L.N. Gumilyov Eurasian National University
Faculty of Information Technology
Department of «Information Systems»

OBJECT DETECTION AND FACE RECOGNITION

Viola Jones Algorithm

IWS №5 by discipline «Computer graphics and pattern recognition»

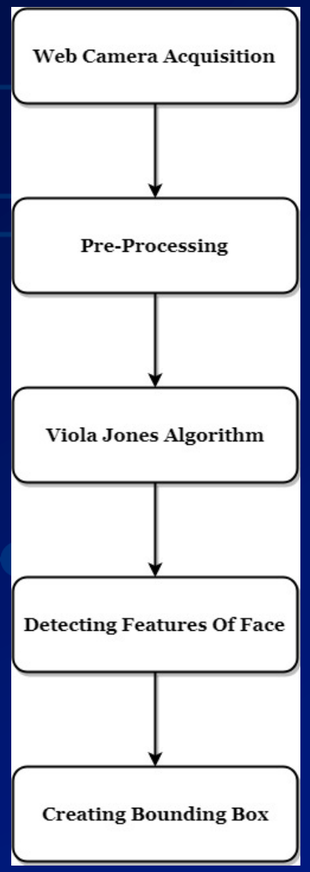
Presented by: Toleubay D.M.
Checked by: Prof. Dr. Zhukabayeva T.K.

INTRODUCTION

Viola Jones algorithm is named after two computer vision researchers who proposed the method in 2001, Paul Viola and Michael Jones in their paper, “Rapid Object Detection using a Boosted Cascade of Simple Features”. Despite being an outdated framework, Viola-Jones is quite powerful, and its application has proven to be exceptionally notable in real-time face detection. This algorithm is painfully slow to train but can detect faces in real-time with impressive speed.

The Viola Jones algorithm has four main steps, which we shall discuss in the sections to follow:

Steps Viola Jones alg.

