

# Proyecto final de prácticas

## Tratamiento digital de imagen. Visión artificial

Participantes:

Hugo Martín Escrihuela

Diego Baldoví Santiago

# Índice:

- [-Introducción](#)
- [-GUI](#)
- [-Algoritmo y código](#)
- [-Resultados](#)
- [-Discusión de los resultados e ideas de mejora](#)
- [-Anexo](#)
- [-Bibliografía](#)

# Introducción

En este proyecto se presenta el trabajo final de prácticas de la asignatura de tratamiento digital de Imagen del 3º curso del Grado de Tecnologías Interactivas.

El propósito principal de este proyecto realizado mediante una GUI de matlab es la detección de bocas y el cálculo del nivel de sonrisa dada la imagen de una cara.

La detección de sonrisa se llevarán a cabo los procesos vistos en clase mediante un algoritmo simple que realizará todos los pasos de manera autónoma o bien también se puede llevar a cabo de manera manual por el usuario introduciendo los valores necesarios de cada proceso, de esta manera es un proceso más interactivo y se pueden corregir cualquier problema que hubiera cometido el algoritmo automático.

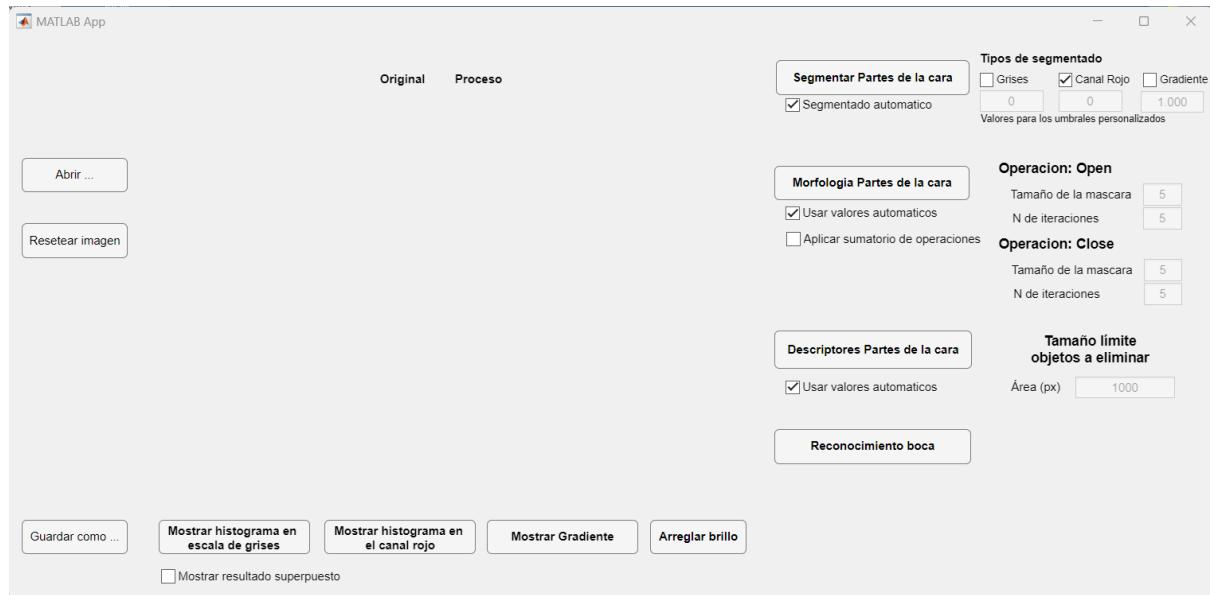
El proceso que se realizará para la detección de sonrisa será el siguiente.

Al cargar una imagen primero se segmentará la imagen de tres formas distintas para obtener la mayor información de la cara; una umbralización simple en la imagen en escala de grises, una umbralización simple en el componente rojo de la imagen y una detección de bordes mediante el gradiente de la imagen, una vez segmentada la imagen aplicaremos un pequeño algoritmo con operaciones morfológicas obteniendo así un segmentado de la boca. Posteriormente se aplicarán descriptores para de los objetos resultantes de la morfología quedarnos con la boca. Para eso se emplearán dos comprobaciones con el fin de que no dé lugar a errores, primero se filtrará por el área de los objetos para eliminar objetos pequeños cercanos a la boca, y después se buscará la boca por posición, teniendo en cuenta que debería estar en mitad del eje X y en la mitad de abajo (aproximadamente 2/3 de la imagen) del eje Y. Finalmente con el objeto de la boca localizado, realizaremos con los dos ejes la relación de aspecto para detectar el porcentaje de sonrisa de la boca, también haremos la relación de aspecto inversa para mayor precisión, al hacer esto se nos muestra el nivel de sonrisa gráficamente en la imagen.

# GUI

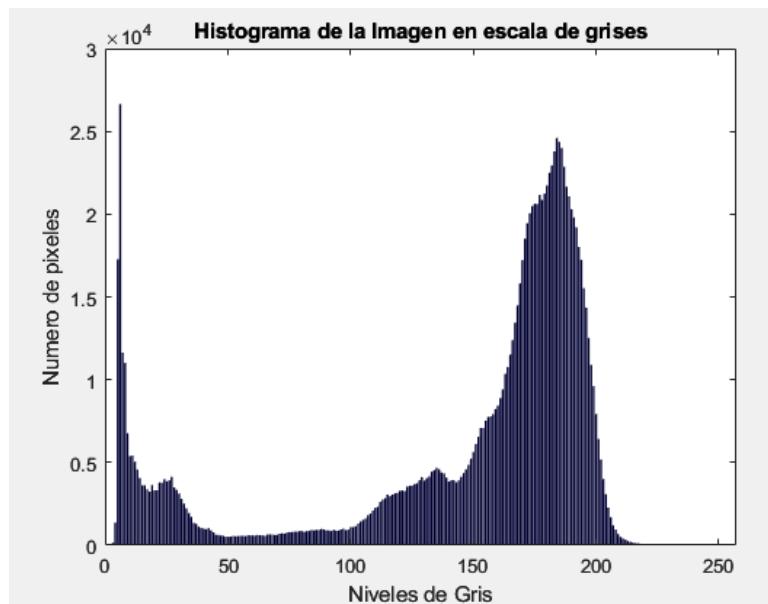
La pantalla principal de nuestra GUI

Esta es la pantalla principal de nuestra GUI, a la izquierda tenemos el botón “Abrir” que cargará la imagen que queramos al axis debajo del título “Original”, bajo el botón abrir tenemos el botón “Resetear imagen” cuya funcionalidad será borrar todas las imágenes que tengamos cargadas como si se acabara de abrir el programa.

