



Universidad Autónoma de Querétaro
Facultad de Ingeniería
Maestría en Ciencias en Inteligencia Artificial

Sistema embebido para la detección de expresiones faciales y clasificación de emociones a través de una máquina de soporte vectorial

TESIS

Que como parte de los requisitos para obtener el grado de
Maestro en Ciencias en Inteligencia Artificial

Presenta:

Ing. Francisco Emiliano Aguayo Serrano

Dirigido por:

Dr. Jesús Carlos Pedraza Ortega

SINODALES

Dr. Jesús Carlos Pedraza Ortega

Presidente

Dr. Juan Manuel Ramos Arreguín

Secretario

Dr. Efrén Gorrostieta Hurtado

Vocal

Dr. Marco Antonio Aceves Fernández

Suplente

Dr. Saúl Tovar Arriaga

Suplente

Centro Universitario
Querétaro, QRO
México.
Julio 2020

RESUMEN

En este trabajo de investigación se muestra la implementación de un sistema de reconocimiento y clasificación de expresiones faciales (EF) en una computadora de escritorio así como en un sistema embebido, utilizando una Máquina de Soporte Vectorial (MSV, o por sus siglas en inglés SVM). En el desarrollo del sistema se sigue una metodología propia que consta de seis pasos. El primer paso es la adquisición de las imágenes, donde se muestran las bases de datos utilizadas; como segundo paso se aplica un pre-procesamiento a las imágenes para su calidad y normalizar los datos; en el tercer paso se detectan rasgos faciales y se resaltan las características de forma y apariencia del rostro; en el cuarto paso se implementa el Análisis Discriminante Lineal (ADL, o LDA por sus siglas en inglés) y el Análisis del Componente Principal (ACP, o PCA por sus siglas en inglés), con esto se reduce y se elimina información que sea redundante o irrelevante; en el quinto paso se entrenan y clasifican los datos, para ello se utiliza SVM con distintos núcleos; como sexto y último paso, se realiza una serie de experimentos aplicando métricas para probar y medir los resultados obtenidos. Como principal contribución o novedad en este trabajo se muestra la utilización del algoritmo K-Means para ayudar con la función de clasificación de SVM, esto con el fin de reducir las clasificaciones incorrectas en el paso de entrenamiento del algoritmo. Se realizaron diferentes pruebas en dos escenarios. El primer escenario solo con caras extraídas y el segundo escenario con la aplicación de técnicas de procesamiento de imágenes. El promedio de exactitud en la clasificación de SVM con LDA + SVM fue del 80 % mientras que LDA + K-MEANS + SVM fue del 96 %.

Palabras clave: Sistema, Reconocimiento, Clasificación, Extracción, Expresiones Faciales. Procesamiento.

SUMMARY

This research work shows the implementation of a facial expression recognition and classification system (FE) in a desktop computer as well as in an embedded system, using a Support Vector Machine (SVM). The development of the system follows its own methodology consisting of six steps. The first step is the acquisition of the images, where the databases used are displayed; as a second step, a pre-processing is applied to the images for their quality and to normalize the data; In the third step, facial features are detected and the shape and appearance characteristics of the face are highlighted; In the fourth step the Linear Discriminant Analysis (LDA) and the Principal Component Analysis (PCA) are implemented, with this reducing and eliminating information that is redundant or irrelevant; In the fifth step, the data is trained and classified, for this, SVM with different cores is used; As the sixth and last step, a series of experiments is carried out applying metrics to test and measure the results obtained. As the main contribution or novelty in this work, the use of the K-Means or K-Means algorithm is shown to help with the SVM classification function, this in order to reduce incorrect classifications in the training step of the algorithm. Different tests were carried out in two scenarios. The first scenario with only drawn faces and the second scenario with the application of image processing techniques. The average accuracy in the classification of SVM with LDA + SVM was 80 % while LDA + K-MEANS + SVM was 96 %.

Keywords: System, Recognition, Classification, Extraction, Facial Expressions.

AGRADECIMIENTOS

En primer lugar quiero agradecer al Consejo Nacional de Ciencia y Tecnología (CONACYT) por la beca recibida durante los dos años en la Maestría en Ciencias de Inteligencia Artificial. Agradezco a la Universidad Autónoma de Querétaro (UAQ) por el apoyo de la beca de inscripción durante el mismo periodo así como la infraestructura prestada. También quiero agradecerle al investigador y profesor de tiempo completo, Dr. Jesús Carlos Pedraza Ortega por dirigir este trabajo de investigación, por su ayuda con la revisión de mi tesis, sus valiosos consejos y comentarios que me han hecho mejorar en esta carrera científica.

Índice general

CAPÍTULOS	PÁGINA
RESUMEN	I
SUMMARY	II
AGRADECIMIENTOS	III
ÍNDICE GENERAL	V
ÍNDICE DE TABLAS	VI
ÍNDICE DE FIGURAS	IX
ACRÓNIMOS	X
1 Introducción	1
1.1 Descripción del problema	1
1.2 Justificación	2
1.3 Hipótesis	2
1.4 Objetivo general	3
1.5 Objetivos específicos	3
1.6 Alcances y limitaciones	3
1.7 Organización de la tesis	4
2 Antecedentes	6
2.1 Aprendizaje máquina	6
2.2 Lenguaje de programación Python	7
2.3 Conceptos Matemáticos	10
2.4 Reconocimiento Facial	19
2.5 Trabajos recientes	21
3 Metodología	26
3.1 Descripción general	26
3.2 Adquisición de imágenes	28
3.2.1 Base de datos pública <i>Cohn-Kanade (CK+)</i>	28
3.2.2 Base de datos pública JAFFE	29

3.2.3	Base de datos pública RAVDESS	30
3.3	Pre-procesamiento	31
3.3.1	Alineación del rostro	31
3.3.2	Detección del rostro	33
3.3.3	Extracción del rostro	34
3.3.4	Aplicación de filtros	36
3.3.5	Normalización de los datos	38
3.4	Reducción de la dimensión de los datos	38
3.4.1	Análisis Discriminante Lineal	39
3.4.2	Análisis del Componente Principal Lineal	46
3.5	Agrupamiento y clasificación de los datos	47
3.5.1	Máquina de Soporte Vectorial	47
3.5.2	K-Means	54
3.6	Entrenamiento y pruebas	65
3.7	Resultados	68
3.7.1	Escenario experimental	68
3.7.2	tablas de resultados	70
4	Conclusiones y trabajo futuro	73
	Bibliografía	74
	Anexos	78

Índice de tablas

TABLA		PÁGINA
1	Resultados de la clasificación SVM con la base de datos JAFFE sin técnicas de pre-procesamiento	70
2	Resultados de la clasificación SVM con la base de datos JAFFE con técnicas de pre-procesamiento	71
3	Resultados de la clasificación SVM con la base de datos CK+ sin técnicas de pre-procesamiento	71
4	Resultados de la clasificación SVM con la base de datos CK+ con técnicas de pre-procesamiento	71
5	Resultados de la clasificación SVM con la base de datos RAVDESS sin técnicas de pre-procesamiento	72
6	Resultados de la clasificación SVM con la base de datos RAVDESS con técnicas de pre-procesamiento	72

Índice de figuras

FIGURA	PÁGINA
1	Tipos de aprendizaje máquina, (Francois, 2017) 6
2	Línea de tiempo de las versiones de <i>python</i> , (History of Python, 2020) 8
3	Interpretación de un vector \vec{u} en un espacio R^2 10
4	Interpretación geométrica de la suma de los vectores \vec{u} y \vec{v} en un espacio R^2 11
5	Interpretación geométrica del producto de un escalar λ por un vector \vec{u} 11
6	Interpretación geométrica de la norma de un vector 12
7	Interpretación geométrica de la distancia entre dos vectores 12
8	Interpretación geométrica del producto punto de dos vectores 13
9	Estructura de los estados de clasificador Haar Cascade, (Viola et al., 2001) 19
10	Ejemplo de características de rectángulo mostradas en relación con la ventana de detección adjunta 20
11	Antes y después del cálculo de la imagen integral donde (a) es la imagen original, (b) es la imagen integral 20
12	Ejemplos de combinaciones de unidades de acción de FACS (Ekman, 1977) 21
13	Diagrama de flujo del sistema (a). Siete triángulos que representan puntos de referencia en un rostro humano (b). Propuestos por (Shah et al., 2017) 22
14	Arquitectura del sistema del método propuesto por (Bellamkonda y Gopalan, 2018) 23
15	Marco del sistema de reconocimiento facial 3D invariante de expresión basado en SVM propuesto por (Leo y Suchitra, 2018) 23
16	Diagrama de bloques del sistema de reconocimiento de expresión facial propuesto por (Kar et al., 2019) 24
17	Pasos de cálculo del umbral adaptativo propuesto por (Makhmudkhujaev et al., 2019). 25
18	Metodología propuesta para el trabajo de investigación. 26
19	Ejemplo de la secuencia de imágenes de la base de datos Cohn-Kanade (CK+). 28
20	Extracto de imágenes con signos de sorpresa de la base de datos pública Cohn-Kanade (CK+). 28
21	Carpetas donde se almacenan las imágenes que forman la base de datos pública <i>Cohn-Kanade</i> (CK+). 29
22	Muestra de imágenes con signos de felicidad de la base de datos pública JAFFE. 30
23	Ejemplo de imágenes seleccionadas manualmente de los videos, mostrando la expresión facial con signo de enojo (Livingstone y Russo, 2018). 31

24	Ejemplo del proceso de rotar un rostro.	32
25	Ejemplo de los valores de rotación (grados) en la base de datos COHN-CANADE CK+.	32
26	Ejemplo de detección del rostro con características Haar en cascada.	34
27	Extracción de rostros de JAFFE con rasgos visibles de tristeza.	34
28	Extracción de rostros de Cohn-Kanade (CK+) con rasgos visibles de enojo.	36
29	Resultados de aplicar un un filtro gaussiano con valores del kernel de (a) 3×3 ; (b) 5×5 ; (c) 7×7	37
30	Ejemplo de aplicación de un filtro promedio y ecualización.	37
31	Representación gráfica de la matriz X	39
32	Representación gráfica del calculo del promedio de cada clase y el promedio total de todas las imágenes	40
33	Representación gráfica de la matriz entre clase S_B	41
34	Representación gráfica de la matriz dentro de la clase S_W	41
35	Representación gráfica del calculo del nuevo espacio dimensional W a partir de los vectores propios	42
36	Nuevo sub-espacio de características proyectadas por LDA de la base de datos JAFFE (a) sin pre-procesamiento, (b) con pre-procesamiento.	43
37	Nuevo sub-espacio de características proyectadas por LDA de la base de datos CK+ (a) sin pre-procesamiento, (b) con pre-procesamiento.	44
38	Nuevo sub-espacio de características proyectadas por LDA de la base de datos RAVDESS (a) sin pre-procesamiento, (b) con pre-procesamiento.	45
39	Representación gráfica del algoritmo del Análisis del Componente Principal	46
40	Comparación gráfica entre LDA y PCA	47
41	Nuevo sub-espacio de características proyectadas por PCA de la base de datos JAFFE (a) con pre-procesamiento, (b) sin pre-procesamiento,	48
42	Nuevo sub-espacio de características proyectadas por PCA de la base de datos CK+ (a) con pre-procesamiento (b), sin pre-procesamiento.	49
43	Nuevo sub-espacio de características proyectadas por PCA de la base de datos RAVDESS (a) con pre-procesamiento, (b) sin pre-procesamiento.	50
44	Ejemplo de clasificación binaria utilizando SVM.	52
45	Interpretación geométrica de los parámetros w y b	53
46	Ejemplo del problema de optimización de calle.	54
47	Ejemplo de la estrategia de clasificación multi-clase uno contra todos.	55
48	SVM con kernel Lineal y Radial utilizando el nuevo sub-espacio de características de PCA en la base de datos JAFFE (a) con pre-procesamiento, (b) sin pre-procesamiento.	56
49	SVM con kernel Lineal y Radial utilizando el nuevo sub-espacio de características de PCA en la base de datos JAFFE (a) con pre-procesamiento, (b) sin pre-procesamiento.	57
50	SVM con kernel Lineal y Radial utilizando el nuevo sub-espacio de características de PCA en la base de datos RAVDESS (a) con pre-procesamiento, (b) sin pre-procesamiento.	58

51	SVM con kernel Lineal y Radial utilizando el nuevo sub-espacio de características de PCA en la base de datos JAFFE (a) con pre-procesamiento, (b) sin pre-procesamiento.	59
52	SVM con kernel Polinomial utilizando el nuevo sub-espacio de características de LDA en la base de datos JAFFE (a) con pre-procesamiento, (b) sin pre-procesamiento.	60
53	SVM con kernel Lineal y Radial utilizando el nuevo sub-espacio de características de LDA en la base de datos CK+ (a) con pre-procesamiento, (b) sin pre-procesamiento.	61
54	SVM con kernel Radial y Lineal utilizando el nuevo sub-espacio de características de LDA en la base de datos RAVDESS (a) con pre-procesamiento, (b) sin pre-procesamiento.	62
55	SVM con kernel Lineal y Radial utilizando el nuevo sub-espacio de características de PCA en la base de datos JAFFE sin pre-procesamiento	63
56	SVM con kernel Lineal y Radial utilizando el nuevo sub-espacio de características de PCA en la base de datos JAFFE sin pre-procesamiento	64
57	Example of the optimization problem in the stret	65
58	Herramientas de hardware y software utilizadas	68
59	Tarjeta de desarrollo Raspbery	68
60	Sistema embebido conectado (a). PCA (b) y LDA + KMEANS + SVM (c).	69
61	Constancia de participación como presentador oral en el Coloquio de la Facultad de Ingeniería de la Universidad Autónoma de Querétaro	78
62	Constancia de cumplimiento del requisito de inglés emitida por la Facultad de Lenguas y Letras de la Universidad Autónoma de Querétaro, equivalente a 550 puntos TOEFL	79

ACRÓNIMOS

DPR: Detección de Puntos de Referencia

CE: Conjunto de Entrenamiento

SVM: (*Support Vector Machine*, Máquina de Soporte Vectorial)

PCA: (*Principal Component Analysis*, Análisis del Componente Principal)

LDA: (*Linear Discriminant Analysis*, Análisis Discriminante Lineal)

ECF: Extracción de Características Faciales

RF: Reconocimiento Facial

EF: Expresiones Faciales

MLPs: Puntos de referencia medios

REF: Reconocimiento de Expresiones Faciales

LS-SVM: (*Least-squares Support Vector machine*, Máquina de vector de soporte de mínimos cuadrados)

RBF: (*Radial Base Function*, Funcion de Base Radial)

JAFFE: (*Japanese Female Facial Expression*, Expresiones Faciales de Mujeres Japonesas)

LBP: (*Local Binary Pattern*, Patrón Binario Local)

Introducción

1.1. Descripción del problema

Los seres humanos son capaces de establecer una comunicación a través de un lenguaje, también son capaces de organizar pensamientos, expresar ideas y emociones que le permiten interactuar con el mundo que los rodea. La expresión facial es una forma de comunicación que exterioriza ideas, pensamientos y emociones, las cuales se pueden expresar a través de contracciones y movimientos de los músculos del rostro. Es claro que las emociones influyen en el desarrollo del ser humano, tal y como se muestra en (López, 2000). También el trabajo de (Retana, 2012) muestra que las emociones son claves en el proceso de aprendizaje y de trabajo. Un mal manejo de las emociones puede reflejar estrés, así como problemas de conducta externa e interna, provocando afectaciones en las habilidades cognitivas, de aprendizaje y concentración. Por estas razones, en este trabajo de investigación se propone combinar dos ideas, las emociones y la inteligencia, al proponer el desarrollo e implementación de un sistema embebido que realice una detección del rostro humano, la extracción de sus características faciales, la clasificación de dichas expresiones y la estimación del estado emocional de una persona. Dicho sistema automático, podría ser utilizado como herramienta que ayude en la investigación de otras áreas del conocimiento como la Psicología, al ser implementado para el monitoreo de las expresiones faciales. Al detectar y clasificar expresiones faciales, es posible llevar a cabo un análisis detallado de la información recopilada. La interpretación de tales expresiones faciales sigue siendo un reto y se necesita mucha investigación, debido a que muchos sistemas automáticos, se centran en reconocer un

conjunto limitado de emociones. Esto se debe, principalmente, a la falta de mejores técnicas de extracción de características, clasificación y análisis sobre la forma en que se relacionan las expresiones faciales con las emociones del ser humano.

1.2. Justificación

En la actualidad uno de los problemas más importantes para el ser humano y al que se le ha puesto especial atención, es la del reconocimiento de EF, debido a que forman una pieza clave en la comunicación no verbal, ya que se ha demostrado de forma empírica que estas EF transmiten emociones e intenciones. Esta área resulta muy interesante y desafiante debido a que las EF pueden ser subjetivas e incluso parecidas en ciertas situaciones, ocasionando confusión al interpretarlas. Es por esto que se ha puesto especial atención en el desarrollo de sistemas automáticos que ayuden a interpretar dichas EF. Este tipo de sistemas se caracterizan por realizar una estimación del estado emocional de una persona, a partir de la captura de la imagen del rostro, aplicando un pre-procesamiento, posteriormente una extracción de características faciales, la clasificación de dichas expresiones (felicidad, tristeza, enojo, disgusto, sorpresa y miedo) consideradas como universales, y por último, determinar la emoción que se está experimentando. Resulta interesante para distintas áreas, como la publicidad, la aplicación de este tipo de sistemas para analizar el rostro de una persona mientras visualiza un anuncio, un comercial de televisión o un producto en una tienda. También se podría utilizar como herramienta medica en personas con autismo o Alzheimer, donde puede resultar útil monitorear la expresión facial de un paciente ante un determinado estímulo para ayudar en el diagnóstico o seguimiento durante la enfermedad. Otra posible aplicación sería como herramienta en interrogatorios policíacos o bien en la evaluación psicológica de una persona.

1.3. Hipótesis

El uso de una máquina de soporte vectorial, permite llevar a cabo una clasificación de expresiones faciales a partir de las características del rostro y así detectar emociones en un sistema embebido de visión por computadora proporcionando un resultado válido y confiable.

1.4. Objetivo general

Desarrollar un sistema de detección de expresiones faciales y clasificación de emociones mediante una Máquina de Soporte Vectorial en un sistema embebido.

1.5. Objetivos específicos

1. Estudiar métodos de detección y clasificación de expresiones faciales relacionados con Máquinas de Soporte Vectorial (SVM).
2. Proponer una metodología de detección y clasificación de emociones.
3. Analizar las características faciales del ser humano.
4. Definir el conjunto de imágenes utilizadas en la metodología.
5. Aplicar mejoras o correcciones en las imágenes.
6. Detectar expresiones faciales humanas.
7. Implementar algoritmos de reducción de dimensión en los datos.
8. Clasificar expresiones faciales en seis emociones.
9. Obtener métricas de evaluación para validar la metodología.

1.6. Alcances y limitaciones

Este trabajo de investigación se limita a la exploración de técnicas y algoritmos de procesamiento de imágenes así como de aprendizaje máquina, con el fin de detectar expresiones faciales de una persona y clasificarlas en 6 emociones consideradas como universales (felicidad, enojo, tristeza, sorpresa, miedo, normal). En la fase de entrenamiento y pruebas se utilizan bases de datos públicas, las cuales presentan inconsistencias en el número de imágenes, individuos y expresiones faciales por persona. Esta investigación se centra únicamente en la forma en la que se tratan las imágenes con los rostros; los pasos que hay que seguir para lograr una buena caracterización de las expresiones faciales; así como la utilización de algoritmos de aprendizaje automático existentes en un prototipo de sistema embebido de bajo costo, para lograr un porcentaje de exactitud aceptable por encima del 90 %.

1.7. Organización de la tesis

Este trabajo de investigación se encuentra dividido en seis capítulos y una serie de anexos los cuales representan las actividades realizadas durante los estudios de maestría. A continuación se describen cada uno de ellos de forma general.

Capítulo 1: Introduce al lector al tema de investigación, aquí se muestra una descripción general y concisa del problema que se aborda. La justificación por la cual se decidió abordar dicho problema. Se realiza una declaración explícita del objetivo general así como los objetivos específicos del trabajo de investigación. Finalmente se explican sus alcances y limitaciones.

Capítulo 2: Explica de forma detallada conceptos claves relacionados con el trabajo de investigación que facilitan su entendimiento, así como el estado del arte de sistemas de reconocimiento y clasificación de EF. Se describen y analizan de forma general las técnicas más utilizadas de procesamiento de imágenes y aprendizaje máquina.

Capítulo 3: Describe de forma detallada, concisa y técnica cada paso de la metodología propuesta. Se ilustra cómo se realizó el caso de estudio partiendo de los datos que se utilizaron, la implementación de los algoritmos y los instrumentos de medición aplicados.

Capítulo 4: Muestra los resultados obtenidos en gráficas y tablas que ilustran las salidas de las pruebas de los algoritmos implementados así como los resultados de las métricas empleadas para validar la metodología. También se realiza una comparación del trabajo propuesto con otros autores en cuestión de métodos, tiempo de procesamiento y eficacia.

Capítulo 5: Define las conclusiones que se obtuvieron de este trabajo de investigación con base en las pruebas realizadas y en la comparación del método propuesto con otros autores. Se hace una reflexión sobre la hipótesis planteada y los resultados obtenidos, además de que se enmarcan las implicaciones y el impacto del trabajo. Finalmente se propone continuar con el tema de investigación explorando otras técnicas y métodos.

Capítulo 6: Muestra actividades complementarias realizadas por el autor durante el curso de la maestría, las cuales se realizaron a lo largo del desarrollo de este proyecto de investigación.

Dirección General de Bibliotecas de la UAQ

Antecedentes

2.1. Aprendizaje máquina

Existen varios enfoques en el proceso de aprendizaje máquina y cada uno de ellos está orientado a resolver distintos tipos de problemas, en la Figura 1 se muestran los cinco tipos de aprendizaje que existen hoy en día.

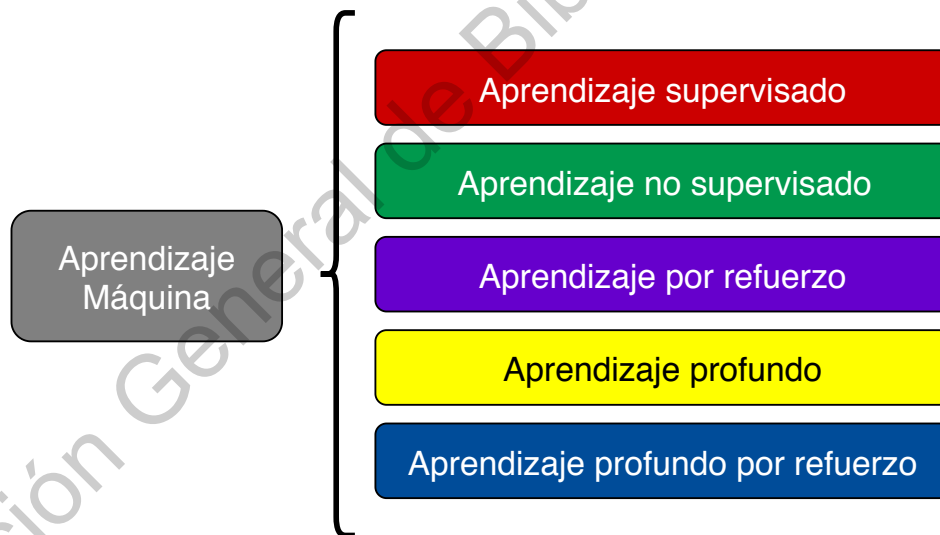


Figura 1. Tipos de aprendizaje máquina, (Francois, 2017)

- **Aprendizaje supervisado:** Es un conjunto de algoritmos de aprendizaje automático que en base a una función, asigna a un conjunto de datos entrada, una salida o etiqueta.
- **Aprendizaje no supervisado:** Es un conjunto de algoritmos de aprendizaje automático que realizan un entrenamiento o agrupamiento en base a inferencias de conjuntos de

datos sin etiquetas.

- **Aprendizaje por refuerzo:** Es un método donde no se busca asociar entradas con una salida. Aquí se busca determinar qué acciones se deben escoger para un agente de software, en un entorno dado con el fin de maximizar alguna función de recompensa.^o premio acumulado.
- **Aprendizaje profundo:** Es un conjunto de algoritmos de aprendizaje automático que intenta modelar abstracciones de alto nivel en datos usando arquitecturas computacionales que admiten transformaciones no lineales múltiples e iterativas de datos expresados en forma matricial o tensorial.
- **Aprendizaje profundo por refuerzo:** Combina las dos áreas de aprendizaje conocidas como aprendizaje por refuerzo y aprendizaje profundo.

2.2. Lenguaje de programación Python

En estos tiempos, la información se ha convertido en uno de los recursos más valiosos con los que el ser humano ha contado en las últimas décadas. Es por esto que en la actualidad la información es conocida como el nuevo petróleo. Estos datos son capturados, ordenados y son analizados con el fin de sacarle provecho en distintos ámbitos, como en la toma de decisiones en negocios, predicción del clima, el estudio de estructuras proteínicas, modelos de negocio, sistemas financieros, etc. Todas estas áreas de conocimiento tienen un enfoque multidisciplinario, debido a que utilizan modelos matemáticos, y el lenguaje de programación Python es una herramienta que ayuda en el procesamiento de estas grandes cantidades de información con dichos modelos de matemáticos.

En este trabajo de investigación se decidió trabajar con el lenguaje de programación que fue creado por (Van Rossum et al., 2007) entre los años 1985 y 1990, *python* es un proyecto de código abierto que está bajo la Licencia Pública General (LPG, o GPL por sus siglas en inglés *General Public License*). Está diseñado para ser un lenguaje altamente legible debido a que utiliza palabras clave en inglés con frecuencia donde otros idiomas usan puntuación, y tiene menos construcciones sintácticas que otros idiomas, en este proyecto de investigación

se está trabajando con la versión 3.7.1 de este lenguaje debido a que es la versión la última versión estable de *python*, tal y como muestra en la Figura 2, donde se visualiza al evolución de dicho lenguaje de programación. A continuación se listan algunas características de este lenguaje que justifican su uso.

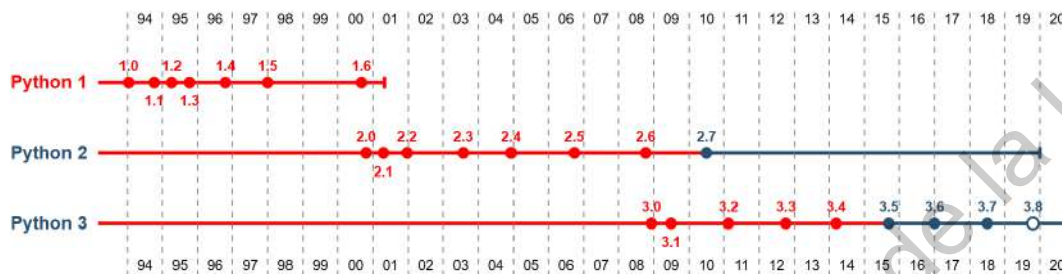


Figura 2. Línea de tiempo de las versiones de *python*, (History of Python, 2020)

- **Alto nivel:** Nivel de abstracción del más alto nivel del lenguaje máquina que traslada los conceptos más técnicos del funcionamiento interno a instrucciones con palabras cortas y fáciles de entender.
- **Lenguaje interpretado:** No necesita ser pre-procesado por un compilador, es posible de ejecutar las instrucciones sin leer y traducir todo el código. Para que esto sea posible hace falta un programa (un intérprete) encargado de traducir cada instrucción escrita con una semántica humana a código máquina.
- **Interactivo:** Es posible modificar el programa mientras se encuentra activo, es decir, el código de programa es la principal interfaz del usuario para ejecutar un proceso
- **Orientado a objetos:** Es un paradigma de programación que trata de trasladar los conceptos de la vida real a la programación mediante el uso de clases (plantillas), es decir, reunir varias técnicas y conceptos, incluyendo la herencia, abstracción, polimorfismo y encapsulamiento.

Para este trabajo de investigación se propuso utilizar algunas librerías, bibliotecas o paquetes de código abierto que cuentan con una licencia de tipo BSD (*Berkeley Software Distribution*), estos paquetes tienen una naturaleza pensada en la computación científica y son ideales para trabajar para un proyecto complejo.

