

UNIVERSIDAD AUTÓNOMA DE QUERÉTARO



FACULTAD DE INGENIERÍA
MAESTRÍA EN CIENCIAS EN INTELIGENCIA ARTIFICIAL

TESIS

Sistema de Reconocimiento Facial Basado en SVM

Presenta:

Pablo Augusto CAMPOS
ZÁRATE

Supervisor:

Dr. Efrén GORROSTIETA
HURTADO

Centro Universitario
Querétaro, QRO
México.
Marzo 2021



Universidad Autónoma de Querétaro

Facultad de Ingeniería

Maestría en Ciencias en Inteligencia Artificial

Sistema de Reconocimiento Facial Basado en SVM

TESIS

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

Presenta:

Pablo Augusto Campos Zárate

Dirigido por:

Dr. Efrén Gorrostieta Hurtado

SINODALES

Dr. Efrén Gorrostieta Hurtado
Presidente

Firma

Dr. Jesús Carlos Pedraza Ortega
Secretario

Firma

Dr. Juan Manuel Ramos Arreguín
Vocal

Firma

Dr. Marco Antonio Aceves Fernández
Suplente

Firma

Dr. Saúl Tovar Arriaga
Suplente

Firma

Centro Universitario
Querétaro, QRO
México.
Marzo 2021

Dirección General de Bibliotecas UAQ

Dirección General de Bibliotecas UAQ

Dedicado a los que ocupan un espacio en mi corazón

Dirección General de Bibliotecas UAQ

Agradecimientos

Gracias a las personas que me apoyaron emocionalmente en cada instante de este proceso, a los profesores que me brindaron algo de su sabiduría, a la Universidad Autónoma de Querétaro por permitirme continuar con mi formación académica y al Consejo Nacional de Ciencia y Tecnología por el apoyo económico.

Dirección General de Bibliotecas UAQ

Dirección General de Bibliotecas UAQ

Abstract

Face recognition is an interesting problem to Computer vision and pattern recognition researchers, due to its inherent difficulties dealing with pose and lighting variation and face expressions it keeps attracting the eye of those enthusiasts. In this work a new face recognition methodology is proposed using 3D data from the Texas 3DFRDatabase and combining local descriptors like Local Derivative Patterns and Angular Radial Signature to extract feature vectors from each face in the database and feeding those vectors to a Support Vector Machines to perform the feature matching, the results over 95 % in verification and recognition task are comparable with state-of-the-art methods. This work is organized as follows, in Chapter 1 you will find an introduction to face recognition and a summary of the state-of-the-art methods, in Chapter 2 some concepts to understand SVM's and LDP and ARS algorithms are reviewed, the description of the proposed methodology is in Chapter 3, in Chapter 4 are the results from the experiments using multiple parameters, finally, conclusions and future work are in Chapter 5.

Dirección General de Bibliotecas UAQ

Resumen

El reconocimiento facial es un problema interesante para la comunidad de visión computacional y reconocimiento de patrones debido a las problemáticas inherentes a este, las variaciones de iluminación, pose y expresiones facial siguen motivando la investigación en este campo. En este trabajo se propone una metodología para un sistema de reconocimiento facial que utiliza información tridimensional del rostro humano de una base de datos de la universidad de Texas y combina los descriptores *Local Derivative Patterns* con *Angular Radial Signature* para obtener los vectores representativos de cada rostro, cada uno de estos vectores se alimentó a una *Support Vector Machine* para realizar la clasificación, los resultados obtenidos son comparables a métodos del estado del arte con *Recognition rate* y *Verification rate* superiores al 95 %.

Este trabajo se divide como sigue, el Capítulo 1 contiene una introducción al reconocimiento facial y una breve revisión al estado del arte, en el Capítulo 2 se revisan los conceptos para el entendimiento de las SVMS y los descriptores LDP y ARS y continúa con la descripción de la metodología propuesta y los datos utilizados en el Capítulo 3, los resultados obtenidos en los experimentos realizados con múltiples parámetros se encuentran en el Capítulo 4 y finalmente las conclusiones y trabajo futuro en el Capítulo 5

Dirección General de Bibliotecas UAQ

Índice general

Agradecimientos

Abstract I

Resumen III

1. Introducción 1

1.1. Reconocimiento de patrones 1

1.2. Reconocimiento facial 1

1.3. Antecedentes 4

1.4. Estado del arte 5

1.4.1. Basados en puntos clave 5

1.4.2. Basados en curvas 6

1.4.3. Basados en superficies 6

1.4.4. *Matching* basado en SVM 6

1.5. Descripción del problema 8

1.6. Justificación 9

1.7. Hipótesis 9

1.8. Objetivos 9

1.8.1. Objetivo general 9

1.8.2. Objetivos específicos 9

2. Marco teórico 11

2.1. Datos linealmente separables 11

2.2. Hiperplanos 12

2.2.1. ¿Qué es un hiperplano? 12

2.2.2. Clasificación de datos a partir de un hiperplano 13

2.3. *Support Vector Machines* 14

2.3.1. Hiperplano del margen máximo 14

2.3.2.	Hiperplano de margen suave	17
2.3.3.	Caso no separable linealmente	18
2.3.4.	Funciones Kernel	20
2.3.5.	Validación y ajuste de hiperparámetros	23
2.4.	<i>Local Normal Derivative Pattern</i>	24
2.4.1.	Normales de superficie	24
2.4.2.	LNDP	25
2.4.3.	<i>Angular Radial Signature</i>	27
3.	Metodología	31
3.1.	Descripción general	31
3.2.	Especificaciones	31
3.3.	Base de datos	33
3.4.	Descriptor	34
3.4.1.	Obtención de las normales de superficie	34
3.4.2.	Máscara binaria	36
3.4.3.	<i>Local Derivative Pattern</i>	38
3.5.	<i>Matching</i> de características	38
3.5.1.	Medición del desempeño	40
4.	Resultados	43
4.1.	Resultados preliminares	43
4.2.	Efectos de los parámetros en la extracción	45
4.2.1.	Efectos de la cantidad de <i>bins</i>	45
4.2.2.	Efectos de la cantidad de ARS	48
4.2.3.	Efectos de las normales de superficie	50
4.3.	Efectos de los hiperparámetros en el entrenamiento	52
4.4.	Comparación con métodos del estado del arte	55
4.5.	Discusión de los resultados	56
5.	Conclusiones y trabajo futuro	63
5.1.	Conclusiones	63
5.2.	Trabajo futuro	64
6.	Anexos	71
6.1.	Anexo I: Ponencia en COMIA 2020	72
6.2.	Anexo II: Artículo publicado	73
6.3.	Anexo III: Prueba de idioma FLL	74

Índice de figuras

1.1.	Modos de trabajo posibles para un sistema de reconocimiento facial.	2
1.2.	Módulos de un sistema de reconocimiento facial	3
2.1.	Representación en el plano de la tabla de verdad de una compuerta OR, la línea separa las salidas 0 y 1.	12
2.2.	Datos linealmente separables en un espacio tridimensional, un plano de dos dimensiones separa los datos de dos clases diferentes	13
2.3.	Un conjunto de datos linealmente separable puede ser dividido por una cantidad infinita de hiperplanos distintos	14
2.4.	En una SVM, se separan los datos con un hiperplano óptimo, el cual maximiza el margen con los datos.	15
2.5.	Diferencia entre vectores de soporte para calcular el margen	16
2.6.	Efecto de diferentes valores para C , una C pequeña resulta en un margen más ancho, por otro lado, utilizar una C más grande dará un margen más pequeño que acepta menos errores.	19
2.7.	El uso de un determinado kernel depende del conjunto de datos que se esté utilizando	22
2.8.	Ejemplo de validación cruzada (muestra tomada de la documentación de [1]), el conjunto de entrenamiento es dividido en <i>5-folds</i> para la validación	23
2.9.	Vector normal a p_i con una vecindad de tamaño 8 estimado ajustando un plano al conjunto de puntos.	24
2.10.	De izquierda a derecha, imagen de rango original de la base de datos <i>Texas 3DFRD</i> , imagen normal del componente x , imagen normal del componente y , imagen normal del componente z	25
2.11.	Imágenes normales en x, y y z divididas en parches, los histogramas de cada parche para cada imagen normal se concatenan, la concatenación de los histogramas de cada imagen normal forman el descriptor final.	26
2.12.	Máscara binaria utilizada para 8 ARS, (basada en la presentada por [2])	28
2.13.	Máscara binaria con distintas cantidades de ARS sobre una imagen de rango de la base de datos	29
2.14.	17 ARS calculadas sobre una superficie tridimensional del rostro.	30

3.1.	Flujo de trabajo para la toma de decisión en una imagen de prueba	33
3.2.	Imagen preprocesada de la base de datos así como varias vistas del perfil tridimensional que se obtienen de ella	35
3.3.	Tiempo de cálculo de normales de superficie con distintos valores de k	36
3.4.	Imágenes normales con distintos valores de k para una imagen de rango, de izquierda a derecha son la normal en x , normal en y y normal en z	37
3.5.	Tiempo de cálculo de la máscara binaria para diferentes cantidades de ARS	38
3.6.	Tiempo de cálculo de características LNDP para distintos números de ARS	39
4.1.	Resultados preliminares	44
4.2.	Tiempos de extracción de características LNDP para una imagen en la base de datos a diferentes escalas	45
4.3.	RR con 17 ARS-LDP 3 y distintas cantidades de <i>bins</i>	48
4.4.	Patrones obtenidos con distintas construcciones de histogramas para 17 ARS y LDP de grado 3 de un solo individuo	49
4.5.	VR para diferentes cantidades de bins y LDP de tercer grado	50
4.6.	RR con diferentes valores de ARS y LDP de grado 3	52
4.7.	VR para diferentes cantidades de ARS y LDP de tercer grado	53
4.8.	RR para diferentes cantidades de ARS y LDP de grado 2	54
4.9.	VR con diferentes cantidades de ARS y LDP de segundo grado	55
4.10.	Región de interés para extracción de las normales de superficie.	55
4.11.	RR para 8 ARS y LNDP de segundo grado utilizando los patrones obtenidos de las normales x,y y z por separado	56
4.12.	RR para 8 ARS y LNDP de segundo grado como un solo patrón.	57
4.13.	Matriz de confusión para los patrones obtenidos utilizando las 3 normales de superficie, 8 ARS y LDP de grado 2.	58
4.14.	RR de 8ARS-LNDP2 con distintos valores de σ y $C = 1$	60
4.15.	RR de 8ARS-LDP2 con kernel polinomial de grado 2 y $C = 0,1$ con normales de superficie	60
4.16.	RR de 8ARS-LDP2 con kernel polinomial de grado 2 y $C = 0,1$ sin normales de superficie	61
4.17.	VR de 8ARS-LDP con kernel polinomial de segundo grado y $C = 0,1$ sin normales de superficie	61
4.18.	Curva CMC que presenta la comparación de los métodos evaluados en la tarea de reconocimiento	62
4.19.	Curva ROC para los resultados comparados en la tarea de verificación, la línea roja intersecta las curvas en 0.1 FAR	62

Índice de tablas

1.1. Principales categorías de descriptores locales.	4
1.2. Métodos locales para el reconocimiento facial en 3D	7
1.3. Sistemas de reconocimiento facial con <i>matching</i> basado en SVM con la RR reportada	8
2.1. Tabla de verdad de una compuerta lógica OR	11
3.1. Particiones propuestas por los creadores de la Texas 3DFRD	34
3.2. Matriz de confusión para un clasificador binario	42
4.1. Verification rate y Recognition rate para puntos fijos en 0.1 FAR y Rank-1	44
4.2. Desempeño del método propuesto utilizando 17 ARS y distintos números de <i>bins</i> en los histogramas con KPCA.	46
4.3. Desempeño del método propuesto utilizando 17 ARS y distintos números de <i>bins</i> en los histogramas sin KPCA.	47
4.4. Desempeño del método propuesto utilizando 17 ARS sin construcción de histogramas	47
4.5. VR y FAR para diferentes cantidades de barras y $\Delta = 10,8$	51
4.6. RR para diferentes cantidades de ARS-LDP3 con el tiempo de extracción	51
4.7. VR con diferentes números de ARS y LDP 3 para $\Delta = 10,8$	52
4.8. RR para diferentes cantidades de ARS-LDP2 con el tiempo de extracción	53
4.9. VR y FAR con distintas cantidades de ARS y LDP 2 para $\Delta = 10,8$	54
4.10. Comparación de resultados de reconocimiento en Rank-1 y verificación a un valor fijo de 0.1FAR	56

Dirección General de Bibliotecas UAQ

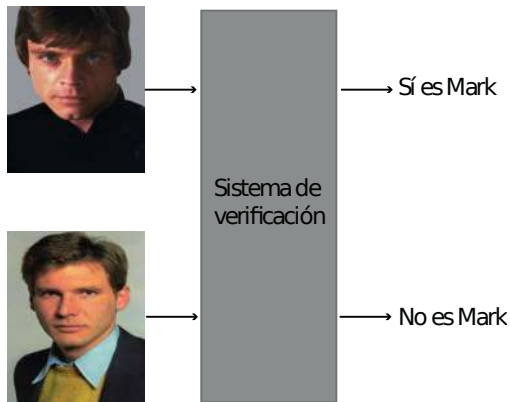
Introducción

1.1. Reconocimiento de patrones

El reconocimiento de patrones o *Pattern Recognition* es una disciplina de las ciencias de la computación, cuyo objetivo es la clasificación de objetos en un número de categorías o clases distintas [3], los objetos a clasificar se denotan como patrones. La historia del reconocimiento de patrones antes de la década de los 60s se basa mayormente en el producto de pura investigación teórica en el campo de la estadística, fue con el desarrollo de las computadoras que la demanda de aplicaciones prácticas de reconocimiento de patrones aumentó. Un campo en el que el reconocimiento de patrones es de gran importancia es el campo de la visión por computadora, donde el sistema captura imágenes a través de una cámara y después de analizarlas, debe determinar qué hay en las imágenes, algunos ejemplos de esto son identificación de huellas dactilares, sistemas de inspección a nivel industrial, autenticación de firmas y reconocimiento facial.

1.2. Reconocimiento facial

El reconocimiento facial es un problema perteneciente al reconocimiento de patrones donde el objeto tridimensional a identificar es el rostro humano, la investigación en reconocimiento facial está motivada por los retos inherentes al problema, así como la gran cantidad de aplicaciones prácticas que tiene. Un sistema de reconocimiento facial debe ser capaz de identificar rostros en imágenes y videos de manera automática y puede operar en dos modos distintos: verificación facial (también conocido como autenticación) e identificación facial (también conocido como reconocimiento) [4]. La verificación es una comparación uno contra uno (1-1) entre una imagen nueva (*query*) que clama cierta identidad y una conocida por el sistema (*template* o *target*) cuya identidad dice ser la imagen en *query*, la identificación es una comparación uno contra varios donde



(a) Sistema de reconocimiento facial que realiza la tarea de la verificación.



(b) Sistema de reconocimiento facial que realiza la tarea de identificación.

Figura 1.1: Modos de trabajo posibles para un sistema de reconocimiento facial.

se trata de determinar la identidad de la imagen en *query* a partir de las imágenes en un conjunto conocido, en otras palabras, la salida de un sistema de verificación deber ser si la imagen en *query* y la imagen en target pertenecen a la misma persona y en un sistema de identificación la salida debe ser la identidad de la persona en el *query*, en la Figura 1.1a se ingresan dos imágenes distintas que dicen ser Mark, la tarea del sistema es decidir si ambas imágenes pertenecen a la identidad que claman ser, por otra parte, el sistema en la Figura 1.1b recibe imágenes como entrada y su tarea es determinar la identidad de las personas en las capturas.

Un sistema de reconocimiento facial se compone de 4 módulos: detección del rostro, alineamiento facial, extracción de características y comparación de características, en la práctica el flujo de trabajo dentro de un sistema de reconocimiento facial se basa en recorrer de izquierda a derecha los módulos como se muestran en la Figura 1.2, es decir, a una imagen de entrada primero se le realiza la detección del rostro para segmentar elementos que no aportan información relevante para la tarea de reconocimiento como lo son el fondo, el cabello, entre otros, el paso siguiente es el alineamiento, aquí se busca normalizar los rostros, comúnmente la normalización se realiza con respecto a propiedades geométricas como la pose y el tamaño y propiedades fotométricas como la iluminación y escala de grises. Después se requiere la extracción de las características, en este paso se obtiene la información de cada imagen que permitirá distinguir entre rostros de diferentes personas, finalmente en la comparación los vectores de características extraídos de una imagen de entrada son comparados con los rostros de una base de datos para devolver la identidad del sujeto en la imagen de entrada.

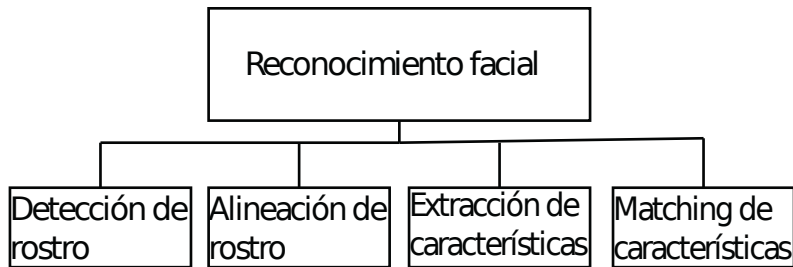


Figura 1.2: Módulos de un sistema de reconocimiento facial

El primer sistema de reconocimiento facial automático fue desarrollado por [5], desde entonces los productos de investigación en este campo han evolucionado al grado de ofrecer reconocimiento en tiempo real y con altos grados de confiabilidad, los algoritmos de reconocimiento facial pueden dividirse en 3 grandes categorías [6]:

- 1 Métodos holísticos: Estos métodos utilizan toda la región del rostro como entrada.
- 2 Basados en características o métodos locales: En estos métodos se extraen características de regiones de rostro como los ojos, nariz y boca para obtener información y se alimenta al clasificador.
- 3 Métodos híbridos: Combina la información de todo el rostro junto con características locales para realizar la clasificación.

Los métodos holísticos debido a que se requiere la información del rostro completo, suelen presentar dificultades cuando se presentan oclusiones o variaciones en la iluminación. El reconocimiento facial en 3D utiliza la información de profundidad del rostro que en los métodos tradicionales 2D no es considerada, algunos métodos holísticos en 3D han demostrado ser funcionales [7–9]. Los métodos basados en características locales tienen ventaja sobre los métodos holísticos debido a que son más robustos a variaciones de pose, iluminación o bajo oclusión [10], los principales descriptores locales en la literatura pueden agruparse en tres categorías: basados en puntos claves, basados en curvas y basados en superficies [11], la Tabla 1.1 muestra los principales descriptores en estas categorías. Los métodos basados en puntos clave obtienen puntos de interés basados en la superficie, el costo computacional de estos descriptores aumenta conforme se aumenta la cantidad de puntos de interés por lo que una parte importante en el desarrollo del descriptor es seleccionar los puntos más efectivos para la tarea de reconocimiento. Los métodos basados en curvas son más robustos respecto a los basados en puntos debido que aprovechan la información geométrica de la forma del rostro, por último, los descriptores basados en superficie obtienen información local de determinadas regiones del rostro que no se deforman durante una expresión facial.

Tabla 1.1: Principales categorías de descriptores locales.

Categoría	Tipo de descriptor
Keypoint-based	SIFT-like [12]
	Mesh-based [13]
	Landmark-based [14]
Curve-based	Countour-based [15]
	Profile-based [2]
Surface-based	LBP-based [16]
	Geometric feature-based [17]

1.3. Antecedentes

En sus comienzos, el RF fue tratado como un problema de reconocimiento de patrones en 2D, los primeros trabajos en los años 70s utilizaban atributos medibles del rostro como características y así se mantuvo mayormente durante los años 80s, fue hasta la década de 1990 que el interés en la investigación en RF aumentó considerablemente gracias a los avances en hardware más capaz de realizar procesamiento de manera más rápida y el creciente interés en las aplicaciones del RF. Los primeros 15 años de estudio del RF se enfocaron en realizar la tarea atacando problemas como la localización de un rostro dentro de una imagen o un video y extrayendo las características de zonas como la nariz, los ojos y la boca mientras que a la par se realizaron avances significativos en el desarrollo de clasificadores que pudieran realizar el reconocimiento de manera satisfactoria. Actualmente los avances en hardware y algoritmos de extracción y clasificación, permiten realizar tareas de verificación e identificación con resultados superiores al 90 % de exactitud en tiempos de ejecución tan reducidos que se considera en tiempo real. En los principios de los 90s, [18] presentó un sistema que realizaba reconocimiento en un tiempo casi instantáneo, en este trabajo se trata el problema como reconocimiento en 2 dimensiones, el sistema proyecta los rostros en las imágenes de entrada a un espacio de características de los rostros de las personas conocidas por el sistema, posteriormente en [19] se propuso utilizar plantillas estándar para determinados elementos del rostro como la boca, los ojos y la nariz, las plantillas se deforman hasta ajustarse de mejor manera a los elementos del rostro, el parámetro resultante se utiliza como descriptor para el reconocimiento, en ambos casos las imágenes utilizadas consistieron en capturas en condiciones controladas. A principios del siglo XX la evolución en la investigación en RF continuo su paso, se comenzaron a ver más trabajos utilizando herramientas de aprendizaje como las *Support Vector Machines* (SVM) como en [20], también se desarrollaron descriptores utilizando patrones binarios [21, 22] así como

bases de datos y protocolos de evaluación para los sistemas de RF [23, 24].

La década siguiente aumentaron los trabajos basados en redes neuronales profundas [25–27] y muchos otros trabajos enfocados en desarrollar sistemas robustos a problemas inherentes al RF como las variaciones de pose e iluminación y oclusiones [28–30], en siguiente sección se discutirán algunos trabajos relevantes de la última década, haciendo enfoque especialmente en aquellos que se basan en el uso de características de tipo local.

1.4. Estado del arte

En esta sección se presentan algunos de los trabajos más relevantes en el campo del reconocimiento facial utilizando descriptores de características de tipo local ya que estos son los que presentan mayor robustez frente a variaciones de pose, expresiones faciales y oclusiones por lo que son más aptos para realizar las tareas de reconocimiento y verificación, los trabajos que se discutirán a continuación son del tipo que tratan el problema como un problema de reconocimiento de patrones en 3D. Los trabajos serán brevemente descritos y se hará mención de las ventajas y/o desventajas que puedan poseer, para mayor detalle en las metodologías se puede referir a la publicación correspondiente a cada trabajo. Para simplificar la descripción se hará uso de las categorías presentadas en la Tabla 1.1.

1.4.1. Basados en puntos clave

Estos sistemas detectan puntos de interés en el rostro, debido a que el costo computacional en estos métodos puede aumentar considerablemente conforme aumenta la cantidad de puntos, es importante escoger los puntos que mejor describan a un individuo para obtener un vector de características eficiente, hay dos pasos que son claves en este tipo de sistemas: 1) Detección de los puntos clave y 2) Descripción de las características.

En [12] los puntos SIFT se extrajeron de los mapas de forma, curvatura, curvatura media y curvatura Gaussiana. En 2016 se presentó un método basado en puntos clave para superficies tridimensionales [31] denominado *Keypoint-based Multiple Triangle Statistics* (KMTS) para realizar la representación de un rostro, de tal manera que sea robusto a expresiones faciales y variaciones de pose y se propone también realizar la clasificación basado en un *Two-Phased Weighted Collaborative Representation Classification* (TPWCRC), los experimentos para el método KMTS-TPWCRC se realizaron con 6 bases de datos distintas. En ese mismo año se propuso aplicar la detección de puntos *Scale-Invariant Feature Transform* (SIFT) en un mapa de forma piramidal [32], calcular los puntos SIFT en mapas piramidales aumenta la confiabilidad respecto al extraerlos de texturas del rostro o mapas de formas. Una manera de detectar puntos clave sin utilizar SIFT se desarrolló por [13] combinando elementos de la transformada *Curvelet* en diferentes direcciones, el resultado fue una extracción de puntos en un menor tiempo que utilizando SIFT y con buena confiabilidad para la tarea de reconocimiento.