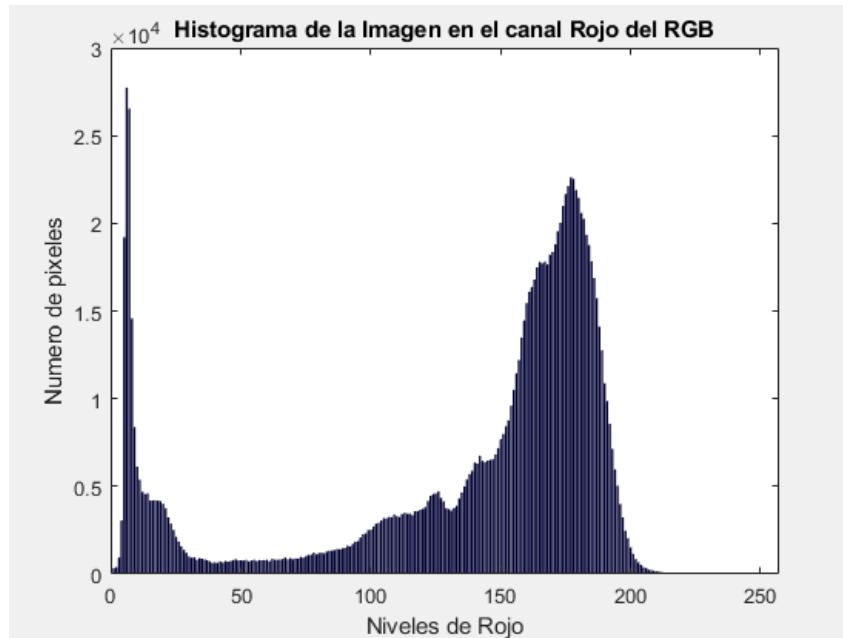


En la parte de abajo tenemos los botones “Guardar como” el cual guardará la imagen que se encuentre en el axis de “Proceso” en nuestro ordenador donde le indiquemos.

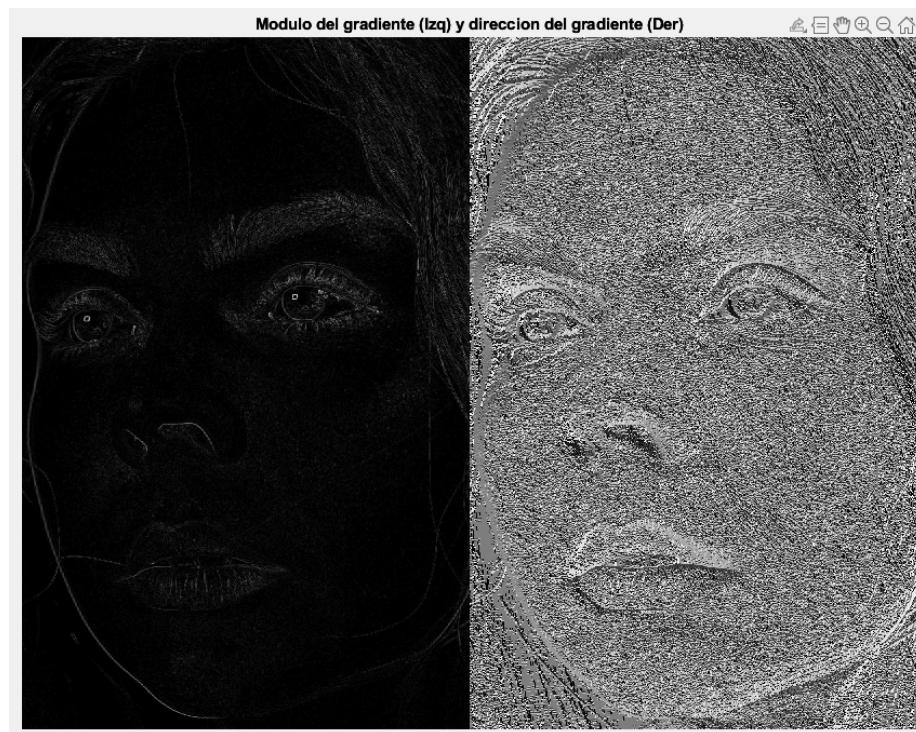
Un botón “Mostrar histograma en escala de grises” que abrirá una nueva figura mostrando el histograma de la imagen cargada en escala de grises.



El botón “Mostrar histograma en el canal rojo” que mostrará en una nueva figura el histograma del canal rojo la imagen cargada RGB.



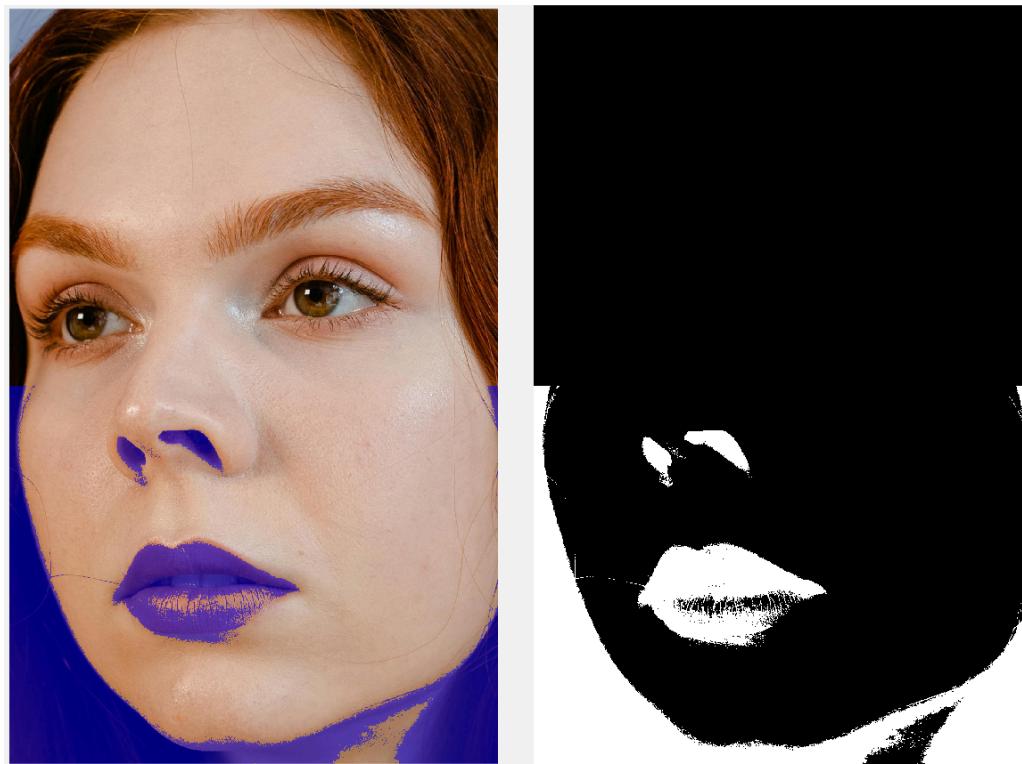
Un botón “Mostrar Gradiente” que mostrará en una nueva figura la dirección y magnitud del gradiente de la imagen cargada.



