



Tecnológico Nacional de México

Instituto Tecnológico de Tijuana



Subdirección académica
Ingeniería Eléctrica y Electrónica

Carrera: *Maestría en Ciencias de la Ingeniería*
Materia: *Visión Artificial*
Profesor: *Dr. Eddie Helbert Clemente Torres*

Resumen de habilidades desarrolladas

Fecha: 1-Marzo-2016

Alumnos:	<u>Zúñiga Alvarez Francisco Fernando</u>	No.Control: <u>G04650183</u>
	<u>Pozos Flores Angel Manrique</u>	No.Control: <u>M07211505</u>

Examen

Utilizar las técnicas vistas en clase como: histogramas, operaciones entre imágenes, binarización por threshold, morfología matemática, etc. Para que dada una serie de imágenes y delimitando el área de la boca se enfatice la región de los labios.

Se sugiere tomar en cuenta fotos bien definidas, de personas de frente y sin variación de escala y rotación.

Para cumplir el objetivo desarrollamos el siguiente código.

```
#include "opencv2/objdetect/objdetect.hpp"
#include "opencv2/highgui/highgui.hpp"
#include "opencv2/imgproc/imgproc.hpp"
#include <iostream>
#include <stdio.h>

using namespace std;
using namespace cv;

int main()
{
    Mat img, gray, equ;
    img = imread("IMAGE.jpg", CV_LOAD_IMAGE_COLOR);
    namedWindow("Original Image", 1);
    imshow("Original Image", img);

    // Archivos .xml
    CascadeClassifier FaceCascade, SmileCascade;
    FaceCascade.load("haarcascade_frontalface.xml");
    SmileCascade.load("haarcascade_mouth.xml");

    cvtColor(img, gray, CV_BGR2GRAY);
    equalizeHist(gray, equ);

    // Detección de la cara
    std::vector<Rect> faces;
    FaceCascade.detectMultiScale(equ, faces, 1.1, 2, 0 | CV_HAAR_SCALE_IMAGE, Size(30, 30));

    for (int i = 0; i < faces.size(); i++)
    {
        // Dibujar un círculo en la cara detectada
        Point center(faces[i].x + faces[i].width*0.5, faces[i].y + faces[i].height*0.5);
        ellipse(equ, center, Size(faces[i].width*0.5, faces[i].height*0.5), 0, 0, 360,
        Scalar(255, 0, 255), 4, 8, 0);

        // Detectar los labios
        Mat FaceROI = equ(faces[i]); //Region of Interest (Face)
        std::vector<Rect> smiles;
        SmileCascade.detectMultiScale(FaceROI, smiles, 1.1, 2, 0 | CV_HAAR_SCALE_IMAGE,
        Size(45, 50));

        for (size_t j = 0; j < smiles.size(); j++)
        {
            Mat LipsROI = FaceROI(smiles[j]); //Región de Interes (Labios)
            Mat TreshLipsROI;

            int scale = 1;
            int delta = 0;
            int ddepth = CV_16S;

            // Generar un gradiente X y Y
            Mat grad, grad_x, grad_y, abs_grad_x, abs_grad_y;

            // Gradiente X
            Sobel(LipsROI, grad_x, ddepth, 1, 0, 3, scale, delta, BORDER_DEFAULT);
            convertScaleAbs(grad_x, abs_grad_x);
```

```

        // Gradiente Y
        Sobel(LipsROI, grad_y, ddepth, 0, 1, 3, scale, delta, BORDER_DEFAULT);
        convertScaleAbs(grad_y, abs_grad_y);

        // Gradiente Total
        addWeighted(abs_grad_x, 0.5, abs_grad_y, 0.5, 0, LipsROI);
        cv::threshold(LipsROI, TreshLipsROI, 25, 255, cv::THRESH_BINARY);
        SmileCascade.detectMultiScale(TreshLipsROI, smiles, 1.1, 2, 0 |
CV_HAAR_SCALE_IMAGE, Size(45, 50));

        namedWindow("ROI", WINDOW_AUTOSIZE);
        imshow("ROI", TreshLipsROI);

    }

}

namedWindow("Face & Mouth", WINDOW_AUTOSIZE);
imshow("Face & Mouth", equ);

waitKey(0);
return 0;
}

```

El código se probó con algunas imágenes tomadas de la página de **FEI Face Database**, además de las fotografías de 2 estudiantes del posgrado, para comprobar su funcionamiento con diferentes fondos, condiciones de iluminación, rotación, etc., estos resultados se muestran en las Figuras



Figura 1. Imagen original.



Figura 2. Imagen original.



Figura 3. Cara detectada (circulo), boca (Recuadro).



Figura 4. Cara detectada (circulo), boca (Recuadro).



Figura 5. Región de interés (labios) extraída y binarizada.



Figura 6. Región de interés (labios) extraída y binarizada.



Figura 7. Imagen original.

