IJARSCT

ISSN (Online) 2581-9429

IJARSCT
Impact Factor: 6.252

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

Volume 2, Issue 2, April 2022

Driver Drowsiness Detection System Using Convolutional Neural Networks

M. Kusuma Sri and T. Annamani
Assistant Professor
Anurag University, Hyderabad, TS, India
kusumasriece@cvsr.ac.in and annamaniece@cvsr.ac.in

Abstract: *In recent years, the rise of car accident fatalities has grown significantly around the world. Hence, road security has become a global concern and a challenging problem that needs to be solved. The deaths caused by road accidents are still increasing and currently viewed as a significant general medical issue. The most recent developments have made in advancing knowledge and scientific capacities of vehicles, enabling them to see and examine street situations to counteract mishaps and secure travelers. Therefore, the analysis of driver's behaviors on the road has become one of the leading research subjects in recent years, particularly drowsiness, as it grants the most elevated factor of mishaps and is the primary source of death on roads. This project presents a way to analyze and anticipate driver drowsiness by applying a Convolutionl Neural Network over a sequence frame driver's face. We used a dataset to shape and approve our model and implemented repetitive neural network architecture multi-layer model-based 3D Convolutional Networks to detect driver drowsiness. After a training session, we obtained a promising accuracy that approaches a 92% acceptance rate, which made it possible to develop a real-time driver monitoring system to reduce road accidents.*

Keywords: Video, CNN, Open-CV, Haar Classifier

I. Introduction

In today's world, every human being uses a vehicle. It is often considered as luxury but it has now become a necessity in a common man's life. People are very much concerned about their safety and also the vehicles safety is it in case of theft or an accident. Motor vehicles are utilized for many purposes like for transporting general public, things and also private journeys. During a very long driving hours the driver will normally get tired and the person feeling drowsy but he still went on driving for reaching the endpoint as soon as possible. The driving in non-favorable/safe condition due to which driver tends to exert himself. When the person feels sleepy he/she still wish to go on driving in spite of the fact that it is very risky. He or she falls asleep and the vehicle is no more in control and collides with other vehicles on road leading to loss of many lives.

There are numerous non-driving related causes to results to accidents including road conditions, the weather and mechanical performance of a vehicle. However, most of the accidents are happened due to driver error. Driver error includes drunkenness, fatigue, and drowsiness. Many factors can affect a driver's ability to control a motor vehicle, such as natural reflexes, recognition and perception. Hence, yearly we are losing thousands of persons in road accidents, cause and important factor for this is drowsiness of the driver.

To improve the driving quality, cars manufactured with smart features like controlling engine speed, transmission, steering and applying break etc. are taken care by advanced software systems. Ad-hoc networks were the first to develop the automatic navigation systems in cars. A noticeable weakness of these systems is that they don't respond in real time to the environmental changes. If prior consideration is time, decision is left with driver. One more technique of checking drowsiness of driver is, facial expressions or physical condition. Hence it is considered as a strong motivation to invent efficient and effective "System to detect driver drowsiness". The former technique, is not reliable. Highly sensitive electrodes would have to be attached on body of the driver, which discomforts driver. Also accuracy monitoring was tedious task. In the second technique, changes in physical actions or moments like i.e. open/closed eyes to detect fatigue measuring is non-intrusive. Also short period of sleeps are good indicators of fatigue. So, it is possible to warn driver in time by monitoring eyes continuously through detecting eye state. Driver assistance system development have been required to

DOI: 10.48175/IJARSCT-3399

prevent the accidents due to driver drowsiness, because all the time he cannot control the vehicles some risks may happened due to driver's tiredness, or inattention. This system helps to keep driver attentive.

1.1 Face Detection

For the face Detection it uses Haar feature-based cascade classifiers is an effective object detection method. It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images. Here we will work with face detection. Initially, the algorithm needs a lot of positive images and negative images (images without faces) to train the classifier. Then we need to extract features from it. For this, Haar features shown in the below image are used. They are just like our convolutional kernel. Each feature is a single value obtained by subtracting sum of pixels under the white rectangle from sum of pixels under the black rectangle.

