VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belagavi: 590 018

A Mini Project Report

On

"Iris-Segmentation Using Mediapipe in Python"

Submitted in partial fulfillment of the requirements for the award of the Degree of Bachelor of Engineering in Computer Science and Engineering.

Submitted by

J.Yeshas -1VE21CS064

Under the Guidance of

Mr. Suresh P

Assistant Professor

Dept. of CSE,

SVCE, Bengaluru.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SRI VENKATESHWARA COLLEGE OF ENGINEERING

Affiliated to VTU Belgaum & Approved by AICTE New Delhi) an ISO 9001:2008 Certified, Kempegowda International Airport road, Vidyanagar, Bengaluru, Karnataka, India-562157

2023– 2024

SRI VENKATESHWARA COLLEGE OF ENGINEERING
Vidyanagar, Bengaluru, Karnataka, India-562157

Department ofComputer Science & Engineering

CERTIFICATE

This is to certify that Mini Project entitled **"Iris-Segmentation Using Mediapipe in Python"** is submitted by **J.Yeshas** bearing **USN 1VE21CS064** on partial fulfillment of sixth semester, Bachelor of Engineering in Computer Science and Engineering, Visvesvaraya Technological University for the academic year 2023-2024.

…………………………………	……………………………..
Signature of Course Teacher with date	**Signature of the HOD with date and seal**

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without complementing those who made it possible, whose guidance and encouragement made our efforts successful.

I would like to express my sincere gratitude to the highly esteemed institution **SRI VENKATESHWARA COLLEGE OF ENGINEERING.** The institution is providing me the knowledge to become a software engineer.

I express my sincere gratitude to **Dr. NAGESWARA GUPTHA M,** Principal, SVCE, Bengaluru for providing the required facility.

I am extremely thankful to **Dr. HEMA M S,** HOD of CSE, SVCE, for providing the support and encouragement.

I am grateful to **Mr. Suresh P,** Asst. Professor, Dept of CSE,SVCE who helped me to complete this project successfully by providing guidance, encouragement and valuable suggestion during the entire period of the project. I thank all the computer science and engineering staff and others who helped me directly or indirectly to meet my project work with grand success.

Finally, I am grateful to my parents and friends for their invaluable support, guidance and encouragement.

J.YESHAS[1VE21CS064]

DEPARTMENT VISION

Global Excellence with Local relevance in Information Science and Engineering Education, Research and Development.

DEPARTMENT MISION

M1: Strive for academic excellence in Information Science and Engineering through student centric innovative teaching-learning process, competent faculty members, efficient assessment and use of ICT.

M2: Establish Centre for Excellence in various vertical of Information Science and Engineering to promote collaborative research and Industry Institute Interaction.

M3: Transform the engineering aspirants to socially responsible, ethical, and technically competent and value added professional or entrepreneur.

PROGRAM OUTCOMES

1. Engineering Knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. Problem Analysis: Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3. Design/development of Solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4. Conduct Investigations of Complex Problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. Modern Tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

6. The Engineer and Society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. Environment and Sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. Individual and Team Work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. Project Management and Finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. Life-long Learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM EDUCATIONAL OBJECTIVES

Knowledge:

Computer Science and Engineering Graduates will have professional technical career in inter disciplinary domains providing innovative and sustainable solutions using modern tools.

Skills:

Computer Science and Engineering Graduates will have effective communication, leadership, team building, problem solving, decision making and creative skills.

Attitude:

Computer Science and Engineering Graduates will practice ethical responsibilities towards their peers, employers and society.

PROGRAM SPECIFIC OUTCOMES

PSO1:

Ability to adopt quickly for any domain, interact with diverse group of individuals and be an entrepreneur in a societal and global setting.

PSO2:

Ability to visualize the operations of existing and future software Application

ABSTRACT

This project explores an integrated approach to iris segmentation using OpenCV and Mediapipe, aiming to achieve real-time and accurate localization of the iris within facial images. Leveraging OpenCV's robust image processing capabilities and Mediapipe's precise facial landmark detection, the system enables rapid and reliable iris segmentation. This capability is essential for applications in biometrics, where secure and precise identification is paramount, and in medical diagnostics, facilitating early detection and monitoring of ocular diseases.

The methodology involves capturing and processing video frames in real-time, applying image processing techniques to detect facial landmarks, and specifically identifying the iris region using geometric and mathematical analyses. Results demonstrate the system's effectiveness in accurately delineating the iris under various lighting conditions and facial occlusions, showcasing its potential in enhancing security systems, medical diagnostics, and interactive technologies.

