International Journal of Advanced Science and Technology
Vol. x, No. x, (20xx), pp. xx-xx

Object Tracking Robot using SIFT based Computer Vision

Deepak Kamath K[1], Dabir Hasan Rizvi[2], Adarsh U[3] and B Karthik[4]

[1,2,3,4]*Department of Electronics and Communication Engineering*

CMR Institute of Technology

Bangalore, Karnataka, India

[1]*dkamath1998@gmail.com*, [2]*dabir.rizvi@gmail.com*,
[3]*adarshumesh657@gmail.com*, [4]*karthik.bataram@gmail.com*

Abstract

Latterly, we have seen an immense increase in the amount of research, development and investment dedicated to the field of Autonomous Vehicle Technology or in common terms - Self-Driving Vehicles. While modern autonomous vehicles commonly use Machine and Deep Learning models using data from sensors to understand the environment and move autonomously, this project demonstrates the use of a digital image processing algorithm- SIFT, to enable a robot to track specific objects in an environment and follow them. The implementation of an efficient and innovative image tracking algorithm for autonomous vehicles is the most important outcome of this work. This paper mentions a few use cases of this project along with the potential and future scope of it.

Keywords: *Autonomous Vehicles, Digital Image Processing, Computer Vision, SIFT*

1. Introduction

An autonomous vehicle is capable of sensing its environment and operating without human involvement. Autonomous vehicles rely on sensors, actuators, complex algorithms, machine learning systems, and powerful processors to execute software. There are several reasons as to why Autonomous Vehicles are required and how they can help revolutionize the modern world. Studies show that around 90% of all vehicular accidents are caused by human error leading to a huge number of unnecessary deaths worldwide. Autonomous Vehicles also bring about automation and comfort to a commuter. Majority of current research and development is focused on self-driving cars and this brings forth the need for industry specific robots that work autonomously or with very little human interference/commands.

Computer Vision enables self-driving cars to make sense of their surroundings. Cameras capture video from different angles around the car and feed it to computer vision software, which then processes the images in real-time to find the extremities of roads, read traffic signs, detect other cars, objects and pedestrians.

2. Related Work

In order to understand the research and development in autonomous driving and computer vision in the last years, it is important to conduct a literature survey thereby understanding the background and potential of this project.

A paper by the name “Algorithms applied in Autonomous Vehicle Systems” [1] discusses the development paths and key algorithms of autonomous vehicles. The first autonomous vehicle concept was realized in 1995 when a vehicle was able to move for about a 1000 km without human assistance using a road network map, landmark of static

objects on a map and dimensions of the road/ path. Fast-forwarding to the 21st century, Tesla launched a not fully autonomous car in the year 2014. Research is still on with a intense competition from several automobile manufacturers.

Specifically talking about path finding, algorithms such as A* and Djikstra's algorithm have been used by autonomous vehicles to find paths based on travel time, distance and fuel consumption.

In the cited paper titled "Line Following Robot Using Image Processing" [2], a robot was developed that had the functionality to follow a line detected by a camera mounted on it. A camera is used to obtain image of the track and then converted into a bitmap image. Least Square Method is used to follow the predefined path. Turns were calculated using the slopes of the line captured within the scope of the camera. Thus, by using image processing the line following robot is guided along the desired path.

The Scale-Invariant Feature Transform(SIFT) algorithm as proposed in researcher David Lowe's- "Distinctive Image Features from Scale-Invariant Keypoints" [3] serves as the backbone of this entire work.

This research aims at the development of one such autonomous vehicle with interesting use cases mentioned later in the paper. The exploitation of a very efficient object feature tracking image processing algorithm(SIFT), the use of modern technologies such as Augmented Reality at the User-Interface and IoT based communication in the working environment makes this paper an impressive state-of-the-art contribution towards science, technology and engineering.

3. Objective and Motivation

The following sections describe the main objectives and motivation behind the development of this project.

3.1. Objective

The basic objective of this project is to enable a camera(mobile phone) mounted robot to efficiently follow any object(image) as selected by the user in an environment as long as the robot and the object are in the range of the same network connection communicating via Wi-Fi. A mobile application that implements SIFT will be deployed for the user to select an object that has to be followed in the environment with the aid of the camera of the mobile phone. The SIFT based application must be able to quickly identify feature points and track the object. The robot has to be able to follow the object even in the case of scale or orientation variance of the tracked object.

