2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N) | 978-1-6654-3811-7/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICAC3N53548.2021.9725638

Object Detection using OpenCV and Python

Ayushi Sharma
B.Tech CSE(AI&ML)
Galgotias University
Greater Noida, India
ayushish2000@gmail.com_

Jyotsna Pathak
B.Tech CSE
Galgotias University
Greater Noida, India
jyotsnapathak24@gmail.com_

Muskan Prakash
B.Tech CSE
Galgotias University
Greater Noida, India
_asmaa.muskan@gmail.com

J N Singh
Galgotias University
Greater Noida, India
drjnsingh@galgotiasuniversity.edu.in

Abstract- The paper has covered topics ranging from how artificial intelligence and machine learning algorithms help in object detection to how OpenCV is such a useful tool for beginners who wish to learn how real time object identification and tracking can be done. It also shows the flexibility of a tracking system to a moving camera, ideal for automotive safety applications. Image identification makes use of techniques like detection of an object, its recognition, and segmentation. The use of artificial intelligence and machine learning enhances the rate of processing the data and maintaining the standard of the outcome. Example, by using artificial intelligence, we can very easily complete difficult tasks.

Keywords—Pedestrian tracking, Moving vehicle, Intelligent vehicles, Unattended driving systems, Intelligent driving decisions

I. INTRODUCTION

OpenCV is a library catering to thousands of ml algorithms and hundreds of functions that support these algorithms. OpenCV approach includes Python, C++ and java and runs on all desktop and mobile systems. It stands for Open-Source Computer Vision Library. OpenCV contains many modules including module for image processing, for identification of object, and ml. By the use of it, we achieve, constrict, build up, replace, retrieve information. The new algorithms, that represents the software is categorized into groups, that include features, learning, and mixture of both known as hybrid. Yet, this tracking has many uses in different areas, some of which are monitoring traffic flows, speeds on different lanes, latest innovations in robotics, supervision, security and video connection. Therefore, this application requires optimal usage of computation, basic transmission, and precision over the network. Returns related to computation and transmission depend upon quantity as well as variety of coordination between cameras used to collect data, distribution and checking to verify the verdict so as to make the estimation errors minimum.

II. WORKING EXAMPLES OF OBJECT IDENTIFICATION:

Nowadays, video identification is established across many domains of corporations. Its usage ranges from video surveillance, sports broadcasting, electrical cars, robot navigation and many more.

How Video Object Detection works:
So, how do we detect a moving object?
The purpose is to record the coordinates of the entity in motion and then encircling the object in the video.

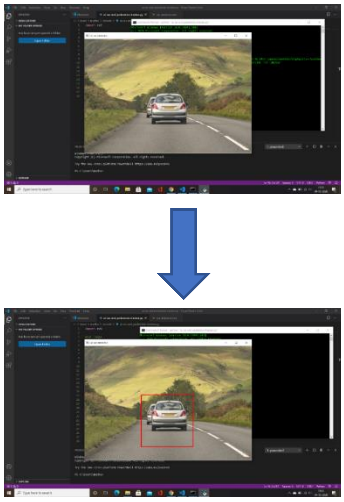

Figure 1: Object Recognition

Then we want our model to detect the operating entity as shown in Figure1. The unit in motion (i.e., car) will be recognized and then encircled as a rectangle.

Either, a model can be trained for object identification or trained model could be used which would identify it the data. Although, both are techniques used in learning. They need labelled information to teach the object identification model. Or we could also use frame differencing approach**.** Whenever an object is seen in motion, it means that the object is at a distinct position at each consecutive frame.

We presume that other than the object nothing moves in a pair of successive frames. This is how frame differencing works.

Cascading:

TrainCascadeObjectDetector is a classifier function. CascadeObject Detectors system object is in MATLAB's

ISBN: 978-1-6654-3811-7/21/$31.00 ©2021 IEEE

computer vision toolbox One of the uses is to build a custom classifier to detect categories of objects which are fixed, for instance face, human body, cars, etc. A group of classifiers arranged in stages is known as Cascade object detector. For instance, classification output from every expert E_n for input x_i is given as $E_n(x_i)$. To distinguish amongst training, $E_n(x_i)$ only two results expected that are written as +1 or −1, i.e., $E_n(x_i) \in [-1, +1]$. The joint opinion of experts is written as $L(x_i)$. It shows the combination as below:

$$L(x_i) = w_1E_1(x_i) + w_2E_2(x_i) + \cdot\cdot\cdot + w_nE_n(x_i)$$

where $E_1(x_i)$, $E_2(x_i)$, . . ., $E_n(x_i)$ represents the decisions from n experts and w_1, w_2, . . ., w_n are weights specified. The steps of for classifier training are:

1. Provided a set to train data with a pair of images (x_i, y_i), where x_i is a true /false picture, and y_i is the tag assigned to each image. The value for true pictures is 1, and that of false pictures is 0.

