

Detección de rostros con tapabocas

Natalia Guevara¹, Matheo Alexander Bermúdez²

¹guevara.pnatalia@javeriana.edu.co ²matheo_bermudez@javeriana.edu.co

Procesamiento de imágenes y visión - Pontificia Universidad Javeriana

2020-III

Fecha de Entrega : Diciembre 4 de 2020

Index Terms—Aspect ratio, phi, python, OpenCV.

I. INTRODUCCIÓN

Actualmente en nuestra sociedad es indispensable el uso de tapabocas, debido a esto hemos decidido diseñar un programa que determine si una persona tiene tapabocas. Esto se realizara con algunas herramientas de visión por computador como umbralización y búsqueda de contornos; y para implementar el programa se usará el software phyton con ayuda de la librería de OpenCV.

II. OBJETIVOS

OBJETIVO GENERAL:

- Implementar un sistema el cual determine si una persona porta uno de los principales elementos de bioseguridad (tapabocas).

OBJETIVOS ESPECÍFICOS:

- Medir el aspect ratio (phi) de distintas personas y sacar medidas estadísticas como promedio, varianza, phi máximo y phi mínimo
- Realizar detección de rostros por contornos, buscando el mínimo rectángulo contenedor del rostro con tapabocas y sin tapabocas

III. DESARROLLO

Para poder detectar si alguna persona tiene tapabocas, se siguió el siguiente proceso:

- Separar el rostro del resto de la imagen por umbralización
- Encerrar el rostro en un rectángulo y tomar las medidas del ancho y el largo para calcular el aspect ratio (phi)
- Determinar si la persona tiene tapabocas de acuerdo con la medida de phi e imprimir un mensaje que le indique si debe ponérselo o no.

De acuerdo al proceso anterior, se plantearon unas restricciones en la operación del programa en tiempo real, las cuales son:

- La persona que utilice el sistema deberá cubrir el cuello y demás zonas descubiertas (con una bufanda por ejemplo), esto con el fin de que solo se detecte el color piel del rostro.
- El fondo de la imagen debe ser en lo posible blanco. Si no es posible, no debe tener elementos de color parecido al de la piel (entre amarillo y café)
- Si la persona tiene algún accesorio en el rostro (gorro, gafas), en lo posible deberá quitárselos y si tiene el pelo largo, en lo posible debe recogerlo.

Con las restricciones y los pasos de solución definidos, se procedió a dividir el programa en 3 partes; la primera parte estaba encargada de medir los datos de phi sin tapabocas y phi con tapabocas a un conjunto de k personas, esto para calcular las fronteras de decisión en el ultimo paso del proceso; la segunda parte se encargaba de tomar el conjunto de k datos y sacar medidas estadísticas como el promedio, la varianza, el phi mínimo y el phi máximo, esto con el fin de conocer el comportamiento del aspect ratio de una muy pequeña población; por ultimo, la tercera parte tenia la función de decidir en tiempo real si la persona en la pantalla tenia tapabocas, valiéndose de las fronteras de decisión (phi mínimo y phi máximo) calculados en la parte 2. El siguiente diagrama de bloques representa el proceso de solución.

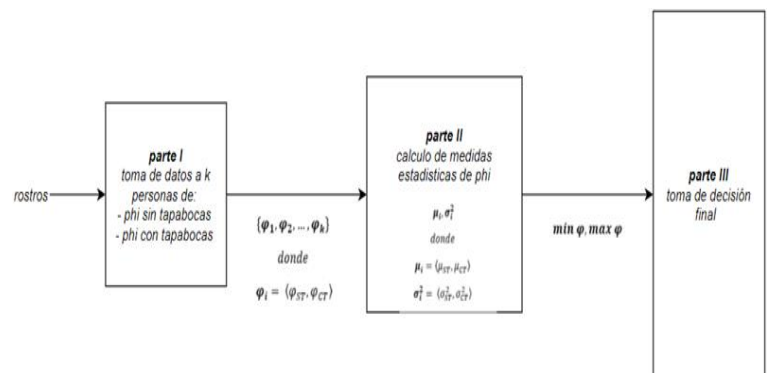


Figura 1. Diagrama de bloques

A continuación se explicará con mas detalle cada una de las partes del programa

III-A. Primera parte: toma de datos

En esta primera parte, se tomaron los datos del phi sin tapabocas (φ_{ST}) y del phi con tapabocas (φ_{CT}) a un conjunto de $k = 8$ personas; con estos dos datos se construyo un vector φ cuyas componentes son (φ_{ST}) y (φ_{CT}).

