

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería



**Sistema de detección de rostro y reconocimiento de gestos
para robot animatrónico**

Trabajo de graduación presentado por Luis Eduardo Ruano Argueta
para optar al grado académico de Licenciado en Ingeniería Electrónica

Guatemala,

2019

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería



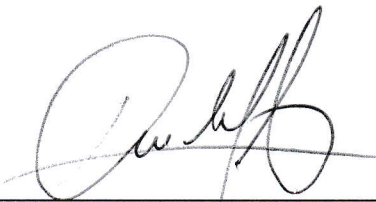
**Sistema de detección de rostro y reconocimiento de gestos
para robot animatrónico**

Trabajo de graduación presentado por Luis Eduardo Ruano Argueta
para optar al grado académico de Licenciado en Ingeniería Electrónica

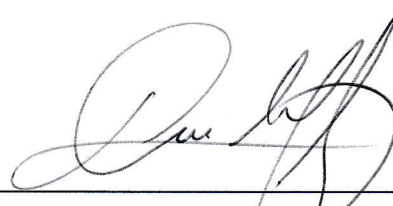
Guatemala,


2019

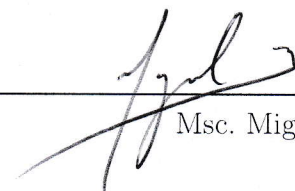
Vo.Bo.:

(f) 
Ing. Luis Pedro Montenegro

Tribunal Examinador:

(f) 
Ing. Luis Pedro Montenegro

(f) 
MSc. Carlos Esquit

(f) 
Msc. Miguel Zea

Fecha de aprobación: Guatemala, 05 de diciembre de 2018.

Como estudiante de Ingeniería Electrónica se obtuvo una experiencia gratificante el poder implementar un sistema de detección de rostro y reconocimiento de expresiones. Ya que a pesar de tener conocimientos avanzados en el área de programación, se tuvo que realizar una revisión bibliográfica extensa acerca de los algoritmos para clasificación y predicción de distribuciones. Esta investigación me abre las puertas a mundos, que hasta el momento de haber iniciado la investigación, permanecían desconocidos.

Estos nuevos mundos de la detección de objetos y procesamiento de imágenes, son temas interesantes. Los cuales al haber realizado un trabajo de graduación con base en la detección de rostro y reconocimiento de expresiones faciales se tiene un buen conocimiento acerca de los temas. Oportunidades como el desarrollo de nuevas tecnologías de carros autónomos y procesamiento de imágenes para procedimientos médicos son dos de las áreas de especialización que ahora vienen a la mente en cuanto a la aplicación de los conceptos desarrollados.

Prefacio	v
Lista de figuras	XIII
Lista de cuadros	XVI
Resumen	XVIII
Abstract	XIX
1. Introducción	1
2. Antecedentes	3
2.1. Detección rostro	3
2.2. Reconocimiento de expresiones faciales	4
2.3. Módulo reconocimiento de gestos proyecto animatrónico UVG	5
3. Justificación	7
4. Objetivos	9
4.1. Objetivo general	9
4.2. Objetivos específicos	9
5. Alcance	11
6. Marco teórico	13
6.1. Aspecto psicológico de las expresiones	13
6.1.1. Universalidad de expresiones	13
6.1.2. Identificadores de expresiones faciales	14
6.2. Clasificador <i>Haar cascade</i>	14
6.2.1. Viola-Jones	15
6.3. Marcas de cara	17
6.3.1. Modelos de apariencia activos	17
6.4. Reconocimiento facial	19

6.4.1.	PCA, <i>Eigen</i> rostros	19
6.4.2.	FISHER FACE	21
6.4.3.	SVM	23
6.5.	Logitech c920	24
6.6.	Sci-Kit Learn	25
6.7.	Validación cruzada	26
6.8.	Matrices de confusión	27
6.8.1.	Exactitud	27
6.8.2.	Sensibilidad	27
6.8.3.	Precisión	28
7.	Metodología	29
7.1.	Obtención de datos	29
7.1.1.	Entrenamientos	29
7.1.2.	Base de datos	30
7.2.	Pruebas	31
7.2.1.	Prueba ambiente abierto	32
7.2.2.	Prueba ambiente cerrado	34
8.	Detección de rostro	37
8.1.	Pruebas Plaza Paiz	37
8.1.1.	Aplicación histograma	37
8.1.2.	Movimiento laterales	39
8.1.3.	Movimientos agresivos	42
8.1.4.	Obstrucción del rostro	43
8.1.5.	Distancia máxima	43
8.2.	Prueba ambiente controlado	45
8.2.1.	Rotación ángulo de visión	46
8.2.2.	Distancia máxima	47
8.3.	Pruebas iluminación menor a 10 lx.	50
8.4.	Pruebas detección de rostro en bases de datos	51
9.	Pruebas específicas algoritmo <i>Fish Face</i>	55
9.1.	Aplicación del algoritmo PCA	55
9.1.1.	<i>Eigen</i> rostros	55
9.1.2.	Reconstrucción imágenes por <i>Eigen rostros</i>	58
9.2.	Aplicación del algoritmo LDA	60
9.2.1.	<i>Fisher rostros</i>	61
9.2.2.	Reconstrucción de imágenes por medio de <i>Fisher rostros</i>	63
9.2.3.	Distribución datos de expresiones a partir de aplicación PCA y LDA	65
10.	Extracción de marcas de cara	71
11.	Reconocimiento de expresiones faciales	75
11.1.	<i>Fisher Face</i>	75
11.1.1.	Base de datos <i>Cohn-Kanade</i>	75
11.1.2.	Base de datos UVG	76
11.1.3.	Base de datos UVG 2 m	76
11.1.4.	Base de datos Ambiente 2	77

11.1.5. Base de datos Personalizada	77
11.2. SVM	80
11.2.1. Base de datos <i>Cohn-Kanade</i>	80
11.2.2. Base de Datos UVG	81
11.2.3. Base de datos UVG 2 m	82
11.2.4. Base de datos Ambiente 2	82
11.2.5. Base de datos Personalizada	83
12. Conclusiones	89
13. Recomendaciones	91
14. Bibliografía	93
15. Anexos	97
15.1. Imágenes bases de datos	97
15.2. Manual	100
15.2.1. Hardware necesario	100
15.2.2. Software necesario	100
15.3. Implementación	100
16. Glosario	105

Lista de figuras

1.	Detección múltiples rostros algoritmos <i>Haar cascade</i>	4
2.	Reconocimiento de rostro por medio de localización de marcas de cara	5
6.	Reconocimiento gesto de saludo	5
3.	Localización de marcas de cara en ovejas	6
4.	Ubicación de las cámaras para la base de datos Multi-PIE	6
5.	Imágenes obtenidas base de datos Multi-PIE	6
7.	Set de rectángulos de características <i>Haar cascade</i>	15
8.	Generación de marcas de rostros por modelos de apariencia activos	19
9.	Set de imágenes iniciales	20
10.	Imágenes reconstruidas por vectores propios	21
11.	Set de imágenes procesamiento Eigen Caras	23
12.	Set de imágenes procesamiento <i>Fisher Face</i>	23
13.	Clasificación binaria por medio de SVM	24
14.	Cámara web <i>Logitech C920</i>	25
15.	Vista de lado obtención bases de datos	32
16.	Vista de planta pruebas ambiente abierto	33
17.	Aplicación de la técnica ecualización del histograma en Plaza Paiz	38
18.	Plano de posicionamiento movimientos laterales	39
19.	Movimiento lateral 1 m algoritmo <i>Haar cascade</i>	40
20.	Movimiento lateral 1m algoritmo marcas de cara	40
21.	Movimiento lateral 1.5 m algoritmo <i>Haar cascade</i>	40
22.	Movimiento lateral 1.5 m algoritmo marcas de cara	41
23.	Movimiento lateral 2 m algoritmo <i>Haar cascade</i>	41
24.	Movimiento lateral 2 m algoritmo marcas de cara	41
25.	Distancias máximas, algoritmo <i>Haar cascade</i> Plaza Paiz	43
26.	Distancias máximas algoritmo de marcas de cara Plaza Paiz	44
27.	Plano de detección Plaza Paiz	45
29.	Detección rostros con rotación <i>Ambiente 2</i>	46
28.	Detección rostros con rotación ambiente UVG	46
30.	Distancias máximas algoritmo <i>Haar cascade</i> para ambiente UVG	47

31.	Distancias máximas algoritmo <i>Haar cascade</i> para <i>Ambiente 2</i>	48
32.	Distancias máximas algoritmo marcas de cara para ambiente UVG	48
33.	Distancias máximas algoritmo marcas de cara para <i>Ambiente 2</i>	49
34.	Detección algoritmo marcas de cara para ambiente oscuro	50
35.	Imágenes con detección de rostros para bases de datos	51
36.	Imágenes con reconocimientos incorrectos de rostro base de datos UVG	52
37.	<i>Eigen rostros</i> base de datos <i>Cohn-Kanade</i>	56
38.	<i>Eigen rostros</i> base de datos <i>UVG</i>	56
39.	<i>Eigen rostros</i> base de datos <i>Ambiente 2</i>	57
40.	<i>Eigen rostros</i> base de datos <i>Personalizada</i>	57
41.	Reconstrucción de imagen por <i>Eigen rostros</i> de expresión neutral para base de datos <i>Cohn-Kanade</i>	58
42.	Reconstrucción de imagen por <i>Eigen rostros</i> de expresión felicidad para base de datos <i>UVG</i>	59
43.	Reconstrucción de imagen por <i>Eigen rostros</i> de expresión felicidad para base de datos <i>Ambiente 2</i>	59
44.	Reconstrucción de imagen por <i>Eigen rostros</i> de expresión felicidad para base de datos <i>Personalizada</i>	60
45.	<i>Fisher rostros</i> base de datos <i>Cohn-Kanade</i>	61
46.	<i>Fisher rostros</i> base de datos <i>UVG</i>	61
47.	<i>Fisher rostros</i> base de datos <i>Ambiente 2</i>	62
48.	<i>Fisher rostros</i> base de datos <i>Personalizada</i>	62
49.	Reconstrucción de imagen por medio de <i>Fisher rostros</i> para base de datos <i>Cohn-Kanade</i>	63
50.	Reconstrucción de imagen por medio de <i>Fisher rostros</i> para base de datos <i>UVG</i>	64
51.	Reconstrucción de imagen por medio de <i>Fisher rostros</i> para base de datos <i>Ambiente 2</i>	64
52.	Reconstrucción de imagen por medio de <i>Fisher rostros</i> para base de datos <i>Personalizada</i>	65
53.	Distribución de expresiones para base de datos <i>Cohn-Kanade</i> utilizando 50 componentes	66
54.	Distribución de expresiones para base de datos <i>Cohn-Kanade</i> utilizando 100 componentes	66
55.	Distribución de expresiones para base de datos <i>UVG</i> utilizando 50 componentes	67
56.	Distribución de expresiones para base de datos <i>UVG</i> utilizando 100 componentes	67
57.	Distribución de expresiones para base de datos <i>Ambiente 2</i> utilizando 50 componentes	68
58.	Distribución de expresiones para base de datos <i>Ambiente 2</i> utilizando 100 componentes	68
59.	Distribución de expresiones para base de datos <i>Personalizada</i> utilizando 50 componentes	69
60.	Distribución de expresiones para base de datos <i>Personalizada</i> utilizando 100 componentes	69
61.	Localización de las marcas de cara	71
62.	Marcas de cara expresión neutral	72

63.	Marcas de cara expresión feliz	72
64.	Marcas de cara expresión enojo	72
65.	Marcas de cara expresión sorpresa	73
66.	Marcas de cara expresión disgusto	73
67.	Imagen con reconocimiento incorrecto de disgusto	78
68.	Imagen con reconocimiento incorrecto de sorpresa	85
69.	Comparación porcentajes de detección algoritmos SVM y <i>Fisher Face</i> para 3 expresiones	86
70.	Comparación porcentajes de detección algoritmos SVM y <i>Fisher Face</i> para 5 expresiones	86
71.	Rostros de neutralidad	97
72.	Rostros de felicidad	98
73.	Rostros de enojo	98
74.	Rostros de sorpresa	99
75.	Rostros de disgusto	99
76.	Salida de datos algoritmo <i>Fisher Face</i>	102
77.	Salida de datos algoritmo <i>SVM</i>	104

Lista de cuadros

1.	Matriz de confusión	27
2.	Matriz de confusión, obteniendo exactitud	28
4.	Matriz de confusión, obteniendo precisión	28
3.	Matriz de confusión, obteniendo sensibilidad	28
5.	Matriz de confusión implementada	36
6.	Porcentajes de detección de rostro Plaza Paiz	38
7.	Porcentajes de detección para movimientos agresivos	42
8.	Porcentajes de detección para obstrucción del rostro	43
9.	Porcentajes de detección de rostros con un ángulo de rotación de 45° para ambiente Universidad del Valle	47
10.	Porcentajes de detección de rostros con un ángulo de rotación de 45° para <i>Ambiente 2</i>	47
11.	Porcentajes de detección rostros de bases de datos	51
12.	Matriz de confusión base de datos <i>Cohn-Kanade</i> para 3 expresiones con algoritmo <i>Fisher Face</i>	75
13.	Matriz de confusión base de datos <i>Cohn-Kanade</i> para 5 expresiones con algoritmo <i>Fisher Face</i>	76
14.	Matriz de confusión base de datos <i>UVG</i> para 3 expresiones con algoritmo <i>Fisher Face</i>	76
15.	Matriz de confusión base de datos <i>UVG</i> para 5 expresiones con algoritmo <i>Fisher Face</i>	76
16.	Matriz de confusión base de datos <i>UVG 2 m</i> para 3 expresiones con algoritmo <i>Fisher Face</i>	76
17.	Matriz de confusión base de datos <i>UVG 2 m</i> para 5 expresiones con algoritmo <i>Fisher Face</i>	77
18.	Matriz de confusión base de datos <i>Ambiente 2</i> para 3 expresiones con algoritmo <i>Fisher Face</i>	77
19.	Matriz de confusión base de datos <i>Ambiente 2</i> para 5 expresiones con algoritmo <i>Fisher Face</i>	77

20.	Matriz de confusión base de datos <i>Personalizada</i> para 3 expresiones con algoritmo <i>Fisher Face</i>	77
21.	Matriz de confusión base de datos personalizada para 5 expresiones con algoritmo <i>Fisher Face</i>	78
22.	Porcentajes de reconocimiento de expresiones para algoritmo <i>Fisher Face</i>	78
23.	Matriz de confusión base de datos <i>Cohn-Kanade</i> para 3 expresiones, SVM lineal	80
24.	Matriz de confusión base de datos <i>Cohn-Kanade</i> para 3 expresiones, SVM polinomial	80
25.	Matriz de confusión base de datos <i>Cohn-Kanade</i> para 5 expresiones, SVM lineal	80
26.	Matriz de confusión base de datos <i>Cohn-Kanade</i> para 5 expresiones, SVM polinomial	81
27.	Matriz de confusión base de datos <i>UVG</i> para 3 expresiones, SVM lineal	81
28.	Matriz de confusión base de datos <i>UVG</i> para 3 expresiones, SVM polinomial	81
29.	Matriz de confusión base de datos <i>UVG</i> para 5 expresiones, SVM lineal	81
30.	Matriz de confusión base de datos <i>UVG</i> para 5 expresiones, SVM polinomial	81
31.	Matriz de confusión base de datos <i>UVG 2 m</i> para 3 expresiones, SVM lineal	82
32.	Matriz de confusión base de datos <i>UVG 2 m</i> para 3 expresiones, SVM polinomial	82
33.	Matriz de confusión base de datos <i>UVG 2 m</i> para 5 expresiones, SVM lineal	82
34.	Matriz de confusión base de datos <i>UVG 2 m</i> para 5 expresiones, SVM polinomial	82
35.	Matriz de confusión base de datos <i>Ambiente 2</i> para 3 expresiones, SVM lineal	82
36.	Matriz de confusión base de datos <i>Ambiente 2</i> para 3 expresiones, SVM polinomial	83
37.	Matriz de confusión base de datos <i>Ambiente 2</i> para 5 expresiones, SVM lineal	83
38.	Matriz de confusión base de datos <i>Ambiente 2</i> para 5 expresiones, SVM polinomial	83
39.	Matriz de confusión base de datos <i>Personalizada</i> para 3 expresiones, SVM lineal	83
40.	Matriz de confusión base de datos <i>Personalizada</i> para 3 expresiones, SVM polinomial	83
41.	Matriz de confusión base de datos <i>Personalizada</i> para 5 expresiones, SVM lineal	84
42.	Matriz de confusión base de datos <i>Personalizada</i> para 5 expresiones, SVM polinomial	84
43.	Porcentajes de reconocimiento expresiones para algoritmo SVM	84
44.	Porcentajes de reconocimiento por expresión para 3 expresiones	86
45.	Porcentajes de reconocimiento por expresión para 5 expresiones	87

El desarrollo de los robots animatrónicos ha ayudado a la generación de nuevas tecnologías en los sistemas de control y en el procesamiento de imágenes. El objetivo general del trabajo era poder implementar por medio de software un programa que pueda detectar el rostro y reconocimiento de expresiones faciales. Para la consecución de este objetivo se realizó una búsqueda bibliográfica de las diferentes cámaras para seleccionar la más adecuada. Partiendo de la elección de la cámara se determinaron los algoritmos que se podía implementar en el programa *Python* para la detección de rostros.

Para el apartado de detección de rostros se realizó una comparación entre dos algoritmos utilizados. Uno que utilizaba algoritmos en cascada y otra que utilizaba las marca de cara. Se analizaron ambos algoritmos en dos ambiente diferentes, entre los que se evaluó el desempeño de estos. En aspectos como la distancia máxima, movimientos laterales, agresividad de movimientos y rotación con respecto del ángulo de visión. Entre los resultados se destaca el alcance que llega a tener el algoritmo *Haar cascade* en ambientes controlados y abiertos en donde presenta un mayor alcance que el algoritmo marcas de cara, en donde llega a tener un valor máximo de 5.31 m. En cambio, el algoritmo marcas de cara presenta una mayor robustez al momento de la de detección cuando existe una rotación de 45° del ángulo de visión con respecto a la cámara. Siendo el porcentaje de detección del algoritmo marcas de cara de 98.67% en comparación al 22.5% del algoritmo *Haar cascade*.

En el apartado del reconocimiento de expresiones faciales se realizó un búsqueda bibliográfica de los algoritmos que se utilizaban para la clasificación de estos. Los algoritmos seleccionados fueron *Fisher Face* y SVM. En el caso del programa *Fisher Face* que se basó en el el algoritmo *Haar cascade*. Para el programa *Fisher Face* se realizó un apartado con la reconstrucción de imágenes y representación de estas, mediante diagramas de dispersión de datos. Con el fin de observar una comparación entre las técnicas PCA y LDA utilizadas en el método *Fisher Face*. En el caso del apartado de SVM se utilizó las marcas de cara para la generación de los diferentes entrenamientos.

Para las bases de datos se utilizó la base de datos descargada *Cohn-Kanade* y las bases de datos generadas *UVG*, *UVG 2 m*, *Ambiente 2* y *Personalizada*. Ya que uno de los objetivos era realizar la prueba en ambientes diferentes, se generaron diferentes entrenamientos tomando en cuenta ciertos aspectos como la iluminación y distancia entre el sujeto y la

cámara. Estas bases de datos sirvieron para generar archivos llamados entrenamientos que sirven al algoritmo para poder predecir las expresiones faciales.

En este apartado se generaron las pruebas para cada uno de los entrenamientos realizando una comparación con el algoritmo *Fisher Face* y SVM. La base de datos con mejor porcentaje de reconocimiento de expresiones fue la base de datos *Personalizada* con valores entre 98.67% y 100%. En el caso de las demás bases de datos se obtuvo mejores porcentajes de reconocimiento con el algoritmo SVM en comparación con el algoritmo *Fisher Face*, ya que para todos los entrenamientos se obtuvieron reconocimiento superiores al 57.53%.

Se tuvo ciertos problemas reconociendo la expresión de disgusto, ya que existía confusión para los algoritmos para reconocer entre disgusto, neutral y enojo. Es por eso que se realizaron pruebas en donde se limitó la cantidad de expresiones detectadas a tres: feliz, enojo y sorpresa. Al realizar esta modificación se tuvo una mejora en cuanto a los porcentajes de reconocimiento ya que el valor más bajo se obtuvo en el ambiente *UVG 2 m*, con un valor de 68.21% para el algoritmo *Fisher Face*.

An animatronic is a device commonly used in amusement parks because of their flexibility to represent almost everything. Based on their interaction this type of robots have with people, the develop of a gesture recognition system for it, is a very useful tool. The objective of the system is to be able to detect a human face and predict their facial expressions. This would help the robot to have a better interaction and could be used to imitate the expression done by the subject.

The system was developed by software using Python language with the help of external libraries as Open CV for computer vision. All the information was captured by a logitech C920 webcam. This was done via iterative process, were several aspects were taken in count like lighting, posture, and reflections to generate two different programs based on different algorithms. The Fisher Face and SVM algorithms were tested in 5 different data sets, were 4 of them were created to have more control of the variables. The data sets include faces for expressions like: neutral, happiness, angry, surprise and disgusts. The data sets were generated in different environments to validate the robustness of the algorithms. The results for the recognition algorithms were generated by confusion-matrix to validate them and to visualize misclassifications.

Based on the results the SVM algorithm implemented using the facial landmarks detection had a better performance specially with the head posture and when implemented with non-controlled data sets. Also, the investigation validates the use of facial landmarks distance based on a central point face to be able to recognize facial expressions through it.

Un robot animatrónico es un robot que intenta, en la medida de lo posible, imitar el comportamiento de un ser vivo o algún otro objeto animado. Sin embargo el animatrónico no solo debe poder imitar el comportamiento, sino su apariencia. Los animatrónicos se utilizan mayormente en películas, parques de diversión y en algunos lugares turísticos modernos.

El animatrónico planteado es un pirata que cuenta con movimiento de cuello, ojos, antebrazo y dedos. El costo de un robot animatrónico es demasiado elevado, en gran parte a los mecanismos de control que utilizan estos ya que debido a su tamaño suelen utilizar mecanismos neumáticos e hidráulicos. Dependiendo del animatrónico se utilizan diferentes mecanismos para poder controlarlo. En este caso se determinó controlarlo todo de forma electrónica y con motores para generar un animatrónico con un costo no tan elevado.

El animatrónico fue dividido en tres módulos principales. El diseño de software para interacción con las personas, el diseño de mecanismos para su movimiento y el diseño de potencia para control del robot.

El módulo a trabajar consiste en el software para interacción con humanos. Este a su vez está dividido en dos módulos. El módulo de reconocimiento de rostros y gestos, por medio del cual será capaz de darle un seguimiento a los rostros ya que gran parte de la comunicación humana se da cuando se está hablando y mirando directamente a la otra persona. Además el módulo contará con un reconocimiento de expresiones faciales, con la utilidad de generar una interacción humano-robot más placentera. Es decir volver al robot lo más empático posible. El módulo utilizará una cámara web tanto para la detección de rostros y gestos. El programa dará como salida el mapeo de las coordenadas en la que se encuentra el rostro y la emoción que más se llega a resaltar en la persona. De esta manera se podrá comunicar con los otros módulos asignados. Esta software será complementado con el módulo de voz, que se encargará tanto de la interacción cómo de complementar la información proporcionada por el apartado de reconocimiento de rostros y gestos.

El trabajo consistirá en una exploración teórica de los algoritmos que se utilizan para el reconocimiento de objetos. Siendo específicamente destinado al reconocimiento de rostro

y características de este para poder detectar las diferencias que permitan detectar las expresiones faciales. Luego de esta exploración se implementó el sistema y se realizaron las pruebas en diferentes ambientes para la validación de los algoritmos.

2.1. Detección rostro

El primer acercamiento del trabajo consistió en la detección y seguimiento de rostro. El trabajo presentado por Chatrath y Pancaj [1] propone un seguimiento de rostros basados en la implementación del algoritmo Viola Davis a partir de una modificación para intentar mejorar la detección de rostros. En el documento [1] se enumeran tres de los pasos clave para la detección de caras:

- Formación de la Imagen Integral.
- Generación del Algoritmo *Adaboost*.
- Aplicación Clasificadores en Cascada.

El trabajo realizado por Chatrath [1] se basó principalmente en la robustez del algoritmo para las condiciones cambiantes, se analizó los resultados para diferentes iluminaciones, fondos cambiantes e inclusive se tomó en cuanto la variación de posición de la cámara. Factores como la inclusión de varios sujetos no representaban alguna dificultad debido a que las zonas de la imagen estaban representadas por la imagen integral. Los resultados para la detección de varios rostros en un mismo cuadro se encuentran en la Figura 1.

La utilización de los algoritmos *Haar cascade* presentan la ventaja en cuanto al tiempo computacional y buen porcentaje de detección en cuanto a detección de objetos, sin embargo el trabajo realizado por Mathews [2] presenta otra alternativa para la detección de rostro por medio de la localización de las marcas de cara. Este tipo de modelos son llamados modelos de apariencia, en el cual basa su detección de rostros no solo en la extracción de las características sino en la forma y apariencia de estas.

La Figura 2 presenta las marcas de cara detectadas para una de las imágenes basadas en la base de Datos LPFW. En el trabajo realizado por Belheumer se realizó un programa capaz

de detectar las marcas de cara en el cual para analizar sus resultados se hizo una comparativa utilizando las bases de datos LPFW y BioID. Belheumer realizó una comparación entre el algoritmo generado y otras herramientas previamente implementadas basándose en la localización correcta de los puntos a partir de la distancia promedio de error en base a la detección ocular [3].

Así como se pueden utilizar estos modelos para la detección de objetos, se puede utilizar para la detección de rostros y rasgos en ovejas. Esta acercamiento fue abordado por Yang [4]. Esto se puede observar en la Figura 3. El objetivo de este trabajo era presentar la robustez que poseen estos algoritmos así como la aplicación de estos métodos en la detección de diferentes objetos.

2.2. Reconocimiento de expresiones faciales

Uno de los métodos para el reconocimiento de expresiones faciales son los métodos *Eigen-Rostros* y *Fisher Face*. En el trabajo presentado por Ottado [5] presenta una comparación para ambos métodos en relación a la base de Datos *Extended Yale B*. En el trabajo realizado por Ottado se presenta una comparación de los métodos en base al número de características utilizadas para la obtención de los porcentajes de reconocimiento.

En el trabajo realizado por Gross en la Universidad de Pittsburgh [6] se realizó una comparación entre los algoritmos *Eigen-Rostros* y *Fisher Face*, además presenta toda la metodología necesaria para la toma y creación de una base de datos. En esta se intentó controlar al máximo las distancias, iluminación, fondos, etc. Con el fin de generar un base de datos en la cual las diferencias se basaran en las emociones expresadas y no en factores externos [6]. La Figura 4 presenta la distribución y el ambiente en donde se realizaron todas las tomas, mientras que la Figura 5 muestras las fotos obtenidas para la generación de la base de datos para un sujeto de una sola expresión.

Aparte de reconocer rostros por medio de *Eigen-Rostros* y *Fisher Face* se pueden llegar a utilizar otros algoritmos como SVM. Esto se aplicó en el trabajo realizado por Phillips, en la cual propone una comparación entre los algoritmos PCA y SVM a partir de una muestra pequeña de sujetos para el reconocimiento de rostros [7]. El trabajo presenta la comparación en base a las características extraídas y validación de reconocimiento de ambos algoritmos.

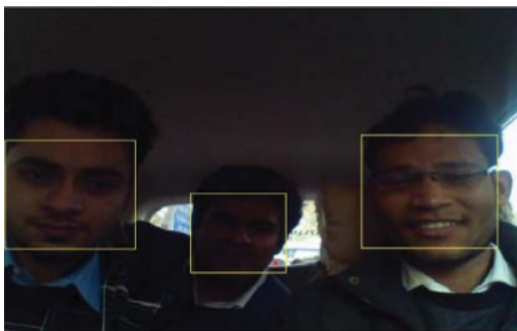


Figura 1: Detección y seguimiento de varios rostros con algoritmo *Haar cascade* [1]



Figura 2: Reconocimiento de rostro por medio de localización de marcas de cara [3]

En el caso del trabajo realizado por Paul Van Gent de la Universidad de Delft, en donde a partir de las marcas de cara se logra entrenar un algoritmos SVM para la localización de las marcas de cara [8].

2.3. Módulo reconocimiento de gestos proyecto animatrónico UVG

En una de las fases previas del proyecto, se le dio un enfoque mas enfocado a todo el cuerpo por medio de la herramienta *Kinect*. En esta fase tal y como se observa en la Figura 6, se enfocó en el reconocimiento de gestos de saludo o movimiento de brazos.

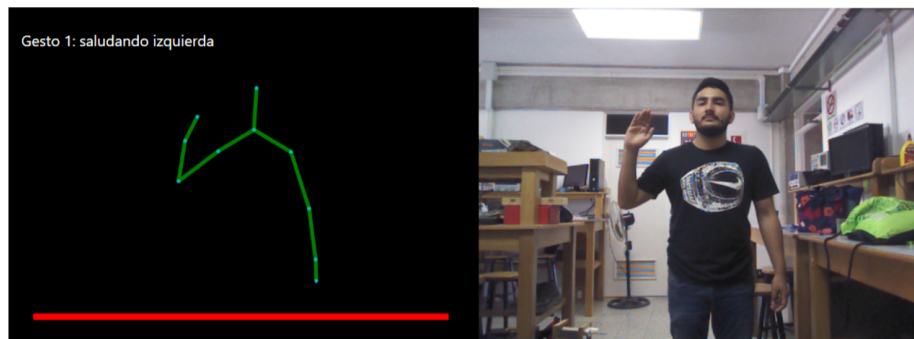


Figura 6: Reconocimiento gesto de saludo por medio de la herramienta *Kinect* [9]

El programa es capaz de darle seguimiento a todo el esqueleto, debido a la utilización de una cámara y un sensor infrarrojo que ayuda a detectar la profundidad a la que se encuentran los objetos. El programa fue analizando el reconocimiento de gestos a diferentes distancias. Se recomendó una distancia de 1.5 m como distancia inicial y se empezó a tomar datos a partir de esta distancia. Este contaba con un micrófono unidireccional para coordinar el módulo de voz del animatrónico.



Figura 3: Reconocimiento de rostro por medio de localización de marcas de cara para ovejas, la columna en rojo presenta la localización incorrecta [4]

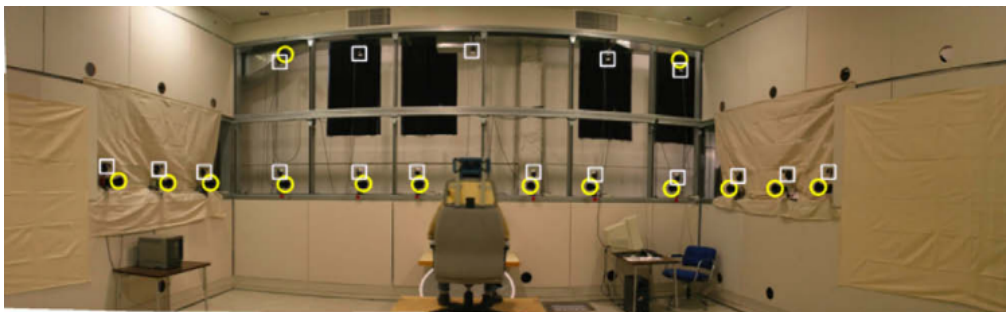


Figura 4: Ubicación de las cámaras para la base de datos Multi-PIE [6]



Figura 5: Imágenes obtenidas base de datos Multi-PIE [6]

Los robots son instrumentos cada vez más comunes en la vida de los seres humanos, ya sea para realizar acciones que se nos hacen imposible o para la simplificación de la vida humana. En el caso de un animatrónico su función no solo se limita al entretenimiento, sino la creación de objetos que ya fueron extintos, como lo es el caso de gran parte de animatrónicos que simulan dinosaurios y sus comportamientos para propósitos educativos. La función del animatrónico va a depender de la selección del animatrónico a realizar y del alcance que se le quiera dar a este. Para este trabajo se estará trabajando en el apartado de detección de rostro y reconocimiento de gestos para el robot animatrónico.

