



Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computación

**Desarrollo de un sistema automático para la verificación de firmas
manuscritas**

Trabajo especial de grado
presentado ante la Ilustre
Universidad Central de Venezuela

Por el Bachiller
Juan Estrada

Para optar al título de
Licenciado en Computación

Tutor: Prof. Rhadamés Carmona

Caracas, Octubre de 2018

Resumen

La firma manuscrita es una medida biométrica asociada a la identidad de una persona, y es el método predilecto para la autenticación y verificación de la identidad en una gran variedad de contextos diferentes. Sectores como la banca, la notaría pública y el comercio hacen uso de la firma manuscrita de una persona como prueba de su identidad, y, entre todas las modalidades biométricas existentes en la actualidad, la firma manuscrita cuenta con el mayor grado de aceptación social y legal, por lo que el mayor reto en cuanto a su uso es la automatización de su verificación y la obtención de resultados de alta precisión para evitar autenticaciones falsas o falsos rechazos. Los sistemas automáticos para la verificación de firmas manuscritas que hacen uso de imágenes digitalizadas de firmas son llamados sistemas off-line mientras que los que hacen uso de dispositivos especiales de captación electrónicos son llamados sistemas on-line. Dependiendo de la aplicación de las técnicas de aprendizaje automático, los sistemas pueden ser dependientes o independientes del usuario. Los sistemas para la verificación de firmas manuscritas constan principalmente de 4 pasos: la captación de los datos, el pre-procesamiento de los mismos, la extracción de características y la verificación.

En el desarrollo del presente trabajo se implementaron tres métodos basados en el procesamiento digital de imágenes para la extracción de características con las cuales realizar la verificación off-line de las firmas. Comparamos mediante las métricas de falsa aceptación y falso rechazo varias técnicas de verificación dependientes e independientes del usuario usando las características extraídas. Comparamos el rendimiento de los métodos usando falsificaciones ingenuas y falsificaciones expertas, extraídas de dos bases de datos, una con firmas realizadas con el alfabeto Bengalí e Hindú, y otra con firmas realizadas con el alfabeto latino.

Palabras claves: Aprendizaje automático, verificación biométrica, procesamiento digital de imágenes, minería de datos.

Caracas, 24 de Octubre de 2018
Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computación



ACTA DEL VEREDICTO

Quienes suscriben, Miembros del Jurado designado por el Consejo de la Escuela de Computación para examinar el Trabajo Especial de Grado, presentado por el Bachiller Juan Sebastián Estrada Del Pino, C.I: 20.978.294 con el título "Desarrollo de un sistema automático para la verificación de firmas manuscritas", a los fines de cumplir con el requisito legal para optar al título de Licenciado en Computación, dejan constancia de lo siguiente:

Leído el trabajo por cada uno de los Miembros del Jurado, se fijó el día 24 de Octubre de 2018, a la 1:00pm, para que su autor lo defendiera en forma pública, en el Centro de Computación Gráfica, lo cual realizó mediante una exposición oral de su contenido, y luego respondió satisfactoriamente a las preguntas que le fueron formuladas por el Jurado, todo ello conforme a lo dispuesto en la Ley de Universidades y demás normativas vigentes de la Universidad Central de Venezuela. Finalizada la defensa pública del Trabajo Especial de Grado, el jurado decidió aprobarlo con una calificación de veinte (20) puntos.

En fe de lo cual se levanta la presente acta, en Caracas el 24 de Octubre de 2018, dejándose también constancia de que actuó como Coordinador del Jurado el Profesor Tutor Rhadamés Carmona.

Dr. Rhadamés Carmona
(Tutor)



Prof. Eugenio Scalise
(Jurado Principal)

Prof. Francisco Moreno
(Jurado Principal)

Índice general

1. Introducción	10
2. Descripción del problema	12
2.1. Solución propuesta	13
2.2. Objetivo general	14
2.3. Objetivos específicos	14
2.4. Metodología de desarrollo	15
2.5. Plataforma de Software	16
2.6. Plataforma de Hardware	16
3. Elementos del procesamiento digital de imágenes	17
3.1. Histogramas	17
3.1.1. Umbralización mediante histogramas	18
3.1.2. Ecuilización de histogramas	18
3.2. Filtros digitales	18
3.2.1. Filtros y operaciones en el dominio espacial	20
3.2.2. Filtros estadísticos	23
3.2.3. Filtros en el dominio de frecuencias	23
3.3. Operaciones Geométricas	28
3.4. Operaciones Morfológicas	30
3.4.1. Elemento estructurante	31
3.4.2. Ajuste e intersección	31
4. Aprendizaje Automático	34
4.1. Aprendizaje Automático	34
4.2. Aprendizaje Supervisado	35
4.2.1. Regresión	35
4.2.2. Máquinas de vectores de soporte	36

4.2.3.	Redes neuronales artificiales	36
4.3.	Aprendizaje no supervisado	39
4.3.1.	Análisis de componentes principales	40
4.3.2.	k -Medias	40
5.	Antecedentes	42
5.1.	Sobre los sistemas de verificación	42
5.2.	Métricas de evaluación	44
5.3.	Adquisición de datos	45
5.4.	Pre-procesamiento	46
5.5.	Extracción de características	47
5.6.	Verificación	50
6.	Diseño e implementación	55
6.1.	Bases de datos	56
6.2.	Pre-procesamiento	56
6.3.	Extracción de características	57
6.3.1.	Morfología matemática y características globales	57
6.3.2.	Centros Geométricos	60
6.3.3.	HOG y LBP	62
6.4.	Verificación	64
6.4.1.	Morfología matemática, HOG y LBP	64
6.4.2.	Centros geométricos	65
7.	Resultados	68
7.1.	Análisis de las características	69
7.2.	Modelos independientes de usuario	70
7.2.1.	Morfología matemática y métricas globales.	70
7.2.2.	HOG y LBP	71
7.2.3.	Modelo basado en morfología matemática con métricas globales y HOG+LBP	73
7.3.	Modelos dependiente de usuario	74
7.3.1.	Morfología matemática y métricas globales.	74
7.3.2.	HOG y LBP	75
7.3.3.	Morfología matemática con métricas globales y HOG+LBP	75
7.4.	Modelos Globales	76
7.4.1.	Modelo basado en PCA	76

7.4.2. Modelo basado en morfología matemática con métricas globales	78
7.4.3. Modelo global basado en HOG+LBP	79
7.4.4. Modelo basado en morfología matemática con métricas globales y HOG+LBP	79
7.5. Modelo basado en centros geométricos	80
7.6. Comparación contra el estado del arte	80
8. Conclusiones y Trabajos Futuros	83
9. Anexos	86

Índice de figuras

3.1. Umbralización mediante histogramas.	19
3.2. Umbralización dinámica.	19
3.3. Ecualización de un histograma.	19
3.4. Especificación de histogramas.	20
3.5. Matriz de convolución.	21
3.6. Matriz de convolución.	21
3.7. Filtros de paso alto y bajo.	24
3.8. Filtro de Sobel.	25
3.9. Filtro de Prewitt	25
3.10. Filtro no-lineal.	26
3.11. Resultado de aplicar la transformación discreta de Fourier. . .	27
3.12. Aplicación de un filtro no-lineal en el espacio de las frecuencias.	27
3.13. Aplicación de una transformación afín.	29
3.14. Aplicación de una transformación proyectiva.	30
3.15. Aplicación de una transformación no-lineal.	30
3.16. Aplicación de una operación de dilatación.	32
3.17. Aplicación de una operación de erosión.	32
3.18. Operaciones morfológicas sobre imágenes a escala de grises. . .	33
4.1. Vector soporte.	37
4.2. Perceptrón multicapa.	38
4.3. Red neuronal recurrente.	38
4.4. Red neuronal de convolución.	39
6.1. Ejemplos de las firmas dentro de las bases de datos usadas. . .	57
6.2. Ejemplo del método basado en morfología matemática.	59
6.3. Aplicación del método basado en centros geométricos	61
6.4. Representación visual de las características extraídas mediante HOG y LBP	63

6.5. Parámetros k y d	66
7.1. Varianza de las diversas características.	69
7.2. Resultados para el método de morfología matemática.	70
7.3. Resultados para el método de morfología matemática y métricas globales.	71
7.4. Resultados para el método de HOG y LBP con una ventana de 32 píxeles.	72
7.5. Resultados para el método de HOG sin LBP con una ventana de 32 píxeles.	73
7.6. Resultados para el método de HOG y LBP con una ventana de 64 píxeles.	73
7.7. Resultados para el método combinado	74
7.8. Resultados para el método de morfología matemática más métricas globales dependiente de usuario.	75
7.9. Resultados para el método de HOG y LBP con una ventana de 32 píxeles dependiente de usuario.	76
7.10. Resultados para el método de HOG y LBP con una ventana de 64 píxeles dependiente de usuario.	77
7.11. Resultados para el método combinado dependiente de usuario.	77
7.12. Resultados para el método basado en PCA.	78
7.13. Resultados para el modelo global basado en morfología matemática y métricas globales.	78
7.14. Resultados para el modelo global basado en HOG+LBP.	79
7.15. Resultados para el modelo global basado en morfología matemática con métricas globales y HOG+LBP	80
7.16. Resultados para el modelo basado en centros geométricos.	82
9.1. Resultados para el modelo basado en morfología matemática, verificado con falsificaciones ingenuas.	86
9.2. Resultados para el modelo basado en morfología matemática más métricas globales, verificado con falsificaciones ingenuas.	87
9.3. Resultados para el modelo basado en HOG+LBP con una ventana de 32 píxeles, verificado con falsificaciones ingenuas.	87
9.4. Resultados para el modelo basado en HOG sin LBP con una ventana de 32 píxeles, verificado con falsificaciones ingenuas.	88
9.5. Resultados para el modelo basado en HOG+LBP con una ventana de 64 píxeles, verificado con falsificaciones ingenuas.	88

9.6.	Resultados para el modelo basado en morfología matemática con métricas globales y HOG+LBP con una ventana de 32 píxeles, verificado con falsificaciones ingenuas.	89
9.7.	Resultados para los modelos dependiente de usuario basados en morfología matemática con métricas globales, verificados con falsificaciones ingenuas.	89
9.8.	Resultados para los modelos dependiente de usuario basados en HOG+LBP con una ventana de 32 píxeles, verificados con falsificaciones ingenuas.	90
9.9.	Resultados para los modelos dependiente de usuario basados en HOG+LBP con una ventana de 64 píxeles, verificados con falsificaciones ingenuas.	90
9.10.	Resultados para los modelos dependiente de usuario basados en morfología matemática con métricas globales y HOG+LBP con una ventana de 32 píxeles, verificados con falsificaciones ingenuas.	91
9.11.	Resultados para el modelo basado en PCA, verificado con falsificaciones ingenuas.	91
9.12.	Resultados para el modelo global basado en morfología matemática con métricas globales y HOG+LBP con una ventana de 32 píxeles, verificado con falsificaciones ingenuas.	92
9.13.	Resultados para el modelo global basado en HOG+LBP con una ventana de 32 píxeles, verificado con falsificaciones ingenuas.	92
9.14.	Resultados para el modelo global basado en morfología matemática con métricas globales, verificado con falsificaciones ingenuas.	93
9.15.	Resultados para el modelo basado en centros geométricos, verificado con falsificaciones ingenuas.	93
9.16.	Diferencia entre el modelo global basado en HOG+LBP y el mismo método entrenado con cada base de datos.	94
9.17.	Diferencia entre el modelo global basado en morfología matemática más métricas globales y el mismo método entrenado con cada base de datos.	95
9.18.	Diferencia entre el modelo global combinado y el mismo método entrenado con cada base de datos.	96
9.19.	Diferencia de las métricas de evaluación entre el método basado en morfología matemática más métricas globales y el método basado en HOG+LBP.	97

9.20. Diferencia de las métricas de evaluación entre el método basado en HOG+LBP y HOG únicamente.	98
9.21. Diferencia de las métricas de evaluación entre el método basado en HOG+LBP y la combinación de todas las características.	99
9.22. Diferencia de las métricas de evaluación entre el método basado en morfología matemática más métricas globales y la combinación de todas las características.	100

Índice de cuadros

5.1. Resultados de Bhunia et al.	53
5.2. Resultados de Dey et al.	53
5.3. Resultados de Haffemann et al.	54
5.4. Resultados de Yilmaz et al.	54
6.1. Resultados de la aplicación de la disyunción exclusiva en la morfología matemática	58
7.1. Estado del arte con la base de datos CEDAR.	81
7.2. Estado del arte con la base de datos BHSig260.	81

Capítulo 1

Introducción

La firma manuscrita es una medida biométrica asociada a la identidad de una persona, y es usada como el método predilecto para la autenticación y verificación de la identidad en una gran variedad de contextos diferentes, por ejemplo, sectores como la banca, la notaría pública y el comercio hacen uso de la firma manuscrita de una persona como prueba de su identidad [40]. La firma manuscrita es un método de verificación de identidad válido cuando se requiere una prueba legalmente válida del conocimiento, aprobación u obligación de dicha persona en cuanto a la elaboración de un documento o del contenido del mismo [18].

En contraste con otros métodos biométricos, la firma no presenta un poder discriminatorio necesariamente mayor; sin embargo, se destaca por su conveniencia y la austeridad en cuanto a los instrumentos necesarios para su captación: mientras los métodos basados en la retina, en las huellas dactilares o en la voz requieren sofisticados dispositivos de captación, basta con un lápiz y un papel para plasmar una firma. Otra ventaja que presenta la firma manuscrita sobre otros métodos de identificación es su capacidad de ser cancelada en caso de *spoofing*¹, pues, en caso de que la forma de la firma sea conocida por un tercero, es posible cambiarla, ya que las características discriminatorias de una firma no son más que el producto de la configuración neuronal y biomecánica propia de cada individuo.

Entre todas las modalidades biométricas existentes en la actualidad, la firma manuscrita es la que cuenta con el mayor grado de aceptación social y legal, por lo que el mayor reto en cuanto a su uso, es la automatización

¹Una situación en la cual una persona o un programa se hace pasar por otro habiendo robado o falsificado el método de autenticación, en este caso, la falsificación de la firma

de su verificación y la obtención de resultados de alta precisión para evitar autenticaciones falsas o falsos rechazos [38].

En la literatura podemos encontrar una plétora de esquemas de verificación. Algunos hacen uso de dispositivos especiales de captura, tales como tablas digitalizadoras, para usar información presente en el proceso de producción (el ángulo del lapiz, la rapidez de trazo, etc) como características discriminantes. Este tipo de esquemas son llamados *on-line*. Otros, hacen uso de la imagen digital de la firma para extraer dicha información; este tipo de esquemas son llamados *off-line*.

A lo largo del presente trabajo implementamos varios métodos para la extracción de características discriminantes de firmas manuscritas, y con estas entrenamos varios modelos de aprendizaje automático para su verificación. El primer método consiste en la aplicación de operaciones morfológicas sobre la imagen de una firma para crear distintos umbrales de variación permitida. Al comparar estos umbrales con otra firma del mismo usuario obtenemos una métrica de la variación entre ambas. Esta información, junto con otras características de carácter global son usadas para entrenar los modelos a usar en la verificación.

Para el segundo método, mediante un algoritmo recursivo creamos un conjunto de puntos que describen los centros de masa prototípicos de las firmas para un usuario. Midiendo distintas firmas de un usuario contra este prototipo conseguimos un umbral que representa la varianza usual entre sus firmas, y mediante este umbral podemos medir el grado de similitud entre distintas firmas y verificarlas.

Finalmente, el último método usa información alrededor de los centros de masa de varias subdivisiones de una firma mediante las características de histograma de gradientes orientados y patrones locales binarios, para entrenar varios clasificadores para la verificación de las firmas.

Este documento está estructurado en 9 capítulos. En el capítulo 2 presentamos la descripción del problema, en el capítulo 3 se tratan los elementos básicos del procesamiento digital de imágenes. En el capítulo 4 se desarrollan algunos conceptos básicos sobre el aprendizaje automático. En el capítulo 5, se exponen algunos trabajos previos, sus puntos en común y sus particularidades. En el capítulo 6 se presenta las consideraciones para el diseño y la implementación del sistema. En el capítulo 7 presentamos los resultados de dicho sistema, en el capítulo 8 presentamos nuestras conclusiones así como ideas para posibles trabajos futuros, y en el capítulo 9 encontramos los anexos.

Capítulo 2

Descripción del problema

La autenticación de la identidad es un paso involucrado en la gran mayoría de transacciones comerciales e interacciones de índole legal entre individuos. Entre los métodos biométricos usados para autenticar la identidad de una persona y relacionarla inequívocamente con el portador, la firma manuscrita está entre los más convenientes y entre aquellos que cuentan con la más amplia adopción. Incluso traspasa fronteras: es un mecanismo compartido internacionalmente entre regiones con culturas y escrituras distintas como China e India.

Sin embargo, las firmas manuscritas pueden ser falsificadas por terceros. Para un ojo experto identificar la falsificación es una tarea razonablemente sencilla, ya que por más laborioso que sea el intento de apropiación, siempre hay detalles que identifican unívocamente al verdadero dueño de la firma. Aún así, no es posible someter todas las firmas recreadas en un instante dado a escrutinio por parte de un ojo experto. Considerando esto, desde principios de la década de los 90 se han venido desarrollando esquemas y técnicas para la verificación automática (por parte de un computador) de las firmas manuscritas con niveles variados de éxito.

La verificación de firmas no es un problema que se considere aún resuelto, pues todavía no existe un sistema con márgenes de error considerado insignificante, siendo que el estado del arte provee una precisión cercana al 90 % (figura 5.2). La falsificación de firmas manuscritas sigue siendo un problema común extendido por todo el mundo, que además de generar graves daños materiales a las víctimas, ocasiona enormes problemas legales a las instituciones que se vean involucradas en estos delitos.

Existen varios enfoques para la verificación de firmas manuscritas, todas

con el objetivo común de usar características discriminatorias con alto poder predictivo para separar las firmas falsas de las firmas auténticas. Los enfoques varían entre las características discriminatorias que usan, el método que arroja la predicción o que aprende la función de discriminación, y cómo se desenvuelve el proceso de captación de datos (On-line u off-line).

Gracias al auge del poder de cómputo en las últimas dos décadas es posible hoy en día eximirse de la ingeniería de características discriminatorias, siendo posible utilizar modelos del aprendizaje supervisado para que extraigan por medio de la optimización de millares de parámetros, características discriminatorias nuevas de forma automática, sin la intervención adicional del ingenio humano.

El creciente poder computacional no sólo posibilita el descubrimiento de características discriminatorias de forma automática, sino que también nos permite agregar nuevas características discriminatorias distintas a las usadas en modelos predictivos ya existentes, pudiéndose hoy en día complementar modelos predictivos con una mayor cantidad de parámetros y grados de libertad con el objeto de mejorar su efectividad. Por lo tanto, hoy en día es posible utilizar una amplia variedad de características discriminatorias, de forma que los casos extremos en donde algunas fallan, estas puedan ser subsanados por aquellas que los presentan bajo sus fortalezas.

Con esto en consideración, puede observarse como posibilidad utilizar el creciente volumen de datos y el auge en el poder de cómputo para la creación de técnicas para la verificación de firmas; se sabe que hoy, existe el volumen de datos necesarios para desarrollar los modelos predictivos más efectivos y complejos de los que se tenga conocimiento.

2.1. Solución propuesta

Planteamos desarrollar y modificar 3 métodos de verificación de firmas offline, dos de ellos ya existentes. Los métodos planteados difieren entre sí en su grado de refinamiento y complejidad, y por consiguiente, planteamos comparar el rendimiento de los métodos. Para el desarrollo del presente planteamiento, disponemos de 3 conjuntos de datos de firmas escritas en distintos alfabetos. La base de datos CEDAR utiliza el alfabeto occidental, y posee 55 usuarios con 24 firmas genuinas y 24 falsificaciones expertas para cada uno. La segunda base de datos contiene 2 sub-conjuntos de firmas. El primero consta de las firmas de 100 usuarios, 24 genuinas y 30 falsificaciones expertas

para cada uno escritas con el alfabeto bengalí, mientras que el segundo posee las mismas características, solo que para 160 usuarios escritas con el alfabeto hindú.

En primer lugar, desarrollaremos los métodos y algoritmos necesarios para preprocesar las imágenes de las firmas. El primer método a implementar considera que el proceso de producción de una firma presenta una importante variación intra-clase, es decir, que para distintos intentos de una misma persona de producir una firma, habrán diferencias en el producto final. Mediante este método pretendemos establecer un margen donde las variaciones intra-clase son aceptadas, y aquellas que posean una varianza mayor a estas sean consideradas como pertenecientes a una clase distinta (la firma es falsa, o no pertenece al usuario) mediante el uso de operaciones morfológicas sobre la imagen.

El segundo método se basa en el cálculo de una serie de centros geométricos a lo largo de las distintas firmas de un usuario para establecer un conjunto de centros prototípicos, y las distancias medias de los centros geométricos de las firmas del individuo con respecto a este conjunto prototípico. Con esta información establecemos un umbral que establece la suma máxima aceptable de estas distancias para aceptar una firma como genuina. Por último, el último método se basa en la obtención de información local alrededor de un número de centros geométricos dentro de las firmas, mediante de un par de métodos basados en histogramas.

2.2. Objetivo general

Implementar 3 métodos offline para la extracción de características de firmas manuscritas, usando distintos métodos de aprendizaje automático para su verificación y comparar su rendimiento a lo largo de un conjunto de bases de datos de firmas.

2.3. Objetivos específicos

- Recolectar, preprocesar y limpiar los datos a ser usados para las etapas subsiguientes. La limpieza de los datos y su preprocesamiento es común para todos los métodos planteados.

- Generar los vectores característicos mediante la extracción de características usando los métodos planteados.
- Entrenar los diversos métodos de clasificación usando los vectores característicos previamente extraídos.
- Evaluar el desempeño de los modelos entre sí, mediante las métricas de falso rechazo y falsa aceptación, usando un sub-conjunto de los datos distinto al usado en la etapa de entrenamiento.

2.4. Metodología de desarrollo

Usaremos la metodología común para la minería de datos. Esto es:

- **Recolección de los datos:** Recolectamos los datos relevantes para el problema propuesto. Específicamente, estos son los datos o las observaciones sobre las cuales se entrenaran los modelos posteriores, o los datos sobre los cuales se desempeñará la evaluación de los mismos.
- **Selección de características:** Seleccionamos las características que puedan ser útiles y relevantes para el problema a tratar. La buena selección de características es fundamental para obtener un modelo que presente un buen rendimiento.
- **Elección del algoritmo:** Elegimos el algoritmo de aprendizaje automático que se desempeñará sobre las características extraídas de los datos recolectados.
- **Elección de parámetros:** Los algoritmos electos en el punto anterior probablemente tengan un conjunto de parámetros que influyan en su funcionamiento. Mediante el conocimiento previo, o la realización de experimentos, elegimos los parámetros adecuados en pro de conseguir el mejor rendimiento posible.
- **Entrenamiento:** Se ajustan los algoritmos escogidos sobre los datos recolectados para que, con suerte, aprenda relaciones entre los datos que puedan ser usados para discriminar entre observaciones y realizar predicciones.