Para cada persona se siguió un protocolo de medición, el cual consistió en tomar 30 datos de (φ_{ST}) y después promediarlos, y tomar 30 datos de (φ_{CT}) y también promediarlos; para esto, la persona tenia que quedarse quieta en una posición mientras se tomaban 10 datos, después se debía alejar de la cámara para tomar otros 10 datos, y por ultimo tenia que acercar el rostro a la cámara para poder tomar los últimos 10 datos; con estos datos se calculo el promedio.

III-B. Segunda parte: cálculos estadísticos de los datos

Para conocer el comportamiento de (φ_{CT}) y (φ_{ST}) en la población de k personas, se optó por recolectar los datos y calcular el promedio, la varianza y el valor máximo y mínimo, estos valores se almacenaron en un vector μ , σ , **max**, y **min** respectivamente. los valores máximo y mínimo posteriormente sirvieron como fronteras de decisión en la tercera parte

$$\text{Promedio} = \mu = \frac{\sum_{i=1}^k \varphi_i}{k} \quad (1)$$

$$\text{Varianza} = \sigma^2 = \frac{\sum_{i=1}^k (\varphi_i - \mu)^2}{k} \quad (2)$$

III-C. Tercera parte, toma de decisión

En esta ultima parte, los valores **max** y **min** fueron usados como fronteras de decisión; Y la persona solo tenia que colocar su rostro a una distancia cualquiera de la cámara, después se tomaba una medida del valor de phi y con esta medida se indagaba a través de condicionales, si phi estaba dentro de los valores máximos y mínimos para los dos casos (con tapabocas o sin tapabocas), de acuerdo a la decisión tomada se imprimía en pantalla uno de los 2 posibles mensajes.

- póngase el tapabocas por favor
- cumple al menos un protocolo de bioseguridad

IV. DIAGRAMA DE FLUJO

A continuación observaremos los diagramas de flujo, es este caso se utilizaran dos, uno para definir los valores de la umbralización y el otro sera para determinar si una persona tiene o no tapabocas.

IV-A. Primer diagrama de flujo

En este diagrama de flujo observamos que inicialmente se crean las variables que definen la umbralización de phi.

Luego de esto se debe saber si $k=27$ esto quiere decir que si se oprimió la tecla 'Esc' o no, si es así no se visualizara nada, de lo contrario se visualizara el vídeo frame por frame

Sabiendo esto al momento de visualizar los frames se realiza un filtrado gaussiano con un kernel de 7×7 y después de esto pasar los frames (los cuales están en BGR) al formato de HSV.

Se identifican los umbrales superiores e inferiores en formato HSV para el color de la piel

Con estos datos y los frames en formato HSV se aplican los operadores morfológicos y se realiza una mascara para poder separar la cara y el fondo de la imagen.

Se define el contorno de la cara, y se crea un rectángulo al rededor de esta del mínimo tamaño posible.

Si la variable phi esta entre el rango de la umbralización sin tapabocas se imprimirá un texto en pantalla el cual indicara a la persona que se debe poner el tapabocas, de lo contrario se vera si esta variable esta en el rango de la umbralización con tapabocas, si es así se imprimirá en pantalla que tiene el tapabocas.

Cada vez que se detecte un rostro este se encerrara en un recuadro.

Podemos observar que la única salida de este bucle así como se indico al comienzo sera el de oprimir la tecla 'ESC' la cual es la única que saca el código del streaming de vídeo.

IV-B. Segundo diagrama de flujo

Al igual que en el primer diagrama de flujo inicialmente se inician las variables en este caso se inician en cero debido a que el valor de estas ira cambiando a medida que se ejecute el código. También indica que la manera de terminar el bucle sera cuando $k=27$, esto quiere decir que sera cuando se oprima la tecla 'ESC'.

En este diagrama de flujo se podrá evidenciar un entrenamiento el cual arroja un valor promedio de phi el cual ayudara sera uno de los limites de umbralización para el código anterior

La parte inicial de este código es bastante similar a la del anterior, debido a que se definen los valores de umbralizacion del color piel, se pasa la imagen a un formato HSV, y se realiza un filtro gaussiano con un kernel de 7×7 , después de esto se aplican los operadores morfológicos y se realiza una mascara para poder separar la cara y el fondo de la imagen. Se define el contorno de la cara, y se crea un rectángulo al rededor de esta del mínimo tamaño posible.

Después de esto se realiza un condicional donde se realizan 70 iteraciones, dentro de este se determina el aspect ratio (phi) que contiene el rectángulo del rostro.

