







## EYE DETECTION LOGIC



Image □ Input to Detection Pipeline

Python Code □ `eye_cascade.detectMultiScale(roi_gray, 1.1, 10)`

C Code □ `cvHaarDetectObjects(eye ROI, cascade, ...)`

Assembly □ SCAN ROI -> CMP brightness -> JMP if shape

Machine Code □ 3C 5F | 7E 04 | EB 02 ...

Output □ Detected eye: ex, ey, ew, eh



YENİ ERKEK ÖĞRENCİ YURDU

1. KAT	2. KAT	3. KAT
4. KAT	5. KAT	6. KAT
7. KAT	8. KAT	9. KAT
10. KAT	11. KAT	12. KAT
13. KAT	14. KAT	15. KAT
16. KAT	17. KAT	18. KAT
19. KAT	20. KAT	21. KAT
22. KAT	23. KAT	24. KAT
25. KAT	26. KAT	27. KAT
28. KAT	29. KAT	30. KAT



## TİM ERKEK ÖĞRENCİ YURDU

1. 100	2. 100	3. 100
4. 100	5. 100	6. 100
7. 100	8. 100	9. 100
10. 100	11. 100	12. 100
13. 100	14. 100	15. 100
16. 100	17. 100	18. 100
19. 100	20. 100	21. 100
22. 100	23. 100	24. 100
25. 100	26. 100	27. 100
28. 100	29. 100	30. 100
31. 100	32. 100	33. 100
34. 100	35. 100	36. 100
37. 100	38. 100	39. 100
40. 100	41. 100	42. 100
43. 100	44. 100	45. 100
46. 100	47. 100	48. 100
49. 100	50. 100	51. 100
52. 100	53. 100	54. 100
55. 100	56. 100	57. 100
58. 100	59. 100	60. 100
61. 100	62. 100	63. 100
64. 100	65. 100	66. 100
67. 100	68. 100	69. 100
70. 100	71. 100	72. 100
73. 100	74. 100	75. 100
76. 100	77. 100	78. 100
79. 100	80. 100	81. 100
82. 100	83. 100	84. 100
85. 100	86. 100	87. 100
88. 100	89. 100	90. 100
91. 100	92. 100	93. 100
94. 100	95. 100	96. 100
97. 100	98. 100	99. 100
100. 100	101. 100	102. 100
103. 100	104. 100	105. 100
106. 100	107. 100	108. 100
109. 100	110. 100	111. 100
112. 100	113. 100	114. 100
115. 100	116. 100	117. 100
118. 100	119. 100	120. 100
121. 100	122. 100	123. 100
124. 100	125. 100	126. 100
127. 100	128. 100	129. 100
130. 100	131. 100	132. 100
133. 100	134. 100	135. 100
136. 100	137. 100	138. 100
139. 100	140. 100	141. 100
142. 100	143. 100	144. 100
145. 100	146. 100	147. 100
148. 100	149. 100	150. 100
151. 100	152. 100	153. 100
154. 100	155. 100	156. 100
157. 100	158. 100	159. 100
160. 100	161. 100	162. 100
163. 100	164. 100	165. 100
166. 100	167. 100	168. 100
169. 100	170. 100	171. 100
172. 100	173. 100	174. 100
175. 100	176. 100	177. 100
178. 100	179. 100	180. 100
181. 100	182. 100	183. 100
184. 100	185. 100	186. 100
187. 100	188. 100	189. 100
190. 100	191. 100	192. 100
193. 100	194. 100	195. 100
196. 100	197. 100	198. 100
199. 100	200. 100	201. 100
202. 100	203. 100	204. 100
205. 100	206. 100	207. 100
208. 100	209. 100	210. 100
211. 100	212. 100	213. 100
214. 100	215. 100	216. 100
217. 100	218. 100	219. 100
220. 100	221. 100	222. 100
223. 100	224. 100	225. 100
226. 100	227. 100	228. 100
229. 100	230. 100	231. 100
232. 100	233. 100	234. 100
235. 100	236. 100	237. 100
238. 100	239. 100	240. 100
241. 100	242. 100	243. 100
244. 100	245. 100	246. 100
247. 100	248. 100	249. 100
250. 100	251. 100	252. 100
253. 100	254. 100	255. 100
256. 100	257. 100	

## FACE DETECTION LOGIC



Image □ Input to Detection Pipeline

Python Code □ `face_cascade.detectMultiScale(gray, 1.3, 5)`

C Code □ `cvHaarDetectObjects(gray, cascade, ...)`

Assembly □ LOOP -> CMP pixel intensity -> JMP

Machine Code □ 3C 7F | 7F 05 | EB 03 ...

Output □ Detected face: x, y, width, height









## MOUTH DETECTION LOGIC

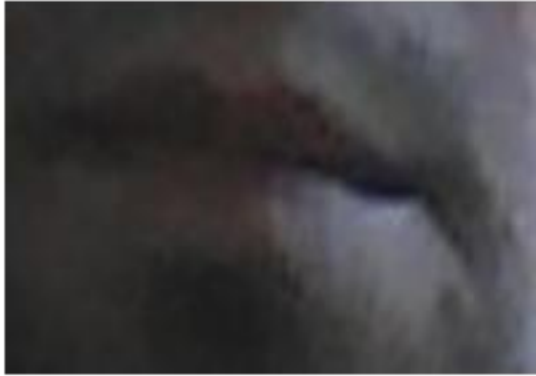


Image □ Input to Detection Pipeline

Python Code □ `mouth_cascade.detectMultiScale(roi_gray, 1.5, 11)`

C Code □ `cvHaarDetectObjects(mouth ROI, ...)`

Assembly □ LOOP over lower half -> CMP -> JMP

Machine Code □ `3C 60 | 7C 03 | EB 01 ...`

Output □ Detected mouth: mx, my, mw, mh