Beer Label Classification for Mobile Applications

Team Name: DIPSum

Repo URL:

https://github.com/Digital-Image-Processing-IIITH/project-dipsum

Team Members:

- Madhav Agarwal (2020900022, PGSSP)
- Siddhant Bansal (2019900091, MS by Research)
- Garima Nishad (2019701029, MS by Research)
- Mundru Yallamanda Rao (2019201029, M.Tech)

Mentor TA: Meher Shashwat Nigam

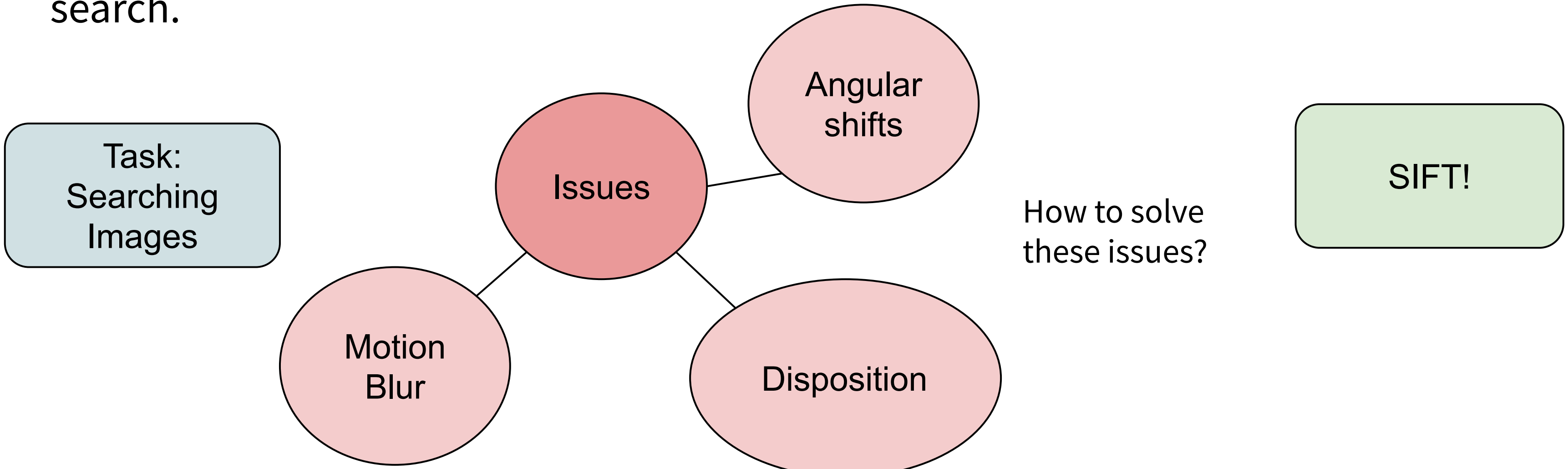
Problem Definition

Matching Beer Bottles to their Labels

- Create an image processing algorithm for the automated identification of beer types using SIFT-based image matching of bottle labels.

SIFT for tackling various image matching issues

- The main problems associated with this task are challenges that are faced in image search in general i.e., searching images irrespective of angle, disposition in the input image.
- In this project we choose to implement SIFT to address the issues w.r.t image search.



Dataset

Database Creation and Pre-processing

- We **have generated** the database containing bottle and label images.
- No more than 5 labels are from the same brewery.
- For each database image, a corresponding query (test) image of a beer bottle with that label was found.

Query Images →

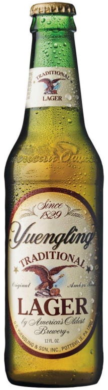
Database →

Protocols followed for data collection

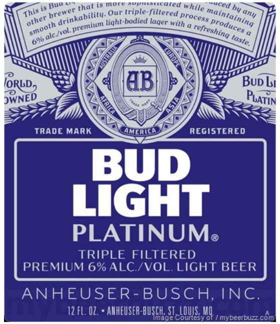
- **260 Images** were downloaded manually from Bing and Google (130 Query, 130 Database)
- **85%** of the query images have a white background
- **15%** of the query images are noisy to analyze the robustness of algorithm
- Aspect ratio of **4:3** is used in most of the query images
- Bottles with very similar labels were avoided

Primary Dataset: Dataset with different labels

- We have collected **100** images in both database and query. Here are some samples:



