iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** I **Month of publication:** January 2022

DOI: https://doi.org/10.22214/ijraset.2022.39860

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 I Jan 2022- Available at www.ijraset.com

Hill Climb Game Play with Webcam Using OpenCV

Chandan Kumar[1], Ms. Akanksha Kochhar[2]
[1]*Bachelor of Technology in, Computer Science and Engineering*
[2]*Asst. Prof. Department of Computer Science, Maharaja Agrasen Institute of Technology, New Delhi*

Abstract: ***Computer vision is a process by which we can understand how the images and videos are stored and manipulated, also it helps in the process of retrieving data from either images or videos. Computer Vision is part of Artificial Intelligence. Computer-Vision plays a major role in Autonomous cars, Object detections, robotics, object tracking, etc.***
OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products.
It comes with a highly improved deep learning (dnn) module. This module now supports a number of deep learning frameworks, including Caffe, TensorFlow, and Torch/PyTorch. This does allow us to take our models trained using dedicated deep learning libraries/tools and then efficiently use them directly inside our OpenCV scripts.
MediaPipe is a framework mainly used for building audio, video, or any time series data. With the help of the MediaPipe framework, we can build very impressive pipelines for different media processing functions like Multi-hand Tracking, Face Detection, Object Detection and Tracking, etc.

I. INTRODUCTION

In this project i have Developed a hand Gesture controlling game in python using OpenCV and Mediapipe. With the help of this system you can move your hill climbing gaming car with the movement of your Hands. With the help of this system you can enjoy virtual drives. I will use the Mediapipe python library to detect hand landmarks. I will be using a Holistic model from Mediapipe Solutions to detect all the face and hand landmarks. I will also be seeing how we can access different landmarks of the face and hands which can be used for different computer vision applications such as sign language detection, drowsiness detection. Hand detection is done on a frame-by-frame basis and the system does not automatically track hand across frames. However, this type of inter-frame tracking is useful as it can enable user interaction where we need to track a hand across a window.
Basically, the MediaPipe uses a single-shot palm detection model and once that is done it performs precise key point localization of 21 3D palm coordinates in the detected hand region.
The MediaPipe pipeline utilizes multiple models like a palm detection model that returns an oriented hand bounding box from the full image. The cropped image region is fed to a hand landmark model defined by the palm detector and returns high-fidelity 3D hand key points.

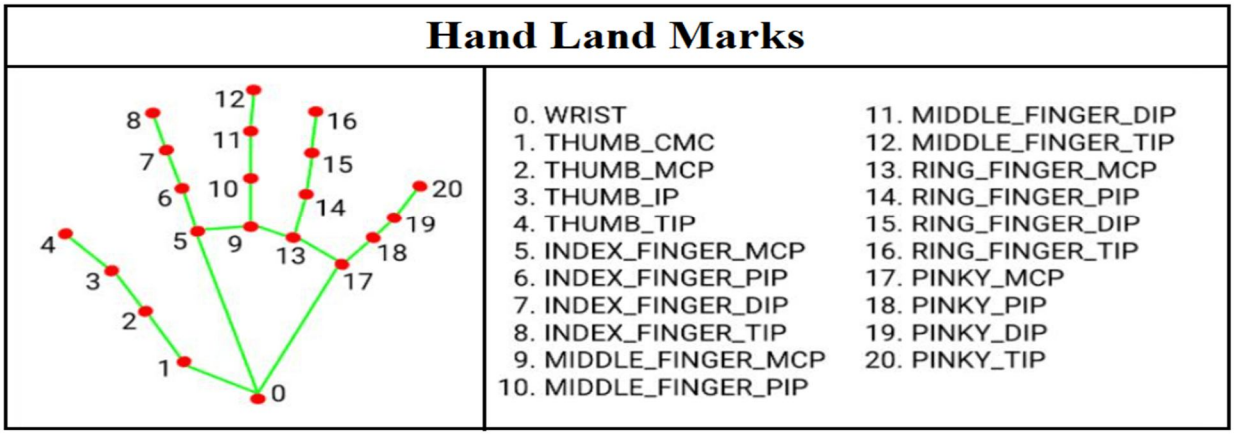


Fig 1. Hand Landmark Model and their Indices.

A. *Need of Study*

1) OpenCV is an open-source library mainly used for computer vision, image processing, and machine learning.
2) It gives better output for real-time data, with the help of OpenCV, we can process images and videos so that the implemented algorithm can be able to identify objects such as cars, traffic signals, number plates, etc., and faces, or even handwriting of a human.
3) With the help of other data analysis libraries, OpenCV is capable of processing the images and videos according to one's desire.
4) With the help of the MediaPipe framework, an impressive ML pipeline can be built for instance of inference models like TensorFlow, TFLite, and also for media processing functions.

B. *Scope of the Study*

Let's rather talk about the scope of image processing in industries, which is rather wide and diverse ranging from medical applications to space research. OpenCV is an optimized library for performing image processing and its usage totally depends on the use-case. An open source community is behind its development and regular updates are made when new algorithms are invented. On the other hand proprietary software like MATLAB's Image processing toolbox is also used in many industrie. They are highly reliable and have excellent documentation for users.

In robotics, it is preferred using OpenCV. It can be tweaked using code and comparatively is less computationally costly than using MATLAB. It's generally easier to run a python code on an on-board linux computer than MATLAB for real-time applications.

C. *Objective of the study*

OpenCV is a great tool for image processing and performing computer vision tasks. It is an open-source library that can be used to perform tasks like face detection, objection tracking, landmark detection, and much more. It supports multiple languages including python, java C++. Although, For this article, we will be limiting to python only.

The library is equipped with hundreds of useful functions and algorithms, which are all freely available to us. Some of these functions are really common and are used in almost every computer vision task. Whereas many of the functions are still unexplored and haven't received much attention yet.

