ROTATIONAL INVARIANCE IN IMAGE RECOGNITION

Report Submitted in Partial Fulfilment of the Requirement for the Degree

of

Bachelor of Technology

By

Sanghamitra Hota

Redg. Number: 1501106521

Under the Guidance of

Mrs. Jyotirmayee Routray

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

COLLEGE OF ENGINEERING AND TECHNOLOGY,

BHUBANESWAR

2018

CERTIFICATE

This is to certify that the project entitled **"Rotational Invariance in Image Recognition"** was carried out by **Sanghamitra Hota**(College of engineering and technology, Bhubaneswar) at bearing Registration No: 1501106521, to the department of Computer Science and Engineering, under our supervision and we consider it worthy of consideration for a partial fulfillment f the requirements for the completion of seminar.

DR.SUBHASISH MOHAPATRA

HOD, CSE

CET, BHUBANESWAR

(MRS. JYOTIRMAYEE ROUTRAY)

LECTURE, DEPT OF CSE

CET, BHUBANESWAR

DECLARATION

I do hereby declare that the project entitled **"Rotational Invariance in Image Recognition"** has been originally done under the guidance of **Mrs Jyotirmayee. Routray,** Department of Computer Science and Engineering, College of Engineering and Technology (CET), Bhubaneswar in fulfilment of my Seminar.

ACKNOWLEDGEMENT

It is great pleasure to express my gratitude and words of appreciation to the people who have been, in various ways, the source of help, inspiration and encouragement in my life. I express my heart felt gratitude to Dr Subasish Mohapatra, Head of Department of Computer Science and Engineering and my guide Mrs. Jyotirmayee Routray for giving me an opportunity to work in her group. I thank her for the guidance and motivation that she provided to me throughout the report writing. It has been a good learning experience to work with her.

TABLE OF CONTENTS

ABBREVIATIONS AND SYMBOLS

SIFT: Scale Invariant Feature Transformation

DoG: Difference of Gaussians

σ: Scale of the key point (standard deviation of the smallest Gaussian used in DoG)

L(x,y,σ): Laplacian function

G(x,y,σ): Gaussian function

I: Image

D(x): DoG function

H: Hessian Matrix

Tr(H): Trace of Hessian Matrix

Det(H): Determinant of Hessian Matrix

m(x, y): Magnitude of key point

θ(x, y): Orientation of key point

LIST OF FIGURES

ABSTRACT

Every image has its own features and for image recognition ,we need to extract those features. Further, an image can be rotated through any angle and for this; rotational invariance techniques are used. For getting the best technique, we are analyzing the feature points and its orientation techniques. In order to extract the required features we are using scale invariant feature transform (SIFT). In this technique, each feature is converted into a descriptor array and that descriptor is fed into the neural network for further process. The main idea is to analyze the orientation based on gradient information.

1. INTRODUCTION

We know the digital image is a matrix of pixel intensity values. In order to get accurate results it is wise to use features of the image instead of taking raw patches of images for machine learning. In order to extract features many methods can be employed which has its own pros and cons. Keeping in mind the aspects of project, SIFT is used to extract the features. SIFT features are invariant to scale, rotation, affine transformation, change in 3D view point and change in illumination. These features are highly distinctive and can be used for image recognition. An important aspect of SIFT is that, it generates a large number of features that densely covers the image range of scales and locations. The feature which makes this technique flexible is key point descriptor which allows a single feature to find its correct match with good probability in large database of features. After finding the feature vectors it can be fed into a neural network. The neural network will be trained with the feature vector of training set of images and tested over another set of images.

2. STEPS INVOLVED

The whole process consists of following steps:

- Extracting of features
- Clustering the features
- Training the neural network

3. EXTRACTING OF FEATURE

In order to extract the features from an image we are using SIFT techniques. [1]

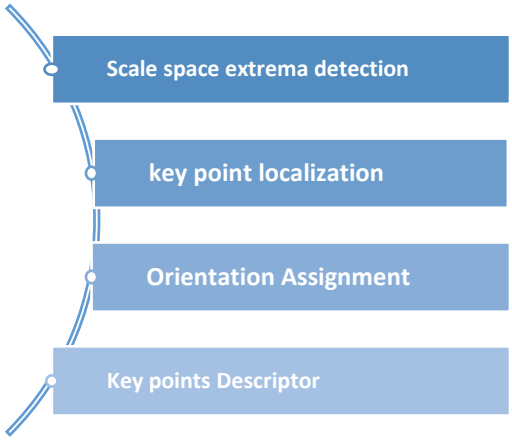


Figure 1: Steps involved in extracting the feature

3.0. FEATURES

The detection and description of local image features can help in image recognition. The SIFT features are local and based on the appearance of the object at particular interest points and are invariant to image scale and rotation. The advantage of extracting local features is making it robust to occlusion and clutter (no prior segmentation). They are easy to extract and allow for correct image recognition with a low probability of mismatch.