1.2 Eye detection

In the system we have used facial landmark prediction for eye detection Facial landmarks are used to localize and represent salient regions of the face, such as: Eyes, Eyebrows, Nose, Mouth, Jawline. Facial landmarks have been successfully applied to face alignment, head pose estimation, face swapping, blink detection and much more. In the context of facial landmarks, our goal is detecting important facial structures on the face using shape prediction methods. Detecting facial landmarks is therefore a twostep process: Localize the face in the image: The face image is localized by Haar feature-based cascade classifiers, then detect the key facial structures on the face ROI: There are a variety of facial landmark detectors, but all methods essentially try to localize and label the following facial regions: Mouth, Right eyebrow, Left eyebrow, Right eye, Left eye, Nose.

1.3 Recognition of Eye's State:

The eye area can be estimated from optical flow, by sparse tracking or by frame-to-frame intensity differencing and adaptive thresholding and finally, a decision is made whether the eyes are or are not covered by eyelids. A different approach is to infer the state of the eye opening from a single image, as e.g. by correlation matching with open and closed eye templates, a heuristic horizontal or vertical image intensity projection over the eye region, a parametric model fitting to find the eyelids, or active shape models. A major drawback of the previous approaches is that they usually implicitly impose too strong requirements on the setup, in the sense of a relative face-camera pose (head orientation), image resolution, illumination, motion dynamics, etc. Especially the heuristic methods that use raw image intensity are likely to be very sensitive despite their real-time performance. Eye Aspect Ratio Calculation for every video frame, the eye landmarks are detected. The eye aspect ratio (EAR) between height and width of the eye is computed.

$$EAR = \frac{\| p2 - p6 \| + \| p3 - p5 \|}{2 \| p1 - p4 \|}$$

Where p1, . . ., p6 are the 2D landmark locations.

The EAR is mostly constant when an eye is open and is getting close to zero while closing an eye. It is partially person and head pose insensitive. Aspect ratio of the open eye has a small variance among individuals, and it is fully invariant to a uniform scaling of the image and in-plane rotation of the face. Since eye blinking is performed by both eyes synchronously, the EAR of both eyes is averaged.

Eye State Determination: Finally, the decision for the eye state is made based on EAR calculated in the previous step. If the distance is zero or is close to zero, the eye state is classified as “closed” otherwise the eye state is identified as “open”. Drowsiness Detection: The last step of the algorithm is to determine the person’s condition based on a pre-set condition for drowsiness. The average blink duration of a person is 100-400 milliseconds (i.e. 0.1-0.4 ofa second). Hence if a person is drowsy his eye closure must be beyond this interval. We set a time frame of 5 seconds. If the eyes remain closed for five or more seconds, drowsiness is detected and alert pop regarding this is triggered.

1.4 Open-CV

The most effortless approach to identify and fragment an item from a picture is the shading based techniques. The item and the foundation ought to have a critical shading distinction so as to effectively portion objects utilizing shading based strategies.1 OpenCV usually captures images and videos in 8-bit, unsigned integer, BGR format. In other words, captured

DOI: 10.48175/IJARSCT-3399

images can be considered as 3 matrices; BLUE, GREEN and RED (hence the name BGR) with integer values ranging from 0 to 255.

