

Final Code

Date	19 November 2022
Team ID	PNT2022TMID40534
Project Name	Project – Real Time Communication System Powered by AI for Specially Abled

```
1  import cv2
2
3  video = cv2.VideoCapture(0)
4
5  while True:
6      ret, frame = video.read()
7      cv2.imshow("Frame", frame)
8      k = cv2.waitKey(1)
9      if k == ord('q'):
10         break
11
12  video.release()
13  cv2.destroyAllWindows()
```

```
1  import cv2
2  import numpy as np
3  from tensorflow.keras.models import load_model
4  from tensorflow.keras.preprocessing import image
5
6  class Video(object):
7      def __init__(self):
8          self.video = cv2.VideoCapture(0)
9          self.roi_start = (50, 150)
10         self.roi_end = (250, 350)
11         self.model = load_model('asl_model.h5') # Execute Local Trained Model
12         # self.model = load_model('IBM_Communication_Model.h5') # Execute IBM Trained Model
13         self.index=['A','B','C','D','E','F','G','H','I']
14         self.y = None
15
16     def __del__(self):
17         self.video.release()
18
19     def get_frame(self):
20         ret, frame = self.video.read()
21         frame = cv2.resize(frame, (640, 480))
22         copy = frame.copy()
23         copy = copy[150:150+200, 50:50+200]
24         # Prediction Start
25         cv2.imwrite('image.jpg', copy)
26         copy_img = image.load_img('image.jpg', target_size=(64,64))
27         x = image.img_to_array(copy_img)
28         x = np.expand_dims(x, axis=0)
29         pred = np.argmax(self.model.predict(x), axis=1)
30         self.y = pred[0]
31         cv2.putText(frame, 'The Predicted Alphabet is: '+str(self.index[self.y]), (100, 50), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 0), 3)
32         ret, jpg = cv2.imencode('.jpg', frame)
33         return jpg.tobytes()
```

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <meta name="viewport" content="width=device-width, initial-scale=1">
5  <style>
6  body {font-family: Arial, Helvetica, sans-serif;}
7
8  /* Full-width input fields */
9  input[type=text], input[type=password] {
10     width: 100%;
11     padding: 12px 20px;
12     margin: 8px 0;
13     display: inline-block;
14     border: 1px solid #ccc;
15     box-sizing: border-box;
16 }
17
18 /* Set a style for all buttons */
19 button {
20     background-color: #273298;
21     color: white;
22     padding: 14px 20px;
23     margin: 8px 0;
24     border: none;
25     cursor: pointer;
26     width: 100%;
27 }
28
29 button:hover {
30     opacity: 0.8;
31 }
```

```
33  /* Extra styles for the cancel button */
34  .cancelbtn {
35      width: auto;
36      padding: 10px 18px;
37      background-color: #f44336;
38  }
39
40  /* Center the image and position the close button */
41  .imgcontainer {
42      text-align: center;
43      margin: 24px 0 12px 0;
44      position: relative;
45  }
46
47  img.avatar {
48      width: 40%;
49      border-radius: 50%;
50  }
51
52  .container {
53      padding: 16px;
54  }
55
56  span.psw {
57      float: right;
58      padding-top: 16px;
59  }
60
61  /* The Modal (background) */
62  .modal {
63      display: none; /* Hidden by default */
64      position: fixed; /* Stay in place */
65      z-index: 1; /* Sit on top */
66      left: 0;
67      top: 0;
68      width: 100%; /* Full width */
69      height: 100%; /* Full height */
```

```
33  /* Extra styles for the cancel button */
34  .cancelbtn {
35    width: auto;
36    padding: 10px 18px;
37    background-color: #f44336;
38  }
39
40  /* Center the image and position the close button */
41  .imgcontainer {
42    text-align: center;
43    margin: 24px 0 12px 0;
44    position: relative;
45  }
46
47  img.avatar {
48    width: 40%;
49    border-radius: 50%;
50  }
51
52  .container {
53    padding: 16px;
54  }
55
56  span.psw {
57    float: right;
58    padding-top: 16px;
59  }
60
61  /* The Modal (background) */
62  .modal {
63    display: none; /* Hidden by default */
64    position: fixed; /* Stay in place */
65    z-index: 1; /* Sit on top */
66    left: 0;
67    top: 0;
68    width: 100%; /* Full width */
69    height: 100%; /* Full height */
```