- Evaluación: Evaluaremos las prestaciones de los algoritmos con datos distintos a aquellos utilizados para la etapa de entrenamiento, para tener así un punto de referencia que nos otorgue una idea sobre su utilidad o su eficacia para el problema a tratar.

2.5. Plataforma de Software

El sistema de desarrollará y se evaluará sobre el sistema operativo Ubuntu 17.10. **OpenCV**, una librería de procesamiento de imágenes, será utilizada para las etapas de pre-procesamiento y extracción de características. Para el entrenamiento de algunos modelos tales como los basados en redes neuronales artificiales, se usará la librería **Keras** para la definición, entrenamiento y verificación de estos, y para otros, como aquellos basados en k -vecinos, Random Forest o máquinas de vectores de soporte, se utilizará **Scikit-learn**, una librería de python para las tareas de minería de datos.

2.6. Plataforma de Hardware

Las características del computador donde se desarrollará y probará el sistema son las siguientes:

- Computador con procesador Intel Core i5-450M
- 4GB de memoria RAM DDR3
- Tarjeta Gráfica Nvidia GeForce 310m
- Un Disco Duro de estado sólido de 128GB, SATA III (6Gbps)

Capítulo 3

Elementos del procesamiento digital de imágenes

En este capítulo se describen algunos conceptos básicos asociados al procesamiento digital de imágenes que son pertinentes para la elaboración del presente trabajo, encontrando su aplicación en la extracción de características y pre-procesamiento de los datos, principalmente. En la sección 3.1 se definen algunos conceptos sobre los histogramas y las operaciones que pueden realizarse a través de estos. En la sección 3.2 se definen conceptos concernientes filtros digitales, mientras que en la sección 3.3 se tratarán algunos conceptos de las operaciones geométricas y morfológicas.

3.1. Histogramas

Los histogramas son la base de muchas técnicas del procesado de imágenes sobre el dominio espacial, y proveen valiosas estadísticas sobre la imagen de gran utilidad a la hora de realizar otro tipo de operaciones, tales como la compresión y la segmentación. El histograma de una imagen digital con intensidades $[0, L - 1]$ puede entenderse como una función discreta $h(r_k) = n_k$, donde r_k es la k -ésima intensidad perteneciente a un canal determinado de una imagen, y n_k es el número total de píxeles con intensidad $r_k \in [0, L - 1]$. Es común normalizar un histograma dividiendo las observaciones de una intensidad n_k por el número total de píxeles en la imagen, denotado por N , de forma que $h(r_k) = \frac{n_k}{N}$ proporcione la probabilidad $p(n_k)$, es decir, la probabilidad de que un píxel cualquiera dentro de la imagen posea una

intensidad r_k [41].

3.1.1. Umbralización mediante histogramas

La umbralización mediante histogramas es una técnica útil para la segmentación y binarización de imágenes. Con un histograma se puede estudiar de forma global o adaptativa las distribuciones de las intensidades de los píxeles de una imagen, así, se pueden asignar valores arbitrarios a píxeles que estén a una distancia dada del centro de las distribuciones. En la figura 3.1 se puede observar que el objeto en primer plano es fácilmente separable del fondo mediante la binarización, pues las intensidades dentro de la imagen tienen dos centros bien definidos. Esta operación es práctica para imágenes con una iluminación homogénea, pero para imágenes con iluminación más variable, se puede realizar la misma operación utilizando un umbral distinto para cada píxel, que dependa del histograma de los píxeles vecinos [10]. Ejemplo de esta técnica es apreciable en la figura 3.2.

3.1.2. Ecuilización de histogramas

Para mejorar el contraste en imágenes que presenten distribuciones de intensidades con centros bien definidos, se pueden manipular los valores de los píxeles para disminuir la varianza entre las intensidades, y de esta forma aumentar el rango dinámico. En el caso específico de la ecualización de histogramas (3.3), se busca mantener el promedio de las intensidades mientras se disminuye la varianza, pero en ocasiones los resultados pueden ser indeseables.

Un concepto relacionado es el de especificación de histogramas (3.4). La idea principal es que, dada una imagen M con un histograma $h_M(r_k)$, se busca una transformación $f(x)$ de los valores de intensidad en M , de forma que el resultado de la transformación M_2 posea el histograma $h_{M_2}(r_k)$ deseado [10].

3.2. Filtros digitales

El concepto de filtrado tiene su origen en el uso de la transformada de Fourier para el procesado de señales en el dominio de la frecuencia [41].

Las imágenes representan variaciones de la intensidad, pudiéndose entender la intensidad como un componente representativo de brillo o color que



Figura 3.1: Umbralización mediante histogramas.

La imagen tiene gran contraste entre el los objetos en primer plano y el fondo, posibilitando la umbralización por medio de histogramas [10]

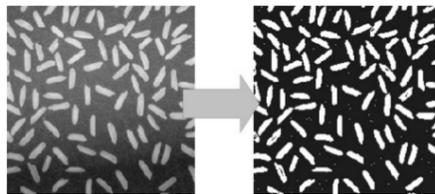


Figura 3.2: Umbralización dinámica.

La iluminación irregular en la imagen la vuelve una candidata perfecta para el umbralizado dinámico [10]



Figura 3.3: Ecualización de un histograma.

El resultado de la ecualización de un histograma [10]



Figura 3.4: Especificación de histogramas.

El resultado de la especificación de histogramas. La imagen superior izquierda es la original [10]

varía en el espacio-imagen. Es posible manipular este componente dependiendo de las relaciones que tiene un punto dado de la imagen con sus vecinos en el dominio espacial, así mismo, también existe representaciones (3.2.3) de una imagen basada en la frecuencia (y la fase de la misma) con que varían las intensidades de sus componentes.

Es factible representar una imagen como un espectro de diferentes frecuencias y transformar de vuelta esta representación al dominio espacial sin ninguna pérdida de información; de igual manera, es posible manipular dicha representación espectral con filtros que incidan en el dominio espacial de la imagen.

3.2.1. Filtros y operaciones en el dominio espacial

El proceso de filtrado en el dominio espacial consiste en el desplazamiento de una máscara de filtrado (o matriz de convolución, fig. 3.5) sobre el dominio de la imagen, calculando la “respuesta” del filtro con una relación predefinida. Los filtros lineales computan esta respuesta con la suma (3.1) de los productos entre los coeficientes del filtro y los píxeles de la imagen bajo el área de la

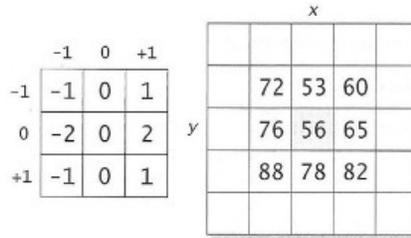


Figura 3.5: Matriz de convolución.

La matriz izquierda es la matriz de convolución, mientras que la derecha representa la imagen [20]

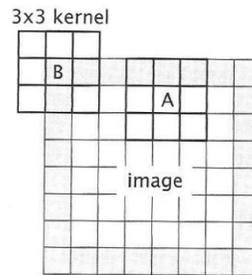


Figura 3.6: Matriz de convolución.

La matriz de convolución recorre punto a punto la imagen, y determina su respuesta con la suma de la multiplicación de los factores $w(s, t)f(x + s, y + t)$ [20]

máscara (3.6).

$$g(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t)f(x + s, y + t) \quad (3.1)$$

Los filtros no lineales pueden calcular la respuesta del filtro de diversas formas, por ejemplo, calculando la mediana de los productos entre los píxeles de la máscara y los píxeles de la imagen [41].

Dependiendo de la máscara o matriz de convolución ($w(s, t)$), existen 3 tipos de operaciones principales [20]:

Filtros lineales: Dependiendo de los elementos de la matriz de convolución, un filtro lineal puede ser de paso bajo o paso alto.

En los filtros de paso bajo, todos los elementos de la matriz de convolución son positivos (3.2). Este tipo de filtros permite que se eliminen o suavicen las características de alta frecuencia en la imagen.

$$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (3.2)$$

Los filtros de paso alto se logran con una combinación de coeficientes positivos y negativos. Un filtro de paso alto que contemple todas las direcciones (una máscara cuadrada) debe tener valores positivos cerca del centro, y valores negativos en la periferia (3.3) Ejemplo de los efectos de ambos tipos de filtros pueden verse en la fig. 3.7.

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad (3.3)$$

Filtros para la detección de bordes: Una de las mayores aplicaciones para las operaciones de convolución es la detección de bordes. Un borde puede definirse como una transición abrupta entre distintas intensidades o colores en una imagen. La mayor parte de las técnicas de convolución para la detección de bordes están basadas en la computación del gradiente discreto. En una imagen el gradiente en y puede ser aproximado de la siguiente forma:

$$g_y(x, y) = f(x, y + 1) - f(x, y - 1) \quad (3.4)$$

y para x :

$$g_x(x, y) = f(x + 1, y) - f(x - 1, y) \quad (3.5)$$

De la representación matricial de las derivadas parciales anteriores $\begin{bmatrix} -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}^T$ se han derivado varias matrices de convolución para la detección de bordes, dos de las más usadas son las matrices de Prewitt (3.6) y Sobel (3.7). La única diferencia entre ambas es que la matriz de Sobel da más importancia a los píxeles alineados en el eje.

$$h_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}, h_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad (3.6)$$

$$h_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, h_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad (3.7)$$

$$\|h\| = \sqrt{h_x^2 + h_y^2} \quad (3.8)$$

$$\theta = \arctan\left(\frac{h_x}{h_y}\right) \quad (3.9)$$

3.2.2. Filtros estadísticos

Los filtros estadísticos son filtros no-lineales con características heterogéneas. Usando el principio de la máscara de convolución, se pueden aplicar operaciones de varios tipos sin estar limitado a las operaciones matemáticas elementales. Por ejemplo, el filtro de la mediana es un filtro de suavizado con excelentes prestaciones al momento de eliminar ruido mientras se intenta preservar las características de alta frecuencia de una imagen [20]. En lugar de promediar la vecindad de un píxel dado con una matriz de convolución, se toma el valor mediano alrededor de los píxeles con una distancia dada en la periferia. Este tipo de filtros pueden ser generalizados para aceptar varias operaciones no lineales asociadas con el rango del valor de un píxel con respecto a los de su vecindad: Mínimo, máximo, rangos arbitrarios, el valor más cercano a una desviación estándar, etc [10].

3.2.3. Filtros en el dominio de frecuencias

Las técnicas para el análisis y la manipulación de la frecuencia espacial están basadas en la teoría de Fourier. La idea principal detrás de esta teoría es que, sin importar que tan compleja sea una función, puede ser representada como una suma de varias funciones sinusoidales (funciones base). Una suma ponderada (donde cada término tiene una importancia variable) de estas funciones base se conoce como una serie de Fourier, y los factores de sesgo de cada función reciben el nombre de coeficientes de Fourier. La transformación desde el dominio espacial de una imagen hasta su dominio espectral cuenta con tres consideraciones:



Figura 3.7: Filtros de paso alto y bajo.

La imagen original se encuentra a la izquierda en ambas comparaciones. En la fila superior, la imagen derecha corresponde al resultado de un filtro de paso bajo, mientras que en la inferior, corresponde al resultado de un filtro de paso alto [22]

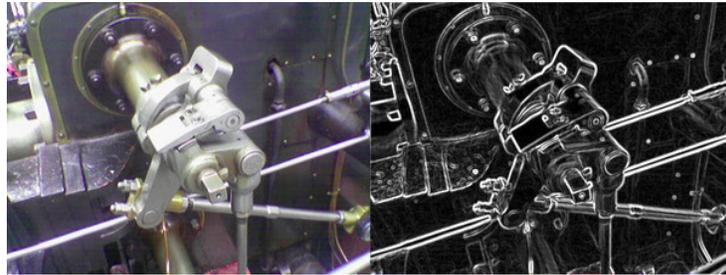


Figura 3.8: Filtro de Sobel.

La imagen original se encuentra a la izquierda. La imagen derecha es el resultado de computar la magnitud del gradiente después de aplicar el filtro de Sobel [16]
Filtros de paso alto y bajo.



Figura 3.9: Filtro de Prewitt.

La imagen original se encuentra a la izquierda. La imagen derecha es el resultado de computar la magnitud del gradiente después de aplicar el filtro de Prewitt [15].

Bidimensionalidad: La transformación deberá ser aplicada a una función de dos variables, $f(x, y)$, que representa el componente de color o brillo en el punto (x, y) .

Muestreo: La función $f(x, y)$ no es una función continua, sino un muestreo discreto del espacio.

Dominio Finito: El dominio sobre el que está definida una imagen $M_{n,m}$ es finito, y va desde $x, y = 0$ hasta $x = n, y = m$.

Estas consideraciones dan paso a la transformada discreta de Fourier (TDF). La TDF es por lo general aplicada sobre imágenes con dimensiones cuadradas ($n = m$) y su expresión es la siguiente:

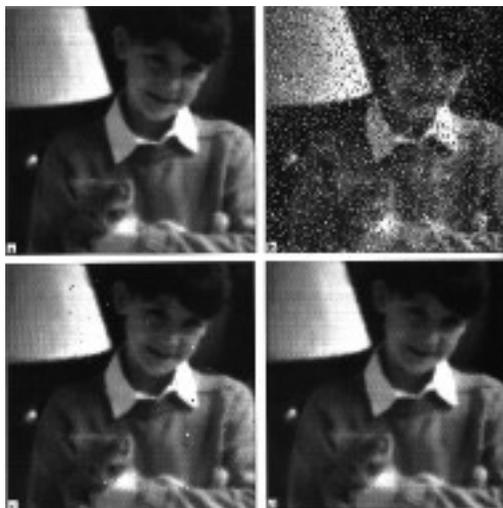


Figura 3.10: Filtro no-lineal.

La imagen superior izquierda es la original, y a la derecha se le añadió ruido. Las imágenes inferiores son el resultado de aplicar un filtro estadístico (la mediana) de 3×3 (izquierda) y 5×5 (derecha) a la imagen alterada [4].

$$F(u, v) = \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \exp \frac{-j2\pi(ux+vy)}{N}$$

Pero puede ser reformulada para imágenes rectangulares [41] :

$$F(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \exp^{-j2\pi(\frac{ux}{M} + \frac{vy}{N})}$$

La TDF resulta en un número complejo, cuyos componentes reales e imaginarios no significan mucho por sí mismos, pero con las siguientes ecuaciones podemos computar la magnitud (3.10) y la fase (3.11) de las funciones base:

$$\|F(u, v)\| = \sqrt{R^2(u, v) + I^2(u, v)} \quad (3.10)$$

$$\theta(u, v) = \tan^{-1} \left[\frac{I(u, v)}{R(u, v)} \right] \quad (3.11)$$

Los resultados de las ecuaciones anteriores pueden ser representados a su vez como imágenes :

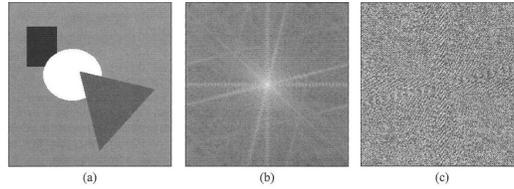


Figura 3.11: Resultado de aplicar la transformación discreta de Fourier.

(a) es la imagen original, (b) es la representación de la magnitud (3.10), y (c) es la representación de la fase (3.11) [20].

Una vez obtenida la representación de la TDF, es posible aplicar los mismos filtros y operaciones del dominio espacial a las imágenes asociadas a la magnitud de la frecuencia y la fase. Por ejemplo, se puede aplicar un filtro de convolución no-lineal para aumentar el brillo de las regiones de alta frecuencia de la imagen original: La imagen también ilustra la posibilidad de revertir el



Figura 3.12: Aplicación de un filtro no-lineal en el espacio de las frecuencias.

La aplicación de un filtro no-lineal, que aumenta el brillo de los píxeles de forma proporcional a la distancia del píxel al centro de la imagen [9]

proceso de transformación, lo cual se logra aplicando la transformada inversa a las imágenes de la magnitud y la fase. La definición de la transformada inversa es la siguiente:

$$F(u, v) = \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \exp^{j\frac{2\pi}{N}(ux+vy)}$$

La única diferencia es el signo del exponente $\frac{j2\pi(ux+vy)}{N}$, que pasa de ser negativo a ser positivo.

Existen otras transformaciones útiles para representar una imagen en su dominio espectral, relacionadas con la transformada de Fourier. Dos casos que vale la pena mencionar por su uso generalizado:

Transformada discreta de Ondícula: Esta transformación, similar a la transformada de Fourier, usa como funciones base ondículas, ondas definidas en un intervalo discreto que empiezan y terminan con amplitud 0. A diferencia de las ondas sinusoidales, las ondículas pueden variar en su localización dentro de su dominio, y aplicando la transformada de ondículas se puede obtener como resultado la correlación entre una ondícula cualquiera y la imagen [14] [41]. La posibilidad de poder construir la ondícula de forma arbitraria posibilita que se pueda usar esta transformación para la adquisición de características arbitrarias dentro de una imagen, por ello, es usualmente usada con este propósito, así como también para la mejora de imágenes (por ejemplo, la eliminación de ruido [39]) y la compresión [14].

Transformada discreta del coseno: Esta transformación expresa la secuencia finita de datos de una imagen en términos de la suma de una serie de funciones del coseno con diferentes frecuencias. La diferencia clave con respecto a la transformada de Fourier es que la transformada del coseno sólo hace uso del componente real (a diferencia de la transformada de Fourier, que tiene componentes reales e imaginarios) [13]. Es comúnmente usada para la compresión y para la extracción de características [17].

3.3. Operaciones Geométricas

Las operaciones geométricas se hacen necesarias para subsanar errores y distorsiones ópticas introducidas por el lente de una cámara, la deformación geométrica de una imagen para conformar a un estándar, etc. Existen multitud de operaciones geométricas, he aquí tres de las más básicas [10]:

Transformaciones afines: Este tipo de transformaciones preservan la forma general de la imagen, y se componen de operaciones de traslación, rotación, escalamiento y shearing.

Transformaciones proyectivas: Las transformaciones proyectivas se diferencian de las afines al no mantener la congruencia entre ángulos. Estas transformaciones se definen a través del mapeo entre un cuadrilátero arbitrario en el espacio-objeto y otro ubicado en el espacio-imagen.

Transformaciones no-lineales: las transformaciones no-lineales no preservan las características de las líneas dentro de una imagen, como el paralelismo entre líneas o la congruencia de los ángulos entre vectores. Estas operaciones son particularmente útiles a la hora de remediar las distorsiones ópticas producidos por lentes gran angular, o incluso las que se originan por el tráfico de la luz por dos medios de transmisión con características distintas.

El conjunto de estas tres posibilita la transformación espacial y geométrica de una imagen. Por ejemplo, las transformaciones afines permiten redimensionar una imagen (fig. 3.13), las proyectivas permiten realizar cambios en la perspectiva aparente del plano de la imagen (fig. 3.14), y las transformaciones no-lineales permiten distorsionarla con la aplicación de funciones arbitrarias (fig. 3.15).

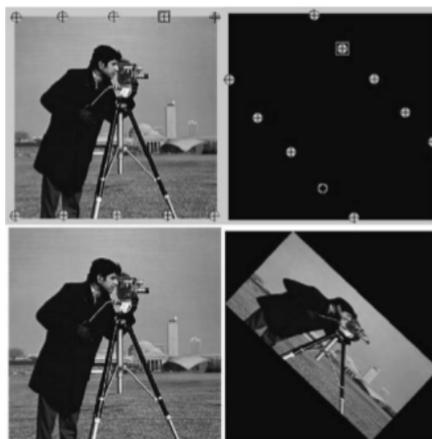


Figura 3.13: Aplicación de una transformación afín.

La aplicación de una transformación afín definida por una matriz de 10 puntos.



Figura 3.14: Aplicación de una transformación proyectiva.

Aplicando una transformación proyectiva se puede simular distintas perspectivas del plano-imagen.

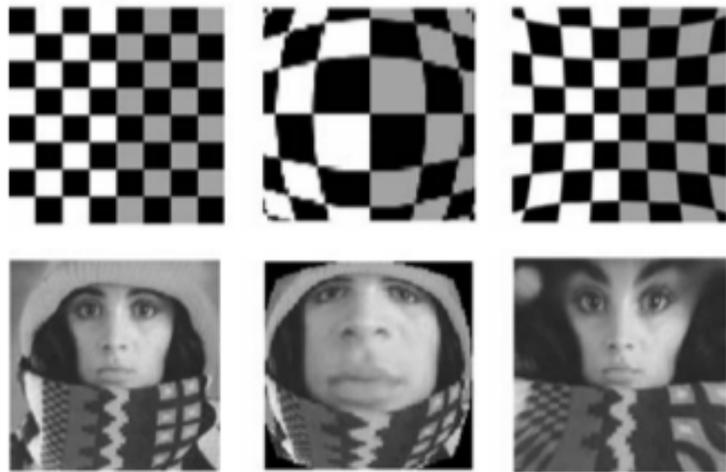


Figura 3.15: Aplicación de una transformación no-lineal.

Con las operaciones no-lineales se pueden conseguir efectos de distorsión como los producidos por lentes gran-angular.

3.4. Operaciones Morfológicas

Las operaciones morfológicas son técnicas de procesado no-lineales que tratan sobre aspectos concernientes a la forma o a la morfología de ciertas características contenidas en una imagen. Estas operaciones son generalmente usadas para eliminar ruido o imperfecciones producidas en alguna etapa previa del procesamiento, tales como aquellos introducidos como consecuencia de una operación de segmentación, o en la aplicación de filtros en el dominio espacial [20]. Las operaciones morfológicas constan de dos elementos básicos, el elemento estructurante, y la operación de ajuste e intersección:

3.4.1. Elemento estructurante

Las técnicas morfológicas por lo general afectan a una imagen mediante un elemento estructurante. El elemento estructurante puede definirse como una matriz $M_{n,m}$ de píxeles $P_{i \leq n, j \leq m} \in \{0, 1\}$, es decir, una matriz cuyos elementos pueden ser 0 o 1. Por ejemplo:

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad (3.12)$$

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (3.13)$$

Un elemento estructurante posee un origen que puede coincidir con la ubicación de alguno de sus componentes, no necesariamente debe estar ubicado dentro del área que delimita al elemento estructurante. La forma como el elemento estructurante es aplicado es similar a cómo se calcula la respuesta de los filtros digitales de convolución, pero con algunas pequeñas diferencias. El proceso mediante el cual se computa la respuesta es llamado ajuste e intersección.