- **OpenCV:** Es una librería escrita para distintos lenguajes de programación y diseñada para abordar problemas relacionados a la visión por computadora. Esta librería fue creada en (Manual, 1999) y desde entonces ha estado en constante evolución. En este trabajo de investigación se utiliza el lenguaje de programación *python* versión 3.8 y *openCV* 4.1.0 .
- **NumPy:** Es el nombre abreviado para *Numerical Python*, es un paquete para el lenguaje de programación *Python*. Con este paquete es posible llevar a cabo operaciones de álgebra lineal, transformadas de Fourier y trabajar con números aleatorios.
- **Matplotlib:** Es un paquete de *python* utilizado para crear gráficas y diagramas 2D. Cuenta con un módulo llamado *pyplot* que facilita las tareas de trazado como histograma, gráficos de barras, espectros de potencia, gráficos de errores, etc. Se usa junto con *NumPy* para proporcionar un entorno que sea una alternativa de código abierto eficaz para *MatLab*.
- **SciPy:** Este paquete diseñado para *python* proporciona una manipulación de matrices N -dimensional, proporciona métodos numéricos fáciles de usar y eficientes, como métodos de integración numérica, interpolación, optimización, álgebra lineal y estadística.
- **Pandas:** Es un paquete de código abierto con licencia BSD que proporciona estructuras de datos de alto rendimiento y fáciles de usar, herramientas de análisis de datos para el lenguaje de programación *Python*. Las áreas de aplicación incluyen los dominios académicos y comerciales, incluyendo finanzas, economía, estadística, análisis, etc.
- **Sklearn:** Es una librería con algoritmos de aprendizaje máquina, bajo licencia BSD para el lenguaje de programación *python*. Incluye métodos de clasificación y regresión. Es necesario tener instaladas las librerías de *numPy* y *sciPy* ya que utiliza algunos de sus algoritmos.

2.3. Conceptos Matemáticos

Espacio vectorial: Es posible definir un espacio vectorial como una estructura matemática que se crea a partir de un conjunto no vacío y que cumple con dos requisitos, el primero es que el conjunto pueda ser sumado con otro conjunto y multiplicado por un escalar, el segundo requisito es que cumpla con una serie de propiedades como: conmutativa, asociativa, elemento neutro, elemento opuesto, distributiva.

Vector: Un vector es una flecha que apunta en una dirección en un espacio R^2 o R^3 . En un espacio R^2 el vector esta representado por un par ordenado tal y como se presenta en la ecuación 1 y como se ilustra en la Figura 3. El origen del vector puede estar ubicado en cualquier lugar del espacio R^2 , por conveniencia el origen del vector se encuentra en la coordenada $(0,0)$. A lo valores de este par ordenado se les conoce como componentes del vector, dicho vector debe cumplir con dos propiedades elementales, la primera es que pueda ser sumado por otro vector y multiplicado por un escalar.

$$\vec{u} = (x, y) \tag{1}$$

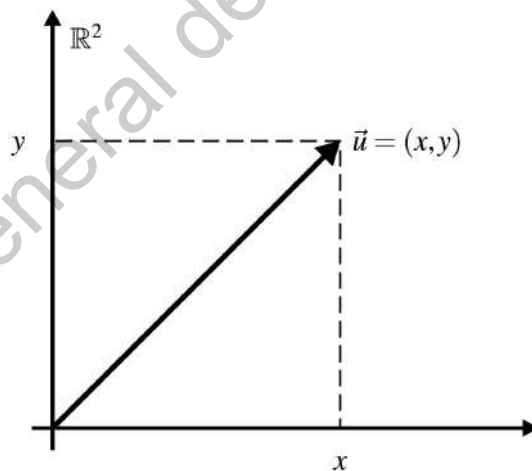


Figura 3. Interpretación de un vector \vec{u} en un espacio R^2

Suma de vectores: La suma de los vectores \vec{u} y \vec{v} se define como la diagonal del paralelogramo que se genera a partir de la suma de los componentes y se denota por la ecuación 2, formando un nuevo vector \vec{w} , tal y como se muestra en la Figura 4.

$$\vec{w} = (\vec{u} + \vec{v}) = (x_1 + x_2, y_1 + y_2) \quad (2)$$

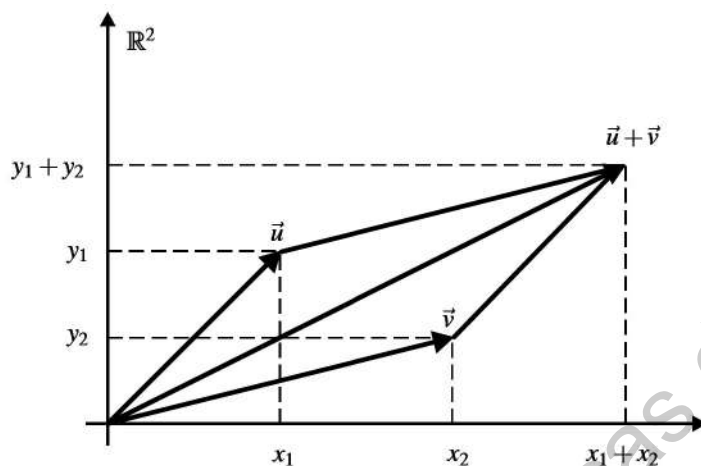


Figura 4. Interpretación geométrica de la suma de los vectores \vec{u} y \vec{v} en un espacio R^2

Multiplicar un escalar por un vector: Es posible multiplicar un número λ que pertenezca a los números reales multiplicando dicho escalar por cada uno de los componentes del vector \vec{u} , tal y como se muestra en la ecuación 3, de esta forma es posible cambiar de tamaño (escalar) o bien cambiar el sentido del vector \vec{u} tal como se muestra en la Figura 5.

$$\lambda \vec{w} = (\lambda x_2, \lambda y_2) \quad (3)$$

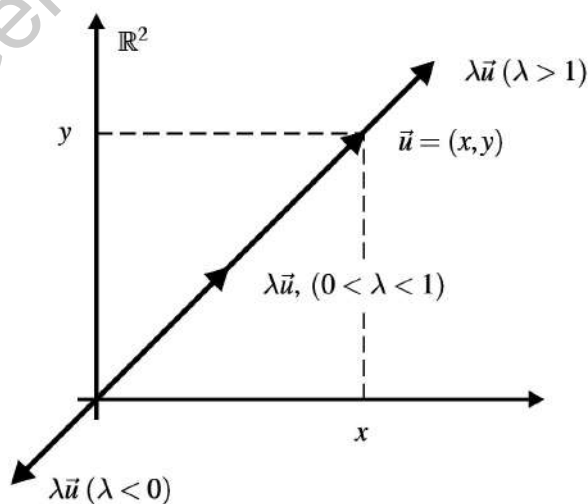


Figura 5. Interpretación geométrica del producto de un escalar λ por un vector \vec{u}

Norma o magnitud: La norma o también conocida como magnitud de un vector \vec{u} es la longitud de dicho vector \vec{u} , tal y como se muestra en la Figura 6 esta distancia esta dada por la ecuación 4.

$$\|\vec{u}\| = \sqrt{x^2 + y^2} \quad (4)$$

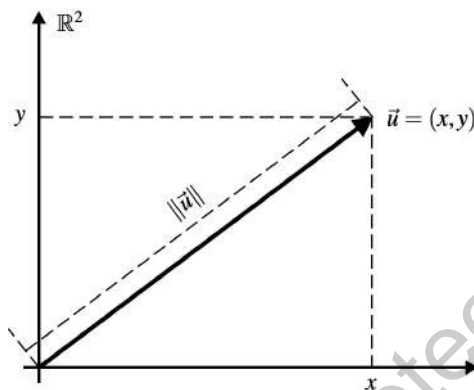


Figura 6. Interpretación geométrica de la norma de un vector

Distancia entre vectores: Teniendo dos vectores dados por \vec{u} y \vec{v} en un espacio R^2 la distancia entre ellos se denota como $d(\vec{u}, \vec{v})$ y se calcula con la norma de la resta entre ambos tal y como se muestra en la ecuación 5 y se ilustra en la Figura 7.

$$d(\vec{u}, \vec{v}) = \|\vec{v} - \vec{u}\| \quad (5)$$

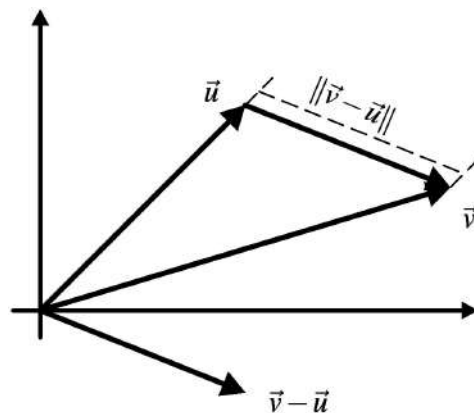


Figura 7. Interpretación geométrica de la distancia entre dos vectores

Producto punto: Esta operación devuelve un escalar, este escalar dice como dos vectores están relacionados entre si. Desde un punto de vista geométrico el producto escalar es la magnitud euclidiana de dos vectores por el coseno del ángulo que existe entre ellos, tal y como se muestra en la ecuación 6. De manera más simple usando propiedades trigonométricas es posible concluir que el producto punto entre dos vectores también se puede calcular la multiplicación de componente a componente y posteriormente sumados, tal y como se muestra en la ecuación 7, y de forma general para dos vectores que tienen un n número de componentes es posible expresar esta operación con la ecuación 8 y se ilustra en la Figura 8.

$$\vec{u} \cdot \vec{v} = \|\vec{u}\| \|\vec{v}\| \cos \theta \quad (6)$$

$$\vec{u} \cdot \vec{v} = x_1 x_2 + y_1 y_2 \quad (7)$$

$$\vec{u} \cdot \vec{v} = \sum_{i=1}^n (x_i y_i) \quad (8)$$

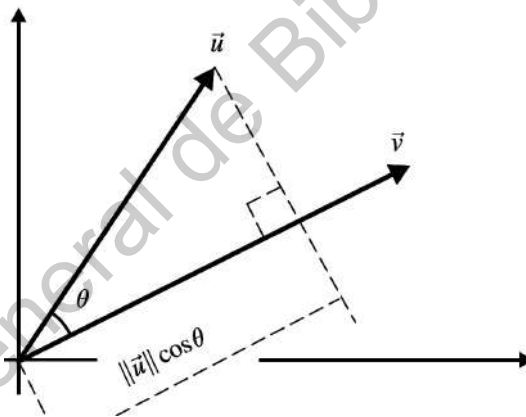


Figura 8. Interpretación geométrica del producto punto de dos vectores

Matriz: Una matriz A es un arreglo de m -filas y n -columnas de $m \times n$ números reales, entonces A es una matriz de tamaño de $m \times n$. A un elemento específico de una matriz se denota por $A = [a_{ij}]$ y se llama componente o entrada de A como se muestra en la ecuación 9

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \vdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \vdots & a_{mn} \end{bmatrix} \quad (9)$$

Matriz cuadrada: Se le conoce como matriz cuadrada A_n a aquella matriz que tiene el mismo número de filas y el mismo número de columnas, también conocida como matriz de orden n . como se muestra en la ecuación 10.

Diagonal: La diagonal de una matriz son los componentes $A = [a_{ij}]$ donde las posiciones de a_{ij} son $i = j$, Tomando los elementos de la matriz de la ecuación 10, los elementos de la diagonal serían 1, 5 y 9.

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \quad (10)$$

Identidad: Se le conoce como matriz identidad a la matriz I donde sus componentes $I = [a_{ij}]$ cumplan las dos condiciones que se muestran en la ecuación 11.

$$a_{ij} = \begin{cases} 1, & \text{si } i = j \\ 0, & \text{si } i \neq j \end{cases} \quad (11)$$

De forma genérica la matriz identidad debe cumplir la forma de la ecuación 12, de tal forma que la matriz identidad siempre es cuadrada, es decir, $I = n \times n$.

$$A = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \vdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \vdots & 1 \end{bmatrix} \quad (12)$$

Transpuesta: Se define como matriz transpuesta como aquella matriz $A = [a_{ij}]$ de tamaño $m \times n$ como $A^T = [b_{ij}]$ donde los elementos $b_{ij} = a_{ji}$, es decir, cambiar los componentes de la matriz de filas por columnas tal y como se muestra en la ecuación 13.

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \end{bmatrix} A^T = \begin{bmatrix} 1 & 5 \\ 2 & 6 \\ 3 & 7 \\ 4 & 8 \end{bmatrix} \quad (13)$$

Matrices triangulares: En la ecuación 14 se muestran dos matrices triangulares. La matriz A de orden n es triangular superior ya que los elementos $[a_{ij}]$ que están por debajo de la diagonal son nulos. La matriz B de orden n es triangular inferior ya que los elementos $[b_{ij}]$ que están por encima de la diagonal son nulos.

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 0 & 6 & 7 & 8 \\ 0 & 0 & 11 & 12 \\ 0 & 0 & 0 & 16 \end{bmatrix} B = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 5 & 6 & 0 & 0 \\ 9 & 10 & 11 & 0 \\ 13 & 14 & 15 & 16 \end{bmatrix} \quad (14)$$

Cofactor: Se define como $c_{ij} = (-1)^{i+j} M_{ij}$, donde c_{ij} es el cofactor del elemento a_{ij} de la matriz A que se muestra en la ecuación 15, y se muestra el calculo en la ecuación 16 y de forma compacta en la ecuación 17.

$$A = \begin{bmatrix} 1 & 0 & 2 \\ -4 & 3 & 1 \\ 0 & 2 & 1 \end{bmatrix} \quad (15)$$

$$c_{12} = (-1)^{1+2} M_{12} = (-1)^3 (-4) = 4 \quad (16)$$

$$M_{12} = \det \begin{vmatrix} -4 & 1 \\ 0 & 1 \end{vmatrix} = -4 \quad (17)$$

Matriz de cofactores: Se define como $Cof(A) = [c_{ij}]$ donde c_{ij} es el cofactor del elemento ij de la matriz A .

Matriz de varianza y covarianza: La covarianza es uno de los índices para determinar la existencia de una relación de tipo lineal entre dos variables de naturaleza cuantitativa. Para ilustrar mejor este concepto, en nuestro caso vamos a hablar de 2 clases o vectores X, Y ,

cada una de ellas cuenta con coordenadas x, y dónde la variable x tiene relación directa con la variable y . Hasta este punto podríamos hablar de la variable peso que esta directamente relacionada con la variable altura, solo por hablar de un ejemplo.

Sistema lineal de ecuaciones: Una ecuación lineal tiene la forma de $y = mx + b$, donde m es la pendiente y b es el valor que intercepta a y , también se le conoce como pendiente-intersección. En un sistema lineal de ecuaciones un conjunto de n ecuaciones lineales de k variables pueden ser representadas por una matriz de ecuaciones, $Ax = b$, donde A es la matriz de coeficientes, x es la columna vector de variables y b es la columna vector de soluciones. Si $k < n$ quiere decir que en general el sistema no tiene solución. Si $k = n$ y la matriz A es no singular, entonces el sistema tiene una única solución de n variables como en la regla general de Cramer donde hay una única solución si A tiene una matriz inversa A^{-1} en este caso $x = A^{-1}b$ donde si $b = 0$ entonces la solución simplemente es $x = 0$. Si A no tiene matriz inversa, entonces el conjunto de soluciones es la traducción de un subespacio de dimensión menor que n o el conjunto vacío. Si dos ecuaciones son múltiplos entre sí, las soluciones son de la forma $x = A + tB$, siendo t un número real. De manera más general, si $k > n$, entonces el sistema está indeterminado.

Transformación lineal: Se denomina aplicación lineal, función lineal o transformación lineal a toda transformación cuyo dominio y contra dominio sean espacios vectoriales que cumpla la siguiente definición: Sean V y W espacios vectoriales sobre el mismo cuerpo K . Una transformación T de V en W es una transformación lineal si para todo par de vectores $u, v \in V$ y para todo escalar $k \in K$ se satisface las ecuaciones 18 y 19:

$$T(u + v) = T(u) + T(v) \quad (18)$$

$$T(ku) = kT(u) \quad (19)$$

Valores propios: Los valores propios son un conjunto especial de escalares asociados con un sistema lineal de ecuaciones que a veces también se conocen como raíces características, valores característicos, valores propios o raíces latentes. Cada valor propio se empareja con el denominado vector propio correspondiente. La descomposición de una

matriz cuadrada A en valores propios y vectores propios se conoce como descomposición propia. El algoritmo de Lanczos es un algoritmo para calcular los valores propios y vectores propios para matrices dispersas simétricas grandes. Sea A una transformación lineal representada por una matriz A . Si hay un vector $X \in \mathbb{R}^2 \neq 0$ tal que la ecuación 20, para algún escalar λ , donde λ es el valor propio de A correspondiente al vector propio X .

$$AX = \lambda X \quad (20)$$

Dada una matriz cuadrada A de $k \times k$, como se muestra en la ecuación 21

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1k} \\ a_{21} & a_{22} & \cdots & a_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ a_{k1} & a_{k2} & \cdots & a_{kk} \end{bmatrix} \quad (21)$$

Si A es matriz de $k \times k$ se dice entonces que: A es invertible, $Ax = 0$ tiene únicamente la solución trivial, la matriz A es equivalente respecto a los renglones a I_n , $Ax = b$ es consistente para toda matriz b de $n \times 1$, $\det(A) \neq 0$, A tiene un rango n , los vectores renglón de A son linealmente independientes, los vectores columna de A son linealmente independientes. Con el valor propio λ entonces los vectores propios correspondientes se muestran en la ecuación 22.

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1k} \\ a_{21} & a_{22} & \cdots & a_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ a_{k1} & a_{k2} & \cdots & a_{kk} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \end{bmatrix} = \lambda \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \end{bmatrix} \quad (22)$$

El cual es equivalente a un sistema homogéneo, mostrado en la ecuación 23 que también se puede escribir como $(A - \lambda I)X = 0$.

$$\begin{bmatrix} a_{11} - \lambda & a_{12} & \cdots & a_{1k} \\ a_{21} & a_{22} - \lambda & \cdots & a_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ a_{k1} & a_{k2} & \cdots & a_{kk} - \lambda \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (23)$$

Donde I es la matriz de identidad, por lo que las soluciones de la ecuación 24.

$$(A - \lambda I)X = 0 \quad (24)$$

Y están dadas por la ecuación 25, donde esta ecuación también se conoce como la ecuación característica de A , y el lado izquierdo se conoce como el polinomio característico.

$$\det(A - \lambda I) = 0 \quad (25)$$

Vectores propios: Los vectores propios son un conjunto especial de vectores asociados con un sistema lineal de ecuaciones como una ecuación matricial. En álgebra lineal, los vectores propios, autovectores o eigenvectores de un operador lineal son los vectores no nulos que, cuando son transformados por el operador, dan lugar a un múltiplo escalar de sí mismos, con lo que no cambian su dirección. Este escalar λ recibe el nombre valor propio, autovector, valor característico o eigenvalor. A menudo, una transformación queda completamente determinada por sus vectores propios y valores propios. Un espacio propio, autoespacio, eigenespacio o subespacio fundamental asociado al valor propio λ es el conjunto de vectores propios con un valor propio común. La descomposición de una matriz cuadrada A en valores propios y vectores propios se conoce como descomposición propia, y el hecho de que esta descomposición siempre es posible que la matriz que consiste en los vectores propios de A sea cuadrada se conoce como el propio teorema de descomposición. Teniendo un vector propio correcto como un vector de columna X_R satisface la ecuación $AX_R = \lambda_R X_R$. Donde A es una matriz, entonces decimos que $(A - \lambda_k I)X_R = 0$, lo que significa que los valores propios correctos deben tener un determinante igual a cero $\det(A - \lambda_R I) = 0$.

2.4. Reconocimiento Facial

La detección facial en tiempo real es una de las tareas más importantes en los campos de inteligencia artificial, visión por computadora y el procesamiento digital de imágenes. Existen casos donde la detección del rostros necesita ser rápida y otros casos el tiempo no es crucial. El algoritmo propuesto por (Viola et al., 2001) es un método que tiene varios enfoques, si la aplicación global está enfocada a tener un costo computacional bajo, este algoritmo ofrece esta ventaja. A pesar de que el costo computacional en el entrenamiento esta asociado al tipo de objeto que se desea detectar y el porcentaje de éxito que se desea obtener, una vez obtenidas dichas características de la detección del rostro, la detección el rostro se hace rápida y de bajo costo computacional. El método de características en cascada está compuesto de dos estados, cada estado compuesto de un clasificador. La tarea de cada clasificador es determinar si en la sub-ventana se encuentra el objeto o si tal vez podría estar el objeto, cuando la sub-ventana es clasificada como objeto no encontrado, inmediatamente se descarta del entrenamiento, sin en cambio cuando la ventana es clasificada como un posible objeto, dicha ventana pasa al siguiente estado hasta que se determina que definitivamente es el objeto que se está buscando, tal y como se describe en la Figura 9.

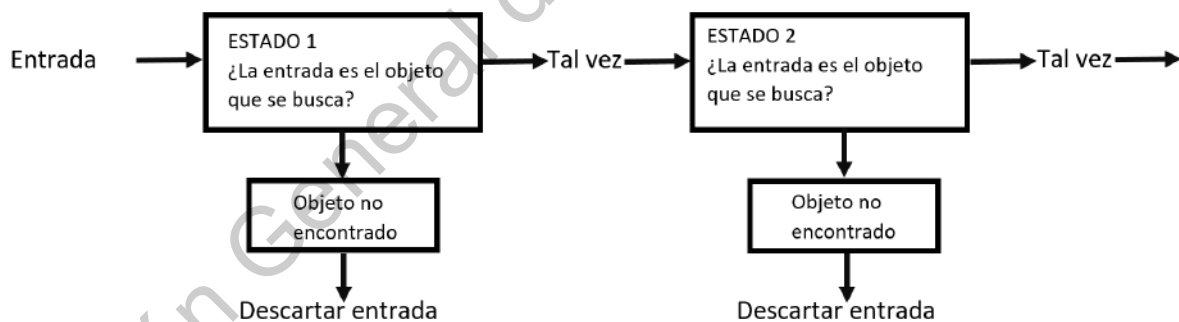


Figura 9. Estructura de los estados de clasificador Haar Cascade, (Viola et al., 2001)

El método de características basadas en clasificadores en cascada fue propuesto por (Viola et al., 2001). Este método realiza un reconocimiento de objetos. Aquí se entrena un conjunto de imágenes negativas (imágenes sin el objeto) y positivas (imágenes con el objeto), para ser reconocido en otra imagen. En definitiva, lo que se busca reconocer es un objeto, que puede ser un reloj, computadora o cualquier tipo de objeto que se quiera detectar. Al obtener el clasificador que se entrena, se extraen las características del objeto, para hacer esto, se utiliza

un kernel de convolución, cada característica es un valor que se obtiene a partir de la ecuación 26 donde la suma de los pixeles de la zona en blanco son restados de la suma de los pixeles en negro, todos los tamaños y ubicaciones de cada kernel son usados para calcular todas las características. El algoritmo de Viola propone utilizar las características Haar a modo de plantillas, tal como se muestra en la Figura 10 donde se utilizan cinco características Haar como patrones o plantillas y el cálculo de la imagen integral, tal y como se muestra en la Figura 11.

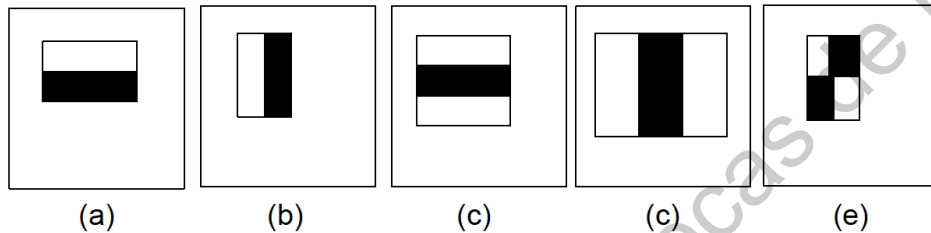


Figura 10. Ejemplo de características de rectángulo mostradas en relación con la ventana de detección adjunta.

$$\sum_{i=1}^N \sum_{j=1}^N X(i, j) 1P(i, j) blanco - \sum_{i=1}^N \sum_{j=1}^N X(i, j) 1P(i, j) negro \quad (26)$$

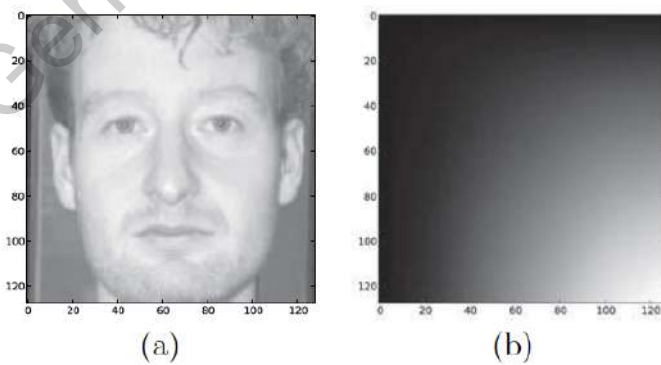


Figura 11. Antes y después del cálculo de la imagen integral donde (a) es la imagen original, (b) es la imagen integral.

2.5. Trabajos recientes

El sistema de codificación de acciones faciales, también conocido como FACS por sus siglas en inglés (*Facial Action Coding System*), fue desarrollado por (Ekman, 1977), dicho sistema realiza una medición de las EF a través de un análisis de las contracciones de los músculos del rostro, las cuales son marcadas como unidades de acción (AUs) o micro-expresiones. Este sistema determina que existe una correlación entre la intensidad de las micro-expresiones y una emoción en específico, dichos grados de intensidad se clasifican como A (ligeramente visible), B (claramente visible), C (pronunciada), D (intensa), así como E (extrema). También se clasifican los movimientos (Ms), códigos de comportamiento (Bc) y descriptores de acción (ADs). Cada unidad de acción es marcada como AU1, AU2, ..., A27. Estas AUs representan una micro expresión en específico tal y como se muestra en la Figura 12.

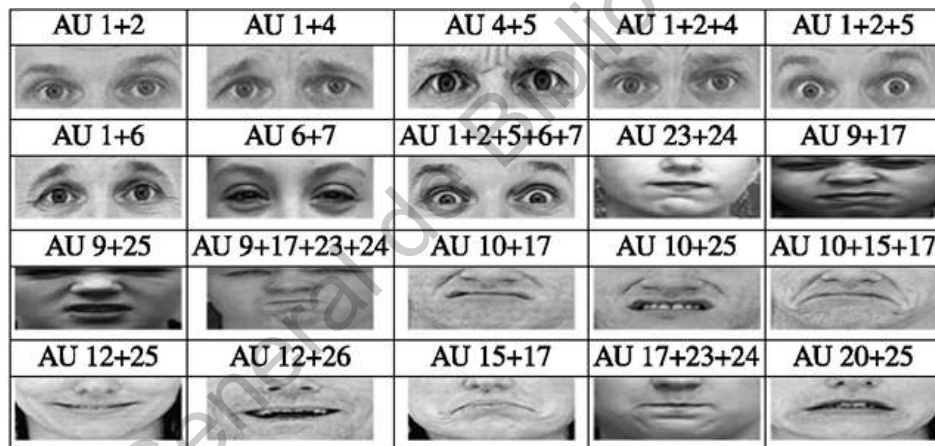


Figura 12. Ejemplos de combinaciones de unidades de acción de FACS (Ekman, 1977)

En este artículo (Shah et al., 2017) se muestra el desarrollo de un marco de trabajo, para la clasificación de EF, en dicho trabajo se propone una técnica que consta de dos pasos, tal y como se muestra en la Figura 13. El primer paso es la extracción de características y la segunda de clasificación. A partir de una imagen de entrada se localizan ciertos puntos de interés, dividiendo el rostro en siete triángulos como puntos de apoyo, para posteriormente ser utilizados como descriptores base. Para la extracción de características locales y globales, se realiza una transformación de Fourier combinado con el algoritmo LDA obteniendo los

discriminantes lineales. En la etapa de clasificación se utiliza SVM aplicando un *kernel* propio. Este marco de trabajo fue evaluado con las bases de datos, JAFFE, Cohn–Kanade (CK) y FER-2013 donde se obtuvieron diferentes mediciones de exactitud en cada una de las seis emociones evaluadas así como de diferentes autores.

