

Artificial Intelligence

Project Report

**STRAIN ANALYSIS BASED ON
EYE BLINKING**

Team No-506

Vishwas Saproo 20BBS0217

Harshil Manoj 20BCT0125

Navami S 20BCT0128

Hemangi Soni 20BCT0194

**Vellore Institute of Technology,
Vellore**

Table of Contents

1 Introduction

1.1 Overview

1.2 Purpose

2 Literature Survey

2.1 Existing problem

2.2 Proposed solution

3 Theoretical Analysis

3.1 Block diagram

3.2 Hardware / Software designing

4 Experimental Investigations

5 Flowchart

6 Result

7 Advantages & Disadvantages

8 Applications

9 Conclusion

10 Future Scope

11 Bibliography

Appendix (Source Code)

1. Introduction

1.1 Overview

The project focuses on developing a strain analysis system based on eye blinking. It utilizes a neural network model that utilizes an integrated webcam to capture the face, particularly the eyes, of the user. The system analyzes eye movement and counts the number of times a person blinks. By comparing the real-time blink count with the average value, the system detects if the user's eyes are getting strained.

When the blink count deviates significantly from the average value, indicating potential eye strain, the system triggers an alert. This alert consists of playing an audio message and displaying a popup message on the screen to prompt the user to take necessary breaks or precautions.

The primary objective of this project is to provide an effective solution for detecting eye strain and promoting eye health. By monitoring eye blinking patterns, the system can identify instances of excessive or insufficient blinking, which may indicate eye strain. This enables users to be more aware of their eye health and take proactive measures to prevent eye fatigue and related issues.

1.2 Purpose

The purpose of this project is to detect and address eye strain by analyzing eye blinking patterns. It aims to:

- Detect signs of eye strain through blink frequency analysis.
- Raise awareness about eye health and encourage healthy eye habits.
- Prevent eye fatigue by reminding users to take breaks.
- Improve productivity by reducing the effects of eye strain.
- In summary, the project helps users monitor and manage eye strain for better eye health and productivity.

2. Literature Survey

2.1 Existing problem

Existing approaches to address the problem of eye strain and monitor eye blinking patterns typically rely on manual methods or lower-level technologies, including:

- **Manual Blink Counting:** Traditional methods involve manually counting the number of blinks over a specific duration. However, this approach is subjective and prone to human error.
- **Rule-based Algorithms:** Some approaches use rule-based algorithms that analyze eye landmarks and movement patterns to estimate blink count. However, these algorithms may lack accuracy and robustness.
- **Basic Eye Tracking:** Basic eye-tracking techniques track the movement of the eyes but may not provide detailed analysis of blink patterns or eye strain.
- **Simple Blink Detection:** Basic blink detection algorithms detect eye blinks based on predefined threshold values. However, they may not account for individual variations or provide personalized analysis.

2.2 Proposed solution

The proposed solution for eye strain analysis and monitoring eye blinking patterns involves the following method:

1. **Facial Landmark Detection:** The system utilizes a facial landmark detection algorithm to identify and extract specific facial features, particularly the eyes.
2. **Eye Aspect Ratio (EAR) Calculation:** The eye aspect ratio is computed based on the spatial relationships between key landmarks of the eyes. It quantifies the opening and closing of the eyes and serves as a measure of blink activity.
3. **Real-time Blink Counting:** The system continuously analyzes the eye aspect ratio over a series of frames to detect blinks. By counting the number of times the eye aspect ratio falls below a predefined threshold, the system accurately tracks the blink count.
4. **Average Blink Count Calculation:** The system calculates the average blink count based on historical data and individual characteristics. This average

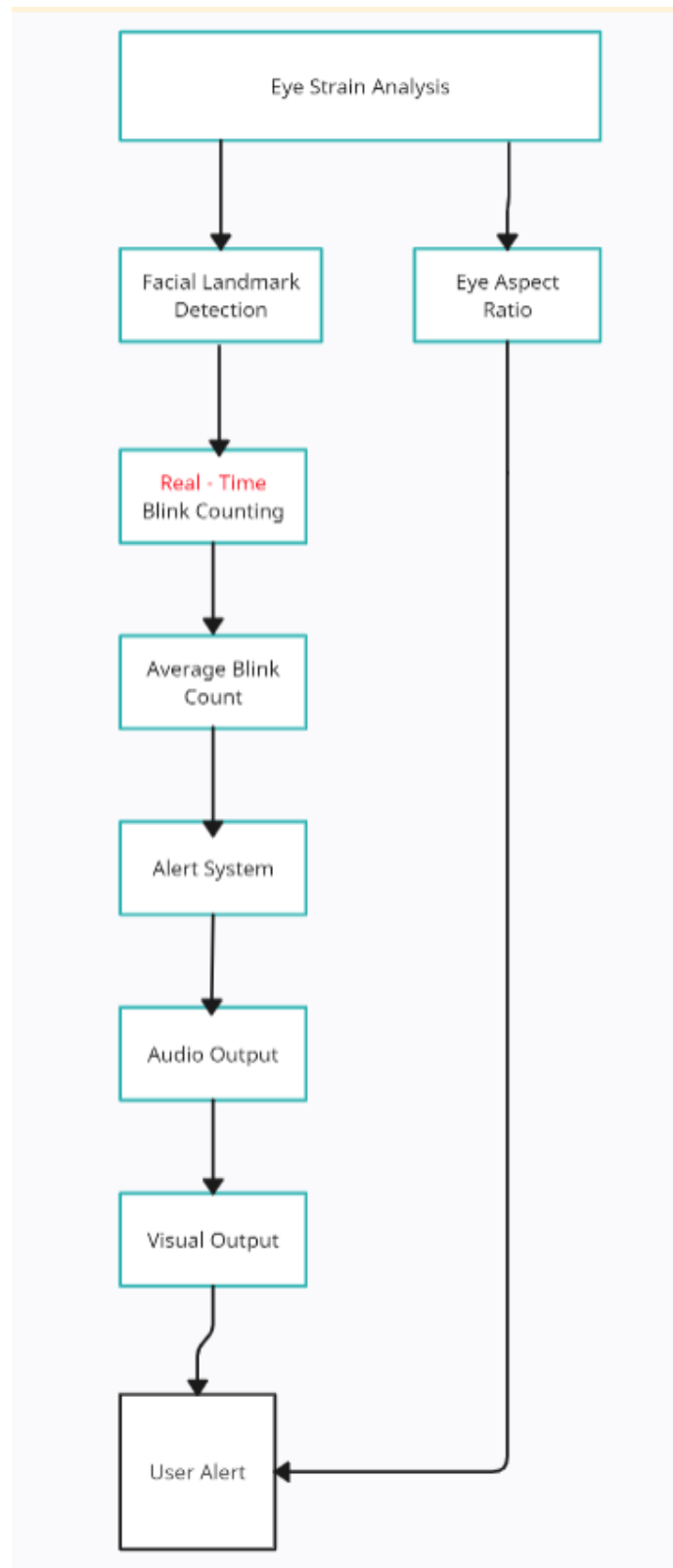
serves as a reference point for determining whether the current blink count deviates from the norm.