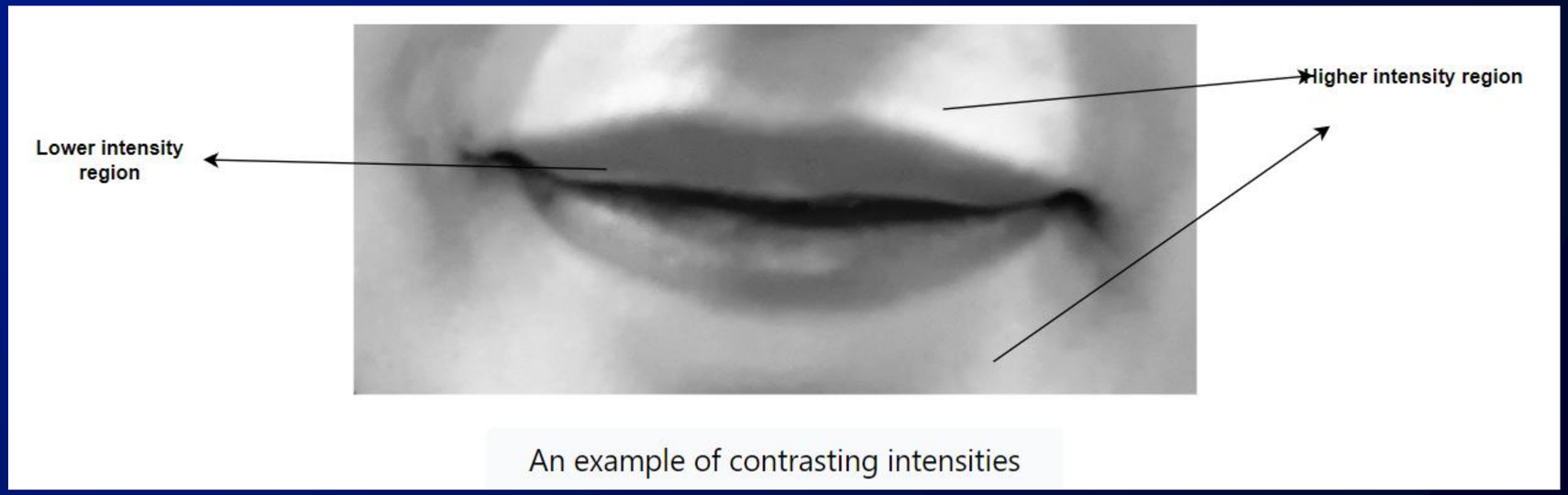


- Selecting Haar-like features
- Creating an integral image
- Running AdaBoost training
- Creating classifier cascades

What are Haar-Like Features?

Narrowing down to the application of object detection, features within any particular object can be described in terms of the difference of light intensities between neighboring landscapes.

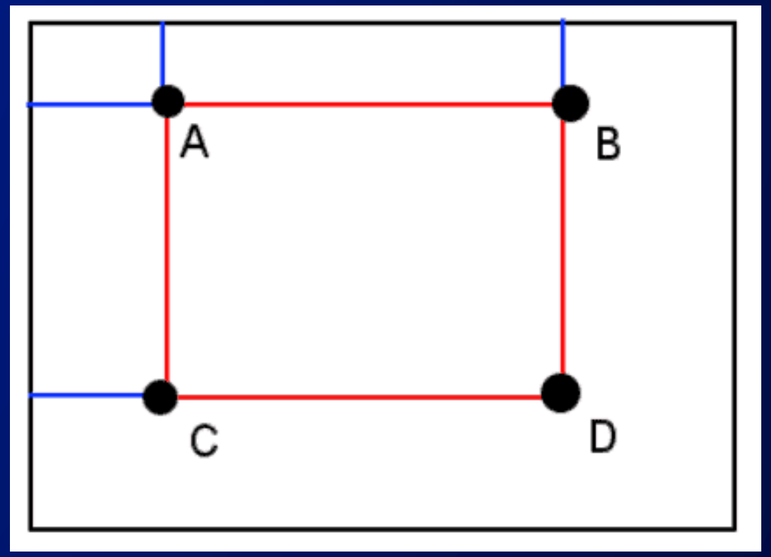