```

106 @-webkit-keyframes animatezoom {
107   from {-webkit-transform: scale(0)}
108   to {-webkit-transform: scale(1)}
109 }
110
111 @keyframes animatezoom {
112   from {transform: scale(0)}
113   to {transform: scale(1)}
114 }
115
116 /* Change styles for span and cancel button on extra small screens */
117 @media screen and (max-width: 300px) {
118   span.psw {
119     display: block;
120     float: none;
121   }
122   .cancelbtn {
123     width: 100%;
124   }
125 }
126 </style>
127 </head>
128 <body>
129
130 <h2 style="text-align: center; color: rgba(1, 2, 2, 0.874);">REAL TIME COMMUNICATION SYSTEM POWERED BY AI FOR SPECIALLY ABLED</h2>
131
132 <button onclick="document.getElementById('id01').style.display='block'">Login</button>
133
134 <div id="id01" class="modal">
135
136   <form class="modal-content animate" action="/action_page.php" method="post">
137     <div class="imgcontainer">
138       <span onclick="document.getElementById('id01').style.display='none'" class="close" title="Close Modal">&times;</span>
139       
140     </div>
141

```

```

142     <div class="container">
143         <label for="uname"><b>Username</b></label>
144         <input type="text" placeholder="Enter Username" name="uname" required>
145
146         <label for="psw"><b>Password</b></label>
147         <input type="password" placeholder="Enter Password" name="psw" required>
148
149         <button type="submit">Login</button>
150         <label>
151             <input type="checkbox" checked="checked" name="remember"> Remember me
152         </label>
153     </div>
154
155     <div class="container" style="background-color: #f1f1f1">
156         <button type="button" onclick="document.getElementById('id01').style.display='none'" class="cancelbtn">Cancel</button>
157         <span class="psw">Forgot <a href="#">password?</a></span>
158     </div>
159 </form>
160 </div>
161 <!doctype html>
162 <html lang="en">
163 <head>
164     <meta charset="UTF-8">
165     <meta name="viewport"
166         content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">
167     <meta http-equiv="X-UA-Compatible" content="ie=edge">
168     <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css">
169     <link rel="stylesheet" href="style.css">
170     <title>Document</title>
171 </head>
172 <body>
173 <div class="display-cover">
174     <video autoplay></video>
175     <canvas class="d-none"></canvas>
176
177     <div class="video-options">
178         <select name="" id="" class="custom-select">

```

```

179         <option value="">Select camera</option>
180     </select>
181 </div>
182
183     <img class="screenshot-image d-none" alt="">
184
185     <div class="controls">
186         <button class="btn btn-danger play" title="Play"><i data-feather="play-circle"></i></button>
187         <button class="btn btn-info pause d-none" title="Pause"><i data-feather="pause"></i></button>
188         <button class="btn btn-outline-success screenshot d-none" title="ScreenShot"><i data-feather="image"></i></button>
189     </div>
190 </div>
191
192 <script src="https://unpkg.com/feather-icons"></script>
193 <script src="script.js"></script>
194 </body>
195 <html><head>
196 </head><body>
197     <video src="" ></video>
198     <br />
199 <button id='flipCamera'>Flip</button>
200 </body>
201 <script>
202     var front = false;
203     var video = document.querySelector('video');
204     document.getElementById('flipCamera').onclick = function() { front = !front; };
205     var constraints = { video: { facingMode: (front? "user" : "environment"), width: 640, height: 480 } };
206     navigator.mediaDevices.getUserMedia(constraints)
207     .then(function(mediaStream) {
208         video.srcObject = mediaStream;
209         video.onloadedmetadata = function(e) {
210             video.play();
211         };
212     })
213     .catch(function(err) { console.log(err.name + ": " + err.message); })
214 </script></html>
215 </html>

```