Dentro de este condicional se encuentra otro condicional el cual tiene 30 iteraciones. En el cual se promedian los 30 datos de phi obtenidos anteriormente, y se imprime en pantalla el promedio de phi, el cual servirá para los umbrales del diagrama de flujo anterior.

V. RESULTADOS

A continuación, se evidenciaran los resultados obtenidos a partir de los problemas y casos mencionados anteriormente:

V-A. Primera parte: Toma de datos

Para la toma de datos contamos con la participación de un total de 9 personas a los que se le midió los dos tipos de phi mencionados anteriormente con el protocolo de medidas también mencionado, a continuación se muestran los resultados

Persona	Phi sin tapabocas	Phi con tapabocas
Nancy	1.3212732919	0.9329533808
Cenaida	1.2931480719	0.942946698
Johanna	1.3171692250	0.89204345
Felipe	1.2909219487	0.961533489
Siervo	1.3220570751	0.909601983
Sebastian	1.3363214675	0.874617644
Clara	1.2286089	0.740657032
Pipe	1.106500553	0.674386597
Laura	1.5059806	0.82225119

Figura 2. Tabla de excel.

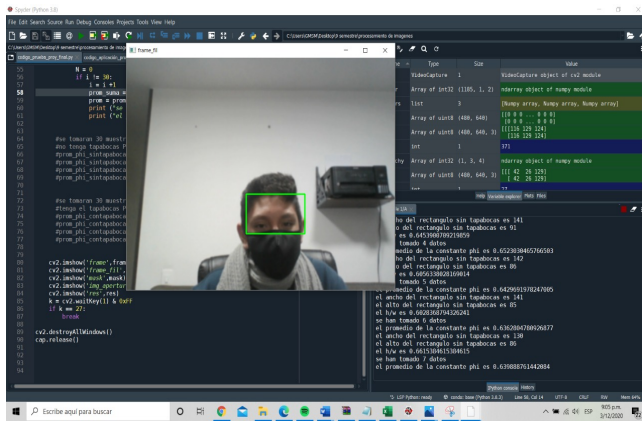


Figura 3. Toma del dato de phi con tapabocas.

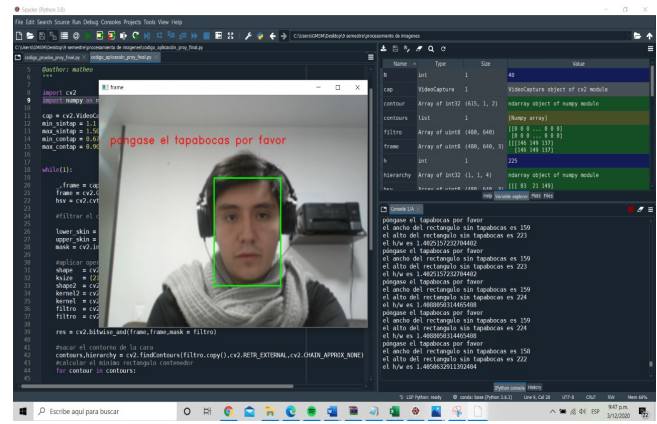


Figura 5. Matheo sin tapabocas.

En la figura 7 se puede observar el momento de la toma de datos, en este caso con tapabocas; además, en la consola, se puede observar el proceso de promedio de la constante phi

V-B. Segunda parte: Cálculos estadísticos de los datos

Con los datos obtenidos, se calcularon las variables estadísticas mencionadas anteriormente, a continuación se muestra el resultado

promedio	1.3024423481	0.8612212738
varianza	0.010949438	0.009539855
minimo	1.1065005530	0.6743865970
maximo	1.5059806000	0.9615334892

Figura 4. Variables estadísticas.

Se puede observar que el conjunto de datos tenía una varianza pequeña, indicando que la constante phi variaba relativamente poco en esa población para los 2 casos (con tapabocas y sin tapabocas)

V-C. Tercera parte: Toma de decisión

Con los valores máximo y mínimo calculados, se obtuvieron los umbrales para los cuales una persona tendría o no el tapabocas, para esta prueba hicimos parte nosotros (Natalia Guevara y Matheo Bermúdez), ya que no habíamos participado en la toma de datos; con esto nos aseguramos que no existiera un sesgo a la hora de probar el sistema, a continuación se muestran los resultados

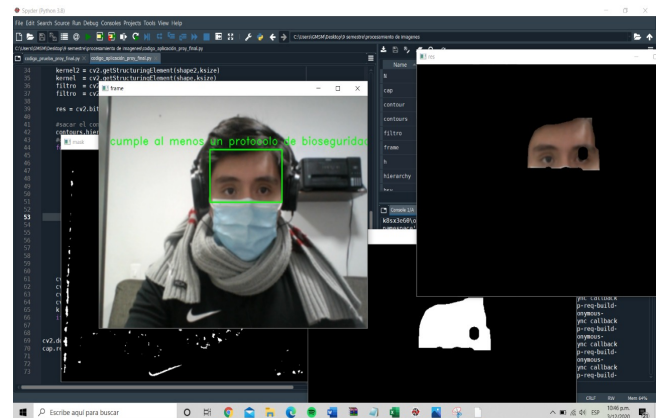


Figura 6. Matheo con tapabocas.