Query images



Database images

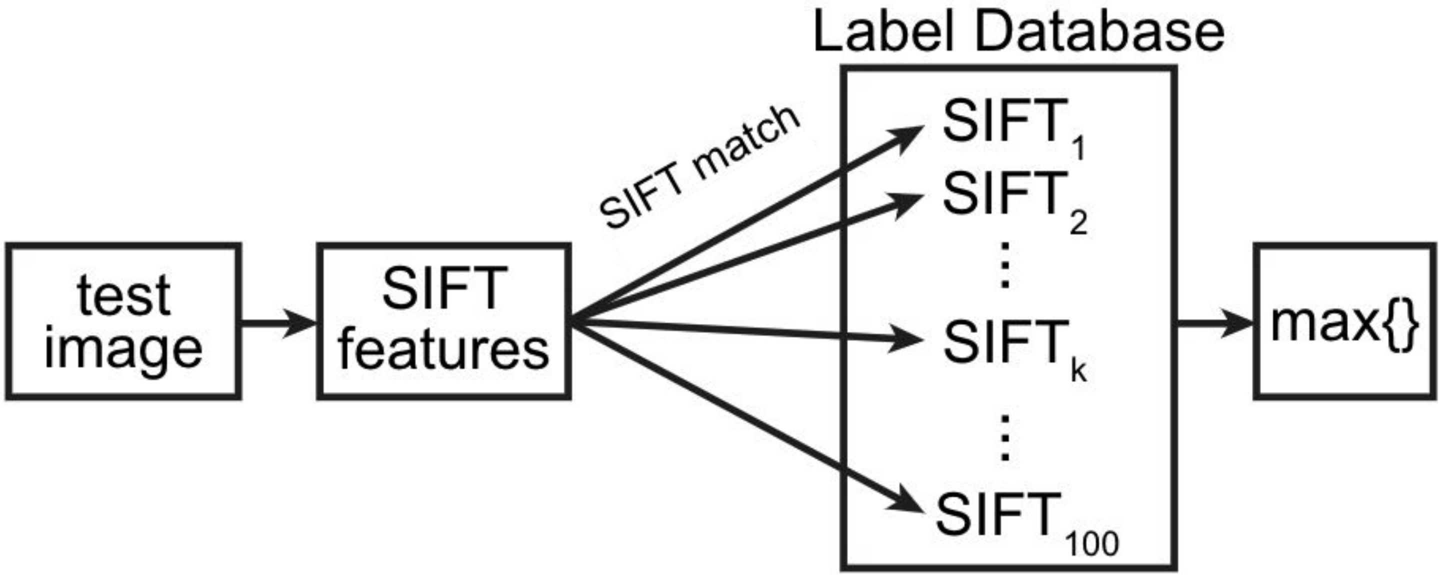
Samuel Adams Dataset: Dataset with similar labels

- We have collected **30** images in both database and query. Here are some samples:

Query images

Database images

Processing strategy



As shown, we'll be extracting SIFT features from the test image. Then, we will match those features with the SIFT features of images from the database. Images having similar features will be queried.

SIFT

SIFT - Keypoint Extraction

- Stands for **S**cale **I**nvariant **F**eature **T**ransform
- Transforms image data into **scale-invariant** coordinates
- Extract distinctive features invariant to image scale and rotation
- Robust to:
  - Affine distortion,
  - Change in 3D viewpoint
- Advantages:
  - Locality
  - Distinctiveness
  - Quantity
  - Efficiency

Steps for Extracting the Key Points

Scale-space peak selection	Keypoint localization	Orientation Assignment	Keypoint descriptor
○ Construct a set of progressively Gaussian blurred images ○ Take differences to get a “difference of Gaussian” pyramid ○ Find local-extrema in this scale-space.	○ Accurately locating the feature key points	○ Assigning orientation to the key points	○ Describing the keypoint as a high dimensional vector

Scale-space peak selection

Construct a set of progressively Gaussian blurred images

↓

Take differences to get a difference-of-Gaussian pyramid

↓

Detect local maxima and minima

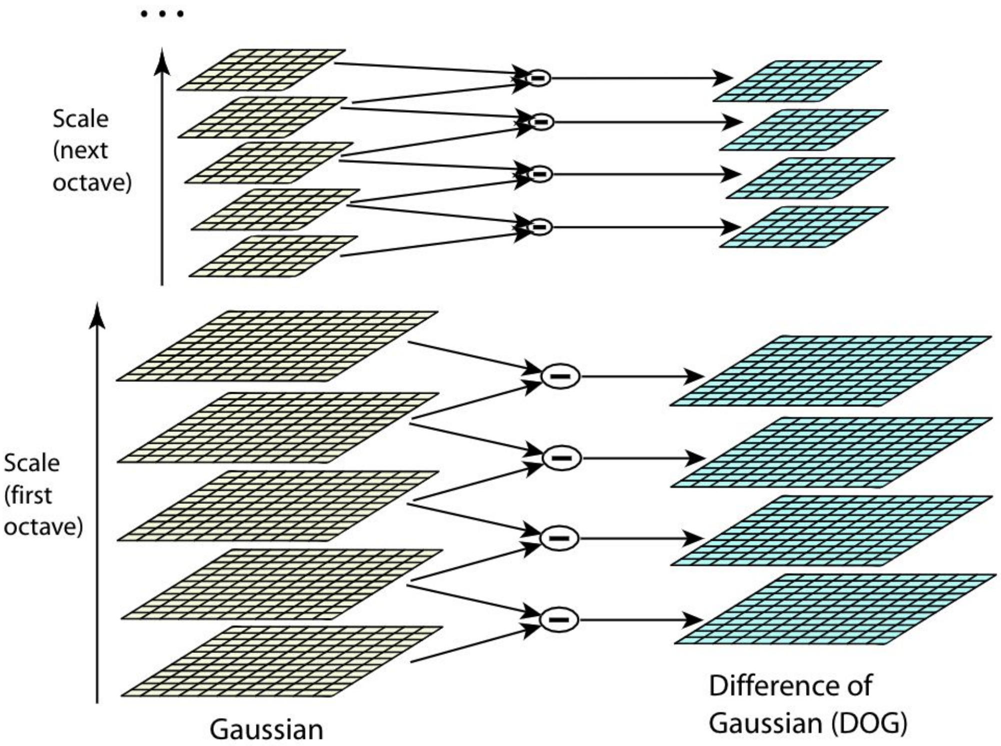


Figure 1: For each octave of scale space, the initial image is repeatedly convolved with Gaussians to produce the set of scale space images shown on the left. Adjacent Gaussian images are subtracted to produce the difference-of-Gaussian images on the right. After each octave, the Gaussian image is down-sampled by a factor of 2, and the process repeated.

https://www.ijcai.org/Proceedings/83-2/Papers/091.pdf

Local extrema detection

To detect the local maxima and minima:

Each sample point is compared to its eight neighbors in the image and nine neighbors in the scale above and below

Point will only be selected if it is larger or smaller than all of these neighbors

Important to determine the frequency of sampling in the image and scale domains that is needed to reliably detect the extrema.

