LEX LOCALIS-JOURNAL OF LOCAL SELF-GOVERNMENT
ISSN:1581-5374E-ISSN:1855-363X
VOL.23,NO.10(2025)

# BOOSTING THE PERFORMANCE OF THE STATE-OF-ART IMAGE CLASSIFICATION TECHNIQUES

**Aabid Vaqar[1*], Dr. Vijay Dhir[2]**

[1*]Research Scholar, Department of Computer Science & Engineering, Sant Baba Bhag Singh University, Khiala, Punjab, India
[2]Professor, Department of Computer Science & Engineering, Sant Baba Bhag Singh University, Khiala, Punjab, India

[*]Corresponding Author: Email ID: aabid.vaqar@gmail.com[1]

**Abstract**
In the modern times, Query by Image is the fad. With the Digital Image Capture Devices made available at very low cost, the image database has been expanding very rapidly. Challenge of maintaining such a humongous Image Database, searching images from it efficiently, is becoming more and more cumbersome. A well organised database can prove to be a great relief. This paper deals with the initial step of classifying images in a very efficient way to lead to a well organised image database from where query image could be retrieved swiftly. Many state-of-the-art Image Classification Techniques including Harris Corner Detection (HCD), Canny's Edge Detection (CED), Laplacian of Gaussian (LoG), Scale Invariant Feature Transform (SIFT) and Speeded up robust features (SURF) are compared and contrasted. Various algorithms for key point matching viz. Oriented FAST Rotaed BRIEF (ORB), KAZE and Accelerated KAZE etc. neural and models like k-Medoids Clustering, k Nearest Neighbourhood (kNN) and Convolution Neural Network (CNN) etc. and networks having various designs – LeNet, AlexNet, SqueezeNet, GoogLeNet, ResNet etc. are compared and contrasted on the basis of their performances keeping in view various parameters like Precision, Recall and Speed. An improvisation is proposed to boost the performance of this phase on the way to a complete algorithm for content-based image retrieval system.

**Keywords**: *Image Classification, SURF, DoG, CNN, kNN*

## 1. Introduction

The rate at which the image data in sample database is growing is much faster than rapid. In such a scenario, appropriate storage after classification has become major challenge in fast retrieval. Thus, developing a system which could automatically comprehend an image from merely a set of sequenced bits attracted major focus of researchers. The process can be subdivided into various phases. The first phase includes database storage in a systematic way such that access to the images stored is easy and fast. The orthodox way to match images to this database was based on meta-data of these images. Certain keywords which were most relevant to the contents of image were stored in the database. Though searching images through these key features was easy but on the whole first analysing each and every image for these keywords has been a tedious task making the overall content based image retrieval system slow and less accurate. So it is commendable to process these images based directly on their contents. Dominant features like colour, texture and shape have been proving path-breaking in the field. Various algorithms have been since then tried to extract these features and classify the images into various categories. Deep Machine learning was the unanimous choice for automated image classification. Since then, a number of algorithms have been tried for the purpose. This paper presents a comparison among them and proposes a hybrid approach for faster and better image classification.

## 2. Managing Image database through Feature Detection

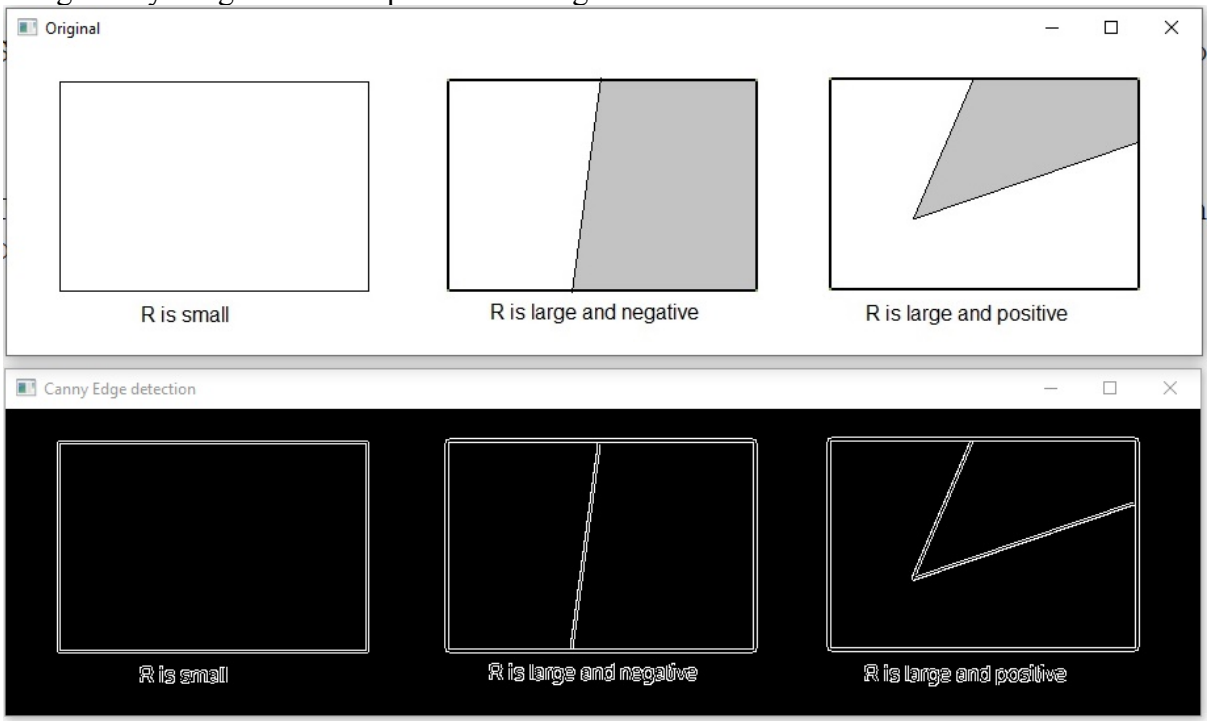
Arrangement of the image database in proper order using appropriate algorithms can facilitate the process of instant search. This arrangement can be made on the basis of features found out of the images in the form of vectors. Over the times the procedure has been evolving with the introduction of newer algorithms.

