iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** V **Month of publication:** May 2022

DOI: https://doi.org/10.22214/ijraset.2022.42523

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue V May 2022- Available at www.ijraset.com

Snake Game Using Hand Recognition System

Yashaswi Raj[1], Rajat Sharma[2]
[1, 2]*Dept. of Computer Science & Engineering, MIET, Meerut, India*

***Abstract:** This paper presents engineering and execution nuances of a hand's movement furthermore motion acknowledgment framework which has been created at Meerut Institute of Engineering and Technology.*
The framework utilizes live 15-piece shading video from an arranged camera, runs in genuine time (On a DEC Alpha, 25 edges per second), and adapts well to foundation mess.
Following is accomplished utilizing the 2D deformable Active Shape Models (savvy snakes), what's more a genetic calculation is utilized to play out an underlying worldwide
picture search. The Point Distribution Model, which was used for both the snakes and the hereditary calculations, is a nonexclusive and extensible model that can be used to follow any 2D deformable object.
***Keywords:** OpenCv, Feature tracking, Saavy Snakes, Genetic Algorithm, Lightning issues.*

I. INTRODUCTION

You can see me playing my all-time favorite game, which I've been playing since I was a kid, in this project. Snake Game is the name of the game. Gestures are used to manipulate the snake's position on the screen. It's your responsibility to keep the snake from hitting the wall boundaries because if it does, the game is finished. For guiding the snake, I utilized a blue-colored item as a pointer.
Gestures are used as commands in vision-based interfaces for video games instead of clicking keys on a keyboard or moving a mouse. To give the user a more natural experience, these interfaces must allow for unintentional movements and continuous gestures.
The traditional snake game is the subject of the paper I choose to concentrate on. This is a game in which the user manipulates a red object in front of a webcam to create a trail that follows the object's image and resembles the shape of a snake. The snake's size keeps growing, and the game becomes increasingly tough to play.
NumPy and OpenCV are the libraries utilized. This project is classified as part of the field of gesture recognition, which is rising in popularity since it makes it very simple for users to complete specific activities. Gesture recognition has slowly but steadily gained a foothold in the gaming business and is now expanding into other areas. It enables a far better user experience and dynamic and user-friendly interfaces when combined with computer vision.
To obtain our results, we employed techniques such as color detection, contour detection drawing, inserting png over the image, and checking for line segment intersection in our work.

II. OPENCV

It's an open source PC vision programming capabilities library aimed at helping developers create apps that take advantage of ongoing PC vision improvements. This system has a BSD permit, allowing it to be used for both business and exploration.
Object ID, division and recognition, face recognition, motion recognition, movement following, diverse advanced mechanics, and so on are some of the model applications in the OpenCV library. This structure, with its extensive features, also necessitates good knowledge of both turn of events and reconciliation processes in order to produce a product application.

A. Detection

The essential advance close by motion acknowledgment frameworks is the identification of hands and the segmentation of the comparing picture locales.
This division is vital on the grounds that it secludes the errand applicable information from the picture foundation, prior to passing them to the ensuing following and acknowledgment stages. Numerous solutions have been offered in the literature that make use of a few different types of visual elements and, for the most part, their combination. Skin tone, shape, movement, and physical models of hands are examples of such highlights. A comparable report on the demonstration of certain hand division methods can be found in Cote et al. (2006) and Zabulis et al. (2009).

©IJRASET: All Rights are Reserved

B. 3D model

For the identification of hands in photographs, a categorization of approaches uses 3D hand models. One advantage of these approaches is that they can achieve view-autonomous recognition. Various models necessitate various picture highlights in order to construct model correspondences. In kinematic hand models, point and line highlights are used to recover points framed. The hand's joints are the joints that connect the fingers to the rest of the body. Following that, hand locations are assessed to guarantee that the 3D model and detected photo highlights correlate accurately. In the writing, various 3D hand models have been proposed. In a few strategies, a deformable model structure is employed to conform a 3D model of the hand to image data. The fitment is led by forces that pull the model in toward the edges of the picture, which also are modulated by forces that assure surface congruency and equality. Incorporating physical input from a human hand into the model improves the cycle. To match the hand model to a photograph of a real hand, trademark focuses on the hand are detected in photos, and virtual springs are proposed to pull these trademark focuses to objective situations on the hand model.