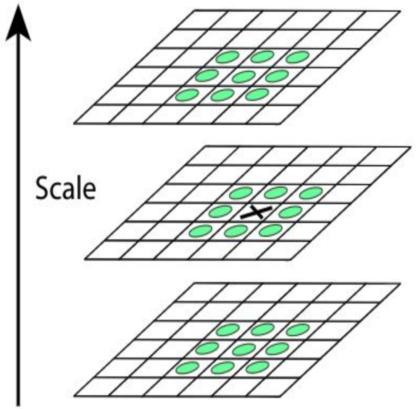
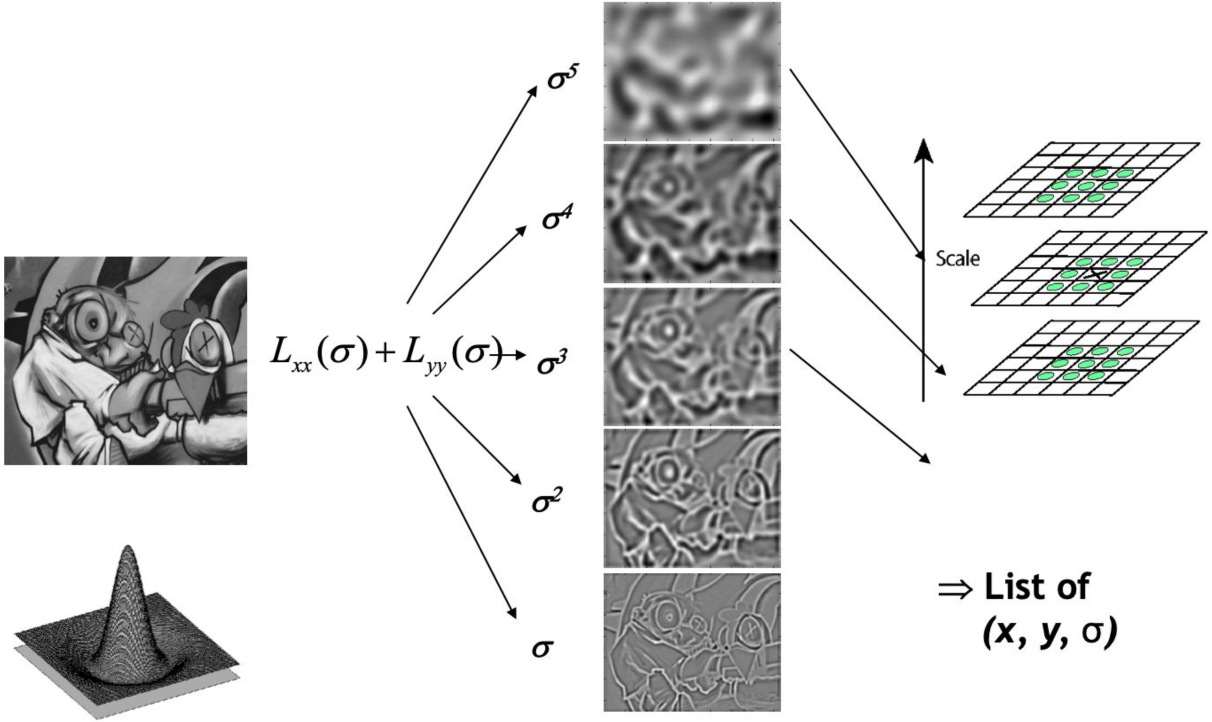


Figure 2: Maxima and minima of the difference-of-Gaussian images are detected by comparing a pixel (marked with X) to its 26 neighbors in 3x3 regions at the current and adjacent scales (marked with circles).

How is scale selection helping?



https://youtu.be/U0wqePj4Mx0

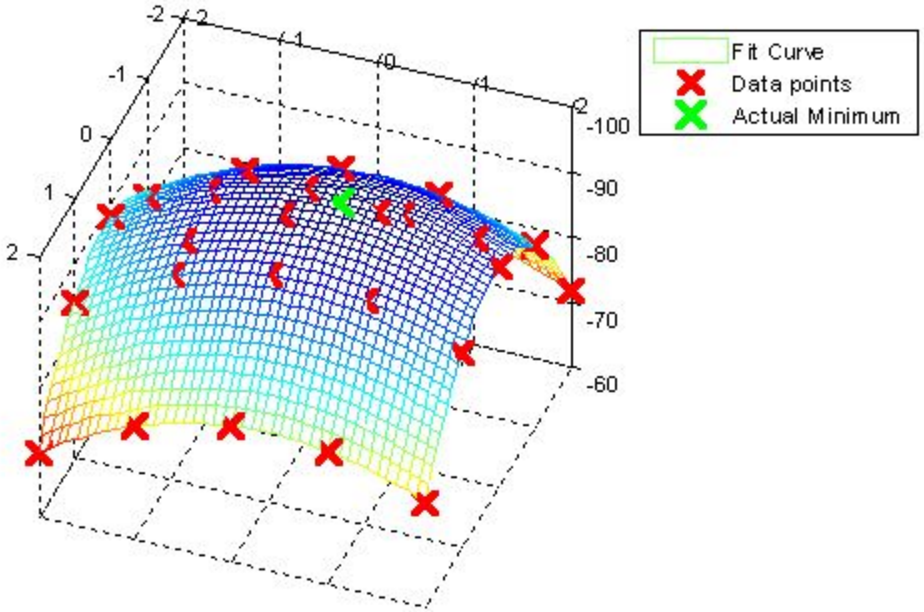
Keypoint Localization

- Fit a quadratic function to the surrounding values (of extremas found in the previous steps) for sub-pixel and sub-scale interpolation.
- Perform Taylor series expansion around that point.