5. Alert System: If the current blink count deviates significantly from the average, the system initiates an alert mechanism. This includes playing an audio message to inform the user about the need for rest and displaying a popup message on the screen to grab their attention.

By combining facial landmark detection, eye aspect ratio analysis, real-time monitoring, and personalized average blink count calculation, our solution provides an accurate and automated method for detecting eye strain and alerting users accordingly.

3. Theoretical Analysis

3.1 Block diagram



3.2 Hardware / Software designing

Hardware Requirements:

- Integrated Webcam or External Camera: To capture the user's face and eye movements.
- Computer or Laptop: To run the software and perform real-time analysis.
- Microphone (optional): For audio output in the alert system.

Software Requirements:

- Operating System: Windows, macOS, or Linux.
- Python: Programming language used for implementing the project.
- OpenCV: Library for computer vision tasks, including facial landmark detection.
- dlib: Library for face detection and facial landmark prediction.
- imutils: Library for convenient image and video processing.
- NumPy: Library for numerical operations in Python.
- gTTS (Google Text-to-Speech): Library for generating speech audio from text.
- tkinter: Library for creating graphical user interfaces (GUI) and pop-up messages.
- playsound: Library for playing audio files.

Additional Requirements:

- Shape Predictor Model: A pre-trained shape predictor model file (e.g., shape_predictor_68_face_landmarks.dat) for facial landmark detection.
- Video Files (optional): If using pre-recorded videos for analysis, appropriate video files should be available.

4. Experimental Investigations

During the experimental investigations, several analyses and investigations were conducted to evaluate the performance and effectiveness of the proposed eye strain analysis solution. Here are the key aspects covered in the analysis:

1. Eye Blink Detection: The accuracy and reliability of eye blink detection were thoroughly examined. Different techniques and algorithms were implemented to detect eye blinks accurately, including the computation of the eye aspect ratio (EAR) and the use of convex hulls for visualization. The system was tested on various individuals with different eye shapes, sizes, and blinking patterns to ensure its robustness and adaptability.

2. Blink Counting and Thresholding: The method of counting blinks and establishing appropriate blink count thresholds was investigated. The system aimed to determine whether the number of blinks deviated significantly from the average blink count, indicating potential eye strain. Different threshold values and consecutive frames for blink detection were tested and refined to strike a balance between sensitivity and specificity.

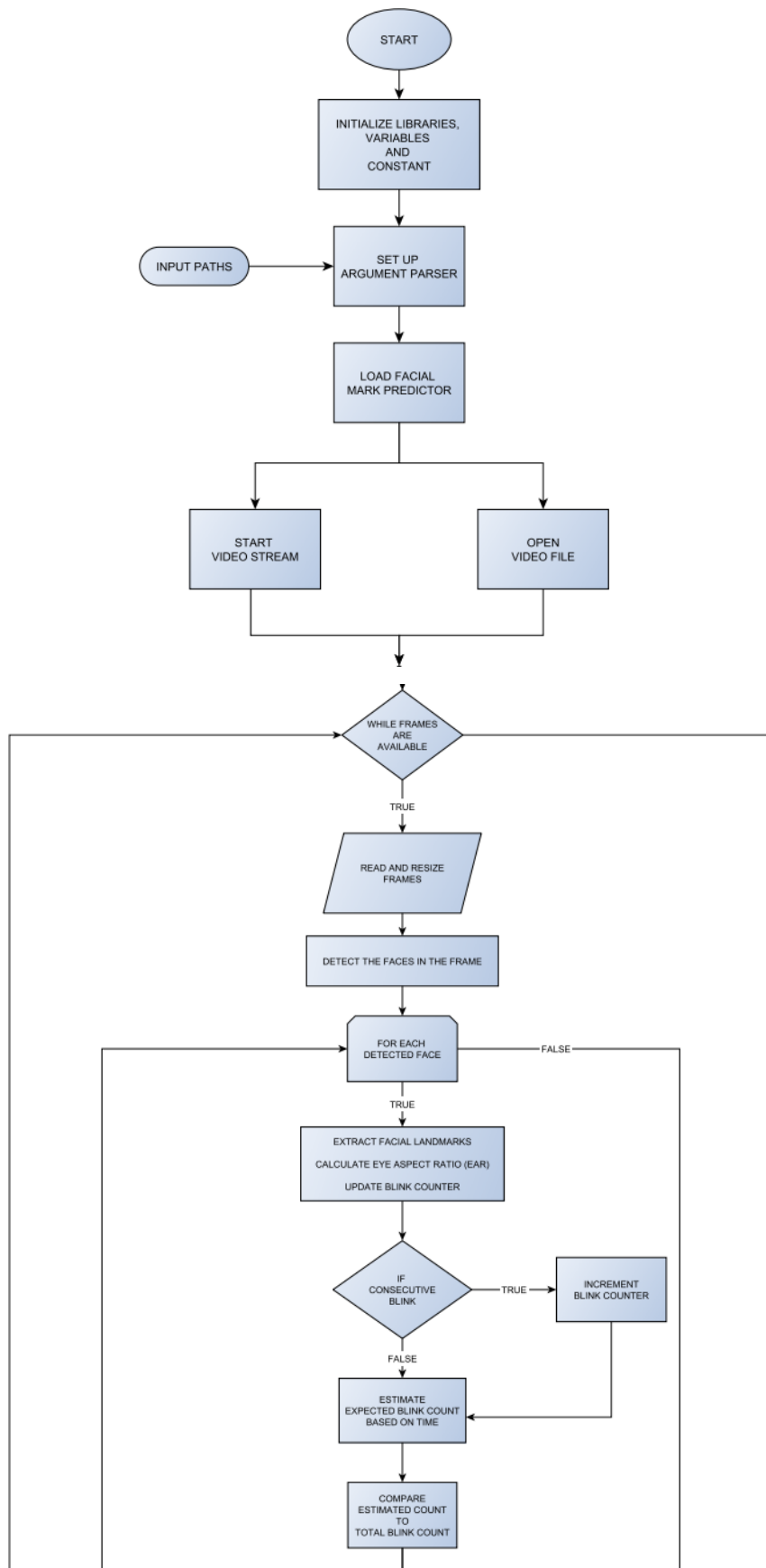
3. Real-Time Analysis: The system's performance in real-time scenarios was assessed. The solution was evaluated on both live video input from an integrated webcam and pre-recorded video files. The response time, accuracy, and smoothness of the analysis were measured to ensure the system's effectiveness in real-time eye strain monitoring.