1.4.2. Basados en curvas

Utilizan conjuntos de curvas en el rostro como características, las curvas pueden ser de contornos o de perfiles, el trabajo de Lei *et al* [2] presenta un descriptor *Angular Radial Signature* con consiste en la obtención de curvas que salen de la punta de la nariz en diferentes direcciones hasta una distancia determinada, un *Kernel Principal Component Analysis* (KPCA) es entrenado para cada curva en determinada dirección y el *matching* se realiza por medio de una SVM, en el trabajo se estudia también la influencia de la cantidad de puntos y direcciones utilizadas. Otro método similar se obtuvo intersectando esferas concéntricas en un punto determinado dentro de un área local, la intersección de distintas esferas resulta en un perfil, esta representación se denomina *Rotation-invariant and Adjustable Integral Kernels* (RAIKs) [33]. Existen también trabajos que utilizan regiones específicas del rostro, como en [34], la región de la nariz se utiliza como objeto para el reconocimiento, la descripción se divide en 5 pasos y se obtienen los parches y curvas más adecuados para enfrentar expresiones faciales.

1.4.3. Basados en superficies

Utilizan características geométricas del rostro en zonas del rostro donde existe poca variación, un ejemplo que está basado en las características tipo *Local Binary Patterns* (LBP) se presenta en [16], donde se propone un método que es capaz de realizar el reconocimiento con partes faltantes de la imagen y expresiones, el descriptor LBP se utiliza en la malla triangular de la imagen. Una extensión del uso de descriptores de covarianza al reconocimiento facial se expone en [35], donde una representación del rostro contiene una cantidad m de puntos de referencia en la superficie, cada punto tiene una región de influencia que se caracteriza por la covarianza de las características geométricas. En [17] se utiliza un nuevo descriptor llamado *regional bounding spherical descriptor* (RBSR) con el que se segmenta una imagen en regiones y cada una de esas áreas es proyectada a una esfera para obtener el descriptor de cada región, los experimentos se realizaron en 3 bases de datos populares.

En la Tabla 1.2 se muestran los sistemas basados en métodos locales revisados previamente, cada uno de ellos posee ciertas ventajas y desventajas sobre otros. A continuación, también se revisarán brevemente los métodos que utilizan SVMs como herramienta para la clasificación de información facial en 3D.

1.4.4. Matching basado en SVM

El RF en 3D no aparece muy frecuentemente en este campo de estudio, sin embargo, hay trabajos importantes de los cuales se hará mención en esta sección. Hayat *et al* [36] utilizaron información de baja calidad capturada con un Kinect para proponer un método eficiente, los modelos del rostro se representan en un *Riemannian manifold*, como elemento para el *matching* se utilizó

Tabla 1.2: Métodos locales para el reconocimiento facial en 3D

Autor	Año	Descriptor	Categoría	Matching	Ventajas
Lei <i>et al</i> [2]	2014	ARS	<i>Curve-based</i>	SVM	Eficiente, expresión
Al-Osaimi [33]	2015	RAIKs	<i>Curve-based</i>	<i>Euclidean dist</i>	Expresión
Ming [17]	2015	RBSR	<i>Surface-based</i>	Regresión	Eficiente, pose
Lei <i>et al</i> [31]	2016	KMTS	<i>Keypoint-based</i>	TPWCRC	Eficiente, información parcial
Elaiwat <i>et al</i> [13]	2015	<i>Curvelet transform</i>	<i>Keypoint</i>	<i>Cosine dist</i>	Iluminación, expresión
Soltanpour and Wu [32]	2016	SIFT	<i>Keypoint-based</i>	<i>Histogram matching</i>	Expresión
Emambakhsh and Evans [34]	2016	Gabor-wavelet	<i>Curve-based</i>	<i>Mahalanobis dist</i>	Expresión
Werghi <i>et al</i> [16]	2016	LBP	<i>Surface-based</i>	<i>Cosine dist</i>	Expresión, información faltante.
Hariri <i>et al</i> [35]	2016	<i>Covariance descriptor</i>	<i>Surface-based</i>	<i>Geodesic dist</i>	Pose, expresión

Tabla 1.3: Sistemas de reconocimiento facial con *matching* basado en SVM con la RR reportada

Autor	Año	Descriptor	Kernel	Ventajas	RR(%)
Lei <i>et al</i>	2014	ARS	KPCA	Eficiente, expresión	96.7
Hiremath and Hiremath	2014	RT+PCA+ LDA	-	Eficiente, iluminación, pose	99.2
Hayat <i>et al</i>	2016	Riemannian manifold	Kernel Stein	Eficiente	97.44
Shi <i>et al</i>	2020	LBP	Lineal	Eficiente	96.83

una SVM en cada subgrupo utilizando un kernel Stein. Mas trabajo con SVMs para el reconocimiento facial se muestra en [37] donde se combinan el uso de la transformada Radon, PCA y *Linear Discriminant Analysis* (LDA) para representar las imágenes faciales y se toman las decisiones con un clasificador SVM. Shi *et al* [38] hacen uso de características LBP para extracción de características en imágenes de rango, mismas que posteriormente son utilizadas por una SVM para tomar las decisiones de verificación e identificación. La Tabla 1.3 muestra los métodos basados en SVM para clasificación con las ventajas que presentan así como la RR reportada en las publicaciones, mismas que servirán como referencia más adelante.

1.5. Descripción del problema

El rostro humano es un objeto que contiene una gran cantidad de información acerca de la identidad de la persona propietaria del rostro y la tarea del reconocimiento es algo que los humanos hacemos con naturalidad, sin embargo, realizar el reconocimiento a través de una computadora es una tarea que aun presenta dificultades debido a que se trata de distinguir entre objetos que pertenecen a una misma clase pero que son muy distintos entre ellos, eso además de los retos que se presentan respecto a las condiciones de captura de las imágenes que serán utilizadas por un sistema, todo el proceso debe realizarse en un tiempo corto de tal manera que el sistema pueda ser útil en la práctica. Los sistemas de reconocimiento facial que parten de imágenes faciales en 2D son vulnerables cuando hay una iluminación poco favorable, además la extracción de las características faciales no es del todo certera debido a la poca información biométrica que ofrecen, la información desaprovechada en el reconocimiento facial en 2D es la que hace vulnerables a estos sistemas también a reconocer incluso una fotografía como el sujeto auténtico. Aunque la

investigación en el reconocimiento facial ha tenido buenos resultados, sigue siendo débil contra expresiones y variaciones de iluminación. El desarrollo de sensores para la captura de imágenes en 3D abrió las puertas a la investigación en sistemas de reconocimiento facial en 3D y ha atraído la atención de los entusiastas en el campo de la visión por computadora. Las imágenes de rostros en 3D son más ricas en información biométrica que las convencionales imágenes en 2D, aportan información característica de cada rostro, que ha permitido a los investigadores desarrollar sistemas de reconocimiento facial más robustos a las expresiones y variaciones de iluminación y precisión mayores al 90 %, aumentando el factor de confiabilidad.

1.6. Justificación

La autenticación biométrica ha ganado importancia a lo largo de los años debido a la necesidad de identificar personas a partir alguna característica biofísica o de comportamiento, las aplicaciones en los campos de seguridad, psicología e incluso medicina plantean una gran oportunidad en el estudio de la identificación biométrica. El reconocimiento facial utiliza las características faciales sin necesidad de tener contacto directo con el sujeto de estudio por lo que ofrece un útil para identificar sujetos a distancia, el uso de información tridimensional del rostro hace que el sistema sea más robusto frente a situaciones con variaciones de iluminación, pose y expresiones faciales.

1.7. Hipótesis

La combinación de características locales y herramientas de aprendizaje SVM, es posible desarrollar un sistema de reconocimiento facial en 3D que sea capaz de discriminar entre rostros con expresiones faciales y variaciones de iluminación obteniendo resultados comparables con métodos del estado del arte en las tareas de verificación e identificación.

1.8. Objetivos

1.8.1. Objetivo general

Implementar un sistema de reconocimiento facial en 3D que utilice descriptores locales tipo LBP y SVM para compararlo con métodos del estado del arte.

1.8.2. Objetivos específicos

- Implementar el sistema utilizando tecnologías y lenguajes de programación.

- Extraer características de una base de datos publica y verificar su costo computacional.
- Utilizar características locales para entrenar un clasificador.
- Probar con imágenes nuevas el desempeño del sistema para obtener las métricas finales.
- Comparar los resultados con métodos del estado del arte para entender la efectividad de los algoritmos.

Dirección General de Bibliotecas UAQ

Marco teórico

2.1. Datos linealmente separables

En un plano bidimensional, un conjunto de datos es linealmente separable si existe una línea recta que pueda dividir la información en dos conjuntos o clases distintas, para entenderlo de otra manera se puede tomar una compuerta lógica OR cuya tabla de verdad se muestra en la Tabla 2.1, los valores en **A** y **B** corresponden a las posibles entradas y **O** la salida para una determinada combinación en las entradas. Si estos valores se pasan a un plano bidimensional, se observa claramente que es posible trazar una línea recta que separe los valores de 0 y 1 en la salida, en la Figura 2.1 se muestra la separación.

De igual forma es posible visualizar la separabilidad lineal de un conjunto de datos en un espacio tridimensional, en este caso la separación se realiza con un plano de dos dimensiones

Tabla 2.1: Tabla de verdad de una compuerta lógica OR

A	B	O
0	0	0
0	1	1
1	1	1
1	0	1

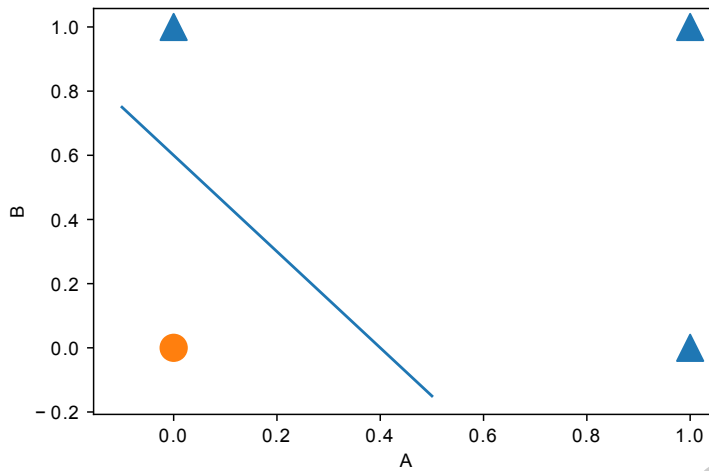


Figura 2.1: Representación en el plano de la tabla de verdad de una compuerta OR, la línea separa las salidas 0 y 1.

como se observa en la Figura 2.2. De manera general se dice que un conjunto de datos etiquetados

$$(y_1, \mathbf{x}_1), \dots, (y_l, \mathbf{x}_l), \quad y_i \in \{-1, 1\} \quad (2.1)$$

son linealmente separables si existe un vector \mathbf{w} y un escalar b de tal forma que se cumplan las desigualdades [39]

$$\begin{aligned} \mathbf{w} \cdot \mathbf{x}_i + b &\geq 1 & \text{si } y_i &= 1 \\ \mathbf{w} \cdot \mathbf{x}_i + b &\leq -1 & \text{si } y_i &= -1 \end{aligned} \quad (2.2)$$

2.2. Hiperplanos

2.2.1. ¿Qué es un hiperplano?

En geometría, un hiperplano es un subespacio con de una dimensión menor que el espacio en el que se encuentra y lo divide en dos secciones [40,41]. Para definir la ecuación del hiperplano se hará referencia a la ecuación de la línea recta

$$x_2 = ax_1 + b \quad (2.3)$$

donde a es la pendiente de la recta y b es el punto donde la recta corta el eje de las ordenadas, pasando todos los elementos en 2.3 al lado izquierdo de la ecuación se obtiene

$$ax_1 - x_2 + b = 0 \quad (2.4)$$

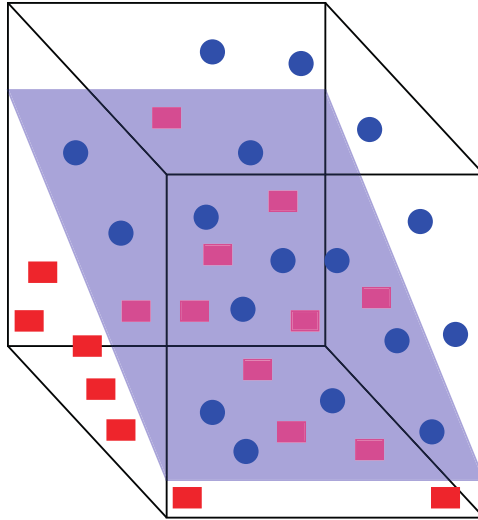


Figura 2.2: Datos linealmente separables en un espacio tridimensional, un plano de dos dimensiones separa los datos de dos clases diferentes

si se definen 2 vectores $\mathbf{w} = (x_1, x_2)$ y $\mathbf{x} = (a, -1)$, la ecuación del hiperplano se define como

$$\mathbf{w} \cdot \mathbf{x} + b = 0 \quad (2.5)$$

misma que puede extenderse a vectores (\mathbf{w}, \mathbf{x}) de cualquier dimensión [40]

NOTA: Si se realiza el análisis inverso, es decir, a partir de 2.5 llegar a la ecuación de la recta 2.3, se observa que, en la ecuación del hiperplano, b no siempre se refiere a la intersección con el eje vertical.

2.2.2. Clasificación de datos a partir de un hiperplano

Un conjunto de datos linealmente separables puede ser dividido en dos clases por medio de un hiperplano, en la Figura 2.3a se encuentra un hiperplano con $\mathbf{w} = (0,75, 1)$ y $b = -9$, que separa los dos tipos de datos en el espacio. Para clasificar un punto nuevo \mathbf{x}_i con el hiperplano dado, los valores de \mathbf{w} y \mathbf{x}_i se evalúan en la Ecuación 2.5. Si se ingresa un punto $\mathbf{x}_1 = (3, 1)$, tras realizar la evaluación $(0,75, 1) \cdot (3, 1) - 9 = -5,75$, por lo que de acuerdo a las condiciones 2.2 $y_1 = -1$. Si de la misma manera se considera un punto $\mathbf{x}_2 = (10, 3)$, el resultado de la evaluación es un valor mayor a 1, por lo que la etiqueta $y_2 = 1$. En la figura 2.3b se observa que un conjunto de puntos linealmente separable puede ser dividido en dos por una cantidad infinita de posibles hiperplanos, entonces ¿cuál es el mejor hiperplano para realizar la separación?.

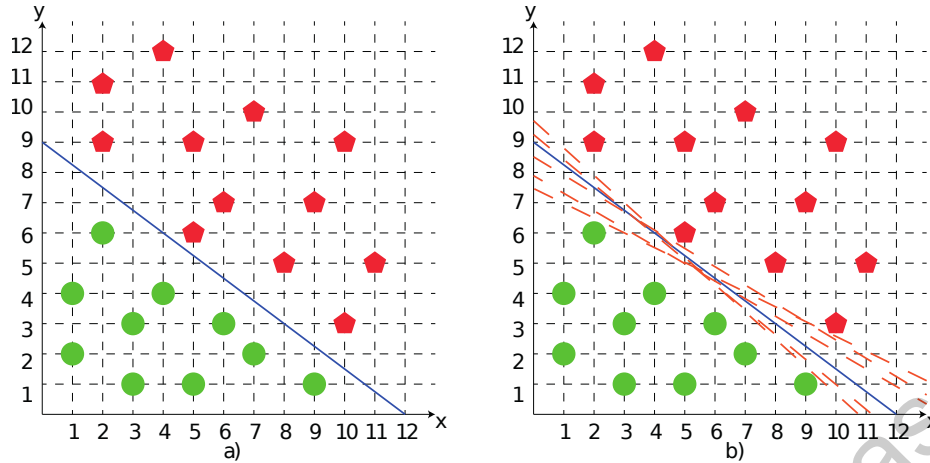


Figura 2.3: Un conjunto de datos linealmente separable puede ser dividido por una cantidad infinita de hiperplanos distintos

2.3. Support Vector Machines

Las SVMs son sistemas de aprendizaje propuestos por Vladimir Vapnik cuyo desarrollo tomó más de 30 años [42], fue completado en tres etapas y “son herramientas poderosas para clasificación” [43] ya que poseen una gran capacidad de generalización. Las SVMs, atacan el problema de la clasificación de datos a partir de un hiperplano por medio de la estimación de un hiperplano óptimo, mismo que separa los datos con una distancia o margen máximo entre los datos más cercanos al hiperplano y este mismo. En la Figura 2.4 se observa un ejemplo del objetivo de la SVM, el hiperplano separa ambas clases con una distancia máxima, los vectores que se encuentran sobre el margen se denominan vectores de soporte, las SVMs son sistemas con una fuerte capacidad de aprendizaje aún con poca información de entrada. En esta sección se revisará la teoría de las SVMs desde el caso más sencillo.

2.3.1. Hiperplano del margen máximo

El concepto del hiperplano óptimo fue introducido por [39] para separar datos sin errores, es el caso de la SVM más simple y también es el menos usado debido a las características que deben poseer los datos. El hiperplano óptimo se definió como una modificación del propuesto en 2.5

$$\mathbf{w}_0 \cdot \mathbf{x} + b_0 = 0 \quad (2.6)$$

y es el único que separa los datos con un margen máximo, el margen máximo determina la distancia máxima entre las proyecciones de dos vectores del conjunto de entrenamiento pertenecientes a diferentes clases. En 2.6, \mathbf{w}_0 es un vector perpendicular al margen sobre el que se proyecta el vector \mathbf{x} para determinar si se encuentra del lado derecho o izquierdo del margen para asignar la

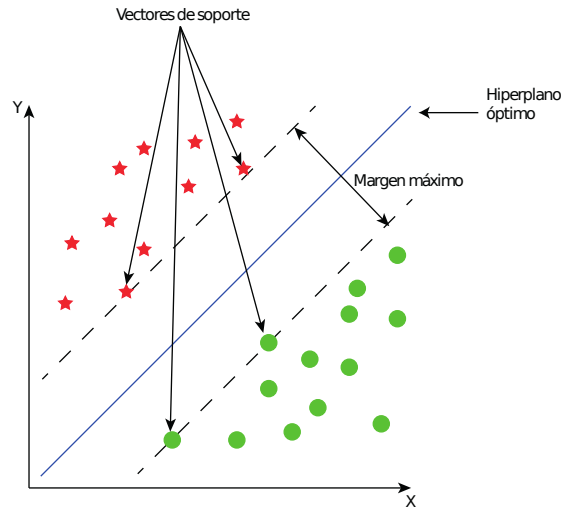


Figura 2.4: En una SVM, se separan los datos con un hiperplano óptimo, el cual maximiza el margen con los datos.

etiqueta, el vector \mathbf{w}_0 puede ser de cualquier longitud, por lo que es necesario establecer condiciones, por lo que las condiciones 2.2 se reescriben de la siguiente manera

$$y_i(\mathbf{w}_0 \cdot \mathbf{x} + b) \geq 1 \quad (2.7)$$

donde $y_i \in \{-1, 1\}$. Los vectores que se encuentran sobre el margen son aquellos para los que se cumple.

$$y_i(\mathbf{w}_0 \cdot \mathbf{x} + b) = 1 \quad (2.8)$$

El ancho del margen se puede entonces calcular con la diferencia entre un vector en el margen positivo y otro en el margen negativo, como se muestra en la Figura 2.5, el vector $(\mathbf{x}_1 - \mathbf{x}_2)$ se proyecta sobre el vector unitario de \mathbf{w}_0 , el vector proyectado será el ancho del margen.

$$margen = (\mathbf{x}_1 - \mathbf{x}_2) \cdot \frac{\mathbf{w}_0}{\|\mathbf{w}_0\|} \quad (2.9)$$

Si \mathbf{x}_1 es una muestra positiva y \mathbf{x}_2 es una muestra negativa, la ecuación 2.8 se puede reescribir para muestras positivas y negativas como

$$\begin{aligned} 1(\mathbf{w}_0 \cdot \mathbf{x}_1 + b) &= 1 \\ -1(\mathbf{w}_0 \cdot \mathbf{x}_2 + b) &= 1 \end{aligned} \quad (2.10)$$

despejando los vectores \mathbf{x}

$$\begin{aligned} \mathbf{x}_1 &= \frac{1 - b}{\mathbf{w}_0} \\ -\mathbf{x}_2 &= \frac{1 + b}{\mathbf{w}_0} \end{aligned} \quad (2.11)$$

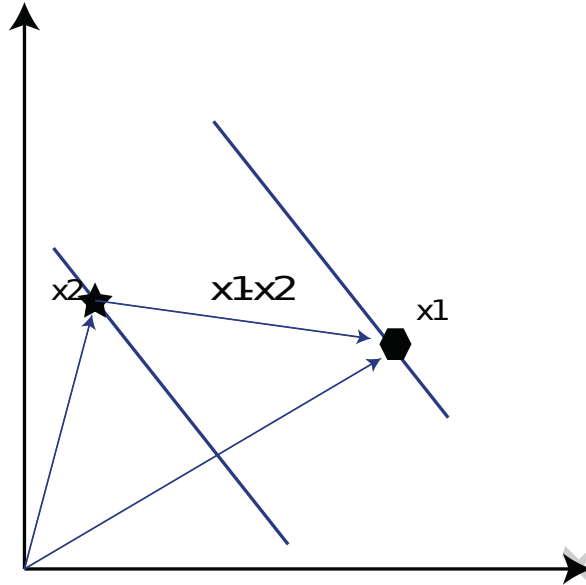


Figura 2.5: Diferencia entre vectores de soporte para calcular el margen

estos valores pueden sustituirse en la ecuación del margen 2.9

$$\text{margen} = \left(\frac{1-b}{\mathbf{w}_0} + \frac{1+b}{\mathbf{w}_0} \right) \cdot \frac{\mathbf{w}_0}{\|\mathbf{w}_0\|} \quad (2.12)$$

de esta manera el ancho del margen se queda de la siguiente manera

$$\text{margen} = \frac{2}{\|\mathbf{w}_0\|} \quad (2.13)$$

el objetivo es maximizar 2.13, lo que quiere decir que se debe minimizar $\|\mathbf{w}_0\|$, esta minimización se puede reescribir como

$$\min \frac{1}{2} \|\mathbf{w}_0\|^2 \quad (2.14)$$

el hiperplano para el cual $\mathbf{w}_0 \cdot \mathbf{w}_0$ es mínimo es el único para el cual se cumplen las condiciones 2.7. Para lograr la optimización sin tener que preocuparse por las condiciones, se utiliza la optimización por Lagrange.

$$L(\mathbf{w}_0, b, \mathbf{A}) = \frac{1}{2} \|\mathbf{w}_0\|^2 - \sum_{i=1}^l \alpha_i [y_i (\mathbf{w}_0 \cdot \mathbf{x}_i + b) - 1] \quad (2.15)$$

donde \mathbf{A} es un vector de coeficientes no negativos de Lagrange y se calculan las derivadas parciales

de L respecto de \mathbf{w} y b y se igualan a cero.

$$\frac{\partial L}{\partial \mathbf{w}_0} = \frac{1}{2} - \sum_{i=1}^l \alpha_i y_i \mathbf{x}_i = 0 \quad (2.16)$$

$$\mathbf{w}_0 = \sum_{i=1}^l \alpha_i y_i \mathbf{x}_i \quad (2.17)$$

$$\frac{\partial L}{\partial b} = - \sum_{i=1}^l \alpha_i y_i = 0 \quad (2.18)$$

$$\sum_{i=1}^l \alpha_i y_i = 0 \quad (2.19)$$

haciendo uso de 2.17 y 2.19 se pueden reescribir 2.15 como

$$L = \frac{1}{2} \left(\sum_{i=1}^n \alpha_i y_i \mathbf{x}_i \right) \cdot \left(\sum_{j=1}^n \alpha_j y_j \mathbf{x}_j \right) - \left(\sum_{i=1}^l \alpha_i y_i \right) \cdot \left(\sum_{j=1}^l \alpha_j y_j \mathbf{x}_i \cdot \mathbf{x}_j \right) - \sum_{i=1}^l \alpha_i y_i b + \sum_{i=1}^l \alpha_i \quad (2.20)$$

$$L = \sum_{i=1}^l \alpha_i + \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j - \sum_{j=1}^l \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \quad (2.21)$$

$$L = \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \quad (2.22)$$

de 2.22 se observa el problema de optimización depende únicamente del producto interno entre pares de muestras. De esta forma, el clasificador puede obtenerse maximizando 2.22 sujeto a 2.19.

$$y(\mathbf{x}) = \sum_{i=1}^l \alpha_i y_i \mathbf{x}_i \cdot \mathbf{x} + b \quad (2.23)$$

la etiqueta de un nuevo punto \mathbf{x} se obtiene tomando el signo resultante de 2.23.