Y por último el botón “Arreglar brillo” el cual aplicará un ecualizado al histograma de la imagen en blanco y negro cargada para alterar el brillo y el contraste y solucionar notablemente los fallos de iluminación que pueda tener una imagen y así poder trabajar con ella.



Bajo el botón de mostrar histograma en escala de grises tenemos un checkbox que, al activarlo, nos permitirá superponer la imagen resultante del proceso encima de la imagen original para ver bien el proceso de manera más cómoda.



A la derecha nos encontramos el proceso principal del proyecto separados por pasos en los diferentes botones.

El botón de segmentado ejecuta el algoritmo de segmentado en la imagen original sin hacerle aún ninguna morfología y muestra la imagen resultante en el axis de “proceso”. Además cuenta con un checkbox que indica si se quiere usar el algoritmo automático para hacer la segmentación o si se hará la segmentación a mano. Al seleccionar que se hará la segmentación a mano se habilitarán los campos a su derecha para indicar cual o cuales métodos de segmentado se quieren aplicar y sus valores para umbralizar, los campos de la umbralización en escala de grises y en el canal rojo aceptan valores entre 0 y 255 incluidos y el campo de segmentación mediante gradiente acepta números de hasta 3 decimales entre el 1 y el 100 incluidos.

Segmentar Partes de la cara

Segmentado automatico

Tipos de segmentado

Grises:  104

Canal Rojo:  120

Gradiente:  6.242

Valores para los umbrales personalizados

El botón de morfología aplica unas operaciones morfológicas para eliminar ruido, segmentaciones erróneas y mejorar la segmentación que nos interesa. También cuenta con la posibilidad, al igual que el anterior botón, de desactivar el algoritmo automático y poder indicar manualmente el tamaño de la máscara que se usará en cada operación y cuantas veces se aplicará. Un último checkbox con el que, si está activado, permitirá al usuario aplicar morfología tantas veces como se pulse el botón de nuevo, al desactivarlo solo se aplicará morfología una vez por muchas veces que se pulse el botón.

Morfología Partes de la cara

Operacion: Open

Usar valores automaticos

Aplicar sumatorio de operaciones

Tamaño de la mascara: 5

N de iteraciones: 5

Operacion: Close

Tamaño de la mascara: 5

N de iteraciones: 5

El botón de los descriptores analiza los objetos obtenidos de la morfología y extrae el objeto referente a la boca que es el necesario para obtener la sonrisa. Viene un valor automático con el valor 1000, el cual representa el área máxima que pueden tener los objetos, esto con el fin de eliminar los objetos pequeños cercanos a la boca. Tenemos la opción, si desactivamos el checkbox, de personalizar el valor del área máxima. Posterior a esta ya encuentra la boca en función de la posición.

Descriptores Partes de la cara

Usar valores automaticos

Tamaño límite objetos a eliminar

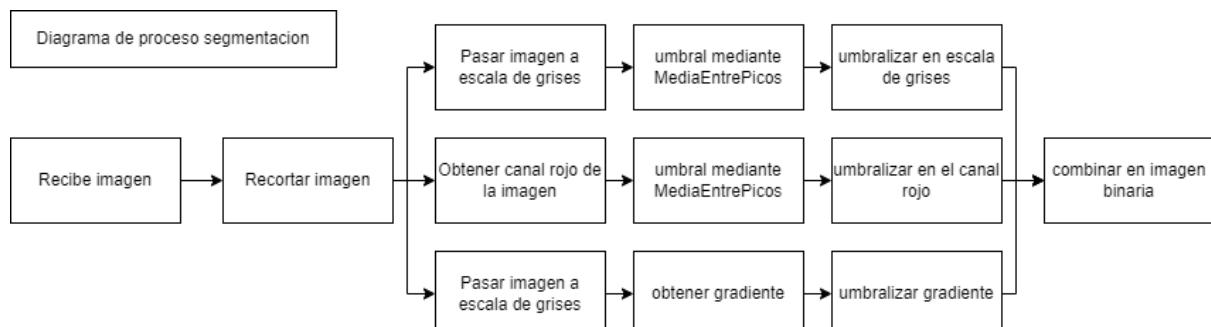
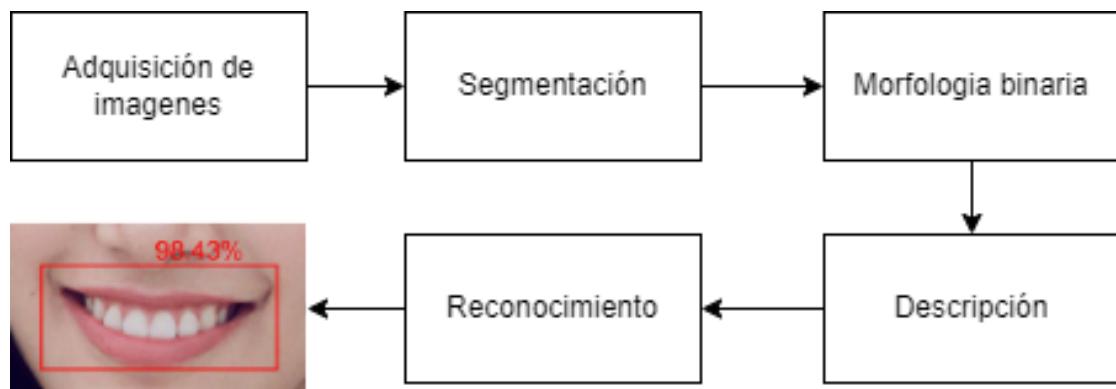
Área (px): 1000

Finalmente, el botón “Reconocimiento boca” realiza el paso final. Una vez que hemos obtenido la boca correctamente procedemos a realizar la relación de aspecto y la relación de aspecto inversa, se usan las dos para más precisión, y así obtenemos el nivel de sonrisa de la cara y lo mostramos en la imagen. Este proceso es automático y no tiene valores personalizables, pues una vez obtengamos la boca de forma correcta no tendremos problema para analizar el porcentaje de sonrisa.

