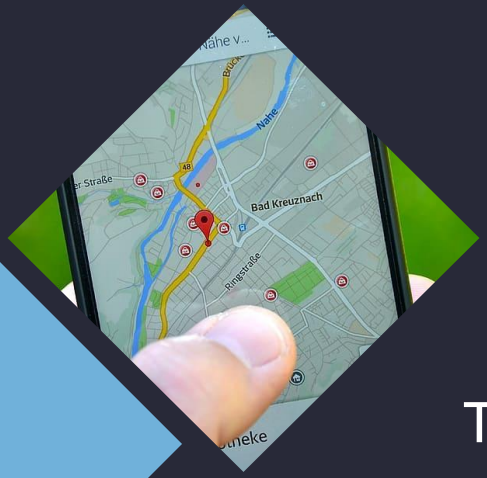
The Lost Student App

Image Retrieval, Image Based Localization
Computer Vision Final Project

Ron Haikin
Roy Amoyal
Omri Hirsh

The Problem

Imagine being a freshman at Ben-Gurion University, stepping onto campus with excitement and anticipation,
only to find yourself lost in a maze of buildings, unsure of where your next class is located



Our Solution

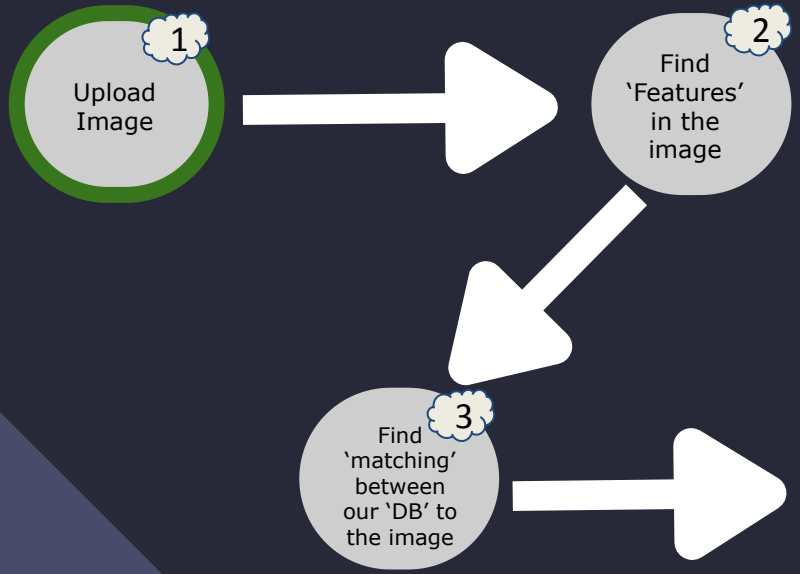
The Lost Student App

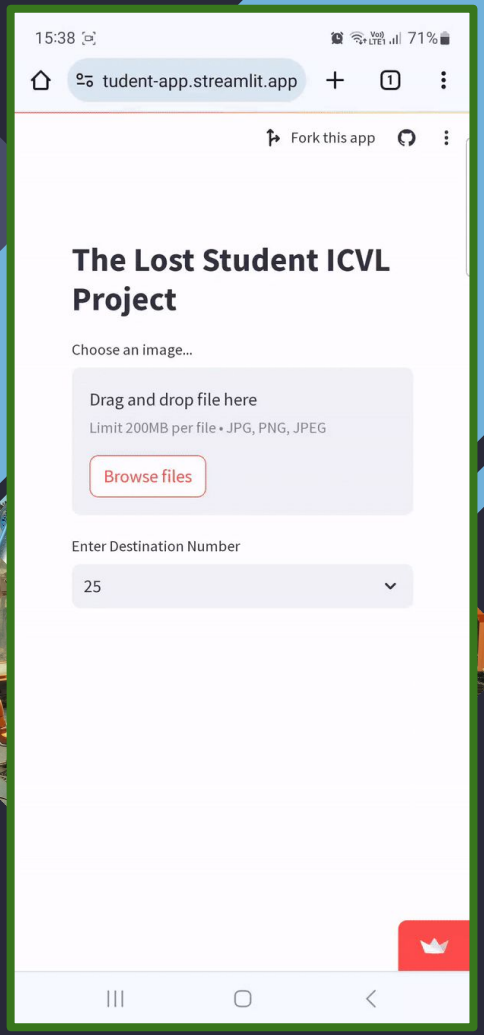
- Smartphone Camera Based
- Easy to use
- Navigation Between Buildings

The Pipeline

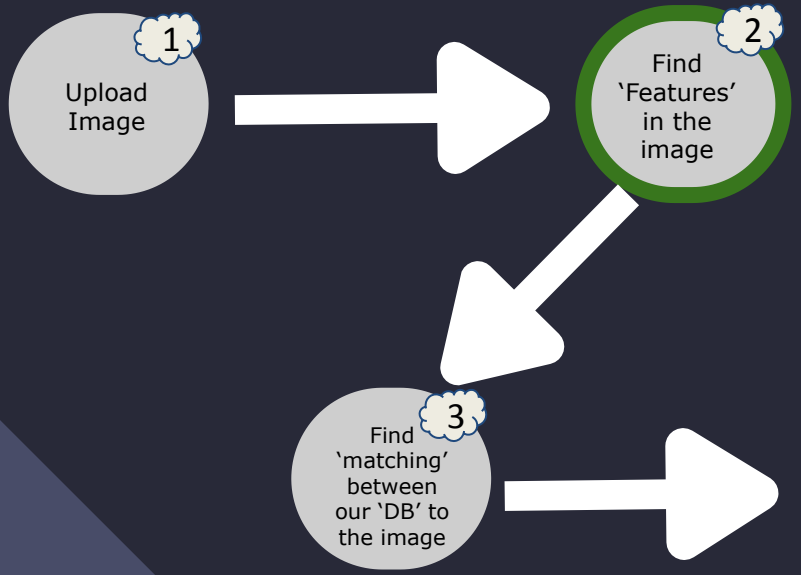
1. Upload Image
2. Find ‘Features’ in the image
3. Find ‘matching’ between our ‘DB’ to the image
4. Show the Path on the Map