2.3.2. Hiperplano de margen suave

En 1995, Vapnik introdujo una modificación al problema, en el nuevo planteamiento no se pretende separar los datos sin errores, sino que se tiene permitida una cantidad de errores de clasificación que se considera aceptable, para esto se introdujo una nueva variable ξ , de manera que las condiciones 2.7 se reescriben como

$$y_i(\mathbf{w}_0 \cdot \mathbf{x} + b) \geq 1 - \xi_i, \quad i = 1, \dots, l, \quad (2.24)$$

$$\xi_i \geq 0, \quad i = 1, \dots, l. \quad (2.25)$$

Minimizando ξ se pueden encontrar el conjunto de puntos en el entrenamiento que representan errores de clasificación, excluyendo estos puntos del conjunto de entrenamiento se puede construir un hiperplano óptimo sin errores con los puntos restantes en el conjunto de entrenamiento. Bajo las nuevas condiciones, la función objetivo 2.14 se convierte en

$$\min \frac{1}{2} \|\mathbf{w}_0\|^2 + C \sum_{i=1}^l \xi_i \quad (2.26)$$

donde C es una variable de penalización o regularización, para un valor de C suficientemente grande se determina el hiperplano que minimiza los errores de clasificación en el entrenamiento y separándolos del resto para separarlos con un hiperplano óptimo, la Figura 2.6 muestra el efecto entre escoger valores pequeños o grandes como penalización, el valor adecuado de C varía dependiendo del conjunto de datos con que se esté trabajando. De la misma forma que con el hiperplano óptimo, se puede utilizar optimización de Lagrange para obtener el problema dual de 2.26

$$\begin{aligned} \max \quad & \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \\ \text{s.a.} \quad & 0 \leq \alpha_i \leq C \\ & \sum_{i=1}^l \alpha_i y_i = 0 \end{aligned} \quad (2.27)$$

2.3.3. Caso no separable linealmente

Hasta ahora se han revisado la separación de datos por medio de un hiperplano en su espacio original. Sin embargo, en pocas ocasiones los datos son linealmente separables, por lo que no es posible trazar una línea recta para separarlos y minimizar la cantidad de errores de clasificación, en estos casos es necesario transformar los datos de un espacio n -dimensional a un espacio N -dimensional ($n < N$) donde sí sea posible una separación lineal

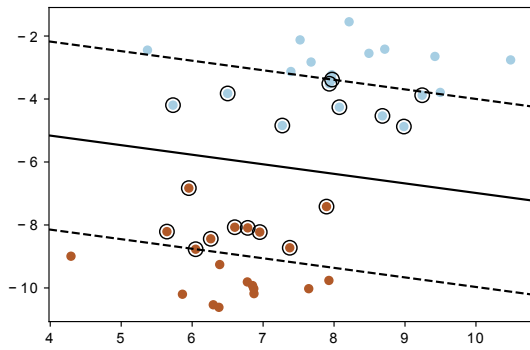
$$\phi : R^n \rightarrow R^N \quad (2.28)$$

y ϕ es la transformación que se mapea en todos los puntos del conjunto de entrenamiento

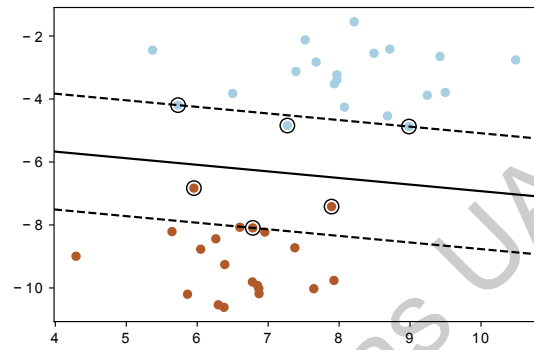
$$\phi(\mathbf{x}_i) = \phi(\mathbf{x}_1), \phi(\mathbf{x}_2), \dots, \phi(\mathbf{x}_l), \quad i = 1, \dots, l. \quad (2.29)$$

Como se vio anteriormente, el vector \mathbf{w}_0 se puede escribir como la combinación lineal de los vectores de soporte, pero esta vez en el nuevo espacio

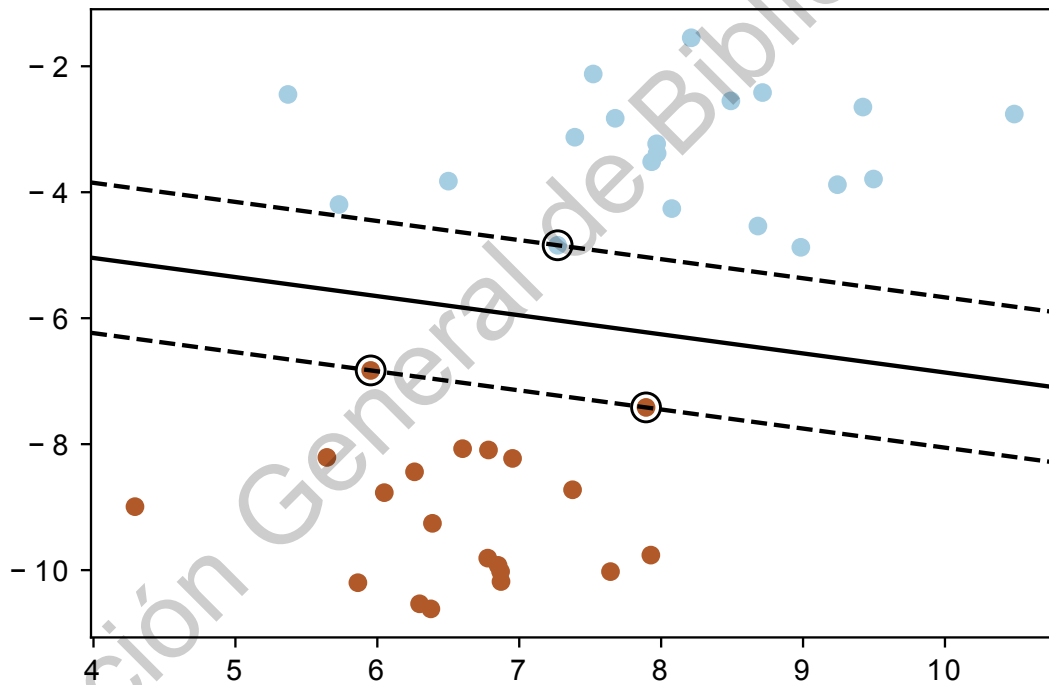
$$\mathbf{w}_0 = \sum_{i=1}^l \alpha_i y_i \phi(\mathbf{x}_i) \quad (2.30)$$



(a) $C = 0,01$



(b) $C = 0,5$



(c) $C = 10$

Figura 2.6: Efecto de diferentes valores para C , una C pequeña resulta en un margen más ancho, por otro lado, utilizar una C más grande dará un margen más pequeño que acepta menos errores.

así, la hipótesis de la SVM se puede reescribir como

$$y(\mathbf{x}) = \sum_{i=1}^l \alpha_i y_i \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}) + b \quad (2.31)$$

$$y(\mathbf{x}) = \sum_{i=1}^l \alpha_i y_i K(\mathbf{x}_i, \mathbf{x} + b) \quad (2.32)$$

donde $K(\mathbf{u}, \mathbf{v}) = \phi(\mathbf{u}) \cdot \phi(\mathbf{v})$ es una función kernel.

$$\begin{aligned} \min \quad & \frac{1}{2} \|\mathbf{w}_0\|^2 + C \sum_{i=1}^l \xi_i \\ \text{s.a.} \quad & y_i(\mathbf{w}_0 \cdot \phi(\mathbf{x}_i) + b) \geq 1 - \xi_i \\ & \xi_i \geq 0. \end{aligned} \quad (2.33)$$

con $i = 1, \dots, l$. Utilizando el mismo método que en el hiperplano de margen suave, el problema de optimización dual se convierte en

$$\begin{aligned} \max \quad & \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \\ \text{s.a.} \quad & 0 \leq \alpha_i \leq C \\ & \sum_{i=1}^l \alpha_i y_i = 0 \end{aligned} \quad (2.34)$$

2.3.4. Funciones Kernel

Las funciones kernel permite mapear implícitamente los datos de entrada a un espacio de características para entrenar una maquina lineal en dicho espacio [41].

Definición 2.1 *Un kernel es una función K , tal que para todo $\mathbf{x}, \mathbf{z} \in X$*

$$K(\mathbf{x}, \mathbf{z}) = \langle \phi(\mathbf{x}) \cdot \phi(\mathbf{z}) \rangle$$

donde ϕ es un mapeo de X a un espacio de características F . [41]

$$\phi : \mathbf{x} \mapsto \phi(\mathbf{x}) \in F$$

En el ejemplo 2.1 también del texto de Cristianini muestra como no es necesario conocer la transformación sin explícitamente evaluar las coordenadas de las proyecciones entre dos puntos

Ejemplo 2.1 *Se considera un espacio de entrada bidimensional $X \subseteq \mathbb{R}^2$ con un mapeo*

$$\phi : \mathbf{x} = (x_1, x_2) \mapsto \phi(\mathbf{x}) = (x_1^2, x_2^2, \sqrt{2}x_1x_2) \in F = \mathbb{R}^3$$

El mapeo pasa los datos de un espacio bidimensional a un espacio tridimensional, este mapeo puede ser evaluado con el producto interno en el espacio F

$$\begin{aligned} \langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle &= \left\langle (x_1^2, x_2^2, \sqrt{2}x_1x_2), (z_1^2, z_2^2, \sqrt{2}z_1z_2) \right\rangle \\ &= x_1^2z_1^2 + x_2^2z_2^2 + 2x_1z_1x_2z_2 \\ &= (x_1z_1 + x_2z_2)^2 = \langle \mathbf{x}, \mathbf{z} \rangle^2 \end{aligned}$$

Por lo que la función

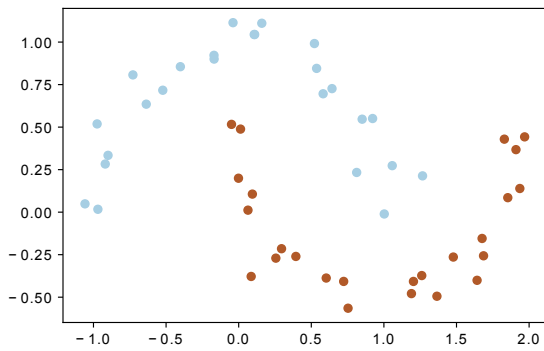
$$k(\mathbf{x}, \mathbf{z}) = \langle \mathbf{x}, \mathbf{z} \rangle^2$$

es una función kernel con F como su espacio de características.

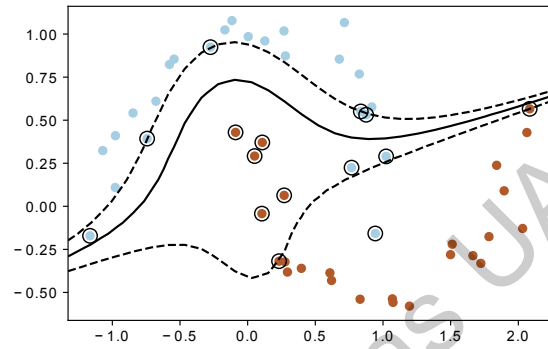
Para que una función kernel sea válida, esta debe cumplir con el teorema de Mercer. Algunas de las funciones kernel más comunes son:

- $K(\mathbf{x}_i, \mathbf{x}) = \mathbf{x}_i^T \mathbf{x}$ (Lineal)
- $K(\mathbf{x}_i, \mathbf{x}) = (\tau + \mathbf{x}_i^T \mathbf{x})^d$ (Polinomial de grado d)
- $K(\mathbf{x}_i, \mathbf{x}) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{x}_i\|^2}{2\sigma^2}\right)$ (Radial Basis Function)
- $K(\mathbf{x}_i, \mathbf{x}) = \tanh(k_1\mathbf{x}_i^T \mathbf{x} + k_2)$ (Multi-Layer Perceptron)

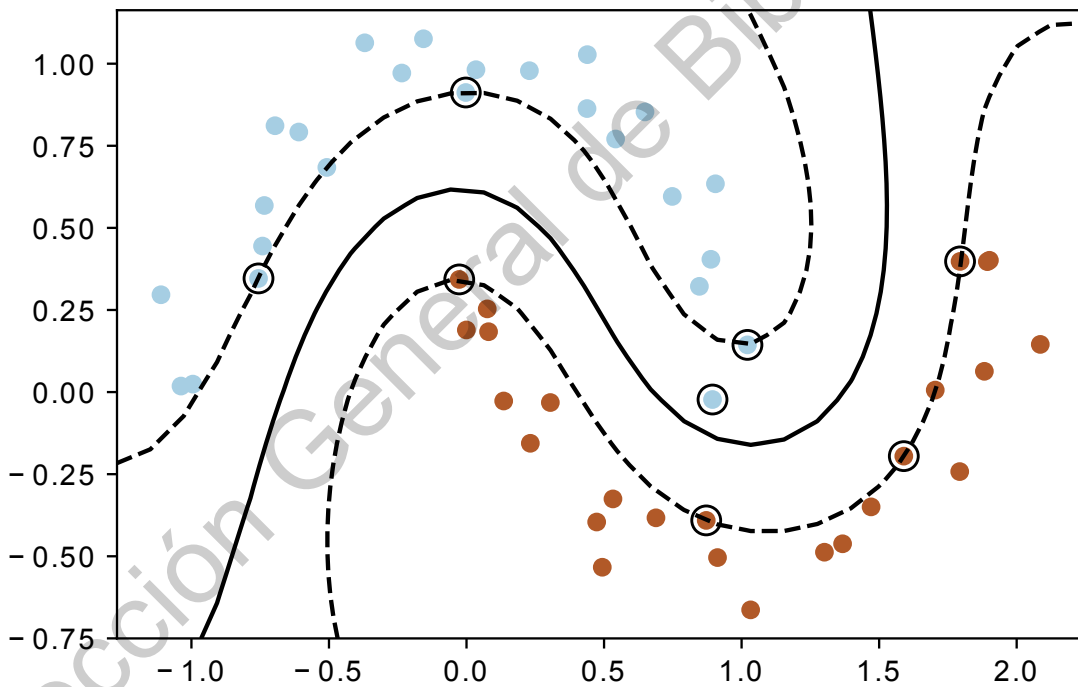
En el caso del kernel MLP, el teorema de Mercer no aplica para todos los valores posibles de k_1 y k_2 . La elección de un determinado kernel dependerá del conjunto de datos con el que se desea trabajar, debido a que no existe una forma de determinar analíticamente un kernel para utilizar con un conjunto de datos dado, la experimentación juega un papel importante. En la Figura 2.7a, se observa un conjunto de datos que claramente no puede ser dividido en dos por medio de una línea recta, o por lo menos no en el espacio en el que se encuentra, aplicando funciones kernel, se puede trazar una superficie de decisión en un espacio diferente, sin embargo, no cualquier transformación es apropiada para todo tipo de datos, como se observa en las Figuras 2.7b y 2.7c hay determinadas funciones kernel que pueden ajustarse mejor a determinado conjunto de datos que otras. La Figura 2.7c muestra que un kernel RBF ofrece una buena frontera de decisión con los datos originales, a comparación de la utilización del kernel polinomial que presenta una mayor cantidad de errores. Los hiperparámetros de las funciones kernel utilizadas en la Figura 2.7 así como el parámetro de regularización C no fueron ajustados por lo que no son los óptimos, pero son suficientemente buenos para expresar la idea general, más adelante se revisará cómo realizar el ajuste de estos hiperparámetros.



(a) Conjunto de datos no separable linealmente



(b) Kernel polinomial de tercer grado para la transformación



(c) Kernel RBF para la separación de los datos

Figura 2.7: El uso de un determinado kernel depende del conjunto de datos que se esté utilizando

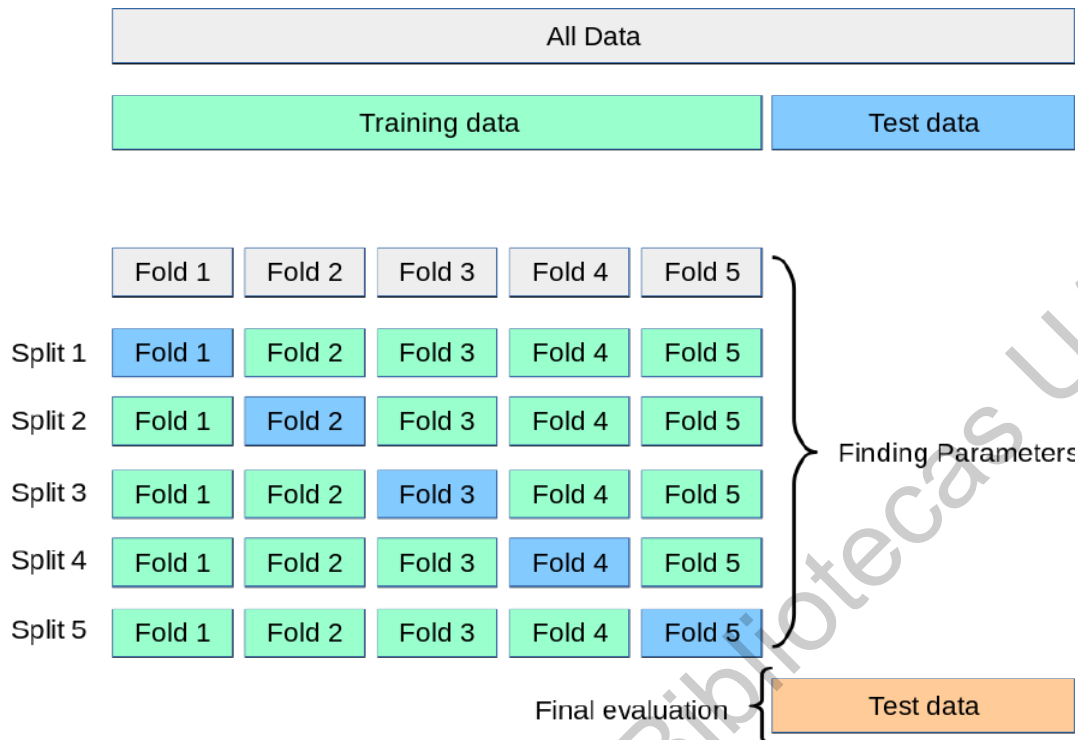


Figura 2.8: Ejemplo de validación cruzada (muestra tomada de la documentación de [1]), el conjunto de entrenamiento es dividido en 5-folds para la validación

2.3.5. Validación y ajuste de hiperparámetros

Los hiperparámetros son parámetros que no son aprendidos por el sistema, por lo tanto, deben definirse previamente a la fase de entrenamiento, los modelos de *Machine Learning* pasan por una fase de validación, en la cual, el desarrollador puede hacer cambios a los hiperparámetros del modelo dependiendo del desempeño obtenido, una manera muy popular de realizar la selección del modelo es la validación cruzada o *cross-validation* (CV), aquí, el conjunto de entrenamiento es dividido en k -folds donde $k - 1$ k -folds son usados para el entrenamiento y el resto para la validación, misma que retorna una métrica de desempeño como puede ser el *accuracy*, este proceso se realiza k -veces de manera que en cada ocasión se utilice un conjunto diferente para el entrenamiento y la prueba, esto puede ser mejor entendido observando la Figura 2.8, el desempeño esperado se encuentra promediando el desempeño de todas las pruebas, para realizar la búsqueda de los parámetros, un método común es la *Grid-search*, en esta se define un espacio de búsqueda o «malla» con los parámetros a probar y cada permutación de parámetros en la malla se utiliza en una CV para determinar el «mejor» conjunto de parámetros a utilizar en el modelo y se escribe «mejor» porque se trata de una búsqueda local, la desventaja de este método es que se trata de una búsqueda exhaustiva y puede tomar tiempos considerables de procesamiento para espacios grandes.

2.4. Local Normal Derivative Pattern

El descriptor LNDP [44] está basado en los descriptores LBP [45] y LDP [46], este descriptor de alto orden aprovecha la información direccional de cada una de las normales de una superficie, ofreciendo características de mejor calidad que los descriptores no direccionales.

2.4.1. Normales de superficie

El cálculo de las normales de superficie es el primer paso para la implementación del LNDP, la estimación de los vectores normales de la superficie busca recuperar la información que se pierde en el proceso de muestreo de la superficie con los sensores de rango, los vectores estimados son descriptores locales [47]. El enfoque más común consiste en encontrar un vector normal para cada punto en la superficie. Dado un conjunto de puntos $P = \{p_1, p_2, \dots, p_n\}$ en una nube $\mathbf{p}_i \in R^3$ donde $\mathbf{p}_i = [p_{ix}, p_{iy}, p_{iz}]^T$ representa las coordenadas 3D de cada punto y a cada punto le corresponde un vector normal $\mathbf{n}_i = [n_{ix}, n_{iy}, n_{iz}]^T$ se define un conjunto de k puntos vecinos para cada punto p_i , de manera que el conjunto de puntos vecinos a \mathbf{p}_i es $Q_i = \{q_{i1}, q_{i2}, \dots, q_{ik}\}$ y se define una matriz aumentada de vecinos

$$\mathbf{Q}_i^+ = [p_i, q_{i1}, q_{i2}, \dots, q_{ik}]^T \quad (2.35)$$

Para encontrar el vector normal \mathbf{n}_i al punto \mathbf{p}_i se ajusta un plano a la matriz 2.35 cuya distancia a cada punto en la matriz aumentada sea mínima, el vector normal al plano es también el vector normal a p_i . La Figura 2.9 muestra un conjunto de puntos \mathbf{Q}_i^+ a los que se ajusta un plano para la estimación de la normal. La combinación de un grafo k -Nearest-Neighbors (KNN) para el

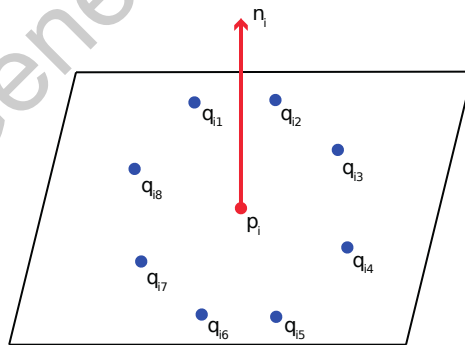


Figura 2.9: Vector normal a p_i con una vecindad de tamaño 8 estimado ajustando un plano al conjunto de puntos.

cálculo de los puntos vecinos y la técnica *PlanePCA* para el ajuste del plano tiene buenos resultados en términos de calidad y desempeño [47]. De esta manera la estimación del plano se obtiene

resolviendo

$$\min_{\mathbf{n}_i} \|[\mathbf{Q}_i^+ - \bar{\mathbf{Q}}_i^+] \mathbf{n}_i\| \quad (2.36)$$

donde $\bar{\mathbf{Q}}_i^+ = \mathbf{1}_k + \frac{1}{k+1}(\mathbf{p}_i + \sum_{j=1}^k \mathbf{q}_{ij})$, resolver 2.36 es equivalente a realizar el *Principal Component Analysis* en la matriz aumentada \mathbf{Q}_i^+ y tomar el componente con menor covarianza. En la figura 2.10 se muestra un ejemplo de las 3 componentes normales de la superficie, la imagen original (extrema izquierda) es parte de la base de datos *Texas 3DFRD* [48–50], como se observa, es claro que las imágenes normales ofrecen mejor información respecto a la imagen de rango original.