Future work entails integrating advanced machine learning algorithms to further improve segmentation accuracy and robustness, expanding the system's capabilities to include pupil tracking and eye movement analysis. Such advancements promise to refine iris segmentation for broader applications in augmented reality interfaces, driver monitoring systems, and personalized healthcare technologies. By bridging technological innovation with practical applications, this project contributes to advancements in both security technology and healthcare diagnostics, offering tangible benefits in safety, health monitoring, and user interactio

CONTENT

CHAPTER-1

INTRODUCTION

Iris segmentation is a crucial task in the field of computer vision and biometrics, which involves the precise localization and delineation of the iris within an eye image. The accuracy of iris segmentation significantly impacts the effectiveness of subsequent stages in iris recognition, making it an essential component in various applications, including security systems, user authentication, and medical diagnostics.

In security systems, accurate iris segmentation ensures that only authorized individuals gain access to restricted areas or information, providing a high level of security. For user authentication, particularly in devices like smartphones and laptops, precise iris segmentation enhances the reliability of biometric authentication, offering a seamless and secure user experience. In medical diagnostics, iris segmentation can aid in the early detection of certain diseases and conditions by analyzing the patterns and features of the iris.

This project leverages advanced image processing techniques and computer graphics to achieve precise and reliable iris segmentation. By utilizing OpenCV, a powerful open-source computer vision library, and Mediapipe, a framework that facilitates the development of machine learning pipelines, combined with the versatility of Python programming, this project aims to create an efficient and accurate iris segmentation system.

OpenCV provides a comprehensive set of tools for image processing and computer vision tasks, including filtering, edge detection, and morphological operations, which are essential for enhancing the quality of eye images and accurately identifying the iris. Mediapipe offers pre-built solutions and customizable pipelines that can be tailored to the specific requirements of iris segmentation, enabling rapid development and deployment. Python, with its simplicity and extensive libraries, serves as the backbone of this project, allowing for seamless integration of OpenCV and Mediapipe functionalities.

Through the integration of these technologies, this project seeks to push the boundaries of iris segmentation accuracy, ensuring that the segmented iris is as close to the true iris region as possible. The outcome of this project holds the potential to improve the performance and reliability of biometric systems, enhance security measures, and contribute to advancements in medical diagnostics.

Through the integration of these technologies, this project seeks to push the boundaries of iris segmentation accuracy, ensuring that the segmented iris is as close to the true iris region as possible. The outcome of this project holds the potential to improve the performance and reliability of biometric systems, enhance security measures, and contribute to advancements in medical diagnostics.

CHAPTER-2

LITERATURE SURVEY

2.1Background and Significance: Iris recognition systems have become very popular because they are both highly reliable and non-invasive. Unlike other biometric methods like fingerprint or facial recognition, iris recognition uses the unique patterns in the iris, which stay the same throughout a person's life. This makes it perfect for high-security uses. The process involves several steps: capturing the image, preprocessing it, segmenting the iris, extracting features, and matching them. Among these, iris segmentation is especially tricky due to different lighting conditions, eyelids and eyelashes getting in the way, and noise in the images.

Lighting can create shadows and highlights that hide parts of the iris, making segmentation difficult. Eyelids and eyelashes can cover parts of the iris, adding to the complexity. Plus, image noise from reflections or motion blur can further complicate the task. Despite these challenges, accurate iris segmentation is crucial for the success of the later stages in iris recognition, as mistakes here can lead to errors in feature extraction and matching, reducing the system's reliability.

2.2Previous Work: Researchers have tried many methods for iris segmentation, from traditional image processing to modern machine learning. Early techniques mainly used edge detection and thresholding to segment the iris. These straightforward methods often struggled with real-world image complexities.

One widely used early technique is the Hough Transform, which finds the circular boundary of the iris by identifying the best-fitting circle for the edge points. Although effective in controlled settings, the Hough Transform can be sensitive to noise and occlusions. Active contours, or snakes, refine the iris boundary by minimizing an energy function. This method is more flexible with irregular shapes but can still be affected by occlusions and noise.

Recently, deep learning-based segmentation networks have shown great potential in overcoming traditional methods' limitations. Convolutional neural networks (CNNs) and other deep learning models can learn complex features from large datasets, enabling more robust and accurate iris segmentation. These models handle variations in lighting, occlusions, and noise, making them ideal for real-world use.

Mediapipe, developed by Google, is a recent framework offering robust solutions for detecting face and hand landmarks. It uses advanced machine learning and optimized pipelines for real-time performance. Mediapipe's modular design allows easy integration and customization, making it a valuable tool for tasks like iris segmentation. By leveraging Mediapipe, this project aims to use state-of-the-art technology for precise and reliable iris segmentation.

In summary, while traditional methods laid the groundwork, modern machine learning techniques and frameworks like Mediapipe provide significant advancements in accuracy and robustness. This project builds on these advancements to develop a cutting-edge iris segmentation system that effectively addresses real-world challenges.

