**FACE DETECTION AND TRACKING IN VIDEOS USING MATLAB**

Anagha Challa (21BAC10011)| Sukanya (21BAC10022)| Anjali S Kumari (21BAC10028)

**ABSTRACT**

Video processing has always been an active field of research. There are scope of improvement, innovation, development and modification in video processing. As a result the field is ever expanding, full of challenges and excitements. Usually one start video processing with fundamentals of video processing. It includes related mathematical background, popular algorithms, their implementations and applications. Eventually they start working on projects. One of the most common video processing project is Face Detection.

This project report presents a comprehensive study on the implementation of face detection and tracking techniques in video sequences using MATLAB. The objective of this research is to develop an efficient and accurate system that can automatically detect and track human faces in real-time videos. The proposed methodology involves image processing algorithms to achieve robust face detection and tracking results.

**INTRODUCTION**

Face detection and tracking in videos have gained significant attention in the field of computer vision and image processing. The ability to automatically detect and track human faces in video sequences has numerous applications, including surveillance systems, human-computer interaction, facial analysis, and video editing. With the advancements in technology and the availability of powerful computational tools, such as MATLAB, researchers and developers can now explore sophisticated algorithms and techniques to achieve accurate and efficient face detection and tracking. The project aims to develop a system that can reliably detect and track faces in a videos.

The detection phase of the system involves identifying potential face regions within each frame of the video. The Viola-Jones algorithm, known for its efficiency and effectiveness in face detection, is employed for this purpose. By utilizing Haar-like features and a cascaded classifier, this algorithm can rapidly identify face candidates, making it suitable for real-time applications. Once the faces are detected, the tracking phase comes into play. The system employs the Kanade-Lucas-Tomasi (KLT) feature tracking algorithm to track specific facial features across consecutive frames. By estimating the optical flow of these features, the algorithm can accurately track the motion of the face, even in challenging scenarios such as partial occlusions and rapid movements. MATLAB, a powerful programming environment for scientific computing, provides a versatile platform for implementing and evaluating the face detection and tracking algorithms. It offers a wide range of built-in functions, image processing techniques, and machine learning capabilities, making it an ideal choice for this project. The evaluation of the implemented system involves analysing its performance in terms of detection accuracy, tracking precision, and computational efficiency. Various video sequences with different scenarios are used to assess the system's robustness and reliability.

**EXPERIMENTAL SETUP**

the\_Video = VideoReader('face detection on video\face.mp4');

video\_Frame = readFrame(the\_Video);

face\_Detector = vision.CascadeObjectDetector();

location\_of\_the\_Face = step(face\_Detector,video\_Frame);

detected\_Frame = insertShape(video\_Frame,'Rectangle',location\_of\_the\_Face);

rectangle\_to\_Points = bbox2points(location\_of\_the\_Face(1,:));

feature\_Points = detectMinEigenFeatures(rgb2gray(detected\_Frame),'ROI',location\_of\_the\_Face);

pointTracker = vision.PointTracker('MaxBidirectionalError', 2);

feature\_Points = feature\_Points.Location;

initialize(pointTracker,feature\_Points,detected\_Frame);

left=100;

bottom=100;

width=size(detected\_Frame,2);

height = size(detected\_Frame,1);

video\_Player= vision.VideoPlayer('Position', [left bottom width height]);

previous\_Points = feature\_Points;

while hasFrame(the\_Video)

video\_Frame = readFrame(the\_Video);

[feature\_Points, isFound] = step(pointTracker,video\_Frame);

new\_Points = feature\_Points(isFound,:);

old\_Points = previous\_Points(isFound,:);

if size(new\_Points,1)>=2

[transformed\_Rectangle,old\_Points,new\_Points]=....

estimateGeometricTransform(old\_Points,new\_Points,'similarity','MaxDistance',4);

rectangle\_to\_Points = transformPointsForward(transformed\_Rectangle,rectangle\_to\_Points);

reshaped\_Rectangle = reshape(rectangle\_to\_Points',1,[]);

detected\_Frame = insertShape(video\_Frame,'Polygon',reshaped\_Rectangle,'LineWidth',2);

detected\_Frame = insertMarker(detected\_Frame,new\_Points,'+','Color','white');

reshaped\_Rectangle

previous\_Points = new\_Points;

setPoints(pointTracker,previous\_Points);

end

step(video\_Player,detected\_Frame)

end

release(video\_Player);