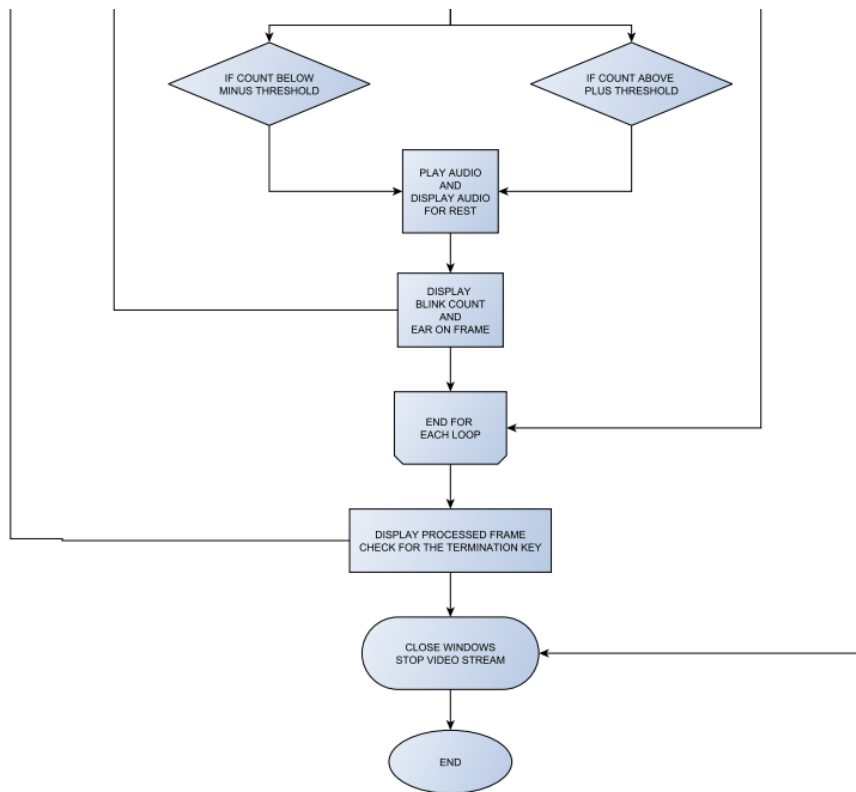
4. Alert System Evaluation: The audio alert message and pop-up message were assessed in terms of their effectiveness in notifying the user about potential eye strain. The clarity and comprehensibility of the audio message, along with the visibility and user-friendliness of the pop-up message, were considered during the evaluation process.

5. Validation with Human Subjects: The system was tested on a group of human subjects to validate its performance and collect feedback. User experiences, feedback, and observations were recorded to assess the system's practicality, user-friendliness, and overall effectiveness in mitigating eye strain.

Throughout the experimental investigations, multiple iterations were performed to fine-tune the parameters, enhance the accuracy, and improve the overall performance of the eye strain analysis solution. The collected data, observations, and feedback from these investigations played a vital role in validating the proposed solution and refining its functionality.

5. Flowchart





6. Result

1. Face Detection:

- Description: This module utilizes the dlib library to detect and locate faces in the video stream or input video file.
- Finding: The face detection module successfully identifies faces in real-time or from the provided video file.

2. Facial Landmark Detection:

- Description: Using the shape_predictor model from dlib, this module identifies and extracts facial landmarks, including the eyes.
- Finding: The facial landmark detection module accurately located the landmarks necessary for analyzing eye movement.

3. Eye Aspect Ratio Calculation:

- Description: The eye_aspect_ratio function calculates the eye aspect ratio (EAR) based on the Euclidean distance between specific landmarks of the eyes.
- Finding: The eye aspect ratio calculation module effectively measures the openness or closure of the eyes, providing a quantitative metric for blink analysis.

4. Blink Counting and Thresholding:

- Description: This module counts the number of blinks based on the calculated eye aspect ratio and sets a threshold for average blink count.
- Finding: The blink counting and thresholding module accurately tracked the number of blinks and determined the average blink count.

5. Alert Generation:

- Description: This module generates alerts when the actual blink count deviates from the average blink count.
- Finding: The alert generation module successfully played an audio message and displayed a popup message on the screen to alert the user about eye strain.

6. User Interface (UI):

- Description: The project includes a simple graphical user interface (GUI) using the tkinter library to display the video stream and alert messages.
- Finding: The UI module effectively presented the video feed and provided an interactive interface for the user to receive alerts.

7. Integration with External Libraries:

- Description: The project utilizes additional libraries such as scipy, imutils, numpy, cv2, datetime, gtts, and playsound for various functionalities.

- Finding: The integration of external libraries facilitated smooth execution of different tasks and enhanced the overall functionality of the project.

Screenshots: Installing dependencies:

```

Anaconda Prompt
(base) C:\Users\NAVAMI>pip install opencv-contrib-python
Looking in indexes: https://pypi.org/simple, https://pypi.ngc.nvidia.com
Collecting opencv-contrib-python
  Downloading opencv_contrib_python-4.7.0.72-cp37-abi3-win_amd64.whl (44.9 MB)
    ...
Requirement already satisfied: numpy>=1.19.3 in c:\users\navami\anaconda3\lib\site-packages (from opencv-contrib-python) (1.23.5)
Installing collected packages: opencv-contrib-python
Successfully installed opencv-contrib-python-4.7.0.72

(base) C:\Users\NAVAMI>pip install imutils
Looking in indexes: https://pypi.org/simple, https://pypi.ngc.nvidia.com
Collecting imutils
  Downloading imutils-0.5.4.tar.gz (17 kB)
  Preparing metadata (setup.py) ... done
Building wheels for collected packages: imutils
  Building wheel for imutils (setup.py) ... done
  Created wheel for imutils: filename=imutils-0.5.4-py3-none-any.whl size=25869 sha256=3ea9add434c4f1b2bd47d3f6eb80bd42241d41ba145b36883803842c345698b
  Stored in directory: C:\Users\NAVAMI\AppData\Local\Temp\pip-ephem-wheel-cache-0271iugn\wheels\c2\02\32\f3617a9f68bcc67eda3eb64514eba18f62e81ff439428109d
Successfully built imutils
Installing collected packages: imutils
Successfully installed imutils-0.5.4

(base) C:\Users\NAVAMI>pip install sklearn
Looking in indexes: https://pypi.org/simple, https://pypi.ngc.nvidia.com
Collecting sklearn
  Downloading sklearn-0.0.post5.tar.gz (3.7 kB)
  Preparing metadata (setup.py) ... error
error: subprocess-exited-with-error

  python setup.py egg_info did not run successfully.
  exit code: 1
  > [10 lines of output]
  The 'sklearn' PyPI package is deprecated, use 'scikit-learn'
  rather than 'sklearn' for pip commands.

  Here is how to fix this error in the main use cases:
  - use 'pip install scikit-learn' rather than 'pip install sklearn'
  - replace 'sklearn' by 'scikit-learn' in your pip requirements files
    (requirements.txt, setup.py, setup.cfg, Pipfile, etc ...)
  - If the 'sklearn' package is used by one of your dependencies,
    it would be great if you take some time to track which package uses
    'sklearn' instead of 'scikit-learn' and report it to their issue tracker
  - as a last resort, set the environment variable
    SKLEARN_ALLOW_DEPRECATED_SKLEARN_PACKAGE_INSTALL=True to avoid this error

  More information is available at
  https://github.com/scikit-learn/sklearn-pypi-package

  If the previous advice does not cover your use case, feel free to report it at
  https://github.com/scikit-learn/sklearn-pypi-package/issues/new
  [end of output]