$$D(\mathbf{x}) = D + \frac{\partial D}{\partial \mathbf{x}}^T \mathbf{x} + \frac{1}{2}\mathbf{x}^T \frac{\partial^2 D}{\partial \mathbf{x}^2}\mathbf{x}$$

- The local point of extremum, is determined by taking the derivative of this function and setting it to 0.

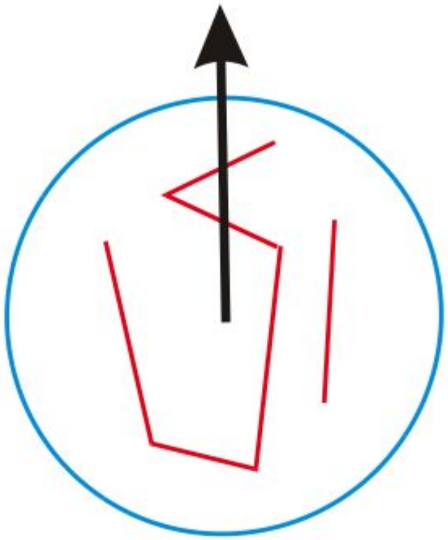
$$\hat{\mathbf{x}} = -\frac{\partial^2 D}{\partial \mathbf{x}^2}^{-1} \frac{\partial D}{\partial \mathbf{x}}.$$

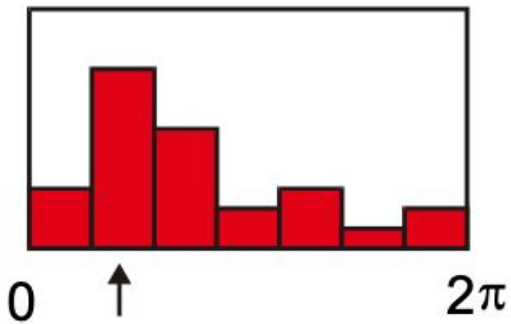


Brown, Matthew, and David G. Lowe. "Invariant features from interest point groups." BMVC. Vol. 4. 2002.

Orientation Assignment

- A consistent orientation is assigned to keypoint for achieving invariance to image rotation;
- Histogram of local gradient directions is computed at the selected scale;
- Now, each keypoint specifies stable 2D coordinates. Which are, x, y, scale, and orientation;
- Select the dominant orientation and normalizing by rotating to the fixed orientation.





https://filebox.ece.vt.edu/~jbhuang/teaching/ece5554-4554/fa16/lectures/Lecture_07_InterestPoints.pdf