### 2.1. Canny Edge detection Algorithm

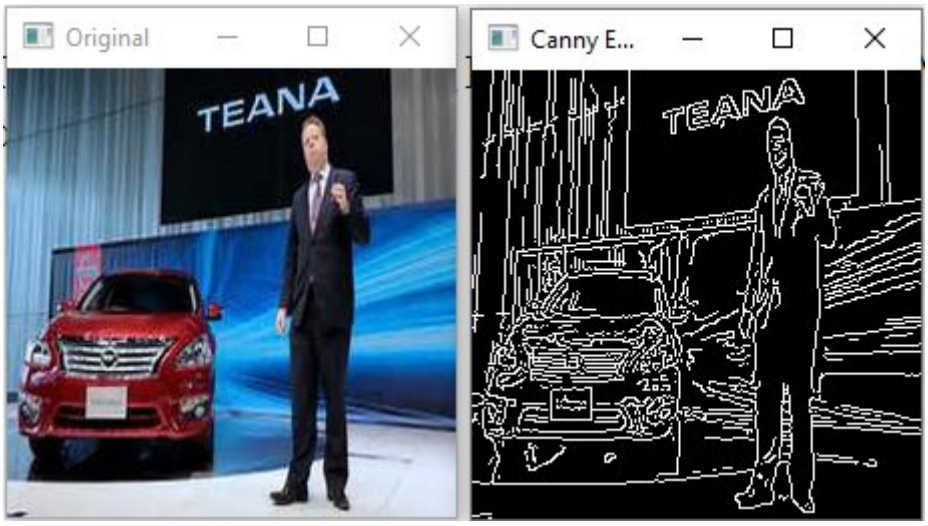
Detecting edges in an image retains only relevant information and discarding the information which is less important with the point of view of image processing. Thus edge detection eventually reduces considerably the data to be processed. Prominent among the algorithm for this purpose is Canny's edge detection Method. An advanced and faster modification proposed by Ruiyuan Liu et al. [1] This algorithm selects the threshold value based on genetic algorithms. The basis of filtering is

$$p_m = \begin{cases} p_{m\,max} \cdot e^{\left[-(f-\frac{f_{avg})}{f_{max}-f_{avg}}\right]} & if\ f > f_{avg} \\ p_{m\,max}\ _{if}\ f < f_{avg} \end{cases}$$

Among them: $f$ is the fitness value of the individual to be mutated; $f_{avg}$ is the average fitness value in the Population; $p_{m\,max}$ is the maximum mutation probability. A precise edge detection using Canny's algorithm is depicted in the figures below.




Another example run of the same algorithm is




## 2.2.Harris Corner Detector

This method is based on considerable change in intensity in some localised neighbourhood as proposed by Javier Sanches et.al. [2] The corner is mathematically detected at the maxima of following function

$$E(h) = \sum G(x)(I(x+h) - I(x))^2$$

where $x$ is the point of interest $h$ is its neighbourhood and $G$ is Gaussian function.

Harris Corner Detection method is based on Eigen values of second moment matrix and involves cumbersome calculations.

In digital form of image, change in intensity along horizontal direction, $I_x$ can be calculated by convolving the image with kernel $[-1 \quad 0 \quad 1]$. Similarly $I_y$ the intensity gradient in vertical direction can be calculated by convolving the image with the kernel $\begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$.

Harris Matrix H is

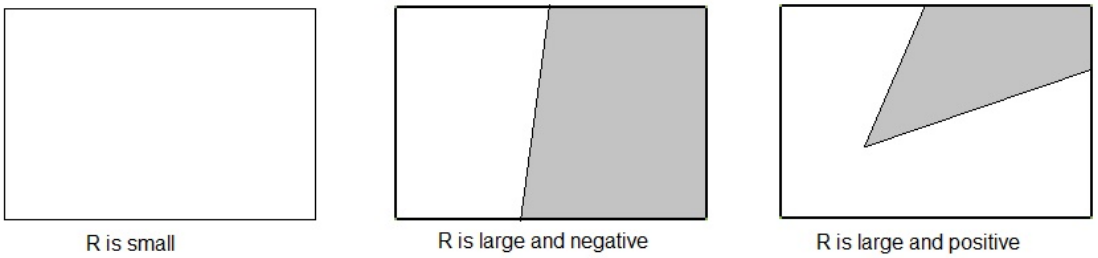
$$H = \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

Then flat, edge or corner detection parameter R is calculated as

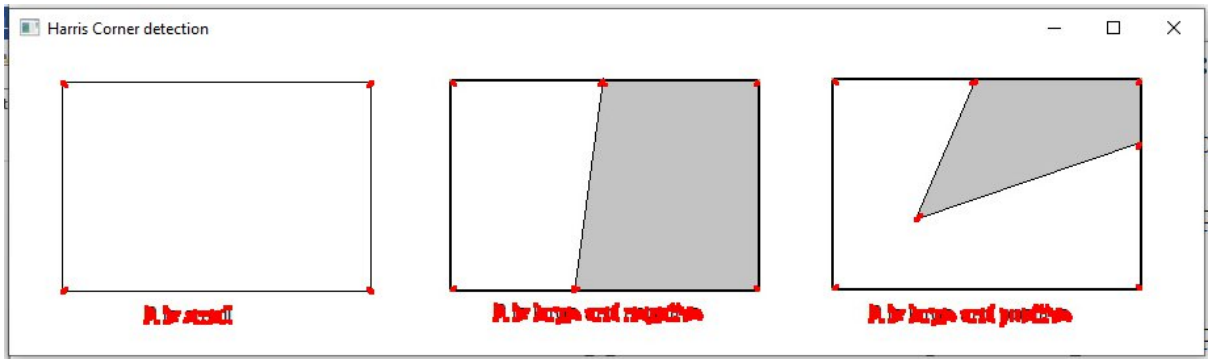
$$R = \det(H) - k \times trace(H)$$

where k is numerical constant.

Depending on the value of R, flat, edge or corner can be detected.




The corners detected in the same image using Harris' Algorithm are as




Another example of Harris Corner Detection is


The significant drawback of this method is that it is not scale invariant.