D. *Computer vision and Digital Image Processing*

The sense of sight is arguably the most important of man's five senses. It provides a huge amount of information about the world that is rich in detail and delivered at the speed of light. However, human vision is not without its limitations, both physical and psychological. Through digital imaging technology and computers, man has transcended many visual limitations. He can see into far galaxies, the microscopic world, the subatomic world, and even "observe" infra-red, x-ray, ultraviolet and other spectra for medical diagnosis, meteorology, surveillance, and military uses, all with great success. While computers have been central to this success, for the most part man is the sole interpreter of all the digital data. For a long time, the central question has been whether computers can be designed to analyze and acquire information from images autonomously in the same natural way humans can. According to Gonzales and Woods, this is the province of computer vision, which is that branch of artificial intelligence that ultimately aims to "use computers to emulate human vision, including learning and being able to make inferences and taking actions based on visual inputs." The main difficulty for computer vision as a relatively young discipline is the current lack of a final scientific paradigm or model for human intelligence and human vision itself on which to build a infrastructure for computer or machine learning. The use of images has an obvious drawback. Humans perceive the world in 3D, but current visual sensors like cameras capture the world in 2D images. The result is the natural loss of a good deal of information in the captured images. Without a proper paradigm to explain the mystery of human vision and perception, the recovery of lost information reconstruction of the world from 2D images represents a difficult hurdle for machine vision. However, despite this limitation, computer vision has progressed, riding mainly on the remarkable advancement of decades-old digital image processing techniques, using the science and methods contributed by other disciplines such as optics, neurobiology, psychology, physics, mathematics, electronics, computer science, artificial intelligence and others. Computer vision techniques and digital image processing methods both draw the proverbial water from the same pool, which is the digital image, and therefore necessarily overlap. Image processing takes a digital image and subjects it to processes, such as noise reduction, detail enhancement, or filtering, for the purpose of producing another desired image as the end result. For example, the blurred image of a car registration plate might be enhanced by imaging techniques to produce a clear photo of the same so the police might identify the owner of the car. On the other hand, computer vision takes a digital image and subjects it to the same digital imaging techniques but for the purpose of analyzing and understanding what the image depicts.

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

For example, the image of a building can be fed to a computer and thereafter be identified by the computer as a residential house, a stadium, high-rise office tower, shopping mall, or a farm barn.

Russell and Norvig identified three broad approaches used in computer vision to distill useful information from the raw data provided by images. The first is the feature extraction approach, which focuses on simple computations applied directly to digital images to measure some usable characteristics, such as size. This relies on generally known image processing algorithms for noise reduction, filtering, object detection, edge detection, texture analysis, computation of optical flow, and segmentation, which techniques are commonly used to pre-process images for subsequent image analysis. This is also considered an uninformed approach.

The second is the recognition approach, where the focus is on distinguishing and labelling objects based on knowledge of characteristics that sets of similar objects have in common, such as shape or appearance or patterns of elements, sufficient to form classes. Here computer vision uses the techniques of artificial intelligence in knowledge representation to enable a "classifier" to match classes to objects based on the pattern of their features or structural descriptions. A classifier has to "learn" the patterns by being fed a training set of objects and their classes and achieving the goal of minimizing mistakes and maximizing successes through a step-by-step process of improvement. There are many techniques in artificial intelligence that can be used for object or pattern recognition, including statistical pattern recognition, neural nets, genetic algorithms and fuzzy systems. The third is the reconstruction approach, where the focus is on building a geometric model of the world suggested by the image or images and which is used as a basis for action. This corresponds to the stage of image understanding, which represents the highest and most complex level of computer vision processing. Here the emphasis is on enabling the computer vision system to construct internal models based on the data supplied by the images and to discard or update these internal models as they are verified against the real world or some other criteria. If the internal model is consistent with the real world, then image understanding takes place. Thus, image understanding requires the construction, manipulation and control of models and at the moment relies heavily upon the science and technology of artificial intelligence.

E. OpenCV

OpenCv is a widely used tool in computer vision. It is a computer vision library for real-time applications, written in C and C++, which works with the Windows, Linux and Mac platforms. It is freely available as open source software from http://sourceforge.net/projects/opencvlibrary/. OpenCv was started by Gary Bradsky at Intel in 1999 to encourage computer vision research and commercial applications and, side-by-side with these, promote the use of ever faster processors from Intel. OpenCV contains optimized code for a basic computer vision infrastructure so developers do not have to re-invent the proverbial wheel. The reference documentation for OpenCV is at http://opencv.willowgarage.com/documentation/index.html. The basic tutorial documentation is provided by Bradsky and Kaehler. According to its website, OpenCV has been downloaded more than two million times and has a user group of more than 40,000 members. This attests to its popularity.

A digital image is generally understood as a discrete number of light intensities captured by a device such as a camera and organized into a two-dimensional matrix of picture elements or pixels, each of which may be represented by number and all of which may be stored in a particular file format (such as jpg or gif). OpenCV goes beyond representing an image as an array of pixels. It represents an image as a data structure called an IplImage that makes immediately accessible useful image data or fields, such as:

1) width – an integer showing the width of the image in pixels
2) height – an integer showing the height of the image in pixels
3) imageData – a pointer to an array of pixel values
4) nChannels – an integer showing the number of colors per pixel
5) depth – an integer showing the number of bits per pixel
6) widthStep – an integer showing the number of bytes per image row
7) imageSize – an integer showing the size of in bytes.
8) roi – a pointer to a structure that defines a region of interest within the image.

OpenCV has a module containing basic image processing and computer vision algorithms. These include:

a) smoothing (blurring) functions to reduce noise,
b) dilation and erosion functions for isolation of individual elements,
c) floodfill functions to isolate certain portions of the image for further processing,
d) filter functions, including Sobel, Laplace and Canny for edge detection,

e) Hough transform functions for finding lines and circles,
f) Affine transform functions to stretch, shrink, warp and rotate images,
g) Integral image function for summing subregions (computing Haar wavelets),
h) Histogram equalization function for uniform distribution of intensity values,
i) Contour functions to connect edges into curves,
j) Bounding boxes, circles and ellipses,
k) Moments functions to compute Hu's moment invariants,
l) Optical flow functions (Lucas-Kanade method),
m) Motion tracking functions (Kalman filters), and
n) Face detection/ Haar classifier.

OpenCV also has an ML (machine learning) module containing well known statistical classifiers and clustering tools. These include:

- Normal/ naïve Bayes classifier
- Decision trees classifier,
- Boosting group of classifiers,
- Neural networks algorithm, and
- Support vector machine classifier.