3.4.2. Ajuste e intersección

El origen del elemento estructurante es desplazado a través del dominio de la imagen, y cada uno de sus píxeles dentro del área que lo delimita es asociado con los píxeles de la imagen que caen bajo su área. Se dice que el elemento estructurante **ajusta** si bajo cada uno de sus elementos con valor 1, los píxeles de la imagen poseen el mismo valor, y se dice que **intersecta** si al menos uno de los píxeles del elemento estructurante tiene un valor positivo que corresponde con el píxel asociado en la imagen [20].

En las figuras 3.16 y 3.17 se puede observar el efecto de procesar distintos elementos estructurantes con las operaciones de ajuste e intersección. Por ejemplo, si computamos la **intersección** del elemento estructurante (3.12) sobre una imagen, el resultado es conocido como dilatación, y si computamos el **ajuste** del elemento estructurante definido en (3.13) sobre una imagen el resultado es conocido como erosión.

La implementación más simple de las operaciones morfológicas asume que la imagen y la operación son binarias, sin embargo, es posible extender

las operaciones morfológicas a imágenes a escala de grises o a color si, por ejemplo, se permite probar el píxel por un valor positivo cualquiera, en vez de estrictamente igual a 1, tal como se puede ver en la figura 3.18.

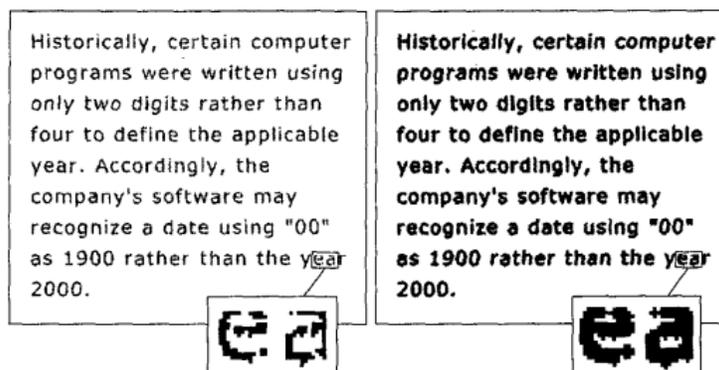


Figura 3.16: Aplicación de una operación de dilatación.

Resultado de una aplicación de dilatación sobre una imagen con ruido.



Figura 3.17: Aplicación de una operación de erosión.

Resultado de la aplicación de erosión sobre una imagen con ruido.



Figura 3.18: Operaciones morfológicas sobre imágenes a escala de grises.

Operaciones morfológicas sobre imágenes a escala de grises [27]. De izquierda a derecha tenemos la imagen original, la aplicación de un filtro de dilatación y la aplicación de un filtro de erosión.

Capítulo 4

Aprendizaje Automático

En este capítulo se define el aprendizaje automático (sección 4.1), algunas técnicas pertenecientes su agrupación supervisada (sección 4.2), y otras pertenecientes a las no-supervisadas (sección 4.3).

4.1. Aprendizaje Automático

El aprendizaje automático es en esencia una forma de estadística aplicada con un énfasis en el uso de computadoras para aproximar con instrumentos probabilísticos, funciones complejas, y con un menor énfasis determinar los intervalos de confianza alrededor de estas funciones [23].

Podemos extender la definición anterior, dentro del contexto del aprendizaje automático, asumiendo que un programa de computador se considera que aprende de una experiencia E con respecto a una tarea T y una métrica de rendimiento P , si su rendimiento en la tarea T , medido por P , mejora con la experiencia E , cualesquiera que sea la experiencia E , la tarea T y la métrica P [35] [23].

De forma más concisa, puede definirse como una serie de técnicas que pueden detectar patrones de forma automática, y posibilitar el uso de dichos patrones para predecir el futuro o realizar otro tipo de toma de decisiones bajo un contexto de incertidumbre [36]. En función de la forma que asume la experiencia E , podemos determinar 3 tipos fundamentales de aprendizaje automático [23]:

1. Aprendizaje supervisado

2. Aprendizaje sin supervisión
3. Aprendizaje por refuerzo

Es importante señalar que en este caso nos referiremos sólo los dos primeros porque el tercero no implica una mayor relevancia a los fines de este trabajo.

4.2. Aprendizaje Supervisado

El aprendizaje supervisado comprende las técnicas de aprendizaje automático en las cuales se pretende modelar una función f que lleve entradas x a su correspondiente salida y , utilizando en el proceso pares $D = (x, y)$ de entrada-salida (la experiencia E) que contienen tanto la entrada x como su valor de salida o etiqueta y [11] [36].

Cada entrada de entrenamiento x es un vector d -dimensional de números que pueden representar infinidad de variables asociadas con algún objeto, no importa la complejidad de su estructura. Estas variables son llamadas características, atributos o covariables. La estructura de la salida o etiqueta y en principio no tiene límite, pero por lo general las técnicas de aprendizaje supervisado se categorizan en función de la naturaleza de la variable.

Cuando y forma parte de un grupo finito, $y_i \in \{1, \dots, k\}$, el problema de aprendizaje se entiende como un problema de clasificación, es decir, este tipo de tareas involucra predecir a cual categoría y_k pertenece una entrada x . Cuando y es una variable escalar con un dominio real, se considera como un problema de regresión, que involucra predecir un valor numérico para una entrada arbitraria [36] [23].

4.2.1. Regresión

Uno de los modelos más usados para la regresión se conoce como regresión lineal. Lineal porque asume que la respuesta es una función lineal de las entradas:

$$y(x) = w^T x + \epsilon = \sum_{j=1}^D w_j x_j + \epsilon \quad (4.1)$$

donde $w^T x$ representa el producto escalar entre el vector x y el vector de pesos de regresión w , siendo ϵ el error residual entre la predicción del modelo y el valor verdadero asociado al vector x [36].

Es posible representar relaciones no-lineales usando combinaciones lineales de otras funciones base, por ejemplo, polinomios, exponenciales o sigmoideas. Incluso funciones bases más sofisticadas (como las producidas por la transformada de Fourier o la transformación de la Ondícula) son válidas para obtener una función de regresión que es no-lineal con respecto al vector x , aunque aún la relación entre estas bases aún sea lineal [11].

Existen también modelos regresivos que hacen uso de funciones lineales a trozos y splines (funciones polinómicas definidas a trozos, con condiciones que aseguren un encaje suave entre los pares contiguos). Los modelos dentro de este tipo de regresión son llamados modelos semi-paramétricos [11].

4.2.2. Máquinas de vectores de soporte

Las máquinas de vectores de soporte pertenecen a una clase de algoritmos que pueden ser usados para la clasificación, regresión, estimación de densidad, detección de novedades, etc. En el caso más simple de clasificación binaria, las máquinas de vectores de soporte encuentran el plano que separa las dos clases entre los datos con el margen más amplio posible (fig 4.1). Esto conlleva a una buena generalización a la hora de clasificar datos no vistos anteriormente, y también permite la aplicación de métodos de optimización especializados que permiten a este método aprender de grandes cantidades de datos [11].

4.2.3. Redes neuronales artificiales

Las redes neuronales artificiales son un modelo computacional inspirado en funcionamiento del sistema nervioso procesando la información usando una perspectiva conexionista [11]. Consisten en un conjunto interconectado de nodos análogos a su contraparte biológica, cuya capacidad de cómputo reside en la relación de la fuerza de las conexiones íter-nodo, o peso de conexión, obtenido a través de un proceso de adaptación al conjunto de observaciones de entrenamiento, aprendiendo a “computar” a través de su experiencia [24].

En su configuración más simple, la señal producida por una entrada correspondiente a una observación, es propagada entre niveles de nodos interconectados de forma similar a las neuronas biológicas. La sinapsis, o la fuerza

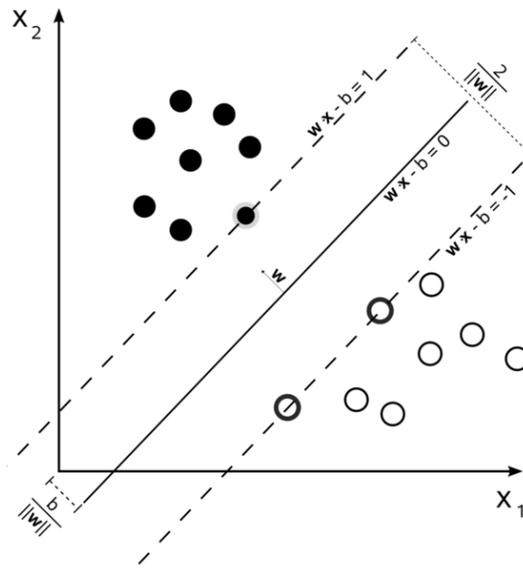


Figura 4.1: Vector soporte.

de conexión íter-nodo, le dan un valor a las partes individuales de la señal, estos pesos pueden ser positivos o negativos, en el caso de las neuronas biológicas, estimulantes o inhibitorios. Las dendritas, que en el caso de las neuronas artificiales son todas las conexiones que inciden en un nodo dado reúnen la información de los niveles anteriores.

En el soma o el nodo en sí la información es acumulada, en el caso artificial, con una función predefinida. En ambos casos, tan pronto como la señal acumulada excede un cierto valor (valor límite), el núcleo de la célula activa un pulso eléctrico que es propagado a las neuronas conectadas en el nivel subsiguiente. En las redes neuronales artificiales, el proceso de acumulación y propagación es realizado con una función de activación, que puede o no ser similar a su contraparte biológica [32]. El ajuste de los pesos entre nodos es realizado a través de un algoritmo basado en el descenso gradiente llamado retro-propagación (back-propagation).

La arquitectura más básica de redes neuronales artificiales es conocida como perceptrón. El perceptrón contiene sólo dos capas de neuronas, que corresponden a una capa de entrada y una de salida. La cantidad de nodos en la capa de entrada es igual a la dimensión del vector de entrada [2]. Existen muchas formas posibles de acomodar los elementos básicos de una red neuronal para realizar distintas tareas.

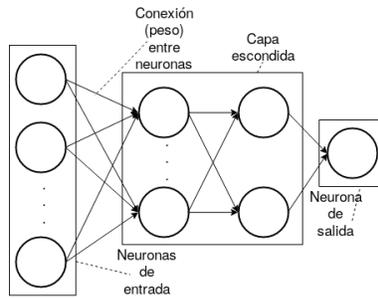


Figura 4.2: Perceptrón multicapa.

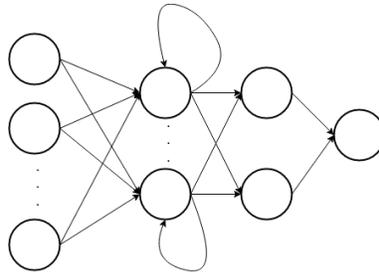


Figura 4.3: Red neuronal recurrente.

Perceptrones multicapa y redes neuronales profundas: Estos modelos (fig. 4.2) también son llamados feedforward ya que el flujo de información fluye desde la entrada, pasando por las operaciones de las capas intermedias, hasta llegar finalmente a la unidad de salida. Además de las capas de salida y entrada, pueden tener niveles intermedios de neuronas artificiales, que reciben por nombre capas escondidas [23].

Redes recurrentes: Las RNA recurrentes (fig. 4.3) son capaces de retroalimentarse por medio de recurrencias, por ejemplo, incluyendo la salida de la red en un tiempo t para la próxima computación en un tiempo $t + 1$. Existen muchas redes recurrentes de formas casi arbitrarias, pero por lo general son usadas para tareas donde se necesite modelar una secuencia, por ejemplo, en el reconocimiento de texto y voz [23].

Redes Convolucionales: Esta arquitectura (fig. 4.4) ha tenido un inmenso éxito en el procesamiento de datos con una topología tipo malla, como las imágenes o las series de tiempo. Por lo general tienen una baja conectividad entre niveles, y las unidades de entrada son activas

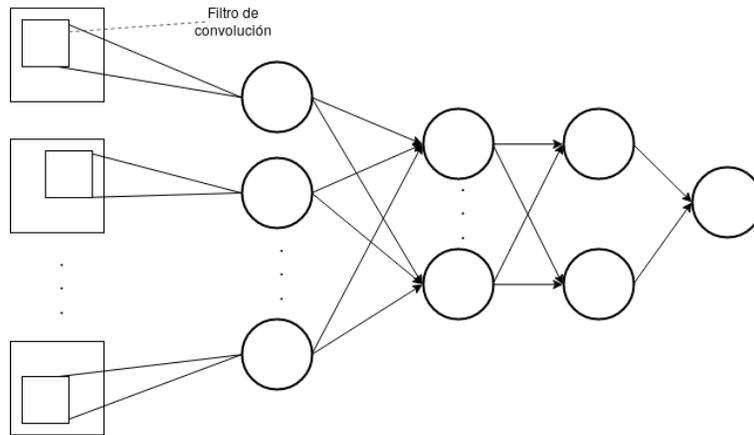


Figura 4.4: Red neuronal de convolución.

por sí mismas, ya que aplican filtros de convolución (ya anteriormente expuestos) a sus entradas [23].

Las RNA existen en concepto desde hace más de 40 años, y el interés que suscitan ha encontrado nuevamente un auge gracias a los avances en el proceso mediante el cual son entrenadas. Este proceso, llamado retropropagación, muy costoso en términos computacionales, se ha vuelto ejecutable gracias al surgimiento de procesadores masivamente paralelos de propósito general. También la creciente cantidad de datos correctamente catalogados y etiquetados gracias a la revolución del internet posibilitó la recopilación de las cantidades de datos necesarios para un entrenamiento efectivo de las RNA.

4.3. Aprendizaje no supervisado

El objetivo de las técnicas de aprendizaje no supervisado es encontrar en los datos patrones o características latentes de interés o utilidad de forma automática. Puede entenderse como el proceso de extracción de información de una distribución de datos sin que se requiera supervisión en forma de etiquetas o valores de salida esperados, ya que el grupo $D = \{x\}$ de entradas son proporcionadas sin estos.

La reducción de ruido en una distribución de datos; la agrupación de datos o el encontrar una representación comprimida de los mismos, son parte de las

tareas que usualmente son conseguidas mediante la aplicación de los algoritmos de aprendizaje no-supervisado, por ejemplo, el análisis de componentes principales y k -medias.

4.3.1. Análisis de componentes principales

El análisis de componentes principales permite reducir la dimensionalidad de los datos. El principio básico detrás de esta técnica se basa en encontrar un grupo de vectores lineales base, ortogonales entre sí, de forma que, al proyectar los puntos sobre uno de estos vectores base, el error cuadrático medio sea mínimo [36].

Estos vectores representan la dirección de máxima información, es decir, que entre pares de componentes (variables en los datos), un primer vector base es definido por la dirección donde se aprecie la máxima varianza entre los datos, siendo el segundo vector base la segunda dirección con mayor información y así sucesivamente [11].

Con el análisis de componentes principales, gracias a la definición de los vectores como ortogonales, es posible aprender una representación de los datos que no posean correlaciones lineales entre ellos, pues al encontrar una matriz de vectores ortogonales $z = W^T x$, nos aseguramos que la matriz z sea diagonal, por lo tanto $var(z) = 0$ [23].

4.3.2. k -Medias

k -medias es un método no-determinístico de agrupación, ampliamente usado en muchas aplicaciones. La idea básica de la agrupación a través de k -medias es que, por cada iteración del algoritmo, dada una agrupación inicial no-óptima, cada observación sea asignada al grupo con el cual la distancia a su centro sea mínima, se calcule nuevamente el punto medio de cada grupo computando la media de las observaciones contenidas en él, y se repita este proceso hasta que sea alcanzado algún criterio de convergencia. [11]

Una de las ventajas de k -medias, es la eficiencia en la codificación de la representación aprendida. Como una observación puede pertenecer a sólo un grupo, el tamaño de su representación aprendida es de tamaño k . Esto confiere ciertas ventajas estadísticas, pues homogeiniza la representación aprendida de todas las representaciones que son asignadas a un mismo grupo, y también posibilita que la representación pueda ser codificada en un solo entero [23].

En cuanto a sus desventajas, tenemos que no existe un único criterio que mida que tan bien la agrupación de los datos se corresponda con características en el mundo real. La naturaleza no determinística puede arrojar agrupaciones distintas entre distintas aplicaciones del mismo algoritmo, arrojando en situaciones representaciones útiles, que estén correlacionadas con una característica concreta, y en otras, produciendo agrupaciones de poca correspondencia con el mundo real [23].

Capítulo 5

Antecedentes

En este capítulo se desarrolla una breve reseña de los antecedentes relacionados a la elaboración de sistemas de verificación off-line para firmas manuscritas. En la sección 5.1 se describen algunos de los pasos y conceptos propios del proceso, y su tratamiento en los trabajos previos son expuestos con mayor detalle en las secciones subsecuentes. En la sección 5.2 se habla de las métricas usadas para la evaluación de los sistemas, en la sección 5.3 se describe el proceso de recolección de datos, luego, en la sección 5.4 se ahonda en el proceso de preprocesamiento de las imágenes, seguido por la sección 5.5, que describe el proceso de extracción de características, y por último, en la sección 5.6 se habla sobre el proceso de verificación.

5.1. Sobre los sistemas de verificación

Según Hafemann [26], los distintos sistemas off-line desarrollados para la verificación de firmas pueden distinguirse en dos grupos :

- Si un único modelo es usado para clasificar las imágenes de las firmas de cualquier usuario, el sistema se denomina independiente del usuario (writer-independent).
- En el caso donde se asigna un modelo de clasificación para cada usuario, se dice que el sistema es dependiente del usuario (writer-dependent).

De acuerdo con Hafeman [26], y Bhumika [1], el proceso para la verificación de firmas puede dividirse en 4 subprocesos:

- Adquisición de los datos: El dato (la firma) es adquirido, es decir, está disponible su representación digital [43]. Esto puede lograrse a través de varios métodos, por ejemplo, con el uso de una cámara, un teléfono móvil, o un escáner [1].
- Preprocesamiento: Es un paso necesario para mejorar la eficacia de la clasificación, y reducir el poder de cómputo necesario tanto para la fase de extracción de características como de la fase de clasificación [1]. Estandarizar la imagen de una firma mediante este subproceso es importante, pues la imagen producto del subproceso de adquisición puede presentar variaciones en cuanto al ancho del lápiz, el tamaño de la firma, su rotación, etc., incluso entre firmas auténticas de una misma persona [26]. Existen multitud de métodos (capítulo 3) , usando los elementos del procesamiento digital de imágenes, disponibles para esta tarea.
- Extracción de características: Se extraen atributos intrínsecos de las firmas que otorgan detalles en la observación de los datos. Cualquier característica puede ser cuantificada [43]. La eficacia de la clasificación depende de la característica, así que su selección es parte fundamental del proceso de verificación [44].

Hafemann [26] otorga dos categorizaciones posibles para las técnicas de extracción. En primer lugar, podrían ser distinguidas entre estáticas y pseudo-dinámicas. Las pseudo-dinámicas buscan recobrar información dinámica del proceso de firmado (la velocidad, presión, etc.), mientras que las estáticas buscan características contenidas en la imagen final.

La segunda categorización diferencia entre propiedades locales y propiedades globales. Las propiedades globales incluyen características tales como la altura, la anchura o las dadas por extractores de características que son aplicados a una imagen entera. Las propiedades locales describen partes de la imagen mediante la segmentación (de acuerdo a componentes conectados) o más comúnmente dividiendo la imagen con un mallado (en coordenadas cartesianas o polares [21]) para luego aplicar extractores de características a cada parte del mallado. Yadav [44] añade a esta última categoría las características de transición (de la variación entre píxeles negros y blancos en una imagen binarizada), mientras que Bhumika [1] añade las características geométricas, que

preservan la información topológica y geométrica así como también sus propiedades globales y locales.

- **Verificación:** En concordancia con lo descrito por Hafemann [26], el proceso de entrenamiento depende del esquema utilizado. Para sistemas dependientes del usuario (writer-dependent) un modelo es entrenado para cada identidad, usando firmas genuinas y falsificaciones aleatorias. Durante la fase operativa, el modelo entrenado para cada identidad es usado para clasificar nuevos ejemplos de firmas como genuinas o falsificaciones. En el esquema independiente del usuario (writer-independent) existe un único modelo para todas las identidades. Durante la fase de pruebas, el modelo es usado para comparar muestras de prueba contra referencias de una firma genuina para realizar una decisión.

5.2. Métricas de evaluación

Para evaluar el rendimiento y la exactitud de la verificación, se utilizan cuatro métricas principales arrojadas por los sondeos de Hafemann [26], Bhumika [1] y Shah [42] :

Proporción de falsa aceptación: Es equivalente al error tipo 2. La proporción de falsa aceptación, o FAR por sus siglas en inglés, es la medida de la probabilidad de que un sistema biométrico acepte de forma errónea un intento de acceso por parte de un usuario no autorizado. Para los propósitos del presente trabajo, esto significa la probabilidad de que una firma falsa sea catalogada como genuina [5]. Viene dada por $FAR = \frac{FP}{N+P}$, con FP como falsos positivos, N como el total de los negativos verdaderos y P como el total de los verdaderos positivos [5].

Proporción de falso rechazo: Equivalente al error tipo 1. Denotado por FRR por sus siglas en inglés, es el complemento del FAR, es decir, representa la probabilidad de que una firma verdadera sea catalogada como falsa. Se define como $FRR = \frac{FN}{N+P}$, con FN como falsos positivos, N como el total de los negativos verdaderos y P como el total de los verdaderos positivos [6].

Proporción de error idéntico: Denotado por EER (Equal Error Rate), se comprende como el punto donde las dos anteriores son iguales, dado el

umbral de discriminación k [45]. Es decir, si el modelo tiene un umbral k mediante el cual se decide la pertenencia de una observación a una clase, el EER se reporta como el umbral k , y la proporción de falsa aceptación o falso rechazo indistintamente, ya que son iguales.

Exactitud: Este número representa el total de casos que fueron catalogados de forma correcta. Se calcula como $ACC = \frac{VP+VN}{P+N}$, con VP como el número de verdaderos positivos, VN el de verdaderos negativos, P como el total de positivos y N como el total de negativos [12].