### 2.3.Laplacian of Gaussian

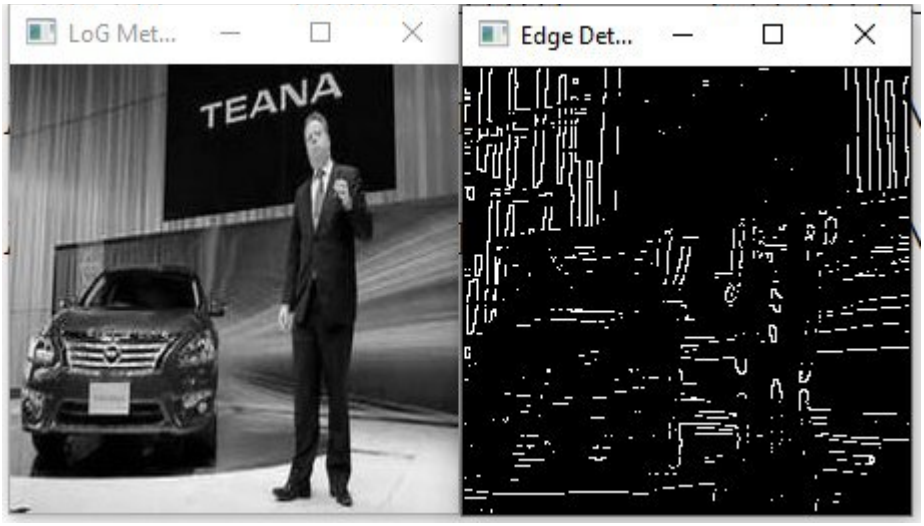
LoG is based on surface fitting technique as proposed by G.E. Sotak et.al. [3] LoG with space constant $\sigma$ at any point $(x,y)$ of an image is given by

$$\nabla^2 G(x,y) = \frac{1}{2\pi\sigma^4}\left(2-\left(\frac{x^2+y^2}{\sigma^2}\right)\right)e^{-x^2+y^2/\sigma^2}$$

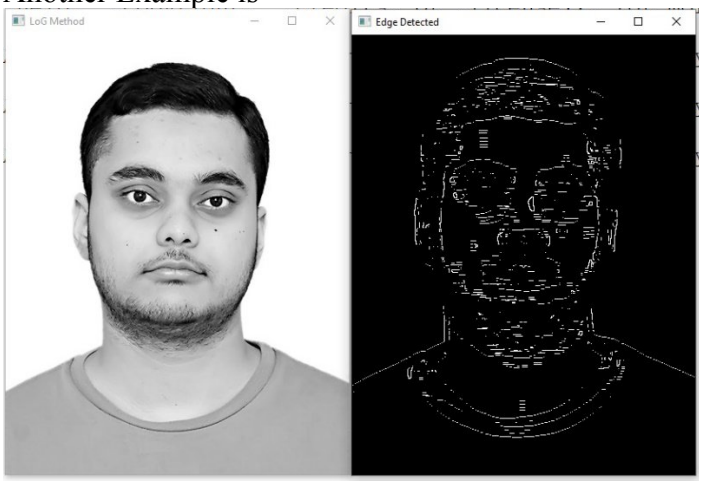
The kernel matrix to be convolved with the image in this case is

$$\begin{bmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & -2 & -1 & 0 \\ -1 & -2 & 16 & -2 & -1 \\ 0 & -1 & -2 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{bmatrix}$$

The performance of LoG Algorithm is indicated as




Another Example is




**2.4 Harris Laplacian detector**:

Harris Laplace detector is just an extension of Harris detector in multi scaled mode thus making it scale-invariant. [4] The structure tensor $\tilde{A}$ is given as

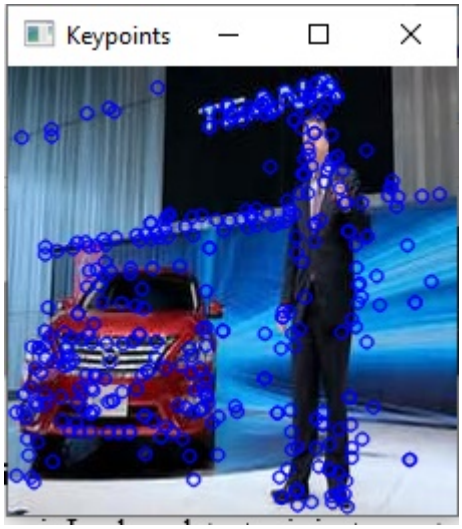
$$\tilde{A} = G(\sigma_I) * \begin{bmatrix} {}^{\alpha}_{x}I^2 & {}^{\alpha}_{x}I\,{}^{\alpha}_{x}I \\ {}^{\alpha}_{x}I\,{}^{\alpha}_{x}I & {}^{\alpha}_{y}I^2 \end{bmatrix}$$

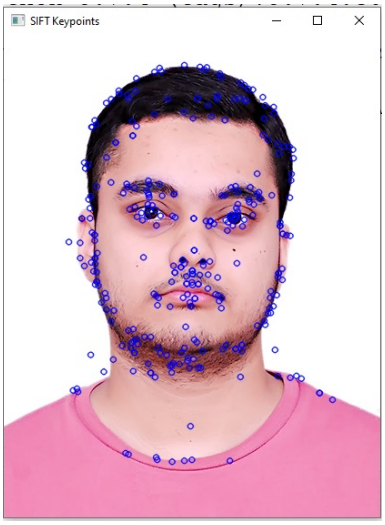
Here ${}^{\alpha}_{x}I$ is the fractional derivative of order $\alpha$ in x direction where as ${}^{\alpha}_{y}I$ is the same in y direction. But this method has a drawback of returning too few corner points.

**2.5 SIFT**

Scale Invariant Feature Transform Algorithm creates scale space after scaling image at many levels and then finding Difference of Gaussian (DoG). Its descriptor is 128 dimensional which is too large and complicates the processing.