II. Literature Survey

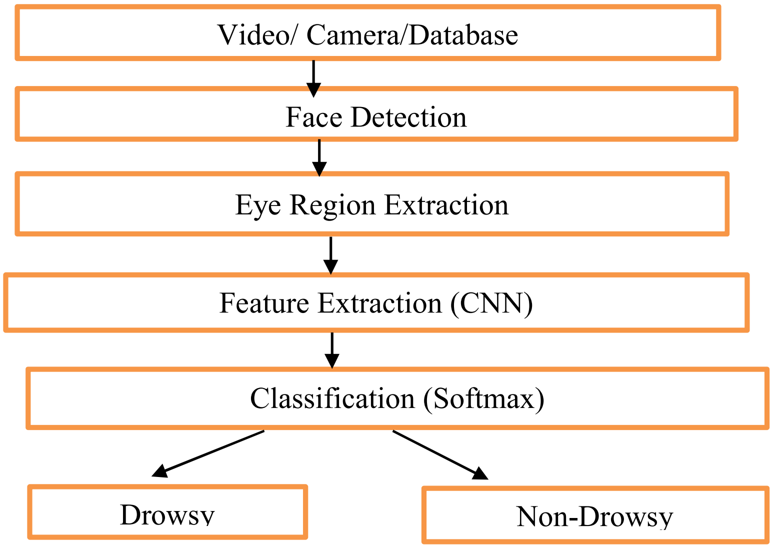
In this section, we have discussed various methodologies that have been proposed by researchers for drowsiness detection and blink detection during the recent years. Manu B.N in 2016, has proposed a method that detect the face using Haar feature-based cascade classifiers. Initially, the algorithm needs a lot of positive images (images of faces) and negative images (images without faces) to train the classifier that will detect the object. So along with the Haar feature-based classifiers, cascaded Adaboost classifier is exploited to recognize the face region then the compensated image is segmented into numbers of rectangle areas, atany position and scale within the original image. Due to the difference of facial feature, Haar-like feature is efficient for real-time face detection. These can be calculated according to the difference of sum of pixel values within rectangle area and during the process the Adaboost algorithm will allow all the face samples and it will discard the non-face samples of images.

Amna Rahman in 2015, has proposed a method to detect the drowsiness by using Eye state detection with Eye blinking strategy. In this method first, the image is converted to gray scale and the corners are detected using Harris corner detection algorithm which will detect the corner at both side and at down curve of eye lid. After tracing the points then it will make a straight line between the upper two points and locates the mid-point by calculation of the line, and it connects the mid-point with the lower point. Now for each image it will perform the same procedure and it calculates the distance ‘d’ from the mid-point to the lower point to determine the eye state. Finally, the decision for the eye state is made based on distance ’d’ calculated. If the distance is zero or is close to zero, the eye state is classified as “closed” otherwise the eye state is identified as “open”.

III. Proposed Method

The different types of methodologies have been developed to find out drowsiness. Physiological level approach: This technique is an intrusive method where in electrodes are used to obtain pulse rate, heart rate and brain activity information. ECG is used to calculate the variations in heart rate and detect different conditions for drowsiness. The correlation between different signals such as ecg (electrocardiogram), EEG(electroencephalogram), and EMG (electromyogram) are made and then the output is generated whether the person is drowsy or not.

Behavioral based approach: In this technique eye blinking frequency, head pose, etc.of a person is monitored through a camera and the person is alerted if any of these drowsiness symptoms are detected.



Take Image as Input from a Camera, Detect Face in the Image and Create a Region of Interest ROI), Detect the eyes from ROI and feed it to the classifier, Classifier will Categorize whether Eyes are Open or Closed, Calculate Score to Check whether Person is Drowsy.

3.1 Haar Cascade Classifier

Haar Cascade Classifier is the classifier, is used to classify based on object feature of interest. Features of the images can be extracted using XML files of haar cascade classifier, use the extracted features for further process as given below. Image features extracted using Haar cascade classifier Detecting of face in fetched image and creating the region of interest (ROI): We don't need color information to detect the objects on the fetched image. For detecting the face region in the image, we need to convert the images into gray-scale. We will be using XML files of Haar cascade classifier for detecting face region. Eyes are detected from Region of Interest then feed it to the classifier: The same steps which are used in detection of face and eyes. First, we need to set our cascade classifier for both eyes with leye (Left Eye) and reye (Right Eye) respectively then we can eyes are detected using [3]. Now data of eyes are extracted from image. By using boundary box of the eye, this can be achieved and so, we can fetch image of an eye from the image. eye only contains data of the left eye. Then, give it as input to CNN classifier, predicted result will be given like eyes are in the state closed or open. Same way process repeat with the right eye data into r_eye. CNN will classify whether an eyes are open or closed [4]: CNN classifier is used to predict the status of eye. If drivers eyes are closed more than fixed threshold value, assuming that the person has drowsiness, by that time system will invoke sound.play() method to make beep sound and alerts the driver.

