repository to track continuous development, experiments, and weekly progress reports, ensuring reproducibility and mentor-friendly progress monitoring.

26 Jan 2026 – Implementation

Topics Covered:

- Formula deduction for score calculation
- Tried different mathematical models for score calculation
- Version control over github

Outcome:

Established a structured GitHub repository to track continuous development, and tried different score calculation formula (weighted average, exponential method,etc.) for optimised solution.

27 Jan 2026 – Further exploration of opencv-python

Topics Covered:

- Visualization of assignment of casualties to camps on the output image
- Refining the score logic
- Version control over GitHub

Outcome:
With the scores calculated I tried to implement a function that shows arrows originating from different rescue pads and pointing to their respective casualties.Adopted linear weighted optimization model for scoring of each casualties.

28 Jan 2026 – Refining the final code and testing a brute force algorithm

Topics Covered:

- Changed the code’s flow from procedural to function wise so that each function can be used independently
- To test different algorithms and correctness of our algorithm, we opted for a backtracking algorithm that goes through each path before presenting an optimised result.
- Tested both the algorithms on different images from the dataset.
- Changed the perspective of the solution from camps to casualties
- Version control over GitHub.

Outcomes:

Now the code looks more presentable with independent function for different tasks and the brute force logic:

As in an image there can be 9 casualties at max and for each casualty there are 3 camps to choose from at max there can be 3^9 options and so for each option some operations to perform for score calculation tho having O(3^n) time complexity makes it bad algorithm but for fixed number of input and for testing purposes it works fine. For each option score calculation is similar as in old versions of the code, here we focus on maximization for each path and then choose the optimized path.

29 & 30 Jan 2026 – Implementation of final operations

- Implementing a function to output the nested list of different casualties assigned to different camps and then sorting images on the basis of final output.
- Version control over GitHub

Outcome:

The code now is complete with all the functions and operations needed to output an optimized solution to the given task also tracked on GitHub with different versions of the program.