The steps involved in SIFT are as follows.

3.1. SCALE SPACE EXTREMA DETECTION

Real world objects are meaningful only at a certain scale. We might see a sugar cube perfectly on a table. But if we look at the entire Milky Way, then it simply does not exist , we aren't able to see the size perfectly. This multi-scale nature of objects is quite common in nature and a scale space attempts to replicate this concept on digital images i.e. how an object seems so small when it is far away but while near it is so prominently visible.

3.1.1 SCALE SPACE

Let us say we want to detect the key points from a monument image, there the visitors might act as false detection points which we do not want to add in our dataset. Hence a proposed way of eliminating the unwanted features is by a Gaussian blur.[8][2] Gaussian blur is the convolution of Image with a Gaussian functions given by equation (1):

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}} \text{---------- (1)}$$

Therefore, the scale space of an image is defined as a function, L(x, y, σ), that is produced from the convolution of a variable-scale Gaussian, G(x, y, σ), with an input image, I(x, y):

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \text{---------- (2)}$$

Where x, y are the coordinates and σ is the standard deviation of Gaussian function.

Why a Gaussian filter?

The purpose of choosing Gaussian kernel instead of other filters is that it is rotationally invariant. Gaussian function convolved with the image give us more stable features than a lot of other possible functions such as gradient, Hessian or Harris corners. [1]

SIFT takes scale space to a new level. We first consider the original image, take the Gaussian blur of the image and progressively increase the amount of blur in multiple of some constant factor k over σ (In Lowe's paper [1] the author took the value of constant factor to be k = 1.414). After that the image is down-sampled by half, and again successive blurred out images is produced. The convolved images of same size form an octave. Down sampling helps to reduce the number of pixels which in other words tends to keep the size of Gaussian small, and saves computational time. According to Lowe's paper [1] it is sufficient to have four octaves and five scale level for one image.

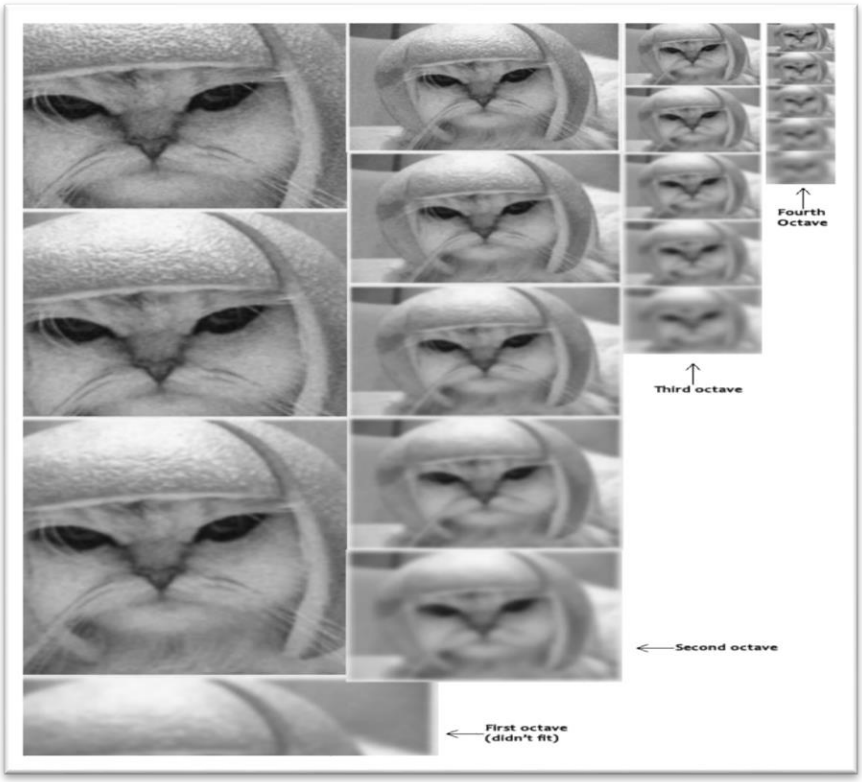


Figure 2: Octaves and increasing blur levels.
(Retrieved from http://aishack.in/tutorials/sift-scale-invariant-feature-transform-scale-space)

3.1.2. DIFFERENCE OF GAUSSIANS

In imaging science, difference of Gaussians is a feature enhancement algorithm that involves the subtraction of one blurred version of an original image from another, less blurred version of the original. Subtracting one image from the other preserves spatial information that lies between the ranges of frequencies that are preserved in the two blurred images.