2. Setting up weights w_1, i= 1/2p, 1/2n for true and false pictures, where p and n are the count of true and false pictures, correspondingly.

3. For f = 1, . . ., F, where F is the number of levels of training and n sets of images.

4. Upgrade weight $w_{f+1,\, i} = w_{f,\, i}\beta^{\,1-fi}$ y, when $e_i = 0$ then x_i is sorted in a correct manner, $e_i = 1$ otherwise and $\beta_y = e_f/ (1 - e_f)$.

III. Image Recognition

Image identification makes use of techniques like detection of an object, its recognition, and segmentation.
The use of artificial intelligence and machine learning enhances the rate of processing the data and maintaining the standard of the outcome. Example, by using artificial intelligence, we can very easily complete difficult tasks.

The more conventional computer vision and ML approach as opposed to deep learning, the following have been useful:

- determining the most appropriate features (HAAR features, image color histogram, etc.)
- inspecting and learning about various techniques provided by OpenCV
- applying grid search to search the suitable classifier

Also, the following problems could arise:

- To decide correct position of our sliding windows
- To make sure that they don't overlap
- Recognizing the *threshold* for overlapping detection
- Using appropriate frame sampling rate
- Keeping min detection count over multiple frames
- Collecting combined window dimensions for avoiding overlapping detections

IV. Comparative Study

A. INTRODUCTION:

To acknowledge individuals in a real video surveillance system, we spot humans automatically. Pedestrian detection detects areas where a human is present.
Person detection is the process of predicting and localizing each person in the image, represented in a rectangular box that is made around the image.

B. METHOD OF PEDESTRIAN TRACING:

It mainly has 2 models:

1. Hand-crafted model which is based on hand-crafted features and is widely used for object detection.
2. Deep learning model uses convolutional neural networks to focus on improved performance.

C. PROBLEM FORMULATION:

Safety standards and accident prevention system in cars have gained notable development. **Advanced driver assistance systems** are commercially available now and mostly found pre-installed by car manufactures. The aim of these systems is to provide automobiles with sensors capable of detecting and acting when threatening situations are faced, so that the driver would be able to avoid a collision. **Detecting pedestrians,** launches an alert and then acts upon their unpredictable behavior.

Thus, the full potential of ADAS systems can be achieved by including AI car and **pedestrian tracing. To** utilize such strong ML with computer vision methods, we employ a **cascade of classifiers**. Each classifier is trained to model a pedestrian. The targets are detected Haar features. This process is done to be able to operate in real-time and will help in future, especially in self-driving cars.

MERITS:

While driving if you take your eyes off of the road to reach for your coffee turn to tell your kids to quiet down, and then suddenly you see a pedestrian is crossing the road right in front of you. You push the brakes—but it might be late.

It is a very scary yet usual scenario. One out of three crashes involve a vehicle going straight as a pedestrian crosses the road. To prevent these mis-happenings, automakers now provide a "pedestrian detection" system in some of their models. The ability to reliably detect vehicles provides huge advantage to everyone.

Haar Cascade Classifiers:

Object Detection using Haar cascade classifiers is a useful detection method. It is a ML method in which a cascade function is taught by providing a lot of true and false images.

For example, when we discuss about face identification. Initially, the algorithm needs lots of true (pictures of appearances) and false (pictures without faces) to train the classifier.