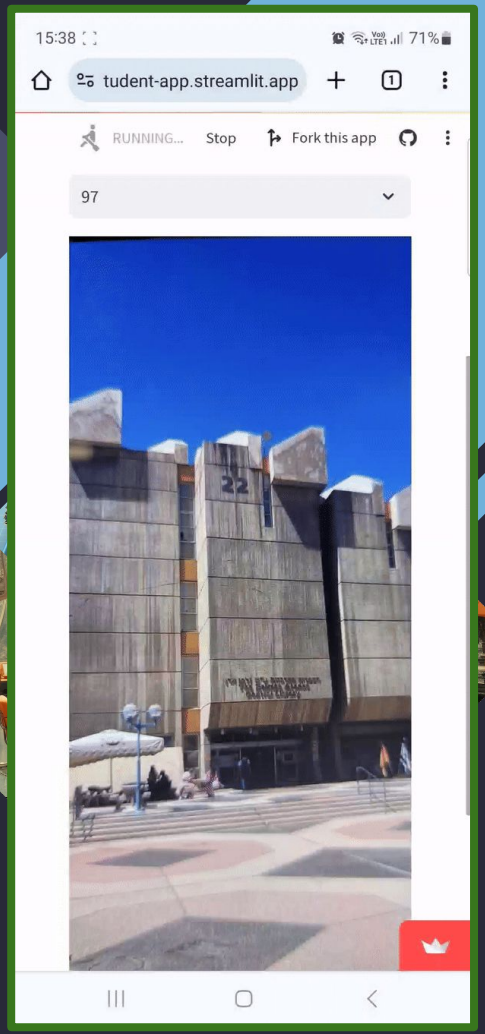
The Pipeline



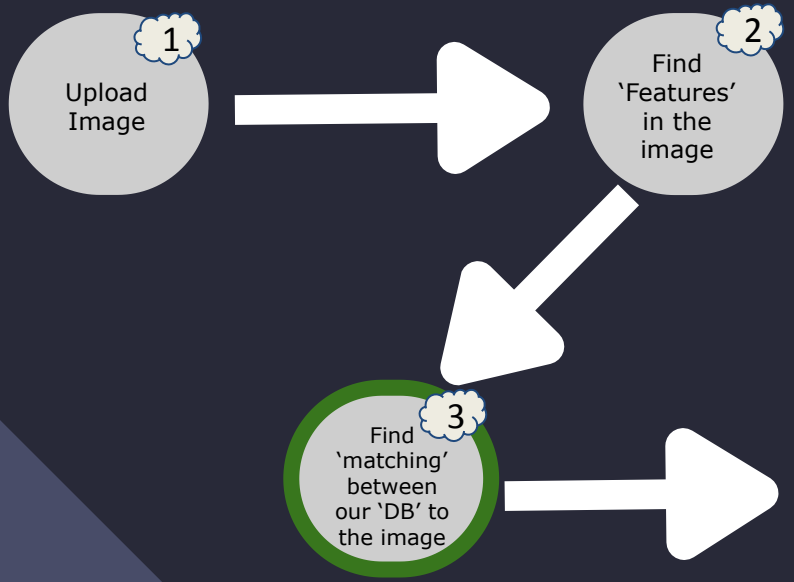


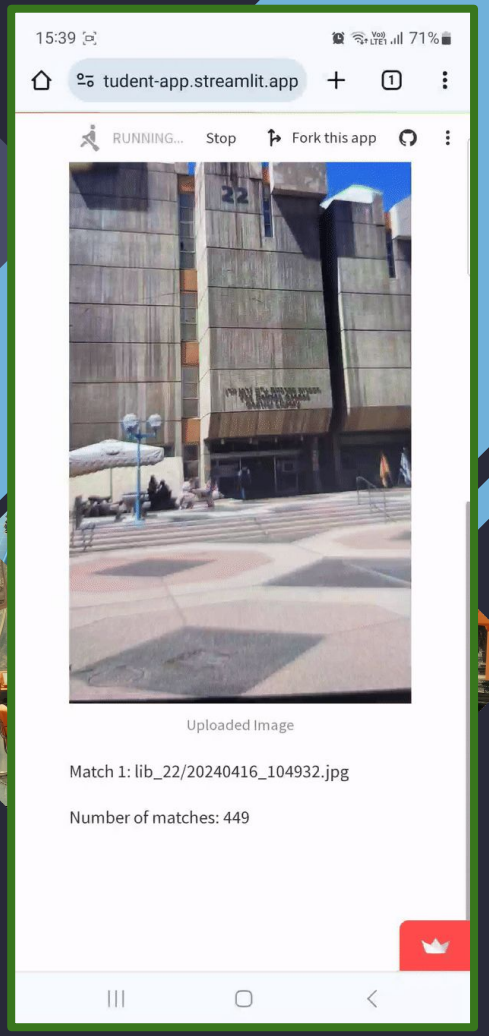
The Pipeline





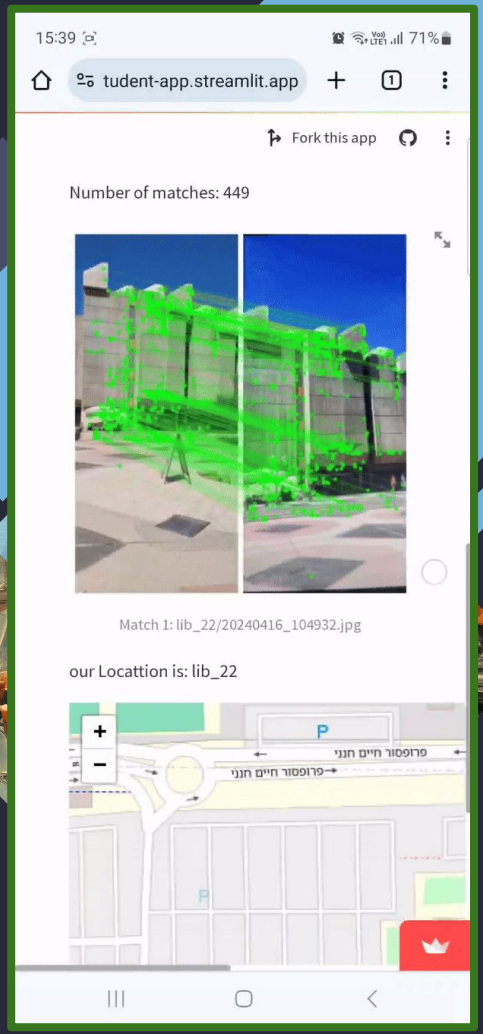
The Pipeline





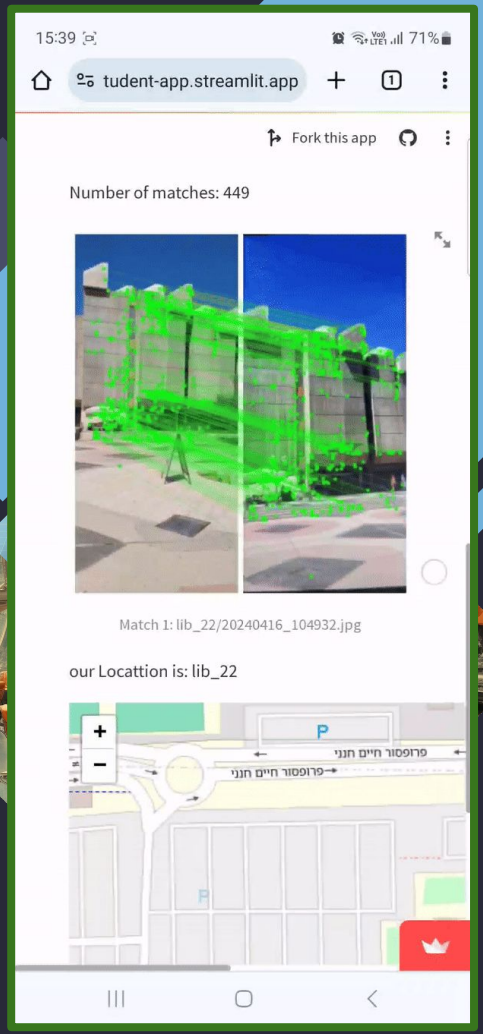
The Pipeline

1. Upload Image
2. Find ‘Features’ in the image