CHAPTER-3

CODE

3.1-Main.py code

```
from operator import rshift

import cv2 as cv

import numpy as np

import mediapipe as mp

mp_face_mesh = mp.solutions.face_mesh

LEFT_EYE =[ 362, 382, 381, 380, 374, 373, 390, 249, 263, 466, 388, 387, 386,385,384,398]

# right eyes indices

RIGHT_EYE=[ 33, 7, 163, 144, 145, 153, 154, 155, 133, 173, 157, 158, 159, 160, 161 , 246]

LEFT_IRIS = [474,475, 476, 477]

RIGHT_IRIS = [469, 470, 471, 472]

cap = cv.VideoCapture(0)

with mp_face_mesh.FaceMesh(

    max_num_faces=1,

    refine_landmarks=True,

    min_detection_confidence=0.5,

    min_tracking_confidence=0.5) as face_mesh:

    while True:

        ret, frame = cap.read()
```

```
    if not ret:
        break
    frame = cv.flip(frame, 1)
    rgb_frame = cv.cvtColor(frame, cv.COLOR_BGR2RGB)
    img_h, img_w = frame.shape[:2]
    results = face_mesh.process(rgb_frame)
    if results.multi_face_landmarks:
        # print(results.multi_face_landmarks[0].landmark)
        mesh_points=np.array([np.multiply([p.x, p.y], [img_w, img_h]).astype(int) for p in
results.multi_face_landmarks[0].landmark])
        # print(mesh_points.shape)
        # cv.polylines(frame, [mesh_points[LEFT_IRIS]], True, (0,255,0), 1, cv.LINE_AA)
        # cv.polylines(frame, [mesh_points[RIGHT_IRIS]], True, (0,255,0), 1, cv.LINE_AA)
        (l_cx, l_cy), l_radius = cv.minEnclosingCircle(mesh_points[LEFT_IRIS])
        (r_cx, r_cy), r_radius = cv.minEnclosingCircle(mesh_points[RIGHT_IRIS])
        center_left = np.array([l_cx, l_cy], dtype=np.int32)
        center_right = np.array([r_cx, r_cy], dtype=np.int32)
        cv.circle(frame, center_left, int(l_radius), (255,0,255), 1, cv.LINE_AA)
        cv.circle(frame, center_right, int(r_radius), (255,0,255), 1, cv.LINE_AA)

    cv.imshow('img', frame)
    key = cv.waitKey(1)
```

```
    if key ==ord('q'):
        break
cap.release()
cv.destroyAllWindows()
```

3.2-Segmnetation.py Code

```
import cv2 as cv
import numpy as np
import mediapipe as mp
mp_face_mesh = mp.solutions.face_mesh

# left eyes indices
LEFT_EYE =[ 362, 382, 381, 380, 374, 373, 390, 249, 263, 466, 388, 387, 386, 385,384, 398 ]
# right eyes indices
RIGHT_EYE=[ 33, 7, 163, 144, 145, 153, 154, 155, 133, 173, 157, 158, 159, 160, 161 , 246]
# irises Indices list
LEFT_IRIS = [474,475, 476, 477]
RIGHT_IRIS = [469, 470, 471, 472]
cap = cv.VideoCapture(0)
with mp_face_mesh.FaceMesh(
  max_num_faces=1,
```

```
refine_landmarks=True,

min_detection_confidence=0.5,

min_tracking_confidence=0.5) as face_mesh:

while True:

    ret, frame = cap.read()

    if not ret:

        break

    frame = cv.flip(frame, 1)

    rgb_frame = cv.cvtColor(frame, cv.COLOR_BGR2RGB)

    img_h, img_w = frame.shape[:2]

    results = face_mesh.process(rgb_frame)

    mask = np.zeros((img_h, img_w), dtype=np.uint8)



    if results.multi_face_landmarks:

        # print((results.multi_face_landmarks[0]))



        # [print(p.x, p.y, p.z ) for p in results.multi_face_landmarks[0].landmark]



        mesh_points=np.array([np.multiply([p.x, p.y], [img_w, img_h]).astype(int)

        for p in results.multi_face_landmarks[0].landmark])
```

```
    # cv.polylines(frame, [mesh_points[LEFT_IRIS]], True, (0,255,0), 1, cv.LINE_AA)
    # cv.polylines(frame, [mesh_points[RIGHT_IRIS]], True, (0,255,0), 1, cv.LINE_AA)

    (l_cx, l_cy), l_radius = cv.minEnclosingCircle(mesh_points[LEFT_IRIS])
    (r_cx, r_cy), r_radius = cv.minEnclosingCircle(mesh_points[RIGHT_IRIS])
    center_left = np.array([l_cx, l_cy], dtype=np.int32)
    center_right = np.array([r_cx, r_cy], dtype=np.int32)
    cv.circle(frame, center_left, int(l_radius), (0,255,0), 2, cv.LINE_AA)
    cv.circle(frame, center_right, int(r_radius), (0,255,0), 2, cv.LINE_AA)

    # cv.circle(frame, center_left, 1, (0,255,0), -1, cv.LINE_AA)
    # cv.circle(frame, center_right, 1, (0,255,0), -1, cv.LINE_AA)

    # drawing on the mask
    cv.circle(mask, center_left, int(l_radius), (255,255,255), -1, cv.LINE_AA)
    cv.circle(mask, center_right, int(r_radius), (255,255,255), -1, cv.LINE_AA)

cv.imshow('Mask', mask)
cv.imshow('img', frame)
key = cv.waitKey(1)
if key ==ord('q'):
```

```
        break

cap.release()

cv.destroyAllWindows()
```

