Performance Analysis of SIFT/SURF Algorithms in Neural Networks for Optimized Feature Detection

Aditya Retissin, Mohammed Jasim

Abstract*: This paper is an experiment on the implementation of scale-invariant feature transform (SIFT) and speeded up robust features (SURF) algorithms into multi-dimensional neural networks. We are attempting to perform a comparative performance evaluation by using different scale factors of the SIFT algorithm in multi-layered neural networks. This method will help us to understand the best way of implementing the above algorithms in neural networks and from a given sample, extracting distinctive invariant features and finding points of interests. Hence performing a large data set computation would be made much easier because of the neural network implementation. The conventional method of performing SIFT has computational limitations and we aim to achieve best possible way of performing the feature detection when using SIFT and neural network combined, hence transcending computational limitations that SIFT previously had. This approach to recognition of features can robustly find results much faster on bigger dataset and at the same time have the benefits of SIFT algorithm

Keywords: Machine Learning, Computer Vision, Neural Network, SIFT (Scale Invariant Feature Transform)

I. INTRODUCTION

In today's world, the application of computer vision in various different field is booming. From medical field, education, military, Online marketing, Surveillance, industrial uses are to name a few. Feature detection and Object classification from digital data that is abundance with the rise of cameras and feasible implementation of same in both industry and consumer devices are plenty. The data acquired from these devices compliment the idea of computer vision and provide a medium for feature detection, classification and various information collection from the given sample of digital images. The potential of computer vision and so this method of approaching feature detection is high now and in the future where digital data would only increase even more than now.

A. Need for Computer Vision

A lot of modern problems can be solved easily with the help of computer vision when it is complimented with the technological advancement. Object reconstruction, Optical character recognition, vision biometrics, forensics, crowd analysis, robotic sketching [1] etc. The technology can be used for benefit of human kind as well as the planet as a whole, with pattern detection in meteorological data and have evacuation warnings for people living near coastal region in case of a disaster.

Analyze long term weather patterns and predict extreme weather, with enough investment in the technology may be even prevent extreme weathers upon predictions and counter acting to the cause.

B. Conventional Problems in Computer vision.

The advent of mobile phones and camera side-by-side took out a major problem the computer vision was facing, the data. Now we have more data than ever to work on, however, to reach an ideal place where we can use this technology feasible, we will need more research and development in semiconductors, IC technology if looked at this from a conventional point of view. However, upon perfecting Quantum computing and have it consumer available will be the time computer vision will reach its peak.

C. Challenges involved in Computer vision

The most difficult challenge in faced computer vision is semantic segmentation. A perfected semantic segmentation would be able to understand data and extract information from a scene just like a human would with their eyes. Segmentation is considered as the biggest challenge because a lot of computational power is required and even with such computational power there could be many errors because of the noise. Segmentation requires pixel-labelling on an individual scale and class.

The noise during the assigning of the class to pixel will create indifferences that would reduce the accuracy and precision of the results, which would be fatal on application which would have consequences and affect security of people or property.

Segmentation on very large data pose a different kind of problem altogether. Segmentation carrying out on larger dataset would require paramount of computational power. The computation time that takes would be crucial as calculation that takes places every millisecond would matter for real time operating systems (RTOS) like an autonomous vehicle using its optical assets and computer vision. Hence a system with very high response time has to be created which would be difficult if segmentation is performed and the dataset is very high.

D. Application

There are a number of works done on computer vision and some of them are cited below

1. Using egomotion as supervision and using object detection [2]

Revised Manuscript Received on April 25, 2019.

Aditya Retissin, Software Engineering, SRM Institute of Science and Technology , Chennai, India.

Mohammed Jasim J S,Mechatronics Engineering, SRM Institute of Science and Technology, Chennai, India.

2. Visual tracking with fully convolutional networks[3]
3. Unmanned aerial vehicles [5]
4. Autonomous Cars
5. Cancer treatments [4]

E. Approach

We are going to conduct this experiment by taking the functionality of the SIFT algorithm and having the individual components of the algorithms put into the multi-layered neural network. Various different types of implementation will be performed with respect to change in scale and octaves of the SIFT algorithm.

II. WORK FLOW OF NEURAL NET

Neural network will help us solve wide range of complex problems. With its unique ability to self-adapt in unsupervised learning, it has helped developers solve various computer vision problems which couldn't be solved before.