3.2. Motivation

Activities in an environment such as a warehouse or a factory involve tasks such as human operators driving industry specific vehicles inside the environment such as forklifts, hand trucks, order pickers etc. These vehicles have to be driven by humans to specific locations such as racks and storage facilities within the environment which may also be hazardous potentially risking the life of the operators. This is what the project aims to eliminate. Eliminating the requirement of an operator in the vehicle also brings about comfort to him if the same work can be done by an autonomous vehicle with very less interference from him. While conventional autonomous vehicles do not have the ability to specifically follow certain objects, this project aims to contribute exactly that functionality to existing autonomous vehicle technology.

4. Scale-Invariant Feature Transform (SIFT)

A specific challenge in tracking moving objects as in the case of tracking moving vehicles is the fact that when the object is selected, the algorithm takes a while to identify its feature points and by the time it actually starts tracking it, the object might have moved further away or changed its orientation by taking a turn for instance. This is the reason behind using a scale independent edge detecting algorithm known as Scale-Invariant Feature Transform(SIFT). A rotation-invariant algorithm such Harris is not good enough because it is not scale-invariant.

A simple explanation of scale-invariance is given by Tony Lindeberg's "Scale-space Theory: A basic tool for analyzing structures at different scales" [4] - an inherent property of objects in the world is that they only exist as meaningful entities over certain ranges of scale. A simple example is the concept of a branch of a tree, which makes sense only at a scale from, say, a few centimeters to at most a few meters. It is meaningless to discuss the tree at the nanometer or the kilometer level. At those scales it is more relevant to talk about the molecules that form the leaves of the tree, or the forest in which the tree grows. Similarly, it is only meaningful to talk about a cloud over a certain range of coarse scales. At finer scales it is more appropriate to consider the individual droplets that consist of water molecules that consist of atoms, which in turn consist of protons and electrons.

A corner may not be a corner if the image is scaled. What this means is that a corner in a small image within a small window is flat when it is zoomed in the same window. This is clearly depicted in the image below.

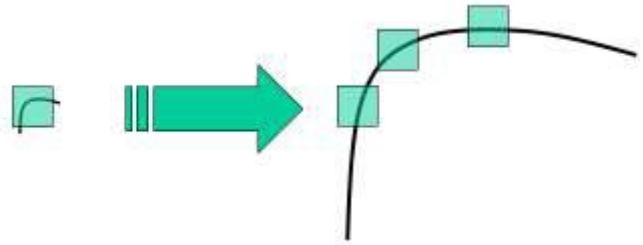

Figure 1. Depiction of change in edge with scale

The scale-invariant feature transform (SIFT) is a feature detection algorithm used in computer vision to detect and describe local features in images. As mentioned earlier, this algorithm was proposed by David G. Lowe in his paper titled "Distinctive Image Features from Scale-Invariant Keypoints" [3] where he described the algorithm in the five following steps described very briefly in this section.

i. Scale-space Extrema Detection
ii. Keypoint Localization
iii. Orientation Assignment
iv. Keypoint Descriptor
v. Keypoint Matching

4.1. Scale-space extrema detection

SIFT algorithm uses Difference of Gaussians(DoG) which is an approximation of Laplacian of Gaussian(LoG). Difference of Gaussian is obtained as the Difference of Gaussian(DoG) blurring of an image with two different σ, let them be σ and $k\sigma$. This process is done for different octaves of the image in Gaussian Pyramid shown in the figure below.