Pattern Recognition and Classifiers In computer vision a physical object maps to a particular segmented region in the image from which object descriptors or features may be derived. A feature is any characteristic of an image, or any region within it, that can be measured. Objects with common features may be grouped into classes, where the combination of features may be considered a pattern. Object recognition may be understood to be the assignment of classes to objects based on their respective patterns. The program that does this assignment is called a classifier.
The most important step is the design of the formal descriptors because choices have to be made on which characteristics, quantitative or qualitative, would best suit the target object and in turn determines the success of the classifier. In statistical pattern recognition, quantitative descriptions called features are used. The set of features constitutes the pattern vector or feature vector, and the set of all possible patterns for the object form the pattern space X (also known as feature space). Quantitatively, similar objects in each class will be located near each other in the feature space forming clusters, which may ideally be separated from dissimilar objects by lines or curves called discrimination functions. Determining the most suitable discrimination function or discriminant to use is part of classifier design. A statistical classifier accepts n features as inputs and gives 1 output, which is the classification or decision about the class of the object. The relationship between the inputs and the output is a decision rule, which is a function that puts in one space or subset those feature vectors that are associated with a particular output. The decision rule is based on the particular discrimination function used for separating the subsets from each other. The ability of a classifier to classify objects based on its decision rule may be understood as classifier learning, and the set of the feature vectors (objects) inputs and corresponding outputs of classifications (both positive and negative results) is called the training set. It is expected that a well-designed classifier should get 100% correct answers on its training set. A large training set is generally desirable to optimize the training of the classifier, so that it may be tested on objects it has not encountered before, which constitutes its test set. If the classifier does not perform well on the test set, modifications to the design of the recognition system may be needed.

F. Moment Invariants

Moment Invariants As mentioned previously, feature extraction is one approach used in computer vision. According to A.L.C. Barczak, feature extraction refers to the process of distilling a limited number of features that would be sufficient to describe a large set of data, such as the pixels in a digital image . The idea is to use the features as a unique representation of the image.
Since a digital image is a two-dimensional matrix of pixel values, region-based object descriptions are affected by geometric transformations, such as scaling, translation, and rotation. For example, the numerical features describing the shape of a 2D object would change if the shape of the same object changes as seen from a different angle or perspective. However, to be useful in computer vision applications, object descriptions must be able to identify the same object irrespective of its position, orientation, or distortion. One of the most popular quantitative object descriptors are moments. The concept of statistical characteristics or moments that would be indifferent to geometric transformations was first formulated by Hu in 1962.

Moments are polynomials of increasing order that describe the shape of a statistical distribution [10]. The order of a moment is indicated by its exponent. The geometric moments of different orders represent different spatial characteristics of the image intensity distribution. A set of moments can thus form a global shape descriptor of an image.

G. The Special Role of Heuristic Rules

Even if computer vision achieves pattern recognition using complex and sophisticated techniques in artificial intelligence, there is a place in it for the application of heuristic rules. A heuristic is knowledge that is empirical but vague. Sometimes in problems of high complexity looking for an exact and exhaustive solution is impracticable or overly expensive, if not impossible. Sometimes no algorithm is known or can be found to provide the solution. Often the intention is only one of exploration where an acceptable estimate or approximation of a possible solution is all that is required for the present. In these cases, a good heuristic or rule of thumb is all that is required to have a good working solution. A computer vision application can be made to capture images containing hand gestures from the webcam camera, detect and recognize the hand gestures, and output the applicable commands to navigate the game.

II. LITERATURE SURVEY

Bhumika Gupta (2017) proposed object detection is a well-known computer technology connected with computer vision and image processing that focuses on detecting objects or its instances of a certain class (such as humans, flowers, animals) in digital images and videos. There are various applications of object detection that have been well researched including face detection, character recognition, and vehicle calculator. Object detection can be used for various purposes including retrieval and surveillance. In this study, various basic concepts used in object detection while making use of the OpenCV library of python 2.7, improving the efficiency and accuracy of object detection are presented.

Kartik Umesh Sharma (2017) proposed an object detection system that finds objects of the real world present either in a digital image or a video, where the object can belong to any class of objects namely humans, cars, etc. In order to detect an object in an image or a video the system needs to have a few components in order to complete the task of detecting an object, they are a model database, a feature detector, a hypothesis and a hypothesis verifier. This paper presents a review of the various techniques that are used to detect an object, localise an object, categorise an object, extract features, appearance information, and many more, in images and videos.

Mukesh Tiwari (2017) presented object detection and tracking as one of the critical areas of research due to routine change in motion of object and variation in scene size, occlusions, appearance variations, and ego-motion and illumination changes. Specifically, feature selection plays a vital role in object tracking. It is related to many real time applications like vehicle perception, video surveillance and so on. In order to overcome the issue of detection, tracking related to object movement and appearance. Most of the algorithm focuses on the tracking algorithm to smoothen the video sequence. On the other hand, few methods use the prior available information about object shape, color, texture and so on. Tracking algorithm which combines the above stated parameters of objects is discussed and analyzed in this research.

Abdul Muhsin M (2019) proposed that everybody deserve to live independently, especially those who disabled, with the last decades, technology gives attention to disabled to make them control their life as much as possible. In this work, an assistive system for blind is suggested, to let him know what is around him, by using YOLO for detecting objects within images and video streams quickly based on deep neural networks to make accurate detection, and OpenCV under Python using Raspberry Pi3. The obtained results indicated the success of the proposed model in giving blind users the capability to move around in unfamiliar indoor outdoor environments, through a user friendly device by person and object identification model.

The goal of this paper is to analyze and review the previous approach towards object tracking and detection using video sequences through different phases. Also, identify the gap and suggest a new approach to improve the tracking of objects over video frames.

III. IMPLEMENTATION OF PROPOSED MODEL

A. Software Requirement Specification

1) Functional Requirements

The functional requirements for a system describe what a system does.

a) The developed system should recognize the hand movement in the video.
b) The system should control the game with respect to hand movement.
c) The system must provide the quality of service to the user.
d) The system must provide accuracy for virtual game control.

2) Normal Requirements

These are the requirements clearly stated by the customer hence requirement must be present for customer satisfaction.

a) N1: Movement of both hands that is right and left should be recognized and
b) N2: system should be able to perform in both high quality and low-quality cameras.
c) N3: Application should be able to match the stored patterns on input video

3) Expected Requirements

These requirements are implicit types of requirements. These requirements are not clearly stated by the user but even though the user expects them.

a) Exp1: Instead of only one hand either left and right application should act on both hand movements.
b) Exp2: Application should be user friendly and also easy to install.
c) Exp3: By using Deep learning bringing more accuracy in object recognition process.
d) Exp4: Application also recognizes other 3d objects too.

4) Nonfunctional Requirements

As the name suggest these are the requirements that are not directly interacted with specific functions delivered by the system.

a) *Functionality:* This software will deliver on the functional requirements.
b) *Availability:* This system will retrieve the hand movement regions only if the image contains hand overlapping on other body parts.
c) *Flexibility:* It provides the users to load the live streaming video easily.
d) *Learn ability:* The software is very easy to use and reduces the learning work.