El desarrollo de tecnología para la detección y reconocimiento de gestos puede que ayude, no solamente a la comunicación humano-computador a hacerla lo más empática posible. Ya que gran parte de estas máquinas están en el día a día, puede ser la base de principios de seguridad en la detección de rostros y expresiones faciales para diferentes sistemas de seguridad. Inclusive podría utilizarse para mejorar la forma de comunicarse con las personas, ya que se tendría un software que permitiría la comprensión de los gestos faciales expresados. Otro de los ámbitos en donde se utiliza el reconocimiento y detección de objetos es en el desarrollo de vehículos autónomos [10], además de las aplicaciones médicas donde se realiza procesamiento de imágenes para detección y clasificación de objetos [11].

Como estudiante de ingeniería electrónica el proyecto será de gran ayuda ya que me permitirá crear un puente entre la conexión humano-robot. Que el animatrónico pueda reconocer y seguir rostros además de expresiones faciales lleva al robot a poder tener una mejor interacción. Esta conexión me permitirá en un futuro poder desarrollar tecnologías que tengan una mejor adecuación a la vida que llevamos actualmente. Además este trabajo me permitirá desarrollar conceptos de procesamiento de imágenes para la detección y clasificación de objetos que me servirán para la creación de nuevas tecnologías.

4.1. Objetivo general

Implementar por medio de software un programa que pueda darle seguimiento a los rostros y el reconocimiento de expresiones faciales de las personas.

4.2. Objetivos específicos

- Determinar el equipo necesario para el seguimiento de rostros y reconocimiento de expresiones faciales tales como felicidad, enojo, asombro y disgusto.
- Determinar e implementar el software específico para el reconocimiento y detección de rostros
- Determinar e implementar el software y base de datos adecuados para el reconocimiento de las expresiones faciales (felicidad, enojo, asombro y disgusto)
- Generar un manual de seguimiento de pasos para la implementación del proyecto para reconocimiento de rostros y expresiones faciales.
- Realizar pruebas a 30 sujetos para la detección de rostros y expresiones faciales en dos ambientes diferentes

El objetivo general del trabajo era poder implementar un sistema de detección de rostro y reconocimiento de expresiones faciales. Por lo cual se definirán los alcances por parte de los dos aspectos a evaluar.

En el caso del primer apartado el programa que se generó es capaz de realizar la detección en un espacio abierto en un radio menor a los 3.5 m para garantizar la detección del rostro del sujeto. Este tendrá como limitación que se tendrá que estar observando directo a la cámara en todo momento, ya que en caso se tenga un ángulo superior a los 45° con respecto al ángulo de visión a una distancia de 1 m, se interrumpirá la detección y se podrían dar resultados falsos. Esto se determinó ya que se simuló que el tamaño del animatrónico fuera de 2 m de largo y que a una distancia de 1 m fuera capaz de poder observar todo el animatrónico, sin que el programa sufriera una detección negativa.

Ya que la captura de la imagen se da en un tamaño 640×480 se tendrá una resolución basado en este tamaño de imagen, por lo cual el mapeo de coordenadas del plano de visión se limitan al tamaño de la imagen mencionado anteriormente. El algoritmo será lo suficientemente capaz de detectar varios rostros al mismo tiempo, sin embargo este le dará prioridad al que se encuentre a una menor distancia de la cámara. Por lo cual solo podrá ser capaz de enviar las coordenadas e información de la expresión para la persona que se encuentre más cercana.

El algoritmo está diseñado para funcionar en espacios con una buena iluminación por lo cual no se recomienda utilizarlos en ambientes en donde se tendrá una iluminación menor a los 150 lx, ya que el algoritmo no será capaz de darle el seguimiento al rostro. Los algoritmos de detección de rostros se basa en la detección del rostro completo, por lo cual los algoritmos presentarán buenos resultados siempre y cuando no exista ningún tipo de bloqueo en alguna parte del rostro. El algoritmo será capaz de detectar los movimientos del sujeto de prueba siempre y cuando no realice movimientos demasiados bruscos, ya que se menciona más adelante en este apartado, se podrán trabajar con una velocidad de tres a cuatro imágenes por segundo.

En el caso del reconocimiento de expresiones se planteo que el algoritmo será capaz de realizar la predicción de las expresiones en los ambientes en donde se generaron las bases de datos, siendo el caso de los ambientes *UVG*, *UVG 2m*, o *Ambiente 2*. Debido a que al momento de generar los entrenamientos se extraen las características más significativas que se obtuvieron al momento de la clasificación de las imágenes. Características que al modificar el ambiente en donde se generaron pueden llegar a tener resultados negativos. El trabajo se limitó a que se implementaran dos algoritmos de reconocimiento de expresiones para hacer una comparación directa entre el método *Fisher Face* y SVM para seleccionar el mejor algoritmo a partir de las pruebas realizadas.

Al momento de querer implementar cualquiera de los dos algoritmos se deberá tener condiciones similares para obtener los mejores resultados de reconocimientos, tal es el caso que si se utilizara el entrenamiento *UVG* se tiene que estar a una distancia aproximada de 80 cm de la cámara para obtener un buen reconocimiento. En el caso de la utilizar el entrenamiento *UVG 2 m* se tendrá que estar a una distancia de 2 m de la cámara.

El algoritmo será capaz de procesar y predecir entre tres a cuatro imágenes por segundo, esto variará dependiendo del tiempo de procesamiento que tarde el algoritmo en procesar la información. El tiempo de procesamiento dependerá de factores como la cantidad de sujetos presentes en el cuadro. Al igual que para la detección de rostros, el algoritmo será capaz de brindar información sobre la expresión en base al sujeto que se encuentre más cercano a la distancia con respecto a la cámara. En el caso que se implemente el programa SVM si el sujeto no se encuentra en el rango de distancia a las que se generaron el entrenamiento no será capaz de proveer información acerca de la expresión mostrada, solo brindará información acerca de la posición del rostro.

En el caso de la implementación del algoritmo *Fisher Face* y se desee generar una nueva base de datos o agregar elementos a estos, no se pondrá generar un entrenamiento con un número mayor de 400 imágenes con una resolución de 350×350 , debido a las limitaciones de memoria que presenta la implementación en *Python*. Se tendrá que reducir la resolución de las imágenes o la cantidad de estas.

6.1. Aspecto psicológico de las expresiones

La emoción “es un proceso de tipo particular de valoración automática influida por nuestro pasado evolutivo y personal, en el que sentimos que está ocurriendo algo importante para nuestro bienestar, produciendo un conjunto de cambios físico y comportamentales para hacernos cargo de una situación”[12]. Uno de los aspectos que separan al ser humano de las máquinas es la capacidad de poder expresar y percibir emociones. Esta habilidad esta descrita desde de la inteligencia emocional. El psicólogo Salovey, lo define como “un subconjunto de la inteligencia social que comprende la capacidad de controlar los sentimientos propios y los de los demás, de discriminar entre ellos y utilizar esta información para guiar nuestro pensamiento”[13]. Esta habilidad de poder discriminar las expresiones y actuar a partir de ella es una de las herramientas que facilitan las interacciones humanas. Charles Darwin fue el primero en relacionar la expresiones faciales con “emociones” y a partir de esta fecha se ha dado un interés en el área psicológica de poder comprender los factores que afectan a estas. Entre los cuales destacan los factores aprendidos a la hora de expresar las emociones.

6.1.1. Universalidad de expresiones

Charles Darwin fue el primero que planteó la universalidad de las expresiones, sin embargo no fue sino hasta partir del año 1950 que empezaron a surgir un mayor número de experimentos para la validación o refutación de esta teoría. Si bien no existe un acuerdo 100 % definido, Ekman propone que existen seis expresiones universales [12]:

- Felicidad
- Enojo
- Miedo

- Sorpresa
- Tristeza
- Asco/disgusto

Cabe destacar que en uno de los experimentos realizados por Ekman para concluir sobre la universalidad de las expresiones, fue realizado en Nueva Guinea, en un aldea aislada del mundo universal. El experimento dio resultados interesantes ya a pesar de que las emociones estaban expresados por sujetos de la cultura occidental, los habitantes de esta población lograron poder reconocer las seis emociones mostradas [12]. Es cierto que al realizar este tipo de experimentos se observó que se mostraba cierta tendencia a reconocer algún tipo de expresión específico, sin embargo se puede concluir que las seis expresiones mencionadas anteriormente corresponden a un grupo de expresiones universales del ser humano. No dependen de los factores culturales para el reconocimiento de estas sino que se tiene un origen biológico.

6.1.2. Identificadores de expresiones faciales

Al ser el ser humano capaz de expresar y reconocer las expresiones, es necesario distinguir las características principales para poder hacer énfasis en estas y tener un mejor reconocimiento. Entre los cuales destacan los gestos faciales, gestuales y el lenguaje que se utiliza [14]. Ya que se tiene un mayor interés en las expresiones faciales, entre estas destacan [15]:

- Las frente/cejas
- Los ojos/párpados
- Parte inferior rostros (boca)

Este resultado no solo estuvo basado en la observación de los gestos, sino que se determinó por medio de medición de impulsos eléctricos en las distintas partes de rostro y se encontró una mayor actividad muscular en estas zonas de la cara [12].

6.2. Clasificador *Haar cascade*

El algoritmo *Haar-cascade Classifier* es un método muy utilizado para el procesamiento, detección y reconocimiento de imágenes. Este método es muy utilizado debido a su bajo costo computacional y por la rapidez que este permite para el procesamiento de la información en tiempo real.

El grupo de algoritmos basados en *Haar-like Features* basa su funcionamiento en la detección de características o rasgos como caras, ojos, brazos u otros a partir de las intensidades luminosas de las regiones rectangulares [16]. Es por eso que es muy común la utilización de

este tipo de algoritmos para ambientes de luz controlados y suele funcionar para personas con tez blanca, ya que a medida que es más oscuro el color de la persona los contrastes entre las intensidades luminosas son menos perceptibles para la cámara y el algoritmo.

Estos rasgos pueden estar asociados, programados o entrenados dependiendo de la plataforma que se utilice para identificarlos. Los rasgos son puestos a prueba en rectángulos, dependiendo del rasgo y resolución que se quiere obtener se pueden obtener de dos, tres o cuatro rectángulos tal y cómo se observa en la Figura 7. [1]. El tipo de rectángulo que se implemente no solo va a depender del rasgo, sino del ambiente y posición del rasgo que se quiere identificar. Los rectángulos presentan siempre una región mayormente sombreada que el otra ya que este se basa en intensidades luminosas. Es por eso que los rectángulos que encuentran los rasgos tiene una parte gris y una parte blanca, que representa este contraste de la identificación de rasgos, pero en términos de números representan 1 y -1 para la simplificación del manejo computacional.

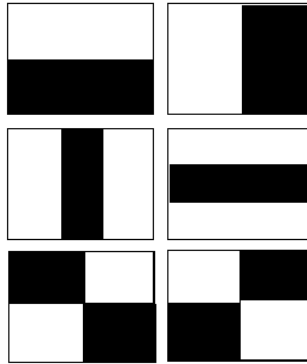


Figura 7: Set de rectángulos de características *Haar cascade*

6.2.1. Viola-Jones

El algoritmo de Viola-Jones basa su funcionamiento en los algoritmos *Haar-cascade like*, pero presenta una mejora de desempeño con respecto a sus predecesores. Este algoritmo presenta dos elementos clave, para mejorar su rendimiento, que son imagen integral y *Ada-boost*. Viola-Jones fue desarrollado por Paul Viola y Michael Jones en 2001 y es ampliamente utilizado en muchos dispositivos móviles para la detección de varios rasgos.

Imagen integral

El principio básico de una cámara es la captura de fotones a través un área o superficie. Por lo cual se genera un matriz y cada pixel representa el fotón capturado en esa dirección. Este arreglo o matriz de la imagen es lo que permite poder manipularla de forma integral. Este manejo de forma integral se basa en otro arreglo de pixeles en donde cada uno almacena "la suma de todos los pixeles que están arriba y hacia la izquierda de ese punto en la imagen original"[16]. Lo cual provee que cada posición, no solo tenga información de ese pixel, sino de los que lo rodean.

La imagen integral permite una forma más eficiente de la implementación de los algoritmos ya que en lugar de compararla con cada uno de los píxeles, se utiliza un algoritmo para comparar los puntos en donde se realizó la suma integral de estos para poder comparar los rasgos de la imagen. Para poder referenciarse a la figura usando imagen integral se utilizan 4 parámetros de referencia para obtener las cuatro esquinas del rectángulo que se está analizando, cabe resaltar que se han realizado modificaciones para que el área que se quiere comparar pueden estar rotada 45 grados, sin tener que agregar otro parámetro extra para referenciarlo [17]. Este principio de imagen integral puede ser llevado a volumen integral para una representación en 3D, no obstante las imágenes se representan por simplicidad en un arreglo de 2D lo cual permite un análisis y procesamiento de imágenes más eficiente que un modelado 3D.

Adaboost

Complementario al uso de la imagen integral, el algoritmo Viola-Jones utiliza un algoritmo llamado *Adaboost*. *Adaboost* es un algoritmo de *machine learning* que combina reglas de procesamiento débiles e inexactas, pero que combinadas logra una identificación precisa y rápida [18]. Lo que diferencia principalmente este algoritmo de otros, es su forma de clasificación de cada una de las instancias OCR. Que por sus siglas en inglés está dado por reconocimiento óptico de caracteres [19]. Esta característica deberá de ser fácil y rápida de detectar, sin tener mucho peso por lo cual su predicción no será la más precisa. Si la hipótesis utilizada cumple con esta característica se conoce como un método débil. El conjunto de todas las posibilidades de las instancias es conocido, como dominio (X).

Los conceptos de dominio, instancia y clasificación, son muy importantes para poder relacionar el algoritmo *Adaboost*. En el pseudocódigo de *Adaboost* en donde se enumeran los pasos para su implementación. El cómo se realiza una distribución de los pesos del algoritmo y de cómo en, cada ronda que se vuelven a llamar los algoritmos y los resultados van cambiando debido a que va reconociendo los clasificadores más débiles y los va mejorando. Esta asignación de pesos la va generando automáticamente hasta que se detecte un mejor peso para cada uno de las características. En el caso del pseudocódigo que se muestra en el siguiente segmento, se asumió un comportamiento binario en la característica. Como respuesta sí o no, es por eso que el mapeo que se desea realizar proporcione resultados de $\{-1,1\}$ [19]. En caso que se desea aumentar la dimensión del algoritmo se debe aumentar los posibles resultados de las salidas, en lugar de limitarlo a un componente binario.

Pseudocódigo *Adaboost*

Se tienen dos vectores, \mathbf{x} de entrada y \mathbf{y} de salida, que se encuentran en el dominio χ

Dados $(x_1, y_1), \dots, (x_m, y_m)$ donde $x_i \in \chi$, $y_i \in \{-1, +1\}$

Se inicializa $D_1(i) = 1/m$ para $i = 1, 2, 3, \dots, m$.

Para $t=1, 2, 3, \dots, T$:

- Entrenar un aprendizaje débil usando la distribución D_t
- Obtener la hipótesis débil de $h_t : \chi \rightarrow -1, +1$
- Seleccionar el valor de h_t que minimice el error:

$$\epsilon_t \doteq Pr_{i \sim D_t}[h_t(x_i) \neq y_i] \quad (1)$$

- Escoger $a_t = \frac{1}{2} \ln \frac{1-\epsilon_t}{\epsilon_t}$.
- Actualizar para $i = 1, 2, 3, \dots, m$:

$$D_{t+1} = \frac{D_t(i)}{Z_t} \times \left\{ \begin{array}{ll} e^{-\alpha t} & \text{si } h_t(x_i) = y_i \\ e^{\alpha t} & \text{si } h_t(x_i) \neq y_i \end{array} \right\} = \frac{D_t \exp(-\alpha_i y_i h_t(x_i))}{Z_t} \quad (2)$$

Donde Z_t es el factor de normalización, elegido para que D_t sea una distribución.

Obteniendo la hipótesis de salida de la forma:

$$H(x) = \text{sign} \left(\sum_{i=1}^T a_t h_t(x) \right) \quad (3)$$

6.3. Marcas de cara

Las marcas de cara, son coordenadas en el rostro ya sea de un humano o animal, en el cual se localizan ciertas características principales del rostro. Entre estos destacan, los ojos, cejas, boca o frente. La detección de estas marcas de cara, pertenece a una clasificación aún más grande de clasificación y detección de objetos. La detección y clasificación de objetos es una de las tareas que los seres humanos realizan a diario, es parte de la rutina diaria. El ser humano logra captar información en forma de imágenes por medio del ojo y procesarlas a través del cerebro para poder hacer el reconocimiento. Esta tarea resulta muy compleja para las computadoras, ya que la variación de tamaño, color, iluminación y ruido son factores que afectan no solo en la obtención de datos sino en la clasificación de estas. La detección de objetos se basa fundamentalmente en la extracción de características, es decir que es lo que hace a este objeto diferente, que hace que puede clasificarse. Este proceso de selección es gran parte de la clave de tener un buen entrenamiento para la detección de objetos. Los algoritmos que sirven para la detección de este tipo de características son los modelos de apariencia activos [20]

6.3.1. Modelos de apariencia activos

Los modelos de apariencia activos son algoritmos que se utilizan para la localización de las características principales de un objeto. Este tipo de algoritmos pueden ser clasificados en dos categorías: basados en textura y basados en forma. Debido a que se centrará principalmente en la detección de marcas de rostros, solo se estarán analizando los algoritmos basados en forma.

Los algoritmos de detección de rostro basados en forma se basan en la geometría del rostro, de esta se obtienen tres componentes principales para tener un buen desempeño: forma, apariencia e instancia:

Forma

Obtienen la "forma" de la instancia (imagen a analizar) y buscan que esta se asemeje lo más posible al modelo. Es decir que puedan encontrar una coincidencia entre la imagen a procesar y el modelo que se está buscando [2]. La ecuación matemática que describe la forma de estos modelos se observa en la ecuación 4.

$$\mathbf{s} = (x_1, y_1, x_2, y_2, \dots, x_n, y_n) \quad (4)$$

Siendo \mathbf{s} un vector forma, generado a partir de las coordenadas de estos vértices de la imagen, el cual puede ser generado a partir de una combinación lineal entre el modelo y n vectores de forma [2]:

$$\mathbf{s} = \mathbf{s}_0 + \sum_{i=1}^n p_i \mathbf{s}_i \quad (5)$$

Para este tipo de algoritmos se suele utilizar PCA, descrito en la sección 6.4.1. El cual genera una reducción y proporciona los componentes principales para la variación entre ellos.

Apariencia

La apariencia es el conjunto de pixeles que se encuentra dentro de la red de vértices que definen la forma [2]. Al ser una imagen que se encuentra entre las coordenadas de la forma, puede ser generada a partir de la imagen modelo A_0 generada a partir de la forma inicial. En donde la ecuación 6 representa las coordenadas de la apariencia y la ecuación 7. Los valores de λ que son llamados parámetros de apariencia.

$$\mathbf{x} = (x, y) \quad (6)$$

$$A(\mathbf{x}) = A_0(\mathbf{x}) + \sum_{i=1}^m \lambda_i A_i(\mathbf{x}) \quad \forall \mathbf{x} \in \mathbf{s}_0 \quad (7)$$

Instancia del modelo

La instancia del modelo utiliza las ecuaciones 5 y 7 para poder generar una instancia, es decir una reconstrucción de la información obtenida por medio de la combinación lineal de la apariencia y forma [2]. Esto permite realizar la asignación al modelo base para la distinción de características. La serie de pasos que involucra la forma, apariencia e instancia está descrita en la Figura 8.

Observando la Figura 8 se ve una de las grandes ventajas que proveen este tipo de algoritmos con respecto a los clasificadores *Haar cascade*. La cual es que permite una mejor detección a pesar de que no existe un alineamiento de la cara con el modelo, se logra poder

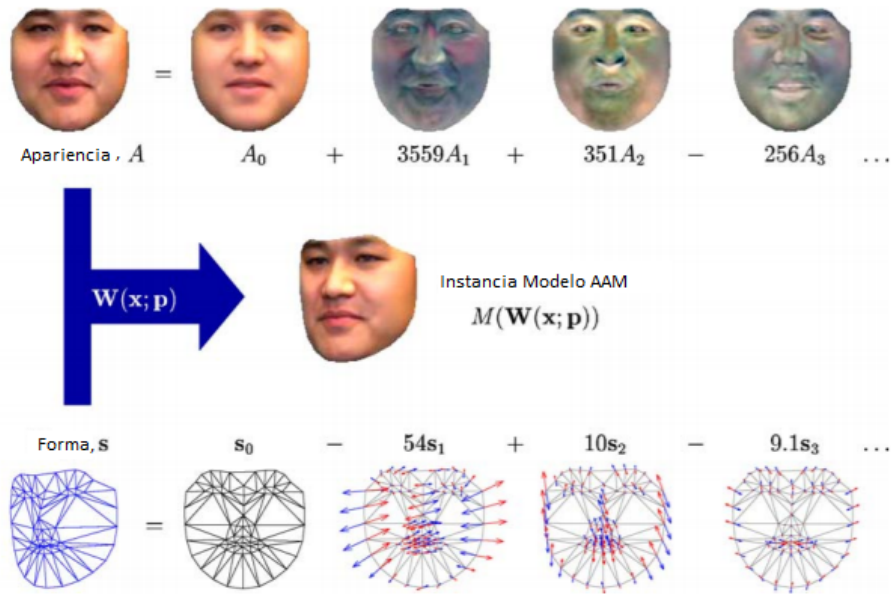


Figura 8: Generación de marcas de rostros por modelos de apariencia activos [2]

detectar los elementos de esta. Esto ya que al basarse en la geometría permite tener un modelo activo, sin embargo el costo computacional y de tiempo es superior a los algoritmos en cascada.

6.4. Reconocimiento facial

Las expresiones son uno de los temas a reconocer para este trabajo, sin embargo no existe un software o una condición específica para reconocer un gesto. Al igual que el ser humano, la computadora se basa en ciertos patrones o elementos claves para poder detectar ciertos gestos, tal es el caso de una sonrisa para saber si esta feliz o un lágrima cercana a los ojos para reconocer que se encuentra triste. Este reconocimiento de patrones o expresiones esta denotado como *Facial Recognition*. Este utiliza distintos métodos para poder distinguir la emoción. A continuación se describen algunos de los métodos o la combinación de estos para el reconocimientos de expresiones faciales:

6.4.1. PCA, *Eigen* rostros

PCA por sus siglas en inglés, se refiere a *análisis del componente principal*, esta es una técnica estadística muy utilizada para el reconocimiento de patrones en datos que contienen multi-variables. Es un método no supervisado, debido a que no necesita que se le proporcionen etiquetas de clase, sino que automáticamente busca las diferencias entre los datos y busca la obtención de los componentes principales. El pseudocódigo utilizado por PCA [21], para un arreglo bidimensional se muestra a continuación:

1. Ordenar los datos en una matriz N^2

$$\mathbf{x} = \{x_1, x_2, x_3, \dots, x_i\} \quad (8)$$

En donde cada fila, representa cada una de las imágenes

2. Obtener una media, de las imágenes de la ecuación 8, μ .
3. Obtener una matriz de la covariancia entre los datos anteriores.

$$S_T = \sum_{k=1}^N (x_k - \mu)(x_k - \mu)^T \quad (9)$$

- Si $x_k - \mu$ tiene un signo positivo, tendrá una relación proporcional.
 - Si $x_k - \mu$ tiene un signo negativo, tendrá una relación inversamente proporcional.
 - Si $x_k - \mu$ es igual a cero, los datos son completamente independientes.
4. Calcular los vectores propios y valores propios a partir de la matriz de covariancia.
 - Se busca determinar el valor propio más grande, y este pasará a ser el componente principal del análisis
 - En este paso se suelen simplificar o reducir las dimensiones, eliminando los valores propios más pequeños y obteniendo una matriz, solo con los vectores propios W que se van a utilizar. Dependiendo de la aplicación o el usuario, este es el punto en donde se suele colocar un límite, para que los valores que estén por debajo de este punto, sean eliminados.

$$W_{opt} = \operatorname{argmax} |W^T S_T W| \quad (10)$$

5. Obtener una matriz de datos nueva, a partir de la multiplicación de la matriz de diferencia de medias, con la matriz de vectores propios seleccionados.
 - Este paso da como resultado una matriz en términos de los vectores propios utilizados, a partir de los datos originales, ver Figura 10.

$$\mathbf{y} = W\mathbf{x} \quad (11)$$

- En lugar de ser píxeles mapeados por coordenadas x y y , estén mapeados a coordenadas de los vectores propios que al ser ortogonales, permite ver las representaciones de los datos.



Figura 9: Set de imágenes iniciales [22]

En el caso específico de reconocimiento facial el método PCA es representado por un arreglo bidimensional en donde cada va a representar la información del color o intensidad de este, en cada uno de los píxeles de una imagen. En el caso de la Figura 9 muestra el set de



Figura 10: Imágenes reconstruidas por vectores propios [22]

datos con las imágenes originales, mientras que la Figura 10 muestra la imagen reconstruida, a partir de los vectores propios obtenidos en los pasos anteriores.

El método es utilizado para simplificar el costo computacional y de orden en la búsqueda de realizar reconocimiento de rostros. La clasificación del algoritmo se mencionó en la sección 6.4 el cual es no supervisado. Debido a que no se le colocan etiquetas para generar un entrenamiento, muestra una versatilidad para la realización de varias tareas como:

- Identificación, lo cual genera etiquetas de los individuos.
- Reconocimiento de persona, el cual el algoritmo decide si el individuo se encuentra en la base de datos.
- Categorización, en la cual la cara debe ser asignada a una clase.

En el caso del proyecto animatrónico, se centrará en la categorización la cual se utilizará para ir catalogando los rostros de los sujetos de prueba a las expresiones, en el caso de identificación se utilizará para que a partir de esas etiquetas se pueden ir clasificando. Este método brindará la posibilidad de reducción de orden, que sirve para aplicarlo a bases de datos con un alto número de características y pocos datos.

6.4.2. FISHER FACE

El método *Fisher Face* es una técnica de computación para el reconocimiento de rostros, esta técnica hace uso del análisis del discriminante lineal de *Fisher* (LDA), de ahí que surge el nombre. Este algoritmo entre dentro de la clasificación de algoritmo supervisados, ya que necesita de conocer los valores reales para poder generar los entrenamientos. Así como para el método de *eigen rostro* utiliza PCA para ir reduciendo y clasificando el orden de los matrices, el método *Fisher Face* utiliza PCA para la reducción de este. El cambio que se presenta con respecto a algoritmo *eigen rostro* es que la transformada que se utiliza en la reducciones busca maximizar las diferencias entre clases y reducir la diferencia entre la clase [5], haciendo una reducción mayor que el algoritmo PCA. La ecuaciones matemáticas que describen estos comportamientos se observan en las ecuaciones 16, 17 en donde la ecuación 16 representa la diferencia que se obtendrá de la separación entre clases. Esta es la que se busca maximizar. En cambio, la que se presenta en la ecuación 17 busca que esta diferencia se minimice. El pseudocódigo utilizado para la implementación de *Fisher Faces* se presenta a continuación:

1. Se consideran dos vectores. Ambos con el mismo número de datos, el primero representa las clasificaciones y el segundo es el vector con los pixeles de las imágenes a analizar.

$$\mathbf{y} = \{y_1, y_2, \dots, y_n\} \quad (12)$$

$$\mathbf{x} = \{x_1, x_2, x_3, \dots, x_n\} \quad (13)$$

2. Se calcula la media total de todas las imágenes μ , y la media total de cada clase μ_i

$$\mu = \frac{1}{N} \sum_{i=1}^N x_i \quad (14)$$

$$\mu_i = \frac{1}{|X_i|} \sum_{x_j \in X_i} x_j \quad (15)$$

3. Se calculan las matrices de dispersión, siendo S_B la matriz de entre clases y S_W la matriz intra-clases.

$$S_B = \sum_{i=1}^c N_i (\mu_i - \mu) (\mu_i - \mu)^T \quad (16)$$

$$S_W = \sum_{i=1}^c \sum_{x_k \in X_i} N_i (x_k - \mu_i) (x_k - \mu_i)^T \quad (17)$$

4. Se procede a calcular la matriz que maximiza la separación de clases, similar al de la forma que se realizó en PCA.

$$W_{opt} = \underset{W}{\operatorname{argmax}} \frac{|W^T S_B W|}{|W^T S_W W|} \quad (18)$$

Sin embargo en el caso específico de procesamiento de imágenes no se puede resolver ya que es una matriz singular. Esto debido a que el número de muestras, suele ser menor que el número de características a analizar. Para que se pueda realizar, el rango de la matriz S_W debe ser a lo sumo $(N-c)$, siendo N el número de muestras y c el número de clases [23]. Es por esta razón que se utiliza el método PCA para esta primera reducción y unos arreglos al método discriminante de *Fisher* para poder resolver. Es por esta modificación que el algoritmo tiene el nombre de *Fisher Faces*.

5. Se obtiene la proyección W , a partir del cálculo de las ecuaciones 19 y 20.

$$W_{pca} = \underset{W}{\operatorname{argmax}} |W^T S_T W| \quad (19)$$

$$W_{lda} = \underset{W}{\operatorname{argmax}} \frac{|W^T W_{pca}^T S_B W_{pca} W|}{|W^T W_{pca}^T S_W W_{pca} W|} \quad (20)$$

$$W_{fish} = w_{lda}^T W_{pca}^T \quad (21)$$

6. Por último se obtiene la transformación de la forma:

$$\mathbf{y} = W_{fish} \mathbf{x} \quad (22)$$



Figura 11: Set de imágenes procesamiento Eigen Caras [5]



Figura 12: Set de imágenes procesamiento *Fisher Face* [5]

El enfoque que se hace en la diferenciación entre clases, hace que el método de *Fisher Face*, se encuentre orientado a la clasificación. Mientras que el método *eigen caras*, se utiliza para representación [22]. Esta diferencia se puede observar en las figuras 11 y 12 en donde se realiza una comparación de cómo se van generando los vectores a medida que se van ingresando nuevas imágenes. De cómo el método genera estas transformaciones, para poder ir generando las comparaciones. En el caso de la Figura 12 de algoritmo *Fisher Face* se observa que existen rasgos más marcados, como la posición de los labios, la nariz y las cejas.

Uno de los requerimientos para la implementación es que los datos deban tener una distribución uniforme. En lo cual se basa el algoritmo para buscar la maximización de diferencias presentes en las ecuaciones 17 y 16. Entre los mayores beneficios del algoritmo es que suele tener un buen porcentaje de rendimiento con pocas características a analizar, por lo cual no demanda de tantos recursos computacionales y puede ser utilizado para procesamiento de imágenes en tiempo real [5].

6.4.3. SVM

Por sus siglas en inglés SVM representa a las máquinas de vectores de soporte. Estos son considerados un grupo de algoritmos implementados para la regresión o clasificación de ciertos patrones. En este caso se estudiará el uso de los SVM para el reconocimiento de patrones en los rostros. Las máquinas de vectores de soporte, fueron algoritmos creados por Vapnik. A pesar de ser un algoritmo supervisado, este no solo busca el mapeo de un valor, hacia una característica en específica, sino que utiliza la distribución de las muestras para poder hacer la clasificación y un mejor mapeo [24]. Originalmente el algoritmo estaba implementado para realizar una clasificación binaria sobre una tendencia lineal, sin embargo actualmente se han desarrollado nuevas implementaciones que permiten la clasificación multi-clase, para distribuciones no necesariamente lineales. A continuación se presenta un algoritmo para el uso de vectores de soporte para dos dimensiones.