5.3. Adquisición de datos

Existen varios repositorios centralizados de imágenes de firmas ya previamente extraídas. Varios de los estudios mencionados usaron datos provistos por estos repositorios. Los repositorios son los siguientes:

GPDS: En su versión actual contiene data de 960 individuos, 24 firmas genuinas además de 30 falsificaciones. Los 24 especímenes genuinos de cada signatario fueron recolectados en un único día de escritura. Las falsificaciones fueron producidas a partir de las imágenes estáticas de las firmas genuinas. Se les permitió a los falsificadores practicar tanto tiempo como requirieran. Cada falsificador generó 3 firmas de 5 distintas personas en una sesión única. Las firmas genuinas dadas como referencias a cada falsificador fueron elegidas de manera aleatoria entre las 24 firmas disponibles. Por lo tanto, de cada firma se pueden encontrar 30 falsificaciones expertas realizadas por 10 falsificadores a partir de 10 especímenes genuinos [34].

MCYT: La información de las firmas fue adquirida usando una pluma fuente y papel sobre una superficie plana. Fueron escogidos 75 individuos para digitalizar sus firmas mediante un escaneo a 600 dpi. El cuerpo resultante está compuesto de 2.250 firmas, con 15 genuinas y 15 falsificaciones expertas para cada uno de los 75 individuos. Para las falsificaciones expertas, se contó con 3 falsificadores por usuario [30].

CEDAR: Contiene firmas de 55 individuos que pertenecen a diversos trasfondos culturales y profesionales. Cada uno de estos individuos realizaron un total de 24 firmas con 20 minutos de diferencia entre sí. Cada

uno de los falsificadores intentaron emular las firmas de 3 personas en 8 oportunidades distintas, para producir un total de 24 falsificaciones expertas por cada genuina. Por lo tanto, la base de datos cuenta con un total de 1.320 firmas genuinas y 1.320 falsificaciones expertas. [19]

BHSig260: Contienen firmas de 260 personas, entre ellas 100 fueron producidas por en el lenguaje Bengalí y 160 en Indio. Los creadores usaron el mismo protocolo que el de GPDS. Por lo tanto, consiste de 2.400 firmas genuinas con 3.000 falsificaciones expertas en Bengalí, y 3.840 firmas genuinas con 4.800 falsificacioens expertas en Indio. [19]

PUC-BR: El repositorio consiste en la digitalización de las firmas asociadas con cheques bancarios. Totaliza 168 firmas genuinas, pero sólo 60 usuarios cuentan falsificaciones, siendo 10 falsificaciones aleatorias y 10 falsificaciones expertas para cada uno [26].

De los estudios citados en este trabajo, sólo Hatkar [29] no especifica cómo se desarrolló la adquisición de los datos.

5.4. Pre-procesamiento

Durante esta fase, Hatkar et al. [29] binarizaron la imagen de la firma para simplificar el proceso de extracción de características. Como el tamaño de las imágenes diferían entre sí, aplicaron transformaciones geométricas para estandarizar el tamaño de su base de conocimiento a 256x256 píxeles. Luego, para obtener una representación cuyas características fuesen invariantes a las propiedades del proceso de firmado (la calidad del papel, el utensilio de escritura, etc.) se le aplicó la operación morfológica de erosión para lograr el adelgazamiento de las líneas. Posteriormente, se construyó un rectángulo sobre la imagen en aras de reducir el área a ser usada para la extracción de características.

En el desarrollo de Ferrer et al. [21] se buscaba obtener el contorno de la firma, además del proceso de estandarizado y adelgazamiento llevado a cabo por Hatkar et al. [29], se le aplicó a la imagen una operación de dilatación, y posteriormente una de llenado donde el espacio entre líneas se le asigna la misma intensidad que el de las mismas, para simplificar la extracción del contorno. Si varios objetos fuesen detectados en la misma imagen después

del proceso anterior, se aplicaba una operación de dilatación horizontal para lograr la conectividad.

Bhunia et al [8] sólo aplicaron el proceso de binarización y reducción de ruido mediante un filtro Gaussiano.

Hafemann et al [25] centraron la imagen en un lienzo usando el centro de masa de la firma (el punto promedio de todos los píxeles que la conforman). Se procedió a remover el fondo del lienzo con el algoritmo de OTSU, asignando el color blanco para el fondo y dejando la firma en escala de grises. Entonces, se invirtieron las intensidades de los píxeles restando el valor de cada píxel a la intensidad máxima (255) de forma que el fondo tuviese valor 0. Posteriormente la imagen fue re-dimensionada (150x220 píxeles).

Dey et al. [19] re-dimensionaron las imágenes a un tamaño fijo de 155x220 usando interpolación bi-lineal, y al igual que Hafemann et al. [25] invirtieron las intensidades de la imagen. Finalmente, los píxeles de cada imagen fueron divididos entre la desviación estándar de las intensidades de los píxeles de cada imagen, proceso el cual el autor hace referencia como “normalización”.

Yilmaz et al. [45] puntualizaron que aunque “la verificación off-line de firmas puede beneficiarse de los pasos de normalización para obtener invarianza de rotación, escala y traslación” ningún procesamiento previo sería aplicado, ya que las características a extraer eran inherentemente invariantes a la traslación y escala, mientras que normalizar en cuanto a la rotación es una transformación complicada que ralentiza el proceso de verificación y entrenamiento.

Kekre et al. [31] aplicaron operaciones de remoción de ruido, de escalamiento, suavizado, normalización de intensidad y de erosión para conseguir los trazos de la firma. Posteriormente, aplicaron una sucesión de operaciones de dilatación para producir el contorno. Este paso se realizó un total de 4 veces rellenando el contorno generado con un color distinto a cada nivel (negro, rojo, verde y azul respectivamente), con un elemento estructurante distinto por cada uno. Cada elemento estructurante posee un radio que cumple con que $r1 < r2 < r3 < r4$. Como resultado, se tiene una estructura con 4 bandas de colores, donde cada banda representa la extensión de la variación de cada píxel y por lo tanto los segmentos de la firma.

5.5. Extracción de características

Hatkar et al. [29] extrajeron las siguientes características:

Histograma horizontal y vertical máximo: Se obtienen calculando la cantidad de píxeles negros en cada fila y columna de la imagen. Aquellas con mayor número son usadas como características.

Centro de masa: Se divide la imagen en dos partes iguales y se calcula el punto promedio de los píxeles negros.

Área normalizada de la firma: Es la proporción entre el área de la imagen sobre el área total de los píxeles que conforman la firma

Proporción de aspecto: Es la razón entre la altura y la anchura de la imagen. Este coeficiente es establecido ya que estas dimensiones por separado pueden variar entre distintas firmas de una misma persona, pero la proporción se mantiene medianamente constante.

Característica a seis pliegues: Se divide la firma en tres partes iguales y se encuentra el cuadro delimitante para cada parte. Entonces se calcula el centro de masa de cada una. Luego se traza una línea horizontal pasando a través de cada uno de los centros de masa anteriores y se calcula el área de la parte inferior y superior generada dentro de cada cuadro delimitante, obteniendo un total de 6 áreas.

Característica de transición: Se recorre la imagen de izquierda a derecha y cada vez que hay una transición de blanco a negro o viceversa, se calcula la razón entre la posición de la transición y el ancho de la imagen recorrida hasta ese punto, y se almacena como una característica. Se repite el proceso para todas las direcciones posibles. También se calcula el número total de transiciones de 0 a 1 y de 1 a 0.

Ferrer et al. [21] tomaron como característica un vector que incluye el número de píxeles negros bajo el radio, la derivada de su longitud y el ángulo de los puntos del contorno de la firma sobre T intervalos discretos. Para esto se define como origen el centro de masa del contorno y en coordenadas polares se obtiene una muestra del mismo con distancias expresadas en ángulos iguales a T . El radio se entiende como la distancia entre el origen y el contorno de la firma, la derivada de la longitud del radio sería la tasa de crecimiento de su longitud entre distintas muestras consecutivas. Una segunda característica viene dada dividiendo la imagen de forma horizontal, partiendo del centro geométrico. Se calcula la distancia de la proyección de

los puntos del contorno sobre el plano. Los puntos se toman en intervalos regulares. Se repite la operación dividiendo de forma vertical.

Bhunia et al [8] utilizaron dos métodos para extraer características en el dominio espectral. El primero es denominado discretización local de la fase, donde se transforma la imagen mediante la transformada discreta de Fourier y se obtiene el valor de la fase para cada píxel de cuatro frecuencias base. Como segunda característica, se aplica una transformación de la ondícula, empleando un filtro de paso bajo y otro de paso alto simultáneamente a la señal de entrada.

Tanto Dey et al. [19] como Hafemann et al [25] utilizaron redes neuronales profundas en la extracción automática de características. Sin embargo, Dey et al. [19] muestran la respuesta de cada nivel de la red neuronal usada.

En Yilmaz et al. [45] dos características son extraídas. La primera, llamada histograma de gradientes orientados, computa el histograma de las orientaciones (entre 0 y 180 grados) del gradiente de una imagen. Para ser calculado, la misma se divide en un mallado regular, y para cada sección se determina el gradiente de cada píxel usando alguno de los filtros de detección de bordes anteriormente expuestos (sección 3.2). La segunda característica recibe por nombre LBP (Local Binary Patterns), que para cada punto en la imagen calcula un número binario concatenando (mediante un patrón predefinido) el valor de sus vecinos (0 o 1), y por último, crea un histograma con el valor en base decimal de cada uno de los valores binarios correspondientes a cada punto.

Kekre et al. [31] utilizó como vector de características el número de ocurrencias de ciertos colores al aplicar una operación X-Or entre el contorno producido en la fase anterior y la firma a verificar. El vector resultante representa la cantidad de píxeles de la firma a verificar que caen dentro o fuera del contorno. El contorno es entendido como la variación permitida entre firmas de una misma persona (la variación intra-clase), así que el número de píxeles de color negro, azul, verde y rojo (el producto de la operación XOR entre imágenes) determina que tan bien se ajusta la firma al patrón generado por las operaciones de dilatación sobre una firma de referencia. La cantidad de píxeles por fuera de los colores anteriormente mencionados se entienden como variaciones por fuera de los parámetros permitidos.

5.6. Verificación

Hatkar et al. [29] usó un perceptrón multicapa entrenado con las características extraídas de 5 firmas genuinas contra sus 5 contrapartes falsas. Posteriormente se midió la generalidad del modelo probando con 995 pares de firmas genuinas y falsas.

Ferrer et al. [21] usan tres modelos para la clasificación. HMM (Hidden Markov Model), SVM (Support Vector Machine) y la distancia euclidiana son las técnicas usadas para la predicción. El primer modelo, basado en HMM, fue entrenado con 4, 8 y 12 pares de firmas genuinas. En la etapa de verificación, para las falsificaciones aleatorias con 4, 8, y 12 muestras los resultados reportados fueron los siguientes: FRR de 4.3%, 2.5% y 2.3%, FAR de 3.8%, 2.4% y 3.3%.

Para las falsificaciones expertas los resultados reportados fueron los siguientes: FRR de 17.3%, 13.4% y 14.1%, FAR de 14.9%, 14.9% y 12.6% para 4, 8 y 12 muestras respectivamente.

El modelo SVM fue entrenado con 12 muestras, con 3 kernels distintos. Los resultados para cada uno son los siguientes: Para el entrenamiento con falsificaciones aleatorias, la tasa de FRR fue de 4.27%, 3.65% y 3.23%, la FAR fue de 3.71%, 3.15% y 2.65% (kernel lineal, polinomial y RBF respectivamente). Con falsificaciones expertas la tasa de FRR fue de 21.06%, 15.41% y 15.41%, mientras que la FAR fue de 18.54%, 15.64% y 13.12% para los respectivos kernels.

Para la distancia euclidiana usaron 8, 12 y 16 muestras en la etapa de entrenamiento. En la verificación del modelo, tenemos los siguientes resultados para los modelos entrenados con 8, 12 y 16 muestras respectivamente: Con falsificaciones aleatorias el FAR se ubica en 6.16%, 5.56% y 5.61%, FRR es de 5.92%, 5.13% y 4.96% respectivamente. Con falsificaciones expertas el FRR resulta ser de 17.29%, 16.21% y 16.39%; el FAR es igual a 18.25%, 15.66% y 15.50% respectivamente.

En el desarrollo de Bhunia et al. [8] el modelo para la verificación constó de dos SVM de una sola clase para cada individuo, una por cada característica. La clasificación de cada firma (genuina o falsificación) estaría dada por el promedio de los valores arrojados por ambos SVM (cuyo valor oscilaría entre -1 y 1). El entrenamiento del modelo se llevó a cabo con cuatro de los repositorios previamente mencionados, MCYT, GPDS-300, BHsig-260 y CEDAR. Para evaluar el modelo se realizaron 5 experimentos por cada repositorio, usando 4, 6, 8, 10 y 12 firmas genuinas y ninguna firma falsa por

cada usuario en el entrenamiento. En la etapa de verificación, para medir la exactitud sobre los repositorios DGPS160 y BHsig-260 se usaron 20, 18, 16, 14 y 12 firmas genuinas y 30 firmas falsas para los modelos entrenados con 4, 6, 8, 10 y 12 muestras respectivamente. Para CEDAR se repitió el mismo procedimiento, sólo que con 24 firmas falsas en lugar de 30, y para MCYT se usaron 15 firmas falsas y 11, 9, 7, 5 y 3 firmas genuinas correspondientes de forma respectiva a los modelos entrenados con 4, 6, 8, 10 y 12 muestras genuinas. Los resultados de estos experimentos pueden apreciarse en la figura 5.1.

Dey et al. [19] usaron una misma red neuronal convolucional para cada observación. La salida de la red neuronal es un vector de 1.024 elementos correspondientes a las características aprendidas de forma automática para cada una de las imágenes en el par de firmas genuina/falsificación. Posteriormente, se estableció un umbral de discriminación para clasificar los dos vectores de características por medio de la distancia euclidiana. Los resultados pueden ser apreciados en la figura 5.2.

El sistema implementado por Hafemann et al. [25] consiste en un ensemble serial de dos redes neuronales distintas. La primera es entrenada para maximizar su poder discriminante entre firmas genuinas y falsificaciones aleatorias. Hafemann nota que el entrenamiento con falsificaciones aleatorias es equivalente a entrenar el modelo con el objetivo de maximizar la distancia computada entre distintos usuarios, ya que la comparación de una firma aleatoria contra una firma genuina es equivalente a realizar la comparación entre firmas genuinas de distintos usuarios. Una vez que se alcanza un margen de error estable, el vector de características arrojado por penúltimo nivel de la red neuronal es extraído. Este vector es entonces usado como entrada de una red neuronal que forma parte de un sistema dependiente del usuario. Para cada usuario se entrena un clasificador basado en SVM que toma como entrada la salida de la segunda red neuronal. Los resultados obtenidos por [25] pueden verse en la figura 5.3. El proceso se llevó a cabo 2 veces, en la primera se usaron únicamente firmas genuinas para entrenar el modelo para cada usuario (en la tabla aparece bajo el nombre *SigNet*), y en la segunda se incluyeron de forma aleatoria firmas falsas (bajo el nombre de *SigNet - F*).

Yilmaz et al. utilizan seis modelos para realizar la verificación. Se dividen entre los dependientes e independientes del usuario. Para cada grupo, se entrenaron 3 máquinas de vectores de soporte, una para cada característica distinta a ser usada para la discriminación. Dos de los modelos dentro de cada grupo utilizan el histograma de gradientes orientados, difiriendo entre

sí por el espacio (uno en coordenadas cartesianas y el otro en coordenadas polares) en el cual fueron computados. El último modelo para ambos grupos fue entrenado con los patrones locales binarios en coordenadas cartesianas. Resultados en la tabla 5.4. En dos experimentos, los modelos se entrenaron con 12 o 5 muestras por usuario, tanto para los dependientes del usuario como para los independientes de usuario.

En Kekre et al. [31] utilizan un clasificador difuso, con clases que representan que tan bien se ajusta una firma al patrón prototipo de la firma. Entre las clases posibles se encuentran Perfect, Acceptable, Okay y Reject. Las muestras dentro de las clases Perfect, Acceptable y Okay son consideradas como válidas, mientras que aquellas dentro de Reject son consideradas inválidas. Un total de 257 pruebas fueron realizadas, usando 3 firmas genuinas de 100 individuos para el entrenamiento. 350 falsificaciones expertas fueron usadas para la verificación, y un total de 257 pruebas fueron realizadas. El sistema arrojó un 100 % de precisión a la hora de rechazar falsificaciones aleatorias, mientras que para las falsificaciones expertas, se obtuvo un FAR de 5.79 %. Para firmas genuinas, el FRR se ubicó en 7.23 %.

Dataset	Entrenamiento		Evaluación		FAR (%)	FRR (%)	AER(%)
	Ng	Nr	Ng	Nr			
MYCT	4	0	11	0	18.12	20.11	19.12
	6		9		13.55	16	14.78
	8		7		11.77	12.1	11.94
	10		5		8.78	10.23	9.5
	12		3		8	9.13	8.57
	4		GDPS-160		20	30	17.89
6	18	15.89		15.18	15.54		
8	18	12.56		11.56	12.06		
10	14	10.89		9.53	10.26		
12	12	8.56		7.5	8.03		
BHSig-260	4	0	20	30	34.12	27.21	30.66
	6		18		27.12	26.12	26.62
	8		16		24.10	26.0	20.05
	10		14		20.1	24.18	22.14
	12		12		18.42	23.1	20.76
CEDAR	4	0	20	24	10.12	9.12	9.62
	6		18		8.2	8.4	8.3
	8		16		7.46	7.86	7.66
	10		14		6.12	7.2	6.66
	12		12		5.01	6.12	5.57

Cuadro 5.1

Dataset	Método	# usuarios	Exactitud	FAR	FRR
CEDAR	SigNet	55	100.0	0	0
GDPS-300		300	76.83	23.17	23.17
		300 (falsificaciones ingenuas)	65.36	34.64	34.64
GDPS-SSC		4000	77.76	22.24	22.24
Bengalí		100	86.11	13.89	13.89
Hindú		100	84.64	15.36	15.36

Cuadro 5.2: Resultados de Dey et al. [19]

Dataset	# muestras por usuario	Características	EER (%)	
GDPS-160	5	SigNet	3.23	
	12		2.63	
GDPS-300	5		3.92	
	12		3.15	
GDPS-160	5		SigNet-F	2.41
	12			1.72
GDPS-300	5	2.42		
	12	1.69		

Cuadro 5.3: Resultados de Haffemann et al. [25]

Características	Método de clasificación	12 muestras	5 muestras
HOG-Polar	USVM	19.58 %	21.73 %
HOG-Mallado		21.13 %	22.65 %
LBP-Mallado		19.84 %	22.90 %
HOG-Polar	GSVM	23.57 %	22.79 %
HOG-Mallado		24.13 %	29.61 %
LBP-Mallado		35.82 %	34.11 %
Todos Combinados		15.41 %	17.65 %

Cuadro 5.4: Resultados de Yilmaz et al. Están dados en función del EER [45].

Capítulo 6

Diseño e implementación

En el desarrollo del presente trabajo se implementaron tres métodos para la extracción de características con las cuales realizar la verificación de las firmas. En primera instancia, basándonos en el trabajo de Kekre et al. [31], usamos la morfología matemática y otras métricas para extraer características globales. En segundo lugar, usamos un método de carácter recursivo para extraer los centros de masa de sucesivas subdivisiones de la imagen, y posteriormente computar los valores medianos de los centros de masa para cada subdivisión con el fin de establecer un umbral que establece la suma de las distancias entre los centros de masa correspondiente a cada subdivisión para cada imagen de un mismo usuario y el valor mediano.

En último lugar, inspirados en el método anterior, usamos una ventana de $N \times N$ píxeles centrados en 30 centros de masa (15 resultado de 4 subdivisiones empezando por el eje horizontal, y 15 empezando por el vertical) computados de forma recursiva, para obtener características locales mediante el uso de histogramas de gradientes orientados y los patrones locales binarios. Posteriormente, usando las características dadas por el primer y tercer método, aplicamos un análisis de componentes principales para reducir la dimensión del vector característico (que contaba con un total de 195 componentes) y contrastamos el poder predictivo de los métodos.

Para la verificación, usamos varios métodos para la clasificación: Máquinas de vectores de soporte, tanto con bases lineales como con bases radiales, Random Forest, k -vecinos, así como redes neuronales artificiales (perceptrón multicapa) fueron los métodos escogidos para realizar esta tarea. En este capítulo se describen las consideraciones para el diseño y la implementación de nuestro sistema. En la sección 6.1 se describen las bases de datos usadas

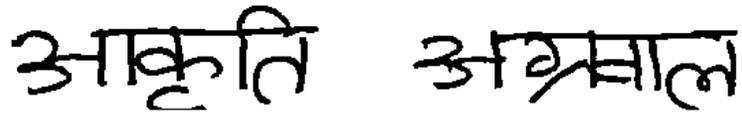
para la verificación y entrenamiento del sistema. En la sección 6.2 se habla de los métodos de preprocesamiento de imágenes aplicados a las bases de datos usadas, en la sección 6.3 se describen los procesos de extracción de características, y de último, en la sección 6.4 detallamos los métodos a usar en el paso correspondiente a la verificación.

6.1. Bases de datos

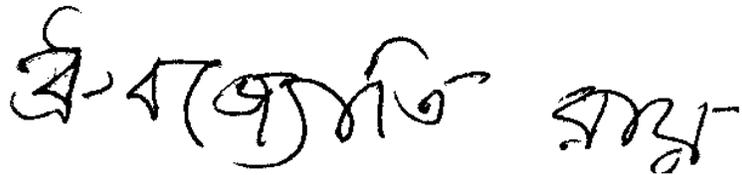
Usamos bases de datos con características heterogéneas para el entrenamiento y la verificación de los métodos propuestos, y como punto de comparación con los trabajos previos, pues son ampliamente utilizadas a lo largo de la literatura. En primer lugar usamos la base de datos que recibe por nombre BHSig260, que a su vez consta de dos sub-bases de datos. La primera de estas posee las firmas de 160 usuarios, escritas en lenguaje Hindú (fig. 6.1a), mientras que la segunda posee las firmas de 100 usuarios escritas en el lenguaje Bengalí (fig. 6.1b). Para ambas existen 24 firmas genuinas y 30 falsificaciones expertas por cada usuario. La segunda base de datos recibe por nombre CEDAR (fig. 6.1c), y posee 55 usuarios con firmas escritas en el alfabeto latino, con 24 firmas genuinas y 24 falsificaciones expertas para cada usuario.