Figura 2.10: De izquierda a derecha, imagen de rango original de la base de datos *Texas 3DFRD*, imagen normal del componente x , imagen normal del componente y , imagen normal del componente z

2.4.2. LNNDP

Cando se tienen las componentes de los vectores normales de la superficie, se procede a los cálculos que darán forma a los vectores de características que podrán ser utilizados más adelante. El LNNDP es un descriptor multidireccional de alto orden, que aprovecha la información de n píxeles aledaños a un punto. Para un componente normal $N(P)$, donde p_i es un punto en la componente, las derivadas de primer orden en las direcciones $\alpha = \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}$ son

$$N'_{0^\circ}(p_i) = N(p_i) - N(Q_{i4}) \quad (2.37)$$

$$N'_{45^\circ}(p_i) = N(p_i) - N(Q_{i3}) \quad (2.38)$$

$$N'_{90^\circ}(p_i) = N(p_i) - N(Q_{i2}) \quad (2.39)$$

$$N'_{135^\circ}(p_i) = N(p_i) - N(Q_{i1}) \quad (2.40)$$

El direccional de segundo orden para una dirección α en un punto de la normal es

$$LNNDP_\alpha^2(p_i) = \{f(N'_\alpha(p_i), N'_\alpha(q_{i1})), f(N'_\alpha(p_i), N'_\alpha(q_{i2})), \dots, f(N'_\alpha(p_i), N'_\alpha(q_{i8}))\} \quad (2.41)$$

Donde $f(*, *)$ es una función binaria que determina el tipo de transición en el patrón, está definida como

$$f(N_{\alpha}^{n-1}(p_i), N_{\alpha}^{n-1}(q_{ij})) = \begin{cases} 0 & , si \quad N_{\alpha}^{n-1}(q_{ij}) \cdot N_{\alpha}^{n-1}(p_i) > 0 \\ 1 & , si \quad N_{\alpha}^{n-1}(q_{ij}) \cdot N_{alpha}^{n-1}(p_i) \leq 0 \end{cases} \quad j = 1, \dots, 8. \quad (2.42)$$

El LNDP de segundo orden, se obtiene concatenando los vectores de 8-bits de cada dirección α

$$LNDP^2(p_i) = \{LNDP_{\alpha}^2 | \alpha = 0^{\circ}, 45^{\circ}, 90^{\circ}, 135^{\circ}\} \quad (2.43)$$

De esta manera el vector resultante es una representación de 32-bits. Para extender el descriptor a un orden mayor de grado n , la forma general es la siguiente

$$LNDP_{\alpha}^n(p_i) = \{f(N_{\alpha}^{n-1}(p_i), N_{\alpha}^{n-1}(q_{i1})), f(N_{\alpha}^{n-1}(p_i), N_{\alpha}^{n-1}(q_{i2})), \dots, f(N_{\alpha}^{n-1}(p_i), N_{\alpha}^{n-1}(q_{i8}))\} \quad (2.44)$$

Para que las características LNDP representen características locales en el rostro, se utiliza un histograma espacial, la imagen original se divide en parches, en cada parche se calculan las características LNDP y se calcula el histograma para cada parche, el descriptor final se obtiene concatenando los histogramas de todos los parches de la imagen. En la figura 2.11 se muestran las 3 imágenes normales divididas en parches, para cada parche se forma un histograma. La concatenación de los histogramas de las imágenes normales forma el vector LNDP final.

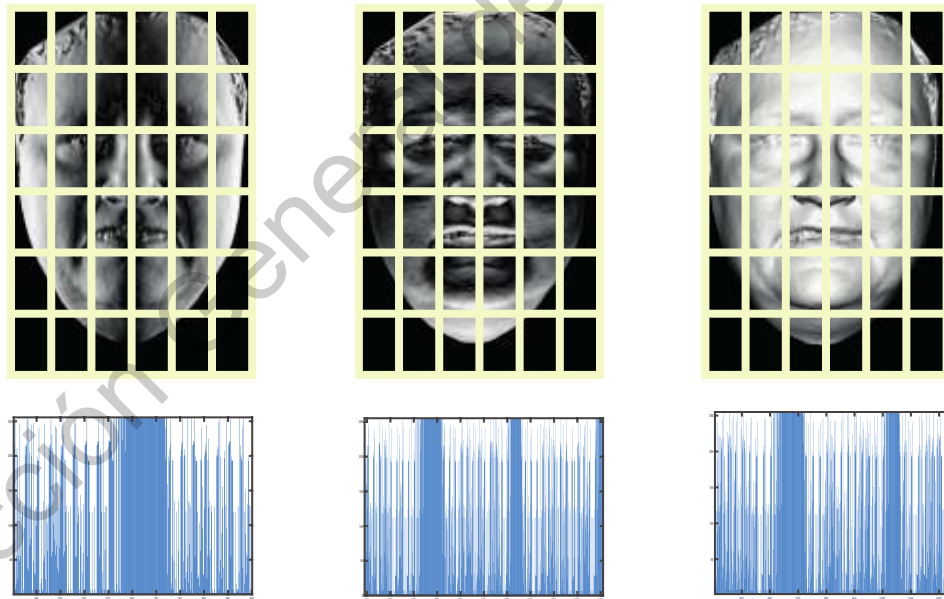


Figura 2.11: Imágenes normales en x, y y z divididas en parches, los histogramas de cada parche para cada imagen normal se concatenan, la concatenación de los histogramas de cada imagen normal forman el descriptor final.

El algoritmo 1 muestra el proceso para la obtención de las características LNDP para una imagen de rango.

Algorithm 1: Descriptor LNDP de grado n

Entrada: Imagen de rango 3D P , tamaño de parche $(k \times l), n$.

Salida : Vector de características LNDP.

```

1 para puntos en  $P$  hacer
2   |   Calcular las componentes normales  $N_x, N_y, N_z$ .
3 fin
4 para cada  $N$  hacer
5   |   Dividir en  $k \times l$  parches. para  $\alpha = 0, 45, 90, 135$  hacer
6     |   para cada parche hacer
7       |   para cada pixel en el parche de  $N$  hacer
8         |   |   Aplicar ecuación 2.44
9           |   fin
10        |   |   Construcción de histograma.
11        |   fin
12        |   |   Concatenar histogramas de diferentes parches.
13        |   fin
14        |   |   Concatenar histogramas de diferentes  $\alpha$ .
15 fin
16 Concatenar histogramas de cada  $N$ .
17 return  $nHLNDP$ 

```

2.4.3. Angular Radial Signature

El descriptor ARS fue propuesto por [2] que utiliza información geométrica del rostro de una manera eficiente, consiste en un conjunto de puntos que emanan de la nariz en diferentes direcciones, para esto Lei *et al* utilizaron una máscara binaria, el número de N de ARS es calculado en base al ángulo deseado entre cada franja de puntos θ (en radianes) con la siguiente ecuación

$$N = \pi/\theta + 1 \quad (2.45)$$

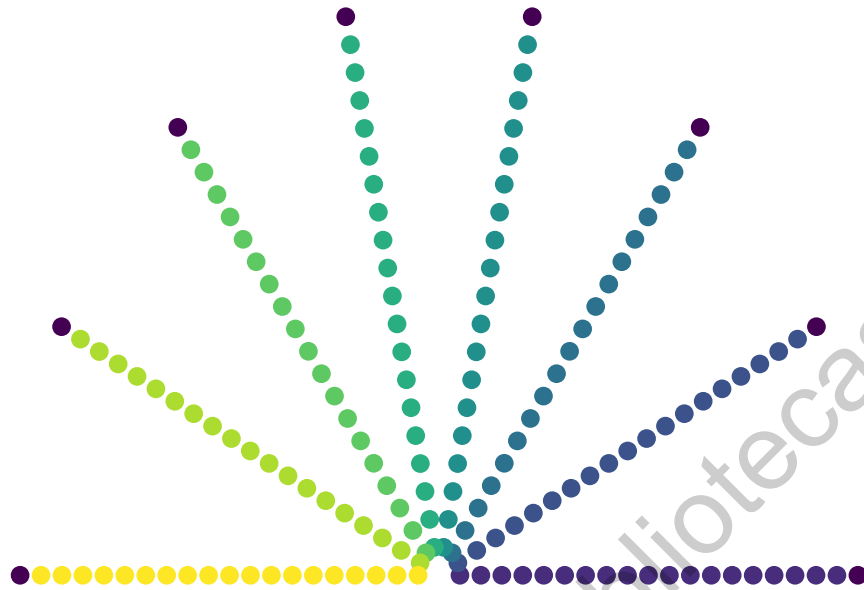


Figura 2.12: Máscara binaria utilizada para 8 ARS, (basada en la presentada por [2])

Cada conjunto de puntos en una θ dirección es una ARS, la Figura 2.12 muestra una máscara para 8 ARS, la máscara se coloca centrada en la punta de la nariz de la imagen facial y se realiza una interpolación bilineal para obtener los valores en cada punto de las ARS, en la misma figura también se observa que los puntos son extraídos únicamente de la parte superior del rostro, esto debido a que es la región semirrígida del rostro, lo que quiere decir que es la zona que sufre poca deformación en presencia de expresiones faciales, las Figuras 2.13a, 2.13b, 2.13c, 2.13d muestran la colocación de la máscaras con 8, 10, 12 y 17 ARS respectivamente sobre una imagen de rango, las ARS tienen su centro en la punta de la nariz del individuo, la ubicación de la nariz ha sido etiquetada manualmente por los desarrolladores de la base de datos. Para visualizar mejor esto se puede referir a la Figura 2.14 donde aparecen vistas en tres dimensiones de 17 ARS sobre la superficie del rostro de la Figura 2.13, aquí ya se puede apreciar que las curvas son significativas cuando se trata de obtener información característica de un individuo, cada uno de los puntos en estas curvas serán utilizados más adelante para la descripción del rostro.



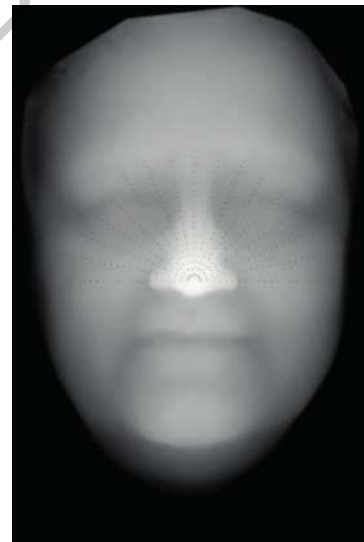
(a) 8 ARS



(b) 10 ARS



(c) 12 ARS



(d) 17 ARS

Figura 2.13: Máscara binaria con distintas cantidades de ARS sobre una imagen de rango de la base de datos

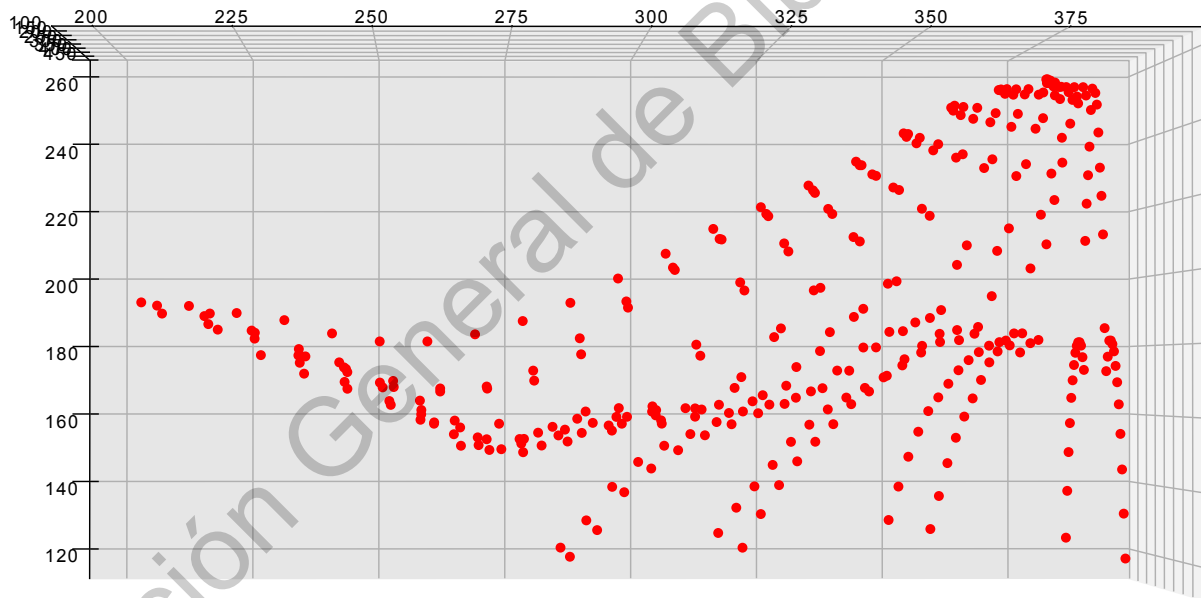
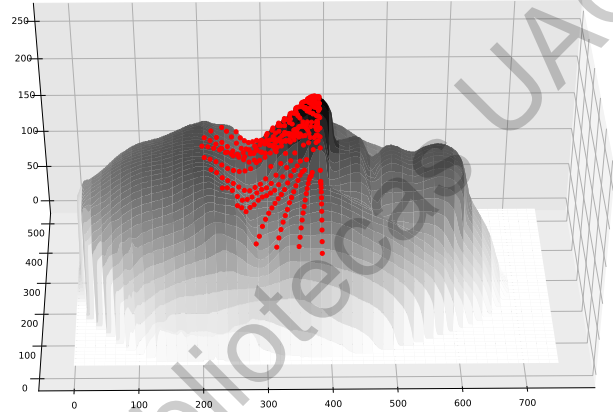
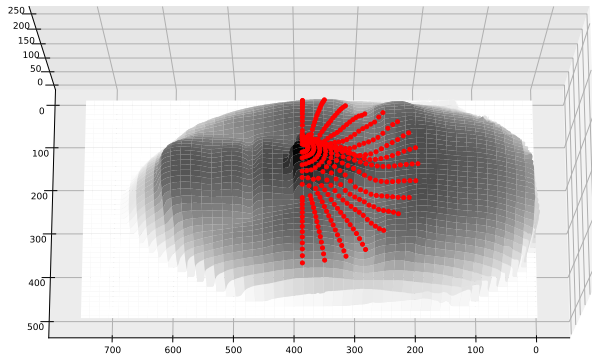


Figura 2.14: 17 ARS calculadas sobre una superficie tridimensional del rostro.

Metodología

3.1. Descripción general

El proceso a seguir para el desarrollo del sistema consta de dos módulos básicos para las imágenes pre-procesadas de la base de datos, el primero consiste en la extracción de las características y el segundo en el *matching* de las mismas, el descriptor propuesto combina las características ARS y LNDP, en el primer módulo se obtendrá la máscara binaria de una imagen, esta máscara se aplica a las imágenes normales correspondientes a la imagen de rango original, después, para cada punto en cada ARS se calculan las características LDP en 4 direcciones distintas, los vectores LDP de cada punto se concatenan en un histograma para cada ARS, lo siguiente es entrenar un KPCA para cada histograma, así es posible realizar el *matching* por medio de una máquina lineal. La Figura 3.1 muestra el flujo de trabajo en el sistema con los dos módulos mencionados previamente, como la base de datos utilizada se encuentra pre-procesada no es necesario hacer uso del resto de los módulos mencionados en la Sección 1, sin embargo también se revisará el efecto de extraer las características LDP en la imagen de rango original, así como omitir el entrenamiento de un KPCA para realizar el entrenamiento de la SVM con distintas funciones kernel. El Algoritmo 2 muestra los pasos a seguir para obtener las características propuestas mediante la combinación de características ARS y LDP, donde P es una imagen de rango que contiene la información 3D del rostro, n es el grado del descriptor LNDP, n_ARS es el número de direcciones ARS y α son las direcciones para calcular el diferencial.

3.2. Especificaciones

Todo lo relacionado a la programación en este trabajo se realizó en el lenguaje Python en su versión 3.7, los paquetes utilizados son *Numpy* [51], *Scipy* [52], *Matplotlib* [53], *Scikit-Learn* [1] y

Algorithm 2: Descriptor propuesto

Entrada: Imagen P , n , n_ARS, α

Salida : Vector de características

```
1 para cada  $p$  en  $P$  hacer
2   |   Calcular las componenter normales  $N_x, N_y, N_z$ 
3 fin
4 para cada  $N$  hacer
5   |   Calcular  $\theta$  utilizando 2.45
6   |   Aplicar máscara binaria
7   |   para cada  $a$  en  $\alpha = 0, 45, 90, 135$  hacer
8     |   para cada punto en la máscara hacer
9       |   Calcular 2.44
10    |   fin
11    |   Construcción de histograma para cada ARS
12    |   fin
13    |   Concatenación de histrogramas de diferentes  $\alpha$ 
14 fin
15 Concatenar histogramas para cada  $N$ 
16 return  $ARS+LNDP$ 
```

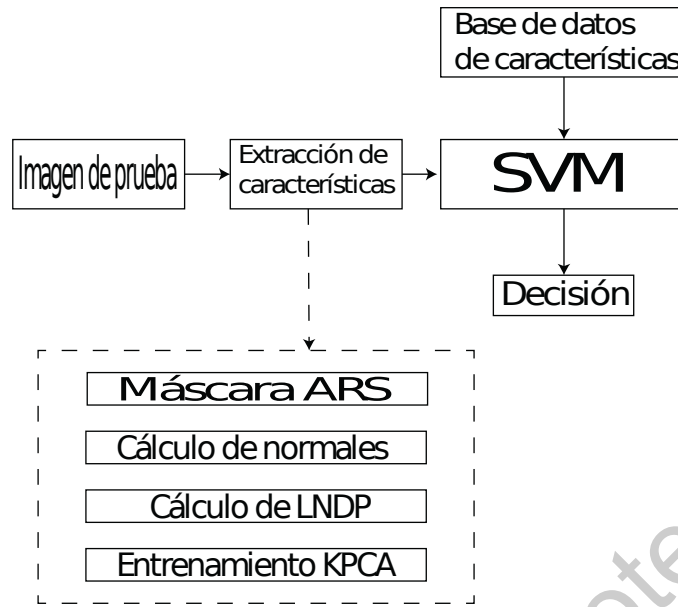


Figura 3.1: Flujo de trabajo para la toma de decisión en una imagen de prueba

Pandas [54] así como las librerías estándar como *math*, las especificaciones del equipo de cómputo utilizado son las siguientes:

- Procesador Intel core i5-10400K
- Memoria RAM 16 GB a 2666 Mhz
- Tarjeta gráfica Nvidia GeForce GTX 1660 Super

3.3. Base de datos

La base de datos utilizada es la *Texas 3DFR Database* [48–50], contiene información facial 2D y 3D de 118 humanos adultos de diferentes grupos étnicos, el número de imágenes por sujeto varía entre 1 y 89 capturas, los 118 sujetos se encuentran en un rango de edad de entre 22-75 años. Las imágenes fueron capturadas usando un sistema estéreo MU-2 fabricado por *3Q Technologies Ltd.* a una distancia conocida de la cámara. En esta base de datos las imágenes 2D y 3D de cada captura están perfectamente alineadas, más información sobre las condiciones de captura puede ser encontrada en el sitio web de la base de datos. Las imágenes de rango tienen un tamaño de 751x501 píxeles con resolución de 0.32 mm y cada valor de z está representado con un número de 8 bits, donde el 255 representa el punto más cercano a la cámara y 0 representa el fondo, las imágenes a color son de 751x501x3 píxeles en RGB a 8 bits. Las 1149 imágenes ya se encuentran pre-procesadas por lo que es posible dedicarse al trabajo únicamente del desarrollo de los algoritmos de reconocimiento, en la base de datos también se encuentran sujetos con expresiones neutrales

Tabla 3.1: Particiones propuestas por los creadores de la Texas 3DFRD

Partición	No. de sujetos	No. de imágenes		
		Neutral	Expresión	Total
Entrenamiento	12	228	132	360
Gallery	105	105	0	105
Probes	95	480	183	663
Resto	13	0	21	21

y no neutrales, ningún sujeto usa lentes o barba. La Figura 3.2 muestra una imagen de la base de datos en 3.2a, 3.2b, 3.2c y 3.2d son vistas del perfil que se puede obtener con los valores de cada pixel en la imagen de rango, como se ve en el perfil lateral 3.2c la punta de la nariz es el punto más lejano al fondo y corresponde al píxel más intenso en la imagen 3.2a.

En la base de datos también se incluyen 25 *fiducial points* para cada imagen, cada uno de los puntos fue etiquetado de manera manual, por lo que se asegura que están colocados correctamente, los puntos para las imágenes 2D y 3D están perfectamente alineados. De igual manera la base de datos tiene particiones propuestas descritas en la tabla 3.1 [49], estas particiones se diseñaron para seguir el protocolo de la *Face Recognition Grand Challenge* (FRGC) [24], los conjuntos *Gallery* y *Probes* conforman el conjunto de entrenamiento.

