Blinking to a New Era of Environmental Interaction
Project-Bitrac

ABSTRACT

This paper introduces an innovative solution for a person to interact with his surroundings with the power of blink. We utilize various modern algorithms and models like Inception v3, a pre-trained Deep Neural Network model for image classification [5], and facial landmark detection algorithm of Kazemi and Sullivan (2014) trained using various datasets like the IBUG dataset to efficiently capture an activation signal in the form of a blink and thereby capture the image of the object to be interacted with. This is achieved using two micro cameras, one on a watch and one hooked to the user. The image is then processed, and the appliance is thereby switched on with an IR signal. A disabled person can deeply interact with all the appliances with the help of a few blink patterns.

***Keywords*—**Artificial Intelligence, ImageNet Dataset, OpenCV

I. INTRODUCTION

Blinking to interact with one's surroundings is a highly powerful tool especially for disabled people. People who are handicapped, bedridden or affected by strokes are usually capable of moving their heads and registering a blink, it is one of the most basic capabilities that is usually the first to be gained and the last to be lost by a human being struck by illness. This project captivates that very capability to allow these people to interact with their surroundings as well as to become independent, helping them to blend into the society and to an extent overcome their sense of loss.

We have created a project to help people interact with their surroundings just with the power of blink. This technology can also ease the daily struggles of many commoners. How many times has one wished to turn off the lights while on the bed or to switch on the AC while sleeping? Forgetting where one has placed the remote is also a commonplace struggle. The capability to do all of the above with the ability to blink is an innovative, efficient, and extremely cheap method of achieving the objective. This project can be made adaptive and/or modified to suit many industrial requirements. It is an extremely useful tool for astronauts in zero gravity situations as blinking to overcome many of their daily life struggles reduces commuting. Our device can also act as an assistant in many experimental setups where there might be a need to change various experimental factors while still doing a specific task.

Many methods have been used to detect the motion of the eye lids and therefore detect a blink. Some of them include frame by frame intensity tracking and thereby using a threshold to estimate if a blink has taken place. Others use a single frame, by performing pixel wise or intensity based comparisons on the extracted face and estimated eye region with respect to images of closed and open eyes. A parametric model fitting to find eye lids or active shapes are also used.

These methods are effective to a certain extend but they are sensitive to environmental factors like lighting and also require many equipment's or have slow processing speeds. Recent innovations in facial landmark detection using regression trees and optimizing square loss error has provided a quick, efficient, and innovative approach around the problem. This method is accurate enough for the project as it is trained on "in the wild datasets". This trained model for facial landmark detection when combined with advanced image processing and recognition models like Inception v3 provides new possibilities. Our project capitalizes on these innovative models by using efficient programming methodology and algorithms. This helps make environmental interaction with the help of blink a reality.

This paper covers: -

• Frame wise extraction of eye coordinates from the facial landmark detector.

• Utilization of the extracted coordinates to detect whether the appropriate blink pattern for appliance activation has taken place.

• Capturing an image based on the blink signal and thereby processing it to identify the required appliance to be activated or deactivated.

II. PROPOSED METHOD

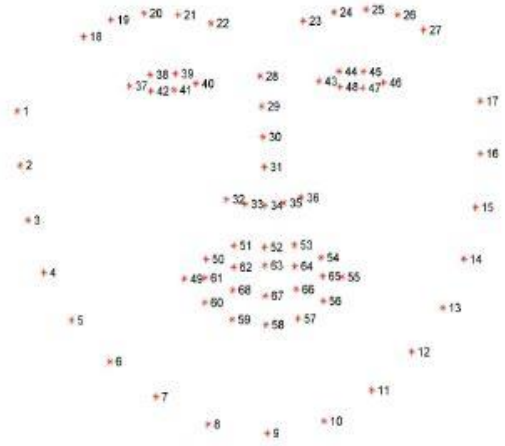


Figure 1: The 68 points mark-up [3].

From dlib, which is an open source toolkit for machine learning algorithms and trained models, we used the algorithm based on 'one-millisecond face alignment with an ensemble of regression trees' by Sullivan and Kazemi (2014) [8]. A model based on the algorithm that has been trained on the IBUG dataset is used in our project. The algorithm uses gradient boosting to learn an ensemble of regression trees that optimizes the loss (error).