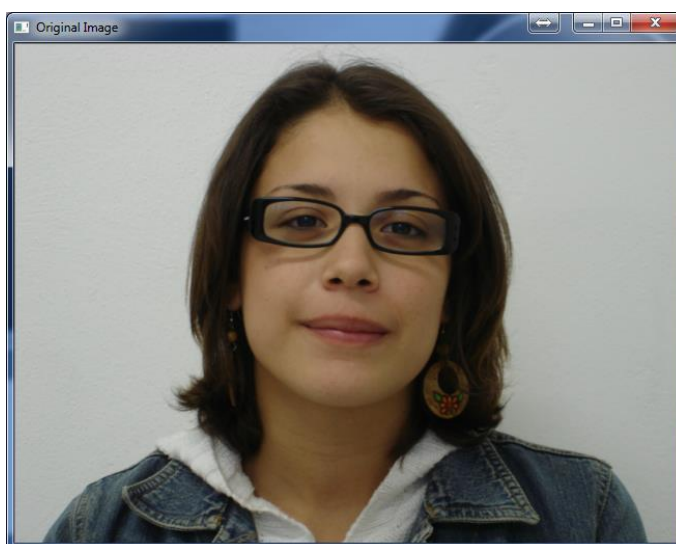


Figura 8. Imagen original.

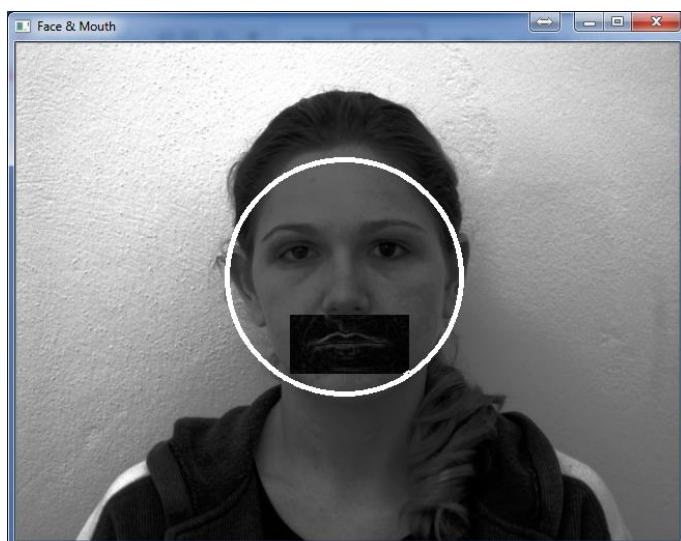


Figura 9. Cara detectada (circulo), boca (Recuadro).



Figura 10. Cara detectada (circulo), boca (recuadro).



Figura 11. Región de interés (labios) extraída y binarizada

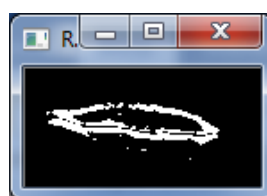


Figura 12. Región de interés (labios) extraída y binarizada.

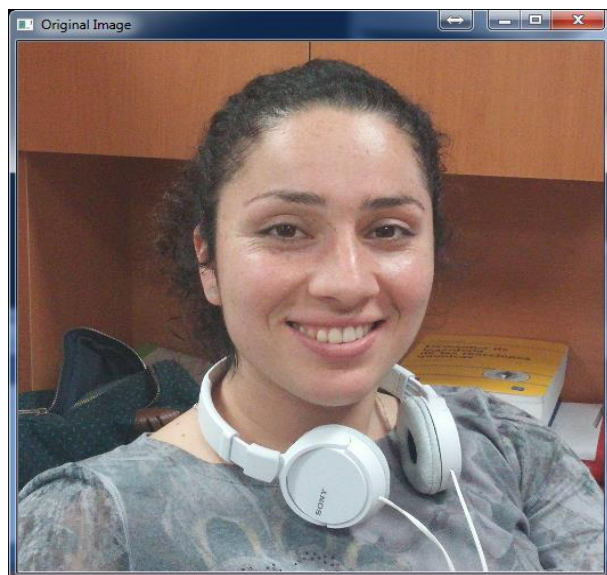


Figura 13. Imagen original.

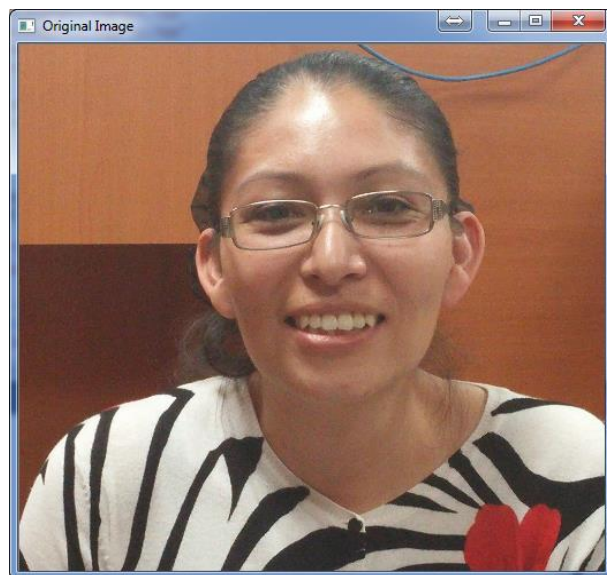


Figura 14. Imagen original.

