PROCESAMIENTO DE IMÁGENES PARA DETECCIÓN FACIAL

El Arte de la Programación - TC1001.1S

Jesús Palomino Hurtado A01638492

León E. García Pérez A00573074

Diego Martínez Medrano A01634562

C. Alexis Sahagún Navarro A01637885

INTRODUCCIÓN

HOMEBREW

¿QUÉ ES? : Es un Sistema de Gestión de Paquetes que simplifica la Instalación, Actualización y Eliminación de los Programas en los Sistemas Operativos Mac OS de Apple y GNU/Linux.

¿CUÁL ES SU USO DENTRO DEL PRESENTE PROYECTO? : A través del gestor se instala el paquete cmake, para compilar todos los archivos cpp.

DLIB

¿QUÉ ES? : Es una Biblioteca de Software Multiplataforma de propósito general escrita en C++. Su diseño está fuertemente influencia por ideas de diseño por contrato e Ingeniería de Software basada en componentes.

¿CUÁL ES SU USO DENTRO DEL PRESENTE PROYECTO? : La librería se usa con el fin de poder emplear sus algoritmos de detección.

XQUARTZ

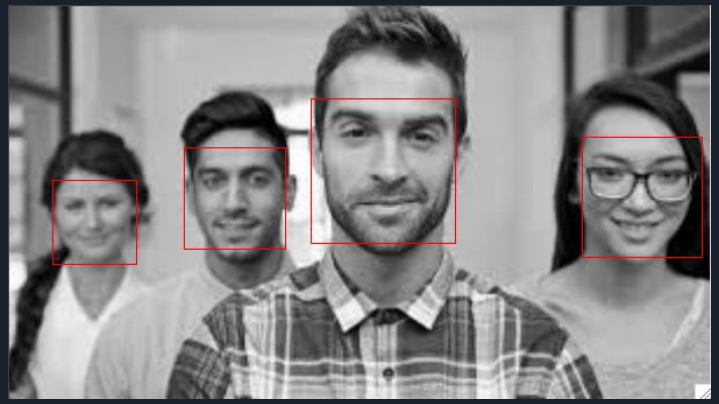
¿QUÉ ES? : Es una versión de código abierto del servidor X. Org X, un componente del sistema X Windows que se ejecuta en Mac OS. Permite que las aplicaciones multiplataforma que usan X11 para la GUI se ejecuten en Mac OS, muchas de las cuales no están diseñadas específicamente para Mac OS.

¿CUÁL ES SU USO DENTRO DEL PRESENTE PROYECTO? : Mostrar las salidas de la ejecución del proyecto.

EL CÓDIGO Y SU FUNCIONAMIENTO

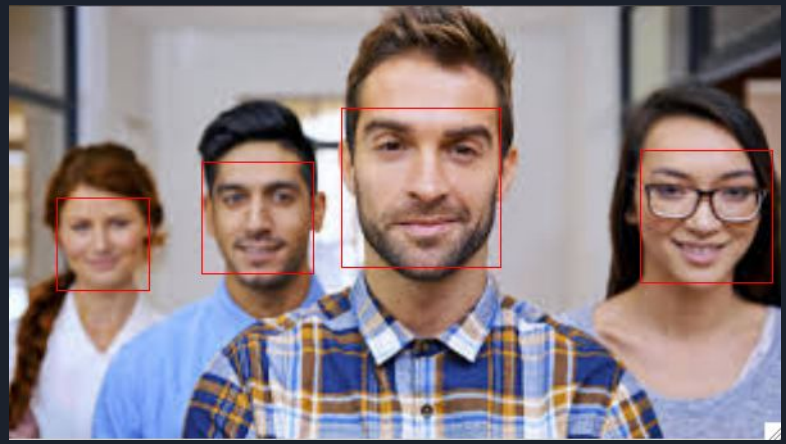
Funcionamiento General del Código

Posterior a la Instalación del Paquete CMake por medio de Homebrew, y de la instalación del Paquete de Herramientas de DLIB. Se procede a una compilación de todos los códigos, archivos cpp, dentro de DLIB. Para posteriormente emplear el ejecutable junto a un archivo de imagen jpg, para mostrar la imagen en blanco y negro con la zona de la cara enmarcada en rojo; estando aquí presente la Deteción Facial.