```
253     display: flex;
254 }
255
256 .controls > button {
257     width: 45px;
258     height: 45px;
259     text-align: center;
260     border-radius: 100%;
261     margin: 0 6px;
262     background: transparent;
263 }
264
265 .controls > button:hover svg {
266     color: white !important;
267 }
268
269 @media (min-width: 300px) and (max-width: 400px) {
270     .controls {
271         flex-direction: column;
272     }
273
274     .controls button {
275         margin: 5px 0 !important;
276     }
277 }
278
279 .controls > button > svg {
280     height: 20px;
281     width: 18px;
282     text-align: center;
283     margin: 0 auto;
284     padding: 0;
285 }
286
287 .controls button:nth-child(1) {
288     border: 2px solid #1a12b3;
289 }
```



```
216 <style>
217 .screenshot-image {
218     width: 150px;
219     height: 90px;
220     border-radius: 4px;
221     border: 2px solid whitesmoke;
222     box-shadow: 0 1px 2px 0 rgba(0, 0, 0, 0.1);
223     position: absolute;
224     bottom: 5px;
225     left: 10px;
226     background: white;
227 }
228
229 .display-cover {
230     display: flex;
231     justify-content: center;
232     align-items: center;
233     width: 70%;
234     margin: 5% auto;
235     position: relative;
236 }
237
238 video {
239     width: 100%;
240     background: rgba(0, 0, 0, 0.2);
241 }
242
243 .video-options {
244     position: absolute;
245     left: 20px;
246     top: 30px;
247 }
248
249 .controls {
250     position: absolute;
251     right: 20px;
252     top: 20px;
```

```
291 .controls button:nth-child(1) svg {
292   color: #2b128e;
293 }
294
295 .controls button:nth-child(2) {
296   border: 2px solid #008496;
297 }
298
299 .controls button:nth-child(2) svg {
300   color: #008496;
301 }
302
303 .controls button:nth-child(3) {
304   border: 2px solid #0048b5;
305 }
306
307 .controls button:nth-child(3) svg {
308   color: #0f0a5b;
309 }
310
311 .controls > button {
312   width: 45px;
313   height: 45px;
314   text-align: center;
315   border-radius: 100%;
316   margin: 0 6px;
317   background: transparent;
318 }
319
320 .controls > button:hover svg {
321   color: rgb(75, 173, 230);
322 }
323 </style>
324
325 <script>
326 // Get the modal
327 var modal = document.getElementById('id01');
```

```
291 .controls button:nth-child(1) svg {
292   color: #2b128e;
293 }
294
295 .controls button:nth-child(2) {
296   border: 2px solid #008496;
297 }
298
299 .controls button:nth-child(2) svg {
300   color: #008496;
301 }
302
303 .controls button:nth-child(3) {
304   border: 2px solid #0048b5;
305 }
306
307 .controls button:nth-child(3) svg {
308   color: #0f0a5b;
309 }
310
311 .controls > button {
312   width: 45px;
313   height: 45px;
314   text-align: center;
315   border-radius: 100%;
316   margin: 0 6px;
317   background: transparent;
318 }
319
320 .controls > button:hover svg {
321   color: rgb(75, 173, 230);
322 }
323 </style>
324
325 <script>
326 // Get the modal
327 var modal = document.getElementById('id01');
```

```

329 // When the user clicks anywhere outside of the modal, close it
330 window.onclick = function(event) {
331     if (event.target == modal) {
332         modal.style.display = "none";
333     }
334 }
335 feather.replace();
336
337 const controls = document.querySelector('.controls');
338 const cameraOptions = document.querySelector('.video-options>select');
339 const video = document.querySelector('video');
340 const canvas = document.querySelector('canvas');
341 const screenshotImage = document.querySelector('img');
342 const buttons = [...controls.querySelectorAll('button')];
343 let streamStarted = false;
344
345 const [play, pause, screenshot] = buttons;
346
347 const constraints = {
348     video: {
349         width: {
350             min: 1280,
351             ideal: 1920,
352             max: 2560,
353         },
354         height: {
355             min: 720,
356             ideal: 1080,
357             max: 1440
358         },
359     }
360 };
361 </script>
362 <script>
363 const getCameraSelection = async () => {
364     const devices = await navigator.mediaDevices.enumerateDevices();

```