For instance, in the case of facial features, lips have a relatively darker intensity than the upper lip and lower-lip chin area. Similarly the edge of forehead is of a relatively higher intensity than that of the hair on the head.



An example of contrasting intensities

Integral images

It can be demonstrated that the sum in any rectangular area requires four values of the integral image, regardless of the window size:



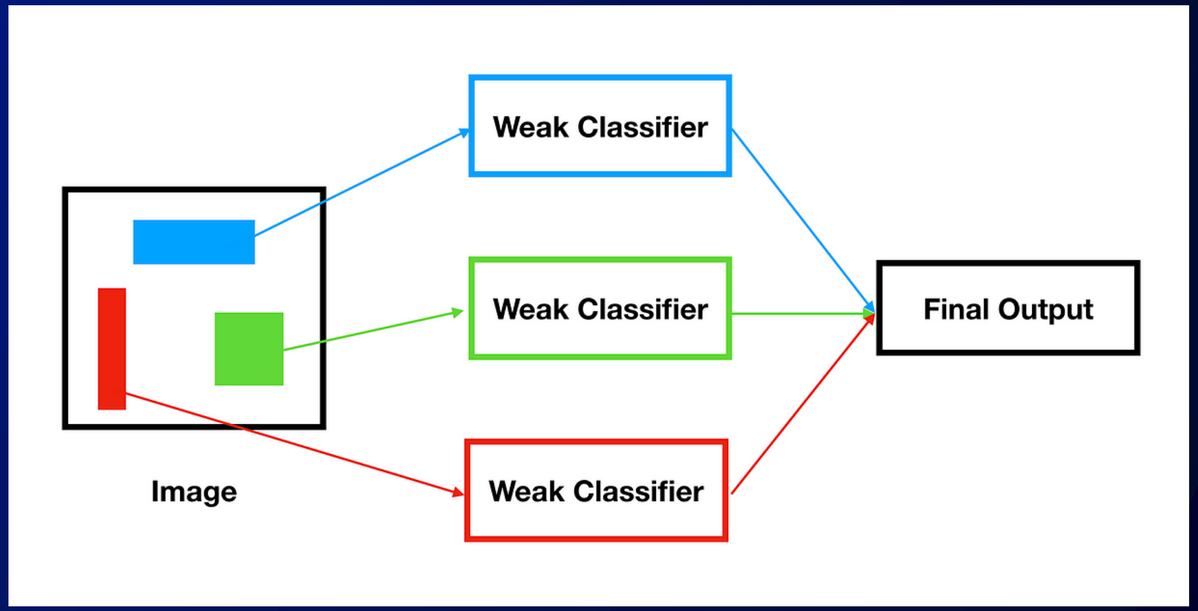
More specifically, the sum of pixel values within the rectangle ABCD shown above can be computed as:

$$\sum_{\substack{x_0<x\leq x_1 \\ y_0<y\leq y_1}} I(x,y) = ii(D) + ii(A) - ii(B) - ii(C). \tag{3}$$

Hence, using this formula, the time complexity of the computation of the Haar-like feature is $o(1)$.

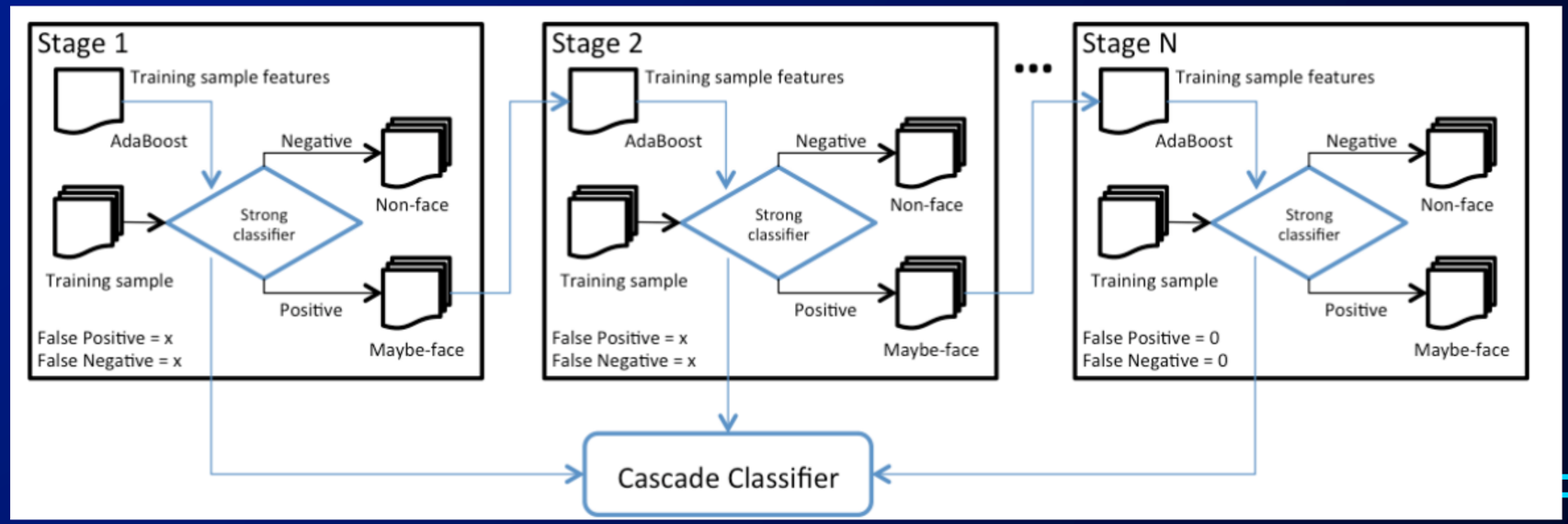
AdaBoost classifier

The number of features is approximately 16000 if the base resolution of the detector is 24×24. However, a small number of these features are useful for detecting faces. Viola-Jones algorithm uses AdaBoost to find the best features and to train a classifier. Each Haar-like feature represents a weak classifier. The final classifier is given by a linear combination of weak classifiers. Larger weights are associated with better classifiers using the AdaBoost learning algorithm.