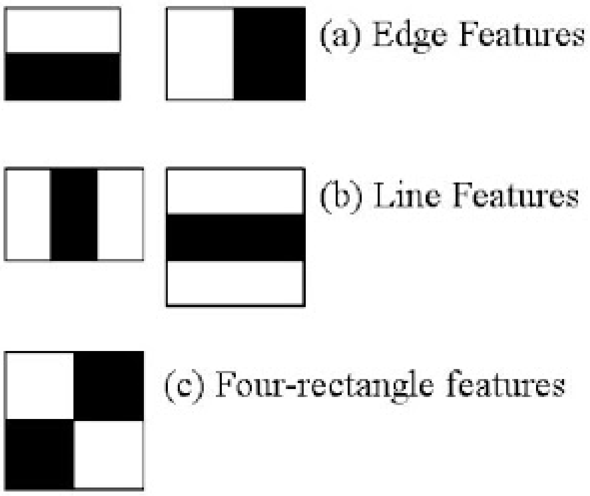

Figure 2: HAAR-CASCADE CLASSIFIER

Presently, all potential dimensions and areas of each kernel is used to determine the features.

For this, we apply every single component on all the training pictures. For each feature, it finds the best limit which will characterize the appearances to positive and negative. There will be mistakes yet we select the highlights with least blunder rate, which implies they are the highlights that most precisely characterize the face and non-face pictures.

V. Car And Pedestrian Tracker:

Intelligent vehicles have the ability to recognize cars and pedestrians and thus, detect dangerous situations ahead of time. We have shown how to use ai and ml algorithms and make a project out of it for beginners using OpenCV and python. An illustration for beginners.

BASIC TOOLS REQUIRED:

- Visual Studio Code
- Open CV
- Python
- Haar Cascade classifier
- XML

OBJECT IDENTIFICATION:

It is defined as procedure:
- Using the primary group of object detection
- Creating a distinctive identity for every primary detection
- Subsequently, tracing all objects when they shift throughout the frames in clip, while preserving the distinctive identities.

Moreover, tracking permits to use distinctive identity all tracked object, hence, attainable since the unique entities in a clip can be accounted for. A good algorithm is fast, has the capacity to manage the object when being tracked and creates a rectangular outline. It also, detects the primary object in one go. It also finds objects that it might have been lost in the middle of the frames

The tracking algorithm depends upon Euclidean distance that exists in the middle of objects which the tracker has been trained on before and new objects between consecutive frames in a video.

The tracking algorithm presumes that when a rectangular outline is made along the entity, it recognises it as an object. These rectangular boxes are built using detectors for instance, solid state drives, haar cascade classifier, etc, using which they identify and run each frame in a video.

VI. Implementation and Description of Project Modules:

We start by giving an input image and video, which then gets converted into black and white, so that processing can be done faster since there are many screens within fraction of seconds. After this, the black and white image is converted to colored when identified as the object and a rectangular box is created around the object.
The same process is done over and over until the last frame of the video or photo is achived.

Module 1(Figure 3,4):

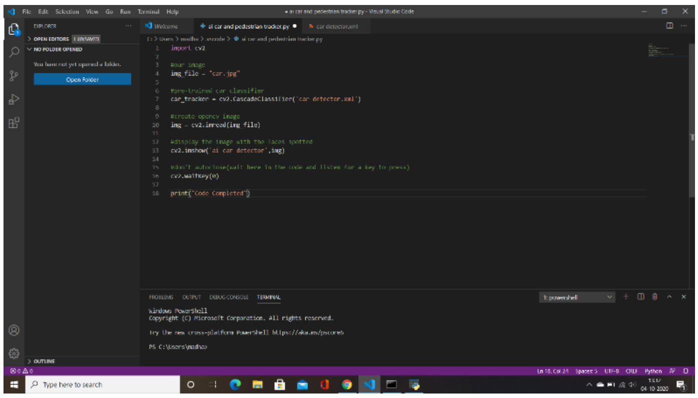

Figure 3: Colored image processing

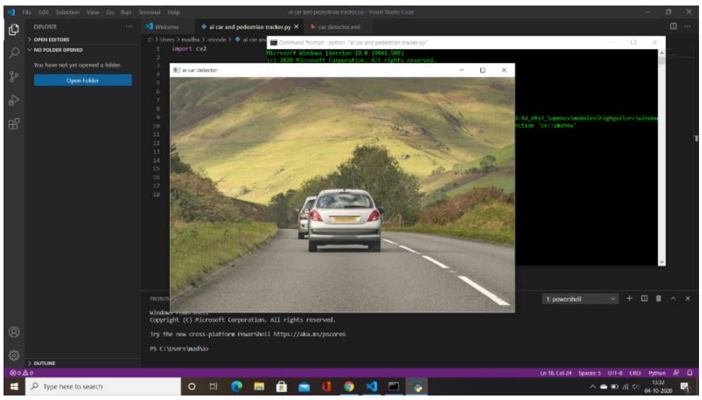

Figure 4: Colored image output

Module 2(Figure 5,6):