**Reconocimiento boca**

# Algoritmo y código

El proceso de detección de sonrisa después de adquirir la imagen sigue los siguientes pasos:



En el paso de segmentación, el algoritmo utilizado recibe una imagen a color y la recorta para quedarnos con la mitad inferior de la cara ya que la boca está aproximadamente en el tercio inferior de la cara y para evitar problemas recortamos un poco menos de un cuarto. Después esa imagen recortada donde sabemos que se encuentra la boca, se envía a 3 funciones de segmentado:

-La primera función pasa la imagen recibida que está a color y la pasa a escala de grises para después hacerle una umbralización simple y obtener la imagen binaria de la segmentación. El umbral se puede obtener de 2 formas, si está desactivado el checkbox de segmentado automático, se utilizará como valor umbral el valor introducido en el campo correspondiente al segmentado por escala de grises, de lo contrario se llamará a la siguiente función:

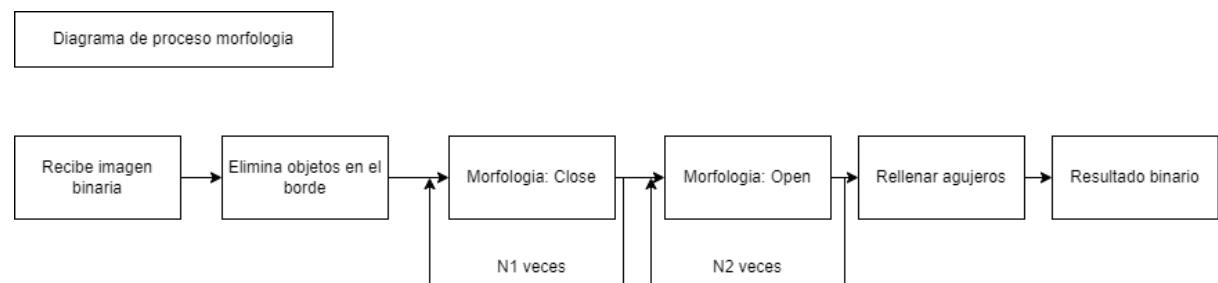
La función “EncontrarMediaEntrePicos” cuyo algoritmo analiza el histograma en busca de picos ya que podemos asumir que el primer pico, aquel más a la izquierda, que sea mayor al 9% del valor máximo (valor obtenido mediante prueba y error) corresponderá a los píxeles que forman las sombras de la cara como puede ser el pelo, las sombras de la nariz o incluso la propia sombra de los labios, después podemos contar como que el segundo pico que cumpla las mismas especificaciones corresponderá a aquellos píxeles claros que corresponden a la piel. Teniendo estos dos picos localizados, el algoritmo obtiene sus

posiciones y calcula la posición media entre ambos ya que los labios siempre serán más claros que los píxeles negros de las sombras pero mucho más oscuros que la piel de esta forma podemos segmentar la imagen en la gran mayoría de casos.

-La segunda segmentación se trata de una segmentación en el canal rojo RGB de la imagen a color. Extraemos la componente verde de la imagen, ya que el color rojo tiene valores muy altos en su propio canal y nosotros buscamos que la boca sea oscura para poder aplicar la función “EncontrarMediaEntrePicos” como en el segmentado anterior y obtener, al igual que en el segmentado anterior, el valor umbral por el que se realizará el segmentado y dará como resultado una imagen binaria.

-El tercer segmentado es un segmentado mediante el gradiente, éste obtiene la imagen a color y la pasa a escala de grises para calcular el gradiente mediante sobel, éste cálculo nos devuelve el umbral medio de ese gradiente, alteramos ligeramente el umbral con el mismo objetivo que “EncontrarMediaEntrePicos” y aplicamos una umbralización al gradiente con el umbral nuevo y obtenemos una imagen binaria con los bordes que no superan el umbral.

De vuelta en la función de segmentación tenemos las tres imágenes binarias de los tres tipos de segmentado y las devolvemos al tamaño de la imagen original quedando las imágenes binarias del mismo tamaño y alineadas con la imagen original que hemos cargado. Después comprueba los checkbox que hay encima de los campos de umbral de los segmentados para comprobar que segmentados utilizar para el resultado final de esta función. Una vez comprobados cuáles se usarán y cuáles no, el algoritmo sumará todas aquellas imágenes que se vayan a utilizar y aquellos píxeles que superen valor 2 (o valor 1 si solo se usa un tipo de imagen) para crear una imagen combinada binaria.



En el paso de morfología binaria el algoritmo recibe la imagen que se encuentre en la variable “`imag_proc_mat`” es decir, la imagen que ya haya sido procesada por ejemplo después de la segmentación. Con esta imagen, la función de morfología elimina por completo todos aquellos objetos que estén en contacto con uno de los bordes, de esta manera eliminamos los objetos que podrían haber sido creados por sombras del cuello, del pelo o en pocos casos de la barba. Despues rellena todos aquellos huecos que hayan quedado vacíos dentro un mismo objeto, generalmente sólo quedarán aquellos que estén dentro de la zona de la boca, y después se aplicarán X veces los procesos de Open y Close con un tamaño de máscara Y, siendo X e Y valores que por defecto serán 5 pero que también pueden editarse en los campos de la GUI. La imagen resultado es una imagen binaria almacenada en “`imag_proc_mat`”.

En el paso de descriptores el algoritmo recibe la imagen binaria procesada en el paso de morfología, de esta sacamos los objetos que contiene, y con estos obtenemos sus propiedades. Agarramos el valor de área máxima y comparamos esta con el valor de área de cada objeto, eliminando los objetos que sean más pequeños que el área máxima, esto con el fin de eliminar pequeños objetos que puedan estar cerca de la boca. Obtenemos una nueva imagen binaria con los objetos que han pasado el filtro y volvemos a obtener las propiedades de estos. Con las dimensiones de la imagen y el centro de los objetos determinamos el objeto que esté en el medio del eje X y aproximadamente en  $\frac{2}{3}$  del eje Y, posición donde debería estar la boca. Finalmente, cuando obtenemos la boca la guardamos en una imagen binaria y obtenemos las propiedades de ésta, las cuales usaremos en reconocimiento.

En el paso de reconocimiento usaremos las propiedades de boca que hemos obtenido en paso de descriptores. El algoritmo de este paso, básicamente realiza la relación de aspecto con los dos ejes, y con el valor obtenido, gracias a un previo análisis de datos, podemos obtener el porcentaje de sonrisa. También realizamos la relación de aspecto inversa con los mismos ejes y sacamos el porcentaje, esto para hacer una media de los porcentajes y obtener un valor más preciso. Finalmente con el valor obtenido dibujamos en la imagen original un rectángulo que rodea la sonrisa y plasmamos el valor para que el usuario lo vea de forma clara.