The algorithm, by using prior measures to manipulate the image data structure, helps in feature selection.

The algorithm based on pixel intensities provides a quick (within a processing time frame of about a millisecond) set of facial landmarks which in the case of the model trained on IBUG dataset is a set of 68 x, y-coordinates (Fig. 1) of the facial landmarks per face in a single image. The set is further divided into specific subsets of coordinates assigned to specific facial features like the eye, mouth, eyebrows etc. These subsets of coordinates can be accessed separately with specific commands. This program when run with frames from a video stream, provides a quick and cheap means of real-time feature tracking that is unaffected by the surroundings i.e. lightings, pixel intensities, and various other environmental factors. This is much more efficient than the conventional method of the eye and face isolation with the help of pixel intensities. This is because the conventional methods are much more time consuming and are much more sensitive to environmental factors as many work within range functions or use time consuming algorithms. For our project we are only interested on the subset of coordinates that deal with the left and the right eye. The model provides us with 6 x, y-coordinates for each of the eyes per frame loaded from the real-time video stream. template. Use this document as an instruction set.

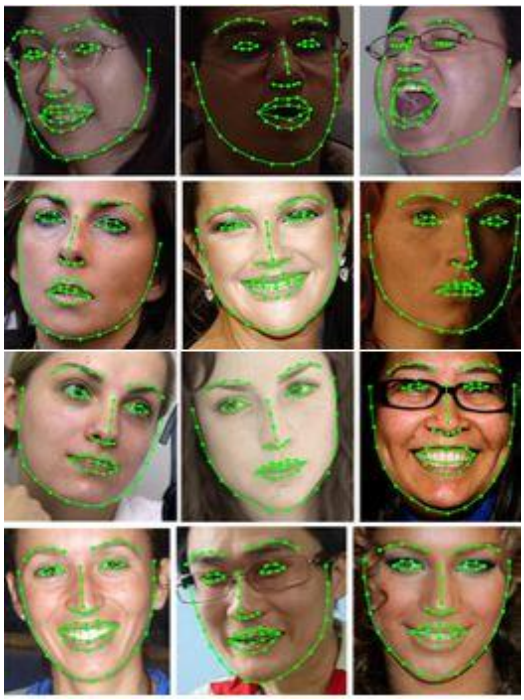


Figure 2: Selected results on the IBUG dataset. An ensemble of randomized regression trees is used to detect 194 landmarks on face from a single image in a millisecond [8].

Blink Detection

This project works on two cameras (micro camera(s) or a mini-spy camera(s)), which are mounted on the eye wear or on the users outfit and on a suitable personal wear like a watch or a ring. This camera acts as the source of the real-time video stream that is fed into the program. In case the camera is attached to a hand wear of the user. The user may activate the device by facing the camera and closing his eyes for 2-3 seconds. After activation the user must blink twice so that the camera can be triggered to capture the image of the object the user wants to control.

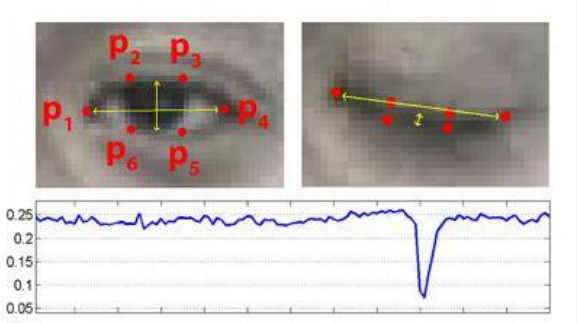


Figure 3: Top-left: A visualization of eye landmarks when then the eye is open.
Top-right: Eye landmarks when the eye is closed.
Bottom: Plotting the eye aspect ratio overtime [6].