In the below diagram, we have the

1. Input Layer
2. Hidden Layer
3. Output Layer

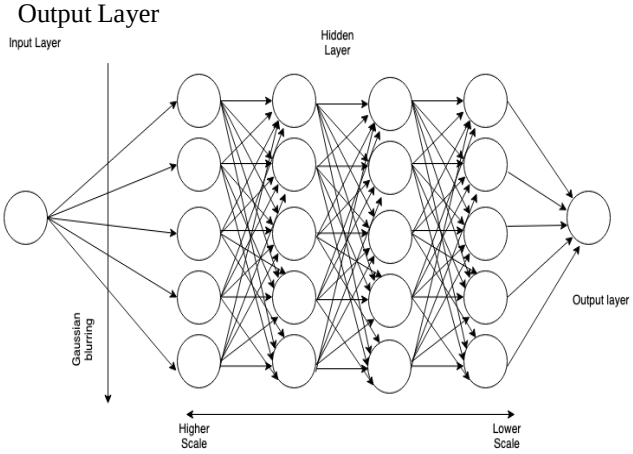


Fig 1 Network's framework

In the neural network(fig 1), the input layer (which takes an image as a parameter) is fed into the network. Along the hidden layer we have 4 octaves. In which each octave consists of nodes containing Scaling and Gaussian Blurring. So as the image is passing through the neural network, we will achieve various scales and blurring of the image, the weights are multiplied with each of the nodes in the network. Each weight consists of different detection algorithm which is used to help find and extract the specific key points of the image which will allow us to get robust results. Key points are unique points which can be found and matched in another image. So, it is important for us to reduce the noise of the image by using the Taylor's Expansion to get more accurate key points. The orientation is then assigned to the key points to achieve scale invariance image rotation. A region neighboring key point depending on the scale. The descriptors are then used to match the feature of the image that is being inputted.

III. REVIEW OF SIFT ALGORITHM

SIFT algorithm is a state of art algorithm which has many different steps to simplify computational limitations from previous object detection algorithms. SIFT converts an image data into local feature vectors called descriptors, which are used later with the key points collected from performing algorithm on all the different scales and Gaussian blurring. Those features have the power to geometric transformations that are constant to scaling and rotation which makes it robust and can get results with good accuracy regardless of orientation. This algorithm is divided into the following four main stages:

1. Scale Space Extrema Detection

This stage of computation takes all the different scales and octaves. It is implemented efficiently by using difference of Gaussian function to identify potential interest points which can be used invariant of rotation or scales. Here, is L(x,y,σ) scale-space image which is a resultant of convolution process that is done between the Gaussian function G(x,y,σ) and the image I(x,y).

$$L(x,y,\sigma) = G(x,y,\sigma) * I(x,y)$$

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

The key points of the image named Difference of Gaussian(DoG) are drawn out from the computational approximation on Gaussian's Laplacian. DoG here is introduced as below,

$$D(x,y,\sigma) = L(x,y,k\sigma) - L(x,y,\sigma)$$

Where L(x,y,kσ) is the previous convolution of the image of scale kσ with Gaussian blurring G(x,y,kσ) .

2. Key point Localization

The scale space extrema have given us enough data with various scales and octaves that we can now use the Difference of Gaussian (DoG) to remove the unwanted extrema that were found from the image. This is carried out using Taylor expansion of scale-space function.

The expansion is as follows:

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

3. Orientation Assignment

Orientation assignments is carried out for assigning vectors for the key points. The key points then are used for all the future calculation with the orientation assigned to it. This is done for robust result generation on various different conditions.Here m(x,y) is the gradient magnitude of the image and θ(x,y) is the orientation which is precomputed using pixel difference.

$$m(x,y) = (L(x+1,y) - L(x-1,y))^2 + \left((L(x,y+1) - L(x,y-1))^2\right)^{\frac{1}{2}}$$

$$\theta(x,y) = tan^{-1}\left(\frac{L(x,y+1) - L(x,y-1)}{L(x+1,y) - L(x-1,y)}\right)$$

4. Key point Descriptor Generation

The data collected from the previous steps of the algorithms are used here. The key points are used to make sense of the object or feature we aim to archive. The image gradient is selectively performed on the image around each key point. The final product allows us to varying levels of local shape distortion and change in illumination.