# Resultados

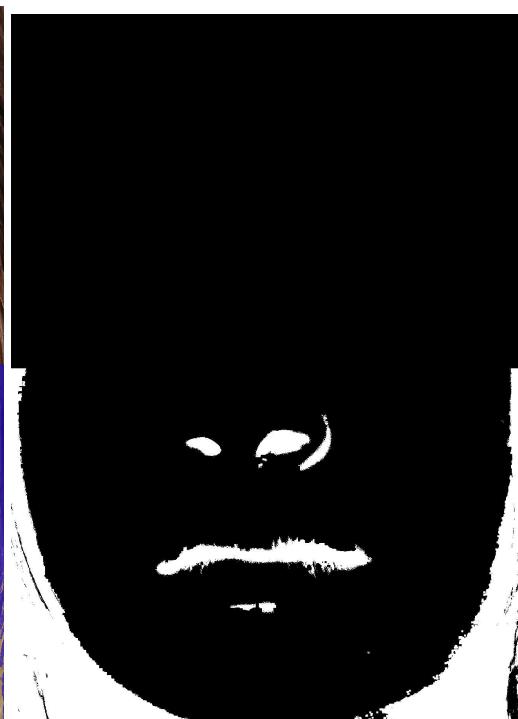
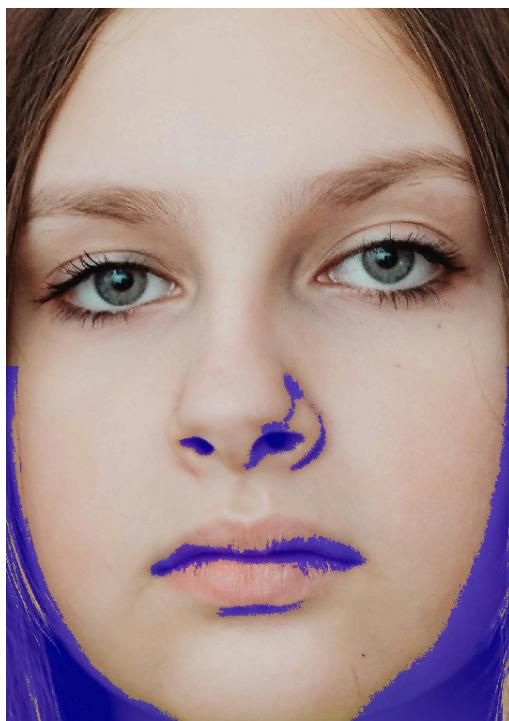
Se mostrarán algunos de los resultados obtenidos con el programa en diferentes casos. Se presentarán a la izquierda la imagen original superpuesta al resultado de cada paso.