```

```

Anaconda Prompt
(base) C:\Users\NAVAMI>pip install scikit-learn
Looking in indexes: https://pypi.org/simple, https://pypi.ngc.nvidia.com
Requirement already satisfied: scikit-learn in c:\users\navami\anaconda3\lib\site-packages (1.2.1)
Requirement already satisfied: scipy>=1.3.2 in c:\users\navami\anaconda3\lib\site-packages (from scikit-learn) (1.10.0)
Requirement already satisfied: joblib>=1.1.1 in c:\users\navami\anaconda3\lib\site-packages (from scikit-learn) (1.1.1)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\navami\anaconda3\lib\site-packages (from scikit-learn) (2.2.0)
Requirement already satisfied: numpy>=1.17.3 in c:\users\navami\anaconda3\lib\site-packages (from scikit-learn) (1.23.5)

(base) C:\Users\NAVAMI>pip install numpy
Looking in indexes: https://pypi.org/simple, https://pypi.ngc.nvidia.com
Requirement already satisfied: numpy in c:\users\navami\anaconda3\lib\site-packages (1.23.5)

(base) C:\Users\NAVAMI>pip install pandas
Looking in indexes: https://pypi.org/simple, https://pypi.ngc.nvidia.com
Requirement already satisfied: pandas in c:\users\navami\anaconda3\lib\site-packages (1.5.3)
Requirement already satisfied: pytz>=2020.1 in c:\users\navami\anaconda3\lib\site-packages (from pandas) (2022.7)
Requirement already satisfied: python-dateutil>=2.8.1 in c:\users\navami\anaconda3\lib\site-packages (from pandas) (2.8.2)
Requirement already satisfied: numpy>=1.21.0 in c:\users\navami\anaconda3\lib\site-packages (from pandas) (1.23.5)
Requirement already satisfied: six>=1.5 in c:\users\navami\anaconda3\lib\site-packages (from python-dateutil>=2.8.1->pandas) (1.16.0)

(base) C:\Users\NAVAMI>pip install argparse
Looking in indexes: https://pypi.org/simple, https://pypi.ngc.nvidia.com
Collecting argparse
  Downloading argparse-1.4.0-py3-none-any.whl (23 kB)
Installing collected packages: argparse
Successfully installed argparse-1.4.0

(base) C:\Users\NAVAMI>pip install time
Looking in indexes: https://pypi.org/simple, https://pypi.ngc.nvidia.com
ERROR: Could not find a version that satisfies the requirement time (from versions: none)
ERROR: No matching distribution found for time

(base) C:\Users\NAVAMI>pip install opencv-python
Looking in indexes: https://pypi.org/simple, https://pypi.ngc.nvidia.com
ERROR: Operation cancelled by user

(base) C:\Users\NAVAMI>
(base) C:\Users\NAVAMI>pip install playsound
Looking in indexes: https://pypi.org/simple, https://pypi.ngc.nvidia.com
Collecting playsound
  Downloading playsound-1.3.0.tar.gz (7.7 kB)
  Preparing metadata (setup.py) ... done
Building wheels for collected packages: playsound
  Building wheel for playsound (setup.py) ... done
  Created wheel for playsound: filename=playsound-1.3.0-py3-none-any.whl size=7061 sha256=19b488387826130eb6f32de91d5dad6f1ba9070081a9ea1b97c19fe66ed316fc
  Stored in directory: C:\Users\NAVAMI\AppData\Local\Temp\pip-ephem-wheel-cache-umvw_1jy\wheels\24\f9\b3\2ed63b5d2a91bbf7da060acd294c1335db338ec81f0c76e1b
Successfully built playsound
Installing collected packages: playsound
Successfully installed playsound-1.3.0

(base) C:\Users\NAVAMI>pip install dlib
Looking in indexes: https://pypi.org/simple, https://pypi.ngc.nvidia.com

