

# Report

## ***1. Approach and Methodology:***

- The program reads two images: the Schedule and the Room Plan, both of which contain components represented by icons.
- The Schedule is preprocessed by converting it into a binary image to isolate icons. We use contour detection to extract individual icons from the Schedule image.
- The extracted icons are then matched against the Room Plan image using template matching (`cv2.matchTemplate`). This function scans through the Room Plan image and searches for areas that match the provided icon.
- A similarity threshold (set to 0.8) is applied, and the number of locations in the Room Plan that meet or exceed the threshold are counted as matches for each icon.

## ***2. Assumptions Made:***

- It is assumed that the icons in the Schedule are distinct and represent objects or components that also appear in the Room Plan.
- The program assumes that the icons appear in their original form in the Room Plan, without any significant scaling or rotation. The template matching method does not account for major transformations.
- The icons extracted from the Schedule are expected to be present in the Room Plan at the same scale and orientation. If the icons in the Room Plan are resized or rotated, the matching accuracy may decrease.

### ***3. Challenges Encountered and How They Were Addressed:***

- **Icon Size and Scaling:** Since template matching works best when the scale and orientation of the icons in both images are the same, any differences could lead to mismatches or missed icons. This was mitigated by testing with a fixed threshold value, but future improvements could include scaling the icons dynamically or using more advanced matching techniques like feature matching (e.g., ORB or SIFT).
- **Performance:** Template matching is computationally expensive, especially for large images and multiple icons. To optimize this, the icons were extracted using contours and only the extracted portions were used for matching, reducing the search area.

### ***4. Explanation of How the Code Works:***

- **Icon Extraction:** The `Schedule_page-0001.jpg` is read and converted to a binary image using a thresholding technique. This makes it easier to identify and isolate the icons from the Schedule image using contour detection. Each detected contour is used to extract an icon, which is stored in a list.
- **Template Matching:** The program uses OpenCV's `matchTemplate` function to scan the Room Plan image for occurrences of each icon from the Schedule. A threshold is applied to control the level of match similarity required. The number of matching locations is counted for each icon and stored in a dictionary.
- **Output:** The results are printed and also saved to a text file for submission.

### ***Sample Output:***

Extracted 5 icons from the schedule.

Matches found: {'Icon\_1': 7498, 'Icon\_2': 127292,

```
'Icon_3':127292 , 'Icon_4': 28309500, 'Icon_5': 2642}
```

This is the snapshot of first five icons.