Thus, the difference of Gaussians is a band-pass filter that discards all but a handful of spatial frequencies that are present in the original grayscale image.[3]

In this step we subtract the Gaussian blurred images of different scales of the same octaves. The formula is stated below:

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y)) .$$
$$= L(x, y, k\sigma) - L(x, y, \sigma) \text{ .----------(3)}$$

This steps reduces the calculation as subtraction is always easier than differentiation.[4]

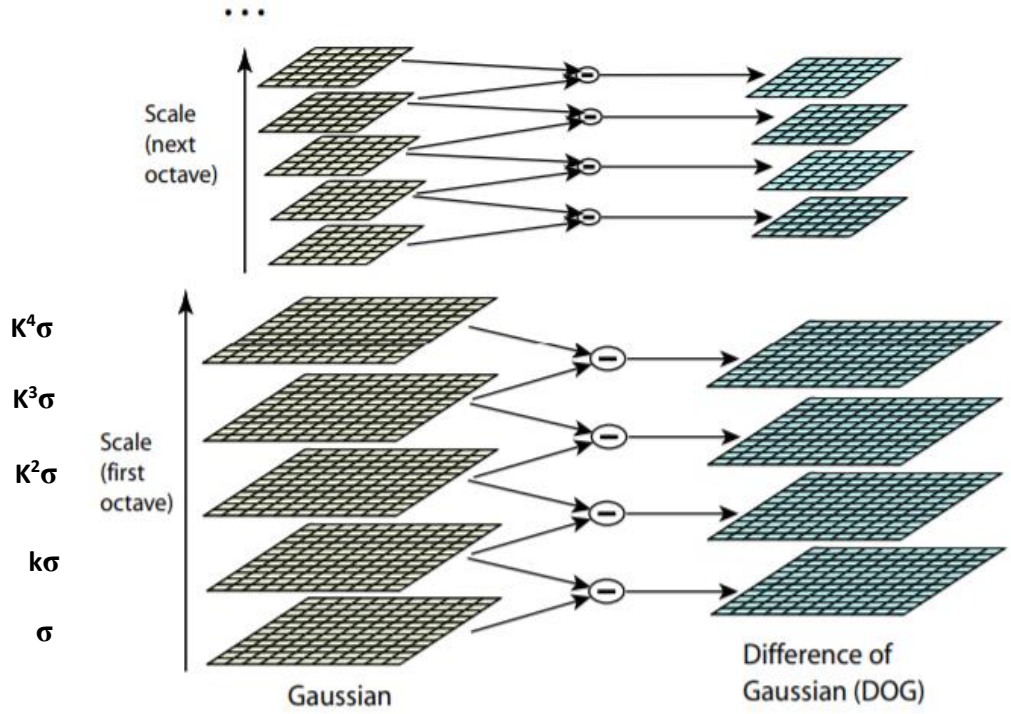


Figure 3: For each octave of scale space, the initial image is repeatedly convolved with Gaussians to produce the set of scale space images shown on the left. Adjacent Gaussian images are subtracted to produce the difference-of-Gaussian images on the right. After each octave, the Gaussian image is down-sampled by a factor of 2, and the process repeated (Retrieved from Lowe, D.G., 2004. Distinctive image features from scale-invariant key points. International journal of computer vision)

Here we simply subtract the pixel value of one layer from its consecutive layer to get the pixel value of the resulting DoG.