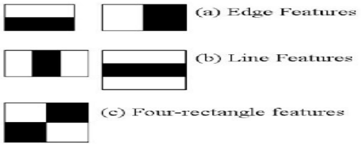


Fig.Haar cascade features

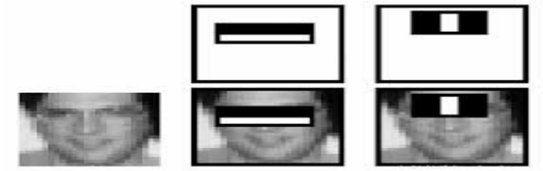


Fig. Eye detecting feature

A **Convolutional Neural Network (CNN)** is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics. The architecture of a ConvNet is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. A collection of such fields overlap to cover the entire visual area.

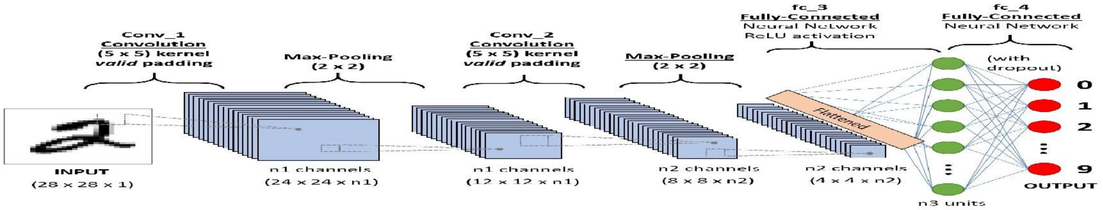


Figure: Layers of CNN

In cases of extremely basic binary images, the method might show an average precision score while performing prediction of classes but would have little to no accuracy when it comes to complex images having pixel dependencies through 'out.A ConvNet is able to **successfully capture the Spatial and Temporal dependencies** in an image through the application of relevant filters. The architecture performs a better fitting to the image dataset due to the reduction in the number of parameters involved and reusability of weights. In other words, the network can be trained to understand the sophistication of the image better. An RGB image which has been separated by its three color planes — Red, Green, and Blue. The role of the ConvNet is to reduce the images into a form which is easier to process, without losing features which are critical for getting a good prediction. This is important when we are to design an architecture which is not only good at learning features but also is scalable to massive datasets.

DOI: 10.48175/IJARSCT-3399

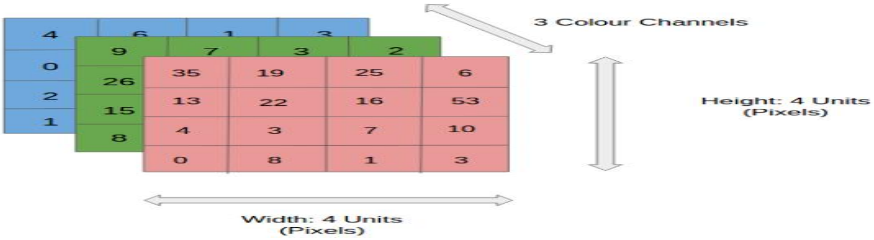


Figure: Convolutionl layers

Image Dimensions = 5 (Height) x 5 (Breadth) x 1 (Number of channels, eg. RGB). In the above demonstration, the green section resembles our 5x5x1 input image, I. The element involved in carrying out the convolution operation in the first part of a Convolutional Layer is called the Kernel/Filter, K, represented in the color yellow. We have selected K as a 3x3x1 matrix. The Kernel shifts 9 times because of Stride Length = 1 (Non-Strided), every time performing a matrix multiplication operation between K and the portion P of the image over which the kernel is hovering. The filter moves to the right with a certain Stride Value till it parses the complete width. Moving on, it hops down to the beginning (left) of the image with the same Stride Value and repeats the process until the entire image is traversed. In the case of images with multiple channels (e.g. RGB), the Kernel has the same depth as that of the input image. Matrix Multiplication is performed between Kn and In stack ([K1, I1]; [K2, I2]; [K3, I3]) and all the results are summed with the bias to give us a squashed one-depth channel Convoluted Feature Output.