6.2. Pre-procesamiento

La base de datos BHSig260 viene pre-procesada por defecto, con un filtro de suavizado y umbralizado ya aplicados. Sin embargo, por errores en la digitalización de las firmas, varias imágenes contienen ruido y claros defectos (que no están presentes en otras muestras de la misma clase), motivo por el cual se procedió a remediarlos por medio de operaciones de erosión, suavizado, umbralización (algoritmo de Otsu) y transformación (recortando el espacio extra en los extremos de la firma). Para el método de morfología matemática se aplicó además un filtro de esqueletización (esqueletización de Zhang-Suen). La base de datos CEDAR presenta las imágenes en escala de grises, y por consiguiente se procedió a umbralizar las imágenes (Otsu) después de ser aplicado un filtro de suavizado Gaussiano. Las imágenes fueron también recortadas, sin embargo, las imágenes no presentaban mayores defectos, y ningún otro paso posterior fue considerado necesario.



(a) Firma Hindú



(b) Bengali



(c) CEDAR

Figura 6.1: Ejemplos de las 3 bases de datos utilizadas.

6.3. Extracción de características

Las firmas de un mismo usuario poseen ciertas variaciones intra-clase, pero asumimos que estas diferencias son menores que las variaciones inter-clase (entre dos usuarios distintos). Una falsificación posee variaciones mayores al ser comparada contra una firma genuina. Para verificar una firma extraemos características cuantificables que nos permitan compararlas entre sí. En este trabajo utilizamos tres métodos para extraer estas características: la morfología matemática, centros geométricos, y el histograma de gradientes orientados (HOG) más patrones locales binarios (LBP).

6.3.1. Morfología matemática y características globales

Morfología matemática

El proceso de producción de una firma presenta una importante variación intra-clase, es decir, que para distintos intentos de una misma persona de producir una firma, habrán diferencias en el producto final. Mediante este método se pretende establecer un margen donde las variaciones intra-clase

Color	Negro	Rojo	Verde	Azul	Fondo	Blanco	A	B	C	Fondo 2
R	0	255	0	0	0	255	0	255	255	255
G	0	0	255	0	100	255	255	0	255	155
B	0	0	0	255	96	255	255	255	0	159

Cuadro 6.1: Posibles resultados de la aplicación de la disyunción exclusiva.

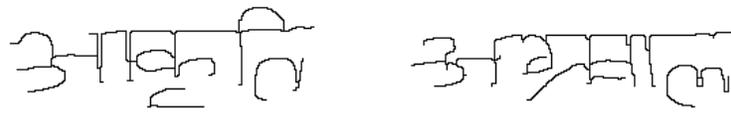
son aceptadas, y aquellas que posean una varianza mayor a estas sean consideradas como pertenecientes a una clase distinta (la firma es falsa, o no pertenece al usuario). Para esto aplicamos una serie de filtros morfológicos al esqueleto de la firma. Este esqueleto (fig. 6.2a) es conseguido a través del filtro de Zhang-Suen, y al resultado le es aplicado una serie de operaciones de dilatación. Cada aplicación sucesiva se efectúa con un radio y color distinto, para un total de 4 iteraciones con los radios $r_1 = 3$, $r_2 = 6$, $r_3 = 10$ y $r_4 = 16$ y los colores negro, rojo, verde y azul, respectivamente. El fondo de la firma es rellenado con un color sólido ($R = 0$, $G = 100$, $B = 96$) (fig. 6.2b).

Para la verificación se toma el esqueleto de la imagen cuestionada, y se efectúa una operación de disyunción exclusiva (xor) sobre los píxeles RGB pertenecientes a esta y los de una firma prototipo (fig. 6.2c), conseguida mediante el proceso descrito anteriormente. Se asume que la firma cuestionada ha sido umbralizada con intensidades en sus píxeles dentro del conjunto $\{0,255\}$, por lo tanto se cuenta con un conjunto definido de posibles valores arrojados por la aplicación de la disyunción exclusiva. Cada uno de estos valores corresponde a un color en formato RGB (tabla 6.1), y representan distintos niveles de variación entre la firma a verificar y la firma prototipo. El color negro representa no-variación, el rojo, verde y azul representan variaciones sobre los márgenes creados por la operación de dilatación, mientras que el color de fondo 1 representa variaciones fuera de estos. Los colores A, B, C y el color de fondo 2 no nos dan información acerca de la variación de la línea.

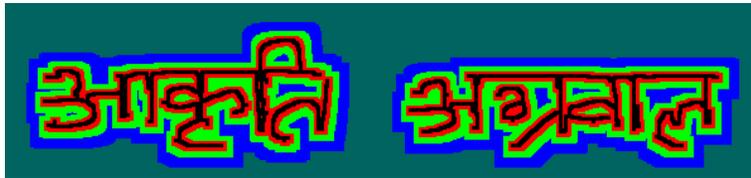
Características globales

Para complementar el método anterior, se extrajeron 5 características adicionales de carácter global:

Centro de masa: El punto resultante de promediar los componentes x, y de todos los píxeles con intensidad $i = 0$ en la imagen.



(a) Ejemplo de la esqueletización Zhang-Suen.



(b) Ejemplo de la aplicación de los distintos niveles de dilatación.



(c) El resultado de la aplicación de la operación de xor entre una firma prototipo y una firma a ser verificada.

Figura 6.2: Ejemplo del método basado en morfología matemática.

Relación de aspecto: Significa la relación entre la longitud y la altura de la firma. Para obtenerla se recorta la imagen al mejor ajuste, y posteriormente se divide la longitud entre la altura.

Densidad: La densidad es la relación entre los píxeles pertenecientes a la firma y aquellos que pertenecen al fondo sobre el cual la firma fue producida. Se obtiene dividiendo la cantidad de píxeles con intensidad $i = 0$ sobre los píxeles con intensidad $i = 1$ de la imagen de la firma.

Proporción de ocupación: Se divide la imagen sobre su centro geométrico, creando dos subdivisiones, la izquierda y la derecha. Posteriormente se divide la cantidad de píxeles con intensidad $i = 0$ de la subdivisión derecha sobre la cantidad de píxeles con intensidad $i = 0$ de la subdivisión izquierda.

Puntos Críticos: Para calcular los puntos críticos, usamos el algoritmo de las esquinas de Harris [28].

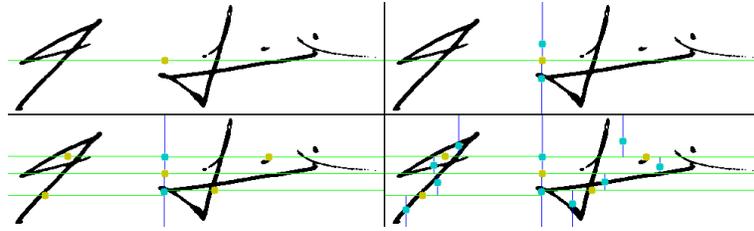
Vector característico

Bajo este método el vector característico cuenta con 15 componentes. Los 10 primeros, concernientes a los colores presentes en la imagen resultante de la aplicación de la operación de disyunción exclusiva, fueron normalizados en relación al número total de píxeles presentes en la imagen, logrando que los valores posibles de estos estén contenidos entre 0 y 1. El componente siguiente fue calculado tomando la distancia euclidiana entre los centros de masa de las dos firmas a comparar, y los 4 componentes restantes fueron normalizados mediante la expresión $c_i = \frac{|y_i - x_i|}{\frac{y_i + x_i}{2}}$, donde y_i y x_i representan la i -ésima característica (relación de aspecto, densidad, proporción de ocupación y puntos críticos) para un par de firmas (y, x) .

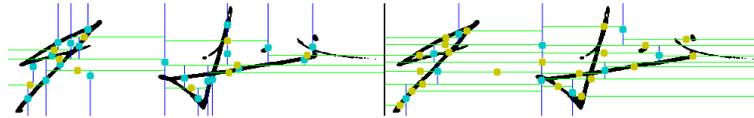
6.3.2. Centros Geométricos

Centros Geométricos y subdivisiones recursivas

En este método, utilizamos un algoritmo recursivo para subdividir la imagen a través de su centro geométrico en ejes alternativos. En un primer paso, se subdivide la imagen en dos cuadrantes a lo largo de uno de los ejes, intersectando el punto p_1 . El punto p_1 se obtiene mediante el cálculo del centro



(a) El resultado de las subdivisiones por cada nivel de profundidad, partiendo por el eje horizontal.



(b) El resultado de la aplicación del algoritmo, partiendo por el eje horizontal en la imagen izquierda, y por el eje vertical en la derecha.

Figura 6.3

geométrico de los píxeles pertenecientes a la firma en una subdivisión dada, es decir, el promedio de los componentes x, y de los píxeles con intensidad $i = 0$. Los cuadrantes resultantes son, a su vez, subdivididos recursivamente a lo largo del eje alternativo, para el primer cuadrante a lo largo de la línea que intersecta al punto p_2 , y a lo largo de la línea que intersecta al punto p_3 para el segundo. Esto se hace sucesivamente hasta llegar a la condición de parada que no es más que la profundidad de recursión pasada al algoritmo (fig. 6.3a). Así, para una profundidad d , tenemos un total de $2^d - 1$ centros geométricos. Para la adquisición de 62 centros geométricos el algoritmo se aplica de forma alternativa, siendo ejecutado una vez empezando por subdividir a lo largo del eje x , y luego se ejecuta de nuevo, empezando por subdividir a lo largo del eje y (fig. 6.3b).

Vector característico

Para este método, el vector característico consta únicamente de los 62 centros geométricos resultantes de la aplicación del método anteriormente descrito. Por lo tanto, el vector está compuesto de 124 valores agrupados en pares, que representan el par de coordenadas (x, y) para cada centro.

6.3.3. HOG y LBP

La hipótesis detrás de este método es que, en primer lugar, los centros de masa en distintas subdivisiones para distintas imágenes pertenecientes a un mismo usuario presentan una pequeña varianza. En segundo lugar, considera que el ambiente alrededor de dichos centros de masa es similar, es decir, la orientación de las líneas, así como las esquinas y la disposición de los detalles del trazo podrían ser similares para distintas firmas de un mismo usuario alrededor de estos centros de masa. Consideramos que esto podría actuar como un descriptor local de la imagen útil para la tarea de clasificación. Para adquirir estas características empleamos dos métodos basados en histogramas: el histograma de gradientes orientados y los patrones binarios locales.

HOG

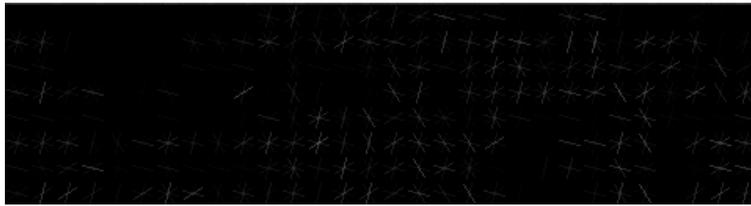
El histograma de gradientes orientados provee la distribución de las orientaciones de las líneas sobre 180° o 360° con una resolución arbitraria. Para calcularlo se toma una imagen en escala de grises y se aplica el operador de Sobel de dimensión $= 1$ (3.7) para cada uno de los ejes, y se calcula la magnitud (3.9) así como la orientación del gradiente 3.8. El histograma se calcula para un rango uniformemente distribuido de orientaciones, de forma que para un histograma de 8 subdivisiones podemos determinar la distribución de las orientaciones con una resolución de 22.5° o 45° , dependiendo si se toma en cuenta el signo del gradiente o no. La votación para cada una de estas subdivisiones del histograma se hace en función de la magnitud del gradiente o de alguna función arbitraria del mismo. Existe una variante que considera los histogramas de forma piramidal, calculando el histograma para un área de una imagen y sus respectivas subdivisiones. Para el presente trabajo, el cálculo se hizo de forma piramidal, utilizando una ventana de 32 o 64 píxeles alrededor del centro de masa correspondiente, y 4 subdivisiones, de 16 y 32 píxeles respectivamente.

LBP

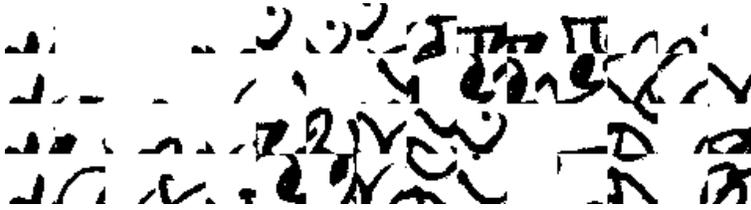
El descriptor LBP se calcula dividiendo la imagen a examinar en celdas de un tamaño determinado, y para cada píxel dentro de la celda, se compara este con cada uno de sus N vecinos, siguiendo la comparación en algún sentido predeterminado. Para aquellos píxeles vecinos cuyo valor sea mayor al píxel central, se considera su valor como 1, y en caso contrario, como 0.



(a) *La representación de los patrones locales binarios.*



(b) *La representación de los histogramas de gradientes orientados.*



(c) *Las dos primeras filas corresponden a las divisiones horizontales y verticales de la firma a verificar, y abajo las de alguna genuina para el usuario.*

Figura 6.4

Concatenando estos valores, siguiendo el sentido predeterminado, se obtiene un número binario de tamaño N . La frecuencia de los números binarios resultantes es contabilizada en un histograma que toma en cuenta los valores resultantes de este procedimiento sobre todos los píxeles dentro de una misma celda. Opcionalmente se puede normalizar el histograma. Los píxeles vecinos a ser tomados en cuenta es un parámetro libre, así como también lo es el radio de la comparación. Por ejemplo, para un radio $r = 1$ y $N = 8$ se toman los 8 píxeles que rodean al píxel central. Para cada uno de los experimentos se calcularon los patrones locales binarios uniformes (fig. 6.4a), en un radio $r = 1$ con $N = 8$.

Vector característico

Para obtener el vector característico a ser utilizado en la tarea de clasificación, tomamos las distribuciones de los gradientes orientados y los patrones locales binarios en ventanas de $N \times N$ píxeles alrededor de 30 puntos correspondientes a los centros geométricos (15 horizontales y 15 verticales) de una firma prototipo y la firma a verificar, utilizando el algoritmo descrito anteriormente (6.3.2) para obtener dichos puntos. Luego calculamos la correlación entre los histogramas a través de una prueba de χ^2 entre los histogramas de las subdivisiones correspondientes para ambas imágenes. Esto otorga un vector de 180 componentes (150 para los histogramas de gradientes orientados, y el resto para los patrones locales binarios) cuyos valores oscilan entre 0 y 1.

6.4. Verificación

Una vez obtenidas las características, utilizamos diversas técnicas del aprendizaje automático para construir un modelo que nos permita clasificar pares de firmas. Diversos métodos son capaces de encontrar distintas correlaciones en los datos para realizar su predicción, así que utilizamos varios métodos, y de esta forma comparar su rendimiento.

6.4.1. Morfología matemática, HOG y LBP

Para la verificación usando estas características utilizamos 4 métodos distintos:

Máquinas de vectores de soporte: Se entrenaron 3 máquinas de vectores de soporte para cada uno de los experimentos. Utilizamos una SVM con kernel lineal, y otra con kernel de base radial. Para una tercera opción, se utilizó ensamble de 10 SVM, cada una entrenada con un décimo de las muestras de entrenamiento. Este tipo de ensambles reciben por nombre clasificadores de Bagging [11].

k-NN: Para este método usamos una búsqueda de k-NN con $k = 29$.

Random Forest: El Random Forest contó, para todos los experimentos, con una profundidad máxima $d = 9$.

Perceptrón multicapa: Entrenamos un clasificador basado en una red neuronal con 4 capas escondidas. La cantidad de neuronas en esta capa fue parametrizada en función al número de muestras usadas en el entrenamiento. El modelo consiste de varios meta-parámetros: Para prevenir un sobre-ajuste en el modelo, el parámetro de dropout especifica la probabilidad con la que una neurona será descartada temporalmente en una instancia de entrenamiento. El valor usado fue de $p = 0,2$. El framework usado para la creación del modelo ofrece varios algoritmos para ejecutar el descenso gradiente; estos algoritmos reciben por nombre optimizadores. Utilizamos el RMSprop para todos los experimentos. Ejecutar el descenso gradiente para cada instancia de entrenamiento puede ser costoso y prevenir una buena generalización a lo largo de todas las muestras. En consecuencia, el meta-parámetro de lote define el número de instancias de entrenamiento entre sucesivas ejecuciones del algoritmo especificado por el optimizador, optamos por un tamaño de lote $b = 128$. Finalmente, la red fue entrenada por un total 128 iteraciones.

Los parámetros k y d fueron escogidos a partir de una búsqueda exhaustiva sobre el intervalo $(1, 30)$ de posibles valores. Para realizar una búsqueda que se aproximara al valor óptimo dentro de nuestros límites de tiempo, optamos por disminuir la dimensionalidad de los vectores característicos por medio de PCA, y así al contar con menos componentes acelerar este proceso. De esta forma, los valores que minimizaran el error total fueron escogidos (fig. 6.5).

6.4.2. Centros geométricos

Para la verificación de los centros geométricos, primero construimos un prototipo de vector característico para cada usuario. Para calcularlo, tomamos n firmas a ser usadas para su construcción y por cada centro geométrico $x_i \in (x_1, x_2, \dots, x_s)$ correspondiente a cada subdivisión i , obtenemos la posición mediana entre n muestras, $x_{i_{med}} = med(x_{i,1}, x_{i,2}, \dots, x_{i,n})$.

En vista de que los elementos a ordenar poseen dos componentes, para encontrar la mediana en este paso utilizamos el algoritmo de Weiszfeld que calcula una aproximación del punto mediano de un conjunto de elementos, refinándola a lo largo de varias iteraciones hasta llegar a una condición de parada. En nuestro caso, consideramos como condición de parada cuando el

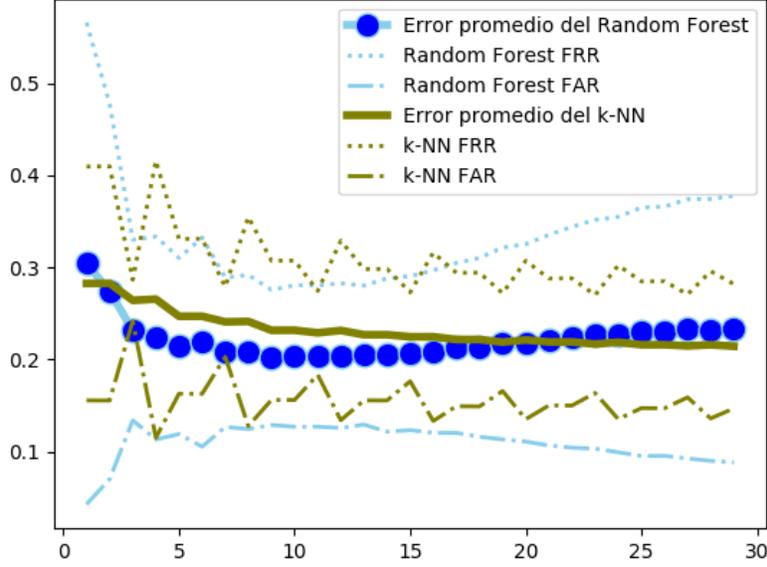


Figura 6.5: Parámetros k y d

El error promedio para los parámetros k y d sobre el intervalo de posibles valores $(1, 30)$.

error absoluto entre iteraciones sucesivas sea igual o menor a 0.01 píxeles. Cada iteración está definida por la fórmula 6.2.

Posteriormente, calculamos las distancias euclidianas de cada uno de los s centros geométricos $x_i \in (x_1, x_2, \dots, x_s)$ para cada una de las muestras con respecto a la mediana calculada en el punto anterior, $d_{i,j} = \sqrt{x_{imed}^2 - x_{i,j}^2}$, obteniendo n distancias por cada centro geométrico. En el siguiente paso, procedemos a calcular el promedio y la desviación estándar (d_{avg} y ρ_i) de las distancias por cada centro geométrico. Estos dos parámetros son usados para calcular el umbral t (6.1).

$$t = \sqrt{\sum_{i=1}^s d_{avg} + \rho_i} \quad (6.1)$$

Para verificar una firma q , calculamos sus centros geométricos y calculamos las s distancias entre estos y su correspondiente x_{imed} , $c_i = \sqrt{x_{imed}^2 - x_{i,q}^2}$.

Si $\sqrt{\sum_{i=1}^s c_i} \leq t$ para las subdivisiones horizontales y verticales, decimos que la firma es válida, en caso contrario la firma es rechazada.

$$y_{i+1} = \frac{\sum_{j=1}^s \frac{x_j}{\|x_j - y_i\|}}{\sum_{j=1}^s \frac{1}{\|x_j - y_i\|}} \quad (6.2)$$

En el siguiente capítulo procedemos a exponer los resultados de la implementación de los métodos de extracción de características y verificación aquí descritos. En el entrenamiento de los modelos de verificación, utilizamos un total de 130272 instancias de entrenamiento, correspondientes a pares de firmas genuina-genuina para un mismo usuario, y 164016 correspondientes a pares genuina-falsificación experta. De los pares genuina-genuina, 22632 corresponde a la base de datos CEDAR, 41400 a la Bengali y las restantes 66240 a la Hindú. Para los pares genuina-falsificación experta, 23616, 54000 y 86400 respectivamente.

Capítulo 7

Resultados

Para medir el rendimiento de los métodos de verificación, usamos 2 de las métricas expuestas en 5.2, el porcentaje de falsa aceptación (FAR) y el porcentaje de falso rechazo (FRR). Usamos el 75 % de las muestras tanto genuinas como falsificaciones expertas de cada una de las bases de datos para entrenar los modelos, y verificamos con el restante 25 %, totalizando 7728 pares de firmas genuina-genuina para la base de datos CEDAR, 13800 para la Bengali y 21528 para la Hindú, con 8064, 18000 y 28800 pares genuina-falsificación experta, respectivamente. Para los modelos globales utilizamos la suma de todas las instancias, es decir, 43056 pares genuina-genuina y 54864 pares genuina-falsificación experta, para un total de 97920 instancias.

Aunque las bases de datos no incluyen falsificaciones ingenuas, para simularlas utilizamos firmas genuinas de los distintos usuarios tal como propone Hafemann [25]. El rendimiento obtenido de los modelos al ser enfrentados a este tipo de falsificación es elevado, las tablas que detallan los resultados por cada uno de los métodos pueden ser encontrados en los anexos 9.

En la construcción del umbral (6.1) para el método basado en centros geométricos, seleccionamos las firmas de acuerdo a su número correlativo dentro del conjunto de firmas para cada usuario. Así, para el umbral construido con $n = 18$ y $n = 12$ firmas, usamos aquellas cuyos números correlativos estuvieran comprendidos entre 1 y n .