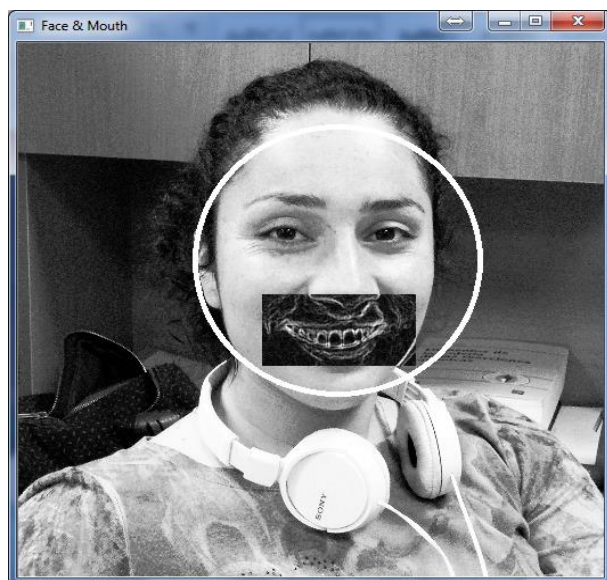


Figura 15. Cara detectada (circulo), boca (Recuadro).



Figura 16. Cara detectada (circulo), boca (Recuadro).

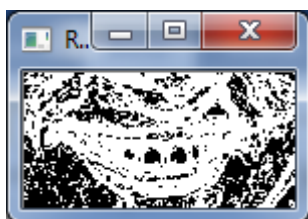


Figura 15. Región de interés (labios) extraída y binarizada.



Figura 18. Región de interés (labios) extraída y binarizada.

Conclusiones

El presente algoritmo desarrollado utiliza muchas de las funciones vistas en clase y desarrolladas en las tareas, como el manejo de la escala de grises, la ecualización del histograma, filtro Sobel para la detección de contornos y thresholding binario, estas funciones fueron aplicadas con el fin de poder enfatizar la región de los labios.

Primeramente, cargamos una imagen tomada de una base de datos [3] a la cual se le hará el procesamiento, para ello se convierte a escala de grises y se ecualiza su histograma, una vez ecualizada ya está lista para poder procesarse de una mejor manera, utilizando los archivos de haarcascade para cara y boca [4] se generan vectores que contendrán estas regiones de interés (ROI), donde primeramente detectamos lo que es la cara y la encerramos en un círculo y una vez que ha sido detectada esta región de interés guardamos esta matriz de valores y la utilizamos para detectar la segunda región de interés la cual es la boca.

Una vez detectada la región de la boca, aplicamos un thresholding para binarizarla y utilizando la función de Sobel definimos sus contornos tomando los gradientes correspondientes para el eje x y el eje y, después sacamos el gradiente total de la imagen y lo imprimimos en una ventana.

Finalmente imprimimos la imagen original junto con la procesada y en otra ventana mostrando la ROI, cumpliendo con ello el fin de enfatizar la región de los labios para una imagen n.

Se tomaron 2 fotografías de compañeras del posgrado con el fin de observar que tan bueno era el algoritmo para poder reconocer y enfatizar los labios la cual resulto 100% positiva, observamos también que existen imágenes que al no tener una adecuada iluminación el algoritmo puede equivocarse e identificar mas puntos como nuestra ROI, por ello decidimos que antes de procesarla fuera corregida por ecualización del histograma y después de ser reconocido el rostro, aplicar una binarización a la ROI interna (boca) para así hacer el énfasis de los labios con el fin de mejorar la precisión del algoritmo.

Referencias

- [1] «Cascade Classifier,» [En línea]. Disponible en: <http://tinyurl.com/jf25mex>
[Último acceso: 27 febrero 2016].
- [2] «Face Detection Using Haar-Cascade,» [En línea]. Disponible en: <http://tinyurl.com/gon58q7>
[Último acceso: 27 febrero 2016].
- [3] «FEI Face Database,» [En línea]. Disponible en: <http://tinyurl.com/h6nbsma>
[Último acceso: 25 febrero 2016].
- [4] «Haar Cascades,» [En línea]. Disponible en: <http://tinyurl.com/jgdy69d>
[Último acceso: 25 febrero 2016].
- [5] «Sobel Derivatives,» [En línea]. Disponible en: <http://tinyurl.com/jdzpxzo>
[Último acceso: 28 febrero 2016].
- [6] Learning Image Processing with OpenCV, Gloria B., Jose L., Ismael S., Open Source.
- [7] OpenCV By Example, Prateek J., David M., Vinicius G., Open Source.