The common drawback of the DoG representations is that the local extrema can also be detected in neighboring contours of straight edges, where the change is only in one direction, which make them less stable and more sensitive to noise or small changes.[5]

3.1.3. LOCAL EXTREMA DETECTION

In this step each sample point is compared to its eight neighbours in the current DoG and nine neighbors at exact location in the scale above and below it. It is selected as a candidate key point only if it is larger than or smaller than all these neighbors. In this process we might get extrema that are proximately close to each other, and are highly unstable to small changes in the image. [1]

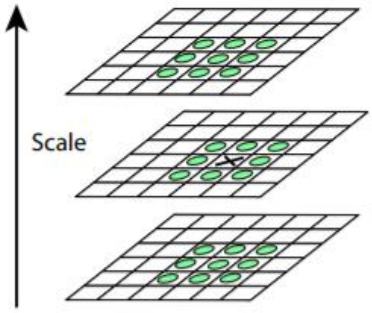


Figure 4: Near 'X', a 3X3 neighborhood is taken then it is matched with the pixels above below and around it.(Retrieved from Lowe, D.G., 2004. Distinctive image features from scale-invariant key points. International journal of computer vision)

3.2. KEYPOINT LOCALIZATION

In this step after finding the candidate's key points, a fit is performed to the nearby data for much accuracy in location, scale, ratio of principle curvature. (As we have a DoG response map, we treat the DoG response around a pixel as surface and compute its bend which should be high on both perpendicular sides.)

3.2.1. SUBPIXEL DETECTION

It is not always possible to find the exact positions of candidate key point. The exact extrema never lies on the pixel but mostly lies in between the pixels. For example, in fig 5.5:

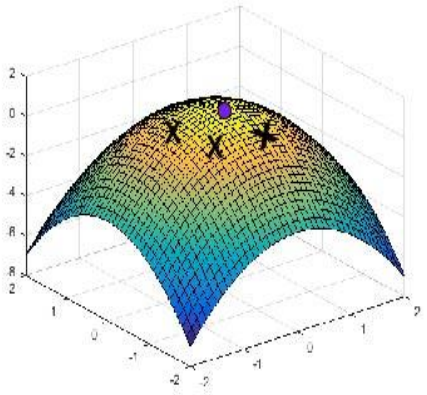


Figure 5: Sub pixels detection

In this the actual extrema is the colored point but in the previous step we got the "X" marked position as the extrema.

In order to find the sub pixels mathematically, Brown developed a method which uses the Taylor's expansion truncated at second order.[6]

$$D(x) = D + \frac{dD^T}{dx} x + 0.5\, x^T \frac{d^2D}{dx^2} x \text{ .----------(4)}$$

Where D(x) is the DoG function, DoG (x, y,σ) and x = (x, y, σ)^T is the offset from this point. The location of the extremum $\hat{\boldsymbol{x}}$,Is determined by differentiating and setting it to zero. If the offset is larger than 0.5 in any dimension, then the extremum lies closer to another candidate key point. In this case, the candidate's key point is changed.

3.2.2. ELIMINATIN GLOW CONTRAST POINTS

Due to re-iterated Gaussian filtering, many extrema exhibit small value to contrast. To discard the keypoints with low contrast, the value of the second-order Taylor expansion D($\hat{\boldsymbol{x}}$) (equation given below) is computed at the offset $\hat{\boldsymbol{x}}$. If this value is less than 0.03(as in Lowe's paper), the candidate key point is discarded.

$$D(\hat{x}) = D + \frac{1}{2}\frac{dD^T}{dx}\hat{X} \text{ .----------(5)}$$

3.2.3. ELIMINATING EDGE RESPONSES

Poorly defined key points have high edge responses. For poorly defined peaks in the DoG function, the principal curvature across the edge would be much larger than the principal curvature along it. In this step the idea is to find the gradients in two directions both being perpendicular to each other. There can be three possibilities:

Flat Region: both the gradient will be small.

Edge Region: The gradient along the edge is small and perpendicular to the edge is big.

Corner Region: Both the gradient is big.

Mathematically, it is done by calculating the principal curvatures amounts from the eigenvalues of the second-order Hessian matrix of DoG.[7][3]

$$H = \begin{bmatrix} D_{XX} & D_{XY} \\ D_{XY} & D_{YY} \end{bmatrix} \text{.----------(6)}$$

The Eigen value of 'H' is proportional to principal curvature. Let α be the eigen value with the largest magnitude and β be the smaller one. Trace and determinant will be calculated as follows:

$$Tr(H) = D_{XX} + D_{YY} = \alpha + \beta \text{ .----------(7)}$$

$$Det(H) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta \text{ .----------(8)}$$

As we are concerned with ratio of Eigen values instead of concrete Eigen values, it is attractive to use Tr (H) and Det (H) in the formulation.