IV. EXISTING MODEL

1. BRISK [6]

Binary Robust invariant scalable key points is very well implemented key point generator. In some of the test, even the state of art algorithms like SURF gets low performance in terms of magnitude.

2. FAST [7]

Feature from accelerated segment test is very fast corner detection algorithm originally published in 2006 by Edward Rosten and Tom Drummond. It is designed to handle real time video processing.

3. BRIEF [8]

Binary robust independent elementary features (BRIEF) is an algorithm that utilized the binary strings for the feature point descriptor. This method of feature detect has had similar results like SURF or U-SURF.

4. SIFT [9]

Scale invariant feature transform is a sate of art algorithm that was proposed by by David G.Lowe. The method has been remade many times over the years to improve the performance and the accuracy. In this paper, we would be using this algorithm for implementation of the same in neural network.

5. SURF

Speeded-up Robust Features is another iteration of SIFT. It is a novel scale and rotation invariant interest point detector and descriptor.

6. DETECTORS

Detectors are calculators which takes the input information from pixel data. Its primary job is to compute abstracts and collect useful information from the image. Below are few generally used detectors.

1. Blob detectors

Blob detectors are made to detect regions of same properties in terms of gradient and color. The regions which have constant or approximate magnitudes are called as a blob.

2. Edge detector

Edge detectors are very important computer vision tool that aims at detecting sharp changes in light intensities (difference in pixel values). A continuous change, forming a continuous pattern would be classified as an edge.

3. Corner detector

Corner detectors are another tool widely used in computer vision. Two edges that have same origin in an angle is generally considered as corner for the logic.

4. Ridge detector

Ridges are interruptions in patterns. Any values that is exponentially higher than the local maxima are considered as ridges and used further.

V. PROPOSED MODEL

The use of deep learning has been excessively used over the years. This concept has opened doors for developers to solve wide ranges of computer vision problems because of its ability to perform Unsupervised learning. Neural networks make classification simplified. In this model, we will be implementing the Scale-invariant feature transform (SIFT) into a Deep Neural Network. SIFT is a feature detecting algorithm to detect features within an image. The object is recognized by individually comparing each feature of the new image to the database. SIFT [10] has good stability and invariance. It detects local key points which contains large amount of information.

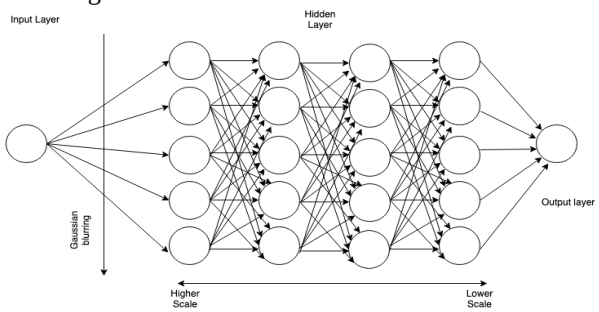


Fig 2 , Model of the Neural Network In the above figure (fig 2) , We can see that input is fed

into the hidden layer which consists of numerous nodes. Every node consists of different Gaussian blurring and various scales of the input.

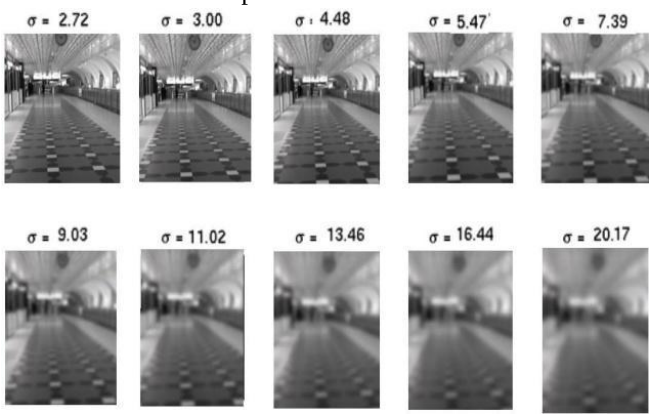


Fig 3 Gaussian blurring

Gaussian blurring is done several times to reduce the image noise and detail so that it makes it much easier to detect local features of the image (fig 3). After we have attained images with different scales and octaves we can now use the Difference of Gaussian to remove the unwanted extrema that were found from the image. Taylor expansion is used to carry out the DoG and remove the unwanted extrema. We calculate them by finding the difference of the first image and the second image and so on. The number of octaves and scale depends on the size of the image.