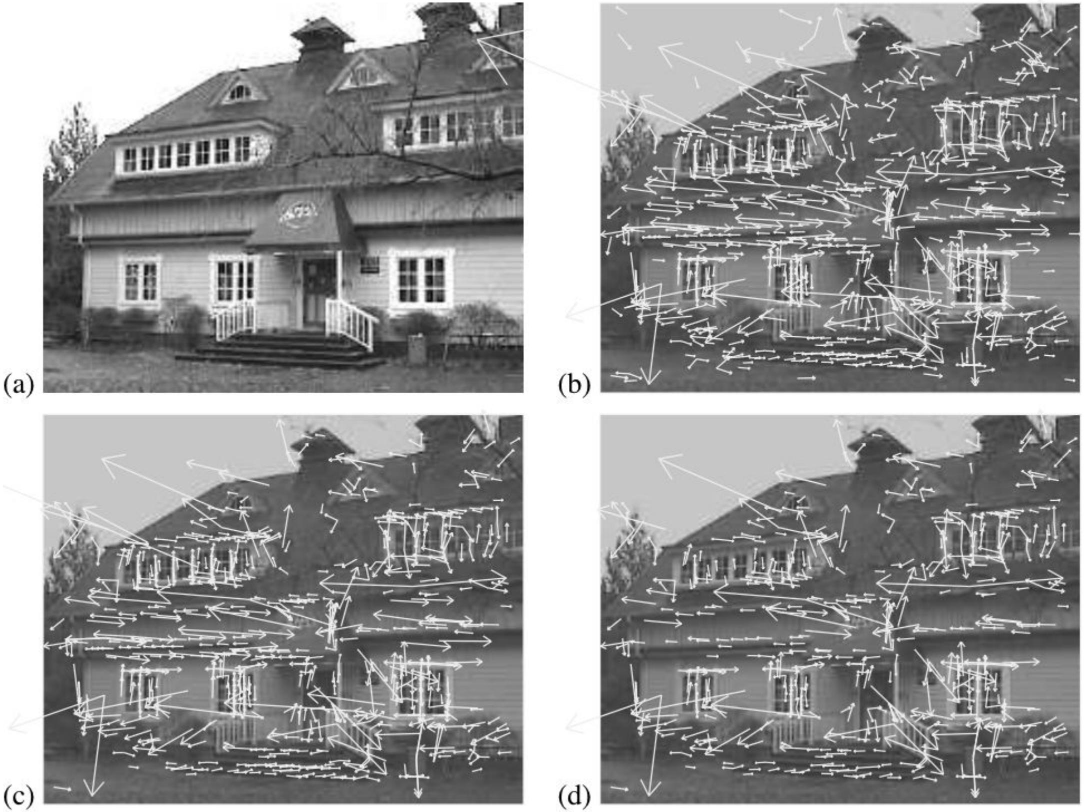
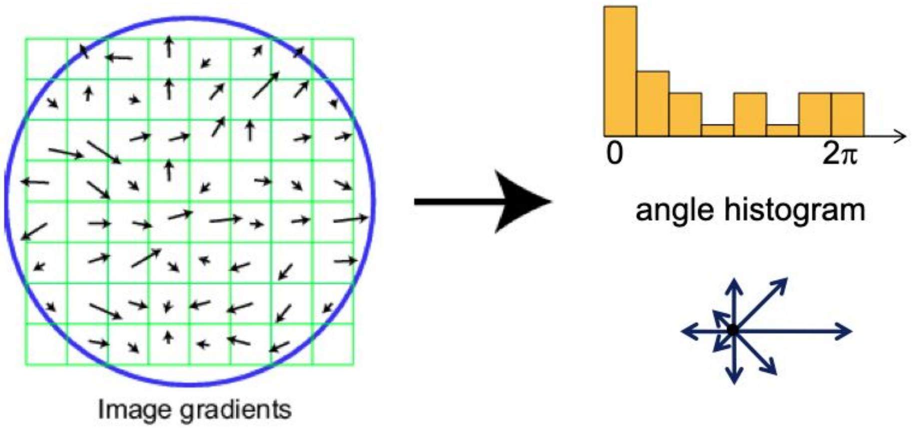
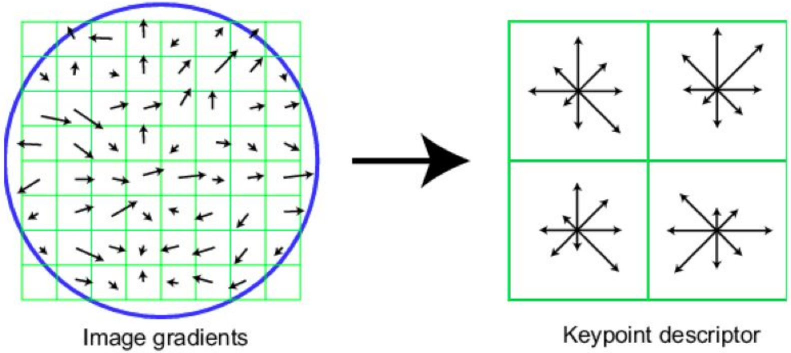


Figure 5: This figure shows the stages of keypoint selection. (a) The 233x189 pixel original image. (b) The initial 832 keypoints locations at maxima and minima of the difference-of-Gaussian function. Keypoints are displayed as vectors indicating scale, orientation, and location. (c) After applying a threshold on minimum contrast, 729 keypoints remain. (d) The final 536 keypoints that remain following an additional threshold on ratio of principal curvatures.

Keypoint descriptor

- Take 16x16 square window around detected feature;
- Compute edge orientation (angle of the gradient - 90°) for each pixel;
- Throw out weak edges;
- Create histogram of surviving edge orientations;
- For SIFT, 8 orientations x 4x4 histograms array results in 128 dimensions.





ORB

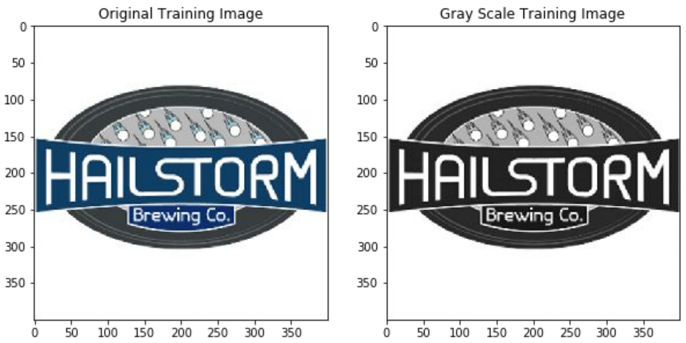
ORB (Oriented FAST and Rotated BRIEF)

- A very fast algorithm that creates feature vectors from detected keypoints
- Invariant to rotations, changes in illumination, and noise

Locate all key points in the training image → ORB creates their corresponding binary feature vectors → Binary feature vectors are grouped together in the ORB descriptor