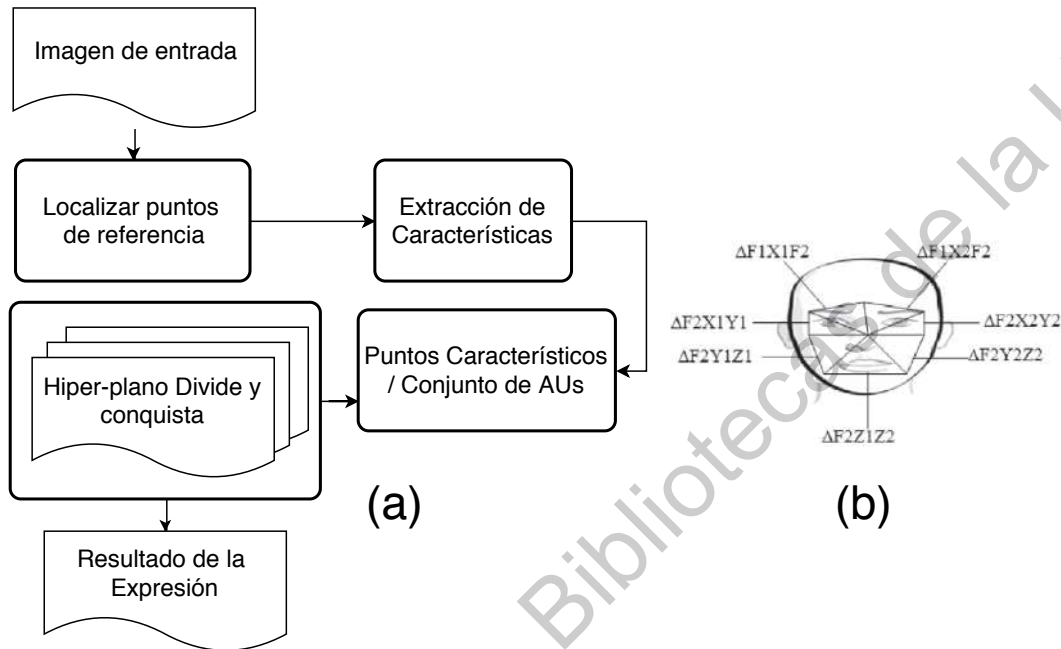


Figura 13. Diagrama de flujo del sistema (a). Siete triángulos que representan puntos de referencia en un rostro humano (b). Propuestos por (Shah et al., 2017)

En la investigación de (Bellamkonda y Gopalan, 2018) se propone un modelo para el REF, tal y como se muestra en la Figura 14, aquí se muestra una metodología que aplica PCA a la imagen recortada del rostro para reducir su dimensión, después se lleva a cabo una extracción de características con ondas de Gabor o bien con la técnica LBP, una vez hecho esto, la información se almacena en una base de datos, estos datos se utilizan posteriormente para realizar la clasificación de las expresiones con SVM y así predecir la emoción que se está experimentando en tiempo real (imagen de entrada capturada con una cámara web). En el modelo propuesto, se obtiene un porcentaje de exactitud en las bases de datos usando ondas de Gabor de 84.17 % con JAFFE, 93.00 % con MMI, y del 85.83 % con Cohn Kanade. En la base de datos JAFFE se obtuvo 88.00 % de exactitud, 88.16 % con MMI y del 96.83 % con Cohn Kanade usando técnicas de extracción de características LBP.

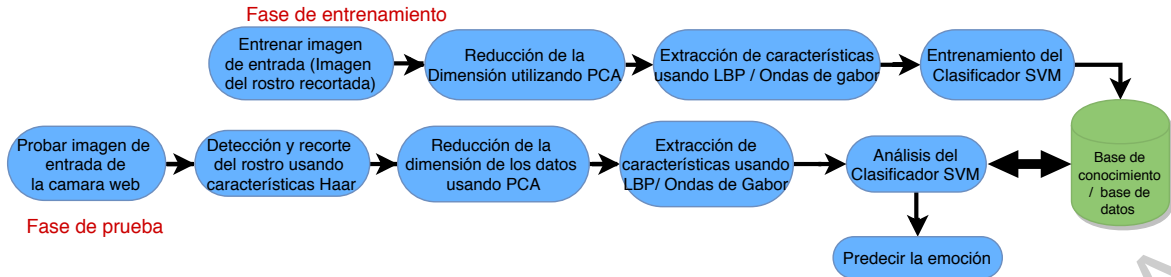


Figura 14. Arquitectura del sistema del método propuesto por (Bellamkonda y Gopalan, 2018)

En el trabajo de (Leo y Suchitra, 2018), se muestra un sistema automático de REF para rostros 3D, tal y como se muestra en la Figura 15, donde se combina la técnica de extracción de características faciales conocida como PCA, así como el algoritmo SVM, donde se implementó un clasificador binario y multiclase, con el fin de llevar a cabo la clasificación de las EF. La tarea de extracción con 3D PCA se apoyó del algoritmo MLPs, que ubica puntos en zonas del rostro, y a partir de estos puntos, se realizó una alineación extrayendo el rostro de una forma muy delimitada.

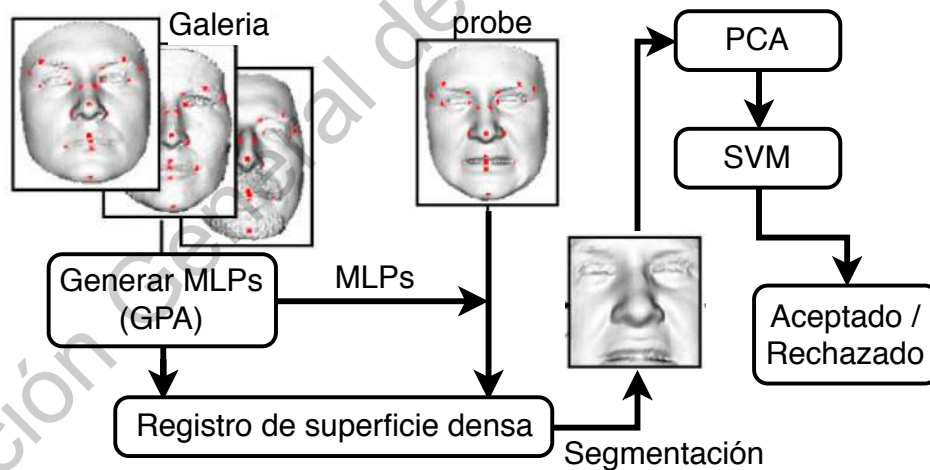


Figura 15. Marco del sistema de reconocimiento facial 3D invariante de expresión basado en SVM propuesto por (Leo y Suchitra, 2018)

En (Kar et al., 2019), se muestra un sistema automático de REF, tal y como se muestra en la Figura 16, donde se propone un esquema de tres pasos. El primer paso consiste en la extracción de características faciales. Utilizando la transformación ondulada tipo II (ripple-

II), esta implementación se hizo de forma híbrida, debido a que combina la extracción de características de apariencia así como geométricas. En el segundo paso se combina PCA y LDA con el fin de reducir la dimensionalidad de los datos debido a que el algoritmo ripplet-II devuelve coeficientes de espacio dimensional alto, quedando así un vector de 6 características, mejorando el tiempo computacional. Como tercer y último paso, se clasificaron las características, donde se utilizó el algoritmo LS-SVM, usando un *kernel* RBF, (Función de Base Radial). Las pruebas y validaciones de este sistema se llevaron a cabo con bases de datos públicas conocidas como CK+ y JAFFE y a pesar de haber obtenido una tasa de exactitud del 98.97% y 99.46% respectivamente en dichas bases de datos públicas, no se documentan resultados con imágenes que no tengan condiciones de iluminación controladas, por lo que imágenes de la vida real afectan el desempeño de dicho sistema, dando lugar a la mejora de dicho sistema, así como poder realizar un sistema que haga la estimación del estado emocional en tiempo real.

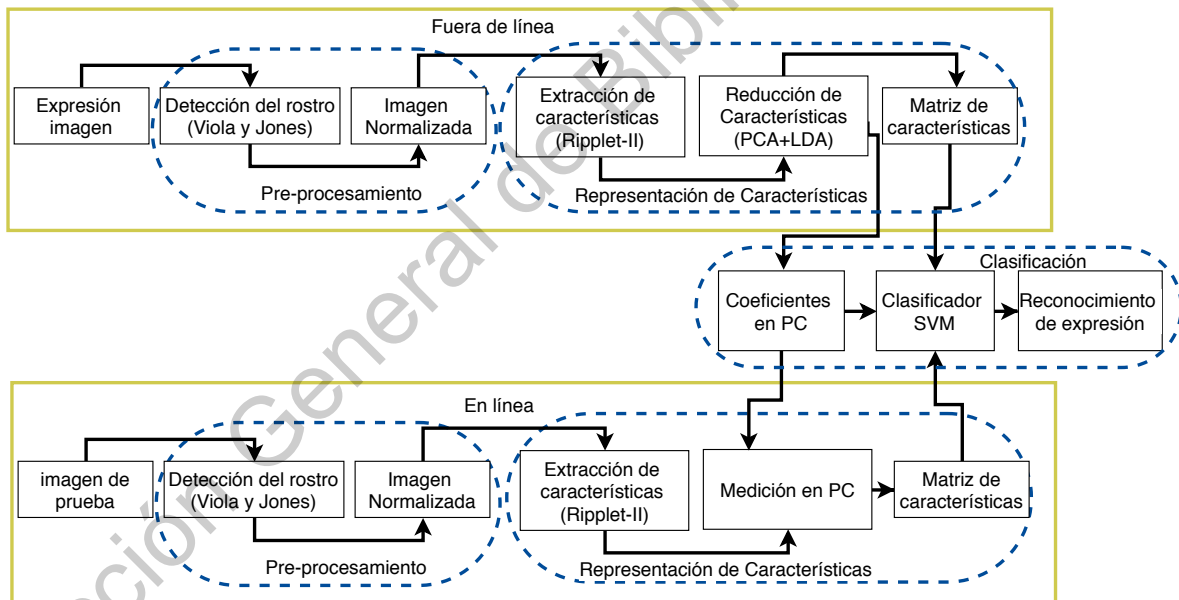


Figura 16. Diagrama de bloques del sistema de reconocimiento de expresión facial propuesto por (Kar et al., 2019)

En el trabajo presentado por (Makhmudkhujaev et al., 2019) se propone un descriptor de contorno denominado *Local Prominent Directional Pattern* (LPDP) que recupera bordes significativos correspondientes a la forma local, asegurando la codificación de la in-

formación del borde a pesar de algunas variaciones de posición y evitando bordes ruidosos. Este descriptor puede representar texturas importantes para ser utilizado en el reconocimiento de EF. LPDP analiza variaciones en la dirección de bordes para recuperar direcciones prominentes que caracterizan la forma local por lo que se garantiza una representación precisa de la forma local. El concepto principal de LPDP es capturar variaciones direccionales locales producidas por diferentes movimientos musculares faciales basados en el histograma de variaciones direccionales. En este trabajo se muestra un extenso trabajo de experimentación realizada no solo en los conjuntos de datos conocidos y desafiantes, sino también en imágenes que tienen diferentes niveles de distorsiones, el descriptor propuesto muestra un rendimiento superior que otros descriptores existentes. En la Figura 17 se muestra un ejemplo de imagen de la cara (a). Magnitudes de gradiente de la imagen de la cara (b). Acumulación de ocurrencias de gradiente de magnitud (c) . Gradiente magnitudes después del umbral (d). Para una mejor visualización del efecto de aplicar un umbral adaptativo, el color brillante representa píxeles planos donde los bordes oscuros en (d).

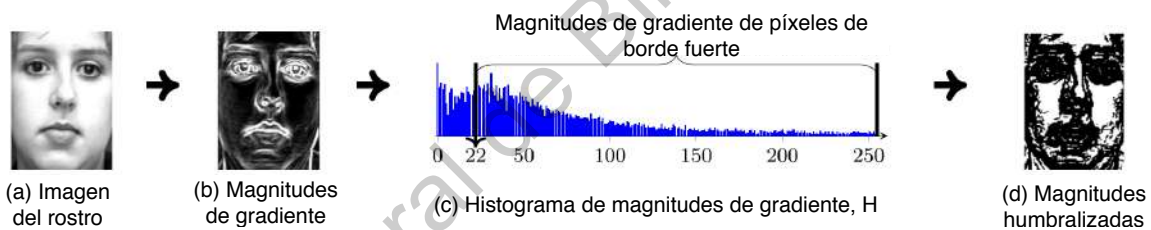


Figura 17. Pasos de cálculo del umbral adaptativo propuesto por (Makhmudkhujayev et al., 2019).

Metodología

3.1. Descripción general

En este trabajo de investigación se lleva a cabo la implementación de una metodología para un sistema de reconocimiento y clasificación de EF robusto y confiable. Tal y como se muestra en la Figura 18, la metodología consta de cinco etapas que conforman el sistema. De forma general se describe la metodología ilustrada en el listado que se muestra a continuación:

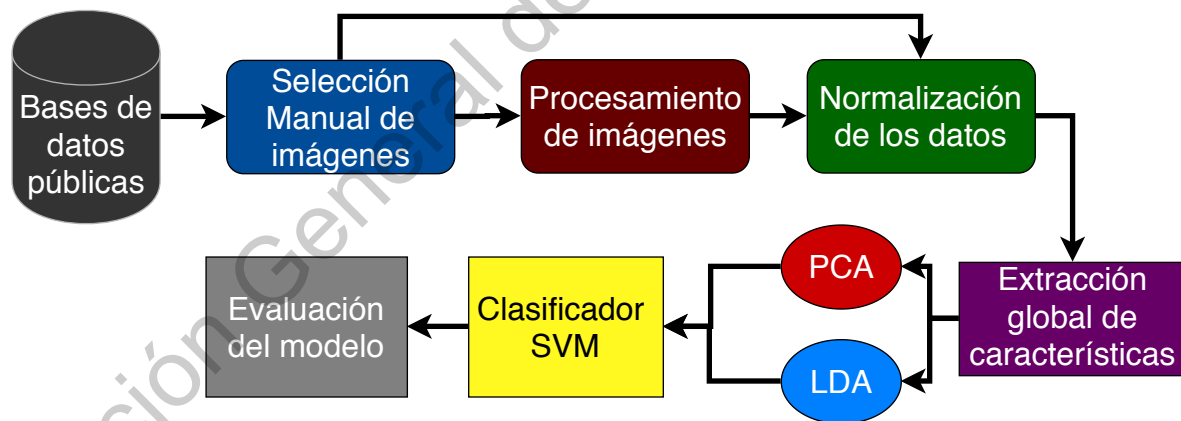


Figura 18. Metodología propuesta para el trabajo de investigación.

- La primera etapa es la adquisición de las imágenes. En esta investigación se utilizan tres bases de datos públicas, las cuales, se solicitaron a sus respectivos administradores y citadas respectivamente. En casos particulares se tuvo que elegir y discriminar algunas imágenes debido a inconsistencias con las bases de datos quedando así un nuevo

conjunto de imágenes. También se utiliza una base de datos propia siguiendo una metodología con un consentimiento informado de cada uno de los participantes para utilizar las imágenes de sus rostros con fines académicos y de investigación científica.

- En la segunda etapa se realiza un pre-procesamiento a la imagen, para utilizarla como entrada en la siguiente etapa. En este paso se detecta el rostro de la persona, se extrae su rostro, se estandarizan las imágenes a un tamaño más pequeño para reducir el tiempo computacional y se aplican distintos algoritmos de procesamiento de imágenes con el fin de mejorar y corregir la imagen.
- En la tercera etapa, se realiza una detección de EF. En este paso, se implementan de forma independiente, tres algoritmos; un algoritmo con el fin de detectar características geométricas del rostro, uno algoritmo para resaltar expresiones que tienen que ver con la apariencia y textura, así como un algoritmo híbrido que permita combinar las características de geométricas y textura.
- Como cuarta etapa se realiza una extracción de características faciales y reducción de la dimensión de los datos a fin de mejorar el tiempo computacional, en este paso se probaron los algoritmos PCA y LDA de forma independiente con imágenes con pre-procesamiento y sin pre-procesamiento.
- Como quinta etapa, el algoritmo KMEANS se usa como una función auxiliar para datos ruidosos para construir un mejor modelo SVM, luego se usan discriminantes lineales originales para la evaluación del modelo o paso de prueba.
- En la sexta etapa, se realizó la clasificación de las características faciales con el algoritmo SVM, utilizando varios tipos de *kernel* para determinar cuál permite hacer una mejor clasificación.
- En la séptima y última etapa de la metodología propuesta se evalúa el modelo con el enfoque de la validación cruzada, partiendo los datos en 10 sub-conjuntos.

3.2. Adquisición de imágenes

3.2.1 Base de datos pública *Cohn-Kanade (CK+)*

La primera adquisición fue de bases de datos públicas, la primer base de datos es (Kanade et al., 2000), conocida como *Cohn-Kanade (CK+)*, esta base de datos brinda una basta colección de imágenes secuenciales de personas mostrando una expresión en forma de pose no espontanea así como una pose espontanea, donde se parte de una expresión que va de un estado neutral hasta un exhibir una expresión facial que detone una emoción en específico, tal y como se muestra en la Figura 19 y 20.



Figura 19. Ejemplo de la secuencia de imágenes de la base de datos Cohn-Kanade (CK+).



Figura 20. Extracto de imágenes con signos de sorpresa de la base de datos publica Cohn-Kanade (CK+).

Debido a que esta base de datos tiene un número de secuencias y expresiones distintas en cada persona y cada individuo cuenta con un número diferente de EF denotando una emoción. Se decidió seleccionar de forma discrecional las imágenes que mejor representen una emoción en concreto, por lo que el procedimiento y criterios fueron los siguientes:

1) Seleccionar tres imágenes de cada individuo que muestren diferentes intensidades de una emoción, 2) No tomar en cuenta que en cada tipo de expresión facial aparezca el mismo individuo (debido a la inconsistencia de la base de datos), 3) Separar las imágenes tomando en cuenta únicamente las EF.

Tal y como se muestra en la Figura 21, se tienen separadas 66 imágenes del mismo tipo de emoción que reflejan 3 tipos de intensidades por persona, teniendo así 22 personas distintas mostrando una misma emoción con tres diferentes intensidades. Al final de haber extraído manualmente cada una de las imágenes de la clase de datos *Cohn-Kanade (CK+)* se separaron dichas imágenes en 6 carpetas más nombradas a partir de el nombre de las emociones conocidas como enojo, felicidad, miedo, neutral, sorpresa y tristeza, tal y como se muestra en la Figura 21, quedando así una nueva base de datos con un total de 396 imágenes de 640 x 490 píxeles cada una.



Figura 21. Carpetas donde se almacenan las imágenes que forman la base de datos pública *Cohn-Kanade (CK+)*.

3.2.2 Base de datos pública JAFFE

La base de datos de (Lyons et al., 1998) contiene 213 imágenes con 7 EF, seis EF básicas más una neutral de 10 mujeres japonesas, es decir, cada mujer expresa 7 adjetivos de emoción a través de una expresión facial. Originalmente las imágenes fueron tomadas en el departamento de psicología de la universidad de Kyushu, Japón. Cada carpeta almacena a una persona diferente con todas las expresiones faciales mencionadas. En este trabajo se extrajeron manualmente las imágenes de las carpetas y se distribuyeron de igual forma que en la base de datos anterior, es decir, 6 carpetas para cada emoción a clasificar. Al ser 3 imágenes con la misma expresión, pero distinta intensidad y teniendo en cuenta que son 10 personas. Al final quedaron 180 imágenes, 30 imágenes por cada emoción. Tal y como se muestra en la Figura 22 donde se muestra el contenido de una de las carpetas, esta carpeta llamada felicidad.

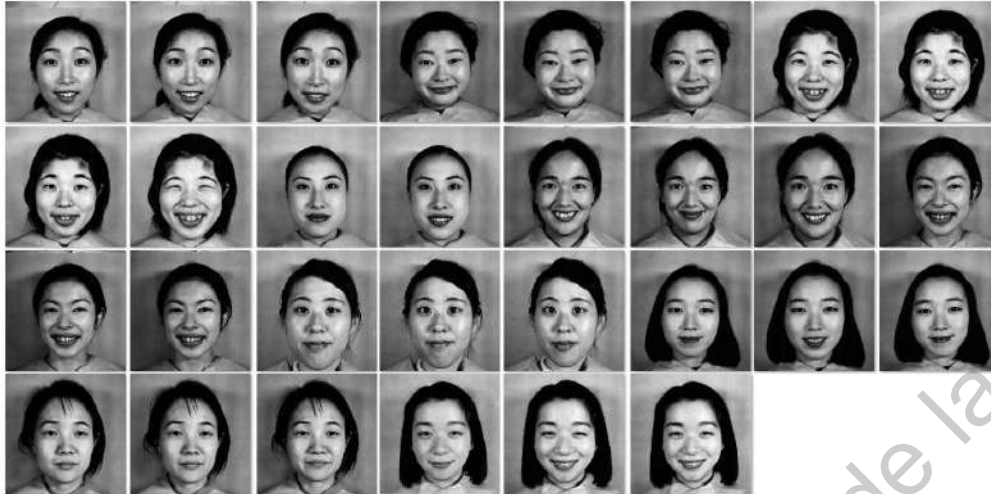


Figura 22. Muestra de imágenes con felicidad de sorpresa de la base de datos publica JAFFE.

3.2.3 Base de datos pública RAVDESS

La base de datos *Ryerson Audio-Visual Database of Emotional Speech and Song* (RAVDESS) (Livingstone y Russo, 2018), es una amplia y extensa colección de videos con 24 actores 12 hombres y 12 mujeres que vocalizan dos declaraciones en un acento de América del Norte con las siguientes frases: "*Kids are talking by the door*" y "*Dogs are sitting by the door*". La Figura 23 muestra un ejemplo de las imágenes extraídas de esta base de datos.

Cada expresión se produce a dos niveles de intensidad emocional (normal y fuerte). Se incluyen las expresiones: neutral, felicidad, tranquilidad, enojo, temor, sorpresa y disgusto. Para este trabajo de investigación se decidió seguir las consideraciones de las 2 bases de datos anteriores, es decir, se consideraron únicamente las emociones de enojo, felicidad, miedo, neutral, sorpresa y tristeza. Además de extraer manualmente las imágenes que se consideran útiles para los propósitos de extracción y clasificación de EF.

Es necesario mencionar que se extrajeron cuadro por cuadro de los videos para tener un conjunto basto de dónde poder tomar las imágenes a utilizar. La única diferencia con esta base de datos es que al ser tan extensa se decidió extraer una mayor cantidad de imágenes con más diversidad en la intensidad de las expresiones.



Figura 23. Ejemplo de imágenes seleccionadas manualmente de los videos, mostrando la expresión facial con signo de enojo(Livingstone y Russo, 2018).

3.3. Pre-procesamiento

En esta sección se explica cada uno de los pasos que tienen relación con el pre-procesamiento de las imágenes utilizadas. Este tipo de pre-procesamiento se lleva a cabo con el fin de mejorar la calidad de las imágenes, tratando problemas comunes en el reconocimiento facial y de EF, estos problemas relacionan con varios factores y que en esta sección se tratan de resolver hasta cierto punto. El primer factor tiene que ver con las condiciones de iluminación de la imagen lo que hace afecta la apariencia del rostro. Como segundo factor esta la pose del rostro donde la pose puede variar de rostro a rostro y al no ser homogéneas afectan el reconocimiento de EFs debido al funcionamiento de los algoritmos de reducción y clasificación de características. Como tercer problema esta el resaltar las EF del rostro ya que una expresión facial no es la misma en el mismo individuo y menos entre individuos.

3.3.1 Alineación del rostro

Con el fin de alinear el rostro de las personas en las imágenes se implementó un algoritmo de propósito general basado en la transformación a fin propuesta en (Hazewinkel, 2001) y así alinear el rostro a una posición vertical. Es necesario conocer el valor del ángulo

que se desea rotar la imagen, el Algoritmo 1 muestra el proceso de rotación a partir del calculo de dicho ángulo, a partir de algunos cálculos geométricos utilizando características faciales. Tal y como se muestra en la Figura 24, a partir de una imagen de entrada de una persona (A), es necesario detectar la nariz y ojos (B), después obtener el valor de las coordenadas centrales de cada ojo y nariz. Al dibujar un triángulo usando las coordenadas de la nariz y los ojos (C) es posible encontrar el punto medio de la línea que une los dos ojos, posteriormente se lleva acabo la detección de rostro (D), obteniendo las coordenadas del centro del lado superior del rectángulo del rostro, se dibuja un segundo triángulo usando las tres coordenadas anteriores. El ángulo que forma la línea que va desde el centro de la nariz hasta el centro del lado superior del rectángulo del rostro y la línea que va desde el centro de la nariz hasta el punto medio de la línea que une a los dos ojos, es el valor del ángulo que se desea rotar, obteniendo así la imagen (E).

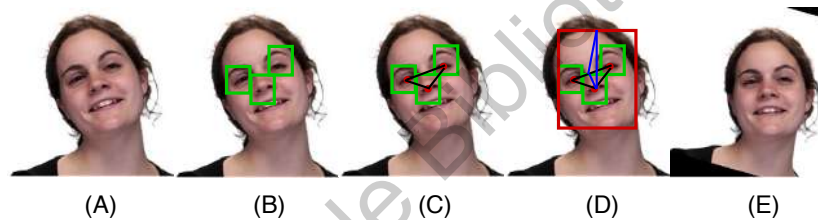


Figura 24. Ejemplo del proceso de rotar un rostro.

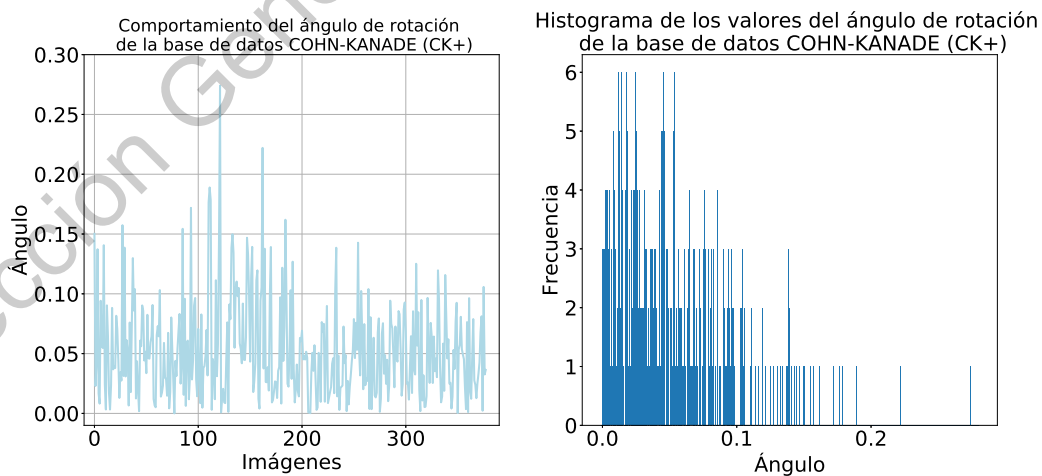


Figura 25. Ejemplo de los valores de rotación (grados) en la base de datos COHN-KANADE CK+.

Algoritmo 1 Rotar imagen I

Entrada: Imagen I de $N \times N$ píxeles

Salida: Imagen rotada img

- 1: $(E_1, E_2) \leftarrow$ Formas cuadradas que indican la detección del ojo izquierdo y derecho
 - 2: $(Ex_1, Ey_1) \leftarrow$ Coordenadas (x, y) del centro de E_1
 - 3: $(Ex_2, Ey_2) \leftarrow$ Coordenadas (x, y) del centro de E_2
 - 4: $N \leftarrow$ Forma cuadrada que indica la detección de la nariz
 - 5: $(Nx, Ny) \leftarrow$ Coordenadas (x, y) del centro de N
 - 6: $(MpEx_1, MpEy_1) \leftarrow$ Coordenadas del punto medio entre E_1 y E_2 con
 $(MpEx_1, MpEy_1) = ((Ex_1 + Ex_2)/2, (Ey_1 + Ey_2)/2)$
 - 7: $R \leftarrow$ Forma cuadrada que indica la detección del rostro
 - 8: $(Rx_1, Ex_2) \leftarrow$ Coordenadas (x, y) del punto medio de la línea superior de R
 - 9: $A \leftarrow$ Obtener el ángulo que se forma entre los segmentos que van de (Nx, Ny) a
 $(MpEx_1, MpEy_1)$
 - 10: $A \leftarrow$ Convertir el valor de A de radianes a grados con
$$A = \frac{A \times 180}{\pi}$$
 - 11: $img \leftarrow$ Rotar la imagen de entrada I , A grados.
-

3.3.2 Detección del rostro

Para la detección del rostro se utiliza un clasificador en cascada basado en características Haar propuesto por (Viola et al., 2001), se utiliza este algoritmo debido a que en este trabajo se busca tener un bajo costo computacional ya que no es la única tarea a realizar en el sistema. Dicho algoritmo tiene un enfoque basado en el aprendizaje automático donde la función en cascada se forma a partir de muchas imágenes positivas y negativas, por lo que el tiempo de entrenamiento si puede ser alto dependiendo del objeto a reconocer, en este caso rostros. El algoritmo realiza un entrenamiento de imágenes donde se escanea n veces la misma imagen, cada vez con un detector de diferente tamaño, esto con el fin de determinar la forma correcta del rostro que se busca reconocer, el enfoque de dicho algoritmo es descartar zonas de la imagen que no contengan el objeto que se busca, es decir, es más fácil descartar un posible resultado que tratar de encontrar el objeto en sí. El entrenamiento queda fuera del trabajo propuesto y no afecta en el desempeño del sistema, se utilizaron características Haar ya entrenadas para llevar a cabo la detección de los rostros, tal y como se muestra la implementación del algoritmo en una imagen de prueba en la Figura 26.

