

# Práctica 2: Segmentación precisa de la imagen de la oreja

*Máster Universitario en Visión Artificial - URJC*



**Vicente Gilabert Maño - Luis ROSARIO TREMOULET**

19/04/2022

# Sumario

|                                |          |
|--------------------------------|----------|
| <b>Introducción</b>            | <b>1</b> |
| Estudio del problema           | 1        |
| Base de datos                  | 2        |
| <b>Solución</b>                | <b>3</b> |
| Lenguaje, ficheros y paquetes  | 3        |
| Funcionamiento del algoritmo   | 4        |
| <b>Evaluación</b>              | <b>4</b> |
| Método de evaluación : IOU     | 4        |
| Generación de los ground truth | 5        |
| <b>Resultados</b>              | <b>6</b> |

# Introducción

## Estudio del problema

Una modalidad biométrica relativamente poco estudiada es el reconocimiento de personas basándose en la imagen de su oreja. La oreja es un rasgo facial muy característico por su morfología que, además de ser relativamente invariante al paso del tiempo, presenta una capacidad discriminatoria bastante alta. Un paso previo a la extracción de características de este rasgo facial para su uso biométrico consiste en disponer de una segmentación o detección precisa de la región de la oreja a partir de una imagen de perfil. Esta tarea puede presentar algunas dificultades al quedar la oreja parcialmente ocluida por el pelo, por el uso de pendientes, piercings, etc.

El objetivo de esta práctica es conseguir una segmentación precisa de este rasgo. Es decir, lo más ajustado a la región de la oreja sin perder información de dicha región facial. Está correcta segmentación, permitirá mejorar los resultados de reconocimiento/verificación de una persona a partir de dicho rasgo. El resultado debería parecerse más o menos a la figura siguiente, en la que aparece un cuadrado rojo que recorta la oreja.

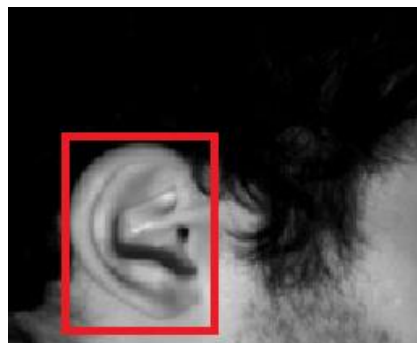


Imagen 1 : Ejemplo de recorte

## Base de datos

La base de datos que tenemos está compuesta por sólo 20 imágenes. Estas imágenes han sido capturadas en el espectro infrarrojo cercano para reducir la influencia de la iluminación. Todas las imágenes están en formato “bitmap” (.bmp).

Al tener sólo 20 imágenes a nuestra disposición, no podemos utilizar una solución de machine learning o deep learning porque no tenemos suficientes datos.

Si intentamos resolver este problema con deep o machine learning puede que obtengamos muy buenos resultados sobre la base de datos ya conocida, pero cuando probemos sobre un nuevo ejemplo, es muy probable que el algoritmo no dé buenos resultados (overfitting).

Por lo tanto, trataremos de resolverlo sin machine learning o deep learning.