Cascade classifier

The cascaded classifier is composed of stages each containing a strong classifier from AdaBoost. The job of each stage is to determine whether a given sub-window is definitely not a face or maybe a face. When a sub-window is classified to be a non-face by a given stage it is immediately discarded. Conversely a sub-window classified as a maybeface is passed on to the next stage in the cascade. It follows that the more stages a given sub-window passes, the higher the chance the sub-window contains a face.



Example

In [ ]:

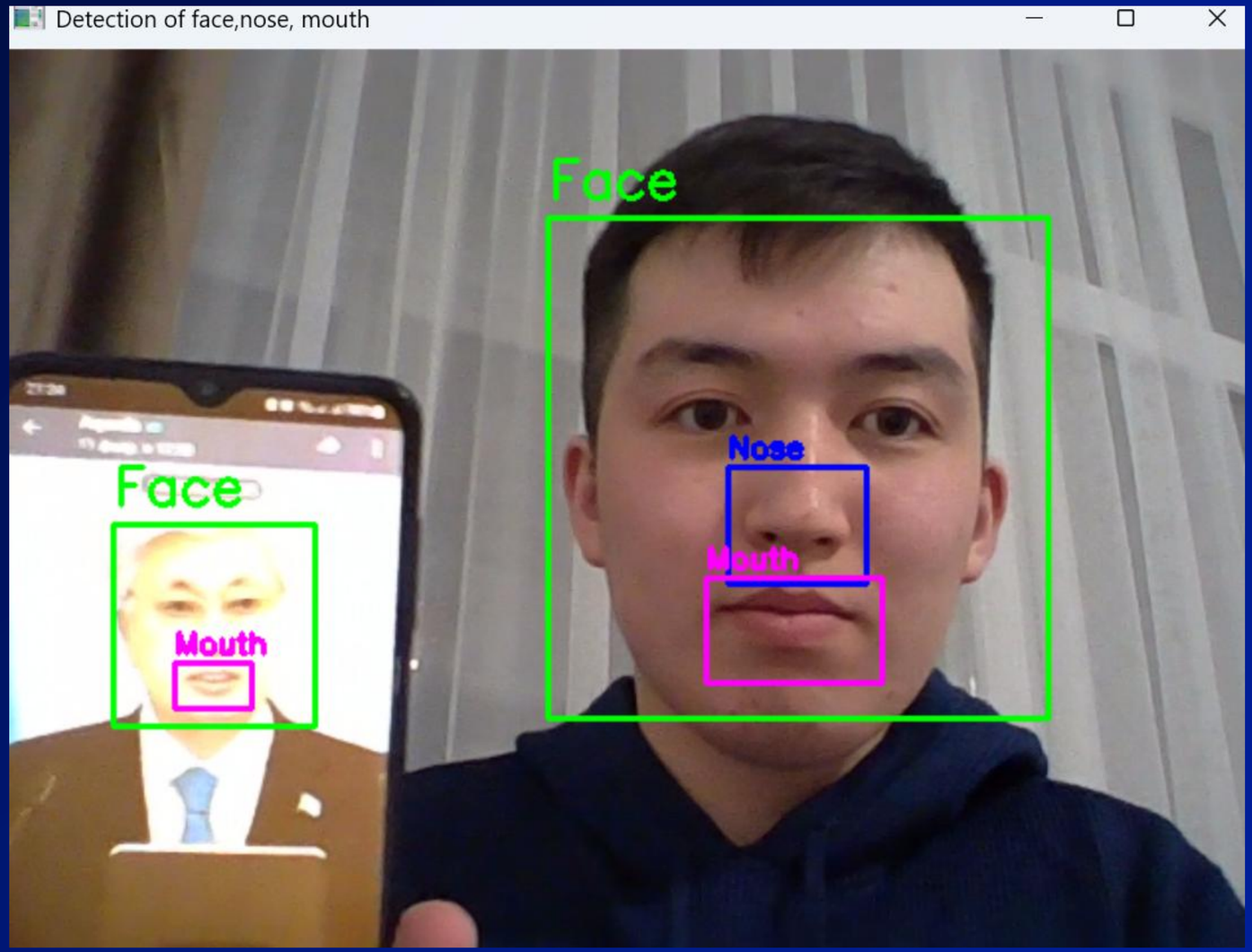
```python
import cv2

img = cv2.imread("C:/Users/opste/Downloads/my_group.jpg")

gray_img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

face_cascade = cv2.CascadeClassifier("C:/Users/opste/Downloads/haarcascade_frontalface_default.xml")
eye_cascade = cv2.CascadeClassifier("C:/Users/opste/Downloads/haarcascade_eye.xml")

# Applying the face detection method on the grayscale image
faces_rect = face_cascade.detectMultiScale(gray_img, 1.1, 9)

# Iterating through rectangles of detected faces
for (x, y, w, h) in faces_rect:
    cv2.rectangle(img, (x, y), (x+w, y+h), (0, 255, 0), 2)
    cv2.putText(img, "Face", (x, y-10), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 0), 2)

    face_roi_gray = gray_img[y:y+h, x:x+w]
    face_roi_color = img[y:y+h, x:x+w]

    eyes = eye_cascade.detectMultiScale(face_roi_gray, 1.1, 10)
    for (ex, ey, ew, eh) in eyes:
        cv2.rectangle(face_roi_color, (ex, ey), (ex+ew, ey+eh), (255, 0, 0), 2)
        cv2.putText(face_roi_color, "Eye", (ex, ey-5), cv2.FONT_HERSHEY_SIMPLEX, 0.4, (255, 0, 0), 1)

cv2.imshow('Detected faces and eyes', img)

cv2.waitKey(0)
```