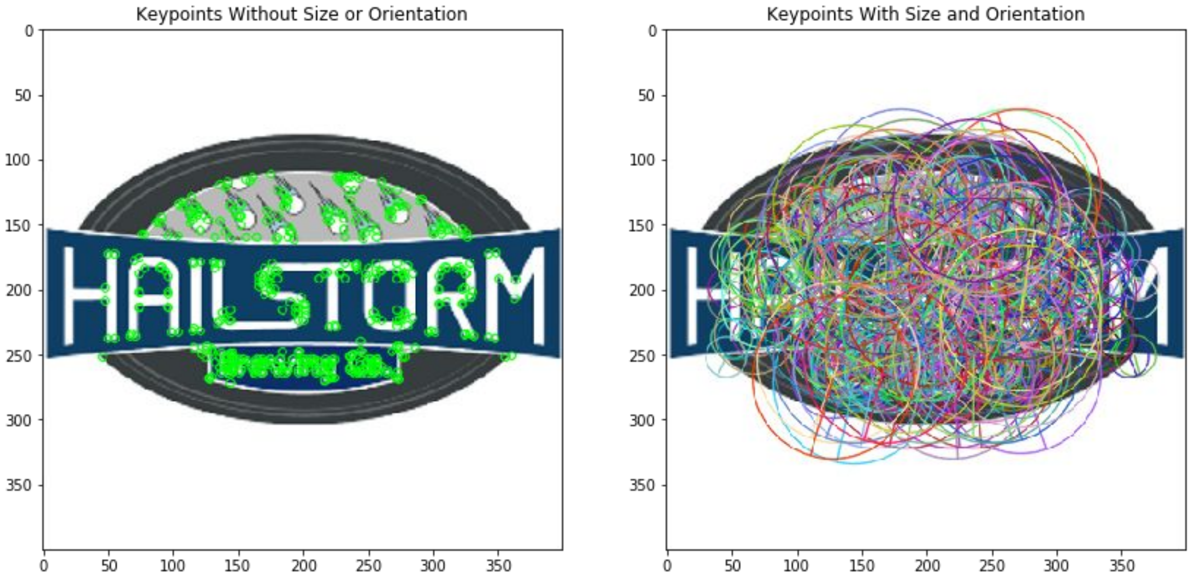
ORB (Oriented FAST and Rotated BRIEF)

Convert query image to grayscale → Initialize the ORB detector and detect the key points in query image and scene → Compute the descriptors belonging to both the images → Match the keypoints using Brute Force Matcher



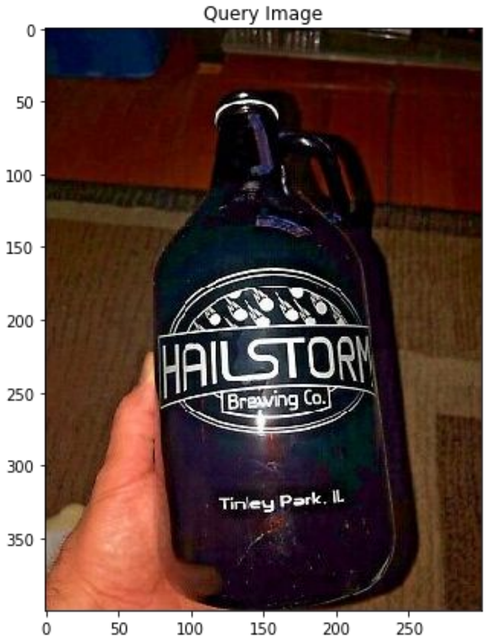
Locating Keypoint

- Right image, every keypoint has a center, a size, and an angle.
- Once the keypoints for the training image have been found and their corresponding ORB descriptor has been calculated, the same thing can be done for the query image.



Feature Matching

- The final step is to perform keypoint matching between the two images using their corresponding ORB descriptors.