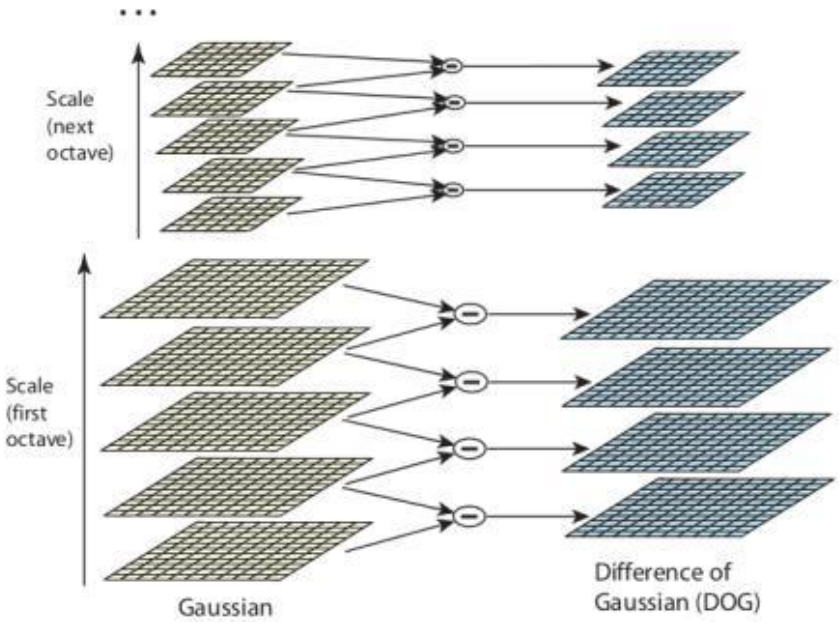

Figure 2. Difference of Gaussian[1]

Once this DoGs are found, images are searched for local extrema over scale and space. For example, one pixel in an image is compared with its 8 neighbors as well as 9 pixels in next scale and 9 pixels in previous scales. If it is a local extrema, it is a potential keypoint. It basically means that keypoint is best represented in that scale. It is shown in Figure 3 below.

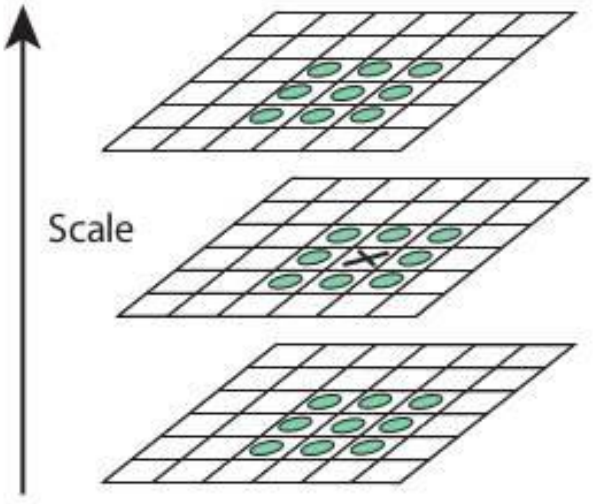

Figure 3. Finding the local extrema over scale and space[1]

4.2. Keypoint Localization

Once potential keypoints locations are found, they have to be refined to get more accurate results. The Taylor series expansion of scale space is used to get a more accurate location of extrema, and if the intensity at this extrema is less than a threshold value (0.03 as per the paper), it is rejected. DoG has higher response for edges and need to be removed. For this, a concept similar to Harris corner detector is used. A 2x2 Hessian matrix (H) is used to compute the principal curvature. In the Harris corner detector that for edges, one Eigen value is larger than the other. If this ratio is greater than a threshold, the keypoint is discarded. So it eliminates any low-contrast keypoints and edge keypoints and what remains is strong interest points.

4.3. Orientation Assignment

Now an orientation is assigned to each keypoint to achieve invariance to image rotation. A neighborhood is taken around the keypoint location depending on the scale, and the gradient magnitude and direction is calculated in that region.

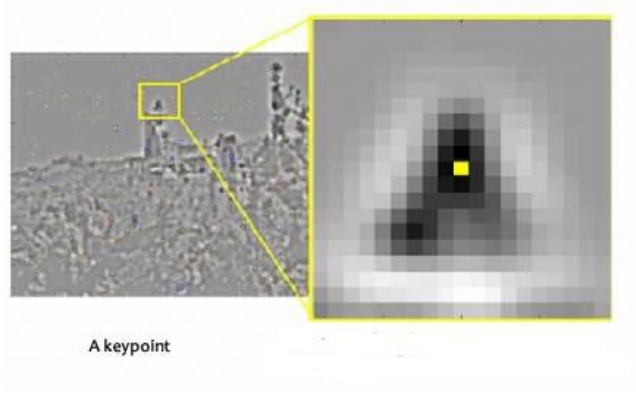

Figure 4. A keypoint and its neighborhood pixels

An orientation histogram with 36 bins covering 360 degrees is created as shown in the figure below. It is weighted by gradient magnitude and a Gaussian-weighted circular window with equal to 1.5 times the scale of keypoint. The highest peak in the histogram is taken and any peak above 80% of it is also considered to calculate the orientation. It creates keypoints with same location and scale, but different directions. It contributes to the stability of matching.