C. Motion

There are a couple of approaches to deal with hand placement that employ movement as a sign. The reason for this is that gesture palm identification necessitates a highly controlled setup since it assumes that the apparent movement in the image is due to hand movement. As a result, the assumption that Hands' appearance changes in the interframe are more consistent than for other items such as clothing, face, and foundation has merit. hand placement.

D. Recognition

The translation of the semantics that the hand(s) area, stance, or signal passes on is the general goal of hand signal acknowledgment. Static and dynamic signals can also be used to categorise vision-based hand motion recognition systems. An overall classifier or a format matcher can be used to distinguish static motions (such as postures). In any instance, dynamic hand motions have a transient viewpoint that necessitates approaches such as Secret Markov Models (HMM) to handle this element, unless the transient aspect is revealed through the hand motion portrayal (for example movement based model).

E. Hidden Markov model

Because of its verifiable response for the division issue, Gee were introduced in the 1990s and quickly became the acknowledgement technique for decision. It is beneficial to first depict stowed away Markov models (Ramage 2007). Consider Markov chains. Markov chains are simply limited state automata in which each state progression circular segment has a corresponding probability value; the probability upsides of the curves departing a single state aggregate to one. Each state could address a variety of possible hand positions when it comes to hand signal recognition. The state advances are concerned with the likelihood of a certain hand posture progressing into another; the corresponding yield picture is concerned with a specific stance; and successions of result images are concerned with a hand motion.

F. Smart Snakes for Feature Tracking

Dynamic Shape Models (Smart Snakes (ASMs)) were created with the intention of accurately finding an element within a still image given a reasonable starting point. The premise is very simple. On the picture, near the element, a contour that represents the state of the element to be found is placed. The shape in the image is drawn to neighbouring edges (force changes) and can be caused to travel towards these edges by accurately twisting (inside imperatives). This approach can be used to track highlights across video edges, with the position of a component in one edge serving as a starting point for the next case.

To establish such an approach, a proper model of the object to be followed must be built. Creating an unbending framework for a hand in a given posture isn't difficult, but catching the way the form can change while avoiding invalid distortions (which is essential for a strong framework) is. A model for, example, head following would appear to be essentially application-specific, requiring a great amount of effort to design.

G. The Point Distribution Model in Brief

The Point Distribution Model (PDM) is an example of variety data that can be constructed via a preparation interaction that requires the input of several different instances of the item to be presented in various positions. Certain milestone focuses, which are focused points on the object1, should be explained in the preparation models. Factual study can be carried out to locate the usual model shape, as well as the important techniques of variety, by consistently marking these focuses across the full preparation set.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue V May 2022- Available at www.ijraset.com

The cycles in question are depicted in Figure 1 as a framework. After changing the preparation models, the mean shape is presented as far as the normal (x; y) directions of the milestone focuses. These can be put together to form a single segment vector x
where n is the model's number of points Sets of x and y displacements for the landmark points illustrate the modes of variation, which can be merged into column vectors vi

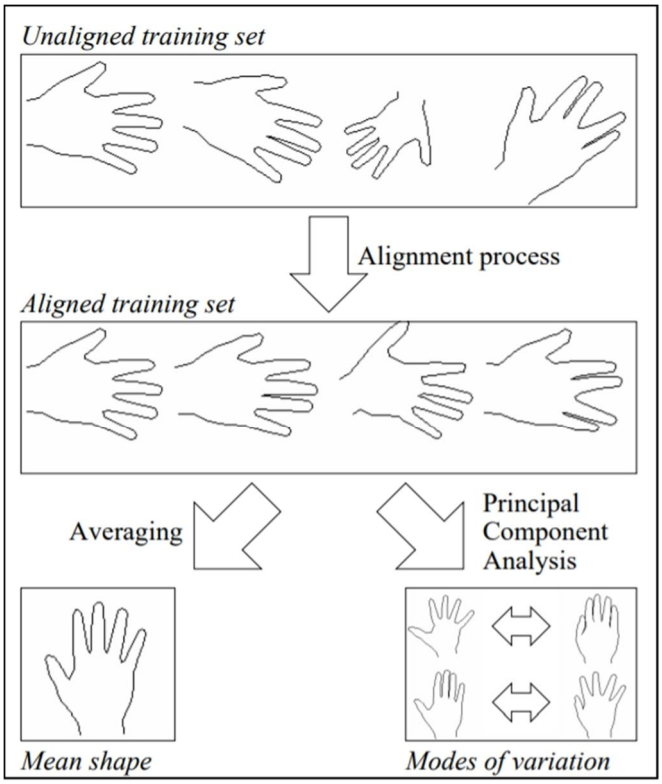


Figure 1: Training a Point Distribution Model

Change the loads in the model to generate examples. Invalid disfigurements are avoided by keeping the wi within proper cutoff points and ignoring the variety vectors vt+1 and onwards, which will frequently show turmoil in the preparation data. Figures 2 and 3 illustrate the effect of different w1 and w2 on the model shape, while leaving any remaining wi at zero. By indicating the model's area, scales, revolution, x-interpretation tx and y-interpretation ty, and posture as far as the variety vector loads wi, the model can be star jected onto a picture.