## Caso: Seria con boca cerrada

Adquisición de imagen



Segmentado



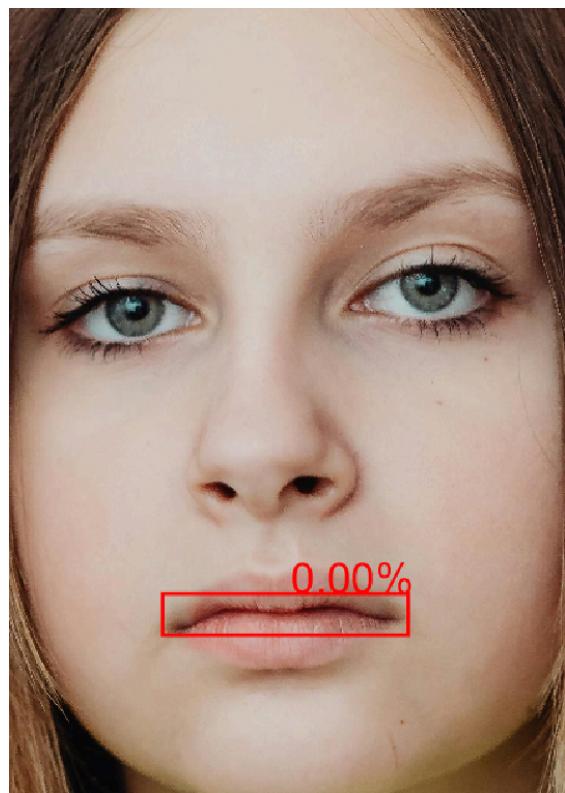
Morfología



Descripción



Reconocimiento



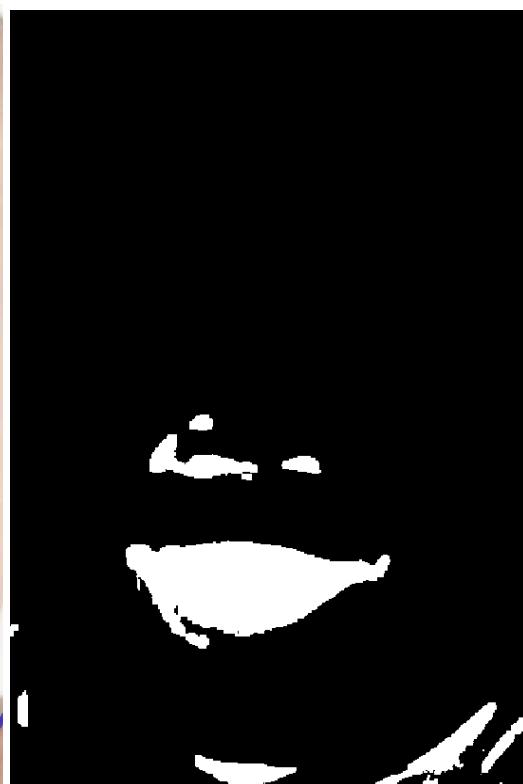
**Caso: Bebe con boca abierta**

Este es el resultado de un bebe sonriendo con la boca abierta

Adquisición de la imagen



Segmentación



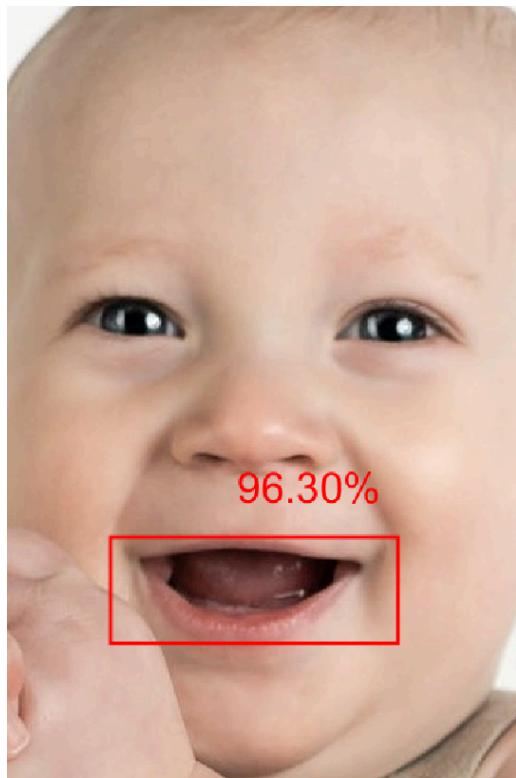
Morfología



Descripción



Reconocimiento



**Caso: Sonriente con boca abierta y dientes**

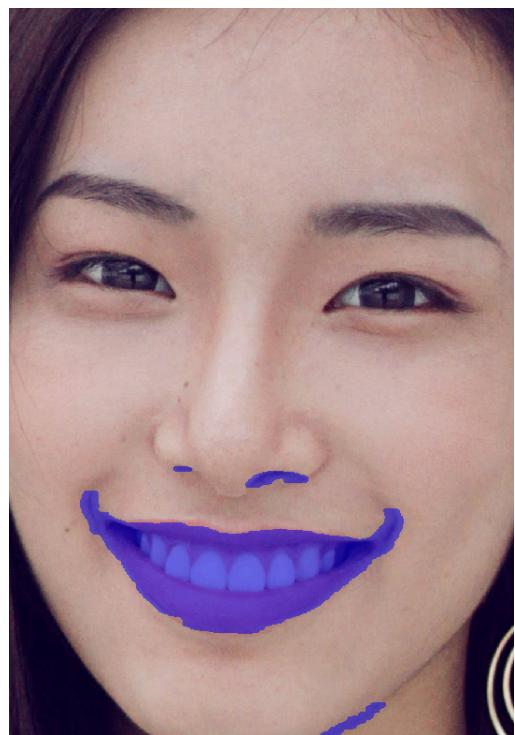
Adquisición de la imagen



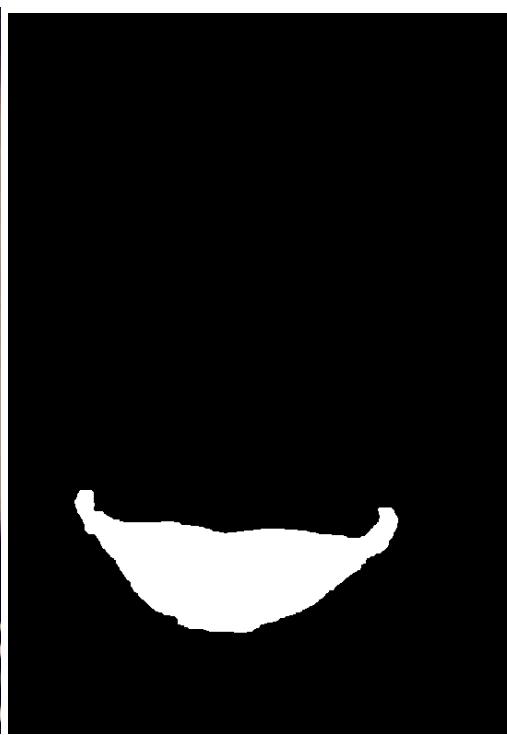
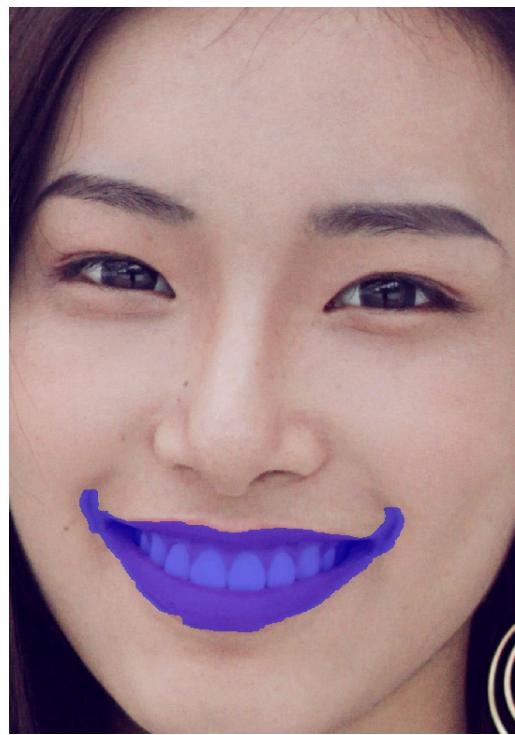
Segmentado