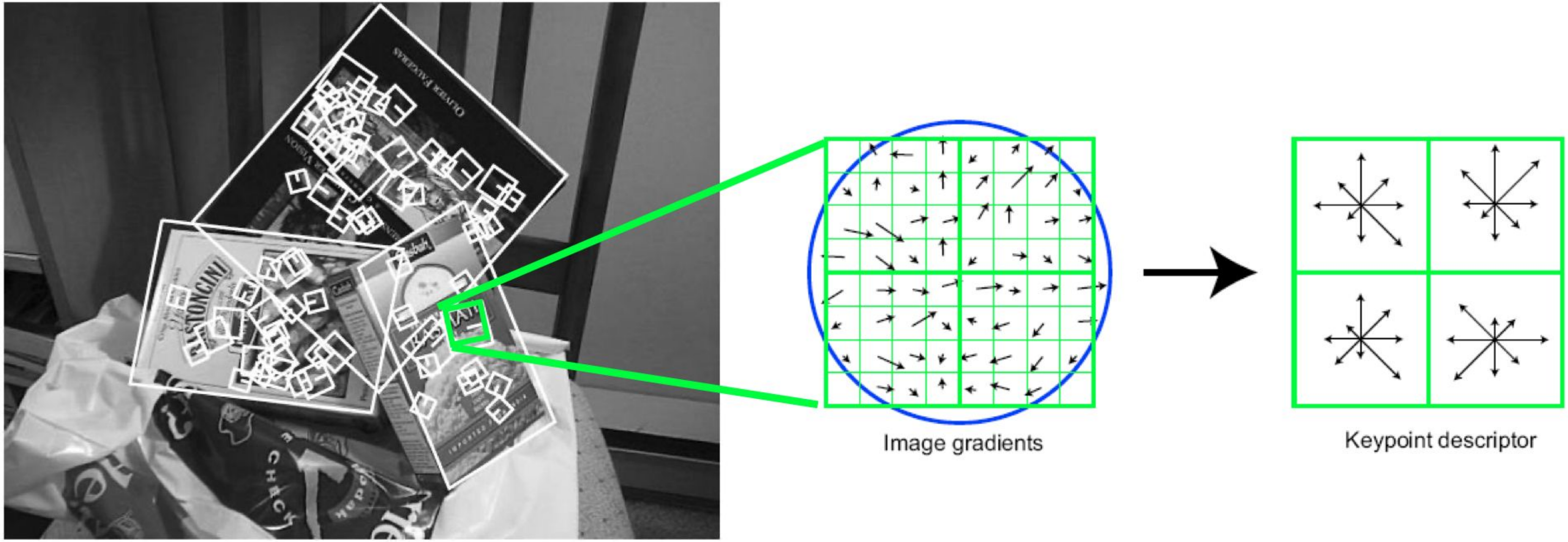
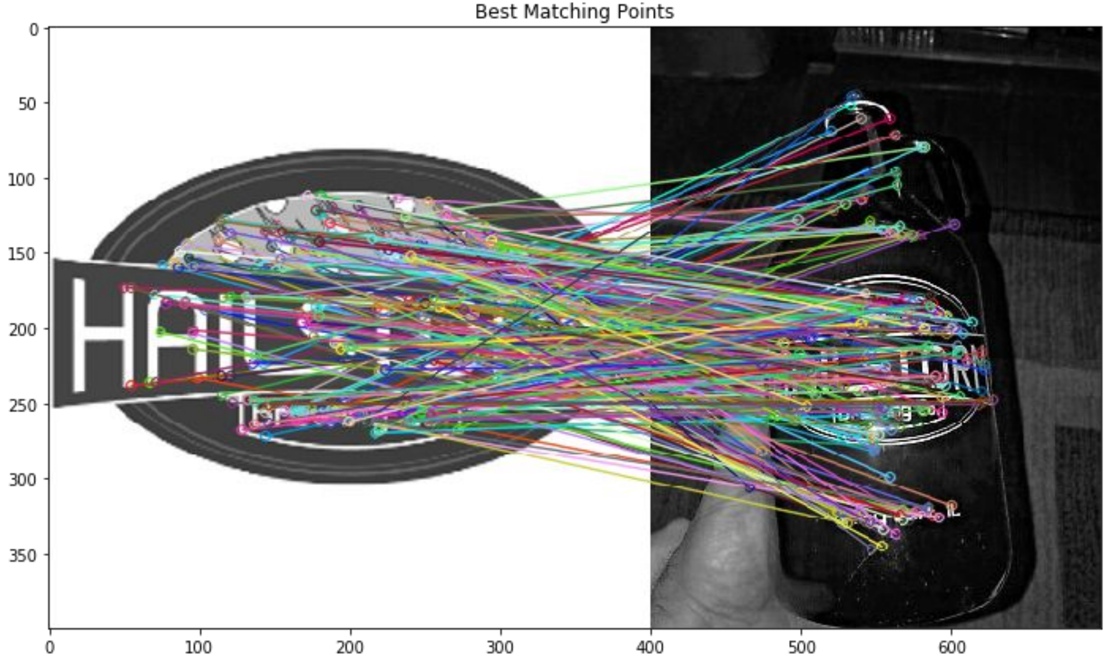


Image gradients
Keypoint descriptor

Feature Matching

- This *matching* is usually performed by a matching function.
- One of the most commonly used matching functions is called *Brute-Force*.



Results

Quantitative Results

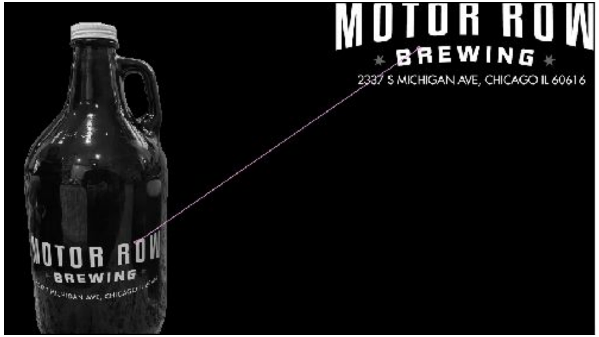
Algorithm	Dataset	Percent Accuracy
SIFT	Primary	100
SIFT	Samuel Adams	100
ORB (500 features)	Primary	31
ORB (500 features)	Samuel Adams	30
ORB (50K features)	Primary	32

SIFT Qualitative Results: Primary Dataset

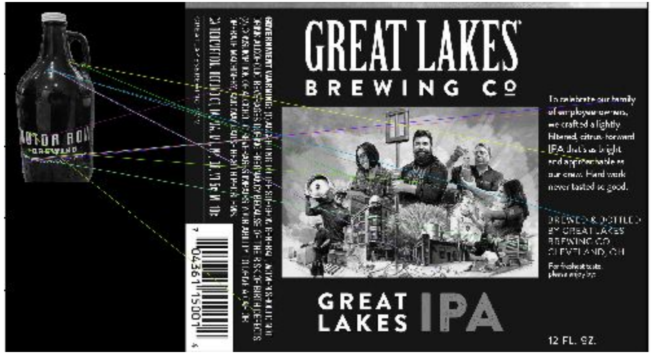
SIFT Qualitative Results: Samuel Adams

ORB Qualitative Results

Correct Matches



Incorrect Match



Actual match for this query

Why is SIFT performing better than ORB?

SIFT

ORB

- In case of **correct** label image, SIFT is able to match more number of points accurately as compared to ORB.
- Being a binary descriptor, ORB focuses on speed whereas, SIFT focuses on precision[1].

[1] https://stackoverflow.com/questions/37617023/why-orb-is-not-as-good-as-sift-in-matching-for-these-two-images

Why is SIFT performing better than ORB?