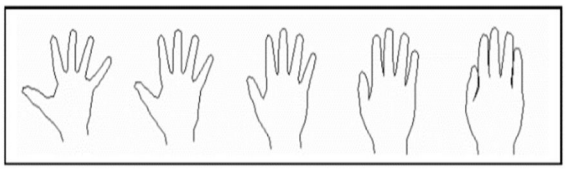


Figure 2: The first mode of variation (w_1 = -100, -50, 0, +50, +100)

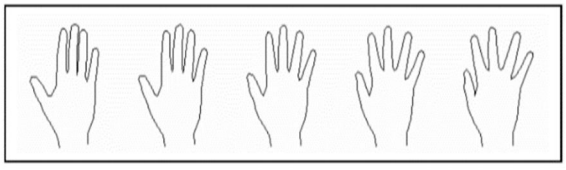


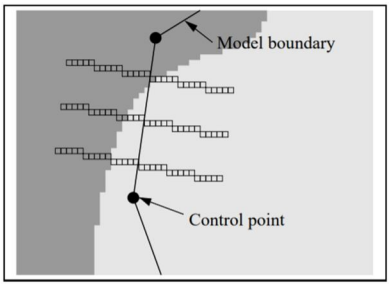
Figure 3: The second mode of variation (w_2 = -100, -50, 0, +50, +100)

©IJRASET: All Rights are Reserved

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue V May 2022- Available at www.ijraset.com

H. Feature Tracking

It's simple to keep track of edges; Look for the most grounded shift in pixel power by looking at lines of pixels ordinary to the model limit in one or the other direction after the model is close to the new edge point..



1) I evaluate pixels ordinary to the model limit for each model point, looking for the most grounded edge.
2) Determine the usual interpretations of dx and dy for each model point. Use these characteristics to understand the complete model, i.e. update the tx and ty coordinates..
3) Remove all of the dxi and dyi's eects from this interpretation. This will leave lingering esteems that should be satisfied in some way..
4) Use a suggested by and big normal scale, ds, and turn d to update s and d.
5) Subtract all dxi and dyi once more from the effects of this overall scale and revolution.
6) . The model's internal imperatives are twisted to force any remaining required developments; the wi bounds are also changed.

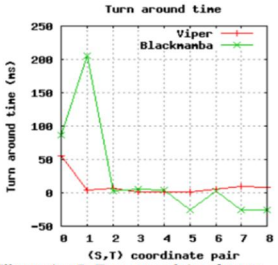


Illustration 5: Turn around time for two path finding algorithms

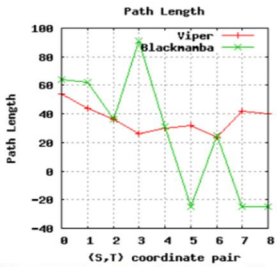


Illustration 6: Length of path returned by the two path finding algorithms

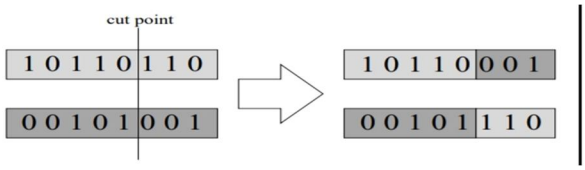
I. Recognition of Gestures

The utilization of the PDM simplies the undertaking of motion acknowledgment, on the grounds that the hand's posture is exemplified by not very many boundaries tx and ty allow you to control the hand's x and y coordinates, as well as its pivot. Observing these limitations over time reveals intricacies in the way the hand moves.

J. Outline of Genetic Algorithms

Genetic calculations (GAs) give an approach to nding maxima in uproarious capacities, where customary techniques can fizzle. The ttest's endurance is fully predicated on the Darwinian hypothesis of evolution. A population of mathematical characteristics (possible maxima) is created arbitrarily given a capacity f for which a largest is sought. The tness esteem f (xi) is calculated for each quality xi. Qualities are then restored probablistically - in the future, qualities with a high level of tness esteem are solid and will survive, whereas qualities with a low level of tness esteem are frail and will perish. Also as in development, the cycles of hybrid (mating) and change are utilized trying to produce an expansive range of qualities.