B. Modules

1) *Pose Estimation:* Human pose estimation from video or a real-time feed plays a crucial role in various fields such as full-body gesture control, quantifying physical exercise, and sign language recognition. For example, it can be used as the base model for fitness, yoga, and dance applications. It finds its major part in augmented reality. Media Pipe Pose is a framework for high-fidelity body pose tracking, which takes input from RGB video frames and infers 33 3D landmarks on the whole human. Current state-of-the-art approach methods rely primarily on powerful desktop environments for inferencing, whereas this method outperforms other methods and achieves very good results in real-time.

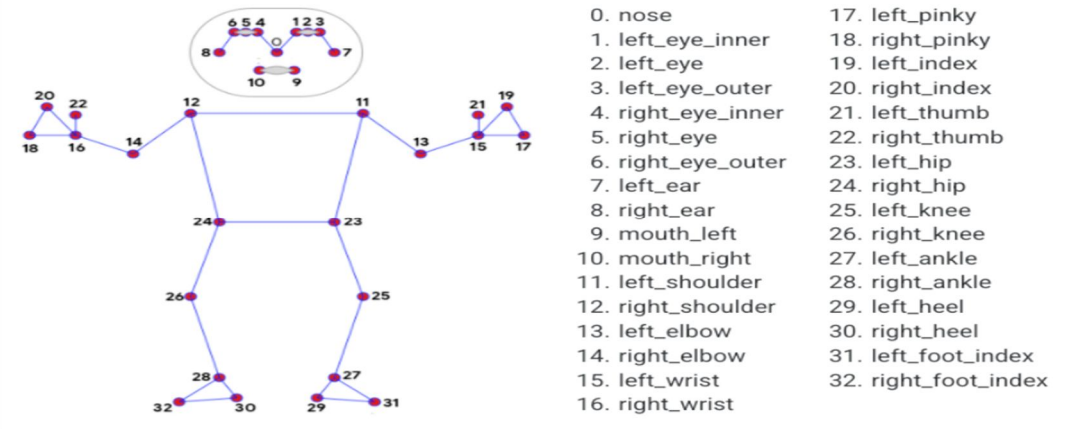


Fig 5. Human Pose landmarks.

2) *Preprocessing:* GrayScale An image is an array, or a matrix, of square pixels(picture elements) arranged in columns and rows. In an (8 bit) gray scale image each picture element has an assigned intensity that ranges from 0 to 255. A gray scale image is what people normally call a black and white image,but the name emphasizes that such an image will also include many shades of gray. A normal gray scale image has 8 bit colour depth- 256 gray scales. A " true color images 24 bit color depth 8 * 8 * 8 bits 256 * 256 * 256 colors = 16 million colors. Some gray scale images have more gray scales, for instance 16 bit =65536 gray scales. There are two general groups of images :vector graphics(or line art) and bitmaps(pixel based or images).

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

3) *Feature Extraction:* In pattern recognition in image processing, Feature extraction is a special form of dimensionality reduction. When the input data to an algorithm is too large to be processed and it is suspected to be notoriously redundant (much data,but not much information) then the input data will be transformed into a reduced representation set of features(also named feature vector). Transforming the input data into the set of features is called features extraction.
4) *Segmentation:* In this part the characters will be identified as letters and the image will be converted to text. After the image is cleaned up and becomes a binary image which contains only the text, the binary image is then saved and the memory is cleaned up. This step is very important to increase the speed of the system.

C. STEP

1) *STEP-1:* Import all the necessary libraries, In our case only three libraries are required.

```
import mediapipe as mp
import cv2
import pydirectinput
```

Fig 2. List of libraries.

2) *STEP-2:* Initializing Holistic model and Drawing utils for detecting and drawing landmarks on the image.

```
mp_drawing=mp.solutions.drawing_utils
mp_holistic=mp.solutions.holistic
mp_pose=mp.solutions.pose
```

Fig 3. Initializing Holistic model

3) *STEP-3:* Detecting Face and Hand landmarks from the image. Holistic model processes the image and produces landmarks for Face, Left Hand, Right Hand and also detects the Pose of the hand move.

a) Capture the frames continuously from the camera using OpenCV.
b) We will take a live feed from the webcam by changing the first line in the main function to "cap = cv2.VideoCapture(0)".
c) Convert the BGR image to an RGB image and make predictions using an initialized holistic model.
d) The predictions made by the holistic model are saved in the results variable from which we can access the landmarks using results.face_landmarks, results.right_hand_landmarks, results.left_hand_landmarks respectively.
e) Draw the detected landmarks on the image using the draw_landmarks function from drawing utils.
f) Use the resulting Image as pose movement according to the axis drawing on the image window.

```
cap=cv2.VideoCapture(0)
cap.set(3,560)
cap.set(4,400)
with mp_holistic.Holistic(min_detection_confidence=0.5, min_tracking_confidence=0.5) as holistic:
    while cap.isOpened():
        success,img=cap.read()
        img=cv2.flip(img,1)
        #color
        img=cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
        results=holistic.process(img)
        img=cv2.cvtColor(img,cv2.COLOR_RGB2BGR)
        height,width,_=img.shape

        try:
            right_hand = (results.pose_landmarks.landmark[mp_pose.PoseLandmark.LEFT_WRIST].x * width,
                          results.pose_landmarks.landmark[mp_pose.PoseLandmark.LEFT_WRIST].y * height)
            results.pose_landmarks.landmark[mp_pose.PoseLandmark.RIGHT_WRIST].y * height)
            y_mid=height//2

            pose="move"
            if(right_hand[1]<y_mid):
                pose="acc"
                pydirectinput.keyDown('right')
                pydirectinput.keyUp('left')
            elif(right_hand[1]>y_mid):
                pose="break"
                pydirectinput.keyDown('left')
```

Fig 4. Detecting Hand landmarks from the image.

D. Sequence Diagram

A sequence diagram is a graphical view of a scenario that shows object interaction in a time based sequence: what happens first, what happens next. Sequence diagrams establish the role of objects and help provide essential information to determine class responsibilities and interfaces. This type of diagram is best used during the early analysis phase in design because they are simple and easy to comprehend. Sequence diagrams are normally associated with use cases.