```

365     const videoDevices = devices.filter(device => device.kind === 'videoinput');
366     const options = videoDevices.map(videoDevice => {
367       return `<option value="${videoDevice.deviceId}">${videoDevice.label}</option>`;
368     });
369     cameraOptions.innerHTML = options.join('');
370   });
371
372   </script>
373   <script>
374
375   play.onclick = () => {
376     if (streamStarted) {
377       video.play();
378       play.classList.add('d-none');
379       pause.classList.remove('d-none');
380       return;
381     }
382     if ('mediaDevices' in navigator && navigator.mediaDevices.getUserMedia) {
383       const updatedConstraints = {
384         ...constraints,
385         deviceId: {
386           exact: cameraOptions.value
387         }
388       };
389       startStream(updatedConstraints);
390     }
391   };
392
393   const startStream = async (constraints) => {
394     const stream = await navigator.mediaDevices.getUserMedia(constraints);
395     handleStream(stream);
396   };
397
398   const handleStream = (stream) => {
399     video.srcObject = stream;
400     play.classList.add('d-none');
401     pause.classList.remove('d-none');

```

```
402     screenshot.classList.remove('d-none');
403     streamStarted = true;
404 };
405
406 getCameraSelection();
407
408 cameraOptions.onchange = () => {
409     const updatedConstraints = {
410         ...constraints,
411         deviceId: {
412             exact: cameraOptions.value
413         }
414     };
415     startStream(updatedConstraints);
416 };
417
418 const pauseStream = () => {
419     video.pause();
420     play.classList.remove('d-none');
421     pause.classList.add('d-none');
422 };
423
424 const doScreenshot = () => {
425     canvas.width = video.videoWidth;
426     canvas.height = video.videoHeight;
427     canvas.getContext('2d').drawImage(video, 0, 0);
428     screenshotImage.src = canvas.toDataURL('image/webp');
429     screenshotImage.classList.remove('d-none');
430 };
431
432 pause.onclick = pauseStream;
433 screenshot.onclick = doScreenshot;
434 </script>
435
436 </body>
437 </html>
```



```

75     </section>
76     <div class="modal fade" role="dialog" tabindex="-1" id="modal-1">
77       <div class="modal-dialog" role="document">
78         <div class="modal-content">
79           <div class="modal-header">
80             <h4 class="modal-title">American Sign Language - Alphabets</h4><button type="button"
81               class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
82           </div>
83           <div class="modal-body"></div>
84           <div class="modal-footer"><button class="btn btn-secondary" type="button"
85             data-bs-dismiss="modal">Close</button></div>
86         </div>
87       </div>
88     </div>
89     <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
90 </body>
91
92 </html>

```

```

1  .bs-icon {
2    --bs-icon-size: .75rem;
3    display: flex;
4    flex-shrink: 0;
5    justify-content: center;
6    align-items: center;
7    font-size: var(--bs-icon-size);
8    width: calc(var(--bs-icon-size) * 2);
9    height: calc(var(--bs-icon-size) * 2);
10   color: var(--bs-primary);
11 }
12
13 .bs-icon-xs {
14   --bs-icon-size: 1rem;
15   width: calc(var(--bs-icon-size) * 1.5);
16   height: calc(var(--bs-icon-size) * 1.5);
17 }
18
19 .bs-icon-sm {
20   --bs-icon-size: 1rem;
21 }
22
23 .bs-icon-md {
24   --bs-icon-size: 1.5rem;
25 }
26
27 .bs-icon-lg {
28   --bs-icon-size: 2rem;
29 }
30
31 .bs-icon-xl {
32   --bs-icon-size: 2.5rem;
33 }
34
35 .bs-icon.bs-icon-primary {
36   color: var(--bs-white);
37   background: var(--bs-primary);

```

```
38 }
39
40 .bs-icon.bs-icon-primary-light {
41   color: var(--bs-primary);
42   background: rgba(var(--bs-primary-rgb), .2);
43 }
44
45 .bs-icon.bs-icon-semi-white {
46   color: var(--bs-primary);
47   background: ■ rgba(255, 255, 255, .5);
48 }
49
50 .bs-icon.bs-icon-rounded {
51   border-radius: .5rem;
52 }
53
54 .bs-icon.bs-icon-circle {
55   border-radius: 50%;
56 }
```