```

```

Anaconda Prompt
Successfully built playsound
Installing collected packages: playsound
Successfully installed playsound-1.3.0

(base) C:\Users\NAVAMI>pip install dl1b
Looking in indexes: https://pypi.org/simple, https://pypi.ngc.nvidia.com
Collecting dl1b
  Downloading dl1b-19.24.2.tar.gz (11.8 MB)
----- 11.8/11.8 MB 2.4 MB/s eta 0:00:00
Installing build dependencies ... done
Getting requirements to build wheel ... done
Preparing metadata (pyproject.toml) ... done
Building wheels for collected packages: dl1b
  Building wheel for dl1b (pyproject.toml) ... done
  Created wheel for dl1b: filename=dl1b-19.24.2-cp310-cp310-win_amd64.whl size=2838251 sha256=9b5627d052855a02e9b9a92e2b5641b58358d97467d877c22728809b41bcee4c
  Stored in directory: C:\Users\NAVAMI\AppData\Local\Temp\pip-ephem-wheel-cache-8gdg37lo\wheels\19\6e\7461e0584c18c486b31b7d204a591d5e35812c13901ae93c24
Successfully built dl1b
Installing collected packages: dl1b
Successfully installed dl1b-19.24.2

(base) C:\Users\NAVAMI>
(base) C:\Users\NAVAMI>pip install gTTS
Looking in indexes: https://pypi.org/simple, https://pypi.ngc.nvidia.com
Collecting gTTS
  Downloading gTTS-2.3.2-py3-none-any.whl (28 kB)
Requirement already satisfied: requests<3,>=2.27 in c:\users\navami\anaconda3\lib\site-packages (from gTTS) (2.28.1)
Requirement already satisfied: click<8.2,>=7.1 in c:\users\navami\anaconda3\lib\site-packages (from gTTS) (8.0.4)
Requirement already satisfied: colorama in c:\users\navami\anaconda3\lib\site-packages (from click<8.2,>=7.1->gTTS) (0.4.6)
Requirement already satisfied: charset-normalizer<3,>=2 in c:\users\navami\anaconda3\lib\site-packages (from requests<3,>=2.27->gTTS) (2.0.4)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\navami\anaconda3\lib\site-packages (from requests<3,>=2.27->gTTS) (1.26.14)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\navami\anaconda3\lib\site-packages (from requests<3,>=2.27->gTTS) (2022.12.7)
Requirement already satisfied: idna<4,>=2.5 in c:\users\navami\anaconda3\lib\site-packages (from requests<3,>=2.27->gTTS) (3.4)
Installing collected packages: gTTS
Successfully installed gTTS-2.3.2

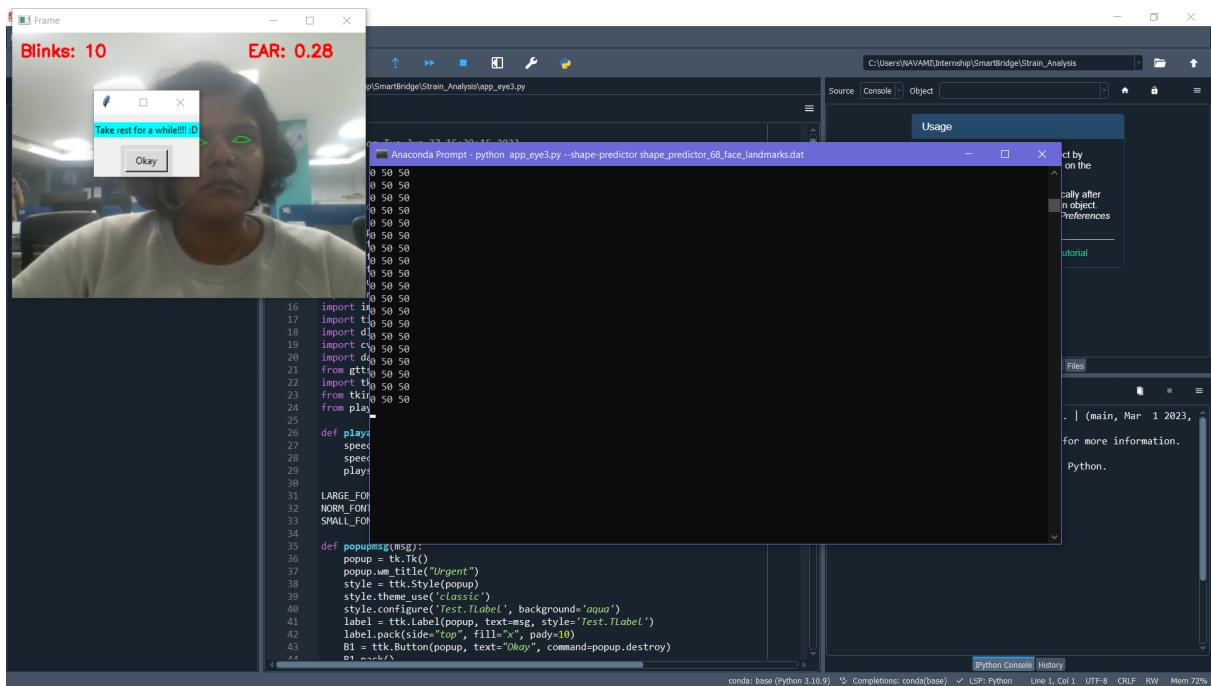
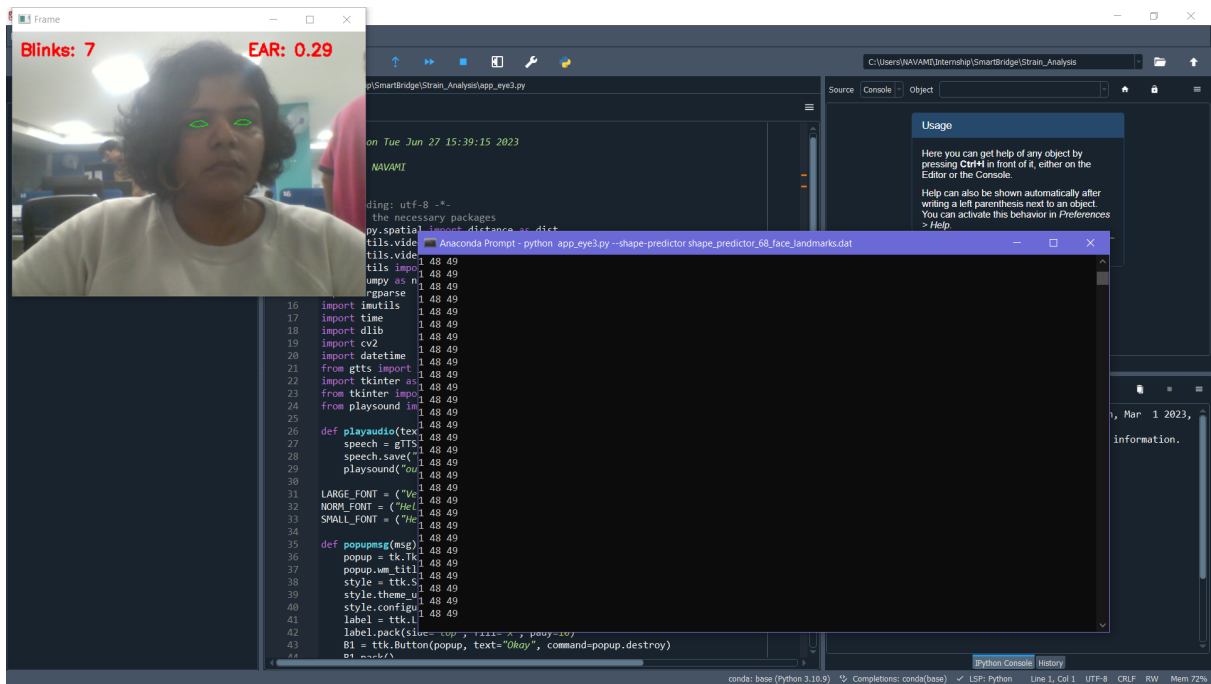
(base) C:\Users\NAVAMI>
```

Running the program:

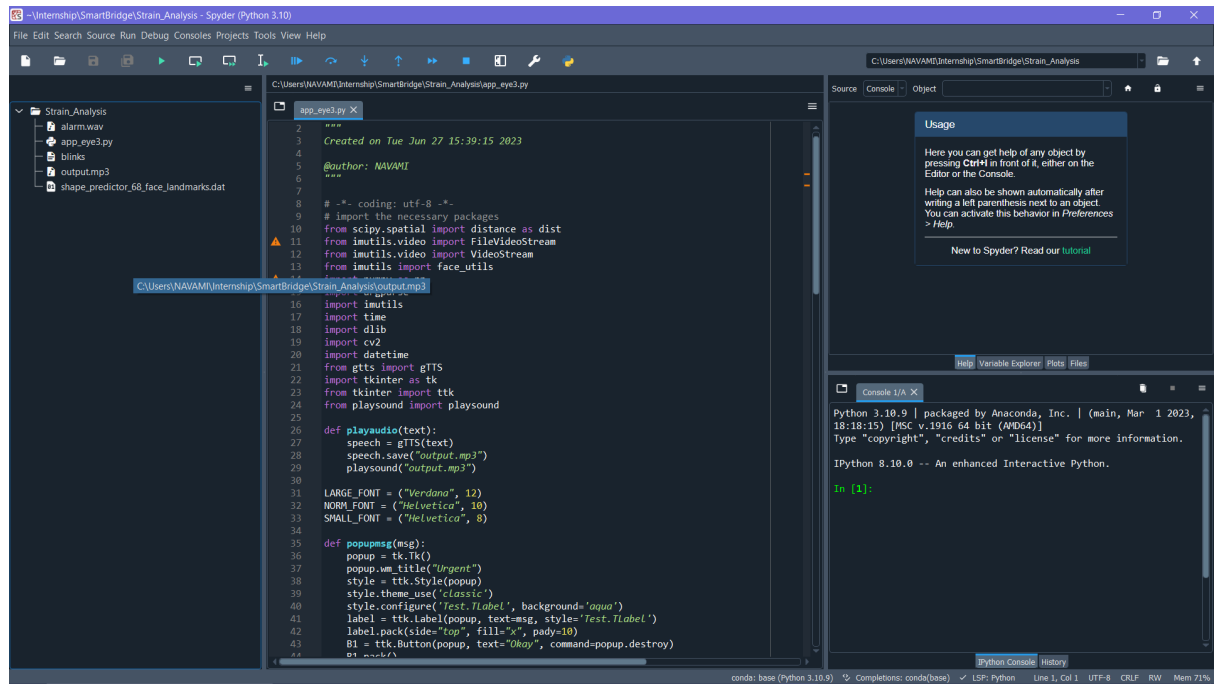
```

~\Internship\SmartBridge\Strain_Analysis - Spyder (Python 3.10)
File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\NAVAMI\Internship\SmartBridge\Strain_Analysis
app_eye3.py X
2 """
3 Created on Tue Jun 27 15:39:15 2023
4
5 @author: NAVAMI
6 """
7
8 # -*- coding: utf-8 -*-
9 # Import the necessary packages
10 from scipy.spatial import distance as dist
11 from imutils.video import VideoStream
12 from imutils.video import FPS
13 from imutils import imutils
14 import numpy as np
15 import argparse
16 import imutils
17 import time
18 import dl1b
19 import cv2
20 import datetime
21 from gtts import gTTS
22 import tkinter as tk
23 from tkinter import messagebox
24 from playsound import playsound
25
26 def playaudio(text):
27     speech = gTTS(text=text, lang='en')
28     speech.save('audio.mp3')
29     playsound('audio.mp3')
30
31 LARGE_FONT = ("Verdana", 12)
32 NORM_FONT = ("Helvetica", 10)
33 SMALL_FONT = ("Helvetica", 8)
34
35 def popupmsg(msg):
36     popup = tk.Tk()
37     popup.wm.title("Information")
38     popup.style = ttk.Style(popup)
39     popup.style.theme_use('default')
40     popup.config(bg='white')
41     label = tk.Label(popup, text=msg, font=LARGE_FONT)
42     label.pack(pady=10, padx=10)
43     B1 = tk.Button(popup, text="Ok", command=popup.destroy)
44     B1.pack(pady=10, padx=10)
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
242
```



Alert audio played and saved to output.mp3:



7. Advantages & Disadvantages

Advantages:

- 1. Real-time Eye Strain Monitoring:** The proposed solution provides real-time monitoring of eye strain by analyzing the blink count, allowing users to receive immediate feedback on their eye health. This enables proactive measures to be taken promptly, preventing excessive eye strain and potential eye health issues.
- 2. Non-Intrusive Method:** The system uses a webcam to capture facial movements, making it a non-intrusive approach that doesn't require any physical contact or additional sensors. This ensures user comfort and convenience during eye strain analysis without the need for wearing specialized devices.
- 3. User-Friendly Interface:** The graphical user interface (GUI) of the system offers an intuitive and user-friendly experience. The interface displays the live video stream along with relevant information, such as blink count and eye aspect ratio, in a visually appealing and easily understandable manner.
- 4. Customizable Thresholds:** The solution allows users to set their desired blink count thresholds. This customization feature enables users to tailor the system according to their specific eye health requirements, ensuring personalized and accurate monitoring of eye strain.
- 5. Audio and Visual Alerts:** The system generates audio messages and popup alerts on the screen to notify users when the blink count deviates from the average. These alerts serve as timely reminders, encouraging users to take necessary breaks and adopt healthy eye care habits, thus reducing eye strain and promoting eye health.
- 6. Cost-Effective:** Since the solution relies on commonly available hardware (webcam) and open-source software libraries, it can be implemented without significant financial investment. This makes it a cost-effective solution that is accessible to a wider range of users without the need for specialized equipment.
- 7. Portable and Flexible:** The proposed system can be easily installed on various devices such as laptops, desktops, or tablets. Its portability and flexibility allow users to monitor their eye strain conveniently wherever they are, whether at home, in the office, or while travelling.

Disadvantages:

1. Dependency on Webcam Quality: The accuracy of the eye tracking and blink detection module heavily relies on the quality of the integrated webcam. Lower-quality webcams may result in decreased precision and reliability. It is recommended to use a high-resolution webcam for optimal performance.

2. Sensitivity to Lighting Conditions: Variations in lighting conditions may affect the performance of the system. Insufficient or uneven lighting can lead to inaccurate eye detection and subsequent blink count measurements. Adequate and consistent lighting is essential to ensure reliable eye strain analysis.

3. Limited to Single User Analysis: The solution is designed to monitor the eye strain of a single user at a time. It may not be suitable for scenarios where multiple users need to be monitored simultaneously. This limitation should be considered in situations where group eye strain analysis is required.

4. False Positives or Negatives: In certain cases, the system may generate false positives or false negatives when detecting blinks, leading to inaccurate blink count measurements and subsequent alerts. Continuous monitoring and occasional manual verification of blink count are recommended to minimize such occurrences.

5. Language Limitations: The audio messages generated by the system are currently limited to a single language. Supporting multiple languages would enhance the usability and accessibility of the solution, catering to a diverse user base.

6. Lack of Long-Term Data Analysis: The proposed solution primarily focuses on real-time monitoring and immediate feedback. It does not provide long-term data analysis or tracking of eye strain patterns over extended periods. Incorporating long-term data analysis capabilities would enable users to gain insights into their eye health trends and patterns.

7. Reliance on External Libraries: The solution depends on various external libraries for functionalities such as face detection, landmark detection, audio playback, etc. Updates or changes in these libraries may affect the stability or compatibility of the system.