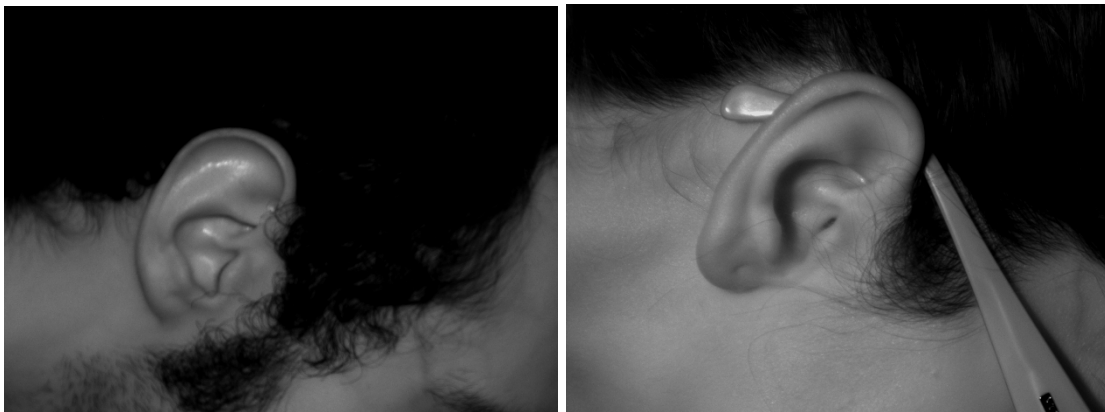


Imagen 2 : Ejemplos de la base de datos

# Solución

## Lenguaje, ficheros y paquetes

Para resolver este problema hemos optado por desarrollar el algoritmo en Python3. Esto nos permitirá poder utilizar una multitud de paquetes.

Hemos desarrollado dos archivos Python: “main.py” (algoritmo para encontrar el área de la oreja) y “groundtruth\_generator.py” (que genera un archivo json con las coordenadas del ground truth), este archivo se explicará en detalle en la parte de evaluación.

Para los paquetes hemos decidido utilizar :

- OpenCV (gestion de imagenes)
- Numpy (Matrices)
- argparse (gestion de argumentos)
- os (gestion de ficheros y carpetas)
- json (para guardar las coordenadas del ground truth)
- scipy.ndimage (obtener el mas grande componente conectado)
- shapely (para calcular las áreas de las máscaras para el IOU)

## Funcionamiento del algoritmo

El funcionamiento técnico de los dos ficheros está explicado en el README.md

La solución que proponemos se divide en seis pasos principales :

Primero leemos la(s) imagen(es) con OpenCV, luego la(s) cambiamos de RGB a una escala de grises. Para entender mejor los siguientes pasos, tomaremos esta imagen como ejemplo:



Imagen 3 : Imagen de perfil (ejemplo número 7 en la base de datos)

En segundo lugar, probablemente el paso más importante, pasamos un filtro llamado “top-hat” sobre la imagen. El filtro “top-hat” se utiliza para realzar los objetos brillantes de interés en un fondo oscuro. La operación de un “black-hat” se utiliza para hacer lo contrario, realzar los objetos oscuros de interés en un fondo brillante.

El filtro “top-hat” necesita un kernel para poder funcionar, probando varios kernels hemos decidido elegir una matriz circular de tamaño 107 x 107. Aquí podemos ver una matriz similar, la matriz circular de tamaño 5 x 5 :

|   |   |   |   |   |
|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 |

Imagen 4 : Matriz circular de tamaño 5 x 5

Aquí podemos ver el resultado del filtro sobre la imagen de perfil :

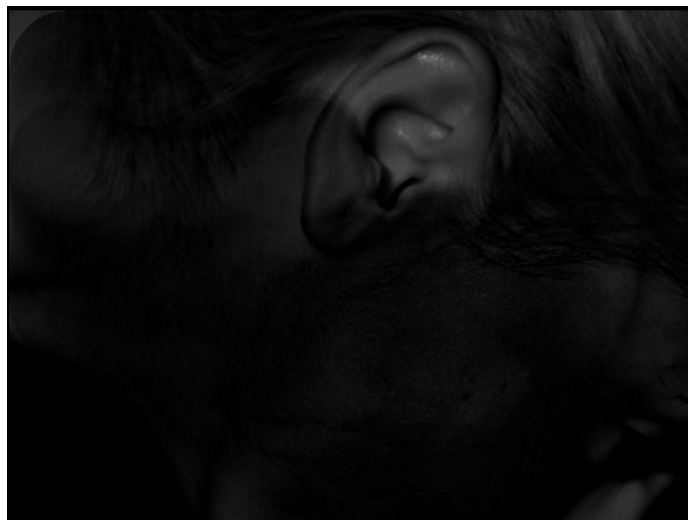


Imagen 5 : Resultado del filtro “top-hat”

En un tercer lugar pasamos el resultado del “top-hat” en blanco y negro. Para convertir la imagen en binaria utilizamos el método de binarización de Otsu.