3.3-Eye-Position.py Code

```
from operator import rshift
import cv2 as cv
import numpy as np
import mediapipe as mp
import math

mp_face_mesh = mp.solutions.face_mesh

LEFT_EYE = [362, 382, 381, 380, 374, 373, 390, 249, 263, 466, 388, 387, 386, 385, 384,
398]
RIGHT_EYE = [33, 7, 163, 144, 145, 153, 154, 155, 133, 173, 157, 158, 159, 160, 161, 246]
RIGHT_IRIS = [474, 475, 476, 477]
LEFT_IRIS = [469, 470, 471, 472]
L_H_LEFT = [33]  # right eye right most landmark
L_H_RIGHT = [133]  # right eye left most landmark
R_H_LEFT = [362]  # left eye right most landmark
R_H_RIGHT = [263]  # left eye left most landmark

def euclidean_distance(point1, point2):
    x1, y1 = point1.ravel()
    x2, y2 = point2.ravel()
    distance = math.sqrt((x2 - x1)**2 + (y2 - y1)**2)
    return distance

def iris_position(iris_center, right_point, left_point):
    center_to_right_dist = euclidean_distance(iris_center, right_point)
```

```
    total_distance = euclidean_distance(right_point, left_point)

    if total_distance == 0:
        raise ValueError("Total distance is zero, which will cause a division by zero error.")

    ratio = center_to_right_dist / total_distance
    iris_position = ""
    if ratio <= 0.42:
        iris_position = "right"
    elif ratio > 0.42 and ratio <= 0.57:
        iris_position = "center"
    else:
        iris_position = "left"

    return iris_position, ratio

cap = cv.VideoCapture(0)

with mp_face_mesh.FaceMesh(
    max_num_faces=2,
    refine_landmarks=True,
    min_detection_confidence=0.5,
    min_tracking_confidence=0.5
) as face_mesh:
    while True:
        ret, frame = cap.read()
        if not ret:
            break
        frame = cv.flip(frame, 1)
        rgb_frame = cv.cvtColor(frame, cv.COLOR_BGR2RGB)
        img_h, img_w = frame.shape[:2]
        results = face_mesh.process(rgb_frame)
        if results.multi_face_landmarks:
            mesh_points = np.array([np.multiply([p.x, p.y], [img_w, img_h]).astype(int) for p in
```