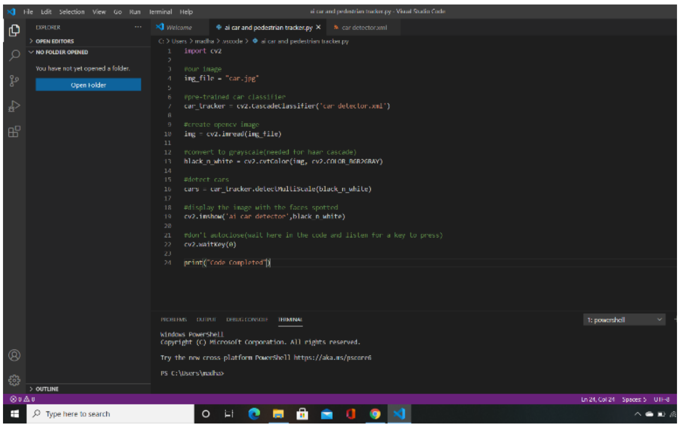

Figure 5: Black & White image processing

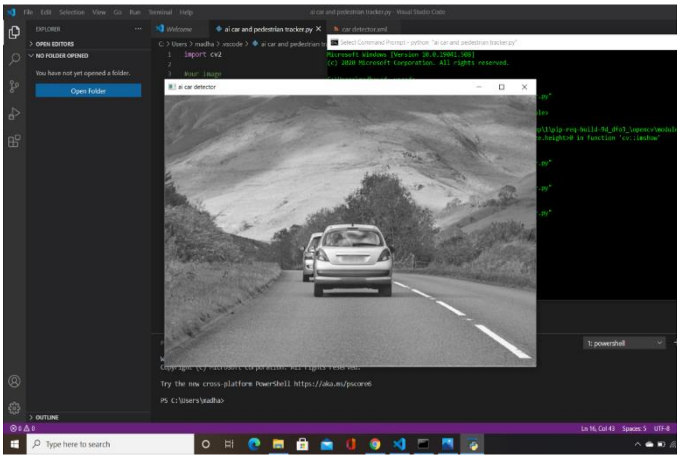

Figure 6: Black & White image output

Module 3(Figure 7,8):

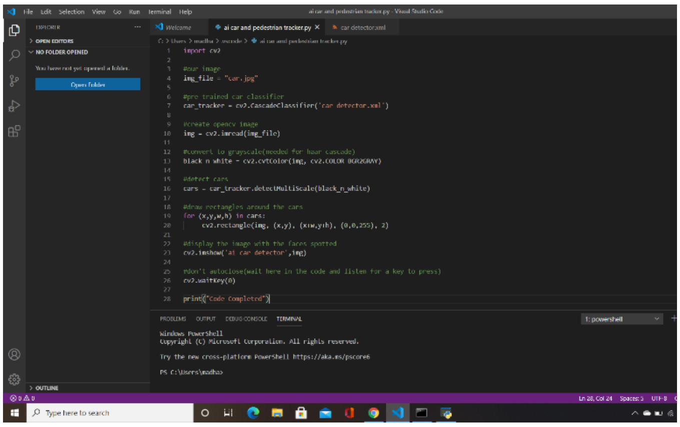

Figure 7: Making rectangles(outline)

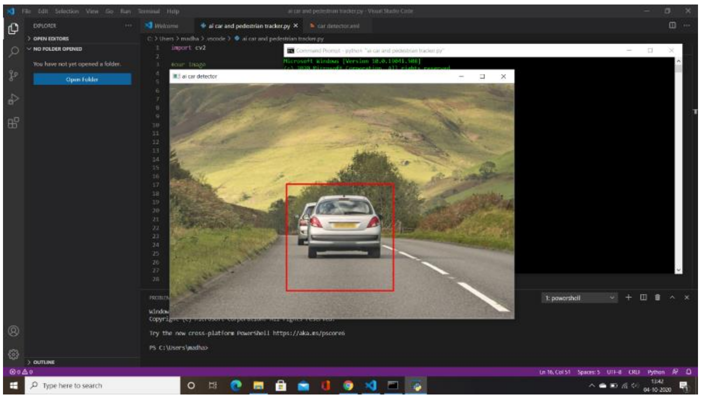

Figure 8: Output of outline around the object

For Example (Code with implementation):