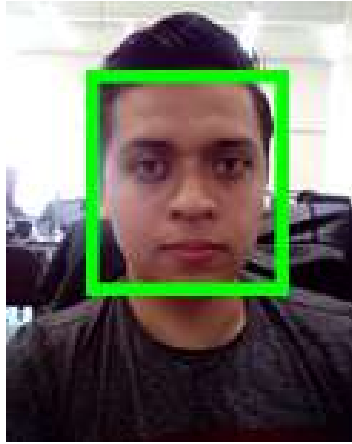


Figura 26. Ejemplo de detección del rostro con características Haar en cascada.

3.3.3 Extracción del rostro

Una vez probado el algoritmo de detección de rostro y obtenidas las imágenes en la etapa de adquisición, se procede a la extracción del rostro de cada una de las imágenes de las bases de datos. Como ejemplo la Figura 27 es una muestra de la aplicación del algoritmo 2.



Figura 27. Extracción de rostros de JAFFE con rasgos visibles de tristeza.

Detectar y extraer la zona del rostro, ayuda a eliminar aquellas zonas de la imagen que no son relevantes y no aportan nada como características del rostro que ayuden a distinguir EF. Una imagen de 2500 píxeles es una buena representación del rostro y no se pierde

información relevante que ayude o no a identificar distintas EF. A pesar de las imágenes parecen estar en escala de grises, realmente cada imagen tiene tres canales de color y al convertirla a escala de grises, se reduce considerablemente el tiempo de procesamiento al tratar cada una de las imágenes en un solo canal.

Algoritmo 2 Extracción de rostros, donde P_1 es la ruta de las imágenes de entrada y P_2 es la ruta de las imágenes de salida

Entrada: P_1, P_2

Salida: N Rostros extraídos

```
1:  $c = 1$ 
2: for each  $C$  in  $P_1$  do
3:    $P_2 = \text{update}(P_2 + \text{str}(c))$ 
4:   for each  $I$  in  $C$  do
5:     for each  $R$  in  $I$  do
6:       if  $R \neq \text{null}$  then
7:          $M = R$ 
8:          $M = \text{redim}(50, 50)$ 
9:          $M = \text{grayscale}(M)$ 
10:         $\text{imwrite}(M, P_2)$ 
11:      else
12:        continue
13:      end if
14:    end for
15:  end for
16:   $c = c + 1$ 
17: end for
```

Aquí se muestra el Algoritmo 2 que realiza esta tarea de forma automática. Las imágenes están almacenadas en el disco local en un directorio, dentro de este directorio estás las demás carpetas con las imágenes clasificadas manualmente tal y como se muestra en la Figura 28, de esta forma se leen cada una de las carpetas y en cada una de ellas las imágenes para ser procesadas por el algoritmo. Este algoritmo itera a través de cada una de las carpetas de la base de datos, y en cada carpeta procede a leer cada una de las imágenes que están almacenadas en dicha carpeta para posteriormente detectar el rostro, realizar un recorte de la zona de interés, hacer una re-dimensión de la imagen a 50x50 píxeles, convertirla a escala de grises y guardarla en la ruta de salida. Al terminar de procesar todas las imágenes de una carpeta, se actualiza la ruta de salida para guardar el siguiente conjunto de imágenes un otra

carpeta y continuar con el proceso en el resto de carpetas e imágenes. Quedando así un total de 396 imágenes de 50x50 píxeles, separadas en 6 carpetas.



Figura 28. Extracción de rostros de Cohn-Kanade (CK+) con rasgos visibles de enojo.

3.3.4 Aplicación de filtros

En esta parte de la metodología se muestran varios filtros que se aplicaron a las imágenes, tal y como se muestra como ejemplo 30 aquí se realiza una operación de tipo convolución que se refiere a una operación matemática donde se hace una copia de cada píxel de una imagen a su respectiva ubicación, pero aplicando una suma, resta, multiplicación o bien combinando las anteriores. Este proceso se utiliza para extraer o mejorar la información de las imágenes, resaltando los bordes, quitando ruido y reduciendo la distorsión. Es conveniente utilizar la convolución cuando las imágenes que esperan ser obtenidas tienen un tipo de distorsión o desenfoco. En este caso el enfoque y propósito de aplicar un filtro es que la imagen que en definitiva es una matriz donde cada píxel es una característica siga una distribución más homogénea.

El filtro promedio consiste en reemplazar cada píxel con el promedio de sí mismo y sus vecinos a partir de un "kernel". Dado que la implementación puede generalizarse como la suma de los valores de píxeles en la región multiplicada por una matriz de pesos numéricos. A este proceso también se le conoce como convolución. En la ecuación 27 se muestra el cálculo sobre una matriz cuadrada de dimensión $2m + 1$ que equivale a un número impar como 3×3 , 5×5 , 7×7 , etc. Donde la matriz de pesos ω contiene coeficientes de valor 1, estos pesos son multiplicados por el valor de los píxeles vecinos de la imagen original y posteriormente

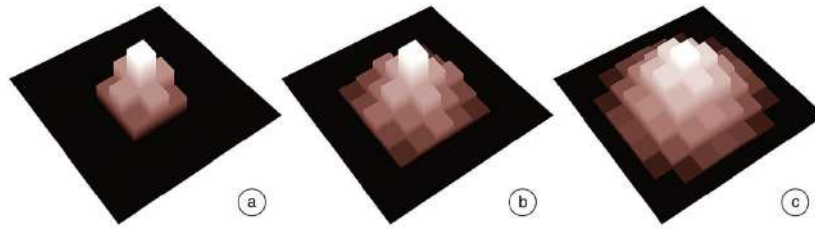


Figura 29. Resultados de aplicar un un filtro gaussiano con valores del kernel de (a) 3×3 ; (b) 5×5 ; (c) 7×7 .

divididos por la suma de tus pesos, el resultado de esta operación nos da el nuevo coeficiente que ocupara en el píxel correspondiente de la nueva imagen.

$$P'_{x,y} = \frac{\sum_{i,j=-m}^m \omega_{i,j} \cdot P_{x+i,y+j}}{\sum_{i,j=-m}^m \omega_{i,j}} \quad (27)$$

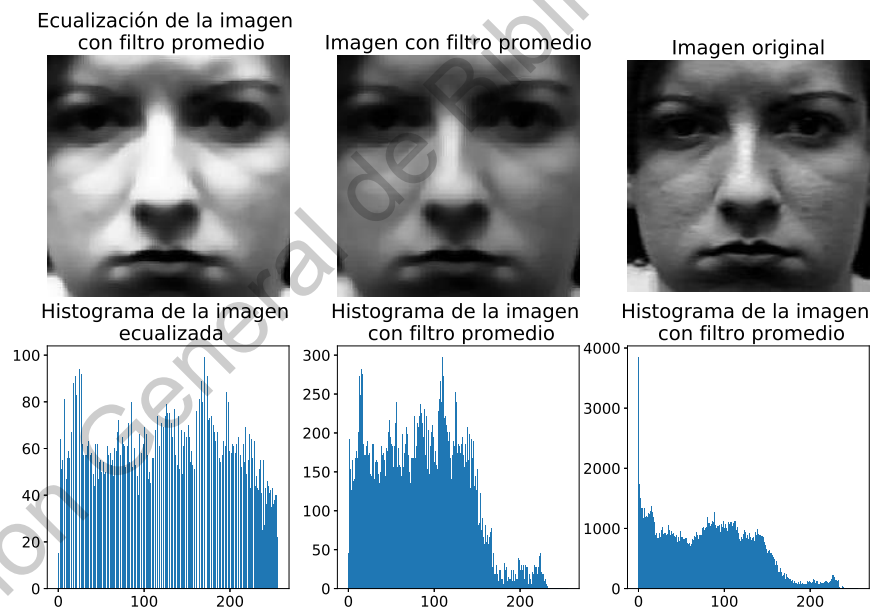


Figura 30. Ejemplo de aplicación de un filtro promedio y ecualización.

Otro filtro que se implementa en esta parte es el Gaussiano donde un conjunto de pesos que se aproxima al perfil de una función gaussiana a lo largo de cualquier fila, columna o diagonal a través del centro. Se caracteriza por una desviación estándar σ expresada en términos de dimensiones en píxeles y se calcula con la ecuación 28. Donde x y la variable y son la distancia en píxeles desde el centro del kernel. El tamaño del kernel es 3 veces la

desviación estándar a cada lado del píxel central, por lo que al agregar otra fila de valores se insertarían números insignificantes pequeños en la matriz. La desviación estándar para estos núcleos es el radio (en píxeles) que contiene el 68 por ciento de la magnitud integrada de los coeficientes, o el volumen debajo de la superficie si el núcleo se representa como un gráfico 3D de los valores de peso como se muestra en la Figura 29. Esta es una generalización de dos dimensiones de la definición familiar de desviación estándar; para una distribución gaussiana unidimensional, el 68 por ciento del área bajo la curva se encuentra dentro de ± 1 desviación estándar.

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} \times e \left(\frac{-(x^2 + y^2)}{2\sigma^2} \right) \quad (28)$$

3.3.5 Normalización de los datos

Cómo última parte del pre-procesamiento de los datos, se aplica una normalización antes de aplicar cualquier algoritmo de aprendizaje máquina, a partir de $Z = \frac{(x-u)}{s}$, donde x son las imágenes, u es el promedio de las imágenes de entrenamiento y s es la desviación estándar de las imágenes, se obtiene una matriz Z con los datos ya normalizados. Debido a que la matriz de covarianza depende de las escalas de medición de las características originales y que el método de PCA produce un sub-espacio de características que maximiza la varianza, es congruente normalizar los datos. Sin la normalización se afecta la eficacia de los algoritmos de aprendizaje máquina ya que las características individuales no se parecen a los datos estándar, distribuidos normalmente (media = 0 y varianza = 1).

3.4. Reducción de la dimensión de los datos

Las imágenes utilizadas tienen distintas dimensiones para representar el número de características. Dichas dimensiones son de 50×50 píxeles, 100×100 píxeles y de 150×150 píxeles, es decir, cada imagen tiene un total de 2500, 1000 y 22500 respectivamente. Cada matriz cuadrada es convertida a una fila con M columnas, por lo que el nuevo conjunto de datos consta de una matriz $X = \{x_1, x_2, \dots, x_N\}$ donde x_i representa i_i imagen, y N el número de ejemplos, cada imagen está representada por M píxeles, tal y como se muestra en la Figura 31.

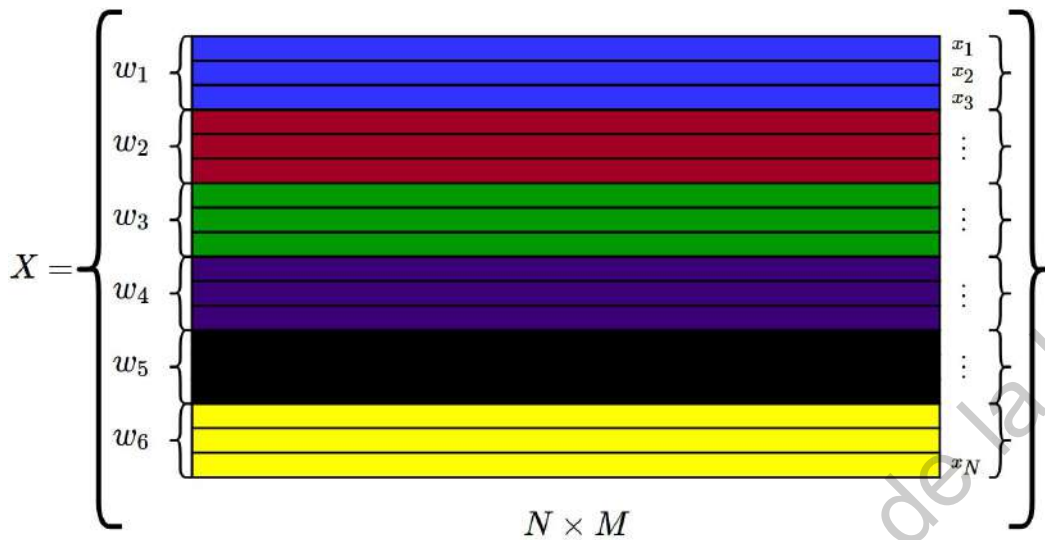


Figura 31. Representación gráfica de la matriz X

En otras palabras, cada imagen x_i se representa como un punto en un espacio M – *dimensional* que se denota como R^M . Para separar las imágenes en clases y facilitar los cálculos se añade una columna de llamada etiqueta, cuya función es la de indicar la emoción que expresa dicha imagen (1 enojo, 2 felicidad, 3 tristeza, 4 sorpresa, 5 miedo, 6 neutral), $w = [w_1, w_2, \dots, w_n]$.

3.4.1 Análisis Discriminante Lineal

La primera reducción de la dimensión de los datos fue con el Algoritmo 3. LDA toma cada columna como característica de cada muestra. En este caso con las imágenes no es posible asegurar que cada píxel de cada columna en todas las muestras represente la misma característica facial del rostro, debido a la orientación de la cara, pose, condiciones de iluminación, oclusión, etc. Los resultados de la aplicación de este algoritmo se muestran en las Figuras 36, 37 y 38

Primero se calcula una matriz de promedios μ tal y como se muestra en la Figura 32 donde queda una matriz de $1 \times M$, donde cada M_i representa el promedio de cada columna, también se muestra el cálculo de los promedios de cada una de las clases donde quedan seis matrices $\{\mu_1, \mu_2, \dots, \mu_6\}$, cada una con una dimensión de $1 \times M$.

Algoritmo 3 Análisis Discriminante Lineal

Entrada: Dada una matriz X de N ejemplos $[x_i]_{i=1}^N$ donde N es el número de filas y M el número de columnas

Salida: Nuevo sub-espacio de características Y

- 1: Calcular la matriz promedio de todos los datos ($1 \times M$)

$$\mu = \frac{1}{N} \sum_{i=1}^N x_i$$

- 2: Calcular la matriz promedio por cada clase ($1 \times M$)

$$\mu_j = \frac{1}{n_j} \sum_{x_i \in w_j} x_i$$

- 3: Calcular la matriz de dispersión entre clase ($M \times M$)

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

- 4: Calcular la matriz de dispersión dentro de la clase ($M \times M$)

$$S_W = \sum_{j=1}^c \sum_{i=1}^{n_j} (x_{ij} - \mu_j) (x_{ij} - \mu_j)^T \text{ donde } x_{ij} \text{ representa el } X^i \text{ ejemplo de la } j^i \text{ clase.}$$

- 5: Obtener los valores λ_M y vectores propios V_M a partir de

$$S_W^{-1} S_B$$

- 6: Ordenar los vectores propios en orden descendente con respecto a sus valores propios.

- 7: proyectar todos los ejemplos de la matriz X en un nuevo espacio dimensional

$$Y = X \times W$$

Promedio de cada clase

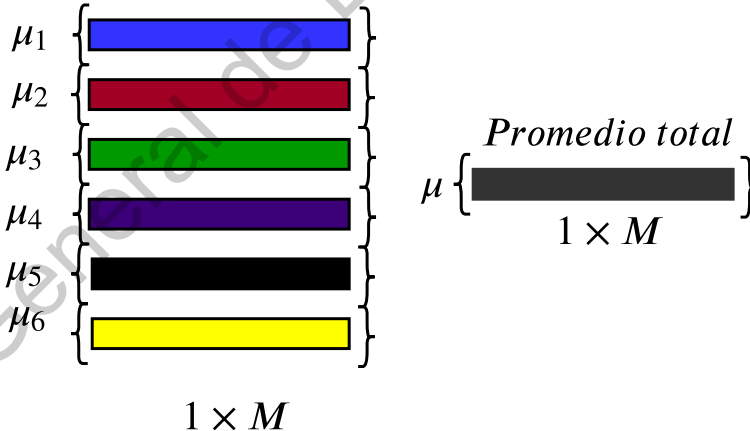


Figura 32. Representación gráfica del cálculo del promedio de cada clase y el promedio total de todas las imágenes

Al calcular la matriz de dispersión entre clases con $S_B = \sum_{i=1}^c N_i (\mu_i - \mu) (\mu_i - \mu)^T$, tal y como se muestra 33. Es posible obtener las distancias que existen entre cada una de las clases así como con el promedio de todos los datos, por lo tanto es posible decir una medida de dispersión que existe entre clases.

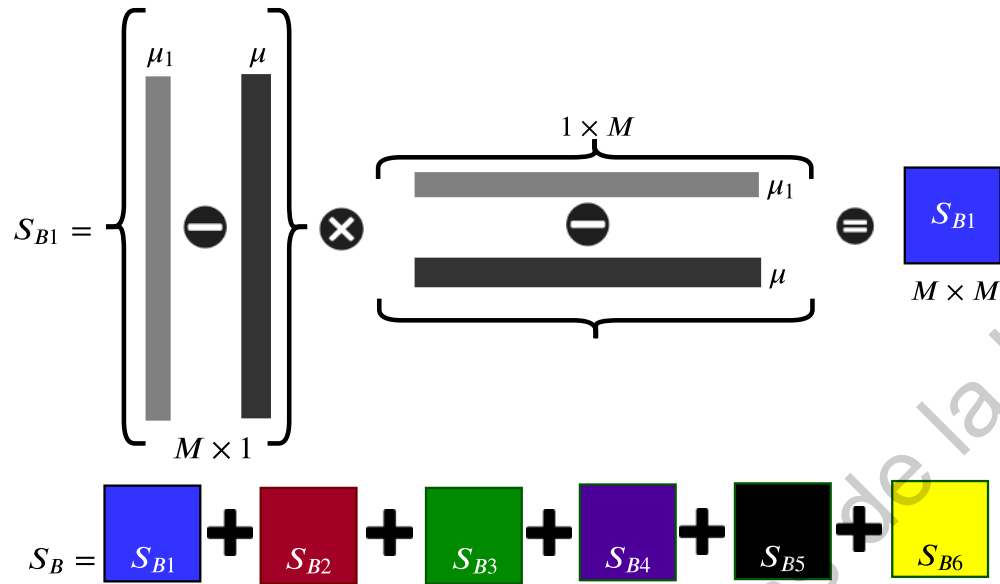


Figura 33. Representación gráfica de la matriz entre clase S_B

La varianza dentro de la clase de la i -ésima clase (S_{W_i}) representa la diferencia entre la media y las muestras de esa clase, tal y como se muestra en la Figura 34. La técnica LDA busca un espacio de menor dimensión, que se utiliza para minimizar la diferencia entre la media proyectada (μ_i) y las muestras proyectadas de cada clase.

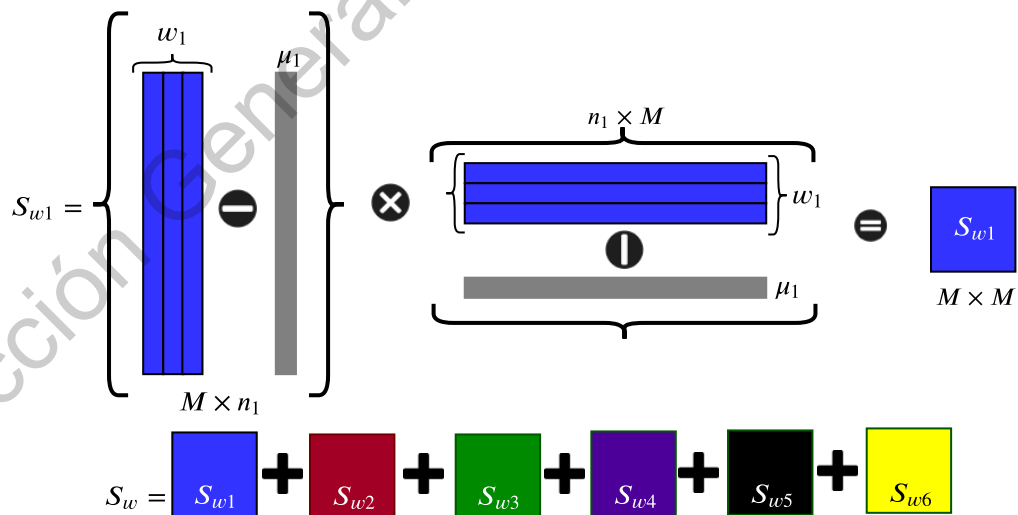


Figura 34. Representación gráfica de la matriz dentro de la clase S_w

Calcular los vectores propios y valores propios de las matrices de dispersión, ordenar

los vectores propios según el orden descendente de sus respectivos valores propios y elegir los k vectores propios con los valores propios más grandes para así formar la matriz W de $d \times k$, donde cada columna del vector W representa un vector propio, tal y como se muestra en la Figura 35. Usar la matriz W para transformar los ejemplos (datos de entrada) en un nuevo sub-espacio de características. Para conseguir esto se realiza la siguiente operación $Y = X \times W$, donde X es la matriz d -dimensional que representa el conjunto de datos (n ejemplos) y Y es la matriz transformada de $n \times d$ -dimensional en el un nuevo sub-espacio.

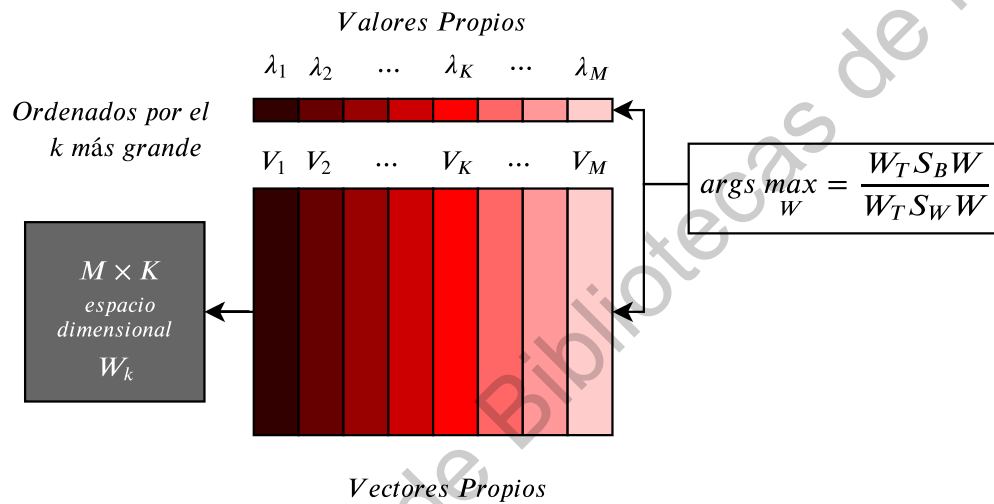


Figura 35. Representación gráfica del calculo del nuevo espacio dimensional W a partir de los vectores propios

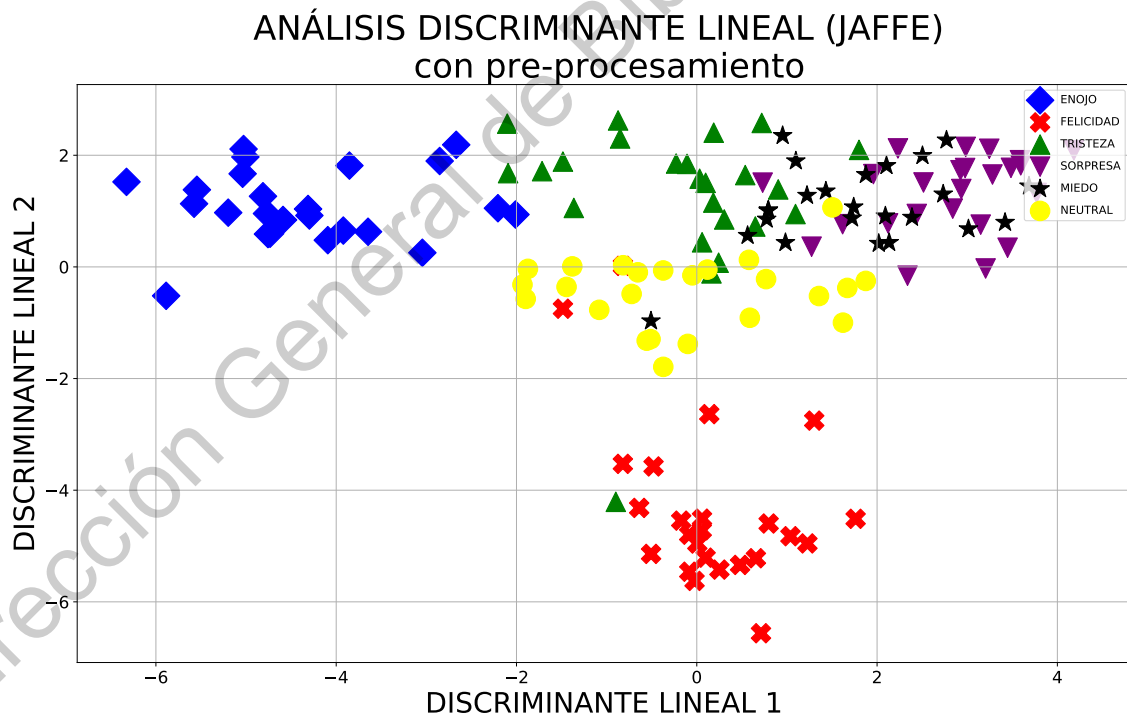
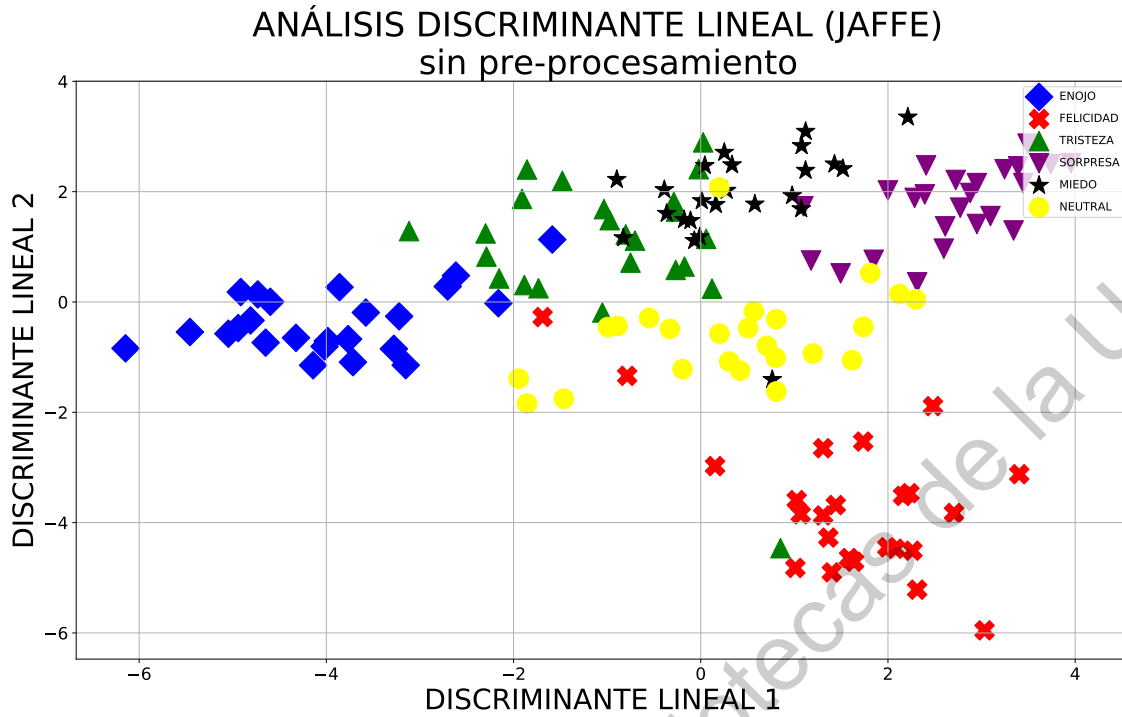


Figura 36. Nuevo sub-espacio de características proyectadas por LDA de la base de datos JAFPE (a) sin pre-procesamiento, (b) con pre-procesamiento.