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

A. Sample Model

The sample model is a generalized implementation the algorithms. Here the frameworks are divided into neural network framework for the understanding and working of neural network and the algorithm framework for how the algorithm assigns values to each nodes of the neural network from the SIFT elements.

B. Sample Neural Network framework

In the below figure (Figure 4), We can see that input is fed into the hidden layer which consists of numerous nodes. Every node consists of different Gaussian blurring and various scales of the input. The number of octaves and scale depends on the size of the image. The first octave (index 0) contains a full resolution of the image. Anything after 1 scales the resolution by half. Whereas anything below 0 increases the resolution of the image.

Along the nodes x1 to x3, y1 to y3, z1 to z3 contains Gaussian blurring of the input and along the nodes x1 - z1, x2 - z2, x3 - z3 contains different scales of the input.

Gaussian blurring is done several times to reduce the image noise and detail that makes it easier to detect the features of the image. After we have attained images with different scales and octaves we can now use the Difference of Gaussian to remove the unwanted extrema that were found from the

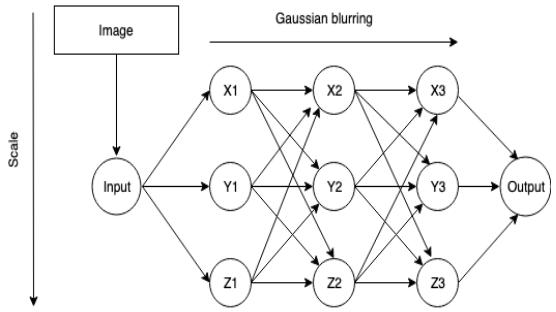


Fig 4, Sample network framework

image. We can calculate the Difference of Gaussian by subtracting the image consecutively.

$$G(x) = \frac{1}{\sqrt{2\pi\sigma}}e^{-\frac{x^2}{2\sigma^2}}$$

The Taylor's expansion is used to achieve the Key Point Localization.

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

C. Sample Algorithmic Framework

The goal here is to achieve optimized feature detection results by implementing it into a neural network. So after having generated the scale space, using that scale space we calculate the Difference of Gaussian. In the above figure 8 we see the scales and the octaves is being passed through the detectors.