A few examples depicting the keypoint (Points of interest) detected using SIFT Algorithm are




## 2.6 The Proposed Algorithm for Edge Detection

As the basis of filtering in Canny's Algorithm is 2D Gaussian function which is continuous in nature, it causes a time delayed performance. It would be wise if it is replaced with Sobel Filters. It would considerably reduce the overhead of very complex calculations. Moreover, as the pixels and their intensity levels being discrete in nature, would fit more into the situation.
The kernels for convolution in x direction is

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

and the same for y direction is

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

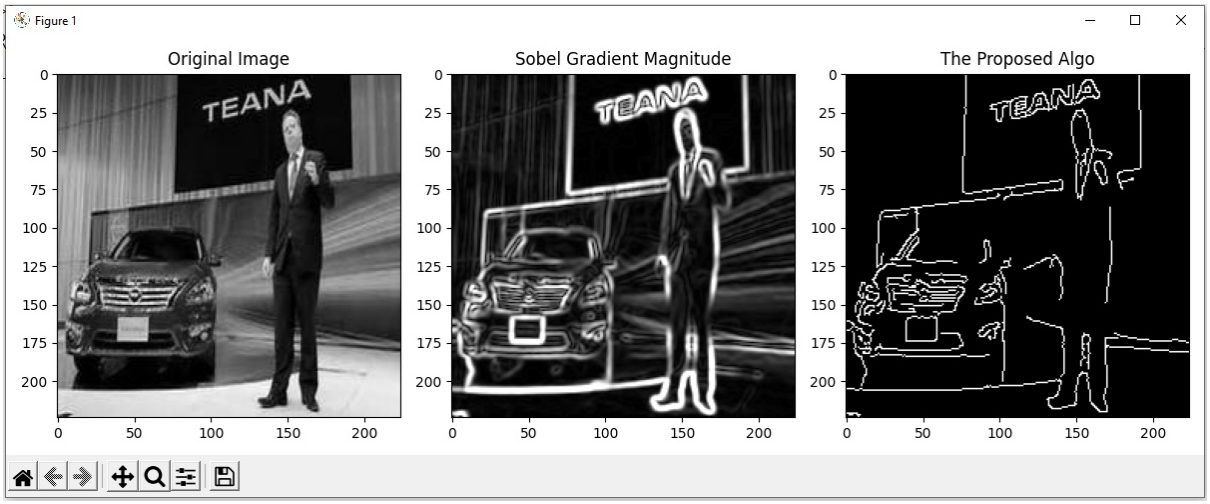
Approximation of gradient is obtained by applying these filters on the gray scale version of the image as below:
Gradient in x direction, $I_x = G_x * I$
Gradient in y direction, $I_y = G_y * I$
To eke out the gradient magnitude promptly, an approximation may be applied as

$$| \nabla I | \approx | I_x | + | I_y |$$

Instead of exact calculation as

$$| \nabla I | = \sqrt{Ix^2 + Iy^2}$$

An example run of edge detection using the proposed algorithm with a complex input image is as below.




The comparison of above discussed algorithms in terms of speed has been represented in the below table.

| **Algoritm** | **Time taken (in Seconds)** | **Speed (Images per second)** |
|---|---|---|
| Canny's Edge Detection Algorithm | 0.8280 sec | 1.207 |
| Harris Corner Detection Algorithm | 1.0155 sec | 0.985 |
| LoG | 356.1547 sec | 0.002 |
| SIFT | 7.9525 sec | 0.127 |
| **The Proposed Algorithm** | **0.6216 sec** | **1.6087** |

*Table 1 : Comparison of various feature detection algorithms*

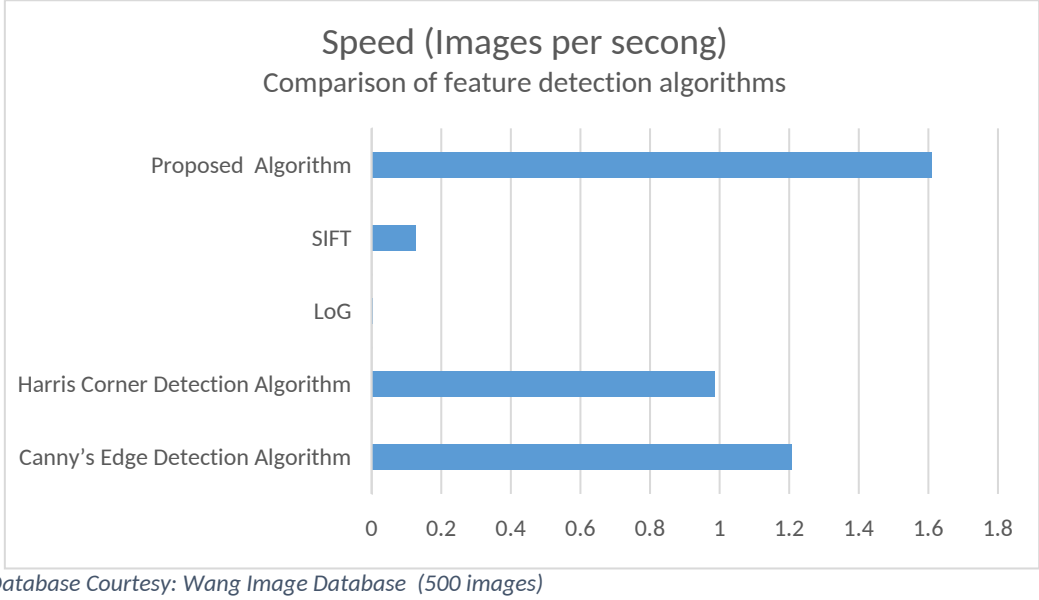



*Image Database Courtesy: Wang Image Database (500 images)*

### 2.6 Experimental Setup

The speed testing program, coded in python, was run on an i5 M480 @ 2.67 GHz & 2.67 GHz and 8GB RAM. The time calculated is for the core feature detection and that performing extra tasks e.g. loading image files and closing etc. is excluded. The values quoted above are an average of 50 runs with a database of 500 images (Courtesy : Wang Image database).

## 3. Image matching

Detection of features follows matching the image on the basis of these Features.

**3.1 Scale Invariant Feature transform (SIFT)**

SIFT algorithm uses DoG (Difference of Gaussian) to make image scale invariant, Taylor's expansion to localise the keypoints, finds the dominant orientation and then finds 128-vector Discriptor.