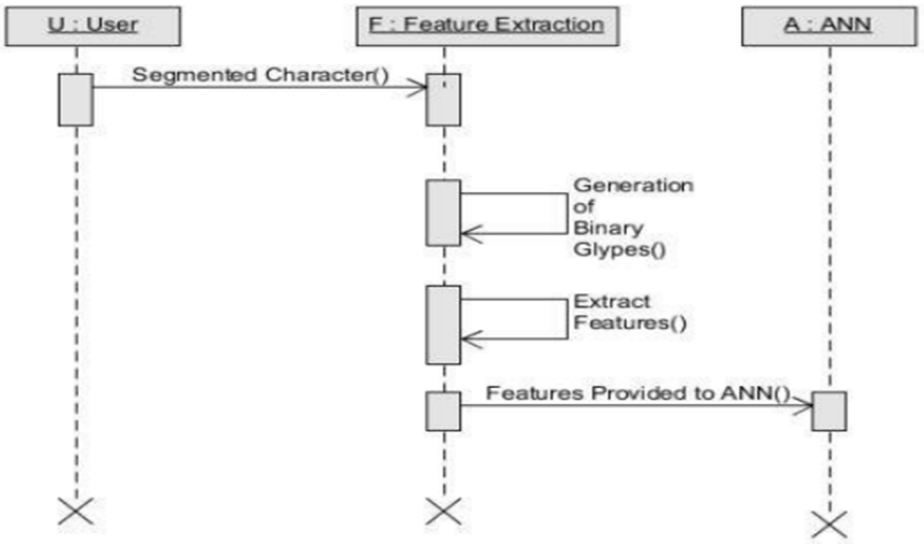


Fig 6: Sequence Diagram of Feature Extraction

E. Data Flow Diagram

Data flow diagram is also called a Bubble Chart is a graphical technique, which is used to represent information flow, and transformers that are applied when data moves from input to output. Data flow diagram represents system requirements clearly and identifies transformers that become programs in design. Data flow diagrams may further be partitioned into different levels to show detailed information flow.

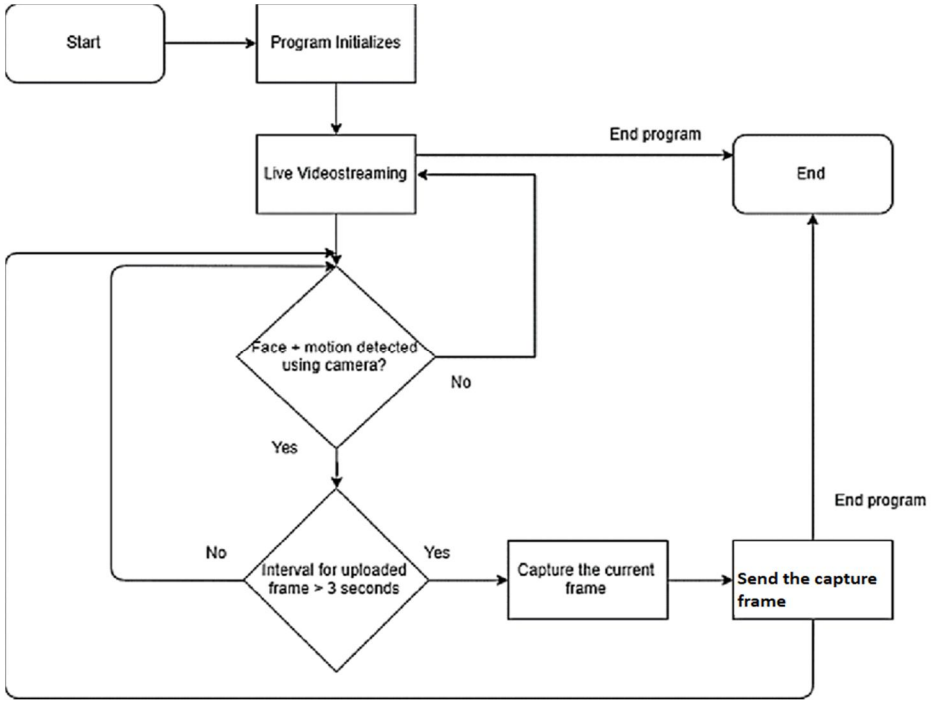


Fig 7: Control Flow Diagram

F. Proposed Algorithm

In this study, we, therefore, propose a gesture recognition algorithm using support vector machines (SVM) and histogram of oriented gradient (HOG) based on the previous work. Besides, we also use the CNN model to classify gestures. The goal of the algorithm is to detect gestures with real-time processing speed, minimize interference, and reduce the ability to capture unintentional gestures. The static gesture controls include on, off, up, and down in this study. Besides, the dynamic gestures in this study include the following:

- Toggle state switch is hand from spread state upwards, into grip state.
- Up order is the hand from outstretched state up to left.
- Down order is handed from outstretched state up to right.

1) *Detecting Object:* The technique has two requirements. The first requirement is to detect whether the image contains the object or not. The second requirement is to find out the position of the subject of the image. As introduced in the previous section, there are many algorithms that perform the task. In this study, the requirement is the accuracy of the results as well as being fast enough to operate for real-time applications. In the system, the action is the object detection operation. Therefore, it is necessary to select techniques with relatively fast calculation speed. Therefore, our implementation idea is to use the multiscale and sliding window technique to separate the image into ROI. We then crop the image from those ROI regions and extract their HOG feature. The SVM technique is used to classify whether an image contains an object or not.
2) *Tracking Object:* When the algorithm has detected the frame containing the object as well as the ROI area, the next thing is to lock and track the target when it is moving or may be partially deformed in the next frames. The use of the tracking object will be necessary since the user of the gesture will take place in a few seconds. If we continue to use classification and detection techniques to conclude, it will be difficult to achieve the desired processing speed or it may lead to false conclusions. For example, the action initiates ending the gesture that comes from two completely different objects. During the intermediate stage between starting and ending of gesture, the subject may be partially deformed. It is difficult to use detection techniques. The requirement for object tracking technique in this problem is fast adhesion with acceptable accuracy. In this study, we select a kernelized correlation filter (KCF) algorithm to track the object. The algorithm has good speed and consistent accuracy. It will not restore tracking target when losing its target that will reduce noise for the gesture control system.
3) *Object Classification Technique:* Object classification techniques are particularly applied for image processing. In artificial intelligence applications, the classifier requirements are to be able to distinguish the gestures that begin with each other. The required accuracy is high and this will ensure the desired level of control accuracy. Processing speed is not required too high since information containing objects is known and categorization does not occur continuously in every frame. When the object (gesture is detected) contains the target, the next task is to recognize them. Once gesture recognition begins, we continue to track the target and end gesture. To perform the task, we choose the convolutional neural network (CNN) model. CNN is used in many problems such as image recognition, video analysis, MRI images, and natural language processing.
4) *Object Detection and Partitioning Techniques:* We choose HOG characteristic extraction technique, combined with the SVM classification algorithm for the proposed model. The HOG characteristic is proposed by Lee and Chung. The typical HOG idea comes from the object of form and state. It can be characterized by the intensity and direction distribution of pixel value and is represented as a vector called a gradient vector. Gradient is a vector of elements that represent how fast a pixel of value changes. The gradient vector value brings a lot of useful information. It represents the change in the luminance value of pixels. The gradient vector value changes when the pixel is in the corner and edge areas of the object. Therefore, the HOG feature is effective in choosing the representation of posture. The essence of the HOG method is to use information about the distribution of intensity gradients or edge directions to describe local objects in an image. The HOG operators are implemented by dividing an image into subregions calling cells. We will compute a histogram of directions of gradients for points for each cell. To combine the histograms together, we get a representation of the original image. To enhance recognition performance, local histograms can be normalized for contrast by calculating an intensity threshold in an area larger than the cell calling blocks.
5) *Gradient:* Descent The goal of all supervised machine learning algorithms is to best estimate a target function (f) that maps input data (X) onto output variables (Y). This describes all classification and regression problems. Some machine learning algorithms have coefficients that characterize the algorithm's estimate for the target function (f). Different algorithms have different representations and different coefficients, but many of them require a process of optimization to find the set of coefficients that result in the best estimate of the target function.