If the determinant is negative, the extremum is discarded (which does not happens generally). Let r be the ratio between the largest magnitude eigen value and the smaller one, so that α = rβ. The inspiration for getting the edge response is derived from Harris corner detector's equation i.e.R = Det (H) – k Tr (H) ^{2}where R is the corner response and k is an empirical constant.[7]

$$\frac{\mathrm{Tr}(H)^2}{\mathrm{Det}(H)} = \frac{(\alpha+\beta)^2}{\alpha\beta} = \frac{(r\beta+\beta)^2}{r\beta^2} = \frac{(r+1)^2}{r} \text{ .----------(9)}$$

Therefore, to check that the ratio of principal curvatures is below some threshold,

$$\frac{\mathrm{Tr}(H)^2}{\mathrm{Det}(H)} < \frac{(r+1)^2}{r} \text{ .} \qquad \text{---------- (10)}$$

In Lowe's paper he took the value of r as 10 which gave him some stable results.

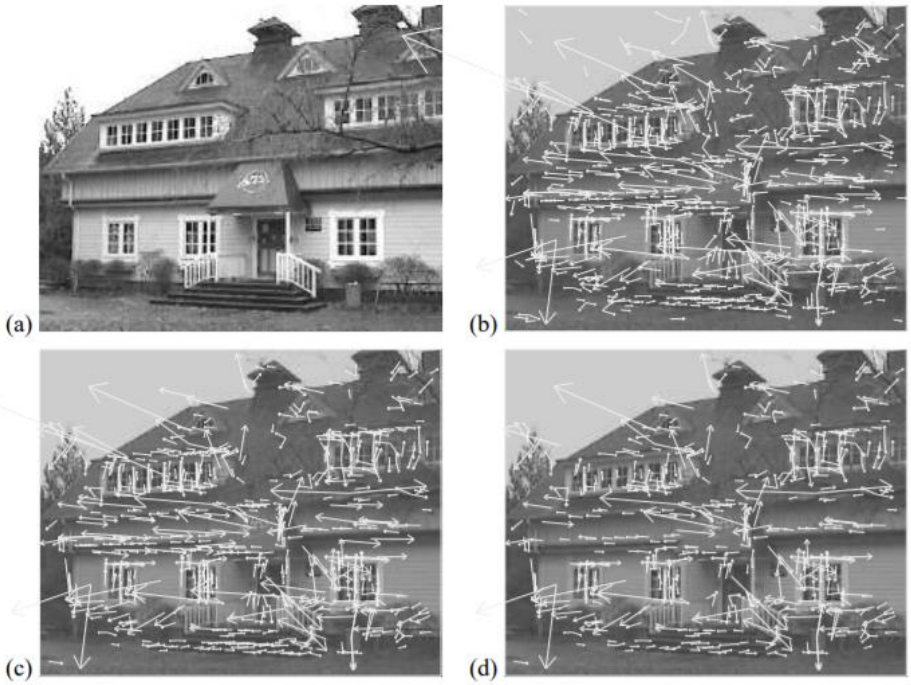


Figure 6: stages of key point selection. (a) The 233x189 pixel original image. (b) The initial 832 key point's locations at extrema of the DoG function. (c) 729 key points remain, after removal of low contrast key points (d) the final 536 key points after removal of edge responses. (Retrieved from Lowe, D.G., 2004. Distinctive image features from scale-invariant key points. International journal of computer vision)

This drastic fall in key points shows how important it is to remove the false points.

3.3. ORIENTATION ASSIGNMENT

In this step, each key point is given some consistent orientation. This is the key step in achieving invariance to rotation as the keypoint descriptor can be represented relative to this orientation and therefore achieve invariance to image rotation.

For each image sample, L(x, y), at this scale, the gradient magnitude,m(x, y), and orientation, θ(x, y), is pre computed using pixel differences:

$$m(x,y) = \sqrt{\big(L(x+1,y) - L(x-1,y)\big)^2 + \big(L(x,y+1) - L(x,y-1)\big)^2}\,. \qquad \text{---------(11)}$$

$$\theta(x,y) = tan^{-1}((L(x,y+1) - L(x,y-1)/(L(x+1,y) - L(x-1,y)))))\,. \qquad \text{---------(12)}$$

A neighborhood N around each key point is considered. The orientation of the gradient of the points in N is represented by a histogram with 36 bins. The peak of histogram is assigned to (x, y, σ), so that the key point is described now by a vector (x, y, σ, θ), where q is the orientation of the peak of histogram. For example: the gradient direction at a certain point (in the "orientation collection region") is 28.759 degrees, then it will go into the 20-29 degree bin. And the "amount" that is added to the bin is proportional to the magnitude of gradient and to a Gaussian-weighted circular window with σ that is 1.5 times that of the scale of the key point.