Quality hybrid comprises of taking two parent qualities (considered as double strings), playing out a join activity some place (haphazardly) and trading the two tails over, as displayed in Figure 5.



©IJRASET: All Rights are Reserved

Mutuation comprises of once in a while presenting arbitrary piece changes (this ought to happen approximately one time in 1,000).The interaction is run for two or three hundred ages (emphasess) and ideally by then, at that point, The ttest traits will take over, providing a few good options for x and increasing competition for f. Hereditary calculations can work when conventional techniques fail, but they are temperamentally probablistic rather than predictive , deterministic; For any given situation, there are numerous potential solutions that can be developed, and it is difficult to promise hundred percent success.

K. Lighting Issues

One of the super introductory issues experienced concerned lighting conditions. Natural eyes can successfully adjust to various light forces; we can also compensate for settings with a lot of contrast, utilise shade data to aid in recognising objects in a scene, and auto-center with ease.

Current camcorders have programmed focusing and brilliance control. Be that as it may, they don't adapt well to high differentiation pictures. For example, When a person sits directly in front of a window, the light from outside causes the management of brilliance automatically to reduce the brightness.

As a result of the entrance, frontal objects become exceedingly dim and difficult to perceive. Furthermore, in the typical circumstance when the camera is mounted on a workstation screen, there is almost no instant light falling on a handed up to the camera, unless there is a window directly behind the screen, which is impossible to select. Similarly, foundation objects, particularly white screen housings with excessively sharp edges, can be exceedingly distracting.

To work on the tracker's exhibition, two procedures were taken.;

one algorithmic and one physical.

The actual solution was to place up-lighters on both sides of the screen, as well as white reflecting cards, to direct light onto the client's content (as opposed to having a spotlight shining in one's eyes). This is significantly superior in terms of overall picture quality. When models (The greyscale image does not do the complete shading form justice, but notice how the hand in the right-hand photo appears lighter.)



The computational measure was to adjust the picture's overall degrees of red, green, and blue. Because a hand has a large red area (when lit properly), red was highlighted. For both green and blue components, the best equilibrium was seen with negative loads. White items behind the scenes, such as screen screen housings, vanished, while esh tones remained perfect. Figure 7 shows the eects.



Figure 7: Images without (left) and with (right) colour enhancement

III. METHODOLOGY

A. Installing Pygame

White items behind the scenes, such as screen screen housings, vanished, while esh tones remained perfect. You can do so by typing the command below.

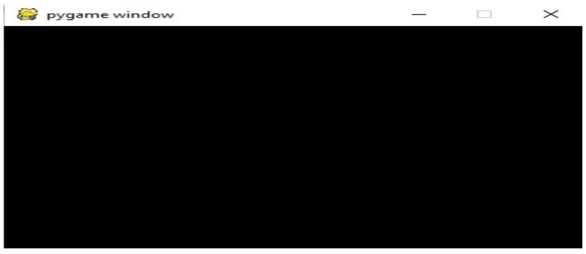
B. Install pygame with pip

Simply import Pygame and start working on your game after that is done. Before continuing, review the Pygame functions used in this Snake Game, as well as their descriptions.

Function	Description
init()	All of the imported Pygame modules are initialised (yields a tuple indicating whether or not initializations were successful.)
display.set_mode()	To build a surface, it takes a tuple or a list as a parameter (tuple preferred)
update()	The screen is updated.
quit()	Everything was uninitialized using this method.
set_caption()	The caption text will be displayed at the top of the display screen.
event.get()	Returns a list of all the events that have occurred.
Surface.fill()	Will fill the surface with a solid color
time.Clock()	Helps track time time
font.SysFont()	From the System font resources, a Pygame font will be created.

C. Construct the Screen

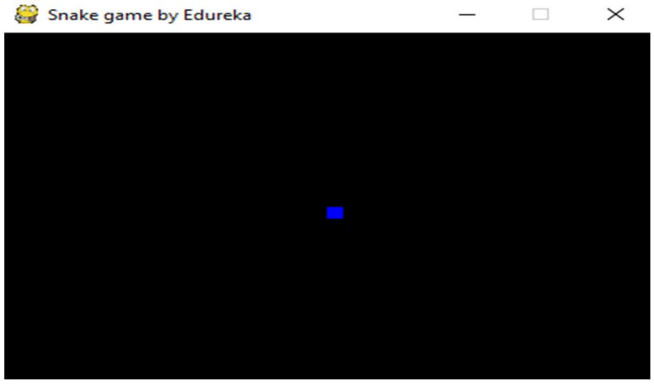
To create the screen, you'll need to use Pygame's display.set mode() function. To initialise and uninitialize everything at the beginning and end of the code, respectively, you'll need to use the init() and quit() functions. The update() method is used to update any changes to the screen. Flip() is a mechanism that operates similarly to the update() function. The update() function only updates the modifications that have been done (but updates the full screen if no arguments are specified), but the flip() method completely redoes the screen.



When you run this code, you'll notice that the screen you saw previously doesn't go away, and it also returns all of the actions that occurred on it. Using the event.get() function, I was able to accomplish this. I also used the display.set caption() function to label the screen **.**

D. Construct the Snake

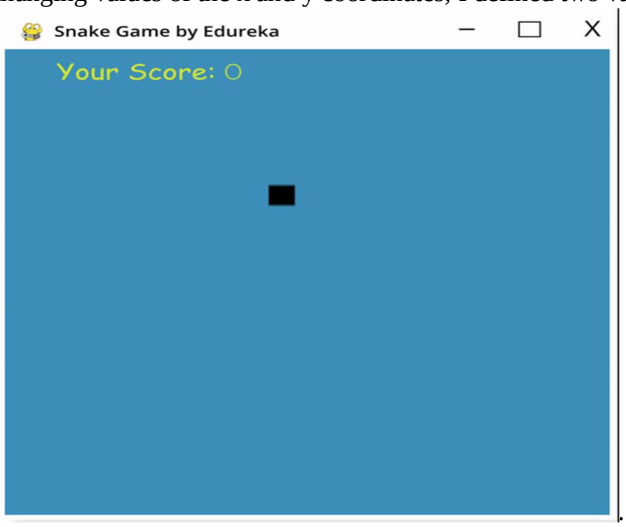
To make the snake, I'll start by setting up a few colour variables to colour the snake, food, and screen, among other things. RGB (Red Green Blue) is the colour scheme used in Pygame. Set all of them to zeros to make the colour black, and all 255s to make the colour white. As a result, we'll have a rectangle for our snake. You can use the draw.rect() function in Pygame to draw rectangles with the specified colour and size**.**



ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue V May 2022- Available at www.ijraset.com

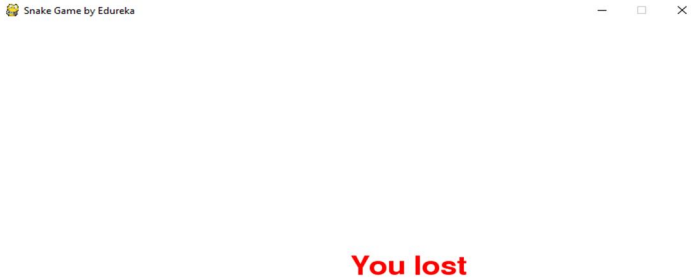
E. Snake Movement

You'll need to use the key events provided by Pygame's KEYDOWN class to move the snake. The snake moves up, down, left, and right using the events K UP, K DOWN, K LEFT, and K RIGHT .Fill() can also be used to change the display screen's colour from black to white. To keep track of the changing values of the x and y coordinates, I defined two variables, x1 change and y1 change.



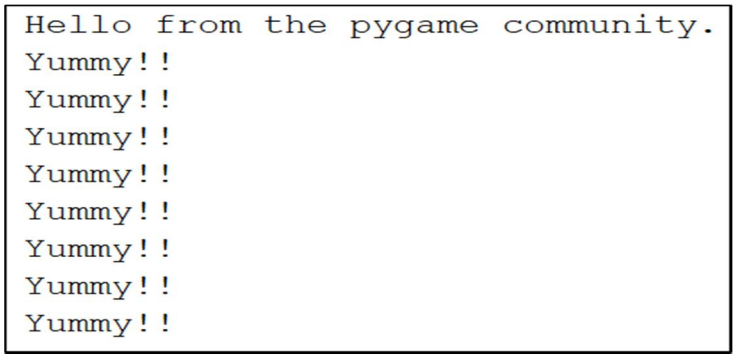
F. When Snake Reaches the limits, the game is over

In this snake game, if the player hits the screen's bounds, he loses. Using an if statement, I declared that the snake's x and y coordinates must be less than or equal to the screen's coordinates. Remember that I've replaced the hardcodes with variables to make future game tweaks easy for you.