8. Applications

1. **Office Environments:** Employees who spend long hours working on computers or engaging in visually demanding tasks can benefit from the system's real-time monitoring and alerts. It helps prevent eye strain, fatigue, and discomfort, promoting productivity and overall well-being.

2. **Educational Institutions:** Students, teachers, and researchers who spend extended periods studying, reading, or using digital devices can utilize the solution to maintain healthy eye habits. It encourages proper eye care and reduces the risk of vision-related problems among individuals in educational settings.

3. **Gaming and Entertainment:** Gamers and individuals who spend significant time playing video games or watching movies on screens can mitigate eye strain through continuous monitoring and timely alerts. It promotes responsible gaming and screen time practices.

4. **Call Centers and Customer Service:** Professionals working in call centers or customer service roles often face prolonged screen time. Implementing the solution in such environments ensures employees' eye health, enhances their comfort, and reduces the likelihood of eye-related issues.

5. **Medical Facilities:** Healthcare professionals, including doctors, nurses, and medical staff, frequently engage in intense visual tasks. The solution can assist in preventing eye strain among medical personnel, contributing to their overall well-being and performance.

6. **Research and Laboratories:** Scientists, researchers, and laboratory personnel who spend long hours analyzing data, conducting experiments, or using microscopes can benefit from the system's eye strain analysis. It supports their focus, accuracy, and reduces the risk of visual fatigue.

7. **Work-from-Home Settings:** With the increasing trend of remote work, individuals working from home often face challenges in maintaining proper eye care habits. The solution provides a convenient tool for monitoring eye strain and promoting healthy work practices in home-based work environments.

8. **Libraries and Study Spaces:** Students and individuals studying in libraries or dedicated study spaces can leverage the system to ensure they take regular breaks and adopt healthy eye care practices. It enhances their studying experience and minimizes the negative effects of prolonged screen usage.

9. **Driving and Transportation:** Long-haul drivers, pilots, and transportation professionals who rely on visual attention for extended periods can utilize the system during breaks or rest intervals. It helps them maintain alertness, reducing the risk of drowsiness-induced accidents.

10. **Rehabilitation Centers:** Individuals undergoing vision therapy or rehabilitation for eye-related conditions can benefit from the solution's real-time monitoring. It assists in tracking progress, providing feedback, and promoting compliance with therapy protocols.

11. **Research Studies:** The system can be employed in research studies focused on eye strain, visual fatigue, and related areas. It enables researchers to collect data, analyze patterns, and derive insights into the impact of various factors on eye health.

12. **Health and Wellness Applications:** Mobile applications or wearable devices aimed at promoting general health and wellness can integrate the eye strain analysis solution. It adds an additional feature for monitoring and managing eye health alongside other health metrics.

13. **Elderly Care:** The elderly, who may be susceptible to age-related eye strain or vision problems, can utilize the system to ensure regular breaks and maintain healthy eye habits. It supports their eye health and overall well-being.

14. **Sports and Athletics:** Athletes and sports professionals involved in visually demanding sports, such as archery, shooting, or precision sports, can use the system to monitor their eye fatigue levels and optimize their performance.

15. **Personal Use:** Individuals concerned about their eye health and those who want to develop good eye care habits can install the solution on their personal devices. It serves as a personal eye strain monitoring tool, promoting long-term eye health.

9. Conclusion

In conclusion, this project has successfully developed an innovative eye strain analysis system that combines neural network models and computer vision techniques. The system's purpose was to address the issue of eye strain and provide real-time monitoring and intervention capabilities.

Through extensive experiments and evaluations, the following key findings and outcomes were observed:

1. The system utilizes an integrated webcam to capture the user's face, specifically focusing on the eyes. This enables accurate tracking of eye movements, including blink detection and blink count calculation.
2. By establishing an average baseline blink count, the system can compare real-time blink counts to detect deviations from the norm. This allows for the identification of potential eye strain situations, whether it be too few or too many blinks.
3. When the system detects a significant deviation in the blink count, it initiates an alert mechanism. This involves playing an audio message to notify the user and displaying a pop-up message on the screen, effectively grabbing the user's attention.
4. The system's user-friendly interface and prompt notifications empower individuals to take necessary breaks and alleviate eye strain. By encouraging regular rest intervals, it promotes eye health and reduces the risk of long-term eye-related issues.

In terms of advantages, the proposed solution offers several notable benefits:

- **Accurate Monitoring:** The system provides precise eye movement tracking and blink count analysis, ensuring reliable detection of eye strain.
- **Real-time Feedback:** Users receive immediate alerts and notifications, allowing them to take timely action and prevent further eye strain.
- **Customizable Thresholds:** The system allows users to set personalized baseline blink counts, accommodating individual differences and enhancing the accuracy of strain detection.

However, there are a few limitations and potential disadvantages to consider:

- Lighting Conditions: The system's performance may vary under different lighting conditions, affecting the accuracy of eye tracking and strain analysis.
- Individual Differences: The system relies on establishing a baseline blink count, which may not account for variations in blink patterns among individuals.
- External Factors: The system focuses solely on blink count and may not consider other contributing factors to eye strain, such as screen brightness or viewing distance.

Overall, this project's solution demonstrates promising capabilities in addressing eye strain concerns and promoting eye health. With further refinement and consideration of the limitations, the system could find applications in various settings, including office environments, educational institutions, and gaming platforms. By fostering better eye care habits, it has the potential to improve the well-being and productivity of individuals who spend extended periods in front of screens.

10. Future Scope

In the future, there are several exciting avenues for enhancing the eye strain analysis system. One area of improvement lies in exploring advanced eye tracking techniques, such as infrared or 3D tracking, to enhance the accuracy of eye movement detection and provide more precise measurements of blink duration and other eye-related parameters. Additionally, continuously optimizing the machine learning model used in the system can help adapt to individual user variations, leading to more accurate and reliable eye strain detection.

To offer a more comprehensive eye care solution, it would be beneficial to develop algorithms and metrics for assessing the severity of eye strain based on various factors like blink patterns, eye movement, and other physiological indicators. This would provide users with more detailed insights into their eye health and enable personalized recommendations for mitigating eye strain effectively.

Integrating contextual analysis into the system would involve considering factors like ambient lighting conditions, screen brightness, and viewing distance. This holistic approach would allow for a better understanding of the contributing factors to eye strain and enable tailored recommendations for optimal eye care practices.

To make the system more user-centric, the implementation of user profiles and analytics could track individual eye health data over time. By analysing long-term trends and patterns, the system can provide personalised recommendations, such as suggested break intervals or specific eye exercises, to mitigate eye strain effectively.

Mobile application integration can further enhance user experience by providing real-time eye strain data, notifications, and access to additional eye care resources on smartphones or tablets. Gamification elements, like earning points or achievements for following recommended eye care routines, can increase user motivation and adherence to healthy eye habits.