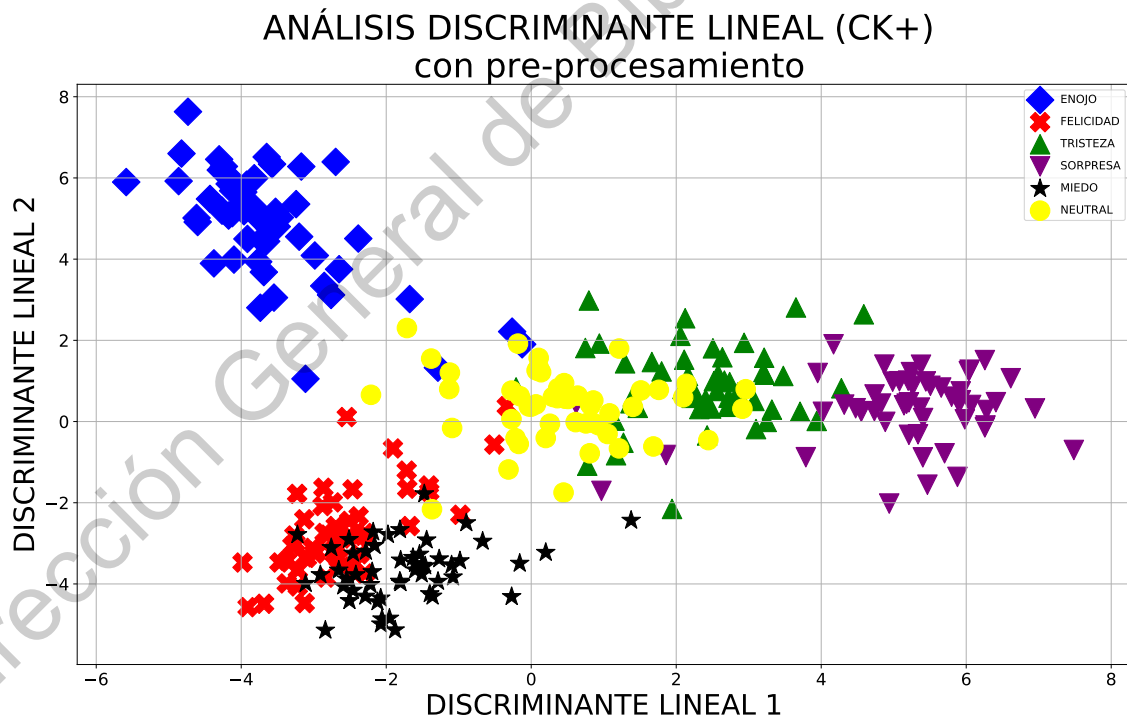
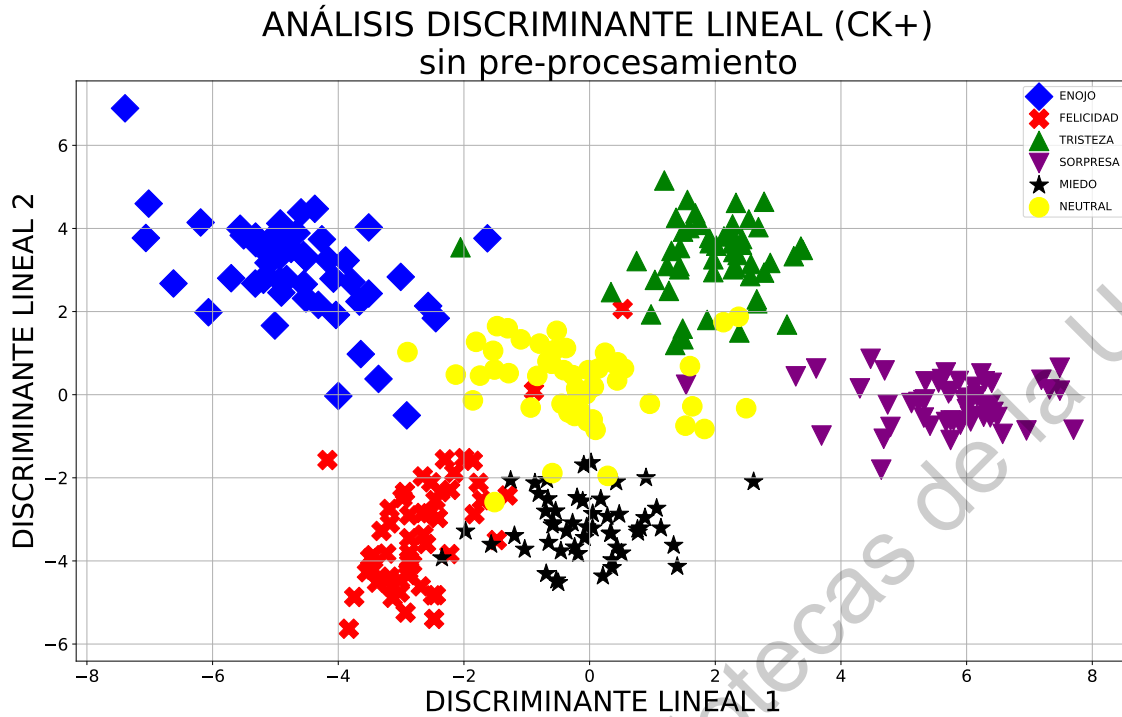


Figura 37. Nuevo sub-espacio de características proyectadas por LDA de la base de datos CK+ (a) sin pre-procesamiento, (b) con pre-procesamiento.

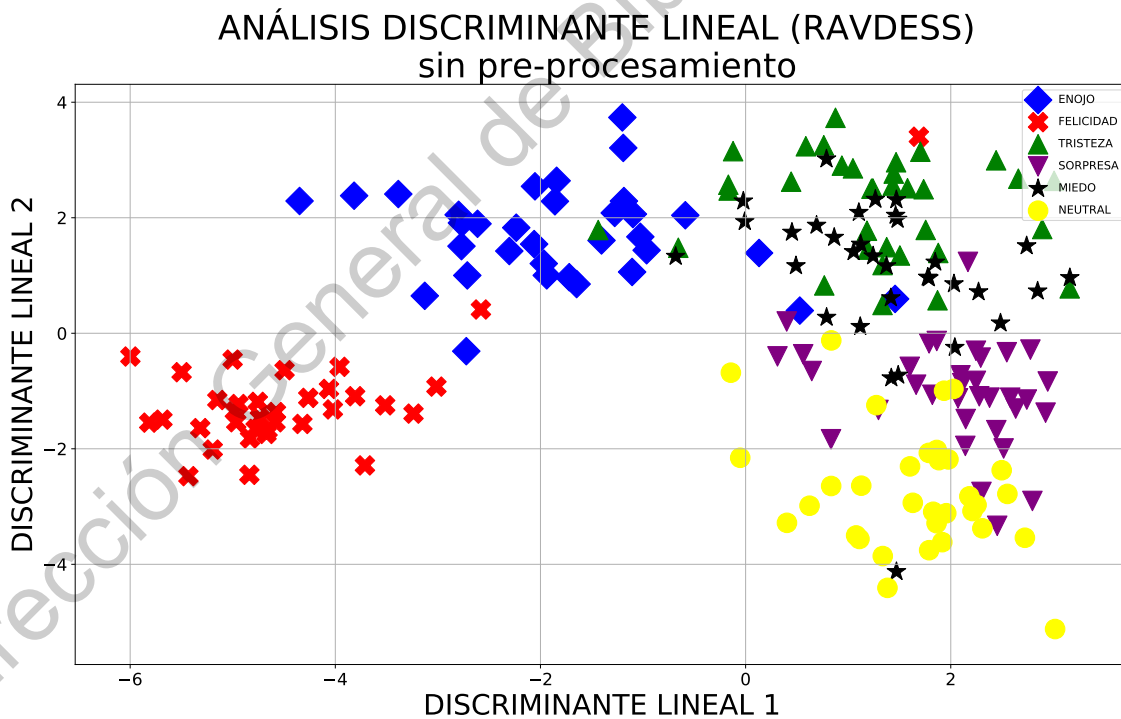
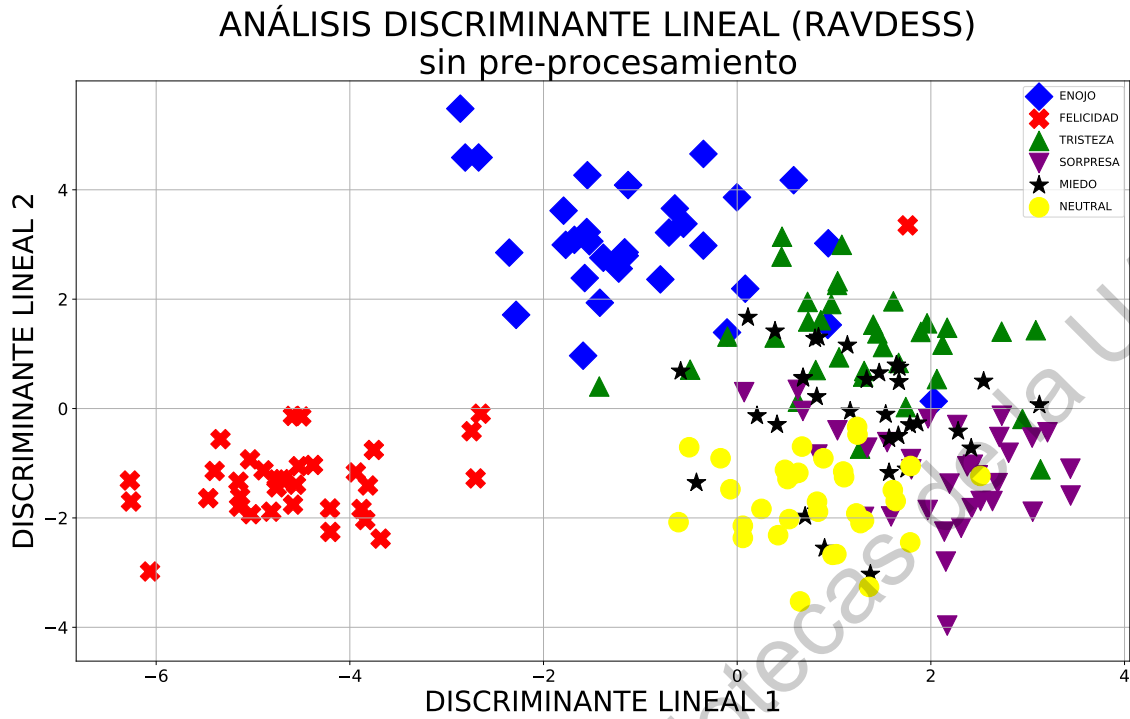


Figura 38. Nuevo sub-espacio de características proyectadas por LDA de la base de datos RAVDESS (a) sin pre-procesamiento, (b) con pre-procesamiento.

3.4.2 Análisis del Componente Principal Lineal

El Algoritmo 4 proyecta un conjunto de datos en un nuevo sub-espacio de características, donde las direcciones de los componentes principales representan la máxima varianza del conjunto de entrenamiento. El algoritmo de PCA tiene varios enfoques y el principal objetivo para este trabajo de investigación, es encontrar una relación entre las características de las imágenes (píxeles), gracias a que utiliza una matriz de covarianza, el algoritmo no toma en cuenta las clases para hacer una separación de los datos, sino que utiliza la correlación lineal que existe entre las características de las imágenes, consiguiendo obtener la máxima varianza posible de los píxeles y así encontrar las direcciones o componentes principales.

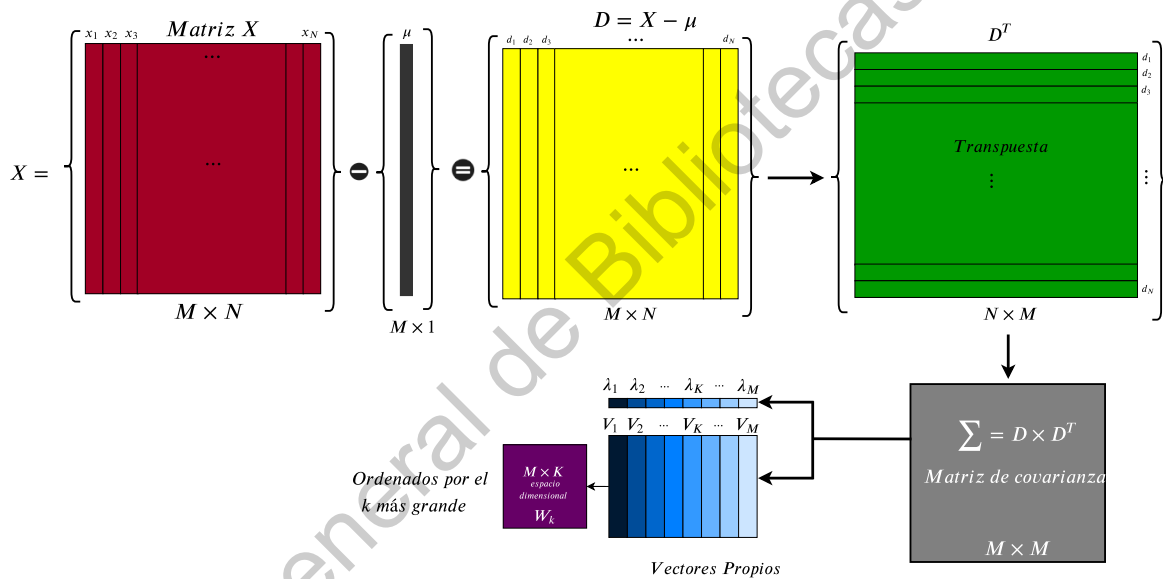


Figura 39. Representación gráfica del algoritmo del Análisis del Componente Principal

Tal y como se muestra en la Figura 39, la matriz de entrada X de N ejemplos o muestras $[x_i]_{i=1}^N$ y M es el número de píxeles. PCA busca proyectar un conjunto de datos con un espacio dimensional de R^M a un nuevo espacio dimensional de R^k donde $k < M$. La principal característica de PCA es calcular la matriz de covarianza de la matriz X , se puede decir que el cálculo de esta matriz se obtiene a partir de restar cada característica de la matriz de X con la matriz promedio de los datos μ , obteniendo así la matriz D , posteriormente multiplicar D por su transpuesta D^T . Los resultados de este algoritmo se muestran en las Figuras 41, 42 y 43

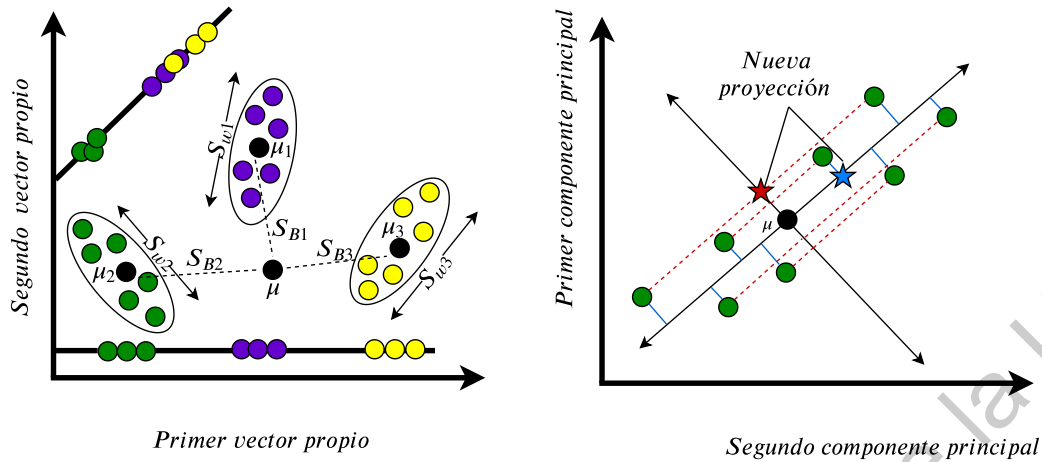


Figura 40. Comparación gráfica entre LDA y PCA

Con la matriz de covarianza Σ es posible calcular los valores y vectores propios los cuáles sirven para formar una matriz W que permitirá proyectar el conjunto de características X en un nuevo sub-espacio de características $Y = X \times W$.

3.5. Agrupamiento y clasificación de los datos

3.5.1 Máquina de Soporte Vectorial

Las Máquinas de Soporte Vectorial (SVM) son un conjunto de algoritmos supervisados de aprendizaje automático desarrollados por (Vapnik y Lerner, 1963) y actualizados en (Vapnik, 1998). Son ideales para resolver problemas de clasificación y regresión, como el reconocimiento de EF (Chen et al., 2015), (Saeed et al., 2018), (Lopes et al., 2018). Este método realiza el etiquetado y la capacitación de la clase para obtener un modelo que determina si una nueva muestra pertenece a una clase específica. La Figura 44 ilustra un problema de clasificación binaria difícil entre 2 clases en un espacio dimensional R^2 donde los puntos azules representan muestras negativas y las rojas muestras positivas. Los puntos rodeados por una circunferencia roja se conocen como vectores de soporte, y la línea negra central es la hipótesis o línea óptima que separa mejor dos clases con un margen duro. Esta hipótesis viene dada por $w^T x + b = 0$, donde w es el vector normalizado que determina la orientación de la hipótesis y b es un valor escalar llamado sesgo que controla su desplazamiento desde el origen como se muestra en la Figura. 45.

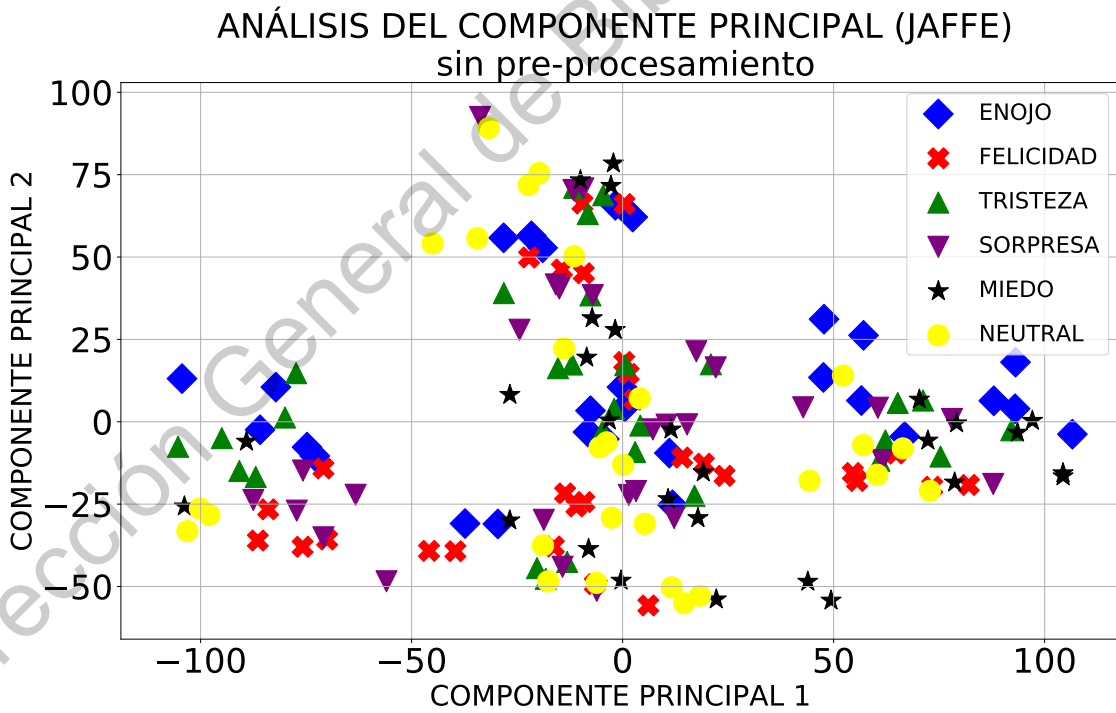
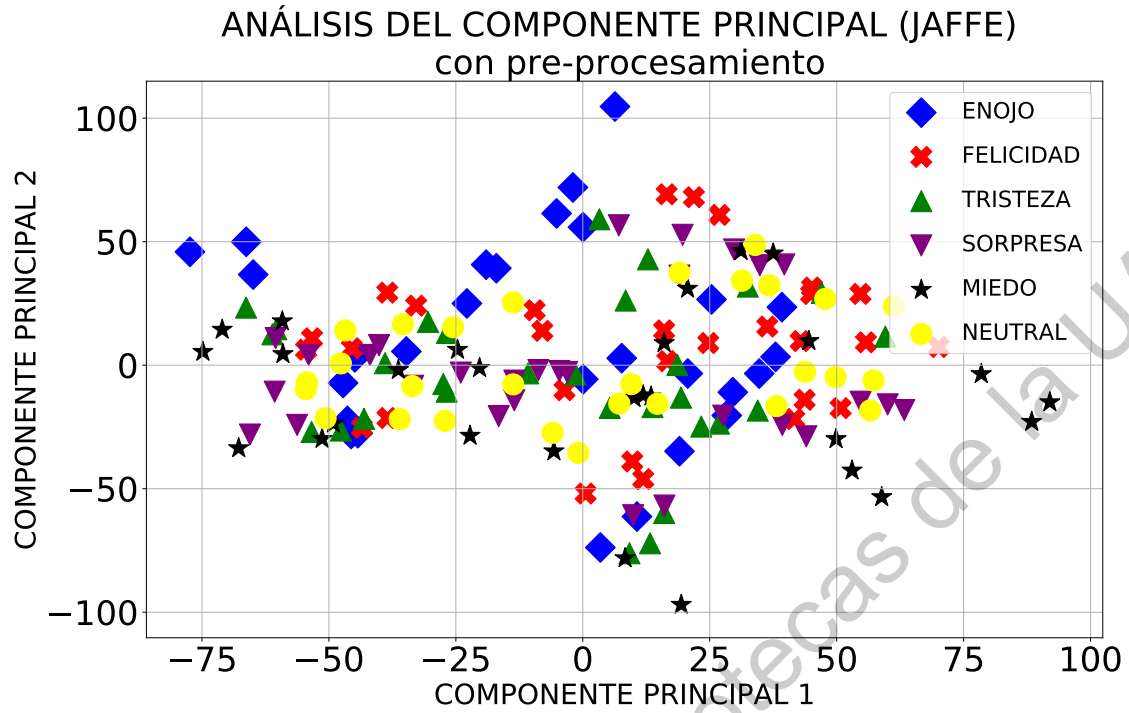


Figura 41. Nuevo sub-espacio de características proyectadas por PCA de la base de datos JAFJE (a) con pre-procesamiento, (b) sin pre-procesamiento.

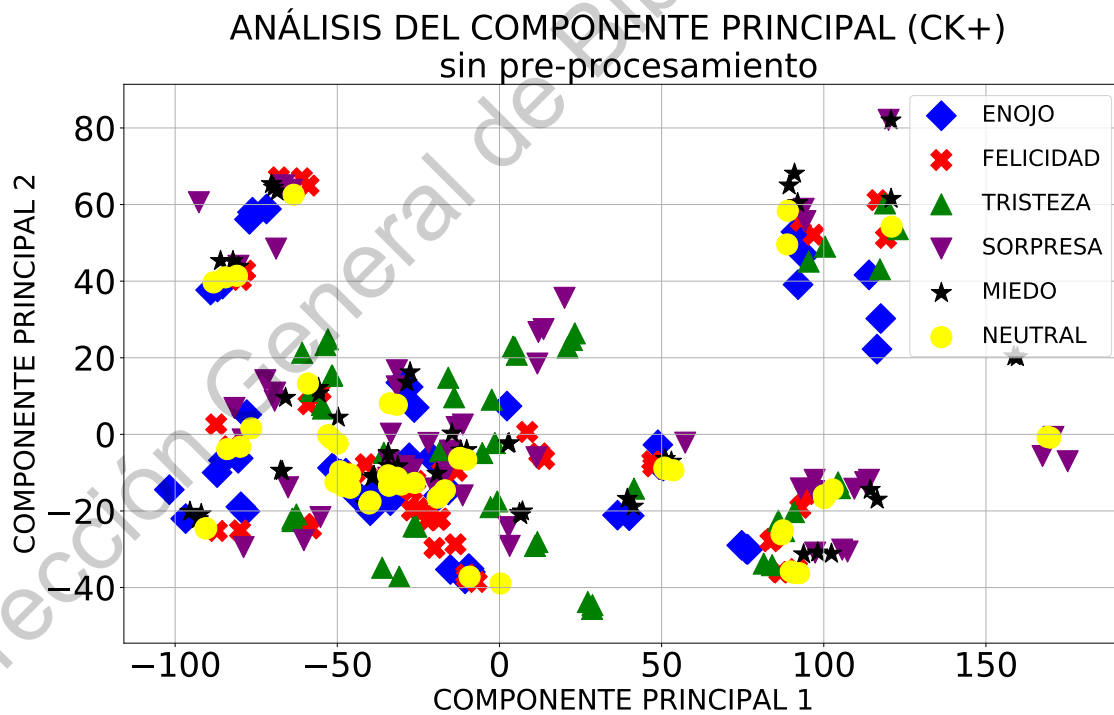
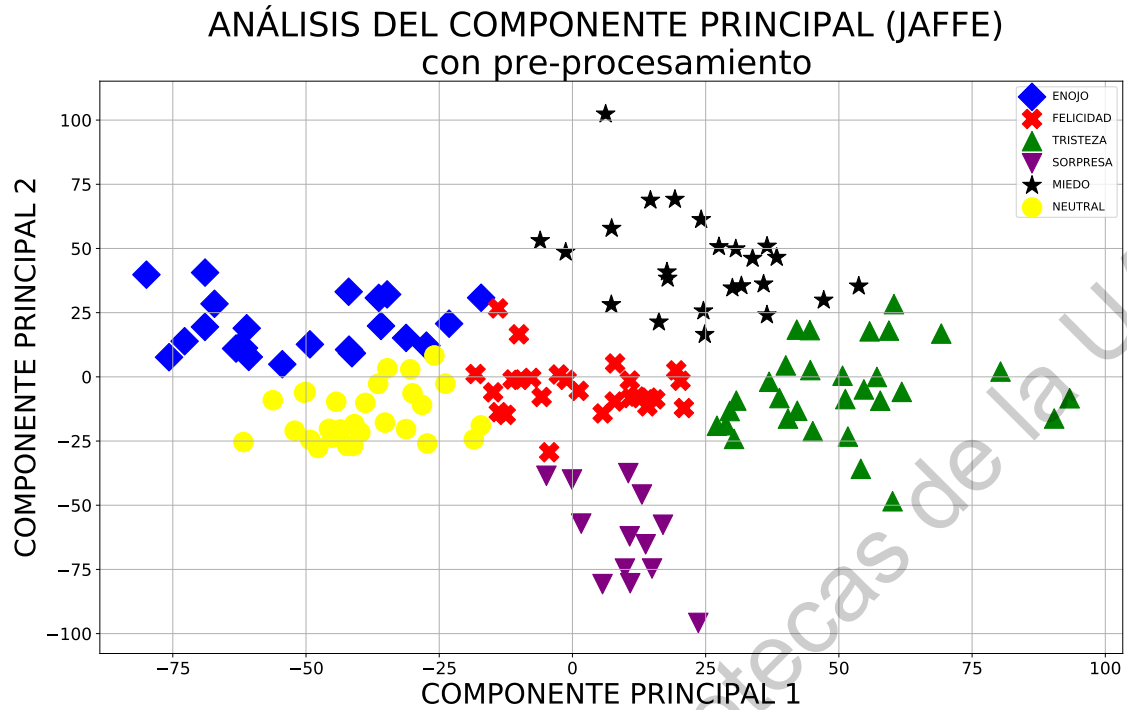


Figura 42. Nuevo sub-espacio de características proyectadas por PCA de la base de datos CK+ (a) con pre-procesamiento, (b) sin pre-procesamiento.