G. Putting the Food In

I'll put some food here for the snake, and when it crosses over, I'll get a message saying "Yummy!!" In addition, if the player loses, I will give the player the option of stopping the game or restarting it..



©IJRASET: All Rights are Reserved

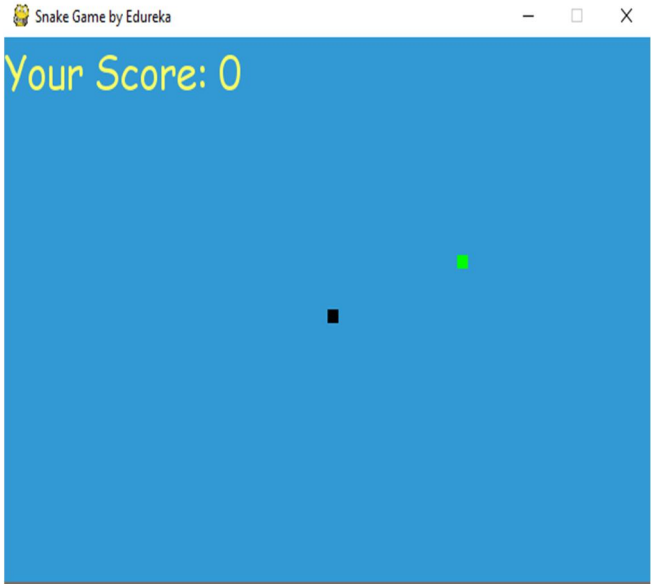
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue V May 2022- Available at www.ijraset.com

H. Increase the Snake's Size

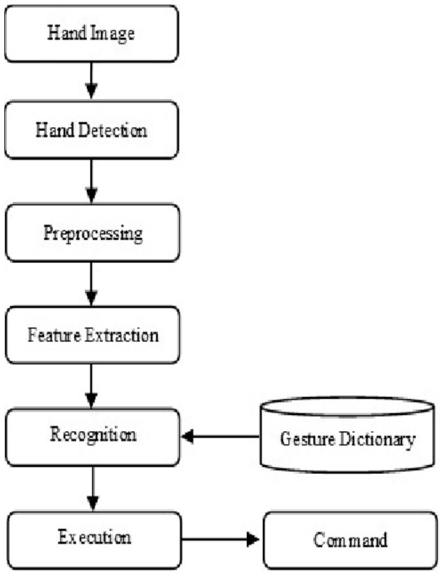
When our snake eats the food, the code below will enlarge it. Furthermore, if the snake collides with his own body, the game will finish and you will be notified that "You Lost!" "Q-Quit or C-Play Again" is the greatest option. As seen in the code below, the snake's length is normally kept in a list, and the initial size is one block..

I. The Score is Presented

Last but not least, we may have to display the player's ranking. We created a new function called "Your score" to do this



J. Flowchart



K. Remarkable Features

1) As the majority of the handling is done before the real game beginnings, it should diminish the handling time during the game-time.

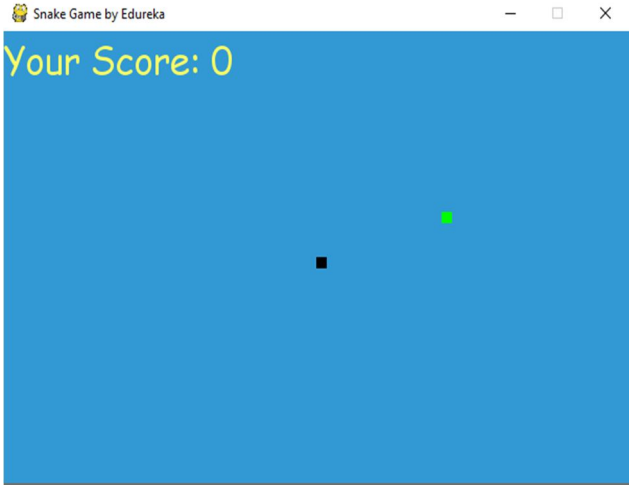
2) The idea of open-square shape permits various way, and thus inside it the snake can be. moved haphazardly towards the predetermined point. Also thus it is wise.