1. Harris Detector
2. Szeliski Detector
3. Triggs Detector
4. Edge Detector

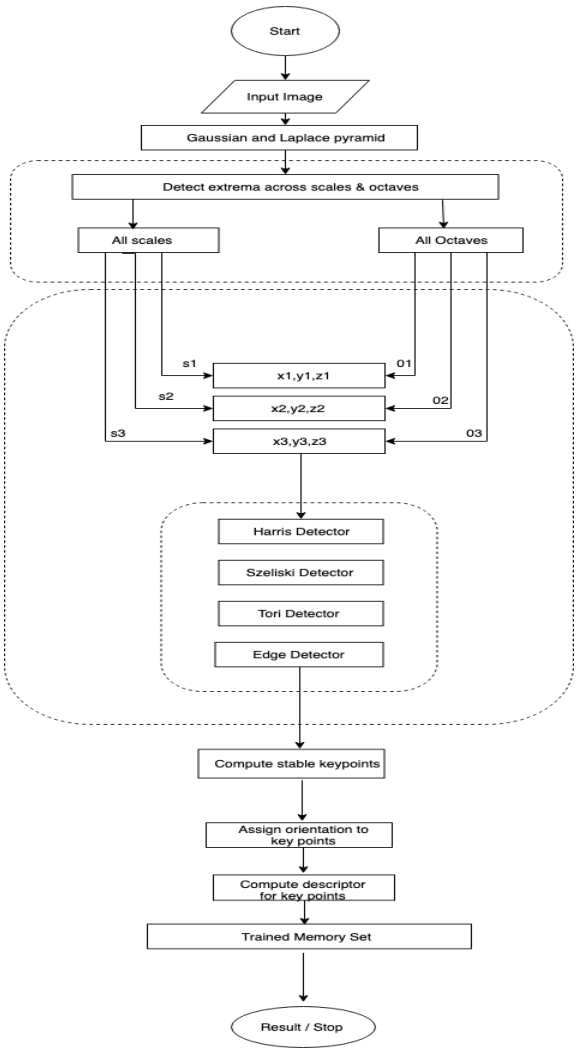


Fig 5 , Algorithm Framework

Then we assign an orientation to the key points to achieve invariance image rotation. A region closer to the key point location depending on the scale. After we have extracted the position of the key points of the image we pass it to the descriptor. Descriptors are used to compare the key points of the images.

Some key factors of descriptors are:

1. The position of the key point should be independent
2. They should be scale independent

3. Robust against image transformation
4. The comparison is done with the training set and the key points are then printed on the image.

1. Accuracy – It is calculated based on the match of the feature in the image.

2. Processing Time – It is calculated on the basis of the speed of the execution of the program.

Response time – Time taken for the system to react to a given event.

Number of classes considered – The number of classes of an object to detect in an image.

. Computation complexity of the algorithm – The complexity of the algorithm used in this model.

VI. RESULTS

The accuracy we can see is 91% for the SIFT algorithm. We have tried the algorithms with 10 classes with 5 sets with SIFT. However, we used 10 classes with 20 sets for the DNN because of DNN's versatility. For the below accuracy score chart(Table I), we can see some of number in the training set had 100 % accuracy on all the 5 sets.

1. Metrics score

Table I , Accuracy score for SIFT

al Values* / Detected Values	1*	2*	3*	4*	5*	6*	7*	8*	9*	10*
1	9	1	0	0	0	0	0	0	0	0
2	1	8	0	0	0	0	0	0	0	0
3	0	1	9	0	0	0	0	0	0	0
4	0	0	1	10	0	0	0	0	0	0
5	0	0	0	0	10	0	0	0	0	0
6	0	0	0	0	0	10	0	0	0	0
7	0	0	0	0	0	0	10	0	0	0
8	0	0	0	0	0	0	0	8	1	0
9	0	0	0	0	0	0	0	2	9	2
10	0	0	0	0	0	0	0	0	0	8

In the below table(Table II), we used the DNN algorithm and achieved an accuracy of 91%. We the algorithm with 20 sets of 10 classes.

Table II , Accuracy score for Deep Neural Net

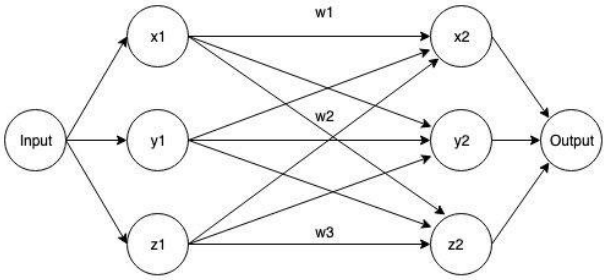
Real Values* / Detected Values	1*	2*	3*	4*	5*	6*	7*	8*	9*	10*
1	19	18	0	0	0	0	0	0	0	0
2	1	1	1	0	0	0	0	0	0	0
3	0	1	19	0	0	0	0	0	0	0
4	0	0	0	20	0	0	0	0	0	0
5	0	0	0	0	20	0	0	0	0	0
6	0	0	0	0	0	20	0	0	0	0
7	0	0	0	0	0	0	20	1	0	0
8	0	0	0	0	0	0	0	19	0	0
9	0	0	0	0	0	0	0	0	19	0
10	0	0	0	0	0	0	0	0	1	20

VII. EXPERIMENTAL SETUP

In this model, the performance is analyzed on this basis of Accuracy, Processing Time, Response Time , Number of Classes Considered and computation Complexity of the Algorithm.

Evaluation metrics :

We are given 3 different algorithms, in which the key point contains different kinds of detectors such as Harris, Szeliski , Tori and edge. The time taken for the SIFT algorithm is 1.3s with a match rate of 94.32% whereas in SURF the time taken is 1.1s with a match rate of 96.03%. We've achieved the highest match rate of 96.03% using the DNN algorithm and the time taken in DNN is 1.8s.