1. Se inicializan dos vectores:

$$\mathbf{x} = \{x_1, x_2, x_3 \dots x_n\}, \quad \mathbf{y} \in \{-1, 1\} \quad (23)$$

- Se encuentran vectores de soporte ϕ , de modo que al multiplicarlo por el vector de entradas \mathbf{x} se obtenga un valor llamado hiperplano que permita la clasificación [25], ecuación 24.

$$\phi \times \mathbf{x} = c \quad (24)$$

- A partir del hiperplano generado se puede hacer la clasificación, analíticamente se observa en la ecuación 25 o gráficamente, en la Figura 13

$$\left\{ \begin{array}{l} \phi \times \mathbf{x}_i > c \quad \text{si} \quad \mathbf{y}_i = 1 \\ \phi \times \mathbf{x}_i < c \quad \text{si} \quad \mathbf{y}_i = -1 \end{array} \right\} \quad (25)$$

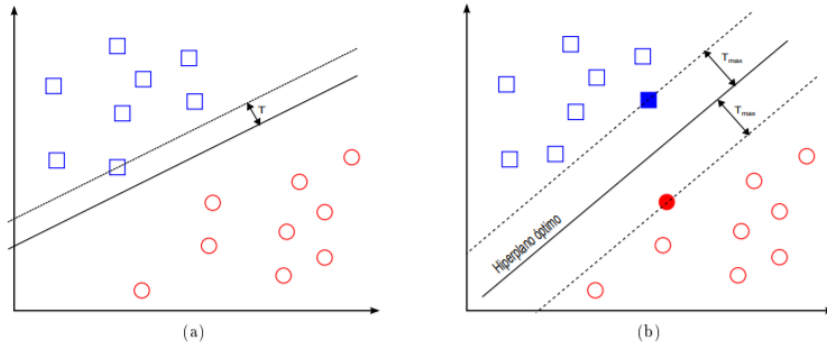


Figura 13: Clasificación binaria por medio de SVM [5] a) Hiperplano no-óptimo, b) Hiperplano optimizado.

Los algoritmos SVM presentan ventajas de clasificación superiores a las de varios métodos de clasificación, sin embargo su tiempo computacional es superior a varias de ellas, debido a la condición que no solo toma en cuenta el valor, sino la distribución de estos. Se debe tener especial cuidado con los métodos de comparación que utilizan los métodos SVM, debido a que si se realizan cierto tipo de comparaciones genera una situación llamada sobre ajuste [24]. El sobre ajuste puede causar que la clasificación que hace de los datos se vuelva muy específica y en el caso que se le de un mayor peso a una de las clases, podría tener una tendencia a solamente detectar una clase, es decir tener una predicción muy sesgada.

Las máquinas de soporte presentan la ventaja que se pueden realizar clasificaciones de distribuciones no uniformes, sin que esto afecte la predicción de los nuevos valores. Esto debido a que en la mayoría de implementaciones permite cambiar el peso de las clasificaciones, ya que al ser un algoritmo supervisado tiene conocimiento de la muestra de entrenamiento para este algoritmo. Estos pesos al ser modificados y con la capacidad del algoritmo de analizar la distribución de estos, permite una mejor clasificación para este tipo de datos.

6.5. Logitech c920

La implementación de algoritmos de procesamiento de imagen, requiere de dispositivos que se encargan de la captación de cuadros para realizar el posterior procesamiento de estos.

El proyecto utiliza la cámara *Logitech c920*. La imagen de la cámara se puede observar en la Figura 14. Las especificaciones técnicas del dispositivo *Logitech c920* son [26]:

- Capacidad de obtención de video a 720p en alta definición hasta un tamaño de imagen de 1280×720 .
- Captura de imágenes de hasta 15 megapíxeles.
- Lente de cristal.
- Auto Enfoque *Carl Zeiss* de 20 pasos.
- Conexión USB 2.0 y 3.0
- Soporte para trípode universal.

Para que que la cámara sea capaz de obtener las definiciones y tamaño de video mencionados necesita de una computadora con especificaciones [26]:

- Procesador 2.4 GHz *Intel Core 2 Duo* o superior.
- 2 GB de memoria RAM o superior.
- Computadora con puerto USB 2.0 o 3.0.



Figura 14: Cámara web *Logitech C920* [26]

6.6. Sci-Kit Learn

La herramienta *Sci-Kit* de *Python* permite la implementación de diferentes métodos de *machine learning*. Esta implementación está basada en *Numpy* y *SciPy* que son herramientas utilizadas en el entorno de programación de *Python*. La mayoría de sus métodos tiene énfasis en *Data Mining* y Análisis de Datos. El módulo permite ciertas acciones como:

- Clasificaciones

- Regresiones
- Cluster
- Reducción de dimensionalidad
- Selección de modelo
- Pre-procesamiento

6.7. Validación cruzada

Validación cruzada corresponde a una técnica estadística implementada en los algoritmos de aprendizaje de computadoras para la selección de los datos de entrenamiento y prueba. Para poder realizar este tipo de pruebas se tiene que realizar con base en algoritmos supervisados [27].

- Datos de entrenamiento:
Este es el grupo de datos que se tiene el conocimiento de la clasificación por parte del usuario y de la computadora. Ya que estos se brindan al ordenador, para que use esta información para generar las clasificaciones.
- Datos prueba:
Corresponden al apartado de los datos que no fueron incluidos en el entrenamiento. De estos solo el usuario conoce la clasificación verdadera de cada uno de estos. Sirven para probar la clasificación del algoritmo según los datos de entrenamiento. A partir de estos datos se pueden sacar parámetros para medir el desempeño del algoritmo.

La validación cruzada utiliza los grupos de datos de entrenamiento y pruebas para realizar una separación entre 70% - 30% o separar los datos y dejar uno fuera. Sin embargo se enfocará en el tipo de validación cruzada que separa el total de datos en el número de grupos que se deseen. Este tipo de validación cruzada es llamada *K-Cross Validation* [28]. De estos grupo se separa uno de los grupos que será utilizado para pruebas, el resto de grupos se utilizarán como entrenamiento del algoritmo. Este proceso se repite el número de grupos que se separaron para poder tener una mejor medida del desempeño del algoritmo, siendo k el número de grupos a implementar [28].

Entre las ventajas que permite el método de validación cruzada resaltan [27]:

- Estimar el desempeño de un algoritmo.
- Comparar el desempeño de dos algoritmos en torno a una misma base de datos.
- Comparar el desempeño del algoritmo en torno a las características que se utilizan para el entrenamiento y reconocimiento.

6.8. Matrices de confusión

A la hora de la presentación de resultados de algoritmos de clasificación, una de las técnicas más utilizadas para la presentación de datos se conocen como matrices de confusión. Estas matrices de confusión son un conjunto de métodos llamados envolventes, ya que permiten obtener información acerca de la exactitud, sensibilidad y precisión de este [29]. Estos factores no solo nos pueden brindar información acerca de que tan bueno es el algoritmo, sino que puede dar indicios sobre las clasificaciones que se están realizando mal y sobre qué factores a tomar más en cuenta que otros. Cabe mencionar que para poder sacar mejores conclusiones de una matriz de confusión es preferible que haya una distribución normal de las clases, es decir que no existan muchas más muestras de una clase que otras.

Las matrices de confusión pueden ser implementadas especialmente cuando se están analizando algoritmos supervisados. En donde se tiene conocimiento de la verdadera clasificación a la que pertenece el objeto. Las matrices de confusión son de tamaño $n \times n$ en donde n representa el número total de clases. En donde las filas contienen los valores verdaderos de la clase y las columnas los valores que se predijeron por medio del algoritmo. El Cuadro 1 presenta una matriz de 2×2 que intenta clasificar si la bebida es gaseosa o jugo. De esta matriz, se puede analizar que identificar mejor la gaseosa que jugo y por tener la misma cantidad de muestras se puede ver la predisposición del algoritmo. Las cantidades que muestran información de la matriz de confusión ya se mencionaron anteriormente, las cuales son: exactitud, sensibilidad y precisión. Para la explicación de estas se estará utilizando una matriz de confusión de 3×3 para la obtención de información sobre estas.

6.8.1. Exactitud

La exactitud nos puede brindar información sobre la cantidad y porcentajes de que el algoritmo acertó en la clasificación [30]. En la matriz de confusión 2 se obtiene de los datos en la diagonal, marcados en gris.

6.8.2. Sensibilidad

La sensibilidad es una medida en torno a una clase. Esta nos dice según la cantidad de valores reales, que tan bueno es el algoritmo para reconocer los objetos de esta clase [30]. La matriz de confusión 3 resalta las casillas a tomar en cuenta si se quiere obtener la sensibilidad de la clase 1.

		Valores predecidos	
		Gaseosa	Jugo
Valores reales	Gaseosa	7	3
	Jugo	4	6

Cuadro 1: Matriz de confusión

Cuadro 2: Matriz de confusión, obteniendo exactitud

6.8.3. Precisión

Al igual que la sensibilidad brinda información en torno a una clase en específica. Este brinda información acerca de a partir de los valores que el algoritmo predijo que eran de esa clase, cuántos realmente pertenecen a esta. En la matriz 4 se resaltan las casillas a tomar en cuenta si se quiere obtener la precisión de la clase 1.

Cuadro 4: Matriz de confusión, obteniendo precisión

Cuadro 3: Matriz de confusión, obteniendo sensibilidad

En este capítulo se describen los pasos y los requerimientos necesarios para la implementación del programa que se generó. Además de las pruebas que se realizaron para la validación de los algoritmos de detección de rostros y reconocimiento de expresiones faciales. También se menciona cómo se generaron las bases de datos y la descripción de los métodos realizados para seleccionar los datos de entrenamiento y prueba.

7.1. Obtención de datos

Parte importante para el apartado de visión de computadoras son los datos que se utilizan para poder generar los entrenamientos. Para este proyecto se utilizaron distintos datos:

7.1.1. Entrenamientos

Frontal Face Classifier

Este es un archivo de clasificador en cascada. Se utilizó la versión del entrenamiento que viene con la versión de *OpenCV* 3.4. Este archivo está especialmente entrenado para detectar rostros que centran su visión hacia la cámara. En caso de querer generar un entrenamiento en cascada con datos propios es posible por medio de *OpenCV*, sin embargo para esta parte se limitó a utilizar el archivo previamente entrenado. Se utilizó para la detección de rostro con el algoritmo *Haar cascade*.

Marcas de cara

El archivo *Face Landmark Detection* obtenido se generó a partir de los métodos de aprendizaje de computadoras de la librería *Dlib C++ Library*. Este posee la ventaja de que fue generado a partir de un histograma orientado a gradientes, sobre un método holístico. El entrenamiento del archivo marcas de cara fue generado por Vahid Kazemi y Joseph Sullivan en forma de cascada utilizando los métodos de apariencia activa para poder generar un reconocimiento de rostro [31]. En este documento se enfocó principalmente en el alineamiento de la cara, lo cual no solo permite poder detectar rostros que se encuentran observando directamente a la cámara, sino rostros que apuntan hacia otras direcciones. El algoritmo para implementar el archivo marcas de cara, fue generado por Kazemi y Sullivan, sin embargo este archivo fue entrenado por Sagonas y Antonakos utilizando la base de datos *300 Faces IN-THE-WILD* [20].

7.1.2. Base de datos

Así como se utilizó archivos previamente entrenados con un set de datos, se tuvo que generar nuevos entrenamientos para el apartado de reconocimiento de expresiones faciales. Esto hace necesario las bases de datos para poder generar entrenamientos a partir de las imágenes.

Base de datos Cohn-Kanade

La base *Cohn-Kanade* [32] contiene 486 fotos de 97 sujetos diferentes. Esta base de datos fue realizada en base a secuencias. Esto debido a que se tomó una expresión del rostro neutral, y se obtuvo una siguiente foto obtenida en el máximo de la expresión solicitada. Esta base de datos fue generada en un ambiente controlado en la Universidad de Pittsburgh. Esta base de datos contiene información sobre siete expresiones diferentes: Enojo, desprecio, disgusto, miedo, felicidad, tristeza y sorpresa. Una de las grandes ventajas de poder utilizar esta base de datos es que cuenta con una validación de emociones, esto se refiere a que no solo se pidió al usuario que realizara la expresión, sino que se hizo una evaluación si realizó la expresión de forma correcta. Estas imágenes fueron pre-procesadas para que se lograra recortar el rostro y solo se utilizó una imagen de tamaño 350×350 pixeles

Base de datos UVG

Esta base de datos se generó a partir de estudiantes y personal de la Universidad del Valle de Guatemala. Se tuvo la colaboración de 67 personas las cuales se encontraban en los rangos de edad de 18 hasta 66 años. Esta base de datos cuenta con 3,350 fotos ya que a cada sujeto se le pidió que expresara por medio de gestos faciales: enojo, felicidad, sorpresa, disgusto y neutro. Cabe mencionar de que se tomó 15 fotos por cada expresión y se seleccionaron las mejores 10 fotos para cada una de estas. Para esta base de datos, no se realizó una validación de emociones, ya que al querer medir el rendimiento que esta podría tener el tiempo real, se limitó a solicitarle a las personas a expresar el gesto y no a delimitarlo de alguna forma.

Es decir, no se corrigió en el modo que expresaron el gesto, solo se solicitó que fuera más expresivo y se seleccionaron las fotos en donde se tuvo picos más altos de expresividad.

Esta base de datos se generó en el laboratorio de Ingeniería Electrónica/Mecatrónica de la Universidad del Valle de Guatemala. Se realizó con una iluminación de 180 lx. La distancia que se encontraba el rostro del dispositivo de captura era de 80 cm, ver Figura 15. La cámara que se utilizó fue una *Logitech C920 HD*, se obtuvo la captura con un tamaño de imagen de 1920×1080 . Este fue procesada para analizarla con un tamaño de 350×350 para unificarla con las demás bases de datos.

En el mismo ambiente de laboratorio y siguiendo los mismos pasos de cantidad de fotos y expresiones, se realizó una segunda base de datos, para rostros que se encontraban a dos metros de la cámara. Al igual que con la base anterior, se seleccionaron las fotos en donde se encontraban los picos de la expresión. Esta implementación contó con la participación de 20 sujetos, ya que el objetivo era analizar los resultados a distancias bastante alejadas de los entrenamientos a cortas distancias.

Base de datos Ambiente 2

La base de datos *Ambiente 2* fue generada a partir de 31 sujetos en un rango de edad de 11 a 80 años. Esta base de datos cuenta con 1550 imágenes, en donde se reflejan expresiones de: enojo, felicidad, sorpresa, disgusto y neutralidad. Para esta base de datos se tomaron 15 fotos por cada expresión y se seleccionaron las 10 mejores fotos para cada una de estas. Al igual que para la base de datos *UVG*, no se realizó la validación de expresiones.

Esta base de datos fue generada en un ambiente de iluminación de 200 lx. La distancia a la que se realizó este base de datos fue de 80 cm, ver Figura 15. La cámara se colocó a una altura del suelo de 120 cm. El dispositivo utilizado fue la misma cámara utilizada para el ambiente *UVG*. El tamaño de las imágenes obtenidas originalmente fue de 1920×1080 y se procesaron para que se obtuviera una imagen final de tamaño 350×350 con solamente el rostro de la persona.

Base de datos personalizada

Esta base de datos se realizó en la misma localización que la base *Ambiente 2*, se mantuvieron la mismas condiciones de iluminación y distancias que se tenían de la cámara. Sin embargo en esta se limitó a la obtención de datos de un solo sujeto, por lo cual las imágenes para cada expresión aumentaron a 30 por gesto. De esta manera el total de estos datos fue de 150 imágenes.

7.2. Pruebas

En esta sección se describieron las pruebas realizadas tanto para la detección de rostros como para el reconocimiento de expresiones. Debido a que existen factores como el fondo



Figura 15: Vista de lado obtención bases de datos

y luz los cuales afectan el rendimiento del programa, se utilizaron dos ambientes diferentes para realizar las primeras pruebas de detección de rostro.

7.2.1. Prueba ambiente abierto

La prueba en la plaza consistió en un fondo el cual tendría que ser móvil, por lo cual se eligió este lugar. Para la prueba, se midió la distancia máxima a la cual los programas tenían alcance y las limitaciones en cuanto existen uno o varios sujetos en el cuadro. La prueba se limitó únicamente a la detección de rostros, ya que debido a la incertidumbre de paso de las personas y otros factores, afectaba el rendimiento del reconocimiento de expresiones.

La prueba del ambiente abierto se realizó en la Plaza Paiz Riera, Universidad del Valle de Guatemala. Estas se realizaron cercano a las 14 horas para que las condiciones de iluminación fueran muy similares. En estas pruebas se utilizaron ambas implementaciones, el algoritmo que utiliza los clasificadores *Haar cascade* y el algoritmo que utiliza las marcas de cara. El dispositivo utilizado fue un cámara *Logitech C920 HD*, con un tamaño de imagen de 640×480 . La cantidad de cuadros por segundo, se definió automáticamente por el programa, dependiendo del tiempo que le tardara el procesamiento de cada uno de los cuadros. La cámara se colocó a una altura de 120 centímetros del suelo y realizó la prueba. La vista en planta de las dimensiones y espacio que se realizó la prueba se puede observar en la Figura 16. Los aspectos que se evaluaron se basaron en la series de pasos que se tienen a continuación:

1. El sujeto se colocó a una distancia inicial de metro y medio de la cámara.
2. El sujeto realizó movimientos laterales a velocidad moderada. Siempre manteniendo la vista hacia la cámara.

3. El sujeto realizó movimientos hacia adelante y atrás a velocidad moderada. Siempre manteniendo la vista hacia la cámara.
4. El sujeto debió repetir los pasos 2 y 3 a mayor velocidad.
 - Para pasos 2,3 y 4 aparte de siempre mantener la vista hacia la cámara, los movimientos se realizaron en un radio de 1 m que estuvo delimitado, ver Figura 16. En estos se analizó si el algoritmo fue capaz de darle seguimiento al rostro por medio de grabación de videos de 10 segundos para los tres pasos.
5. El sujeto se colocó en el punto inicial y bloqueó parte de su rostro, simulando el gesto de “cubrirse las manos con el rostro debido al sol”.
6. El sujeto caminó alejándose de la cámara, hasta el punto que el algoritmo ya no pudo detectar el rostro.
7. Se agregó otro sujeto a la prueba y tuvo que repetir los pasos 2 al 6.
8. Se repitieron los pasos del 1 al 7 para el siguiente algoritmo (Se realizó la primera prueba con el algoritmo basado en clasificadores *Haar cascade* y luego realizó la prueba para el algoritmo marcas de cara).

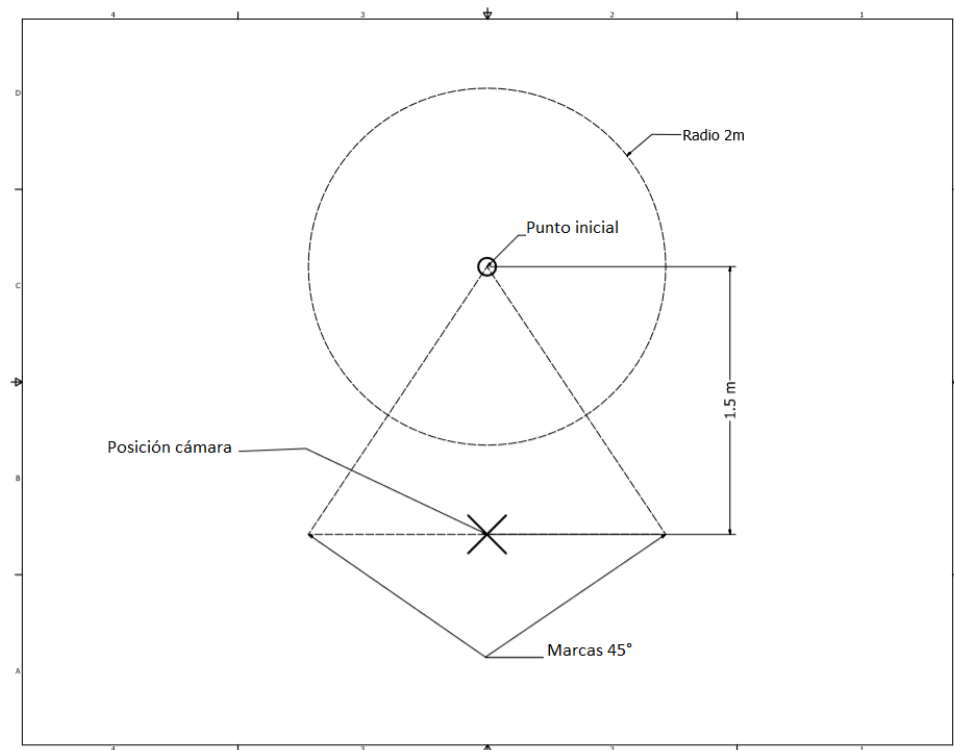


Figura 16: Vista de planta pruebas ambiente abierto

7.2.2. Prueba ambiente cerrado

La prueba de ambiente cerrado, no solo se limitó al seguimiento de rostros, sino que se implementó el reconocimiento de expresiones faciales. Para este segundo aspecto se evaluó el desempeño de la predicción de datos con varios entrenamientos, es decir, con diferentes bases de datos.

Prueba seguimiento de rostro

La primer prueba de ambiente controlado se realizó en el laboratorio de Ingeniería Electrónica/Mecatrónica de la Universidad del Valle de Guatemala. Estas pruebas se realizaron en un horario cercano a las 15 horas, tratando de mantener una iluminación uniforme. La prueba se realizó con todas las luces encendidas del laboratorio. El promedio de iluminación para esta primera prueba fue de 180 luxes. La prueba de seguimiento de rostro se realizó colocando la cámara a una altura de 120 centímetros. Para esta prueba no se analizaron los movimientos laterales ya que al ser un espacio controlado, suelen ser más pequeños. Tampoco se tomó en cuenta tapar parte del rostro con las manos, simulando el bloqueo del Sol. A continuación se describen los pasos para la evaluación de esta prueba:

1. El sujeto se colocó en la posición inicial a metro y medio de la cámara.
2. El sujeto rotó su rostro hasta el punto en donde el programa no fuera capaz de detectar el rostro o hasta una marca determinada de 45° .
3. El sujeto se alejó de la cámara hasta el punto en donde no logró detectar el rostro.
4. Se repitieron los pasos 1, 2 y 3 agregando otro sujeto a la prueba.
5. Se repitieron los pasos del 1 al 4 para el siguiente algoritmo. Primero se realizaron las pruebas para el algoritmo *Haar cascade* y luego se realizó la prueba para el algoritmo marcas de cara.

La segunda prueba del seguimiento de rostro se realizó en el *Ambiente 2*, donde se recopilaron los datos de la base de datos *Ambiente 2*. La prueba de seguimiento se realizó en un ambiente controlado de luz, en donde se tuvo una iluminación promedio de 200 lx. El dispositivo de captura utilizado fue el mismo y se colocó a una altura de 120 centímetros. En esta prueba se realizaron los pasos 1 al 5 del apartado 7.2.2.

Para el *Ambiente 2* se realizaron pruebas en cuanto a la distancia de alcance máximo en condiciones de luz no óptimas, con una iluminación promedio por debajo de los 10 lx. Por las malas condiciones de luz, esta prueba se limitó a determinar si el algoritmo era capaz de poder detectar un rostro y la distancia máxima a la cual era capaz de darle seguimiento.

Prueba reconocimiento de expresiones faciales

Las pruebas de reconocimientos de expresiones faciales se realizaron basadas en los datos obtenidos por las bases de datos. Para esto se generaron entrenamientos, que utiliza la computadora para poder predecir las expresiones mostradas.

- Entrenamiento *Cohn-Kanade*:
Esta base de datos contiene los archivos de la secuencia de imágenes como un archivo de texto mencionando la expresión. Para esto se obtuvo una captura de la imagen neutral, como para la expresión en su estado pico. Es por esto que el número de imágenes con neutral era más elevado a las demás expresiones. Para evitar que se tuviera algún tipo de sesgo, se utilizó la validación cruzada con una $k=10$. Esto significó que se subdividió el grupo de imágenes en 10 grupos aleatorios, de los cuales uno se utilizó para pruebas y el resto se utilizó para entrenar el algoritmo. Se realizaron 10 diferentes corridas para validar los resultados de cada uno de los grupos que se seleccionaron.
- Entrenamiento base de datos *UVG, Ambiente 2 y Personalizado*:
Para estas bases de datos se tuvo una estructura que en cada folder con el nombre de cada personas se tenía cinco diferentes carpetas con las cinco expresiones mostradas. Enumeradas desde 1 hasta el número de imágenes que se tuvo. Para estas bases de datos se tuvo igual número de imágenes por sujeto de cada gesto, siendo 50 imágenes por persona, 10 por expresión. A pesar de que se tuvo el mismo número de imágenes por cada expresión se utilizó el mismo método implementado con el entrenamiento *Cohn-Kanade* de validación cruzada con un $k=10$. La diferencia es que en lugar de juntar todas las imágenes y subdividirlos en grupos se diez, se generó un arreglo aleatorio con todos los nombres de las personas de la base de datos y estos se separaron en diez grupos para la realización de las pruebas. Esto con el fin de que se pudiera generalizar de mejor manera los resultados.

En este apartado, los datos que se mencionan que fueron utilizados para el entrenamiento y para pruebas, provienen de las imágenes obtenidas en las bases de datos. Para esto se tuvo que convertir la imagen en un escala de blanco y negro entre un número de 0 a 255 que representaban la intensidad del color. A partir del número que representaba cada uno de los pixeles se generó un vector por cada una de las imágenes representadas. Al tener un gran grupo de imágenes se formó una matriz en la cual cada una de las filas representó la información de cada una de las imágenes.

Uno de los métodos por el cual se obtuvo los resultados de las pruebas fue por medio de las matrices de confusión, la cual se observa en el Cuadro 5, un ejemplo de la matriz generada para los entrenamientos de cinco expresiones. Estas matrices fueron generadas para cada unas de las pruebas realizadas, tanto para el algoritmo de *Fisher Face* como para las máquinas de vectores de soporte. Debido a que se tuvo conocimiento previo del gesto, se corrió el programa para predecir el gestos que este reconocía, así se fue realizando con los grupos de pruebas generados por la validación cruzada. A partir de estos se generaron la matrices de confusión ya que se conocían ambos valores, los valores reales y los valores predichos.

Pruebas específicas algoritmo *Fisher Face*

En el caso del algoritmo *Fisher Face* se utilizaron los diagramas de dispersión y la reconstrucción de imágenes por medio de *Eigen-Rostros* y *Fisher Face*. Esto fue posible en parte gracias a que los métodos realizaron una reducción de orden, la cual permitió obtener

este tipo de datos. Este tipo de datos junto con las matrices de confusión permitió el análisis gráfico de la separabilidad de las expresiones y las características a tomar en cuenta.

- Diagramas de dispersión:
 Para la generación de estos se utilizó la ayuda de la librería *Sci-Kit* para la representación gráfica de la clasificación generada por los algoritmos. Debido a que el algoritmo *Fisher Face* basó su funcionamiento en análisis del componente principal el cual se pudo obtener por medio de resolución la matriz de valores propios. Se seleccionó por medio de esta técnica los componentes principales que se utilizaron para la clasificación de datos. Debido a que para esta sección se utilizó la instancia del algoritmo en *Sci-Kit* se obtuvo una gráfica de dispersión a través del comando "plt.scatter", el cual devolvió los datos representados a partir de los valores de los componentes principales de la imagen. Este método se utilizó tanto para *Eigen-Rostros* como para *Fisher Face* para ver la dispersión a partir de las transformaciones de la matriz de datos generada por la intensidad de los pixeles de cada una de las imágenes.
- Reconstrucción de imágenes:
 Este elemento se utilizó para el algoritmo *PCA* como para el algoritmo *LDA* implementados en el programa *Fisher Face*. Sin embargo debido a la representación visual que presentan estos se utilizó principalmente para la determinación de la cantidad de componentes principales a seleccionar en el paso del algoritmo *PCA*. Como este algoritmo obtuvo los valores propios de la distribución, se obtuvo los vectores propios del grupo de imágenes. Estos sirvieron para la reconstrucción de de la imagen, por medio del método en *Sci-Kit* llamado "pca.inverse_transform()", el cual revirtió el proceso realizado, tomando en cuenta solo los componentes principales seleccionados y no todos los componentes de la imagen. Ya que en caso que se utilice todos los componentes se obtiene la imagen 100% original. Este proceso se realizó para *LDA* cómo *PCA* al momento de la reconstrucción se obtuvo diferentes resultados, en parte debido a las diferentes transformaciones que se realizaron en las matrices para cada uno de los algoritmos.

Para ambos procesos de diagramas de dispersión y reconstrucción de resultados se presentaron según diferentes número de componentes utilizados en el proceso del algoritmo *PCA*. Se inició mostrando los resultados con base en 10 vectores propios y se fue subiendo hasta el valor de 100 vectores propios. Ya que sobre el algoritmo *PCA* se logró obtener el control del número de componentes a utilizar a diferencia del algoritmo *LDA*, en donde el número de componentes lo reduce al número de clases menos uno.

		Valores predichos				
		Neutral	Feliz	Enojo	Sorpresa	Disgusto
Valores reales	Neutral					
	Feliz					
	Enojo					
	Sorpresa					
	Disgusto					

Cuadro 5: Matriz de confusión implementada

Como ya se mencionó en el capítulo de metodología para la parte de detección de rostros se utilizó en dos ambiente diferentes. Uno en un ambiente abierto (Plaza Paiz UVG) y otros en un ambiente controlado. Para la prueba de la plaza se analizaron cuatro situaciones principales:

- Movimiento laterales
- Movimientos agresivos
- Obstrucción de rostro
- Distancia máxima

8.1. Pruebas Plaza Paiz

8.1.1. Aplicación histograma

Una de las variables con mayor incidencia en el algoritmo es la luz, debido a que la cámara web cuenta con un función de auto-enfoque que abre o cierra el lente dependiendo de la cantidad de luz que detecta y hacia donde quiere enfocar. Se tiene diferentes lecturas de los cuadros en el caso de un ambiente no controlado. Estas diferentes lecturas no solo afectan el enfoque que tiene, sino el tiempo de respuesta de cuadros por segundo, el tiempo computacional que se va a tardar y la calidad y tamaño que envía la imagen.

Para analizar más detalladamente la variable de luz se decidió realizar una prueba en el ambiente controlado la cual no tuvo el mejor desempeño. La Figura 17 a) muestra la imagen

de un sujeto con la captura normal de la cámara. En esta se observa que la cámara no es capaz de enfocar correctamente el rostro, por lo cual el programa no lo logra detectar de manera correcta el rostros. La solución de este problema se obtuvo evitando los reflejos y las intensidad de luz demasiado altas, además en lugar de analizar el cuadro captado por la cámara en blanco y negro, se realizó una modificación previa para análisis. Esta modificación consistió en generar una ecualización de histograma [11], lo cual hace que resalte de mejor manera los contrastes de la imagen. Este proceso se observa claramente en la Figura 17 b).