Primera Modificación (Cada modificación está señalada por su orden dentro del repositorio en GitHUB)

Se crea un array de pixeles de color, RGB, para procesar la imagen, y así poder hacer el display de ésta al final, conservando los colores originales, en vez de la que es originalmente: en blanco y negro.



Segunda Modificación (Cada modificación está señalada por su orden dentro del repositorio en GitHUB)

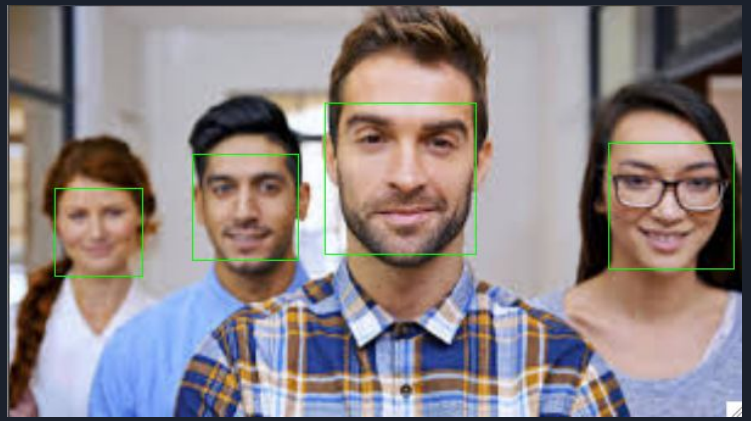
Se usan los métodos de la clase Rectángulo, dentro de los Archivos de DLIB, para poder obtener las coordenadas inferiores y superiores de ambos bordes de cada Rectángulo, que representa la zona donde hubo detección facial. Se realiza una impresión.

```
Procesando imagen personas.jpg

Cara 1 detectada: Top: 121      Left: 150      Bottom: 208     Right: 237
Cara 2 detectada: Top: 79       Left: 259      Bottom: 203     Right: 383
Cara 3 detectada: Top: 112      Left: 492      Bottom: 215     Right: 595
Cara 4 detectada: Top: 149      Left: 37       Bottom: 221     Right: 109
```

Tercera Modificación (Cada modificación está señalada por su orden dentro del repositorio en GitHUB)

Para el número total de caras, del Vector que contenía los bordes, con la función size: se imprime la cantidad de rectángulos del vector, siendo la cantidad total de caras detectadas. A su vez, se realiza un cambio de color en el rectángulo que enmarca las caras: en la función del Overlay con los Bordes, se cambia el parámetro del color RGB.



Cuarta Modificación (Cada modificación está señalada por su orden dentro del repositorio en GitHUB)

Se detecta si se manda más de una imagen desde la terminal; cada vez que se imprime una imagen, solicita un enter para continuar con la siguiente imagen, tomando la consideración de impresión de formato en al terminal para hacer la diferenciación, mostrando un mensaje de continuar. En la última imagen, solamente solicita un enter para finiquitar.

```
Número de caras detectadas: 4

--------------------------------------------------------------------------------
Presiona enter para continuar con la siguiente imagen...


Procesando imagen personas2.jpg

Cara 1 detectada: Top: 141      Left: 323      Bottom: 227     Right: 410
Cara 2 detectada: Top: 100      Left: 112      Bottom: 204     Right: 215
Cara 3 detectada: Top: 204      Left: 227      Bottom: 307     Right: 330
Cara 4 detectada: Top: 26       Left: 227      Bottom: 112     Right: 314
Cara 5 detectada: Top: 215      Left: 54       Bottom: 319     Right: 158
Cara 6 detectada: Top: 192      Left: 400      Bottom: 296     Right: 503
Cara 7 detectada: Top: 83       Left: 467      Bottom: 170     Right: 553

Número de caras detectadas: 7

--------------------------------------------------------------------------------
Presiona enter para continuar con la siguiente imagen...
```

DEMOSTRACIÓN DEL CÓDIGO Y SU FUNCIONAMIENTO

GRACIAS