Layer 1	Weights	Layer 2
x1 = Scale 1, Gaussian Blur 1	w1 = Harris Detector	x2 = Scale 2, Gaussian Blur 1
y1 = Scale 1, Gaussian Blur 2	w2 = Edge Detector	y2 = Scale 2, Gaussian Blur 2
z1 = Scale 1, Gaussian Blur 3	z1 = Corner Detector	z2 = Scale 2, Gaussian Blur 3

Fig 6 , Sample network with weights assigned

Table III, Key points found and match rate in all algorithms

	Time(s)	Keypoints 1	Keypoints 2	Match rate (%)
SIFT	1.3	736	975	94.32%
SURF	1.1	831	987	94.82%
DNN	1.8	809	1002	96.03%

VIII. CONCLUSION

1) The way of carrying out feature detection in this paper is shown particularly useful in various areas. This method can be used with very large dataset because of the use of neural network usage. Various different tests were conducted under different conditions to test robustness, accuracy and other metrics of the neural network implementation of SIFT. This method can be used primarily for object detection but it has also application in 3D reconstruction, RTOS control etc.

2) Future work can be conducted on this topic for implementing this method on high definition video and real-time systems. The fluidity of the algorithm because of the neural network makes this an ideal contender for application in autonomous cars, unmanned aerial vehicle etc.

ACKNOWLEDGMENT

The authors would like to thank Andrew Ng for his many computer vision classes. We would also like to thank Sree. Varshan, Muthu Krishiv, Mahima Bhandari for their heartful suggestions towards fulfilment of this paper.

To measure the performance of the algorithm, we have various evaluation metrics such as Accuracy, Processing Time, Response Time, Numbers of classes considered and the Computation complexity of the algorithm. DNN has the highest complexity compared to SIFT and SURF. SURF has an accuracy of 90% whereas SIFT achieved an accuracy of 91%. SIFT and DNN has corresponding accuracy but the complexity of SIFT is medium whereas in DNN it's elevated.

Table IV, Overall score for SIFT, SURF and DNN

Metric	Accuracy	Processing Time (ms)	Response Time(ms)	Number of class considered	Computation complexity of the algorithm
SIFT	91%	900	400	5	Medium
SURF	90%	800	300	5	Medi um
DNN	91%	1300	500	20	Very High

REFERENCES

1. Mayank Tiwari., Subir Singh Lamba and Bhupendra Gupta., "An image processing and computer vision framework for efficient robotic sketching," in International Conference on Robotics and Smart Manufacturing (RoSMa), 2018.
2. Pulkit Agrawal., Joa˜o Carreira and Jitendra Malik., "Learning to See by Moving ,"in UC Berkeley , 2015.
3. Lijun Wang1,2, Wanli Ouyang2, Xiaogang Wang2, and Huchuan Lu1 "Visual tracking with fully convolutional networks , " in ICCV , 2015.
4. Shivangi Jain., Vandana jagtap and Nitin Pise., "Computer aided Melanoma skin cancer detection using Image Processing," in International Conference on Intelligent Computing, Communication & Convergence , 2015
5. AbdullaAl-Kaff., David Martín., Fernando García., Arturo de laEscalera and JoséMaría Armingol., "Survey of computer vision algorithms and applications for unmanned aerial vehicles." In MSC 00-01;99-00, 2018
6. Stefan Leutenegger., Margarita Chli and Roland Y. Siegwart.," BRISK: Binary Robust Invariant Scalable Keypoints." in ETH Zu¨rich , 2006.
7. Edward Rosten and Tom Drummond., "Machine learning for high-speed corner detection" inCAM.AC ,2006
8. Michael Calonder., Vincent Lepetit., Christoph Strecha and Pascal Fu., "BRIEF: Binary robust independent elementary features."in LNCS, volume6314 , 2010
9. D. Lowe, "Object recognition from local scale-invariant features". Proc. of the International Conference on Computer Vision, pp. 1150-1157 1999
10. http://www.measurement.sk/2013/Jian.pdf A Comparative Study of SIFT and its Variants

AUTHORS PROFILE

Aditya Retissin
Software Engineering Graduate of SRM Institute of Science and Technology.

Mohammed Jasim
Mechatronics Engineering Graduate of SRM Institute of Science and Technology.