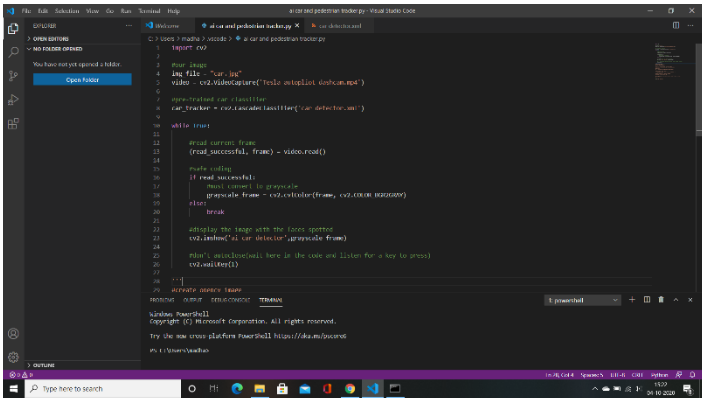

Figure 9: Final Code Snippet

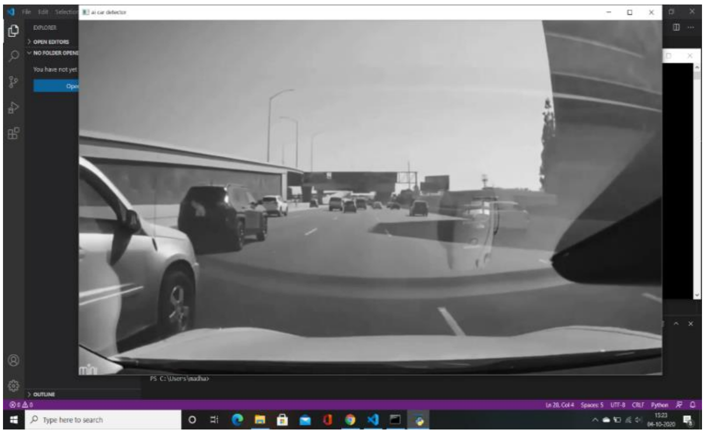

Figure 10(Black & white o/p)