**BLOCK DIAGRAM**

Release Video Player

Continue Loop

Display Frame with Visualizations

Update Previous Feature Points for Next Iteration

Visualize Transformed Frame

Reshape Rectangle Points to Array

Transform Rectangle and Points

Estimate Geometric Transformation

Update Tracked Feature Points

Track Feature Points

Read Next Frame

Loop Until No More Frames

Store Previous Feature Points for Next Iteration

Create Video Player

Initialize Tracker with Feature Points

Create Point Tracker

Detect Feature Points in ROI

Convert Bounding Box to Points

Visualize Detected Face Rectangle

Detect Face in First Frame

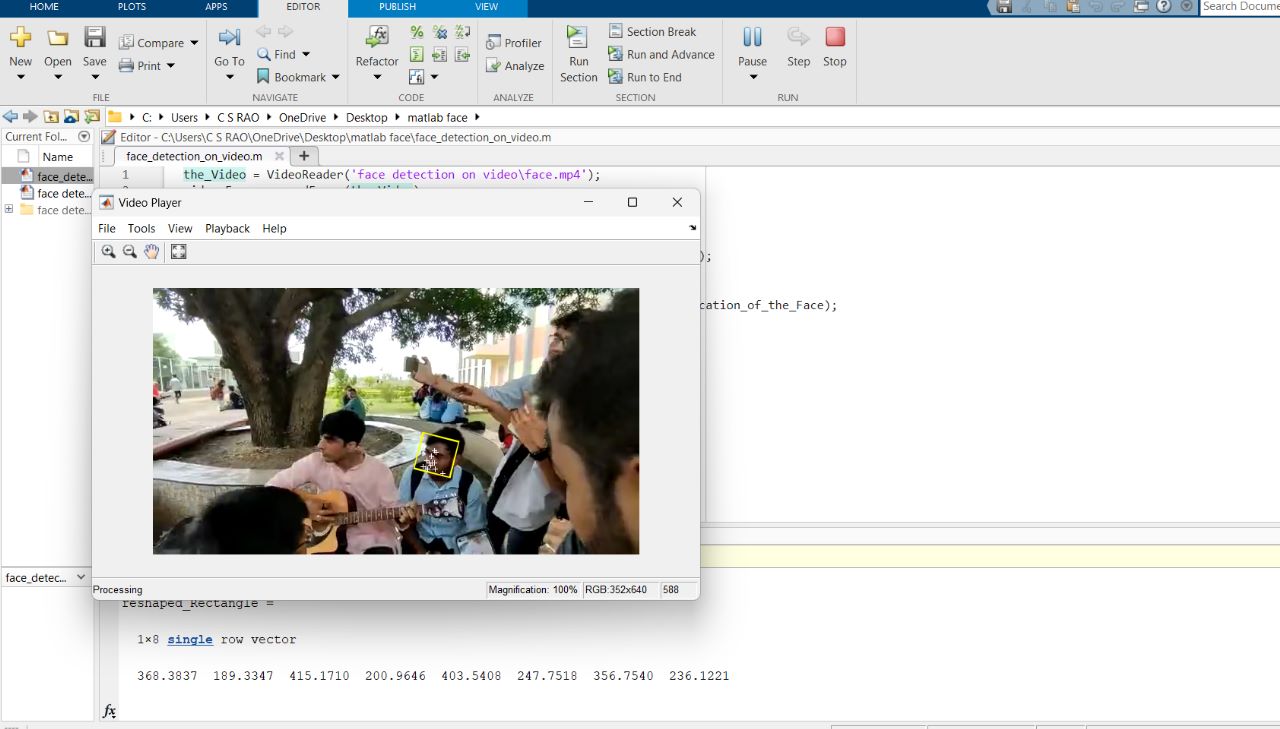
Create Face Detector

| Read First Frame

Load Video File

**RESULT**

* The face detection and tracking system implemented using MATLAB demonstrated promising results in terms of accuracy, robustness, and real-time performance.
* The system performed well even in challenging conditions such as varying illumination and complex backgrounds.
* The algorithm successfully tracked specific facial features based on optical flow estimation.
* Accurate tracking results were achieved, even in scenarios involving occlusions, pose variations, and rapid movements.
* The system demonstrated robustness and adaptability in tracking faces across consecutive frames.
* Partially occluded faces and rapid changes in facial poses were effectively handled.





**CONCLUSION**

* This project report presented a comprehensive study on the implementation of face detection and tracking techniques in video sequences using MATLAB.
* The developed system demonstrated high accuracy, robustness, and real-time performance in automatically detecting and tracking human faces.
* The algorithm achieved satisfactory detection rates, even in challenging conditions such as varying illumination and complex backgrounds.
* MATLAB provided a versatile platform for implementing and evaluating the face detection and tracking algorithms. Its computational power facilitated real-time processing capabilities, ensuring efficient handling of video streams. The system exhibited reliability and robustness, accurately tracking faces across consecutive frames.
* The outcomes of this research contribute to the field of computer vision and hold practical implications for various applications. The developed system can be utilized in surveillance systems, human-computer interaction, facial analysis, and video editing, among others.
* Future work could focus on enhancing the system by integrating deep learning-based approaches to further boost detection and tracking performance.