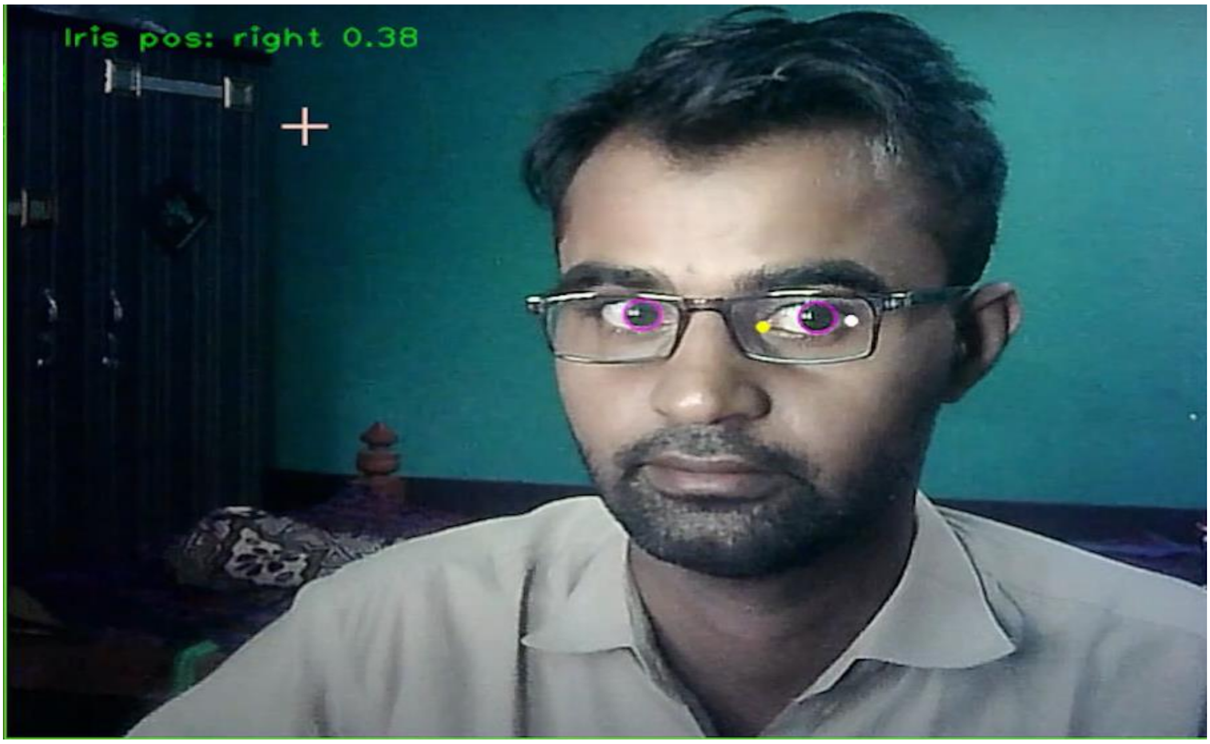
```
results.multi_face_landmarks[0].landmark])
            (l_cx, l_cy), l_radius = cv.minEnclosingCircle(mesh_points[LEFT_IRIS])
            (r_cx, r_cy), r_radius = cv.minEnclosingCircle(mesh_points[RIGHT_IRIS])
            center_left = np.array([l_cx, l_cy], dtype=np.int32)
            center_right = np.array([r_cx, r_cy], dtype=np.int32)
            cv.circle(frame, center_left, int(l_radius), (255, 0, 255), 1, cv.LINE_AA)
            cv.circle(frame, center_right, int(r_radius), (255, 0, 255), 1, cv.LINE_AA)
            cv.circle(frame, mesh_points[R_H_RIGHT][0], 3, (255, 255, 255), -1, cv.LINE_AA)
            cv.circle(frame, mesh_points[R_H_LEFT][0], 3, (0, 255, 255), -1, cv.LINE_AA)
            try:
                iris_pos, ratio = iris_position(center_right, mesh_points[R_H_RIGHT][0],
mesh_points[R_H_LEFT][0])
                cv.putText(frame, f"iris_pos:{iris_pos} {ratio:.2f}", (30, 30),
cv.FONT_HERSHEY_PLAIN, 1.2, (0, 255, 0), 1, cv.LINE_AA)
            except ValueError as e:
                cv.putText(frame, str(e), (30, 30), cv.FONT_HERSHEY_PLAIN, 1.2, (0, 0, 255),
1, cv.LINE_AA)
        cv.imshow('img', frame)
        key = cv.waitKey(1)
        if key == ord('q'):
            break

cap.release()
cv.destroyAllWindows()
```