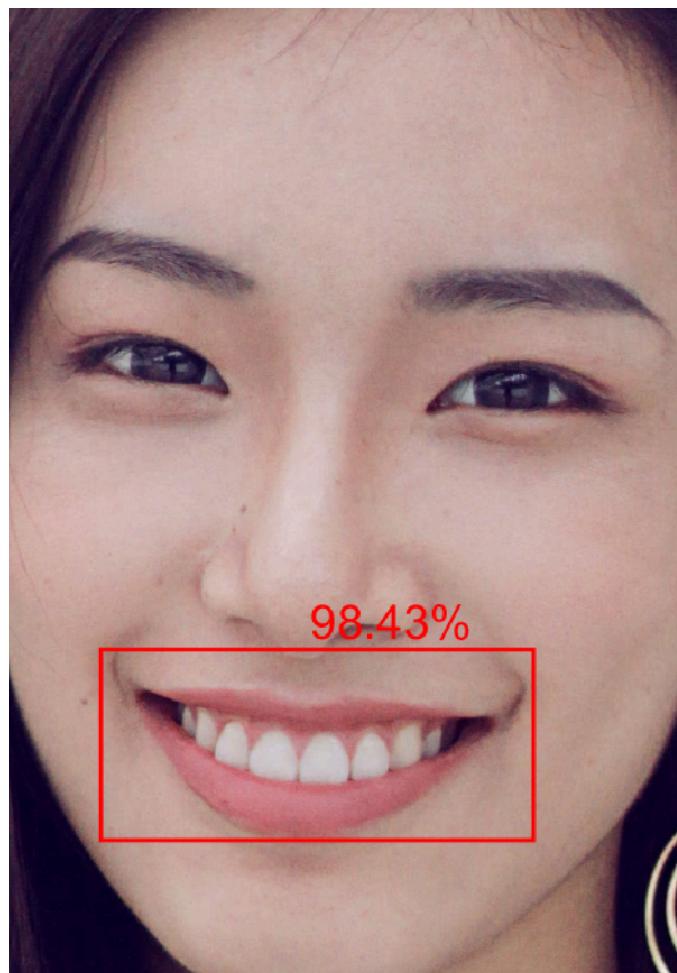
Morfología



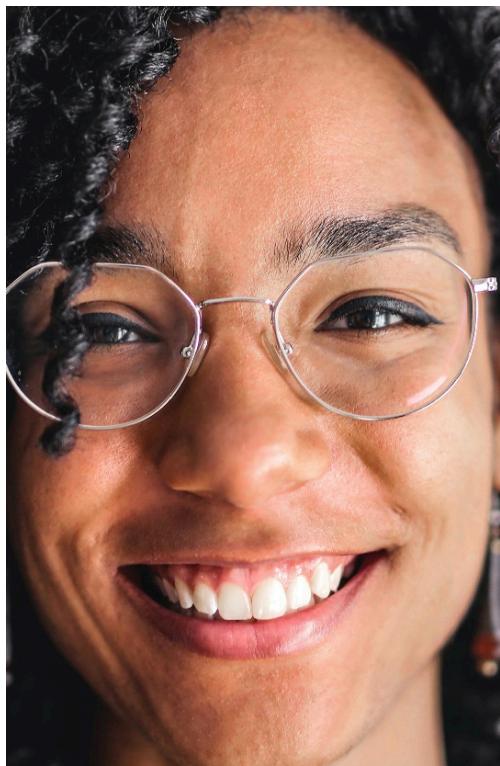
Descripción



Reconocimiento



**Caso: error iluminación no uniforme y arreglado mediante función**  
Adquisición de la imagen



Segmentado



Al haber una iluminación no uniforme el algoritmo falla y usaremos el botón de “Arreglar brillo” para mejorar el resultado (no se mostrarán los cambios del arreglo arreglada en la imagen original superpuesta)

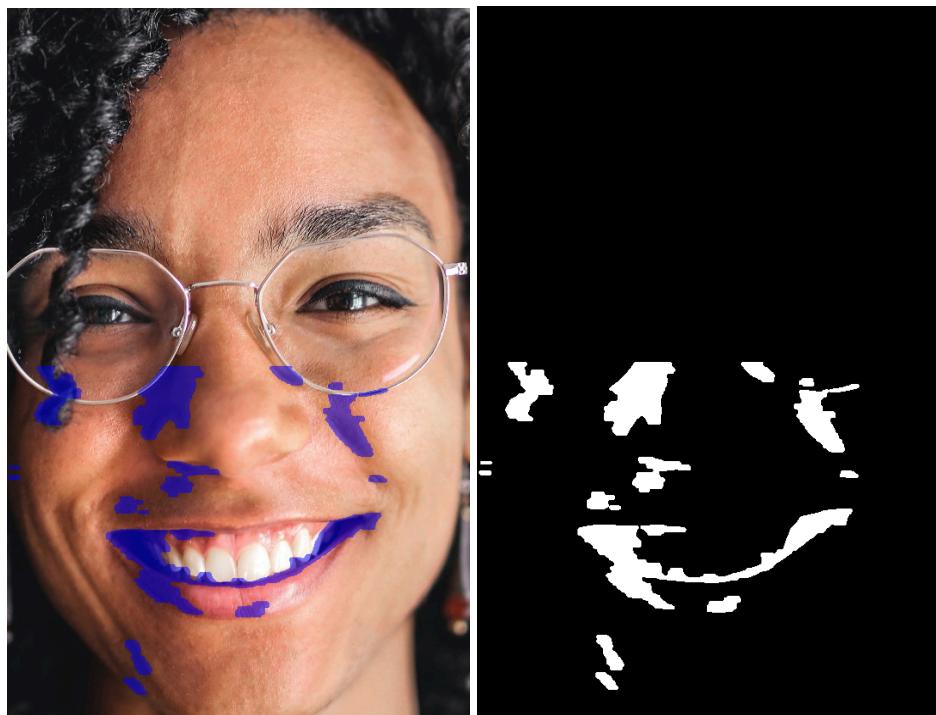
Arreglo de brillo: resultado en escala de grises de la ecualización del histograma