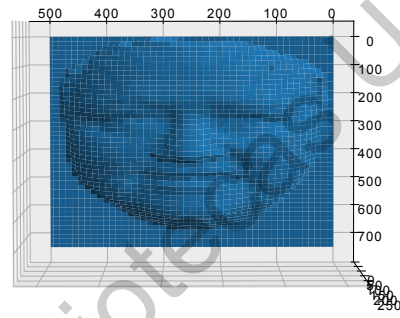
3.4. Descriptor

3.4.1. Obtención de las normales de superficie

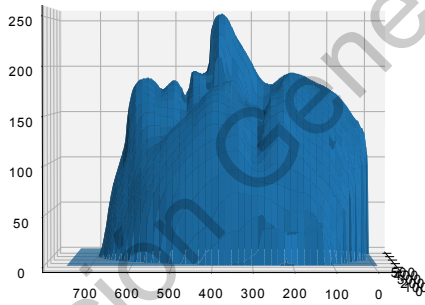
Para encontrar las componentes normales a un punto p_i en una nube de puntos $p_i \in R^3$, se requiere una matriz aumentada como 2.35, para encontrar los k vecinos más cercanos a p_i se utilizó el algoritmo KNN, después se resolvió 2.36 cada punto en la imagen de entrada. De acuerdo con el trabajo de Klasing *et al* [47], se espera que el método de ajuste de plano por PCA tenga un comportamiento similar a $O(d^3)$, donde d es el número de columnas en la matriz aumentada Q_i^+ . Conforme se aumenta la cantidad k de vecinos, la calidad de las normales aumenta de igual manera que el tiempo de cómputo, la Figura 3.3 muestra cómo el tiempo de ejecución aumenta conforme se aumenta la cantidad de vecinos utilizada para la estimación de las normales, se observa que el comportamiento sí se acerca al esperado, Aumentar el valor de k no solo afecta el tiempo de cómputo, también mejora la calidad de la estimación, por lo que las normales resultan más confiables conforme se aumenta el valor de k . De acuerdo con Klasing *et al*, un valor de $k =$



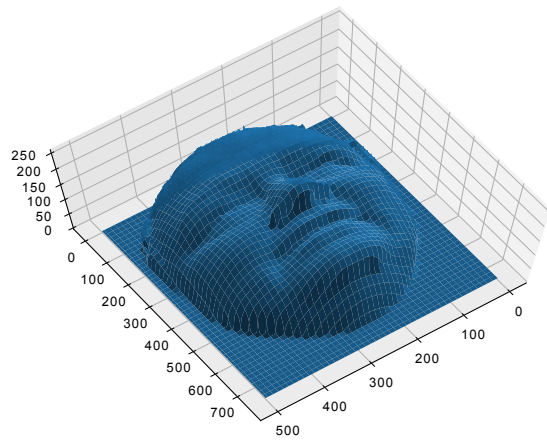
(a)



(b)



(c)



(d)

Figura 3.2: Imagen preprocesada de la base de datos así como varias vistas del perfil tridimensional que se obtienen de ella

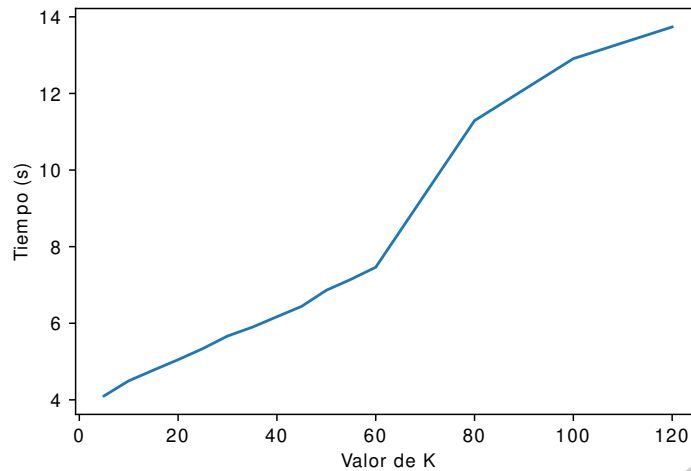


Figura 3.3: Tiempo de cálculo de normales de superficie con distintos valores de k

25 ofrece buena calidad en las componentes normales, la Figura 3.4 muestra las componentes normales estimadas con diferentes valores de k , visiblemente no hay cambios significativos a partir de 25 vecinos, por lo que este valor es una buena opción en cuanto a tiempo (5.33s) y calidad, los tiempos en la Figura 3.3 corresponden a todos los puntos en la imagen ($751 \times 501 = 376251$), por lo que eliminar información no útil (ej. El fondo negro o la zona no-rígida del rostro) de las imágenes puede reducir el tiempo de cómputo de las normales

3.4.2. Máscara binaria

De acuerdo con el trabajo de Lei [2], los mejores resultados se obtuvieron a partir de 17 ARSs, por lo que se este se usará como punto de partida, cada punto en una ARS tiene una separación de 3mm y las imágenes en la base de datos tienen una resolución de 0.3 mm en cada eje, esto se usó para generar un vector horizontal (ángulo cero) de 20 puntos con una separación de 10 pixeles, este vector se rotó a cada ángulo de las distintas ARS. La Figura 3.5 muestra que aun utilizando un numero alto de ARS el desempeño es muy bueno ya que se puede obtener la máscara de manera instantánea, por lo que un valor de 17 ARSs no presentara ningún inconveniente en cuanto a eficiencia se refiere, la máscara se aplica a cada imagen normal obtenida en el paso anterior. En la publicación de Lei *et al* también se propone aprender cada ARS con un kernel PCA, más se revisará si aprender las características con un KPCA también ofrece un mejor desempeño que sin el KPCA para las características propuestas.

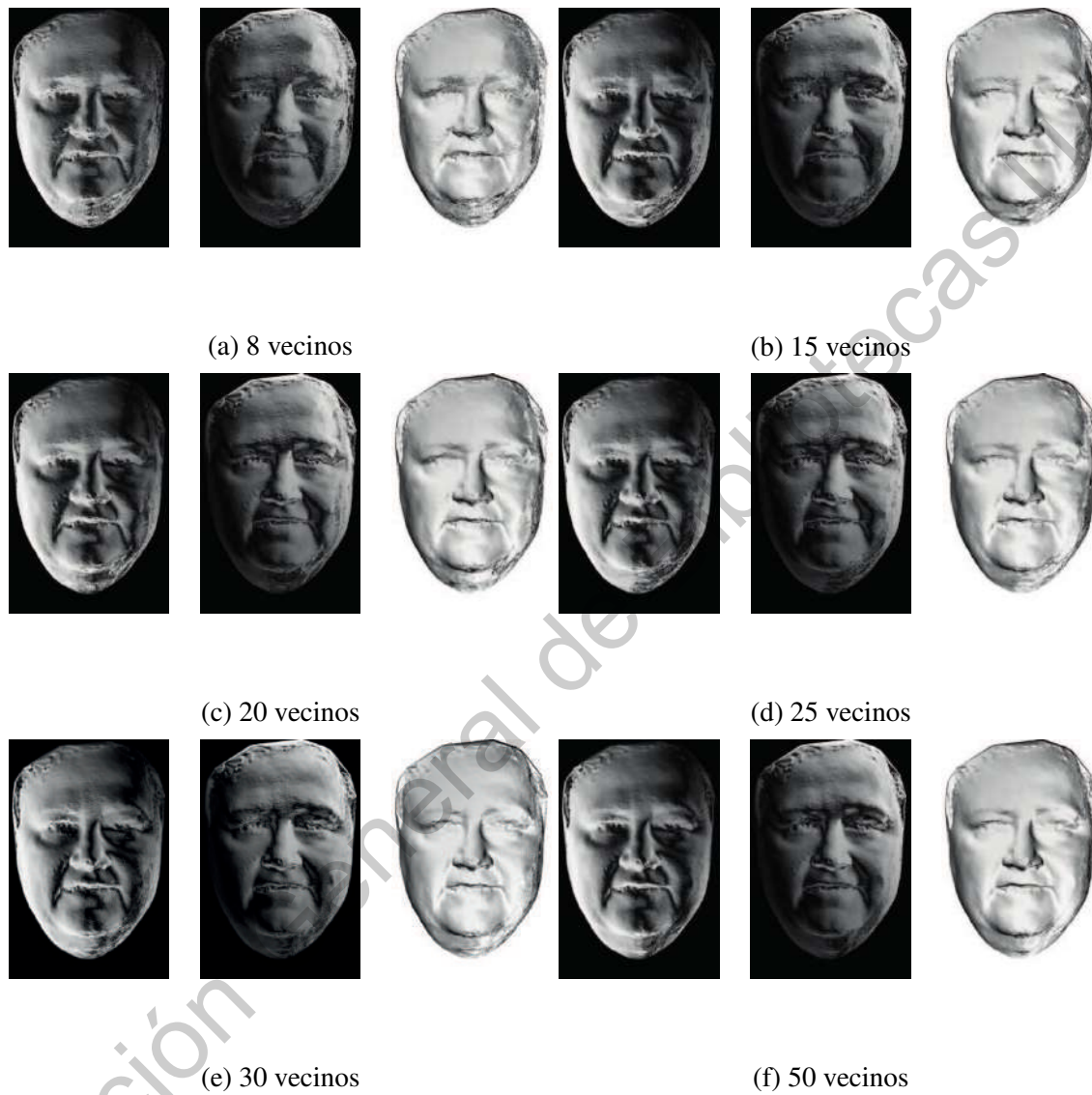


Figura 3.4: Imágenes normales con distintos valores de k para una imagen de rango, de izquierda a derecha son la normal en x , normal en y y normal en z

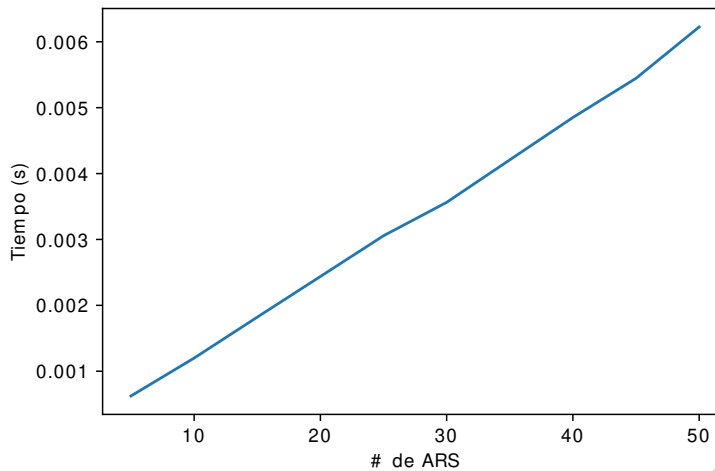


Figura 3.5: Tiempo de cálculo de la máscara binaria para diferentes cantidades de ARS

3.4.3. *Local Derivative Pattern*

Una vez obtenidos los puntos de la máscara binaria, el siguiente paso es obtener las características direccionales para cada punto en la máscara, en lugar de dividir las imágenes en parches como sugiere el Algoritmo 1 las características LDP en sus 4 direcciones se obtendrán de cada dirección de ARS por lo que se obtendrá un histograma para cada ARS, de acuerdo con la literatura [44,46] un descriptor de tipo LDP de grado 3 ofrece el mejor desempeño en clasificación, esto debido a que para descriptores de mayor grado se puede obtener ruido en lugar de información útil para identificar a una persona por lo que en este trabajo también se utilizara el mismo valor para el descriptor LDP, igualmente más adelante se revisará el efecto de utilizar el descriptor en la imagen de rango contra las imágenes normales. En la Figura 3.6 se comparan los tiempos de cálculo para características LNDP de grados 2 y 3, es claro que un descriptor de segundo grado ofrece un tiempo de procesamiento más largo, aunque para un número de ARS bajo, la diferencia se considera pequeña, conforme se aumenta el número de ARS la diferencia en el tiempo muestra una tendencia a hacerse más notoria, también se observa que para 17 ARS el tiempo se vuelve un poco largo para tareas de reconocimiento en tiempo real, por lo que se explorarán los efectos con números más pequeños de ARS así como aplicar el descriptor sin utilizar las componentes normales de la imagen original.

3.5. *Matching de características*

La decisión sobre la identidad de los sujetos en el conjunto de prueba se realizó con SVMs, hay dos formas de atacar este problema de clasificación diferentes metodologías, la primera fue la

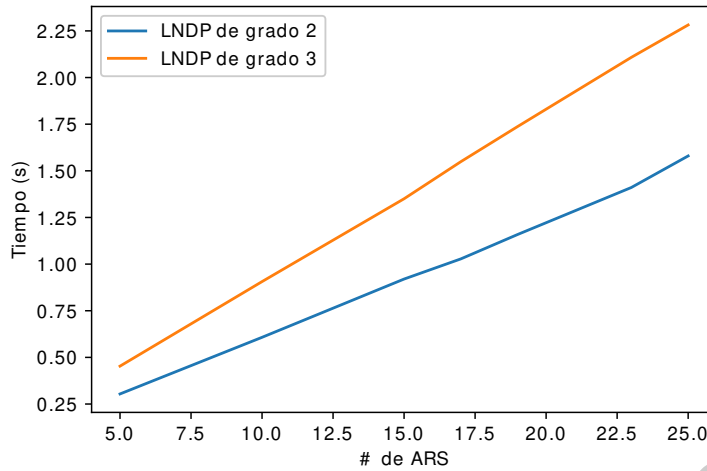


Figura 3.6: Tiempo de cálculo de características LNDP para distintos números de ARS

propuesta en [55] y convierte el problema del reconocimiento facial en un problema binario, un conjunto de entrenamiento $T = \{t_1, t_2, \dots, t_l\}$ con K clases distintas se convierte en un conjunto de dos clases C_1 y C_2 , mismas que son usadas para entrenar una SVM, el conjunto C_1 es un conjunto con la diferencia entre imágenes pertenecientes a la misma clase

$$C_1 = \{t_i - t_j | t_i \sim t_j\} \quad (3.1)$$

la segunda clase se define con la diferencia entre imágenes que no pertenecen a la misma clase

$$C_2 = \{t_i - t_j | t_i \not\sim t_j\} \quad (3.2)$$

la SVM resultantes del entrenamiento se utiliza para obtener las medidas de similitud del conjunto de entrenamiento, para la tarea de verificación se ingresa una prueba p que dice ser un individuo g_i en la galería, la prueba se acepta si la medida de similitud d es menor o igual que un umbral definido Δ . Para la identificación, una prueba p debe ser identificada dentro de la galería, la medida de similitud se calcula para p con todos los individuos en la galería, de esta manera la identidad de la prueba se obtiene del individuo en la galería cuya similitud d con la prueba p es menor

$$d_l = \sum_{i=1}^l \alpha_i y_i K(\mathbf{x}_i, \mathbf{g}_j - \mathbf{p}) + b \quad (3.3)$$

La segunda metodología trata el reconocimiento como un problema de $K = \{k_1, k_2, \dots, k_m\}$ clases, utilizando el enfoque *OneVsAll* se entrenan K clasificadores binarios, cada uno de estos considera una clase k_j como la clase positiva y el resto como una sola clase negativa, de esta forma se puede dividir un problema multiclase en K problemas binarios. Cada uno de estos métodos tiene sus ventajas y desventajas, por ejemplo, el uso de un clasificador binario simplifica el problema de optimización ya que solo es necesario encontrar una superficie de decisión que separe dos

clases distintas, sin embargo, se requiere un tiempo de procesamiento adicional para encontrar los conjuntos C_1 y C_2 mencionados anteriormente y puede haber errores derivados de las diferencias entre las imágenes comparadas, cuando se trata el problema como multiclase, la cantidad de clasificadores binarios a entrenar puede aumentar considerablemente cuando aumenta el número de sujetos dentro de la base de datos pero no requiere de almacenar una galería dentro de la memoria del sistema ya que la información de los individuos conocidos ya se encuentra en las superficies de decisión encontradas cuando se resuelven los problemas de optimización por lo que el tiempo de comparación puede reducirse respecto a un sistema con galerías extensas. Por esta razón se escogió utilizar el sistema multiclase utilizando el enfoque 1VA.

3.5.1. Medición del desempeño

El desempeño en el reconocimiento facial se reporta en tres tareas estándar: verificación e identificación en conjunto abierto y en conjunto cerrado [56], comúnmente para realizar estas tareas se requieren 3 conjuntos de datos, una galería y dos pruebas, la galería G es un conjunto de capturas de individuos conocidos por el sistema, las dos pruebas son conjuntos de muestras que se presentan al sistema para realizar la verificación e identificación, los dos conjuntos de pruebas se llamarán P_G y P_N , en el primero se concentran muestras de individuos que existen en la galería y en el segundo contiene muestras de personas desconocidas para el sistema. La identificación en conjunto cerrado es la medida más popular en la comunidad sobre la identificación en conjunto abierto, por lo que esta métrica será la utilizada para identificación. Cuando se trata el reconocimiento como un problema multiclase no se requiere de una galería como un conjunto separado, los individuos conocidos por el sistema serán aquellos presentados durante la fase de entrenamiento. También se hará uso de una matriz de confusión para reportar el desempeño general del clasificador.

Verificación

Como se mencionó en el Capítulo 1, la tarea de verificación consiste en la comparación de una imagen de prueba p que clama la identidad de una persona en la galería, el objetivo es determinar si dicho clamado es cierto, el método utilizado para la verificación se conoce como el método del verdadero impostor, consiste en utilizar las dos galerías para calcular las medidas de similitud con personas en la galería, las medidas pueden ser un *match* cuando la muestra en la prueba si corresponde a una persona en la galería y *nonmatch* cuando la persona en la prueba no corresponde a una en la galería. En verificación es común encontrar dos métricas distintas, *verification rate* (VR) y *false alarm rate* o *false acceptance rate* (FAR) que muestran la cantidad de individuos que están siendo correctamente verificados y los que no (personas desconocidas por el sistema siendo aceptadas) respectivamente. De acuerdo con Li y Jain, en el método del verdadero impostor la VR para un umbral Δ se calcula como

$$P_V(\Delta) = \frac{|\{p_j : s_{ij} \geq \Delta, id(g_i) = id(p_j)\}|}{|P_G|} \quad (3.4)$$

donde p_j y g_i son imágenes en los conjuntos de prueba y galería respectivamente, s_{ij} es la medida de similitud entre g_i y p_j , $id(g_i)$ es la identidad de la persona g_i , la cantidad de *match scores* $|P_G|$ se obtiene utilizando solamente la prueba P_G , para la FAR, las *nonmatch scores* $|P_N||G|$ son las diferencias entre los individuos en la galería y P_N

$$P_{FA}(\Delta) = \frac{|\{s_{ij} : s_{ij} \geq \Delta\}|}{|P_N||G|} \quad (3.5)$$

Los resultados par diferentes valores de Δ se grafican en una curva *Receiving Operating Characteristic* (ROC)

Identificación de conjunto cerrado

Mediante esta métrica se busca encontrar si la identidad de una prueba se encuentra entre las primeras n coincidencias, en este método se calculan las medidas de similitud de un prueba p_j con la galería G y se ordenan, la *identification rate* (IR) o *recognition rate* (RR) para un rango n , la cantidad de pruebas en un rango n o menor se calcula como

$$P_I(n) = \frac{|\{p_j : rank(p_j) \leq n\}|}{|P_G|} \quad (3.6)$$

El desempeño de la identificación en conjunto cerrado se reporta mediante una curva *Cumulative Match Characteristic* (CMC), el eje horizontal es el rango en escala logarítmica y en el eje vertical se tiene $P_I(n)$

Matriz de confusión

Existen otras métricas comunes para medir el desempeño de un clasificador binario, el modelo de la SVM puede predecir una etiqueta para un dato en el conjunto de prueba, estas predicciones son comparadas con las etiquetas verdaderas del conjunto de datos en una matriz de confusión. Para un conjunto de entrenamiento con etiquetas $\{p, n\}$ y si las predicciones de un modelo para el mismo conjunto son etiquetas $\{Y, N\}$, en la Tabla 3.2 se muestra la matriz de confusión para un problema de clasificación binario, TP, FP, TN, FN corresponden a *true positive*, *false positive*, *true negative*, *false negative* respectivamente, una predicción puede caer en cualquiera de estas cuatro.

- TP: Instancia que es positiva y se clasifica como positiva.
- FP: Instancia negativa clasificada como positiva.
- FN: Instancia positiva clasificada como negativa.
- TN: Instancia negativa que se calsifica como negativa.

Tabla 3.2: Matriz de confusión para un clasificador binario

		Clase Real	
		p	n
predicción	Y	TP	FP
	N	FN	TN

En un sistema de clasificación se buscan minimizar los FP y FN que representan errores de clasificación, también de la matriz de confusión se pueden obtener las siguientes métricas [57]

$$sensitivity = \frac{TP}{P} \quad (3.7)$$

$$precision = \frac{TP}{TP + FP} \quad (3.8)$$

$$F1score = \frac{2}{1/precision + 1/recall} \quad (3.9)$$

$$accuracy = \frac{TP + TN}{P + N} \quad (3.10)$$

la *sensitivity* indica cuántas de las muestras positivas fueron calculadas como positivas, *precision* dice de las muestras clasificadas como positivas cuántas en realidad son positivas, la *F1score* o *Fmeasure* combina ambas métricas para dar un solo resultado, finalmente, *accuracy* muestra cuántas de las muestras positivas y negativas fueron correctamente clasificadas. Para extender la matriz a un clasificador multiclase simplemente se agregan las demás etiquetas reales y las predicciones, la diagonal en la matriz cuadrada resultantes serán los TP para cada clase, en las columnas se encontraran los FN y los FP en las filas, el desempeño general del sistema se puede obtener promediando los valores de *sensitivity*, *precision* y *accuracy* para cada clase.

Resultados

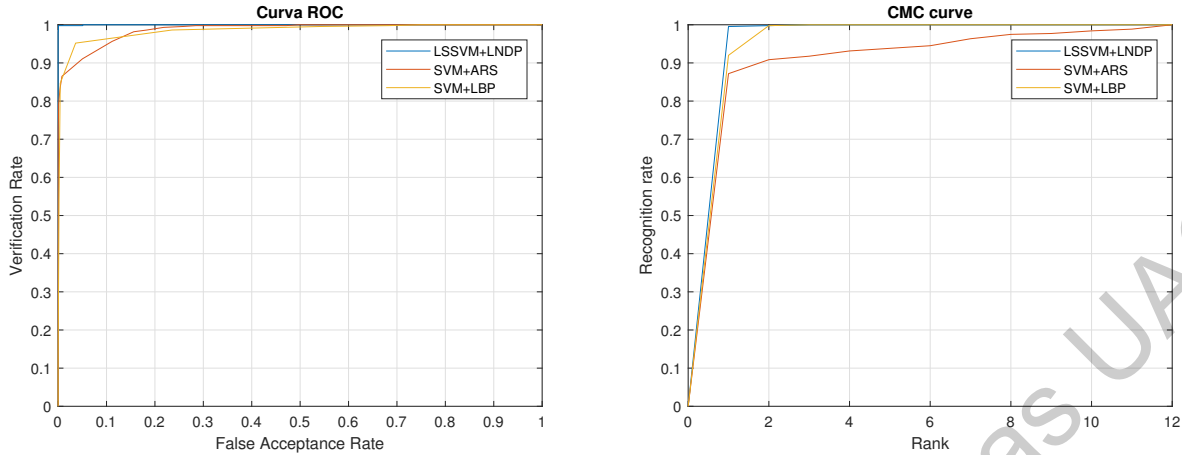
En esta sección se presenta el desempeño del método propuesto mediante pruebas en la *Texas 3DFRD* tomando en cuenta diferentes aspectos

- Se evalúa el desempeño utilizando diferentes parámetros del descriptor propuesto.
- Se evalúa el desempeño para diferentes funciones kernel e hiperparámetros en la SVM.
- Se realiza una comparación con diferentes métodos del estado del arte.

Es importante mencionar que los conjuntos de prueba no serán los mismos para las dos diferentes tipos de SVM a utilizar, en el caso del problema binario se utilizan las particiones tal como se muestran en la Tabla 3.1, tratándose del problema multiclase, las particiones de galería y prueba se unieron en un solo conjunto, esto se debe a que en esta metodología no se requiere de una galería para realizar la comparación con las imágenes en la prueba, para poder realizar los experimentos de identificación en el conjunto cerrado solo se toman en cuenta las imágenes en el conjunto de prueba cuyas identidades son conocidas por el sistema mediante la fase de entrenamiento, en la tarea de verificación el conjunto de prueba se dividió en dos de tal manera que se obtuvieron los conjuntos P_G y P_N mencionados en la sección anterior.

4.1. Resultados preliminares

Los primeros experimentos se realizaron para verificar el comportamiento de las características tipo LDP (más específicamente LNDP) en conjunto con una maquina de soporte de vectores. Utilizando la *Texas 3DFRDatabase* se implementaron los 2 modos de trabajo, verificación e identificación, utilizando un universo cerrado, ambos ofreciendo un alto desempeño en términos de VR y RR. En la Figura 4.1 se muestran el *benchmarking* de un sistema que utiliza *Least Square SVM* (LSSVM) en conjunto con características LNDP contra otros métodos del estado del arte que



(a) VR de los experimentos preliminares

(b) RR de los experimentos preliminares

Figura 4.1: Resultados preliminares

Tabla 4.1: Verification rate y Recognition rate para puntos fijos en 0.1 FAR y Rank-1

Método	VR en 0.1 FAR (%)	RR en Rank-1 (%)	Tiempo de extracción (s)
SVM+ARS [2]	95.65	87.19	0.0477 ± 0.0017
SVM+LBP [38]	95.8	91.99	0.0000536 ± 0.00014
LSSVM+LNDP	99.54	99.77	5.92 ± 0.1836

hacen uso de técnicas similares, donde claramente el uso de LSSVM-LNDP tiene desempeño en ambas tareas en lo que a VR y RR corresponde, sin embargo, la extracción características LNDP para una imagen completa en la base de datos es muy exhaustiva para el computador, la Figura 4.2 muestra el tiempo en segundos que toma extraer las características para una imagen completa a diferentes escalas respecto al tamaño original y se observa que aún haciendo un re-escalamiento al 10 % el tiempo parece lento para un sistema de RF en tiempo real, en la Tabla 4.1 se encuentra la información correspondiente a las curvas en 4.1 más los tiempos de extracción correspondientes a cada método donde se observa que a pesar de ser superior en comportamiento, utilizar LSSVM con características LNDP conlleva un mayor tiempo de cálculo.