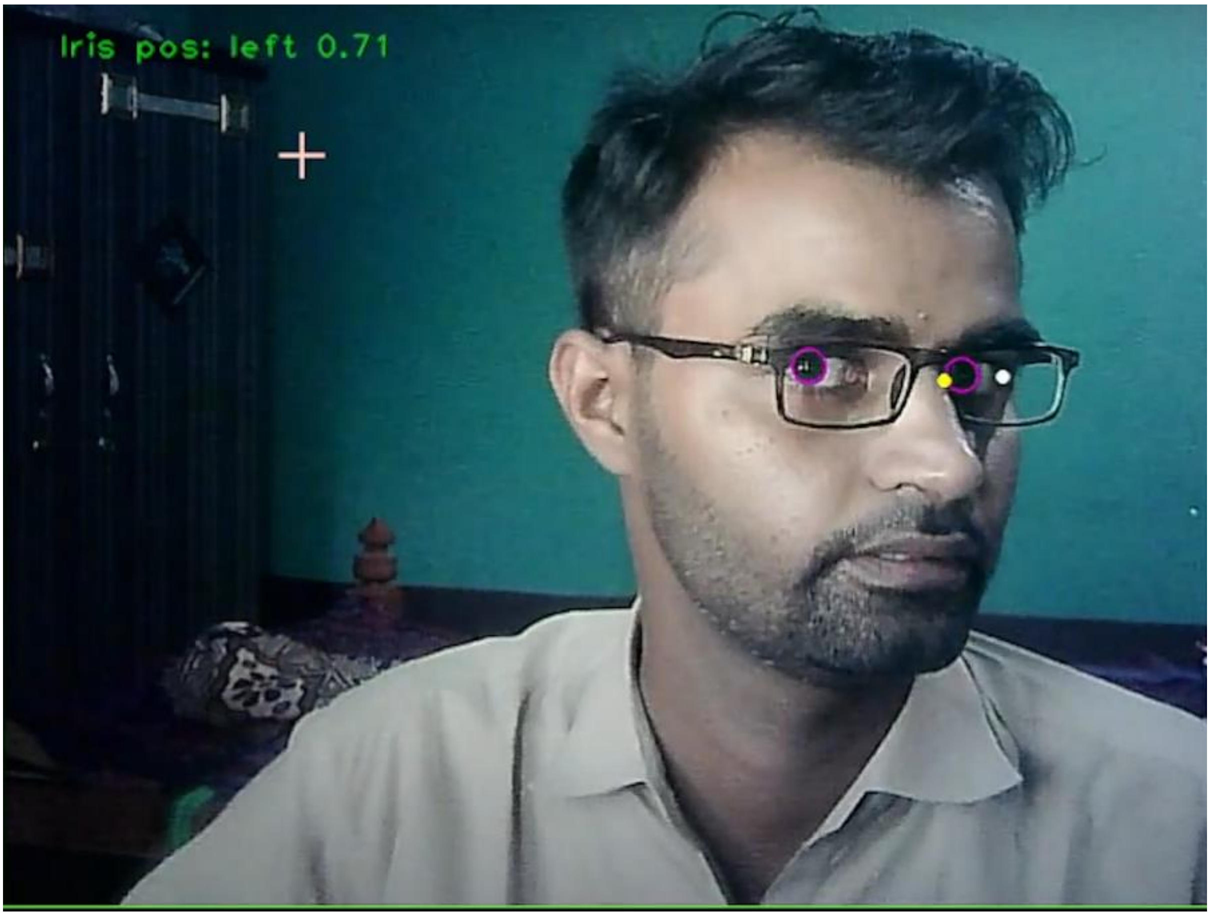
CHAPTER-4

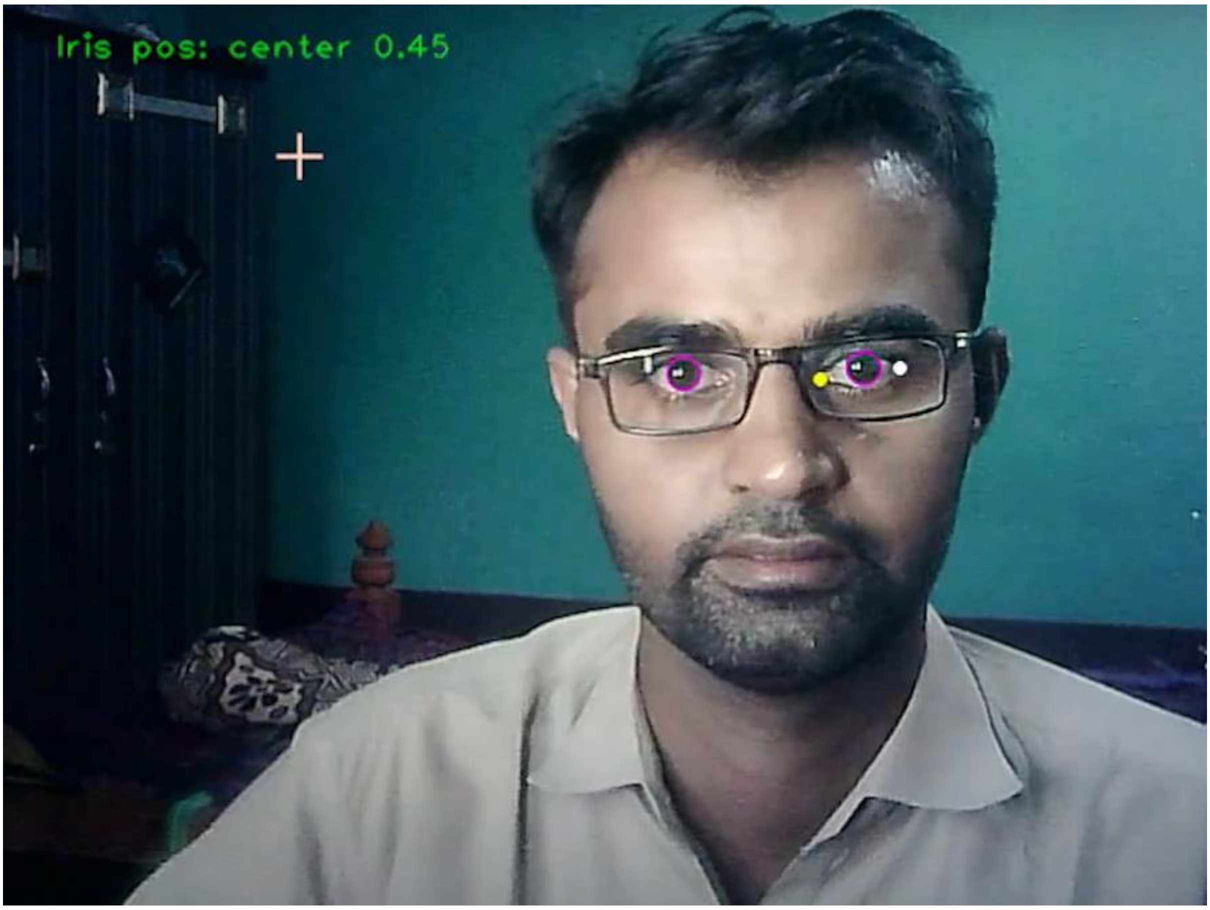
OUTPUTS

Main.py

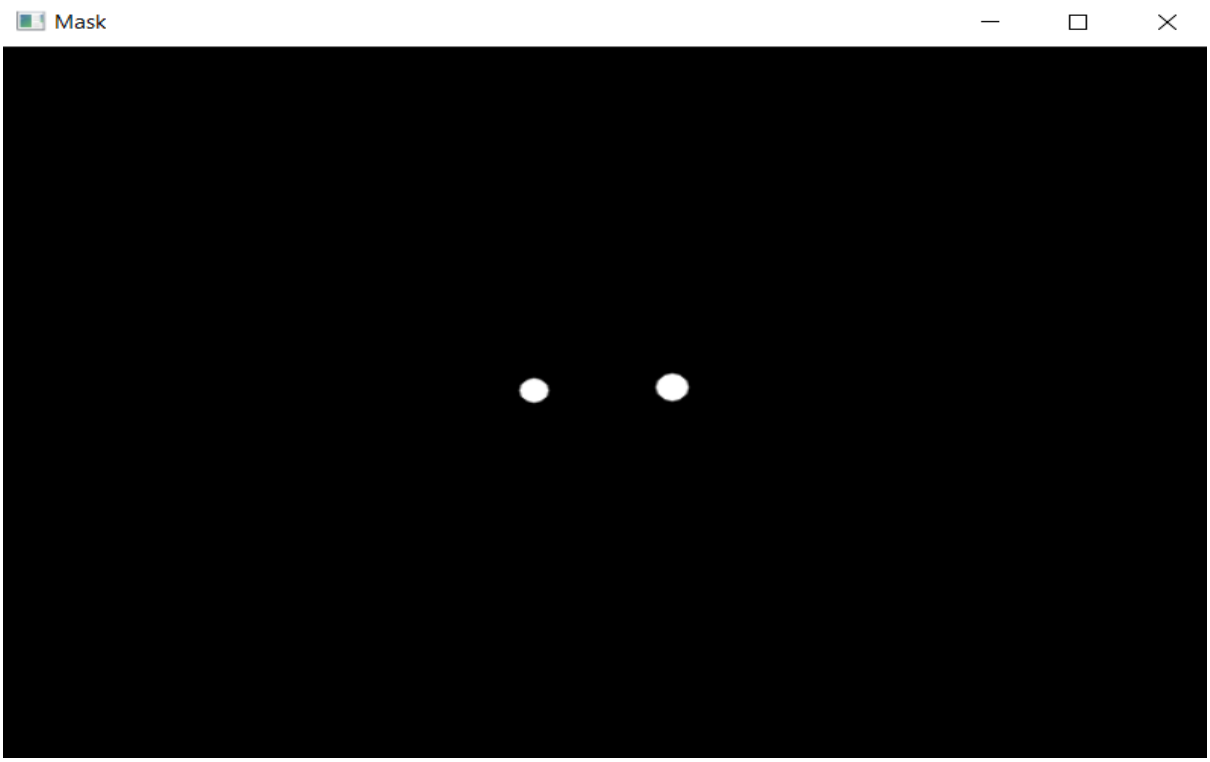
Eye_position.py







Segmentation_mask.py



CHAPTER-5

APPLICATIONS OF IRIS-SEGMENTATION USING MEDIAPIPE IN PYTHON

5.1Biometric Security Systems: Enhancing Security and Access Control, Iris recognition is incredibly reliable for verifying identity. With accurate iris segmentation, we can ensure that only authorized people get access to sensitive places like government buildings, military bases, and research labs, keeping these areas secure.

5.2 Medical Diagnostics: Assisting Ocular Health Monitoring, When it comes to eye health, precise iris segmentation can make a big difference. It helps doctors diagnose and monitor diseases like glaucoma, diabetic retinopathy, and iris melanoma more effectively. Clear images of the iris mean better detection and treatment, which ultimately leads to healthier patients.