3. Find ‘matching’ between our ‘DB’ to the image
4. Show the Path on the Map



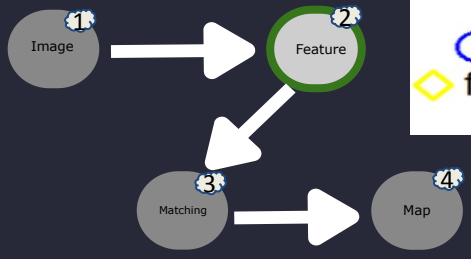
The Pipeline

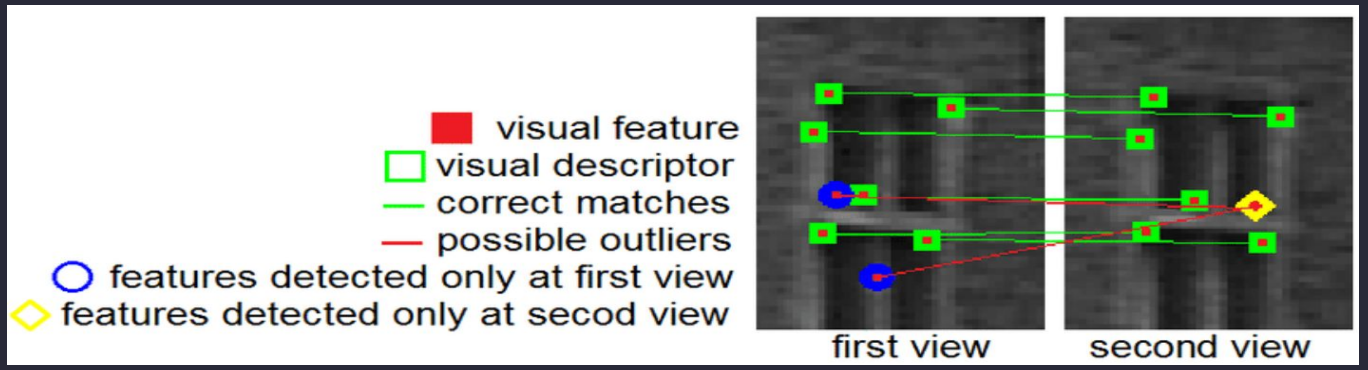
1. Upload Image
2. Find ‘Features’ in the image
3. Find ‘matching’ between our ‘DB’ to the image
4. Show the Path on the Map