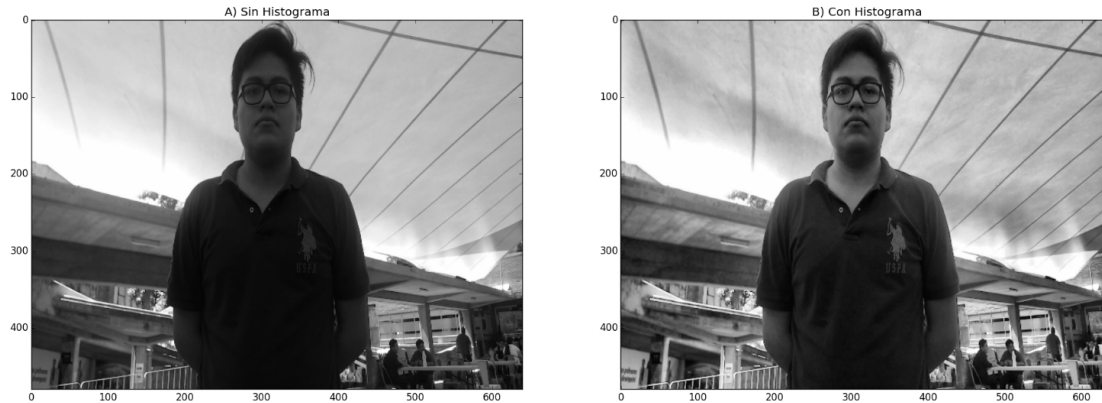


Figura 17: Aplicación de la técnica ecualización del histograma en Plaza Paiz a) Sin ecualizador b) Con ecualizador

Para el ser humano observado la Figura 17 se ve claramente una diferenciación en el reconocimiento de la imagen, sin embargo se analizó si la diferencia es igual de perceptible para el ordenador que para el ojo humano. En esta prueba se determinaron los porcentajes de detección de rostros del algoritmo, aplicando la ecualización de histograma en ambos algoritmos. Los resultados se observan en el Cuadro 6

Algoritmo	% Cuadros correctos	% Cuadros incorrectos
<i>Haar cascade</i>	38.5 %	61.5 %
<i>Haar cascade con histograma</i>	83 %	17 %
Marcas de cara	86 %	14 %
Marcas de cara con histograma	96 %	4 %

Cuadro 6: Porcentajes de detección de rostros para algoritmos *Haar cascade* y marcas de cara en Plaza Paiz con aplicación de ecualización de histograma.

Analizando los datos obtenidos en el Cuadro 6 se puede determinar una mejora de detección de rostros aplicando la ecualización con histograma. Esta técnica mejoró sustancialmente el rendimiento especialmente para el algoritmo *Haar cascade*, ya que al utilizar patrones basados en los contrastes, permite que estos sean aún más reconocibles. Esto es claro ya que pasó de un porcentaje de 38.5 % a un 83 % de detección de rostros. En el caso del entrenamiento marcas de cara, la aplicación del ecualizador presentó una mejora ya que paso de 86 % a 96 %, sin embargo no presentó la mejora sustancial presentada por el algoritmo *Haar cascade*. Esto es debido a que el algoritmo marcas de cara no solo toma en cuenta la detección de patrones por medio de contrastes, sino además toma en cuenta la forma y

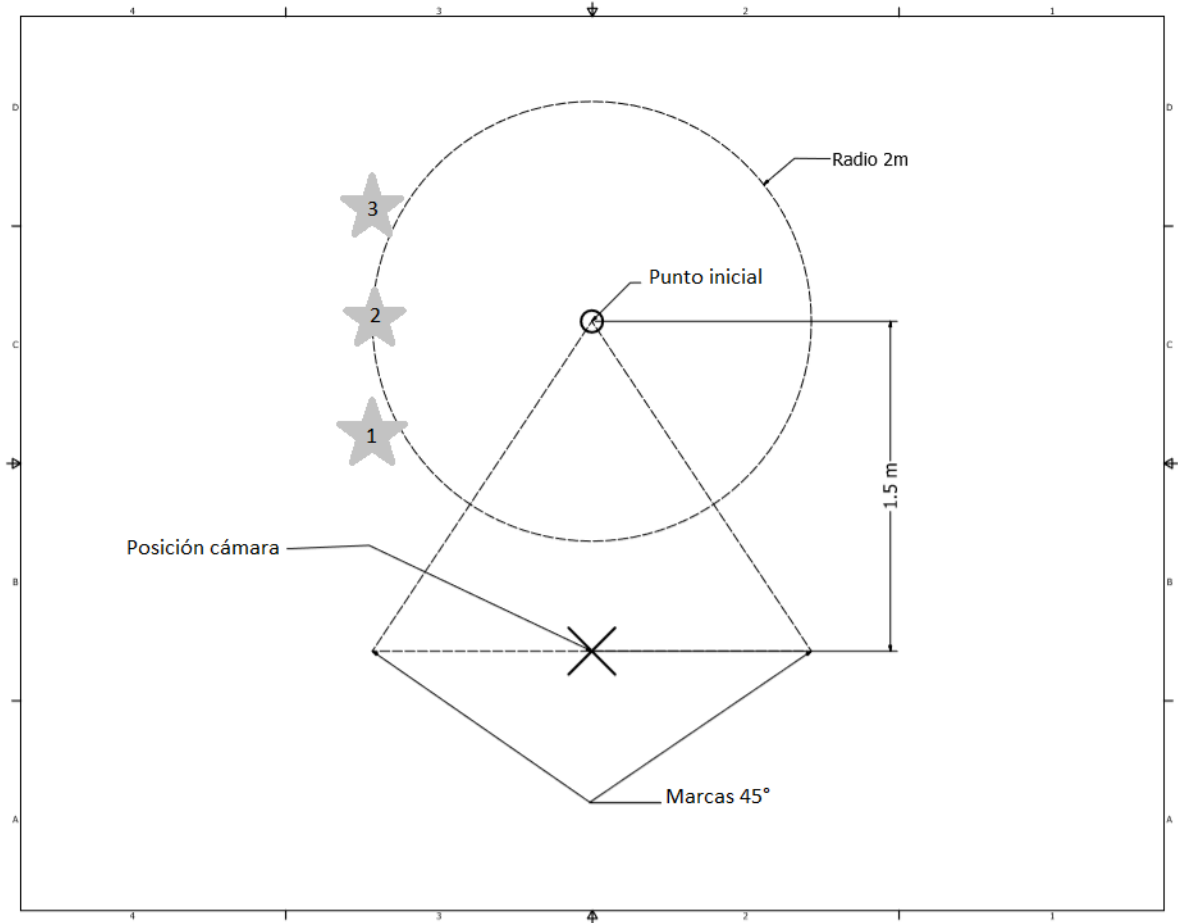


Figura 18: Plano de posicionamiento movimientos laterales 1) Movimiento lateral a 1 m eje vertical 2) Movimiento lateral a 1.5 m eje vertical 3) Movimiento lateral a 2 m eje vertical

apariciencia del rostro.

A pesar que la mejora de rendimiento será mas evidenciabile en ambientes con reflejos o ambientes abiertos en donde la iluminaci3n no se encuentre al 100% controlada, ambos algoritmos presentaron una mejora aplicando la ecualizaci3n de histograma. Por tal raz3n en el resto de pruebas de esta secci3n y futuras se aplic3 el ecualizador con histograma para tratar de buscar el mejor desempe1o en cada una de las pruebas.

8.1.2. Movimiento laterales

Bas3ndose en el plano, de la Figura 16, se tuvo un 3rea de trabajo en la cual se estuvo analizando el alcance que este pod3a tener, siendo esta un 3rea de radio de 2 m alrededor de un punto colocado a 1.5 m de donde se encontraba la c3mara. En la Figura 18 se observa la disposici3n de las posiciones en las cuales se realiz3 el an3lisis de los movimientos laterales. En las figuras 19, 21 y 23 se observan los resultados para el algoritmo *Haar cascade* y en las figuras 20, 22 y 24 para el algoritmo que utiliza las marcas de cara.



Figura 19: Movimiento lateral a una distancia vertical 1 m de la cámara, algoritmo *Haar cascade*



Figura 20: Movimiento lateral a una distancia vertical 1 m de la cámara, algoritmo marcas de cara



Figura 21: Movimiento lateral a una distancia vertical 1.5 m de la cámara, algoritmo *Haar cascade*



Figura 22: Movimiento lateral a una distancia vertical 1.5 m de la cámara, algoritmo marcas de cara



Figura 23: Movimiento lateral a una distancia vertical 2 m de la cámara, algoritmo *Haar cascade*



Figura 24: Movimiento lateral a una distancia vertical 2 m de la cámara, algoritmo marcas de cara

Algoritmo	% Cuadros correctos	% Cuadros incorrectos
<i>Haar cascade</i>	77.66 %	22.34 %
Marcas de Cara	94.66 %	5.34 %

Cuadro 7: Porcentajes de detección para movimientos agresivos

De las figuras 19 y 20 se puede determinar que ambos algoritmos ya no fueron capaces de reconocer rostros en esas posiciones del campo de visión, que se encuentran a una distancia vertical y horizontal de un metro de la cámara. Sin embargo esto no es debido al desempeño del algoritmo sino está más limitado al campo de visión que puede tener la cámara, ya que tanto en la Figura 19 y 20 solo se cuenta con una parte del rostro, la cual no es suficiente para identificarlo.

En el caso de las figuras 21 y 22 es aún más evidente que la no detección de los resultados a una distancia de 1 m es debido al campo de visión de la cámara. Ya que en la distancia de 1.5 m ambos algoritmos logran obtener una identificación positiva del rostro. El mismo resultado se obtuvo para las distancias de 2 m en las figuras 23 y 24. A partir de los resultados de las figuras 21, 22, 23 y 24 se puede determinar que el usuario puede realizar movimientos laterales y ambos programas serán capaces de poder darle un seguimiento, no existe la necesidad de que se encuentre al centro de la cámara para poder detectarlo.

8.1.3. Movimientos agresivos

Para los movimientos agresivos se realizaron movimientos laterales y frontales dentro del radio de trabajo predefinido en la Figura 16. El en Cuadro 7 muestra los resultados obtenidos para ambos entrenamientos.

En el Cuadro 7 se puede determinar cómo el rendimiento del algoritmo marcas de cara, es superior al algoritmo *Haar cascade*. Con una diferencia bastante considerable del 17%. Analizando tanto la tabla, como los vídeos de los cuales se obtuvieron los resultados, en los momentos que se tiene fallo con el algoritmo *Haar cascade* es cuando el sujeto no tiene el rostro observando directamente hacia la cámara. Si bien se indicó que se debía tener la mirada siempre enfocada hacia la cámara, fue uno de las variables que no se pudo controlar debido a la velocidad y agresividad de los movimientos.

A pesar de que el algoritmo basado en las marcas de cara, tuvo un mejor desempeño hay que destacar que los fallos que tuvo, fue cuando se dieron los cambios de dirección, ya que en este momento el cuadro se aprecia borroso y el auto enfoque de la cámara no es lo suficientemente capaz de ajustarse a la velocidad que se está dando el cambio de dirección. Otro de los momentos en los que se dio los errores, fue cuando observaba hacia la cámara pero el rostro no se encontraba alineado.

Otro de los aspectos que llamó al atención es que el algoritmo marcas de cara no presentó ningún falso positivo a diferencia del algoritmo *Haar cascade* ya que este en los momentos de cambio de dirección, presentaba que se detectaba más de un rostro. A pesar de que podía estar reconociendo el rostro de la persona, si el algoritmo reconocía más de un rostro los resultados se tomaban como cuadro incorrecto.

Algoritmo	% Cuadros correctos	% Cuadros incorrectos
<i>Haar cascade</i>	0 %	100 %
Marcas de cara	0 %	100 %

Cuadro 8: Porcentajes de detección para obstrucción del rostro

8.1.4. Obstrucción del rostro

Como se mencionó en el apartado de metodología, para esta prueba se buscaba analizar los resultados cuando la persona se cubría parte del rostro debido al Sol, los resultados de esta prueba fueron negativos en su totalidad, ya que ambos algoritmos no lograron tener ninguna lectura correcta cuando se obstruía parte del rostro con las manos simulando taparse del Sol. Estos resultados se evidencian en el Cuadro 8 en donde el 100 % de los cuadros analizados con obstrucciones, no se logró detectar el rostro.

Estos resultados están validados ya que para la detección de los rostros como tal, se tiene que tener algún rostro para detectarlo y al no poder tener un rostro completo, no es posible reconocerlo. Ya que la detección de rostro se basa en la detección de patrones y al bloquear parte del patrón hace que la computadora según los entrenamientos *frontalface.xml* y *FacialLandmarks.dat* no logre desempeñarse de la mejor forma. Esta es una de las desventajas de enfocarse en la parte del rostro y no hacer un enfoque holístico de todo el cuerpo humano, ya si se enfocara en el reconocimiento del cuerpo completo, cubrir parte del rostro podría no ser lo suficiente como para que el algoritmo no sea capaz de detectarlo. Es decir el porcentaje del objeto bloqueado a detectar representaría un porcentaje menor que si solo se enfoca en el rostro.

A pesar de que en esta prueba no se tuvo ningún sujeto utilizando gorra (elementos que bloquea parte del rostro), en las pruebas en ambiente cerrado se tuvo un sujeto que estaba utilizando esta y los resultados fueron positivos ya que sí logró detectarlo. En caso que se llegara a implementar el sistema en un ambiente abierto en donde el uso de elementos para bloquear el Sol es común, se tendría que realizar un análisis posterior del usos de elementos tales como gorras y viceras para poder generalizar aún mas sobre el uso de estos en los algoritmos presentados.

8.1.5. Distancia máxima

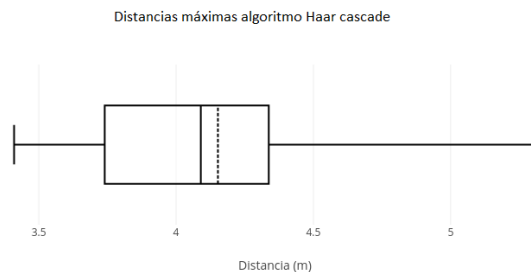


Figura 25: Distancias máximas, algoritmo *Haar cascade* Plaza Paiz

- Valor mínimo = 3.41 m
- $Q_1 = 3.74$ m
- Mediana = 4.09 m
- Media = 4.15 m
- $Q_3 = 4.33$ m
- Valor máximo = 5.31 m

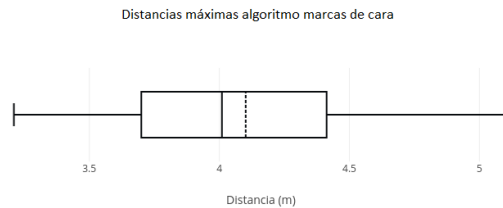


Figura 26: Distancias máximas algoritmo de marcas de cara Plaza Paiz

- Valor mínimo = 3.21 m
- $Q_1 = 3.7$ m
- Mediana = 4.01 m
- Media = 4.10 m
- $Q_3 = 4.41$ m
- Valor máximo = 5.11 m

Los diagramas de caja y bigote representados en las figuras 25 y 26 brindan información acerca de las distancias máximas para cada sujeto además de la distribución que estos tienen. Cabe señalar que el algoritmo *Haar cascade* pareciera tener una mejora en cuanto al alcance que puede tener el algoritmo y esto se evidencia principalmente en los valores máximo y mínimo los cuales ambos son superiores a los presentados por el algoritmo de marcas de cara. El valor mínimo para el algoritmo *Haar cascade* es de 3.41 m, mientras que para el algoritmo de marcas es de 3.21 m. Pareciera que se tiene 20 cm de mejora entre los valores a destacar. Este dato pareciera verse respaldando analizando el valor máximo del Figura 25 con un valor de 5.31 m, comparado con 5.11 m de la Figura 26. Tanto para el valor mínimo como para el máximo habiendo mejorías de 20 cm entre los valores límite.

Sin embargo tomando en cuenta otros valores como lo son la media o mediana de estos datos pareciera que no existe una gran diferencia. En la Figura 25 se tienen valores de 4.09 y 4.15 m de mediana y media respectivamente. Mientras que para la Figura 26 se tienen valores de 4.01 m y 4.10 m habiendo diferencias solamente entre los cinco y ocho centímetros respectivamente. Otra medida que revela desempeños muy similares entre los algoritmos es

que el 50 % de los datos para la Figura 25 se encuentra entre 3.74 m y 4.33 m, mientras que para la Figura 26 el 50 % se encuentra entre 3.7 m y 4.41 m.

Tomando en consideración todos los resultados obtenidos para esta prueba, se puede llegar a determinar que son muy similares entre ambos algoritmos, sin embargo el algoritmo *Haar cascade* se puede llegar a implementar en la búsqueda de una distancia mayor. Estos resultados, junto con los del apartado de movimientos laterales, dan como conclusión una zona promedio de reconocimiento en la cual el algoritmo puede trabajar bajo condiciones de ambiente abierto, ambiente que se puede ver en la Figura 27. En donde las líneas gruesas representan el ángulo de visión de la cámara y el área sombreada gris, representa la zona en donde el algoritmo empieza a presentar limitaciones en cuanto al alcance máximo que este tiene.

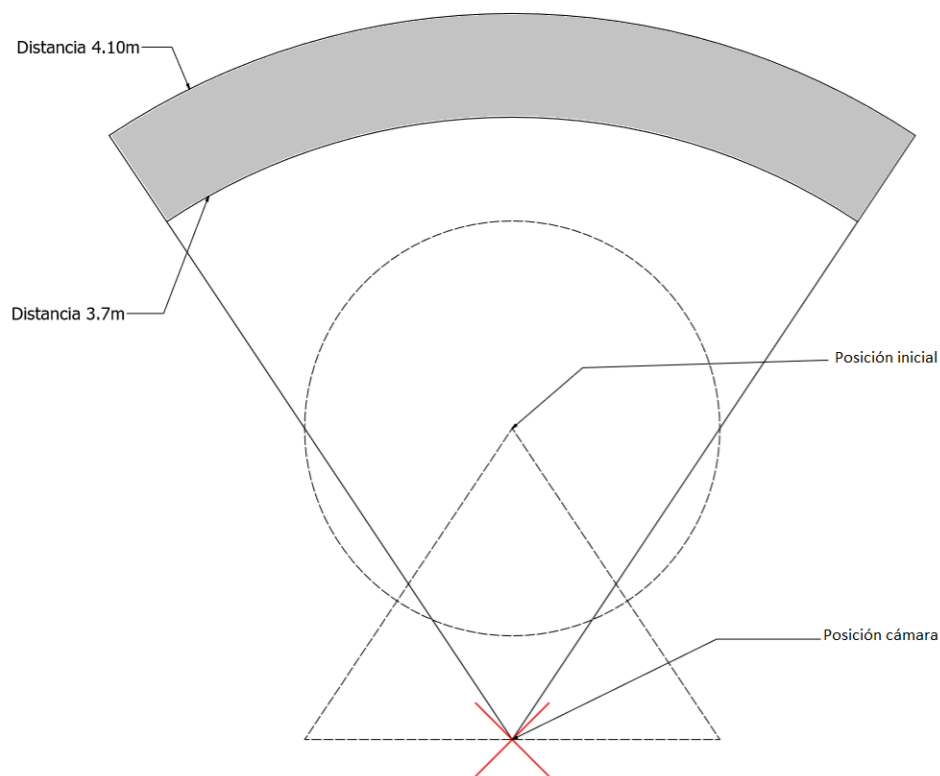


Figura 27: Plano de detección Plaza Paiz

8.2. Prueba ambiente controlado

A diferencia de la prueba de ambiente abierto, en esta sección se analizaron solamente dos situaciones:

- Rotación del ángulo de visión
- Distancia máxima



(a) Algoritmo *Haar cascade*

(b) Algoritmo marcas de cara

Figura 29: Detección rostros con rotación ángulo de visión 45° , ambiente 2

Ya que el a ser un ambiente más cerrado, se disminuirá la cantidad de movimiento lateral, al igual que los movimientos agresivos. No se incluyó tampoco el apartado de obstrucción de rostro, ya que al ser un ambiente de luz controlada, no es necesario instrumentos de protección de luz solar.

8.2.1. Rotación ángulo de visión



(a) Algoritmo *Haar cascade*

(b) Algoritmo marcas de cara

Figura 28: Detección rostros con rotación ángulo de visión 45° , ambiente Universidad del Valle

Las figuras 28, 29 presentan una comparación visual de la detección de rostros de ambos algoritmos. En las figuras 28 y 29 se puede observar, como con los sujetos de prueba se tuvo una detección correcta de rostro solamente con el algoritmo marcas de cara. A pesar de que en la Figura 28b se logra una detección, no todas las marcas de cara están correctamente localizadas. En el apartado de detección de rostro no mostrará ningún inconveniente, sin embargo en la detección de expresiones tendrá efectos por la localización defectuosa de las marcas. A pesar de eso la marca de color blanco en la Figura 28b pareciera localizarse en el

Algoritmo	% Cuadros correctos	% Cuadros incorrectos
<i>Haar cascade</i>	27.5 %	72.5 %
Marcas de Cara	92.5 %	7.5 %

Cuadro 9: Porcentajes de detección de rostros con un ángulo de rotación de 45° para ambiente Universidad del Valle

centro del rostro, lo cual es el principal enfoque de esta sección para la detección de este.

Los resultados completos de la detección de rostros con una rotación de 45°, se presentan en los cuadros 9 y 10, en estas tablas se respalda la información proveída por las figuras 28 y 29 de las cuales se obtiene un mejor rendimiento a partir del algoritmo marcas de cara para la detección de rostros con un ángulo de visión rotado. El algoritmo *Haar cascade* no presenta un buen detección ya que solo es capaz de detectar el 27.5 % de los datos, habiendo una diferencia de 65 % de rendimiento entre ambos algoritmos para el Universidad del Valle.

Parte de estos resultados coinciden con los resultados del análisis de movimientos agresivos para el ambiente abierto, en donde se tuvo malos reconocimientos en los momentos en los que el sujeto no esta dirigiendo su campo de visión directamente hacia la cámara. Como se mencionó parte de este es debido a que el entrenamiento *frontal_face.xml* está diseñado para rostros frontales, a diferencia del entrenamiento *FacialLandmarks.dat* que toma en cuenta la forma. La validación de ambas pruebas permite descartar el elemento de cambio de fondo o condiciones cambiantes de luz para la falta de reconocimiento a rostros rotados. Es una limitación que presentan los entrenamientos a la hora de la detección de rostros con campos de visión de 45°.

8.2.2. Distancia máxima

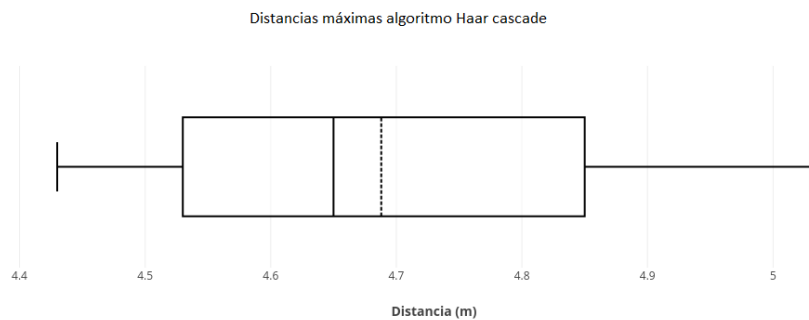


Figura 30: Distancias máximas algoritmo *Haar cascade* para ambiente UVG

Algoritmo	% Cuadros correctos	% Cuadros incorrectos
<i>Haar cascade</i>	22.5 %	77.5 %
Marcas de Cara	91.5 %	8.5 %

Cuadro 10: Porcentajes de detección de rostros con un ángulo de rotación de 45° para *Ambiente 2*

- Valor mínimo = 4.43 m
- $Q_1 = 4.53$ m
- Mediana = 4.65 m
- Media = 4.69 m
- $Q_3 = 4.85$ m
- Valor máximo = 5.03 m

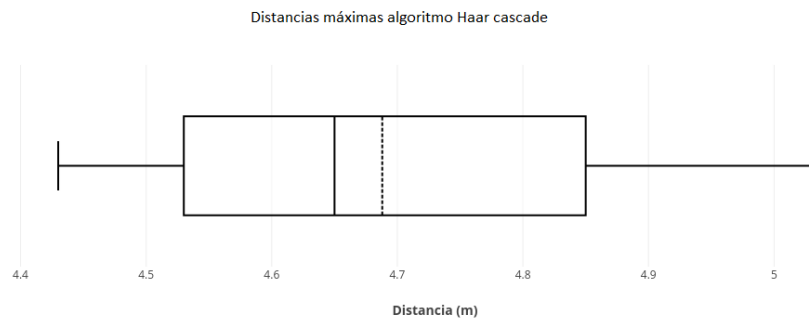


Figura 31: Distancias máximas algoritmo *Haar cascade* para *Ambiente 2*

- Valor mínimo = 4.4 m
- $Q_1 = 4.53$ m
- Mediana = 4.66 m
- Media = 4.72 m
- $Q_3 = 4.84$ m
- Valor máximo = 5.1 m

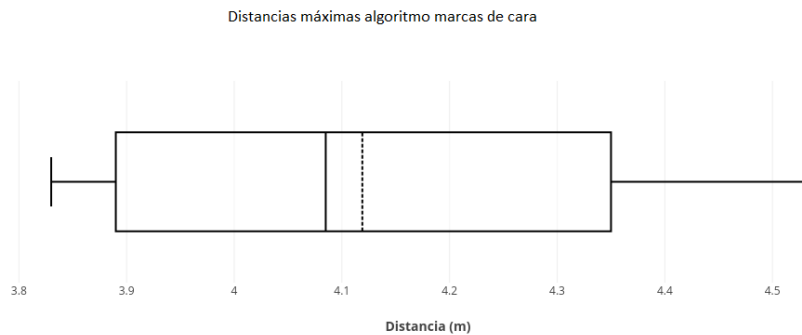


Figura 32: Distancias máximas algoritmo marcas de cara para ambiente UVG

- Valor mínimo = 3.83 m
- $Q_1 = 3.89$ m
- Mediana = 4.01 m
- Media = 4.12 m
- $Q_3 = 4.35$ m
- Valor máximo = 4.53 m

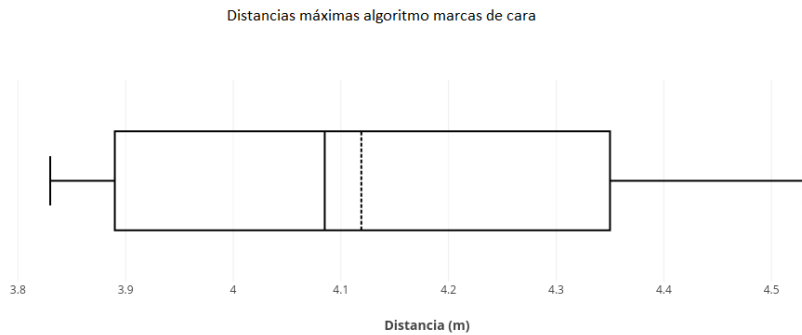


Figura 33: Distancias máximas algoritmo marcas de cara para *Ambiente 2*

- Valor mínimo = 3.81m
- $Q_1 = 3.86$ m
- Mediana = 4.08m
- Media =4.12m
- $Q_3 =4.36$ m
- Valor máximo = 4.67

Para la prueba de distancia máxima se determinó el radio de visión que podía tener el programa para la detección de rostros. Eso se ve reflejado en las figuras 30,31, 32 y 33. Comparando los resultados de ambos algoritmos se puede evidenciar que el algoritmo *Haar cascade* presenta un mejor rendimiento ya que no solo se obtuvo un valor mínimo de 4.43 m muy superior al 3.83 m del otro algoritmo, sino que la diferencia entre medias es de 57 cm. Lo cual representa un mejora entre ambos algoritmos.

A la hora de realizar una comparación entre los datos obtenidos en los ambientes controlados de las figuras 30,31, 32 y 33, con las del ambiente abierto de las figuras 25 y 26 se puede evidenciar que en términos de media, la prueba de ambiente controlado presenta un mejor rendimiento para el algoritmo *Haar cascade*. En el caso del algoritmo marcas de cara se presenta una mejora en la detección del valor mínimo ya que paso de ser 3.41 m en el ambiente abierto a 3.83 m para el ambiente controlado.

Otro de los resultados presentes es el valor máximo obtenido para las pruebas. En la Figura 25 se observa el valor de 5.31 m que fue el más alto para todas las pruebas, sin embargo este fue obtenido para la prueba de ambiente abierto, contrario a lo que se podría pensar la distancia máxima obtenido en el ambiente controlado fue de 5.03 m. Esto lleva a determinar que el límite del algoritmo se encuentra cercano a los 5 m y no es debido al ambiente sino a la definición que se tiene de las características para identificar el rostro a largas distancias. A pesar de que la prueba de ambiente controlado no mejoró en cuanto al valor máximo, se tiene una menor desviación entre los datos presentados para esta área, esto se observa en los valores presentados por los cuartiles uno y tres de las figuras 30 y 32. Por lo cual a diferencia de las pruebas de ambiente abierto en donde se tiene un rango entre 3.41 m y 5.31 m, el rango para las pruebas de ambiente controlado es de 3.83 m a 5.03 m lo cual permite describir de mejor manera las limitaciones físicas del algoritmo.

8.3. Pruebas iluminación menor a 10 lx.

Esta prueba se realizó con pocos sujetos debido a las limitaciones que se obtienen con una iluminación que no supera los 10 lx. Este medida se obtuvo con la aplicación *Light Meter*. Esta prueba se realizó en el *Ambiente 2*, utilizando diferente horario e iluminación para su ejecución a diferencia de la utilizada para la recolección de la base de datos.

Debido a la dificultad de la implementación de estas pruebas, los resultados se limitarán a realizar una descripción de los resultados observados. Durante la implementación de algoritmo *Haar cascade* no se lograron obtener reconocimientos del rostro. Sin embargo para el algoritmo marcas de cara, se lograron obtener capturas, tal y como lo muestra la Figura 34.

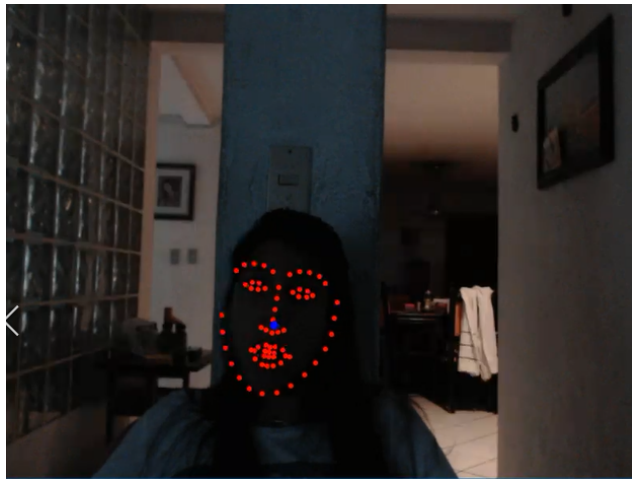


Figura 34: Detección algoritmo marcas de cara para ambiente oscuro

El algoritmo logró mantener el seguimiento de rostro hasta una distancia de 1.67 m de la cámara. Sin embargo a diferencia de los ambientes con mejor iluminación el programa tenía lapsos en los cuales no era capaz de darle seguimiento al rostro, es decir presentaba falsos negativos.

8.4. Pruebas detección de rostro en bases de datos



Figura 35: Imágenes con detección de rostros para bases de datos con algoritmos *Haar cascade* y marcas de cara

Base de datos	<i>Haar cascade</i>	Marcas de cara
Cohn-Kanade	99.91 %	100 %
UVG	85.29 %	99.04 %
Ambiente 2	98.74 %	100 %
Personalizada	100 %	100 %

Cuadro 11: Porcentajes de detección rostros de bases de datos

La detección del rostro para las imágenes de la base de datos resulta en un paso vital para la generación de entrenamientos. Ya que a partir de la detección del rostro se extraen las características, que luego se pueden utilizar para diferentes situaciones, siendo para este trabajo el reconocimiento de expresiones.