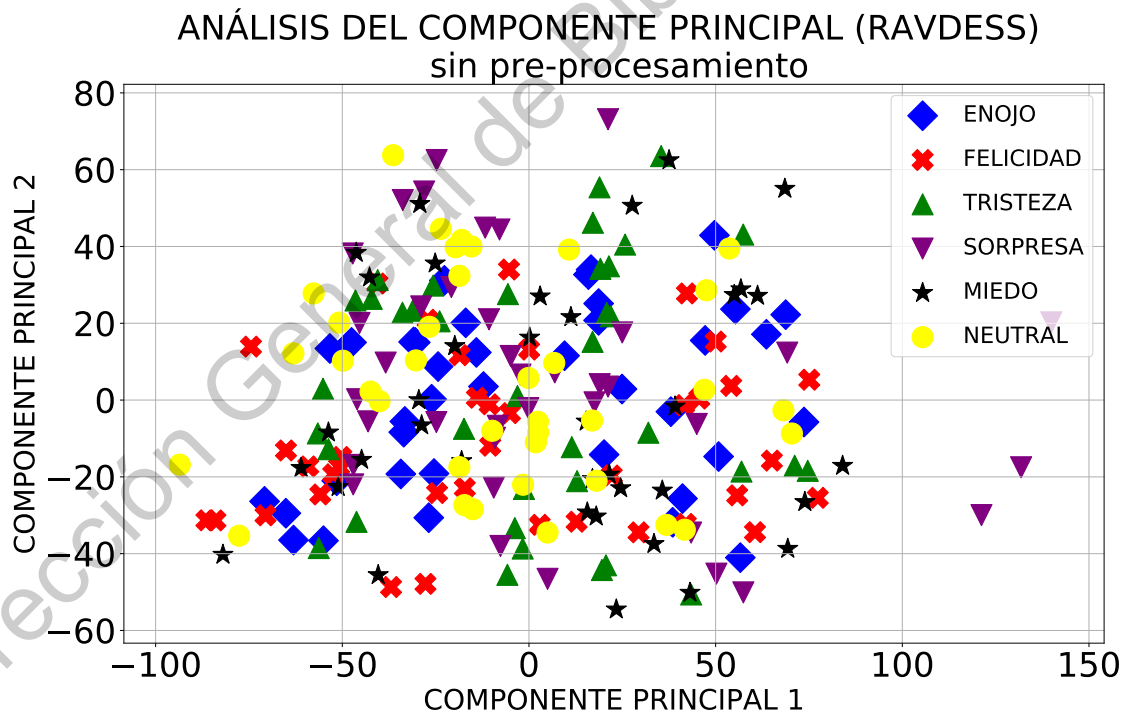
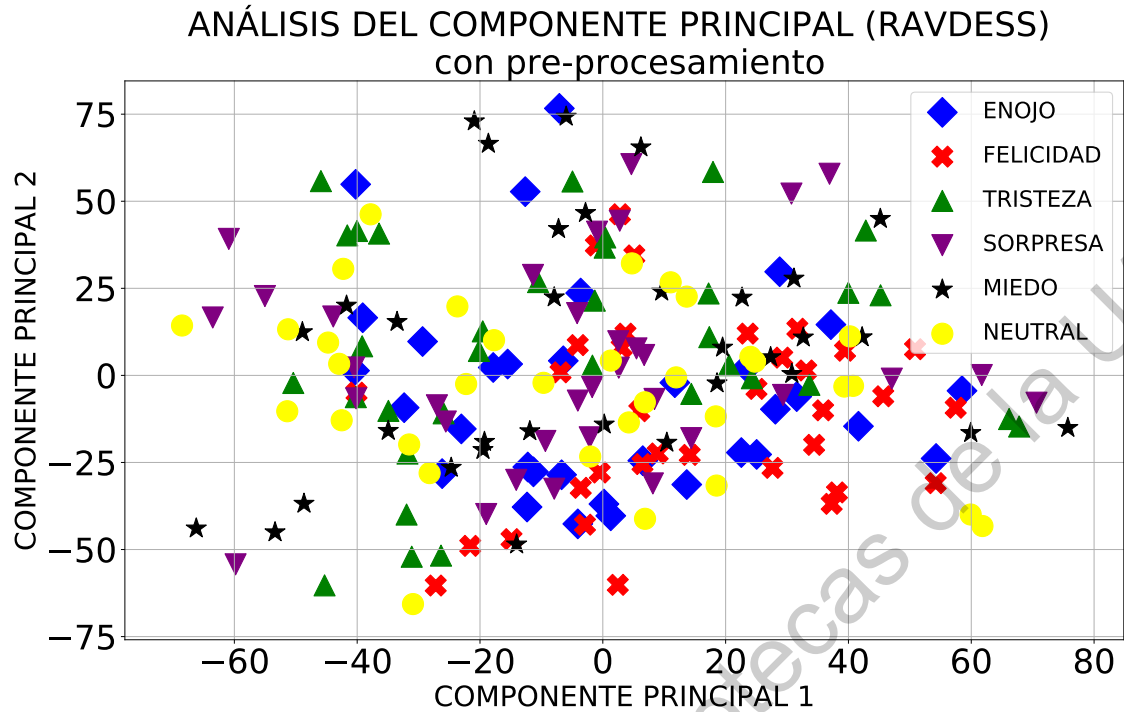


Figura 43. Nuevo sub-espacio de características proyectadas por PCA de la base de datos RAVDESS (a) con pre-procesamiento, (b) sin pre-procesamiento.

Algoritmo 4 Análisis del Componente Principal

Entrada: Dada una matriz X de N ejemplos $[x_i]_{i=1}^N$ donde N es el número de filas y M el número de columnas

Salida: Nuevo sub-espacio de características Y

1: Calcular la matriz promedio de todos los datos ($1 \times M$)

$$\mu = \frac{1}{N} \sum_{i=1}^N x_i$$

2: Restar la matriz promedio de cada uno de los ejemplos x_i

$$D = d_1, d_2, \dots, d_N = \sum_{i=1}^N x_i - \mu$$

3: Calcular la matriz de covarianza de D

$$\Sigma = \frac{1}{N-1} D \times D^T$$

4: Calcular los vectores propios V y valores propios λ de la matriz de covarianza de Σ

5: Ordenar los vectores V con relación a sus correspondientes valores propios λ ordenados de forma descendente.

6: Seleccionar los vectores propios con los valores propios más largos, $W = v_1, \dots, v_k$, los k vectores seleccionados formaran la matriz W para proyectar los datos en un nuevo espacio dimensional Y .

7: Proyectar D en un nuevo espacio dimensional utilizando W .

$$Y = W^T \times D$$

En este trabajo de investigación, hemos tratado con un problema de clasificación multi-clase con el algoritmo de SVM. Utilizando el enfoque de uno contra todos para clasificar seis EF en seis emociones generales, lo que significa que hemos realizado seis clasificadores binarios. Sin embargo, esto podría resultar en una etiqueta asignada a múltiples clases simultáneamente, así como no se puede asignar ninguna clase a un ejemplo en una región específica que tenga una predicción negativa.

Para resolver el problema anterior, se ha tomado la solución de Vladimir Vapnik. Usando la clase del clasificador para la cual el valor de la función de decisión es el máximo. Debido a que las características devueltas por los algoritmos PCA y LDA no son linealmente separables, se ha implementado otra solución Vapnik (Cortes y Vapnik, 1995) que permite algunos errores en el paso de clasificación, el objetivo de este enfoque no es cometer errores de clasificación cero, sino que sean errores sena pocos tanto como sea posible. Originalmente el algoritmo SVM clasificaba los datos linealmente separables con un hiperplano óptimo que formaba una calle o brecha calculado por $\frac{2}{\|w\|}$ resolviendo un problema de optimización, y separando clases con dos líneas con restricciones. La primera línea $wx + b = -1$ separa las muestras azules (la muestra de la clase negativa debe estar debajo del hiperplano

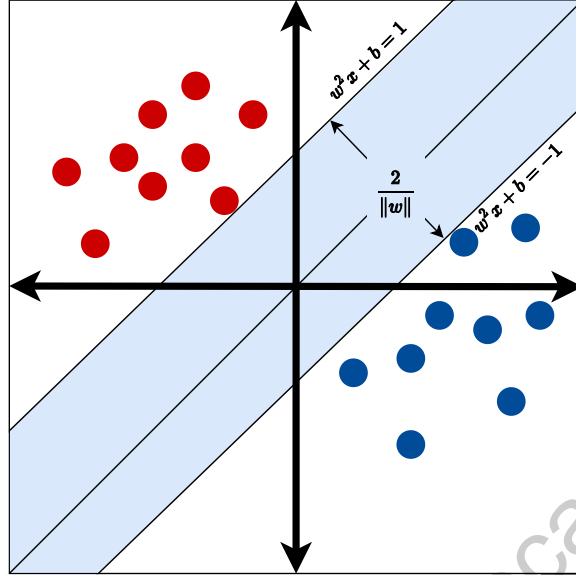


Figura 44. Ejemplo de clasificación binaria utilizando SVM.

$y_i(w x_1 + b) \leq -1$). La segunda línea $w x + b = 1$ separa las muestras rojas (la muestra de la clase positiva debe estar por encima del hiperplano $y_i(w x_1 + b) \geq 1$). Con datos ruidosos permitimos que las muestras estén dentro del margen o en el lado equivocado del hiperplano con una variable de holgura ζ que mide la distancia de una muestra positiva X_+ al hiperplano de margen positivo $w x + b = 1$, cuando X_+ está en el lado equivocado.

El objetivo de SVM es maximizar el ancho de la calle y dibujar una línea conocida como el límite de decisión, esto significa que el ancho de la calle es inversamente proporcional a la longitud del vector normalizado w , esto implica un problema de optimización, donde desea maximizar el margen, el ancho de la calle o el espacio lo más posible, mientras que la norma vectorial w se minimiza.

El ancho de la calle es inversamente proporcional a la norma del vector w . Para seguir el enfoque de margen suave, tenemos que resolver el problema de optimización de w ahora llamado regularizador, y calcular el valor escalar b . Restamos ζ_n del margen donde $\zeta_n \geq 0$. Para fomentar la clasificación correcta de las muestras, agregamos ζ_n from al objetivo

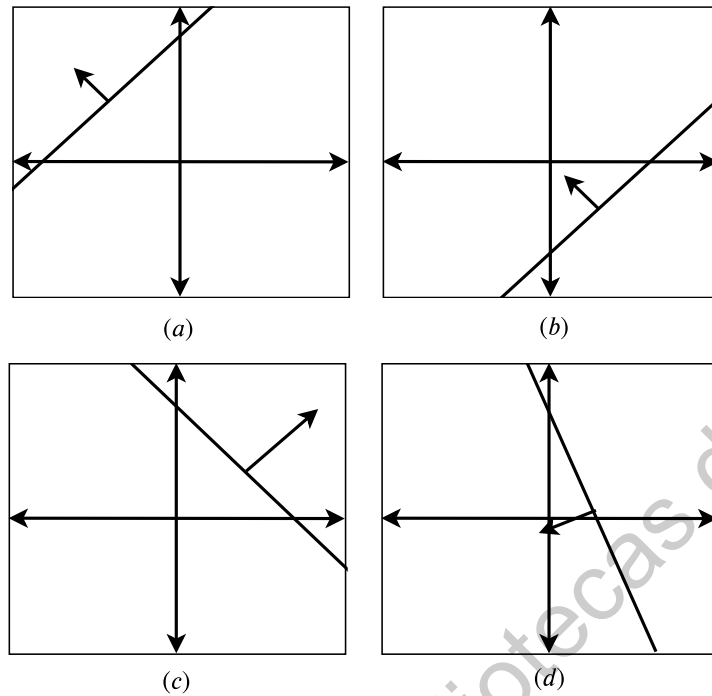


Figura 45. Interpretación geométrica de los parámetros w y b .

como se muestra en la Ecuación 29. El parámetro $C \geq 0$ intercambia el tamaño del margen y la cantidad total de holgura ese parámetro de regularización. El término de margen en la función objetivo que se muestra en la Ecuación 30 es el término de regularización donde un valor grande de C implica baja regularización, ya que un mayor peso de las variables de holgura dan más prioridad a los ejemplos que no se encuentran en el lado correcto de El límite de decisión.

$$\min_{w,b,\zeta} \frac{1}{2} \|w\|^2 + C \sum_{n=1}^N \zeta_n \quad (29)$$

$$\min_{w,b} \frac{1}{2} \|w\|^2 + C \sum_{n=1}^N \{0, 1 - y_n(\langle w, x_n \rangle + b)\} \quad (30)$$

$$y_n(\langle w, x_n \rangle + b) \geq 1 - \zeta_n \quad (31)$$

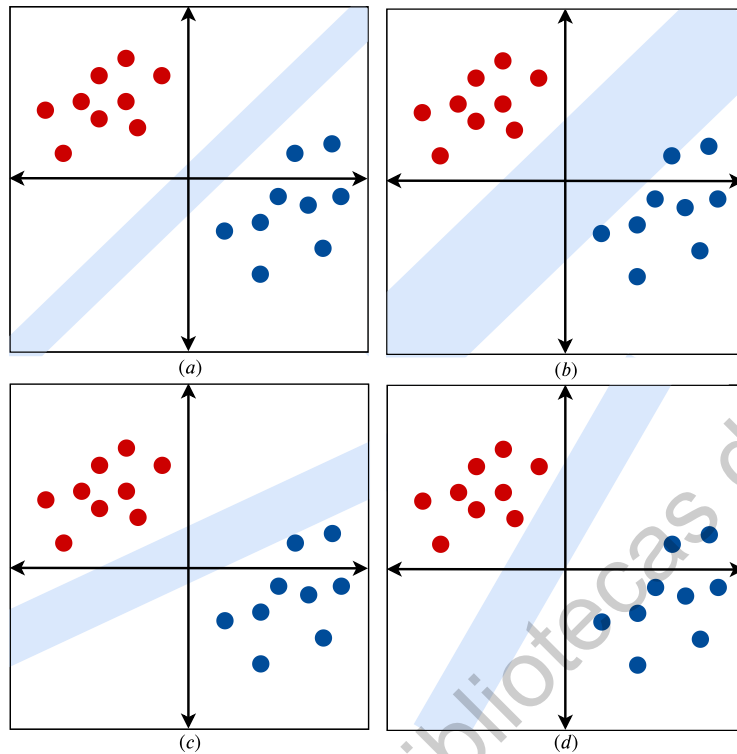


Figura 46. Ejemplo del problema de optimización de calle.

3.5.2 K-Means

El algoritmo K-Means propuesto por (Lloyd, 1957) es ampliamente conocido y consiste en dos pasos explicados en el Algoritmo 1. El objetivo principal de K-means es agrupar datos no etiquetados en k subconjuntos donde los grupos se forman a partir del valor medio del cálculo de la distancia euclidiana de todas las muestras con respecto al centroide. Cuando todos los centroides dejan de cambiar, significa que el algoritmo K-means ha convergido. En el trabajo de (Ye et al., 2008) K-Means se usa para la selección y agrupamiento simultáneo de un subespacio LDA. Se han utilizado algoritmos similares para la agrupación semi-supervisada de expresiones desconocidas con redes neuronales en (Jalal y Tariq, 2019). En el trabajo propuesto por (Wu et al., 2019) K-means y LDA son combinados para la reducción de la dimensionalidad, donde se emplea el algoritmo de agrupamiento para generar etiquetas de clase para datos no etiquetados y LDA se usa para investigar la representación de datos de baja dimensión. En este trabajo de investigación, K-means se usa para agrupar

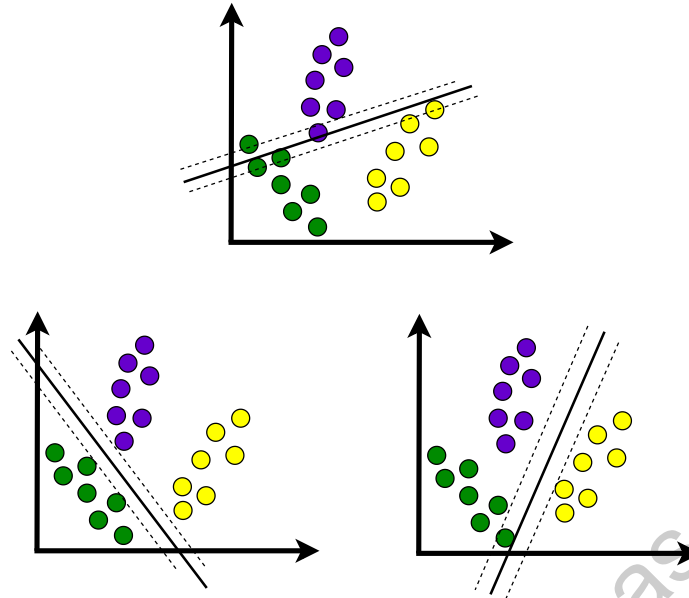


Figura 47. Ejemplo de la estrategia de clasificación multi-clase uno contra todos.

muestras ruidosas y construir un mejor modelo SVM para el problema de clasificación de expresión facial como se muestra en la Figura 55 y 56.

Algoritmo 5 K-Means Clustering

Entrada: Dado un conjunto X de N muestras y k números de clusters

Salida: Nuevo conjunto de k sub-conjuntos

- 1: Seleccionar de forma aleatoria k muestras de X como centroide de los k clusters y asignarlos a *centroids*
 - 2: **while** *centroids* no cambien **do**
 - 3: De acuerdo con el centroide del k cluster, asignar cada muestra al centroide más cercano k .
 - 4: Calcular los centroides de nuevo
 - 5: **end while**
-

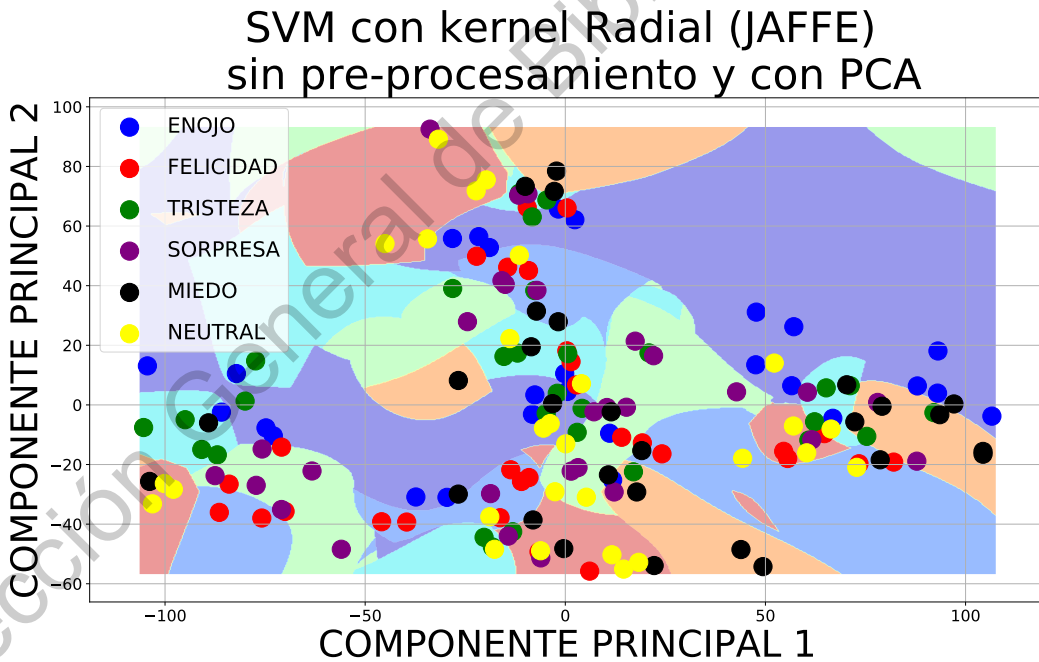
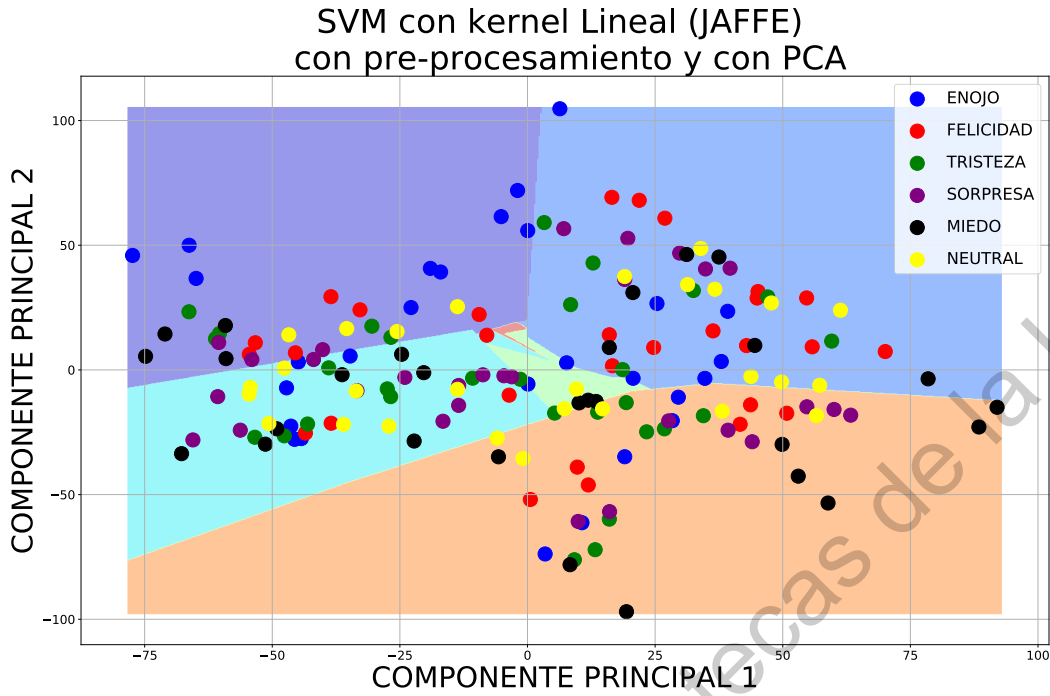


Figura 48. SVM con kernel Lineal y Radial utilizando el nuevo sub-espacio de características de PCA en la base de datos JAFPE (a) con pre-procesamiento, (b) sin pre-procesamiento.

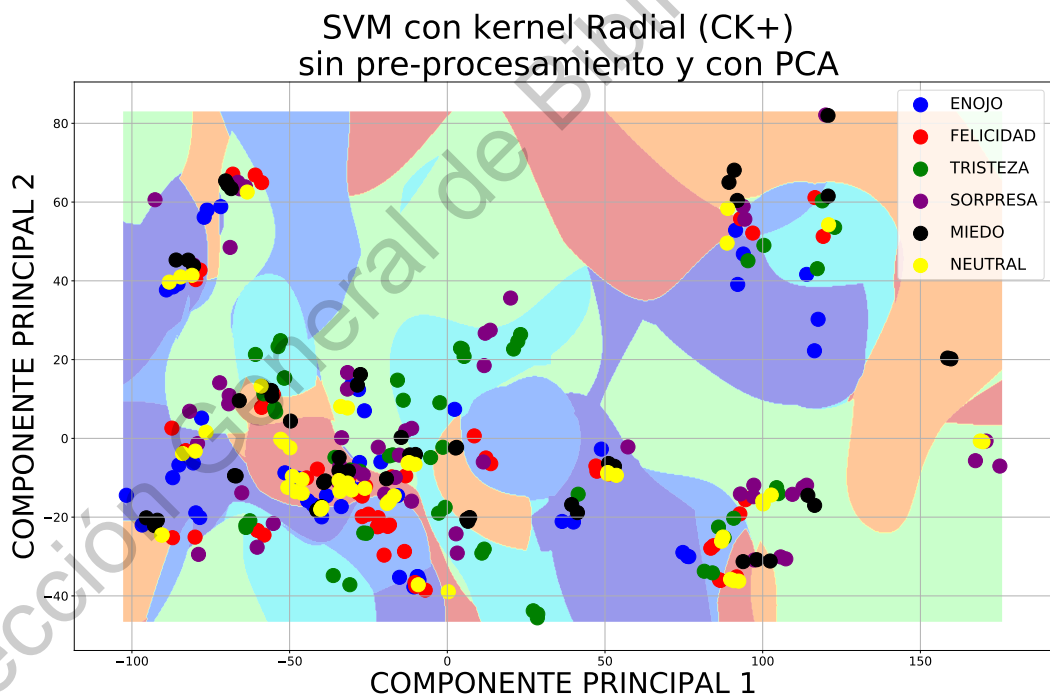
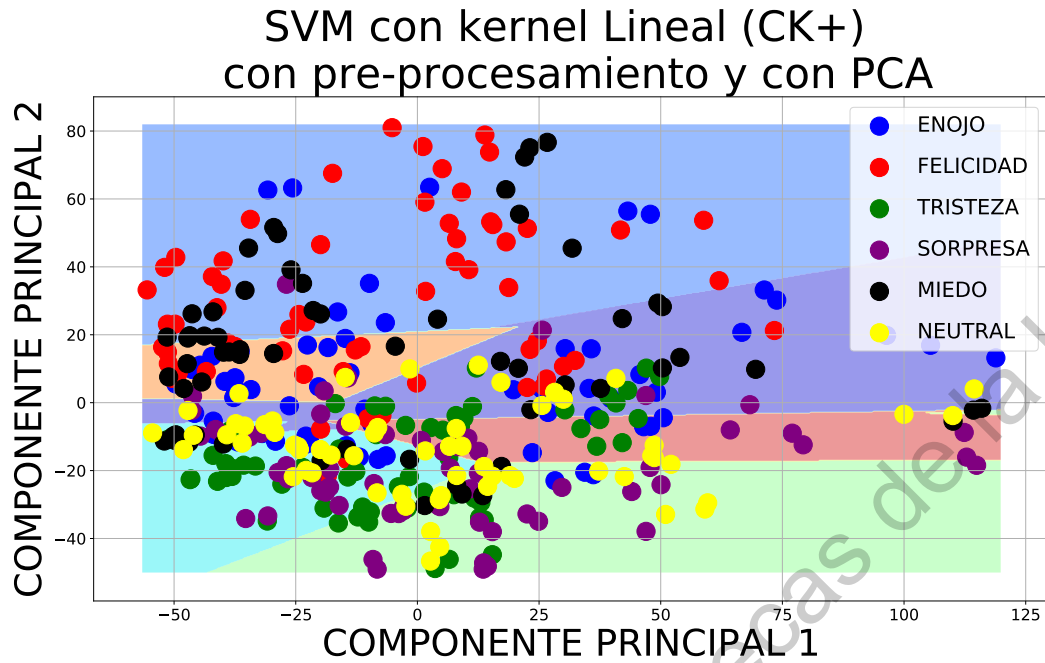


Figura 49. SVM con kernel Lineal y Radial utilizando el nuevo sub-espacio de características de PCA en la base de datos JAFFE (a) con pre-procesamiento, (b) sin pre-procesamiento.

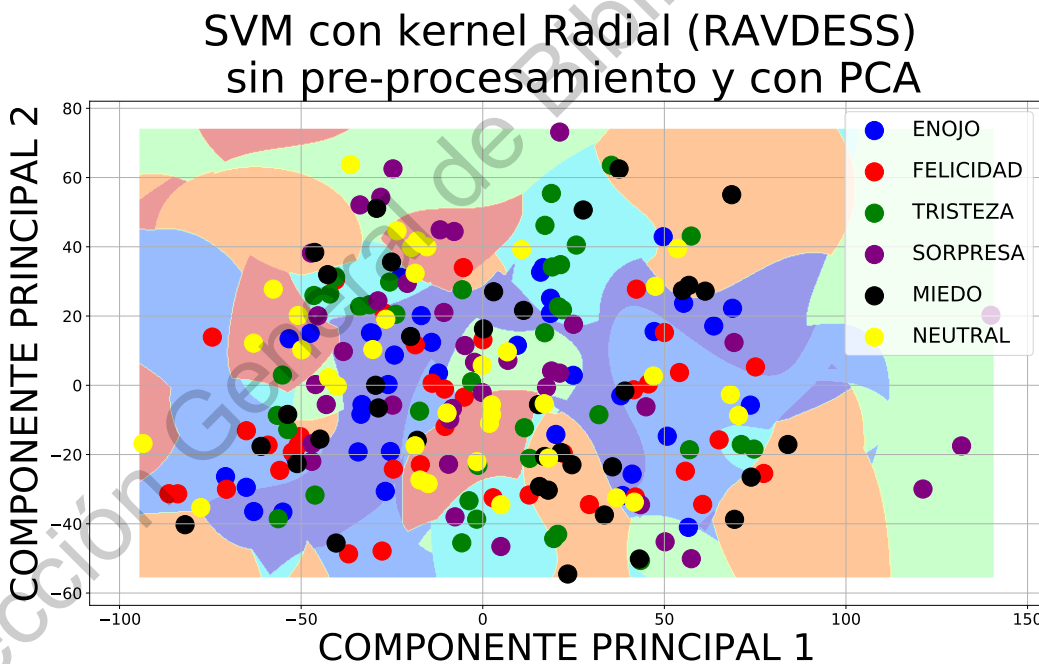
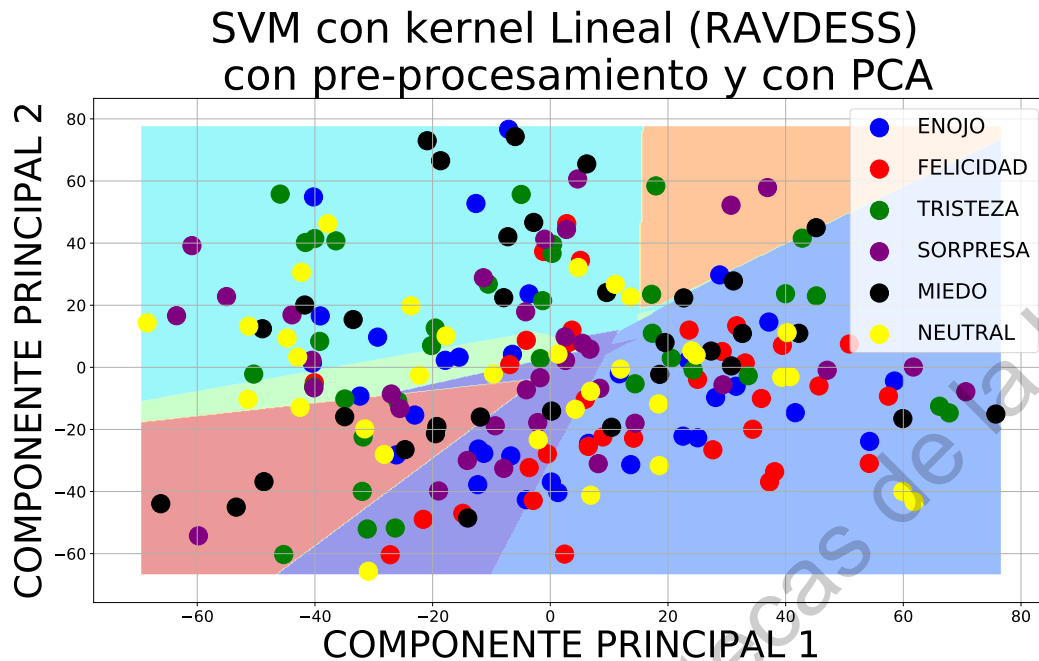


Figura 50. SVM con kernel Lineal y Radial utilizando el nuevo sub-espacio de características de PCA en la base de datos RAVDESS (a) con pre-procesamiento, (b) sin pre-procesamiento.