7.1. Análisis de las características

Antes de realizar el entrenamiento de los métodos y la verificación, procedimos a analizar la varianza de los vectores característicos para cada uno de los métodos de extracción de características. La varianza puede servir como un indicador del poder discriminatorio de una característica, y es uno de los pasos previos para seleccionar las características a ser utilizadas por el algoritmo de análisis de componentes principales. Sin embargo, no analizamos la varianza de las características arrojadas por este último, al haber sido estas obtenidas de manera automática. Los resultados se encuentran en la figura 7.1

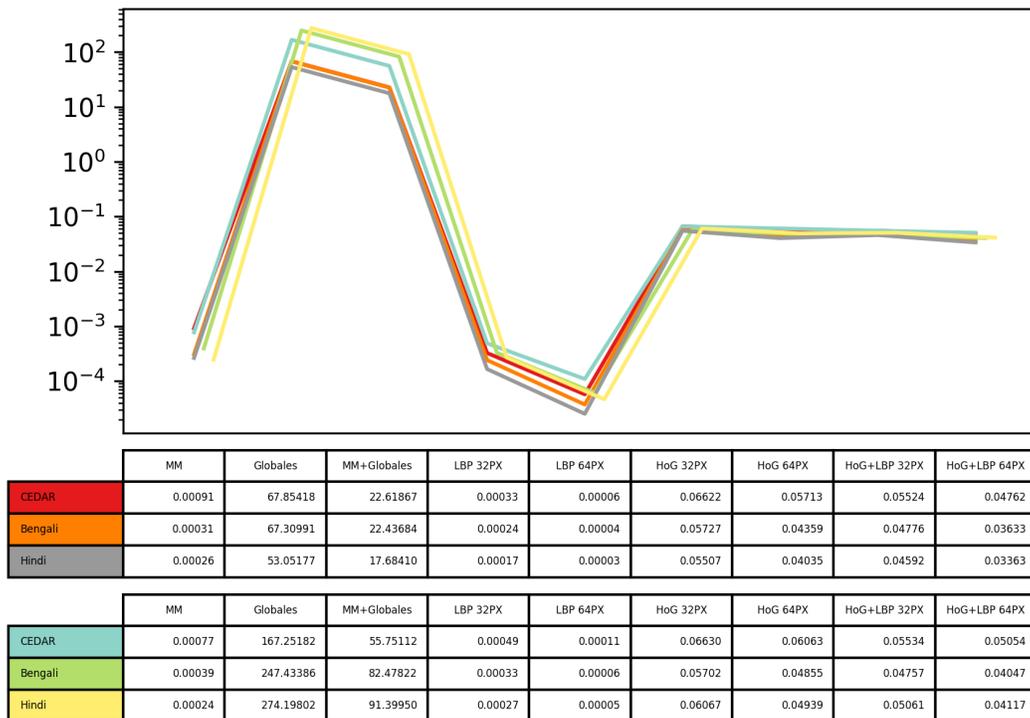


Figura 7.1: Varianza de los vectores característicos.

La varianza está expresada en una escala logarítmica, con los resultados en la tabla superior pertenecientes a los pares genuina-genuina, y los de la tabla inferior a los pares genuina-falsificación experta.

De acuerdo a estos resultados, la mayor varianza pertenece a las métricas globales, seguido por los histogramas de gradientes orientados con ventanas

de 32 y 64 píxeles. La morfología matemática posee una varianza similar a los patrones locales binarios con ventanas de 32 y 64 píxeles, ubicándose en el antepenúltimo lugar, seguido por estos últimos. Con esto, podemos especular que los métodos con menor tasa de error serán aquellos basados en las métricas globales y los histogramas de gradientes orientados.

7.2. Modelos independientes de usuario

Los modelos independientes de usuario son entrenados con la unión de las instancias de entrenamiento para una misma base de datos, de forma que el modelo resultante pueda verificar para cualquier usuario que no haya sido usado en la etapa de entrenamiento o del cual no se posean firmas disponibles.

7.2.1. Morfología matemática y métricas globales.

Para un primer experimento probamos la capacidad discriminatoria de la morfología matemática sin usar las características globales. De los resultados (fig. 7.2) entendemos que la morfología matemática por sí sola tiene un escaso poder discriminatorio, lo cual es claro en el análisis de la varianza de los vectores característicos. Los mejores resultados los encontramos bajo la base de datos Hindú, donde el clasificador de vector soporte (SVC) con kernel lineal tiene como tasas de error promedio de 18.9 % (FRR=21.9, FAR=15.9).

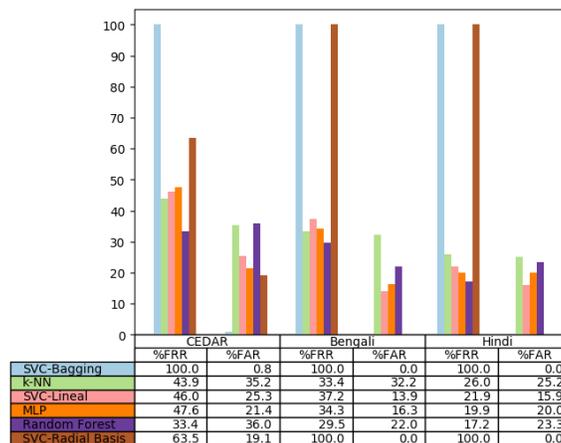


Figura 7.2: Resultados para el método de morfología matemática.

Sin embargo, cuando complementamos la morfología matemática con las métricas globales (fig. 7.3), los resultados mejoran a lo largo de todas las bases de datos, aunque nuevamente es bajo la Hindú que se muestran los mejores resultados, sobre todo con el perceptrón multicapa, obteniendo una tasa de error promedio de 11.6 % (FRR=11.5 %, FAR=11.7 %).

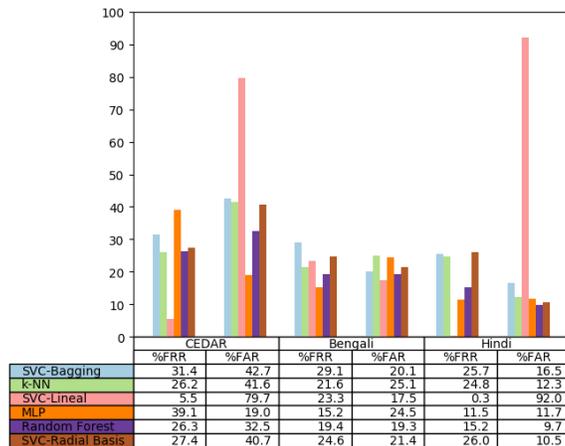


Figura 7.3: Resultados para el método de morfología matemática y métricas globales.

7.2.2. HOG y LBP

En esta sección, presentamos 3 experimentos con las características extraídas mediante el histograma de gradientes orientados y patrones locales binarios. Probamos los métodos con ventanas de 32 y 64 píxeles, y además, analizamos el rendimiento del histograma de gradientes orientados por cuenta propia con una ventana de 32 píxeles.

Ventana de 32 píxeles

En este experimento, usamos el histograma de gradientes orientados y los patrones locales binarios con una ventana de 32 píxeles (fig. 7.4).

En comparación al método basado en morfología matemática y métricas globales, la proporción de falsa aceptación disminuye de forma general para las bases de datos CEDAR y la Bengali, y se mantiene comparable en la

Hindú. La menor tasa de error promedio la encontramos en el clasificador basado en el SVC de kernel radial (FRR=14.5 %, FAR=15.0 %) y el clasificador de Bagging (FRR=15.4 %, FAR=14.1 %), con 14.75 % para ambos.

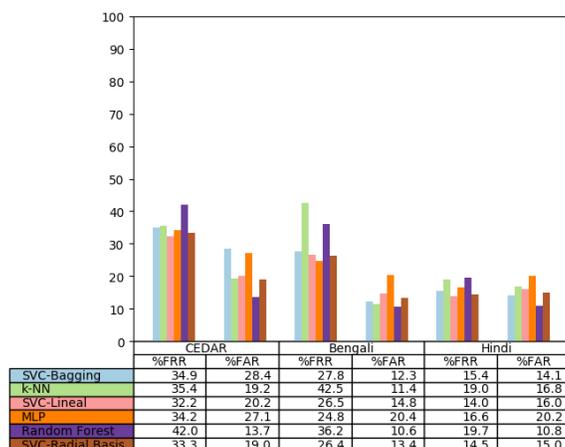


Figura 7.4: Resultados para el método de HOG y LBP con una ventana de 32 píxeles.

Como la varianza de los histogramas de gradientes orientados es muy superior a la presentada por los patrones locales binarios, experimentamos con la clasificación sin usar estos últimos (fig. 7.5), y los resultados presentan, como era de esperarse, una muy ligera variación.

Nuevamente, el error promedio más bajo se encuentra con el clasificador basado en el SVC de kernel radial bajo la base de datos Hindú, con 14.74 % (FRR=14.5 %, FAR=15.0 %).

Ventana de 64 píxeles

Repetimos el proceso anterior, esta vez usando una ventana de 64 píxeles. Los resultados (fig. 7.6) son similares a los obtenidos con una ventana de menor tamaño, considerando que la varianza presentada en el vector característico de este método es ligeramente menor. En este experimento, la tasa de error promedio más baja se encuentra nuevamente en la base de datos Hindú, con 13.75 % (FRR=14.1 %, FAR=13.4 %) usando un clasificador basado en Random Forest.

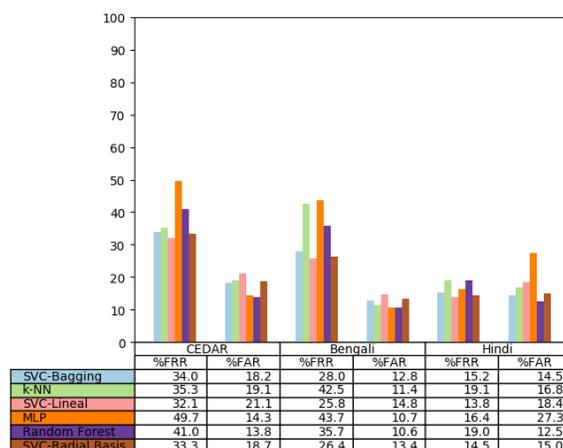


Figura 7.5: Resultados para el método de HOG sin LBP con una ventana de 32 píxeles.

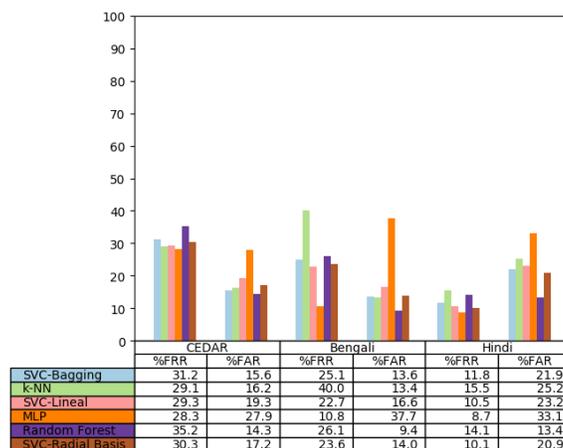


Figura 7.6: Resultados para el método de HOG y LBP con una ventana de 64 píxeles.

7.2.3. Modelo basado en morfología matemática con métricas globales y HOG+LBP

Combinamos las características de las dos secciones anteriores, entrenando los modelos con cada una de las bases de datos (7.7). Para la Bengali y la Hindú, la tasa de falsa aceptación mejora de forma considerable, mientras que la tasa de falso rechazo presenta un leve retroceso.

Sin embargo, este modelo presenta la tasa de error promedio mínima dentro del presente trabajo para los modelos independientes de usuario, nuevamente bajo la base de datos Hindú, con un 10.49 % (FRR=11.9 %, FAR=9.1 %) para el clasificador basado en Random Forest, seguido muy de cerca por el clasificador basado en el SVC con kernel radial, con 10.52 % (FRR=14.0 %, FAR=7.0 %). En los anexos (sección 9) se encuentran las figuras donde presentamos la diferencia en las tasas de error entre este método y los métodos basados en morfología matemática con métricas globales, y HOG con LBP, respectivamente.

7.3. Modelos dependiente de usuario

En este tipo de modelos se entrena un clasificador por cada usuario. Las tasas de error reportadas son el promedio del error dado a lo largo de todos los modelos de cada usuario.

7.3.1. Morfología matemática y métricas globales.

Las tasas de error promedio son más bajas en comparación a su contraparte independiente de usuario. El mejor resultado lo tenemos bajo la base de datos Bengali, con una tasa de error promedio de 2.75 % (FRR=4.1 %, FAR=1.4 %) mediante un Random Forest (fig. 7.8).

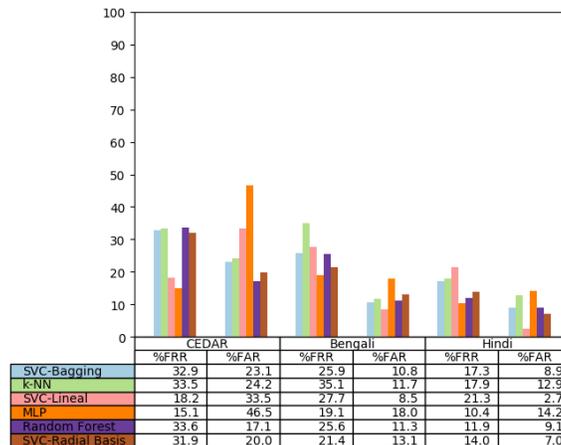


Figura 7.7: Resultados para el método combinado.

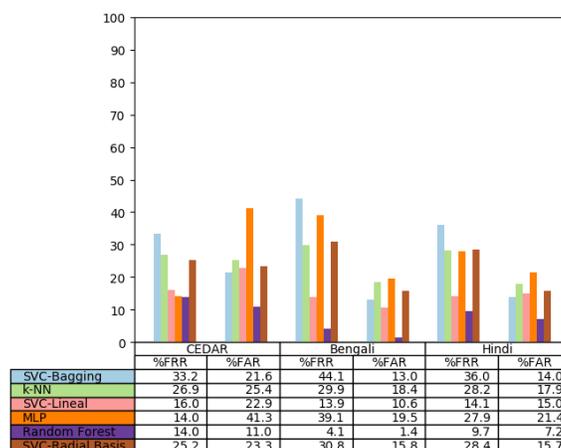


Figura 7.8: Resultados para el método de morfología matemática más métricas globales dependiente de usuario.

7.3.2. HOG y LBP

Ventana de 32 píxeles

La máquina de vectores de soporte con kernel lineal bajo la base de datos Bengalí arroja un minúsculo 0.7% de error promedio (FRR=0.2%, FAR=1.2%), así los resultados usando una ventana de 32 píxeles se ubican muy por debajo del mejor resultado usando un modelo independiente de usuario (fig. 7.9).

Ventana de 64 píxeles

El mejor resultado de los modelos dependiente de usuario lo encontramos bajo la ventana de 64 píxeles (fig. 7.10), con un error promedio de apenas 0.35% (FRR=0.4%, FAR=0.3%) usando un Random Forest.

7.3.3. Morfología matemática con métricas globales y HOG+LBP

Y por último, nuevamente encontramos que el mejor rendimiento bajo estas características viene de la mano de un Random Forest bajo la base de datos Bengalí. Con un error promedio de 0.75% (FRR=0.9%, FAR=0.6%), la combinación de todas las características sigue teniendo un buen rendimiento

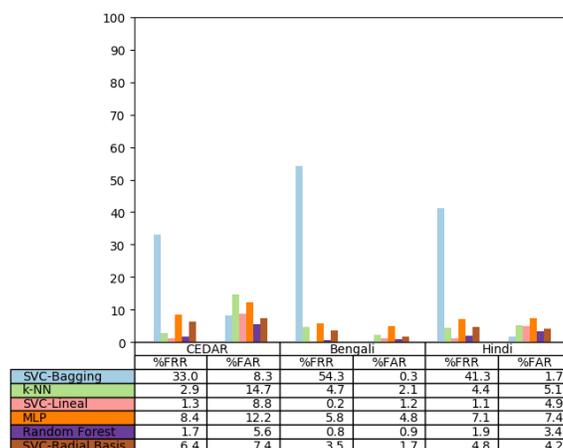


Figura 7.9: Resultados para el método de HOG y LBP con una ventana de 32 píxeles dependiente de usuario.

(fig. 7.11), pero comparativamente no se lleva el mejor puesto tal como sucede con los modelos independientes de usuario.

7.4. Modelos Globales

Con los modelos globales pretendemos añadir más instancias para el entrenamiento, de forma que los métodos de aprendizaje automático tengan un espacio más grande sobre el cual ajustar el modelo y así mejorar su capacidad de aprendizaje. Tomamos los mismos métodos para la extracción de características que los expuestos en las secciones anteriores, y entrenamos un único modelo con la concatenación de las 3 bases de datos. De forma similar procedimos con las instancias reservadas para el propósito de verificación.

7.4.1. Modelo basado en PCA

En este experimento reducimos la dimensionalidad de los vectores característicos a través de un proceso automático de análisis de componentes principales. Combinamos los componentes de los vectores característicos de la morfología matemática con métricas globales y el histograma de gradientes orientados con patrones locales binarios, creando así un vector característico de 195 componentes. Mediante el análisis de componentes principales reduci-

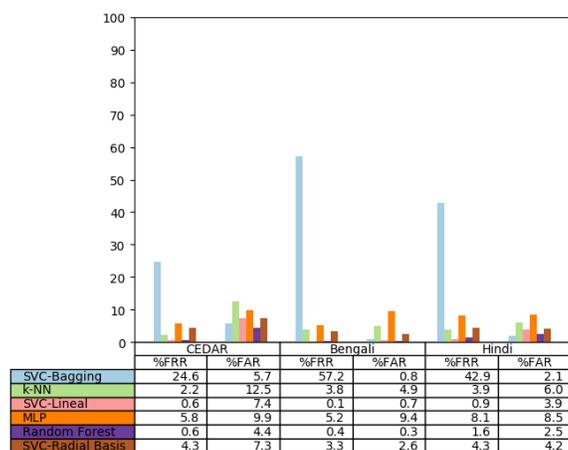


Figura 7.10: Resultados para el método de HOG y LBP con una ventana de 64 píxeles dependiente de usuario.

mos este vector a uno de 10 componentes (94.87% de reducción), y entrenamos los métodos anteriores con la unión de todas las muestras de las 3 bases de datos (fig. 7.12). La tasa de error promedio más baja se ubica en 18.83% (FRR=24.2%, FAR=13.5%) para el perceptrón multicapa, aunque todos los métodos tienen un rendimiento similar.

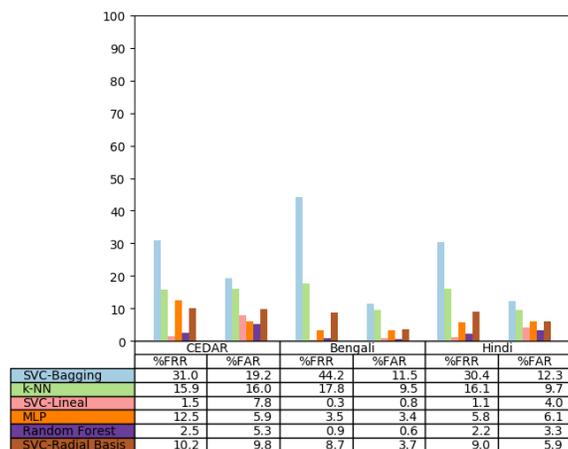


Figura 7.11: Resultados para el método combinado dependiente de usuario.

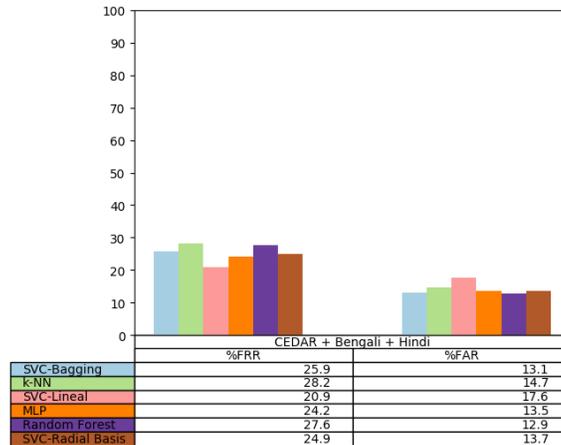


Figura 7.12: Resultados para el método basado en PCA.

7.4.2. Modelo basado en morfología matemática con métricas globales

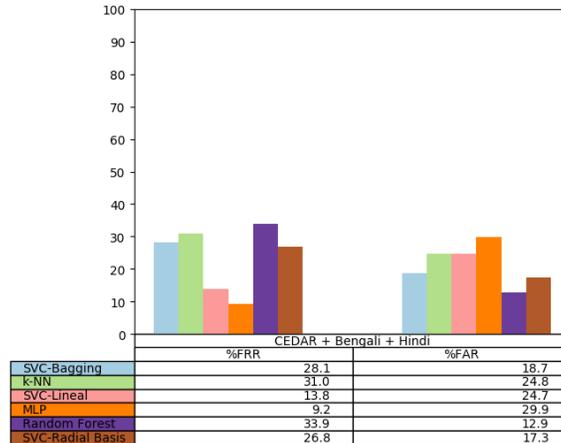


Figura 7.13: Resultados para el modelo global basado en morfología matemática y métricas globales.

El poder discriminatorio falla en mejorar apreciablemente con la adición de más muestras de entrenamiento (fig. 7.13). Aún así, nuevamente el Random Forest es el método de clasificación con la tasa de error promedio más baja, con 18.06 % (FRR=33.0 %, FAR=12.9 %).

7.4.3. Modelo global basado en HOG+LBP

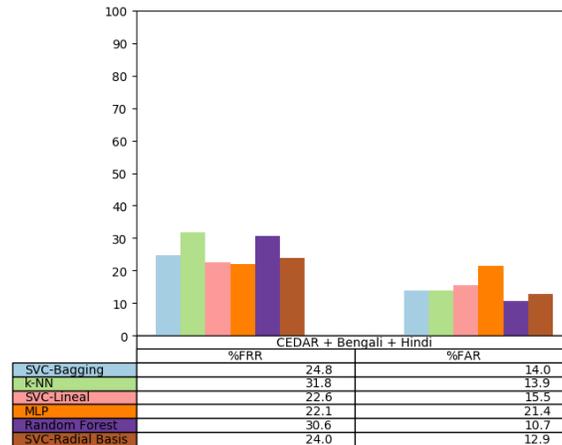


Figura 7.14: Resultados para el modelo global basado en HOG+LBP.

El tamaño de ventana usado es de 32 píxeles, exhibiendo un resultado similar a los obtenidos en el entrenamiento con cada base de datos por separado (fig. 7.14). En este caso, el mejor rendimiento pertenece al SVC con kernel radial, con 18.44 % (FRR=24.0 %, FAR=12.9 %) para la tasa de error promedio.