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

The evaluation of how close a fit a machine learning model is to the target function can be calculated a number of different ways, often specific to the machine learning algorithm. The cost function involves evaluating the coefficients in the machine learning model by calculating a prediction for the model for each training instance in the dataset and comparing the predictions to the actual output values and calculating a sum or average error (such as the Sum of Squared Residuals or SSR in the case of linear regression). From the cost function a derivative can be calculated for each coefficient so that it can be updated using exactly the update equation described above. The cost is calculated for a machine learning algorithm over the entire training dataset for each iteration of the gradient descent algorithm. One iteration of the algorithm is called one batch and this form of gradient descent is referred to as batch gradient descent. Batch gradient descent is the most common form of gradient descent described in machine learning.

a) *Learning Rate:* The size of these steps is called the learning rate. With a high learning rate we can cover more ground each step, but we risk overshooting the lowest point since the slope of the hill is constantly changing. With a very low learning rate, we can confidently move in the direction of the negative gradient since we are recalculating it so frequently. A low learning rate is more precise, but calculating the gradient is time-consuming, so it will take us a very long time to get to the bottom.

b) *Cost Function:* A Loss Functions tells us "how good" our model is at making predictions for a given set of parameters. The cost function has its own curve and its own gradients. The slope of this curve tells us how to update our parameters to make the model more accurate. Now let's run gradient descent using our new cost function. There are two parameters in our cost function we can control: m (weight) and b (bias). Since we need to consider the impact each one has on the final prediction, we need to use partial derivatives. We calculate the partial derivatives of the cost function with respect to each parameter and store the results in a gradient.

6) *Softmax:* In mathematics, the softmax function, also known as softargmax or normalized exponential function, is a function that takes as input a vector of K real numbers, and normalizes it into a probability distribution consisting of K probabilities proportional to the exponentials of the input numbers. That is, prior to applying softmax, some vector components could be negative, or greater than one; and might not sum to 1; but after applying softmax, each component will be in the interval, and the components will add up to 1, so that they can be interpreted as probabilities. The function is usually used to compute losses that can be expected when training a data set. Known use-cases of softmax regression are in discriminative models such as CrossEntropy and Noise Contrastive Estimation. These are only two among various techniques that attempt to optimize the current training set to increase the likelihood of predicting the correct word or sentence. (We will touch upon the aforementioned techniques in the next few posts, so stay tuned for those.) If you look at it from the outset, the definition may sound off as trivial but in the realm of Machine Learning and NLP, this regression function has been useful as a baseline comparator. Researchers who design new solutions have to carry out experimentation keeping the softmax results as a reference. However, it should be noted that softmax is not ideally used as an activation function like Sigmoid or ReLU (Rectified Linear Units) but rather between layers which may be multiple or just a single one. Implementing the code is extraordinarily easy and the fun part is that it's only one line considering we have the necessary Python helper functions at our disposal. In the Github gist below I have used both Numpy and Tensorflow to write the Softmax function as explained in the previous section. The two mentioned libraries are popularly used to carry out mathematical and neural network-related operations. Furthermore, the larger input components will correspond to larger probabilities. Softmax is often used in neural networks, to map the non-normalized output of a network to a probability distribution over predicted output classes.

IV. EXPERIMENTAL RESULTS

A. Source Code

```
import mediapipe as mp
import cv2
import pydirectinput

mp_drawing=mp.solutions.drawing_utils
mp_holistic=mp.solutions.holistic
mp_pose=mp.solutions.pose

cap=cv2.VideoCapture(0)
cap.set(3,560)
```



```
cap.set(4,400)
with mp_holistic.Holistic(min_detection_confidence=0.5, min_tracking_confidence=0.5) as holistic:
    while cap.isOpened():
        success,img=cap.read()
        img=cv2.flip(img,1)
        #color
        img=cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
        results=holistic.process(img)
        img=cv2.cvtColor(img,cv2.COLOR_RGB2BGR)
        height,width,_=img.shape
        try:
            right_hand = (results.pose_landmarks.landmark[mp_pose.PoseLandmark.LEFT_WRIST].x * width,
                          results.pose_landmarks.landmark[mp_pose.PoseLandmark.LEFT_WRIST].y * height)
            results.pose_landmarks.landmark[mp_pose.PoseLandmark.RIGHT_WRIST].y * height)
            y_mid=height//2

            pose="move"
            if(right_hand[1]<y_mid):
                pose="acc"
                pydirectinput.keyDown('right')
                pydirectinput.keyUp('left')
            elif(right_hand[1]>y_mid):
                pose="break"
                pydirectinput.keyDown('left')
                pydirectinput.keyUp('right')
        except:
            pass
        cv2.putText(img,pose,(20,8),cv2.FONT_HERSHEY_PLAIN,2,(255,255,0),2)
        cv2.line(img,(0,y_mid),(width,y_mid),(255,0,255),2)
        cv2.imshow('CarGame',img)
        if cv2.waitKey(1)==ord('q'):
            break;
cap.release()
cv2.destroyAllWindows()
```

B. Experimental Outputs

When an image is fed into a trained model the following caption is generated:

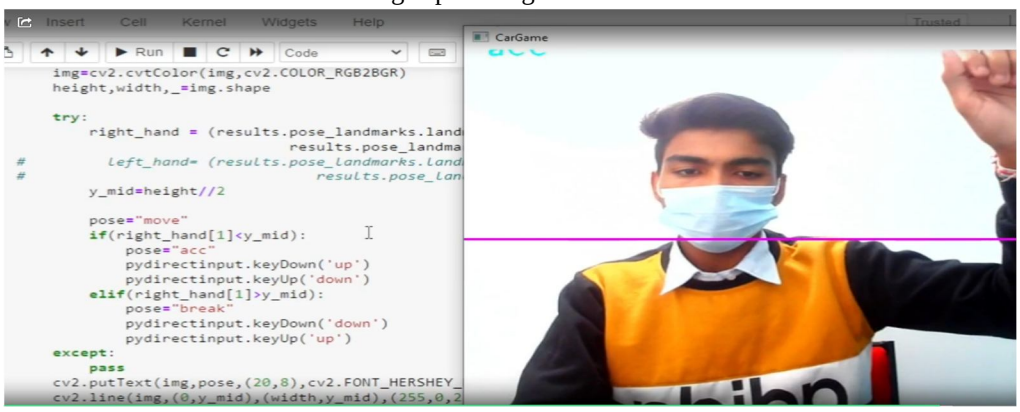


Fig: 8 prediction output

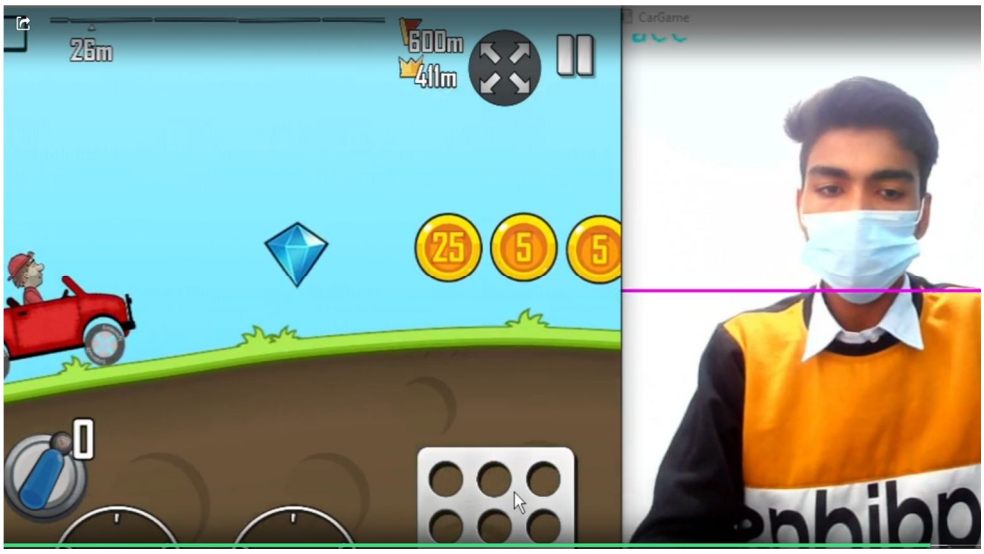


Fig: 9 prediction output.

V. CONCLUSION AND FUTURE SCOPE

A. Conclusion

Real-time Hand Gesture Recognition is one of the most common problems in computer vision. There are many techniques for face detection and face landmarks drawing. The most efficient techniques are created with the help of deep learning models. But if we try to make the models from scratch, it requires huge computational power, sophisticated knowledge as well as a dataset. In most cases, it's a problem for the common people. Mediapipe library is amazing in making the difficult task easy for us. The library facilities a customized built-in model. In this article, we have just shown the simple and easy process of face detection and face landmarks drawing using MediaPipe. In the next articles, many more easy and useful techniques will be shown with the help of MediaPipe.