En el caso de la Figura 35 donde se muestran ocho imágenes, representando cada una de las bases de datos a utilizar. Estas imágenes representan la expresión de neutralidad expresada en los diferentes ambientes. Como se observa en la Figura 35 se obtuvieron resultados positivos en cuanto a la detección para estas imágenes, tanto para el algoritmo *Haar cascade* como marcas de cara. Sin embargo al observar el Cuadro 11, que muestra la detección del rostro para todas las imágenes de cada uno de las bases de datos, observamos que hubieron imágenes en las cuales no detectó ningún rostro, por lo tanto no formarán a ser parte de la base de datos.

Analizando los resultados del Cuadro 11 se puede determinar que el algoritmo marcas de cara presenta un mejor rendimiento en la detección de estos rostros, ya que en tres de las cuatro bases de datos muestra un porcentaje de reconocimiento del 100 %. Además que en cada una de las bases presenta un resultado igual o una mejora en cuanto al rendimiento. Las imágenes en las cuales no se tuvo resultados positivos fueron aquellas que presentaron alguna rotación con respecto al ángulo de visión recto, ya sea en un ángulo de elevación o inclinación. Esto se observa en la Figura 36 en las cuales no se logró la detección del rostro para ambos algoritmos.



Figura 36: Imágenes con reconocimientos incorrectos de rostro base de datos UVG

En el capítulo 8 se pueden evaluar los objetivos:

- Determinar e implementación del software específico para el reconocimiento y detección de rostros
- Realizar pruebas a 30 sujetos para la detección de rostros y expresiones faciales en dos ambientes diferentes

Se obtuvieron resultados positivos en el caso de detección de rostros y se realizó una comparación entre ambos algoritmos para la determinación de la mejor solución. Se tuvo un gran rendimiento al presentar alcances máximo superiores a los cinco metros en las pruebas de ambiente abierto. También cabe destacar que para ambos algoritmos en estas condiciones se obtuvieron resultados de reconocimiento superiores al 75 %, por lo cual se ha logrado cumplir con el objetivo de determinación que tanto el algoritmo basado en el entrenamiento *frontal_face.xml* y *FacialLandmarks.dat* son parte fundamental de la detección de rostros, junto con las técnicas de procesamiento implementadas para mejorar aún mas el desempeño del algoritmo.

Estas técnicas incluyeron al aplicación de ecualización de histograma cuyos resultados en el Cuadro 6 presentan una mejora en cuenta a la detección de rostros. Tanto para ambos ambientes se pudo determinar las limitaciones que pueden llegar a tener los algoritmos, tales son el caso de los movimientos laterales, movimientos agresivos y rotaciones del ángulo de visión. Parte de estas pruebas permite validar la robustez del algoritmo para ambiente diferentes, lo cual cumple con la realización del objetivo de la implementación de pruebas a 30 sujetos en dos ambientes diferentes.

Pruebas específicas algoritmo *Fish Face*

Este capítulo corresponderá exclusivamente a los procesos que se aplicaron para el algoritmo *Fisher Face*, utilizado para el reconocimiento de expresiones faciales a partir de los métodos PCA y LDA. Como se mencionó en el apartado de metodología, aquí se analizarán las imágenes de todas las bases de datos. Estas imágenes fueron pre-procesadas, ya que se utilizó el método *Haar cascade* para la identificación del rostro y se recortó la imagen para que solo estuvieran características presentes de este. El tamaño de todas las imágenes en esta sección fue de 350×350 píxeles. Este pre-procesamiento de la imágenes facilitó la unificación de criterios a la hora de las técnicas aplicadas. Entre los criterios que resaltaron fueron la distinción de características principales e iluminación. Los resultados presentados en el apartado de *Eigen rostros* y *Fisher rostros* no se separaron por bases de datos.

9.1. Aplicación del algoritmo PCA

9.1.1. Eigen rostros

Como se mencionó en el apartado teórico del trabajo, el primer paso la realización del algoritmo *Fisher Face* es la aplicación del análisis del componente principal, PCA. Ya que este método devuelve los valores propios y vectores propios. En este apartado se centrará en los vectores propios devueltos de la matriz de covarianza a partir de las imágenes ingresadas, que al tratarse de rostros se les llama *Eigen Rostros*. Se obtuvo los resultados para cada una de las bases de datos utilizadas, estos se pueden observar en las figuras 37, 38, 39 y 40. En este apartado se utilizó un número de componentes de 150, por lo cual se debieron de generar 150 *Eigen Rostros*, sin embargo se limitarán a presentar los nueve *Eigen Rostros* principales obtenidos por la técnica PCA, que fueron obtenidos de encontrar los vectores propios principales.

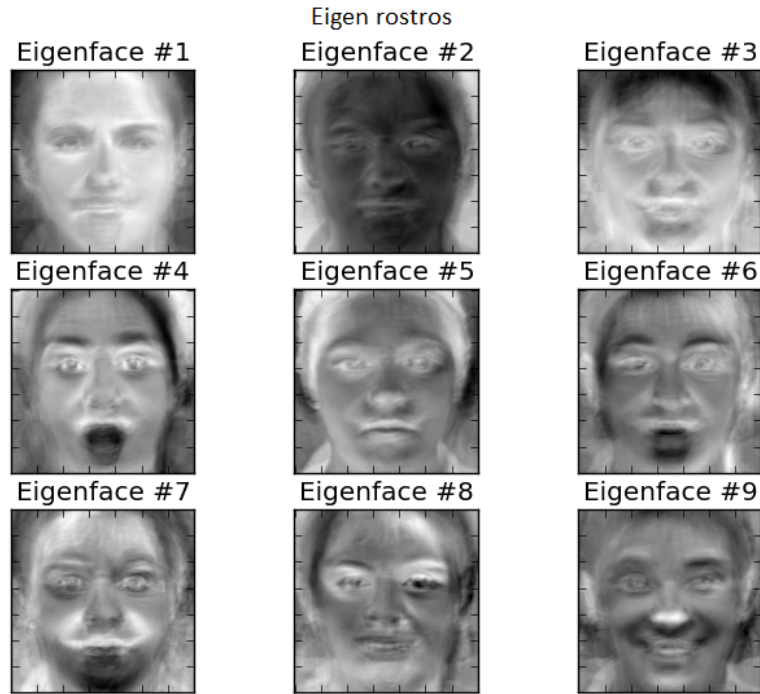


Figura 37: *Eigen rostros* base de datos *Cohn-Kanade*

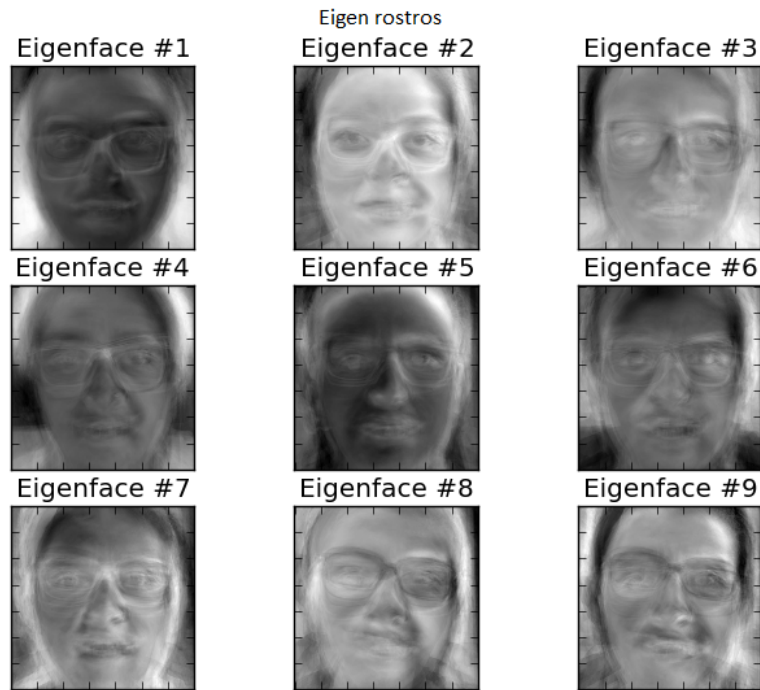


Figura 38: *Eigen rostros* base de datos *UVG*

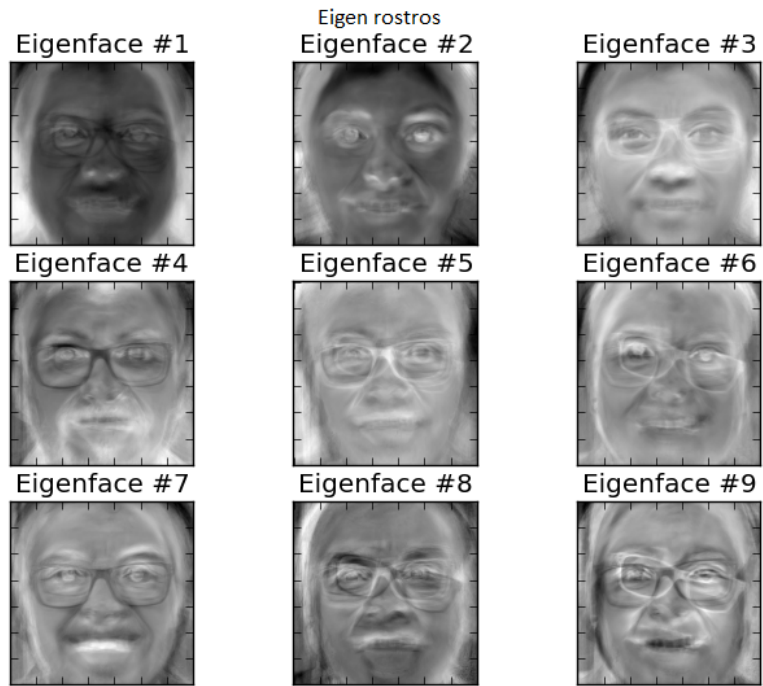


Figura 39: *Eigen rostros* base de datos *Ambiente 2*

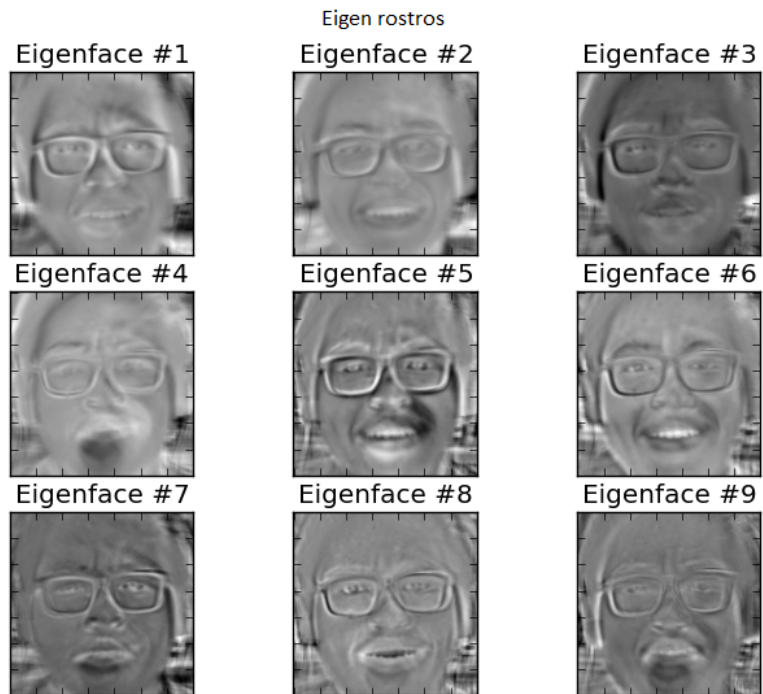


Figura 40: *Eigen rostros* base de datos *Personalizada*

Las figuras 37, 38, 39 y 40 sirven para la clasificación de imágenes en el caso que solo se estuvieran analizando el algoritmo PCA. En el caso de la Figura 40 se obtienen imágenes más uniformes y se logran diferenciar un mayor número de rasgos a diferencia del resto de base de datos. Esto es en gran parte a la variabilidad que contienen las demás bases de datos a diferencia de la Figura 40 la cual es una base de datos personalizada. La misma variabilidad es la que no permite la identificación de patrones claros en las figuras 37, 39 y 38, ya que la morfología de cada una de las caras es diferente a pesar de poder tener rasgos similares. Las posiciones tamaños y orientaciones pueden variar y afectar el desempeño del algoritmo.

9.1.2. Reconstrucción imágenes por *Eigen rostros*

A partir de los *Eigen rostros* generados en la sección 9.1, se puede realizar una reconstrucción de las imágenes, esto puede servir para determinar como parámetro para definir la cantidad de componentes necesarios para la identificación correcta de las imágenes. Los rostros en las figuras 41, 42, 43 y 44 representan las reconstrucción de las imágenes en las bases de datos con base en el número de vectores propios utilizados para la reconstrucción. Para todas las reconstrucciones se utilizaron hasta noventa *Eigen vectores*, ya que estos se limitaban por la cantidad de imágenes.

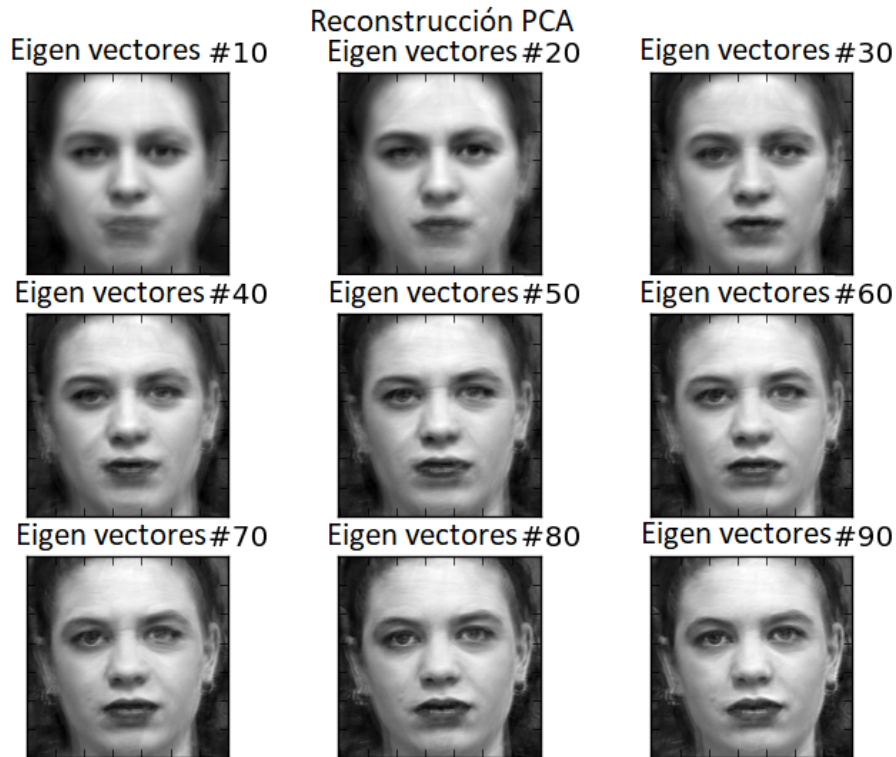


Figura 41: Reconstrucción de imagen por *Eigen rostros* de expresión neutral para base de datos *Cohn-Kanade*

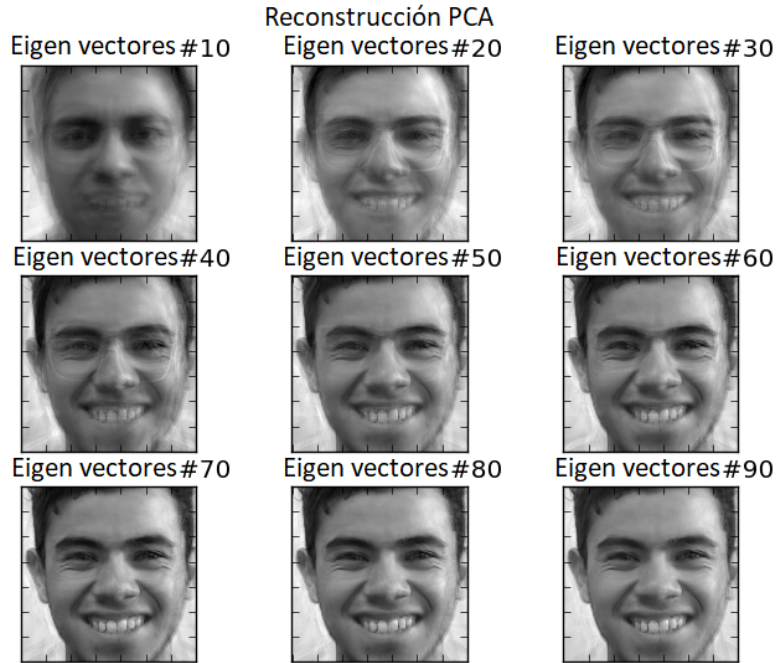


Figura 42: Reconstrucción de imagen por *Eigen rostros* de expresión felicidad para base de datos *UVG*

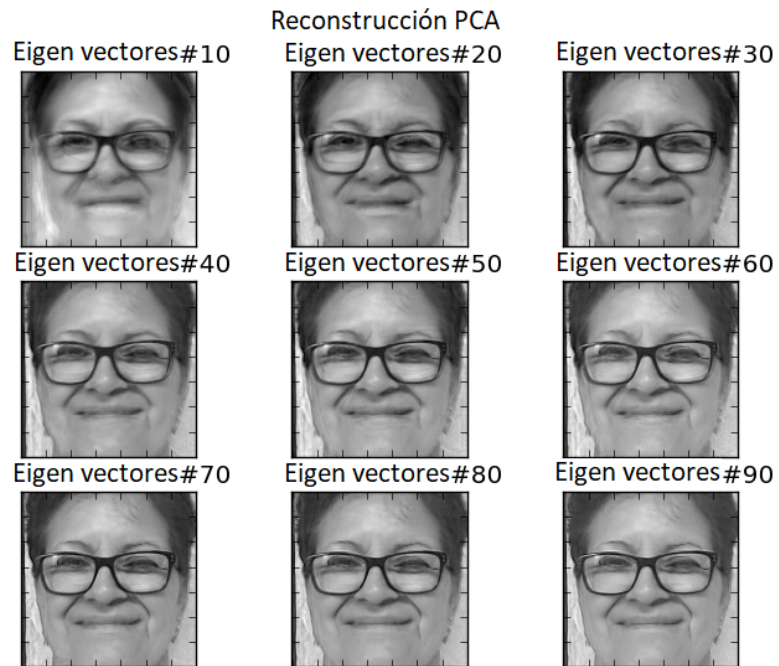


Figura 43: Reconstrucción de imagen por *Eigen rostros* de expresión felicidad para base de datos *Ambiente 2*

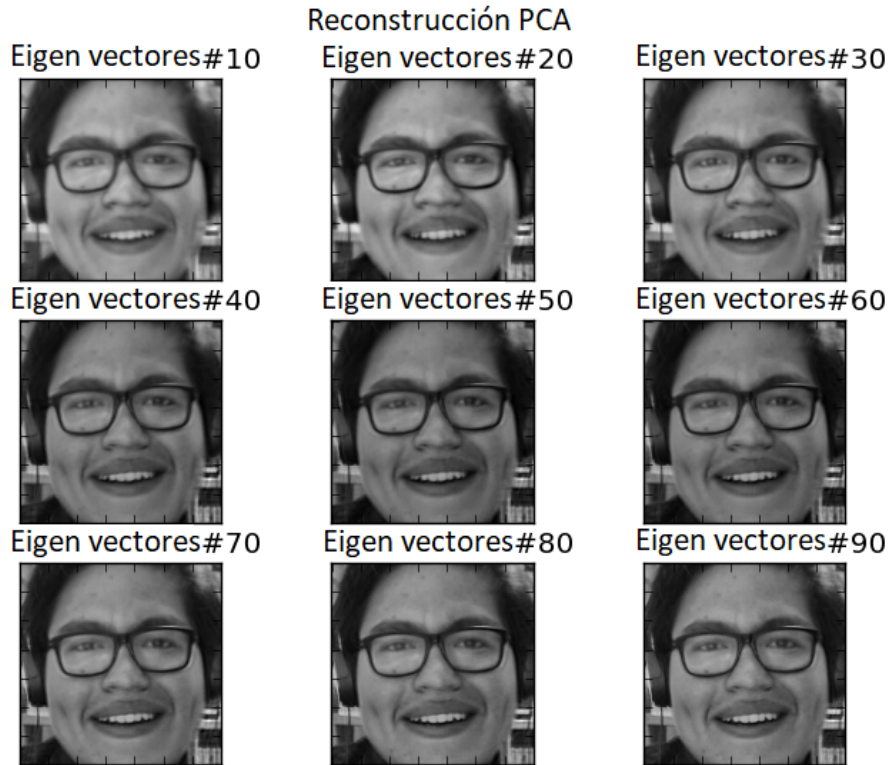


Figura 44: Reconstrucción de imagen por *Eigen rostros* de expresión felicidad para base de datos *Personalizada*

Las base de datos *Cohn-Kanade* y *UVG* son las que mayor número de sujetos de prueba se tienen. Es por eso que al observar las figuras 37 y 42 se empieza a observar de manera clara sin perturbaciones la imagen que se intenta reconstruir hasta que se tiene un número superior a los 50 vectores propios, es decir se necesitan más de 50 componentes principales para su representación. En el caso de la Figura 43 si bien se puede observar la imagen a partir de 20 vectores propios, se pueden observar detalles específicas de la sonrisa hasta que se tienen 40 componentes para su representación. Por último, la Figura 44 a partir de 10 vectores propios se pueden determinar gran parte de la imagen junto con su característica. Ya que no se tiene mucha variabilidad en los rasgos y posiciones al tratarse de una base de datos personalizada.

9.2. Aplicación del algoritmo LDA

Como se mencionó en el apartado teórico, se debe de haber realizado el algoritmo PCA para realizar el algoritmo LDA, debido a que en gran parte de estas situaciones el número de características es muy superior al número de imágenes que se tienen. Los resultados a continuación se realizaron a partir de la aplicación previa de la técnica PCA a partir de 150 componentes, para luego realizar la aplicación del método LDA.

9.2.1. *Fisher rostros*

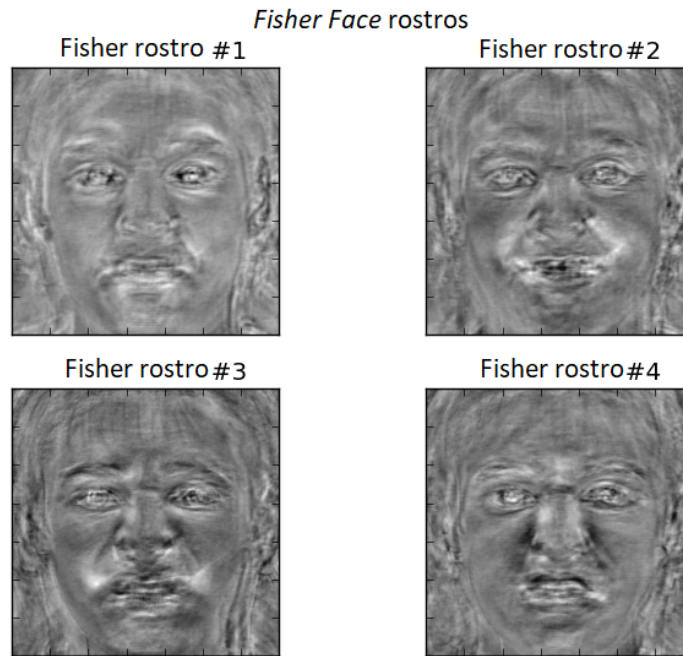


Figura 45: *Fisher rostros* base de datos *Cohn-Kanade*

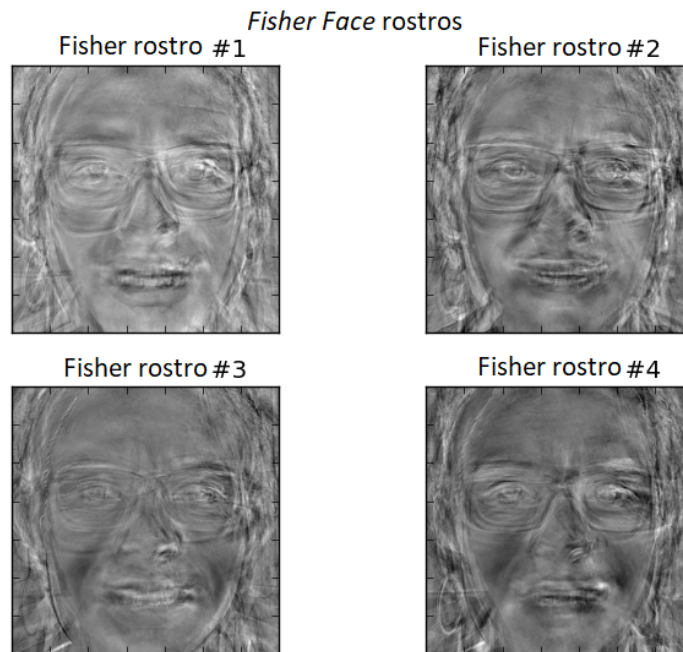


Figura 46: *Fisher rostros* base de datos *UVG*

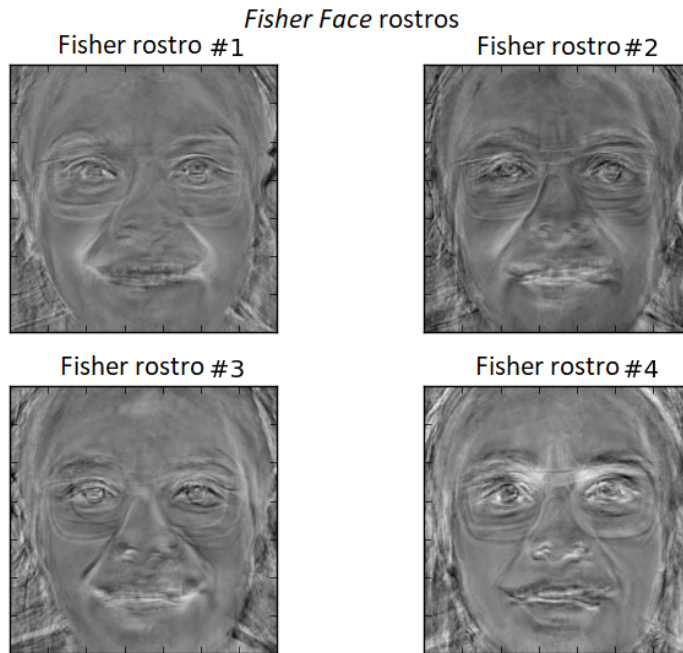


Figura 47: *Fisher rostros* base de datos *Ambiente 2*

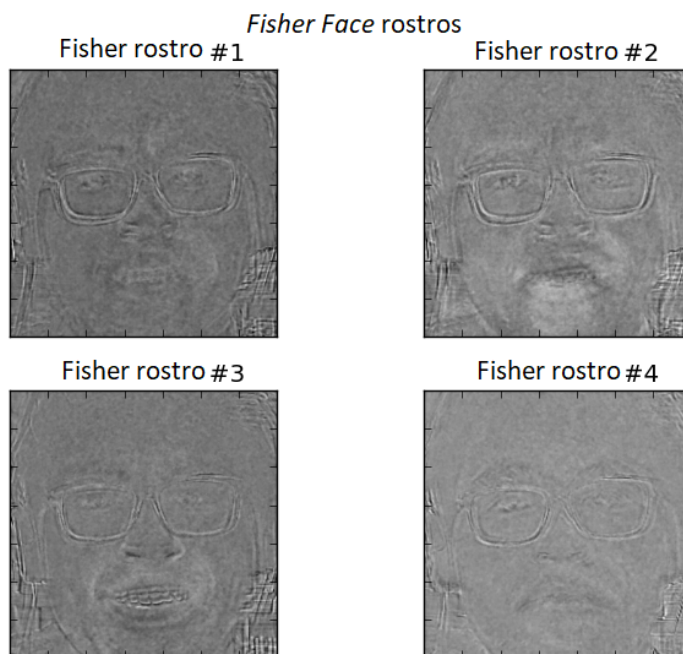


Figura 48: *Fisher rostros* base de datos *Personalizada*

Las figuras 45, 46, 47 y 48 representan los *Fisher Rostros* para las expresiones mostradas para cada una de las bases de datos. Como estas bases de datos cuenta con cinco clases, la reducción de orden que realiza el algoritmos PCA, devuelve una matriz de Orden $C - 1$, siendo C el número de clases. Es por eso que solo se puede observar cuatro *Fisher Rostros* por cada base de datos. Ya que la base de datos se limitó a las expresiones de neutralidad, felicidad, enojo, sorpresa y disgusto.

A diferencia de las imágenes en la sección 9.1 al estar orientado a clasificación resaltará las características del rostros que utilizará para la diferenciación de clases. En estas podemos ver que resalta los aspectos de los ojos, cejas y especialmente de la boca. Debido a que resalta las características diferentes para la clasificación, en el caso del Figura 48 pareciera que no resaltara muchos alguna característica y esto es debido a que se trata de una base de datos personalizado las diferencias son menos perceptibles en comparación si se tiene un grupo variado de sujetos como en el resto de base de datos. En las figuras 46 y 47 se observa que se marcó el rasgo de lentes, ya que gran parte de los sujetos que realizaron la prueba portaron lentes característica que diferencia algunos sujetos.

9.2.2. Reconstrucción de imágenes por medio de *Fisher rostros*

Debido a que cuando se le aplica al técnica LDA se busca no solo la separación entre clases, sino la unión entre datos de la misma. La reconstrucción de imágenes a partir de los *Fisher Rostros* no pareciera ser la mas adecuada, en comparación con los *Eigen Rostros* que si están orientados a la representación.