There are two types of Pooling: Max Pooling and Average Pooling. **Max Pooling** returns the **maximum value** from the portion of the image covered by the Kernel. On the other hand, **Average Pooling** returns the **average of all the values** from the portion of the image covered by the Kernel. Max Pooling also performs as a **Noise Suppressant**. It discards the noisy activations altogether and also performs de-noising along with dimensionality reduction. On the other hand, Average Pooling simply performs dimensionality reduction as a noise suppressing mechanism. Hence, we can say that **Max Pooling performs a lot better than Average Pooling**. The Convolutional Layer and the Pooling Layer, together form the i-th layer of a Convolutional Neural Network. Depending on the complexities in the images, the number of such layers may be increased for capturing low-levels details even further, but at the cost of more computational power. After going through the above process, we have successfully enabled the model to understand the features. Moving on, we are going to flatten the final output and feed it to a regular Neural Network for classification purposes.

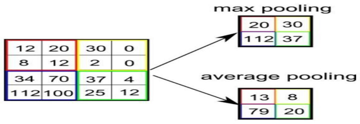
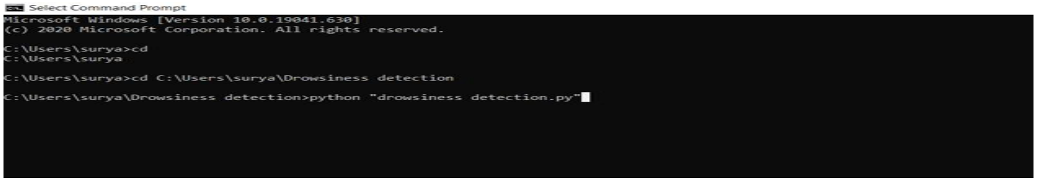


Figure: Pooling methods

Adding a Fully-Connected layer is a (usually) cheap way of learning non-linear combinations of the high-level features as represented by the output of the convolutional layer. The Fully-Connected layer is learning a possibly non-linear function in that space. Now that we have converted our input image into a suitable form for our Multi-Level Perceptron, we shall flatten the image into a column vector. The flattened output is fed to a feed-forward neural network and backpropagation applied to every iteration of training. Over a series of epochs, the model is able to distinguish between dominating and certain low-level features in images and classify them using the **Softmax Classification technique.**

IV. Result And Discussion

To start the system, we need to open a command prompt, go to the directory where our main file "drowsiness detection.py" exists. Run the script with this command.



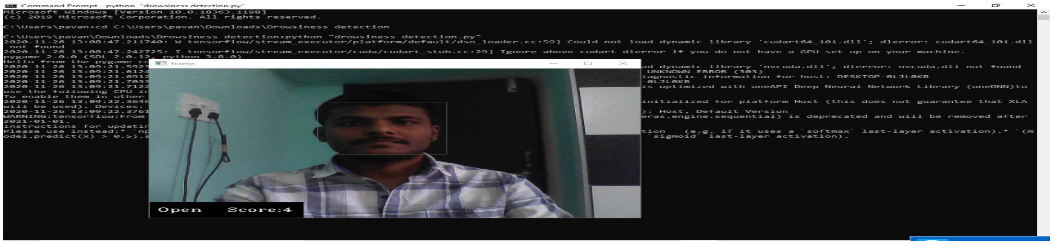
```
Microsoft Windows [Version 10.0.19041.630]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\surya>cd
C:\Users\surya

C:\Users\surya>cd C:\Users\surya\Drowsiness detection

C:\Users\surya\Drowsiness detection>python "drowsiness detection.py"
```

It may take a few seconds to open the webcam and start detection.