The highest peak in the histogram is detected, and then any other local peak that is within 80% of the highest peak is used to also create a key point with that orientation. The new key point will have the same scale and location but its orientation will be the other peak.

3.4. KEY POINT DESCRIPTOR

In this step a unique finger print is found for each key point. In this step it achieve invariance to illuminance and partially to affine transformation. A 16×16 neighborhood is taken around the key point. This 16×16 is broken into sixteen 4×4 windows. Within each 4×4 window, gradient magnitude and orientations is calculated. Then histogram of 8 bins each is created using the set of orientation. And the amount added to the bin depends on the magnitude of the gradient. The magnitudes are further weighted by a Gaussian function with σ equal to one half the width of the descriptor window.

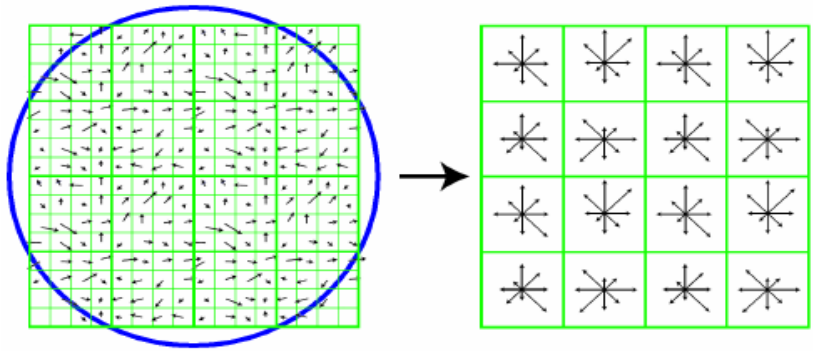


Figure 7: key point Descriptor - 4x4 descriptors computed from a 16x16 sample array (Retrieved from Lowe, D.G., 2004. Distinctive image features from scale-invariant key points. International journal of computer vision)

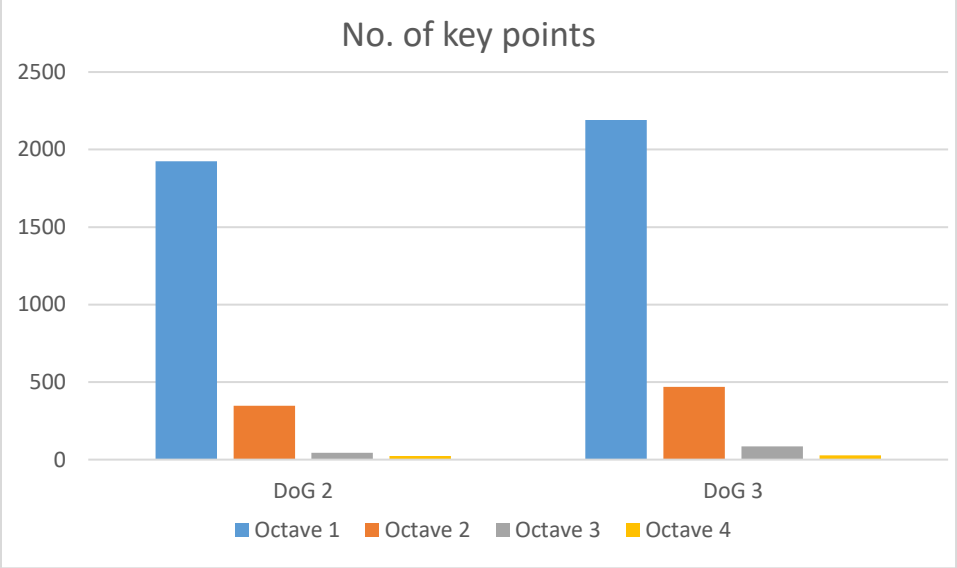
The descriptor then becomes a vector of all the values of these histograms. Since there are 4 x 4 = 16 histograms each with 8 bins the vector has 128 elements. This vector is then normalized to unit length in order to enhance invariance to affine changes in illumination.

When we rotate the sample, the gradient orientation will also change. To achieve rotation independence, the key point's rotation is subtracted from each orientation. Thus, each gradient orientation is relative to the key point's orientation. In order to achieve stability to illumination changes, a two-step normalization process is used. The key descriptor is first normalized to unit length then any element larger than 0.2 is clipped to 0.2. The clipped vector is renormalized to unit length as final SIFT descriptor.