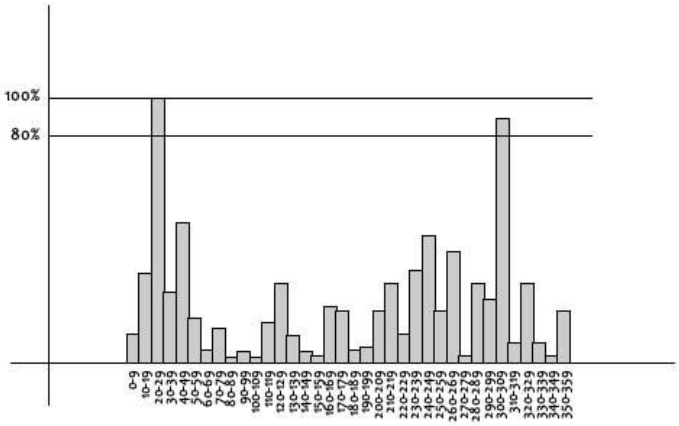

Figure 5. Orientation Histogram

4.4. Keypoint Descriptor

Now the keypoint descriptor is created. A 16x16 neighbourhood around the keypoint is taken. It is divided into 16 sub-blocks of 4x4 size as shown in Figure 6.

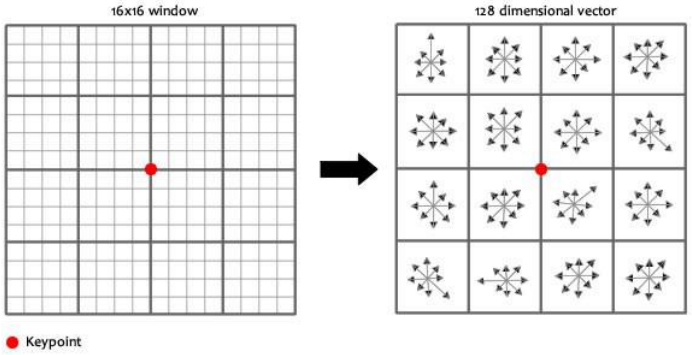

Figure 6. SIFT Descriptor[1]

For each sub-block, 8 bin orientation histogram is created as shown in the figure below.

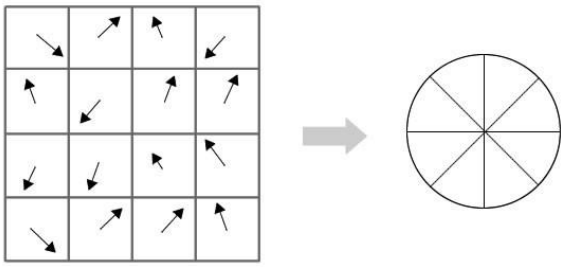

Figure 7. Example of an 8 bin orientation histogram[1]

So 4 X 4 descriptors over 16 X 16 sample array were used in practice. 4 X 4 X 8 directions give 128 bin values. It is represented as a feature vector to form keypoint descriptor. This feature vector introduces a few complications that we need to get rid of before finalizing the fingerprint.

Rotation dependence - The feature vector uses gradient orientations. Clearly, if you rotate the image, everything changes. All gradient orientations also change. To achieve rotation independence, the keypoint's rotation is subtracted from each orientation. Thus each gradient orientation is relative to the keypoint's orientation.

Illumination dependence - If we threshold numbers that are big, we can achieve illumination independence. So, any number (of the 128) greater than 0.2 is changed to 0.2. This resultant feature vector is normalized again.

4.5. Keypoint Matching

Keypoints between two images are matched by identifying their nearest neighbors. But in some cases, the second closest-match may be very near to the first. It may happen due to noise or some other reasons. In that case, ratio of closest-distance to second-closest distance is taken. If it is greater than 0.8, they are rejected. It eliminates around 90% of false matches while discards only 5% correct matches, as per David Lowe's [3] paper.

5. Implementation

As mentioned earlier, this paper focuses on the application of the SIFT algorithm to aid the movement of an object following robot.

Specific use-cases where this robot can prove to be handy are as follows:

1. At a warehouse to guide vehicles to a rack of an item to be picked.
2. On the road to make an autonomous car follow another car or a street sign.
3. In military applications such as target following missiles or drones.

OpenCV is a library of programming functions mainly aimed at real-time computer vision. This library has dedicated functions and modules to implement several image processing algorithms including the SIFT algorithm. The SIFT based mobile application was developed using the OpenCV library at the back-end.

The result of the OpenCV implementation of SIFT on an image that shows the identified feature points is shown in the figure below.