SIFT

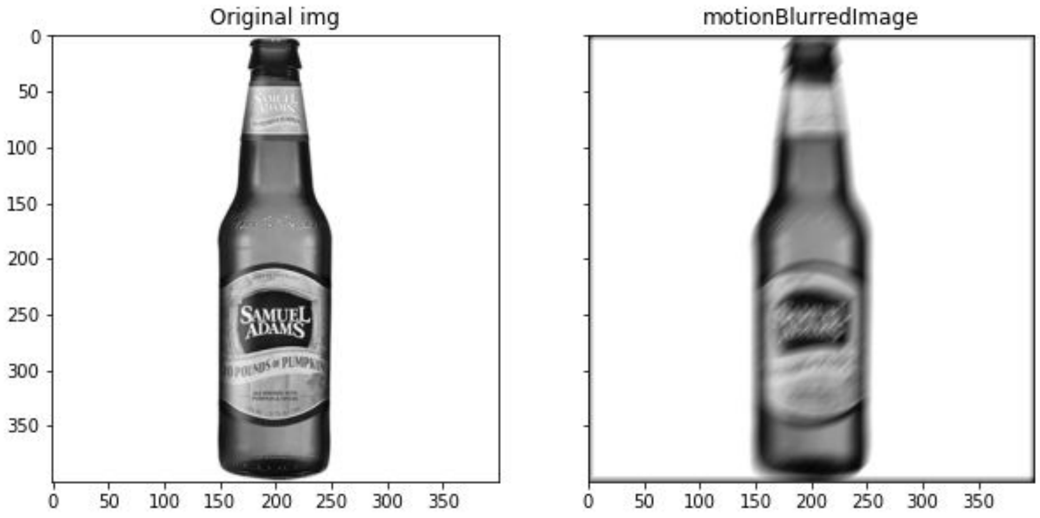
ORB

- In case of **incorrect** label image, SIFT matches very less number of points as compared to ORB.
- As shown in [1], ORB is more sensitive to noise as compared to SIFT.
- A smaller descriptor of ORB size can also be a possible reason for such results.

[1] Helia Sharif, Matthew, A comparison of prefilters in ORB-based object detection, Pattern Recognition Letters, 2017.

Camera motion vs SIFT

- Camera motion was simulated for testing robustness of the SIFT algorithm.



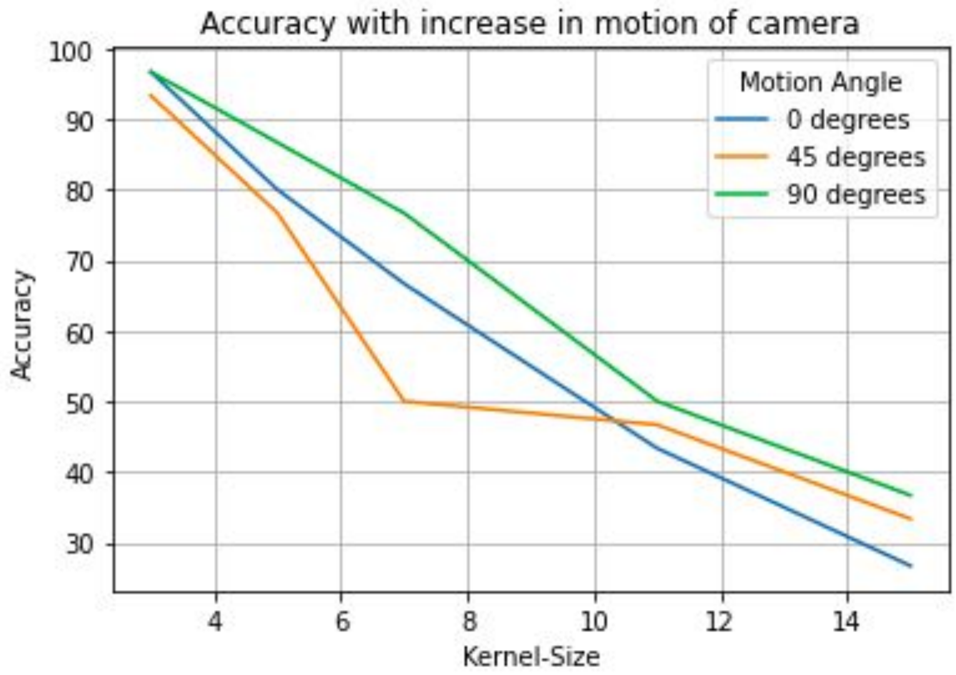
Original image and simulated image with camera motion of 11 pixels at an angle of 45°.

$$\frac{1}{5}\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Matrix used for generating vertical blur.

https://www.geeksforgeeks.org/opencv-motion-blur-in-python/

Camera motion vs SIFT



As the pixels of simulated camera motion increase

Accuracy of the model decreases

Future Work and Conclusion

Future work and Conclusion

- Update the dataset with more noisy images
- Analyse other algorithms like SURF
- Narrow down cases where ORB performs better than SIFT

- As claimed, SIFT proves to be rotation and scale invariant
- On our dataset, SIFT performs better as compared to ORB
- SIFT also proves to be more robust to noise and more descriptive as compared to ORB
- However, SIFT does not work well when camera motion is taken into account

Thank you and cheers 🍻!

Contributions

Name	Tasks Done
Madhav Agarwal (2020900022)	Dataset (P: 60%, SA: 50%), SIFT (20%), Visualization, End-to-end Pipeline, Demo, Documentation (README), Presentation.
Siddhant Bansal (2019900091)	Dataset (SA: 50%), SIFT (80%), Feature Matching, ORB (30%) Visualization, Demo, Documentation (README), Presentation.
Garima Nishad (2019701029)	Dataset (P:20%), ORB (70%), Presentation.
Yallamanda Mundru (2019201029)	Dataset (P:20%), Camera Motion, Presentation