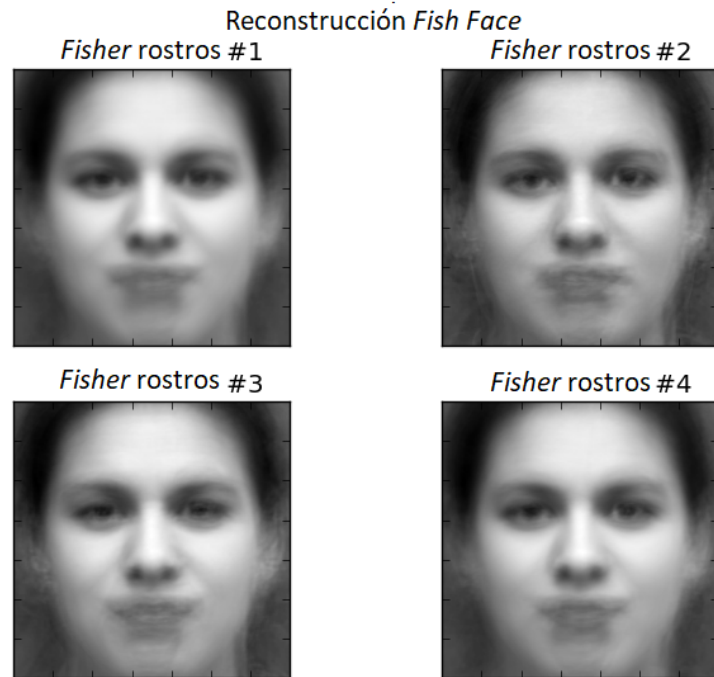


Figura 49: Reconstrucción de imagen por medio de *Fisher rostros* para base de datos *Cohn-Kanade*

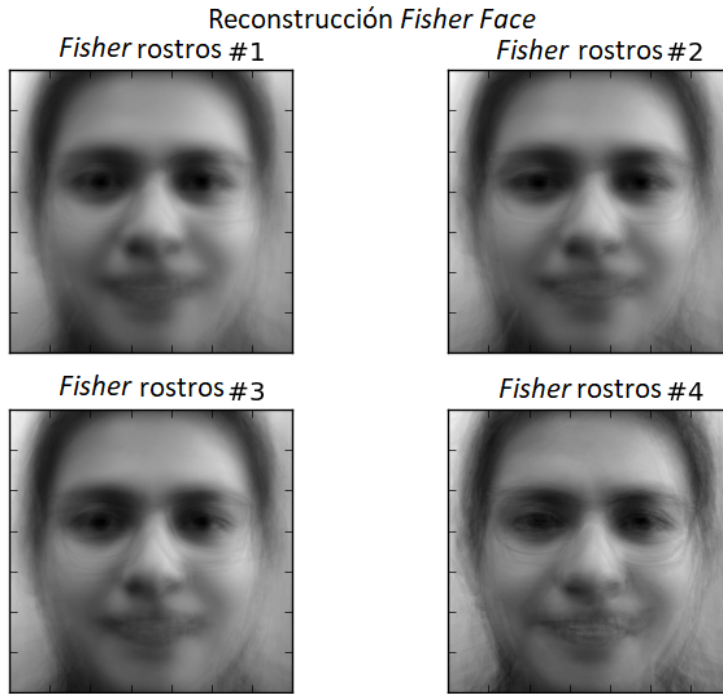


Figura 50: Reconstrucción de imagen por medio de *Fisher rostros* para base de datos *UVG*

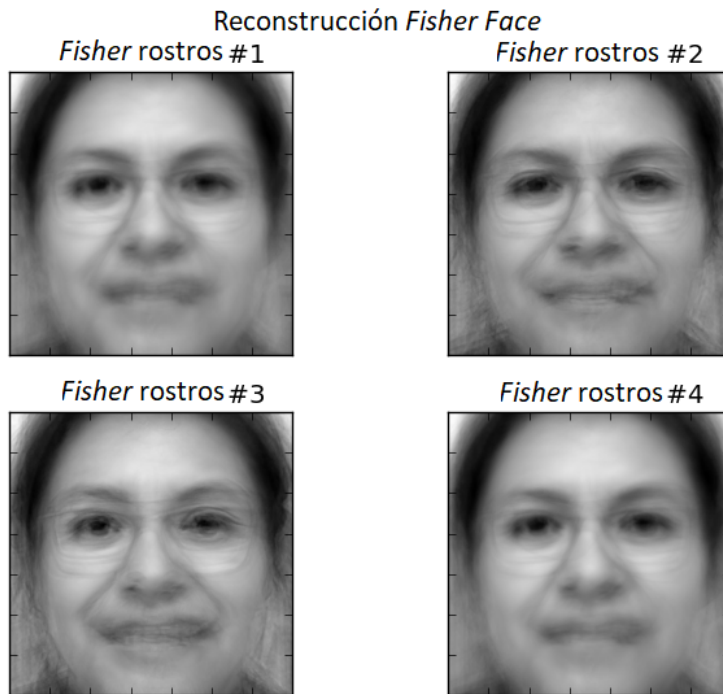


Figura 51: Reconstrucción de imagen por medio de *Fisher rostros* para base de datos *Ambiente 2*

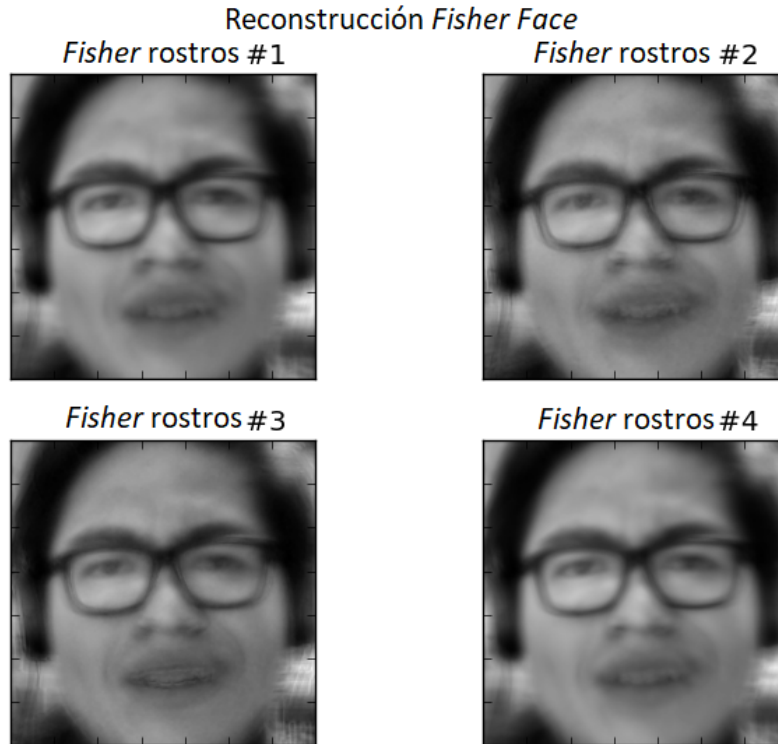


Figura 52: Reconstrucción de imagen por medio de *Fisher rostros* para base de datos *Personalizada*

Las figuras 49, 50, 51 y 52 representan las reconstrucciones de las imágenes a partir de la matriz devuelta por el método LDA. La reconstrucción de las imágenes a partir del métodos LDA tuvo los resultados esperados al no tener una gran reconstrucción de las imágenes, es decir no podemos concluir cual fue la imagen a partir de esta reconstrucción. Incluso en la Figura 52 que se generó a partir de una base de datos personalizada no permite la identificación clara del rostro a mostrar. Cabe mencionar que la reconstrucción de las imágenes en las figuras 49, 50, 51 y 52, se generó a partir de las mismas imágenes utilizadas en la sección de reconstrucción de *Eigen rostros* presentes en las figuras 41, 42, 43 y 44.

9.2.3. Distribución datos de expresiones a partir de aplicación PCA y LDA

A partir de la aplicación de las técnicas PCA y LDA, se puede generar una distribución de los datos. En este caso se generaron distribuciones a partir de los dos componentes principales de separabilidad. Las figuras 53, 55, 57 y 59 representan la separabilidad que existe en la clasificación cuando se toman los 50 componentes principales de la base de datos, mientras que las figuras 54, 56, 58 y 60 muestran la distribución de los datos cuando se utilizan 100 componentes principales. Esta sección se encuentra orientada a la separación de los datos de cada una de las bases para poder realizar las detecciones correctas de las expresiones.

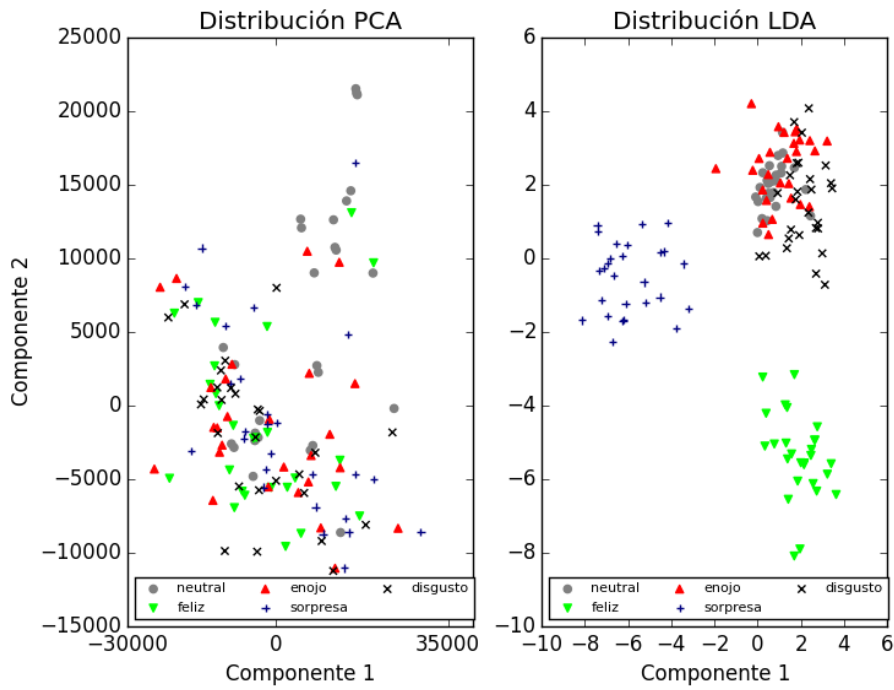


Figura 53: Distribuciones PCA vs. LDA de expresiones: neutral, feliz, enojo, sorpresa y disgusto utilizando 50 componentes para base de datos *Cohn-Kanade*

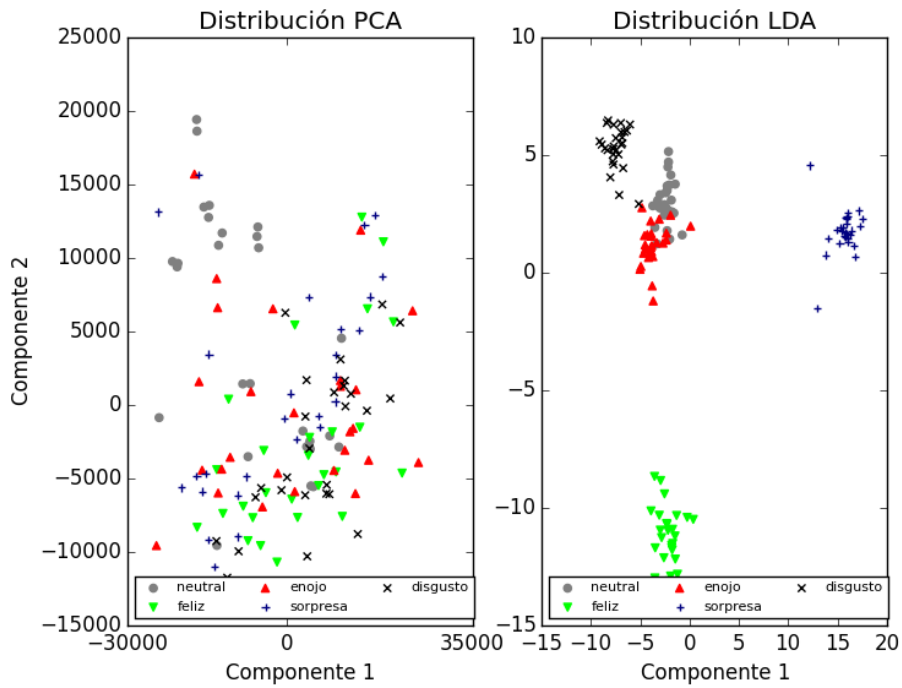


Figura 54: Distribuciones PCA vs. LDA de expresiones: neutral, feliz, enojo, sorpresa y disgusto utilizando 100 componentes para base de datos *Cohn-Kanade*

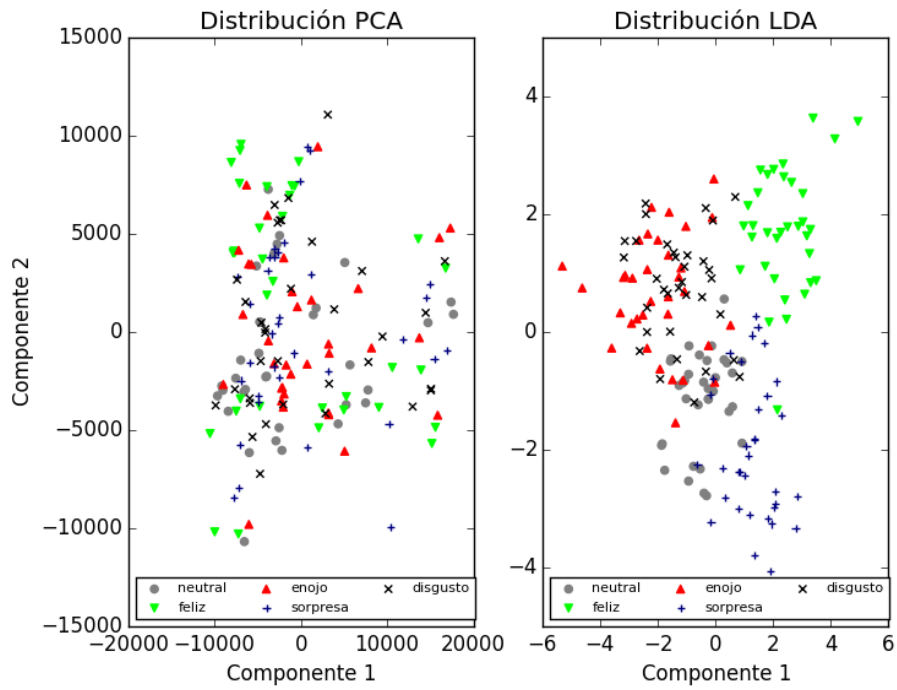


Figura 55: Distribuciones PCA vs. LDA de expresiones: neutral, feliz, enojo, sorpresa y disgusto utilizando 50 componentes para base de datos *UVG*

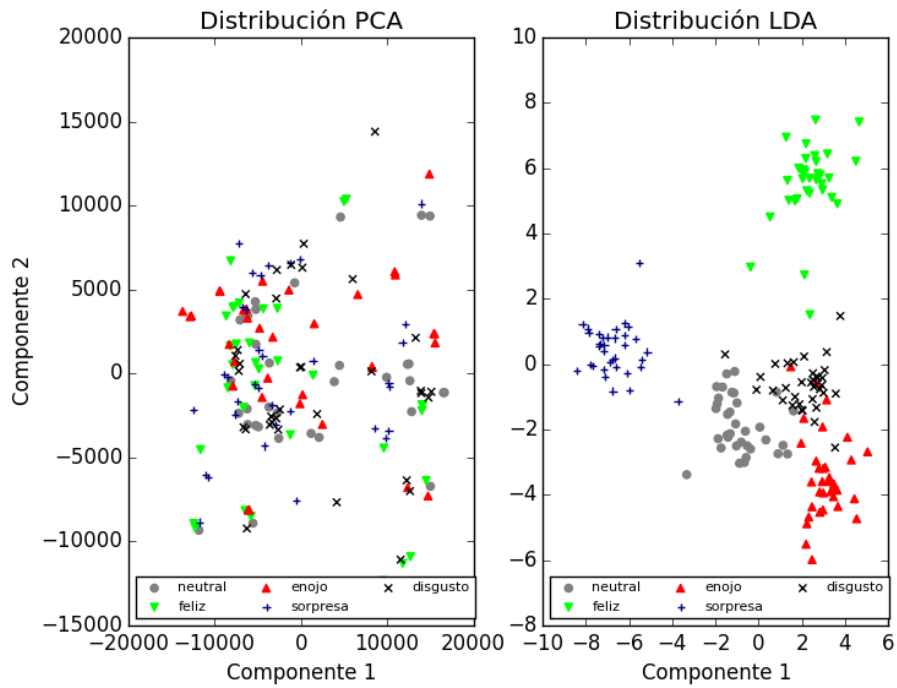


Figura 56: Distribuciones PCA vs. LDA de expresiones: neutral, feliz, enojo, sorpresa y disgusto utilizando 100 componentes para base de datos *UVG*

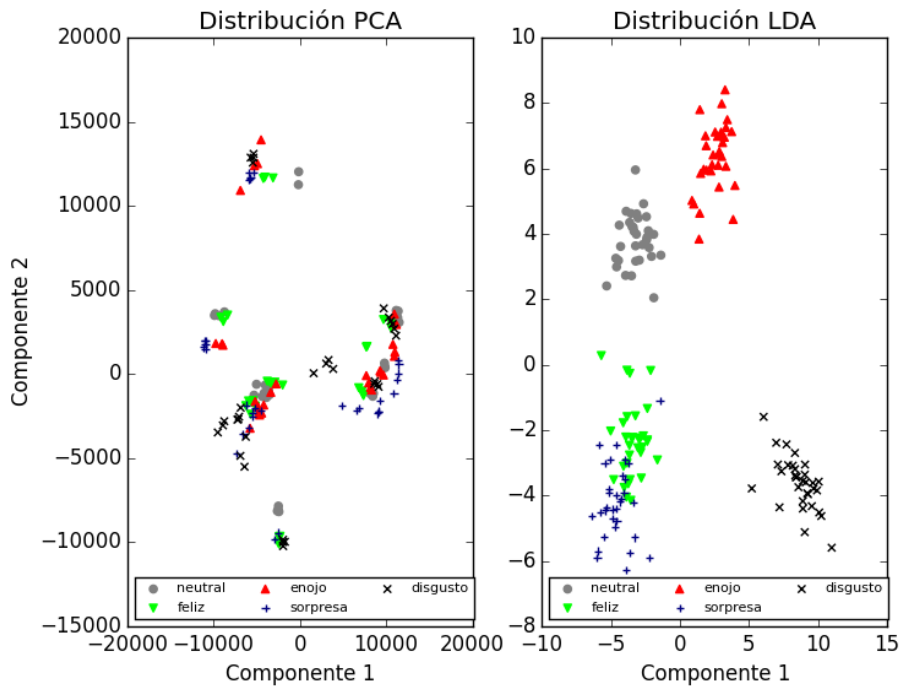


Figura 57: Distribuciones PCA vs. LDA de expresiones: neutral, feliz, enojo, sorpresa y disgusto utilizando 50 componentes para base de datos *Ambiente 2*

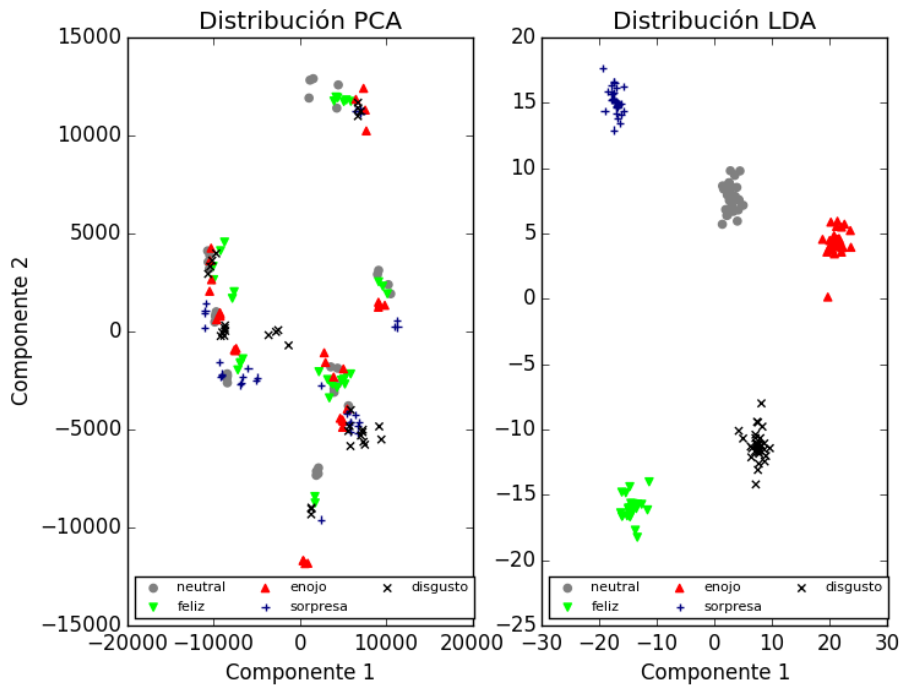


Figura 58: Distribuciones PCA vs. LDA de expresiones: neutral, feliz, enojo, sorpresa y disgusto utilizando 100 componentes para base de datos *Ambiente 2*

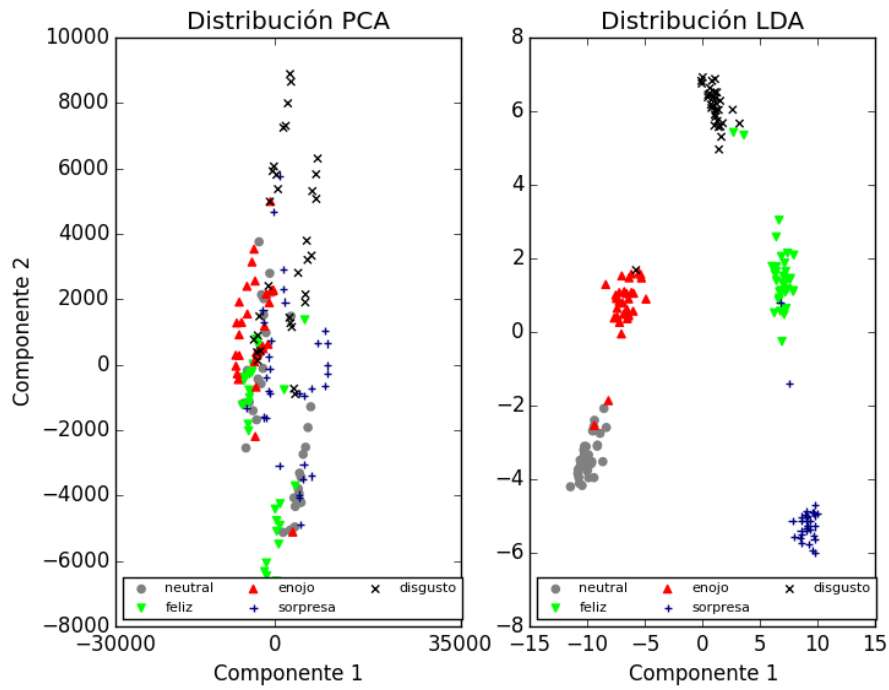


Figura 59: Distribuciones PCA vs. LDA de expresiones: neutral, feliz, enojo, sorpresa y disgusto utilizando 50 componentes para base de datos *Personalizada*

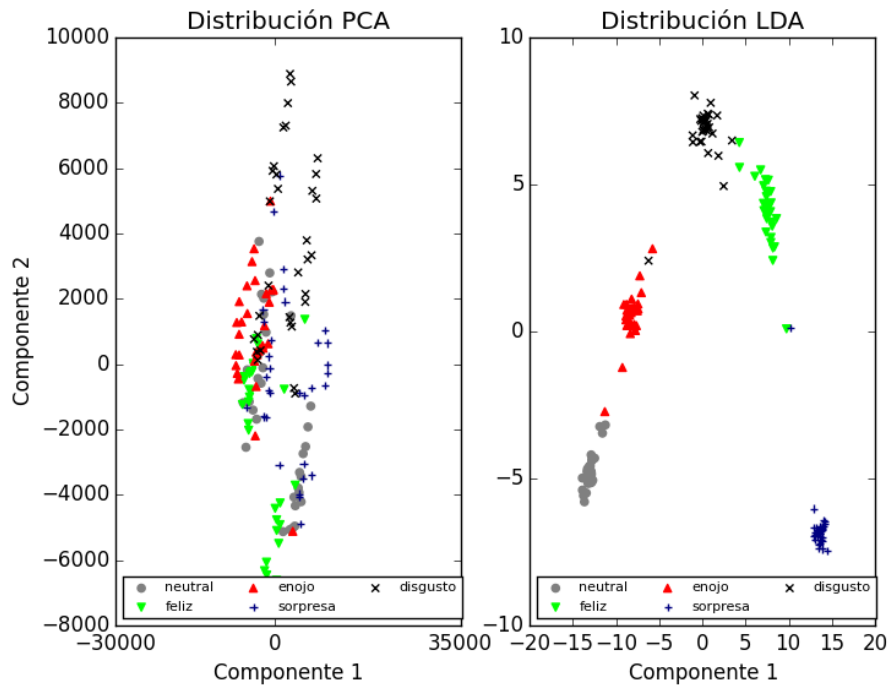


Figura 60: Distribuciones PCA vs. LDA de expresiones: neutral, feliz, enojo, sorpresa y disgusto utilizando 100 componentes para base de datos *Personalizada*

Las figuras 53 y 54 muestran la distribución para la base de datos *Cohn-Kanade*, en ambas figuras se puede ver la diferencia de separabilidad que presentan ambos algoritmos, siendo la distribución del algoritmo LDA la que presenta una mayor separación entre los datos. En el caso de las figuras 53 y 54 se puede observar que puede existir una confusión entre las imágenes entre enojo, disgusto y neutral.

En el caso de la base de datos UVG se tuvieron resultados similares en las distribuciones a las presentes en la base de datos *Cohn-Kanade*. Se tuvo un mayor separación de los datos para el algoritmo LDA. Sin embargo al observar la Figura 55 que representa la distribución utilizando 50 componentes principales en el paso PCA, se observó que estos no son los suficientes, ya que no se logra reconocer los centroides de las distribuciones. En el caso de la Figura 56 ya muestra una mejor distribución, al igual que para la distribución de 100 componentes para LDA se volvió a tener una menor separación entre las expresiones de enojo, disgusto y neutral.

La base de datos del *Ambiente 2* presenta una mejor distribución a comparación que las bases *UVG* y *Cohn-Kanade*, ya que al observar la Figura 57 muestra una gran separabilidad entre las clases. En el caso de la Figura 57 muestra una posible confusión entre la expresión de sorpresa y felicidad, sin embargo en la Figura 58 que ya utiliza 100 componentes, se observa que ya no se tienen ninguna confusión. Inclusive presenta una mayor separación entre las clases de enojo, disgusto y neutralidad en comparación con las bases de datos *UVG* y *Cohn-Kanade*.

En la base de datos *Personalizada* se logran identificar los centroides a partir del algoritmo PCA, esto se observa en las figuras 59 y 60. Esta base de datos es la que mejor separación presenta entre todas las bases y en parte es a que debido a que no se tiene una mayor compresión entre elementos de la misma clase, por lo cual la aplicación del LDA pareciera ser más efectiva. En el caso de la Figura 60 presenta una ligera confusión entre la expresión de felicidad y disgusto.

Los datos variarán dependiendo del entrenamiento, sin embargo las distribuciones en las figuras 53, 54, 55 y 56 sirven para poder generalizar las expresiones para una población más grande. Ya que la muestra de cada una de estas bases es más significativa al poseer un mayor número de sujetos en la prueba. Por lo cual se puede determinar que entre mayor se el número de componentes utilizados para la reducción de orden en el método PCA, es más probable que se tenga una mejor clasificación a la hora de la predicción de expresiones.

Extracción de marcas de cara

Gran parte del problema del reconocimiento de gestos a través del rostro es la selección de características para realizar la clasificación. Ya que en el caso del algoritmo *Fisher Face* se utilizan todas las intensidades de píxeles del rostro y a partir de estos se obtienen los componentes principales. Sin embargo la marcas de cara son una de las formas a partir de las cuales se pueden extraer las características principales sin tener un gran número de estos, que permitirán a partir de muestras más pequeñas obtener mejores clasificaciones. En este trabajo se utilizó el archivo *FacialLandmarks.dat* el cual obtiene las marcas de cara que se observan en la Figura 61.

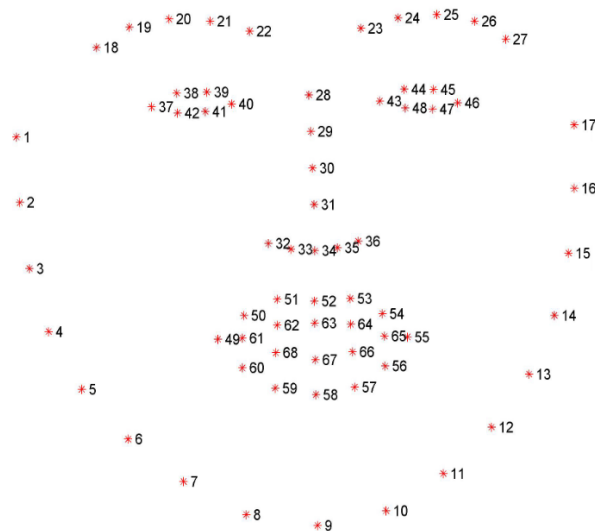


Figura 61: Localización de las marcas de cara con el archivo *FacialLandmarks.dat* [20]

Ya que uno de los objetivos era la implementación de un sistema que lograra la detección de expresiones, se lograron generar las marcas de cara a partir de las imágenes de cada una de las bases de datos. En las figuras 62, 63, 64, 65 y 66 se puede determinar las marca de cara para cada una de las expresiones para la base de datos *Personalizada*.

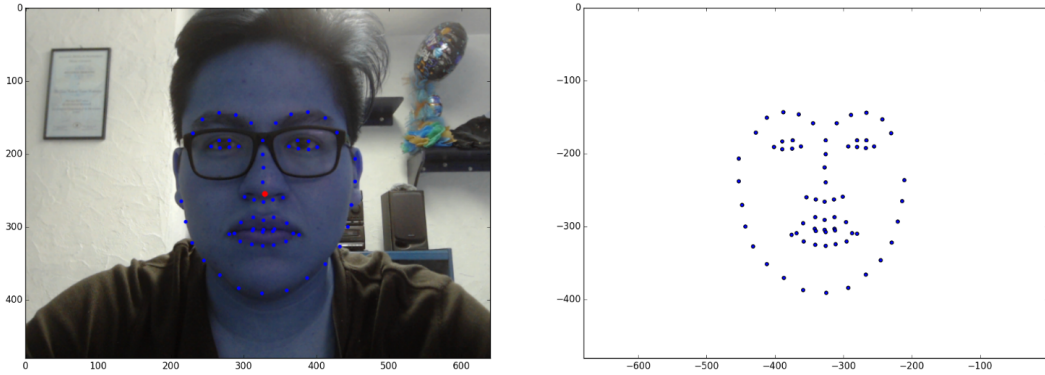


Figura 62: Marcas de cara expresión neutral

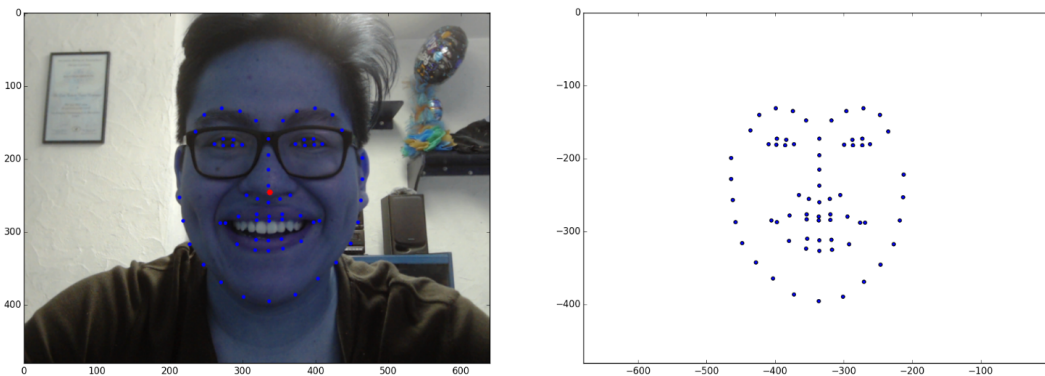


Figura 63: Marcas de cara expresión feliz

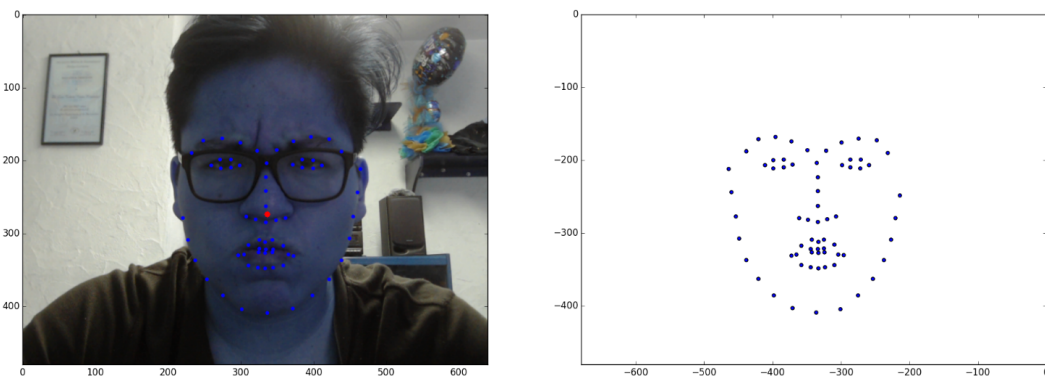


Figura 64: Marcas de cara expresión enojo

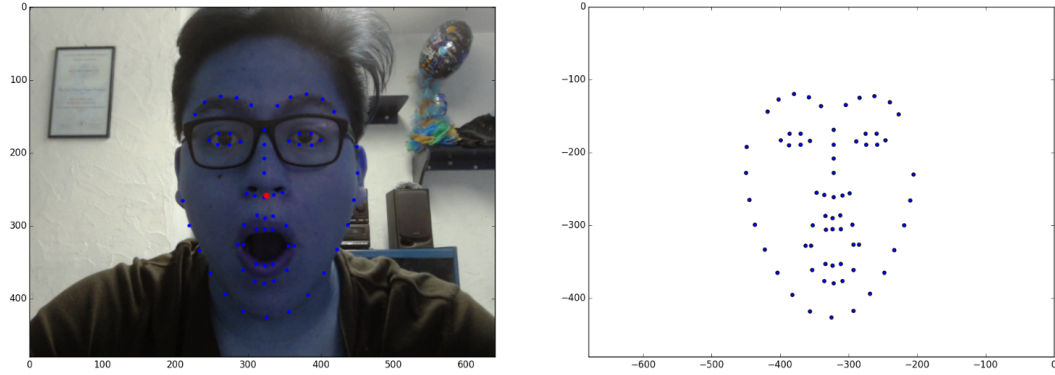


Figura 65: Marcas de cara expresión sorpresa

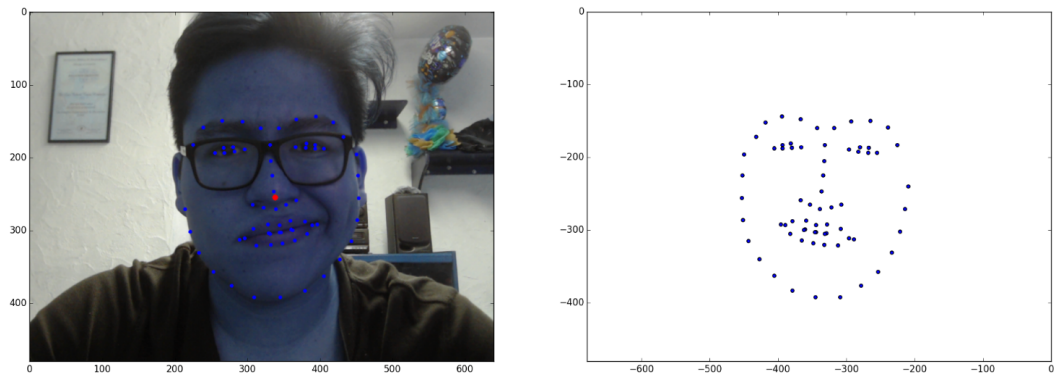


Figura 66: Marcas de cara expresión disgusto

Ya que el algoritmo basará las clasificaciones según las distancias medidas desde el punto rojo ubicado en el centro del rostro a cada uno de las marcas de cara localizadas, se deberá tener diferencias significativas para poder realizar la clasificación de las expresiones.