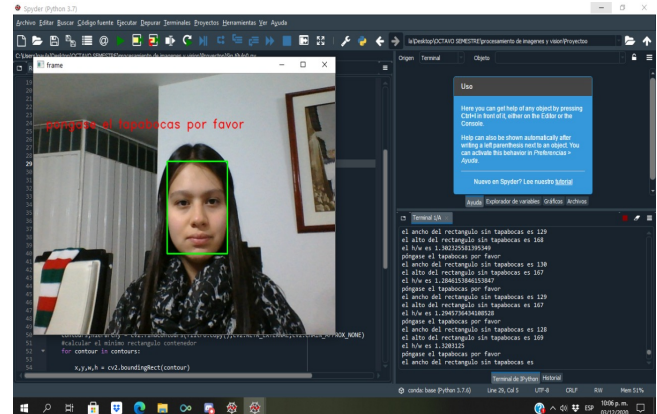


Figura 7. Natalia sin tapabocas.

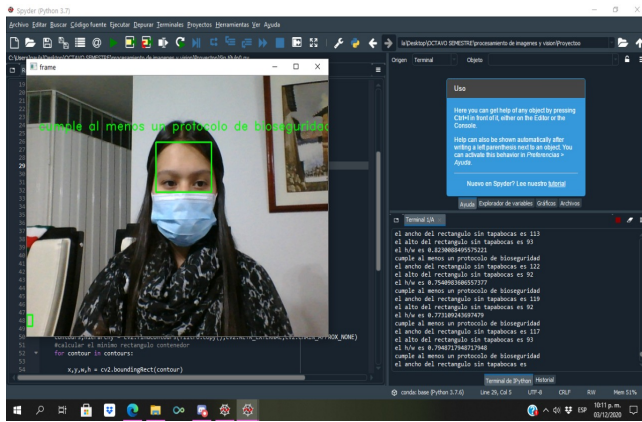


Figura 8. Natalia con tapabocas.

V-D. Código implementado

Para implementar el diseño anterior se dividió el código en 2 partes, la primera parte se encargaba solo de tomar los datos de phi; y la segunda parte se encargaba de imprimir los 2 posibles mensajes de acuerdo al valor de phi calculado para una persona. a continuación se explican las partes relevantes del código de la primera parte

```
while(1):
    _frame = cap.read()
    frame = cv2.GaussianBlur(frame,(7,7),0) #aplicar filtro para quitar ruido

    hsv = cv2.cvtColor(frame,cv2.COLOR_BGR2HSV) #pasar a HSV

    #filtrar el color de piel https://es.wikipedia.org/wiki/Carne\_\(color\)

    lower_skin = np.array([0,10,60])
    upper_skin = np.array([20,150,255]) #umbrales para el color piel
    mask = cv2.inRange(hsv,lower_skin,upper_skin) #para mostrar solo el rostro
```

Figura 9. selección de valores de umbral

En esta parte del código empieza el ciclo while y se escogen los valores de umbralización de acuerdo al rango de colores piel de las personas, se encontró que esos valores eran los mas adecuados para dejar pasar una gran variedad de rostros, so pena de dejar pasar otros objetos con colores parecidos al color piel

```
#aplicar operadores morfológicos
shape = cv2.MORPH_RECT
ksize = (21,21)
shape2 = cv2.MORPH_ELLIPSE
kernel2 = cv2.getStructuringElement(shape2,ksize)
kernel = cv2.getStructuringElement(shape,ksize)
filtro = cv2.morphologyEx(mask,cv2.MORPH_OPEN,kernel)
filtro = cv2.morphologyEx(filtro,cv2.MORPH_CLOSE,kernel2)

res = cv2.bitwise_and(frame,frame,mask = filtro)

#sacar el contorno de la cara
contours,hierarchy = cv2.findContours(filtro.copy(),cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_NONE)
```