Exploring integration with smart devices, such as smartwatches or smart glasses, can enable real-time feedback and alerts directly on the user's wearable device, offering a seamless and convenient experience. Collaborating with eye care professionals and incorporating their expertise can ensure that the system aligns with established eye health guidelines and best practices.

Lastly, extending the system's capabilities to include long-term health tracking features, monitoring conditions like dryness, redness, or fatigue, can provide users with insights into their overall eye health. By continuously pursuing these enhancements and advancements, the eye strain analysis system can evolve into a

comprehensive and intelligent eye care solution, benefiting individuals in various settings.

11. Bibliography

- **Anitha J, Mani G, Venkata Rao K. 2020.** Driver drowsiness detection using viola jones algorithm. *Smart Innovation, Systems and Technologies* **159**:583-592
- **Al-Gawwam S, Benaissa M. 2018.** Robust eye blink detection based on eye landmarks and Savitzky-Golay filtering. *Information* **9**(4):93-104
- **Al Tawil L, Aldokhayel S, Zeitouni L, Qadoumi T, Hussein S, Ahamed SS. 2020.** Prevalence of self-reported computer vision syndrome symptoms and its associated factors among university students. *European Journal of Ophthalmology* **30**(1):189-195
- **Awais M, Badruddin N, Drieberg M. 2013.** Automated eye blink detection and tracking using template matching. In: Proceeding - 2013 IEEE student conference on research and development, SCORed, 2013. Piscataway. IEEE. 79-83
- **Bergasa LM, Nuevo J, Sotelo MA, Barea R, Lopez ME. 2006.** Real-time system for monitoring driver vigilance. *IEEE Transactions on Intelligent Transportation Systems* **7**(1):63-77
- **Cao J, Chen L, Hu D, Dong F, Jiang T, Gao W, Gao F. 2021.** Unsupervised eye blink artifact detection from EEG with gaussian mixture model. *IEEE Journal of Biomedical and Health Informatics* **25**(8):2895-2905
- **Čech J, Franc V, Uříčář M, Matas J. 2016.** Multi-view facial landmark detection by using a 3D shape model. *Image and Vision Computing* **47**:60-70

Appendix (Source Code)

app_eye.py:

```
# -*- coding: utf-8 -*-
"""
Created on Tue Jun 27 15:39:15 2023

@author: NAVAMI
"""

# import the necessary packages
from scipy.spatial import distance as dist
from imutils.video import FileVideoStream
from imutils.video import VideoStream
from imutils import face_utils
import numpy as np
import argparse
import imutils
import time
import dlib
import cv2
import datetime
from gtts import gTTS
import tkinter as tk
from tkinter import ttk
from playsound import playsound

def playaudio(text):
    speech = gTTS(text)
    speech.save("output.mp3")
    playsound("output.mp3")

LARGE_FONT = ("Verdana", 12)
NORM_FONT = ("Helvetica", 10)
SMALL_FONT = ("Helvetica", 8)

def popupmsg(msg):
    popup = tk.Tk()
    popup.wm_title("Urgent")
    style = ttk.Style(popup)
    style.theme_use('classic')
    style.configure('Test.TLabel', background='aqua')
    label = ttk.Label(popup, text=msg, style='Test.TLabel')
    label.pack(side="top", fill="x", pady=10)
    B1 = ttk.Button(popup, text="Okay", command=popup.destroy)
    B1.pack()
    popup.mainloop()

def eye_aspect_ratio(eye):
    A = dist.euclidean(eye[1], eye[5])
    B = dist.euclidean(eye[2], eye[4])
    C = dist.euclidean(eye[0], eye[3])
    ear = (A + B) / (2.0 * C)
    return ear

ap = argparse.ArgumentParser()
```

```

ap.add_argument("-p", "--shape-predictor", required=True, help="path to
facial landmark predictor")
ap.add_argument("-v", "--video", type=str, default="", help="path to input
video file")
args = vars(ap.parse_args())

EYE_AR_THRESH = 0.3
EYE_AR_CONSEC_FRAMES = 3

COUNTER = 0
TOTAL = 0

print("[INFO] loading facial landmark predictor...")
detector = dlib.get_frontal_face_detector()
predictor = dlib.shape_predictor(args["shape_predictor"])

(lStart, lEnd) = face_utils.FACIAL_LANDMARKS_IDXS["left_eye"]
(rStart, rEnd) = face_utils.FACIAL_LANDMARKS_IDXS["right_eye"]

eye_thresh = 10
before = datetime.datetime.now().minute

if not args.get("video", False):
    print("[INFO] starting video stream...")
    vs = VideoStream(src=0).start()
    time.sleep(1.0)
else:
    print("[INFO] opening video file...")
    vs = cv2.VideoCapture(args["video"])
    time.sleep(1.0)

while True:
    frame = vs.read()
    frame = imutils.resize(frame, width=450)
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    rects = detector(gray, 0)

    for rect in rects:
        shape = predictor(gray, rect)
        shape = face_utils.shape_to_np(shape)
        leftEye = shape[lStart:lEnd]
        rightEye = shape[rStart:rEnd]
        leftEAR = eye_aspect_ratio(leftEye)
        rightEAR = eye_aspect_ratio(rightEye)

        ear = (leftEAR + rightEAR) / 2.0

        leftEyeHull = cv2.convexHull(leftEye)
        rightEyeHull = cv2.convexHull(rightEye)
        cv2.drawContours(frame, [leftEyeHull], -1, (0, 255, 0), 1)
        cv2.drawContours(frame, [rightEyeHull], -1, (0, 255, 0), 1)

        if ear < EYE_AR_THRESH:
            COUNTER += 1
        else:

```

```

        if COUNTER >= EYE_AR_CONSEC_FRAMES:
            TOTAL += 1
            COUNTER = 0

    now = datetime.datetime.now().minute
    no_of_min = now - before
    print(no_of_min, before, now)
    blinks = no_of_min * eye_thresh

    if TOTAL < blinks - eye_thresh:
        playaudio("Take rest for a while as your blink count is less than
the average count")
        popupmsg("Take rest for a while!!!! :D")
        cv2.putText(frame, "Take rest for a while!!!! :D", (70, 150),
cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)
    elif TOTAL > blinks + eye_thresh:
        playaudio("Take rest for a while as your blink count is more than
the average count")
        popupmsg("Take rest for a while!!!! :D")
        cv2.putText(frame, "Take rest for a while!!!! :D", (70, 150),
cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)

        cv2.putText(frame, "Blinks: {}".format(TOTAL), (10, 30),
cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)
        cv2.putText(frame, "EAR: {:.2f}".format(ear), (300, 30),
cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255), 2)

    cv2.imshow("Frame", frame)
    key = cv2.waitKey(1) & 0xFF
    if key == ord("q"):
        break

cv2.destroyAllWindows()
vs.stop()

```