**3.2 Speeded up Robust Feature (SURF)**

This fast and reliable method can be better understood if divided into two phases- Feature Detection/Extraction and Feature Description [5]. In the first phase Points of interest also called keypoints are detected. This phase may involve following steps.

- Hessian Matrix found out.

Hessian matrix which is the second order partial spatial derivative of the image, has a great role in image quality enhancement and detection of important features like edge and point-of-interest.[6] Hessian matrix has the potential of filtering out point structures and linear structures based on isotropy. In a two-dimensional space point structures are isotropic whereas linear structures are anisotropic ones.

Second order derivative of Gaussian is found out as below:

$$\frac{\partial^2 G(x,y)}{\partial x^2} = -\frac{1}{2\pi\sigma^4}\left(1-\frac{x^2}{\sigma^2}\right)e^{-\frac{x^2+y^2}{2\sigma^2}}$$

$$\frac{\partial^2 G(x,y)}{\partial y^2} = -\frac{1}{2\pi\sigma^4}\left(1-\frac{y^2}{\sigma^2}\right)e^{-\frac{x^2+y^2}{2\sigma^2}}$$

$$\frac{\partial^2 G(x,y)}{\partial x\partial y} = \frac{xy}{2\pi\sigma^6}e^{-\frac{x^2+y^2}{2\sigma^2}}$$

The Hessian Matrix of a pixel at point $(x,y)$ is something like

$$H(G(x,y) = \begin{bmatrix} \frac{\partial^2 G(x,y)}{\partial x^2} & \frac{\partial^2 G(x,y)}{\partial x\partial y} \\ \frac{\partial^2 G(x,y)}{\partial x\partial y} & \frac{\partial^2 G(x,y)}{\partial y^2} \end{bmatrix}$$

- DoG calculated

Instead of Laplacian of Gaussian, Difference of Gaussian using q-Gaussian kernels would yield excellent results by feature enhancement of blurred images through Gaussian blurring.[8] q-Gaussian kernel provides extra entropic index as compared to classical Gaussian kernel.

- All this is applied on integral image instead of image itself.
- Determinant of Hessian Matrix is found out.

This procedure is scale invariant i.e. instead of scaling up the image, the filters are scaled up.

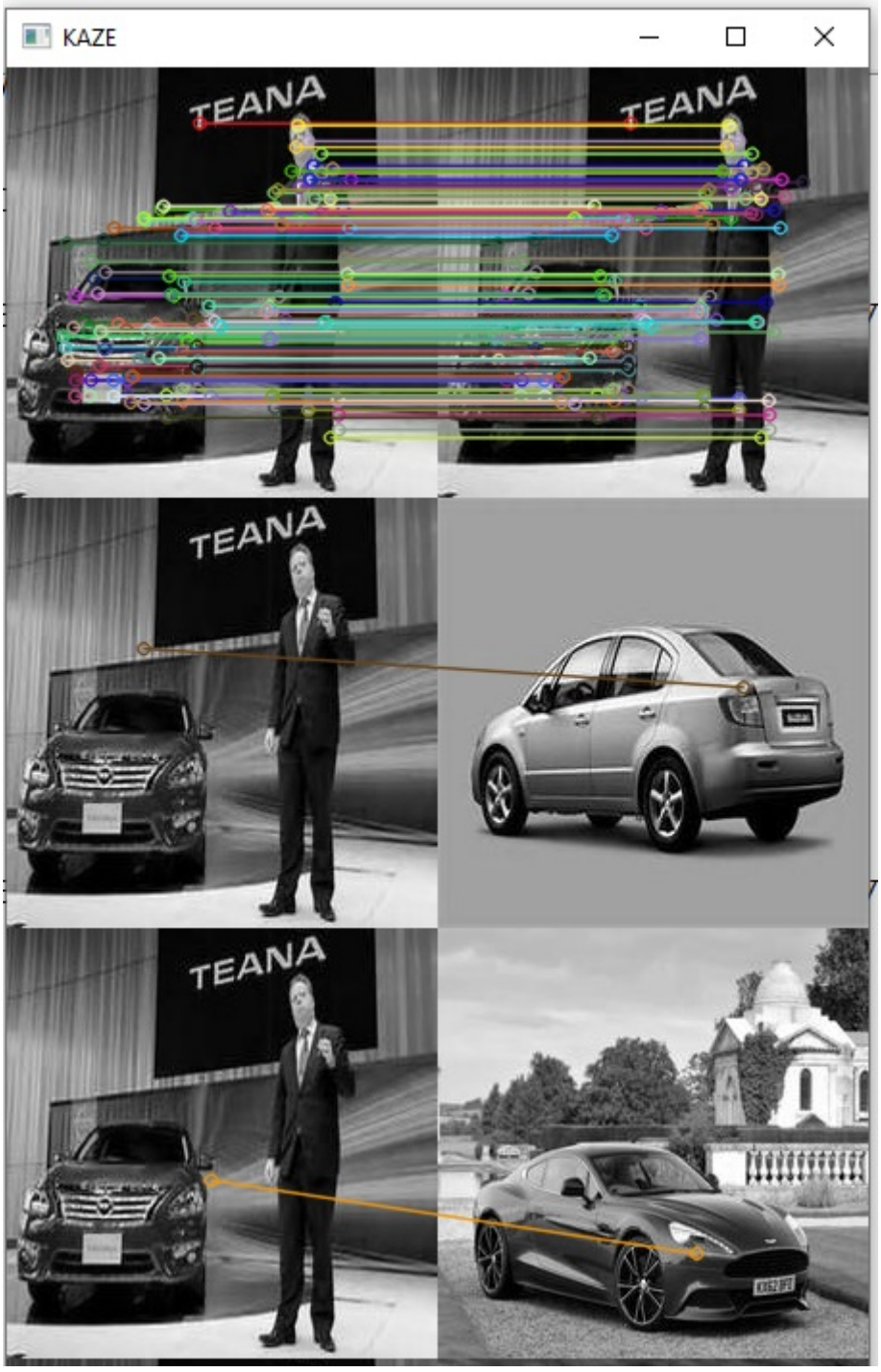
The second phase in which the Feature Vector is calculated involves the following steps.