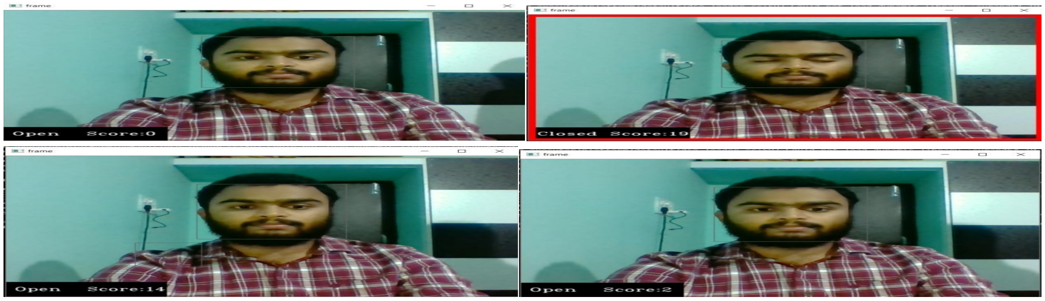


```
Command Prompt - python  "drowsiness detection.py"
Microsoft Windows [Version 10.0.18363.1198]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\pavan>cd C:\Users\pavan\Downloads\Drowsiness detection

C:\Users\pavan\Downloads\Drowsiness detection>python "drowsiness detection.py"
2020-11-26 13:08:47.211740: W tensorflow/stream_executor/platform/default/dso_loader.cc:59] Could not load dynamic library 'cudart64_101.dll'; dlerror: cudart64_101.dll
 not found
2020-11-26 13:08:47.242725: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if you do not have a GPU set up on your machine.
pygame 2.0.0 (SDL 2.0.12, python 3.8.0)
Hello from the pygame co
2020-11-26 13:09:21.5923                              ad dynamic library 'nvcuda.dll'; dlerror: nvcuda.dll not found
2020-11-26 13:09:21.6124                               UNKNOWN ERROR (303)
2020-11-26 13:09:21.6912                              iagnostic information for host: DESKTOP-0LJL0KB
2020-11-26 13:09:21.7033                              -0LJL0KB
2020-11-26 13:09:21.7122                              is optimized with oneAPI Deep Neural Network Library (oneDNN)to
use the following CPU in
To enable them in other
2020-11-26 13:09:22.3646                              initialized for platform Host (this does not guarantee that XLA
will be used). Devices:
2020-11-26 13:09:22.3761                              ): Host, Default Version
WARNING:tensorflow:From                               eras.engine.sequential) is deprecated and will be removed after
2021-01-01.
Instructions for updatin
Please use instead:* `np                              tion   (e.g. if it uses a `softmax` last-layer activation).* `(m
odel.predict(x) > 0.5).a                               `sigmoid` last-layer activation).
```

When eye is in closed state for long time and score value keeps on increased and reached threshold value, so system given warning issued to wake him up from drossiness



To check the working state of the developed system, we have given the images as Input from a webcam. The webcam periodically captures the series of images then gives them as an input to system. System fetches Region of Interest (Eye) and checks the state of an eye. If an eye is in open state, score value set to 0 otherwise score value keeps on increased by value one. In Figure, an eye's are in open state, so system has displayed the status as open and score is 0. So person is not feeling drowsiness. Consider Figure, person is feeling drowsiness, that's why an eye state is changed from open to closed and score value keeps on increasing. But the warning is not issued, because score value is not reached threshold value. Let us consider Figure, an eye state is in closed state for long time and score value keeps on increased and reached threshold value, so system given warning issued to wake him up from drossiness.

V. Conclusion

The proposed system can be used to detecting the drowsiness of person based on his eyes opening and closing state and blink rate. Once the person has drowsiness, system will calculate the eye open and closing ratio and compare with the threshold. If the values are not within the given threshold, system came to conclusion that person has drowsiness, then it will issue an alert and try to keep him awaken to avoid accidents. The system works well even under low light conditions if the camera delivers better output. The proposed system gives a portable solution which can be easily implemented in various devices. The proposed system not required any kind of expensive hardware or mechanical devices like they already available ones in the market.

References

[1]. Kirti Dang, Shanu Sharma, "Review and comparison of face detection algorithms", 7th International Conference on Cloud Computing, Data Science & Engineering - Confluence, 2017

[2]. Jongkil HyunCheol-Ho Choi, Byungin Moon, "Hardware Architecture of a Haar Classifier Based Face Detection System Using a Skip Scheme", IEEE International Symposium on Circuits and Systems (ISCAS),2021

[3]. jain, "face detection in color images; r. L. Hsu, m. Abdel-mottaleb, and a. K. Jain.

[4]. Wang Yang;Zheng Jiachun, "Real-time face detection based on YOLO" 1st IEEE International Conference on Knowledge Innovation and Invention (ICKII), 2018

[5]. Kartika Candra Kirana,Slamet Wibawanto,Heru Wahyu Herwanto, "Redundancy Reduction in Face Detection of Viola-Jones using the Hill Climbing Algorithm", 4th International Conference on Vocational Education and Training (ICOVET), 2020