Estos primeros resultados fueron utilizados para la escritura de un artículo de divulgación científica publicado en *Research in computing science* del Instituto Politécnico Nacional [58].

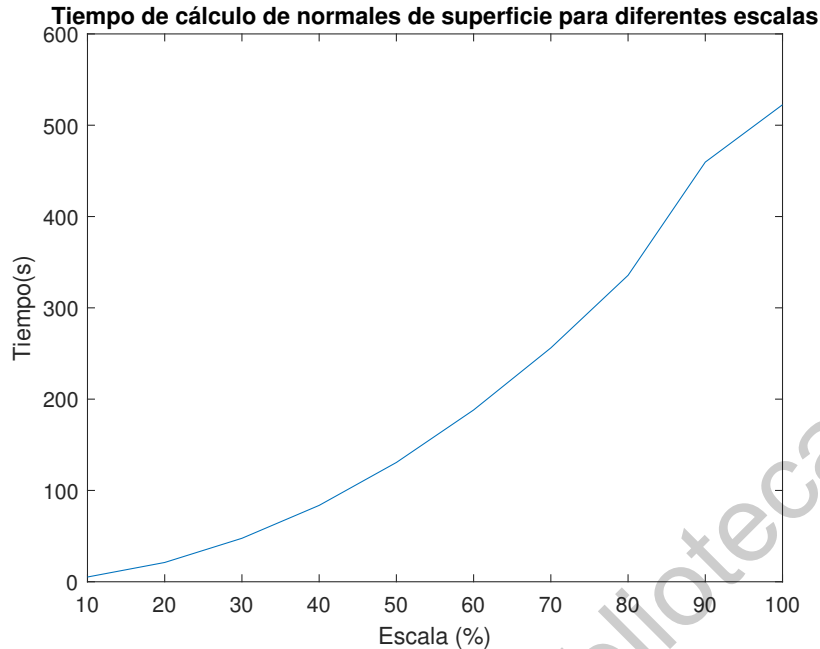


Figura 4.2: Tiempos de extracción de características LNDP para una imagen en la base de datos a diferentes escalas

4.2. Efectos de los parámetros en la extracción

Como se ha mencionado anteriormente, en la fase de extracción de características hay diferentes parámetros que pueden influir en el desempeño del sistema de RF, en esta sección se revisarán los efectos de distintos valores de ARS, grado de las características LDP y la cantidad de *bins* utilizadas en los histogramas.

4.2.1. Efectos de la cantidad de *bins*

Los primeros experimentos se fijaron utilizando 17 ARS con diferentes números de *bins* en los histogramas para analizar el efecto que tienen en la RR, inicialmente se utilizaron $nBins = \{6, 8, 10, 15, 50, 100, 200\}$ con un descriptor LDP de grado 3 y se utilizó el kernel PCA con kernel Gaussiano para transformar las características previo a la fase de entrenamiento de la SVM lineal, como se observa en la Tabla 4.2 con números bajos de *bins* el desempeño del clasificador es inferior al 90 % de *accuracy*, a partir de 50 *bins* los valores ya superan el 90 %, lográndose el resultado más alto en 50. Este desempeño está por debajo de los sistemas en el estado del arte mostrados en la Tabla 1.3, que aunque no se estén evaluando las mismas métricas por el momento, se puede obtener una buena idea general de cual será el desempeño en verificación e identificación en base

Tabla 4.2: Desempeño del método propuesto utilizando 17 ARS y distintos números de *bins* en los histogramas con KPCA.

Cantidad de <i>bins</i>	<i>Precision</i>	<i>Sensitivity</i>	<i>F1score</i>	<i>Accuracy</i>
6	0.61	0.64	0.60	0.66
8	0.77	0.84	0.78	0.83
10	0.80	0.88	0.82	0.86
15	0.86	0.93	0.87	0.92
50	0.89	0.96	0.90	0.94
100	0.84	0.94	0.88	0.93
200	0.87	0.93	0.87	0.92

al *accuracy*, lo siguiente es verificar si el mapeo de las características mediante KPCA está afectando negativamente el desempeño del sistema, utilizando los mismos conjuntos de entrenamiento y prueba se realizan ahora los experimentos sin la transformación KPCA y utilizando una SVM con kernel Gaussiano, la Tabla 4.3 muestra que no entrenar con KPCA aumenta el desempeño cuando se utilizan valores pequeños de *bins* y para valores grandes (a partir de 50) los resultados disminuyeron ligeramente, en este experimento utilizar 15 barras ofreció los resultados más altos, aunque por poca diferencia. Sin embargo, estos valores aún se encuentran por debajo de un desempeño de los mostrados en la Tabla 1.3, por lo que el siguiente paso es utilizar las características sin realizar la construcción de histogramas, es decir, en el Algoritmo 2 saltar los pasos 11 y 13 para formar un solo vector de características LDP en los puntos ARS, de esta manera se obtuvieron los resultados presentados en la Tabla 4.4, mismos que son valores superiores a los resultados en las Tablas 4.2 y 4.3, por lo que no realizar la construcción de los histogramas se presenta como la mejor decisión en este escenario y se considera que de esta manera los resultados en verificación e identificación serán competitivos. En la Figura 4.3 se muestra la RR utilizando distintas cantidades de barras en los histogramas sin la transformación KPCA, aquí claramente se observan 2 cosas, la primera es que las métricas presentadas previamente para medir el desempeño del clasificador no tienen por que coincidir con la RR, la segunda es que, como ya se mencionó anteriormente, no hay una relación, lineal o de otro tipo, aparente con el número de *bins* y un mejor desempeño, también es notorio que el no realizar la construcción de histogramas obtiene mejores resultados hasta el Rank-5, a partir de este, la diferencia es mínima. El mejor resultado para RR-1 es del 97.55 %, mismo que alcanza el 99.77 % en RR-2 para llegar al 100 % en el RR-4. Es razonable pensar que con la cantidad de puntos que se están utilizando para extraer información utilizar histogramas de los valores resultantes cause pérdidas de información, como se observa en la Figura 4.4, utilizar pocas barras en los histogramas (6,8 y 10) ocasiona que la información se encuentre muy concentrada

Tabla 4.3: Desempeño del método propuesto utilizando 17 ARS y distintos números de *bins* en los histogramas sin KPCA.

Cantidad de <i>bins</i>	<i>Precision</i>	<i>Sensitivity</i>	<i>F1score</i>	<i>Accuracy</i>
6	0.76	0.83	0.76	0.82
8	0.83	0.90	0.84	0.88
10	0.83	0.90	0.84	0.88
15	0.88	0.95	0.89	0.94
50	0.86	0.93	0.87	0.92
100	0.86	0.92	0.87	0.91
200	0.85	0.92	0.85	0.90

por lo que diferentes individuos pueden dar patrones similares, sin embargo, cuando la cantidad de barras es muy grande (100,200) la información puede encontrarse muy dispersa y muchos valores no utilizados tendrán cuenta en cero, esto también vuelve difícil distinguir entre individuos, utilizar 15 *bins* muestra una mejor distribución de la información aunque si bien el desempeño es mejor respecto a las demás construcciones, sigue siendo inferior que el método sin construcción.

Tabla 4.4: Desempeño del método propuesto utilizando 17 ARS sin construcción de histogramas

<i>Precision</i>	<i>Sensitivity</i>	<i>F1score</i>	<i>Accuracy</i>
0.94	0.98	0.95	0.98

En el modo de verificación los resultados no son muy diferentes a los resultados en reconocimiento, no realizar la construcción de un histograma refleja los mejores resultados en *Verification Rate* y *False Acceptance Rate*, en los experimentos que se utilizaron histogramas con distintas cantidades de barras, 6 *bins* siguen ofreciendo el desempeño más pobre, las demás cantidades de *bins* ofrecen prácticamente los mismos resultados como se observa en la Figura 4.5 se observan las curvas ROC utilizando diferentes umbrales, hay que recordar que el objetivo es aumentar la VR manteniendo la FAR como la mínima posible, para entender la diferencia la Tabla 4.5 muestra los resultados de VR y FAR para un valor $\Delta = 10,8$ para las distintas cantidades de *bins*, claramente se nota que sin la construcción se obtiene una FAR similar a los demás experimentos pero con resultados en VR superiores.

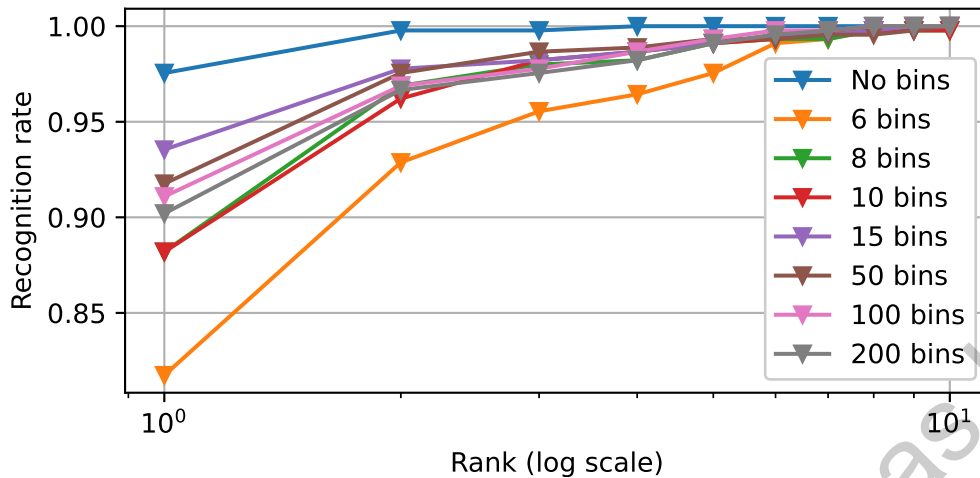
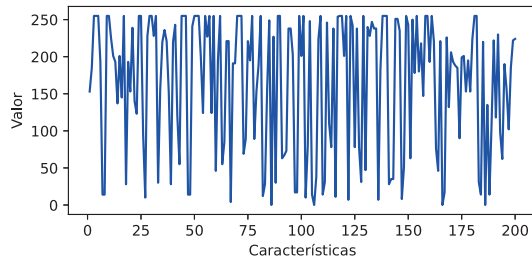


Figura 4.3: RR con 17 ARS-LDP 3 y distintas cantidades de *bins*

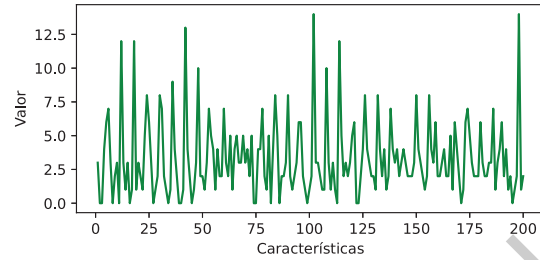
4.2.2. Efectos de la cantidad de ARS

En la sección anterior se observó que utilizar 17 ARS con un descriptor LDP de tercer grado ofrece resultados en RR sobre el 97 %, sin embargo, como se mencionó en la Sección 3.4.3 el tiempo de cálculo para esta cantidad de ARS se vuelve lento, por lo que en esta Sección se explorará como afecta la cantidad de ARS el desempeño del sistema siendo 17 el número máximo a evaluar debido a que un valor superior conlleva tiempos de extracción más largos de lo deseado. En la Figura 4.6 se observan los valores de RR con diferentes valores de ARS, es claro que utilizar 17 ARS presenta un mejor desempeño que menos ARS a excepción de RR-1 para 15 ARS, donde la diferencia es mínima, la Tabla 4.6 muestra los valores de RR-1 para cada valor de ARS así como el tiempo de extracción de características, en la tarea de reconocimiento el *top match* o Rank 1 es más importante que la identificación en rangos superiores, tomando esto en cuenta, utilizar 15 ARS presenta la mejor opción para obtener un buen desempeño tanto en reconocimiento como en tiempo, el tiempo de extracción se puede reducir utilizando 12 ARS sacrificando 1.56 % en RR, un mejor equilibrio entre el tiempo y RR se encuentra para 15 ARS, desafortunadamente tratar de reducir en mayor medida el tiempo reduciendo las ARS conlleva un empobrecimiento en el reconocimiento.

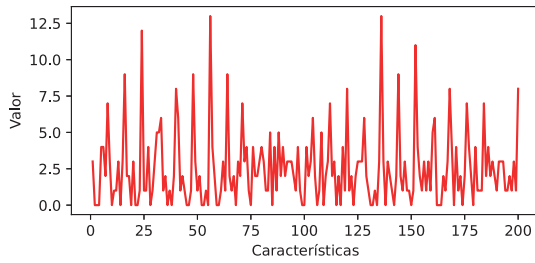
En cuanto a la verificación con el descriptor de tercer grado, se obtienen resultados similares para las diferentes cantidades de ARS, la Figura 4.7 contiene las curvas correspondientes a estos experimentos, para comparar los diferentes experimentos también se hace uso de una tabla con los valores de VR y FAR a un umbral determinado $\Delta = 10,8$, esta información se concentra en la Tabla 4.7 y muestra que 15 ARS también son mejor opción que 17 ARS ya que aunque la VR es la misma, utilizar 15 ARS disminuye la cantidad de impostores aceptados erróneamente por el sistema. Por otro lado, sí es posible reducir el tiempo cambiando el orden de las características LDP de grado 3 a grado 2, esto sabemos que reduce el tiempo de extracción como se ve en la Figura 3.6 pero para comprender como se ve afectado el desempeño, se realizaron experimentos con los



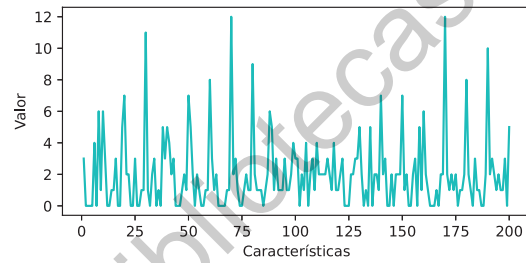
(a) Sin construcción



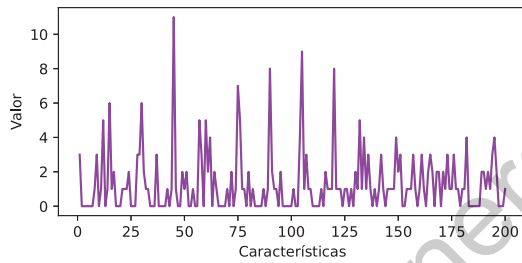
(b) 6 bins



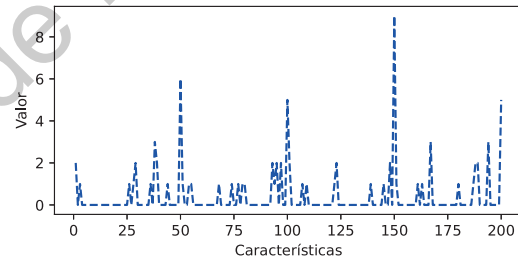
(c) 8 bins



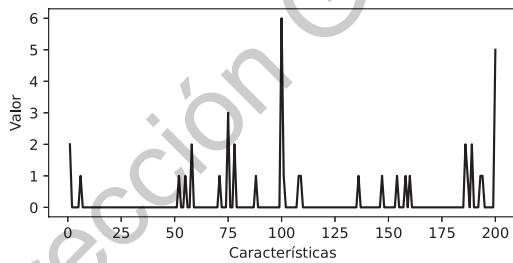
(d) 10 bins



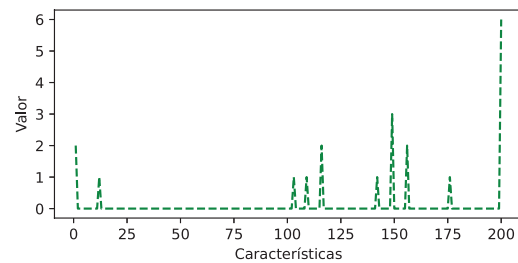
(e) 15 bins



(f) 50 bins



(g) 100 bins



(h) 200 bins

Figura 4.4: Patrones obtenidos con distintas construcciones de histogramas para 17 ARS y LDP de grado 3 de un solo individuo

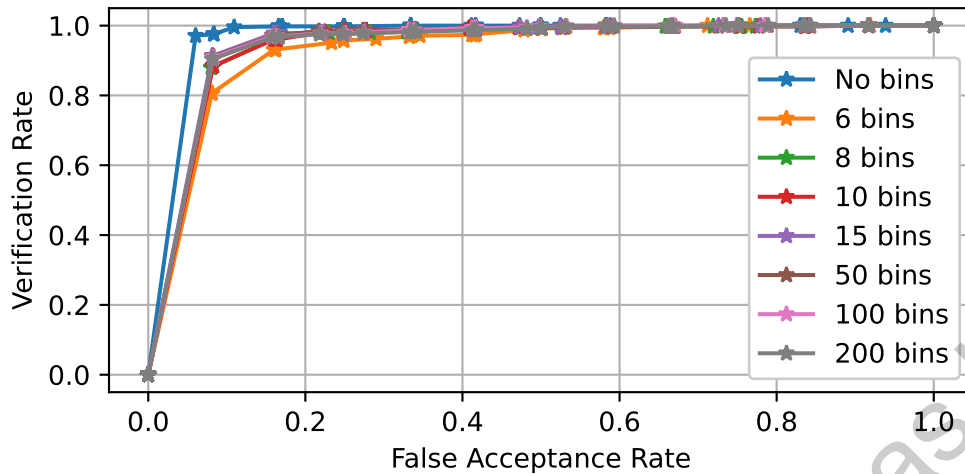


Figura 4.5: VR para diferentes cantidades de bins y LDP de tercer grado

mismos datos utilizados anteriormente, los resultados se muestran en la Figura 4.8, en esta ocasión utilizar características de 2 grado mostró una mejora importante utilizando 8 ARS, donde la RR aumentó casi un 7 %, también es fácil observar que en RR-1 para 17 ARS sigue siendo ligeramente inferior que con 15 direcciones en la máscara cuyo RR-1 resulta igual que para 12 ARS en estos datos, siendo así, la mejor combinación en parámetros para obtener mejores resultados en RR y tiempo de extracción se presenta para 12 ARS y LDP de segundo grado, también es fácil observar que cualquier cantidad de ARS-LDP2 alcanza el 100 % de reconocimiento en RR-2, por lo que podemos afirmar que este tipo de características son mejores respecto a las de tercer grado, si se desea mejorar el tiempo de extracción es posible utilizar 8 ARS sin sacrificar mucho reconocimiento. En la tarea de verificación presentada en la Figura 4.9, se observa un comportamiento que parece ser el mismo para los distintos valores de ARS utilizados en el experimento, sin embargo, hay unas sutiles diferencias que se pueden observar haciendo uso de la Tabla 4.9, en este caso el mejor desempeño se lo adjudica el sistema con 8 ARS en cuanto a FAR con una ligera desventaja en la VR respecto a los demás sistemas.

4.2.3. Efectos de las normales de superficie

En esta sección se evaluará el desempeño del sistema cuando se utilizan las normales de superficie, como se vio anteriormente, el tiempo de estimación de las normales para toda la información en una imagen es lento e innecesario tomando en cuenta que solo se requiere la información normal en una determinada región del rostro, por lo tanto, lo primero en realizarse fue un recorte de las imágenes faciales para quedarnos únicamente con la región de interés, en esta es donde se estimaron las normales de superficie, en la Figura 4.10 se muestra el recorte del rostro para utilizar únicamente la región de interés para extraer las normales de superficie, de cada imagen normal se realizó la extracción de características LNDP y se calculó la RR para cada normal por separado, los resultados se muestran en la Figura 4.11, el desempeño claramente es pobre, en RR-1 se encuentra

Tabla 4.5: VR y FAR para diferentes cantidades de barras y $\Delta = 10,8$

Cantidad de bins	VR(%)	FAR(%)
No bins	97.55	8.33
6	80.62	8.08
8	87.97	8.06
10	88.41	8.28
15	91.31	8.18
50	91.09	8.3
100	90.86	8.3
200	90.42	8.21

Tabla 4.6: RR para diferentes cantidades de ARS-LDP3 con el tiempo de extracción

# ARS	RR-1(%)	Tiempo de extracción (s)
8	92.87	0.75
12	96.21	1.1
15	97.77	1.35
17	97.55	1.54

por debajo del 50 % siendo la coordenada z la resulta mejor para discriminar entre individuos, con estos resultados no parece posible que aumentando el número de ARS se pueda mejorar la tasa de reconocimiento, por lo que el siguiente paso es realizar los experimentos utilizando las características de las 3 normales de superficie como un solo patrón para averiguar si realmente vale la pena hacer uso de las normales en este descriptor.

En la Figura 4.12 se utilizaron las características LNDP de las 3 normales de superficie como un solo vector de características, el desempeño permanece en un RR-1 de aproximadamente el 50 %, logrando superar el 100 % hasta el RR-10, mismo que es un rango inaceptable tomando en cuenta el tamaño de la galería, por lo que no se considera viable utilizar normales de superficie en este descriptor ya que presenta una tasa de reconocimiento bastante baja a comparación de no utilizar las normales, adicionalmente la estimación de los vectores normales conlleva un mayor tiempo de procesamiento, este comportamiento se puede deber a que realizar esta extracción genera patrones muy similares para diferentes individuos y por lo tanto es difícil para el clasificador separar entre

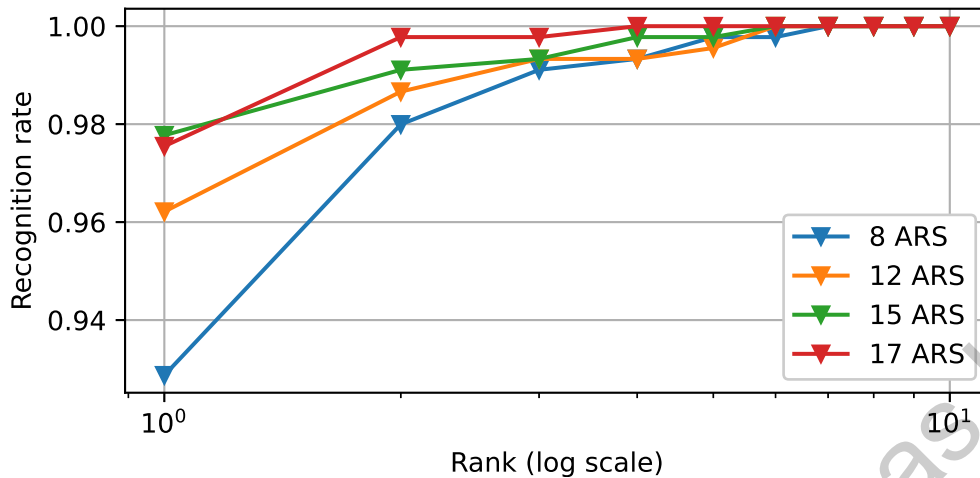


Figura 4.6: RR con diferentes valores de ARS y LDP de grado 3

Tabla 4.7: VR con diferentes números de ARS y LDP 3 para $\Delta = 10,8$

# ARS	VR(%)	FAR(%)
8	92.65	8.3
12	96.21	8.3
15	97.55	8.28
17	97.55	8.33

clases, en la Figura 4.13 se muestra la matriz de confusión del clasificador, se observa que en la mayoría de los individuos se tiene una exactitud menor al 50 %, el problema de separabilidad de los datos en este escenario puede también estar afectado por los hiperparámetros para el entrenamiento de la SVM, mismos que hasta ahora no han sido ajustados para cada tipo de datos, para poder desestimar las normales de superficie dentro del descriptor utilizado se requiere ajustar dichos hiperparámetros para revisar el comportamiento del sistema y decidir finalmente si las normales ofrecen mejor desempeño para la extracción de los patrones propuestos. En siguiente sección se evaluará el efecto de las constantes C y σ sobre los resultados del sistema para 8 ARS con y sin la estimación de las normales de superficie.