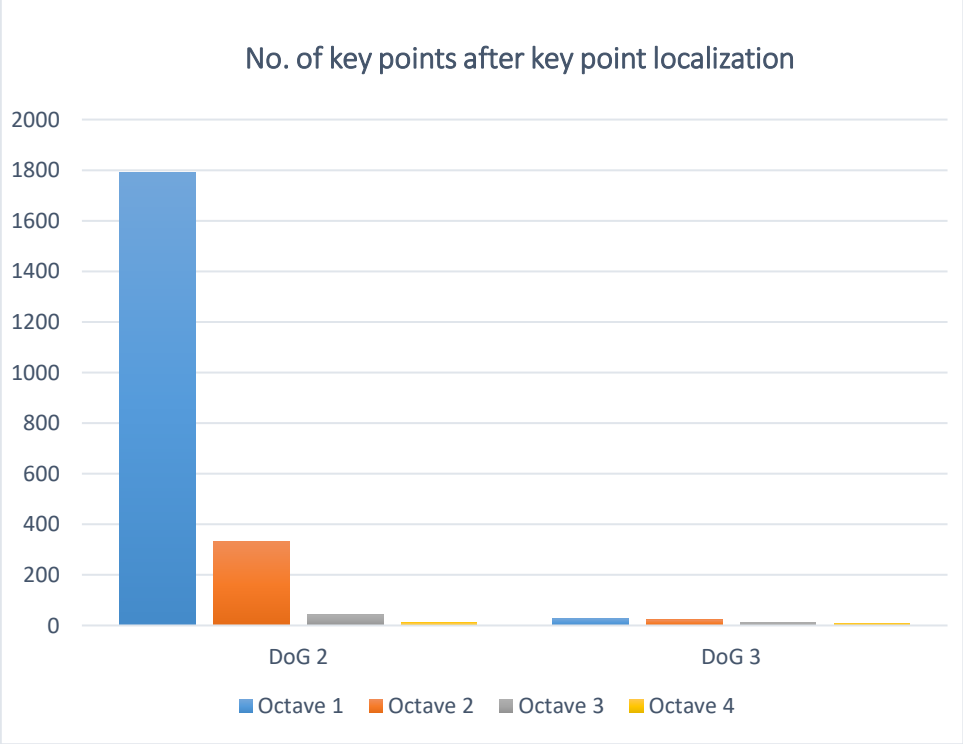
From the above procedure, we can observe that SIFT actually computes a 3D histogram of spatial location and gradient orientations, weighted by gradient magnitude and Gaussian function.

4. RESULTS AND DISCUSSION

- For simplicity, the value of scale parameter was taken 1.6.
- In first octave, the DoG matrix doesn't show much difference in pixel change, so it gives a lot of false key points if we decrease the threshold and no key points if we increase it.
- The number of key points for a 50×50 pixels of a **4-digit** image in DoG2 for first octave is 1924 whereas for second octave, the number of key points fall to 374. Hence, it is better to consider the key point descriptors of second and third octave for further computations.



- The number of key points decreases even more after eliminating the edge response and



removal of low contrast points. For DoG2 of second octave, the key points decreased from 374 to 332.

- In order to achieve rotational invariance, a 10×10 neighborhood around each key point was taken. The histogram for each key point is different from other.
- In order to check if SIFT works for complex image, an image of a building was taken. The red circles are the key points found from SIFT. The number of key points found in figure 9 is 1112.

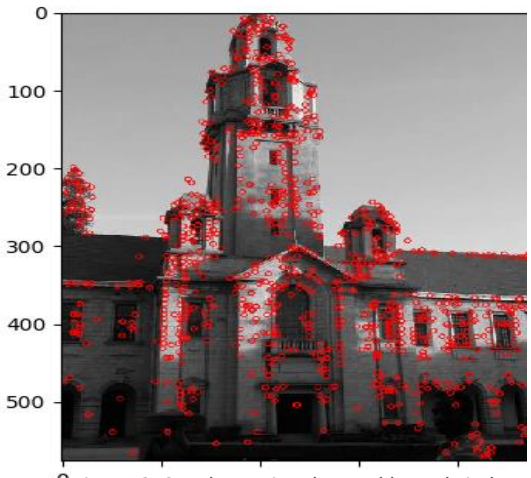


Figure 9: SIFT key points located by red circles.

- For feature matching in two images, FLANN based matching was used.