For the actual detection of the blink the program extracts the 6 x, y-coordinates (Fig. 3). of the eyes from each frame and calculates the distance C between the eyelids, Based on the formula: -

$$C = \frac{||p_2 - p_6|| + ||p_3 - p_5||}{2}$$

This formula calculates the coordinate difference of a set of symmetrically located points on the upper and the lower eye lids and then averages the similarly calculated values for various points on the eye lids to find the mean distance between the eye lids for each eye. Then an experimentally calculated lower limit is set for C. If the value of C falls below this value a blink is said to have taken place. A counter keeps track of the number of blinks and the frame rate of the video stream acts as a timer to calculate the time between two consecutive blinks and also calculate the time the value of C has been below the threshold value during the course of a blink. If the value of C falls below the threshold range for a certain number of frames (i.e. number of frames that is equivalent to 2 seconds) then an image is captured. This image is then classified using Inception as shown in Figure. Alternate suitable methods can be employed for activation like alternate blinks from the left and the right eyes. These processes help distinguish from the regular activities of the human eye and therefore act as a distinct source of device activation and prevents any error or unwanted appliance activation.

T.V. Control	Blink Pattern
T.V. ON/OFF	Blink Twice
Volume UP	Left-Eye Wink
Volume DOWN	Right-Eye Wink

Table 1: Example in which a handicapped patient wishes to deeply interact with his surroundings.

Image Processing

We have used the Inception v3 model which has nearly 25 million parameters and uses 5 billion multiply-add operations for classifying a single image [5]. When an image is sent for classification to this model, the model classifies the image and gives a wide range of probabilities. Inception also takes care to focus on and detect the primary object under consideration in the provided photo. For example, I fan image of a T.V. is sent for classification into the trained model, the T.V. will then be generally classified as ['television, television system', 'home theater', 'entertainment center'] by Inception. These probabilities will be much greater in the event of a T.V. than any other object and hence we collected a large sample of images under different lighting conditions, from different angles etc. and classified them using Inception. Then we determined all the possible appliances that get detected with high probabilities in case of a T.V. After this we used the formula: -

$$\sum x_i y_j$$

Where Xi denotes the weight assigned to the object on the basis its frequency of appearance and probability in the above tested sample under different conditions and Xj denotes the obtained probability of a common object after classification by Inception. Here the objects that are defined as common (the ones that take part in the equation) are the ones that have quite often shown a probability of 0.3 or above in the sample. Then we obtain two values one is of the detected object with the highest probability and the second is the weighted average of the other common objects as per the above formula. If the calculated average and the highest probability meet certain requirements we can ascertain that the detected object is in fact a television. This algorithm is highly effective as there are few appliances to be detected in comparison to the number of appliances Inception has been trained to detect. This will ensure an error free method of object detection in case of changing lighting and low video camera quality. The cameras mounted onto the glasses (will behave as 360-degree camera) can take care of both the blink detection and object detection and therefore acting like a single video stream source. If the camera used to detect the blink is placed on a hand wear like a ring or a watch then a second camera must be conveniently located on the face or has to be pinned to the shirt pocket or shoulders for image processing and object detection. Some users may find the first method of camera placement more user friendly than the second and some may find the second method more user friendly than the first.

Examples of Images Classified Using Inception v3

When the model runs, the script will produce the following output [4] : -



Figure 4: Photo taken from [7]

Output: -
television, television system 0.908759
screen, CRT screen 0.0482494
monitor 0.0208261
web site, website, internet site, site 0.00312534
entertainment center 0.00225773

Object Activation

Now, most of the task is completed we have detected an appropriate signal in the form of a blink and we have detected the object to be controlled be it a television set, a set of bulbs or the air conditioning system. Now, all that is left is to send the appropriate signal and perform the required activity. For this, different ways are employed in accordance with the user's available resources. If the user has a smart home, then the "smart appliances" are linked with the program and based on the detected object different commands are passed on to the system and the detected object is controlled. If no "smart appliances" are present, then an alternative means of IR transmission can be employed (even a conveniently placed mobile phone next to the T.V. can perform this task). For example, to switch on the television we have used the "Desktop TV Remote" app which allows the user to control any Smartphone's IR Blaster via a PC [2].