- Obtaining Descriptor Component (usually 2x2)
- Orientation of descriptor component along dominant direction
- Obtain HAAR wavelet response to obtain uniform orientation

This descriptor is 64 dimensional which is more concise as compared to that in SIFT. This makes SURF three times faster than SIFT with lesser complexity.

### 3.3 KAZE and Accelerated KAZE (AKAZE)

SIFT and SURF both use linear diffusion for blurring image. This may affect the natural boundary of the object as well. Which results into lesser gradient between the boundary and non-boundary region. KAZE whereas offers non-linear diffusion for edge detection with more clarity. This non-linear provides larger gradient between points of interest and the noise component.



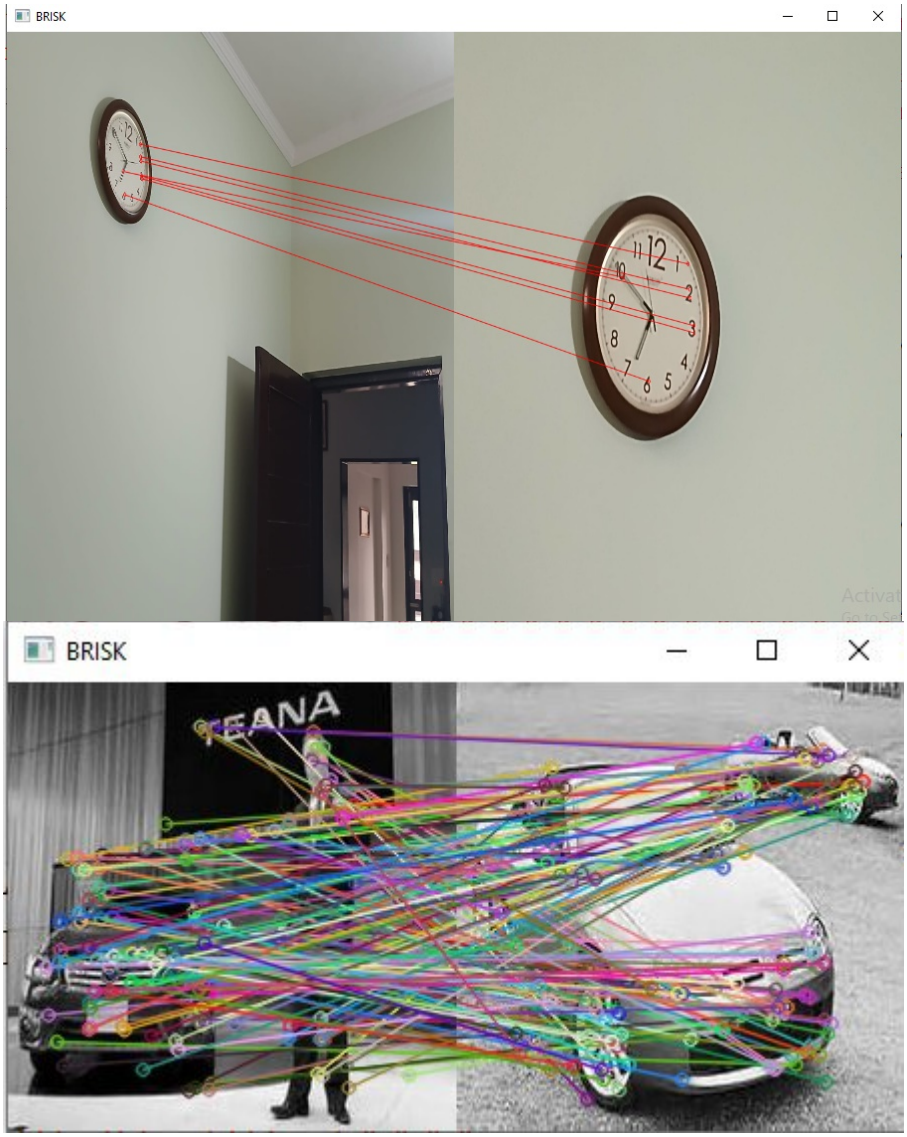

### 3.4 Oriented FAST and Rotated BRIEF (ORB)

ORB method is a diffusion of two methods- FAST and BRIEF. FAST (Features from Accelerated Segment Test) is implemented to find corners primarily based on Canny's Edge

Detection method and Harris corner score is used to rank the key-points detected.. It uses the gradient of intensity variation as the basis of filtering. Inclusion of Orientation and Rotation components makes ORB invariant of direction and alignment which is an added benefit. BRIEF algorithm i.e. Binary Robust Independent Elementary Features computes feature matching vectors.

### 3.5 Binary Robust Invariant Scalable Keypoints (BRISK)

As proposed by Swati Shilaskar et.al.[9]




### 3.6 The Proposed improvisation in the existing algorithms

The ORB algorithm, though considered to be most time efficient and accurate, yet it is scale invariant. So it is prone to fail in matching different sized images which is he most probable case. Hence to add up to the precision of ORB, an additional step may be introduce to make it scale invariant. For this purpose, there is no better algorithm than finding Difference of Gausian. So, before finding the keypoints and descriptor, DoG may be applied to the input image. This will definitely improve the accuracy and will not have any toll on speed either.