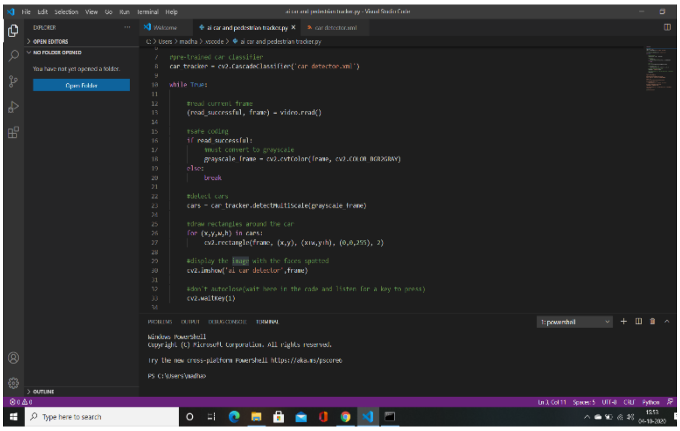

Figure 11: Code Snippet

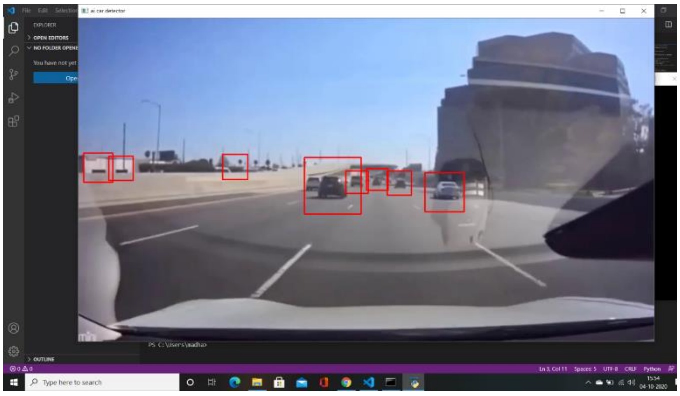

Figure 12: Running output

VII. Architecture Diagram for Proposed Method:

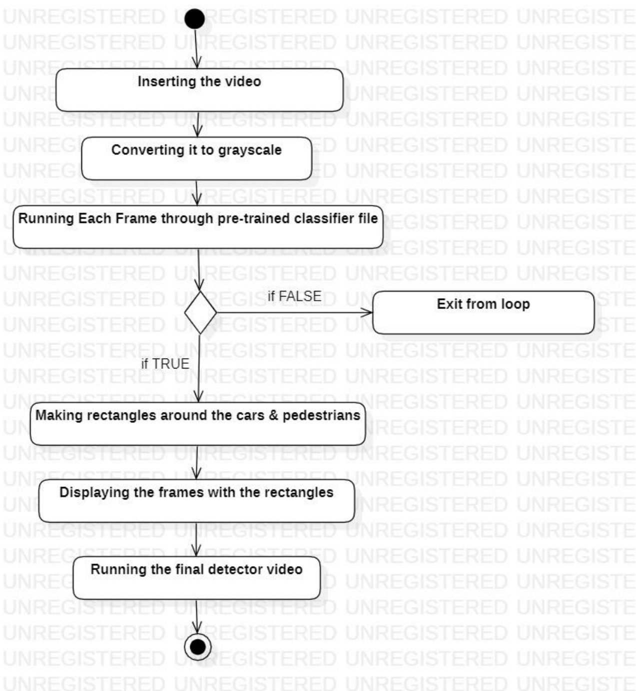

Figure 13. Flow Chart

VIII. Conclusion:

So, artificial intelligence effects the future of each and every single industry and humans as a whole. It also acts as the main operator of emerging technologies such as big data, robotics and IoT. Using machine learning and computer vision for detection and classification of different activities is very important. Like observing which direction, the driver is looking when he operates the vehicle, how fast he's driving, which direction he's driving, locating the people that surround him, etc. Therefore, main goal is to prevent accidents by increasing efficiency.

References:

[1]. [1] Viola, P & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR, 2001), December 8-14, 2001, Kauai, HI, USA.

[2]. [2] Liao, S., Jain, A.K., Li, S. Z. (2016). A fast and accurate unconstrained face detector. IEEE Transaction of Pattern Analysis and Machine Intelligence, Vol 38, No 2.

[3]. [3] Luo, D., Wen, G., Li, D., Hu, Y., and Huna, E. (2018). Deep learning-based face detection using iterative bounding-box regression. Multimedia Tools Applications.

[4]. [4] Zhang, Y., Wang, X., and Qu, B. (2012). Three-frame difference algorithm research based on mathematical morphology. Proceedings of 2012 International Workshop on Information and Electronics Engineering (IWIEE), pp. 2705 – 2709.

[5]. [5] Canny, J. (1986). A computational approach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume: PAMI-8, No: 6, pp. 679-698, November 1986.

[6]. [6] Li, J. and Ding, S. (2011). A research on improved Canny edge detection algorithm. Proceedings of the International Conference on Applied Informatics and Communication, pp. 102 – 108, Communications in Computer and Information Science (CCIS), Vol 228, Springer-Verlag.

[7]. [7] Lucas, B. D. & Kanade, T. (1981). An iterative image registration technique with an application to stereo vision.

[8]. [8] Ren, Z., Yang, S., Zou, F., Yang, F., Luan, C., and Li, K. (2017). A face tracking framework based on convolutional neural networks and Kalman filter. Proceedings of the 8th IEEE International Conference on Software Engineering and Services Science, pp. 410-413.

[9]. [9] Mingxing, J., Junqiang, D., Tao, C., Ning, Y., Yi, J., and Zhen, Z. (2013). An improved detection algorithm of face with combining AdaBoost and SVM. Proceedings of the 25th Chinese Control and Decision Conference, pp. 2459-2463.

[10]. [10] Altun, H., Sinekli, R., Tekbas, U., Karakaya, F. and Peker, M. (2011). An efficient color detection in RGB space using hierarchical neural network structure. Proceedings of 2011 International Symposium on Innovations in Intelligent Systems and Applications, pp. 154-158, Istanbul, Turkey.