What are features?

- **Features in an image are distinctive points that represent important visual information, such as corners or blobs.**
- **Descriptors are numerical representations of these features that encode their appearance or surroundings, enabling algorithms to match and compare them efficiently for tasks like object recognition or image stitching.**

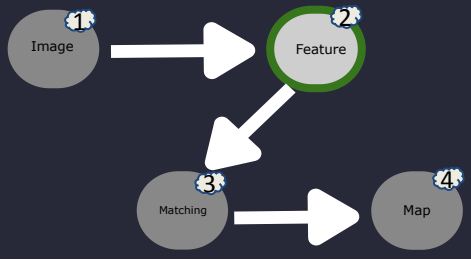


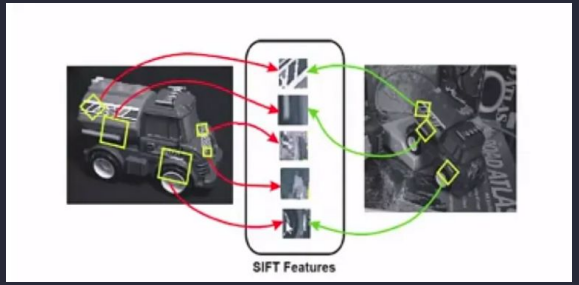


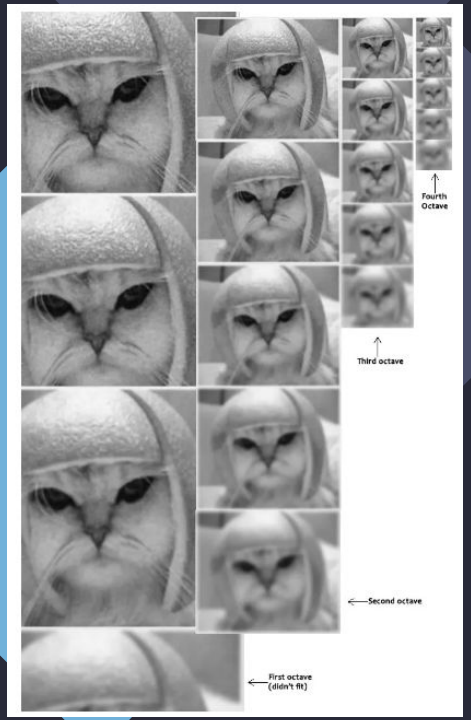
SIFT - Scale Invariant Feature Transform

SIFT features ={Location, scale, orientation, descriptor}

- **Location:** Pixel location of keypoint in image
- **Scale:** Scale of the keypoint that we got the maximum respace for the keypoint
- **Orientation:** Orientation of each keypoint by computing gradient based on the histogram representation
- **Descriptor:** A 128-diminal floating point numbers that stores the local surrounding relations to the orientation



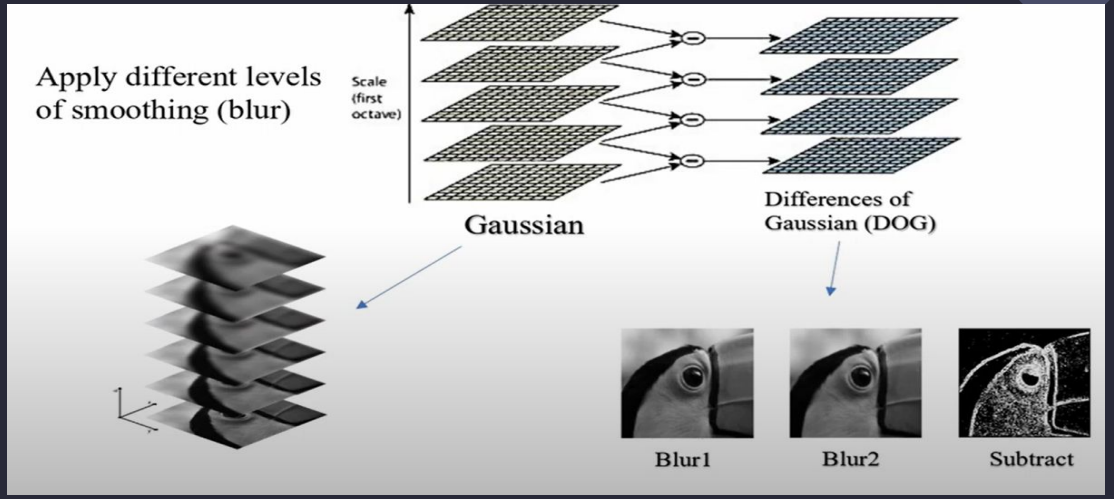


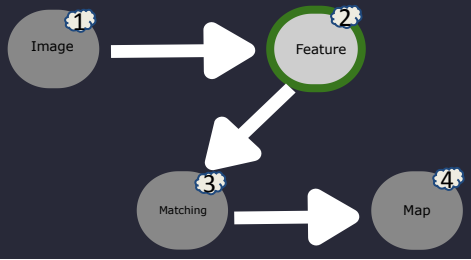


SIFT Workflow

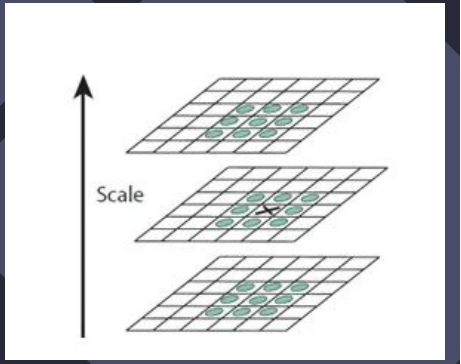
Scale-Space Extrema Detection:

- Build a scale space by applying Gaussian blurring and downsampling the image.



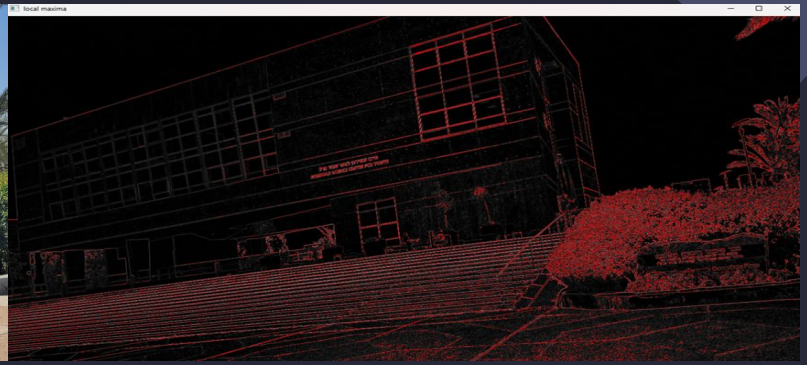


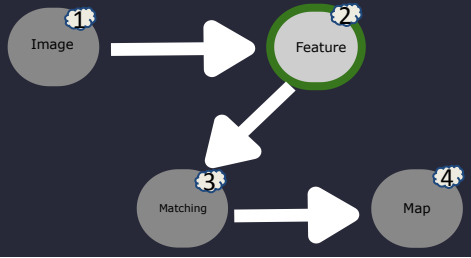
SIFT Workflow



Scale-Space Extrema Detection:

- Detect potential keypoints (extrema) across scales by comparing each pixel with its neighbors.



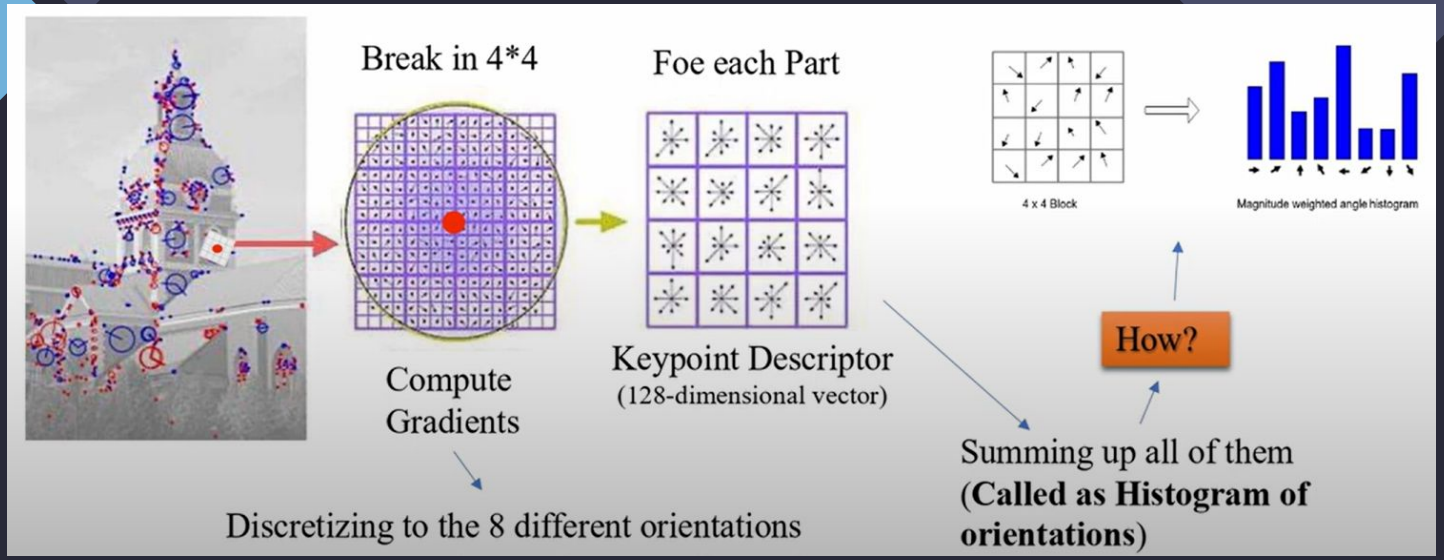


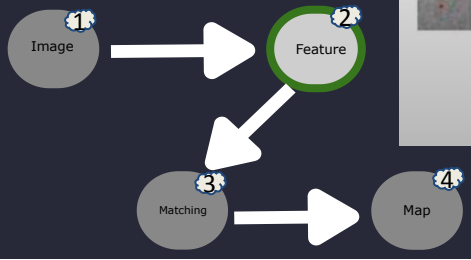
8	4	0
0	237	1
32	64	128

SIFT Workflow

Keypoint descriptor:

- Generate a feature descriptor vector for each keypoint based on local image gradients.

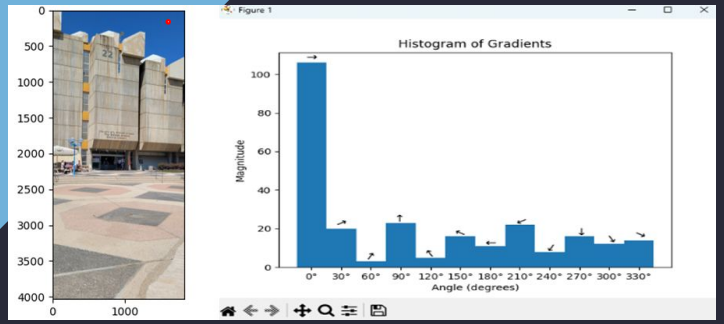


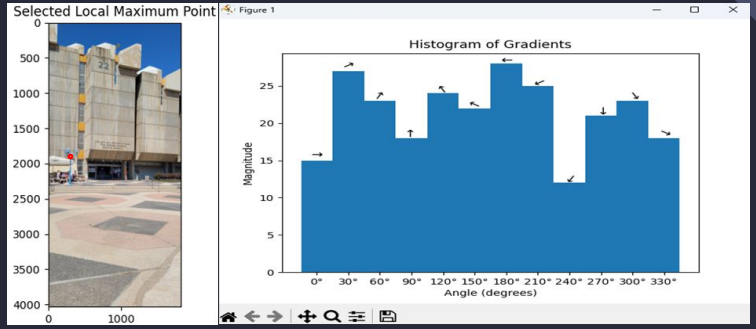


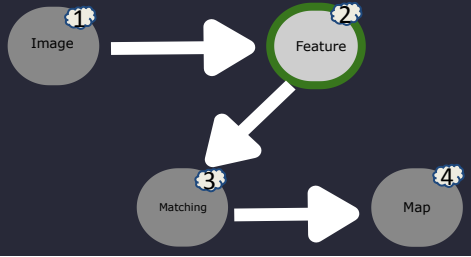
SIFT Workflow

Orientation Assignment:

- Compute gradients orientations histograms around each keypoint and assign a dominant orientation.
- Keypoint descriptors will be invariant to image rotation due to this orientation assignment.

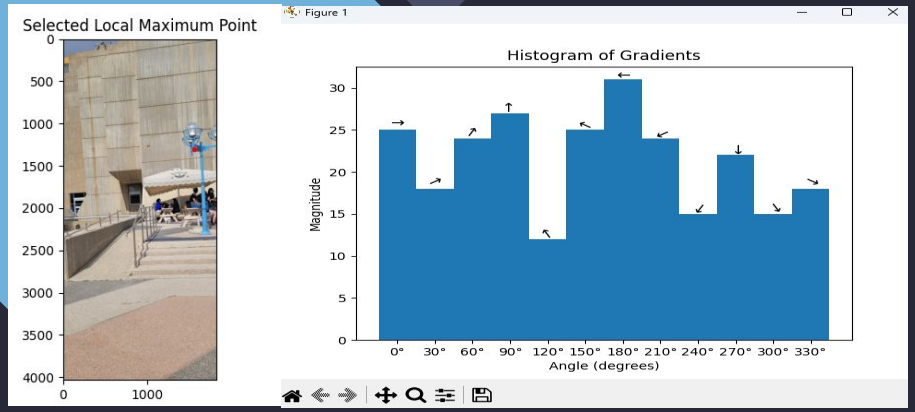


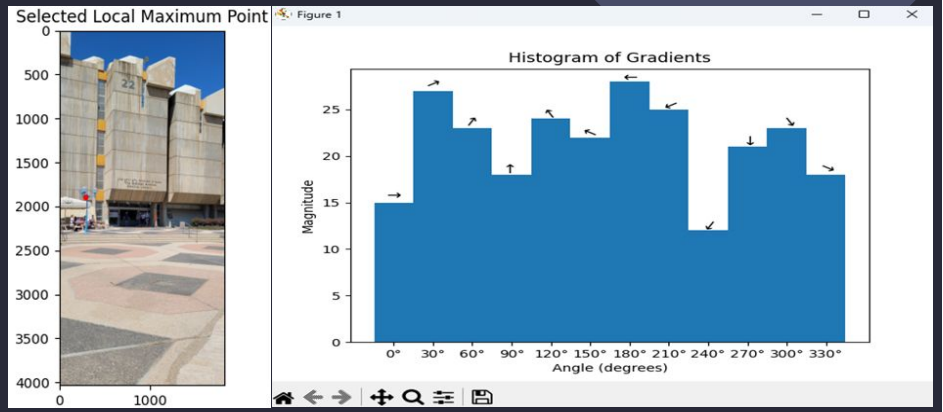


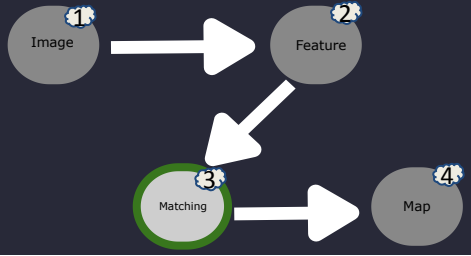


SIFT + Old School Matching Limitations

$$L2\_Norm(\mathbf{descriptor}_1, \mathbf{descriptor}_2)$$

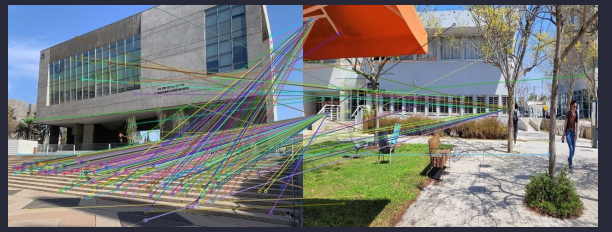






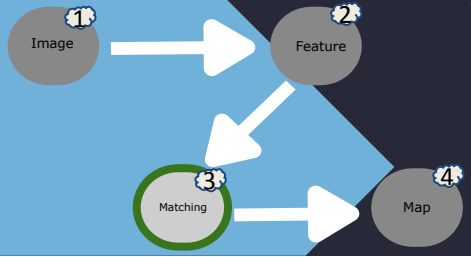
SIFT, SURF, ORB, AKAZE and other classic features and descriptors can't handle different perspectives with the classics matching techniques!

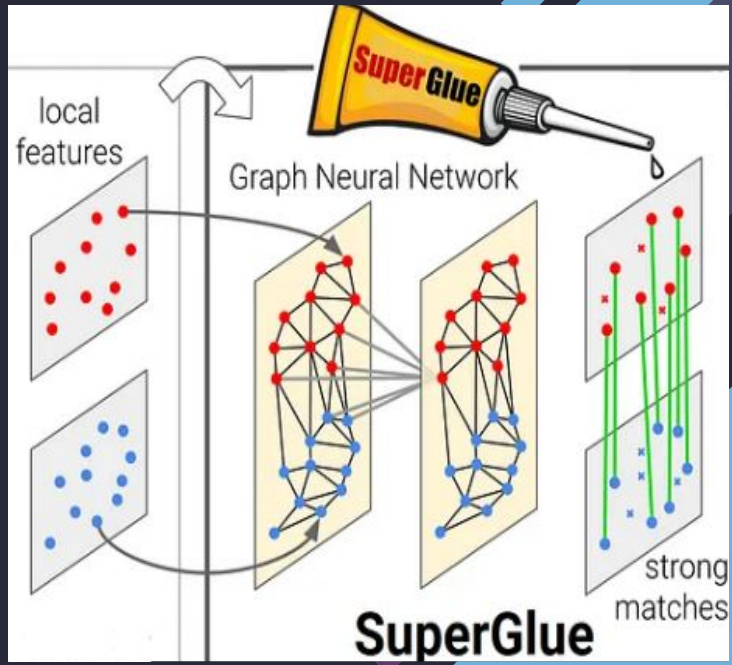
We still want to be able to use them, maybe we can change the matching process?



Matching Computation- Exploring “Light Glue”

- Deep neural network for matching keypoints in images
  - Input: Descriptors Vector 128 Dim - Image1, Descriptors Vector 128 Dim - Image2
  - Output: Matching Between the keypoints
- An improvement to “SuperGlue”, the previous state-of-the-art method
- Achieves high accuracy while being much faster
- Adaptively reduces complexity for easy image pairs



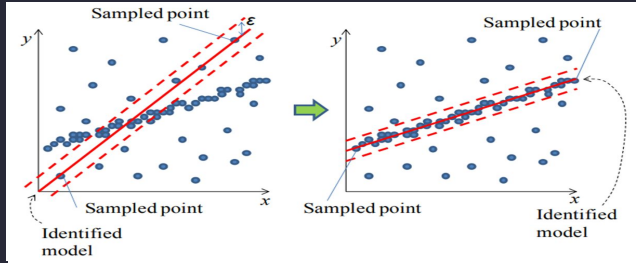


Features matching - Outlier Detection

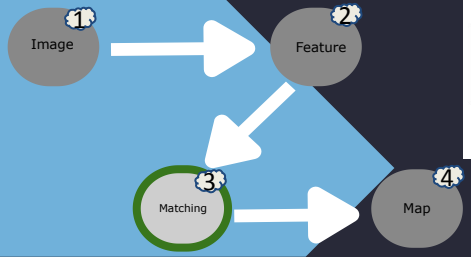
- We used Fundamental matrix that represents a geometric relationship between two images of the same scene taken from different viewpoints.