### 3.7 Indexing the Features

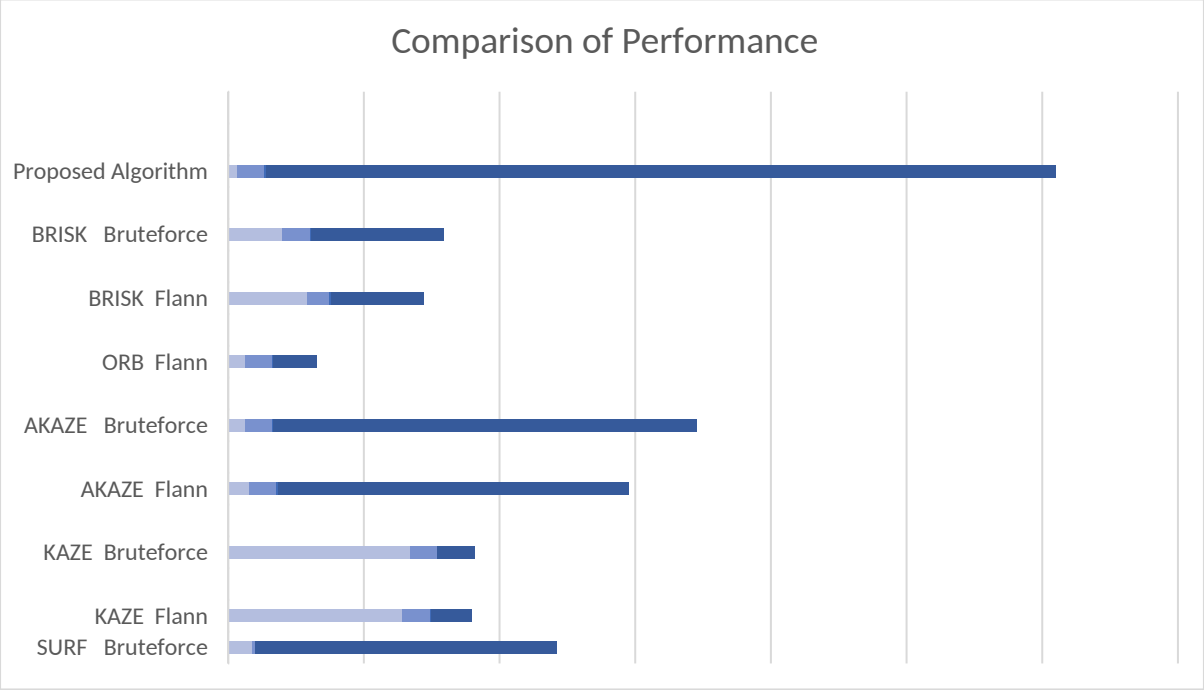
Whereas these key features are identified in the form of descriptors. It is proposed that image matching can be considerably improved on the basis of precision and accuracy if these kepoints are stored in a well organised way so that their rematching should be least iterative. For this, two mainstream indexing algorithms viz. BruteForce and FLANN (Fast Library for Approximate Nearest Neighbours) were compared in tandem with various feature detection algorithms. Their performance is compared as follows.

**3.8 Comparison of Performance**

| **Algorithm** | **Time*** | **Precision** | **Recall** | **Speed†** |
|---|---|---|---|---|
| SIFT Feature Detection with Flann Matching | 17.71 | 0.99 | 0.53 | 0.056 |
| SIFT Feature Detection with Bruteforce Matching | 9.26 | 0.97 | 0.21 | 0.108 |
| SURF Feature Detection with Flann Matching | 9.31 | 0.98 | 0.28 | 0.107 |
| SURF Feature Detection with Bruteforce Matching | 9.01 | 0.96 | 0.18 | 0.111 |
| KAZE Feature Detection with Flann Matching | 64.32 | 1.0 | 0.60 | 0.015 |
| KAZE Feature Detection with Bruteforce Matching | 66.99 | 1.0 | 0.12 | 0.014 |
| AKAZE Feature Detection with Flann Matching | 7.73 | 1.00 | 0.72 | 0.129 |
| AKAZE Feature Detection with Bruteforce Matching | 6.37 | 1.0 | 0.25 | 0.156 |
| ORB Feature Detection with Flann Matching | 6.23 | 0.98 | 0.63 | 0.160 |
| BRISK Feature Detection with Flann Matching | 29.20 | 0.8 | 0.8 | 0.034 |
| BRISK Feature Detection with Bruteforce Matching | 20.15 | 1.00 | 0.23 | 0.049 |
| **Proposed Algorithm** (DoG Improvised ORB Feature Detection with Bruteforce Matching) | 3.43 | 1.00 | 0.52 | 0.291 |

* Time (in Seconds) for matching one image with 500 images database.

† speed for matching images with database in one second




## 4. Improving Image search using deep learning

The feature extraction can further be boosted with the introduction of machine learning. Out of many, a few algorithms have really proved to be very efficient if used for CBIR. CNN and kNN methods are the most useful in this context.

### 4.1 Advent in CNN

In the earlier phase, image classification was done through simple Artificial Neural Networks. In the traditional method, the training parameters become too large even for images with very low resolution. Convolution Neural Network (CNN) has evolved over the time to find solutions to this problem. The major evolution has happened in the architecture of learning network as below.

***4.1.1 LeNet :*** it's a simple most Feed Forward CNN containing three segments- convolution, pooling and non-linear activation function. This network contains seven layers which train parameters [10].

***4.1.2 Alexnet :*** it is a 5+3 layered feed forward CNN. First five layers are convoluiton based and then a set of three fully connnected layers.[11]. Ciseran et.al. expanded it for multiple image databases[12].

***4.1.3 SqueezeNet :*** Landola et.al. in 2016 proposed a squeezed version of AlexNet and claimed their network be faster having 50 times lesser parameters requiring less than iMB memory instead of 250MB[13]. Ganesh and Abhinav proposed further improvement in terms of even smaller network with lesser number of parameters so that algorithm may fit easily into memory [14].

***4.1.4 GoogLeNet :*** with Inception Modules[15], GoogLeNet revolutionised the way images were interpreted.

***4.1.5 ResNet:*** Residual Neural Network[16] proposes that instead of introducing more and more layers, proper connections among them can solve the purpose without expanding the network wastefully.

***4.1.6 DenseNet :*** Huang et.al. [17] proposed further extension in connectivity of various blocks in the network to produce results with greater efficiency.

**Inception model Version3**

Inception v3[18] runs three convolution layers with different sized filters (1×1, 3×3 and 5×5) and one 3×3 maxpooling layer. All of these are run in parallel and the results are concatenated to obtain a more reliable feature vector. As the multiple channels are run parallel than in tandem, it is rather a wider network as the other networks are deeper. This architecture of learning networks makes it faster as most of the layers start working simultaneously instead of waiting for the output from previous layer. In addition, it reduces the chances of vanishing gradient. Higher versions of inception further reduces the parameter size by factorising the convolution. This can be achieved by replacing one 5×5 convolution with two 3×3 convolutions in tandem. Further factorising 3×3 convolution into 1×3 and 3×1 can reduce the parameter size upto 33%.