Figura 10. procesamiento y generación del contorno del rostro

En la figura anterior se puede observar que se aplican una serie de operadores morfológicos con kernel grande para eliminar los objetos que no se hayan podido filtrar con la

umbralización y así mismo no afectar al reconocimiento del rostro; después, se procede a encontrar el contorno del rostro; es normal que aun con estos filtros, existan objetos que no se hayan podido filtrar, debido a las condiciones de iluminación o simplemente por las condiciones del fondo.

```
N = N + 1
#calcular el minimo rectangulo contenedor de la cara
for contour in contours:

    x,y,w,h = cv2.boundingRect(contour)
    cv2.rectangle(frame,(x,y),(x+w,y+h),(0,255,0),2)

#esta parte del codigo se encarga de recoger datos y promediarlos
if N == 70:
    print ("el ancho del rectangulo sin tapabocas es",w)
    print ("el alto del rectangulo sin tapabocas es",h)
    phi = h/w
    print ("el h/w es",phi)
    N = 0
    if i != 30:
        i = i +1
        prom_suma = (prom_suma + phi)
        prom = prom_suma/i
        print ("se han tomado", i, "datos")
        print ("el promedio de la constante phi es", prom)
```

Figura 11. promedio de la constante phi

Por ultimo, esta parte del código se encarga de encerrar el rostro en un rectángulo y sacar las medidas del alto y ancho del mismo, después, la variable N se encarga de controlar cada cuanto se toma el dato de la constante phi sobre la persona y cada cuanto se promedia; es decir, cada vez que N sea igual a 70 se reinicia esta variable, se toma el dato de phi y se promedia con los otros datos anteriores hasta que el contador i llegue a 30; en este caso ya se tendría el promedio de phi (dependiendo de si la persona tiene o no tiene tapabocas).

Para la parte 2 del código, la primera parte que se encarga de umbralizar, procesar y encontrar los contornos es la misma, pero después de ese proceso el programa se encarga de tomar la decisión final, como se muestra en la siguiente figura

```
for contour in contours:

    x,y,w,h = cv2.boundingRect(contour)
    cv2.rectangle(frame,(x,y),(x+w,y+h),(0,255,0),2)
    phi = h/w
    print ("el h/w es",phi)

#clasificador con los valores max y min de phi
if phi <= max_sintap and phi >= min_sintap:
    cv2.putText(frame,"pongase el tapabocas por favor",(25,10),2,0.8,(0,0,255),1,cv2.LINE_AA)
    print("pongase el tapabocas por favor")
else:
    if phi <= max_contap and phi >= min_contap:
        cv2.putText(frame,"cumple al menos un protocolo de bioseguridad",(25,100),2,0.8,(0,255,255),1,cv2.LINE_AA)
        print("cumple al menos un protocolo de bioseguridad")
```

Figura 12. clasificador

En este caso, se saca el valor en tiempo real de phi, y este valor es evaluado constantemente por esos 2 condicionales, cuyos valores fueron sacados de los cálculos estadísticos de la parte 2 de todo el sistema (los valores máximo y mínimo para

los 2 casos, con tapabocas o sin tapabocas); sin embargo a estos valores se les sumo un pequeño offset de 0.1 ya que con la poca cantidad de datos recolectados, el clasificador seguía siendo muy sesgado a la población a la cual fueron tomados los datos.

VI. CONCLUSIONES

- Si bien el programa mostraba una buena precisión a la hora de predecir si una persona tenía tapabocas, tenía el problema de detectar objetos con color parecido al color piel, haciendo que el rectángulo contenedor también encerrara, creando un error grande en la medida; este problema se pudo solucionar de manera aceptable aplicando operaciones morfológicas con kernels grandes.
- Las condiciones de iluminación afectaban la operación del sistema, pues en superficies lambertianas, la luz del medio se reflejaba, cambiando el color de objetos como repisas o incluso el pelo, esto generaba una fuente de ruido que afectaba la predicción del programa
- Debido a la poca cantidad de datos tomados, no se tenía suficiente criterio para evaluar a cualquier población debido a la diferencia de tamaños en el rostro, sin embargo, agregando un offset de 0,1 a los valores mínimo y máximo de ϕ , se pudo ampliar el rango de medida para las personas que no fueron evaluadas
- El programa tuvo una muy buena precisión para diferentes distancias de la cara respecto a la cámara, sin embargo, para distancias lejanas, el programa no reconocía los rostros, ya que el valor de ϕ para esas distancias se encontraba justo en el rango entre el valor máximo de ϕ_{CT} y el valor mínimo de ϕ_{ST}

VII. REFERENCIAS

[1] Apuntes de la clase de procesamiento de imágenes y visión

[2] [Online]. Available: [https://es.wikipedia.org/wiki/Carne\(color\)](https://es.wikipedia.org/wiki/Carne(color))