7.4.4. Modelo basado en morfología matemática con métricas globales y HOG+LBP

Para las características basadas en HOG y LBP, la ventana usada fue de 32 píxeles. Al igual que en los experimentos anteriores, las métricas de rendimiento se encuentran cerca o ligeramente por debajo de las obtenidas con los modelos individuales, sin embargo, la combinación de las características parece ser igualmente efectiva en este modelo global. La tasa de error promedio mínima se ubica en 14.92 % (FRR=21.7 %, FAR=8.2 %), conseguido con el SVC de kernel lineal.

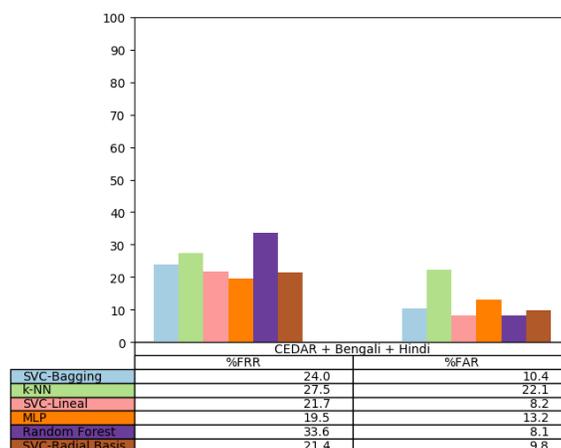


Figura 7.15: Resultados para el modelo global basado en morfología matemática con métricas globales y HOG+LBP

7.5. Modelo basado en centros geométricos

Utilizamos 12 y 18 firmas para la construcción del vector prototipo y el umbral para la verificación (fig. 7.16). A diferencia de los modelos anteriores, el mejor rendimiento es presentado por la base de datos CEDAR, con una tasa de error promedio de 18.25 % (FRR=15.8 %, FAR=20.7 %), conseguido con 12 muestras para la construcción del umbral.

7.6. Comparación contra el estado del arte

En las siguientes tablas podemos comparar el rendimiento de los métodos desarrollados en el presente trabajo contra algunos encontrados en el estado del arte. Comparamos en función de la base de datos usada, y denotamos la naturaleza del método, es decir, dependiente o independiente del usuario. CEDAR es la base de datos más popular entre las que utilizamos, y en la tabla 7.1 podemos apreciar que el rendimiento para los modelos dependiente de usuario es excelente sin llegar a poder ser considerado como estado del arte. Por otra parte, el rendimiento del mejor modelo independiente de usuario está firmemente anclado al último lugar.

BHSig260, tal vez por estar escrita en un alfabeto foráneo para la mayor parte de la comunidad investigadora, cuenta con muy pocas referencias

	Método	FRR	FAR	AER	Tipo
CEDAR	HOG+LBP 64PX (RF)	0.6	4.4	2.5	DU
	HOG+LBP 64PX (k-NN)	29.1	16.2	22.64	IU
	LPQ+DWT [8]	5.01	6.12	5.57	IU
	SigNet [19]	0	0	0	IU
	SigNet-F [26]	4.63	4.63	4.63	DU
	Graph Matching [26]	7.7	8.2	7.9	DU
	Morfología [26]	12.39	11.23	11.81	IU
	Surroundness [26]	-	8.33	8.33	IU
	Chain Code [26]	9.36	7.84	7.84	DU
	Curvelet Transform [26]	-	-	5.6	DU
	Aprendizaje de características [26]	-	-	4.63	DU

Cuadro 7.1: Estado del arte con la base de datos CEDAR.

	Método	FRR	FAR	AER	Tipo
BHSig260	HOG+LBP 64PX (RF)(B)	0.4	0.3	0.35	DU
	HOG+LBP 64PX (RF)(H)	1.5	2.6	2.0	DU
	HOG+LBP 64PX (B+H)	1.07	1.71	1.36	DU
	MM+MG+HOG+LBP 32PX (SVC-RBF)(B)	21.39	13.07	17.23	IU
	MM+MG+HOG+LBP 32PX (RF)(H)	11.9	9.2	10.49	IU
	MM+MG+HOG+LBP 32PX (B+H)	17.74	11.58	14.76	IU
	LPQ+DWT [8]	18.42	23.1	20.76	IU
	SigNet [19]	14.62	14.62	14.62	IU
	ULBP & LBP [37]	32.72	32.72	32.72	IU
	Fuzzy Similarity [3]	17.69	28.61	23.08	DU

Cuadro 7.2: Estado del arte con la base de datos BHSig260.

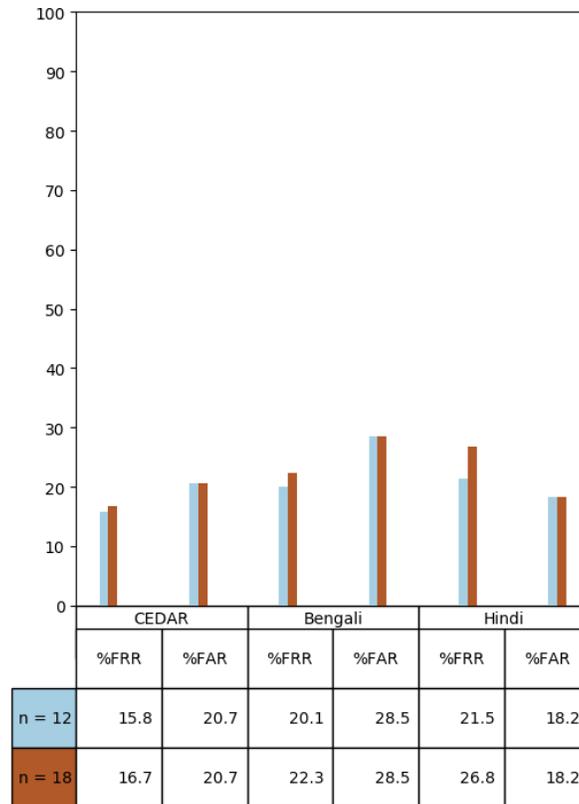


Figura 7.16: Resultados para el modelo basado en centros geométricos.

de las cuales podamos hacer uso para comparar el rendimiento obtenido. Además, en los antecedentes encontrados son pocos los que distinguen entre las dos bases de datos que la conforman, por lo cual comparamos el mejor rendimiento obtenido con cada uno de los alfabetos por separado. Dentro del pequeño universo disponible para contrastar los resultados, nuestros métodos propuestos obtienen excelentes resultados, en particular los propios de los modelos dependientes de usuario entrenados con el alfabeto Bengali, y en el caso de los independientes de usuario, la concatenación de todas las características con el alfabeto Hindú (tabla 7.2).

Capítulo 8

Conclusiones y Trabajos Futuros

A lo largo del presente trabajo implementamos una serie de métodos para la extracción de características a ser usadas en una posterior etapa de clasificación. La naturaleza de cada método varía con respecto a los otros, pues utilizamos tanto características globales (la morfología matemática y las métricas globales) como locales para intentar construir un sistema de clasificación robusto. Para esta etapa de clasificación, probamos una serie de técnicas que se ubican entre las más populares y efectivas en la literatura.

Los métodos usados exhibieron resultados variados, pero generalmente buenos. El rendimiento varía en función de las características, la base de datos y métodos de clasificación usados, y se obtuvieron tasas de error tan bajas como 10.49 % (FRR=11.9 %, FAR=9.1 %) con falsificaciones expertas y 0.7 % de FAR con falsificaciones ingenuas con modelos independientes de usuario. Para los dependientes de usuarios, los resultados mejoran de forma general para las falsificaciones expertas, con el mejor error promedio ubicándose en 0.35 % (FRR=0.4 %, FAR=0.3 %) y presentan una leve aunque significativa regresión para las falsificaciones ingenuas, con un FAR de 2.7 % para el mejor resultado, mientras que los modelos globales, agnósticos en cuanto al alfabeto, no mejoran en ninguna categoría.

Sin embargo, a pesar de los resultados positivos, se presentaron una serie de limitantes para el desarrollo del presente trabajo, entre las cuales tenemos la falta de disponibilidad de un mayor poder de cómputo y capacidad de memoria. Por ejemplo, podríamos haber experimentado con una mayor resolución y una mayor profundidad en el esquema piramidal en la extracción

de características a través del histograma de gradientes orientados, pero la incapacidad de albergar el resultante vector característico en memoria principal haría del entrenamiento de los modelos una tarea impráctica en cuanto al tiempo, puesto que la diferencia en tiempo de acceso entre la memoria principal y la memoria de disco es de 3 órdenes de magnitud en el mejor escenario posible. En otro caso concreto tenemos que el entrenamiento de los modelos globales basados en esta técnica tuvieron que ser efectuados sin la interfaz gráfica del sistema operativo, puesto que los escasos 300MB de memoria que ocupa eran necesarios para proseguir con el entrenamiento sin incurrir en una ralentización por paginación.

Además, en un principio se buscó emplear dos métodos distintos a los usados para la verificación, sin embargo, la carencia material de ciertos dispositivos de hardware imposibilitó la prosecución de estas opciones. Para ser precisos, originalmente se planteó usar redes artificiales convolucionales y redes artificiales siamesas para el descubrimiento automático de características, tal como se plantea en el marco teórico. De este modo, fue la imposibilidad de proseguir estas alternativas las que inspiró la creación de un método propio en el presente trabajo (el histograma de gradientes orientados alrededor de los centros geométricos).

Por otra parte tenemos que la calidad de las bases de datos, aunque aceptable, dista de ser ideal. Si bien se tuvo mucho cuidado a la hora de llevar a cabo la limpieza de los datos, es imposible aseverar con total seguridad que ningún sesgo haya sido introducido en esta etapa. Las bases de datos de mayor calidad, y que además cuentan con la mayor cantidad de referencias a lo largo de la literatura, tales como GDPS o MYCT (mencionadas en la sección 5.3), requieren de coordinación entre las instituciones académicas y la prosecución de un proceso burocrático (que conlleva la firma de un acuerdo legal) para poder ser utilizadas.

Como trabajos futuros proponemos profundizar en el desarrollo de los métodos expuestos. Por ejemplo, emplear una mayor profundidad en el esquema piramidal del histograma de los gradientes orientados, así como extraer otro tipo de características globales.

Algunos métodos dentro del estado del arte utilizan clasificadores híbridos [7], promediando las predicciones de modelos dependientes e independientes de usuario, ya que en vista de la discrepancia en el rendimiento de nuestros métodos dependientes de usuario a la hora de clasificar falsificaciones ingenuas, pensamos que esto podría ser una buena alternativa de investigación.

También creemos que medir el rendimiento de nuestros métodos propuestos, con el fin de establecer un punto de comparación más amplio y así constatar la generalidad de estos, podría ser un buen emprendimiento investigativo. Así mismo, a lo largo de la literatura encontramos otros sistemas que no se limitan a procesar imágenes binarizadas, recomendamos extender este trabajo a imágenes en otros formatos de color.

Además, en nuestro método propuesto usamos una mezcla de descriptores con invariantes (LBP uniforme) y variantes a la rotación (HOG); de este último existen versiones invariantes a la rotación que podrían mejorar el rendimiento de clasificación [33]. El panorama del aprendizaje automático cambia a un ritmo vertiginoso, de igual forma recomendamos proseguir con la utilización de otros métodos a los utilizados en el presente trabajo, o adaptar nuestra propuesta a los que surgan posteriormente.

Capítulo 9

Anexos

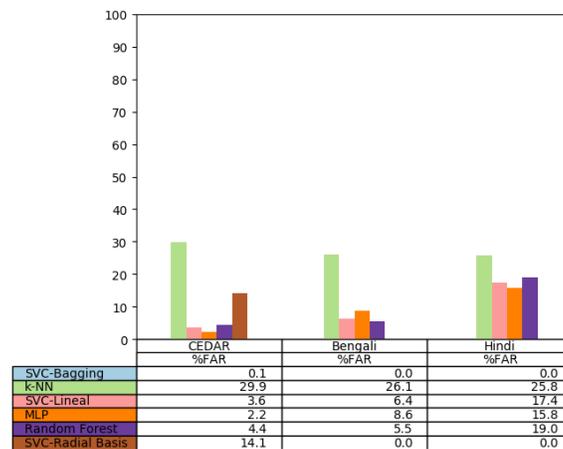


Figura 9.1: Resultados para el modelo basado en morfología matemática, verificado con falsificaciones ingenuas.

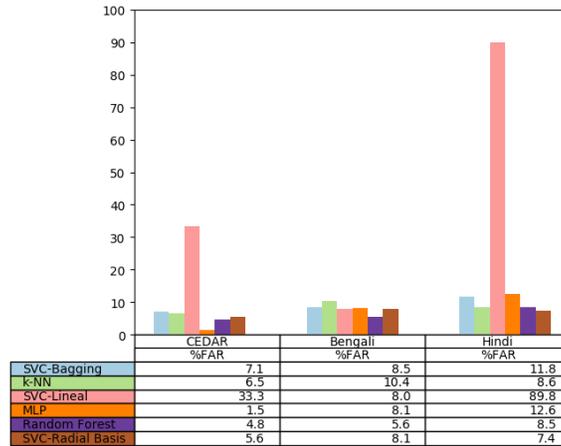


Figura 9.2: Resultados para el modelo basado en morfología matemática más métricas globales, verificado con falsificaciones ingenuas.

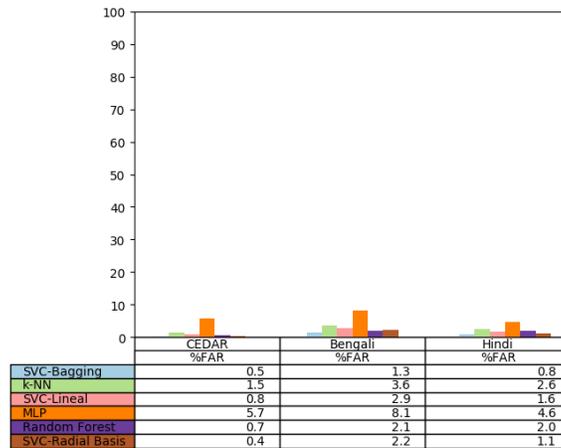


Figura 9.3: Resultados para el modelo basado en HOG+LBP con una ventana de 32 píxeles, verificado con falsificaciones ingenuas.

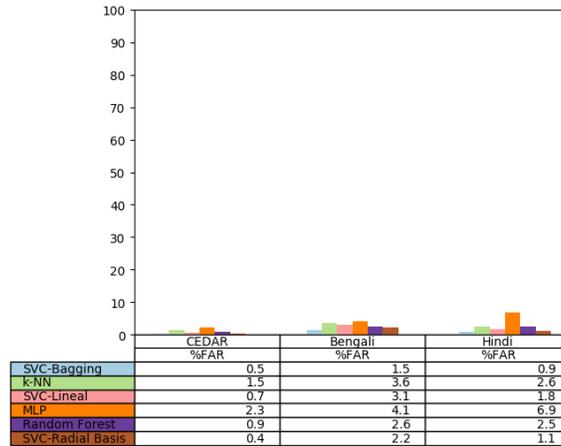


Figura 9.4: Resultados para el modelo basado en HOG sin LBP con una ventana de 32 píxeles, verificado con falsificaciones ingenuas.

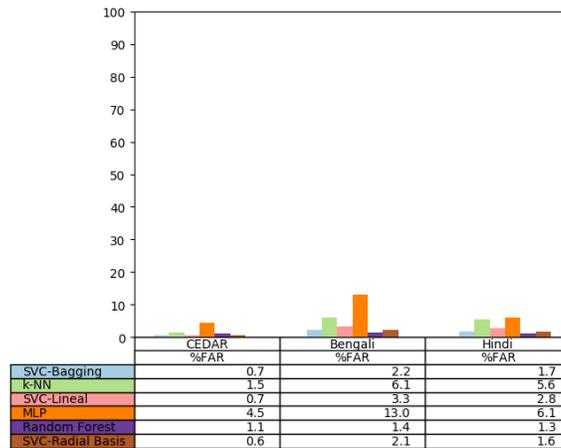


Figura 9.5: Resultados para el modelo basado en HOG+LBP con una ventana de 64 píxeles, verificado con falsificaciones ingenuas.

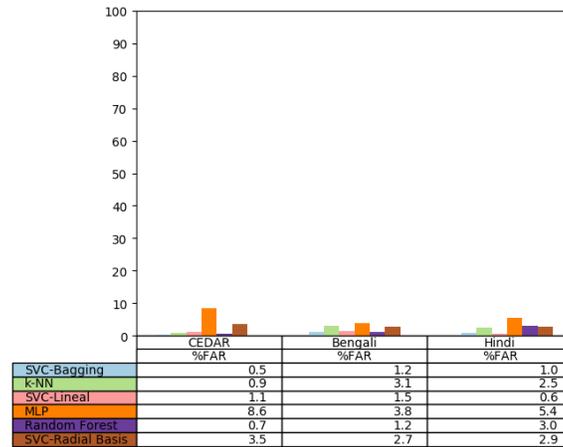


Figura 9.6: Resultados para el modelo basado en morfología matemática con métricas globales y HOG+LBP con una ventana de 32 píxeles, verificado con falsificaciones ingenuas.

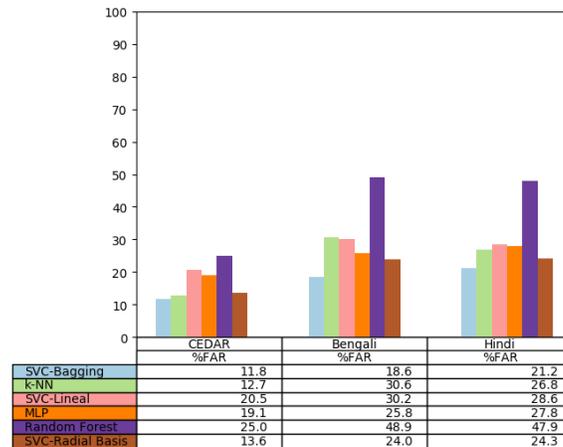


Figura 9.7: Resultados para los modelos dependiente de usuario basados en morfología matemática con métricas globales, verificados con falsificaciones ingenuas.

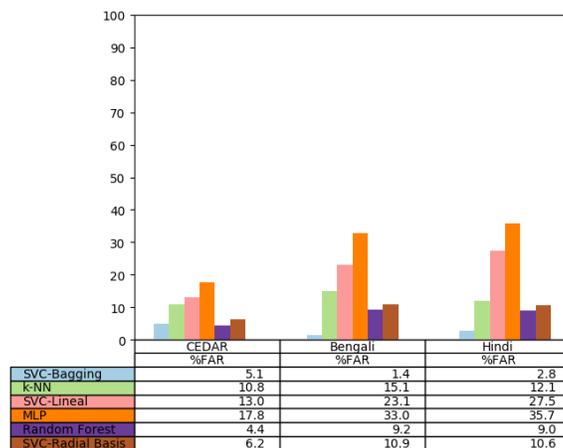


Figura 9.8: Resultados para los modelos dependiente de usuario basados en HOG+LBP con una ventana de 32 píxeles, verificados con falsificaciones ingenuas.

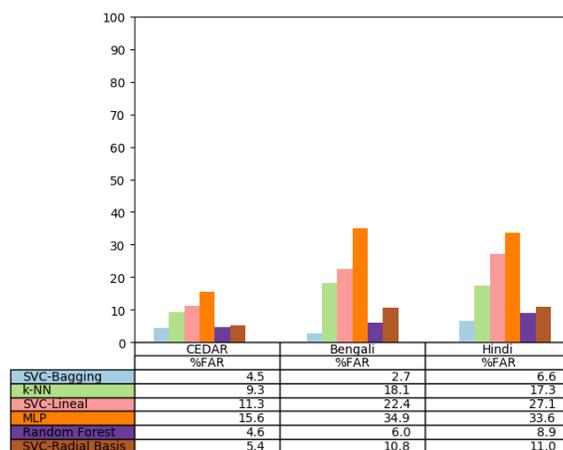


Figura 9.9: Resultados para los modelos dependiente de usuario basados en HOG+LBP con una ventana de 64 píxeles, verificados con falsificaciones ingenuas.

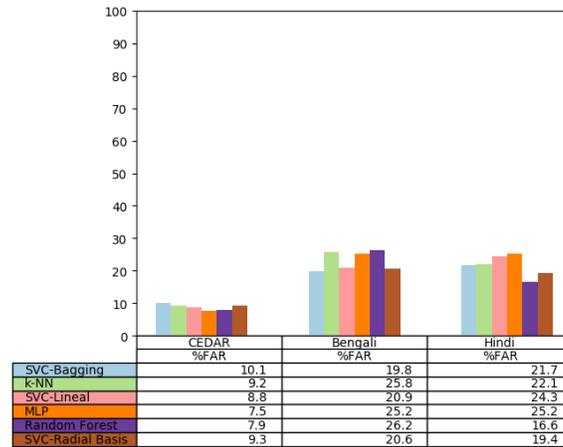


Figura 9.10: Resultados para los modelos dependiente de usuario basados en morfología matemática con métricas globales y HOG+LBP con una ventana de 32 píxeles, verificados con falsificaciones ingenuas.

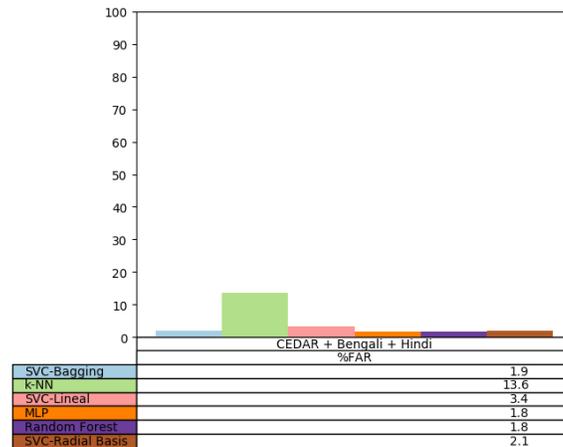


Figura 9.11: Resultados para el modelo basado en PCA, verificado con falsificaciones ingenuas.