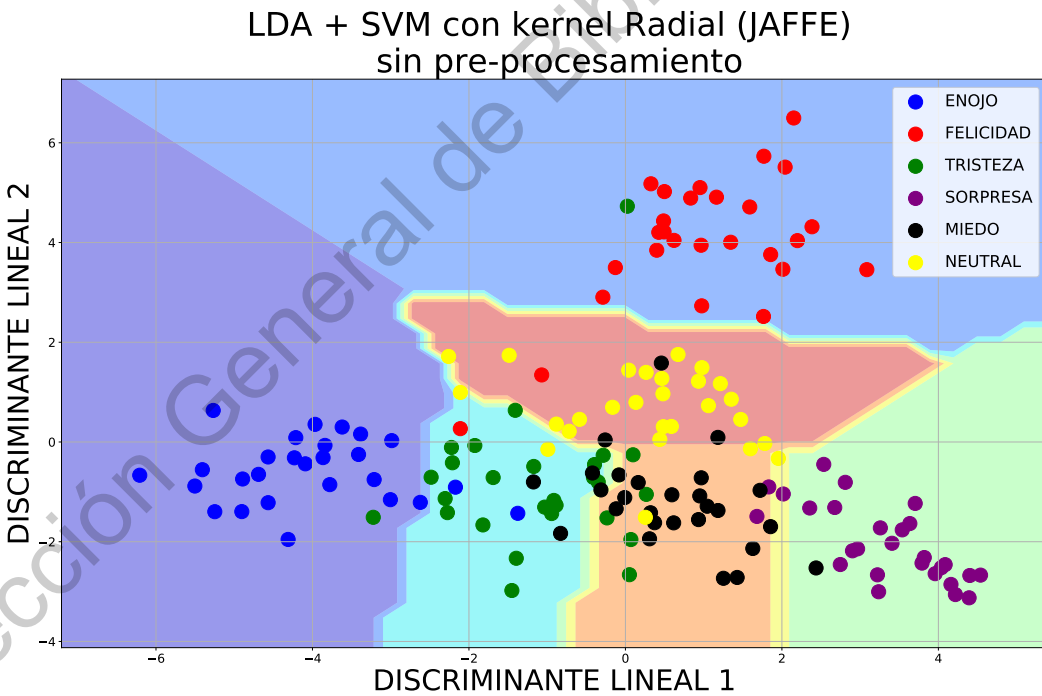
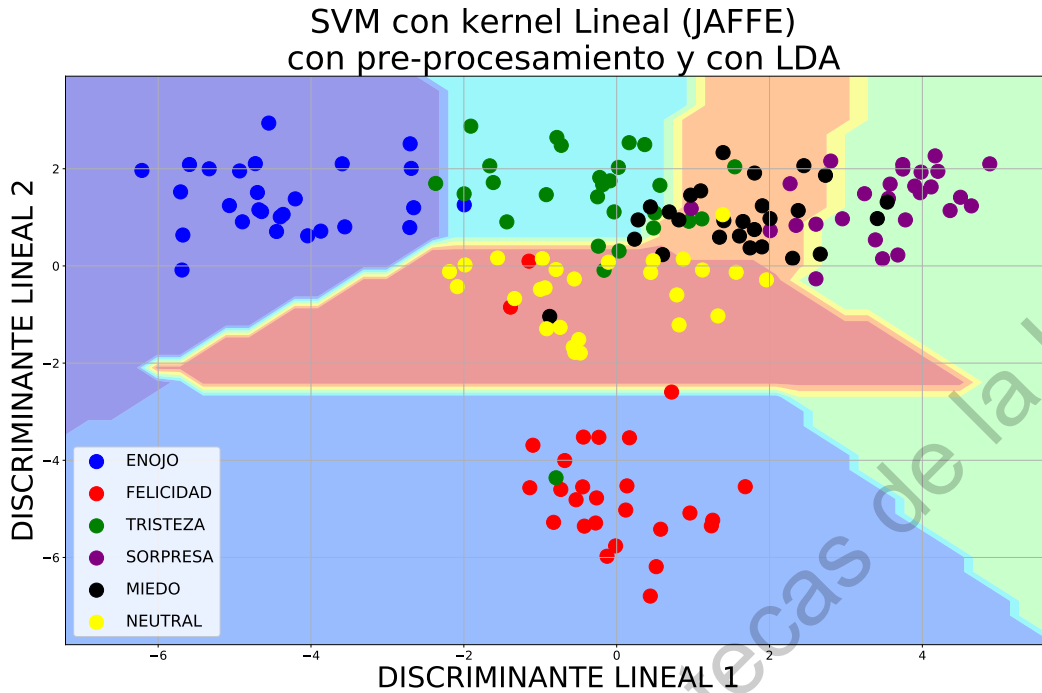


Figura 51. SVM con kernel Lineal y Radial utilizando el nuevo sub-espacio de características de PCA en la base de datos JAFEE (a) con pre-procesamiento, (b) sin pre-procesamiento.

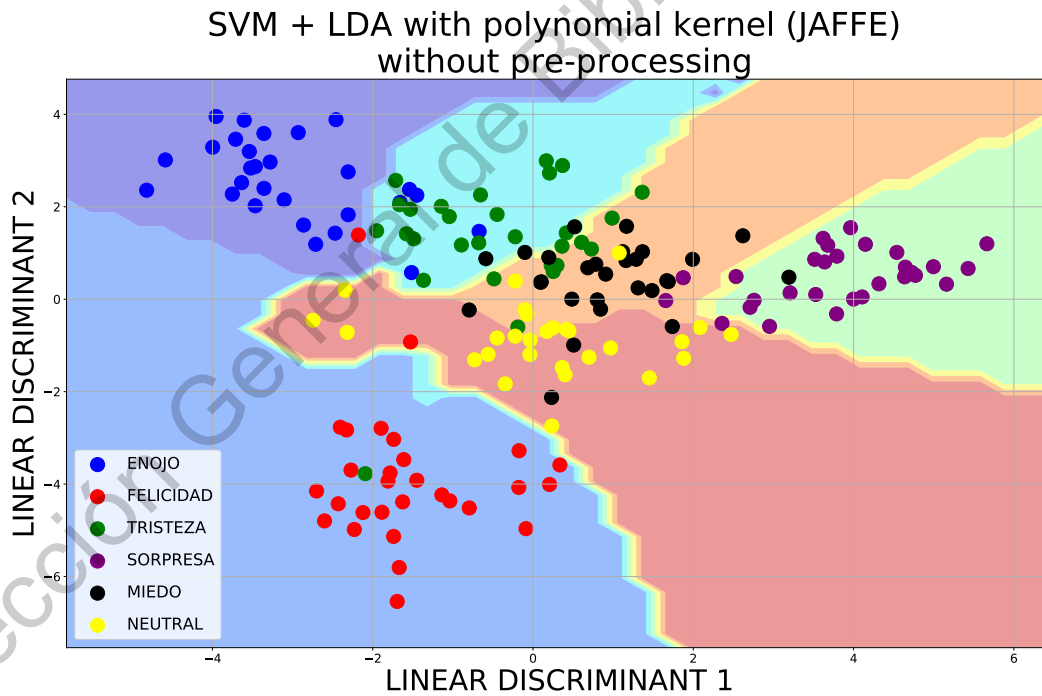
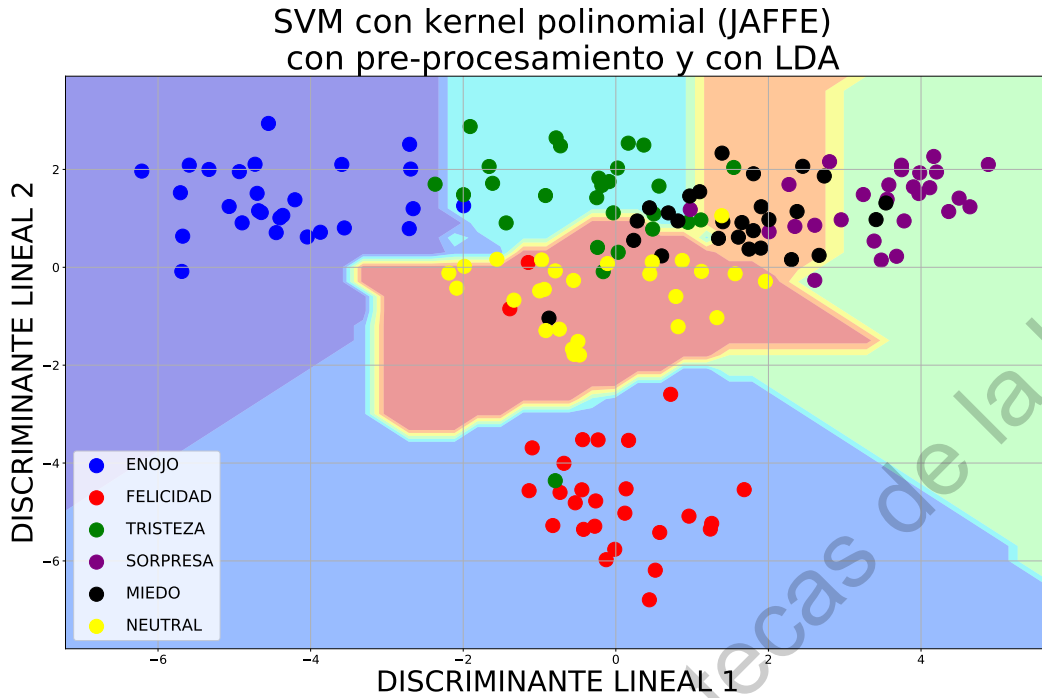


Figura 52. SVM con kernel Polinomial utilizando el nuevo sub-espacio de características de LDA en la base de datos JAFEE (a) con pre-procesamiento, (b) sin pre-procesamiento.

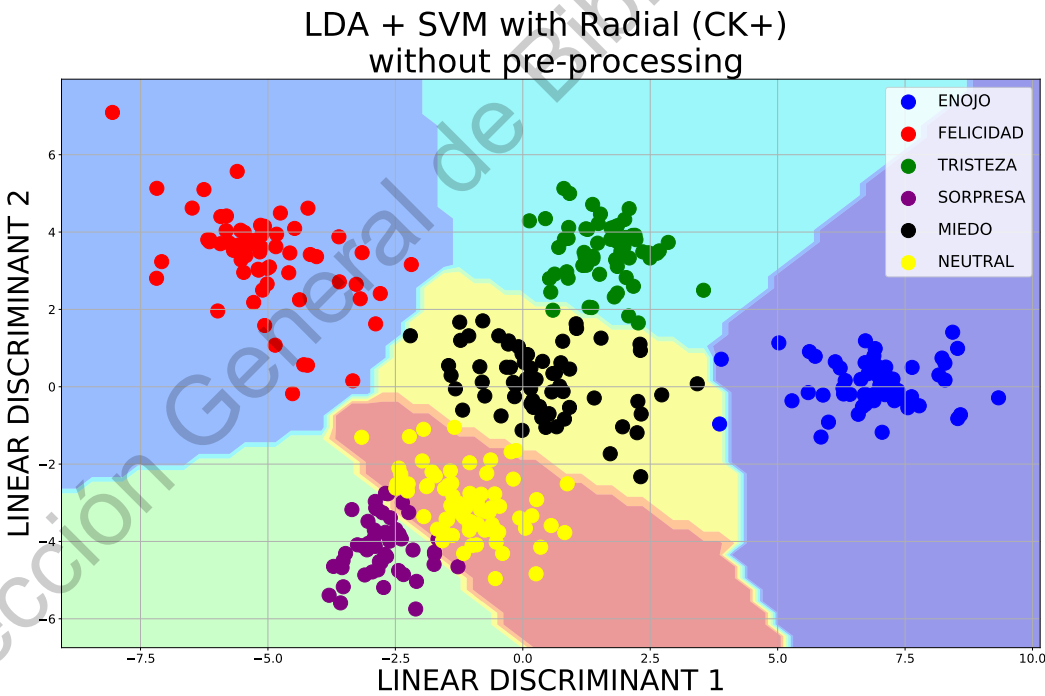
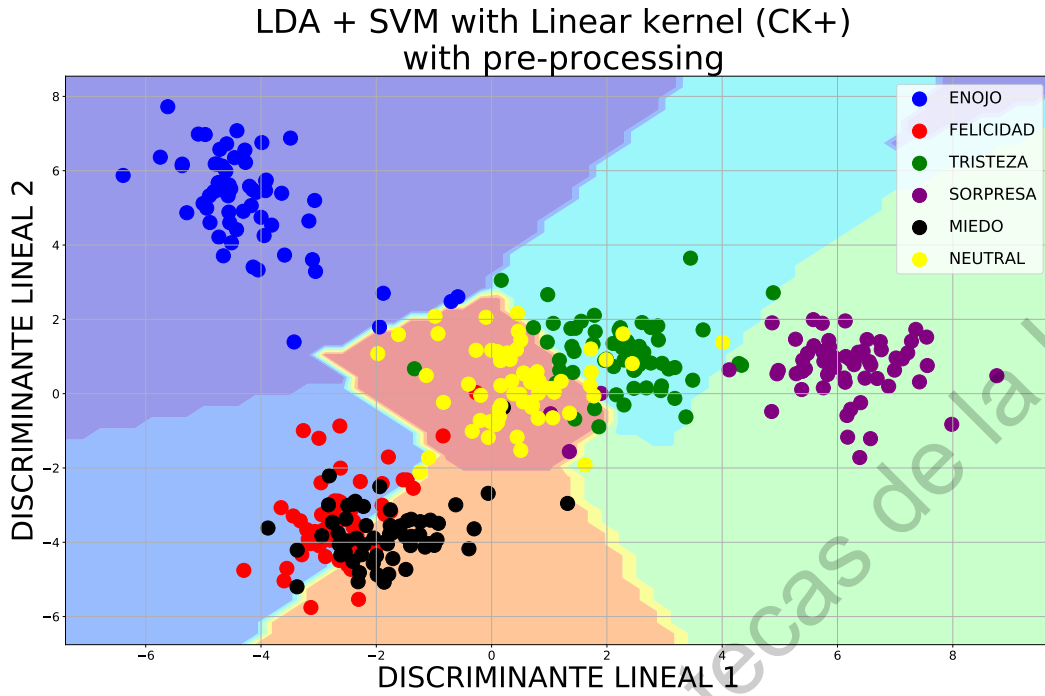


Figura 53. SVM con kernel Lineal y Radial utilizando el nuevo sub-espacio de características de LDA en la base de datos CK+ (a) con pre-procesamiento, (b) sin pre-procesamiento.

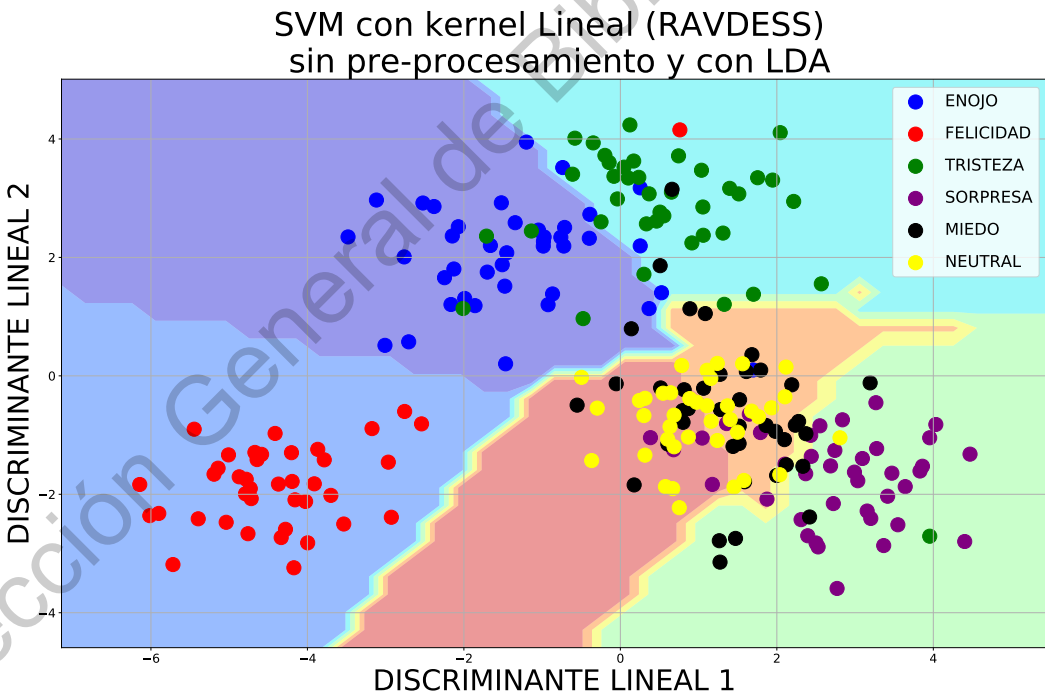
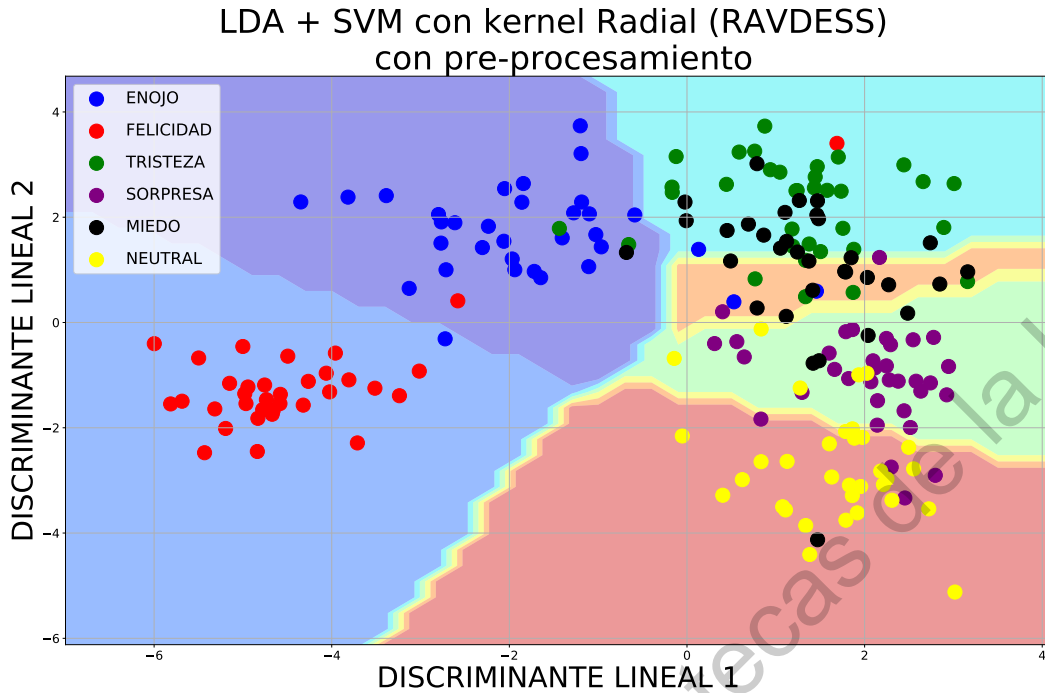


Figura 54. SVM con kernel Lineal y Radial utilizando el nuevo sub-espacio de características de LDA en la base de datos RAVDESS (a) con pre-procesamiento, (b) sin pre-procesamiento.

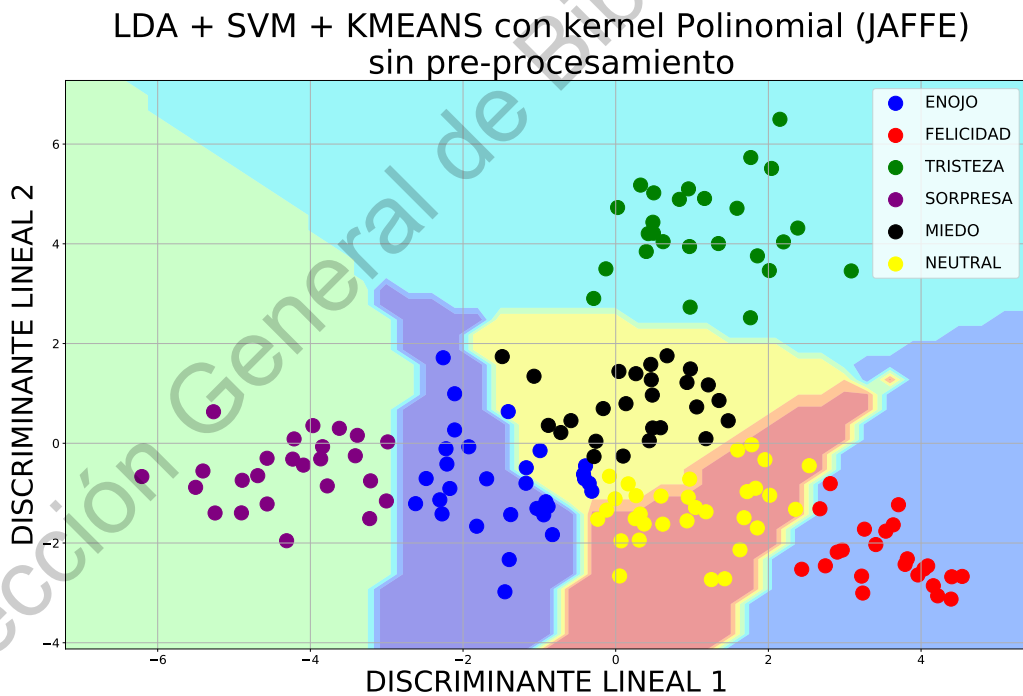
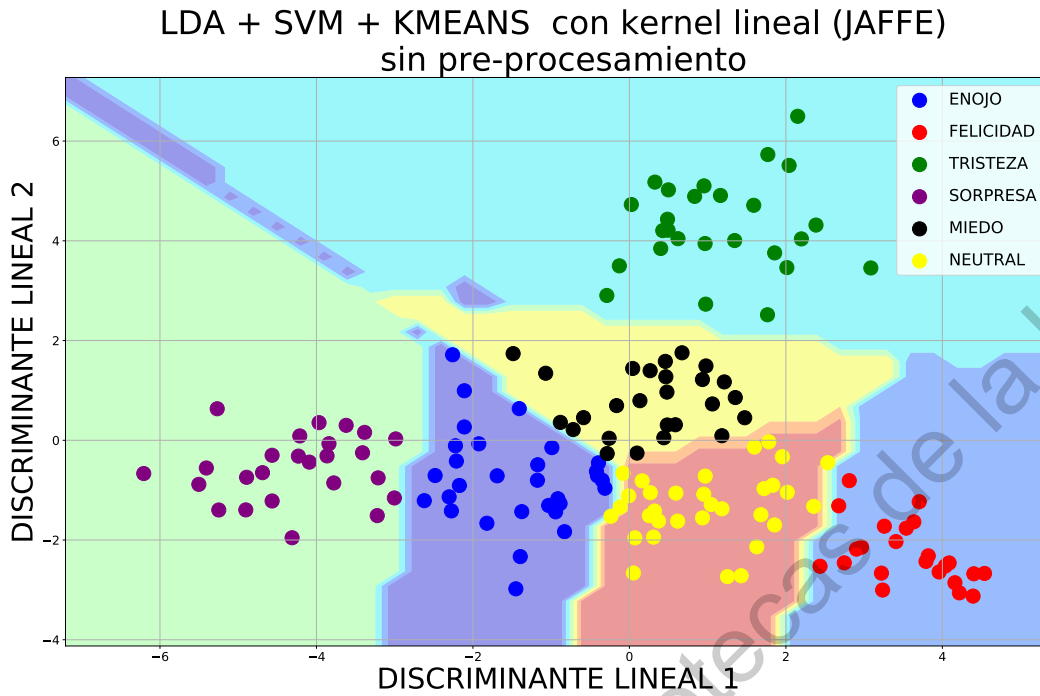
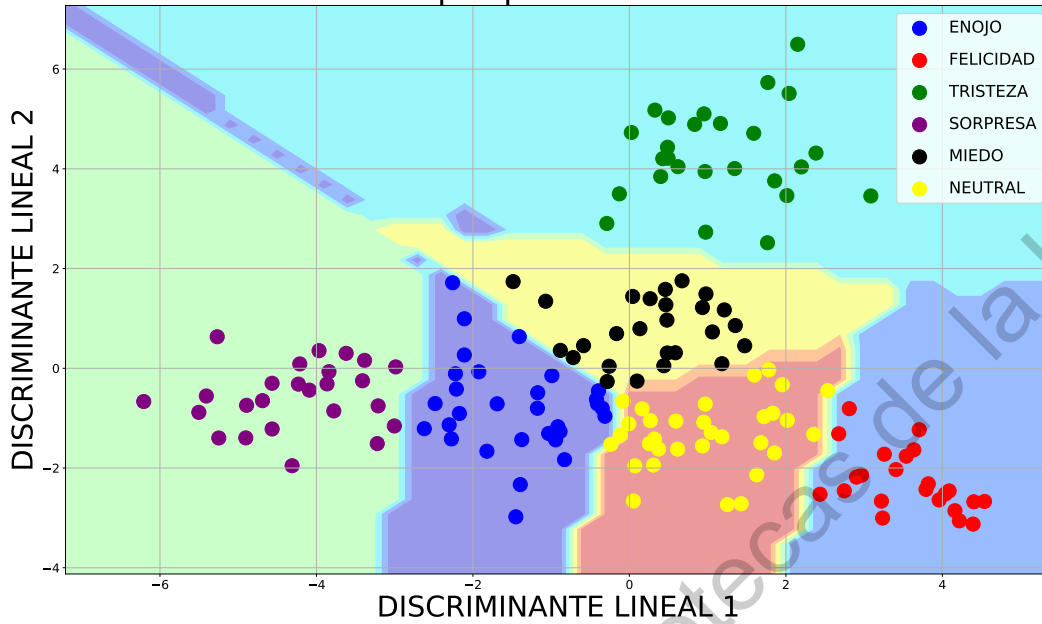


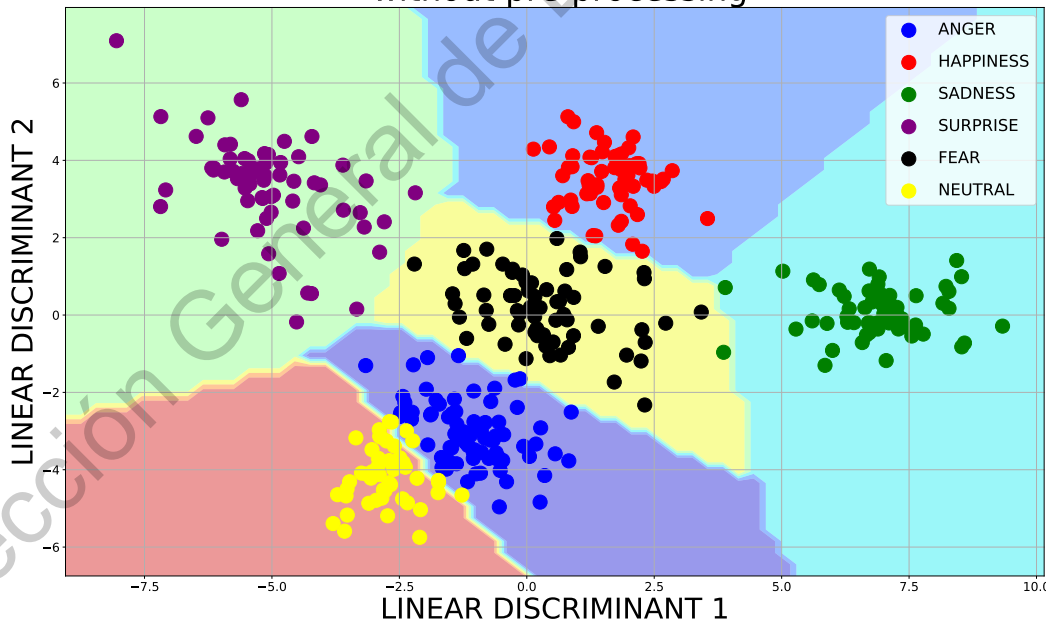
Figura 55. SVM con kernel Lineal y Radial utilizando el nuevo sub-espacio de características de PCA en la base de datos JAFEE sin pre-procesamiento

LDA + SVM + KMEANS con kernel lineal (JAFEE)
sin pre-procesamiento



(a) h

LDA + KMEASN + SVM with Radial (CK+)
without pre-processing



(b) h

Figura 56. SVM con kernel Lineal y Radial utilizando el nuevo sub-espacio de características de PCA en la base de datos JAFEE sin pre-procesamiento

3.6. Entrenamiento y pruebas

Se realizaron varios experimentos con las tres bases de datos en dos escenarios diferentes. Primer escenario que se explica en detalle en Algoritmo 6, tenemos imágenes con preprocesamiento. El segundo escenario que se explica en detalle en Algoritmo 7, tenemos imágenes sin preprocesamiento (imagen con cara extraída y nuevo tamaño). Cada escenario tiene tres configuraciones diferentes de los experimentos usando PCA + SVM, LDA + SVM y LDA + KMEANS + SVM, donde obtenemos precisión de clasificación usando la función de base lineal, radial y polinomial. Cada experimento se validó mediante validación cruzada donde las muestras de los algoritmos PCA (Componentes principales) y LDA (Discriminadores lineales) se dividieron en subconjuntos k de igual tamaño. Se usó un subconjunto para probar el paso (10%) y los otros nueve subconjuntos se usaron en el paso de entrenamiento (90%). El promedio de los resultados de n de los pliegues de k se calculó para producir una estimación de precisión única. En el caso de LDA + KMEANS + SVM, KMEANS se aplicó solo en los datos utilizados para el entrenamiento (90%). KMEANS se usa solo como una función auxiliar para datos ruidosos para construir el modelo SVM, luego se usan discriminantes lineales originales para la evaluación del modelo o paso de prueba.