Deeper networks undoubtedly produce more accuracy but beyond a certain limit they will suffer from vanishing gradient. Wider networks are faster but more the channels more is the complicacy level. Higher resolution means better picture information which will definitely enhance the pace of learning. But resolution beyond certain limit will definitely have lesser performance over data ratio.

So choosing optimal values of these three factors viz. depth i.e. no of layers in tandem, channels i.e. layers running in parallel and the resolution of image makes the network an EfficientNet.

## 4.2. k-Nearest Neighbour method

After choosing a suitable value for k i.e. no. of neighbours on the basis of Euclidean distance, the data point is assigned the category which has maximum occurrence in this neighbourhood. Yanhui Guo et.al.[19] proposed a method clubbing guided filters with kNN classification to produce highly accurate results in image classification in hyperspectral region.

## 4.3. K-Mediod Clustering Method

Bhat Aruna[20] proposed k-mediod clustering to be improvised for image classification. The technique specifically known as Partitioning Around Mediod (PAM) was used to divide the image data set in clusters having nearest possible centroid.

# 5. Conclusion

It is observed that Speeded Up Robust Feature Detection Algorithm offers many benefits over the previously prevailing algorithms in terms of scale invariance, concise dimensionality etc.

k-NN method produces results efficiently in a non-noisy environment when they are no outliers present. But in case of presence of noise and outliers, k mediod clustering appears to be more accurate and efficient. Among the CNN, ResNet, DenseNet and GoogLeNet has obtained better accuracy where as SqueezeNet focuses on raising performance by reducing the memory size requirement.

As far as feature detection algorithms are concerned, Canny's Edge Detection Algorithm is quite time efficient. But the performance is further improved if Sabel's filtering function is used. Whereas a combination of ORB with Bruteforce Matching outsmarted all other combinations in terms of all the assessing parameters viz. Speed Precision and Recall.

Therefore, we propose a hybridised algorithm viz Improvised Canny's Edge detection based oriented FAST and rotated BRIEF with accelerated matching through Bruteforce algorithm and for an improved precision, deep learning neural methods may be involved.

# References:

[1] Liu, R., & Mao, J. (2018). Research on Improved Canny Edge Detection Algorithm. *MATEC Web of Conferences 232, 03053*.

[2] Sanchez, J., Monzon, N., & Salgado, A. (2018). An Analysis and Implementation of Harris Corner Detector . *Image Processing Online*.
[3] Boyer, K. L., & Sotak Jr., G. E. (1989). The Laplacian of gaussian Kernel: A formal Analysis and Design Procedure for fast, accurate convolution and full frame output. *Computer Vision, Graphics and Image Processing*, 147-189.
[4] Adams, M. (2019). *The Fractional Harris-Laplace Feature Detector.*
[5] Allaberdiev, S., Yakhyoev, S., Fatkhullayev, R. and Chen, J. (2019) Speeded-Up Robust Feature Matching Algorithm Based on Image Improvement Technology. Journal of Computer and Communications, 7, 1-10.
[6] Chen, X., Wu, Y., Zhu, C., & Liu, H. (2022). Image Quality Enhancement Algorithm using Hessiona Matrix. *Journal of New Media*.
[7] Chen, X., Wu, Y., Zhu, C., & Liu, H. (2022). Image Quality Enhancement Algorithm using Hessiona Matrix. *Journal of New Media*.
[8] Assirati, L., Silva, N. R., Berton, L., Lopes, A. A., & Bruno, O. M. (2014). Performing Edge Detection by Difference of Gaussians using q-Gaussian Kernels. Journal of Physics.
[9] Shilaskar, Swati & Bhatlawande, Shripad & Chavare, Ranveer & Joshi, Rushikesh & Jaiswal, Sakshi. (2023). Vision Based Fall Detection with BRISK Feature Descriptor. 1-7. 10.1109/ICEEICT56924.2023.10157061.
[10] LeCun, Y.; Boser, B.; Denker, J. S.; Henderson, D.; Howard, R. E.; Hubbard, W.; Jackel, L. D. (1989). "Backpropagation Applied to Handwritten Zip Code Recognition". Neural Computation. 1 (4)
[11] Krizhevsky, Alex; Sutskever, Ilya; Hinton, Geoffrey E. (2017). "ImageNet classification with deep convolutional neural networks"
[12] Cireşan, Dan; Meier, Ueli; Schmidhuber, Jürgen (2012). Multi-column deep neural networks for image classification. 2012 IEEE Conference on Computer Vision and Pattern Recognition. New York, N
[13] Landola, Forrest N; Han, Song; Moskewicz, Matthew W; Ashraf, Khalid; Dally, William J; Keutzer, Kurt (2016). "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size".
[14] Ganesh, Abhinav. "Deep Learning Reading Group: SqueezeNet". Kdnuggets(2018)
[15] Ning Bi, Jiahao Chen and Jun Tan. “The Handwritten Chinese Character Recognition Uses Convolutional Neural Networks with the GoogLeNet” (2019)
[16] He, Kaiming; Zhang, Xiangyu; Ren, Shaoqing; Sun, Jian (2016). Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas, NV, USA: IEEE. pp. 770–778
[17] Huang, Gao; Liu, Zhuang; van der Maaten, Laurens; Q. Weinberger, Kilian (2018). "Densely Connected Convolutional Networks".
[18] Szegedy, C., Vanhoucke, V., Ioffe, S., Schlen, J., & Wojna, Z. (2015). Rethinking the Inception Architecure for Computer Vision. *Arxiv - Computer Vision and Pattern Recognition*.
[19] Yanhui Guo, Siming Hanb, Ying Lia, CuifenZhanga, and YuBaic. “K-Nearest Neighbor combined with guided filter for hyperspectral image classification” Procedia Computer Science 129 (2018) 159–165
[20] Aruna Bhatt. “K-Medoids Clustering Using Partitioning Around Medoids For Performing Face Recognition”, International Journal of Soft Computing, Mathematics and Control (IJSCMC), Vol. 3, No. 3,(2014)