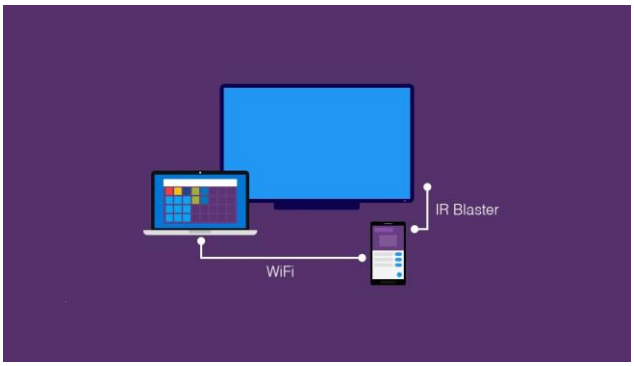


Figure 5: Device linkage using Desktop T.V. Remote [2].

The app can also help control a wide range of appliances like the LED lights, air conditioning systems, heaters, projectors etc. Once registered the mobile is automatically detected when in the vicinity and thereby no further activation is required. Moreover, the app runs in the background of the system and therefore acts as a reliable medium. The app sends a signal via the Wi-Fi system to the mobile from which the required IR frequency is blasted to turn on the device. The app contains the information regarding the official brand IR frequencies for various operations of a device. This information is available for a wide range of brands like Samsung, Acer, Toshiba etc. and therefore acts as a reliable source regardless of the brand of the appliance.

Advancing the Project

We have explained how our project uses the model trained on the **IBUG** dataset for blink detection. Even though the **IBUG** dataset is reliable enough in low light conditions for blink detection, this can be replaced by the model trained on the **HELEN** dataset which is much more precise as compared to the **IBUG** dataset and we are currently working to incorporate this into our project and take it to the next level. Fig. 6 shows the comparison of **IBUG** and **HELEN** datasets.

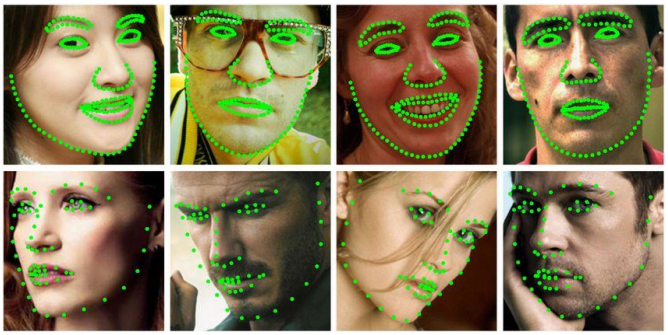


Figure 6: Row 1: HELEN Dataset (194 landmarks)
Row 2: IBUG Dataset 68 landmarks [9]

We are also working towards incorporating Finger Gesture based interaction into this project. For this, we are using OpenCV for detecting the finger and also for shape detection of the figure drawn. For example, A rectangle drawn would mean to increase or decrease volume in case of a Television.

Working of the Gesture Program

The figure shown below (Fig. 7), is the output made using an index finger in the OpenCV program [1].

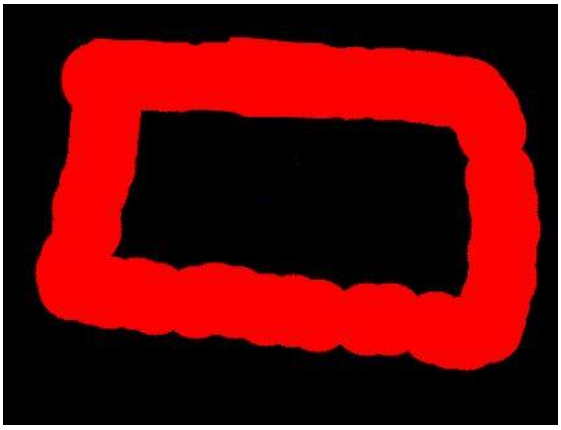


Figure 7: A rectangular shape drawn using OpenCV

After this, the image shown above (Fig. 7) is sent into the shape detection program and the script on running with this image shows the following output: -
Rectangle
This can then be trained to perform specific operations.