Para hacer un binarización se utiliza generalmente un umbral elegido arbitrariamente. En cambio, el método de Otsu evita tener que elegir un valor y lo determina automáticamente. Consideremos una imagen con sólo dos valores

distintos (imagen bimodal), en la que el histograma sólo constaría de dos picos. Un buen umbral estaría en el medio de esos dos valores. Del mismo modo, el método de Otsu determina un valor óptimo de umbral global a partir del histograma de la imagen.



Imagen 6 : Resultado de la binarización de Otsu

En un cuarto lugar tratamos de determinar la mayor área conectada de la imagen (para encontrar la oreja). Hemos realizado esta etapa con la ayuda de la librería “scipy”.



Imagen 7 : Mayor área conectada de la imagen



El penúltimo paso es encontrar los píxeles derecho, izquierdo, superior e inferior de la mayor área conectada para dibujar la predicción. Hemos optado por calcular las coordenadas de la detección como el algoritmo YOLO con: x, y, anchura y altura (x, y, w, h)

Finalmente, en el último paso dibujamos la predicción (en rojo), el ground truth (en verde) y calculamos el resultado de la métrica IOU que veremos a continuación. Aquí nuestra puntuación es de 0,762, que es una buena puntuación.

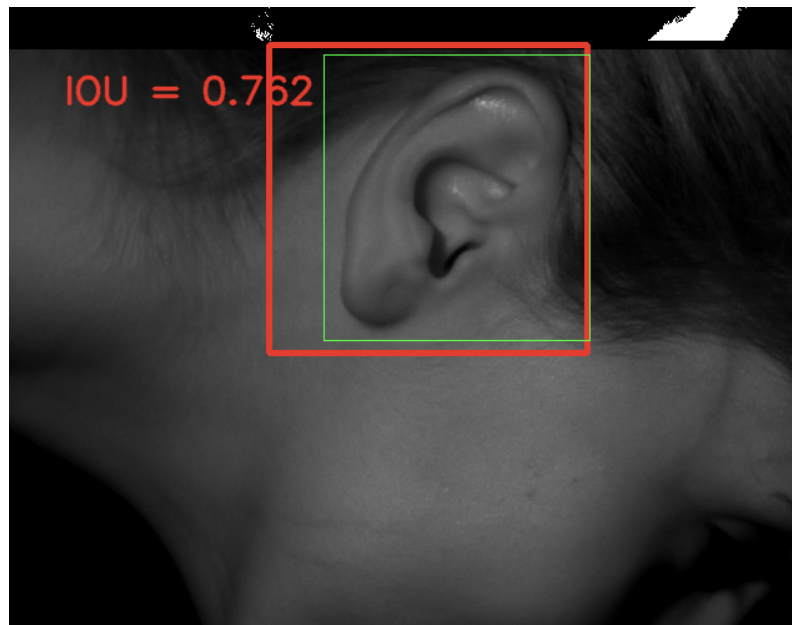


Imagen 8 : Resultado final

## Evaluación

## Método de evaluación : IOU

Para evaluar nuestro algoritmo utilizamos la métrica IOU (Intersection over Union)

En los problemas de detección de objetos, el IoU (intersection over union) evalúa el solapamiento entre la máscara que hay que detectar (GT, “ground truth”) y la máscara predicha (PD, “prediction”). Se calcula como el área de intersección entre el ground truth y la predicción dividida por el área de unión de las dos, es decir

$$\text{IoU} = \frac{\text{area}(gt \cap pd)}{\text{area}(gt \cup pd)}$$

Esquemáticamente, IoU se define como

$$IOU = \frac{\text{area of overlap}}{\text{area of union}} = \frac{\text{area of overlap}}{\text{area of union}}$$

La métrica IoU varía entre 0 y 1, donde 0 significa que no hay solapamiento y 1 implica un solapamiento perfecto entre el ground truth y la predicción. Con la

métrica IoU, tenemos que definir un umbral (por ejemplo,  $\alpha$ ) que se utilizará para distinguir una detección válida de una inválida.