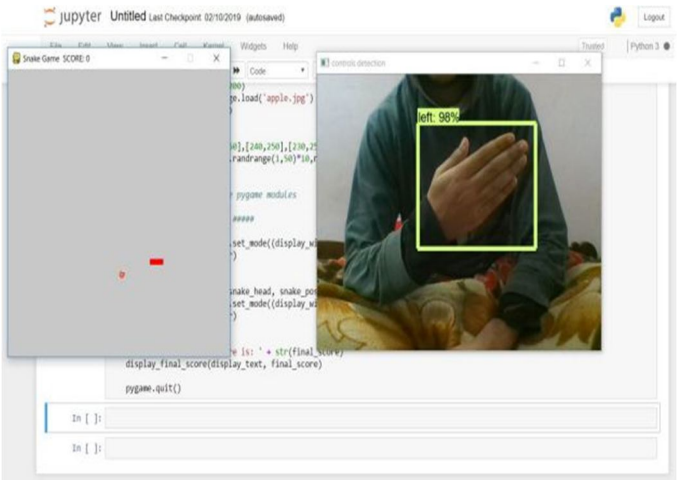
3) Moreover, during the estimation of the way the snake can move inside the FOR in which it presently lie.

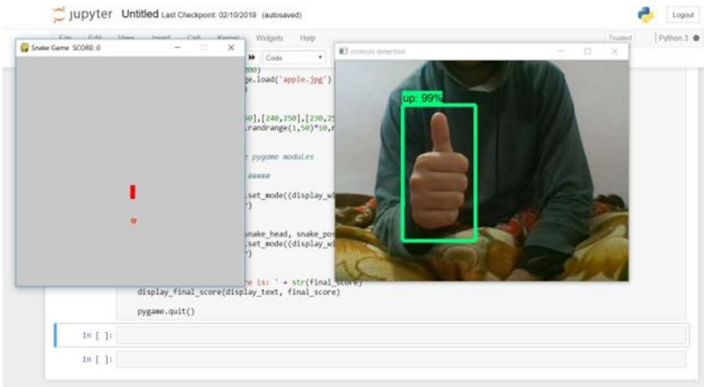
©IJRASET: All Rights are Reserved

IV. RESULTS

We have now reached the conclusion of our Python Snake Game article. I want to believe that you are clear with everything that has been imparted to you in this article.