Las diferencias se irán generando a partir de la Figura 62 que presenta las marcas de la figura neutral. En la Figura 63 que muestra la expresión de felicidad, se pueden observar diferencias en el área de las cejas y principalmente en el área de la boca, en donde las marcas de cara se extienden a lo largo del eje X, para formar la sonrisa. Además existe una mayor separación en las marcas generadas en la parte inferior de la boca. En la Figura 64 que representa la expresión de enojo, se puede observar que todas las marcas se encuentran más concentradas a partir del punto medio. La contracción de las cejas y boca son de los rasgos que más se logran diferenciar para esta emoción. Para la expresión de sorpresa la cual se encuentra en la Figura 65 se puede observar el efecto contrario al enojo, ya que en esta emoción se alejaron la mayoría de marcas del centro del rostro. Se evidencia principalmente en el arqueado de las cejas y en la apertura de los labios y ojos. La expresión de disgusto está representado en la Figura 66 en este se puede observar que parte de las marcas de cara se encuentran orientadas hacia uno de los lados de la imagen. Las principales diferencias se encuentran en la desviación de boca y el cierre parcial del ojo hacia el cual se desvían las marcas de cara.

Las marcas de cara por lo cual presentan un apoyo tanto en la detección de rostro y servirán como parte fundamental del software a implementar para la reconocimiento de expresiones. Por lo cual se logra reconocer el tipo de equipo y software que se se necesita para el reconocimiento de expresiones faciales, el cual es uno de los objetivos específicos de este trabajo.

Reconocimiento de expresiones faciales

El reconocimiento de expresiones faciales basó su funcionamiento en la detección de características para poder generar entrenamientos los cuales puedan predecir las expresiones mostradas por los usuarios a través de sus gestos. Las secciones 11.1 y 11.2 mostrarán las matrices de confusión generadas a partir de una de las iteraciones generadas en la validación cruzada. Ambas secciones se subdividirán para mostrar los datos de cada uno de los entrenamientos generados.

Para esta sección se agregó un nuevo entrenamiento generado llamado *UVG 2 m*, el cual se realizó en el mismo ambiente utilizada para la base de datos *UVG*, pero se realizó para una distancia de toma de muestras a 2 m de la cámara. Este con el fin de validar el reconocimiento de expresiones para una distancia más alejada de la cámara con respecto a las demás bases de datos.

11.1. *Fisher Face*

11.1.1. Base de datos *Cohn-Kanade*

	Feliz	Enojo	Sorpresa
Feliz	21	0	0
Enojo	0	13	1
Sorpresa	0	0	25

Cuadro 12: Matriz de confusión base de datos *Cohn-Kanade* para tres expresiones con algoritmo *Fisher Face*

	Neutral	Feliz	Enojo	Sorpresa	Disgusto
Neutral	109	0	0	0	0
Feliz	1	24	0	0	0
Enojo	10	0	6	0	0
Sorpresa	3	0	0	27	0
Disgusto	2	0	0	0	19

Cuadro 13: Matriz de confusión base de datos *Cohn-Kanade* para cinco expresiones con algoritmo *Fisher Face*

11.1.2. Base de datos UVG

	Feliz	Enojo	Sorpresa
Feliz	97	12	10
Enojo	16	101	2
Sorpresa	19	18	63

Cuadro 14: Matriz de confusión base de datos *UVG* para tres expresiones con algoritmo *Fisher Face*

	Neutral	Feliz	Enojo	Sorpresa	Disgusto
Neutral	54	8	17	26	14
Feliz	1	99	2	13	4
Enojo	16	17	49	4	33
Sorpresa	22	16	14	61	5
Disgusto	10	13	27	24	25

Cuadro 15: Matriz de confusión base de datos *UVG* para cinco expresiones con algoritmo *Fisher Face*

11.1.3. Base de datos UVG 2 m

	Feliz	Enojo	Sorpresa
Feliz	66	71	3
Enojo	0	139	1
Sorpresa	17	38	74

Cuadro 16: Matriz de confusión base de datos *UVG 2 m* para tres expresiones con algoritmo *Fisher Face*

	Neutral	Feliz	Enojo	Sorpresa	Disgusto
Neutral	15	5	34	10	26
Feliz	0	57	5	8	19
Enojo	0	7	53	0	25
Sorpresa	6	1	13	41	20
Disgusto	0	5	25	14	29

Cuadro 17: Matriz de confusión base de datos *UVG 2 m* para cinco expresiones con algoritmo *Fisher Face*

11.1.4. Base de datos Ambiente 2

	Feliz	Enojo	Sorpresa
Feliz	37	0	17
Enojo	0	35	19
Sorpresa	0	0	48

Cuadro 18: Matriz de confusión base de datos *Ambiente 2* para tres expresiones con algoritmo *Fisher Face*

	Neutral	Feliz	Enojo	Sorpresa	Disgusto
Neutral	33	8	5	8	0
Feliz	19	18	7	1	9
Enojo	36	5	5	8	0
Sorpresa	27	6	0	11	10
Disgusto	18	3	0	23	4

Cuadro 19: Matriz de confusión base de datos *Ambiente 2* para cinco expresiones con algoritmo *Fisher Face*

11.1.5. Base de datos Personalizada

	Feliz	Enojo	Sorpresa
Feliz	21	0	0
Enojo	0	18	0
Sorpresa	0	0	20

Cuadro 20: Matriz de confusión base de datos *Personalizada* para tres expresiones con algoritmo *Fisher Face*

	Neutral	Feliz	Enojo	Sorpresa	Disgusto
Neutral	17	0	0	0	0
Feliz	0	21	0	0	0
Enojo	0	0	20	0	0
Sorpresa	0	0	0	14	0
Disgusto	0	0	0	0	21

Cuadro 21: Matriz de confusión base de datos personalizada para cinco expresiones con algoritmo *Fisher Face*

Entrenamiento	3 Expresiones	5 Expresiones
Cohn Kanade	98.33 %	92.03 %
UVG	77.21 %	50.17 %
UVG 2m	68.21 %	46.65 %
Ambiente 2	76.92 %	26.89 %
Personalizado	100 %	100 %

Cuadro 22: Porcentajes de reconocimiento de expresiones para algoritmo *Fisher Face*



Figura 67: Imagen con reconocimiento incorrecto de disgusto para base de datos ambiente 2

Para la validación de este algoritmo se utilizó un diseño iterativo, ya que se fueron generando los valores de exactitud y las matrices de confusión para ir validando los entrenamientos generados. Al principio se planteó el reconocer cinco expresiones en tiempo real. Sin embargo los resultados presentes en las matrices de confusión 13, 15, 19 junto con su exactitud en el Cuadro 22 se puede determinar que el reconocimiento de todas las expresiones al mismo tiempo no se obtienen los mejores resultados. Tal es el caso del dato de

Ambiente 2 del Cuadro 22 el cual es de 26.89%. Observando la matriz de confusión del *Ambiente 2*, Cuadro 19, se puede determinar que el algoritmo está confundiendo todas las expresiones pero muestra una predisposición para la expresión neutral, es decir se está dando un sobre-entrenamiento. Este sobre-entrenamiento se puede dar debido a varios factores pero principalmente porque no se tuvo la mejor extracción de características a la hora de la toma de datos. Otro de los datos que se puede obtener del Cuadro 19 es que las expresiones de disgusto y enojo fueron las que tuvieron mayor problema de identificación ya que en la diagonal de la matriz, los valores correctos de estas expresiones fueron cinco y cuatro respectivamente. La Figura 67 presenta una de las imágenes de la base de datos del *Ambiente 2*, la cual fue catalogada como enojo, sin embargo el sujeto mostró los gestos expresados en la Figura 67 cuando se le pidió expresar disgusto. Inclusive para el ser humano la Figura 67 pudo haber sido confundida con enojo, al no ser el sujeto de prueba muy expresivo.

Las bases de datos que mostraron mejores resultados para el reconocimiento de las cinco expresiones fueron *Cohn-Kanade* y *Personalizada*. En el caso de la base de datos *Cohn-Kanade* se dio debido al gran control de los factores que se tiene en esta base de datos, tales como: iluminación, postura, distancia y validación de expresión. En el caso de la base *Personalizada* observando la Figura 21 se determinó que se tuvo un 100% de exactitud. Al ser una base de datos con un solo sujeto y ser el algoritmo enfocado al reconocimiento de patrones, se tendrá una mejor detección al no haber tanta variación. Sin embargo esto tendrá problemas a la hora de generalizar este entrenamiento para sujetos diferentes al de la prueba.

Basados en los resultados obtenidos de la exactitud presentados en el Cuadro 22, matrices de confusión 13, 15, 17, 19 y las distribuciones del algoritmos generadas en la sección 9.2.3 se logró determinar que la mayoría de confusión se da entre las expresiones de neutralidad, enojo y disgusto. La expresión de disgusto siendo la expresión que mayor dificultad presenta a la hora de identificarlo. Se decidió incluir la expresión de enojo ya que esta es la que genera una menor confusión entre las expresiones de sorpresa y felicidad. Es debido a estos resultados que se implementó entrenamientos basados en tres expresiones presentes en las matrices de confusión 12, 14, 16, 18 y 20.

La modificación de limitar a reconocer tres emociones le dio una mejora en la exactitud de los entrenamientos. Esto se puede observar en el Cuadro 22 en donde con excepción de la base de datos *Personalizada*, todas aumentaron su porcentaje. Todos los porcentajes de reconocimiento para tres expresiones se encuentran por encima del 68%. A pesar de esta reducción siguen habiendo confusiones, tal y como se observa en la matriz 14 en donde las expresiones de felicidad y sorpresa son las que más se confunden con esta reducción a tres expresiones. Para el caso de tres expresiones la que reconoce con mayor facilidad es la de enojo especialmente para las bases de datos *UVG* y *UVG 2 m* en la cual las matrices 14 y 16 muestran valores de 139 y 101 resultados positivos en torno al reconocimiento de enojo. Esto también se puede ver gráficamente en la Figura 56 en donde las expresión de enojo, es la que mayor diferencia presenta entre las las expresiones de felicidad y sorpresa.

El porcentaje de exactitud más bajo presente en el Cuadro 22 para tres expresiones es la del entrenamiento *UVG 2 m* con un valor de 68.21%. Esto puede ser debido a que esta fue la base de datos que se obtuvo a una mayor distancia, por lo cual se puede tener mayor pérdida de resolución en cuanto a las demás bases de datos, ya que al estar más alejada pueden omitirse ciertos rasgos. Es por eso que se esperan mejores resultados en el caso del

entrenamiento *UVG*, ya que este se realizó bajo las mismas condiciones de iluminación y lugar, solo se modificó la distancia de la captura de datos.

11.2. SVM

Los entrenamientos generados por las implementaciones SVM, presentan otra alternativa a los entrenamientos generados por el método *Fisher Face*. En el caso del métodos SVM pueden tener diferentes implementaciones dependiendo de la forma del kernel. Para esta sección se realizó un análisis basado en dos tipos de kernel: lineal y polinomial de grado tres. Además este método tiene un parámetro c , que permite controlar la agresividad del método a la hora de generar la clasificación. En el caso del kernel lineal, se limitó a dejar el valor por defecto que corresponde a uno. En cambio para el kernel polinomial se modificó y se presentan los resultados para el valor en el cual se tuvo mejores resultados.

En esta sección se presentan los resultados obtenidos para el reconocimiento de expresiones en base a la extracción de marcas de cara y generando un entrenamiento en base al algoritmo SVM. Debido a la mejora de rendimiento mostrado en el algoritmo *Fisher Face* se tomó en cuenta mostrar resultados para 3 y 5 expresiones.

11.2.1. Base de datos *Cohn-Kanade*

	Feliz	Enojo	Sorpresa
Feliz	21	0	0
Enojo	2	12	0
Sorpresa	1	0	24

Cuadro 23: Matriz de confusión base de datos *Cohn-Kanade* para tres expresiones, SVM lineal

	Feliz	Enojo	Sorpresa
Feliz	21	0	0
Enojo	1	13	0
Sorpresa	1	0	24

Cuadro 24: Matriz de confusión base de datos *Cohn-Kanade* para tres expresiones, SVM polinomial grado tres, $C=2^{*-5}$

	Neutral	Feliz	Enojo	Sorpresa	Disgusto
Neutral	92	1	0	0	0
Feliz	2	17	0	0	2
Enojo	4	0	9	0	1
Sorpresa	1	0	0	24	0
Disgusto	4	0	2	0	12

Cuadro 25: Matriz de confusión base de datos *Cohn-Kanade* para cinco expresiones, SVM lineal

	Neutral	Feliz	Enojo	Sorpresa	Disgusto
Neutral	88	0	3	2	0
Feliz	2	19	0	0	0
Enojo	4	0	10	0	0
Sorpresa	0	0	0	25	0
Disgusto	4	3	3	0	8

Cuadro 26: Matriz de confusión base de datos *Cohn-Kanade* para cinco expresiones, SVM polinomial grado tres, $C=2^{*-5}$

11.2.2. Base de Datos UVG

	Feliz	Enojo	Sorpresa
Feliz	90	7	22
Enojo	26	74	19
Sorpresa	14	6	82

Cuadro 27: Matriz de confusión base de datos *UVG* para tres expresiones, SVM lineal

	Feliz	Enojo	Sorpresa
Feliz	77	22	20
Enojo	7	105	0
Sorpresa	21	20	60

Cuadro 28: Matriz de confusión base de datos *UVG* para tres expresiones, SVM polinomial grado tres, $C=2^{*-5}$

	Neutral	Feliz	Enojo	Sorpresa	Disgusto
Neutral	67	15	6	21	10
Feliz	1	99	1	10	8
Enojo	25	3	51	7	32
Sorpresa	9	23	2	71	13
Disgusto	6	32	20	1	43

Cuadro 29: Matriz de confusión base de datos *UVG* para cinco expresiones, SVM lineal

	Neutral	Feliz	Enojo	Sorpresa	Disgusto
Neutral	75	6	9	18	11
Feliz	9	74	1	18	17
Enojo	15	16	58	0	29
Sorpresa	20	7	0	86	5
Disgusto	26	15	6	10	45

Cuadro 30: Matriz de confusión base de datos *UVG* para cinco expresiones, SVM polinomial grado tres, $C=2^{*-5}$

11.2.3. Base de datos UVG 2 m

	Feliz	Enojo	Sorpresa
Feliz	92	19	29
Enojo	22	100	18
Sorpresa	16	7	107

Cuadro 31: Matriz de confusión base de datos *UVG 2 m* para tres expresiones, SVM lineal

	Feliz	Enojo	Sorpresa
Feliz	109	18	13
Enojo	20	109	11
Sorpresa	16	6	108

Cuadro 32: Matriz de confusión base de datos *UVG 2 m* para tres expresiones, SVM polinomial grado tres, $C=2^{*-5}$

	Neutral	Feliz	Enojo	Sorpresa	Disgusto
Neutral	92	2	13	29	4
Feliz	20	67	4	44	5
Enojo	51	8	54	3	24
Sorpresa	4	12	0	119	5
Disgusto	27	38	7	13	45

Cuadro 33: Matriz de confusión base de datos *UVG 2 m* para cinco expresiones, SVM lineal

	Neutral	Feliz	Enojo	Sorpresa	Disgusto
Neutral	86	9	15	7	23
Feliz	21	87	0	11	21
Enojo	25	6	83	1	25
Sorpresa	28	25	1	85	1
Disgusto	39	28	7	0	56

Cuadro 34: Matriz de confusión base de datos *UVG 2 m* para cinco expresiones, SVM polinomial grado tres, $C=2^{*-5}$

11.2.4. Base de datos Ambiente 2

	Feliz	Enojo	Sorpresa
Feliz	27	5	22
Enojo	0	54	0
Sorpresa	4	0	44

Cuadro 35: Matriz de confusión base de datos *Ambiente 2* para tres expresiones, SVM lineal

	Feliz	Enojo	Sorpresa
Feliz	28	0	26
Enojo	14	37	3
Sorpresa	1	0	47

Cuadro 36: Matriz de confusión base de datos *Ambiente 2* para tres expresiones, SVM polinomial grado tres, $C=2^{*-5}$

	Neutral	Feliz	Enojo	Sorpresa	Disgusto
Neutral	13	7	17	9	8
Feliz	9	27	0	5	13
Enojo	6	0	27	1	20
Sorpresa	10	5	7	30	2
Disgusto	4	3	11	8	22

Cuadro 37: Matriz de confusión base de datos *Ambiente 2* para cinco expresiones, SVM lineal

	Neutral	Feliz	Enojo	Sorpresa	Disgusto
Neutral	26	1	6	20	1
Feliz	5	38	0	11	0
Enojo	0	5	42	0	0
Sorpresa	0	5	0	49	0
Disgusto	0	24	8	7	9

Cuadro 38: Matriz de confusión base de datos *Ambiente 2* para cinco expresiones, SVM polinomial grado tres, $C=2^{*-5}$

11.2.5. Base de datos Personalizada

	Feliz	Enojo	Sorpresa
Feliz	30	0	0
Enojo	0	30	0
Sorpresa	0	0	31

Cuadro 39: Matriz de confusión base de datos *Personalizada* para tres expresiones, SVM lineal

	Feliz	Enojo	Sorpresa
Feliz	30	0	0
Enojo	0	30	0
Sorpresa	0	0	31

Cuadro 40: Matriz de confusión base de datos *Personalizada* para tres expresiones, SVM polinomial grado tres, $C=2^{*-5}$

	Neutral	Feliz	Enojo	Sorpresa	Disgusto
Neutral	30	0	0	0	0
Feliz	0	30	0	0	0
Enojo	0	0	30	0	0
Sorpresa	0	0	0	31	0
Disgusto	0	0	0	0	30

Cuadro 41: Matriz de confusión base de datos *Personalizada* para cinco expresiones, SVM lineal

	Neutral	Feliz	Enojo	Sorpresa	Disgusto
Neutral	30	0	0	0	0
Feliz	0	30	0	0	0
Enojo	1	0	29	0	0
Sorpresa	1	0	0	30	0
Disgusto	0	0	0	0	30

Cuadro 42: Matriz de confusión base de datos *Personalizada* para cinco expresiones, SVM polinomial grado tres, $C=2^{*-5}$

Base de Datos	Kernel	3 Expresiones	5 Expresiones
<i>Cohn-Kanade</i>	Lineal	95 %	90.06 %
<i>Cohn-Kanade</i>	Polinomial	96.67 %	87.77 %
UVG	Lineal	72.35 %	57.46 %
UVG	Polinomial	72.89 %	58.68 %
UVG 2 m	Lineal	72.92 %	54.64 %
UVG 2 m	Polinomial	79.51 %	57.53 %
Ambiente 2	Lineal	80.12 %	45.07 %
Ambiente 2	Polinomial	71.79 %	63.81 %
Personalizada	Lineal	100 %	100 %
Personalizada	Polinomial	100 %	98.67 %

Cuadro 43: Porcentajes de reconocimiento expresiones para algoritmo SVM

Las matrices 23, 27, 31, 35 y 39 representan los resultados para tres expresiones basados en kernel lineal. Por medio del Cuadro 43 se puede determinar que el que tuvo peor rendimiento fue el entrenamiento *UVG* con un valor de 72.35 %. Observando los datos de la matriz 27 se puede determinar que la expresión que menos logró reconocer fue la expresión de enojo. Sin embargo el mayor número de falsos negativo se dio en la expresión de felicidad, la cual la confundió con la expresión de sorpresa.

Así como se obtuvo para los entrenamientos para reconocer tres emociones con kernel lineal, las matrices 24, 28, 32, 36 y 40 se generaron a partir de un kernel polinomial de grado tres. Tal y como se observa en el Cuadro 43 el entrenamiento que genera mayor dificultad para el reconocimiento de expresiones fue para la base de datos *Ambiente 2* con 71.79 %. Al observar la matriz 36 se obtuvo que la emoción que mejor reconoció fue el enojo y volvió a tener una mayor confusión para las expresiones de felicidad y sorpresa. Esta confusión se



Figura 68: Imagen con reconocimiento incorrecto de sorpresa para base de datos ambiente 2

puede observar en la Figura 68 que presenta una imagen la cual fue catalogada inicialmente como sorpresa, pero que el algoritmo la clasificó como feliz.

Realizando una comparación entre los valores presentes en el Cuadro 43 entre los entrenamientos de tres expresiones entre los diferentes kernels, no existe diferencia significativa entre los entrenamientos. Ya que para los entrenamientos de *Cohn-Kanade*, *UVG* y *UVG 2m* se obtuvo mejores resultados, solamente se tuvo disminución para la base de Datos Ambiente 2. Esto es en parte debido a que la distribución de datos va a variar dependiendo del entrenamiento, es por eso que se recomendaría un kernel polinomial para los entrenamientos, exceptuando el Ambiente 2 si se quieren reconocer tres expresiones.

Las matrices 25, 29, 33, 37 y 41 representan los resultados para cinco expresiones basados en kernel lineal. El entrenamiento que presentó el menor porcentaje de exactitud fue la base de datos *Ambiente 2*. El cual tuvo un valor de 45.07%. Para esta la mayor confusión se dio entre las emociones de neutral y enojo, tal y como se observa en la matriz 37.

Los resultados generados a partir de los entrenamientos con reconocimiento de cinco expresiones basados en kernel polinomial se observan en las matrices 26, 30, 34, 38 y 42. En el caso de los entrenamientos se observa por medio del Cuadro 43 que la que obtuvo el desempeño mas bajo fue el entrenamiento *UVG 2 m* con un valor de 57.53%. Observando la matriz 34 del entrenamiento *UVG 2 m*. Se puede determinar que la mayor confusión se da entre las expresiones de neutralidad y disgusto, siendo la expresión de disgusto la que mayor dificultad tuvo de reconocer.

Según los cuadros 22 y 43 se puede determinar que si existe una mejora de desempeño cuando se intentan clasificar entre tres y cinco expresiones. Siendo para los algoritmos *Fisher Face* y *SVM* la clasificación de tres expresiones un problema que puede abordar de mejor manera.

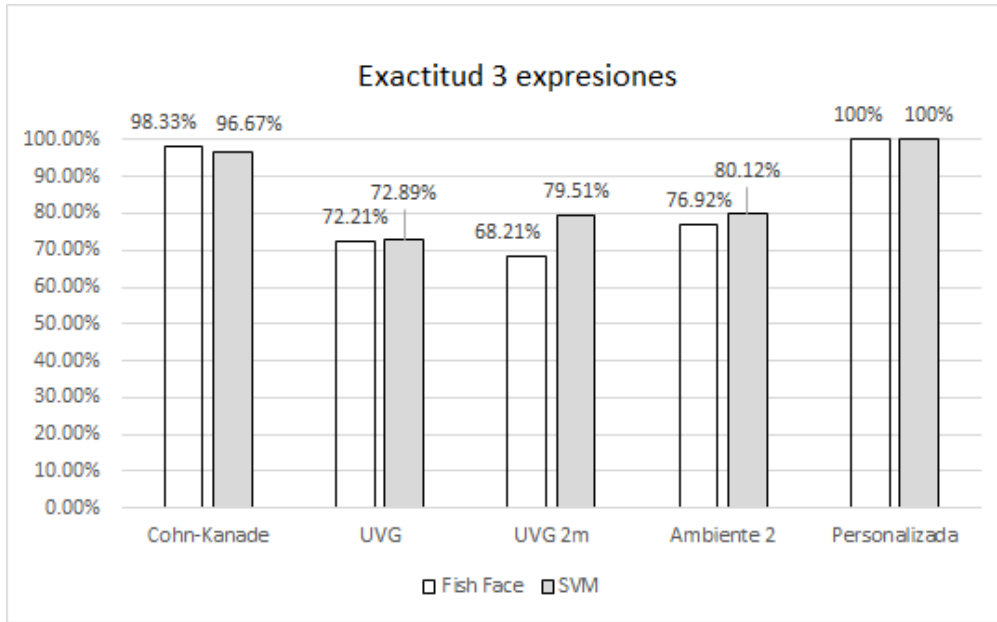


Figura 69: Comparación porcentajes de detección algoritmos SVM y *Fisher Face* para tres expresiones

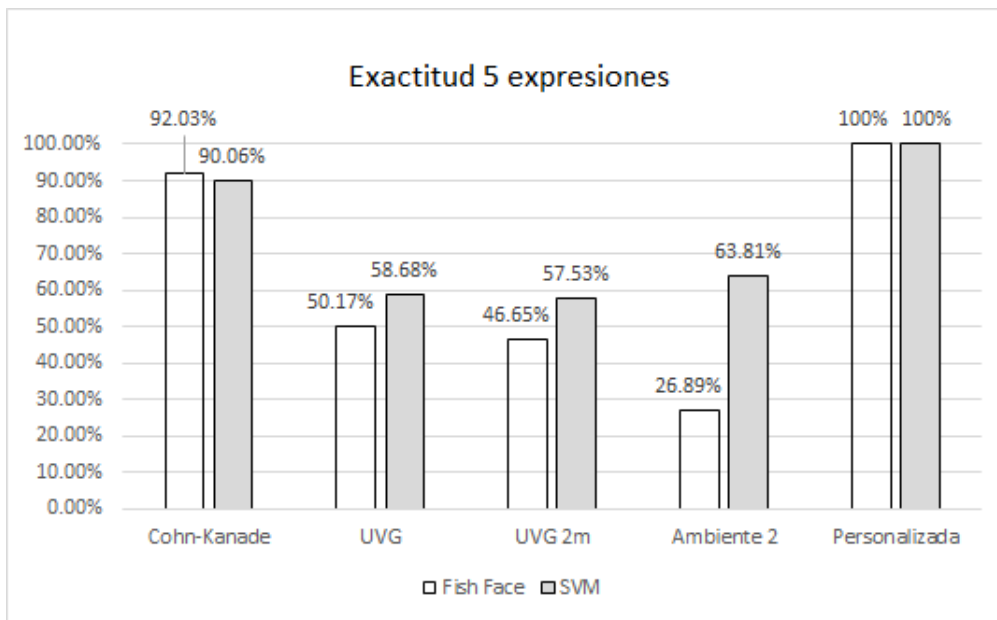


Figura 70: Comparación porcentajes de detección algoritmos SVM y *Fisher Face* para cinco expresiones

	Feliz	Enojo	Sorpresa
<i>Fisher Face</i>	73 %	79 %	77 %
SVM	79 %	87 %	85 %

Cuadro 44: Porcentajes de reconocimiento por expresión, para *feliz*, *enojo* y *sorpresa*

	Neutral	Feliz	Enojo	Sorpresa	Disgusto
<i>Fisher Face</i>	72 %	75.31 %	60 %	63 %	52.75 %
SVM	74.52 %	78.14 %	74.49 %	82.22 %	59.28 %

Cuadro 45: Porcentajes de reconocimiento por expresión, para *neutral, feliz, enojo, sorpresa y disgusto*

Para realizar una comparación entre los algoritmos se generaron gráficas de barra en las figuras 69 y 70. En el caso de la comparación de tres expresiones presentes en la Figura 69 el algoritmo SVM presenta un mejor rendimiento en comparación con el algoritmo *Fisher Face*, la mayor diferencia se presenta en la base de datos *UVG 2 m*. En donde pasó de tener un 26.89 % con el algoritmo *Fisher Face* a un 65.81 % con el algoritmo SVM.

Para la comparación entre ambos algoritmos con base en cinco expresiones los resultados se encuentran presentes en la Figura 70. En esta Figura se puede observar que el algoritmo SVM presenta un mayor eficiencia, especialmente para las bases de datos *UVG 2 m* y *Ambiente 2*.

En el único caso que el algoritmo *Fisher Face* presenta un mejor reconocimiento es para la base de datos *Cohn-Kanade*. Este resultado se da tanto para el entrenamiento de tres y cinco emociones. Esto es debido al gran control que se tuvo en la toma de datos. Otro de los datos que se destaca en las figuras 69 y 70 es que para la base de datos personalizada se logran obtener 100 % de reconocimiento para ambos algoritmos.

Aparte de la comparación en base a las bases de datos, se realizó una comparación del reconocimiento de las expresiones faciales. Estas comparaciones se encuentran presentes en los cuadros 44 y 45. En el caso de la tabla 44 que presenta los resultados para tres expresiones, el valor más bajo con el algoritmo *Fisher Face* es de 73 % para la expresión de felicidad. Mientras que para el algoritmo SVM el valor más bajo fue para la expresión de felicidad con un valor de 79 %. En ambos algoritmos la expresión que mayor dificultad presenta de identificar la de felicidad, mientras que la expresión que mayor precisión presenta es la de enojo. Esto es debido a que los puntos de marcas de cara para la expresión de enojo, presentan una mayor separabilidad de las expresiones de felicidad y sorpresa respectivamente.