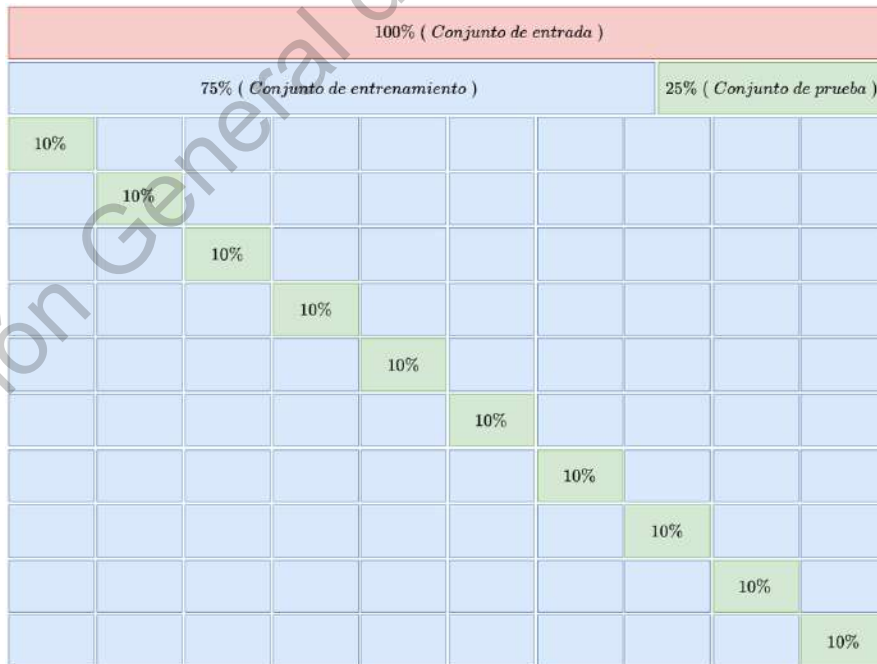


Figura 57. Example of the optimization problem in the stret

Algoritmo 6 Pasos a seguir para el experimento propuesto en el escenario 1

Entrada: Dado un conjunto $data_set$ de n imágenes con $N \times M$ píxeles

Salida: Exactitud de la clasificación de las expresiones faciales

- 1: Transformar $data_set$ en $filtered_faces$ usando el algoritmo 1
 - 2: Transformar $filtered_faces$ en $normalized_data$ con la normalización Z
 - 3: Transformar $normalized_data$ en $rotated_faces$ usando el algoritmo 1
 - 4: Transformar $rotated_faces$ en $extracted_faces$ usando el algoritmo 2
 - 5: Transformar $extracted_faces$ en una matriz X_1 de N filas y M columnas, donde N_i representa cada imagen
 - 6: Transformar $extracted_faces$ en una matriz X_2 de N filas y M columnas, donde M_i representa cada imagen
 - 7: Transformar la matriz X_2 en una matriz X_PCA de 2 componentes principales usando el algoritmo 4
 - 8: Transformar la matriz X_1 en una matriz X_LDA de 2 discriminantes lineales usando el algoritmo 3
 - 9: Transformar la matriz X_LDA en una matriz X_LDA_KM con 6 grupos usando el algoritmo K-Means
 - 10: Usando X_PCA , separar el conjunto en 10 sub-conjuntos. 9 sub-conjuntos para entrenamiento (90 %) y 1 para pruebas (10 %). Ejecutando el enfoque de la validación cruzada 10 veces con el modelo SVM y kernel.
 - 11: Usando X_LDA , separar el conjunto en 10 sub-conjuntos. 9 sub-conjuntos para entrenamiento (90 %) y 1 para pruebas (10 %). Ejecutando el enfoque de la validación cruzada 10 veces con el modelo SVM y kernel.
 - 12: Usando X_LDA_KM , separar el conjunto en 10 sub-conjuntos. 9 sub-conjuntos para entrenamiento (90 %) y 1 para pruebas (10 %). Ejecutando el enfoque de la validación cruzada 10 veces con el modelo SVM y kernel.
-

Algoritmo 7 Pasos a seguir para el experimento propuesto en el escenario 2

Entrada: Dado un conjunto *data_set* de *n* imágenes con $N \times M$ píxeles

Salida: Exactitud de la clasificación de las expresiones faciales

- 1: Transformar *data_set* en *extracted_faces* con el algoritmo 2
 - 2: Transformar *extracted_faces* en *normalized_data* con la normalización Z
 - 3: Transformar *normalized_data* en una matriz X_1 de N filas y M columnas, donde N_i representa cada imagen
 - 4: Transformar *normalized_data* en una matriz X_2 de N filas y M columnas, donde M_i representa cada imagen
 - 5: Transformar la matriz X_2 en una matriz X_PCA de 2 componentes principal con el algoritmo 4
 - 6: Transformar la matriz X_1 en una matriz X_LDA de 2 discriminantes lineales con el algoritmo 3
 - 7: Transformar la matriz X_LDA es una matriz X_LDA_KM con 6 grupos usando el algoritmo K-Means
 - 8: Usando X_PCA , separar el conjunto en 10 sub-conjuntos. 9 sub-conjuntos para entrenamiento (90 %) y 1 para pruebas (10 %). Ejecutando el enfoque de la validación cruzada 10 veces con el modelo SVM y kernel.
 - 9: Usando X_LDA , separar el conjunto en 10 sub-conjuntos. 9 sub-conjuntos para entrenamiento (90 %) y 1 para pruebas (10 %). Ejecutando el enfoque de la validación cruzada 10 veces con el modelo SVM y kernel.
 - 10: Usando X_LDA_KM , separar el conjunto en 10 sub-conjuntos. 9 sub-conjuntos para entrenamiento (90 %) y 1 para pruebas (10 %). Ejecutando el enfoque de la validación cruzada 10 veces con el modelo SVM y kernel.
-

3.7. Resultados

3.7.1 Escenario experimental

Los experimentos se realizaron usando Python 3.7.6, OpenCV 4.1.2, pandas 0.25.3, numpy 1.18.1, matplotlib 3.1.2, sklearn 0.22.1 y más bibliotecas de Python relacionadas con la ciencia de datos y la máquina aprendizaje. Además, la computadora usada fue una computadora portátil HP Pavilion Gaming 15-cx0001la con un Intel[®] Core[™] i5 + 8300H (frecuencia base de 2,3 GHz, hasta 4 GHz con Intel[®] Turbo Boost Tecnología, 8 MB de caché, 4 núcleos), unidad de estado sólido de 250GB Samsung M.2 y 16 GB de SDRAM DDR4-2666 con Windows 10 Home 64 como sistema operativo.

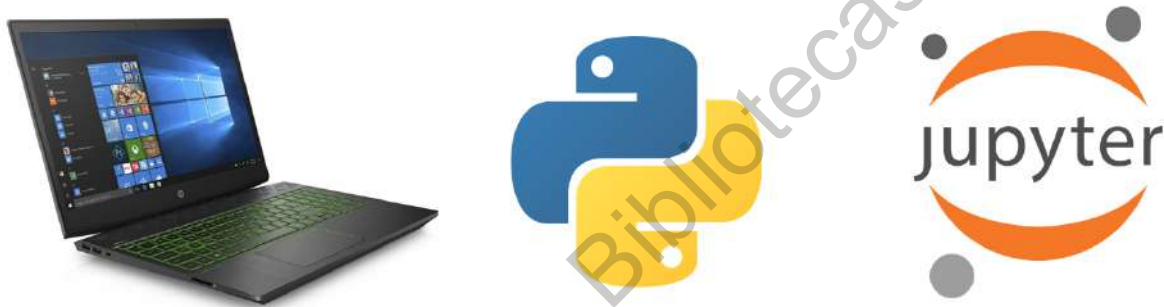


Figura 58. Herramientas de hardware y software utilizadas

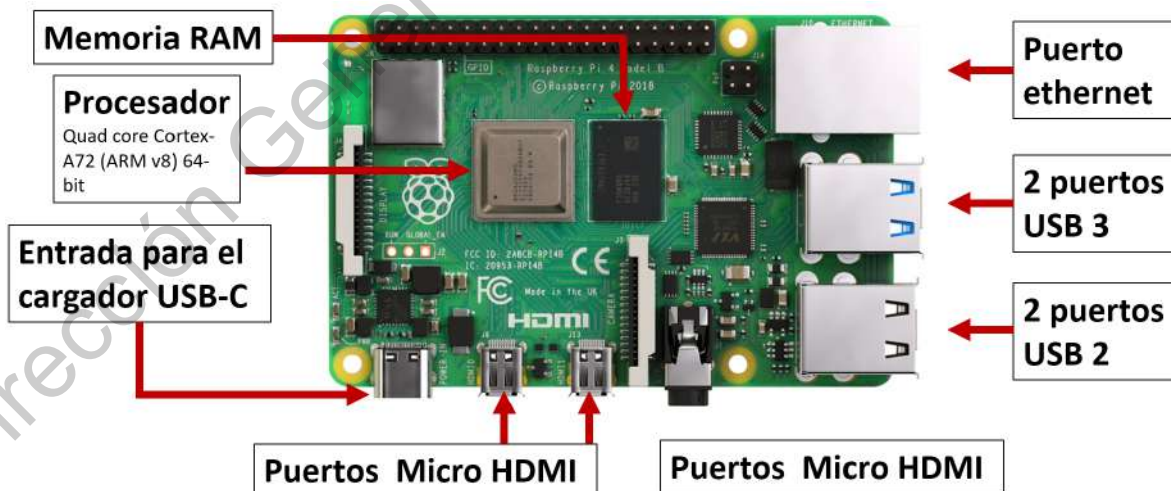
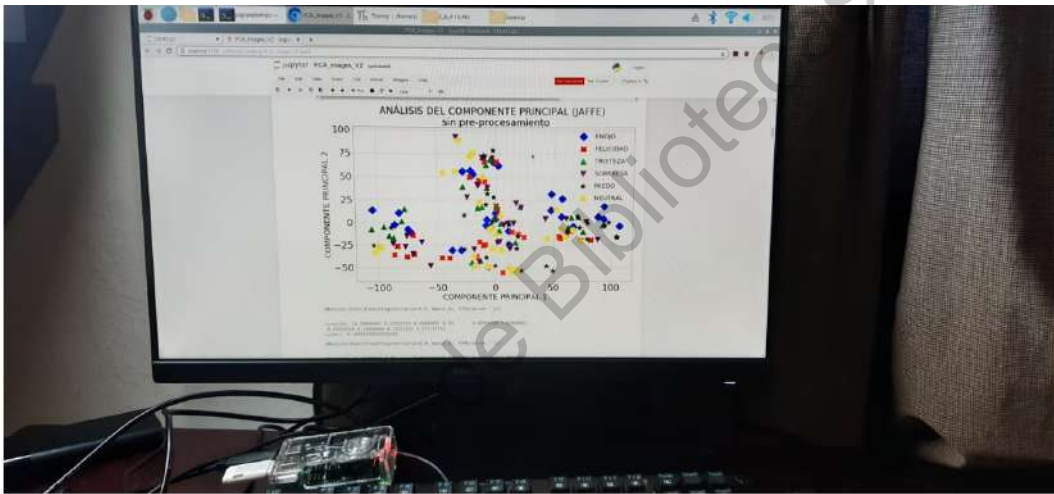


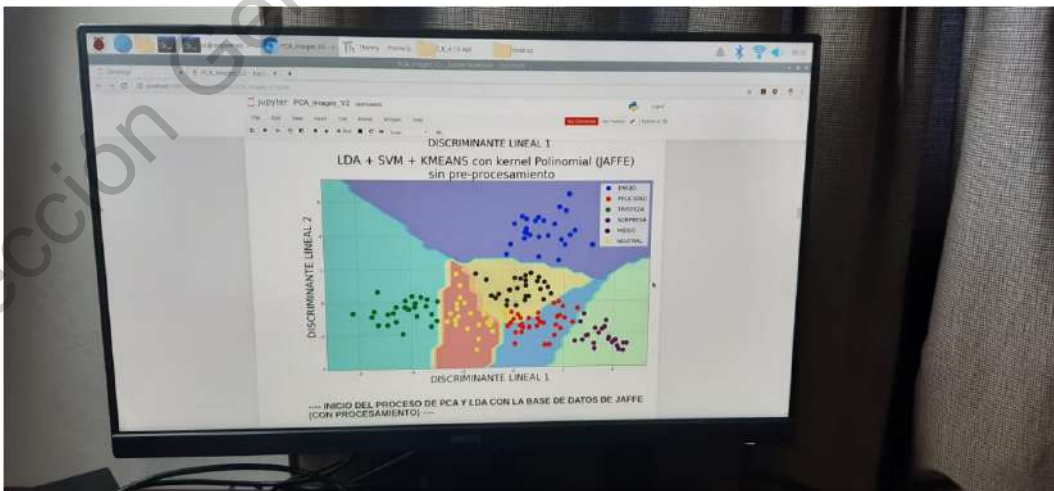
Figura 59. Tarjeta de desarrollo Raspberry



(a)



(b)



(c)

Figura 60. Sistema embebido conectado (a). PCA (b) y LDA + KMEANS + SVM (c).

3.7.2 tablas de resultados

Las tablas 2 a 7 muestran los resultados experimentales utilizando las tres bases de datos mencionadas anteriormente. La primera columna muestra el identificador del experimento de acuerdo con su orden y el nombre de la base de datos. La segunda columna muestra el kernel utilizado por el clasificador SVM. La tercera columna muestra la precisión obtenida de los componentes principales clasificados por SVM. La cuarta columna muestra la precisión obtenida de discriminantes lineales clasificados por SVM. La quinta columna muestra la precisión obtenida de discriminantes lineales agrupados por el algoritmo de K-medias y luego clasificados por SVM. Todos los resultados de PCA + SVM muestran un bajo rendimiento por debajo de 50 % de precisión en el paso de clasificación porque los componentes principales obtenidos del algoritmo PCA están completamente mezclados. Además, el núcleo de Polinomial requiere mucho tiempo en todos los experimentos de PCA + SVM. Por estas razones, PCA será mejor al obtener imágenes comprimidas como técnica de preprocesamiento para luego obtener discriminantes lineales con datos separables. El método propuesto LDA + K-Means + SVM logra una precisión media de reconocimiento de expresión facial de 96.01 % mientras que LDA + SVM logra una precisión media de 81.39 %, ambos resultados con imágenes sin algoritmos de procesamiento de imágenes utilizados, precisión media de imágenes con Los algoritmos de procesamiento de imágenes utilizados fueron 81.39 % y 77.18 % respectivamente.

Test ID	Kernel	PCA + SVM	LDA + SVM	LDA + K-Means + SVM
01-JAFFE	Lineal	12.22 %	80.83 %	97.63 %
02-JAFFE	Radial	17.22 %	80.55 %	95.16 %
03-JAFFE	Polinomial	Consumo alto de tiempo	79.72 %	94.33 %

Tabla 1. Resultados de la clasificación SVM con la base de datos JAFFE sin técnicas de pre-procesamiento

Test ID	Kernel	PCA + SVM	LDA + SVM	LDA + K-Means + SVM
04-JAFFE	Lineal	16.11 %	76.94 %	94.63 %
05-JAFFE	Radial	25.24 %	73.88 %	92.48 %
06-JAFFE	Polinomial	Consumo alto de tiempo	71.38 %	90.18 %

Tabla 2. Resultados de la clasificación SVM con la base de datos JAFFE con técnicas de pre-procesamiento

Test ID	Kernel	PCA + SVM	LDA + SVM	LDA + K-Means + SVM
01-CK	Lineal	13.27 %	93.88 %	99.05 %
02-CK	Radial	21.66 %	94.22 %	99.11 %
03-CK	Polinomial	Consumo alto de tiempo	91.55 %	98.44 %

Tabla 3. Resultados de la clasificación SVM con la base de datos CK+ sin técnicas de pre-procesamiento

Test ID	Kernel	PCA + SVM	LDA + SVM	LDA + K-Means + SVM
04-CK	Lineal	33.44 %	82.94 %	95.15 %
05-CK	Radial	38.44 %	82.44 %	93.32 %
06-CK	Polinomial	Consumo alto de tiempo	82.38 %	91.45 %

Tabla 4. Resultados de la clasificación SVM con la base de datos CK+ con técnicas de pre-procesamiento

Test ID	Kernel	PCA + SVM	LDA + SVM	LDA + K-Means + SVM
01-RAVD	Lineal	18.61 %	71.94 %	91.14 %
02-RAVD	Radial	23.47 %	72.77 %	94.89 %
03-RAVD	Polinomial	Consumo alto de tiempo	67.08 %	94.42 %

Tabla 5. Resultados de la clasificación SVM con la base de datos RAVDESS sin técnicas de pre-procesamiento

Test ID	Kernel	PCA + SVM	LDA + SVM	LDA + K-Means + SVM
04-RAVD	Lineal	19.02 %	76.11 %	92.94 %
05-RAVD	Radial	21.66 %	76.25 %	95.24 %
06-RAVD	Polinomial	Consumo alto de tiempo	72.36 %	95.11 %

Tabla 6. Resultados de la clasificación SVM con la base de datos RAVDESS con técnicas de pre-procesamiento

Conclusiones y trabajo futuro

En este trabajo de investigación, se propone la implementación de un sistema de reconocimiento de EF utilizando algunos algoritmos de procesamiento de imágenes y aprendizaje automático, además se incluyen algunas definiciones y expresiones matemáticas se explican en un alto nivel así como figuras y diagramas para facilitar su comprensión.

En base a los resultados presentados en este trabajo, está claro que, en este caso, las técnicas de procesamiento de imágenes, como el filtrado y la ecualización no ayudan con la separación de datos en el paso de reducción de dimensionalidad y no aumentan la precisión del algoritmo SVM por lo que podría omitirse este paso. Aunque podría ser obvio que la rotación de la imagen ayuda como un paso de preprocesamiento para solucionar el problema de la pose facial, no medimos el impacto de la implementación de este algoritmo sin el filtrado y la ecualización que podrían probarse y presentarse en un trabajo futuro.

En el paso de reducción de dimensionalidad, hemos visto que el algoritmo LDA puede resolver mejor el problema de reducción, así como el problema de separación de clases mediante la extracción de discriminantes lineales basados en la varianza entre clases y dentro de la clase para luego usar los nuevos datos proyectados como un nuevo conjunto de características que podemos clasificar. PCA no muestra un buen rendimiento con el problema de separación de datos en relación con LDA porque los componentes principales solo consideran la variación de los datos para proyectar un nuevo subespacio de características para reconstruir los datos originales. En lugar de usar los componentes principales como un nuevo

conjunto de características, el algoritmo PCA podría usarse para extraer las características faciales más importantes y luego reconstruir una parte de las imágenes aumentando o disminuyendo la varianza total de los componentes principales en futuras investigaciones.

En el paso de clasificación, hemos propuesto utilizar el algoritmo SVM con diferentes configuraciones presentadas en el paso de escenario experimental. La contribución más importante en este trabajo de investigación es el uso del algoritmo K-means para ayudar en el proceso de construcción del modelo SVM, en la sección de resultados experimentales se muestra que la precisión SVM aumenta cuando los datos de entrenamiento se agrupan por K-means, luego las pruebas Los datos se utilizan para evaluar el modelo SVM. El algoritmo K-means no evita completamente los errores en el paso de clasificación para construir el modelo, pero reducimos las clasificaciones erróneas en el paso de entrenamiento.

Una limitación del enfoque propuesto usando LDA + K-MEANS + SVM es que K-means crea un agrupamiento diferente cada vez y cambia el color de las muestras, lo que dificulta la reorganización de las muestras de prueba. Las estrategias para evitar este problema podrían proponerse un algoritmo híbrido con un enfoque supervisado. Además, los algoritmos de procesamiento de imágenes no ayudan en la extracción de rasgos faciales globales.

La investigación futura incluye la experimentación con otros tipos de características faciales, como características geométricas o el uso de descriptores faciales utilizando más bases de datos e imágenes, así como comparar el clasificador SVM con diferentes algoritmos de clasificación para el reconocimiento de EF.

Bibliografía

Bellamkonda, Sivaiah y NP Gopalan. 2018. A Facial Expression Recognition Model using Support Vector Machines.

Chen, Jinhui, Tetsuya Takiguchi y Yasuo Arika. 2015. A robust SVM classification framework using PSM for multi-class recognition. EURASIP Journal on Image and Video Processing 2015(1):7.

Cortes, Corinna y Vladimir Vapnik. 1995. Soft margin classifiers. Machine learning 20:273–297.

Ekman, P. 1977. Facial action coding system. Biological and cultural contributions to body and facial movement :34–84.

Francois, Chollet. 2017. Deep learning with Python.

Hazewinkel, Michiel. 2001. Affine transformation. Encyclopedia of Mathematics, Springer .

History of Python. 2020.

Jalal, Ahsan y Usman Tariq. 2019. Semi-supervised clustering of unknown expressions. Pattern Recognition Letters 120:46–53.

Kanade, Takeo, Jeffrey F Cohn y Yingli Tian. 2000. Comprehensive database for facial expression analysis. Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580). IEEE :46–53.

Kar, Nikunja Bihari, Korra Sathya Babu, Arun Kumar Sangaiah y Sambit Bakshi. 2019. Face expression recognition system based on ripplelet transform type II and least square SVM. Multimedia Tools and Applications 78(4):4789–4812.

- Leo, M Judith y S Suchitra. 2018. SVM Based Expression-Invariant 3D Face Recognition System. *Procedia computer science* 143:619–625.
- Livingstone, Steven R y Frank A Russo. 2018. The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English. *PloS one* 13(5):e0196391.
- Lloyd, Stuart P. 1957. Least Squares Quantization in PCM. *IEEE Transactions on Information* .
- Lopes, Nuno, André Silva, Salik Ram Khanal, Arsênio Reis, João Barroso, Vitor Filipe y Jaime Sampaio. 2018. Facial emotion recognition in the elderly using a SVM classifier. 2018 2nd International Conference on Technology and Innovation in Sports, Health and Wellbeing (TISHW). *IEEE* :1–5.
- López, Yolanda Olvera. 2000. *Inteligencia emocional*. Plaza y Valdes.
- Lyons, Michael J, Shigeru Akamatsu, Miyuki Kamachi, Jiro Gyoba y Julien Budynek. 1998. The Japanese female facial expression (JAFFE) database. *Proceedings of third international conference on automatic face and gesture recognition*. :14–16.
- Makhmudkhujaev, Farkhod, M Abdullah-Al-Wadud, Md Tauhid Bin Iqbal, Byungyong Ryu y Oksam Chae. 2019. Facial expression recognition with local prominent directional pattern. *Signal Processing: Image Communication* 74:1–12.
- Manual, OpenCV. 1999. Open source computer vision library. Intel Corporation, USA 2001.
- Retana, José Ángel García. 2012. La educación emocional, su importancia en el proceso de aprendizaje. *Revista Educación* 36(1):1–24.
- Saeed, Saeeda, Junaid Baber, Maheen Bakhtyar, Ihsan Ullah, Naveed Sheikh, Imam Dad y Anwar Ali Sanjrani. 2018. Empirical evaluation of SVM for facial expression recognition. *Int. J. Adv. Comput. Sci. Appl* 9(11):670–673.

- Shah, Jamal Hussain, Muhammad Sharif, Mussarat Yasmin y Steven Lawrence Fernandes. 2017. Facial expressions classification and false label reduction using LDA and threefold SVM. Pattern Recognition Letters .
- Van Rossum, Guido et al. 2007. Python Programming Language. USENIX annual technical conference. 41 p. 36.
- Vapnik, Vladimir y AJ Lerner. 1963. Generalized portrait method for pattern recognition. Automation and Remote Control 24(6):774–780.
- Vapnik, VN. 1998. Statistical Learning Theory Wiley-Interscience. New York .
- Viola, Paul, Michael Jones et al. 2001. Rapid object detection using a boosted cascade of simple features. CVPR (1) 1(511-518):3.
- Wu, Tong, Yanni Xiao, Muhan Guo y Feiping Nie. 2019. A General Framework for Dimensionality Reduction of K-Means Clustering. Journal of Classification :1–16.
- Ye, Jieping, Zheng Zhao y Mingrui Wu. 2008. Discriminative k-means for clustering. Advances in neural information processing systems. :1649–1656.

Anexos



Figura 61. Constancia de participación como presentador oral en el Coloquio de la Facultad de Ingeniería de la Universidad Autónoma de Querétaro



UNIVERSIDAD AUTÓNOMA DE QUERÉTARO
FACULTAD DE LENGUAS Y LETRAS



A QUIEN CORRESPONDA:

La que suscribe, Directora de la Facultad de Lenguas y Letras, hace **C O N S T A R** que

AGUAYO SERRANO FRANCISCO EMILIANO

Presentó el **Examen de Manejo de la Lengua** efectuado el día cinco de noviembre de dos mil diecinueve, en el cual obtuvo la siguiente calificación:

8

Se extiende la presente a petición de la parte interesada, para los fines escolares y legales que le convengan, en el Campus Aeropuerto de la Universidad Autónoma de Querétaro, el día diecinueve de noviembre de dos mil diecinueve.

Atentamente,
"Enlazar Culturas por la Palabra"



LIC. LAURA PÉREZ TÉLLEZ



LPT/evm*CL*FLL-C.-783

UNIVERSIDAD AUTÓNOMA DE QUERÉTARO
Campus Aeropuerto, Anillo Vial Fray Junípero Serra S/N, Querétaro, Qro.
C.P. 76140, Tel. 01 (442) 192 12 00 Dirección Ext. 61010,
Secretaría Administrativa 61300, Posgrado 61140, Licenciatura 61070,
Centro de Lenguas 61050, Secretaría Académica 61100 y Planeación 61110

SOMOS UAQ
EDUCAR. CRECER. CONSOLIDAR

Figura 62. Constancia de cumplimiento del requisito de inglés emitida por la Facultad de Lenguas y Letras de la Universidad Autónoma de Querétaro, equivalente a 550 puntos TOEFL