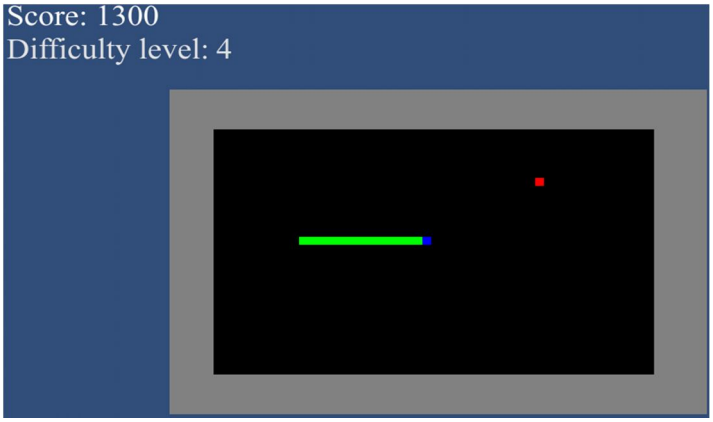


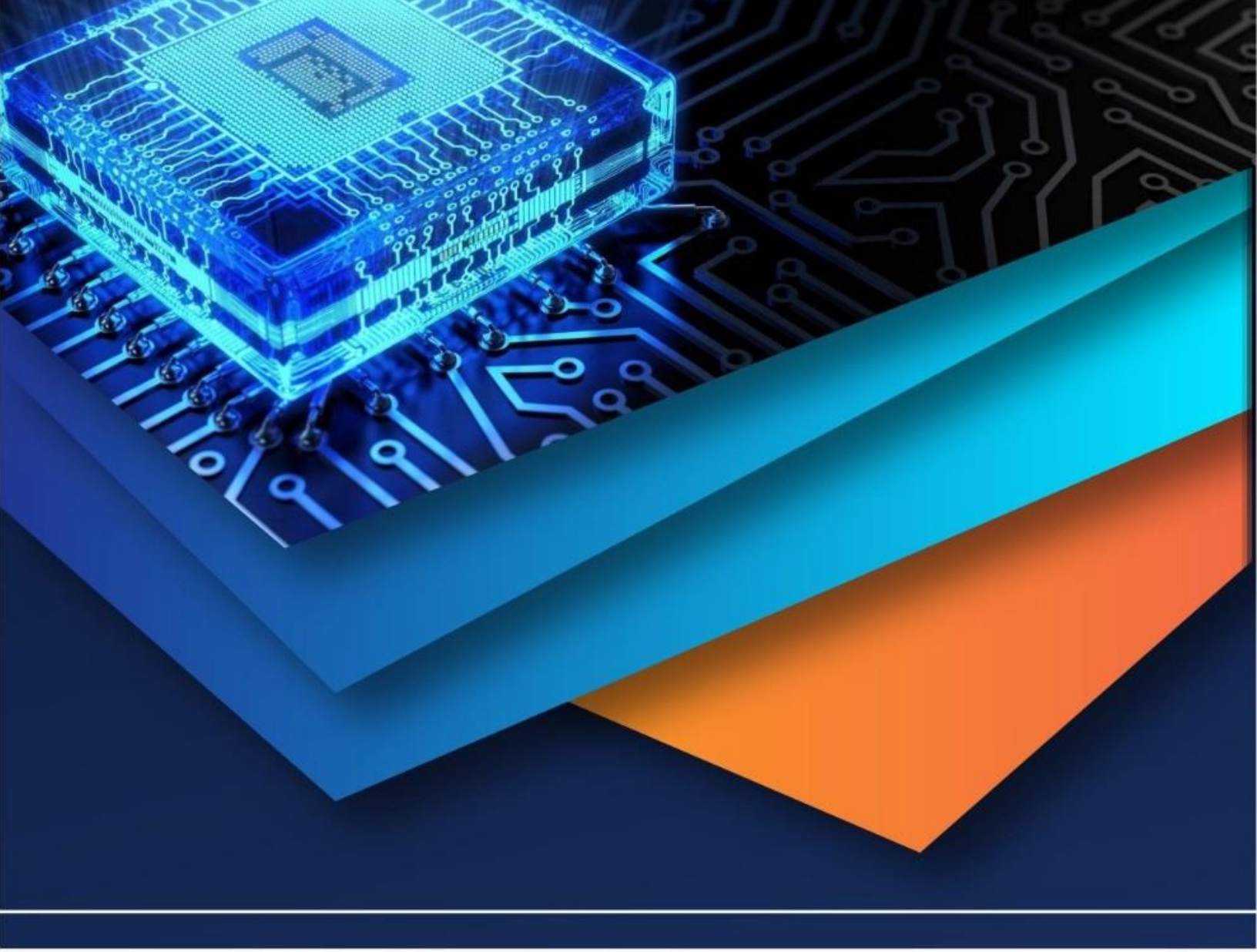
V. CONCLUSION

Active Shape Models have been proved to provide a reasonable premise for realtime include following, primarily in the 2D space. The procedure's ability to stretch out to 3D is a major concern. The Point Distribution Model that was used effectively expands into three aspects, However, in this scenario, the capacity to follow only 2D video sequences for input is a constraint. Sound system vision or data concealment are two possible paths**.**

REFERENCES

[1] B. Dorner. Hand shape identication and following for gesture based communication understanding. Taking a gander at People Workshop, Chambery, France, 1993.
[2] J.M. Rehg and T. Kanade. Visual following of high DOF verbalized constructions: An application to human hand following. In Proceedings of the European Conference on Computer Vision, volume II, pages 35{45, Stockholm, Sweden, 1994. Springer-Verlag.
[3] A. Blake, R. Curwen, and A. Zisserman. A system for spatiotemporal control in the following of visual shapes. Global Journal of Computer Vision, 11(2):127{145, 1993.
[4] C. Kervrann and H. Heitz. Vigorous following of stochastic deformable models in long picture arrangements. In Proceedings of the IEEE International Conference on Image Processing, volume III, pages 88{92, Austin, Texas, US, 1994. IEEE Computer Society Press.
[5] T.F. Cootes and C.J. Taylor. Dynamic shape models - 'Savvy Snakes'. In Proceedings of the British Machine Vision Conference, pages 266{275, Leeds, UK, 1992. SpringerVerlag.
[6] A. Slope, T.F. Cootes, and C.J. Taylor. A nonexclusive framework for picture understanding utilizing exible formats. In Proceedings of the British Machine Vision Conference, pages 276{285, Guildford, UK, 1993. BMVA Press





10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)