## Generación de los ground truth

Las 20 imágenes dadas no estaban etiquetadas (no tenían un GT). Por eso hemos desarrollado un algoritmo "groundtruth\_generator.py" que utilizamos para etiquetar las orejas. Con este algoritmo podemos ver la imagen, seleccionar el área del GT y genera un archivo JSON con las coordenadas x, y, width y height. (para utilización mirar el README.md)

## Resultados

Con nuestro algoritmo conseguimos muy buenos resultados con un IOU entre 0,7 y 0,9. Tenemos cuatro tipos de resultados posibles por el algoritmo:

- Una imagen con sólo la predicción del área de la oreja
- Una imagen con la predicción y su puntuación IOU
- Una imagen con la predicción, su puntuación IOU y la verdad del terreno
- Una imagen con los diferentes pasos del algoritmo

(Para saber como hacer mirar README.md, parámetro "show")

Aquí podemos ver el resultados con el parámetro igual a "detail", donde se pueden ver todos los pasos del algoritmo :

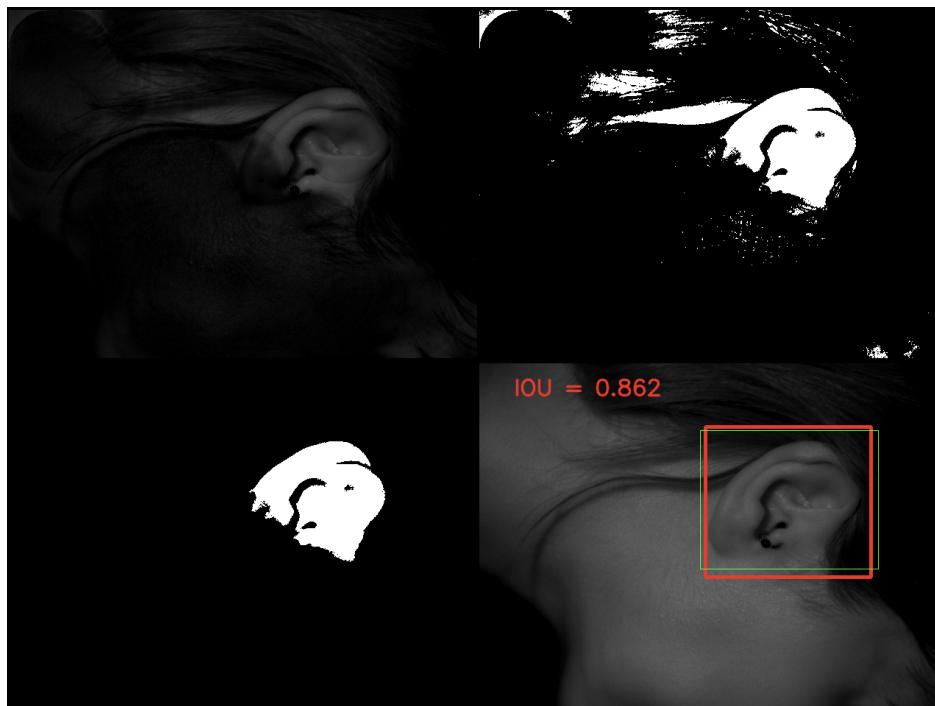


Imagen 9 : Los diferentes pasos del algoritmo

A continuación podemos ver algunos resultados con el ground truth y el IOU score, donde la máscara roja es la predicción y la verde el GT:

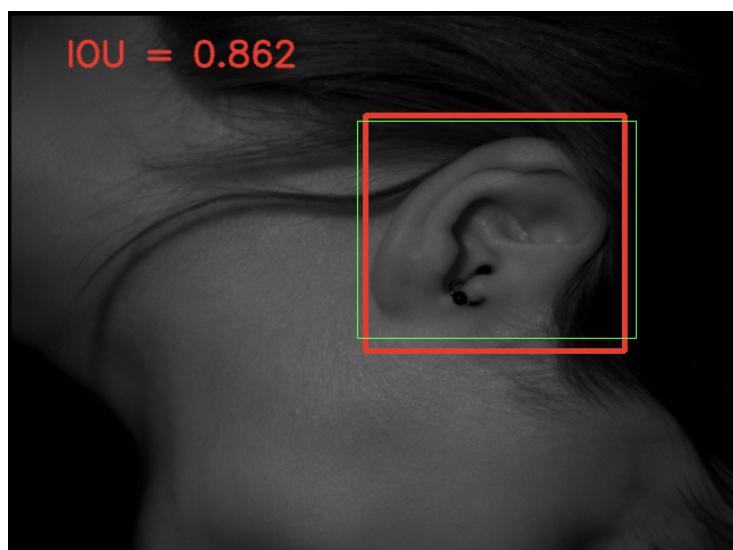


Imagen 10 : Resultado con pendiente

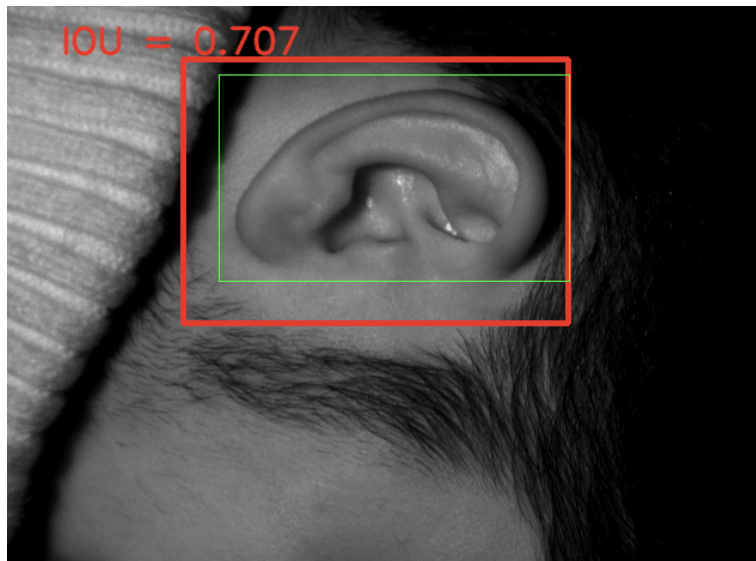


Imagen 11 : Resultado con mucha barba

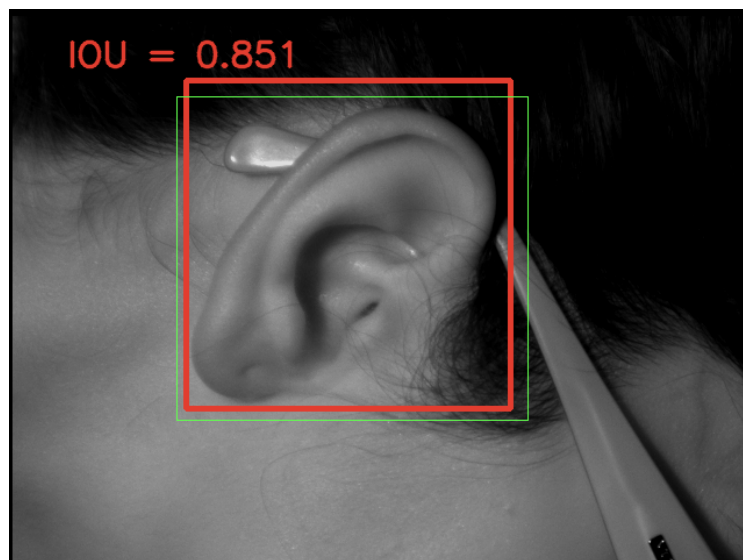


Imagen 12 : Resultado con gafas

Finalmente, podemos ver en estos tres ejemplos que el algoritmo funciona bastante bien y da muy buenos resultados aunque la oreja esté parcialmente ocluida por el pelo, por el uso de pendientes, piercings, barba, etc.