4.3. Efectos de los hiperparámetros en el entrenamiento

Los hiperparámetros son parámetros que no son aprendidos durante la fase de entrenamiento y por lo tanto deben definirse a priori, anteriormente ya se ha mencionado el efecto que pueden tener

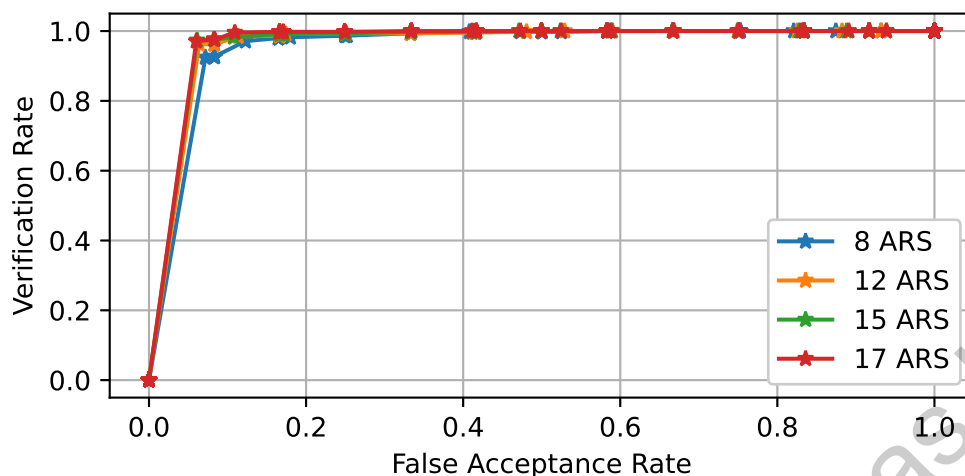


Figura 4.7: VR para diferentes cantidades de ARS y LDP de tercer grado

Tabla 4.8: RR para diferentes cantidades de ARS-LDP2 con el tiempo de extracción

# ARS	RR-1(%)	Tiempo de extracción (s)
8	99.1	0.45
12	99.77	0.525
15	99.77	0.807
17	99.55	0.919

los hiperparámetros en el entrenamiento de una SVM, de manera general un valor de penalización C muy grande puede resultar en un *overfitting*, de igual manera valores muy grandes de σ pueden resultar en regiones muy específicas en la superficie de decisión, es por esto por lo que el ajuste de los hiperparámetros resulta de gran importancia en el entrenamiento. Se realizó una validación cruzada y la búsqueda de los parámetros se hizo con *Grid search*. Para comenzar a entender el efecto que tienen estos hiperparámetros en la RR se refiere a la Figura 4.14, donde se utilizan los patrones obtenidos de las imágenes de rango utilizando 8 ARS-LNDP2 con las componentes normales compuestas como un solo vector, se observa que para valores más grandes de σ el desempeño es incluso inferior, lo que puede ser un indicio de *overfitting*, para los valores más grandes $\sigma = 100$ y $\sigma = 5,2e - 4$ se observa la RR más baja, para valores pequeños es donde se reportan los mejores resultados, sin embargo, esto no parece ser una regla general ya que se observa un valor de σ pequeño con una RR también baja, esto es porque la constante C no está siendo ajustada a la par para encontrar la mejor combinación posible dentro un espacio de búsqueda definido, para realizar el ajuste entonces se definió $\sigma = \{1e - 16, 1e - 15, 1e - 14, 1e - 13, 5,3e - 4, 0,1, 1, 10\}$ y $C = \{0,1, 0,25, 0,5, 0, 75, 1, 5, 10\}$ y se realizó la búsqueda de malla con CV, también se incluyó

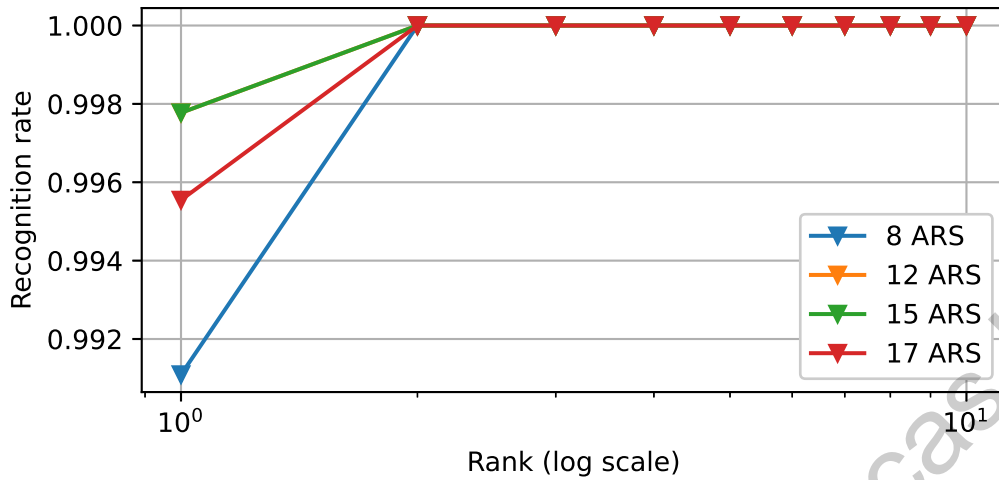


Figura 4.8: RR para diferentes cantidades de ARS y LDP de grado 2

Tabla 4.9: VR y FAR con distintas cantidades de ARS y LDP 2 para $\Delta = 10,8$

# ARS	VR(%)	FAR(%)
8	98.88	8.16
12	99.77	8.30
15	99.55	8.30
17	99.55	8.30

en la búsqueda al kernel polinomial con $d = \{2, 3, 4\}$, tras realizar el ajuste se encontró que el mejor desempeño se encuentra utilizando un kernel polinomial de segundo grado y el parámetro $C = 0,1$, en la Figura 4.15 se observa que no hay cambios significativos respecto a los obtenidos previamente sin ajuste, a diferencia que anteriormente se utilizó un kernel RBF, el 90 % en RR se alcanza hasta el RR-5, por lo que no se considera que estas características (haciendo uso de las normales) representan una opción aceptable.

Utilizando los mismos espacios de búsqueda para 8ARS-LDP2 sin el uso de las normales de superficie los parámetros que ofrecen el mejor desempeño son igualmente un kernel polinomial de segundo grado con la constante $C = 0,1$, la RR utilizando estos valores en el entrenamiento se muestra en la Figura 4.16, nótese que la RR-1 es superior al 99 % por lo que se puede asegurar que el clasificador tiene buena capacidad de generalización, si se revisa la verificación con los mismos parámetros, se obtiene la gráfica de la Figura 4.17 donde se encuentran los valores $VR = 0,9710$ y $FAR = 0,0821$ para $\Delta = 10,8$, valores que son inferiores a los encontrados en la Tabla 4.9, lo que da el indicio que realizar el ajuste de los parámetros pensando en optimizar la tarea de

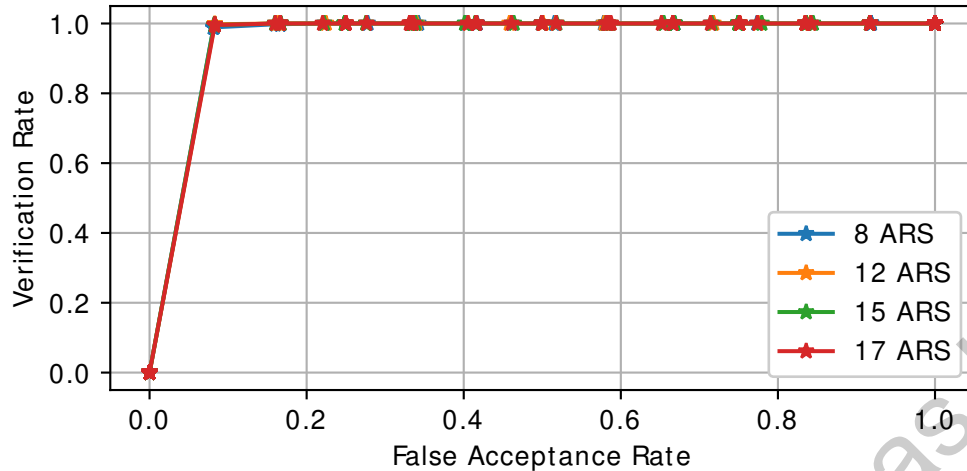


Figura 4.9: VR con diferentes cantidades de ARS y LDP de segundo grado

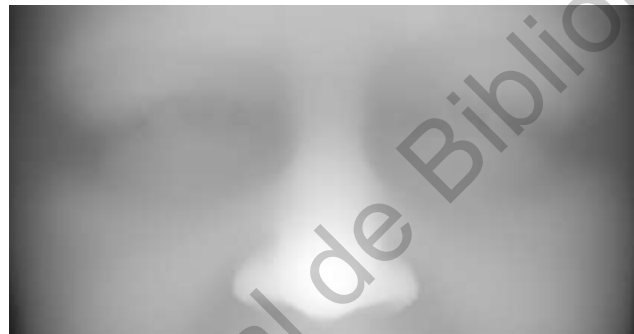


Figura 4.10: Región de interés para extracción de las normales de superficie.

reconocimiento descuida el desempeño del sistema cuando trabaja en verificación

4.4. Comparación con métodos del estado del arte

Para asegurar una comparación justa, se utilizaron los mismos conjuntos de entrenamiento y pruebas para cada uno de los métodos, también todos los experimentos se realizaron en el mismo computador con las especificaciones mostradas en el Capítulo 3, los sistemas a comparar son los presentados en [38] y [2] y las métricas comparadas son RR para la identificación y VR y FAR para la verificación. Las Figuras 4.18 y 4.19 muestran las curvas CMC y ROC del reconocimiento y la verificación respectivamente, en lo que corresponde al reconocimiento, se observa que para un Rank-1, el método propuesto tiene una clara ventaja sobre los otros 2 debido a que el primer *match* es en la mayoría de las ocasiones un *match score*, con la verificación no es igual de claro, esto porque con valores bajos de FAR (menores a 0.05) el desempeño no es superior, sin embargo, en un

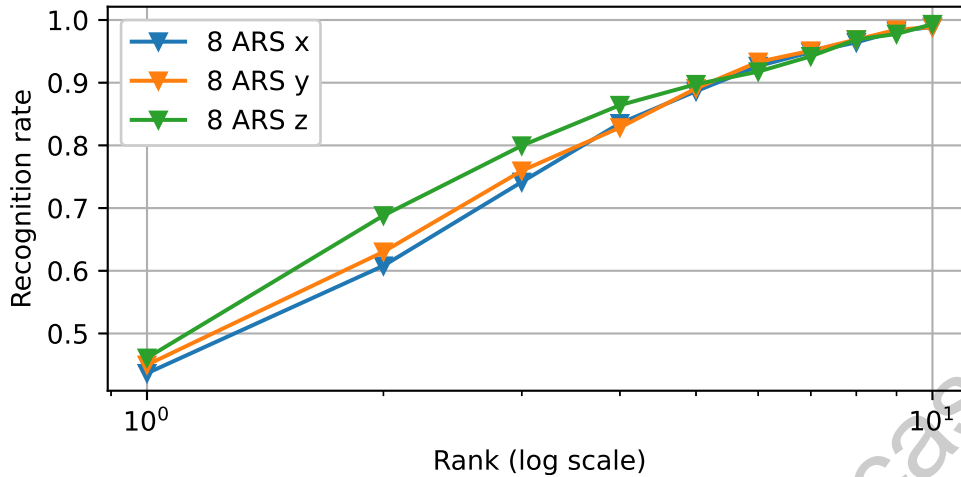


Figura 4.11: RR para 8 ARS y LNDP de segundo grado utilizando los patrones obtenidos de las normales x,y y z por separado

0.1FAR la verificación logra superar el 97 %, lo que significa que cuando se aceptan erróneamente el 10 % de los impostores se aceptan correctamente el 97.1 % de los individuos conocidos por el sistema. La Tabla 4.10 muestra los valores de RR y VR para valores fijos de rango y FAR respectivamente, donde se observa que el método propuesto en este trabajo supera a los propuestos por [38] y [2] en términos de *Verification rate* y *Recognition rate*

Método	RR-1 (%)	VR 0.1FAR (%)
SVM + ARS [2]	92.65	92.65
SVM + LBP [38]	91.75	91.53
Propuesto	98.88	97.1

Tabla 4.10: Comparación de resultados de reconocimiento en Rank-1 y verificación a un valor fijo de 0.1FAR

4.5. Discusión de los resultados

Se realizaron experimentos utilizando los mismos conjuntos para el entrenamiento y la prueba, la validación de cada entrenamiento se hizo mediante una validación cruzada asegurando que

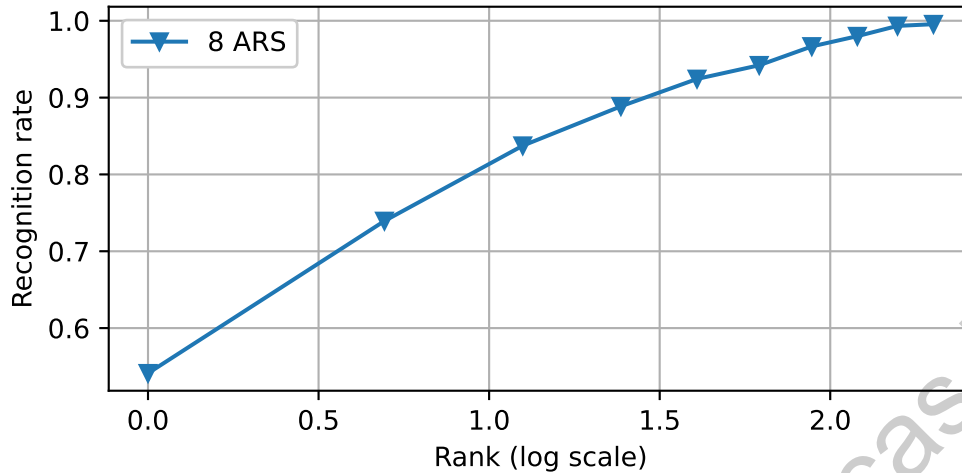


Figura 4.12: RR para 8 ARS y LNDR de segundo grado como un solo patrón.

todos los datos fueron tomados en cuenta durante esta etapa, haciendo cambios en los parámetros del algoritmo para la extracción de las características y los hiperparámetros del clasificador se encontró un sistema capaz de alcanzar un alto desempeño en reconocimiento y verificación aún cuando se utilizan imágenes con variaciones de iluminación y expresiones faciales en el conjunto de prueba, en las primeras pruebas se encontró que realizar una construcción de histogramas de cada dirección ARS disminuye el *accuracy* del sistema respecto a no realizar la construcción de los histogramas, esto debido a que con la baja cantidad de puntos utilizados, la información en los histogramas puede encontrarse muy dispersa cuando se trata de cantidades grandes de *bins* dando resultados de entre el 92 % y 94 % cuando se hace un entrenamiento de un modelo KPCA para cada dirección ARS y entre 90 % y 92 % sin el mapeo KPCA, para cantidades bajas de barras en los histogramas, en ambos casos no se superó el 90 %, utilizar todas las direcciones de ARS como un solo vector presentó el más alto hasta el momento con un valor del 98 % de *accuracy* y también fue superior en RR y VR, como la cantidad de direcciones ARS influye en el tiempo de extracción, se utilizaron diferentes cantidades de ARS con descriptores de segundo y tercer grado, utilizando 15 ARS casi siempre se alcanzaron los mejores resultados respecto a otras cantidades (ver Tablas 4.6 a 4.9) pero también conlleva un mayor tiempo para la extracción, utilizar 8 ARS ofreció un desempeño en el reconocimiento del 92 % y el menor tiempo consumido en la extracción, cuando el grado de descriptor se utilizó de segundo grado se aumentó la VR y RR respecto al tercer grado, esto debido a que cada punto ARS ya tiene una separación de 3 mm entre cada uno, por lo que la información direccional de tercer grado no aprovechada de igual manera que la de segundo grado, usar también el descriptor de segundo grado redujo el tiempo de extracción de 0.75 s a 0.45 s para 8 ARS. Cuando se trató de los experimentos que envuelven a las normales de superficie, los resultados no resultaron favorables para el descriptor, de acuerdo con lo reportado en los experimentos iniciales, las normales de superficie representan información importante que sería de utilidad en los experimentos futuros, sin embargo, los resultados en VR y RR a penas superiores al 50 % de VR y RR no se alinearon a lo esperado previamente, una posible razón es el hecho de que en este caso la

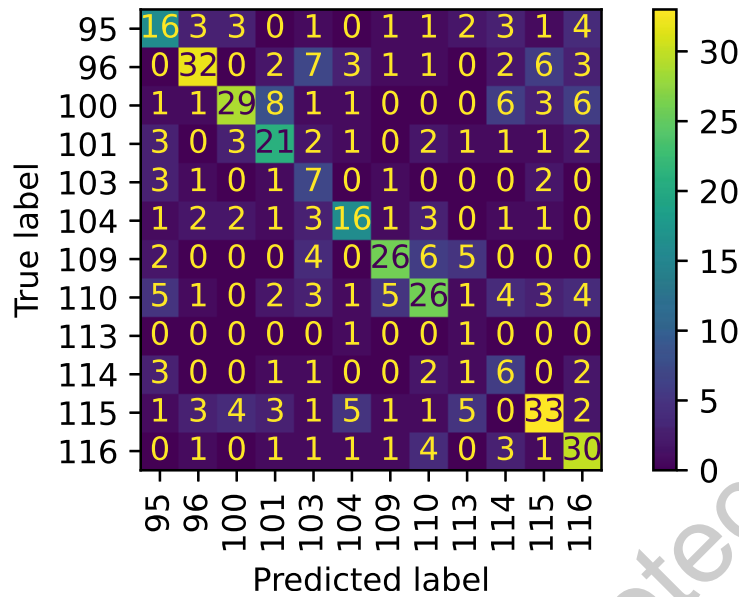


Figura 4.13: Matriz de confusión para los patrones obtenidos utilizando las 3 normales de superficie, 8 ARS y LDP de grado 2.

dimensionalidad de los datos aumenta y esto puede estar afectando negativamente el desempeño de la SVM utilizada para la clasificación, estos resultados también pueden estar relacionados con la dispersión de los datos utilizados en los rostros humano, la variabilidad con las distancias utilizadas puede estar ofreciendo patrones similares para diferentes individuos provocando falsos positivos en todas las clases como lo muestra la Figura 4.13, para poder asegurar cuales son las causas de este comportamiento hay más investigación y una experimentación con una mayor cantidad de datos que se decidió no continuar en este trabajo debido a los resultados ya obtenidos sin el uso de normales ya es satisfactorio para el propósito de esta investigación, continuar con el uso de las normales de superficie significa también un mayor tiempo de cómputo para obtención de las características. Los resultados de los métodos utilizados para la comparación resultaron similares a los presentados en las publicaciones originales a pesar de tratarse de conjuntos de datos distintos, por lo que se considera la experimentación y réplica de los mismos es acertada, la metodología propuesta se encuentra por encima de los mismos cuando se trata de medidas de verificación e identificación, sin embargo, hay trabajo que se puede realizar para el mejoramiento de los tiempos de procesamiento, desde una optimización del código y las estructuras de datos hasta una revisión o proposición de algoritmos que permitan disminuir la complejidad del descriptor. Los resultados arrojados por la experimentación permitieron entonces reescribir el Algoritmo 2 de manera que se eliminaron los pasos que no aportaron utilidad mejorar el desempeño del sistema, quedando como se ve en el Algoritmo 3

Algorithm 3: Descriptor propuesto después de la experimentación

Entrada: Imagen P, n, n_ARS, α

Salida : Vector de características

```
1 Calcular  $\theta$  utilizando 2.45
2 Aplicar máscara binaria a P
3 para cada  $d$  en  $n\_ARS$  hacer
4   para cada  $a$  en  $\alpha = 0, 45, 90, 135$  hacer
5     para cada punto en la máscara hacer
6       Calcular 2.44
7     fin
8     Concatenar para cada p
9   fin
10  Concatenar vector para cada  $\alpha$ 
11 fin
12 Concatenar histogramas para cada d
13 return  $ARS+LDP$ 
```

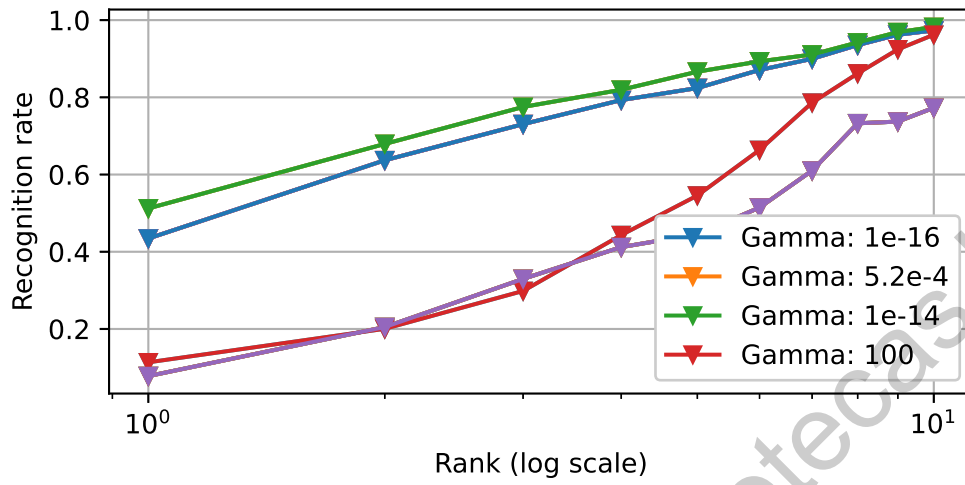


Figura 4.14: RR de 8ARS-LNDP2 con distintos valores de σ y $C = 1$

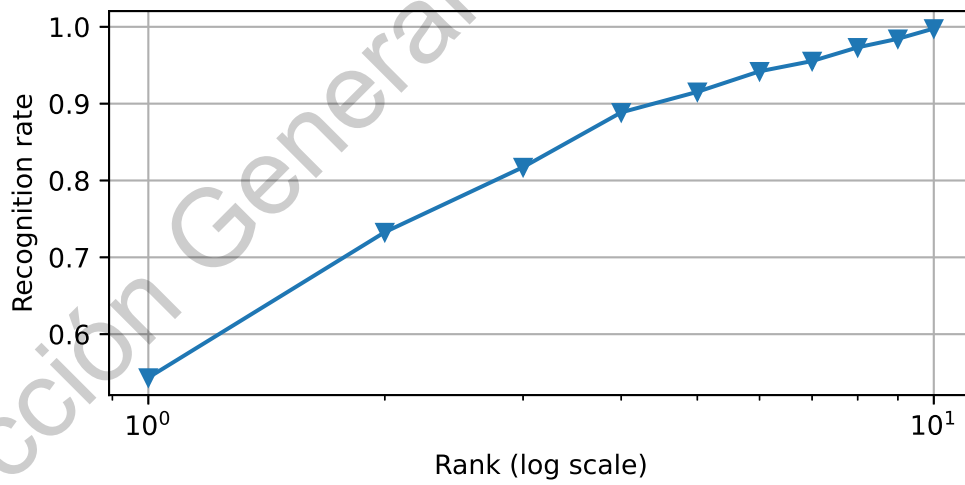


Figura 4.15: RR de 8ARS-LDP2 con kernel polinomial de grado 2 y $C = 0,1$ con normales de superficie