5.3 Human-Computer Interaction (HCI): Enabling Intuitive and Natural Interfaces: Imagine controlling your devices just by looking at them! Accurate iris and eye-tracking technology can make this a reality, especially in augmented reality (AR) and virtual reality (VR). This creates more immersive gaming experiences, smoother virtual meetings, and easier remote collaboration.

5.4 Driver Monitoring Systems: Improving Road Safety, Keeping drivers safe is a top priority. By using iris and eye-tracking technology, we can monitor drivers for signs of drowsiness or distraction. If the system detects a problem, it can alert the driver, helping to prevent accidents and keep everyone on the road safer.

5.5 **Consumer Electronics Enhancing User Experience**: Our gadgets can get even smarter with iris and eye-tracking tech. Think about your phone or tablet recognizing you for secure access, scrolling automatically as you read, or sending notifications only when you're paying attention. It makes using these devices more intuitive and personal.

5.6 Smart Home Systems: Intelligent Environment Control: Imagine your home adjusting to you automatically. With eye-tracking integrated into smart home systems, your home can adjust the lighting, temperature, or even turn on your favorite show based on where you're looking. It's all about making your living space more comfortable and responsive to your needs.

5.7 Retail and Advertising: Personalized Customer Engagement, Retail and advertising can get a major upgrade with eye-tracking. By understanding where customers are looking, businesses can tailor their ads and product displays to catch your eye more effectively. This means better-targeted marketing and a more engaging shopping experience.

5.8 Assistive Technologies: Supporting Individuals with Disabilities, For people with disabilities, iris and eye-tracking technology can be life-changing. Eye-controlled communication devices, for example, allow those with limited mobility to interact with computers and communicate more easily, significantly improving their quality of life.

CHAPTER-6

CONCLUSION

This project represents a significant advancement in iris segmentation through the synergistic use of OpenCV and Mediapipe. By harnessing OpenCV's robust image processing capabilities and Mediapipe's precision in facial landmark detection, the system achieves real-time iris localization with high accuracy. This capability is pivotal for applications in biometrics, where precise identification is crucial for security systems and access control mechanisms. Moreover, in fields like medical diagnostics, the ability to accurately segment the iris aids in the early detection and monitoring of ocular diseases such as glaucoma and diabetic retinopathy. The integration of these tools not only ensures efficient processing of video frames but also enhances the system's reliability in adverse conditions such as varying lighting and facial occlusions.

Looking ahead, future iterations of this system could explore the integration of advanced machine learning techniques. Deep learning models, for example, could further refine iris segmentation by learning complex patterns and variations in iris structures. This enhancement would not only improve accuracy but also extend the system's capabilities to handle more diverse and challenging scenarios. Additionally, expanding the system's functionalities to include pupil tracking and eye movement analysis presents promising avenues for research and development. These extensions could enable applications in gaze-based interaction systems for augmented reality (AR) and virtual reality (VR), enhancing user experiences with intuitive interfaces that respond to eye movements in real time.

Beyond technical enhancements, ongoing research could focus on optimizing the system's performance across different demographic groups and environmental conditions. Ensuring robustness and reliability across diverse user populations, including individuals with varying iris shapes and sizes, remains a critical area of improvement. Furthermore, exploring interdisciplinary collaborations with medical professionals could provide valuable insights into refining the system for clinical applications. By bridging the gap between technological innovation and healthcare needs, this project aims to contribute to advancements in both biometric security and medical diagnostics, ultimately benefiting society by improving safety, health monitoring, and interactive technology interfaces.

In conclusion, the integration of OpenCV and Mediapipe in this project exemplifies a practical and effective approach to iris segmentation. Its applications span across multiple domains, from enhancing security measures to supporting medical diagnoses and advancing human-computer interaction. As technology continues to evolve, further innovations in machine learning and interdisciplinary research hold promise for expanding the capabilities of iris segmentation systems, paving the way for more sophisticated and impactful applications in the future.

CHAPTER-7

REFERENCES

- Donald D Hearn, M Pauline Baker and WarrenCarithers: Computer Graphics with OpenGL 4th Edition, Pearson, 2014
- S. Sridhar, Digital Image Processing, second edition, Oxford University press 2016.
- Edward Angel: Interactive Computer Graphics- A Top Down approach with OpenGL, 5th edition. Pearson Education, 2008
- James D Foley, Andries Van Dam, Steven K Feiner, John F Huges Computer graphics with OpenGL: Pearson education
- https://www.geeksforgeeks.org/introduction-to-computer-graphics/
- https://www.geeksforgeeks.org/opencv-python-tutorial/
- https://nptel.ac.in/courses/106/106/106106090/
- https://nptel.ac.in/courses/106/102/106102063/
- https://nptel.ac.in/courses/106/103/106103224/
- https://nptel.ac.in/courses/106/102/106102065/
- https://www.tutorialspoint.com/opencv/