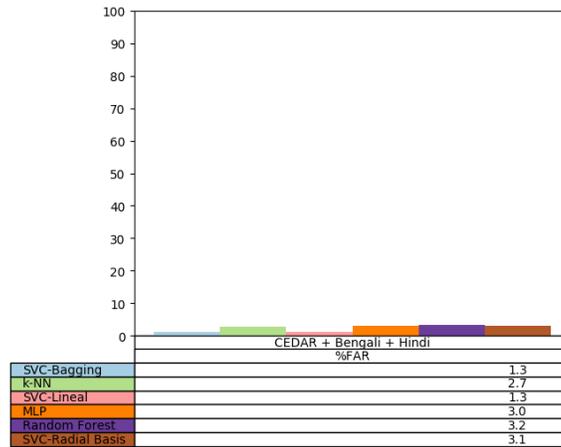


Figura 9.12: Resultados para el modelo global basado en morfología matemática con métricas globales y HOG+LBP con una ventana de 32 píxeles, verificado con falsificaciones ingenuas.

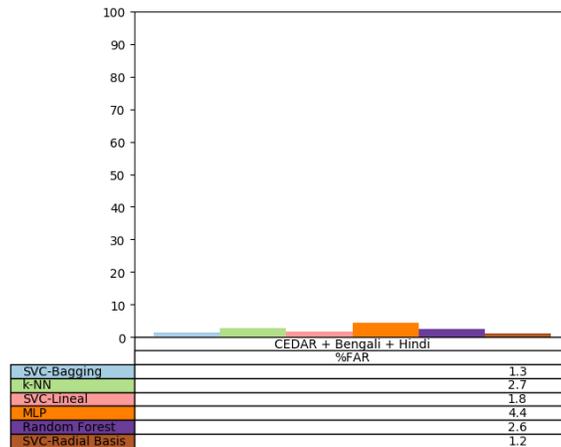


Figura 9.13: Resultados para el modelo global basado en HOG+LBP con una ventana de 32 píxeles, verificado con falsificaciones ingenuas.

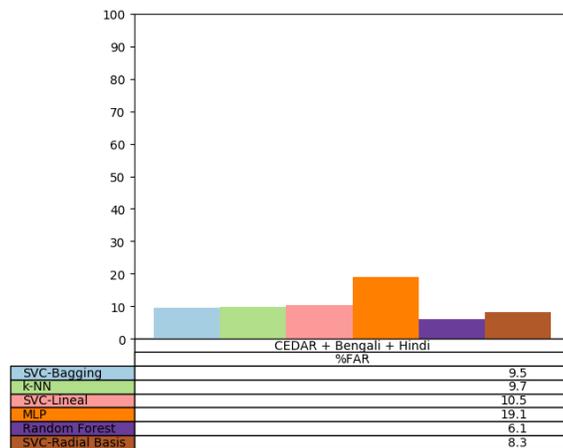


Figura 9.14: Resultados para el modelo global basado en morfología matemática con métricas globales, verificado con falsificaciones ingenuas.

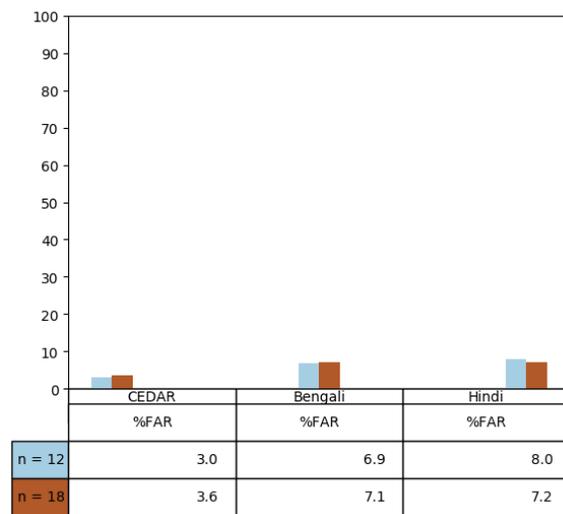


Figura 9.15: Resultados para el modelo basado en centros geométricos, verificado con falsificaciones ingenuas.

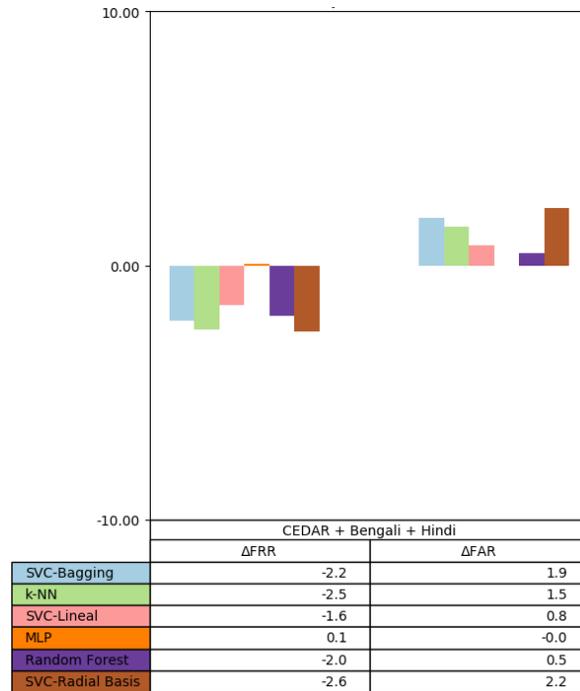


Figura 9.16: Diferencia entre el modelo global basado en HOG+LBP y el mismo método entrenado con cada base de datos.

Las tasas de error para los modelos individuales están ponderadas de acuerdo al número de observaciones pertenecientes a cada base de datos. Para la tasa de falso rechazo, 0.1737 para CEDAR, 0.3177 para la Bengali y 0.5084 para la Hindi. Para la tasa de falsa aceptación, 0.1439 para CEDAR, 0.3292 para la Bengali y 0.5267 para la Hindi. Con esto buscamos establecer la diferencia entre un modelo idealizado, donde el rendimiento es equivalente a la tasa promedio de los tres modelos individuales para cada base de datos, y el arrojado al entrenar el modelo con la concatenación de las tres bases de datos.

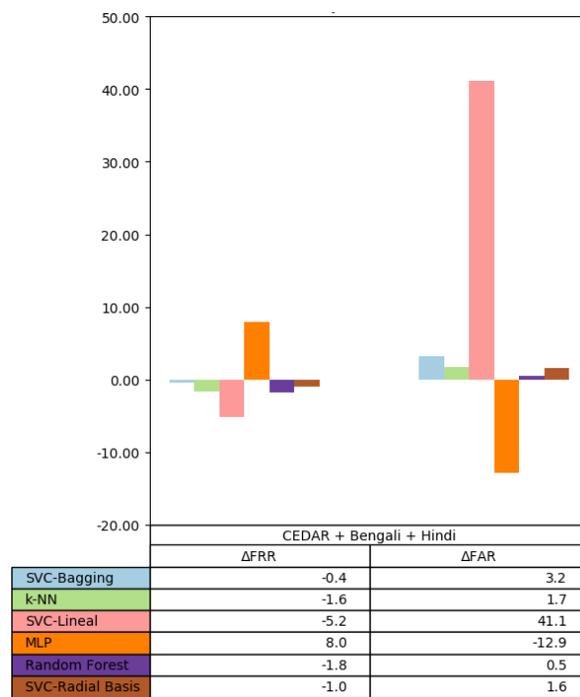


Figura 9.17: Diferencia entre el modelo global basado en MM+Métricas globales y el mismo método entrenado con cada base de datos.

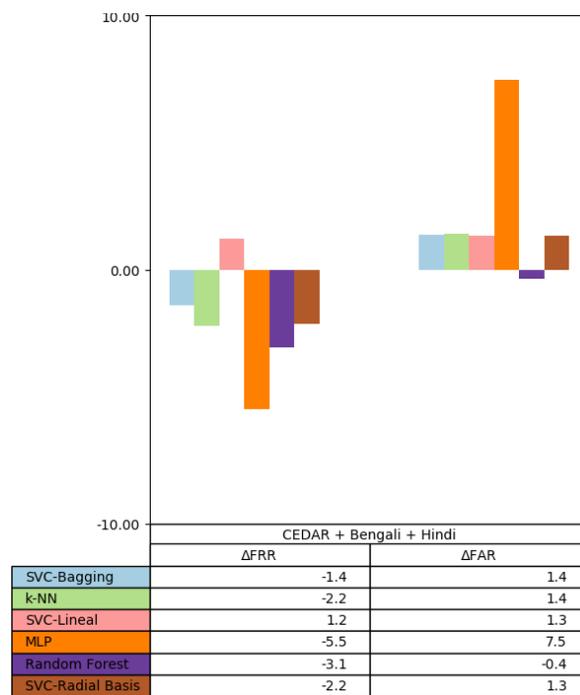


Figura 9.18: Diferencia entre el modelo global combinado y el mismo método entrenado con cada base de datos.

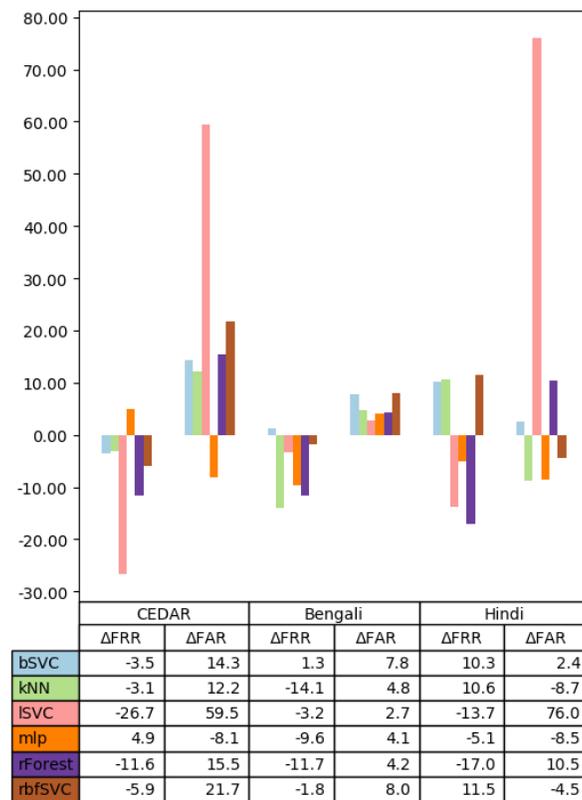


Figura 9.19: Diferencia de las métricas de evaluación entre el método basado en MM+Métricas globales y el método basado en HOG+LBP.

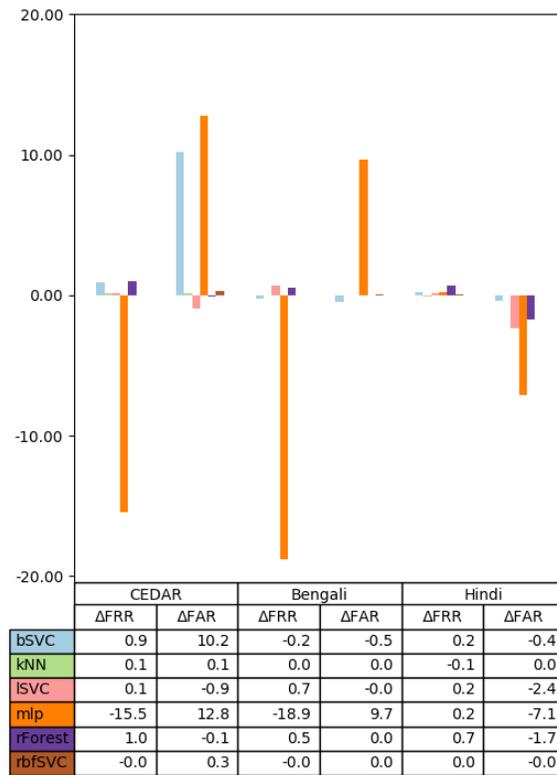


Figura 9.20: Diferencia de las métricas de evaluación entre el método basado en HOG+LBP y HOG únicamente.

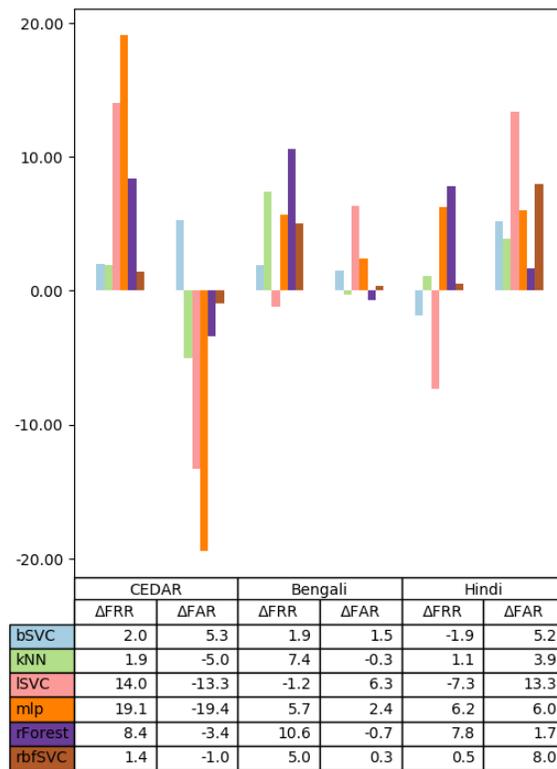


Figura 9.21: Diferencia de las métricas de evaluación entre el método basado en HOG+LBP y la combinación de todas las características.

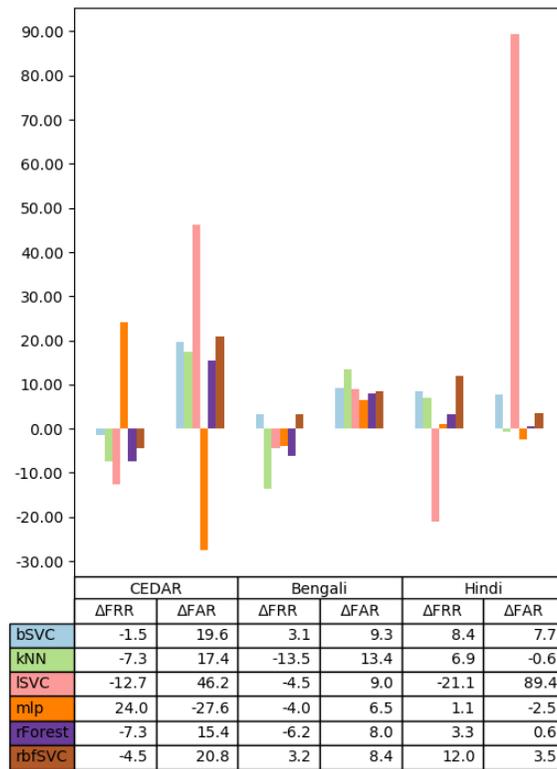


Figura 9.22: Diferencia de las métricas de evaluación entre el método basado en morfología matemática con métricas globales y la combinación de todas las características.

Bibliografía

- [1] A, Pattel Bhumika y Shashwat Kumar: *A Survey on Handwritten Signature Verification Techniques*. International Journal of Advance Research in Computer Science and Management Studies, 3, Julio 2015.
- [2] Aggarwal, Charu C.: *Data Mining: the textbook*. Springer, 2015.
- [3] Alaei, Alireza, Srikanta Pal, Umapada Pal y Michael Blumenstein: *An Efficient Signature Verification Method Based on an Interval Symbolic Representation and a Fuzzy Similarity Measure*. PP:1–1, Mayo 2017.
- [4] Baldwin: *Median filter*. http://situs.biomachina.org/hn06/talks/Baldwin/convolution_filters_new.pdf, 2017. [Online; Recuperado el 22-Diciembre-2017].
- [5] Beal, Vangie: *False Acceptance*. https://www.webopedia.com/TERM/F/false_rejection.html. [Online; Recuperado el 7-Enero-2018].
- [6] Beal, Vangie: *FRR - false rejection rate*. https://www.webopedia.com/TERM/F/false_acceptance.html. [Online; Recuperado el 7-Enero-2018].
- [7] Berkay Yilmaz, Mustafa y Kagan Ozturk: *Hybrid User-Independent and User-Dependent Offline Signature Verification With a Two-Channel CNN*. En *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.
- [8] Bhunia, Ankan Kumar, Alireza Alaei y Partha Pratim Roy: *Signature Verification Approach using Fusion of Hybrid Texture Features*. arXiv:1709.09348v1, Sep 2017.

- [9] Brayer, John M.: *Fourier*. <https://www.cs.unm.edu/~brayer/vision/fourier.html>, 2017. [Online; Recuperado el 22-Diciembre-2017].
- [10] Chris Solomon, Toby Breckon: *Fundamentals of digital image processing*. Willy-Blackwell, 2011.
- [11] Claude Sammut, Geoffrey I. Webb: *Encyclopedia of Machine Learning and Data Mining*. Springer US, 2017.
- [12] contributors, Wikipedia: *Sobel Operator*. https://en.wikipedia.org/wiki/Evaluation_of_binary_classifiers. [Online; Recuperado el 7-Enero-2018].
- [13] contributors, Wikipedia: *Discrete cosine transform* — *Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/wiki/Discrete_cosine_transform, 2017. [Online; Recuperado el 22-Diciembre-2017].
- [14] contributors, Wikipedia: *Discrete wavelet transform* — *Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/wiki/Discrete_wavelet_transform, 2017. [Online; Recuperado el 22-Diciembre-2017].
- [15] contributors, Wikipedia: *Prewitt Operator*. https://en.wikipedia.org/wiki/Prewitt_operator, 2017. [Online; Recuperado el 22-Diciembre-2017].
- [16] contributors, Wikipedia: *Sobel Operator*. https://en.wikipedia.org/wiki/Sobel_operator, 2017. [Online; Recuperado el 22-Diciembre-2017].
- [17] Dabbaghchian, Saeed, Masoumeh P. Ghaemmaghami y Ali Aghagolzadeh: *Feature extraction using discrete cosine transform and discrimination power analysis with a face recognition technology*. *Pattern Recognition*, 43(4):1431 – 1440, 2010, ISSN 0031-3203. <http://www.sciencedirect.com/science/article/pii/S0031320309004142>.
- [18] David M. Hastings, J.D.: *Corpus Juris Secundum*. West, 2010.
- [19] Dey, Sounak, Anjan Dutta, J. Ignacio Toledo, Suman K.Ghosh, Josep Lladós y Umapada Pal: *SigNet: Convolutional Siamese Network for Writer Independent Offline Signature Verification*. arXiv:1707.02131v2, Sep 2017.

- [20] Efford, Nick: *Digital Image Processing: An Introduction using Java*. Addison-Wesley, 2000.
- [21] Ferrer-Ballester, Miguel Angel, Jesús B. Alonso y Carlos Manuel Travieso-González: *Offline geometric parameters for automatic signature verification using fixed-point arithmetic*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 27:993–997, 2005.
- [22] Fletcher, Tom: *Convolution*. <http://www.coe.utah.edu/~cs4640/slides/Lecture5.pdf>, 2017. [Online; Recuperado el 22-Diciembre-2017].
- [23] Goodfellow, Ian, Yoshua Bengio y Aaron Courville: *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.
- [24] Gurney, Kevin: *An introduction to neural networks*. UCL Press, 1997.
- [25] Hafemann, Luis G. y Luiz S. Oliveira Robert Sabourin: *Learning Features for Offline Handwritten Signature Verification using Deep Convolutional Neural Networks*. arXiv:1705.05787v1, May 2017.
- [26] Hafemann, Luis G., Robert Sabourin y Luiz S. Oliveira: *Offline Handwritten Signature Verification -Literature Review*. arXiv:1507.07909v4, Oct 2017.
- [27] Hansen, Dan Witzner: *Edges and Binary Images*. <http://slideplayer.com/slide/4593340/15/images/78/Morphology+operators+on+grayscale+images.jpg>, 2015. [Online; Recuperado el 25-Diciembre-2017].
- [28] Harris, Chris y Mike Stephens: *A combined corner and edge detector*. En *Alvey vision conference*, volumen 15, páginas 10–5244. Citeseer, 1988.
- [29] Hatkar, Pallavi V.: *Offline handwritten signature verification using Neural Network*. 2, Enero 2015.
- [30] J. Fierrez-Aguilar, N. Alonso-Hermira, G. Moreno Marquez y J. Ortega-Garcia: *An Off-line Signature Verification System Based on Fusion of Local and Global Information*”.

- [31] Kekre, H. B., V. A. Bharadi, S. Gupta, A. A. Ambardekar y V. B. Kulkarni: *Off-line signature recognition using morphological pixel variance analysis*. En *ICWET*, 2010.
- [32] Kriesel, David: *A Brief Introduction to Neural Networks*. 2007. <http://www.dkriesel.com>.
- [33] Luo, Z., J. Chen, T. Takiguchi y Y. Ariki: *Rotation-invariant histograms of oriented gradients for local patch robust representation*. En *2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, páginas 196–199, Dec 2015.
- [34] Miguel A. Ferrer, Francisco Vargas, Aythami Morales y Aaron Ordoñez: *Robustness of Off-line Signature Verification based on Gray Level Features*. *IEEE Transactions on Information Forensics and Security*, 7, Junio 2012.
- [35] Mitchell, Tom: *Machine Learning*. McGraw Hill, 1997.
- [36] Murphy, Kevin P.: *Machine learning, A Probabilistic Perspective*. MIT Press, 2012.
- [37] Pal, S., A. Alaei, U. Pal y M. Blumenstein: *Performance of an Off-Line Signature Verification Method Based on Texture Features on a Large Indic-Script Signature Dataset*. En *2016 12th IAPR Workshop on Document Analysis Systems (DAS)*, páginas 72–77, April 2016.
- [38] Pal S, Pal U, Blumenstein M: *Computational Intelligence in Digital Forensics: Forensic Investigation and Applications*. Springer International Publishing, 2014.
- [39] Pragada, S. y J. Sivaswamy: *Image Denoising Using Matched Biorthogonal Wavelets*. En *2008 Sixth Indian Conference on Computer Vision, Graphics Image Processing*, páginas 25–32, Dec 2008.
- [40] Prathiba M K, Basavaraj L: *Histogram Based on Line Signature Verification System*. *International Journal of Applied Engineering Research*, 2017.
- [41] Rafael C. Gonzalez, Richard E. Woods: *Digital Image Processing*. Prentice Hall, 2002.

- [42] Shah, Abdul Salam, Shaheed Zulfikar Ali Bhutto y Asadullah Shah: *An Appraisal of Off-line Signature Verification Techniques*. 2015.
- [43] Thakare, Bhushan y Hemant Deshmukh: *Signature Verification Techniques: State of Art Survey*. International Journal of Control Theory and Applications, 2017.
- [44] Yadav, Madhuri, Alok Kumar, Tushar Patnaik y Bhupendra Kumar: *A Survey on Offline Signature Verification*. International Journal of Engineering and Innovative Technology, 2017.
- [45] Yilmaz, Mustafa Berkay, Berrin Yanikoglu, Caglar Tirkaz y Alisher Kholmatov: *Offline signature verification using classifier combination of HOG and LBP features*, Octubre 2011.