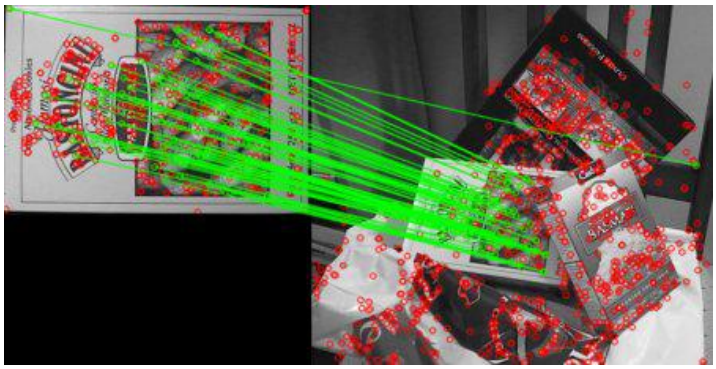

Figure 8. Identified feature points matched with an object

A NodeMCU was used to enable the Wi-Fi communication between the SIFT application and the motors of the robots wheels.

The functionality of the robot is shown in the figure below

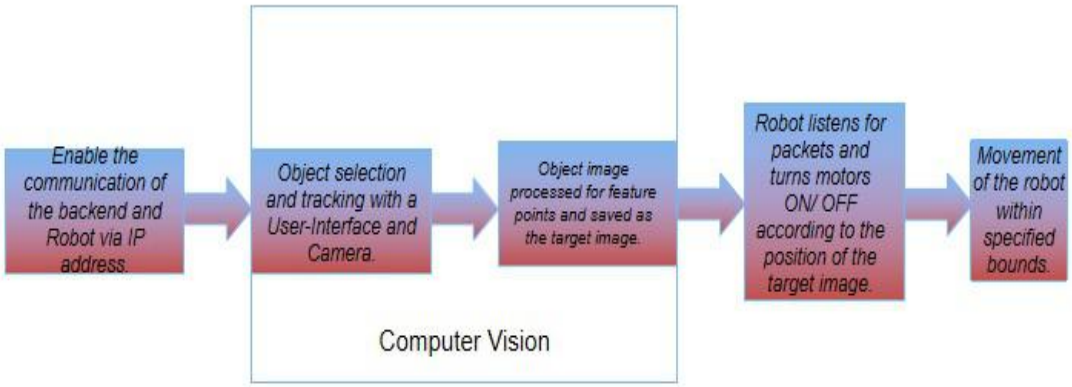

Figure 9. Functionality of the Robot

On the selection of an object detected by the mobile camera, the robot should be able to move towards that object until it reaches the minimum distance that it should maintain with the object.

When the object is aligned and in front of the robot, both the driving wheels are functional but when the object moves towards the right, the right wheel is instructed to be stationary while the left wheel is functional pushing the entire load of the robot towards the right and vice versa when the object moves towards the left. This describes the object following functionality and turning mechanism of the robot.

6. Results and Performance Analysis

The main objective of this project was to implement a very efficient object tracking robot and this required a very robust and efficient object tracking image processing algorithm. SIFT was chosen for this purpose and a mobile application was developed to interface the robot and the back-end image processing and feature extraction.

As expected the SIFT algorithm was very efficient in tracking objects kept in multiple orientations and scales as shown in figures that follow. If the object has enough feature points, the application mentions that the image is of high quality and begins tracking it. A green coloured boundary was included to represent the object that is being tracked. Low quality images and images in poor lighting conditions were not able to be tracked by the application.

The following results depict the tracking of an object in this case a book, in different orientations and scales.