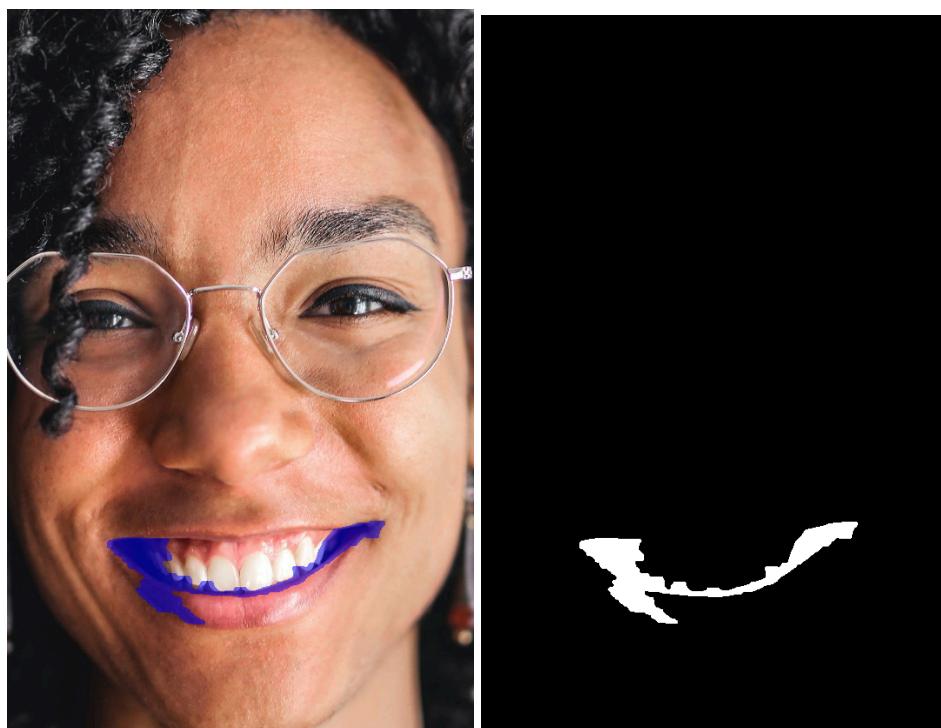
Segmentado tras el arreglo



Morfología



Descripción

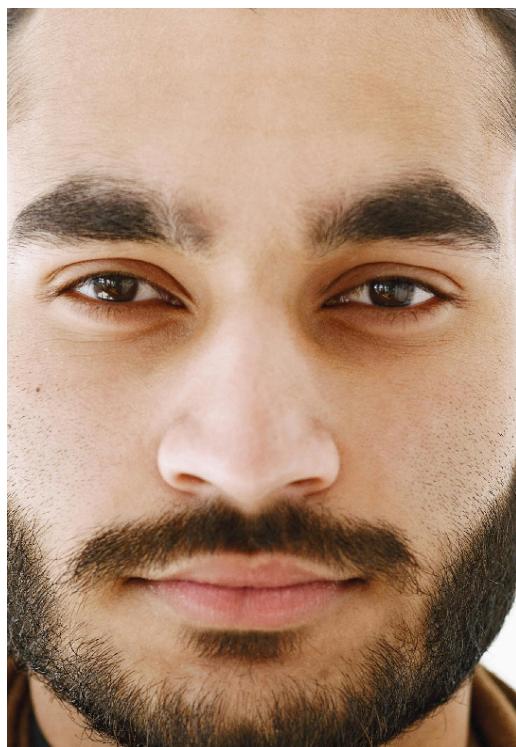


Reconocimiento



**Caso: error por barba**

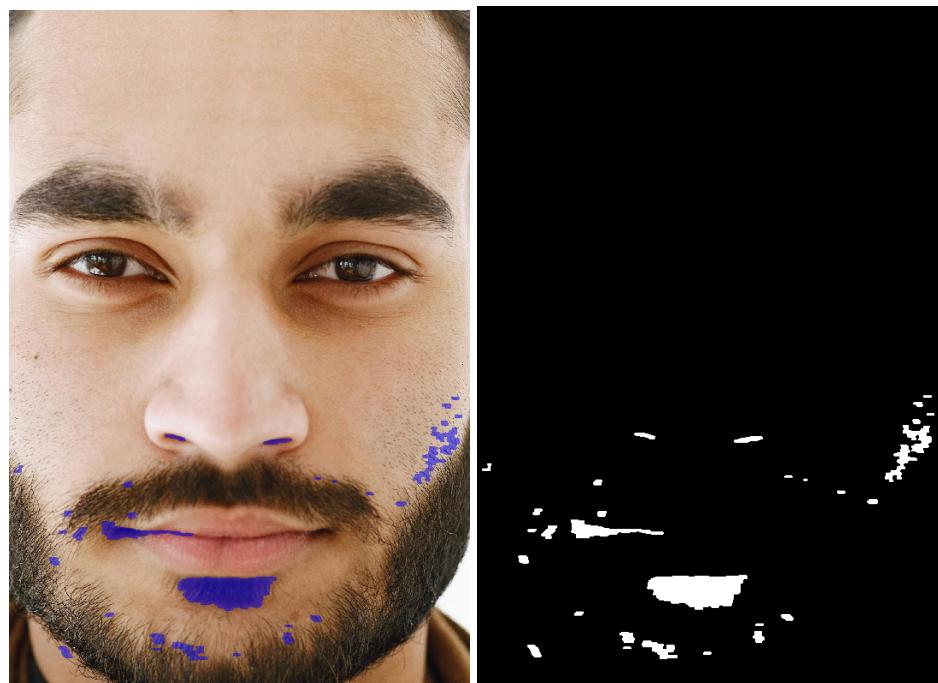
Adquisición de la imagen



Segmentación



Morfología



Al hacer morfología de una imagen que ha tenido problemas de segmentación no podemos obtener la boca y el proceso se da por fallido.

# Discusión de los resultados e ideas de mejora

Como hemos podido ver el programa funciona correctamente mientras no haya perturbaciones en la imagen que afecten al proceso como pueden ser:

La mala iluminación, la barba, las personas de piel oscura o perturbaciones en la cara como brillantes, pintura y otros objetos.

Una idea a futuro para solucionar la segmentación en aquellas personas de piel oscura o que tengan barba es modificar y mejorar el algoritmo del segmentado mediante gradiente ya que de entre los tres tipos de segmentado este es el más útil en estos casos donde los picos de píxeles oscuros son inestables y es el segmentado de entre los tres que tiene un algoritmo más tosco y menos trabajado.

El proceso también tiende a fallar en aquellas imágenes donde la cara está de lado o inclinada, para solucionar ese problema podríamos implementar una detección de ojos y usar su posición relativa el uno del otro para identificar si está inclinada la cara o no y poder corregir la inclinación de ser así.

El algoritmo para arreglar la mala iluminación de algunas imágenes quizá podría arreglarse usando una umbralización local a la hora de segmentar en vez de intentar alterar la iluminación de toda la imagen ya que este proceso afecta mucho al histograma y hace que el algoritmo automático de la segmentación dé muchos problemas.

Otra solución para las barbas podría ser añadir área mínima en el apartado de descriptores, esto con el fin de obtener un objeto en específico que pese a no ser perfecto puede detectar en muchos casos la boca.

En cuanto al reconocimiento se podría tratar de añadir más métodos efectivos con el fin de combinarlos y aumentar la precisión, tales como trabajar de alguna forma con las propiedades de convexidad o investigar nuevas propiedades más complejas que Matlab no proporciona de forma básica.

## Anexo

Hemos utilizado un total de 50 imágenes en el desarrollo y testeo de este programa, en el lugar de presentación, es decir, el espacio compartido, habrá un archivo excel en el que se mostrarán los valores obtenidos por el algoritmo y los valores para una umbralización aceptable o ideal de cada uno de los tres tipos de segmentación antes y después de haber hecho el arreglo de brillo.

En la sección de “tras arreglo de brillo” no se encuentra la segmentación por el canal rojo ya que para segmentar por el canal rojo se necesita una imagen a color y el arreglo de brillo devuelve una imagen en escala de grises por lo que no se puede utilizar tras el arreglo. Aquellos cuyo valor indica “err” en rojo significa que no se ha encontrado valor para el que la segmentación sea aceptable.

En la entrega de este proyecto hemos dado un zip con las imágenes seleccionadas después de un filtrado de las 50 con las que empezamos, en este excel se muestra el estudio que hemos hecho de todas las 50 imágenes, por lo que hay imágenes en el excel que no se encuentran en el zip de presentación.

Nota: Los valores personalizables de la parte de descriptores no aparecen en el excel puesto que no precisan de tanta precisión y su uso es muy intuitivo.

# Bibliografía

-PDFs de teoría y prácticas de la asignatura de Tratamiento digital de imagen.

-Documentación oficial de MatLab

[https://es.mathworks.com/help/images/index.html?s\\_tid=CRUX\\_topnav](https://es.mathworks.com/help/images/index.html?s_tid=CRUX_topnav)  
<https://es.mathworks.com/help/images/ref/regionprops.html#buoixjn-1-properties>  
<https://es.mathworks.com/help/images/morphological-filtering.html>  
<https://es.mathworks.com/help/images/ref/imclearborder.html>  
<https://es.mathworks.com/help/images/ref/imsplit.html>  
<https://es.mathworks.com/help/images/ref/labeloverlay.html#d126e91126>

-Memoria de fin de carrera de Tamara Herrero Vez de Ingeniería de Telecomunicación de la Universidad Carlos 3 de Madrid, Marzo 2010

<https://e-archivo.uc3m.es/bitstream/handle/10016/11170/PFC%20Tamara%20Herrero%20Vez.pdf?sequence=1>

-ChatGPT: funcionalidades básicas con lenguaje de MatLab

<https://chat.openai.com>