

# Documentación Detallada del Código de Reconocimiento Facial en Python

Este documento proporciona una explicación detallada del código Python que realiza reconocimiento facial utilizando diversas bibliotecas como `face_recognition`, `OpenCV (cv2)`, `numpy`, entre otras. A continuación, se describe paso a paso la funcionalidad de cada línea de código, así como el propósito y funcionamiento de cada biblioteca utilizada.

```

1 import React, {Component} from 'react';
2
3 class App extends Component {
4   constructor(props) {
5     super(props);
6     this.state = {
7       // Initial state
8       // ...
9     };
10    this.handleClick = this.handleClick.bind(this);
11  }
12
13  handleClick() {
14    // Handle click event
15    // ...
16  }
17
18  render() {
19    return (
20      // JSX
21    );
22  }
23 }
24
25 export default App;
26
27 // ...
28
29 // ...
30
31 // ...
32
33 // ...
34
35 // ...
36
37 // ...
38
39 // ...
40
41 // ...
42
43 // ...
44
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 // ...
```

## 1. Librerías Utilizadas

- `face_recognition`: Biblioteca que permite la detección y reconocimiento facial en imágenes. Se basa en `dlib` y permite codificar rostros para su comparación posterior.
- `cv2` (OpenCV): Utilizada para la captura de imágenes desde la cámara, conversión de formatos de color y visualización de imágenes.
- `os`: Permite interactuar con el sistema de archivos, como leer directorios o rutas.
- `pathlib.Path`: Facilita el trabajo con rutas de archivos y carpetas de forma más intuitiva y multiplataforma.
- `numpy`: Usada para operaciones matemáticas, como calcular la distancia entre codificaciones faciales.
- `datetime`: Permite obtener la hora actual para registrar los ingresos de personas.
- `time.sleep`: Utiliza una pausa en segundos, usada para dar tiempo antes de tomar una foto.

## 2. Explicación Detallada del Código

A continuación, se explican todas las líneas y bloques del código en orden lógico.

### Inicialización y Preparación

Se importan las bibliotecas necesarias. Luego, se define la ruta actual usando `os.getcwd()`. Posteriormente, se notifica al usuario que se está ajustando la cámara, se imprime una cuenta regresiva y se espera un segundo entre cada número para preparar al usuario antes de tomar la foto.

### Captura de Imagen desde la Cámara

Se accede a la cámara utilizando `cv2.VideoCapture(0, cv2.CAP_DSHOW)`. Se descartan 10 fotogramas para dejar que la cámara se estabilice (autoajuste de luz, enfoque, etc.). Luego se toma una captura y se libera la cámara.

### Lectura de Imágenes de Empleados

Se construye la ruta a la carpeta 'database'. Se listan todos los archivos dentro de esa carpeta. Cada archivo se asume que es una imagen, se carga y se guarda tanto la imagen como el nombre (sin extensión) en listas separadas.

### Función `codificar(imágenes)`

Esta función convierte las imágenes a formato RGB, luego codifica cada rostro en una representación numérica mediante `face_recognition.face_encodings`. Solo se agregan codificaciones válidas.

### **Función registrar\_ingresos(persona)**

Lee el archivo CSV de registro. Si el archivo no existe, lo crea. Si la persona ya está registrada, actualiza su hora de ingreso. Si no, agrega una nueva entrada con el nombre y hora actual.

### **Codificación de imágenes de empleados**

Se llama a la función codificar para obtener la codificación facial de cada imagen en la base de datos.

### **Procesamiento de la captura y reconocimiento facial**

Si la captura fue exitosa, se detectan las caras usando face\_locations y se codifican usando face\_encodings. Se comparan las codificaciones con las de la base de datos. Se mide la distancia facial y se obtiene el índice de menor distancia (más parecido). Si la distancia es menor a 0.6, se considera una coincidencia y se saluda a la persona. También se registra su ingreso.

### **Dibujado y visualización de resultados**

Se dibujan rectángulos alrededor de las caras detectadas. Se muestra el nombre sobre un rectángulo relleno. Finalmente, se muestra la imagen usando cv2.imshow.