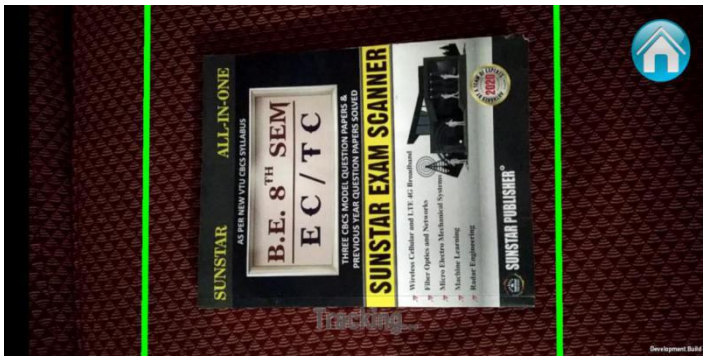

Figure 10. The object being tracked

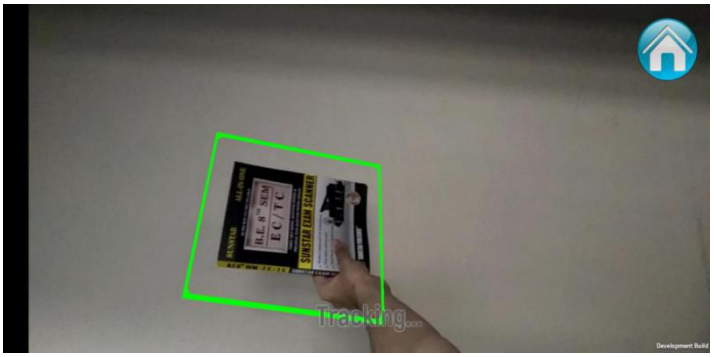

Figure 11. Tracking the scale changed object

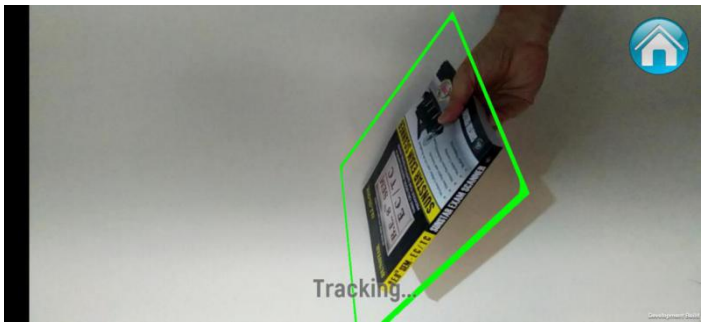

Figure 12. Tracking the orientation changed object

The robot was able to follow the target object(book) tracked by the application communicating via the NodeMCU Wi-Fi module intern communicating with the motors.

However, as mentioned earlier, the application was not able to track objects in poor lighting and determines the image as a low quality frame because enough feature points could not be obtained as showed below.

Figure 13. Tracking failure in low lighting

7. Conclusion and scope for future work

As shown in the results and implementation section of this paper, a lot of keypoints were identified in very less time and the robot was able to follow them even when the object was rotated and scaled but not all object images can yield such good results and good target images must be images with enhanced local contrast, a lot of details and good environmental lighting.

The problem of poor lightning can be solved using illumination techniques based on digital image processing and the same goes with improving contrast and details with techniques such as contrast histogram flattening, image sharpening etc. Another drawback of this implementation is the fact that the robot will stop moving if the camera is blocked or no longer in the scope of the object due to any disturbance in the environment or a technical error in the camera. This can be solved by saving the approximate location of the object based on signal strength but this only works in the case of static objects. Similarly if the robot is blocked by an obstacle, it must be able to take a turn as required to avoid collision using ultra sonic sensors for instance or image processing itself. These are the short term goals to improve the functionality of the robot.

References

[1] Bugała, Michał, “Algorithms applied in Autonomous Vehicle Systems”,**(2018)**.
[2] J. Sarwade, S. Shetty, A. Bhavsar, M. Mergu and A. Talekar, "Line Following Robot Using Image Processing",3rd International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2019, pp. 1174-1179, **(2019)**.
[3] Lowe, David, “Distinctive Image Features from Scale-Invariant Keypoints”, International Journal of Computer Vision, **(2014)**.
[4] Lindeberg, Tony, “Scale-Space Theory: A Basic Tool for Analyzing Structures at Different Scales”, Journal of Applied Statistics, **(1994)**.

Authors

Deepak Kamath K is pursuing his Bachelors in Electronics and Communication Engineering at CMR Institute of Technology, Bangalore, India. His research interests include Machine Learning, Image Processing and Data Analysis.

Dabir Hasan Rizvi is pursuing his Bachelors in Electronics and Communication Engineering at CMR Institute of Technology, Bangalore, India. His research interests include AI Programming, Augmented Reality and Game Development.

Adarsh U is pursuing his Bachelors in Electronics and Communication Engineering at CMR Institute of Technology, Bangalore, India. His research interests include Network Security and Ethical Hacking.

B Karthik is pursuing his Bachelors in Electronics and Communication Engineering at CMR Institute of Technology, Bangalore, India. His research interests include Data Analysis and Machine Learning.