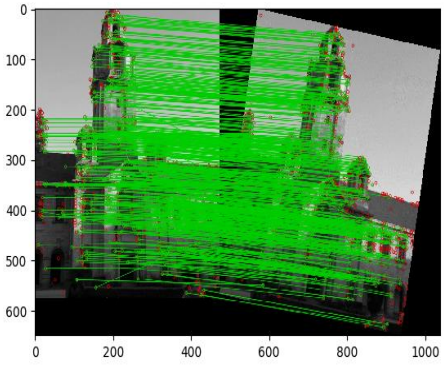


Figure A

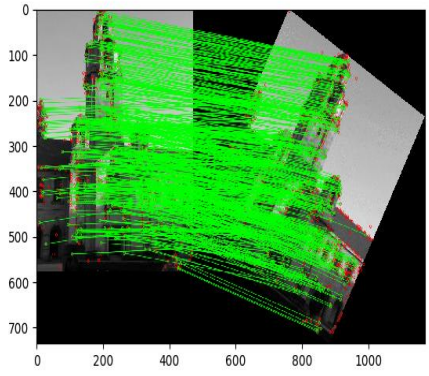


Figure B

- In figure A and figure B, the test image was rotated by 10 degrees and 30 degrees respectively, then the feature matching was done. In SIFT it handled rotational invariance

properly with a lower probability of mismatch.(Green ones shows the feature matching, red ones shows the unmatched feature.)

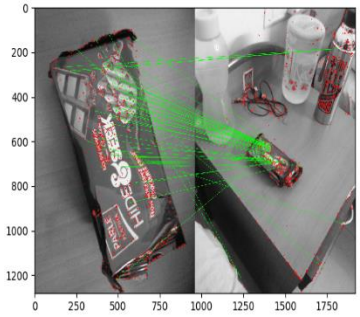


Figure 10: Feature Matching

- In case of object recognition in figure 10, feature matching was done. Out of 548 feature points, 360 were matched properly. It showed an error of 34 % (approx.) (Green ones show the feature matching and red ones show the unmatched features)

5. CONCLUSION AND FUTURE WORK

The objective of this report is to develop an image recognition system which uses rotation invariant local image features. It introduces the basic notations for detecting and extracting image features, then describes the properties of perfect feature detectors. Furthermore, the feature descriptors are explained in details. Finally, feature matching is done in different rotations, change in view point to check the accuracy of SIFT algorithm.
Finally, we get a 128 dimension vector for each key point. Now, the number of key points varies from image to image. In order to bring the SIFT features to fixed size, clustering is done. After getting the fixed size of vectors, we can train the neural network with feature vectors and use it for image recognition.

REFERENCES

1. Lowe, D.G., 2004. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, *60*(2), pp.91-110.
2. Mark, N. and Alberto, S.A., 2008. Feature extraction and image processing. *Nixon, Mark S. Amsterdam*.
3. Davidson, W. and Abramowitz, M., 2006. Molecular expressions microscopy primer: Digital image processing-difference of gaussians edge enhancement algorithm. *Olympus America Inc., and Florida State University*.
4. Lindeberg, T., 1993. Detecting salient blob-like image structures and their scales with a scale-space primal sketch: A method for focus-of-attention. *International Journal of Computer Vision*, *11*(3), pp.283-318
5. Mikolajczyk, K. and Schmid, C., 2004. Scale & affine invariant interest point detectors. *International journal of computer vision*, *60*(1), pp.63-86.
6. Brown, M. and Lowe, D.G., 2002, September. Invariant Features from Interest Point Groups. In *BMVC* (Vol. 4).
7. Harris, C. and Stephens, M., 1988, August. A combined corner and edge detector. In *Alvey vision conference* (Vol. 15, No. 50, pp. 10-5244).
8. L. G. Shapiro & G. C. Stockman, "Computer Vision", page 137~150. Prentice Hall, (2001).
9. Wagner, D., Reitmayr, G., Mulloni, A., Drummond, T. and Schmalstieg, D., 2008, September. Pose tracking from natural features on mobile phones. In *Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality* (pp. 125-134). IEEE Computer Society.
10. Lowe, D.G., 1999. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on* (Vol. 2, pp. 1150-1157).
11. Toews, M., Wells, W., Collins, D.L. and Arbel, T., 2010. Feature-based morphometry: Discovering group-related anatomical patterns. *NeuroImage*, *49*(3), pp.2318-2327.
12. Cui, Y., Hasler, N., Thormählen, T. and Seidel, H.P., 2009, July. Scale invariant feature transform with irregular orientation histogram binning. In *International Conference Image Analysis and Recognition* (pp. 258-267). Springer, Berlin, Heidelberg.