https://github.com/Daniyar-manatuly/Pattern-recognation/blob/main/Haar%20cascade.ipynb

Result: My Group

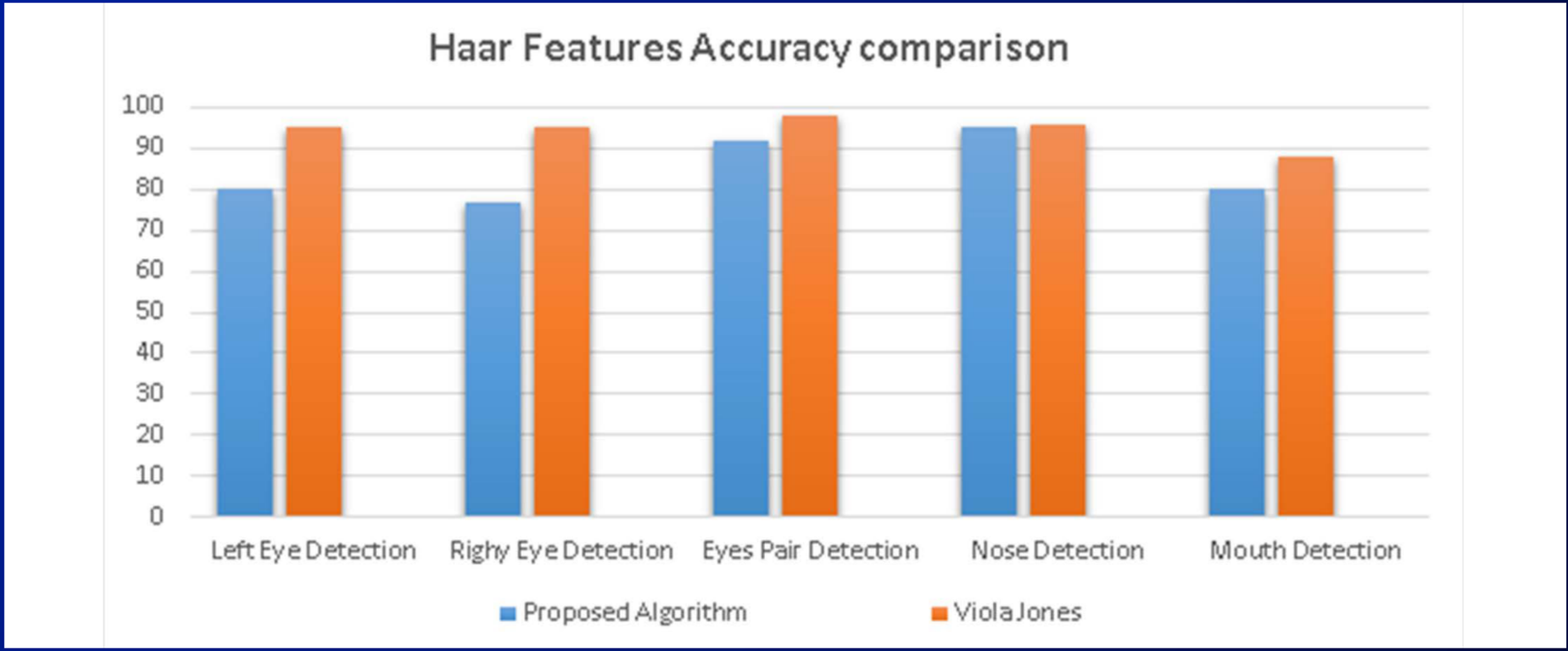
https://github.com/Daniyar-manatuly/Pattern-recognation/blob/main/Haar%20cascade.ipynb

Result:



https://github.com/Daniyar-manatuly/Pattern-recognation/blob/main/Real-time%20detection%20face%2C%20nose%2Cmouth.ipynb

Comparison in research



Masud, U., Saeed, T., Malaikah, H., Ul Islam, Muhammad F., Abbas, G. Smart Assistive System for Visually Impaired People Obstruction Avoidance Through Object Detection and Classification. IEEE Access, 2022. DOI: 10.1109/ACCESS.2022.3146320

Conclusion

The Viola-Jones algorithm revolutionized real-time face detection with its Haar-like features, integral images, and cascade classifiers, enabling fast and efficient processing. However, it struggles with variations in lighting, pose, and occlusion.

While modern deep learning methods offer higher accuracy, Viola-Jones remains relevant for resource-limited applications due to its speed and simplicity.

References

1. Face Detection using Viola Jones Algorithm. Great Learning. https://www.mygreatlearning.com/blog/viola-jones-algorithm/
2. Matlab code for Eye Tracking. Pantechsolutions. https://www.pantechsolutions.net/matlab-code-for-eye-tracking
3. Understanding and Implementing the Viola-Jones Image Classification Algorithm. Medium. https://medium.datadriveninvestor.com/understanding-and-implementing-the-viola-jones-image-classification-algorithm-85621f7fe20b
4. Masud, U., Saeed, T., Malaikah, H., Ul Islam, Muhammad F., Abbas, G. Smart Assistive System for Visually Impaired People Obstruction Avoidance Through Object Detection and Classification. IEEE Access, 2022. DOI: 10.1109/ACCESS.2022.3146320

LET'S CONNECT

Thank you for considering Wardiere Inc. as your technology partner. We look forward to helping your business reach new heights with our innovative solutions.