V. CONCLUSION

First and foremost, we would like to thank **Kazemi and Sullivan** for providing an efficient and quick facial landmark detection system both of which has played a positive role in making this project a reality. We are currently trying to efficiently incorporate the system into wheelchairs, hospital cots and electric beds in our neighborhood and nearby hospitals so as to clearly test their efficiency in helping people with a wide variety of needs. We have seen that slight modifications help in incorporating the model into a variety of environments in a feasible and user-friendly way. We have also included models on hand gesture recognition which complements blink activation and helps make the model much more user friendly for those who are capable of utilizing these aspects. Moreover, by training models that capture specific appliances which correspond to specific users, we see that we are able to cut down the processing time to a large extend, making the appliance extremely fast and much more reliable. We hope that with further improvement and effort we would be able to transform this idea to the next stage and truly make a positive difference in the community.

REFERENCES

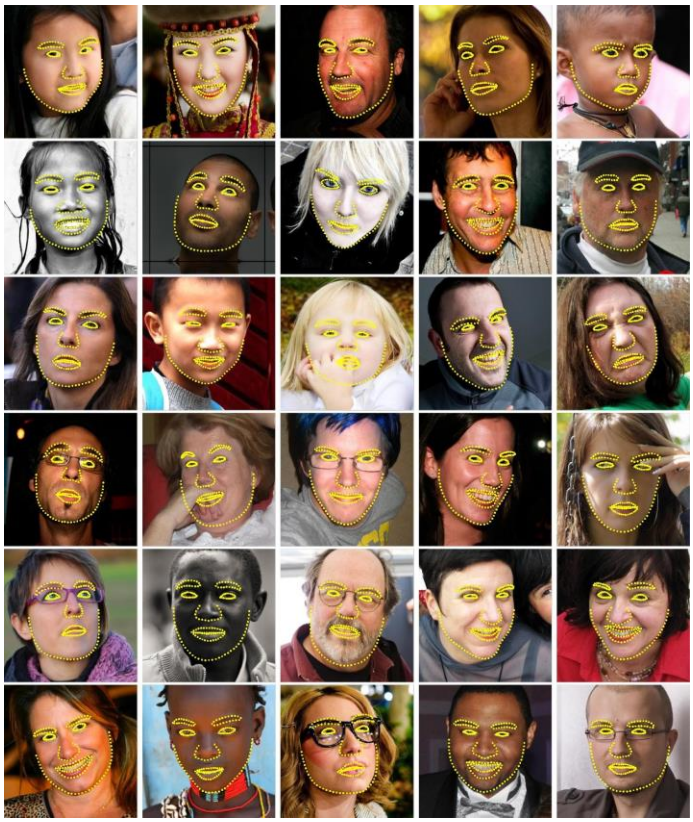


Figure 8: Final results on the HELEN database [8].

[1] Adrian Rosebrock. Opencv shape detection. https://www.pyimagesearch.com/2016/02/08/opencv-shape-detection/, 2016

[2] Desktop tv remote. https://play.google.com/store/apps/detailsid=tomaszczart.com.desktoptvremote&hl=en.

[3] Facial point annotations. https://ibug.doc.ic.ac.uk/resources/facial-point-annotations/

[4] Image recognition. https://www.tensorflow.org/tutorials/image_recognition

[5] Tensorflow tutorial - inception model. http://www.datascienceassn.org/content/tensorflow-tutorial-inception-model

[6] Tereza Soukupova and Jan Cech. Real-time eye blink detection using facial landmarks

[7] Ultra hd oled tv, lg. https://www.euronics.ee/t-en/81211/tvs-and-home-cinema-systems/ 65-ultra-hd-oled-tv-lg/oled65w7v-aee

[8] Vahid Kazemi and Josephine Sullivan. One millisecond face alignment with an ensemble of regression trees.

[9] Yiru Zhao Qinchuan Zhang Lizhuang Ma Zhiwen Shao, Shouhong Ding. Learning deep representation from coarse to fine for face alignment. https://www.researchgate.net/publication/305388996_Learning_Deep_Representation_from_Coarse_to_Fine_for_Face_Alignment.