En el caso del Cuadro 45 presenta una comparación con los porcentajes de sensibilidad para cada una de las expresiones en los entrenamientos de cinco expresiones. Tanto para el algoritmo *Fisher Face* como para el algoritmo SVM, la expresión de enojo fue la que menor porcentaje de sensibilidad presentó. Siendo de 52.75 % para el algoritmo *Fisher Face* y 59.28 % para SVM. El Cuadro 45 presenta la justificación de la reducción a tres expresiones, especialmente para el algoritmo SVM en donde las expresiones de neutralidad, feliz, enojo y sorpresa presentaron mejores resultados. Sin embargo ya que se buscaba el reconocimiento de expresiones se decidió limitar a tan solo tres expresiones dejando fuera la expresión de neutralidad.

Esta sección se logra englobar discusión entorno a cada uno de los objetivos específicos. En el caso del objetivo de determinar el equipo necesario para el seguimiento de rostro y reconocimiento de expresiones faciales tales como felicidad, enojo, sorpresa y disgusto. Ya que por medio de la cámara *Logitech C920* se logró obtener los marcos necesarios tanto para la detección de rostro y reconocimiento de expresiones faciales. Junto con un computador

presente y un programa generado por medio del lenguaje *Python*, se logró obtener porcentajes de reconocimiento de expresiones en su mayoría superiores al 50 %, tal y cómo lo muestran los cuadros 22 y 43.

Como se mencionó en el capítulo de marcas de cara, esto fue parte fundamental para el reconocimiento de expresiones, junto con el algoritmo SVM se logró obtener reconocimientos de hasta el 100 % en el caso de los entrenamientos personalizados. En este apartado se realizó una comparación entre ambos algoritmos para poder generar una solución que abordara todos los entrenamientos propuestos y a partir de ahí elegir el método deseado. A partir de las figuras 69 y 70 se puede observar la mejora de rendimiento que presenta el algoritmo SVM por el cual se utilizará este para entrenar y generar el algoritmo de reconocimiento de expresiones.

En la búsqueda de los objetivos que involucraban las pruebas de 30 sujetos en dos lugares diferentes y la implementación de un software que lograra el reconocimiento de expresiones faciales. Se obtuvo la captura de datos y las pruebas para cada uno de estos entrenamientos, con cada uno de los algoritmos que se implementaron. Esto con el fin de determinar el algoritmo que mostrará el mejor rendimiento para poder implementarlo en el ambiente que fuese necesario. Siempre con la búsqueda de una generalización de este.

Según todos los resultados descritos se generó un manual de aplicación. En este se encuentran las indicaciones detalladas en cuanto a la aplicación, utilización y modificación del código en base a lo que desea realizar el usuario. El manual se encuentra en la sección de anexos. Este forma parte del objetivo de generar un manual de seguimiento de pasos para la implementación del programa para el reconocimiento de rostro y expresiones faciales. Esto con el fin de poder darle un seguimiento al proyecto y facilitar futuras implementaciones de este.

- Las características de captación de video de tamaño 640×480 a cinco cuadros por segundo, imágenes de tamaño 1920×1080 y un lente con auto-enfoque, fueron características necesarias para la detección de rostro y reconocimiento de expresiones faciales.
- El algoritmo marcas de cara presenta resultados de hasta 92.5 % de detección de rostro cuando el sujeto de prueba tiene un ángulo de visión de 45° .
- La aplicación de ecualización de histograma presenta una mejora en la detección de rostros del algoritmo *Haar cascade* con un valor de 44.5 % de diferencia en las pruebas de *Ambiente Abierto*.
- El algoritmo *Haar cascade* logró la detección de rostro a una distancia máxima de 5.31 m, mientras que el algoritmo marcas de cara tuvo un alcance máximo de 5.11 m.
- La base de datos personalizada generó los mejores resultados en cuanto a la detección y reconocimiento de expresiones llegando a obtener resultados de 100 % de detección y reconocimiento.
- Entre las bases de datos con más de un sujeto de prueba, la base *Cohn-Kanade* es la que mejores resultados presenta en cuanto al reconocimiento de expresiones faciales con porcentajes por encima del 90 % para las pruebas del Algoritmo *Fisher Face* y *SVM*.
- El algoritmo SVM presenta mejor desempeño que el algoritmo *Fish Face* en el reconocimiento de expresiones en las bases de Datos *UVG*, *UVG 2m* y *Ambiente 2*, siendo valores superiores a 57 %.
- La expresión de disgusto fue la que presentó el porcentaje de reconocimiento más bajo con un valor de 52.75 %.
- Para el caso del algoritmo *Fisher Face* y SVM se presentan porcentajes de reconocimiento superiores cuando se limita la detección de expresiones a *feliz*, *enojo* y *sorpresa*.

Como se mencionó en gran parte del trabajo, el mayor desafío en los algoritmos de clasificación es la extracción de características, por lo cual se recomienda para futuros trabajos se siga desarrollando algoritmos que busquen y extraigan las características más importantes para que se mejore el porcentaje de detección de los algoritmos. Parte de esto será seguir tratando de optimizar los algoritmos que ya se generaron e intentar generar resultados basados en algoritmos de redes neuronales convolucionales, las cuales pueden llegar a generar mejores resultados en el reconocimiento de expresiones faciales.

- Si en caso se desea continuar con la implementación de este trabajo, se recomienda continuar a partir del reconocimiento de tres expresiones faciales con el algoritmo SVM para obtener mejores resultados de reconocimientos.
- Se recomienda que si se desea generar una base de datos, se utilice un entorno más controlado en el cual se tenga un control de la postura por medio de la silla para que no existan rotaciones con respecto del ángulo de captura de la cámara.
- Se recomienda implementar un algoritmo que logre detectar un menor número de marcas de cara, que permita tiempo de procesamientos más cortos.
- Se recomienda generar un mayor número de entrenamientos en ambiente diferentes, que permita al sistema tener una mayor robustez de implementación para cada uno de estos ambientes.
- Se recomienda implementar otros algoritmos de *machine learning* a partir del reconocimiento de marcas de cara, como lo sería el caso de las redes neuronales.
- Se recomienda implementar un sistema que permite el reconocimiento de otras expresiones faciales tales como: tristeza o miedo.
- Se recomienda al departamento de ingeniería electrónica de la Universidad del Valle de Guatemala a buscar asociaciones y convenios que permitan que se siga invirtiendo en el

proyecto animatrónico para se puedan realizar proyectos más complejos que requieran costos más elevados.

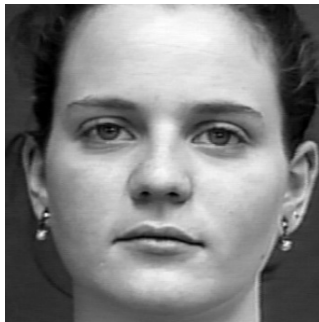
- Se recomienda a la Universidad del Valle de Guatemala a continuar con la investigación del módulo de seguimiento de rostro y reconocimientos de expresiones para ampliar la robustez del algoritmo y permitir el desarrollo de nuevas tecnologías.

-
- [1] J. Chatrath, P. Gupta, P. Ahuja, A. Goel y S. MArora, “REAL TIME HUMAN FACE DETECTION AND TRACKING”, dirección: http://vstechnologies.net/wp-content/uploads/2018/01/DIP_8.pdf.
 - [2] I. Matthews y S. Baker, “Active Appearance Models Revisited”, *International Journal of Computer Vision*, vol. 60, n.º 2, págs. 135-164, nov. de 2004, ISSN: 0920-5691. DOI: 10.1023/B:VISI.0000029666.37597.d3. dirección: <http://link.springer.com/10.1023/B:VISI.0000029666.37597.d3>.
 - [3] P. N. Belhumeur, D. W. Jacobs, D. J. Kriegman y N. Kumar, “Localizing Parts of Faces Using a Consensus of Exemplars”, inf. téc. dirección: https://neerajkumar.org/projects/face-parts/base/papers/nk_cvpr2011_faceparts.pdf.
 - [4] H. Yang, R. Zhang y P. Robinson, “Human and sheep facial landmarks localisation by triplet interpolated features”, en *2016 IEEE Winter Conference on Applications of Computer Vision, WACV 2016*, 2016, ISBN: 9781509006410. DOI: 10.1109/WACV.2016.7477733.
 - [5] G. Ottado, “Reconocimiento de caras: Eigenfaces y Fisherfaces”,
 - [6] R. Gross, I. Matthews, J. Cohn, T. Kanade y S. Baker, “Multi-PIE”, *Image and Vision Computing*, 2010, ISSN: 02628856. DOI: 10.1016/j.imavis.2009.08.002.
 - [7] P. J. Phillips, “Support Vector Machines Applied to Face Recognition”, inf. téc.
 - [8] P. van Gent, “Emotion Recognition Using Facial Landmarks, Python, Dlib and OpenCV”, inf. téc., 2016. dirección: <http://www.paulvangent.com/2016/08/05/emotion-recognition-using-facial-landmarks/>.
 - [9] L. Aguirre y R. Jo, “Consolidado Animatronics Trabajo de graduación en modadlidad de Megaproyecto para optar por el grado de Ingeniería Electrónica Universidad del Valle de Guatemala”, inf. téc.
 - [10] A. Teichman y S. Thrun, “Practical object recognition in autonomous driving and beyond”, inf. téc. dirección: <https://cs.stanford.edu/people/teichman/papers/arso2011.pdf>.

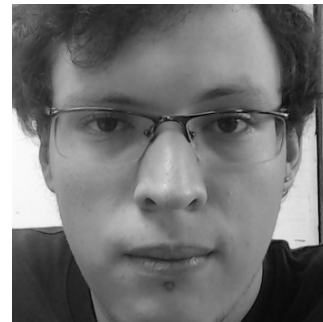
- [11] J. Ware e I. Ciuca, “A neural network based integrated image processing environment for object recognition in medical applications”, en *Proceedings of Computer Based Medical Systems*, IEEE Comput. Soc. Press, págs. 149-154, ISBN: 0-8186-7928-X. DOI: 10.1109/CBMS.1997.596425. dirección: <http://ieeexplore.ieee.org/document/596425/>.
- [12] P. Ekman, S. L. David, R. Matsumoto, H. Oster, E. L. Rosenberg y K. R. Scherer, “Facial Expression and Emotion”, inf. téc.
- [13] “DEFINICIONES DE INTELIGENCIA EMOCIONAL”, inf. téc., 2018. dirección: https://www.aepap.org/sites/default/files/457-469_inteligencia_emocional.pdf.
- [14] S. Iglesias-Hoyos, A. del Castillo Arreola y J. I. Muñoz-Delgado, “Reconocimiento facial de expresión emocional: diferencias por licenciaturas”, *Acta de Investigación Psicológica*, 2016, ISSN: 20074719. DOI: 10.1016/j.aiprr.2016.07.001.
- [15] C. F. Keating, A. Mazur, M. H. Segall, P. G. Cysneiros William T Divale, J. E. Kilbride Suntaree Komin, P. Leahy Blake Thurman y W. Smith Colleges Rolf Wirsing, “Culture and the Perception of Social Dominance From Facial Expression”, inf. téc. 4, 1981, págs. 615-626.
- [16] E. Parra, B. Tutores, Á. Rodríguez Vázquez y J. F. Berni, “Aceleración del algoritmo de Viola-Jones mediante rejillas de procesamiento masivamente paralelo en el plano focal”, dirección: <http://bibing.us.es/proyectos/abreproy/90325/fichero/TrabajoFinGrado.pdf>.
- [17] K. G. Derpanis, “Integral image-based representations”, dirección: http://www.cs.yorku.ca/~kosta/CompVis_Notes/integral_representations.pdf.
- [18] R. E. Schapire, “Explaining AdaBoost”, dirección: <https://www.cs.princeton.edu/~schapire/papers/explaining-adaboost.pdf>.
- [19] "48740_7p_8291_000, “Boosting Foundations and Algorithms”, vol. 10, n.º 17, 2012. dirección: https://doc.lagout.org/science/0_Computer%20Science/2_Algorithms/Boosting_%20Foundations%20and%20Algorithms%20%5BSchapire%20%26%20Freund%202012-05-18%5D.pdf.
- [20] S. Zafeiriou, G. Tzimiropoulos y M. Pantic, “300 W: Special issue on facial landmark localisation “in-the-wild””, *Image and Vision Computing*, vol. 47, págs. 1-2, mar. de 2016, ISSN: 02628856. DOI: 10.1016/j.imavis.2016.03.009. dirección: <http://linkinghub.elsevier.com/retrieve/pii/S0262885616300385>.
- [21] L. I. Smith, “A tutorial on Principal Components Analysis”, 2002.
- [22] K. Kim, “Face Recognition using Principle Component Analysis”, dirección: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.383.6655&rep=rep1&type=pdf>.
- [23] P. Wagner, “Face Recognition with Python”, inf. téc., 2012. dirección: <http://www.bytefish.de>.
- [24] C. J. Burges, “A tutorial on support vector machines for pattern recognition”, *Data Mining and Knowledge Discovery*, 1998, ISSN: 13845810. DOI: 10.1023/A:1009715923555.
- [25] V. N. Vapnik, *Statistical Learning Theory*. Wiley-Interscience, 1998.
- [26] *Logitech HD PRO WEBCAM C920*, Logitech, ene. de 2012.

- [27] P. Refaeilzadeh, L. Tang y H. Liu, “C Cross-Validation”, inf. téc. dirección: <http://leitang.net/papers/ency-cross-validation.pdf>.
- [28] I. H. (H. Witten, E. Frank, M. A. (A. Hall y C. J. Pal, *Data mining : practical machine learning tools and techniques*, pág. 621, ISBN: 9780128043578.
- [29] S. Visa, B. Ramsay, A. Ralescu y E. Van Der Knaap, “Confusion Matrix-based Feature Selection”, inf. téc. dirección: <http://ceur-ws.org/Vol-710/paper37.pdf>.
- [30] C. Beleites, R. Salzer y V. Sergo, “Validation of Soft Classification Models using Partial Class Memberships: An Extended Concept of Sensitivity & Co. applied to the Grading of Astrocytoma Tissues”, vol. 122, págs. 12-22, 2013. DOI: 10.1016/j.chemolab.2012.12.003. dirección: <http://dx.doi.org/10.1016/j.chemolab.2012.12.003>. <http://arxiv.org/abs/1301.0264>, includingthesupplementaryfiguresandtables..
- [31] V. Kazemi y J. Sullivan, “One millisecond face alignment with an ensemble of regression trees”, en *2014 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, jun. de 2014, págs. 1867-1874, ISBN: 978-1-4799-5118-5. DOI: 10.1109/CVPR.2014.241. dirección: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6909637>.
- [32] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar e I. Matthews, “The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression”, inf. téc. dirección: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.182.3759&rep=rep1&type=pdf>.

15.1. Imágenes bases de datos



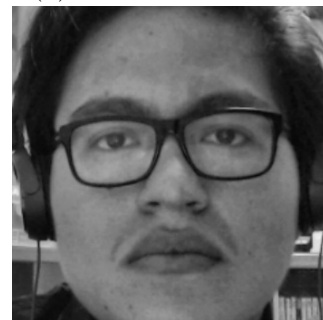
(a) Base de datos *Cohn-Kanade*



(b) Base de datos *UVG*



(c) Base de datos *Ambiente 2*

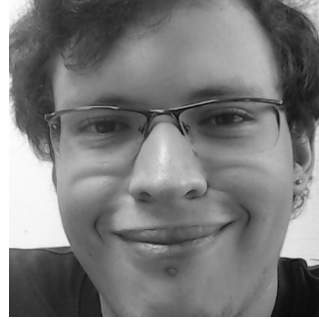


(d) Base de datos *Personalizada*

Figura 71: Rostros de neutralidad



(a) Base de datos *Cohn-Kanade*



(b) Base de datos *UVG*

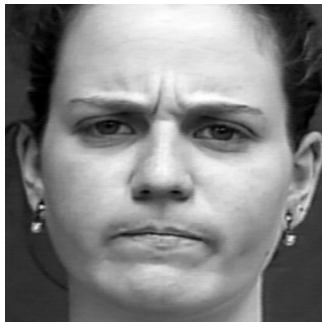


(c) Base de datos *Ambiente 2*

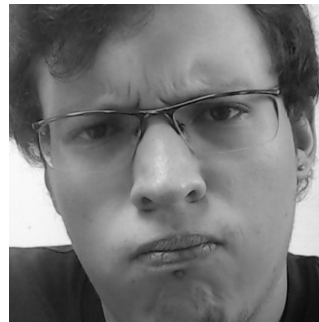


(d) Base de datos *Personalizada*

Figura 72: Rostros de felicidad



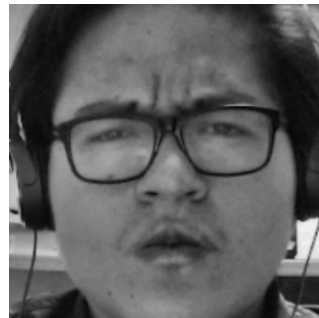
(a) Base de datos *Cohn-Kanade*



(b) Base de datos *UVG*



(c) Base de datos *Ambiente 2*

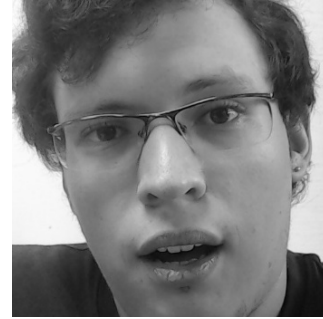


(d) Base de datos *Personalizada*

Figura 73: Rostros de enojo



(a) Base de datos *Cohn-Kanade*



(b) Base de datos *UVG*



(c) Base de datos *Ambiente 2*

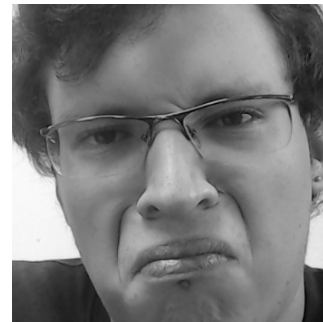


(d) Base de datos *Personalizada*

Figura 74: Rostros de sorpresa



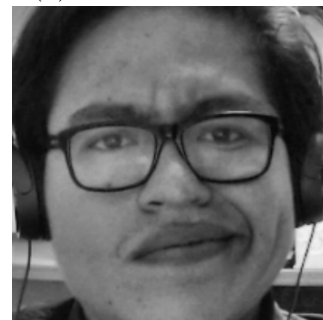
(a) Base de datos *Cohn-Kanade*



(b) Base de datos *UVG*



(c) Base de datos *Ambiente 2*



(d) Base de datos *Personalizada*

Figura 75: Rostros de disgusto

15.2. Manual

15.2.1. Hardware necesario

- Procesador i3, similar o superior. Las pruebas se realizaron con un procesador i7 y una tarjeta grafica *Intel HD Graphics* ¹.
- Cámara web con la capacidad de obtener imágenes superior a 5 cuadros por segundo. La cámara utilizada fue un *Logitech C920 HD PRO*.

15.2.2. Software necesario

- Python Versión 2.7.10
- OpenCV Versión 2.4.0
- Base de Datos de Rostros ².
- Haarscade_Frontal_Face_Default.xml
- Sklearn Python
- FacialLandmarks.DAT
- Dlib (para poder utilizar archivo *Facial Landmarks*)

15.3. Implementación

Programa *Fisher Face*

- I Colocar archivos, ReconocimientoRostro, Update_Model, Haarscade_Frontal_Face_Default.xml en una misma carpeta.
- II Generar una carpeta de nombre *dataset*. En el caso de ya tener separada la base de datos, separar en esta carpeta los archivos por emoción, colocando número desde cero hasta el numero de imágenes que se tengan por emoción. Si no se tiene una base de datos, basta con crear la carpeta *dataset*.
- III Abrir el archivo ReconocimientoRostro, y modificar la linea 31 a partir de las expresiones que se quieran detectar. En el caso del proyecto, se identificarán cinco expresiones:
 - Neutral
 - Disgusto
 - Enojo
 - Felicidad

¹Se incluyen el equipo utilizado ya que depende del tamaño del tamaño de la base de datos, puede llegar a afectar el rendimiento del programa en tiempo real

²Opcional, ya que se puede generar la propia

- Sorpresa

Se puede agregar o modificar dependiendo de la base de datos o la característica que se quiere identificar. Se recomienda no clasificar más de ocho características diferentes debido el algoritmo *Fisher Face* tiene un mejor rendimiento cuando éstas se reducen. El tiempo computacional aumentará al momento de aumentar las características a detectar, lo cual será algo a tomar en cuenta dependiendo del equipo. Es importante mencionar que esta implementación se encuentra con la limitante de memoria de direcciones por medio de *python* de 32 bits, por lo cual el set de imágenes tendrá que ser menor a 400, de lo contrario no podrá hacer un direccionamiento correcto y se obtendrán errores de falta de memoria.

IV En el caso que se quiere utilizar el entrenamiento previamente generado, colocar el archivo *Entrenamiento.xml* que servirá para la predicción de rostros. Si ya se tiene este, adelantarse al paso 6. En caso no se tenga el archivo o se dese generar un entrenamiento personalizado, continuar al paso 5.

V Correr el archivo *ReconocimientoRostro*, este mostrará una interfaz en consola, que va dictando las instrucciones:

"Porfavor hacer emoción + **EMOCIÓN** + cuando acaba el temporizador y mantenerla constante durante 5 seg"

Para cada una de las diferentes emociones a reconocer. Para este paso se debe estar frente a la cámara sin ninguna otra persona en el marco visual, para poder realizar la clasificación de manera correcta. El programa en lugar de capturar imágenes durante cinco segundos, obtiene 15 imágenes que representan esa expresión. En esta etapa puede mostrar mensajes como:

- "No se detecta ningún rostro"
- "Múltiples rostros detectados"

El programa automáticamente no guardará las imágenes que poseen alguno de estos errores. El programa indicará la siguiente instrucción luego de haber obtenido las 15 imágenes de cada expresión.

Una vez finalizado mostrará dos mensajes de que se logró completar el entrenamiento:

- El primero de estos mostrará el tamaño del set de imágenes que se estará analizando³.
- El segundo mostrará un mensaje de que finalizó y que se encuentra listo para poder detectar las expresiones faciales En este paso se observa que en la carpeta de trabajo se generó un archivo *Entrenamiento.xml* esta es el archivo que se utilizará para poder realizar las comparaciones y detección de expresiones.

VI En caso de seguir agrandando la muestra de imágenes, se repite el paso 4. Solo tomar en cuenta que se deben mantener las mismas expresiones si se quiere ir agregándolo al entrenamiento.

³En caso de un set demasiado grande, mostrará un mensaje de error

- VII El archivo *ReconocimientoRostro.py* esta previamente determinado para obtener las expresiones, en tiempo real. Por lo cual al llegar a este paso, no queda más que poner a prueba el programa. En este paso el programa genera una ventana extra, para que se pueda ver en tiempo real, los cuadros captados por la cámara web.
- VIII La ventana que se genera con el rostro, tiene dibujado un recuadro en la cual se puede observar el rostro en el plano de la cámara. Este recuadro sirve para poder determinar las coordenadas que se encuentra el rostro.
- IX Las salidas finales del programa consisten en un arreglo de tres espacios, como se observa en la Figura 76:

S_0 . Salida Expresion:

- 1) Neutral
- 2) Enojo
- 3) Disgustado
- 4) Felicidad
- 5) Sorpresa

S_1 . Coordenada X, centro de rostro.

S_2 . Coordenada Y, centro de rostro.

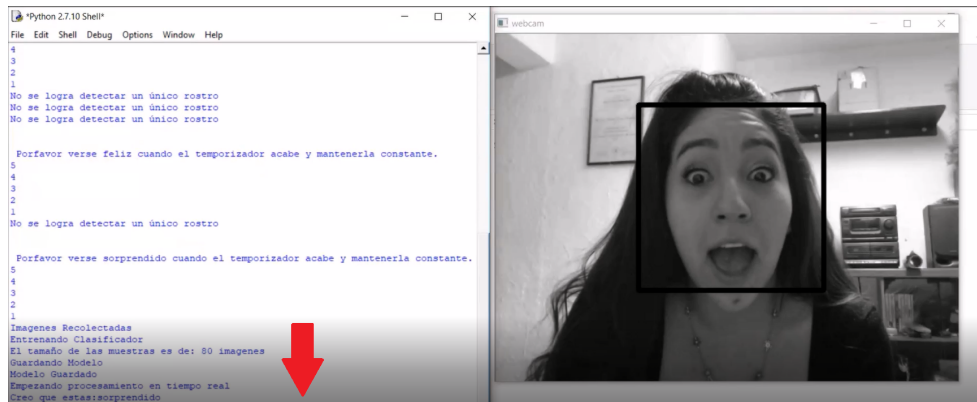


Figura 76: Salida de datos algoritmo *Fisher Face*

Programa SVM

En este programa se implementó utilizando un programa para poder encontrar las marcas de cara, además del algoritmo SVM para la predicción del reconocimiento de gestos. En este se buscaban dos formas diferentes tanto de detectar el rostro como la predicción de las expresiones que al momento de compararlos nos diera la ventaja de cada uno y se decidiera la implementación de uno.

- I Definir una carpeta o localización en donde se encuentre el programa *VersiónFinalSVM.py* junto con el archivo *FacialLandmarks.dat*, este último contiene información de entrenamiento para la detección de rostros.

- II En el espacio donde se está trabajando, crear un carpeta con el nombre *dataset*. Al igual que para el programa *Fisher Face*, en caso de que se tenga una base de datos descargada o propio. Separar esta carpeta por gestos y dentro de ella las imágenes que identifiquen estas expresiones. De no tener una base de datos, limitarse a generar la carpeta *dataset*.
- III Abrir el archivo *VersiónFinalSVM.py*, en el caso de la línea 20, se tienen un listado con las expresiones a reconocer. De forma predefinida se tiene que pueda reconocer cinco expresiones:
- Neutral
 - Disgusto
 - Enojo
 - Felicidad
 - Sorpresa

Tal y como se mencionó anteriormente, estas se pueden agregar o eliminar ciertas expresiones. En el caso de querer modificar esta situación se deben de generar nuevos archivos de entrenamiento que se realizarán a partir del paso 5. A diferencia del método *Fisher Face* las características que utiliza para clasificar cada imagen se limitan a los 68 puntos de cara que logra identificar, por lo cual el set de datos puede ser aún más grande. Actualmente se lograron generar entrenamientos con sets de datos de más de 3,000 fotos. Sin embargo es de tomar en cuenta que el tiempo que se tarda en generar estos entrenamientos es superior que cuando se trabajaba con un número de datos cercanos a los 300 datos.

- IV En el caso que se quiere utilizar un entrenamiento previamente generado, colocar el archivo *EntrenamientoSVM.xml* dentro de la carpeta de trabajo para que pueda acceder a este archivo. En el caso que se que no se tenga esta, o se quiere realizar modificaciones o se quiere personalizar el entrenamiento continuar con el paso 5. En caso de usar el archivo previamente mencionado, avanzar al paso 6.
- V Correr el archivo *VersiónFinalSVM.py* en el caso que automáticamente no reconozca el archivo de entrenamiento, solicitará que al usuario que seleccione si quiere generar un base de datos, o si quiere generar un entrenamiento a partir de datos previamente seleccionados.
- Base de Datos Externa: En el caso que se seleccione esta opción debe asegurarse que los archivos ya se encuentran separados en las carpetas mencionadas en el paso 2. Tomar en cuenta que entre mayor es la cantidad de fotos que se tengan, el tiempo de espera sera mayor.
 - Base Personalizada: En el caso de haber seleccionado esta opción el programa mostrará un interfaz en consola con las instrucciones que tendrá que realizar el usuario. "Porfavor hacer emoción + **EMOCIÓN** + cuando acaba el temporizador y mantenerla constante durante 5 segundos". Para cada unas de las expresiones a reconocer. Tener en cuenta que el entrenamiento sera entrenado para reconocer las expresiones a esta distancia. Así que realizar esta paso en el ambiente exacto en donde se tendrá que reconocer las expresiones o en un ambiente muy similar en el cual se realizará la predicción. El programa obtiene 15 imágenes durante aproximadamente estos 15 segundos, sin embargo el programa puede mostrar mensajes de error como:

- "No se detecta ningún rostro"
- "Múltiples rostros detectados"

En el caso de mostrar alguno de estos datos, el programa no guardará los datos de ese cuadro de captura, sino seguirá obteniendo datos, hasta que se completen las 15 imágenes por expresión. Un vez finalizado para las 5 expresiones, mostrará un mensaje que se está inicializando el entrenamiento y se generará un archivo llamado *EntrenamientoSVM.xml* en la carpeta donde se está trabajando.

VI En este paso ya se tiene el archivo de entrenamiento y el programa listo para el reconocimiento de gestos. Por lo cual a partir de este punto, el programa generará una ventana la cual contiene los cuadros obtenidos por la cámara y en caso de reconocer algún rostro mostrará una marca de cara sobre la imagen. En el caso que se encuentre en el rango que el programa pueda reconocer gestos, no solo colocará una marca en medio de su rostro, sino que identificará las 68 marcas de cara.

VII Las salidas finales que tendrá este programa consisten en un arreglo de tres espacios, como se observa en la Figura 77:

S_0 . Salida expresión:

- 1) Neutral
- 2) Enojo
- 3) Disgustado
- 4) Felicidad
- 5) Sorpresa

S_1 . Coordenada X, centro de rostro.

S_2 . Coordenada Y, centro de rostro.

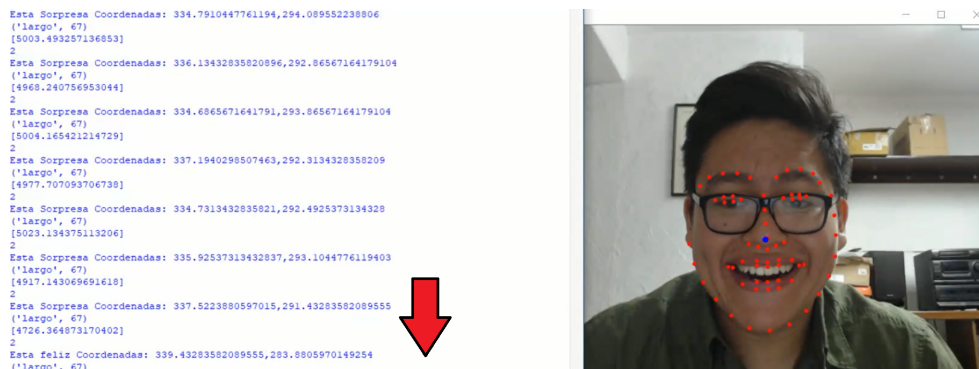


Figura 77: Salida de datos algoritmo SVM

Animatrónico Robot diseñado para simular apariencia y comportamiento de una persona, animal u objeto. 21

Característica En procesamiento de imágenes, las características se refieren a los elementos que resultan claves para la detección de un objeto, en este caso rostro. 14–18, 21–23, 26

Entrenamiento En el ámbito de clasificación, el entrenamiento se refiere al grupo de datos del cual se extraerán características principales que logran generar una diferenciación para la clasificación del grupo de datos de pruebas. 17, 21, 24, 26

Gestos faciales Expresiones realizadas con la cara, las cuales el ser humano puede interpretar para predecir emociones. 14

Vectores propios También conocidos como *Eigen valores* los vectores propios brindan información acerca la naturaleza de transformación lineal de la matriz. 20, 21