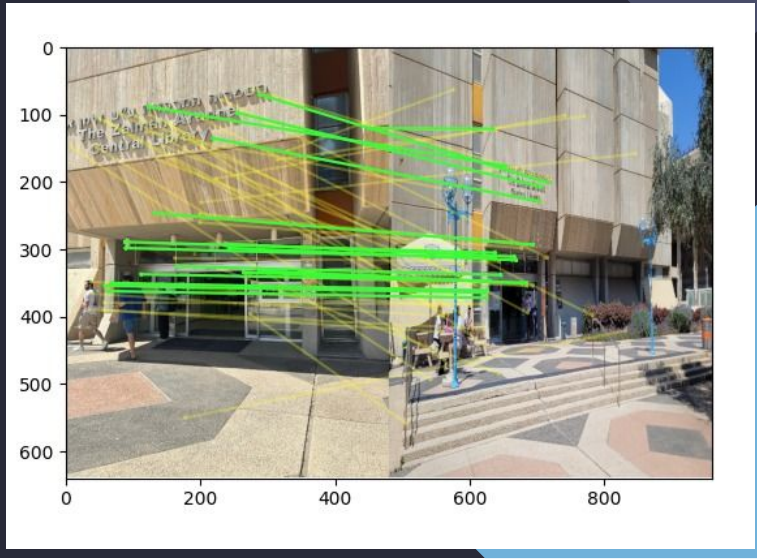
$$[u_l \quad v_l \quad 1]\begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix}\begin{bmatrix} u_r \\ v_r \\ 1 \end{bmatrix} = 0$$

- To compute the Fundamental Matrix, we utilized 2D keypoints matches and applied **Robust** Linear Least Squares with the RANSAC (Robust Maximum Likelihood Estimation) algorithm.



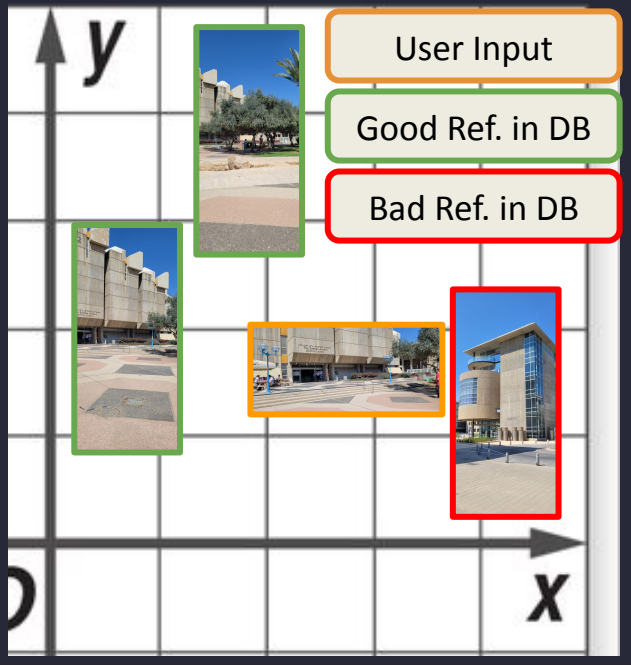
$$\begin{bmatrix} u_1u_1' & u_1v_1' & u_1 & v_1u_1' & v_1v_1' & v_1 & u_1' & v_1' & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_nu_v' & u_nv_n' & u_n & v_nu_n' & v_nv_n' & v_n & u_n' & v_n' & 1 \end{bmatrix}\begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ \vdots \\ f_{33} \end{bmatrix} = 0$$

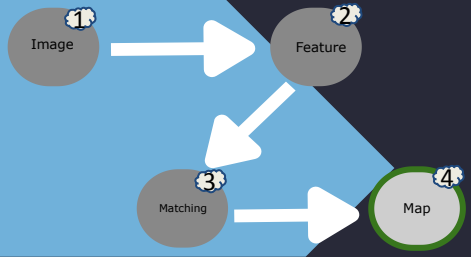




Finding Where You are- k Nearest Neighbor

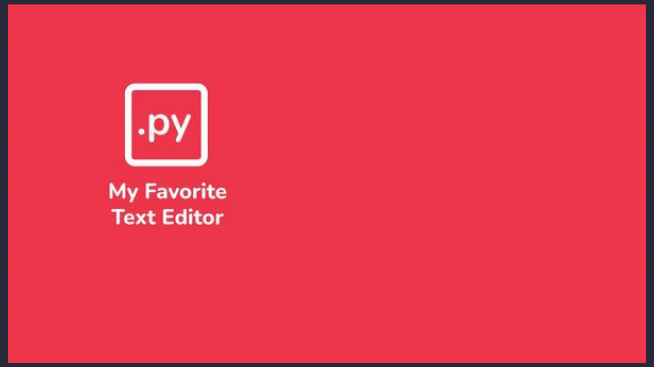
- Took top k images
- By majority vote we choose the building





WebApp - Streamlit.io

- https://streamlit.io/
- Easy to Use
- Best for Phone and Computer use
- Easy to Program
- Python Based



Real-World Applications

- Integrate with BGU student App
- Offer the app for hospitals and large public facilities
- The Lost Student App's technology can be extended into many sectors like robotics, military, healthcare, and more

אוניברסיטת בן-גוריון בנגב
Ben-Gurion University of the Negev

Future Directions

- Improve the UI
- Shorten computation time
- Integrate with BGU app
- Integrate the app with google maps

Conclusion

We solved a Real Life problem
We used a cross platform App and Computer Vision Algorithms
We enjoyed the process of learning

Now it's time for **you** to use the app:

"Embark on a seamless journey through campus with the Lost Student App, your trusted companion for effortless navigation and image-based localization."

Thanks!

Do you have any questions?

ronkh@post.bgu.ac.il,
amoyalr@post.bgu.ac.il,
omrihir@post.bgu.ac.il

Our App
GitHub