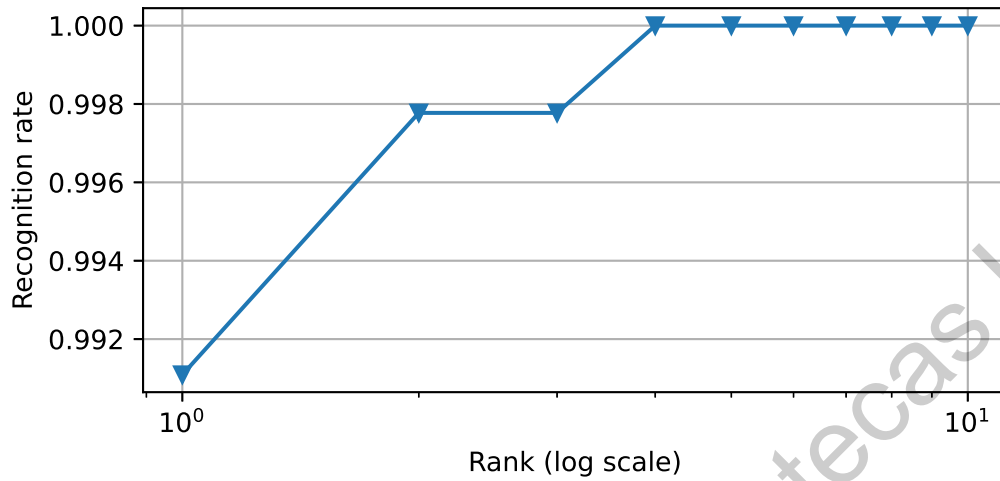


Figura 4.16: RR de 8ARS-LDP2 con kernel polinomial de grado 2 y $C = 0,1$ sin normales de superficie

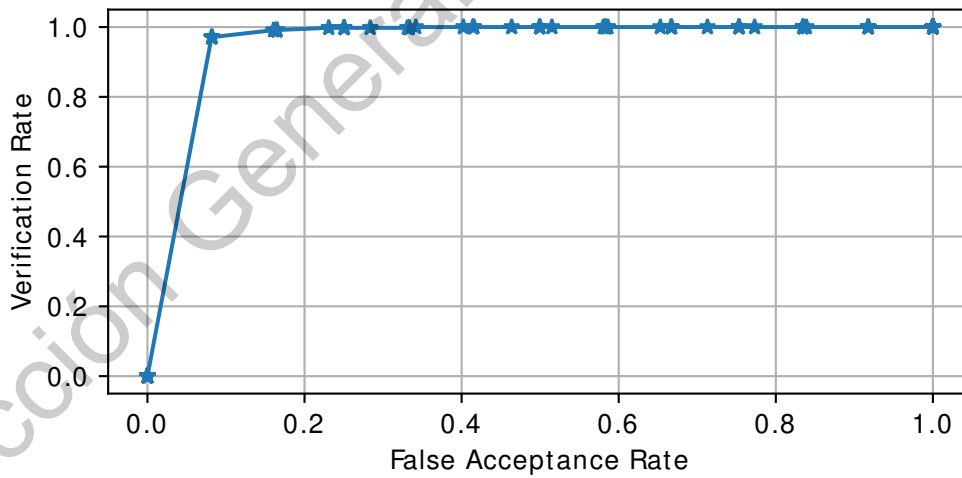


Figura 4.17: VR de 8ARS-LDP con kernel polinomial de segundo grado y $C = 0,1$ sin normales de superficie

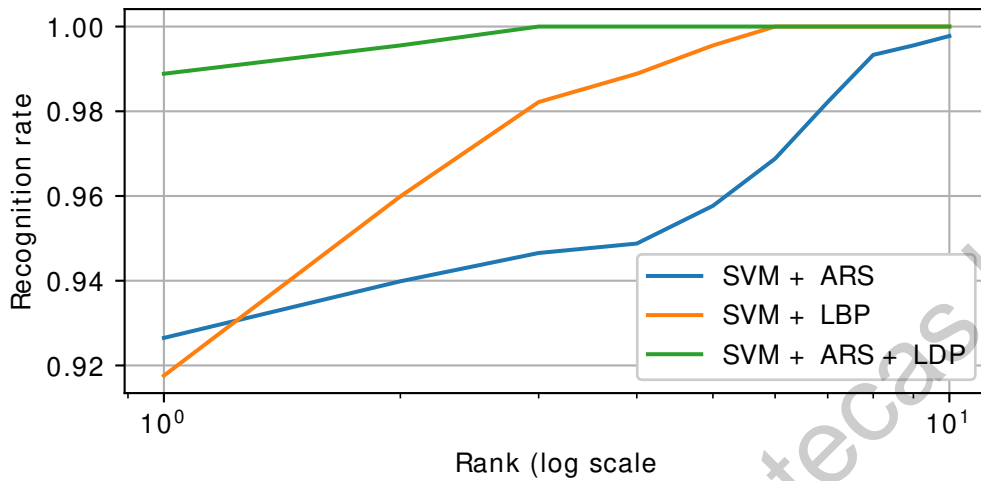


Figura 4.18: Curva CMC que presenta la comparación de los métodos evaluados en la tarea de reconocimiento

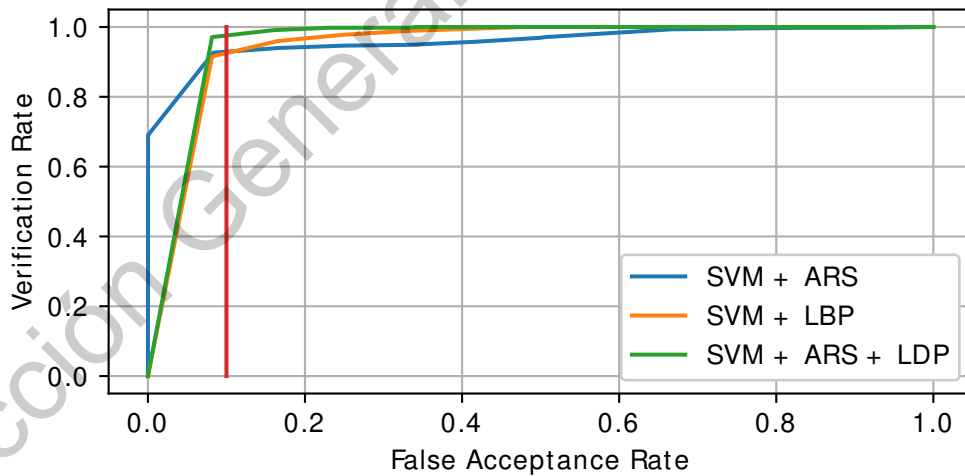


Figura 4.19: Curva ROC para los resultados comparados en la tarea de verificación, la línea roja interseca las curvas en 0.1 FAR

Conclusiones y trabajo futuro

5.1. Conclusiones

Se encontró que el uso de características direccionales como lo son las LDP ofrecen información de buena calidad para el reconocimiento facial, aunque como se mostró también conllevan un tiempo de extracción considerable cuando se obtiene la información del rostro completo y más aún cuando se involucra el cálculo de las normales de superficie por lo que se propuso la combinación de los descriptores ARS y LDP para obtener información de gran calidad para el diseño del sistema mediante la cual se alcanzó un desempeño superior en las tareas de reconocimiento y verificación con un tiempo todavía menor a un segundo, por lo que se considera competitivo con los métodos comparados.

Dentro del proceso de la extracción de las características se encontró que no realizar construcciones de histogramas presenta la mejor opción para mantener un tiempo de extracción menor y un mejor desempeño, el uso de las normales de superficie no ofreció los resultados que se esperaban aunque esto puede ser un síntoma de *overfitting* debido a la dimensionalidad de los datos que se están utilizando para el entrenamiento, sin embargo, no usar las normales ya ofrece una RR superior al 99 % en un Rank-1 por lo que se decidió proseguir por este camino sin realizar una mayor investigación en las posibles causas del bajo desempeño con las normales. La base de datos utilizada contiene imágenes con variaciones de iluminación y expresiones ligeras en los rostros, por la metodología propuesta sí es capaz de discriminar los rostros en estas condiciones con resultados superiores al 90 % y mejor en términos de VR y RR que los métodos utilizados en la comparación, aunque con un tiempo de procesamiento más elevado.

5.2. Trabajo futuro

Para que este sistema pueda ser totalmente funcional debe ser posible el reconocimiento de personas de manera *online*, por lo que la continuación de este trabajo consiste en el desarrollo de los módulos presentados en el Capítulo 1 que no se revisaron en este trabajo, así como el sistema para la obtención de las imágenes considerando las posibles variaciones de pose e iluminación y una interfaz gráfica que permita ingresar nuevos individuos al sistema y administrar la información biométrica conocida, también se debe trabajar en aumentar la VR para valores más pequeños de FAR para aumentar el área bajo la curva ROC.

Dirección General de Bibliotecas UNQ

Bibliografía

- [1] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [2] Y. Lei, M. Bennamoun, M. Hayat, and Y. Guo, “An efficient 3d face recognition approach using local geometrical signatures,” *Pattern Recognition*, vol. 47, no. 2, pp. 509–524, 2014. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0031320313003166>
- [3] S. Theodoridis and K. Koutroumbas, “Chapter 1 - introduction,” in *Pattern Recognition*, 4th ed., S. Theodoridis and K. Koutroumbas, Eds. Boston: Academic Press, 2009, pp. 1–12.
- [4] S. Z. Li and A. K. Jain, “Introduction,” in *Handbook of Face Recognition*, 2nd ed., S. Z. Li and A. K. Jain, Eds. Springer-Verlag London, 2011, pp. 1–12.
- [5] T. Kanade, “Picture processing by computer complex and recognition of human faces,” Ph.D. dissertation, Kyoto University, 1973.
- [6] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, “Face recognition: A literature survey,” *ACM Comput. Surv.*, vol. 35, no. 4, p. 399–458, Dec. 2003. [Online]. Available: <https://doi.org/10.1145/954339.954342>
- [7] T. Russ, C. Boehnen, and T. Peters, “3d face recognition using 3d alignment for pca,” in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2, 2006, pp. 1391–1398.
- [8] H. Mohammadzade and D. Hatzinakos, “Iterative closest normal point for 3d face recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 2, pp. 381–397, 2013.
- [9] S. Z. Gilani, A. Mian, F. Shafait, and I. Reid, “Dense 3d face correspondence,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 7, pp. 1584–1598, 2018.

- [10] N. Pears and A. Mian, "3d face recognition," in *3D Imaging, Analysis and Applications*, N. Pears, Y. Liu, and P. Bunting, Eds. London: Springer, 2012, pp. 311–366.
- [11] S. Soltanpour, B. Boufama, and Q. Jonathan Wu, "A survey of local feature methods for 3d face recognition," *Pattern Recognition*, vol. 72, pp. 391–406, 2017. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0031320317303072>
- [12] T. Inan and U. Halici, "3-d face recognition with local shape descriptors," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 577–587, 2012.
- [13] S. Elaiwat, M. Bennamoun, F. Boussaid, and A. El-Sallam, "A curvelet-based approach for textured 3d face recognition," *Pattern Recognition*, vol. 48, no. 4, pp. 1235–1246, 2015. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0031320314004300>
- [14] X. Lu, A. K. Jain, and D. Colbry, "Matching 2.5 d face scans to 3d models," *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 1, pp. 31–43, 2005.
- [15] L. Li, C. Xu, W. Tang, and C. Zhong, "3d face recognition by constructing deformation invariant image," *Pattern Recognition Letters*, vol. 29, no. 10, pp. 1596–1602, 2008.
- [16] N. Werghi, C. Tortorici, S. Berretti, and A. Del Bimbo, "Boosting 3d lbp-based face recognition by fusing shape and texture descriptors on the mesh," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 5, pp. 964–979, 2016.
- [17] Y. Ming, "Robust regional bounding spherical descriptor for 3d face recognition and emotion analysis," *Image and Vision Computing*, vol. 35, pp. 14–22, 2015. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0262885614001875>
- [18] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of cognitive neuroscience*, vol. 3, no. 1, pp. 71–86, 1991. [Online]. Available: <https://direct.mit.edu/jocn/article/3/1/71/3025/Eigenfaces-for-Recognition>
- [19] A. L. Yuille, P. W. Hallinan, and D. S. Cohen, "Feature extraction from faces using deformable templates," *International journal of computer vision*, vol. 8, no. 2, pp. 99–111, 1992.
- [20] Guodong Guo, S. Z. Li, and Kapluk Chan, "Face recognition by support vector machines," in *Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580)*, 2000, pp. 196–201.
- [21] T. Ahonen, A. Hadid, and M. Pietikäinen, "Face recognition with local binary patterns," in *Computer Vision - ECCV 2004*, T. Pajdla and J. Matas, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 469–481.
- [22] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: Application to face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, pp. 2037–2041, 2006.

- [23] P. J. Phillips, Hyeonjoon Moon, S. A. Rizvi, and P. J. Rauss, "The feret evaluation methodology for face-recognition algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1090–1104, 2000.
- [24] P. J. Phillips, P. J. Flynn, T. Scruggs, K. W. Bowyer, Jin Chang, K. Hoffman, J. Marques, Jaesik Min, and W. Worek, "Overview of the face recognition grand challenge," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, 2005, pp. 947–954 vol. 1.
- [25] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," pp. 1–12, 2015. [Online]. Available: <https://ora.ox.ac.uk/objects/uuid:a5f2e93f-2768-45bb-8508-74747f85cad1>
- [26] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 499–515.
- [27] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu, "Cosface: Large margin cosine loss for deep face recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2018/html/Wang_CosFace_Large_Margin_CVPR_2018_paper.html
- [28] D. Yi, Z. Lei, and S. Z. Li, "Towards pose robust face recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 3539–3545. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2013/html/Yi_Towards_Pose_Robust_2013_CVPR_paper.html
- [29] C. Ding and D. Tao, "Robust face recognition via multimodal deep face representation," *IEEE Transactions on Multimedia*, vol. 17, no. 11, pp. 2049–2058, 2015.
- [30] S. Soltanpour and Q. M. J. Wu, "Weighted extreme sparse classifier and local derivative pattern for 3d face recognition," *IEEE Transactions on Image Processing*, vol. 28, no. 6, pp. 3020–3033, 2019.
- [31] Y. Lei, Y. Guo, M. Hayat, M. Bennamoun, and X. Zhou, "A two-phase weighted collaborative representation for 3d partial face recognition with single sample," *Pattern Recognition*, vol. 52, pp. 218–237, 2016. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0031320315003660>
- [32] S. Soltanpour and Q. J. Wu, "Multimodal 2d–3d face recognition using local descriptors: pyramidal shape map and structural context," *IET Biometrics*, vol. 6, no. 1, pp. 27–35, 2016. [Online]. Available: <https://digital-library.theiet.org/content/journals/10.1049/iet-bmt.2015.0120>
- [33] F. R. Al-Osaimi, "A novel multi-purpose matching representation of local 3d surfaces: A rotationally invariant, efficient, and highly discriminative approach with an adjustable sensitivity," *IEEE Transactions on Image Processing*, vol. 25, no. 2, pp. 658–672, 2016.

- [34] M. Emambakhsh and A. Evans, "Nasal patches and curves for expression-robust 3d face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 5, pp. 995–1007, 2017.
- [35] W. Hariri, H. Tabia, N. Farah, A. Benouareth, and D. Declercq, "3d face recognition using covariance based descriptors," *Pattern Recognition Letters*, vol. 78, pp. 1–7, 2016. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0167865516300320>
- [36] M. Hayat, M. Bennamoun, and A. A. El-Sallam, "An rgb-d based image set classification for robust face recognition from kinect data," *Neurocomputing*, vol. 171, pp. 889–900, 2016. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0925231215010048>
- [37] P. Hiremath and M. Hiremath, "3d face recognition based on radon transform, pca, lda using knn and svm." *International Journal of Image, Graphics & Signal Processing*, vol. 6, no. 7, 2014.
- [38] L. Shi, X. Wang, and Y. Shen, "Research on 3d face recognition method based on lbp and svm," *Optik*, vol. 220, p. 165157, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0030402620309931>
- [39] V. Vapnik, "Estimation of dependences based on empirical data," *Addendum 1*, 1982.
- [40] A. Kowalczyk, *Support Vector Machines Succinctly*. Syncfusion, 2017.
- [41] N. Cristianini and J. Shawe-Taylor, *Kernel Methods: an overview*, 1st ed., N. Cristianini and J. Shawe-Taylor, Eds. Cambridge: Cambridge University Press, 2004.
- [42] V. Vapnik, *Estimation of dependences based on empirical data*. Springer Science & Business Media, 2006.
- [43] Shu-Xia Lu and Xi-Zhao Wang, "A comparison among four svm classification methods: Lsvm, nlsvm, ssvm and nsvm," in *Proceedings of 2004 International Conference on Machine Learning and Cybernetics (IEEE Cat. No.04EX826)*, vol. 7, 2004, pp. 4277–4282 vol.7.
- [44] S. Soltanpour and Q. M. J. Wu, "High-order local normal derivative pattern (Lndp) for 3d face recognition," in *2017 IEEE International Conference on Image Processing (ICIP)*, 2017, pp. 2811–2815.
- [45] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002.
- [46] Baochang Zhang, Yongsheng Gao, Sanqiang Zhao, and Jianzhuang Liu, "Local derivative pattern versus local binary pattern: Face recognition with high-order local pattern descriptor," *IEEE Transactions on Image Processing*, vol. 19, no. 2, pp. 533–544, 2010.

- [47] K. Klasing, D. Althoff, D. Wollherr, and M. Buss, “Comparison of surface normal estimation methods for range sensing applications,” in *2009 IEEE International Conference on Robotics and Automation*, 2009, pp. 3206–3211.
- [48] S. Gupta, K. R. Castleman, M. Markey, and A. C. Bovik. Texas 3d face recognition database. [Online]. Available: <http://live.ece.utexas.edu/research/texas3dfr/index.htm>
- [49] S. Gupta, K. R. Castleman, M. K. Markey, and A. C. Bovik, “Texas 3d face recognition database,” in *2010 IEEE Southwest Symposium on Image Analysis Interpretation (SSIAI)*, 2010, pp. 97–100.
- [50] S. Gupta, M. K. Markey, and A. C. Bovik, “Anthropometric 3d face recognition,” *International journal of computer vision*, vol. 90, no. 3, pp. 331–349, 2010.
- [51] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. Fernández del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, “Array programming with NumPy,” *Nature*, vol. 585, p. 357–362, 2020.
- [52] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burrowski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, Í. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python,” *Nature Methods*, vol. 17, pp. 261–272, 2020.
- [53] J. D. Hunter, “Matplotlib: A 2d graphics environment,” *Computing in Science Engineering*, vol. 9, no. 3, pp. 90–95, 2007.
- [54] W. McKinney, “Data structures for statistical computing in python,” in *Proceedings of the 9th Python in Science Conference*, vol. 445. Austin, TX, 2010, pp. 51–56.
- [55] P. Phillips, “Support vector machines applied to face recognition,” *Advances in Neural Information Processing Systems*, vol. 11, pp. 803–809, 1998.
- [56] S. Z. Li and A. K. Jain, *Evaluation methods in face recognition*, 2nd ed. Springer-Verlag London, 2011, pp. 329–347.
- [57] T. Fawcett, “An introduction to roc analysis,” *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006, rOC Analysis in Pattern Recognition. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S016786550500303X>
- [58] P. Campos-Zarate, E. Gorrostieta-Hurtado, and J.-C. Pedraza-Ortega, “Reconocimiento facial basado en least-squares support vector machine y características lndp,” *Research in computing science*, vol. 149, no. 8, pp. 841–851, 2020.

[59] S. Z. Li and A. K. Jain, Eds., *Handbook of Face Recognition*, 2nd ed. Springer-Verlag London, 2011.

Dirección General de Bibliotecas UAQ

Anexos

Dirección General de Bibliotecas UAQ

6.1. Anexo I: Ponencia en COMIA 2020



UNIVERSIDAD AUTÓNOMA
DE CIUDAD JUÁREZ

LA SOCIEDAD MEXICANA DE INTELIGENCIA ARTIFICIAL
Y LA UNIVERSIDAD AUTÓNOMA DE CIUDAD JUÁREZ

OTORGAN ESTE CERTIFICADO A:

Pablo Augusto Campos-Zarate, Efrén Gorrostieta-Hurtado y Jesús Carlos Pedraza-Ortega

POR LA PRESENTACIÓN DEL ARTÍCULO TITULADO:

Reconocimiento facial basado en Least-Square Support Vector Machine y características LNDP.

en el XII Congreso Mexicano de Inteligencia Artificial - COMIA 2020

Cd. Juárez, Chihuahua, México, del 5 al 7 de agosto de 2020



Dr. Félix Castro Espinoza
Presidente SMIA



Dr. Oscar Herrera Aleántara
Presidente Comité de Programa



Dr. Noé A. Castro Sánchez
Presidente Comité de Programa



Dr. Manuel de J. Mandayapa Alfaro
Comité Local COMIA



Sección de Inteligencia Artificial
Sociedad Mexicana de Inteligencia Artificial, A.C.
www.smia.org.mx
Calle de los Batanes 100, Ciudad Juárez, Chihuahua, México D.F. 07739 • Tel: 5621 1234 • Fax: 5621 1234 • Email: info@smia.org.mx

6.2. Anexo II: Artículo publicado

ISSN 1870-4069

Reconocimiento facial basado en Least-Squares Support Vector Machine y características LNDP

Pablo Augusto Campos-Zárate, Efrén Gorrostieta-Hurtado,
Jesús Carlos Pedraza-Ortega

Universidad Autónoma de Querétaro,
Facultad de Ingeniería,
México

{pacaza60, efrengorrostieta}@gmail.com,
caryoko@yahoo.com

Resumen. El reconocimiento facial en 3D se ha vuelto de gran interés para los investigadores en el campo de la visión artificial debido a las ventajas que posee contra el reconocimiento tradicional en 2D, en este trabajo se presenta un sistema de reconocimiento facial en 3D que utiliza características LNDP del rostro junto con un clasificador LSSVM que conserva las ventajas del SVM tradicional sin la necesidad de resolver un problema de programación cuadrático. Los experimentos realizados en la *Texas 3DFRD* muestran que este sistema es capaz de verificar e identificar rostros de la base de datos con valores de 99%.

Palabras clave: Reconocimiento facial, clasificación multiclase, least-squares support vector machine, normales de superficie.

Facial Recognition Based on Least Squares Support Vector Machine and LNDP Features

Abstract. 3D face recognition has become very popular among computer vision researchers due to its advantages against traditional 2D face recognition, this paper presents a 3D face recognition LNDP+LSSVM-based system which keeps the benefits of a traditional SVM without having to solve a quadratic-programming problem but a linear-programming problem. Experiments conducted on the Texas 3DFR Database show that LNDP-LSSVM method can achieve verification and recognition rate over 99%.

Keywords: Face recognition, multiclass classification, least-squares support vector machine, surface normals.

1. Introducción

Un sistema de reconocimiento de rostros puede trabajar de dos formas, para autenticación o para identificación facial [1], la tarea del reconocimiento, que es

6.3. Anexo III: Prueba de idioma FLL



UNIVERSIDAD AUTÓNOMA DE QUERÉTARO
FACULTAD DE LENGUAS Y LETRAS



Campus Aeropuerto, 03 de septiembre de 2020.
FLL/SA/HCA/2020/1018

C. Campos Zárate, Pablo Augusto
Exp. 219781
Facultad de Ingeniería
Presente.

Me permito comunicar a usted que en Sesión Ordinaria del H. Consejo Académico de la Facultad de Lenguas y Letras, celebrada el **03 de septiembre de 2020**, se acordó **Aprobar** su solicitud de **convalidación** de la Constancia del Examen de Manejo de la Lengua (8-), expedida por el CELyC, por el Examen de Comprensión de Textos en Inglés

Sin más por el momento, me es grato quedar a sus órdenes para cualquier aclaración.

Atentamente,
“Enlazar Culturas por la Palabra”


Dra. Ma. de Lourdes Rico Cruz
Secretaria Académica



C.c. Expediente.
Archivo.

MLRC/*mlp