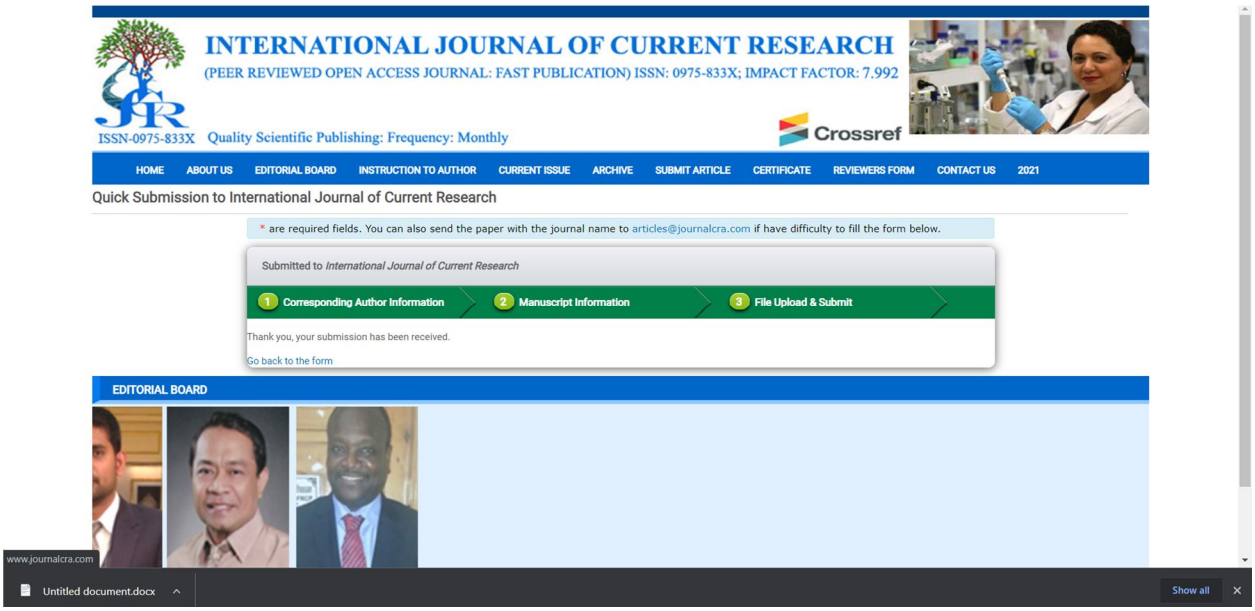
B. Final Thoughts

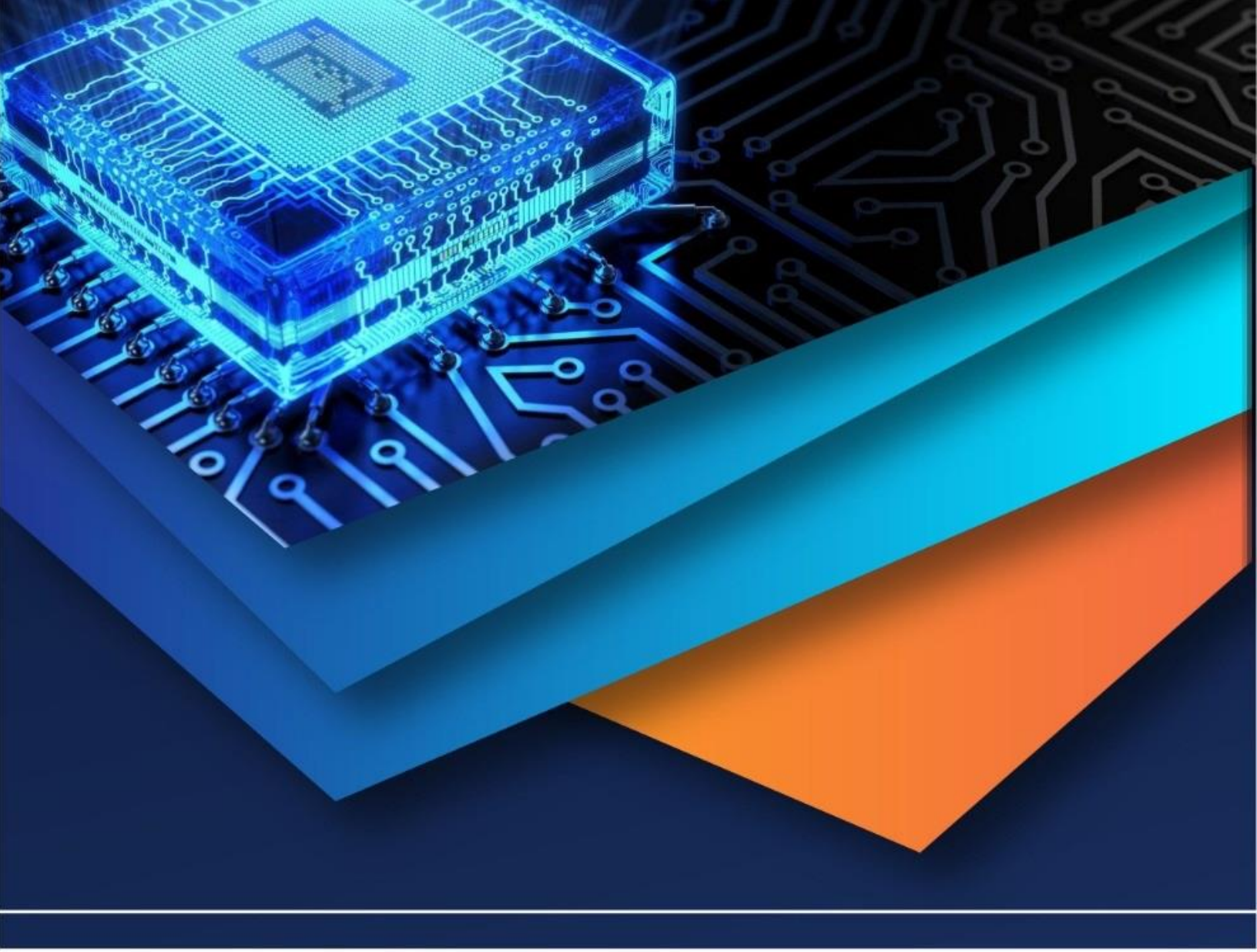
As AI algorithms continue to mature (accuracy, speed), there is potential to leverage these advances in building better interactions. This could be generative interfaces that tailor content to the user or predict their intended interaction, models that enable new types of vision-based interactions, conversational UI etc. Across these areas, it is increasingly important to study the mechanics of such interactions, rigorously test these interactions and create design patterns that inform the use of AI models as first class citizens in interaction design.

REFERENCES

[1] Md. Hasanuzzaman, T. Zhang, V. Ampornaramveth, H. Gotoda, Y. Shirai, and H. Ueno, "Adaptive visual gesture recognition for human-robot interaction using a knowledge-based software platform", Robotics and Autonomous Systems, vol. 55, Issue 8, 31 August 2007, pp. 643-657.
[2] R.C. Gonzalez and R.E. Woods, Digital Image Processing, Prentice Hall, 2008.
[3] M. Sonka, V. Hlavac, and R. Boyle, Image Processing, Analysis, and Machine Vision, PWS Publishing, 1999.
[4] R. Szeliski, Computer Vision: Algorithms and Applications, Springer, 2011.
[5] S.E. Umbaugh, Digital Image Processing and Analysis, CRC Press, 2011.
[6] S. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, Prentice Hall, 2010.
[7] OpenCv, http://www.willowgarage.com/pages/software/opencv, accessed 11 May 2011.
[8] A.L.C. Barczak, Computer Vision: Study Guide, IIMS, Massey University, 2010.
[9] G. Bradsky and A. Kaehler, Learning OpenCV: Computer Vision with the OpenCV Library, O' Reilly, 2008.
[10] J.R. Parker, Algorithms for Image Processing and Computer Vision, Wiley Publishing, 2011
[11] R. Mukundan and K.R. Ramakrishnan, Moment Functions in Image Analysis, World Scientific, 1998.
[12] M.K. Hu, "Visual pattern recognition by moment invariants," IRE Transactions on Information Theory, vol. 8, pp. 179-187, 1962.
[13] Y. Freund and R.E. Schapire, "A short introduction to boosting," Journal of Japanese Society for Artificial Intelligence, vol. 14, no. 5, pp. 771-780, 1999.
[14] A.L.C. Barczak, Computer Vision: Study Guide, IIMS, Massey University, 2010.
[15] G. Bradsky and A. Kaehler, Learning OpenCV: Computer Vision with the OpenCV Library, O' Reilly, 2008.
[16] J.R. Parker, Algorithms for Image Processing and Computer Vision, Wiley Publishing, 2011
[17] R. Mukundan and K.R. Ramakrishnan, Moment Functions in Image Analysis, World Scientific, 1998. [12] M.K. Hu, "Visual pattern recognition by moment invariants," IRE Transactions on Information Theory, vol. 8, pp. 179-187, 1962. [13] Y. Freund and R.E. Schapire, "A short introduction to boosting," Journal of Japanese Society.

Proof of Research Paper Publishing





10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY