Universidad de las Palmas de Gran Canaria

Escuela de Ingeniería de Telecomunicación y Electrónica

PROYECTO DE FIN DE CARRERA

DETECCIÓN E IDENTIFICACIÓN DE MÚLTIPLES TIPOS DE ESCRITURA EN IMÁGENES DE TEXTO

Titulación: Ingeniero de Telecomunicación

Autora: Nayara Rodríguez Rodríguez

Tutores: Dr. Miguel Ángel Ferrer Ballester

Dra. Cristina Carmona Duarte

Fecha: Junio 2015

UNIVERSIDAD DE LAS PALMAS
DE GRAN CANARIA

ESCUELA DE INGENIERÍA DE TELECOMUNICACIÓN Y ELECTRÓNICA

PROYECTO FIN DE CARRERA

DETECCIÓN E IDENTIFICACIÓN DE MÚLTIPLES TIPOS DE ESCRITURA EN IMÁGENES DE TEXTO

Autora: Nayara Rodríguez Rodríguez

Tutores: Dr. Miguel Ángel Ferrer Ballester

Dra. Cristina Carmona Duarte

Titulación: Ingeniero de Telecomunicación

Fecha: Junio 2015

UNIVERSIDAD DE LAS PALMAS
DE GRAN CANARIA

ESCUELA DE INGENIERÍA DE TELECOMUNICACIÓN Y ELECTRÓNICA

PROYECTO FIN DE CARRERA

DETECCIÓN E IDENTIFICACIÓN DE MÚLTIPLES TIPOS DE ESCRITURA EN IMÁGENES DE TEXTO

HOJA DE FIRMAS

Alumno/a:

Fdo.: Nayara Rodríguez

Tutor/a: **Tutor/a:**

Fdo.: D. Miguel A. Ferrer Fdo.: Dña. Cristina Carmona

Fecha: Junio 2015

UNIVERSIDAD DE LAS PALMAS
DE GRAN CANARIA

ESCUELA DE INGENIERÍA DE TELECOMUNICACIÓN Y ELECTRÓNICA

PROYECTO FIN DE CARRERA

DETECCIÓN E IDENTIFICACIÓN DE MÚLTIPLES TIPOS DE ESCRITURA EN IMÁGENES DE TEXTO

HOJA DE EVALUACIÓN

Calificación: _____

Presidente

Fdo.:

Vocal **Secretario/a**

Fdo.: Fdo.:

Titulación: Ingeniero de Telecomunicación

Fecha: Junio 2016

AGRADECIMIENTOS

En primer lugar, quisiera dedicar este Proyecto de Final de Carrera a mis padres, a mi hermana Keila, a mis abuelas y a mi familia en general. He conseguido llegar hasta aquí porque ustedes son los que siempre han creído en mí, porque han estado a mi lado tanto en mis triunfos como en mis fracasos, porque han sabido siempre qué necesitaba oír para continuar andando, porque nunca me han fallado. Gracias.

No me olvido de mi segunda familia, de mis telecos. Nadie mejor que nosotros sabe que para estudiar ingeniería de telecomunicaciones tienes que tener algo especial, algo distinto, pero sobre todo, tienes que ser fuerte y estar listo para enfrentarte a lo que sea. He tenido la suerte de encontrarme a grandes personas a lo largo de mi andadura universitaria. Hemos compartido risas, lágrimas, locuras, vivencias, café, apuntes a todo color y suficientes anécdotas como para escribir un libro. A todos ustedes, a los que siguen y a los que se han ido: Gracias por haberme hecho sentir siempre como en casa.

También quiero agradecer su tiempo, su paciencia, sus consejos y su apoyo a mis tutores, Miguel Ángel y Cristina, que han hecho que este último paso haya sido agradable y sencillo. Ha sido un verdadero placer trabajar con ustedes.

Y finalmente a mis amigos, a mis profesores, a los músicos que han participado conmigo en los conciertos de música clásica y moderna de la escuela, a los integrantes del equipo de mentoría de alumnos con discapacidad, a los compañeros del laboratorio y del Grupo de Procesado Digital de la Señal, y a todos los que, de una manera u otra, han estado presentes en estos últimos seis años: Gracias.

"Si no puedes volar, corre, si no puedes correr, camina, si no puedes caminar, gatea. Sin importar lo que hagas, sigue avanzado hacia adelante"

Martin Luther King Jr.

Índices

Tabla de contenido

Capítulo 1. Introducción 25

1.1. La digitalización de documentos 27

1.2. El análisis de imágenes de texto 28

1.3. El reconocimiento óptico de caracteres (OCR) 30

1.4. Alfabetos en el mundo 33

1.5. Objetivos del proyecto 37

1.6. Metodología 38

1.7. Peticionario 41

1.8. Estructura de la memoria 41

Capítulo 2. Estado del arte 45

2.1. Introducción 47

2.2. Métodos de Reconocimiento de Escritura. 47

2.3. Reconocimiento de escritura basado en la estructura 48

2.3.1. Técnicas de identificación a nivel de página 49

2.3.2. Técnicas de identificación a nivel de párrafo. 50

2.3.3. Técnicas de identificación a nivel de línea 52

2.3.4. Técnicas de identificación a nivel de palabra. 53

2.4. Reconocimiento de escritura basado en la apariencia visual. 54

2.4.1. Técnicas de identificación a nivel de página. 55

2.4.2. Técnicas de identificación a nivel de párrafo. 57

2.4.3. Técnicas de identificación a nivel de palabra. 58

2.5. Tabla resumen de resultados 59

Capítulo 3. Propuesta de clasificador 61

3.1. Introducción 63

3.2. Descripción de los parámetros de textura 63

3.2.1. Local Binary Pattern (LBP) 63

3.2.1.1. Definición 63

3.2.1.2. Programación 65

3.2.2. Orientation of Local Binary Pattern (OLBP) 65

3.2.2.1. Definición 65

3.2.2.2. Programación 67

3.2.3. Local Directional Pattern (LDP) 68

3.2.3.1. Definición 68

3.2.3.2. Programación 71

3.2.4. Local Derivative Pattern (LDerivP) 72

3.2.4.1. Definición 72

3.2.4.2. Programación 77

3.3. El clasificador LS-SVM 78

3.3.1. La Máquina de Vectores de Soporte (SVM) 78

3.3.2. La función kernel, la búsqueda de rejilla y la validación cruzada. 80

3.3.3. La Máquina de Vectores de Soporte de Mínimos Cuadrados (LS-SVM) 82

3.4. Magnitudes y curvas de medida de evaluación de los resultados obtenidos. 83

Capítulo 4. Estudio de laboratorio 87

4.1. Introducción 89

4.2. Diseño de la base de datos 89

4.3. Elaboración de la base de datos 90

4.4. Separación de líneas 93

4.5. Pruebas realizadas y resultados 96

4.5.1. Procedimiento seguido 96

4.5.2. Resultados de la experimentación 100

Capítulo 5. Estudio realista 105

5.1. Introducción 107

5.2. Diseño de la base de datos 107

5.3. Elaboración de la base de datos 110

5.4. Segmentación 112

5.4.1. Segmentación de líneas 112

5.4.2. Segmentación de palabras 115

5.5. Estadísticas y verificación 119

5.6. Pruebas realizadas y resultados 122

5.6.1. Diseño de la secuencia de entrenamiento 125

5.6.2. Procedimiento de extracción de características 126

5.6.3. Ajuste del clasificador 129

5.6.4. Estudio por documentos 129

5.6.5. Estudio por líneas 136

5.6.6. Estudio por palabras 141

Capítulo 6. Comprobaciones sobre texto manuscrito 147

6.1. Introducción 149

6.2. Diseño de la base de datos 149

6.3. Elaboración de la base de datos 150

6.4. Separación de líneas y palabras 152

6.5. Pruebas realizadas con texto manuscrito y resultados 153

6.5.1. Procedimiento seguido 153

6.5.2. Estudio por documentos 154

6.5.3. Estudio por líneas 157

6.5.4. Estudio por palabras 161

Capítulo 7. Conclusiones y líneas futuras 165

7.1. Introducción 167

7.2. Cumplimiento de objetivos y resumen del trabajo realizado 167

7.3. Conclusiones 169

7.4. Aportaciones, líneas futuras y posibles mejoras 170

Referencias bibliográficas 173

Planos y programas 181

8.1. Introducción. 183

8.2. Programas implementados. 183

8.2.1. Programas del estudio de laboratorio 183

8.2.2. Programas del estudio realista sobre texto impreso y manuscrito. 185

8.2.2.1. Base de datos 185

8.2.2.2. Extracción de características 187

Pliego de condiciones 189

9.1. Introducción 191

9.2. Pliego de condiciones técnicas 191

9.2.1. Requisitos mínimos 191

9.2.2. Instalación y ejecución del software 192

9.3. Pliego de condiciones legales 192

Presupuesto 195

10.1. Declaración Jurada 197

10.2. Desglose del presupuesto 198

10.2.1. Amortización del inmovilizado material. 198

10.2.1.1. Amortización del material hardware 199

10.2.1.2. Amortización del material software 200
10.2.2. Trabajo tarifado por tiempo empleado. 201
10.2.3. Gastos por desplazamientos realizados. 203
10.2.4. Costes del material fungible 203
10.2.5. Cuotas de redacción 204
10.3. Derechos de visado 205
10.3.1. Gastos de tramitación y envío 206
10.3.2. Presupuesto antes de impuestos 206
10.3.3. Presupuesto después de impuestos 207

Anexo I . Contenido del DVD-R 209
11.1. Introducción 211
11.2. Descripción del contenido 211

ÍNDICE DE FIGURAS

Figura 1. Jerarquía de las categorías en el procesamiento de documentos..... 30

Figura 2. Flujo de trabajo de un OCR..... 31

Figura 3. Pasos a seguir en el procesado de un documento..... 33

Figura 4. Mapa de difusión de las principales escrituras del mundo..... 34

Figura 5. Esquema con ejemplos de los distintos tipos de escritura..... 37

Figura 6. Diagrama de flujo que resume la metodología seguida en este proyecto..... 41

Figura 7. Diagrama resumen del método de Spitz para la identificación de la escritura..... 49

Figura 8. Flujo de trabajo de la primera estrategia de Chaudhuri y Sheth..... 48

Figura 9. Flujo de trabajo de la segunda estrategia de Chaudhuri y Sheth..... 51

Figura 10. Flujo de trabajo de la tercera estrategia de Chaudhuri y Sheth..... 51

Figura 11. Método de Pal y Chaudhuri para la identificación de documentos hindúes con múltiples alfabetos..... 52

Figura 12. Diagrama de bloques del procedimiento para el reconocimiento de escritura propuesto por Tan..... 56

Figura 13. Procedimientos de identificación de escritura a nivel de palabras propuestos por Dhanya et al..... 58

Figura 14. Ejemplo de cálculo del LBP..... 64

Figura 15. Ejemplo de cálculo del OLBP..... 66

Figura 16. Máscaras de la respuesta de bordes de Kirsch en ocho direcciones..... 68

Figura 17. Proceso para calcular el código binario del LDP..... 69

Figura 18. Ejemplo de cálculo del LDP..... 69

Figura 19. Respuestas de bordes antes (a) y después (b) de la rotación de la imagen..... 70

Figura 20. Ejemplo de cálculo del LDP invariante a la rotación..... 71

Figura 21. Ejemplo de vecindad de 8 píxeles alrededor de Z_0..... 73

Figura 22- 1. Ilustraciones de las ocho primeras plantillas LDerivP..... 74

Figura 22- 2. Ilustraciones de las ocho plantillas restantes LDerivP..... 75

Figura 23. Significados de los patrones denotados con "0" y "1" por el LDerivP de segundo orden..... 75

Figura 24-a. Ejemplo de cálculo del operador LDerivP de segundo orden..... 76

Figura 24-b. Aplicación de plantillas para el cálculo del operador LDerivP de segundo orden..... 77

Figura 25. Representación del funcionamiento de un SVM, que coloca un hiperplano que separa dos clases distintas..... 79

Figura 26. Diagrama de flujo para la creación de la primera base de datos del estudio inicial..... 90

Figura 27. Muestras de los distintos tipos de escritura incluidos en la primera base de datos..... 91

Figura 28. Muestras de los distintos tipos de escritura incluidos en la segunda base de datos..... 92

Figura 29. Procedimiento de segmentación de las líneas de un documento..... 94

Figura 30. Proceso para separar líneas conflictivas..... 96

Figura 31-a. Diagramas de flujo de las fases del proceso de identificación del tipo de escritura en el estudio de laboratorio..... 97

Figura 31-b. Diagramas de flujo de las fases del proceso de identificación del tipo de escritura en el estudio de laboratorio..... 97

Figura 32. Ilustración de la división de una línea en cuatro bloques y el cálculo de sus respectivos histogramas LBP..... 98

Figura 33. Ejemplo de los histogramas LBP, OLBP, LDP y LDerivP de una línea..... 99

Figura 34. Curvas CMC obtenidas entrenando con el 30% de la primera base de datos y realizando el test con las imágenes restantes..... 101

Figura 35. Curvas CMC obtenidas entrenando con el 30% de la primera base de datos y realizando el test con la segunda base de datos..... 103

Figura 36. Documentos físicos digitalizados para formar parte de la base de datos..... 108

Figura 37. Ejemplos de muestras de la base de datos a tres niveles: documento, línea y palabra escritos en Oriya..... 109

Figura 38. Diagrama de bloques del proceso de elaboración de la base de datos de documentos..... 110

Figura 39. Diagrama de bloques del proceso de elaboración de la base de datos de líneas y palabras..... 111

Figura 40. Proceso seguido para la segmentación de líneas..... 113

Figura 41. Ejemplos de documentos que necesitan ser segmentados manualmente..... 114

Figura 42. Diagrama de flujo del proceso de segmentación de líneas..... 115

Figura 43. Ejemplos de líneas en alfabeto latino, tailandés, árabe, malayalam y japonés respectivamente..... 115

Figura 44. Diagrama de bloques que resume el proceso de segmentación de palabras del primer sistema desarrollado..... 116

Figura 45-a. Línea escrita en tailandés con separadores entre palabra y palabra..... 117

Figura 45-b. Línea escrita en tailandés con separadores dividiendo la primera palabra de la línea..... 118

Figura 45-c. Línea escrita en tailandés con separadores dividiendo la segunda palabra de la línea..... 118

Figura 45-d. Línea escrita en tailandés con separadores dividiendo la tercera palabra de la línea..... 118

Figura 45-e. Línea escrita en tailandés con separadores dividiendo la tercera palabra de la línea..... 119

Figura 46. Esquema general del funcionamiento del sistema de reconocimiento del tipo de escritura..... 124

Figura 47. Pasos seguidos en la selección de las muestras dedicadas a entrenamiento y test..... 126

Figura 48. Pasos seguidos en la extracción de características de las imágenes de la base de datos..... 128

Figura 49. Procedimiento de división de documentos en bloques de 128x128 píxeles solapados un 15%..... 130

Figura 50. Líneas en Hindi y Gurmukhi donde se aprecia su similitud visual..... 135

Figura 51-a. Distribución de la base de datos de texto impreso..... 150

Figura 51-b. Distribución de la base de datos de texto manuscrito..... 150

Figura 52. Fragmentos de documentos físicos digitalizados para formar parte de la base de datos de texto manuscrito..... 151

Figura 53. Diagrama de bloques del proceso de elaboración de la base de datos de texto manuscrito..... 151

Figura 54. Fragmento de documento escrito a mano en bengalí..... 152

Figura 55. Fragmento de documento escrito a mano en bengalí segmentado trazando líneas a mano alzada..... 153

Figura 56. Fragmentos de documentos manuscritos redactados en bengalí con inclinaciones, anchos de trazo y caligrafías distintas..... 157

Figura 57. Diagrama de flujo del programa principal del estudio realista para texto impreso y manuscrito..... 157

ÍNDICE DE TABLAS

Tabla 1. Resumen de resultados de métodos para identificar la escritura..... 59

Tabla 2. Contenido de la primera base de datos del estudio de laboratorio..... 92

Tabla 3. Contenido de la primera base de datos del estudio de laboratorio..... 93

Tabla 4. Tasa de identificación de escritura variando el porcentaje de la base de datos destinado a entrenamiento..... 100

Tabla 5. Resultados obtenidos entrenando con el 30% de la primera base de datos y realizando el test con las imágenes restantes..... 101

Tabla 6. Resultados obtenidos entrenando con el 30% de la primera base de datos y realizando el test con la segunda base de datos..... 103

Tabla 7. Contenido de la base de datos realizada durante el estudio realista a partir de documentos físicos..... 120

Tabla 8. Contenido de la base de datos de palabras destinadas al proceso de test..... 121

Tabla 9. Experimentos previstos para identificar la escritura en texto impreso..... 123

Tabla 10. Resultados de los experimentos sobre texto impreso a nivel de documentos aplicando técnicas de análisis de texturas..... 131

Tabla 11-a. Resultados de los experimentos sobre texto impreso a nivel de documentos aplicando el LBP junto con otro de los patrones locales..... 132

Tabla 11-b. Resultados de los experimentos sobre texto impreso a nivel de documentos aplicando las combinaciones restantes usando dos patrones locales..... 132

Tabla 12. Resultados de los experimentos sobre texto impreso a nivel de documentos usando todas las combinaciones posibles con tres patrones locales..... 134

Tabla 13. Resultados de los experimentos sobre texto impreso a nivel de documentos combinando los efectos de todos los patrones locales..... 134

Tabla 14. Resultados de los experimentos sobre texto impreso a nivel de líneas aplicando técnicas de análisis de texturas..... 136

Tabla 15-a. Resultados de los experimentos sobre texto impreso a nivel de líneas aplicando el LBP junto con otro de los patrones locales..... 137

Tabla 15-b. Resultados de los experimentos sobre texto impreso a nivel de líneas aplicando las combinaciones restantes usando dos patrones locales..... 138

Tabla 16. Resultados de los experimentos sobre texto impreso a nivel de líneas usando todas las combinaciones posibles con tres patrones locales..... 139

Tabla 17. Resultados de los experimentos sobre texto impreso a nivel de líneas combinando los efectos de todos los patrones locales..... 140

Tabla 18. Resultados de los experimentos sobre texto impreso a nivel de palabras aplicando técnicas de análisis de texturas..... 142

Tabla 19-a. Resultados de los experimentos sobre texto impreso a nivel de palabras aplicando el LBP junto con otro de los patrones locales..... 142

Tabla 19-b. Resultados de los experimentos sobre texto impreso a nivel de palabras aplicando las combinaciones restantes usando dos patrones locales..... 143

Tabla 20. Resultados de los experimentos sobre texto impreso a nivel de palabras usando todas las combinaciones posibles con tres patrones locales..... 144

Tabla 21. Resultados de los experimentos sobre texto impreso a nivel de palabras combinando los efectos de todos los patrones locales..... 145

Tabla 22. Contenido de la base de datos manuscrita..... 152

Tabla 23. Experimentos previstos para identificar la escritura en texto manuscrito..... 154

Tabla 24. Resultados de los experimentos sobre texto manuscrito a nivel de documentos aplicando técnicas de análisis de texturas..... 155

Tabla 25-a. Resultados de los experimentos sobre texto manuscrito a nivel de documentos aplicando el LBP junto con otro de los patrones locales..... 155

Tabla 25-b. Resultados de los experimentos sobre texto manuscrito a nivel de documentos aplicando combinaciones restantes usando dos patrones locales..... 155

Tabla 26. Resultados de los experimentos sobre texto manuscrito a nivel de documentos usando todas las combinaciones posibles con tres patrones locales..... 156

Tabla 27. Resultados de los experimentos sobre texto manuscrito a nivel de documentos combinando los efectos de todos los patrones..... 156

Tabla 28. Resultados de los experimentos sobre texto manuscrito a nivel de líneas aplicando técnicas de análisis de texturas..... 157

Tabla 29-a. Resultados de los experimentos sobre texto manuscrito a nivel de líneas aplicando el LBP junto con otro de los patrones locales..... 158

Tabla 29-b Resultados de los experimentos sobre texto manuscrito a nivel de líneas aplicando las combinaciones restantes usando dos patrones locales..... 158

Tabla 30. Resultados de los experimentos sobre texto manuscrito a nivel de líneas usando todas las combinaciones posibles con tres patrones locales..... 158

Tabla 31. Resultados de los experimentos sobre texto manuscrito a nivel de líneas combinando los efectos de todos los patrones locales..... 159

Tabla 32. Resultados de los experimentos sobre texto manuscrito a nivel de líneas aplicando técnicas de análisis de texturas y dos porcentajes de entrenamiento distintos..... 159

Tabla 33-a. Resultados de los experimentos sobre texto manuscrito a nivel de líneas aplicando el LBP junto con otro de los patrones locales y dos porcentajes de entrenamiento distintos..... 160

Tabla 33-b. Resultados de los experimentos sobre texto manuscrito a nivel de líneas aplicando las combinaciones restantes usando dos patrones locales y dos porcentajes de entrenamiento distintos..... 160

Tabla 34. Resultados de los experimentos sobre texto manuscrito a nivel de líneas usando todas las combinaciones posibles con tres patrones locales y dos porcentajes de entrenamiento distintos..... 160

Tabla 35. Resultados de los experimentos sobre texto manuscrito a nivel de líneas combinando los efectos de todos los patrones locales y usando dos porcentajes de entrenamiento distintos..... 160

Tabla 36. Resultados de los experimentos sobre texto manuscrito a nivel de palabras aplicando técnicas de análisis de texturas..... 161

Tabla 37-a. Resultados de los experimentos sobre texto manuscrito a nivel de palabras aplicando el LBP junto con otro de los patrones locales..... 161

Tabla 37-b. Resultados de los experimentos sobre texto manuscrito a nivel de palabras aplicando las combinaciones restantes usando dos patrones locales..... 162

Tabla 38. Resultados de los experimentos sobre texto manuscrito a nivel de líneas usando todas las combinaciones posibles con tres patrones locales..... 162

Tabla 39. Resultados de los experimentos sobre texto manuscrito a nivel de palabras combinando los efectos de todos los patrones locales..... 162

Tabla 40. Precios y coste de la amortización del hardware..... 200

Tabla 41. Precios y coste de la amortización del software..... 200

Tabla 42. Factor de corrección del COIT según las horas trabajadas..... 202

Tabla 43. Honorarios del proyecto según tiempo y fase del mismo..... 203

Tabla 44. Costes del material fungible del proyecto..... 204

Tabla 45. Presupuesto P para el cálculo del coste de redacción del proyecto..... 204

Tabla 46. Presupuesto P para el cálculo del coste del visado del proyecto..... 205

Tabla 47. Presupuesto sin impuestos..... 207

Tabla 48. Presupuesto con impuestos..... 207

ÍNDICE DE ECUACIONES

Ec. 1. Expresión del operador LBP 64
Ec. 2. Función signo 64
Ec. 3. Primer paso cálculo OLBP 66
Ec. 4. Expresión del operador OLBP original 66
Ec. 5. Expresión del operador OLBP usada en este proyecto 66
Ec. 6. Expresión del operador LDP 69
Ec. 7. Valor de los bits para el cálculo del LDP 69
Ec. 8. Expresión del LDP invariante a la rotación 71
Ec. 9. Valor de la derivada a 0º 73
Ec. 10. Valor de la derivada a 45º 73
Ec. 11. Valor de la derivada a 90º 73
Ec. 12. Valor de la derivada a 135º 73
Ec. 13. Expresión del operador LDerivP direccional de segundo orden 73
Ec. 14. Función de codificación binaria para el cálculo del LDerivP 73
Ec. 15. Expresión del operador LDerivP de segundo orden 74
Ec. 16. Expresión del kernel RBF (1) 81
Ec. 17. Expresión del kernel RBF (2) 81
Ec. 18. Fórmula de cálculo lineal de cuota de amortización de inmovilizado material 199
Ec. 19. Honorario por tiempo trabajado en proyecto de telecomunicación 201
Ec. 20. Honorario por tiempo trabajado en este proyecto 202
Ec. 21. Gastos por desplazamiento en este proyecto 203
Ec. 22. Costes de redacción de un proyecto de telecomunicación 204
Ec. 23. Costes de redacción de este proyecto 205
Ec. 24. Derechos de visado de un proyecto de telecomunicación 205
Ec. 25. Derechos de visado de este proyecto 206

Capítulo 1. Introducción

1.1. La digitalización de documentos

El siglo XXI se caracteriza por el avance y expansión de la digitalización y el control de la información a nivel global. La tecnología se ha convertido en una pieza clave de nuestro día a día, aparatos electrónicos como nuestro móvil o nuestro ordenador se han vuelto compañeros casi imprescindibles para hacer frente a nuestra rutina diaria.

Se dice que vivimos en la era de la información. El mundo entero está conectado gracias al avance y el fácil acceso a Internet. Acciones como hablar con personas que se encuentran en el otro lado del planeta o acceder a bibliotecas que se localizan a cientos de kilómetros desde nuestra propia casa, son acontecimientos que forman parte de nuestra vida cotidiana, a pesar de que hace tan solo unas décadas podrían haber sido parte de la trama de una novela de ciencia ficción.

Este proyecto se centra en unos de los problemas más interesantes en lo relativo al salto a lo digital que estamos viviendo actualmente. En concreto, trataremos la transformación de los textos, al paso del papel al formato digital.

Se denomina *digitalización de documentos* al proceso por el cual un documento en soporte papel se transforma en un documento en un soporte lógico, accesible desde un ordenador o cualquier plataforma digital como puede ser un teléfono móvil [1].

La digitalización de documentos y archivos tiene múltiples ventajas [2], algunas de ellas son las siguientes:

- La reducción del espacio físico que ocupa la documentación en papel.
- Permite la disponibilidad para la consulta de ejemplares en todo momento y en cualquier lugar o incluso, la posibilidad de permitir su acceso únicamente a ciertas personas autorizadas, aumentando así el nivel de seguridad, lo cual es útil, por ejemplo, en el caso de la documentación judicial.
- Todo documento digital se encuentra disponible de forma simultánea para varias personas a la vez, sin necesidad de copias físicas. Esto también posibilita la distribución de parte del contenido o la totalidad del mismo por correo electrónico u otros sistemas de difusión modernos, de manera inmediata y sin tener que enviar las mencionadas copias físicas ni esperar su recepción.
- La digitalización de archivos también posibilita que estos sean más fáciles de localizar y estén mejor ordenados. Esto a su vez, evita la pérdida de gran parte de los documentos y archivos que suele producirse con el paso del tiempo.

- Otro aspecto positivo destacable es que los documentos digitales sirven de copia de seguridad de los originales, de tal forma que se evita el desgaste de éstos, su robo o pérdida en caso de incendio, vandalismo, inundación, etc. como ya ha ocurrido con multitud de documentos históricos.

En conclusión, la digitalización de documentos y archivos tiene muchos beneficios y es un proceso que resulta imprescindible llevar a cabo, dado el auge de las tecnologías de la información y la comunicación, así como la necesidad de conservación y acceso a documentación que hasta hoy se encuentra en papel.

Centrémonos en el caso de textos históricos [3]. En España tenemos un gran patrimonio cultural del que no somos conscientes y que se considera uno de los más valiosos del mundo por su volumen y por la cantidad de información contenida en sus documentos, que actualmente están siendo digitalizados. Contamos por ejemplo, con el Archivo de Indias, que es una colección única que ha sido declarada Patrimonio de la Humanidad, y también podemos destacar el Archivo General de la Administración, el Archivo de la Corona de Aragón o el Archivo de Simancas entre muchos otros, todos ellos textos de gran valor histórico que es importante conservar.

Sin duda, la digitalización de documentos de carácter histórico nos ayuda a acercarnos a la historia y a entenderla mejor, pero estos no son los únicos archivos cuya digitalización es interesante. El cambio de formato de los documentos jurídicos, administrativos y sanitarios conlleva un mejor funcionamiento de los procedimientos en las instituciones y supone además un gran ahorro en cuanto a espacio y coste, e incluso en términos medioambientales.

1.2. El análisis de imágenes de texto

La consecuencia directa de la digitalización de documentos es la existencia de una gran demanda de software para proceder a la extracción, análisis y almacenamiento de la información de los documentos físicos para su posterior acceso en entornos digitales para su consulta, o incluso para su edición, con altas exigencias de automatización, eficiencia y eficacia.

Todos estos procesos son parte de lo que se conoce como *análisis de imágenes de texto*, que es un campo de estudio que ha tenido un rápido crecimiento y ha suscitado un gran interés dentro de las investigaciones científicas de las últimas décadas. Siendo el objetivo

del análisis de imágenes de documentos reconocer los componentes textuales y gráficos contenidos en imágenes de texto, y extraer la información que sea precisa, como lo haría un ser humano [4].

Esta disciplina no trabaja únicamente en la digitalización de documentos propiamente dicha. En la actualidad, se están llevando a cabo muchas investigaciones para el desarrollo de técnicas con otros propósitos, como pueden ser: establecer la autenticidad o falsedad de un documento mediante la detección de ciertas características del papel como la antigüedad de la tinta; o analizar el texto de un escrito para determinar si la caligrafía del mismo es una imitación de la original.

Dicho esto, para efectuar el análisis de una muestra de texto es necesario seguir una serie de pasos:

1. En primer lugar, se debe escanear el documento a analizar. Esta tarea sólo ha de realizarse una vez, quedando a partir de entonces disponible una imagen de éste, en el ordenador o en la plataforma deseada.
2. Después, se asigna un nombre de archivo y se crean una serie de datos que se relacionan con el tipo de contenido. Estos datos pueden asignarse de forma manual o automática a partir de la información contenida en ciertas áreas de dicho documento.
3. Finalmente, la imagen obtenida de los puntos anteriores pasa a ser procesada mediante técnicas de procesamiento para analizar determinados aspectos sobre el documento, como puede ser el tipo de escritura del mismo, que es el tema principal del presente proyecto.

Si centramos nuestra atención en esta última fase, encontramos dos categorías [5] en las que podemos englobar las distintas técnicas involucradas en el análisis de una imagen de texto (ver Figura 1):

- El procesado gráfico se encarga de los componentes no textuales y simbólicos que conforman diagramas de líneas, delimitando líneas rectas entre secciones de texto, logotipos de empresas, etc.
- Las técnicas de procesado textual por su parte, se ocupan de analizar los componentes textuales de la imagen. Las tareas que se ejecutan en este caso son:
    - La determinación de la inclinación del documento, puesto que resulta importante detectar si existen defectos en la dirección del texto como consecuencia de la fase de escaneado, para llevar a cabo estudios posteriores de las características del escrito.

- La búsqueda de columnas, párrafos, líneas de texto y palabras.
- Y el reconocimiento del texto y sus atributos tales como tamaño, tipo de letra, etc. mediante técnicas software como el reconocimiento óptico de caracteres (OCR).

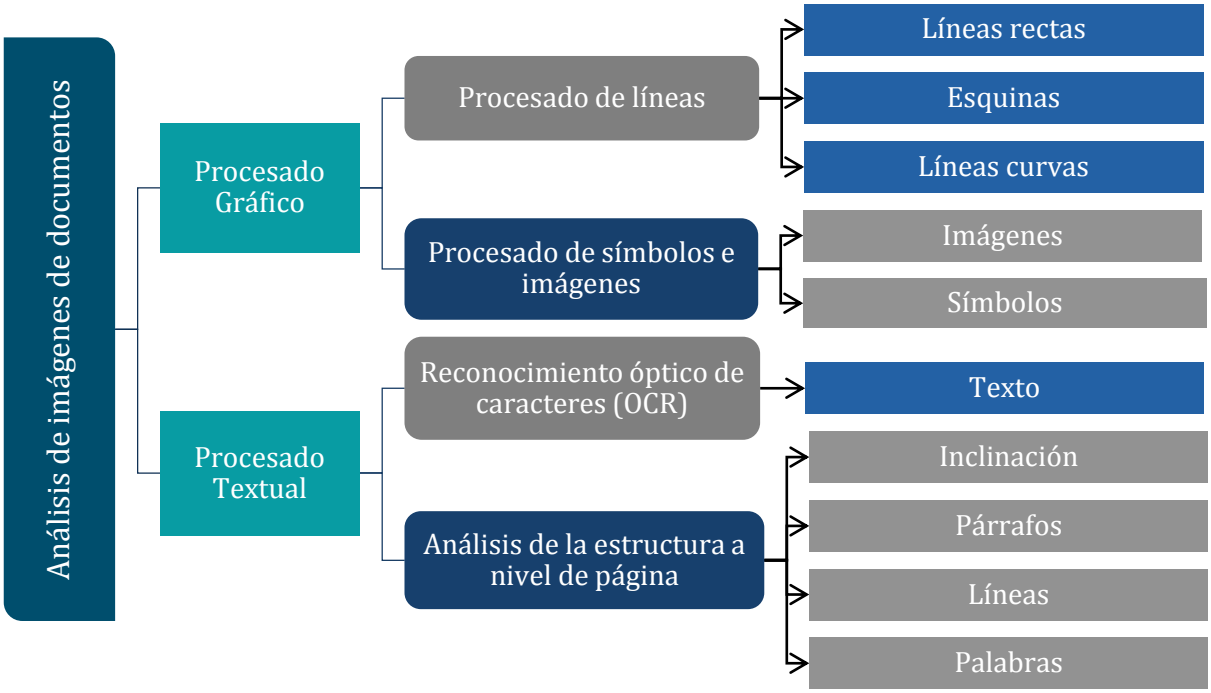


Figura 1. Jerarquía de las categorías en el procesamiento de documentos.

1.3. El reconocimiento óptico de caracteres (OCR)

El *reconocimiento óptico de caracteres u OCR* [6, 7], es una herramienta fundamental en el campo de análisis de documentos. Esta tecnología pretende funcionar de forma similar al ojo humano a la hora de reconocer objetos, siendo su objetivo digitalizar documentos al escanear los caracteres contenidos en una imagen de texto, de manera que estos se vuelven reconocibles para nuestro ordenador o para la plataforma digital que queramos emplear.

La mayoría de los sistemas OCR incluyen un escáner óptico para la lectura de texto y un sofisticado software de análisis de imágenes. Empleando una combinación de hardware (tarjetas de circuitos especializados) y software para reconocer los caracteres, aunque algunos sistemas de bajo coste hacen todo únicamente a través de software.

Cuando se pretende emplear un OCR para la digitalización de un documento, se comienza escaneando el mismo, luego se somete al proceso de reconocimiento, se capturan los datos en formato de texto y finalmente se procede a la verificación de dichos datos antes de exportar el documento digital final. Este proceso se encuentra representado en la figura siguiente.

Figura 2. Flujo de trabajo de un OCR.

Por otro lado, la secuencia de ejecución del OCR conforma varias tareas:

1. En primer lugar, se inicia el análisis de la estructura de la imagen o el documento escaneado, dividiendo cada página en diferentes elementos, como imágenes, tablas y bloques de textos. Una vez se descartan los elementos no textuales se procesa el texto, tal que cada línea se divide en palabras y cada palabra se divide en letras.
2. Una vez hecho esto, el software inspecciona la imagen píxel a píxel para reconocer los caracteres que constituyen el texto, buscando formas que coincidan con los patrones que incorpora el OCR.
3. En función del nivel de complejidad o grado de desarrollo del software, éste buscará coincidencias con los caracteres y fuentes disponibles en el programa, o tratará de identificar los caracteres a través del análisis de sus características.
4. Cuando se han detectado todos los elementos del texto, el programa los compara con un patrón compuesto por un conjunto de imágenes (de hecho, lo habitual es que los OCR incluyan diccionarios en varios idiomas que permitan que esta tarea se lleve a cabo).
5. Finalmente, y después de procesar una gran cantidad de hipótesis probabilísticas, el programa toma una decisión y muestra un texto editable, que se ha de someter a un proceso de verificación antes de exportar los datos finales en el formato digital deseado.

Los sistemas de reconocimiento óptico de caracteres son muy usuales en la actualidad y ello se debe a que su uso tiene multitud de ventajas:

- En primer lugar, permiten una mejor búsqueda y recuperación de documentos, puesto que el software ofrece la posibilidad de buscar texto dentro del documento completo. Hecho que es posible gracias al proceso de creación de metadatos, el cual genera un índice de palabras clave que son reconocidas de manera automática.
- También abren las puertas a la explotación de otros usos del documento puesto que los textos digitalizados se vuelven editables y ello hace que puedan ser sometidos a otros procesos. Por ejemplo, si usamos más programas tras el OCR, podemos transformar el texto de un documento a líneas de braille o archivos de audio, lo cual facilita el acceso a la información de personas invidentes.
- Es destacable mencionar el ahorro en términos de memoria que supone el uso de un OCR, y es que el espacio de almacenamiento requerido por un archivo en formato de texto es mucho menor al requerido por una imagen.

Sin embargo, los OCR tienen una serie de problemas aún por resolver a día de hoy. Este proyecto se centra en uno de los principales inconvenientes de estos sistemas, que consiste en que su funcionamiento es dependiente del lenguaje que procesen, es decir, es necesario saber en qué idioma está escrito el texto de análisis de antemano para poder emplear el OCR más adecuado. De forma genérica, el programa supondrá que el documento está redactado en un único tipo de escritura y realizará el reconocimiento acorde a esta hipótesis.

Consecuentemente, un OCR resulta poco eficaz si se desea analizar documentos que contengan dos o más tipos de escritura. Esto ocurre frecuentemente en países con varios idiomas oficiales como puede ser la India, que es un territorio con un gran número de lenguajes y escrituras que coexisten en toda su geografía. Así, un documento puede contener texto en inglés, en hindi y en otra lengua oficial de la región de donde proceda dicho archivo, como puede ser bengalí, persa o canarés.

En efecto, la identificación del tipo de escritura es un desafío relevante en el campo del análisis de documentos, porque se trata de un procedimiento necesario si se desea que una imagen de texto sea procesada por los OCR más adecuados.

La solución a esta problemática pasa por el procesado de cada palabra del texto que compone el documento antes de someterlo a las técnicas requeridas por el OCR. Por consiguiente, si se quiere conocer los tipos de escritura contenidos en un determinado documento de forma precisa para posteriormente analizarlo con un sistema OCR, los pasos a seguir son los siguientes (ver Figura 3):

1. *Segmentación del documento en líneas y palabras:*

   En primer lugar se debe separar el documento en líneas y posteriormente en palabras, para poder analizar todas y cada una de las estructuras de texto de la imagen y proceder a su identificación individualmente. De hecho, lo habitual es encontrar textos redactados mayoritariamente en un tipo de escritura determinado, que incluyen además palabras en otro tipo de escritura, motivo por el que es importante poder trabajar a nivel de palabra.

2. *Identificación del tipo de escritura de cada palabra:*

   Se pondrán en práctica las técnicas que procedan para saber a qué tipo de escritura corresponde cada una de las unidades de texto anteriormente separadas. En próximos capítulos se explicarán los métodos propuestos en este Proyecto de Final de Carrera para cumplir con este objetivo.

3. *Elección del OCR más adecuado en cada caso:*

   Una vez sepamos qué tipos de escritos contiene nuestro documento, será más sencillo decidir qué OCR es más apropiado en cada caso para, finalmente, digitalizar el texto del mismo siguiendo los pasos vistos con anterioridad.

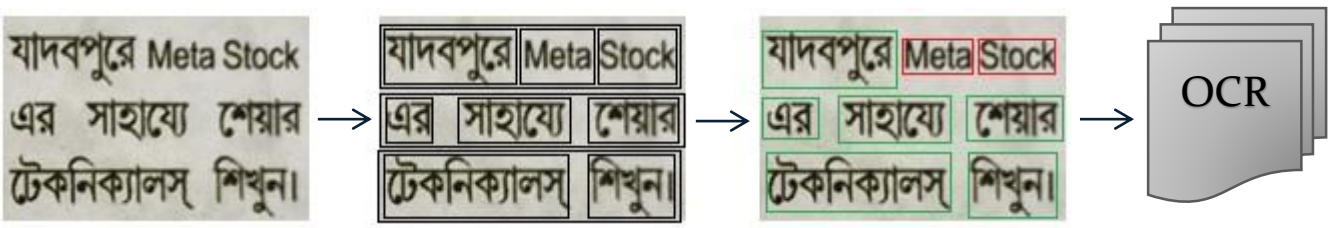


Figura 3. Pasos a seguir en el procesado de un documento.

1.4. Alfabetos en el mundo

Un *sistema de escritura* [8, 9] es un tipo de sistema simbólico usado para representar elementos o declaraciones expresables en el lenguaje. La mayoría de nosotros trabajamos con el alfabeto latino, pero la tipografía no conoce fronteras, y existen decenas de escrituras menos conocidas en todo el mundo con una estética y una estructura distintas.

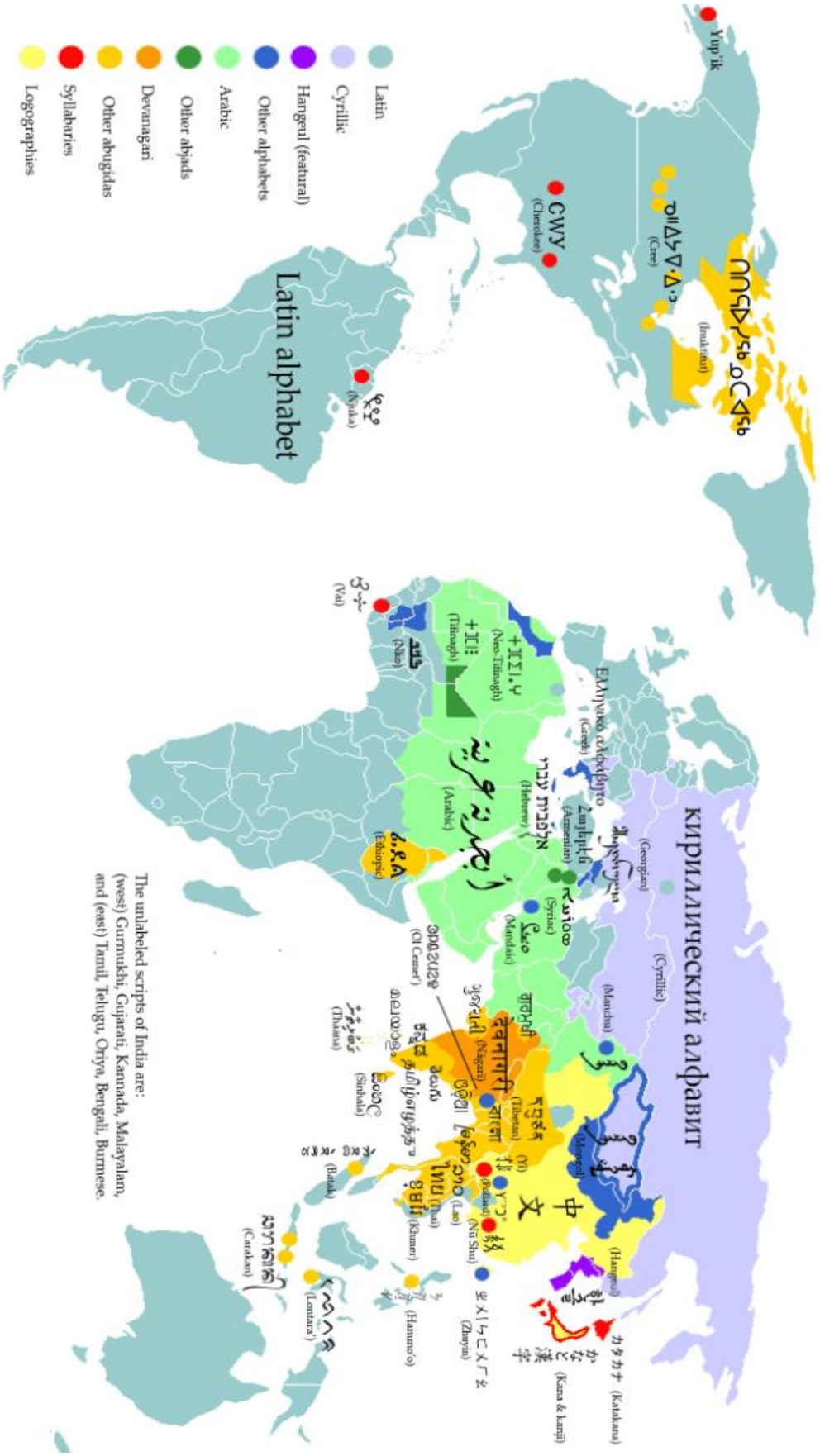


Figura 4. Mapa de difusión de las principales escrituras del mundo.

Centrando nuestra atención en la Figura 4, se observa que el alfabeto latino se usa en la mayoría de los territorios del mundo pero también existen varias áreas geográficas que engloban una gran cantidad de escrituras distintas, como puede verificarse en la zona del mapa correspondiente a Asia.

En concreto, podemos ver que la India es la región con más escrituras distintas a lo largo y ancho de su geografía [10]. En este país existen 18 lenguas oficiales que coexisten con otras muchas no reconocidas oficialmente. De hecho, las escuelas de la India enseñan 58 idiomas diferentes, se publican periódicos en 87 idiomas, se emiten programas de radio en 71 y se estrenan películas en 15 distintos. Muchas de las escrituras analizadas en este proyecto se emplean en distintos lugares de este país: los alfabetos hindi o devanagari, bengalí, canarés, gujarati, gurmukhi, malayalam, oriya, urdu, tamil y telugu.

Asimismo, los principales sistemas de escritura del mundo pueden ser clasificados en cuatro grandes categorías:

- **Escrituras alfabéticas:**

Un alfabeto es un sistema de escritura segmentado en el cual un grupo estandarizado de letras (grafemas) tiene un equivalente fonético. La palabra alfabeto deriva "*alpha*" y "*beta*", las primeras dos letras del alfabeto griego y es aquí donde podemos encontrar las escrituras griega, rusa o latina, las cuales también han sido objeto de análisis de este trabajo.

- **Abyad:**

Un abyad o abjad es un sistema de escritura alfabético donde hay un símbolo por consonante, por lo que también se denominan alfabetos consonánticos.

Éstos se diferencian de otros alfabetos en que tienen caracteres sólo para sonidos consonantes, mientras que las vocales no son frecuentes, aunque sí es cierto que en algunos casos éstas son escritas en forma de marcas diacríticas.

Pertenecen a este grupo la escritura árabe o la persa.

- **Abugida:**

Los abugidas (a veces abúgidas), también llamados alfasilabarios o alfabetos silábicos, son sistemas de escritura que cuentan con caracteres vocálicos y consonánticos. A diferencia del alfabeto o de los abyads, las consonantes llevan una vocal inherente, generalmente 'a'. Para cambiar esta combinación se recurre a los signos diacríticos que denotan cambio de vocal o ausencia de ésta.

En muchos abugidas la modificación es la adición de un signo vocal, pero existen muchas otras posibilidades, tales como la rotación del signo básico, la adición de un signo diacrítico, etc.

Las escrituras de la zona sur de Asia como son el devanagari, el bengalí o el gujarati que ya mencionamos con anterioridad, entre otras muchas escrituras propias del subcontinente hindú, así como el tailandés o el lao se incluyen dentro de esta categoría.

- **Logográficas**

En un sistema logográfico, cada carácter (logograma) representa una sola palabra gramatical y es más precisa que un morfema.

El sistema logográfico de mayor relevancia es el chino, que comprende alrededor de 50.000 caracteres.

- **Escrituras silábicas:**

Como su propio nombre indica, en un sistema silábico cada carácter representa o aproxima sílabas completas, tal que la unión de estos símbolos compone las palabras.

Así, un símbolo representa normalmente una consonante seguida de una vocal, o una única vocal, como ocurre por ejemplo en el japonés, que sería parte de este grupo.

- **Característicos**

Un sistema de escritura característico representa detalles más concretos que un alfabeto. Esto significa que cada símbolo representa no un fonema completo, sino que se refiere a ciertas características fonéticas que componen un fonema, como pronunciación y lugares de articulación. Este es el caso del coreano.

Sin embargo, muchos sistemas de escritura no pueden ser clasificados en un único tipo, por ejemplo, el japonés emplea elementos de la escritura china por lo que podría considerarse como parte de la categoría logográfica pero a su vez cada símbolo chino se asocia con una sílaba por lo que también podría considerarse una escritura silábica.

Otro hecho a destacar es que existen países con más de un idioma oficial y en ocasiones, estos tienen escrituras completamente distintas. En esta línea, también hay que tener cuenta que el alfabeto latino al estar tan extendido, siendo el método de expresión del idioma inglés que es el segundo con más hablantes en el mundo, ha sido acogido en muchos lugares donde la estructura de la escritura oficial no coincide con este alfabeto, llegando a mezclarse ambos en algunos documentos.

Por todo esto, resulta necesaria la creación de técnicas que permitan facilitar la identificación del tipo de escritura dentro de un documento redactado en más de un idioma y con más de un alfabeto distinto. Con este avance podrían desarrollarse sistemas más eficientes de digitalización de documentos y archivos, además de abrir las puertas a nuevas aplicaciones relacionadas con el análisis de documentos como pueden ser la conversión de texto escrito a texto audible o la traducción automática de un escrito dado.

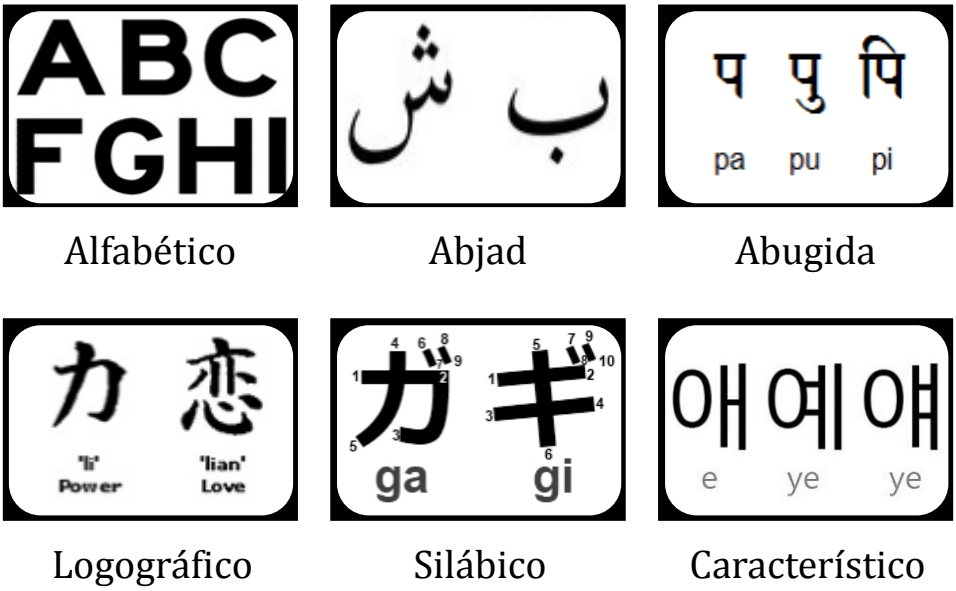


Figura 5. Esquema con ejemplos de los distintos tipos de sistemas de escritura.

1.5. Objetivos del proyecto

El principal objetivo de este Proyecto de Fin de Carrera es desarrollar un sistema que permita identificar los distintos tipos de escritura contenidos en la imagen de un documento.

Por otro lado, uno de los objetivos más importantes a cumplir con el presente proyecto es el de estudiar la aplicación de los descriptores empleados en procesos de análisis de texturas en reconocimiento de escritura. En concreto, se estudió la viabilidad de los *LBP* (*Local Binary Pattern*), los OLBP (*Ortogonal Local Binary Pattern*), los LDerivP (*Local Derivative Pattern*) y los *LDP* (*Local Directional Pattern*) que se explicarán en puntos posteriores de esta memoria. También se pretende saber si existe o no una mejora usando estos descriptores frente a los más usuales como puede ser el filtro de Gabor.

Otro objetivo a cumplir en este proyecto es comprobar si las técnicas que proponemos son válidas, no sólo para texto impreso sino también para la identificación del tipo de escritura de texto manuscrito, ya que no existen, a día de hoy, sistemas que sean capaces de llevar esto a cabo de forma eficaz y con un porcentaje de acierto aceptable.

Para completar la información anterior y lograr la consecución de los propósitos mencionados, ha sido necesario alcanzar otra serie de objetivos secundarios o específicos durante la realización del proyecto, como son: analizar y estudiar la documentación de referencia, escanear distintos documentos para confeccionar una base de datos , organizar dicha base de datos o indagar sobre técnicas de procesado de la imagen a fin de mejorar la calidad de las mismas, antes de que estas fueran procesadas para su identificación mediante algoritmos de análisis de texturas.

1.6. Metodología

El primer paso en la realización de este Proyecto de Fin de Carrera fue la creación de una primera base de datos para corroborar si los parámetros de análisis de texturas podían ser aplicados también para la identificación de tipos de escritura.

Con el objeto de confeccionar esta base de datos inicial, se tomaron muestras seleccionando fragmentos de novelas famosas, artículos periodísticos, etc. en inglés y en español. Seguidamente, se tradujeron a hindi, gujarati, japonés, canarés, bengalí, tamil, telugu, griego, ruso y árabe mediante el traductor de Google. Se guardó la imagen del cuadro de resultados de la traducción, haciendo una captura de pantalla y guardando la imagen del texto.

El siguiente paso consistió en confeccionar un programa que extrajera los resultados de pasar las imágenes que componían la base de datos, por cada uno de los filtros correspondientes a los parámetros de análisis de texturas.

Se podría decir que, desde un punto de vista visual, un texto es un conjunto de texturas y patrones que forman un escrito. Por este motivo, en este proyecto se proponen algoritmos basados en análisis de texturas para la identificación del tipo de escritura. Asimismo, los patrones locales son los descriptores empleados de forma extensa y con mejores resultados en la disciplina del análisis de texturas. Por consiguiente, en este proyecto se ha empleado el histograma de los patrones locales como una descripción de la distribución de las distintas direcciones en el trazo de cada escritura, que es una característica propia de cada tipo que hace que se diferencie claramente del resto.

Los parámetros escogidos para realizar la identificación, fueron la versión básica de los LBP o *Local Binary Patterns* y tres modificaciones del mismo: los OLBP, LDP y LDerivP. Los OLBP u *Orientation of Local Binary Patterns* representan de forma más explícita la información sobre la orientación de los trazos; los LDP o *Local Directional Patterns*, son similares a los anteriores con la diferencia de que producen un patrón mucho más estable en presencia de ruido y se centran en las direcciones más predominantes; y finalmente, los LDerivP o *Local Derivative Patterns*, son menos sensibles que los LDP al ruido aleatorio y estudian las variaciones de dirección con un orden elevado.

Así, el programa para obtener los resultados de aplicar estas técnicas consta de tres fases diferenciadas:

- En primer lugar, se procede a la lectura y adecuación de la imagen del documento con el fin de aclarar y mejorar el objeto de análisis, que son los trazos del texto.
- El segundo paso consiste en dividir el documento en líneas usando la envolvente convexa.
- En el último paso, se toma cada línea obtenida a partir del procedimiento anterior, se divide en cuatro franjas horizontales y se pasa cada una de estas porciones por los algoritmos de análisis de textura que se citaron con anterioridad, obteniendo finalmente una serie de descriptores por cada línea que se almacenan en un vector.

Una vez obtenidos los vectores de parámetros, estos son introducidos en un clasificador SVM, para así comprobar si nuestro sistema inicial es capaz de diferenciar los distintos tipos de escritura que habíamos incluido.

Como consecuencia del éxito de la fase inicial, se decidió crear una base de datos más extensa, compleja y realista. Para cumplir con este propósito, se nos facilitó una serie de documentos desde el *Indian Statistical Institute* y concretamente, la *Computer Vision and Pattern Recognition Unit*.

Se escanearon distintos fragmentos de periódicos y libros con alfabetos, fuentes y estilos distintos, los cuales fueron nombrados y organizados manualmente. Las imágenes de texto obtenidas fueron segmentadas, primero en líneas y luego en palabras. Para ello, se desarrollaron una serie de programas que llevaran a cabo estas separaciones de forma semiautomática, por la complejidad que ello suponía. Cabe mencionar que decidimos implementar una solución que permitiera la interacción del usuario, para así realizar la división en líneas y palabras correctamente, verificando además que el resultado final de este proceso era el deseado.

A continuación y de forma análoga al primer experimento, se procedió a la implementación de un sistema que extrajera las características de las distintas imágenes a partir de someter las mismas a los distintos filtros LBP, OLBP, LDerivP y LDP.

En este caso, el programa consta de dos partes. La primera de ellas coincide con la lectura y adecuación de imágenes ya mencionada con anterioridad, y la segunda se adecuó para trabajar con documentos, líneas y palabras de forma distinta. Los documentos se dividen en bloques de 128x128 píxeles solapados un 20% y estos bloques son los que se someten a la parametrización. Las líneas y palabras seguían la metodología que se presentó inicialmente, dividiéndose en cuatro franjas horizontales antes de pasar por los filtros y extraer sus características.

Visto el programa principal, el procedimiento final seguido en este experimento comienza por escoger las escrituras a enfrentar y a qué nivel se desea trabajar, para hacer más clara esta explicación, supongamos que queremos identificar líneas en hindi e inglés, es decir, queremos saber si nuestro sistema es capaz de distinguir entre los alfabetos latino y devanagari a nivel de línea. El procedimiento seguido es el siguiente:

1. Se escogen las muestras destinadas a entrenamiento y las destinadas al proceso de test pertenecientes a cada tipo de escritura.
2. Seguidamente, se pasan las imágenes de entrenamiento y test por el programa de extracción de características explicado anteriormente, obteniendo así los vectores de entrenamiento y test respectivamente.
3. Finalmente, estos vectores son introducidos en el clasificador SVM, obteniendo como resultado las tablas de confusión y los distintos porcentajes de acierto usando LBP, OLBP, LDerivP, LDP o la combinación de algunos, o todos ellos.

También se nos facilitó una serie de imágenes de texto manuscrito desde el *Indian Statistical Institute* para probar la eficacia de nuestro planteamiento en documentos de esta naturaleza. Estas muestras fueron tomadas de distintos autores en diferentes tipos de escritura, por lo que cada caligrafía era distinta así como el ancho del trazo de las mismas y el tamaño de la fuente.

Las imágenes de documentos manuscritos fueron divididas en líneas y palabras también de forma semiautomática y se elaboró una base de datos de texto manuscrito de forma similar a la creada para texto impreso. Finalmente, se procedió a su análisis del mismo modo que con las imágenes de la base de datos principal y se estudiaron los resultados obtenidos.

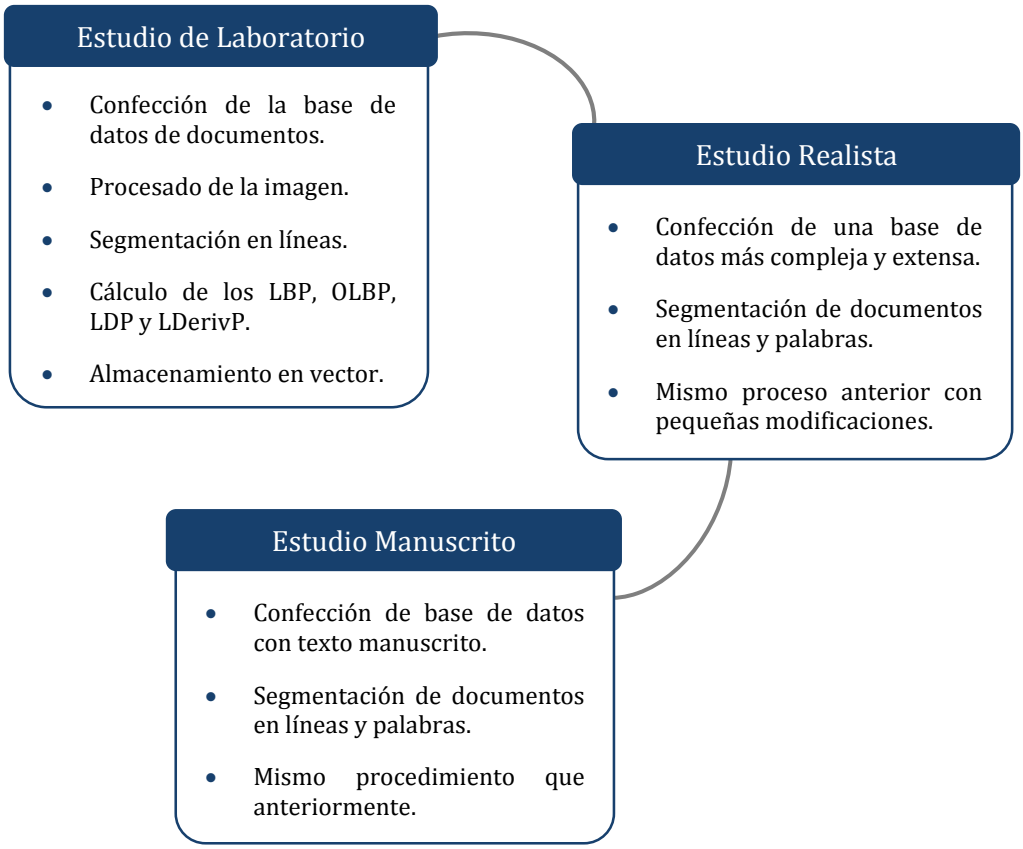


Figura 6. Diagrama de flujo que resume la metodología seguida en este proyecto.

1.7. Peticionario

Actúa como peticionario de este proyecto la Escuela de Ingeniería de Telecomunicación y Electrónica (EITE) de la Universidad de Las Palmas de Gran Canaria (ULPGC), siendo la realización de este Proyecto de Fin de Carrera requisito indispensable para la obtención del título de Ingeniero de Telecomunicación.

1.8. Estructura de la memoria

La memoria de este Proyecto de Fin de Carrera consta de 6 capítulos. A continuación, se describe brevemente el contenido de cada uno de ellos.

Capítulo 1. Introducción. En este capítulo se realiza una primera aproximación a la temática del proyecto. Se comenta en qué se fundamenta el campo del análisis de imágenes de documentos y cuáles son sus principales líneas de trabajo. También se explica en qué consiste la digitalización de documentos y la importancia de este proceso en la actualidad.

A su vez, se nombran las principales características, ventajas y desventajas de los sistemas de reconocimiento óptico de caracteres (OCR), y se presenta la problemática que conlleva el procesado de documentos con más de un tipo de escritura para este tipo de sistemas. De igual forma, se comentan las características de los textos en la India, una de las regiones geográficas donde coexisten un mayor número de tipos de escritura distintos. Finalmente, se fijan los objetivos del proyecto y se especifica la metodología adoptada durante el desarrollo del mismo.

Capítulo 2. Estado del Arte. En este capítulo se analizan algunas de las técnicas existentes en el campo del análisis de imágenes de texto dentro del estado del arte, comentando brevemente sus características.

Capítulo 3. Propuesta de Clasificador. En el cuarto capítulo se realiza un estudio de los parámetros de análisis de texturas que se proponen como alternativa a los métodos actuales de reconocimiento y detección del tipo de escritura, aportando además parte de su programación. Finalmente, se describe el clasificador SVM, que es el sistema escogido para verificar la idoneidad de los parámetros de análisis propuestos en este proyecto.

Capítulo 4. Estudio de Laboratorio. En este capítulo se describe el procedimiento para el diseño y elaboración de la primera base de datos, así como la implementación del primer sistema desarrollado para su análisis y los resultados de este estudio inicial, el cual se llevó a cabo con el objetivo de comprobar la viabilidad de la aplicación de técnicas destinadas al análisis de textura en la identificación de tipos de escritura en imágenes.

Capítulo 5. Estudio Realista. En este capítulo se especifica el proceso seguido para conformar la base de datos final con la que se ha trabajado. Por consiguiente, se describen los procedimientos seguidos para la adquisición de las imágenes de los documentos de cada tipo de escritura a partir de los ejemplares físicos, así como la obtención de las imágenes de las líneas y palabra extraídas de dichas muestras. También se detallan en este apartado las pruebas realizadas sobre la mencionada base de datos, y se exponen los resultados obtenidos tras la aplicación de los parámetros de análisis de textura.

Capítulo 6. Comprobaciones sobre texto manuscrito. En este capítulo se presentan los resultados aplicando los parámetros de análisis de texturas en imágenes de texto manuscrito. En consecuencia, se explica cuál ha sido el procedimiento seguido para la conformación de la base de datos de documentos manuscritos y qué pruebas se han llevado a cabo sobre la misma.

Capítulo 7. Conclusiones y líneas futuras. En este capítulo se extraen las conclusiones generales del proyecto que se ha desarrollado y se indican las principales líneas de mejora para futuros trabajos relacionados.

Además de los capítulos que se han enumerado, este proyecto cuenta con una serie de documentos y anexos que complementan el contenido de la memoria:

Referencias bibliográficas. Se incluyen la bibliografía consultada durante la realización de este proyecto.

Pliego de condiciones. Se indica con qué recursos y bajo qué condiciones se ha desarrollado este Proyecto de Fin de Carrera. En consecuencia, se nombran las especificaciones materiales y de equipos, y las condiciones para su correcta ejecución. De igual forma, se dan las instrucciones necesarias para que el material y la información que se encuentran en el DVD puedan ser utilizados de forma adecuada. Finalmente, se enumeran las condiciones legales que rodean este trabajo.

Planos y programas. Se ilustrarán las principales características de los programas que han sido desarrollados en este proyecto. No se han incluido planos en este apartado puesto que no ha sido necesaria la generación de ninguno a lo largo de este proyecto.

Presupuesto. Se enumeran y se detallan los conceptos que componen el presupuesto del presente proyecto.

Anexo I. Contenido del DVD-R. Este último anexo, incluye un DVD con la información generada en este trabajo, según el modelo normalizado de la E.I.T.E.

Capítulo 2. Estado del arte

2.1. Introducción

Actualmente, se emplean una gran variedad de escrituras para redactar documentos en distintos idiomas a nivel mundial. En un entorno multilingüe y de múltiples alfabetos como es el nuestro, es imprescindible conocer la escritura utilizada en un documento antes de emplear un algoritmo de reconocimiento de caracteres, y escoger las técnicas de análisis de documentos apropiadas.

El reconocimiento de escritura resulta de gran utilidad para la lectura de documentos en los que coexisten distintas escrituras a lo largo de diferentes párrafos, bloques de texto, líneas o palabras dentro de una misma página. El análisis de tales documentos se desarrolla en dos etapas: en primer lugar, se procede a la identificación y separación de las diferentes regiones de texto escrito en el documento, y seguidamente, se lee cada región escrita individualmente usando el sistema OCR correspondiente.

Asimismo, la identificación de la escritura de forma automática es crucial para satisfacer la creciente demanda de sistemas de procesado electrónico de grandes volúmenes de documentos redactados en diferentes escrituras. Esto es importante, por ejemplo, para realizar transacciones comerciales en toda Europa y Oriente, y tiene una gran relevancia en países como la India, que como ya hemos mencionado con anterioridad, tiene muchas lenguas y escrituras estatales oficiales. Debido a esto, ha habido un creciente interés a lo largo de los últimos años en que la tecnología OCR sea capaz de trabajar con múltiples tipos de escritura.

En vista de esto, se han desarrollado varios métodos para la identificación de la escritura de forma automática. Todos ellos pueden ser englobados principalmente en dos categorías, por un lado, tenemos las técnicas basadas en la estructura del texto y por otro, las que se centran en la apariencia visual del mismo.

Este apartado pretende dar una visión general de las diferentes metodologías de identificación de escritura más populares.

2.2. Métodos de Reconocimiento de Escritura.

La identificación de escritura se basa en el hecho de que cada tipo de escritura tiene una distribución espacial única, así como una serie de atributos visuales que permiten distinguirlo del resto. Por lo tanto, la tarea básica en el reconocimiento de la escritura es

idear técnicas para descubrir estas características dentro de un documento determinado y luego, clasificar el tipo de escritura del documento a partir de las mismas [11, 12].

Si nos basamos en la clase de características que se detectan y analizan, los métodos para el reconocimiento de escritura pueden dividirse en dos grupos: los basados en la estructura y los que se centran en la apariencia visual. Y a su vez, las técnicas de reconocimiento de escritura incluidas en cada una de estas dos categorías pueden ser clasificadas según el nivel del escrito a analizar, es decir, se aplican distintos métodos si se analizan páginas completas, párrafos, líneas o palabras.

2.3. Reconocimiento de escritura basado en la estructura.

En general, los tipos de escritura difieren entre sí en su estructura, las conexiones entre sus trazos y en los estilos que tienen los distintos conjuntos de caracteres.

Uno de los primeros planteamientos que se desarrollaron con el fin de identificar los problemas asociados con el reconocimiento del tipo de escritura se expone en el trabajo *Connected Components in Binary Images: The Detection Problem* [13]. Éste propone un método que consiste en extraer los componentes conectados de un documento, y luego analizar sus formas y estructuras con el fin de revelar las características morfológicas de la escritura que se use en ese documento.

En esta línea, se comprobó que en los textos impresos en latín, griego, etc., cada carácter completo o parte del mismo supone un componente conectado. Análogamente, en los documentos escritos a mano, los caracteres de una palabra o parte de la misma pueden tocarse entre sí formando un único componente conectado. Así, en escrituras tales como devanagari, bengalí, árabe, etc., una palabra o una parte de la misma puede ser un componente conectado.

A partir de los buenos resultados logrados es este estudio, se llevaron a cabo nuevos métodos de identificación de escritura basados en la extracción y el análisis de componentes conectados, los cuales se encuentran incluidos dentro de la categoría de las técnicas basadas en el análisis de la estructura del texto. Veamos algunos ejemplos clasificados según el tamaño del área del texto sobre la que se aplican.

2.3.1. Técnicas de identificación a nivel de página

Uno de los investigadores más importantes en el ámbito de la identificación de escritura usando las relaciones espaciales de las estructuras de los distintos caracteres que componen el texto, fue Spitz.

Su primera publicación [14] presentaba un sistema capaz de diferenciar los alfabetos japonés y latino en documentos impresos. En ella se empleó la densidad óptica de caracteres para clasificar cada una de las líneas individuales en un documento escrito en los tipos de escritura mencionados.

En otro de sus trabajos [15], Spitz empleó la distribución de las concavidades ascendentes de los caracteres para discriminar las escrituras japonesa y latina, consiguiendo entonces un 100% de éxito.

En su tercer estudio más importante [16], desarrolló además un clasificador de dos etapas para lograr la identificación de las escrituras. Este sistema combina las características de sus otras dos propuestas. Así, en la primera etapa del clasificador se distingue el alfabeto latino del japonés mediante la comparación de las varianzas de las distribuciones de sus concavidades ascendentes. Y por último, en la segunda etapa se realiza un análisis de la distribución de la densidad óptica de los caracteres del texto. Además, este último sistema va un paso más allá y también distingue entre distintos idiomas que utilizan el alfabeto latino observando los códigos más frecuentes de la forma de los caracteres. Este método alcanza una precisión del 93% aproximadamente.

A continuación, se muestra un diagrama de bloques que representa el flujo de información en este último proceso:

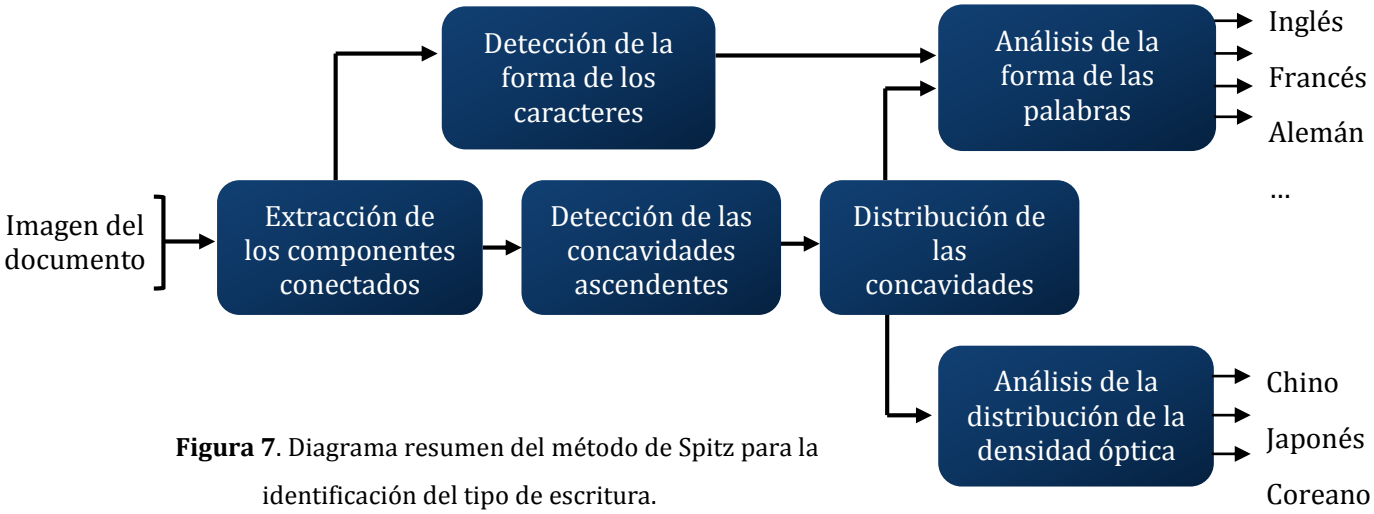


Figura 7. Diagrama resumen del método de Spitz para la identificación del tipo de escritura.

Posteriormente, los trabajos de Spitz fueron extendidos por Lee, Nohl y Baird en el artículo "*Language Identification in Complex, Unoriented, and Degraded Document Images*" [17]. En esta ocasión, el tipo de escritura de un documento impreso se identifica analizando algunas características estructurales adicionales y tomando una decisión mediante la técnica del más votado, usando para tal fin los resultados de la clasificación de texto ya reconocido. En este estudio se emplean los parámetros planteados por Spitz además del análisis de la distribución de la altura de los caracteres y de los perfiles superior e inferior de la envolvente de cada carácter.

Los resultados experimentales mostraron que este método permite separar documentos chinos y japoneses de documentos redactados en inglés, francés, alemán, español e italiano, consiguiendo un 98,16 por ciento de acierto.

2.3.2. Técnicas de identificación a nivel de párrafo.

Los métodos de identificación de la escritura que pertenecen al grupo anterior requieren grandes bloques de texto, tal que exista suficiente información para lograr la extracción de las características de la estructura del tipo de escritura. En este sentido, las técnicas que hemos citado ofrecen un buen rendimiento cuando se utilizan para la identificación de la escritura a nivel de página, pero no pueden mantenerlo cuando se aplican sobre un bloque de texto más pequeño.

En los documentos con más de un tipo de escritura es necesario identificar y separar las diferentes regiones de texto como párrafos, líneas, palabras o incluso caracteres dentro de cada página del documento. Esto es particularmente importante en un país como la India, que alberga una gran variedad de escrituras como pueden ser devanagari, bengalí, tamil, telugu, malayalam u oriya, entre muchas otras.

Así, se han desarrollado múltiples sistemas que realizan el reconocimiento de la escritura a nivel de párrafo, involucrando la diferenciación de varios de los alfabetos mencionados anteriormente. Uno de los trabajos más completos en esta línea fue llevado a cabo por Chaudhury y Sheth, quienes publicaron la obra "*Trainable Script Identification Strategies for Indian Languages*" [18]. En ella se plantearon tres estrategias diferentes para reconocer la escritura contenida en un bloque de texto de un documento impreso.

1. En la primera técnica, se analizan los bloques de texto mediante el perfil de proyección horizontal de los mismos, extrayendo los coeficientes de la transformada de Fourier.

En cuanto al proceso de clasificación, se decide cuál es la escritura de cada bloque de acuerdo a la distancia euclídea en el espacio de Eigen.

La tasa de reconocimiento media obtenida, realizando pruebas para distinguir entre escritura latina, devanagari, telugu y malayalam, fue de aproximadamente 85%.

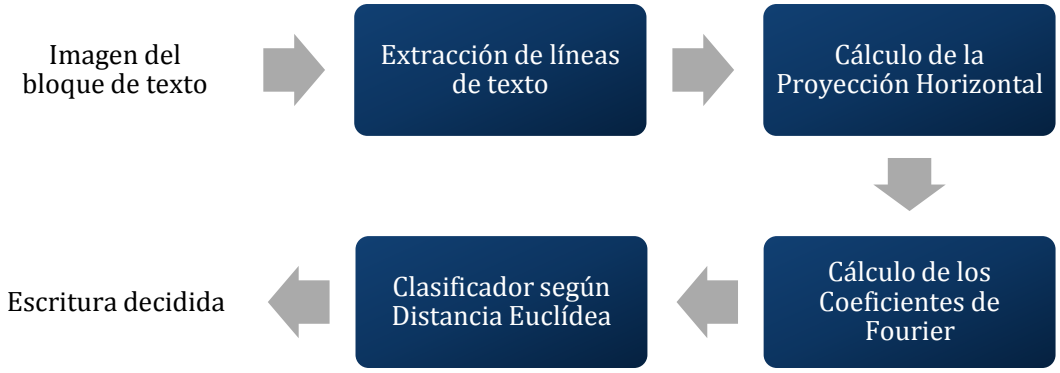


Figura 8. Flujo de trabajo de la primera estrategia de Chaudhury y Sheth

2. El segundo sistema se basa en características relacionadas con los componentes conectados en estos bloques de texto, empleando las medias y las desviaciones estándar de las salidas de un filtro Gabor de seis canales.

   La clasificación en este caso, se lleva a cabo mediante la distancia de Mahalanobis y la tasa de reconocimiento media en esta ocasión fue de 95%.

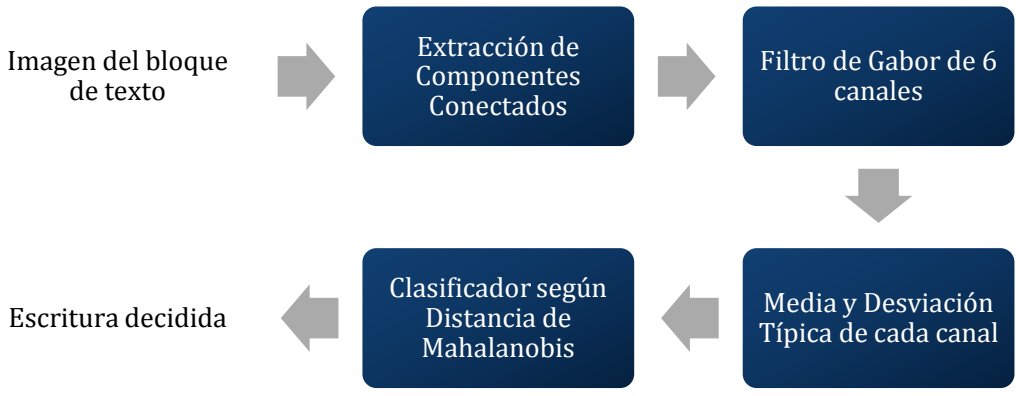


Figura 9. Flujo de trabajo de la segunda estrategia de Chaudhury y Sheth

3. El último método se centra en la distribución de las proporciones entre la anchura y la altura de los componentes conectados presentes en el documento.

   Para la clasificación también se utiliza la distancia de Mahalanobis y se alcanzó una tasa de reconocimiento media del 89%.

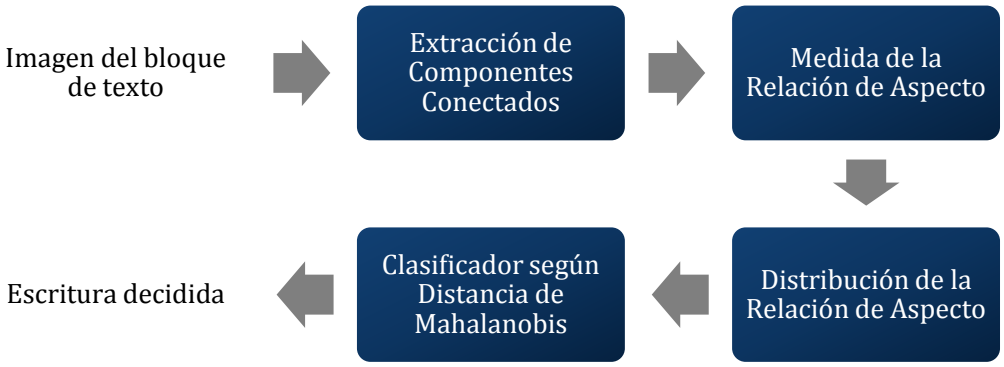


Figura 10. Flujo de trabajo de la tercera estrategia de Chaudhury y Sheth

2.3.3. Técnicas de identificación a nivel de línea.

Un estudio relevante sobre identificación de escritura a nivel de línea en documentos de la India se titula *"Identification of Different Script Lines from Multi-Script Documents"* [19], y fue publicado por Pal y Chaudhuri.

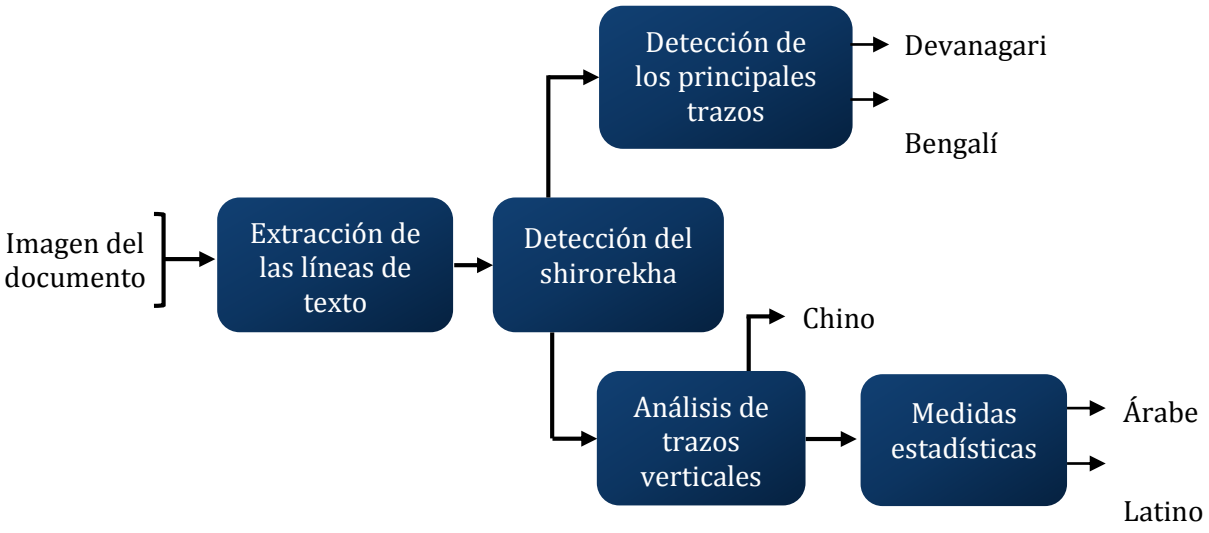


Figura 11. Método de Pal y Chaudhuri para la identificación de documentos hindúes con múltiples alfabetos

El proceso seguido se representa en la Figura 10 y se trata de un sistema automático para la identificación de líneas de texto usando los alfabetos bengalí, latino, chino, árabe y devanagari en documentos impresos.

Éste separa en primer lugar las líneas de texto del documento de entrada y posteriormente lleva a cabo una serie de comprobaciones para determinar el tipo de escritura del documento.

La primera comprobación que se realiza es si existe una característica propia de algunas escrituras hindúes, que consiste en una línea horizontal que se extiende sobre los caracteres llamada "shirorekha". Esto permite separar los alfabetos devanagari y bengalí de las escrituras árabe, latina y china.

Una vez hecho esto, se distinguen las líneas en bengalí de las de devanagari mediante la observación de la presencia de ciertos trazos específicos de cada uno de estos alfabetos en el texto, y se reconocen las líneas escritas en chino comprobando si existen caracteres con cuatro o más trazos verticales.

Por último, las líneas en inglés que usan alfabeto latino se separan de las líneas en árabe mediante medidas estadísticas. Una de estas características es la distribución de los puntos de la parte inferior de los caracteres, por ejemplo, los puntos de los caracteres escritos en alfabeto latino, se distribuyen únicamente a lo largo de la línea de base y la línea de fondo de los caracteres, mientras que en árabe se distribuyen al azar.

Usando todas estas características estructurales, las tasas de identificación fueron 97.32, 98.65, 97.53, 96.05 y 97.12 por ciento para las escrituras bengalí, latina, china, árabe, devanagari, respectivamente, con una precisión global del 97,33 por ciento.

2.3.4. Técnicas de identificación a nivel de palabra.

En comparación con la identificación del tipo de escritura en párrafos y líneas de texto, el reconocimiento a nivel de palabra en un documento cuando éste contiene más de un alfabeto es un proceso mucho más complejo. Esto se debe a que la información disponible en unos pocos caracteres dentro de una palabra puede no ser suficiente para el propósito que nos ocupa. Este hecho ha motivado a muchos investigadores a buscar soluciones a este problema, intentando incluso realizar el reconocimiento a nivel de caracteres.

Uno de los primeros trabajos sobre la identificación de la escritura a nivel de carácter, fue expuesto por Lee y Kim en "*Multi-Lingual, Multi-Font, Multi-Size Large-Set Character Recognition Using Self-Organizing Neural Network*" [20], donde trataron de resolver este problema usando redes neuronales.

Las redes neuronales implementadas en este trabajo son capaces de determinar a qué escritura pertenece cada carácter y lo clasifica en cuatro grupos: latino, chino, coreano y mixto. Los caracteres que se incluyen en el grupo mixto, son aquellos que no pueden ser clasificados por la red con plena confianza y se identifican en el siguiente nivel empleando

una serie de procedimientos de clasificación más detallados, haciendo uso del aprendizaje basado en la cuantificación de vectores.

Con el fin de evaluar el rendimiento del sistema propuesto, se llevaron a cabo experimentos con 3.367.200 caracteres y se obtuvo tasa de reconocimiento de más de 98,27 por ciento.

Más tarde, en "*Page Segmentation Using Script Identification Vectors: A First Look*" [21], se extiende este trabajo incluyendo la separación de las diferentes regiones de texto en un documento impreso con más de un alfabeto.

En este caso, cada símbolo textual (carácter, palabra o parte de una palabra) dentro de un documento, se hace corresponder con un conjunto de símbolos plantilla, y se clasifica la clase de escritura según con qué símbolo plantilla guarde mayor grado de coincidencia.

Se observó que este método ofrece una buena respuesta en todos los casos, excepto en aquellos visualmente similares, como pueden ser el alfabeto latino frente al cirílico o el latino frente al griego; mientras que se obtuvieron mejores resultados en pares de escritura con apariencias visuales distintas, como ocurre al comparar el alfabeto árabe con el latino, el latino frente al japonés o el coreano.

2.4. Reconocimiento de escritura basado en la apariencia visual.

Generalmente, los tipos de escritura se diferencian entre sí por la forma de los caracteres y cómo éstos se agrupan en palabras, las palabras en frases, etc. Esto le da a los distintos alfabetos diferentes apariencias visuales que permiten que puedan distinguirse y este es realmente el método utilizado por un observador casual para realizar la distinción por sí mismo. Por lo tanto, una forma natural de identificar la escritura de un documento consiste en analizar su aspecto visual.

Por otro lado, el aspecto visual es a menudo relacionado con la textura. Así, un bloque de texto escrito con un alfabeto determinado forma un patrón de textura característico y por consiguiente, el problema de la identificación del tipo de escritura se reduce a un análisis de textura, pudiendo entonces emplear cualquier algoritmo de clasificación de textura para realizar esta tarea.

El método propuesto en este Proyecto de Final de Carrera se encuentra englobado en esta categoría, puesto que se plantea un sistema capaz de distinguir entre múltiples tipos de alfabeto haciendo uso de algoritmos propios del análisis de texturas.

Así, se describen a continuación algunas de las técnicas más usuales para reconocer distintos alfabetos en imágenes de documentos mediante su aspecto visual.

2.4.1. Técnicas de identificación a nivel de página.

Uno de los primeros sistema de análisis de texturas aplicado al reconocimiento de la escritura, aparece descrito en el trabajo "*Rotation Invariant Texture Features and Their Use in Automatic Script Identification*" [22], que emplea un filtro de Gabor para la identificación de escritura en texto impreso.

El sistema en cuestión extrae bloques de texto uniformes del documento y lleva a cabo el análisis de textura de cada uno de ellos, extrayendo sus características mediante el uso de un filtro de Gabor de 16 canales.

La respuesta promedio de cada canal proporciona una medida característica para cada alfabeto, la cual es robusta frente al ruido pero dependiente de la rotación. Por consiguiente, para lograr invariancia frente a la rotación, se calculan los coeficientes de Fourier del conjunto de las 16 salidas, una por cada canal.

Para finalizar, se compara el vector de características generado a partir del bloque de texto de entrada con los vectores de características representativos de cada tipo de escritura, utilizando la distancia euclidiana. Estos vectores se forman calculando el vector promedio de características obtenido a partir de un amplio conjunto de documentos de entrenamiento redactados mediante una escritura concreta.

Este método logró una tasa de precisión global del 96,7% al discriminar documentos en escritura china, latina, griega, rusa, persa y malayalam.

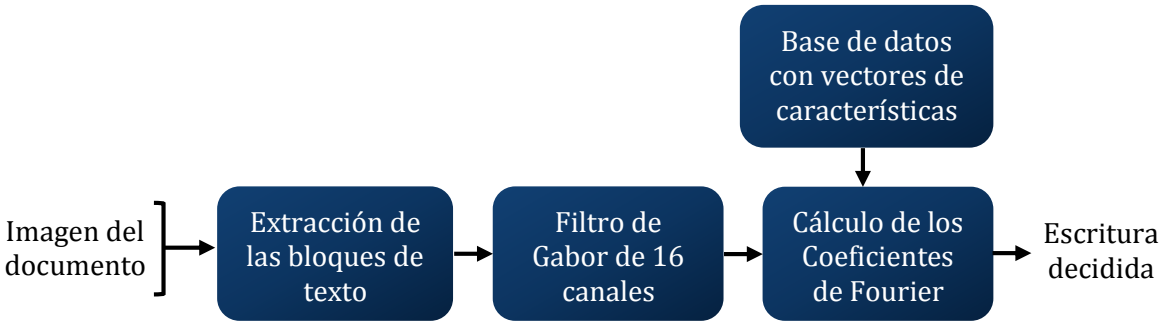


Figura 12. Diagrama de bloques del procedimiento para el reconocimiento de escritura propuesto por Tan

Emplear filtros de Gabor para identificar el tipo de escritura es una práctica muy extendida y que por lo general alcanza buenos resultados. Sin embargo, uno de los problemas asociados al uso este tipo de filtros es el alto coste computacional que supone el filtrar imágenes muchas veces. Con el fin de encontrar una solución, se propuso la identificación del tipo de escritura en documentos impresos utilizando filtros Gabor orientables [23].

Esta técnica explota la capacidad del filtro de Gabor para ser orientado y eso permite reducir el número de filtrados de imagen durante el proceso de identificación. Para llevar a esto cabo se diseña un banco de filtros Gabor que permite extraer parámetros invariantes con la orientación para discriminar escrituras que contengan caracteres que sean similares en forma e incluso, distinguir aquellos con caracteres en común.

Aplicando esto, se logró una tasa de reconocimiento del 98,5 por ciento al discriminar textos escritos con los alfabetos chino, japonés, coreano y latino, y además, el número de operaciones de filtrado de imagen se redujo en un 40 por ciento.

Cabe destacar que aunque los esquemas de reconocimiento de escritura basados en filtros de Gabor han mostrado un buen rendimiento, su aplicación se limita únicamente a los documentos impresos. Esto se debe a que las variaciones en el estilo de escritura, en el tamaño de los caracteres y en el espaciado entre palabras y líneas, hacen que el proceso de reconocimiento se vuelva difícil y poco fiable cuando se trata de analizar texto manuscrito o con distintas fuentes o tamaños. Es por ello que si se desea usar filtros de Gabor es necesario realizar un pre-procesado de las imágenes de los documentos antes de aplicarlo, a fin de compensar las diferentes variaciones que puedan estar presentes.

2.4.2. Técnicas de identificación a nivel de párrafo.

El estudio de nuevas características de textura que sirvieran para clasificar los distintos tipos de escritura presentes en un documento, fue un trabajo impulsado especialmente por Busch.

En su obra "*Texture for Script Identification*" [24] se emplean parámetros relacionados con ondículas como su energía, su media logarítmica o su coocurrencia logarítmica entre otros. Se experimentó sobre una base de datos con ocho tipos de escritura, concretamente: latina, coreana, japonesa, griega, cirílica, hebrea, devanagari y farsi.

En estas pruebas, se binarizaron en primer lugar las imágenes del documento impreso, se corrigió su inclinación y se normalizaron los bloques de texto. A fin de reducir las dimensiones de los vectores de características y mejorar la tasa de precisión del proceso de clasificación, se aplicó la función lineal discriminante de Fisher y posteriormente se procedió a realizar la clasificación.

Se observó entonces que la coocurrencia logarítmica de las ondículas es el parámetro que produce mejores resultados, obteniendo un error de clasificación del 1 por ciento. Por el contrario, los peores resultados se obtuvieron empleando las matrices GLCM que generaron un error de clasificación del 9.1 por ciento.

Sin embargo, este enfoque tiene un problema, y es que generar un modelo por cada clase de escritura se vuelve útil cuando cada texto está escrito utilizando una sola fuente o usando sólo fuentes visualmente similares. No obstante, existe un gran número de fuentes dentro de cada tipo de escritura, y éstas son muy variables en lo que a su apariencia se refiere. Debido a este hecho, es poco probable que un modelo generado a partir de un conjunto de fuentes identificara correctamente una imagen de texto con una fuente inédita del mismo alfabeto. Por ejemplo, en el caso expuesto si repetimos los experimentos usando una fuente distinta y desconocida para el sistema, el error aumenta, pasando del 1 y 9,1 por ciento al 15,9 y 13,2 por ciento respectivamente.

En vista de esto, Busch propuso la caracterización de múltiples fuentes en una sola escritura de manera más adecuada mediante la creación de varios modelos por cada clase de escritura [25].

Así, se procedió a la división de cada tipo de escritura en 10 subclases, tal que cada subclase se correspondía con una de las fuentes incluidas dentro de ese alfabeto concreto. Una vez hecho esto, se llevaba a cabo un análisis discriminante y se procedía a realizar la clasificación igual que en el caso anterior.

Tras hacer esto, surge una mejora significativa en comparación con los resultados obtenidos utilizando un solo modelo y varias fuentes, reduciéndose el error a 2,1 y 12,5 por ciento respectivamente para los dos casos mencionados.

2.4.3. Técnicas de identificación a nivel de palabra.

La identificación de escritura a nivel de la palabra se llevó a cabo con éxito en numerosos artículos usando técnicas similares a las ya vistas.

Ejemplo de ello son dos trabajos con el título "*Script Identification in Printed Bilingual Documents*" [26]. Se trata de dos métodos distintos, el primero de ellos consiste en estructurar las palabras en tres zonas espaciales diferentes empleando la información sobre la distribución espacial de las palabras en dichas zonas. La segunda técnica analiza la distribución de energía direccional de las palabras usando para este fin filtros de Gabor con frecuencias y orientaciones adecuadamente ajustadas. Ambos algoritmos se basan en las siguientes premisas:

- Generalmente, los puntos de los caracteres latinos se distribuyen espacialmente por las zonas media y superior. Sólo unos pocos caracteres en minúsculas lo hacen en la zona inferior. En tamil, los caracteres se distribuyen en las zonas superior e inferior.
- El alfabeto latino contiene un mayor número de trazos verticales y oblicuos. En tamil sin embargo, predominan los trazos horizontales y verticales.
- La relación de aspecto de los caracteres en tamil es generalmente mayor que la de los caracteres latinos.

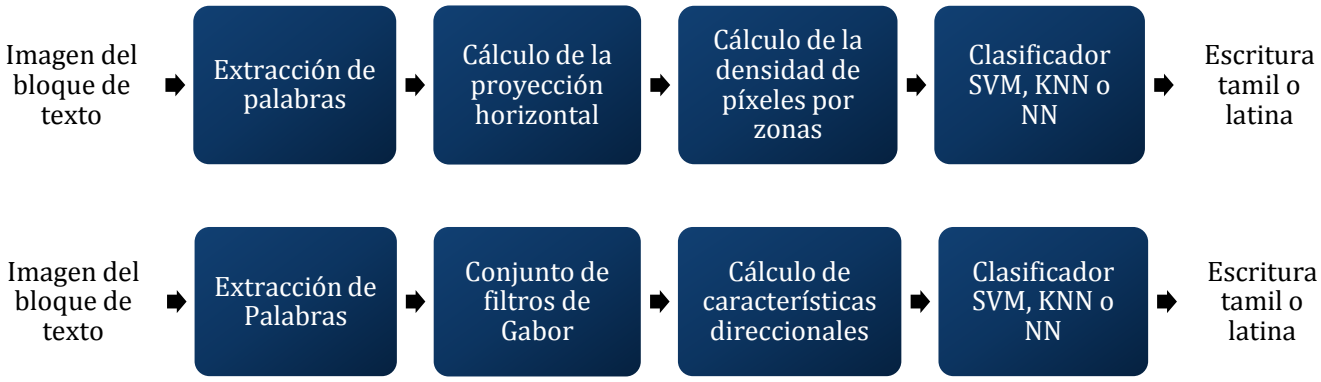


Figura 13. Procedimientos de identificación de escritura a nivel de palabras propuestos por Dhanya et al.

Todo esto sugiere que los procedimientos que se establecen en ambos métodos, pueden ser útiles en la distinción de palabras en las escrituras latina y tamil.

Así, en ambos casos se separan las palabras del documento y se calculan las características que corresponda. Las espaciales, se obtienen mediante el cálculo de la concentración de píxeles en cada zona tras el cálculo de la proyección horizontal, mientras que las características direccionales son las respuestas de los filtros de Gabor. Los valores extraídos se clasifican usando clasificadores SVM, KNN o del vecino más cercano, lo cual permite además cuantificar cuán bueno es cada clasificador para nuestro propósito.

Tras aplicar lo anterior, se observó que las características direccionales funcionan mejor que las espaciales, obteniéndose una tasa de precisión del 96 por ciento empleando un clasificador SVM. Esto se puede atribuir al hecho de que los filtros de Gabor pueden tener en cuenta la naturaleza general de las escrituras de una forma más adecuada.

2.5. Tabla resumen de resultados

En este apartado se resumen los resultados de todas las técnicas de identificación de escritura en la tabla siguiente:

Clase de método	Nivel	Autor	Escrituras a distinguir	Tasa de identificación
Según estructura	Página	Spitz	Japonesa, latina	93%
		Lee et al.	China, japonesa, latina	98,16%
	Párrafo	Chaudhury et al.	Latina, devanagari, telugu, malayalam	85%
				95%
				89%
	Línea	Pal et al.	Latina, bengalí, árabe, china, devanagari	97,33%
		Lee et al.	Latina, coreana, china	98,27%
Según apariencia visual	Página	Pan et al.	Latina, china, griega, rusa, persa, malayalam	96,7%
	Párrafo	Busch	Latina, coreana, japonesa, griega, cirílica, hebrea, devanagari, farsi	98,9%
	Palabra	Dhanya et al.	Tamil, latina	96%

Tabla 1. Resumen de resultados de métodos para identificar la escritura.

Capítulo 3. Propuesta de clasificador

3.1. Introducción

En este capítulo se explicarán los operadores propios del análisis de texturas que empleamos en este trabajo: LBP, OLBP, LDP y LDerivP. Además, se añadirá parte de la programación que hemos desarrollado para el cálculo de estos parámetros.

También se describen las principales características del clasificador LS-SVM que utilizamos en este proyecto.

3.2. Descripción de los parámetros de textura

3.2.1. Local Binary Pattern (LBP)

El ***Local Binary Pattern*** (LBP) es un operador de textura muy eficiente y a la vez de gran simplicidad, que etiqueta los píxeles de una imagen aplicando un umbral sobre la vecindad alrededor de cada píxel usando el valor del píxel central. Debido a su simplicidad computacional y su capacidad discriminativa, el operador LBP se ha convertido en un método popular en diversas aplicaciones, como puede ser el reconocimiento del tipo de escritura [27, 28].

3.2.1.1. Definición

El ***Local Binary Pattern*** se puede ver como un método que unifica los modelos tradicionales estadísticos y estructurales de análisis de textura.

Quizás la propiedad más importante del operador LBP en aplicaciones del mundo real es su robustez a los cambios en escala de grises causados, por ejemplo, por las variaciones de iluminación. Otra propiedad importante es su simplicidad computacional, lo que hace que sea útil, por ejemplo, en el análisis de imágenes en tiempo real.

Aplicado a imágenes en escala de grises, el LBP puede verse como la unión de las direcciones binarias del gradiente. El histograma de los patrones LBP obtenidos contiene información acerca de la distribución de los bordes, manchas y otras figuras locales en una imagen, las cuales se pueden utilizar como características del trazo del texto y es por eso que el LBP resulta eficiente a la hora de identificar los distintos tipos de escrituras.

En el artículo "*Texture Analysis with Local Binary Patter*", Topi Mäenpää y Matti Pietikäinen definen el operador LBP como una medida de textura invariante a escala de

grises en una vecindad local. El operador LBP original, etiqueta los píxeles de una imagen mediante el uso de un umbral aplicado a cada uno de los píxeles que componen una vecindad de tamaño 3x3 y posteriormente, concatena los resultados para formar un número.

Supongamos que una imagen dada se define como $I(Z) = I(x, y)$. El operador LBP transforma la imagen de entrada convirtiéndola en $LBP(Z) = LBP(x, y)$ de la siguiente manera:

$$LBP(Z_c) = \sum_{p=0}^{7} s(I(Z_p) - I(Z_c)) \times 2^p \quad \text{(Ec. 1)}$$

$$s(l) = \begin{cases} 1 & , l \geq 0 \\ 0 & , l < 0 \end{cases} \quad \text{(Ec. 2)}$$

Donde $s(l)$ representa la función escalón unitario e $I(Z_p)$ es el valor de cada uno de los píxeles que forman el conjunto de los 8 vecinos alrededor del píxel central Z_c.

El código $LBP(Z)$ contiene información acerca de la estructura a la que pertenece el píxel, como puede ser el borde del trazo, las esquinas de los trazos, el fin del trazo, el interior de las letras o el fondo, entre otros. Las características de la parte superior, inferior y el área central de las líneas difieren por el hecho de ser distintas y distintivas dentro de cada tipo de escritura, y por esa razón se modela la distribución espacial de los patrones locales, dividiendo la línea en una serie de regiones adyacentes y calculando el histograma en cada una de ellas con el fin de no perder la ubicación de las diferentes estructuras internas de la imagen.

La figura siguiente muestra un ejemplo de cómo obtener el LBP, estableciendo el umbral en el valor del píxel central. Aquellos píxeles alrededor del central que sean mayores al mismo se etiquetan con un "1", y los menores con un "0". Una vez establecido el umbral se conforma la cadena binaria y se convierte a un término decimal.

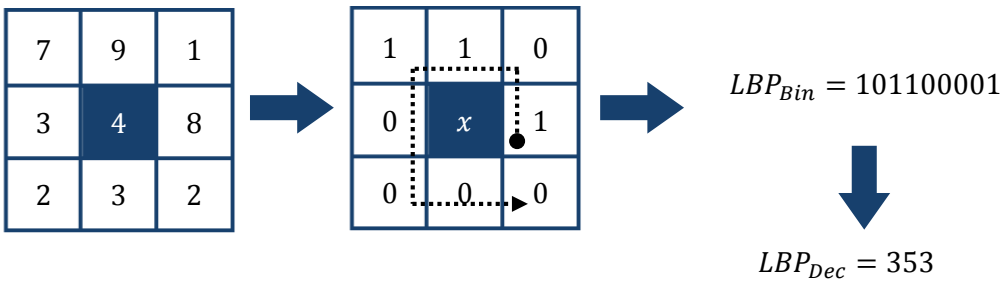


Figura 14. Ejemplo de cálculo del LBP.

3.2.1.2. Programación

El LBP es un operador que trabaja con imágenes en escala de grises. En nuestros sistemas las imágenes son convertidas a formato lógico antes de extraer sus características, por lo que se encuentran en blanco y negro antes de obtener el LBP de las mismas.

Así, el proceso seguido para hallar el LBP consiste en primer lugar, en tomar un conjunto de píxeles de tamaño 3x3 (mv) rodeando uno central ($Z_c = (ifil, icol)$) de la imagen de entrada ($reglone$). A continuación, se toman los bits de manera ordenada como se vio en el ejemplo del punto anterior. Y finalmente, usando el vector resultante, se transforma ese código binario ($codebit$) en un número decimal, obteniendo así el valor del píxel de la nueva imagen $Ilbp$.

```
mv=reglone(ifil-1:ifil+1,icol-1:icol+1);
mv=mv>mv(2,2);
if sum(mv(:))>0
    mvfile=mv(:);
    codebit=mvfile([8 7 4 1 2 3 6 9]);
    Ilbp(ifil,icol)=bi2de(codebit');
end
```

3.2.2. Orientation of Local Binary Pattern (OLBP)

El operador ***Orientation of Local Binary Pattern*** (OLBP) se diferencia del LBP original en que permite representar de forma más explícita la información de la orientación de los trazos, que son los elementos más importantes para llevar a cabo la identificación del tipo de escritura.

3.2.2.1. Definición

Este operador aparece descrito en el artículo "*Hand Vein Recognition Base on Orientation of LBP*" [29]. En este trabajo, se propone una nueva forma de representar las características de las venas: el *Orientation of Local Binary Pattern* (OLBP), que es una extensión del *Local Binary Pattern* (LBP), que ya hemos comentado en el punto anterior.

Lo que hace interesante al operador OLBP es que puede representar adecuadamente la información de la orientación de los píxeles de las venas en el caso del artículo citado, aunque también se ha comprobado que también tiene esa ventaja aplicado a los trazos de un texto entre muchas otras aplicaciones.

El OLBP de un píxel Z_c se calcula siguiendo una serie de pasos:

1. En primer lugar, calculamos el LBP original para crear una secuencia binaria de ocho bits:

$$s\left(I(Z_p) - I(Z_c)\right); \quad p = 0, \dots, 7. \qquad \text{(Ec. 3)}$$

2. A continuación, hemos de encontrar el índice de comienzo ($Start$) y el de fin (End) de la cadena de ceros más larga dentro de la secuencia obtenida del paso anterior.

3. La media entre los puntos de inicio y fin representa la orientación del LBP: el OLBP. Éste puede obtenerse como sigue:

$$OLBP(Z_c) = round(Start + End/2)\ mod\ 8 \qquad \text{(Ec. 4)}$$

Donde la función $round()$ redondea un número al entero más cercano y mod es la operación que permite obtener el complementario aritmético.

Cabe destacar que en este proyecto hemos realizado una modificación en la fórmula original del OLBP, ya que se observó que se obtenían mejores resultados al aplicarlo sobre imágenes de texto. En vista de esto, la expresión del OLBP queda como sigue:

$$OLBP(Z_c) = Start + End \qquad \text{(Ec. 5)}$$

Veamos con un ejemplo cómo calcular el OLBP de un píxel determinado. Dada una matriz de píxeles de tamaño 3x3 y siguiendo el procedimiento descrito anteriormente se tiene:

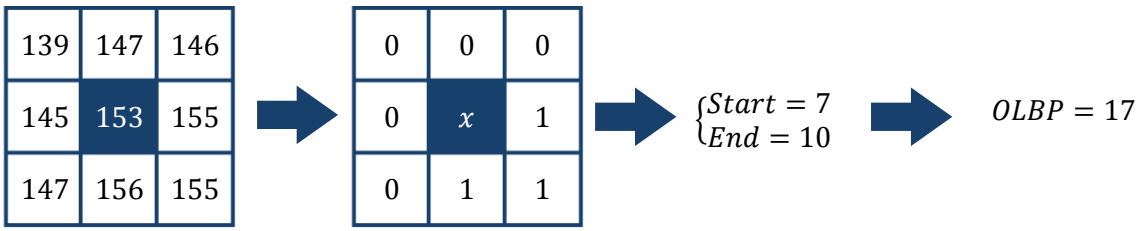


Figura 15. Ejemplo de cálculo del OLBP.

3.2.2.2. Programación

Nuestro sistema sigue cada uno de los puntos descritos y trabajamos como en el caso del LBP, con imágenes binarias, donde los píxeles valen "1" o "0".

Primero, se extrae un bloque de píxeles de tamaño 3x3 (mv) de la imagen de análisis ($reglone$), luego se toman todos los píxeles ($codebit$) alrededor del central ($Z_c = (ifil, icol)$):

```
mv=reglone(ifil-1:ifil+1,icol-1:icol+1);
mv=mv>mv(2,2);
mvfile=mv(:);
codebit=mvfile([8 7 4 1 2 3 6 9]);
```

Seguidamente se buscan el comienzo ($start$) y el fin (fin) de las cadenas de bits, calculando para ello el LBP mediante la función $diff()$, que permite restar elementos de un vector elemento a elemento. Localizando en la misma sentencia aquellos casos donde se obtuviera un valor menor de cero en el caso del índice de inicio, o un uno en el índice de fin.

```
start=find(diff([1; codebit; codebit])==-1)-1;
fin=find(diff([codebit; codebit; 1])==1)-1;
[long,indlong]=max(fin-start+1);
```

En la última línea de código, se guarda el índice que haga que la diferencia entre las cadenas de principio y fin en ese punto sea máxima. Se suma una unidad para diferenciar el código OLBP del resultado obtenido cuando la diferencia entre píxeles es cero por tratarse de valores idénticos.

Finalmente, se guarda el valor del OLBP como valor del píxel central Z_c, siendo éste cero si no se ha podido encontrar ningún punto de inicio:

```
if isempty(start)
    Iolbp(ifil,icol)=0;
else
    Iolbp(ifil,icol)=start(indlong)+fin(indlong);
end
```

3.2.3. Local Directional Pattern (LDP)

El ***Local Directive Pattern*** (LDP) codifica un patrón local de manera más eficiente que el LBP y produce resultados más estables que el OLBP. Esto se debe a que permite calcular los valores de las respuestas de los bordes en diferentes direcciones, utilizando estos resultados para codificar la textura de una imagen [30].

3.2.3.1. Definición

Habitualmente, los investigadores utilizan el cambio de magnitud del gradiente en una dirección específica alrededor de los píxeles para codificar la textura de una imagen. En lugar de comparar el valor de la intensidad de la región vecina, estos métodos comparan la magnitud del gradiente del píxel vecino a largo de una dirección específica y lo codifican como un LBP original. En consecuencia, estas técnicas son incapaces de codificar la totalidad de la información, lo cual puede lograrse si se analizan las diferentes magnitudes de las respuestas de los borde a lo largo de múltiples direcciones en torno a un píxel en particular.

La función LDP es un código binario de ocho bits que se asigna a cada píxel de una imagen de entrada. Este patrón se calcula comparando el valor de la respuesta de los bordes de un píxel en diferentes direcciones. Para cumplir con este propósito, pueden emplearse múltiples metodologías de detección de bordes, como son el detector de borde de Kirsch, el de Prewitt o el de Sobel. Entre todos ellos, el detector de bordes de Kirsch permite detectar diferentes respuestas direccionales con mayor precisión que otros sistemas de la misma índole, debido a que esta técnica considera los ocho vecinos que rodean un píxel.

Así, dado un pixel central en la imagen, los ocho valores de respuesta de borde direccionales $\{mi\}, i = 0, 1, \ldots, 7$ se calculan a partir de las máscaras Kirsch M_i en ocho orientaciones diferentes centradas en su posición. Estas máscaras se muestran a continuación:

$$\begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix} \quad \begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix} \quad \begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} \quad \begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}$$

$$Este\ M_0 \qquad Nordeste\ M_1 \qquad Norte\ M_2 \qquad Noroeste\ M_3$$

$$\begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix} \quad \begin{bmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{bmatrix} \quad \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix} \quad \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{bmatrix}$$

$$Oeste\ M_4 \qquad Sudoeste\ M_5 \qquad Sur\ M_6 \qquad Sureste\ M_7$$

Figura 16. Máscaras de la respuesta de bordes de Kirsch en ocho direcciones.

Las máscaras de Kirsch se denominan también de brújula porque se definen considerando una máscara simple y rotándola en las ocho direcciones principales de la brújula: Norte, Noroeste, Oeste, Suroeste, Sur, Sureste, Este y Noreste.

No obstante, hemos de tener en cuenta que los valores de la respuesta no son igual de importantes en todas las direcciones. La presencia de esquinas o bordes presentan altos valores en algunas direcciones particulares. Por lo tanto, nos interesa conocer las k direcciones más importantes con el fin de generar el LDP. En consecuencia, los k bits direccionales más altos de la respuesta b_i se ponen a "1". Y a los (8-k) bits restantes del patrón LDP de 8 bits, se les asigna un "0". Por último, el código LDP se deriva de la expresión:

$$LDP_k = \sum_{i=0}^{7} b_i(m_i - m_k) \times 2^i \qquad \text{(Ec. 6)}$$

$$b_i(a) = \begin{cases} 1 & ; a \geq 0 \\ 0 & ; a < 0 \end{cases} \qquad \text{(Ec. 7)}$$

Donde m_k es la k-ésima respuesta direccional más significativa.

Para ilustrar este proceso de forma gráfica, se muestra el proceso seguido para transformar la máscara de la respuesta en la cadena de bits del LDP y posteriormente, se incluye un ejemplo de cómo calcular un código LDP con $k = 3$.

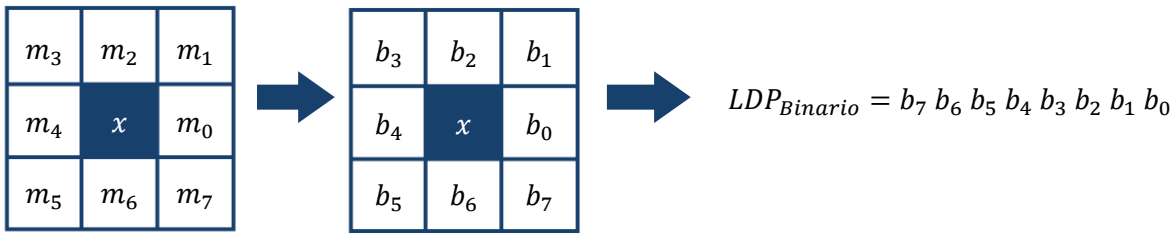


Figura 17. Proceso para calcular el código binario del LDP

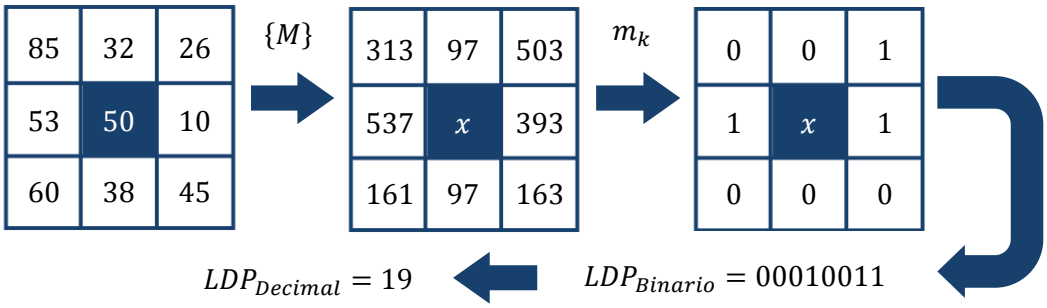


Figura 18. Ejemplo de cálculo del LDP

Un inconveniente que presenta el *Local Directional Pattern* es que la rotación o giro de cualquier imagen altera la distribución de la intensidad espacial de la misma y como consecuencia, los valores de respuesta en los bordes cambian en todas las direcciones y se genera un código LDP completamente distinto.

No obstante, tras la observación de los valores de las respuestas se ha visto que la posición relativa de estos valores, si nos centramos en las respuestas más fuertes, no se ven afectadas por la rotación de la imagen. Por ejemplo, consideremos la imagen con los ocho valores de respuesta de los bordes que se indican en la Fig. 17 (a). Si la imagen se gira 90° en sentido antihorario, el valor de la intensidad de esa imagen cambia, lo cual lleva a un cambio en los valores de respuesta de los bordes, como podemos ver en la Fig. 14 (b). Si comparamos estas dos figuras, se comprueba que los valores de la respuesta de los bordes simplemente varían su dirección 90° en sentido antihorario.

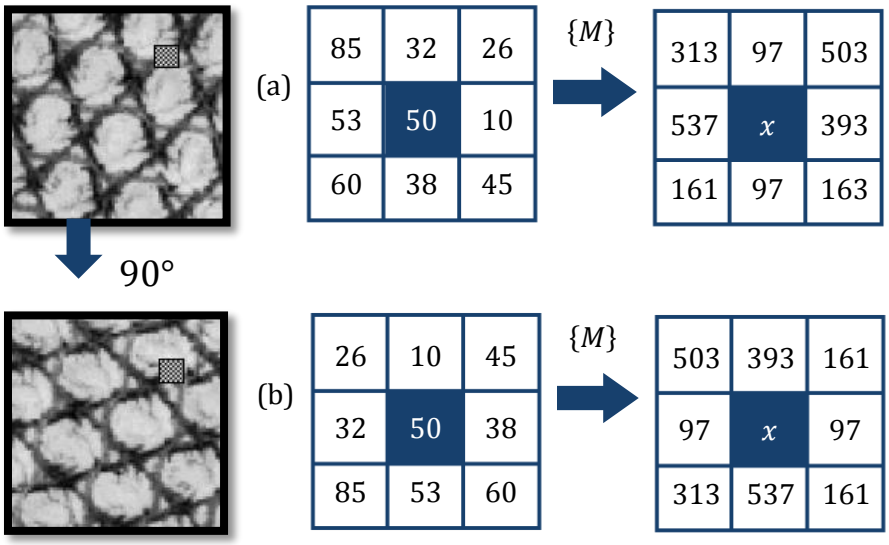


Figura 19. Respuestas de bordes antes (a) y después (b) de la rotación de la imagen

Debido a esta observación, se propuso un método simple para lograr que el LDP fuera invariante a la rotación. Esto se consiguió mediante la aplicación de una operación de desplazamiento circular sobre el valor del código binario original. Aplicado al ejemplo anterior, el resultado sería:

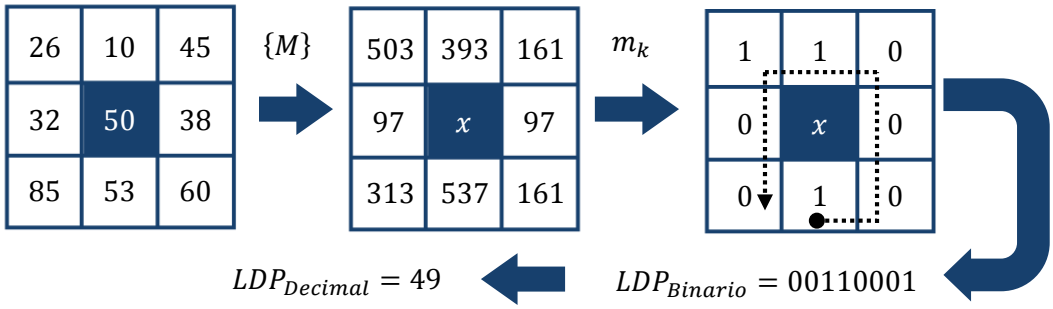


Figura 20. Ejemplo de cálculo del LDP invariante a la rotación

Para que este cambio se complete con éxito, la dirección de la respuesta de borde más alta pasa a ser la dirección dominante del código LDP. El valor del bit asociado a la dirección dominante se establece como el bit menos significativo del código. Y posteriormente, los demás bits se desplazan circularmente el mismo número de posiciones que la dirección dominante, obteniéndose así el código final. Por ejemplo, si en el código original la posición de los bits es "abcdefgh" y el bit de dominante es "c", entonces el valor del bit de la posición "c" debe moverse cinco posiciones para situarse como el que está más a la derecha. Así que todos los demás bits también se mueven cinco posiciones circularmente, y el código normalizado resulta entonces "defghabc". Da igual cuánto se gire la imagen, si seguimos este procedimiento, la cadena de bits final será siempre la misma.

Con esta modificación se consigue generar códigos LDP invariantes con la rotación, esta variante se denomina LDP^{ri} y su expresión matemática es:

$$LDP^{ri} = ROR(LDP, d-1) \qquad \text{(Ec. 8)}$$

Donde d es la posición del bit con la mayor respuesta de borde.

Esta mejora del LDP no se contempla en este proyecto.

3.2.3.2. Programación

El primer paso seguido a la hora de extraer los códigos LDP de las imágenes de nuestra base de datos, a fin de reconocer la escritura de las mismas, consistió en aplicar las ocho máscaras de Kirsch a la imagen de análisis ($reglone$):

```
reglonldp=zeros(8,nf,nc);
reglonldp(1,:,:)=filter2([-3 -3  5;-3 0  5;-3 -3  5],reglone);
reglonldp(2,:,:)=filter2([-3  5  5;-3 0  5;-3 -3 -3],reglone);
reglonldp(3,:,:)=filter2([ 5  5  5;-3 0 -3;-3 -3 -3],reglone);
reglonldp(4,:,:)=filter2([ 5  5 -3; 5 0 -3;-3 -3 -3],reglone);
reglonldp(5,:,:)=filter2([ 5 -3 -3; 5 0 -3; 5 -3 -3],reglone);
reglonldp(6,:,:)=filter2([-3 -3 -3; 5 0 -3; 5  5 -3],reglone);
reglonldp(7,:,:)=filter2([-3 -3 -3;-3 0 -3; 5  5  5],reglone);
reglonldp(8,:,:)=filter2([-3 -3 -3;-3 0  5;-3  5  5],reglone);
```

A continuación, se calcularon los códigos LDP colocando cada uno de los píxeles $Z = (ifil, icol)$ de la imagen como centro. Además, hemos de tener en cuenta que se trabajó con vecindades de 3x3 píxeles, por lo que se realizó el cálculo considerando $k = 3$.

Los códigos LDP se generaron mediante la localización de los k bits dominantes, tal que a éstos se les asignó un "1" y al resto, un "0". Una vez obtenido el código LDP en binario, se pasó a decimal para que su valor fuera asignado al píxel de análisis, tal y como puede verse en las líneas de código siguientes:

```
mvldp=reglonldp(:,ifil,icol);
[Y,rank]=sort(abs(mvldp));
codebitldp=zeros(1,8);
codebitldp(rank(end:-1:end-2))=1;
Ildp(ifil,icol)=bi2de(codebitldp);
```

3.2.4. Local Derivative Pattern (LDerivP)

El ***Local Derivative Pattern*** (LDerivP) codifica información direccional en la vecindad de un píxel. Permite capturar información discriminativa más detallada que la obtenida usando los operadores anteriores acerca de la estructura local. Esto es posible debido a que realiza el cálculo de derivadas locales direccionales de órdenes elevados [31].

3.2.4.1. Definición

En este Proyecto de Final de Carrera hemos considerado el enfoque propuesto en el trabajo "*Local derivative pattern versus local binary pattern: face recognition with high-order local pattern descriptor*", en el cual se investiga la viabilidad y eficacia de la utilización de los patrones locales de orden elevado para la representación de caras. Estudio en el que se propone un operador que cuantifica las variaciones direccionales de las derivadas de orden

elevado basándose en una función de codificación binaria, el cual se denomina *Local Derivative Pattern*.

Así, dada una imagen $I(Z)$, denotaremos las derivadas de primer orden a lo largo de las direcciones $0°, 45°, 90°\ y\ 135°$ como $I'_\alpha(Z)$ donde $\alpha = 0°, 45°, 90°\ y\ 135°$. Las cuatro derivadas de primer orden en $Z = Z_0$ se pueden escribir como:

$$I'_{0°}(Z_0) = I(Z_0) - I(Z_4) \quad \text{(Ec. 9)}$$

$$I'_{45°}(Z_0) = I(Z_0) - I(Z_3) \quad \text{(Ec. 10)}$$

$$I'_{90°}(Z_0) = I(Z_0) - I(Z_2) \quad \text{(Ec. 11)}$$

$$I'_{135°}(Z_0) = I(Z_0) - I(Z_1) \quad \text{(Ec. 12)}$$

Siendo Z_0 el píxel central y Z_i con $i = 1,2, \dots, 8$, los píxeles que lo rodean. De forma que el área de píxeles queda numerada como:

Z_1	Z_2	Z_3
Z_8	Z_0	Z_4
Z_7	Z_6	Z_5

Figura 21. Ejemplo de vecindad de 8 píxeles alrededor de Z_0

Haciendo uso de estas expresiones, el LDerivP direccional de segundo orden ($LDerivP^2_\alpha(Z_0)$), en la dirección α y en el punto $Z = Z_0$ se define como:

$$LDerivP^2_\alpha(Z_0) = \left\{f\left(I'_\alpha(Z_0), I'_\alpha(Z_1)\right), f\left(I'_\alpha(Z_0), I'_\alpha(Z_2)\right), \dots, f\left(I'_\alpha(Z_0), I'_\alpha(Z_8)\right)\right\} \quad \text{(Ec. 13)}$$

Donde $f(\dots, \dots)$es una función de codificación binaria para determinar los tipos de transiciones de patrones locales. Este elemento codifica la coocurrencia de las dos direcciones de las derivadas en diferentes píxeles vecinos de forma que:

$$f\left(I'_\alpha(Z_0), I'_\alpha(Z_1)\right) = \begin{cases} 0, & si\ I'_\alpha(Z_i) \cdot I'_\alpha(Z_0) > 0 \\ 1, & si\ I'_\alpha(Z_i) \cdot I'_\alpha(Z_0) \leq 0 \end{cases} \quad ; i = 1, 2, \dots, 8. \quad \text{(Ec. 14)}$$

Considerando todo lo anterior, el *Local Derivative Pattern* de segundo orden ($LDerivP^2(Z)$) se crea concatenando los cuatro LDerivP direccionales de 8 bits resultantes del análisis en cada dirección α:

$$LDerivP^2(Z) = \left\{LDerivP_\alpha^2(Z)|_{\alpha=0°,45°,90°,135°}\right\} \quad \text{(Ec. 15)}$$

Cabe mencionar que es posible generalizar el orden del LDerivP y así trabajar a órdenes aún mayores.

Las comparaciones entre las derivadas direccionales definidas en (Ec. 13) se ilustran en las 16 plantillas de la Figura 20, las cuales reflejan varias relaciones espaciales distintivas en una región local. Las primeras cuatro plantillas (a-i), se refieren a la dirección 0°, tal que (a-1) permite calcular $f\left(I_{0°}^{n-1}(Z_0), I_{0°}^{n-1}(Z_1)\right)$ y $f\left(I_{0°}^{n-1}(Z_0), I_{0°}^{n-1}(Z_5)\right)$, con (a-2) se calcula $f\left(I_{0°}^{n-1}(Z_0), I_{0°}^{n-1}(Z_2)\right)$ y $f\left(I_{0°}^{n-1}(Z_0), I_{0°}^{n-1}(Z_6)\right)$, (a-3) está asociada con el cálculo de $f\left(I_{0°}^{n-1}(Z_0), I_{0°}^{n-1}(Z_3)\right)$ y $f\left(I_{0°}^{n-1}(Z_0), I_{0°}^{n-1}(Z_7)\right)$ y finalmente, (a-4) sirve para calcular $f\left(I_{0°}^{n-1}(Z_0), I_{0°}^{n-1}(Z_4)\right)$ y $f\left(I_{0°}^{n-1}(Z_0), I_{0°}^{n-1}(Z_8)\right)$.

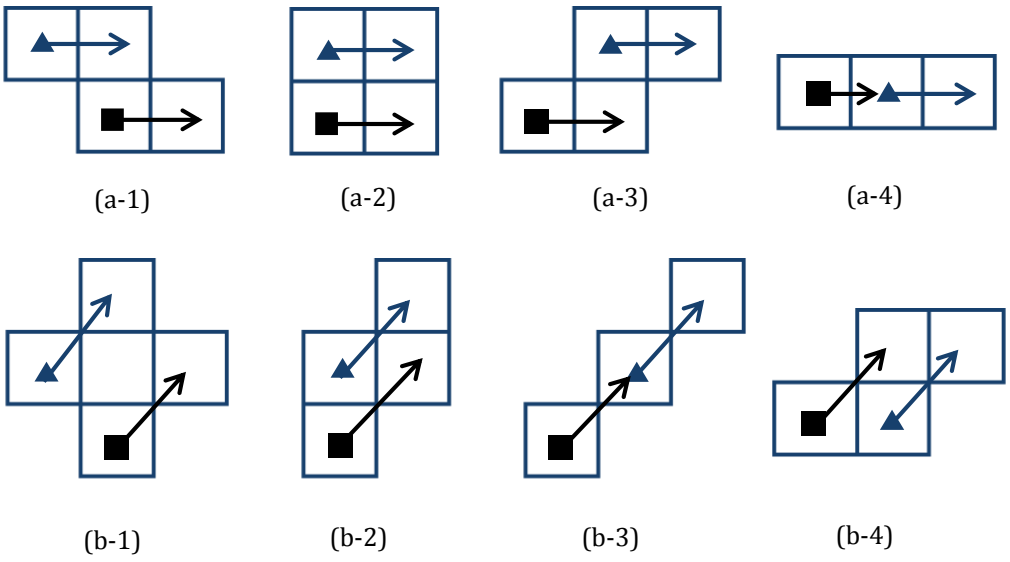


Figura 22- 1. Ilustraciones de las ocho primeras plantillas LDerivP

Las siguientes cuatro plantillas (b-i) trabajan sobre la dirección 45° y permiten realizar los mismo cálculos ya expuestos. De la misma forma, las plantillas (c-i) se aplican a 90° y el último conjunto (d-i) a 135°. En todas ellas, el cuadrado representa el primer punto de referencia, y el triángulo el segundo.

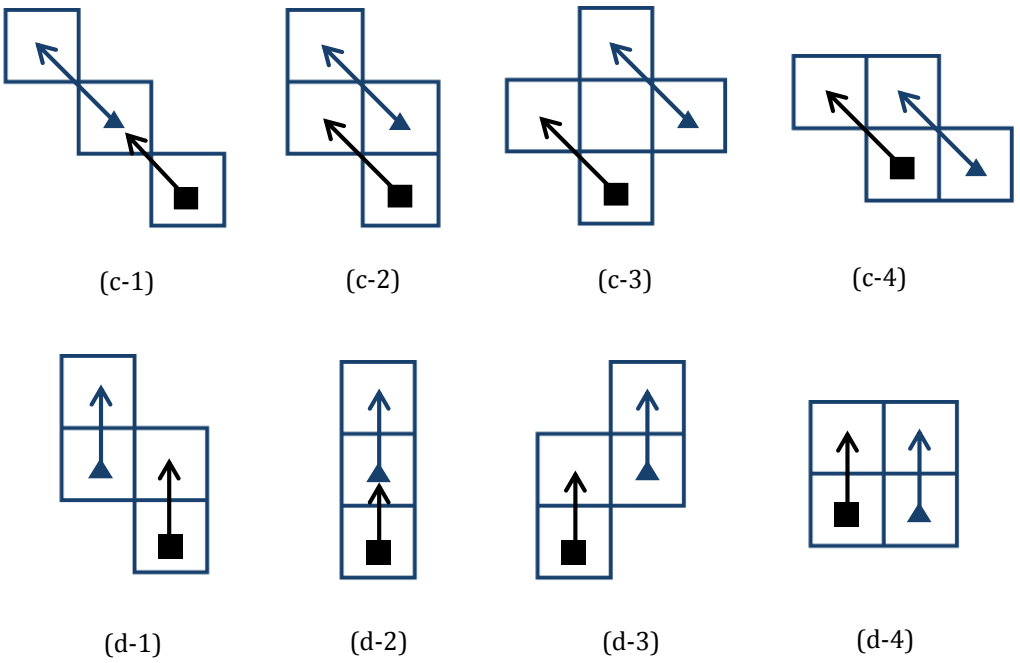


Figura 22-2. Ilustraciones de las ocho plantillas restantes de LDerivP

Cada una de las plantillas que se incluyen en la Fig. 20 pueden clasificarse en función del número de puntos que contengan, que pueden ser 3 (Fig 21-a) o 4 puntos (Fig 21-b).

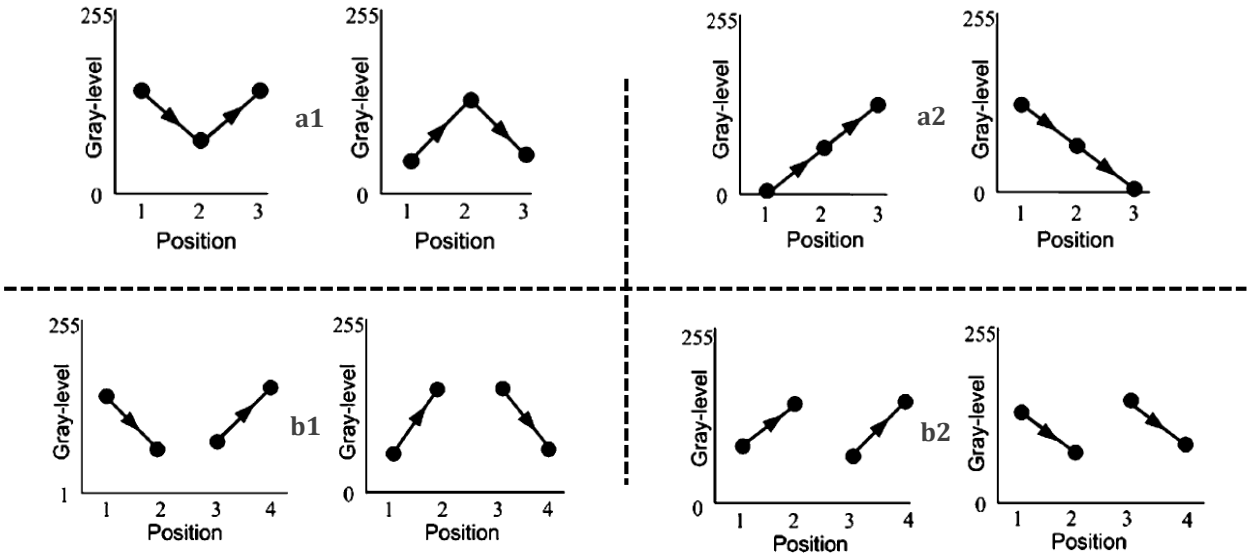


Figura 23. Significados de los patrones denotados con "0" y "1" por el LDerivP de segundo orden.

En el caso de una plantilla de 3 puntos y atendiendo a la expresión (6), asignamos un "0" si se sigue un patrón monótonamente creciente o decreciente (Fig 21_1-a1), y señalizamos con un "1" aquellos patrones que contengan un "punto de inflexión" (Fig 21_1-

a2). Del mismo modo, para una plantilla de 4 puntos, si el patrón no es monótonamente creciente (Fig. 21_1-b1) se etiqueta como "1" y por el contrario, si lo es, se le asigna un "0" (Fig. 21_1-b2).

El proceso para calcular el LDerivP de segundo orden o $LDerivP_{\alpha}^{2}(Z_0)$, es similar en todas las direcciones y sobre cualquier punto. Para entender mejor cómo se lleva a cabo este cálculo, vamos a resolver un ejemplo y nos centraremos en hallar su valor en la dirección $\alpha = 0°$ y en el punto $Z = Z_0$ resaltado en el centro de la imagen.

2	5	3	5	1
6	7	9	1	5
2	3	4	8	2
3	2	3	2	9
1	2	3	2	1

Figura 24-a. Ejemplo de cálculo del operador LDerivP de segundo orden.

En primer lugar, se escoge la plantilla adecuada según la dirección que se pretenda analizar. En este caso, al estar trabajando con $\alpha = 0°$, se utilizan las cuatro primeras plantillas que se incluyen en la Fig. 20 (a) y éstas se aplican sobre la imagen alineando las dos referencias con el píxel central.

Así, al aplicar la plantilla (a-1) y tomar el punto Z_0 como primera referencia, dado que los valores que indican las flechas son mayores a los que se encuentran en el punto de origen de las mismas, se da un aumento monótono que coincide con el caso visto en la figura (14-b2) y en consecuencia, se asigna un “0” a este patrón.

Por otro lado, si aplicamos la plantilla (a-2) comprobamos que no existe este aumento, sino que las flechas irían en sentido contrario y en consecuencia, se asigna un “1” al patrón. Este proceso se repite con el resto de las plantillas y se analiza el comportamiento de los patrones para crear finalmente una cadena binaria con los resultados. En este caso, el valor la cadena final es: "01010100" para la dirección 0°.

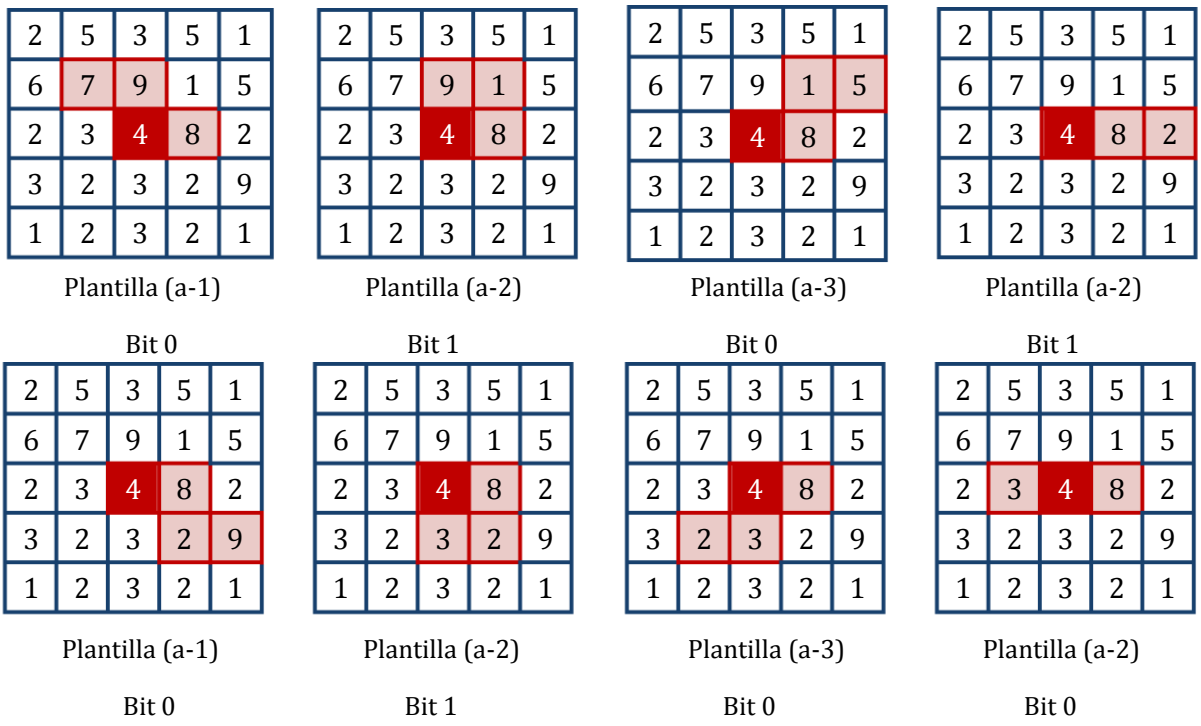


Figura 24-b. Aplicación de plantillas para el cálculo del operador LDerivP de segundo orden.

Una vez hecho esto, se siguen los mismos pasos en el resto de direcciones para así obtener las cadenas binarias en cada caso. Por último, se concatenan todos los cuatro resultados finales de 8 bits cada uno, para crear una cadena final de 32 bits, como se indica en la expresión (7). El resultado de este último paso se ilustra en la siguiente figura:

$$\left.\begin{aligned} LDP^2_{0°}(Z_0) &= 01010100 \\ LDP^2_{0°}(Z_0) &= 00101111 \\ LDP^2_{0°}(Z_0) &= 11010000 \\ LDP^2_{0°}(Z_0) &= 11000110 \end{aligned}\right\} \quad LDP^2(Z_0) = 01010100001011111101000011000110$$

3.2.4.2. Programación

El primer paso que se lleva a cabo consiste en hallar las derivadas de primer orden ($reglonLDerivP$), a partir de diferencias entre píxeles de una imagen ($reglone$) en las cuatro direcciones $0°, 45°, 90°$ y $135°$:

```
reglonLDerivP(2:end-1,2:end-1,1)=reglone(2:end-1,2:end-1)-reglone(2:end-1,3:end);
reglonLDerivP(2:end-1,2:end-1,2)=reglone(2:end-1,2:end-1)-reglone(1:end-2,3:end);
reglonLDerivP(2:end-1,2:end-1,3)=reglone(2:end-1,2:end-1)-reglone(1:end-2,2:end-1);
reglonLDerivP(2:end-1,2:end-1,4)=reglone(2:end-1,2:end-1)-reglone(1:end-2,1:end-2);
```

Una vez hecho esto, se toman conjuntos de píxeles de tamaño 3x3 alrededor de un punto $Z = (ifil, icol)$. En cada uno de estos grupos, se asigna un "0" o un "1" cuando corresponda y se genera un vector de 8 bits por cada una de las direcciones. Finalmente, se convierte cada vector a un término decimal y éste pasa a ser el valor del píxel central de análisis, formando así una nueva imagen filtrada ($ILDerivP$):

```
for iLDerivP=1:4
    mv=reglonLDerivP(ifil-1:ifil+1,icol-1:icol+1,iLDerivP);
    mv=(mv.*mv(2,2))<=0;
    codebit=mv([8 7 4 1 2 3 6 9]);
    ILDerivP(ifil,icol,iLDerivP)=bi2de(codebit);
end
```

3.3. El clasificador LS-SVM

3.3.1. La Máquina de Vectores de Soporte (SVM)

La teoría de las ***Máquinas de Vectores de Soporte*** (SVM por su nombre en inglés *Support Vector Machine*) fue desarrollada por Vapnik basándose en la idea de minimización del riesgo estructural. Los SVM se usan en reconocimiento de firmas o imágenes, entre muchas otras aplicaciones, como puede ser la identificación del tipo de escritura [32, 33, 34, 35].

El clasificador SVM mapea los puntos de entrada a un espacio de características de una dimensión mayor, para luego encontrar el hiperplano óptimo que los separe y maximice el margen entre las clases. Estos sistemas se presentan como una técnica robusta para proceso de clasificación y regresión aplicadas a grandes conjuntos de datos complejos con ruido. Esto quiere decir que trabajan con variables inherentes al modelo, que en el caso de otros métodos aumentan la posibilidad de error en los resultados, pues resultan difíciles de cuantificar y observar.

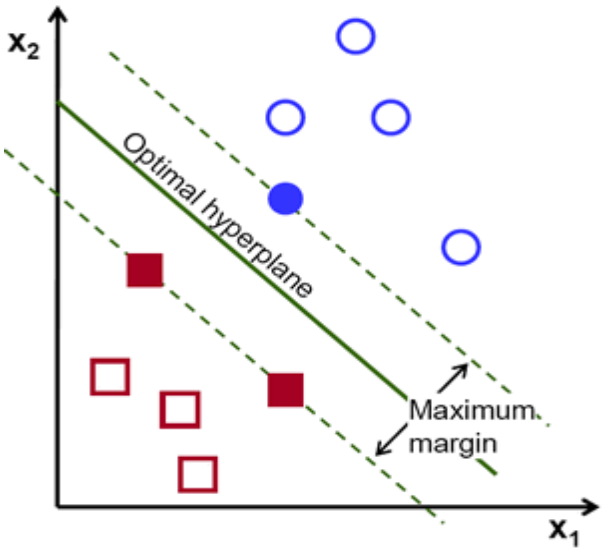


Figura 25. Representación del funcionamiento de un SVM, que coloca un hiperplano que separa dos clases distintas.

Las Máquinas de Vectores Soporte tienen ciertas características que han hecho que resulten más adecuados que otras técnicas populares de clasificación. Una de las más notables es que las SVM pertenecen a la categoría de aprendizaje automático o estadístico. Este tipo de aprendizaje se basa en el propósito de hacer que las máquinas puedan ir aprendiendo, a través de ejemplos, qué salidas son correctas para qué entradas.

Otro aspecto a mencionar, es que las SVM se diferencian de otros algoritmos de aprendizaje en que aplican un nuevo principio inductivo, buscando la minimización del riesgo estructural. Además, emplean una función núcleo o kernel, la cual le atribuye una gran capacidad de generalización, incluso cuando el conjunto de entrenamiento es pequeño. Por consiguiente, tanto la capacidad de generalización como el proceso de entrenamiento de la máquina no dependen necesariamente del número de atributos, y ello hace que sean la elección correcta en problemas de alta dimensionalidad.

Sin embargo, uno de los inconvenientes más importantes de este tipo de clasificador y que debemos tener en consideración es lo que se conoce como sobreentrenamiento. Esta situación se da, como su nombre indica, cuando se ha entrenado un sistema más de lo debido, por lo que el aprendizaje de los datos de entrenamiento resulta tan estricto que no se pueden clasificar bien aquellos ejemplos nunca antes vistos por el sistema, es decir, se produce una mala generalización del modelo.

Asimismo, las SVM pertenecen a la familia de clasificadores lineales, puesto que introducen separadores lineales o hiperplanos en espacios de características de muy alta dimensionalidad. Esto no quiere decir que su uso se limite únicamente a la clasificación en

problemas donde las clases puedan ser separadas linealmente, las SVM pueden ser utilizadas tanto para separar varias clases de datos que puedan llevarse a cabo linealmente, como en aquellos caso donde no pueda llevarse a cabo un enfoque lineal y esa es una de sus principales características.

Por un lado, si las clases de análisis son linealmente separables, las SVM forman hiperplanos que separan los datos de entrada en dos o más subgrupos que poseen una etiqueta propia. Existe sólo un hiperplano de separación óptimo, de forma que la distancia entre el mismo y el valor de entrada más cercano es máxima, lo que permite forzar la generalización de la máquina que se esté construyendo. Aquellos puntos sobre los cuales se apoya el margen máximo son los denominados vectores de soporte.

Por otro lado, si no son linealmente separables, como es el caso de los tipos de escritura, existen dos posibles enfoques según la problemática a la que nos enfrentemos. El primer caso ocurre cuando los datos pueden ser separables maximizando el margen pero en un espacio de características con mayor dimensionalidad, el cual se obtiene a través de la trasformación de las variables del espacio de entrada usando una función kernel. El segundo caso es el denominado “Soft Margin” o margen blando, y se da cuando no es posible encontrar una trasformación de los datos que permita separarlos linealmente, bien sea en el espacio de entrada o en el espacio de características.

3.3.2. La función kernel, la búsqueda de rejilla y la validación cruzada.

Desde que las SVM empezaron a ganar popularidad como sistemas de clasificación, se han creado y aplicado multitud de técnicas basadas en el truco del kernel o el *kernel trick*. Toda esa familia de técnicas se denomina Métodos de Kernel [37].

El *Kernel Trick* o truco del kernel se ha utilizado para obtener extensiones no lineales de otros algoritmos, cómo por ejemplo los algoritmos de clustering o los k-vecinos más cercanos. Los métodos así obtenidos son independientes de la estructura concreta de los datos de entrada y, por lo tanto, una vez definida la función de kernel, la complejidad computacional del sistema es independiente del número de variables de entrada, ya que pasan a estar definidas en base a valores resultados de la evaluación de una función sobre pares de datos de entrada.

Así, se conoce como función núcleo o kernel a un producto interno en el espacio de características, que tiene su equivalente en el espacio de entrada. Entre los kernels más

comunes, se encuentran: la función lineal, polinomial, RBF (Radial Basis Function), ERBF (Exponential Radial Basis Function), entre otros.

En este proyecto se trabaja principalmente con el kernel RBF. El kernel gaussiano o RBF es muy popular y tiene un buen comportamiento especialmente en ausencia de conocimiento sobre los datos y el dominio. Estas funciones se consideran modelos no paramétricos, lo cual significa que la complejidad del modelo es potencialmente infinita porque el número de funciones analíticas son infinitas.

El kernel RBF se define mediante la expresión:

$$K\ (x,z)\ =\ exp\ (-\frac{\|x-y\|^2}{2\sigma^2}) \qquad \text{(Ec. 16)}$$

Pudiéndose escribir como:

$$K\ (x,z)\ =\ exp\ (-d\ (x,z)^2\ \gamma) \qquad \text{(Ec. 17)}$$

Donde: $\gamma\ =\ {}^{1}/_{2\sigma^2}$

En estas expresiones, σ es el ancho del kernel. Un elevado parámetro σ implica a su vez una baja γ, y ello conlleva un ajuste más suave y por lo tanto, los puntos de las vecindades pasan a tener una influencia mayor y viceversa. Cuando $lím_{\sigma \to 0}$, significa que un punto no está influenciado o correlado con cualquier punto.

Cuando se emplea el kernel RBF, el clasificador es descrito mediante superficies en forma de campana cuyo centro se sitúa en cada vector de soporte y la anchura de cada superficie es directamente proporcional a σ. El valor óptimo de σ debe estar en algún lugar entre la distancia mínima y máxima entre pares de datos, y dicha distancia es dependiente de los datos.

Una alternativa muy generalizada a la hora de aproximar el valor de γ es mediante la búsqueda de rendija o grid-search y la validación cruzada (cross-validation), que es la metodología seguida en este proyecto.

La búsqueda de rejilla (grid-search) consiste en buscar los γ más adecuados para clasificar con precisión datos desconocidos y se aplica validación cruzada sobre los datos de entrenamiento con objeto de evitar el sobreentrenamiento.

La validación cruzada (cross-validation), por su parte, es una técnica utilizada para evaluar los resultados de un análisis estadístico y garantizar que son independientes de la

división entre datos de entrenamiento y test. Se basa en repetir y calcular la media aritmética obtenida de las medidas de evaluación sobre diferentes particiones.

En concreto, en la validación cruzada de K iteraciones o *K-fold cross-validation* los datos de muestra se dividen en K subconjuntos. Uno de los subconjuntos se utiliza como datos de prueba y el resto $(K-1)$ como datos de entrenamiento. El proceso de validación cruzada es repetido durante k iteraciones, con cada uno de los posibles subconjuntos de datos de test. Finalmente se realiza la media aritmética de los resultados de cada iteración para obtener un único resultado. Este método es muy preciso, puesto que la evaluación se realiza a partir de K combinaciones de datos de entrenamiento y test, pero aun así tiene una desventaja, y es que es lento desde el punto de vista computacional.

En la práctica, la elección del número de iteraciones (K) depende de la medida del conjunto de datos. Lo más común es utilizar la validación cruzada de 10 iteraciones (*10-fold cross-validation*), la cual empleamos en este trabajo para la selección de los parámetros de la secuencia de entrenamiento.

3.3.3. La Máquina de Vectores de Soporte de Mínimos Cuadrados (LS-SVM)

La ***Máquina de Vectores de Soporte de Mínimos Cuadrados*** (LS-SVM) es una reformulación del SVM estándar que resuelve los problemas de clasificación en sistemas lineales indefinidos [38, 39].

Para diferenciar las SVM originales de las LS-SVM debemos identificar la forma mediante la cual se encuentra el hiperplano óptimo que maximice las márgenes entre éste y los puntos de ambas clases dentro del conjunto de datos de entrada. Así, la SVM original soluciona dicho problema mediante el principio de *structural risk minimization* que es un principio inductivo para la selección del modelo utilizado para el aprendizaje del sistema a partir de conjuntos de datos de entrenamiento finitos; mientras que los sistemas LS-SVM emplean un conjunto de ecuaciones lineales como solución a esta problemática. Por otro lado, en el SVM tradicional muchos valores de soporte son cero, tal que los valores diferentes de cero se corresponden a los vectores de soporte; mientras que en la LS-SVM los valores de soporte son proporcionales a los errores.

El LS-SVM ha sido una propuesta muy generalizada en el campo del procesamiento de imágenes, aplicándose en infinidad de experimentos como son: el análisis de imágenes médicas diagnósticas, tales como MRI (Magnectic Resonance Imaging) y MRS (Magnetic

Resonanc Spectroscopy), la interpretación de imágenes provenientes de datos genéticos, el tratamiento de imágenes en el ámbito de la seguridad, como la detección de objetos en imágenes infrarrojas o la detección de rostros, entre muchas otras aplicaciones.

En resumen, en este trabajo se emplea un clasificador LS-SVM para llevar a cabo la identificación del tipo de escritura, usando para ello el kernel RBF, la búsqueda de rejilla y la validación cruzada de 10 iteraciones obteniendo, como veremos en capítulos posteriores de esta memoria, buenos resultados. Cabe mencionar que además, se han adoptado técnicas de uno-contra-resto, las cuales consisten en entrenar k clasificadores, tal que una clase específica es la positiva y el resto de clases pasan a ser una clase negativa, y se predice la clase para todos los clasificadores, de forma que la clase asignada es aquella con la que se consigue margen mayor, es decir, aquella que haya sido considerada como positiva en más de un clasificador.

3.4. Magnitudes y curvas de medida de evaluación de los resultados obtenidos.

En este apartado se enumeran las magnitudes y las gráficas o curvas que han sido empleadas en el proyecto para evaluar el rendimiento de los sistemas desarrollados.

Tasa de identificación (Identification Rate)

Valor que cuantifica cuántas muestras que forman parte de la base de datos son correctamente identificadas.

Tasa de falsa aceptación (FAR – False Acceptance Rate):

Se trata de una medida estadística que mide cuántas veces el sistema produce una falsa aceptación. En nuestro trabajo, este parámetro da idea de cuántas muestras de escritura son erróneamente aceptadas al realizar una comparación. Se representa de forma gráfica mediante la curva FAR.

Tasa de falso rechazo (FRR – False Rejection Rate):

Esta medida estadística también se emplea para medir el rendimiento del sistema y representa el porcentaje de veces que el sistema produce un falso rechazo. Esto ocurre cuando una muestra de escritura no es reconocida correctamente, siendo rechazada en el proceso de comparación con su propio alfabeto. Se expresa gráficamente por la curva FRR.

Tasa de igual error (EER – Equal Error Rate)

La tasa EER se obtiene de la combinación de los dos parámetros anteriores, siendo el punto de corte entre las curvas FRR y FAR. Por lo general, cuánto más bajo sea el valor de la tasa de igual error, mayor será la precisión del sistema pues eso significa que se produce un número menor de falsas aceptaciones y falsos rechazos. Se representa gráficamente mediante la curva EER.

Rango–n:

Medida característica de los procesos de identificación que indica la capacidad que tiene el sistema para identificar los distintos tipos de escritura, según el puesto en el que localice el sistema dicha clase a reconocer. De esta forma se puede obtener el grado de fiabilidad del sistema, en base a localizar la escritura en las 5 primeras posiciones, las 10 primeras, las 20 primeras etc.

Característica de funcionamiento del receptor (ROC – Receiver Operating Characteristics)

Se trata de un método para cuantificar el rendimiento en términos de precisión en la medida de un sistema de identificación. Las curvas ROC comparan la tasa de falsa aceptación con la tasa de verificación, es decir, comparan las veces que una muestra se acepta correctamente con las veces en que su aceptación no debería haberse producido.

Característica de coincidencia acumulativa (CMC – Cumulative Match Characteristic)

La característica CMC muestra la frecuencia con la cual aparece la clase correcta de la muestra en los distintos rangos (1, 5, 10, 100, etc.), según la tasa de coincidencia, que viene representada por la tasa de identificación en nuestro trabajo.

Para hacer más clara esta explicación, supongamos cuatro tipos de escritura distintos o clases. Si comparamos una imagen de texto con cada uno de los modelos generados con los patrones locales, obtendremos una puntuación que se corresponde con la tasa de identificación en cada caso.

Así, si al realizar la prueba logramos una puntuación superior a las demás en una de esas clases, habremos reconocido el tipo de escritura en Rango 1 indicando que se trata del primer valor más alto. Sin embargo, si obtenemos dos porcentajes altos y cercanos en varias clases y el mayor de ellos no es el correcto, estaríamos obteniendo un falso positivo o un

caso de falsa aceptación, por lo que el rango en este caso se correspondería con el orden de la clase correcta sería Rango 2.

Las curvas CMC se construyen teniendo en cuenta la tasa de identificación global en cada rango.

Capítulo 4. Estudio de laboratorio

4.1. Introducción

En este capítulo y los siguientes, se describirán los procedimientos desarrollados en este proyecto para la implementación de un sistema de reconocimiento de escritura haciendo uso de los patrones locales y el clasificador ya explicados.

Al empezar con este proyecto decidimos realizar un estudio inicial con el propósito de comprobar la viabilidad de las técnicas de análisis de texturas que habíamos seleccionado para la identificación del tipo de texto, dado que nunca habían sido empleadas con este fin. Para ello, fue necesaria la conformación de una base de datos completa desde cero, la cual debía contener muestras de distintos tipos de escritura del mundo, pues no existe ningún conjunto de imágenes de esta naturaleza para ser usada con fines científicos. Además, se crearon múltiples programas para así poder extraer los parámetros de cada alfabeto involucrado en nuestro estudio y proceder a la clasificación y evaluación del sistema completo.

En los puntos siguientes se explicará cómo se han confeccionado las bases de datos y cuáles han sido los procedimientos seguidos para desarrollar el primer sistema llevado a cabo durante este proyecto. Finalmente, se presentarán los resultados obtenidos tras la aplicación de dicho sistema sobre las imágenes de texto que forman parte de las bases de datos.

4.2. Diseño de la base de datos

En el estudio inicial se confeccionaron dos bases de datos distintas. La primera de ellas debía contener al menos diez tipos de escritura diferentes, para así verificar la aptitud de nuestro sistema. Ésta estaba compuesta, en primer lugar, de algunos tipos de escritura europeos, incluimos por tanto la escritura latina, que es la más habitual entre los lenguajes de la zona, el ruso y el griego. Por otro lado, dado que la India es el país donde este sistema resultaría más útil, se añadieron muchos de los principales idiomas de esta región: el hindi, el bengalí, el gujarati, el canarés, el telugu, el tamil y el urdu. Finalmente, se optó por incluir también escrituras asiáticas populares, en concreto, se incorporaron el japonés y el árabe. La segunda base de datos nos fue facilitada por el *Indian Statistics Institute* de Calcuta, y estaba formada por algunos documentos redactados en escrituras hindúes, en concreto, se incluían muestras en bengalí, canarés, gujarati, hindi y urdu.

La primera base de datos, fue realizada en el laboratorio, como parte de este proyecto y se usó tanto para entrenar como para probar el sistema de identificación. La segunda sin embargo, se dedicó únicamente al proceso de test, con el objetivo de corroborar que los modelos obtenidos entrenando con la primera eran adecuados para caracterizar el tipo de escritura, sin necesidad de mantener las mismas fuentes o estilos de texto.

4.3. Elaboración de la base de datos

En la confección de la primera base de datos se utilizó el traductor de Google como herramienta principal. Para obtener las imágenes de texto, se procedió como sigue:

1. En primer lugar, se seleccionaron catorce novelas universales como *Orgullo y Prejuicio* o *Robinson Crusoe* en versión digital, y seis artículos sobre tecnología de revistas y periódicos digitales. De cada una de estas fuentes, se tomó una sección de texto de más de cinco líneas.
2. Seguidamente, estos bloques de texto se introdujeron en el traductor de Google y se tradujeron de su idioma original (español o inglés) a todos los lenguajes mencionados en el punto anterior.
3. El texto obtenido de la traducción se copió a un editor de texto para posteriormente ser convertido a imagen mediante una captura de pantalla. Las imágenes resultantes fueron almacenadas en formato TIFF.

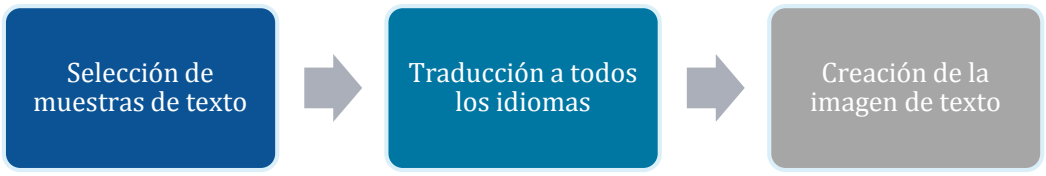


Figura 26. Diagrama de flujo para la creación de la primera base de datos del estudio inicial.

Así, cada una de las escrituras de análisis contaba con un directorio propio, y contenía veinte bloques de texto distintos en total. Cada una de las muestras de los distintos tipos de escritura se nombraba haciendo alusión a la escritura en que se redactaban, el título del documento del que fueron extraídos y se les asignaba también un índice numérico, de forma que todas las imágenes con el mismo índice coincidían en contenido pero se encontraban traducidas al idioma pertinente.

A continuación, se presenta una figura que ejemplifica una misma muestra de texto traducida a cada uno de los tipos de escritura que forman la primera base de datos. Todas ellas se han creado a partir de un fragmento de la novela *Robinson Crusoe*:

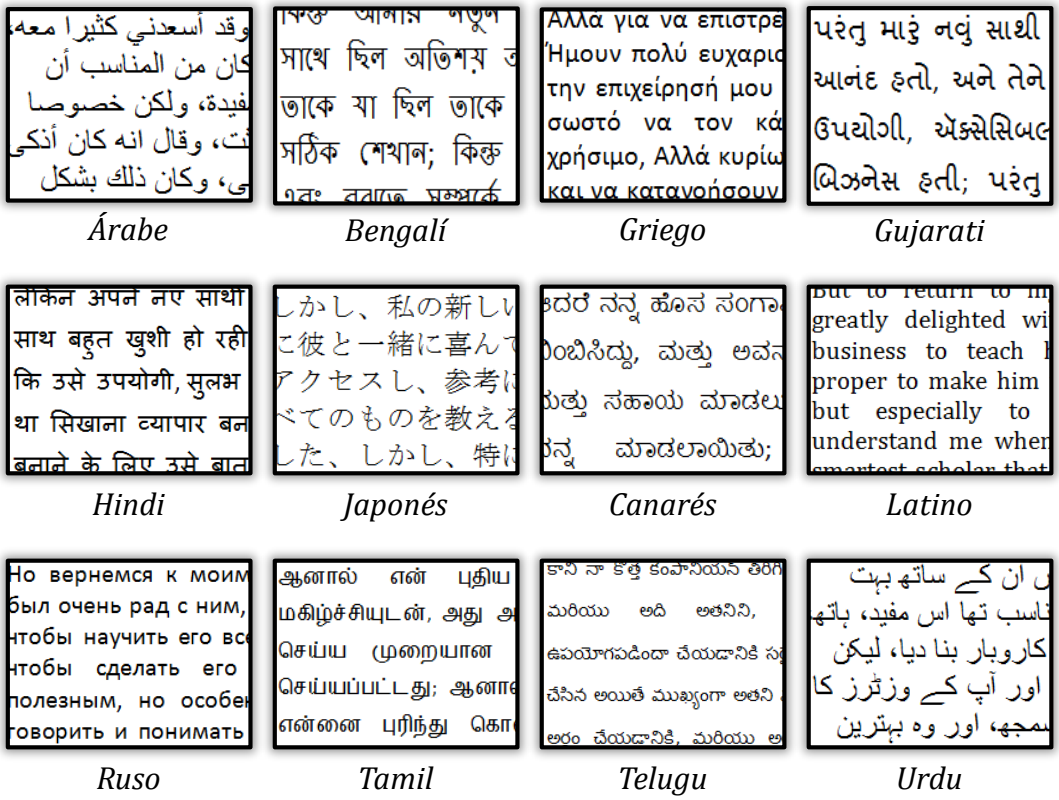


Figura 27. Muestras de los distintos tipos de escritura incluidos en la primera base de datos.

La segunda base de datos, por otro lado, guarda una importante diferencia con respecto a la anterior, y es que todas las imágenes que forman parte de la misma fueron escaneadas directamente desde libros, periódicos y otras fuentes físicas a 150 dpi. Como ya se ha comentado en el punto anterior, esta base de datos nos fue facilitada por el *Indian Statistical Institute* y contiene cinco escrituras comunes con la primera base de datos, incluyendo además otras escrituras no contempladas en ella.

Cabe mencionar que algunas de las muestras de la segunda base de datos mezclaban escritura latina con escritura hindú, dado que habían sido extraídas de un diccionario. Este hecho resultó de utilidad para comprobar si el sistema podía trabajar adecuadamente a nivel de línea si la mayor parte de la misma estaba escrita en un alfabeto, y sólo una o dos palabras se encontraban redactadas en otro distinto.

Así, algunos ejemplos de esta base de datos de imágenes escaneadas se muestran en la figura posterior.

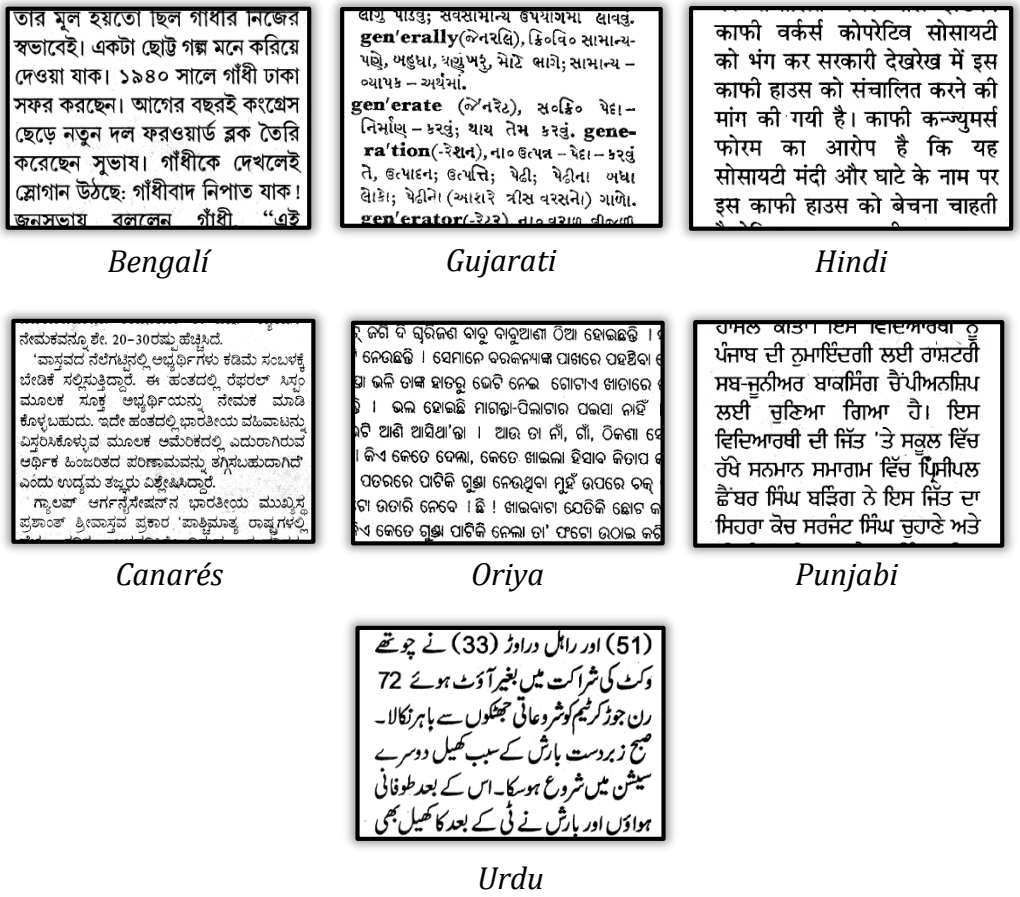


Figura 28. Muestras de los distintos tipos de escritura incluidos en la segunda base de datos.

En esta línea, se recogen en las siguientes tablas el número de imágenes que incluyen cada una de las bases de datos anteriormente mencionadas, tras finalizar los procesos de confección y organización de las mismas:

Primera Base de Datos – Traductor de Google	
Tipo de Escritura	**Número de Documentos**
Árabe, bengalí, canarés, griega, gujarati, hindi, japonesa, latina, rusa, tamil, telugu, urdu	20
TOTAL	**240**

Tabla 2. Contenido de la primera base de datos del estudio de laboratorio

Segunda Base de Datos – Documentos escaneados	
Tipo de Escritura	**Número de Documentos**
Bengalí	5
Canarés	4
Gujarati	3
Hindi	5
Oriya	20
Punjabi	20
Urdu	2
TOTAL	**59**

Tabla 3. Contenido de la segunda base de datos del estudio de laboratorio

4.4. Separación de líneas

Un aspecto importante a la hora de reconocer el tipo de escritura es establecer a qué nivel se quiere trabajar, es decir, si queremos realizar la identificación sobre documentos, párrafos, líneas o palabras. En este estudio inicial se decidió trabajar a nivel de línea, es por ello que fue necesario que parte de la programación se dedicara a dividir las imágenes de los documentos de la base de datos en líneas de texto.

De entre todos los mecanismos existentes a día de hoy para realizar esta tarea, se decide hacer uso del *Convex Hull* o la envolvente compleja. Esta técnica permite dibujar el contorno de todas las áreas de texto dentro de una imagen, siempre y cuando se realice un proceso previo de etiquetado que detecte todos los objetos presentes en la misma.

El procedimiento seguido para separar las líneas da comienzo con el cálculo de la envolvente compleja de la imagen, obteniendo los contornos de cada una de las palabras que la componen. Una vez hecho esto, se rellena el interior de los contornos resultantes y se realiza la operación morfológica de dilatación, de esta forma se consigue marcar todas y cada una de las líneas del documento de análisis.

Además, con el objeto de mejorar los resultados del paso anterior, se desarrolló un programa que permite eliminar pequeñas imperfecciones de la imagen como manchas o puntos de pequeño tamaño. Se establece para ello un umbral o área mínima, tal que si se detecta un objeto cuya superficie no supera dicho valor, queda eliminado durante este proceso.

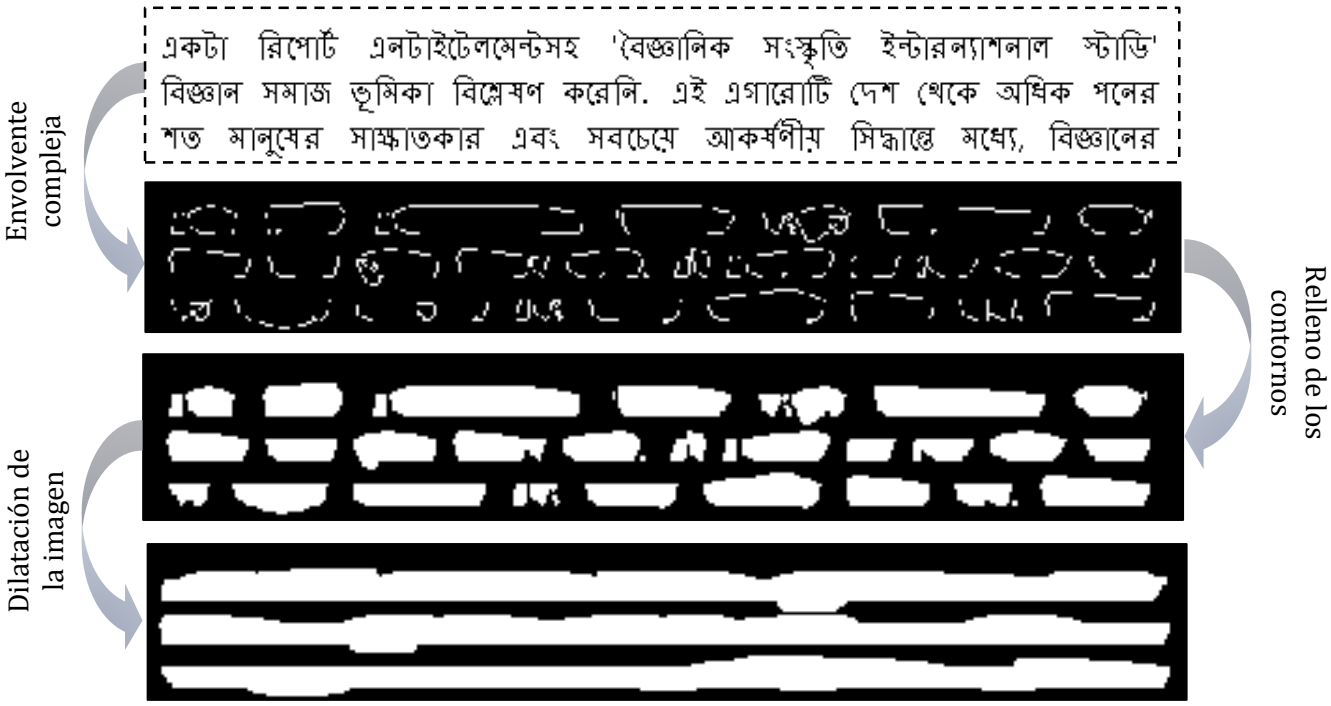


Figura 29. Procedimiento de segmentación de las líneas de un documento.

Tras tener marcadas todas las líneas del documento como se muestra en la Figura 27, se etiquetan y cuentan los objetos presentes en la imagen. A partir de este momento, se avanza objeto a objeto, guardando las líneas y sus contornos de forma individual, para así aplicar sobre ellas las técnicas propuestas en este proyecto con el fin último de identificar el tipo de escritura.

No obstante, esta sería la situación ideal donde cada objeto representa una línea y por lo tanto, las líneas no se tocan, pero también puede ocurrir que existan líneas que sí se toquen y deban ser separadas antes de guardarlas. Para determinar cuál es la situación en la que nos encontramos, tomamos cada objeto y realizamos un histograma horizontal que permita determinar si dicho elemento contiene más de una línea. Si al llevar a cabo este paso se corrobora que sólo hay una línea dentro del objeto, se procede normalmente, pasando a extraer las características de cada línea, en caso contrario, es necesario realizar nuevas comprobaciones que permitan que las líneas se separen correctamente.

En caso de detectar que un objeto contiene más de un elemento en su interior, nuestro propósito principal ha de consistir en separar esos elementos o líneas. Consecuentemente, se diseñó una rutina (ver Figura 30) que posibilitaba determinar si era posible segmentar las líneas de forma automática o si por el contrario debíamos separarlas manualmente fuera del programa.

La mencionada rutina comenzaba con la erosión del objeto conflictivo y la posterior eliminación de aquellas marcas de muy pequeño tamaño que aparezcan tras la aplicación de esta operación. Si después de llevar esto a cabo, se obtenían dos objetos, éstos se correspondían con las dos líneas separadas, no obstante, podían darse algunos casos especiales tras el paso anterior que complicaban en cierta medida la segmentación.

Por un lado, lo habitual es que no haya más de dos líneas unidas entre sí, pero antes la posibilidad de que fueran más de dos se decidió que si se detectaban más de dos elementos, nos quedaríamos con los dos superiores y avisaríamos al usuario. Cabe destacar que partimos del supuesto de que todas las líneas de un mismo tamaño son del mismo tamaño.

También puede ocurrir que al realizar la erosión una o todas las líneas desaparezcan. Esto puede indicar que la escritura debido a su propia estructura provoque que el sistema crea que se trate de dos líneas diferentes siendo una sola. Si nos encontramos en estas circunstancias, no habría ningún problema y sería posible seguir adelante, para verificar este hecho, se lleva a cabo una comprobación usando el ancho de la línea, que es un parámetro estimado a priori. En este sentido, si se observa que sí que existen dos líneas y que ambas han desaparecido, significa que debido a que las líneas están tan juntas, no pueden ser separadas automáticamente. En tal caso, se tendría que separar las líneas de forma manual aunque el programa continuaría normalmente y el objeto se consideraría una línea completa con la finalidad no interrumpir el programa.

Si resulta posible separar los dos elementos satisfactoriamente o se demuestre que se trata de una línea única, se recupera el aspecto que tenía el objeto antes de ser erosionado. Para guardar las líneas unidas por separado, se traza una línea horizontal que sirva de frontera entre ambas, con el fin de determinar el contorno de cada línea individual y guardar cada una de ellas de forma individual, permitiendo así continuar con el programa y extraer sus características adecuadamente.

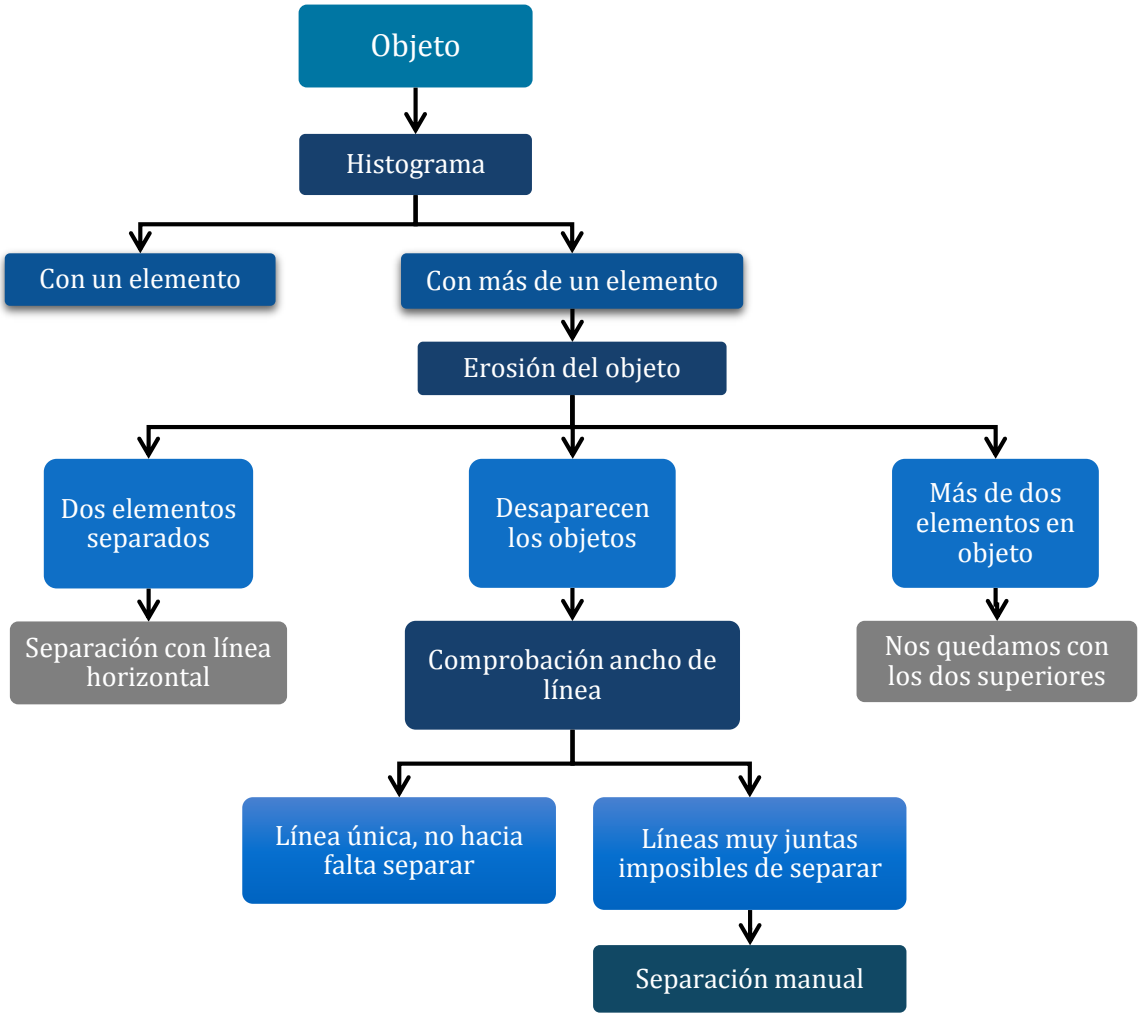


Figura 30. Proceso para separar líneas conflictivas.

4.5. Pruebas realizadas y resultados

4.5.1. Procedimiento seguido

El estudio de laboratorio que estamos tratando en este capítulo constaba de dos fases:

1. Por un lado, la primera de ellas se encargaba de la extracción de parámetros y el procesado de las imágenes, tal que se tomaban las muestras de la base de datos conformada usando el traductor de Google, se pasaba de imagen RGB a imagen lógica, se segmentaban las muestras resultantes en líneas y se extraían las características de cada una de ellas con el fin de obtener los parámetros de cada tipo de escritura.

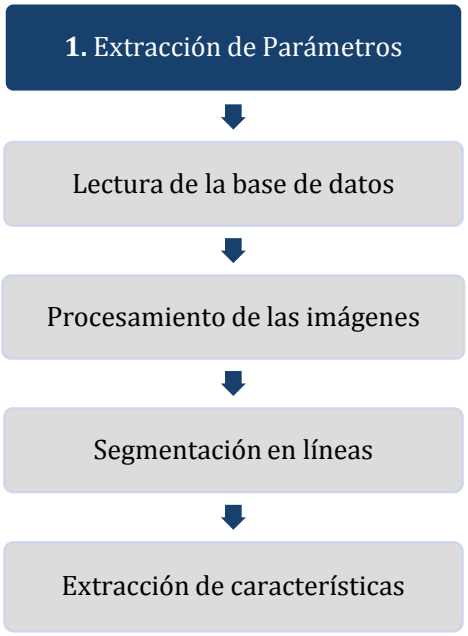


Figura 31-a. Diagramas de flujo de las fases del proceso de identificación del tipo de escritura en el estudio de laboratorio

2. Por otro lado, en la segunda fase se llevaba a cabo el proceso de clasificación, en el cual se empleaba la información obtenida del primer paso para realizar la identificación del tipo de escritura dividiendo las bases de datos en muestras de entrenamiento y de test, y usando un clasificador LS-SVM para obtener el porcentaje de acierto del sistema.

Figura 31-b. Diagramas de flujo de las fases del proceso de identificación del tipo de escritura en el estudio de laboratorio

El primer sistema que se desarrolló en este estudio inicial, por tanto, constaba de cuatro partes: la lectura de la base de datos, el procesamiento de las imágenes, la segmentación del documento y la extracción de parámetros.

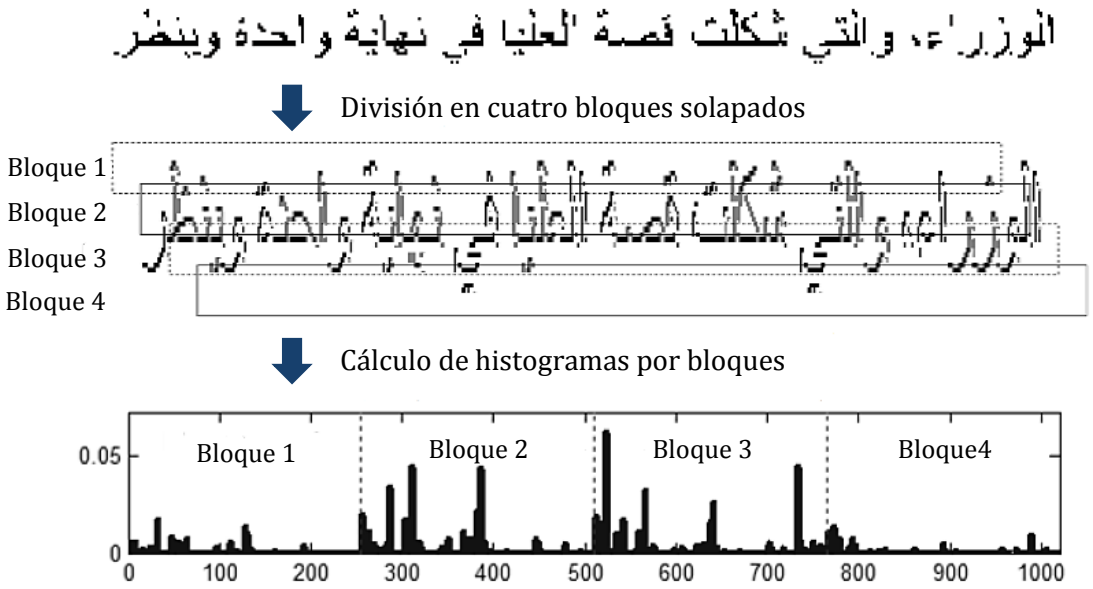


Figura 32. Ilustración de la división de una línea en cuatro bloques y el cálculo de sus respectivos histogramas LBP.

El programa que se llevó a cabo para realizar la caracterización de las imágenes de la base de datos, comienza leyendo la primera base de datos y extrayendo los documentos uno a uno para tratarlos por separado. A continuación, se pasa la imagen a color, a escala de grises y finalmente a formato lógico, de forma que la muestra final se presenta con el fondo negro y el texto blanco. Una vez hecho esto, se divide el documento de análisis en líneas, de acuerdo con la metodología descrita en el punto anterior, haciendo uso de la envolvente compleja y tomando las decisiones que se precisen en caso de que exista algún tipo de conflicto. Y finalmente, se aplican las técnicas de análisis de texturas sobre las líneas segmentadas.

Asimismo, para caracterizar cada línea, se optó por dividir el contenido de la misma en cuatro bloques horizontales con un cierto porcentaje de solapamiento y añadir un marco en lugar de caracterizar la línea completa. Posteriormente, se calculaban los patrones locales LBP, OLBP, LDP y LDerivP de cada uno de los cuatro bloques y por último, se hallaba el histograma de los resultados.

Tras realizar el cálculo de las características de todos los fragmentos de la línea, se concatenaban todos los histogramas y se normalizaba la energía del conjunto final. Podemos ver un ejemplo de estos resultados finales en la Figura 33. Cada uno de estos valores se almacenaba en una matriz como características de la línea de análisis, antes de volver a empezar con el procedimiento descrito y así analizar la siguiente línea del documento.

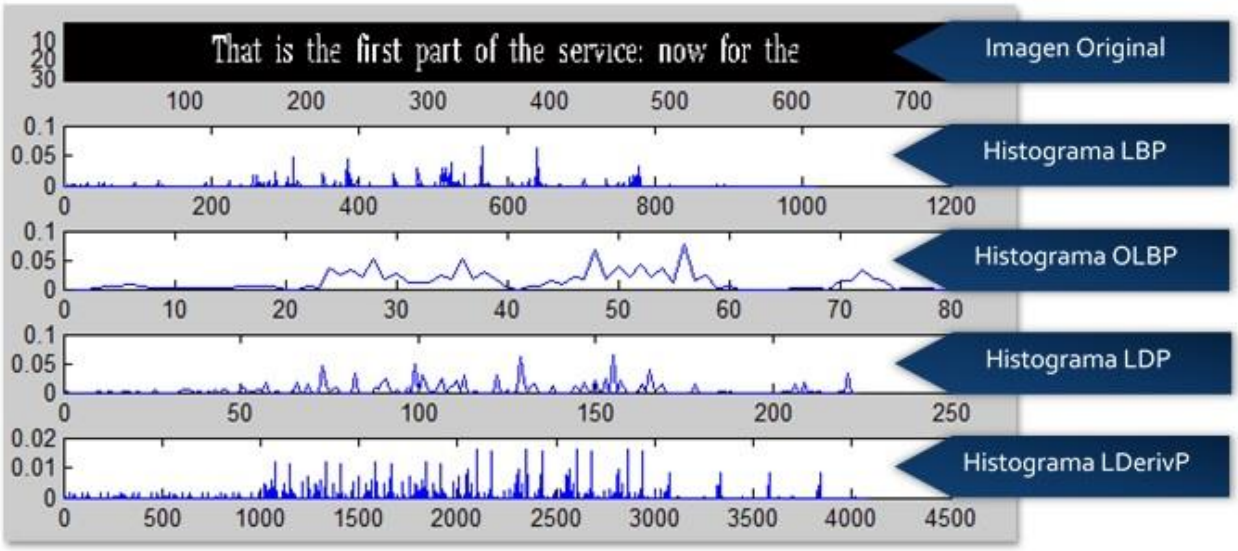


Figura 33. Ejemplo de los histogramas LBP, OLBP, LDP y LDerivP de una línea.

Tras obtener la citada matriz de características con todos los resultados de las imágenes de la base de datos, se comenzaba con la segunda fase del estudio: el proceso de clasificación e identificación del tipo de escritura.

Nuestro sistema de clasificación empezaba dividiendo la base de datos en dos grupos: muestras de entrenamiento y muestras de test; para así crear los vectores de características para entrenar y probar nuestro clasificador. Éstos eran introducidos en el clasificador LS-SVM y se enfrentaban todos los tipos de escritura para comprobar cuán bueno era el enfoque escogido de acuerdo con lo explicado en el capítulo anterior.

En cuanto a los resultados, al final de todo el proceso se obtienen tanto las tablas de confusión, como las tasas de identificación y las curvas de coincidencia acumulativas (CMC) que ya hemos explicado en el Capítulo 3. Estas últimas son las que se han escogido como una forma adecuada de representar cuán buena es la identificación dependiendo del patrón local que se emplee.

Por consiguiente y para obtener las curvas CMC, la primera base de datos se divide al azar en dos partes no solapadas. La primera parte se utiliza para entrenamiento y la segunda parte para test. La segunda base de datos, como ya se comentó al principio del capítulo, se destina únicamente a test. La división aleatoria de la base de datos y el cálculo de las curvas CMC se hacen diez veces para así evitar un resultado expresado en mínimos locales. En consecuencia, los resultados finales que se presentan más adelante son la media de estos diez experimentos.

4.5.2. Resultados de la experimentación

Se llevaron a cabo dos pruebas para comprobar la idoneidad de nuestro planteamiento, una usando solamente la primera base de datos y otra usando las dos bases de datos, las de imágenes de texto creadas en este estudio y la de imágenes escaneadas que teníamos a nuestra disposición.

A. *Evaluación de la primera base de datos*:

En este caso, el esquema se evalúa entrenando y realizando pruebas con la primera base de datos. Las muestras de entrenamiento se obtienen mediante la selección de un porcentaje aleatorio de la base de datos y las muestras restantes se usan para el proceso de test.

Decidimos hacer varias pruebas con objeto de determinar qué porcentaje de datos de entrenamiento obtenía mejores resultado. La tasa de identificación obtenida variando ese porcentaje aplicando cada uno de los parámetros de análisis se ilustra en la siguiente tabla:

Porcentaje de entrenamiento	Número de líneas para entrenamiento	LBP	OLBP	LDP	LDerivP
1%	41	84,53%	85,35%	85,87%	84,96%
5%	179	96,39%	96,68%	96,15%	95,91%
10%	353	98,89%	98,55%	98,75%	98,84%
20%	699	99,37%	99,18%	99,61%	99,47%
30%	1046	99,66%	99,51%	99,61%	99,56%
40%	1393	99,61%	99,47%	99,71%	99,56%
50%	1745	99,61%	99,61%	99,80%	99,61%
60%	2094	99,51%	99,61%	99,71%	99,42%
70%	2438	99,51%	99,42%	99,80%	99,71%

Tabla 4. Tasa de identificación de escritura variando el porcentaje de la base de datos destinado a entrenamiento.

Se puede observar en la Tabla 4 que la tasa de precisión en la identificación del tipo de escritura no mejora si se supera un porcentaje de entrenamiento del 30%.

Además, si comparamos los resultados de los diferentes patrones locales, podemos ver que los resultados obtenidos mediante la aplicación de los OLBP y LDP superan a los valores asociados a los parámetros LDerivP y LBP, lo cual acentúa la importancia de contar con una buena densidad de dirección de los trazos para la adecuada identificación de la escritura.

Por otro lado, en la siguiente tabla se incluyen los distintos comportamientos del sistema, mostrando para ello diferentes resultados en función del rango, entrenando con un 30% de las líneas de texto de la primera base de datos y realizando el proceso de test con las imágenes restantes.

Rango / Mejor Elección	LBP	OLBP	LDP	LDerivP
1	99,34%	99,38%	99,21%	99,34%
2	99,75%	99,79%	99,83%	99,75%
3	99,91%	99,83%	99,95%	99,85%
4	100%	100%	99,95%	100%
5	100%	100%	100%	100%

Tabla 5. Resultados obtenidos entrenando con el 30% de la primera base de datos y realizando el test con las imágenes restantes.

Se puede observar que el rango 1 o la tasa de identificación teniendo en cuenta solamente la mejor opción no coincide con la que aparece en la Tabla 4, esto ocurre porque esta última presenta los resultados promediados, mientras que en la Tabla 5 se presenta el resultado exacto del experimento. También se comprueba que al calcular el rango 4 el rendimiento del sistema pasa a ser prácticamente del 100%, y ello demuestra que el uso de técnicas de análisis de texturas es válido también para la identificación de escritura. Para representar mejor los resultados obtenidos, se incluye la curva CMC en la Figura 34.

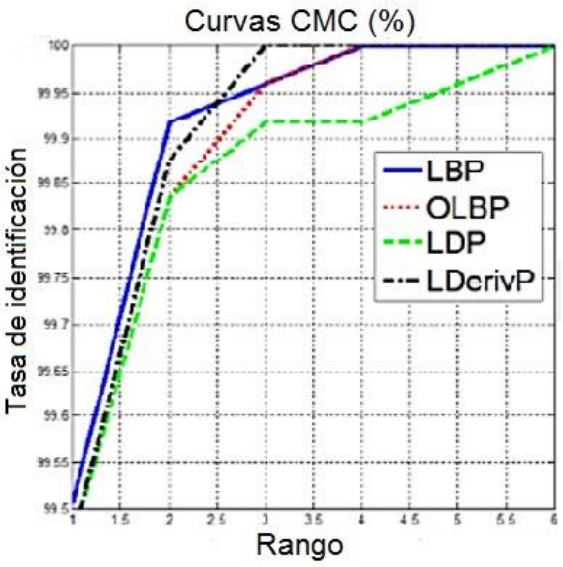


Figura 34. Curvas CMC obtenidas entrenando con el 30% de la primera base de datos y realizando el test con las imágenes restantes.

Cabe destacar que los casos donde se detectó una mayor confusión se dieron lugar al enfrentar la escritura árabe con la urdu y la griega con la rusa. Tras investigar acerca del primer par de escrituras descubrimos que comparten la mayor parte de los caracteres, tal que el alfabeto urdu contiene pocos elementos que lo distingan del árabe; consecuentemente, la similitud estructural es muy alta y al sistema le resulta complicado reconocer a qué escritura pertenecen las líneas de análisis. En el caso de las escrituras rusa y griega ocurre algo similar, las estructuras de las palabras guardan un cierto grado de parentesco que hace que en ocasiones nuestro sistema se confunda en la identificación.

B. Evaluación de la segunda base de datos

Tratando de evaluar la capacidad de generalización del sistema de detección del tipo de escritura propuesto, conservamos los modelos de escritura obtenidos entrenando con el 30% de las líneas de texto de la primera base de datos pero realizamos el test con las líneas de la segunda base de datos.

Asimismo, los resultados de este experimento se muestran en la Tabla 6 mientras que la Figura 33 muestra las curvas CMC. Hemos de destacar que hay sólo 5 idiomas comunes entre la primera y la segunda base de datos.

Rango	LBP	OLBP	LDP	LDerivP
1	84,57%	90,05%	86,81%	84,07%
2	92,53%	95,02%	91,29%	92,03%
3	97,26%	97,01%	94,27%	95,77%
4	98,75%	99,00%	99,00%	97,01%
5	99,25%	99,50%	99,50%	98,01%
6	99,50%	99,75%	99,75%	98,75%
7	99,50%	99,75%	99,75%	99,25%
8	99,75%	99,75%	99,75%	99,25%
9	99,75%	100%	99,75%	99,75%
10	100%	100%	100%	100%

Tabla 6. Resultados obtenidos entrenando con el 30% de la primera base de datos y realizando el test con la segunda base de datos.

En la presente tabla se puede observar que en este experimento hasta llegar al rango 9 no tenemos un 100% de rendimiento. También, se comprueba que el OLBP supera claramente al resto de los patrones locales, debido a la manera en que este parámetro resalta las direcciones de los trazos de las distintas escrituras.

Además, en este caso, la mayor parte de las confusiones se producen en la escritura urdu por las razones que ya comentamos en el punto anterior.

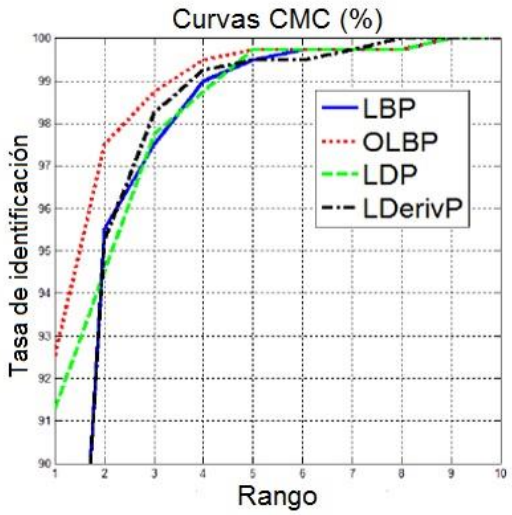


Figura 35. Curvas CMC obtenidas entrenando con el 30% de la primera base de datos y realizando el test con la segunda base de datos.

Capítulo 5. Estudio realista

5.1. Introducción

Tras comprobar la validez y eficacia de los patrones locales en el reconocimiento del tipo de escritura mediante el trabajo llevado a cabo en el estudio inicial, nos propusimos la creación de un sistema más complejo donde se analizara el efecto de los parámetros de textura en el reconocimiento del tipo de escritura a nivel de documentos, líneas y palabras. Este nuevo paso implicaba la confección una base de datos mucho mayor con documentos reales escaneados, que contaran con imperfecciones del papel y distintos tipos de estilos y tamaños de fuentes.

En este quinto capítulo de la memoria, se explica cómo se ha llevado a cabo la conformación de la nueva base de datos y los procedimientos seguidos para, partiendo de imágenes de documentos, obtener las de las líneas y posteriormente, las de las palabras. Además, se describe el procedimiento seguido en la extracción de características en cada nivel de muestras, en la elección de las muestras de entrenamiento y de test y en la clasificación. Añadiendo al final de este capítulo los resultados obtenidos una vez completados una serie de experimentos.

5.2. Diseño de la base de datos

El objetivo principal de este Proyecto de Final de Carrera es comprobar si el análisis de los patrones locales es una buena opción para realizar la identificación del tipo de escritura de un documento. Sin embargo, la primera base de datos llevada a cabo durante el estudio inicial no estaba compuesta de documentos reales, sino que se habían confeccionado diversas muestras en el propio laboratorio con tamaños, estilos y fuentes de texto muy similares, sobre fondos muy limpios y sin imperfecciones. Estos documentos se alejan de la realidad sobre la que se pretende aplicar nuestro sistema, por ejemplo, en el caso de documentos históricos, los textos suelen estar degradados, el papel desgastado o manchado y las fuentes suelen ser distintas de un documento a otro.

Consecuentemente, vimos necesaria la creación de una nueva base de datos con más tipos de escritura, más documentos y más variedad en lo que a fuentes, características y entorno del texto se refiere.

Además, queríamos trabajar a todos los niveles posibles, esto quiere decir que sabiendo que nuestra propuesta tenía buenos resultados si se aplicaba a líneas, pensamos

que también sería útil ver el efecto que éste tenía también sobre documentos y palabras. Por este motivo, se estableció que la base de datos debía contener muestras de todos los niveles. Así, a partir de los documentos se extraían las líneas y con éstas, las palabras. Por consiguiente, nuestro primer propósito en lo referido al diseño de la base de datos, era que ésta contara con al menos diez documentos, cincuenta líneas y quinientas palabras por cada tipo de escritura que se incluyera.

Para crear nuestra base de datos, desde el *Indian Statistical Institute* y el *Grupo de Procesamiento Digital de la Señal* se nos facilitó un gran número de ejemplares físicos, concretamente, periódicos, libros y revistas, con muchas escrituras distintas de la India y de Asia en general.

Figura 36. Documentos físicos digitalizados para formar parte de la base de datos

Cabe mencionar también que a medida que avanzaba el proyecto, el equipo colaborador del *Indian Statistical Institute,* amplió nuestra base de datos con palabras de algunas escrituras hindúes como el bengalí o el malayalam. Decidimos incluir estas muestras en nuestra base de datos en un directorio a parte y se estableció que se emplearían únicamente durante proceso de test, pues no disponíamos de las líneas y los documentos de los que habían sido extraídas como ocurría con la totalidad de los textos que se incluían en la base de datos que estábamos confeccionando.

Una vez organizados todos los documentos y elaborada la base de datos como describiremos en el punto siguiente, contábamos con doce tipos de escritura distintos en total:

- Árabe
- Bengalí
- Canarés
- Gujarati
- Gurmukhi
- Hindi
- Japonesa
- Malayalam
- Oriya
- Latina
- Tailandesa
- Telugu

Un hecho destacable es que nuestra base de datos cuenta con una peculiaridad, y es que las imágenes de las palabras, las líneas y los documentos no tienen el mismo aspecto. Los documentos son imágenes en color, las líneas se encuentran en escala de grises y las palabras son imágenes en blanco y negro, donde el texto se expresa en color blanco y el fondo en negro (ver Figura 37). Esto es debido a que quisimos que nuestros formatos coincidieran con los de las imágenes que íbamos recibiendo desde la India, con la finalidad de que el procesado de los documentos, las líneas y las palabras fueran similares con alguna pequeña diferencia.

No obstante, a la hora de extraer las características de las muestras establecimos en el primer estudio que las imágenes debían pasadas a formato lógico y posteriormente, se requería que fuera invertidas, quedando el texto en blanco y el fondo en negro. De forma que realmente el hecho de que tengamos imágenes en color, en escala de grises o en blanco y negro no afecta demasiado a la parte fundamental del sistema, que es la aplicación de los distintos patrones locales.

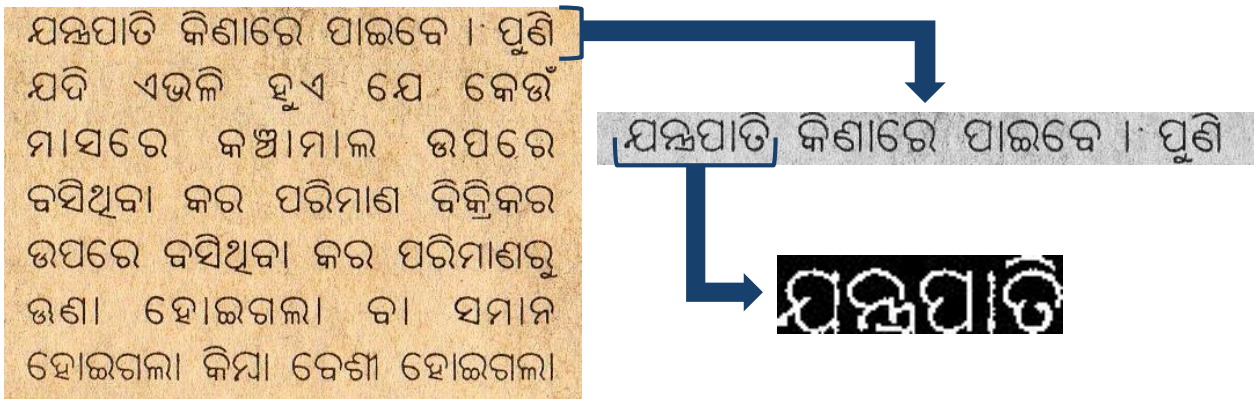


Figura 37. Ejemplos de muestras de la base de datos a tres niveles: documento, línea y palabra escritos en Oriya.

5.3. Elaboración de la base de datos

La base de datos que creamos en este estudio nos llevó mucho más tiempo que la del primero y guardaba una mayor complejidad, tanto en el proceso de captura de las imágenes como en la organización de los directorios donde se encontraban contenidas las imágenes. El procedimiento seguido para obtener las imágenes de los documentos se presenta a continuación:

1. Se seleccionaban los bloques de texto o artículos que iban a servir como muestras dentro de los documentos físicos correspondientes a cada tipo de escritura, evitando aquellos con números y con palabras redactadas en otra clase de escritura distinta.
2. Se escaneaba la página donde se encontraba el texto seleccionado a 300 dpi. Mediante al software asociado a la imagen, hacíamos además que la imagen obtenida estuviera en formato TIFF.
3. Se comprobaba la inclinación y el aspecto de la imagen obtenida del paso anterior, dado que durante el proceso de escaneado, especialmente en el caso de periódicos y documentos grandes era complicado que los documentos no se arrugasen, doblasen. Si la imagen no estaba recta y se podía corregir, se usaba un programa de edición fotográfica para hacerlo.
4. Se recortaba el texto de interés de la página completa, mediante un programa de edición fotográfica.
5. Se nombraba la imagen usando cuatro letras para el nombre de la escritura a la que pertenecía y tres dígitos que se correspondían con su índice, que dependía del orden en que hubiese sido capturada.
6. Finalmente, se almacenaba la imagen en el directorio que contenía los documentos impresos, en la carpeta del tipo de escritura en que estaba escrito.

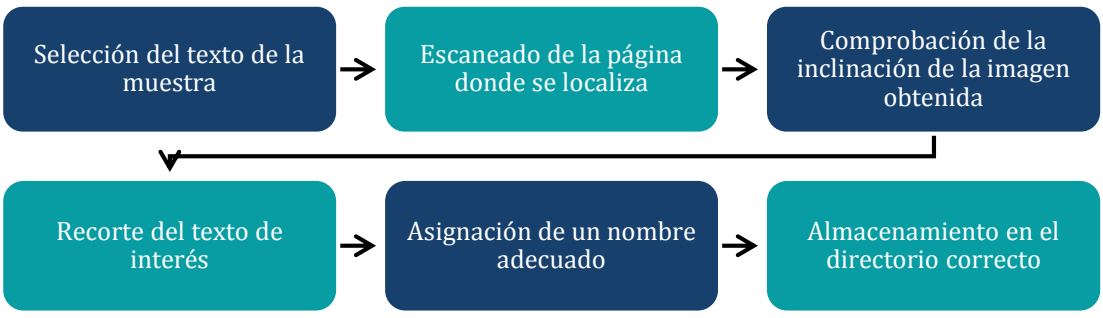


Figura 38. Diagrama de bloques del proceso de elaboración de la base de datos de documentos.

La siguiente fase en la elaboración de la base de datos, consistió en completar el apartado que contenía las líneas. Para ello se transformaban las imágenes de los documentos a escala de grises, se segmentaban en líneas mediante un proceso semi-automático que explicaremos con posterioridad, y se almacenaban las líneas recortadas en formato TIFF. Además, se les otorgaba el mismo nombre que el documento de origen, con la diferencia de que se les asignaba un segundo índice de tres dígitos que se correspondía con el número de la línea en el documento.

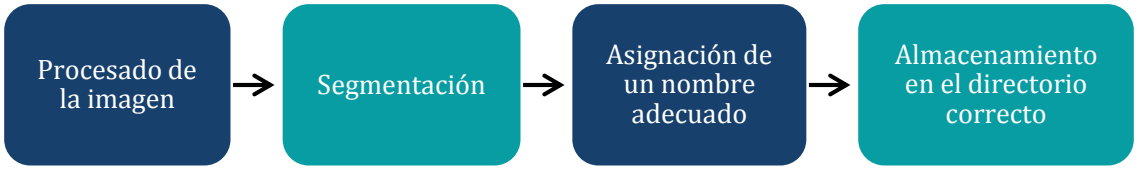


Figura 39. Diagrama de bloques del proceso de elaboración de la base de datos de líneas y palabras.

Análogamente y para poner fin a la constitución de la base de datos, se procedía a convertir las líneas en palabras. En primer lugar, se pasaba de imágenes en escala de grises a imágenes lógicas, luego se invertía el resultado y finalmente, se dividía en palabras. De este modo, al final del proceso descrito se obtenían las palabras separadas, todas ellas con el texto en blanco y el fondo en negro. En lo relativo al nombre del archivo, se procedía de forma similar al caso de las líneas, tal que se usaba el mismo nombre de la línea de origen pero se les añadía un nuevo índice, de tal forma que en el nombre de la imagen aparecía el número del documento de origen, la línea y la posición concreta de la palabra dentro de ella, todos estos valores contaban con tres dígitos cada uno.

Por otro lado, la base de datos de palabras para test cuyas muestras se enviaron desde la India, también tuvieron que ser modificadas y nombradas. Nos percatamos de que era necesario implementar un sistema que impusiera un formato de imagen determinado, debido a que se nos enviaron imágenes en TIF, JPEG y PNG, cuando la base de datos estaba formada por imágenes de tipo TIF únicamente.

También tuvimos que emplear un software llamado *Lupas Rename* que permitía modificar los nombres de una gran cantidad de ficheros y automatizar la asignación de índices, ya que las imágenes se nombraban en la India de acuerdo con sus coordenadas en la línea de donde se extraían. Como pretendíamos que todas las imágenes tuvieran asignado un nombre similar, seleccionábamos un grupo de palabras y les asignábamos tres dígitos para el tipo de escritura seguidos por un “2”, que servía de indicador de que la imagen

pertenecía a la base de datos de test, además de tres dígitos que representaban el índice de un documento, otros tres para indicar la línea de procedencia y tres más para el orden de la palabra. Estos últimos valores se asignaban de forma aleatoria, procurando que el número de palabras por línea fuera similar a la media del tipo de escritura en cuestión.

5.4. Segmentación

Una de las etapas más laboriosas en la creación de la base de datos es la segmentación. Debido a la naturaleza de las imágenes, fue necesario desarrollar una serie de procedimientos semiautomáticos tanto para dividir los documentos en líneas como para obtener las palabras de las líneas.

Además, el hecho de no conocer las escrituras con las que trabajábamos añadió una mayor complejidad a la importante tarea de segmentar las imágenes. En consecuencia, como veremos en las líneas siguientes, se tomaron una serie de decisiones a la hora de dividir las palabras contenidas en una línea concreta. De manera que se obtenían y almacenaban extractos de texto de un cierto número de caracteres como palabras aunque no lo fueran realmente. Así, conseguíamos que en casos como el del tailandés, donde puede haber palabras de más de quince caracteres, se formara una base de datos de esta escritura con imágenes de palabras de distintos tamaños, al dividir esas grandes palabras en fragmentos de texto más pequeños.

5.4.1. Segmentación de líneas

El planteamiento inicial llevado a cabo para dividir los documentos en líneas fue aplicar los mismos conceptos que en el caso del primer estudio. Recordemos que este software estaba basado en el uso de la envolvente compleja, y para hacer pruebas con él tuvo que ser editado para que una vez obtenidas las líneas se guardaran en el directorio correcto con el nombre apropiado sin aplicar sobre ellas los parámetros de textura como ocurría en el software original.

No obstante, después de hacer varias pruebas comprobamos que en algunos documentos cuyo papel era antiguo o se encontraba algo desgastado, la separación de líneas no se hacía correctamente. Esto se debía a que esas imperfecciones hacían que los contornos de las líneas no se establecieran de una manera adecuada o se entrara en algunos de los casos de error que se evaluaban en la descripción del capítulo anterior. Por todo esto, el

programa consideraba como líneas únicas algunos bloques que contenían en su interior dos o más líneas y la nombraba como si así lo fuera, lo cual resultaba complicado de arreglar sobre todo si el error se producía en las primeras líneas de un documento con decenas de ellas.

Por este motivo se desarrolló un nuevo programa que trataba de otra forma las escrituras que entraban en conflicto.

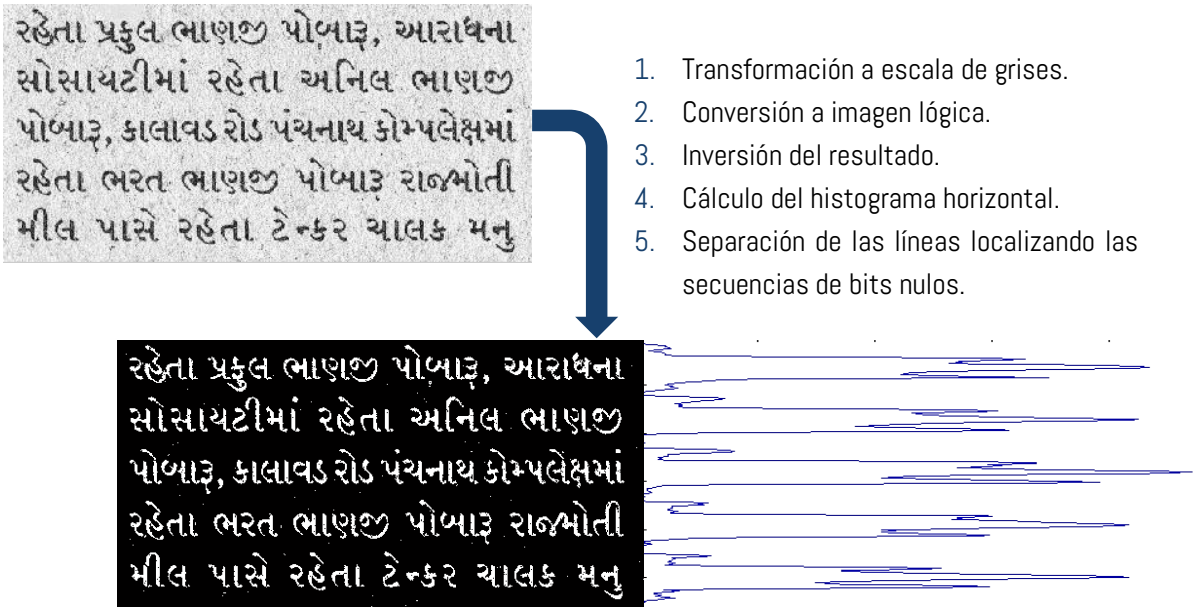


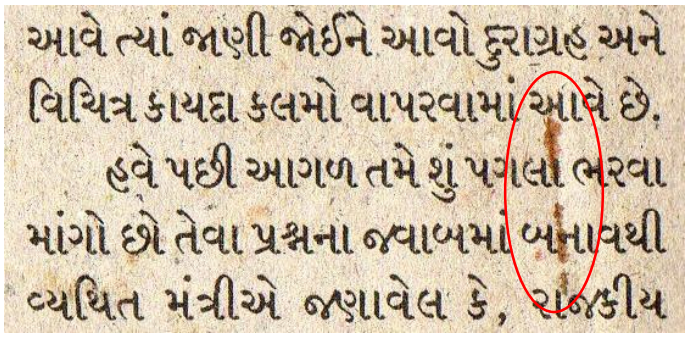
Figura 40. Proceso seguido para la segmentación de líneas.

Como podemos ver en la Figura 40, éste realizaba el histograma horizontal de todo el documento y separaba las líneas de acuerdo con las secuencias de valores nulos que encontraba en el vector obtenido como resultado de la aplicación del histograma. Este programa además permitía recortar las líneas añadiendo un marco de unos pocos píxeles alrededor de las mismas, y también incluir en ellas las marcas, puntos, líneas y otras características de escrituras como la latina o la árabe mediante la adición de una condición asociada al tamaño de estas secuencias nulas.

Aplicando todo lo anterior conseguimos solventar muchos casos que entraban en conflicto pero no todos, y es que algunas líneas no podían ser separadas siquiera a simple vista. Por ejemplo, en ocasiones ocurría que algunas imperfecciones presentes en el papel o en la tinta del documento físico hacían que las líneas estuvieran unidas una vez procesada la imagen del documento, como ocurre en la Figura 41-a, y este hecho hacía imposible la segmentación vía software de estas líneas problemáticas.

También existían casos en varias escrituras donde las líneas estaban muy próximas de forma nativa en el documento físico, y en consecuencia no era posible separarlas trazando una línea recta horizontal que era el método aplicado por el programa para delimitar cada una de las líneas de texto. Un ejemplo de ello se ilustra en la muestra representada en la Figura 41-b.

(a)



(b)

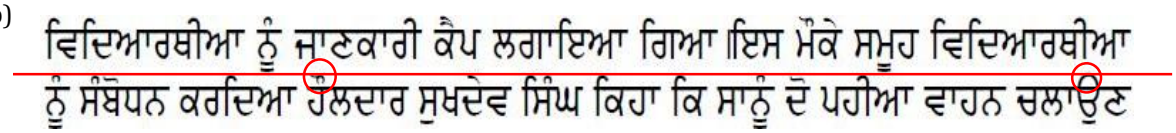


Figura 41. Ejemplos de documentos que necesitan ser segmentados manualmente.

Consecuentemente, en todas las imágenes de este tipo fue necesario segmentar las líneas manualmente usando un software de edición de imágenes. Cabe mencionar que al hacer esto resultaba imperativo el ajuste de los índices de las líneas una vez separadas. Para hacer esto último también se usó el software *Lupas Rename,* cuyo funcionamiento ya hemos comentado durante la explicación relativa a la elaboración de la base de datos.

Así, la metodología seguida finalmente para la segmentación de líneas consistía en separar las líneas del documento usando los programas ya mencionados, de tal forma que si al final se obtenían varias líneas unidas en una única imagen, éstas se separaban a mano, se editaban para eliminar los elementos o marcas sobrantes y se renombraban todas las muestras de las líneas conflictivas y las posteriores a las mismas, con el objeto de adaptar y ordenar los índices de todas y cada una de las líneas de texto de un documento.

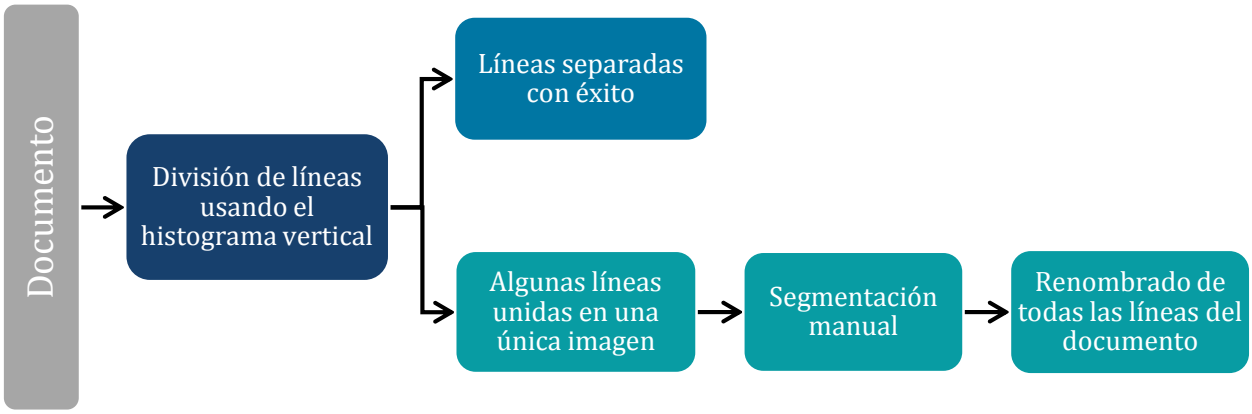


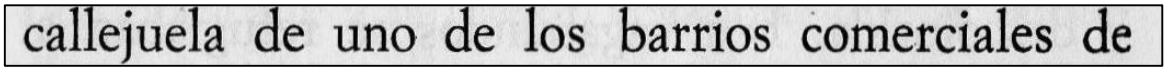
Figura 42. Diagrama de flujo del proceso de segmentación de líneas.

5.4.2. Segmentación de palabras

Una vez separadas y organizadas todas las líneas pertenecientes a cada clase de escritura en sus respectivos directorios, el siguiente paso consistió en separar las palabras de dichas líneas de texto. Para esta tarea también se siguió un método semiautomático.

En este sentido, se plantearon varias soluciones al problema de la segmentación a este nivel, una de ellas pasaba por llevar a cabo un histograma vertical y determinar dónde se encontraban los espacios entre palabra y palabra analizando los nulos obtenidos. Esta propuesta resultaba válida y eficaz en aquellas escrituras donde las palabras estaban claramente delimitadas como es el caso de la latina, en ella los amplios espacios entre palabra y palabra hacen que no resulte complejo segmentar la línea, tal y como puede observarse en la primera imagen de la Figura 43.

Sin embargo, nuestro desconocimiento acerca de la mayor parte de las escrituras incluidas en la base de datos resultaba especialmente problemático a la hora de segmentar líneas compuestas de palabras cuyo comienzo y fin no eran fáciles de discernir a simple vista. En concreto, las escrituras árabe, tailandesa, japonesa y malayalam fueron las más conflictivas, hecho que puede comprobarse en las imágenes siguientes.



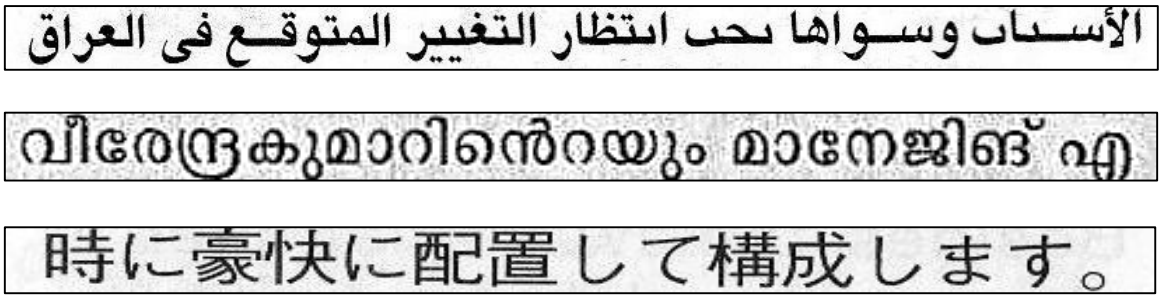


Figura 43. Ejemplos de líneas en alfabeto latino, tailandés, árabe, malayalam y japonés respectivamente.

Un hecho a destacar es que, en el caso concreto de la escritura árabe, se decidió que lo mejor era que la obtención de palabras fuera llevada a cabo por alguien que conociera el alfabeto en cuestión y en consecuencia, el equipo del *Indian Statistical Institute* realizó esta tarea por nosotros.

Para tratar con todos los tipos de escritura a excepción del árabe, incluyendo aquellas con una mayor complejidad, se desarrolló un procedimiento semiautomático que permitía dividir las líneas en palabras mediante la interacción de un usuario externo que tomara las decisiones apropiadas para realizar la segmentación con éxito. De esta forma el usuario tomaba partido en cierta medida en la tarea de mejorar los resultados obtenidos en el caso de la aplicación de un sistema automático basado en histogramas.

La primera versión del sistema propuesto analizaba las líneas y colocaba rectas verticales indicando el lugar donde éste intuía que se separaban unas palabras de otras, analizando los huecos más amplios de la línea. El usuario podía entonces añadir nuevas rectas o eliminar las existentes hasta considerar que la segmentación se realizaba de forma adecuada. Una vez éste confirmaba que la línea estaba dividida en palabras correctamente, el programa almacenaba los resultados con el nombre y los índices correctos.

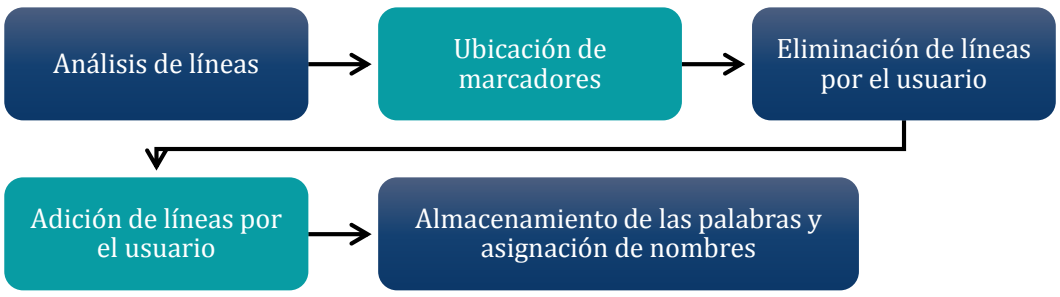


Figura 44. Diagrama de bloques que resume el proceso de segmentación de palabras del primer sistema desarrollado.

No obstante, esta propuesta no solucionaba el mayor conflicto al que nos enfrentábamos en el proceso de segmentación, que era provocado por las líneas escritas en japonés, malayalam y tailandés. En los textos escritos en los mencionados alfabetos podían formarse líneas compuestas de palabras de una gran longitud, llegando incluso a ocupar con una única palabra una línea de texto completa. En consecuencia, se realizó una modificación del programa para añadir una etapa adicional que se ejecutara tras la anterior y permitieran que estas grandes palabras se dividieran en otras más pequeñas.

Para la programación del sistema final se partió por tanto del primer desarrollo para la segmentación de líneas, se añadieron nuevas opciones y se aumentaron los pasos donde el usuario podía interactuar con el sistema y contribuir a la toma de decisiones. En consecuencia, el nuevo método realizaba cuatro operaciones distintas antes de almacenar las palabras separadas. Su funcionamiento se detalla a continuación:

1. En primer lugar, se añadió un apartado que posibilitaba eliminar áreas completas de la imagen, como por ejemplo signos de puntuación, los cuales dificultan el proceso de segmentación o incluso imperfecciones de la imagen que habían sido pasadas por alto al segmentar el documento en líneas. El usuario debía simplemente trazar una figura haciendo cuatro clics con el ratón, y el interior del cuadrilátero descrito al unir con rectas dichos puntos desaparecía tras el cuarto clic. Esta etapa podía repetirse tantas veces se requiriera, ya que se preguntaba al usuario si deseaba eliminar nuevas superficies o si por el contrario, prefería seguir adelante con el programa.

2. Una vez superado lo anterior, el programa analizaba la imagen en busca de los huecos entre palabras más notables como ocurría en la primera versión y establecía aquellas líneas separadoras que creía conveniente. Entonces, el usuario debía decidir si los marcadores establecidos eran correctos o si no.

Figura 45-a. Línea escrita en tailandés con separadores entre palabra y palabra.

Si se informaba al programa de que la segmentación no era correcta, se pasaba a añadir y eliminar separadores. De esta forma, en primer lugar se le preguntaba al usuario si creía necesario eliminar alguna de las marcas establecidas, en caso de ser así,

éste debía hacer clic sobre la línea a suprimir. El programa eliminaba entonces la línea más cercana a la posición señalada por el usuario.

Si no era pertinente eliminar alguna de las líneas trazadas o si se llevaba a cabo la supresión de marcas satisfactoriamente, confirmándose que se habían borrado todas las líneas que se creía conveniente, el programa continuaba consultando al usuario para saber si quiere añadir nuevos separadores además de los ya marcados o no.

En caso afirmativo, se debía hacer clic sobre dos puntos de la ventana que ilustraba la imagen. Después del segundo clic, el sistema trazaba una recta que partía del margen superior, pasando por ambos puntos hasta llegar al inferior. El hecho de permitir la elección de dos puntos, hacía posible que el separador pudiera ser una línea diagonal en lugar de vertical.

3. Seguidamente y una vez establecidos los separadores más claros entre palabras, se dividían las mismas en sucesiones de dos caracteres, tres caracteres, etc. y nuevamente, se preguntaba al usuario si se encontraba conforme con la segmentación o deseaba añadir o eliminar marcadores.

   Este paso se repetía con todas las palabras contenidas entre los separadores dibujados en el punto anterior.

Figura 45-b. Línea escrita en tailandés con separadores dividiendo la primera palabra de la línea.

Figura 45-c. Línea escrita en tailandés con separadores dividiendo la segunda palabra de la línea.

Figura 45-d. Línea escrita en tailandés con separadores dividiendo la tercera palabra de la línea.

4. Una vez el usuario verificara que cada palabra se dividía en los fragmentos apropiados, el sistema almacenaba cada uno de los resultados finales. Para completar esta tarea, se vaciaba o rellenaba con píxeles de valor “0” aquellas partes de la imagen que rodeaban

el contenido de la palabra de análisis, el cual se hallaba encerrado por dos separadores o un separador y un margen. Y antes de guardar, se realizaba un histograma vertical y otro horizontal para quedarnos solamente con el texto y eliminar las partes sobrantes de la imagen de la línea completa.

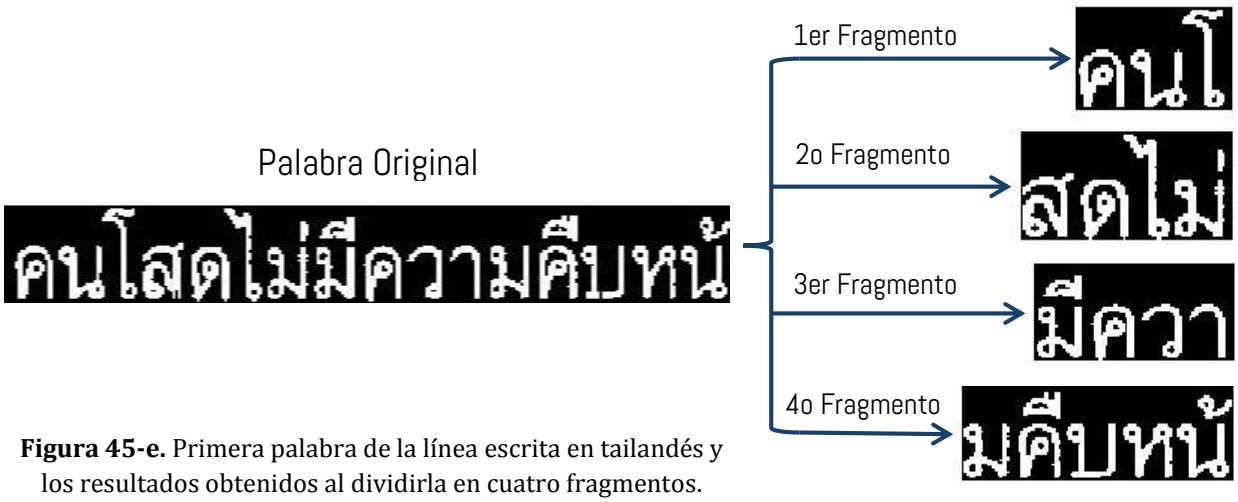


Figura 45-e. Primera palabra de la línea escrita en tailandés y los resultados obtenidos al dividirla en cuatro fragmentos.

En ocasiones era necesario llevar a cabo un paso más debido a que el software que diseñamos para separar las líneas rellenaba todas las partes de la imagen que sobraban pero no editaba los bordes, por lo que digamos que al guardar las palabras finales si había alguna línea con caracteres que llegaran hasta el último píxel de los márgenes superior e inferior esos puntos quedaban estáticos y por lo tanto no se efectuaba correctamente el descarte de los fragmentos de la imagen no pertenecientes a la palabra segmentada. Por consiguiente se creó un programa que buscaba estas secuencias de píxeles presentes en los márgenes, cuyo funcionamiento era similar al de la función que borraba pequeñas marcas. Éste localizaba y etiquetaba aquellos puntos que no superaban un área de 30 píxeles, los eliminaba y recortaba la palabra correctamente, guardando el resultado bajo el mismo nombre que tenía la imagen antes de ser modificada.

5.5. Estadísticas y verificación

Tras el fin de todos los procedimientos para elaborar la base de datos, que consistían en escanear los documentos físicos, seleccionar y almacenar las muestras consideradas documentos, segmentar dichas imágenes en líneas y luego en palabras, y finalmente, organizar todas las carpetas de forma apropiada, se obtuvo una base de datos más realista y completa.

Un aspecto a tener en cuenta es que uno de nuestros objetivos iniciales consistía en escanear prácticamente el mismo número de documentos para cada clase de alfabeto, de tal forma que la base de datos, al menos a este nivel, quedara equilibrada. Sin embargo, cuando terminamos de elaborar toda la base de datos y pedimos al equipo del *Indian Statistical Institute* que revisara nuestro trabajo, nos comentaron que teníamos dos carpetas separadas para el mismo tipo de escritura, una de ellas se correspondía con el punjabi y otra con el gurmukhi. Nuestro error radicó en que el nombre del alfabeto en cuestión es gurmukhi, pero el idioma que se redacta haciendo uso de éste se denomina punjabi. Es por ello que decidimos combinar ambas carpetas y renombrar los documentos, líneas y palabras apropiadamente, y también es por este motivo por el que se trata del directorio que incluye un mayor número de documentos.

Por otro lado, contábamos también con algunas clases de escrituras donde los documentos físicos eran muy pequeños o los textos seleccionados tenían pocas líneas. Esto ocurría por ejemplo en las muestras redactadas en japonés, las cuales se encontraban en una guía turística de bolsillo. Así, para poder trabajar con una cantidad de texto similar en todos los casos, decidimos aumentar el número de documentos de estos casos concretos.

Dicho esto, los datos relativos al contenido de la base de datos final de nuestro estudio realista se exponen en la Tabla 7. Haciendo uso de la misma, podemos comprobar que se cumplen todos los propósitos que marcamos al inicio del diseño de la base de datos, ya que tenemos más de diez tipos de escritura distintos, con más de diez documentos, cincuenta líneas y cien palabras cada uno.

Base de datos realista			
Tipo de Alfabeto	*Nivel*		
	Documentos	*Líneas*	*Palabras*
Árabe	51	1105	8141
Bengalí	54	472	2674
Gujarati	56	398	2365
Gurmukhi	120	1075	11021
Hindi	67	401	3233
Japonés	80	565	2619
Canarés	67	597	2357

Malayalam	70	730	5502
Oriya	44	560	2401
Latino	56	972	8143
Telugu	67	494	2209
Tailandés	64	467	4034
TOTAL	796	7836	54704

Tabla 7. Contenido de la base de datos realizada durante el estudio realista a partir de documentos físicos.

Análogamente, no debemos olvidar que teníamos un conjunto de imágenes de palabras redactadas en diversas escrituras hindúes que se empleaban solamente en el proceso de test. El contenido de la misma se resume a continuación:

Base de datos de test	
Tipo de Alfabeto	*Número de Palabras*
Bengalí	2674
Gujarati	2365
Gurmukhi	11021
Hindi	3233
Canarés	2357
Malayalam	5502
Oriya	2401
TOTAL	796

Tabla 8. Contenido de la base de datos de palabras destinadas al proceso de test.

5.6. Pruebas realizadas y resultados

El presente proyecto tiene como objetivo comprobar si es válido aplicar técnicas de análisis de texturas sobre imágenes de texto para reconocer el tipo de escritura, siendo uno de las utilidades principales del mismo el de servir de etapa previa a un sistema de reconocimiento óptico de caracteres (OCR).

Los OCR suelen emplearse para convertir documentos físicos en ficheros digitales, de ahí la importancia de que nuestro sistema sea capaz de trabajar con documentos escaneados con todas las imperfecciones y problemáticas que ello conlleva. El paso previo y fundamental que se debía llevar a cabo para hacer esto posible era la confección de una base de datos completa que contara con un amplio abanico de tipos de papel, fuentes y estilos de texto, para así generalizar aún más el comportamiento de nuestro sistema.

Cuando se tuvo la base de datos diseñada, confeccionada y verificada, pasamos a adaptar el enfoque seguido en el estudio de laboratorio al nuevo marco de trabajo, razón por la cual se tuvo que establecer una nueva forma de proceder.

Para empezar, cambiamos la manera de efectuar los experimentos. Anteriormente, se enfrentaban todos los tipos de escritura y mediante la tabla de confusión se determinaba cuáles de ellos eran más difíciles de distinguir usando patrones locales. Este planteamiento era factible porque trabajábamos con muy pocas muestras, pues recordemos que la base de datos inicial estaba compuesta de doscientas imágenes, concretamente, teníamos veinte documentos en cada una de las diez escrituras a distinguir.

Sin embargo, la nueva base de datos contiene muchísimas más muestras, que aumentan de forma exponencial a medida que reducimos el tamaño del texto de análisis. Por consiguiente, decidimos realizar pruebas enfrentando dos escrituras concretas. Además, con objeto de acercarnos aún más a la realidad, consultamos a compañeros del *Indian Statistical Institute* para conocer cuáles eran las combinaciones de escrituras que se daban en la documentación hindú con más frecuencia. Confeccionamos una tabla con las más probables (ver tabla 8).

Experimentos		
Nº de experimento	***Escritura 1***	***Escritura 2***
1	Latina	Bengalí
2	Oriya	Bengalí
3	Latina	Oriya
4	Latina	Árabe
5	Latina	Gujarati
6	Latina	Gurmukhi
7	Latina	Hindi
8	Latina	Japonesa
9	Latina	Canarés
10	Latina	Malayalam
11	Latina	Telugu
12	Latina	Tailandesa
13	Hindi	Bengalí
14	Hindi	Gujarati
15	Hindi	Gurmukhi
16	Hindi	Canarés
17	Hindi	Malayalam
18	Hindi	Telugu

Tabla 9. Experimentos previstos para identificar la escritura en texto impreso.

Como se puede percibir, la mayor parte de las combinaciones que se contemplan se constituyen manteniendo como protagonistas los alfabetos latino e hindi.

El motivo por el cual se planteó de esta manera es que la escritura latina puede aparecer en palabras concretas en documentos redactados en el resto de clases contenidas en nuestra base de datos. De hecho, en el momento de la selección de los bloques de texto

que conformaban los documentos, tuvimos que descartar un gran número de ellos porque ocurría esto mismo.

El hindi por otro lado, es la lengua oficial de la India, país donde este sistema resultaría de gran utilidad. Los alfabetos a los que éste aparece ligado en la tabla son tipos de escritura que se dan en regiones concretas de este país, empleándose en lenguas también oficiales pero de menos relevancia que el hindi. Por todo esto, es importante que llevemos a cabo las combinaciones que se indican en la tabla para así acercarnos en la medida de lo posible a las situaciones a las que nos enfrentaríamos si aplicáramos nuestro planteamiento a la realidad.

Descrita la hoja de ruta a seguir en lo relativo a las comprobaciones que debe llevar a cabo nuestro sistema, pasamos a diseñar las distintas etapas con las que debemos contar para que el proceso de identificación se complete con éxito.

Así, de forma genérica y de acuerdo con el estudio de laboratorio realizado a priori, se establece que en primer lugar han de definirse qué conjuntos de muestras estarán destinados a entrenamiento y test. Posteriormente, deben generarse los modelos de cada tipo de escritura aplicando los distintos patrones locales sobre las imágenes de la base de datos, para así poder llevar a cabo la creación de los vectores de características de entrenamiento y de test que servirán de entrada al clasificador LS-SVM. Y para acabar, se ha de realizar el proceso de clasificación.

Figura 46. Esquema general del funcionamiento del sistema de reconocimiento del tipo de escritura

En los próximos apartados se explicarán de forma detallada cada una de estas etapas y se mostrarán los resultados del método completo en el estudio realizado con documentos, líneas y palabras impresas.

5.6.1. Diseño de la secuencia de entrenamiento

En el estudio de laboratorio se verificó que el porcentaje de muestras de entrenamiento con mejores resultados era del 30%. Consecuentemente, se dispuso en este nuevo trabajo que lo apropiado era mantener esta cantidad de muestras para entrenar el sistema como punto de partida.

Se desarrolló un programa para completar esta tarea. Dicho software se encargaba de almacenar en un fichero de texto los nombres de las imágenes que constituían el conjunto de muestras de entrenamiento y en otro, el que se empleaba para test.

De acuerdo con esto, se debía introducir previamente el nombre de aquellas escrituras que se pretendía enfrentar, el nivel al que se requería el estudio (documentos, líneas o palabras) y los porcentajes de muestras entrenamiento para cada uno de los alfabetos a caracterizar, lo cual permitía que un tipo de escritura pudiera estar más entrenada que otra para ver los efectos de este hecho.

Hemos de tener en cuenta que, dado que nuestra base de datos no cuenta con el mismo número de imágenes en todas las categorías como se pudo comprobar en la Tabla 7, resulta imperativo realizar un ajuste que permita que el número de imágenes destinadas a la operación de entrenamiento sea similar en ambas clases. Por consiguiente, una vez establecidos los porcentajes de entrenamiento, el programa calcula la moda del total de imágenes de los alfabetos a analizar y a partir de estos datos calcula la moda, tal que este es el valor sobre el que se aplican los porcentajes fijados. De esta manera conseguimos entrenar con el mismo número de imágenes de cada escritura, siempre que el porcentaje de entrenamiento coincida en ambas clases.

Un paso adicional que efectúa este sistema al trabajar con líneas consiste en descartar aquellas con muy pocas palabras. La motivación de esta condición radica en que pretendemos hacer un estudio a nivel de línea que podría verse afectado si tratásemos realmente con una única palabra, ya que en esta última el nivel de información que podemos extraer es mucho menor, afectando sobre manera a los resultados que aportan los patrones locales al procesar la línea completa. Para llevar a cabo esta tarea, nuestro desarrollo analiza cada línea, determinando si el contenido de las mismas es suficiente como para ser procesada.

Así, se cumple con este propósito realizando un histograma vertical de la imagen y comprobando si existe un mayor porcentaje de franjas vacías en la misma en comparación con aquellas que contienen texto. En caso de hallarnos ante una línea excesivamente corta

ésta pasa a ser descartada y si por el contrario, es lo suficientemente larga se anota su nombre en el fichero de entrenamiento o test que corresponda.

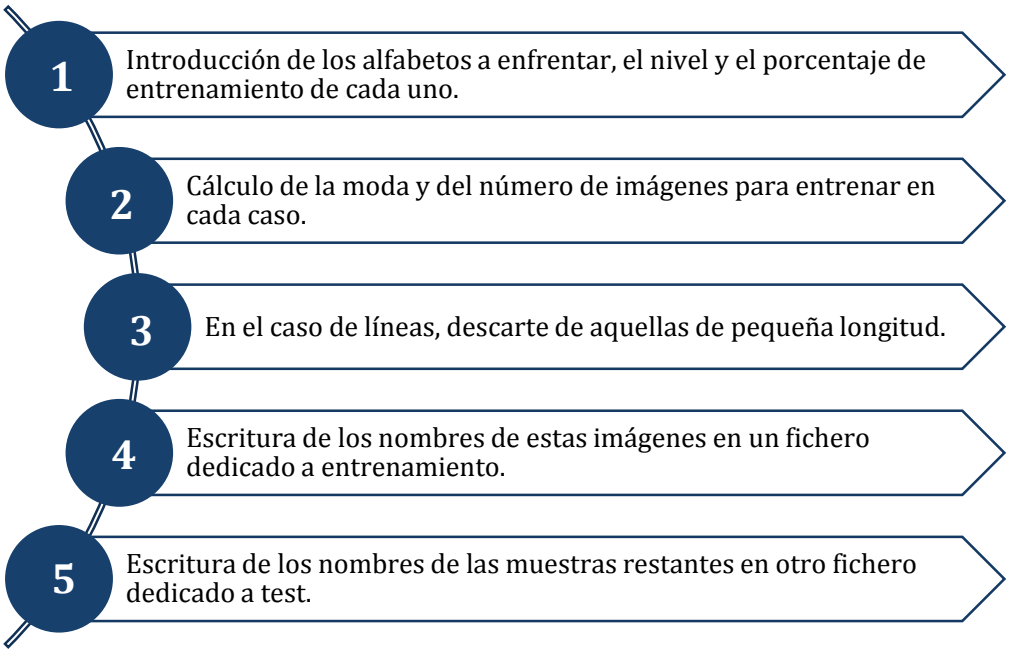


Figura 47. Pasos seguidos en la selección de las muestras dedicadas a entrenamiento y test

5.6.2. Procedimiento de extracción de características

La extracción de características se basa en los mismos conceptos que emplea el estudio inicial: se procesan las imágenes, se dividen en un cierto número de bloques, se aplican los patrones locales a cada uno de ellos, se calcula el histograma en cada caso, se concatenan los resultados y se normaliza la energía.

La primera diferencia es que la lectura de datos se realiza de acuerdo con el listado de nombres contenidos en los ficheros que se generaron durante la fase inicial de división de la base de datos en entrenamiento y test. Así, si está generando el vector de características de las muestras dedicadas a entrenar el sistema, se extraen las imágenes una a una de acuerdo con el listado que corresponda según el alfabeto y el nivel sobre el que se realiza el estudio

Otro aspecto a destacar es que en nuestro primer desarrollo una de las fases más importantes consistía en separar los documentos en palabras. En esta ocasión las imágenes

ya se introducen en el sistema en el nivel concreto al que deben ser procesadas, razón por la cual se elimina toda esta etapa.

Por otro lado, una de las novedades más notables que se introdujo en el nuevo programa de extracción fue la necesidad de indicar los alfabetos con los que se quería trabajar y a qué nivel. Recordemos que nuestro primer sistema calculaba los parámetros de todos los tipos de escritura de la base de datos, que estaba compuesta de doscientos documentos. Durante este proceso se añadían todos los valores obtenidos en una matriz de parámetros, la cual se dividía en entrenamiento y test al iniciar el proceso de clasificación. En el nuevo software se generan directamente los vectores de entrenamiento y test de las escrituras de análisis, motivo por el cual es imprescindible que se seleccionen previamente las escrituras sobre las que se va a trabajar y a qué nivel se encuentran.

En esta línea y como detallaremos más adelante, dependiendo del nivel al que se trabaje las tareas que se incluyen en el procesado de la imagen varían, pues las muestras llegan al sistema en condiciones distintas. A pesar de ello, todas las imágenes se convierten a formato lógico con independencia de si se trata de documentos, líneas y palabras, antes de ser divididas y de aplicar las técnicas de análisis de texturas.

Por otra parte, la división de las muestras de texto en bloques también guarda dependencia con el nivel de las mismas. Concretamente, en las palabras y las líneas la división en bloques se ejecuta de forma análoga a como se hacía en el estudio inicial, es decir, cada muestra se fragmenta en cuatro bloques horizontales y son estos los que se analizan usando los patrones locales. Pero si centramos nuestra atención en los documentos, no sería un planteamiento válido, pues en la mayor parte de los casos los patrones locales se aplicarían sobre grandes superficies de texto pudiendo ser de una o varias líneas. Es por esto que los documentos se dividen en bloques de unas dimensiones concretas, que establecimos de forma semejante a como lo hacían diversos autores de estudios sobre identificación de la escritura usando filtros de Gabor, los cuales se encuentran incluidos en el segundo capítulo y la bibliografía.

En todos los casos la extracción de características como tal, se ejecutaba de la misma forma. Así, aplicamos los patrones locales LBP, OLBP, LDP y LDerivP sobre cada uno de los fragmentos de texto y realizamos el histograma de los datos resultantes, tal que cuando se obtenían los valores de la totalidad de los bloques se concatenaban y se normalizaba la energía. Coincidiendo con los pasos seguidos en el primer estudio.

Una vez hayan sido analizadas todas las imágenes de la base de datos de las escrituras que se especificaron al inicio del mismo, pasan a guardarse los vectores de características. Quisimos que el software encargado de la parametrización fuera lo más general posible y por consiguiente, antes de la ejecución del mismo era necesario establecer qué vector es el que se pretende calcular, es decir, debe especificarse si se desea generar el de entrenamiento o el de test. Por todo esto, al final de esta etapa se guardan dos vectores independientes por cada escritura, uno de entrenamiento y otro de test. Cada vector se nombraba haciendo mención al nombre de la escritura y al nivel en que se estuviese trabajando.

Asimismo, no sólo se guardaban los datos resultantes sino que también se creaba un vector de índices, tal que cada imagen procesada era registrada a la vez que se guardaban los valores derivados de aplicar los patrones locales. La generación de este elemento posibilitaba una mejor detección de errores.

Para resumir todo lo expresado en este apartado se presenta un esquema con la sucesión de acciones que se llevan a cabo para realizar la extracción de características de la nueva base de datos en la Figura 48.

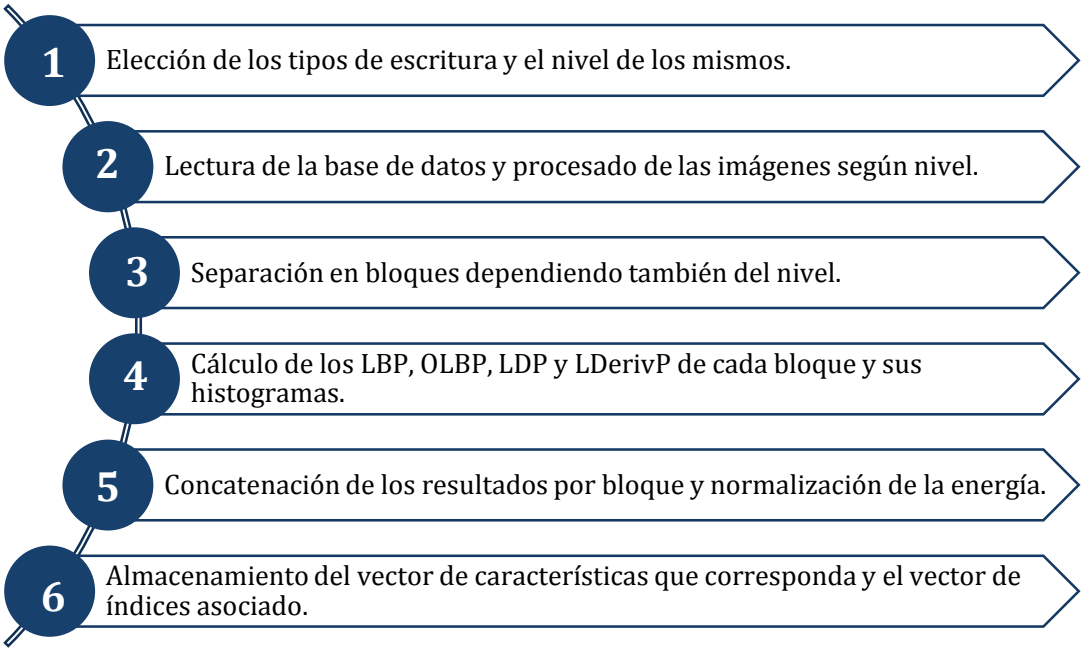


Figura 48. Pasos seguidos en la extracción de características de las imágenes de la base de datos

5.6.3. Ajuste del clasificador

Una de las principales ventajas de guardar cuatro vectores cada vez que ejecutábamos el sistema de parametrización era que se permitía que éstos fueran aprovechados en más de un experimento, sin necesidad de repetir varias veces la generación de vectores de la misma clase de escritura. No obstante, la principal consecuencia de esta manera de proceder era que se forzaba la unión de los vectores de entrenamiento y test de cada alfabeto de análisis antes de comenzar la fase de clasificación y ser introducidos en el LS-SVM.

Con este propósito se creó un programa que buscaba los vectores de las dos escrituras involucradas en un experimento concreto mediante su nombre. Luego, los concatenaba creando entonces los vectores de índices, entrenamiento y test conjuntos de la prueba. Para acabar, todos ellos eran guardados en la carpeta que contenía las funciones que componían el clasificador LS-SVM, para que este los tomara durante su ejecución.

En cuanto al uso del clasificador, no hay ningún cambio relevante. El único detalle a tener en cuenta es que el clasificador LS-SVM cuenta con dos variables que han de ajustarse según la naturaleza de los datos a clasificar. Éstos se denominan gamma y sigma, y son los parámetros que permiten que el hiperplano de separación entre clases se trace adecuadamente.

Al final el proceso de clasificación, se obtenían los resultados en los mismos formatos que en el estudio de laboratorio. Todos ellos se comentan en los apartados siguientes.

5.6.4. Estudio por documentos

La primera consideración que se efectúa al trabajar con documentos es que las imágenes se encuentran guardadas a color, y ello implica que sea necesario transformar la imagen a escala de grises, luego a formato lógico y finalmente invertirla, antes de proceder a la extracción de características.

Por otro lado, una de las incógnitas a resolver en el presente proyecto es determinar cuán buenas son las técnicas de análisis de imagen aplicadas al reconocimiento del tipo de escritura en comparación con los métodos ya existentes, como por ejemplo los filtros de Gabor, que son los más ampliamente utilizados y que obtienen unos mejores resultados.

En la mayor parte de la bibliografía, estos filtros no se aplican sobre el documento completo, sino que analizan la imagen parte por parte. Es por esto que decidimos que, después de transformar el documento en una imagen en formato lógico y antes de calcular

los patrones locales de la imagen, debíamos dividirlo en fragmentos de un tamaño determinado.

Concretamente, se propuso extraer bloques de 128x128 píxeles solapados un 15% como se hacía en varios trabajos incluidos la bibliografía. Tal que estos fragmentos eran los que se analizaban con las técnicas de textura propuestas.

Las líneas de código que se ejecutaban para realizar esta tarea determinaban en primer lugar en cuántos bloques de 128x128 píxeles se podía dividir el documento y para finalizar, mediante el movimiento de punteros en vertical y horizontal, se seleccionaban y almacenaban aquellos fragmentos de la imagen que se habían previsto.

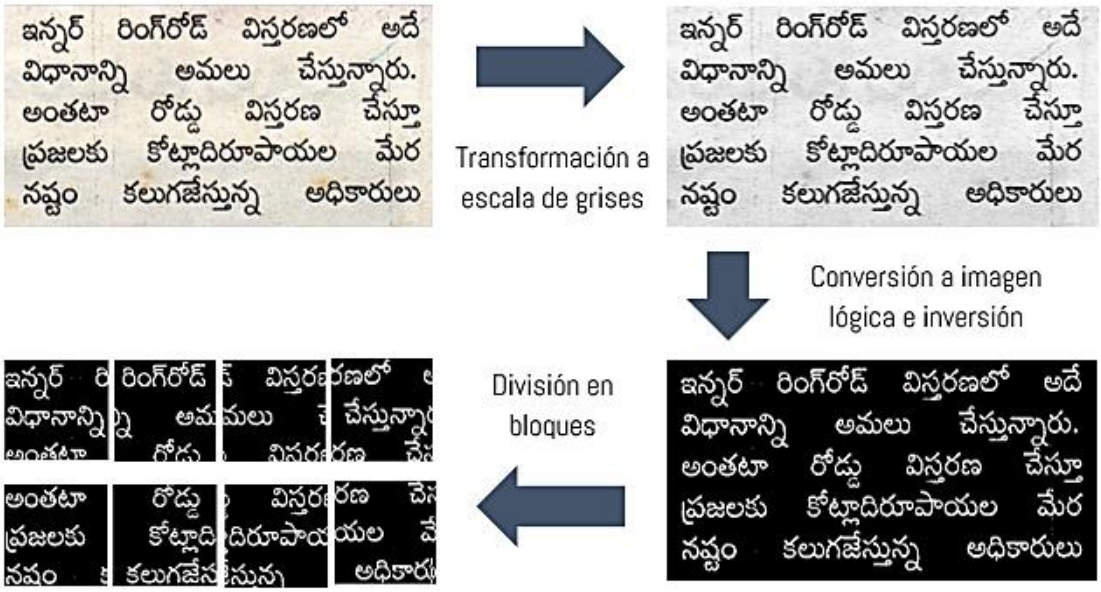


Figura 49. Procedimiento de división de documentos en bloques de 128x128 píxeles solapados un 15%.

Aquellas zonas de la imagen que quedaban fuera de la división en bloques del documento se descartaban, puesto que consideramos que la información aportada por estos fragmentos de imagen con poco texto no tenía gran relevancia, por lo que el hecho de eliminarlos no afecta a los resultados finales.

Otra de las acciones que se llevaban a cabo con anterioridad al cálculo de los patrones locales era comprobar si estos fragmentos estaban vacíos o si su contenido no era suficiente como para obtener resultados válidos. Para ello se estableció que se descartarían aquellos bloques que no contuvieran más de un 20% de sus píxeles ocupados por el texto.

Todo esto sumado a los procedimientos generales de extracción de características y clasificación descritos en puntos anteriores, dio lugar los resultados expresados en términos de tasa de reconocimiento que se representan en la Tabla 10. Cabe mencionar que los experimentos se efectúan tomando un 30% de las muestras de ambas escrituras para entrenar y el resto para testear.

1ª Escritura	2ª Escritura	LBP	OLBP	LDP	LDerivP
Árabe	Latina	94,229%	96,608%	97,679%	94,186%
Bengalí	Latina	99,444%	99,610%	98,887%	99,499%
Gujarati	Latina	95,548%	96,992%	97,920%	95,870%
Gurmukhi	Latina	59,791%	63,989%	63,864%	59,332%
Hindi	Latina	99,527%	97,549%	98,228%	99,527%
Japonesa	Latina	99,288%	99,779%	98,821%	99,189%
Canarés	Latina	98,327%	98,595%	98,829%	98,494%
Malayalam	Latina	97,292%	91,425%	95,197%	97,260%
Oriya	Latina	94,184%	94,794%	93,574%	94,517%
Telugu	Latina	97,297%	97,150%	97,709%	97,415%
Tailandesa	Latina	99,134%	98,484%	98,917%	99,025%
Bengalí	Hindi	94,416%	82,772%	86,007%	94,653%
Gujarati	Hindi	69,329%	70,313%	73,238%	70,955%
Gurmukhi	Hindi	51,059%	53,029%	53,773%	50,898%
Canarés	Hindi	69,152%	83,001%	78,406%	67,609%
Malayalam	Hindi	97,034%	99,364%	97,881%	97,034%
Oriya	Hindi	84,375%	75,893%	77,041%	86,820%
Telugu	Hindi	85,738%	97,399%	89,597%	82,215%
Bengalí	Oriya	97,558%	98,126%	98,467%	97,615%

Tabla 10. Resultados de los experimentos sobre texto impreso a nivel de documentos aplicando técnicas de análisis de texturas.

El clasificador también podía combinar dos o más parámetros de textura a la hora de reconocer la escritura. Así, en las Tablas 11-a y 11-b se encuentran recogidos los resultados

si se combinan dos de los patrones locales, en la Tabla 12 si se usan tres de éstos y en la Tabla 13 se muestran los valores obtenidos al combinar los efectos de las cuatro técnicas.

1ª Escritura	2ª Escritura	LBP + OLBP	LBP + LDP	LBP + LDerivP
Árabe	Latina	97,888%	98,611%	95,605%
Bengalí	Latina	99,777%	99,555%	99,722%
Gujarati	Latina	98,275%	98,529%	97,277%
Gurmukhi	Latina	64,137%	64,013%	61,007%
Hindi	Latina	99,823%	99,764%	99,764%
Japonesa	Latina	99,926%	99,632%	99,632%
Canarés	Latina	99,610%	99,498%	99,052%
Malayalam	Latina	96,809%	98,195%	98,711%
Oriya	Latina	96,435%	95,410%	96,451%
Telugu	Latina	97,944%	98,414%	97,650%
Tailandesa	Latina	99,648%	99,621%	99,567%
Bengalí	Hindi	91,782%	91,485%	96,238%
Gujarati	Hindi	72,334%	75,344%	71,365%
Gurmukhi	Hindi	53,011%	52,339%	51,360%
Canarés	Hindi	81,381%	79,306%	71,223%
Malayalam	Hindi	98,729%	98,729%	98,305%
Oriya	Hindi	81,250%	81,633%	86,534%
Telugu	Hindi	97,148%	92,617%	88,909%
Bengalí	Oriya	99,148%	99,148%	99,659%

Tabla 11-a. Resultados de los experimentos sobre texto impreso a nivel de documentos aplicando el LBP junto con otro de los patrones locales.

1ª Escritura	2ª Escritura	OLBP + LDP	OLBP + LDerivP	LDP + LDerivP
Árabe	Latina	98,782%	97,793%	98,592%
Bengalí	Latina	99,777%	99,777%	99,555%
Gujarati	Latina	98,782%	98,376%	98,579%

Gurmukhi	Latina	66,020%	63,907%	63,449%
Hindi	Latina	98,848%	99,744%	99,705%
Japonesa	Latina	99,779%	99,926%	99,484%
Canarés	Latina	99,582%	99,610%	99,442%
Malayalam	Latina	95,551%	96,217%	98,130%
Oriya	Latina	95,713%	97,123%	96,173%
Telugu	Latina	98,002%	97,767%	98,355%
Tailandesa	Latina	99,729%	99,811%	99,729%
Bengalí	Hindi	89,228%	91,551%	91,683%
Gujarati	Hindi	74,596%	74,116%	75,895%
Gurmukhi	Hindi	54,146%	52,876%	52,113%
Canarés	Hindi	84,961%	84,169%	80,366%
Malayalam	Hindi	99,576%	98,941%	98,517%
Oriya	Hindi	75,638%	85,102%	84,439%
Telugu	Hindi	95,190%	97,148%	90,772%
Bengalí	Oriya	99,659%	98,808%	99,148%

Tabla 11-b. Resultados de los experimentos sobre texto impreso a nivel de documentos aplicando las combinaciones restantes usando dos patrones locales.

1ª Escritura	2ª Escritura	LBP + OLBP + LDP	LBP + OLBP + LDerivP	LBP + LDP + LDerivP	OLBP + LDP + LDerivP
Árabe	Latina	99,201%	98,135%	98,719%	99,172%
Bengalí	Latina	99,777%	99,833%	99,777%	99,777%
Gujarati	Latina	99,061%	98,808%	98,833%	99,036%
Gurmukhi	Latina	65,494%	63,274%	62,983%	65,332%
Hindi	Latina	99,882%	99,941%	99,882%	99,823%
Japonesa	Latina	99,889%	99,926%	99,853%	99,853%
Canarés	Latina	99,749%	99,777%	99,721%	99,721%
Malayalam	Latina	97,679%	98,001%	98,839%	97,485%
Oriya	Latina	96,388%	97,683%	96,643%	97,123%

Telugu	Latina	98,472%	98,120%	98,472%	98,179%
Tailandesa	Latina	99,783%	99,783%	99,783%	99,946%
Bengalí	Hindi	92,475%	94,851%	95,446%	94,059%
Gujarati	Hindi	74,161%	73,015%	75,006%	76,018%
Gurmukhi	Hindi	53,319%	52,277%	51,559%	53,188%
Canarés	Hindi	83,644%	82,423%	80,163%	85,090%
Malayalam	Hindi	99,576%	98,729%	98,941%	99,364%
Oriya	Hindi	80,357%	86,607%	86,607%	82,653%
Telugu	Hindi	95,973%	97,148%	92,617%	95,470%
Bengalí	Oriya	99,148%	99,489%	99,489%	99,319%

Tabla 12. Resultados de los experimentos sobre texto impreso a nivel de documentos usando todas las combinaciones posibles con tres patrones locales.

1ª Escritura	2ª Escritura	LBP + OLBP + LDP + LDerivP
Árabe	Latina	99,239%
Bengalí	Latina	99,833%
Gujarati	Latina	99,137%
Gurmukhi	Latina	64,175%
Hindi	Latina	99,941%
Japonesa	Latina	100,000%
Canarés	Latina	99,888%
Malayalam	Latina	98,711%
Oriya	Latina	97,353%
Telugu	Latina	98,649%
Tailandesa	Latina	99,892%
Bengalí	Hindi	95,000%
Gujarati	Hindi	74,944%
Gurmukhi	Hindi	52,609%
Canarés	Hindi	84,242%

Malayalam	Hindi	99,576%
Oriya	Hindi	84,736%
Telugu	Hindi	96,141%
Bengalí	Oriya	99,319%

Tabla 13. Resultados de los experimentos sobre texto impreso a nivel de documentos combinando los efectos de todos los patrones locales.

Como se puede comprobar los métodos con los que se obtiene una mejor respuesta son aquellos donde el OLBP y el LDP aparecen, lo cual verifica la conclusión extraída del primer estudio y es que las direcciones de los trazos son elementos fundamentales en la identificación del tipo de escritura.

Analizando todas las tablas, vemos también que las tasas de reconocimiento aumentan a medida que combinamos un mayor número de patrones locales, alcanzando las más altas mediante la mezcla de todos ellos en la mayor parte de los casos que se plantean.

No obstante, aunque podemos afirmar que los resultados obtenidos son buenos en general, hay experimentos donde las tasas de acierto son tan bajas que indican que prácticamente el sistema no es capaz de distinguir las escrituras. Esto ocurre especialmente al trabajar con el alfabeto gurmukhi y ello parece deberse a que el aspecto de este alfabeto es tal que si lo comparamos con el hindi, encontramos un alto grado de similitud a simple vista; y si lo comparamos con el alfabeto latino, presenta una estructura parecida.

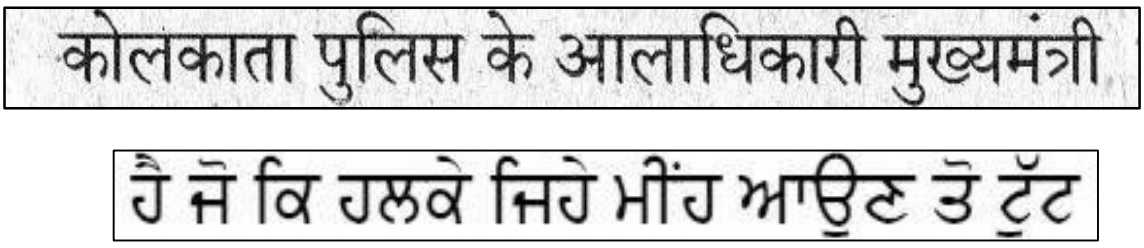


Figura 50. Líneas en Hindi y Gurmukhi donde se aprecia su similitud visual.

5.6.5. Estudio por líneas

Las líneas de la base de datos se encuentran en escala de grises, ergo es necesario convertir la imagen a formato lógico e invertirla con anterioridad al cálculo de los patrones locales.

Una de las grandes ventajas de este conjunto es que nuestro estudio inicial trabajaba con líneas y ello nos permitió corroborar que nuestro enfoque al tratar con ellas era suficientemente válido. Por consiguiente, el proceso seguido con las líneas se mantuvo tal y como estaba, esto quiere decir que las líneas con el fondo negro y el texto en blanco resultantes del paso anterior se dividían en cuatro bloques horizontales, se calculaban los LBP de cada franja, se realizaba el histograma de los valores adquiridos y finalmente, se concatenaban los datos de dichos bloques para finalmente normalizar la energía.

En esta línea y procediendo de forma análoga a la exposición de los resultados a nivel de documentos las tasas de reconocimiento tras la aplicación de los métodos de análisis de texturas sobre líneas de texto se representan en la Tabla 14. Tomando también en este caso un 30% de las muestras de ambas escrituras para entrenar y el resto para testear.

1ª Escritura	2ª Escritura	LBP	OLBP	LDP	LDerivP
Árabe	Latina	92,251%	92,385%	92,318%	92,318%
Bengalí	Latina	98,941%	100,000%	100,000%	100,000%
Gujarati	Latina	96,399%	99,909%	99,362%	99,180%
Gurmukhi	Latina	92,595%	92,867%	92,867%	92,799%
Hindi	Latina	99,729%	100,000%	100,000%	100,000%
Japonesa	Latina	98,841%	99,657%	99,957%	99,828%
Canarés	Latina	99,491%	100,000%	98,728%	99,703%
Malayalam	Latina	97,484%	97,971%	96,726%	96,063%
Oriya	Latina	98,886%	99,829%	99,686%	99,914%
Telugu	Latina	99,122%	100,000%	99,298%	99,298%
Tailandesa	Latina	98,760%	99,646%	99,646%	100,000%
Bengalí	Hindi	87,883%	88,190%	85,123%	88,344%
Gujarati	Hindi	92,674%	98,718%	99,817%	97,527%
Gurmukhi	Hindi	55,435%	55,435%	55,435%	55,435%

Canarés	Hindi	76,413%	77,150%	77,150%	76,536%
Malayalam	Hindi	68,237%	68,237%	68,237%	68,237%
Oriya	Hindi	80,000%	80,000%	80,000%	80,000%
Telugu	Hindi	86,539%	86,982%	86,834%	86,834%
Bengalí	Oriya	100,000%	100,000%	99,592%	99,864%

Tabla 14. Resultados de los experimentos sobre texto impreso a nivel de líneas aplicando técnicas de análisis de texturas.

Asimismo, mediante la mezcla de dos de los parámetros de textura para la identificación de la clase de escritura, se obtuvieron los valores expresados en las Tablas 15-a y 15-b,

1ª Escritura	2ª Escritura	LBP + OLBP	LBP + LDP	LBP + LDerivP
Árabe	Latina	92,385%	92,183%	92,251%
Bengalí	Latina	100,000%	100,000%	100,000%
Gujarati	Latina	99,362%	98,997%	98,906%
Gurmukhi	Latina	93,614%	94,565%	95,477%
Hindi	Latina	100,000%	100,000%	100,000%
Japonesa	Latina	99,828%	99,828%	99,742%
Canarés	Latina	100,000%	99,746%	99,915%
Malayalam	Latina	99,391%	98,539%	98,620%
Oriya	Latina	99,914%	100,000%	100,000%
Telugu	Latina	99,912%	99,649%	99,473%
Tailandesa	Latina	99,823%	99,734%	100,000%
Bengalí	Hindi	88,497%	87,922%	88,267%
Gujarati	Hindi	98,535%	98,535%	97,070%
Gurmukhi	Hindi	55,758%	56,087%	56,114%
Canarés	Hindi	77,027%	76,659%	75,799%
Malayalam	Hindi	68,237%	68,237%	68,136%
Oriya	Hindi	80,000%	80,000%	80,000%

Telugu	Hindi	86,834%	86,686%	86,982%
Bengalí	Oriya	100,000%	99,864%	100,000%

Tabla 15-a. Resultados de los experimentos sobre texto impreso a nivel de líneas aplicando el LBP junto con otro de los patrones locales.

1ª Escritura	2ª Escritura	OLBP + LDP	OLBP + LDerivP	LDP + LDerivP
Árabe	Latina	92,385%	92,385%	92,318%
Bengalí	Latina	100,000%	100,000%	100,000%
Gujarati	Latina	100,000%	99,818%	99,453%
Gurmukhi	Latina	92,935%	93,003%	93,682%
Hindi	Latina	100,000%	100,000%	100,000%
Japonesa	Latina	100,000%	100,000%	100,000%
Canarés	Latina	100,000%	100,000%	99,830%
Malayalam	Latina	98,945%	98,458%	98,377%
Oriya	Latina	99,914%	100,000%	100,000%
Telugu	Latina	99,956%	100,000%	99,737%
Tailandesa	Latina	99,646%	99,911%	99,823%
Bengalí	Hindi	87,781%	88,344%	88,037%
Gujarati	Hindi	100,000%	99,451%	99,817%
Gurmukhi	Hindi	56,046%	55,978%	56,612%
Canarés	Hindi	77,150%	77,027%	76,331%
Malayalam	Hindi	68,136%	68,337%	68,237%
Oriya	Hindi	80,000%	80,000%	80,000%
Telugu	Hindi	86,243%	86,834%	86,834%
Bengalí	Oriya	100,000%	100,000%	100,000%

Tabla 15-b. Resultados de los experimentos sobre texto impreso a nivel de líneas aplicando las combinaciones restantes usando dos patrones locales.

Análogamente, si se usaban tres de los cuatro parámetros locales las tasas de reconocimiento quedaban como sigue:

1ª Escritura	2ª Escritura	LBP + OLBP + LDP	LBP + OLBP + LDerivP	LBP + LDP + LDerivP	OLBP + LDP + LDerivP
Árabe	Latina	92,385%	92,385%	92,318%	92,385%
Bengalí	Latina	100,000%	100,000%	100,000%	100,000%
Gujarati	Latina	99,909%	99,727%	99,180%	99,909%
Gurmukhi	Latina	95,177%	95,448%	96,467%	94,375%
Hindi	Latina	100,000%	100,000%	100,000%	100,000%
Japonesa	Latina	100,000%	100,000%	100,000%	100,000%
Canarés	Latina	100,000%	100,000%	100,000%	100,000%
Malayalam	Latina	99,675%	99,675%	99,269%	99,432%
Oriya	Latina	100,000%	100,000%	100,000%	100,000%
Telugu	Latina	100,000%	100,000%	99,824%	99,912%
Tailandesa	Latina	99,823%	100,000%	100,000%	99,911%
Bengalí	Hindi	89,264%	89,110%	89,571%	88,957%
Gujarati	Hindi	99,817%	99,634%	99,451%	100,000%
Gurmukhi	Hindi	54,294%	54,552%	55,254%	54,891%
Canarés	Hindi	77,027%	76,904%	76,659%	77,150%
Malayalam	Hindi	68,136%	68,036%	68,036%	68,036%
Oriya	Hindi	80,000%	80,000%	80,000%	80,000%
Telugu	Hindi	86,982%	86,982%	86,982%	86,982%
Bengalí	Oriya	100,000%	100,000%	100,000%	100,000%

Tabla 16. Resultados de los experimentos sobre texto impreso a nivel de líneas usando todas las combinaciones posibles con tres patrones locales.

Finalmente, si usamos todos los patrones locales los porcentajes quedan de la forma que aparece ilustrada en la Tabla 17, la cual se presenta a continuación:

1ª Escritura	2ª Escritura	LBP + OLBP + LDP + LDerivP
Árabe	Latina	92,453%
Bengalí	Latina	100,000%
Gujarati	Latina	99,818%
Gurmukhi	Latina	95,245%
Hindi	Latina	100,000%
Japonesa	Latina	100,000%
Canarés	Latina	100,000%
Malayalam	Latina	99,756%
Oriya	Latina	100,000%
Telugu	Latina	100,000%
Tailandesa	Latina	100,000%
Bengalí	Hindi	89,494%
Gujarati	Hindi	99,817%
Gurmukhi	Hindi	54,076%
Canarés	Hindi	76,904%
Malayalam	Hindi	68,237%
Oriya	Hindi	80,000%
Telugu	Hindi	86,834%
Bengalí	Oriya	100,000%

Tabla 17. Resultados de los experimentos sobre texto impreso a nivel de líneas combinando los efectos de todos los patrones locales.

Las líneas de texto permiten un mejor análisis de los trazos y permiten que el sistema modele mucho mejor cada tipo de escritura en comparación con los documentos, ya que estos últimos contenían varios fragmentos de texto que hacían que fuera algo más compleja la solución del problema con la aplicación de los patrones locales Es por esto que en la mayor parte de los casos, las el análisis a nivel de líneas logra un mejor resultado.

Por otro lado, se verifica una vez más que los parámetros que consiguen una tasa de reconocimiento mayor son el OLBP y el LDP. También se cumple que a medida que usamos más métodos combinados el efecto se vuelve más positivo.

Otro hecho destacable es que a este nivel el sistema ya es capaz de distinguir adecuadamente las escrituras latina y gurmukhi, y sin embargo, mantiene unos pésimos resultados al enfrentar los alfabetos hindi y gurmukhi.

5.6.6. Estudio por palabras

El estudio a nivel de palabras es el más interesante de todos los que hemos completado hasta ahora. El motivo de esta afirmación es que en los textos con más de un tipo de escritura lo más común es encontrar una palabra contenida en una frase con un tipo de escritura distinto. En consecuencia, este es el apartado con el abanico más amplio de posibles aplicaciones.

Las imágenes de las palabras no coinciden en términos de aspecto con las líneas o los documentos, ya que se representan directamente en blanco y negro como si ya hubieran sido sometidas a un proceso binarización y se tratase de imágenes lógicas con el fondo negro y los caracteres de texto en blanco. No obstante, el sistema las trataba como imágenes en escala de grises que había que transformar a binarias con la diferencia de que al final de este paso no había necesidad de invertir la imagen resultante.

Al igual que ocurría con las líneas, las palabras se dividían en cuatro franjas horizontales y su caracterización se ejecutaba también de la misma forma. El motivo por el que se actuó así era que esta separación permitía detectar ciertas características propias de cada tipo de escritura presentes en sus caracteres o palabras. Por ejemplo, hay muy pocas escrituras cuyos trazos se distribuyan de forma uniforme como ocurre con el hindi o el gurmukhi, otras pueden distinguirse de estas por la existencia de un patrón distinto con símbolos que aparecen en ocasiones en la parte inferior de las palabras como en el canarés, o en la superior, como sucede en la escritura tailandesa. Por todo lo dicho este procedimiento también resulta válido en el caso de las palabras.

Así, se muestran a continuación las tasas de reconocimiento a nivel de palabras, los cuales son fruto de la aplicación de los patrones locales LBP, OLBP, LDP y LDeriP. Escogiendo también en esta ocasión un 30% de las imágenes de las escrituras enfrentadas para entrenar y el resto para testear.

1ª Escritura	2ª Escritura	LBP	OLBP	LDP	LDerivP
Árabe	Latina	99,670%	99,535%	99,482%	99,541%
Bengalí	Latina	99,624%	99,435%	99,329%	99,555%
Gujarati	Latina	97,892%	97,684%	98,565%	97,823%
Gurmukhi	Latina	80,696%	80,731%	80,602%	80,804%
Hindi	Latina	99,640%	99,210%	99,269%	99,502%
Japonesa	Latina	98,046%	97,548%	96,051%	96,029%
Canarés	Latina	98,690%	98,627%	98,705%	98,970%
Malayalam	Latina	91,248%	91,414%	90,545%	92,220%
Oriya	Latina	97,237%	98,539%	97,581%	97,283%
Telugu	Latina	98,851%	98,543%	98,293%	98,535%
Tailandesa	Latina	98,460%	96,701%	97,144%	96,126%
Bengalí	Hindi	95,575%	95,863%	96,191%	97,632%
Gujarati	Hindi	97,595%	98,887%	99,090%	98,500%
Gurmukhi	Hindi	50,000%	50,000%	50,000%	50,000%
Canarés	Hindi	98,024%	98,467%	98,545%	97,150%
Malayalam	Hindi	70,501%	70,397%	70,475%	70,501%
Oriya	Hindi	99,809%	99,177%	99,531%	99,678%
Telugu	Hindi	95,129%	99,295%	98,372%	98,305%
Bengalí	Oriya	92,361%	92,602%	92,495%	92,602%

Tabla 18. Resultados de los experimentos sobre texto impreso a nivel de palabras aplicando técnicas de análisis de texturas.

De manera similar, se presentan las tablas de resultados obtenidos al combinar dos de los parámetros de textura para la identificación de la clase de escritura.

1ª Escritura	2ª Escritura	LBP + OLBP	LBP + LDP	LBP + LDerivP
Árabe	Latina	99,895%	99,825%	99,877%
Bengalí	Latina	99,875%	99,902%	99,957%
Gujarati	Latina	99,362%	99,395%	99,417%

Gurmukhi	Latina	82,688%	83,010%	84,068%
Hindi	Latina	99,777%	99,841%	99,894%
Japonesa	Latina	99,075%	98,676%	98,902%
Canarés	Latina	99,631%	99,604%	99,582%
Malayalam	Latina	93,695%	93,397%	94,316%
Oriya	Latina	99,077%	98,705%	98,868%
Telugu	Latina	99,612%	99,483%	99,590%
Tailandesa	Latina	99,002%	98,975%	98,754%
Bengalí	Hindi	98,288%	98,222%	98,815%
Gujarati	Hindi	99,402%	99,593%	99,258%
Gurmukhi	Hindi	49,837%	49,955%	50,001%
Canarés	Hindi	99,198%	99,377%	98,851%
Malayalam	Hindi	70,462%	70,488%	70,527%
Oriya	Hindi	99,905%	99,881%	99,976%
Telugu	Hindi	99,555%	99,150%	99,093%
Bengalí	Oriya	92,628%	92,468%	92,578%

Tabla 19-a. Resultados de los experimentos sobre texto impreso a nivel de palabras aplicando el LBP junto con otro de los patrones locales.

1ª Escritura	2ª Escritura	OLBP + LDP	OLBP + LDerivP	LDP + LDerivP
Árabe	Latina	99,807%	99,868%	99,825%
Bengalí	Latina	99,899%	99,834%	99,924%
Gujarati	Latina	99,271%	99,012%	99,301%
Gurmukhi	Latina	82,043%	82,421%	82,952%
Hindi	Latina	99,661%	99,714%	99,746%
Japonesa	Latina	98,694%	98,678%	98,168%
Canarés	Latina	99,579%	99,675%	99,677%
Malayalam	Latina	93,127%	94,212%	93,646%
Oriya	Latina	99,176%	99,295%	99,001%
Telugu	Latina	99,402%	99,512%	99,493%

Tailandesa	Latina	98,585%	98,078%	98,362%
Bengalí	Hindi	98,466%	98,931%	98,885%
Gujarati	Hindi	99,521%	99,521%	99,561%
Gurmukhi	Hindi	49,872%	50,050%	49,708%
Canarés	Hindi	99,322%	99,018%	99,066%
Malayalam	Hindi	70,462%	70,514%	70,501%
Oriya	Hindi	99,881%	99,905%	99,940%
Telugu	Hindi	99,708%	99,708%	99,538%
Bengalí	Oriya	92,602%	92,628%	92,628%

Tabla 19-b. Resultados de los experimentos sobre texto impreso a nivel de palabras aplicando las combinaciones restantes usando dos patrones locales.

Continuamos mostrando las tasas de reconocimiento de cada uno de los experimentos combinando tres de los cuatro parámetros locales disponibles.

1ª Escritura	2ª Escritura	LBP + OLBP + LDP	LBP + OLBP + LDerivP	LBP + LDP + LDerivP	OLBP + LDP + LDerivP
Árabe	Latina	99,895%	99,939%	99,895%	99,895%
Bengalí	Latina	99,978%	99,973%	99,978%	99,946%
Gujarati	Latina	99,613%	99,648%	99,681%	99,510%
Gurmukhi	Latina	82,931%	83,315%	83,609%	82,865%
Hindi	Latina	99,883%	99,862%	99,936%	99,836%
Japonesa	Latina	99,301%	99,398%	99,182%	99,149%
Canarés	Latina	99,835%	99,879%	99,780%	99,882%
Malayalam	Latina	94,387%	94,983%	94,799%	94,661%
Oriya	Latina	99,429%	99,495%	99,258%	99,539%
Telugu	Latina	99,712%	99,767%	99,690%	99,745%
Tailandesa	Latina	99,234%	99,126%	99,108%	98,899%
Bengalí	Hindi	99,024%	99,334%	99,163%	99,447%
Gujarati	Hindi	99,737%	99,677%	99,761%	99,832%

Gurmukhi	Hindi	49,911%	50,068%	50,079%	49,939%
Canarés	Hindi	99,593%	99,467%	99,407%	99,497%
Malayalam	Hindi	70,361%	70,384%	70,397%	70,319%
Oriya	Hindi	99,952%	99,976%	99,976%	99,976%
Telugu	Hindi	99,806%	99,733%	99,636%	99,854%
Bengalí	Oriya	92,708%	92,762%	92,762%	92,762%

Tabla 20. Resultados de los experimentos sobre texto impreso a nivel de palabras usando todas las combinaciones posibles con tres patrones locales.

Por último, empleando todos los patrones locales los resultados quedan como sigue:

1ª Escritura	2ª Escritura	LBP + OLBP + LDP + LDerivP
Árabe	Latina	99,921%
Bengalí	Latina	100,000%
Gujarati	Latina	99,769%
Gurmukhi	Latina	83,373%
Hindi	Latina	99,947%
Japonesa	Latina	99,554%
Canarés	Latina	99,938%
Malayalam	Latina	95,378%
Oriya	Latina	99,585%
Telugu	Latina	99,823%
Tailandesa	Latina	99,269%
Bengalí	Hindi	99,489%
Gujarati	Hindi	99,856%
Gurmukhi	Hindi	49,963%
Canarés	Hindi	99,689%
Malayalam	Hindi	70,496%
Oriya	Hindi	99,976%

Telugu	Hindi	99,879%
Bengalí	Oriya	92,788%

Tabla 21. Resultados de los experimentos sobre texto impreso a nivel de palabras combinando los efectos de todos los patrones locales.

El gran reto al que nos enfrentamos al identificar las escrituras a nivel de palabras radica en la poca información acerca de los trazos que se puede extraer a partir de una palabra, fundamentalmente en aquellas muestras donde dicha palabra esté compuesta por uno o dos caracteres.

Por consiguiente, era de esperar que las tasas de reconocimiento disminuyeran en comparación con los mostrados a nivel de línea. Aunque eso no quiere decir que los resultados obtenidos sean malos, pues recordemos que en las técnicas más usuales de reconocimiento a partir de la apariencia visual del texto, el valor más alto de acierto del sistema era del 96% el cual se supera en la mayoría de los experimentos llevados a cabo en este trabajo.

Otra de las conclusiones que se extraen de las tablas de este apartado es que el sistema es incapaz de distinguir las escrituras hindi y gurmukhi a un nivel tan bajo. Por consiguiente, una posible solución a esta problemática sería combinar estas técnicas con métodos estructurales que analicen, por ejemplo, la distribución de los píxeles en cada caso. Incluso, a simple vista parece factible contar los espacios cerrados en una línea, pues las redactadas en hindi, por lo general, tienen un mayor número de estas características que las escritas en gurmukhi. Esta puede ser una línea de trabajo futura para mejorar aún más las tasas de reconocimiento.

Capítulo 6. Comprobaciones sobre texto manuscrito

6.1. Introducción

La identificación del alfabeto en escritura manuscrita sigue siendo a día de hoy un frente abierto para la comunidad científica dedicada al análisis de documentos. De hecho, las técnicas más populares como el filtro de Gabor o el análisis de componentes conectados, no han demostrado ser eficaces sobre textos de esta naturaleza.

En el presente capítulo se incluye el estudio realizado acerca del reconocimiento de alfabetos en imágenes de escritura manuscritas usando las técnicas de análisis de texturas propuestas.

6.2. Diseño de la base de datos

Al igual que sucedía con los textos impresos, no existe en la actualidad una base de datos pública que contenga textos manuscritos en distintos tipos de escrituras o idiomas.

Es por ello que al inicio de este trabajo fuera necesario componer una base de datos completas que contuviera documentos, líneas y palabras, para así ejecutar los mismos experimentos que se llevaron a cabo en los textos impresos.

Una de las principales problemáticas en este sentido residía en conseguir suficientes muestras manuscritas como para tener una base de datos de una longitud semejante a la correspondiente a las imágenes de texto impreso. Esto fue posible gracias al equipo del *Indian Statistical Institute* que colaboraron con nosotros una vez más aportando imágenes de documentos manuscritos en bengalí y oriya. También se consiguió una cantidad relevante de documentos de esta naturaleza escritos en alfabeto latino por parte del equipo de la *División de Procesado Digital de la Señal*.

Además, diseñamos un nuevo esquema para organizar la base de datos y los ficheros con los nombres de las imágenes de entrenamiento y test producto de la división de la mencionada base de datos en entrenamiento y test.

En el esquema representado en la Figura 51 se ilustra cómo se disponen las muestras y ficheros de entrenamiento y test en nuestra base de datos.

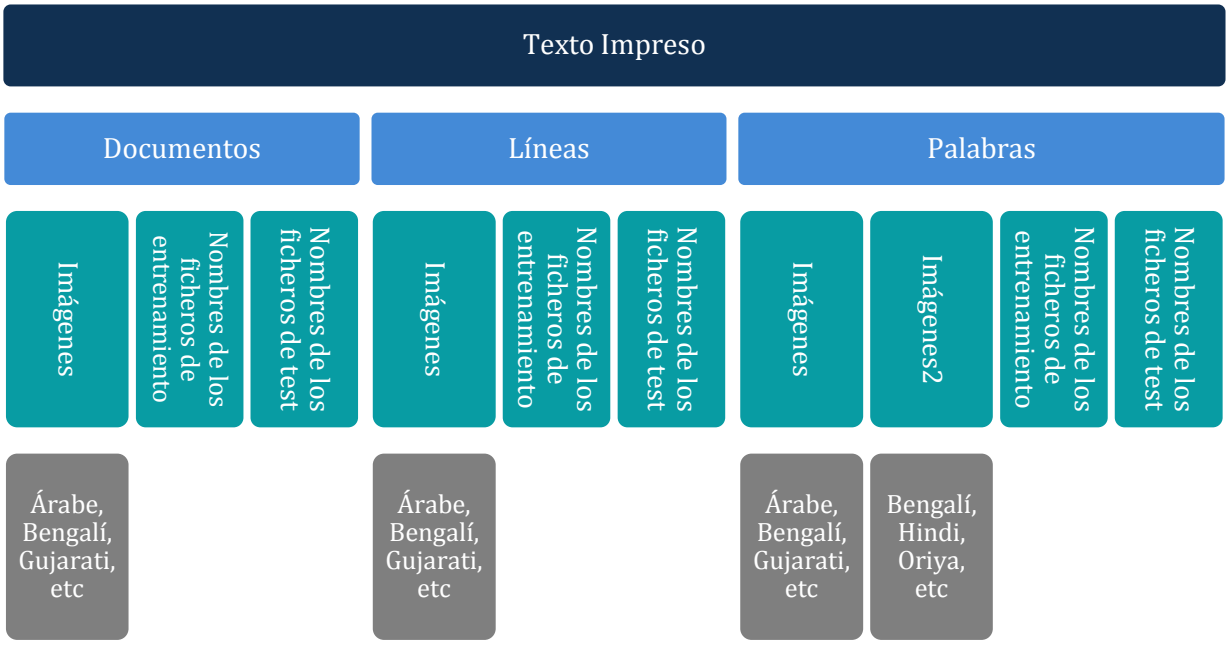


Figura 51-a. Distribución de la base de datos de texto impreso.

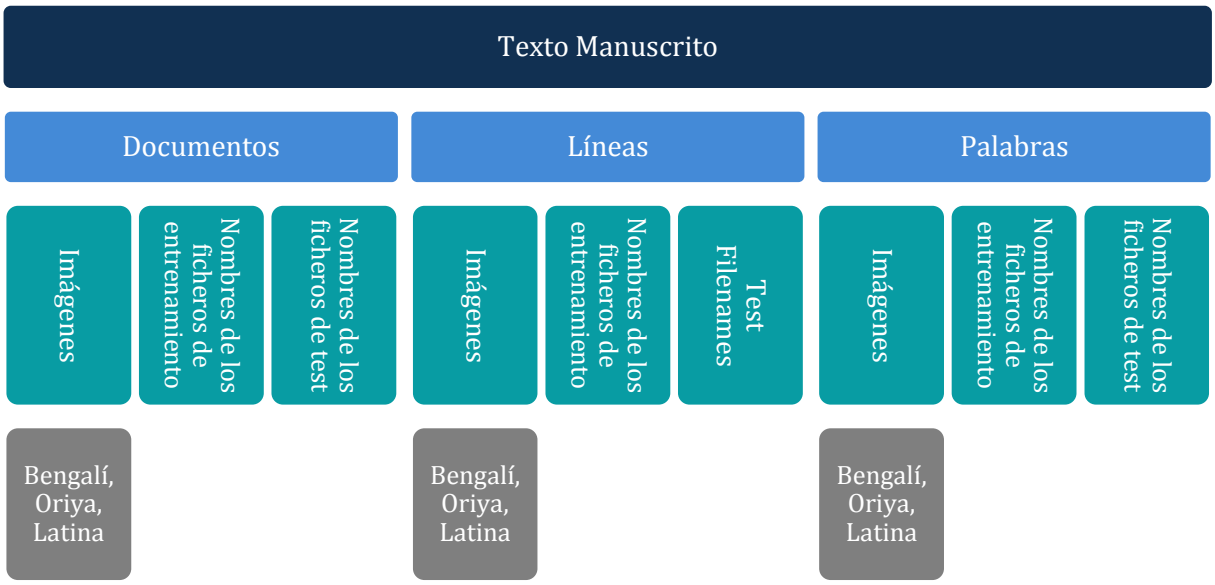


Figura 51-b. Distribución de la base de datos de texto manuscrito.

6.3. Elaboración de la base de datos

Una interesante característica de la base de datos manuscrita es que las muestras aportadas por los distintos equipos de investigación colaboradores fueron redactadas a mano en documentos físicos, para posteriormente ser escaneadas y organizadas en ficheros como se dispuso en el punto anterior.

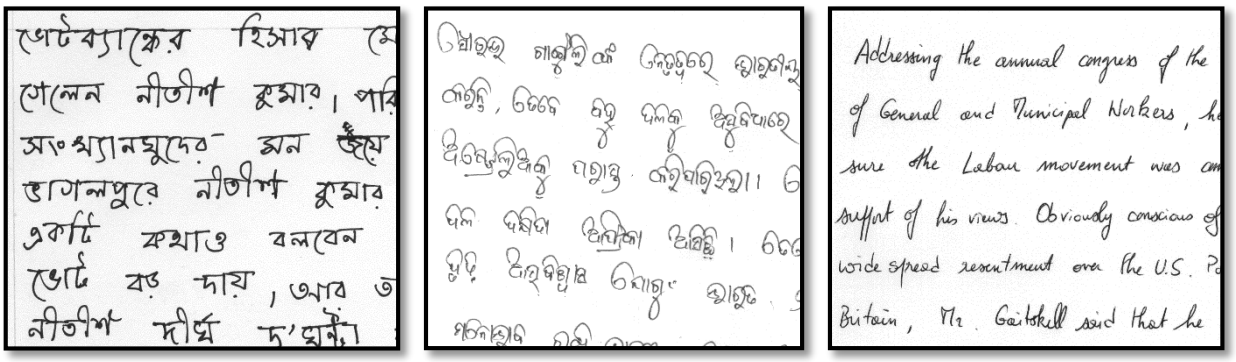


Figura 52. Fragmentos de documentos físicos digitalizados para formar parte de la base de datos de texto manuscrito.

El procedimiento a seguir en la elaboración de la base de datos es muy similar al desarrollado en el estudio de texto impreso. Así, una vez almacenados y apropiadamente nombrados los distintos documentos de la base de datos se llevó a cabo la segmentación de estas muestras en líneas y luego en palabras. Durante este proceso estos nombres asignados coincidían con la dinámica establecida en el trabajo anterior, es decir, empleando cuatro dígitos en representación del tipo de escritura de origen, tres más para el documento, seguidos de otros tres para indicar el número de la línea dentro del documento y los restantes tres para señalar el orden de la palabra dentro de dicha línea.

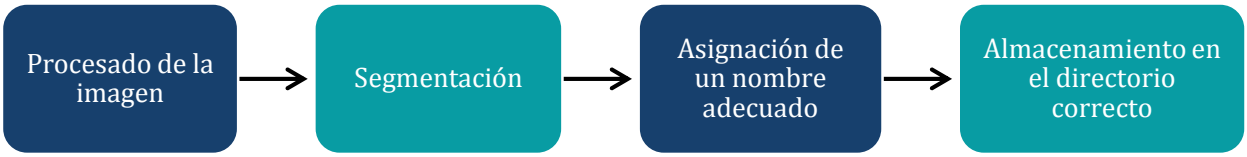


Figura 53. Diagrama de bloques del proceso de elaboración de la base de datos de texto manuscrito.

Como se indica en el diagrama de bloques anterior, para que toda la base de datos mantuviese un aspecto similar, las imágenes a niveles más bajo requerían de un procesado adicional. Sin embargo, a diferencia de la base de datos del estudio de texto impreso, las líneas obtenidas tras la segmentación de los documentos eran almacenadas con el texto en blanco y el fondo en negro, al igual que las palabras.

La base de datos una vez terminada se componía del siguiente número de imágenes clasificadas en la Tabla 22 según el nivel y el tipo de escritura:

Base de datos manuscrita			
Tipo de Escritura	*Nivel*		
	Documentos	Líneas	Palabras
Bengalí	67	515	3613
Oriya	50	1156	8410
Latina	90	764	5593
TOTAL	207	2435	17616

Tabla 22. Contenido de la base de datos manuscrita.

6.4. Separación de líneas y palabras

La segmentación de imágenes de texto manuscrito es la más complicada de la que hemos visto hasta ahora, especialmente en el momento de la división de documentos en líneas.

Una de las peculiaridades más notables de la escritura manuscrita es que las líneas de los documentos no tienen la misma inclinación, algunas incluso describen líneas curvas, y la mayoría no tienen la misma inclinación que las demás.

Un ejemplo ilustrativo de esta problemática se presenta en la Figura 54 que se localiza junto a estas líneas.

Se puede comprobar que en este documento es imposible dividir las líneas trazando una recta horizontal, por lo que el software desarrollado para segmentar las líneas de forma automática en el trabajo anterior no sería efectivo.

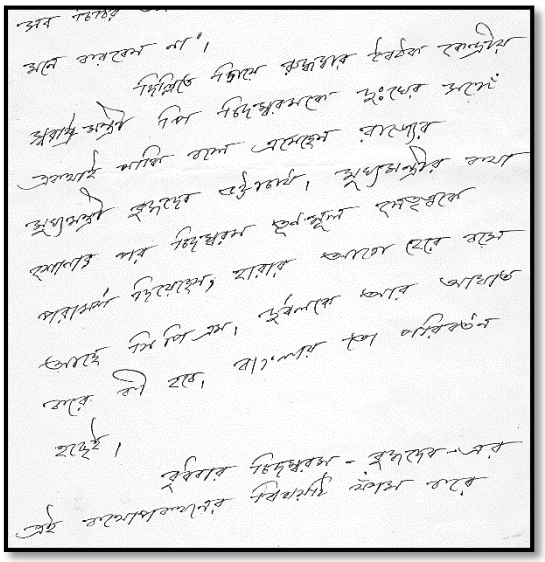


Figura 54. Fragmento de documento escrito a mano en bengalí.

Además, dado que hay trazos muy próximos de una línea a otra, la estrategia de establecer contornos usando la envolvente compleja tampoco sería apropiada en este caso.

Por estos motivos, la segmentación en líneas debía hacerse a mano necesariamente. Así que decidimos usar un editor de imágenes para trazar líneas a mano alzada que

separaran las distintas líneas de texto de los documentos una a una. Quedando las imágenes de los documentos de forma similar a la representada en la Figura 55.

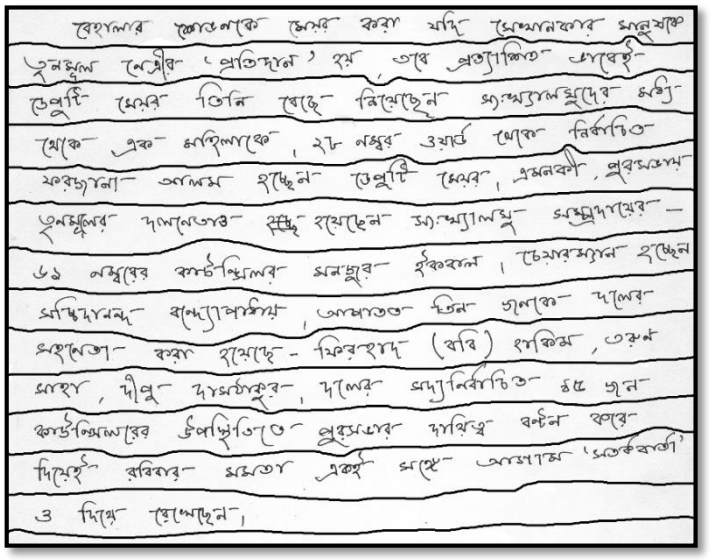


Figura 55. Fragmento de documento escrito a mano en bengalí segmentado trazando líneas a mano alzada.

Cuando acabamos de editar todos los documentos manuscritos, desarrollamos un nuevo software que detectaba esas marcas que habíamos dibujada y las usara como limitadores de cada línea, para así recortarlas y guardarlas con el nombre adecuado.

Por otro lado, para segmentar las palabras usamos el mismo método semiautomático que desarrollamos para segmentar las imágenes impresas con espacios bien definidos. Sin olvidar que en esta ocasión las líneas llegaban al sistema en blanco y negro, por lo que no era necesario invertirlas una vez fueran transformadas a formato lógico.

6.5. Pruebas realizadas con texto manuscrito y resultados

6.5.1. Procedimiento seguido

En primer lugar, se establecieron las distintas combinaciones donde se pretendía aplicar nuestro sistema. Como sólo contábamos con tres tipos de escritura, la combinación de los mismos dio lugar a los tres experimentos mostrados en la Tabla 23:

Experimentos		
Experimento	*Escritura 1*	*Escritura 2*
1	Latina	Bengalí
2	Oriya	Bengalí
3	Latina	Oriya

Tabla 23. Experimentos previstos para identificar la escritura en texto manuscrito.

En cuanto al procedimiento que se siguió en este caso, no fue necesario modificar ningún aspecto relevante en lo referido al funcionamiento de los sistemas involucrados en la identificación del tipo de escritura.

De forma que se escogían los tipos de escritura a enfrentar, el nivel de las mismas y el porcentaje de entrenamiento de ambas para dividir la base de datos en muestras de entrenamiento y test. Se escribían entonces los nombres de las imágenes en los ficheros de entrenamiento y test según correspondiera y tras dar por finalizado este paso, se extraían las características de ambos conjuntos mediante el cálculo de los LBP, OLBP, LDP y LDerivP de cada imagen manuscrita. Finalmente, los vectores de entrenamiento y test obtenidos se concatenaban de acuerdo con el experimento a llevar a cabo y éstos eran introducidos en el sistema de clasificación.

6.5.2. Estudio por documentos

Los documentos manuscritos fueron tratados de forma similar a los impresos. De tal forma que se mantuvo la división de los mismos en bloques de 128x128 píxeles solapados un 15% antes de ejecutar el cálculo de los patrones locales.

Asimismo, los resultados a este nivel producto de la aplicación de las técnicas propuestas en el presente proyecto se ilustran en la Tabla 24. Todos ellos fueron obtenidos aplicando un porcentaje de entrenamiento del 30%.

1ª Escritura	2ª Escritura	LBP	OLBP	LDP	LDerivP
Bengalí	Latina	66,891%	68,011%	66,979%	66,728%
Oriya	Latina	72,037%	69,768%	70,470%	71,223%
Bengalí	Oriya	65,839%	60,972%	63,499%	67,541%

Tabla 24. Resultados de los experimentos sobre texto manuscrito a nivel de documentos aplicando técnicas de análisis de texturas.

Por otro lado, al combinar dos de los parámetros de textura propuestos, las tasas de reconocimiento aumentan quedando como sigue:

1ª Escritura	2ª Escritura	LBP + OLBP	LBP + LDP	LBP + LDerivP
Bengalí	Latina	71,345%	69,948%	69,949%
Oriya	Latina	73,818%	73,865%	74,165%
Bengalí	Oriya	66,018%	66,942%	68,665%

Tabla 25-a. Resultados de los experimentos sobre texto manuscrito a nivel de documentos aplicando el LBP junto con otro de los patrones locales.

1ª Escritura	2ª Escritura	OLBP + LDP	OLBP + LDerivP	LDP + LDerivP
Bengalí	Latina	71,148%	71,181%	70,059%
Oriya	Latina	72,716%	73,763%	73,755%
Bengalí	Oriya	64,621%	67,454%	68,314%

Tabla 25-b. Resultados de los experimentos sobre texto manuscrito a nivel de documentos aplicando las combinaciones restantes usando dos patrones locales.

Análogamente, al combinar tres de los cuatro patrones locales se obtienen los siguientes resultados en términos de tasa de reconocimiento:

1ª Escritura	2ª Escritura	LBP + OLBP + LDP	LBP + OLBP + LDerivP	LBP + LDP + LDerivP	OLBP + LDP + LDerivP
Bengalí	Latina	72,480%	72,423%	71,428%	72,504%
Oriya	Latina	74,584%	75,121%	75,067%	74,769%
Bengalí	Oriya	67,420%	68,950%	69,425%	68,423%

Tabla 26. Resultados de los experimentos sobre texto manuscrito a nivel de documentos usando todas las combinaciones posibles con tres patrones locales.

Por último, si se combinan los efectos de todas las técnicas, el resultado es:

1ª Escritura	2ª Escritura	LBP + OLBP + LDP + LDerivP
Bengalí	Latina	73,118%
Oriya	Latina	75,494%
Bengalí	Oriya	69,571%

Tabla 27. Resultados de los experimentos sobre texto manuscrito a nivel de documentos combinando los efectos de todos los patrones

Los porcentajes obtenidos en estos experimentos son mucho más bajos que aquellos hallados en el estudio de texto impreso a nivel de documentos, al comparar los mismos tipos de escritura.

Esto puede suceder por muchos motivos. En primer lugar, no todas las muestras de la misma escritura son del mismo autor, ello implica que el texto en términos estructurales varíe de una muestra a otra, ya que la caligrafía no coincide. También ocurre que nuestro estudio está centrado principalmente en los trazos de las muestras de texto de las imágenes de análisis, si las líneas de un documento están inclinadas en diferente medida, los patrones locales cometen errores en sus cálculos, pues no se está analizando la dirección real de los mismos.

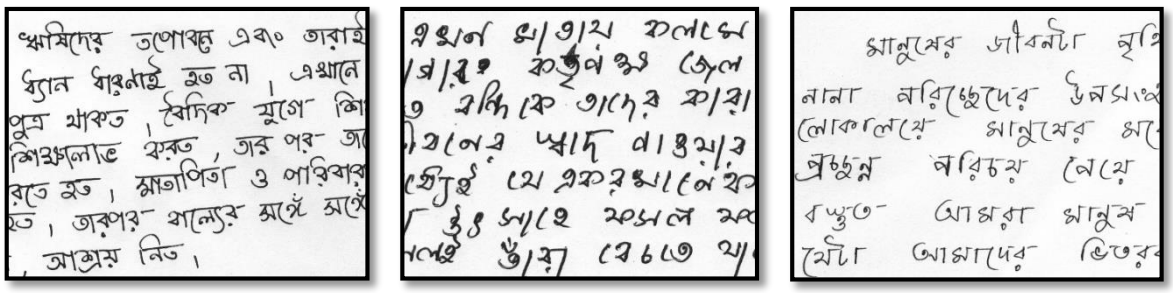


Figura 56. Fragmentos de documentos manuscritos redactados en bengalí con inclinaciones, anchos de trazo y caligrafías distintas.

Por otro lado, al contrario que en el estudio sobre texto impreso, se verifica que los métodos con los que se obtiene una mejor respuesta son aquellos donde el LBP y el LDerivP participan. Este hecho era de esperar debido a lo que comentábamos en el punto anterior, LDP y OLBP modelan las escrituras de acuerdo con la dirección de sus trazos, análisis que no es válido en este tipo de muestras.

6.5.3. Estudio por líneas

Con las líneas se siguió exactamente la misma dinámica que en los demás trabajos, es decir, se extraen las características de los fragmentos fruto de la división de las mismas en cuatro franjas horizontales.

En este sentido, los porcentajes de tasa de reconocimiento que se alcanzan a nivel de línea sobre texto manuscrito, empleando para ello un porcentaje de entrenamiento del 30% en ambos alfabetos enfrentados, se muestran en la Tabla 28.

1ª Escritura	2ª Escritura	LBP	OLBP	LDP	LDerivP
Bengalí	Latina	57,221%	57,339%	57,177%	57,339%
Oriya	Latina	71,790%	74,444%	74,074%	73,673%
Bengalí	Oriya	90,761%	80,049%	85,182%	83,414%

Tabla 28. Resultados de los experimentos sobre texto manuscrito a nivel de líneas aplicando técnicas de análisis de texturas.

Si mezclamos dos o más de los parámetros de textura propuestos, las tasas de reconocimiento varían. Asimismo, la Tabla 29-a y la Tabla 29-b muestran los resultados al combinar dos de los patrones, la Tabla 30 por su parte se rellena con los valores obtenidos aplicando tres de los parámetros y por último, la Tabla 31 presenta los porcentajes alcanzados trabajando con todas las técnicas.

1ª Escritura	2ª Escritura	LBP + OLBP	LBP + LDP	LBP + LDerivP
Bengalí	Latina	57,789%	58,132%	57,961%
Oriya	Latina	73,901%	72,654%	74,049%
Bengalí	Oriya	89,689%	91,801%	91,908%

Tabla 29-a. Resultados de los experimentos sobre texto manuscrito a nivel de líneas aplicando el LBP junto con otro de los patrones locales.

1ª Escritura	2ª Escritura	OLBP + LDP	OLBP + LDerivP	LDP + LDerivP
Bengalí	Latina	57,386%	57,505%	57,828%
Oriya	Latina	75,000%	74,877%	74,667%
Bengalí	Oriya	85,852%	84,995%	88,815%

Tabla 29-b. Resultados de los experimentos sobre texto manuscrito a nivel de líneas aplicando las combinaciones restantes usando dos patrones locales.

1ª Escritura	2ª Escritura	LBP + OLBP + LDP	LBP + OLBP + LDerivP	LBP + LDP + LDerivP	OLBP + LDP + LDerivP
Bengalí	Latina	56,477%	56,534%	56,487%	56,432%
Oriya	Latina	74,769%	74,815%	74,815%	75,093%
Bengalí	Oriya	91,226%	90,997%	92,849%	88,424%

Tabla 30. Resultados de los experimentos sobre texto manuscrito a nivel de líneas usando todas las combinaciones posibles con tres patrones locales.

1ª Escritura	2ª Escritura	LBP + OLBP + LDP + LDerivP
Bengalí	Latina	56,250%
Oriya	Latina	74,815%
Bengalí	Oriya	92,344%

Tabla 31. Resultados de los experimentos sobre texto manuscrito a nivel de líneas combinando los efectos de todos los patrones locales.

Podemos comprobar que existe una mejora en los experimentos que enfrentan las escrituras latina y bengalí con la oriya pero no sucede lo mismo al enfrentar las líneas redactadas en los alfabetos bengalí y latino.

La explicación de esto es que las escrituras bengalí y latina son las que más muestras tienen con esas variaciones de inclinación, lo cual provoca que en el paso de la división de la línea en franjas horizontales el análisis de características que se ejecutaba en las líneas impresas no se lleve a cabo apropiadamente. Mientras que antes analizábamos la distribución de los píxeles y la densidad de las distintas direcciones de los trazos de manera fiable, ahora no podemos estar seguros de que el análisis que se está realizando sirva para cumplir con los mismos objetivos.

Con objeto de comprobar si entrenando el sistema con más muestras de la escritura bengalí con respecto a las de la latina se podía obtener así un modelo más cerrado de la primera y en consecuencia, mejores resultados, se repitieron los experimentos anteriores entrenando con un 50% de las líneas en bengalí y un 30% de las líneas en alfabeto latino. Los resultados se resumen en las Tablas 32, 33-a, 33-b, 34 y 35 que se adjuntan en las líneas siguientes.

1ª Escritura	2ª Escritura	LBP	OLBP	LDP	LDerivP
Bengalí	Latina	68,767%	69,710%	69,131%	68,302%

Tabla 32. Resultados de los experimentos sobre texto manuscrito a nivel de líneas aplicando técnicas de análisis de texturas y dos porcentajes de entrenamiento distintos.

1ª Escritura	2ª Escritura	LBP + OLBP	LBP + LDP	LBP + LDerivP
Bengalí	Latina	73,674%	73,442%	73,143%

Tabla 33-a. Resultados de los experimentos sobre texto manuscrito a nivel de líneas aplicando el LBP junto con otro de los patrones locales y dos porcentajes de entrenamiento distintos.

1ª Escritura	2ª Escritura	OLBP + LDP	OLBP + LDerivP	LDP + LDerivP
Bengalí	Latina	73,700%	72,911%	73,006%

Tabla 33-b. Resultados de los experimentos sobre texto manuscrito a nivel de líneas aplicando las combinaciones restantes usando dos patrones locales y dos porcentajes de entrenamiento distintos.

1ª Escritura	2ª Escritura	LBP + OLBP + LDP	LBP + OLBP + LDerivP	LBP + LDP + LDerivP	OLBP + LDP + LDerivP
Bengalí	Latina	74,138%	73,254%	73,475%	73,522%

Tabla 34. Resultados de los experimentos sobre texto manuscrito a nivel de líneas usando todas las combinaciones posibles con tres patrones locales y dos porcentajes de entrenamiento distintos.

1ª Escritura	2ª Escritura	LBP + OLBP + LDP + LDerivP
Bengalí	Latina	74,039%

Tabla 35. Resultados de los experimentos sobre texto manuscrito a nivel de líneas combinando los efectos de todos los patrones locales y usando dos porcentajes de entrenamiento distintos.

Se produce una gran mejoría simplemente aplicando porcentajes de entrenamiento distintos. Lo cual nos lleva a pensar que el hecho de que el sistema caracterice un mayor número de líneas en bengalí donde la variedad de caligrafías, inclinaciones y estilos es tan desigual hace que se modelen mejor las características generales que todos los tipos distintos de muestras de texto tienen en común.

6.5.4. Estudio por palabras

Una vez más, decidimos no modificar la manera de trabajar y seguimos la misma estrategia que en el estudio de texto impreso, tal que las palabras eran binarizadas y divididas en cuatro partes antes de ser sometidas al cálculo de los LBP, OLBP, LDP y LDerivP para posteriormente ejecutar el proceso de clasificación. En este caso además se estableció que el porcentaje de entrenamiento del sistema sería del 30% en ambos tipos de escritura.

En esta línea, los resultados expresados en términos de tasa de reconocimiento que se alcanzan usando palabras manuscritas se muestran en la Tabla 36.

1ª Escritura	2ª Escritura	LBP	OLBP	LDP	LDerivP
Bengalí	Latina	63,100%	63,352%	62,995%	63,264%
Oriya	Latina	73,382%	74,524%	74,125%	74,123%
Bengalí	Oriya	74,315%	73,409%	74,488%	74,616%

Tabla 36. Resultados de los experimentos sobre texto manuscrito a nivel de palabras aplicando técnicas de análisis de texturas.

De manera análoga, estos porcentajes aumentan en la medida en que combinamos los efectos de las distintas técnicas de análisis. Dicho esto, si se trabaja con dos de los cuatro patrones locales a la vez se obtiene:

1ª Escritura	2ª Escritura	LBP + OLBP	LBP + LDP	LBP + LDerivP
Bengalí	Latina	64,014%	64,050%	64,230%
Oriya	Latina	74,693%	74,246%	74,112%
Bengalí	Oriya	76,693%	76,572%	76,599%

Tabla 37-a. Resultados de los experimentos sobre texto manuscrito a nivel de palabras aplicando el LBP junto con otro de los patrones locales.

1ª Escritura	2ª Escritura	OLBP + LDP	OLBP + LDerivP	LDP + LDerivP
Bengalí	Latina	63,674%	63,891%	63,963%
Oriya	Latina	74,909%	74,922%	74,549%
Bengalí	Oriya	76,487%	76,776%	76,947%

Tabla 37-b. Resultados de los experimentos sobre texto manuscrito a nivel de palabras aplicando las combinaciones restantes usando dos patrones locales.

Asimismo, las tasas de reconocimiento conseguidas mezclando tres patrones locales y dejando uno de ellos fuera del cálculo son las siguientes:

1ª Escritura	2ª Escritura	LBP + OLBP + LDP	LBP + OLBP + LDerivP	LBP + LDP + LDerivP	OLBP + LDP + LDerivP
Bengalí	Latina	63,333%	63,509%	63,630%	63,266%
Oriya	Latina	75,428%	75,478%	75,086%	75,468%
Bengalí	Oriya	78,223%	78,178%	78,029%	78,284%

Tabla 38. Resultados de los experimentos sobre texto manuscrito a nivel de líneas usando todas las combinaciones posibles con tres patrones locales.

Para acabar, si se efectúa la combinación de todos los patrones locales disponibles los resultados de la identificación de los distintos tipos de escritura quedan:

1ª Escritura	2ª Escritura	LBP + OLBP + LDP + LDerivP
Bengalí	Latina	63,233%
Oriya	Latina	75,494%
Bengalí	Oriya	74,815%

Tabla 39. Resultados de los experimentos sobre texto manuscrito a nivel de palabras combinando los efectos de todos los patrones locales.

Mediante la observación de los datos contenidos en las tablas, se verifica que a medida que se disminuye el nivel la tasa de acierto es mayor. Lo cual corrobora nuestra teoría de

que la mala inclinación de las líneas afecta al cálculo de los parámetros locales, puesto que en las palabras prácticamente no se aprecian los efectos de la misma.

No obstante, los resultados siguen estando muy lejos de los que se lograron en el caso del análisis de texto impreso y ello señala que las técnicas de análisis de textura no son del todo adecuadas para identificar el tipo de escritura en texto manuscrito.

Capítulo 7. Conclusiones y líneas futuras

7.1. Introducción

En este capítulo se exponen las principales conclusiones extraídas de la realización de este Proyecto de Final de Carrera, así como las posibles mejoras y las nuevas líneas de trabajo futuras que puedan efectuarse en el futuro.

Para ello, en primer lugar, se evaluará el grado en que se han conseguido los objetivos planteados al principio de esta memoria y se comentarán además los aspectos más relevantes que se desprenden de los resultados obtenidos. Seguidamente, se realizará un resumen del trabajo realizado y se enumerarán las conclusiones a las que hemos llegado. Finalmente, el último bloque del capítulo se dedicará a enumerar las aportaciones que ha generado el proyecto, proponer algunas mejoras y enunciar las posibles líneas futuras de investigación en relación con el mismo.

7.2. Cumplimiento de objetivos y resumen del trabajo realizado

Haciendo un análisis de los objetivos que fueron presentados al principio de esta memoria podemos afirmar que se han cumplido todos ellos de manera satisfactoria. En ese sentido, no sólo se ha alcanzado su cumplimiento a nivel conceptual con la demostración de la viabilidad de nuestro planteamiento sino además, se ha hecho todo con el número de recursos adecuados en términos de materiales, tiempo, etc. Como consecuencia también podemos valorar positivamente el planteamiento, la estructura y la metodología que se han adoptado en este proyecto.

Asimismo, este Proyecto de Final de Carrera surge como respuesta a la problemática existente en la actualidad a la hora de digitalizar documentos escritos con varios tipos de escritura. Éste se ha propuesto con el objeto de innovar en la identificación de los tipos de escritura en imágenes de texto, usando para ello técnicas de gran popularidad en el análisis de texturas: los patrones locales Local Binary Pattern, Orientation of Local Binary Pattern, Local Directional Pattern y Local Derivative Pattern. Asimismo, para corroborar la validez de estos métodos aplicados al campo de análisis de texto, se llevaron a cabo tres estudios distintos cuyas características más relevantes se recuerdan a continuación.

Por un lado, en el estudio de laboratorio que se desarrolló inicialmente se diseñó una base de datos compuesta de 20 imágenes de documentos en diez tipos de escritura distintos usando como herramienta principal el traductor de Google. Además, dicha base de datos se

completó con algunas imágenes aportadas por el equipo colaborador del Indian statistical Institute y que fueron obtenidas a partir de digitalizar una serie de documentos físicos y que se usaron únicamente en el proceso de test.

Durante esta primera etapa del proyecto se desarrollaron una serie de programas para poder realizar la identificación del tipo de alfabeto a nivel de línea. Concretamente, el sistema inicial se componía de dos partes diferenciadas: la extracción de características y la clasificación. Por un lado los documentos eran segmentados en líneas y luego éstas se dividían a su vez en cuatro franjas horizontales, que eran los elementos sobre los cuales se calculaban los patrones locales a fin de modelar cada tipo de escritura. Posteriormente, se hallaba el histograma de cada bloque y se concatenaban los resultados componiendo poco a poco el vector de parámetros que contenía las características de cada escritura. Finalmente, se pasaba a llevar a cabo la fase de clasificación, en la cual se aplicaba un clasificador LS-SVM que enfrentaba todos los tipos de escritura incluidos en la base de datos.

Otro detalle de interés es que se realizan dos experimentos, en primer lugar se empleó únicamente la primera base de datos tanto para entrenar como para probar el sistema, y posteriormente, se entrenó con el primer conjunto de imágenes y se realizó el test con la segunda base de datos, que nos habían enviado desde la India. En general, los resultados obtenidos fueron satisfactorios y demostraron que los métodos de análisis de texturas son útiles para reconocer el tipo de escritura a nivel de línea con más de un 90 por ciento de acierto. Para corroborar este hecho, se decidió realizar un trabajo más complejo y más amplio.

Por consiguiente, el siguiente paso seguido en el proyecto consistió en llevar a cabo un estudio con un nivel de realismo mayor, conformando para ello una base datos cuyos documentos fueran obtenidos a partir de ejemplares físicos y que contarán con imperfecciones en el papel o en la tinta, con distintas fuentes y estilos, puesto que esos son los casos donde nuestro sistema podría tener una mayor utilidad fuera del laboratorio.

Así, se escanearon una serie de documentos como periódicos, libros, revistas, etc. y éstos fueron segmentados en líneas y en palabras componiendo así una base de datos completa a tres niveles. En cuanto a la extracción de parámetros, en esta ocasión se llevaron a cabo una serie de pasos adicionales a los efectuados en el primer estudio. El sistema completo dividía en primer lugar la base de datos en muestras de entrenamiento y muestras de test, posteriormente se procedía al análisis de las imágenes de ambos grupos mediante el cálculo de patrones locales aprovechando la programación del primer estudio. La única diferencia a considerar, es que en cada nivel las imágenes se dividían en fragmentos de

forma diferente y dado que su aspecto no era el mismo, se requería de un procesado especial en cada caso. Finalmente se procedía a introducir los valores obtenidos en el clasificador LS-SVM para obtener las tasas de reconocimiento.

En términos de experimentación, en este segundo planteamiento se realizó un mayor número de pruebas y se comprobó la fiabilidad del sistema a medida que se disminuía el porcentaje de información a extraer a partir de los trazos, ya que los documentos, las líneas y las palabras no tienen la misma cantidad de información. De tal forma que, salvo algunas excepciones, los resultados volvieron a ser satisfactorios, corroborando así la idoneidad de los patrones locales para el reconocimiento de alfabetos dentro de una imagen de texto impreso.

El último trabajo que se ejecutó durante este proyecto se realizó centrando la atención del mismo sobre imágenes de texto manuscrito, que son uno de los frentes abiertos en el campo del análisis de imágenes de texto. En este caso también se tuvo que crear una base de datos desde cero, no obstante, desde la India nos facilitaron un gran número de documentos redactados en dos tipos de escritura diferentes para poder trabajar con ellos y hacer las comprobaciones pertinentes, los cuales fueron separados en líneas y posteriormente en palabras. No tuvimos que modificar el planteamiento seguido en el estudio de texto impreso pero los resultados no fueron tan buenos con en el resto de los casos dada la naturaleza de las muestras de nuestra base de datos, donde las diferencias caligráficas y de inclinación no hicieron posible un mayor porcentaje de éxito en el reconocimiento del tipo de escritura.

7.3. Conclusiones

Una de las principales conclusiones que se pueden extraer de este proyecto es que aún existe margen para innovar en la identificación del tipo de escritura y que es necesario seguir investigando en este campo para obtener aún mejores resultados que abran la puerta a nuevas aplicaciones.

Por otra parte, se ha hecho notar la importancia de contar con una buena base de datos en procesos de reconocimiento. En nuestro caso, una de las tareas más complejas y tediosas fue la creación de esta pieza fundamental de nuestros estudios. Y es que todos los pasos que han de llevarse a cabo para obtener un conjunto adecuado de muestras conllevan un grado de complejidad considerable, hecho que se ha contemplado a lo largo de toda esta memoria. Además, una mala elección de los elementos de análisis puede conllevar la

obtención de resultados erróneos o negativos, invalidando procedimientos que tal vez hubieran funcionado correctamente sobre un entorno distinto.

Asimismo, y centrándonos en la propuesta principal de este proyecto que es el uso de técnicas de análisis de texturas sobre imágenes de texto, podemos afirmar que los LBP, OLBP, LDP y LDerivP son adecuados para reconocer escrituras a cualquier nivel, ya sea documentos, líneas o palabras en imágenes de texto impreso. En la mayor parte de los casos contemplados en la experimentación, se llegó a alcanzar e incluso superar una tasa de reconocimiento del 90%. Por lo que, en comparación con los sistemas de uso más común para la resolución de la problemática en que no hemos centrado, nuestro sistema logra un alto porcentaje de éxito con un coste computacional más bajo y un planteamiento más sencillo.

El texto manuscrito por otro lado, sigue siendo un problema no resuelto siquiera por estos parámetros que han demostrado ser apropiados sobre texto impreso. No obstante, los malos resultados que se logran en este estudio podrían achacarse a la base de datos, por lo que debería servir de motivación para el desarrollo de nuevas técnicas y estrategias a partir de las que hemos propuesto en este trabajo.

En relación con lo anterior, una última conclusión que puede ser extraída tras la última fase de experimentación es que las muestras no deben estar inclinadas, para no calcular valores erróneos al analizar las direcciones de los trazos, que son el pilar principal sobre el que se sustenta el reconocimiento de alfabetos usando patrones locales.

7.4. Aportaciones, líneas futuras y posibles mejoras

En cuanto a las aportaciones de nuestro proyecto, enumeraremos a continuación las más destacables:

- Una gran ventaja de este proyecto es que realmente sirve como línea de partida para el desarrollo de nuevos métodos que complementen al que hemos propuesto. Y es que hemos demostrado que un texto, en cierta manera, puede equipararse a una textura y puede por tanto, analizarse de acuerdo con los métodos desarrollados en este campo.
- En esta línea, hemos sumado también un nuevo campo de aplicación a los patrones locales, los cuales se destinaban normalmente a trabajados donde se involucraran texturas.

- Por otro lado, se ha creado una base de datos completa y de una extensión considerable que puede servir de objeto de experimentación en otros trabajos que no sólo se dediquen a la identificación del tipo de escritura, sino que se desarrollen en el campo del análisis de imágenes de texto en cualquiera de sus vertientes.
- Asimismo, en este proyecto aporta un elevado número de programas con diferentes fines. Entre los cuales destacan la extracción de características de una imagen de texto a nivel de líneas, documentos y palabras en un único software; la división de una base de datos para experimentos que involucren dos clases, a partir de dos porcentajes distintos de entrenamiento y dando como resultado una serie de ficheros de texto que contienen los nombres de los archivos a analizar en cada caso; y la segmentación de documentos en líneas y de líneas en palabras siguiendo una metodología semi-automática donde el usuario participa en el proceso de separación de forma activa.

Por otro lado, existen múltiples opciones que podrían implementarse para mejorar el rendimiento de nuestro sistema pero que no se han llevado a cabo durante el desarrollo de este proyecto. Algunas de ellas se listan a continuación:

- Centrándonos en la identificación de escritura en texto de escritura, podría desarrollarse un programa que corrija la inclinación de las líneas para así minimizar lo máximo posible el efecto negativo que ésta tiene sobre los resultados del sistema.
- Por otro lado, se podría incorporar a nuestro sistema un conjunto de filtros de Gabor, que son los más ampliamente empleados en nuestro campo de trabajo y que tienen una muy buena respuesta al ser aplicados sobre texto. Tal vez con esta medida consigamos mejorar un poco más la tasa de reconocimiento lograda a costa de aumentar un poco el coste computacional de nuestro sistema.
- También podríamos incorporar más escrituras como la coreana o la griega para así seguir generalizando nuestro sistema.
- Por otro lado, sería adecuado modificar la base de datos tal que todas las imágenes tengan el mismo aspecto en todos los niveles, por ejemplo, transformándolas a blanco y negro.

Otro aspecto que cabe mencionar es que inicialmente, nuestro sistema fue diseñado para servir como etapa previa de un OCR para identificar el tipo de escritura antes de aplicarlo para digitalizar un documento cualquiera. En consecuencia, una opción que podríamos explotar en el futuro consiste en el diseño de un sistema combinado que incluya nuestro sistema y un OCR, de tal forma que se pueda digitalizar un documento pasando por

todas las etapas necesarias para que la tarea se lleve a cabo con éxito. Sería necesario crear una interfaz que permita también el manejo de un escáner para procesar el documento físico pasado a imagen, a nivel de palabra, aplicando el OCR a ese nivel y guardando el resultado en un formato digital y editable.

En esta línea, también sería interesante hacer nuestro sistema portable y usarlo en un teléfono móvil o una tablet, combinándolo con un OCR y la cámara que incorporan estos dispositivos. Así, si sacáramos una fotografía con el teléfono, podríamos detectar el texto, determinar a qué escritura pertenece, digitalizar el texto analizando palabra a palabra y luego pasar los resultados a un traductor. Esta aplicación sería útil para aquellas personas con discapacidad auditiva que deciden hacer un viaje al extranjero sin conocer el idioma, tener a mano este sistema podría facilitar mucho esta experiencia.

En conclusión, la identificación del tipo de escritura es una tarea que puede dar lugar a muchas y distintas aplicaciones si se consiguen buenos resultados y se asocia con más programas.

Referencias bibliográficas

[1] Beatriz Lampreabe Martínez "Metodología de Digitalización de Documentos". Eusko Jaurlaritzaren Informatika Elkartea - Sociedad Informática del Gobierno Vasco. 2008. Documentación de referencia.

http://www.zuzenean.euskadi.eus/s68-contay/eu/contenidos/informacion/modelo_gestion_documental/eu_modgesdo/adjuntos/Metodolog%C3%ADa%20de%20Digitalizaci%C3%B3n%20de%20Documentos.pdf

[2] Digitalización. Anobium – Innovación en sistemas de información y documentación S. L.. Página web.

http://www.anobium.es/ventajas-de-la-digitalizacion-de-documentos

[3] Celso Gonzáles Cam, "La Importancia de la Digitalización de Archivos para la Biblioteca", Convención Nacional de Centros Binacionales (Perú), Octubre de 2007.

http://eprints.rclis.org/10647/1/La_importancia_de_la_digitalizaci%C3%B3n_de_archivos_para_la_bi%E2%80%A6.pdf

[4] L. O'Gorman and R. Kasturi, "Documents Image Analysis", IEEE Computer Society Executive Briefings, ISBN 0-8186-7802-X, 1997.

[5] R. Kasturi et al., "Document image analysis: A primer", Sadhana, vol. 27, part 1, pp. 3–22, Feb. 2002.

[6] Reconocimiento óptico de caracteres (2014). Wikipedia. Página web.

http://en.wikipedia.org/wiki/Optical_character_recognition

[7] "What is OCR and OCR technology", ABBYY. Página web.

http://www.abbyy.com/finereader/about-ocr/what-is-ocr/

[8] "Alfabetos de ayer y hoy". PROEL – Promotora Española de Lingüística, España. Página web.

http://www.proel.org/index.php?pagina=alfabetos

[9] Irene Thompson, "Writing Systems". AWL – About World Languages. Página web.

http://aboutworldlanguages.com/writing-systems

[10] "South Asian Writing Systems". Ancient Scripts. Página web.

http://www.ancientscripts.com/sa_ws.html

[11] D. Ghosh, T. Dube and A. P. Shivaprasad, "Script Recognition - A Review", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.32, no.12, pp.2142-2161, Dec. 2010.

[12] U. Pal, "Automatic Script Identification: A Survey", J. Vivek, vol. 16, no. 3, pp. 26-35, 2006.

[13] C. Ronse and P.A. Devijver, "Connected Components in Binary Images: The Detection Problem". John Wiley & Sons, 1984.

[14] A.L. Spitz, "Multilingual Document Recognition", *Proceedings of the International Conference on Electronic Publishing, Document Manipulation and Typography*, pp. 193-206, Sept. 1990.

[15] A.L. Spitz and M. Ozaki, "Palace: A Multilingual Document Recognition System", *Proceedings of the International Association for Pattern Recognition Workshop Document Analysis Systems*, pp. 16-37, Oct. 1994.

[16] A. L. Spitz, "Determination of the Script and Language Content of Document Images", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19,no. 3, March 1997.

[17] D. S. Lee et al., "Language Identification in Complex, Unoriented, and Degraded Document Images", *Proceedings of the International Association for Pattern Recognition Workshop Document Analysis Systems*, pp. 76-98, Oct. 1996.

[18] S. Chaudhury and R. Sheth, "Trainable Script Identification Strategies for Indian Languages", *Proceedings of the International Conference for Document Analysis and Recognition*, pp. 657-660, Sept. 1999.

[19] U. Pal and B. B. Chaudhuri, "Identification of different script lines from multi-script documents", *Image and Vision Computing*, vol. 20, no.13-14, pp. 945-954, 2002.

[20] S. W. Lee and J. S. Kim, "Multi-Lingual, Multi-Font, Multi-Size Large-Set Character Recognition Using Self-Organizing Neural Network", *Proceedings of the International Conference for Document Analysis and Recognition*, vol. 1, pp. 28-33, Aug. 1995.

[21] J. Hochberg et al., "Page Segmentation Using Script Identification Vectors: A First Look", *Proceedings of the Symposium on Document Image Understanding Technology*, pp. 258-264, Apr.-May 1997.

[22] T. N. Tan, "Rotation invariant texture features and their use in automatic script identification", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no.7, pp. 751-756, July 1998.

[23] W. M. Pan, C. Y. Suen, T. D. Bui, "Script identification using steerable Gabor filters", *Eighth International Conference on Document Analysis and Recognition*, vol. 2, pp. 883-887, 29 Aug.-1 Sept. 2005.

[24] A. Busch et al., "Texture for Script Identification", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 11, pp. 1720-1732, Nov. 2005.

[25] A. Busch, "Multi-Font Script Identification Using Texture-Based Features", *Proceedings of the International Conference for Image Analysis and Recognition*, pp. 844-852, Sept. 2006.

[26] D. Dhanya et al., "Script Identification in Printed Bilingual Documents", *Sadhana*, vol. 27, no. 1, pp. 73-82, Feb. 2002.

[27] T. Mäenpää, M. Pietikäinen, "Texture Analysis with local binary Patterns", in C.H. Chen, P.S.P. Wang (eds.): Handbook of Pattern Recognition and Computer Vision, 3rd edn. World Scientific, pp. 197-216, 2005.

[28] S. Brahnam, L. C. Jain, L. Nanni, A. Lumini Editors (2014). " Local Binary Patterns: New Variants and Applications". *Editorial Spring*. ISSN 1860-9503.

[29] W. Bu, X. Wub, E. Gao, "Hand Vein Recognition Based on Orientation of LBP", Proceedings of SPIE, vol. 8371, pp. 83711Y-1 a 83711Y-12, Biometric Technology for Human Identification, 2012.

[30] T. Jabid, M. H. Kabir, O. Chae, "Local Directional Pattern (LDP) – A Robust Image Descriptor for Object Recognition", *7th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp.482,487, Aug. 29th 2010 - Sept. 1st 2010.

[31] B. Zhang, Y. Gao, S. Zhao, J. Liu, "Local Derivative Pattern Versus Local Binary Pattern: Face Recognition With High-Order Local Pattern Descriptor", *IEEE Transaction on Image Processing*, vol.19, no.2, pp. 533-544, Feb. 2010.

[32] U. S. N. Raju, A. S. Kumar, B. Mahesh, B. E. Reddy, "Texture Classification With High Order Local Pattern Descriptor: Local Derivative Pattern", *Global Journal of Computer Science and Technology*, vol 10, Issue 8, pp. 72-76, September 2010.

[33] N. Cristianini, J.S. Taylor, "Support Vector Machines and other Kernel-based Learning Methods". Cambridge University Press, 2000.

[34] V. Jakkula**, "**Tutorial on Support Vector Machine (SVM)", School of EECS, Washington State University. Tutorial.

http://www.ccs.neu.edu/course/cs5100f11/resources/jakkula.pdf

[35] N. Cristianini, J.S. Taylor, "Support Vector Machines and other Kernel-based Learning Methods". Cambridge University Press, 2000.

[36] "Introducción al Diseño de Experimentos para el Reconocimiento de Patrones". Capítulo 7: Máquinas de Vectores Soporte. Curso de doctorado impartido por Dr. Quiliano Isaac Moro Dra. Aránzazu Simón Hurtado. Enero 2006. Material de clase.

http://www.infor.uva.es/~isaac/doctorado/Cap07_SVM.pdf

[37] J-P. Vert et al., "A primer on kernel methods". Centre for Computational Biology. 2006.

http://cbio.ensmp.fr/~jvert/publi/04kmcbbook/kernelprimer.pdf

[38] J. A. K. Suykens, T. V. Gestel, J. D. Brabanter, B. D. Moor, J. Vandewalle, Least Squares Support Vector Machines, World Scientific Publishing Co. Pvt. Ltd, Singapore, 2002.

[39] J. Ye and T. Xiong, "SVM versus Least Squares SVM," in the 11th International Conference on Artificial Intelligence and Statistics (AISTATS), 2007, pp. 640-647.

[40] "Derechos de vidaso para 2014 a percibir por el COIT en el visado obligatorio de trabajos profesionales". COIT – Colegio Oficial de Ingenieros de Telecomunicación, España. Material de referencia.

https://www.coit.es/web/ejercicio_profesional/svc/dv/2014_peticion_expresa.pdf

[41] R. Berwick, **"**An Idiot's guide to Support vector machines (SVMs)". Material de referencia.

http://www.svms.org/tutorials/Berwick2003.pdf

[42] A. McAndrew, "Introduction to Digital Image Processing with Matlab", Course Technology, 1st Edition, 2004.

[43] B. B.Chaudhuri and U. Pal, "Skew angle detection of digitized Indian Script documents", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.19, pp.182-186, 1997.

[44] C. Solomon, T. Breckon, "*Fundamentals of Digital Image Processing: An Approach with Examples in Matlab"*. John Wiley & Sons, Ltd., 2010.

[45] D. Ghosh, T. Dube and A. P. Shivaprasad, "Script Recognition - A Review", IEEE Trans.on PAMI, vol.32, no.12, pp.2142-2161, Dec. 2010.

[46] E. Laorden, "Descripción, comparación y ejemplos de uso de las funciones de la toolbox de procesado digital de la imagen de Matlab". Proyecto Final de Carrera de la Escuela de Ingeniería Técnica de Telecomunicación, Universidad Politécnica de Madrid.

http://oa.upm.es/14016/2/PFC_EDUARDO_LAORDEN_FITER_B.pdf

[47] Francisco José Ribadas, "TEMA 6. SVM – Support Vector Machines (Máquinas de Vectores Soporte)", Asignatura Modelos de Razonamiento y Aprendizaje, Universidad de Vigo. Material de clase.

http://ccia.ei.uvigo.es/docencia/MRA/1213/transparencias/Tema6.pdf

[48] G. S. Peake, T. N. Tan, "Script and language identification from document images", *Workshop on Document Image Analysis (DIA '97)* pp. 10-17, 20 Jun. 1997.

[49] J. A. K. Suykens, K. De Brabanter, P. Karsmakers, F. Ojeda, C. Alzate, K. Pelckmans, "LS-SVMlab Toolbox User's Guide version 1.8". Página web.

http://www.esat.kuleuven.be/sista/lssvmlab/

[50] J. Cheng, X. Ping, G. Zhou, Y. Yang, "Script Identification of Document Image Analysis", *1ST International Conference on Innovative Computing, Information and Control* (ICICIC '06), vol.3, pp. 178-181, Aug. 30th 2006-Sept. 1st 2006.

[51] J. Hochberg, P. Kelly, T Thomas and L. Kerns, "Automatic script identification from document images using cluster-based templates", IEEE PAMI, vol. 19, pp. 176-181, 1997.

[52] J. Pan, Y. Tang. "A rotation-robust script identification based on BEMD and LBP", In *International Conference on Wavelet Analysis and Pattern Recognition (ICWAPR)*, pp. 165 - 170, 10 - 13 July 2011.

[53] K. Roy, S. K. Das, S.M. Obaidullah, "Script Identification from Handwritten Document", *Third National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG),* pp. 66-69, 15-17 Dec. 2011.

[54] M. A. Ferrer, A. Morales, U. Pal, "LBP based line-wise script identification", In Proc. 12th ICDAR, pp. 369-373, 2013.

[55] Mingji Piao, Rongyi Cui, "An Approach to Script Identification in Multi-language Text Image", *6th International Conference on Intelligent Networks and Intelligent Systems (ICINIS)*, pp.248-251, 1-3 Nov. 2013.

[56] M. Pietikaainen, T. Ojala, Z. Xu, "Rotation invariant texture classification using feature distributions", *Pattern Recognition*, vol. 33, no. 1, pp. 43-52, 2000.

[57] P. K. Aithal, G. Rajesh, D. U. Acharya, M. Krishnamoorthi, N. V. Subbareddy. "Script identification for a Tri-lingual document". *Communications in Computer and Information Science*. 2011.

[58] P. Zhang, J. Peng, "SVM vs. Regularized Least Squares classification. 17th International Conference on Pattern Recognition, pages 176–179, 2004.

[59] R. C. Gonzalez, R. E. Woods, L. E. Steven (2009). *Digital Image Processing Using MATLAB*.

[60] S. Chanda, U. Pal, K. Franke, F. Kimura (2010). Script identification - A Han & Roman script perspective. In *Proceedings - International Conference on Pattern Recognition*.

[61] S. Ghosh, B. B. Chaudhuri. "Composite script identification and orientation detection for Indian text images", 11th International Conference on Document Analysis and Recognition, pp. 294 - 298, 2011.

[62] U. Pal and B. B. Chaudhuri, "Automatic separation of words in Indian multi-lingual multi-script documents", 4th International Conference on Document Analysis and Recognition, pp. 576-579, 1997.

[63] U. Pal and B. B. Chaudhuri, "Script line separation from Indian multi-script documents", 5th International Conference on Document Analysis and Recognition, pp 406-409, 1999.

[64] U. Pal and B. B. Chaudhuri, "Identification of different script lines from multi-script documents", Image and Vision computing, vol. 20, no.13-14, pp. 945-954, 2002.

[65] U. Pal, A. Belaïd and Ch. Choisy, "Touching numeral segmentation using water reservoir concept", Pattern Recognition Letters, vol.24, pp. 261-272, 2003.

Nota: Todos los documentos electrónicos han sido visualizados por última vez el 29 de Mayo de 2014.

Planos y programas

8.1. Introducción.

En este capítulo se explica el funcionamiento general de cada uno de los algoritmos que se han desarrollado para la realización de este proyecto.

Sólo están incluidos los algoritmos o *scripts* definitivos, no los casos de algoritmos provisionales o desarrollados a modo de prueba. Para una mejor comprensión, se han dividido todos ellos según la fase del flujo de trabajo del proyecto a la que pertenecen. Asimismo, y de forma general, cada uno de los programas ha sido implementado de forma individual.

Es de destacar que no se ha generado plano alguno en la realización de este proyecto.

8.2. Programas implementados.

8.2.1. Programas del estudio de laboratorio

Borrarobjetosmenores.m

Programa que borra objetos muy pequeños dentro de una imagen de entrada a partir de un área mínima. A su salida, se obtiene la imagen una vez eliminados esos objetos, los objetos más grandes etiquetados de arriba abajo, las áreas de esos objetos y el orden de los mismos, tal que el primero es el mayor.

Calcpaso.m

Software que calcula el paso mínimo entre dos picos de la señal de entrada mediante el período de la misma.

Picosyvalles.m

Esta función calcula los picos y valles presentes en una cierta señal de entrada, devolviendo la posición de los máximos o picos y el mínimo entre valles.

Dividebasededatos.m

Programa para dividir la base de datos en entrenamiento y test, eligiendo las muestras de forma aleatoria para así conformar los vectores de entrenamiento y test que se introducen al LSSVM.

Demostrador3.m

Este programa servía para extraer las características de las muestras de la base de datos usando los patrones locales LBP, OLBP, LDP y LDerivP. Para ello se llevaban a cabo una serie de pasos:

1) Lectura de la base de datos de documentos.
2) Separación del documento seleccionado en líneas mediante el cálculo de la envolvente compleja.
3) División de cada línea obtenida en cuatro bloques horizontales.
4) Aplicación de los patrones locales sobre cada uno de los bloques.
5) Cálculo de los histogramas de cada bloque.
6) Concatenación de los resultados de cada uno de los bloques de la línea y normalización de la energía.
7) Almacenamiento de parámetros.

LSSVMsimuladorScript.m

Clasificador LSSVM editado para dividir la base de datos en entrenamiento y test, y además mostrar las curvas CMC y las tablas de confusión.

TestDemostrador3conBaseDatos1.m

Programa obtenido mediante la combinación del código de los dos anteriores con el fin de obtener los modelos de la base de datos obtenida mediante el traductor de Google y realizar el test usando las imágenes de texto impreso facilitadas por el *Indian Statistical Institute*.

8.2.2. Programas del estudio realista sobre texto impreso y manuscrito.

8.2.2.1. Base de datos

Docs2lines.m

Este programa se desarrolló para segmentar los documentos en líneas haciendo uso del histograma horizontal de las muestras como proceso principal para delimitar el principio y final de las líneas.

GenerarBaseDatosLineasenPalabras.m

Sistema semiautomático que permite segmentar las líneas en palabras. Este cuenta con tres opciones principales que consisten en eliminar áreas realizando cuatro clicks con el ratón sobre la imagen de la línea, y añadir o eliminar marcadores de separación De esta forma, el sistema además de establecer las marcas que considere, hace posible que el usuario realice las modificaciones que crea necesarias y verifique la correcta segmentación de la línea. El proceso seguido es:

1) Lectura de la base de datos y procesado de la imagen.
2) Opción de eliminar áreas, dibujando una figura con cuatro clicks sobre la imagen.
3) Muestra de los separadores determinados de forma automática.
4) Opción de eliminar marcas de separación con dos clicks.
5) Opción de añadir nuevos separadores también con dos clicks.
6) Guardar palabras con el nombre adecuado y habiendo realizado un histograma para eliminar las partes sobrantes de la imagen.

GenerarBaseDatosLineasenPalabrasThaiAutomatico.m

Este programa es similar al anterior pero fue desarrollado para trabajar con las escrituras malayalam, tailandés y japonés. Esto se debe a que el desconocimiento acerca de la estructura de los textos en estos alfabetos hacía que fuera necesario llevar a cabo una estrategia distinta. Dado que en estos casos una palabra podía ser en ocasiones una línea completa, fue necesario la incorporación de un apartado que se encargara de dividir los elementos detectados en secuencias de dos símbolos, tres, cuatro y así sucesivamente. El método que se ha efectuado en este caso consta de los siguientes pasos:

1) Lectura de la base de datos y procesado de la imagen.
2) Opción de eliminar áreas, dibujando una figura con cuatro clicks sobre la imagen.

3) Muestra de los separadores entre palabras determinados de forma automática.
4) Opción de eliminar marcas de separación con dos clicks.
5) Opción de añadir nuevos separadores también con dos clicks.
6) Análisis de cada palabra, añadiendo marcas tras dos, tres, cuatro símbolos, etc.
7) Opción de eliminar separadores.
8) Opción de añadir marcas de separación con un único click.
9) Guardar palabras con el nombre adecuado, realizando un histograma para eliminar las partes sobrantes de la imagen.

Eliminar_marcas.m

Programa cuya finalidad era eliminar pequeñas marcas que aparecían en los extremos de las imágenes obtenidas de los dos sistemas anteriores.

lineasproblematicas.m

Software encargado de separar las líneas en aquellos documentos donde la única estrategia válida de segmentación era llevar a cabo dicho procedimiento de forma manual.

Estos casos se daban mayoritariamente sobre texto manuscrito. Por ello, este programa recibe una imagen en escala de grises, marcada mediante un editor de imagen para indicar así dónde se han de separar sus líneas. Esta imagen es procesada a fin de encontrar dichas marcas y guardar cada una de las líneas que contiene el documento en cuestión, en archivos separados y correctamente numerados.

Cambiarnombreyadecuarapariencia.m

El *Indian Statistical Institute* nos facilitó una serie de imágenes correspondientes a palabras que estaban en distintos formatos, con el fondo blanco y el texto en negro y con un nombre de archivo no coincidente con el de nuestra base de datos.

En consecuencia, se elaboró un software capaz de modificar el formato de las imágenes a tipo TIFF, que también hiciera posible que la imagen tuviera el fondo negro y el texto en blanco como el resto de palabras de la base de datos y finalmente, que el nombre de la misma contara con 26 dígitos para así hacer más fácil la modificación del mismo mediante el programa *LupasRename*.

8.2.2.2. Extracción de características

Prog_general.m

Programa principal que a partir del nivel de trabajo (líneas, documentos o palabras), la naturaleza del texto (impreso o manuscrito), los tipos de escritura a enfrentar y los porcentajes de entrenamiento de cada uno, realiza las siguientes tareas:

1) División de las muestras de la base de datos de cada una de las escrituras involucradas en el experimento en entrenamiento y test.
2) Extracción de características de las muestras tras aplicar los patrones locales.
3) Generación de los vectores de entrenamiento y test a introducir en el clasificador LSSVM.
4) Proceso de clasificación del tipo de escritura, obteniendo las curvas CMC, las tablas de confusión y las tasas de identificación producto de la combinación de dos o más patrones locales.

Trainortest_largelines.m

Este software se encarga de almacenar en un fichero de texto los nombres de las imágenes que forman el conjunto de muestras de entrenamiento y en otro, el que se empleará para test. Para ello se introduce un determinado porcentaje de entrenamiento para cada uno de los tipos de escritura a comparar y se calculará sobre la moda del total de imágenes por cada uno de estos alfabetos para aplicar dicho porcentaje y calcular así el número de muestras de entrenamiento, tal que las restantes se destinan al proceso de test.

En este caso, como paso previo, se determina si la línea tiene suficiente contenido como para ser procesada, pretendiendo así eliminar las líneas que contengan una única palabra ya que afectan negativamente a los resultados a nivel de línea.

Trainortest_others.m

Este programa realiza el mismo procedimiento que el anterior, con la salvedad de que no es necesario descartar ninguna muestra pues se aplica sobre documentos y palabras.

Params.m

Este es el software encargado de la extracción de características a partir de los distintos patrones locales (LBP, OLBP, LDP y LDerivP) y es muy similar a Demostrador3.m visto en el apartado anterior.

Los documentos son divididos en bloques de 128x128 píxeles solapados un 20% y además se descartan aquellos cuya información no supera el 20% del total del área del mismo, antes de proceder al cálculo de los patrones locales. Las palabras y las líneas por otro lado, son divididas en cuatro franjas horizontales antes de hacer ese cálculo.

Los pasos que se llevan a cabo son:

1) Lectura de la base de datos al nivel que corresponda usando para ello la lista con los nombres de los ficheros de entrenamiento o test según se especifique en el programa.
2) Procesado de las imágenes según nivel y naturaleza.
3) División de cada muestra en bloques.
4) Aplicación de los patrones locales sobre cada uno de estos bloques.
5) Cálculo de los histogramas de los mismos.
6) Concatenación de los resultados de cada uno de dichos fragmentos y normalización de la energía.
7) Almacenamiento de parámetros y actualización del vector de índices.

Select_exp.m

Este programa se encarga de encontrar y concatenar los vectores generados a partir de los scripts involucrados en los experimentos a realizar, para así obtener los vectores de entrenamiento, test e índices en el directorio que contiene los programas del clasificador.

LSSVMsimuladorScript.m

Clasificador LSSVM editado para simplemente obtener los vectores de entrenamiento y test de la carpeta de trabajo, representar las curvas CMC y mostrar las tablas de confusión.

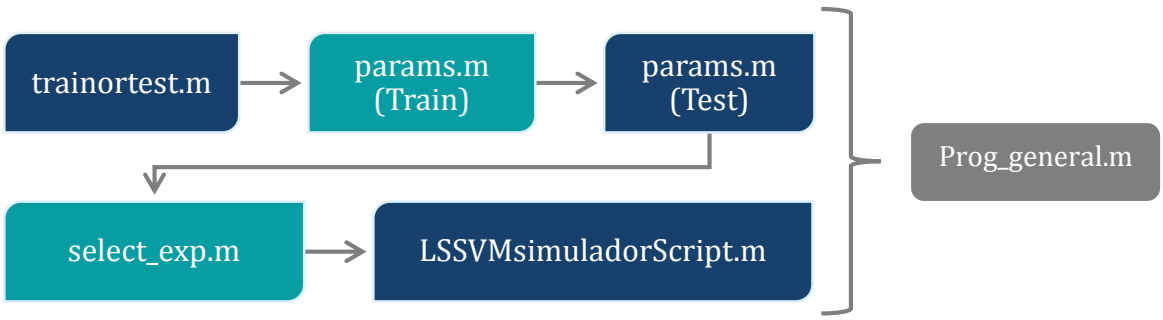


Figura 57. Diagrama de flujo del programa principal del estudio realista para texto impreso y manuscrito

Pliego de condiciones

9.1. Introducción

El pliego de condiciones es el documento en el que se hallan as cláusulas que regulan los derechos, responsabilidades, obligaciones y garantías mutuas entre los diferentes agentes asociados al proyecto. Consecuentemente, recoge las exigencias de índole técnica y legal que han de regir la ejecución del proyecto, teniendo efectos vinculantes.

En el presente apartado se ha dividido en dos partes:

- ***Pliego de condiciones técnicas.*** Conforman esta categoría todas las condiciones necesarias para la correcta instalación y ejecución de los programas desarrollados en el proyecto. Con este fin, se comentan los recursos materiales y herramientas necesarias.
- ***Pliego de condiciones legales.*** Contiene las condiciones o pautas legales en relación al desarrollo, aplicación y distribución del proyecto. Se tratan los derechos de autor, la licencia, las restricciones o garantías, entre otros temas de interés.

9.2. Pliego de condiciones técnicas

9.2.1. Requisitos mínimos

Recursos Hardware:

- Procesador > i5. Se recomienda un procesador i7.
- Memoria > 8 GB de RAM. Se recomienda una memoria de 16 GB de RAM.
- Monitor o pantalla. Se recomienda una resolución de 1024x728 píxeles.
- Teclado y ratón.
- Se recomienda el uso de un disco duro externo para la realización de copias de seguridad periódicas.

Algunos de los requisitos anteriores pueden ser cubiertos empleando un ordenador portátil o de sobremesa con unas características acordes a las anteriormente citadas.

Recursos Software:

- Sistema operativo mínimo: Microsoft Windows XP.
- Versión herramienta software: Mathworks Matlab R2007a.

- Librería de *Matlab Image Processing Toolbox.*
- Adobe Acrobat Reader para la lectura de los ficheros complementarios del directorio del DVD.
- Base de datos incluida en el DVD.
- Se recomienda la instalación de un procesador de texto y de hojas de cálculo, para poder trabajar sobre los resultados de la aplicación y así elaborar la documentación que se precise a partir de los mismos.

9.2.2. Instalación y ejecución del software

El software que se ha desarrollado a lo largo de este Proyecto de Final de Carrera se puede ejecutar desde cualquier ubicación a nivel de directorio, siempre que se cumpla con los requisitos hardware y software que se exigen. En esta línea, se puede trabajar con él sin necesidad de trabajar sobre el *Current Directory* de Matlab, únicamente se debe iniciar dicha aplicación y asignar la ubicación donde se encuentre el software de este proyecto como directorio predefinido.

Se recomienda asimismo consultar el manual de Ayuda o la página web de la aplicación en caso de dudas en lo que respecta al uso de la misma.

Hemos de resaltar que un cambio en la organización de carpetas o archivos del proyecto puede provocar el mal funcionamiento de éste, especialmente si las modificaciones se producen en las bases de datos.

En caso de modificar el código, será necesario revisar todos los ficheros y adaptar el nombre de cada una de las carpetas que se definen como directorios provisionales, tanto en la programación del estudio de laboratorio como en la del estudio realista.

Cabe destacar la importancia de la incorporación de la base de datos para poder ejecutar el código desarrollado en este proyecto.

9.3. Pliego de condiciones legales

Concesión de licencia:

Este Proyecto de Final de Carrera es propiedad de la Universidad de Las Palmas de Gran Canaria y cualquier usuario debe estar de acuerdo en obligarse por los términos y

condiciones establecidas en esta licencia del proyecto aceptando todas sus cláusulas. El uso de los distintos programas o ficheros, o de una copia en cualquier ordenador o computadora sólo podrá darse bajo la autorización expresa de la autora, los tutores del proyecto y de la Escuela de Ingeniería de Telecomunicación y Electrónica de la Universidad de Las Palmas de Gran Canaria.

Derechos de autor:

Este proyecto y su documentación asociada están protegidos por las leyes de propiedad intelectual que le sean aplicables así como las disposiciones de los tratados internacionales. Por consiguiente, el usuario deberá utilizar el material relativo al proyecto como cualquier producto protegido por derechos de autor. Sin embargo, se permitirá que el usuario pueda realizar una copia de los códigos fuente de programación y de la documentación del proyecto siempre que exista una autorización previa de la autora, los tutores del proyecto y de la Escuela de Ingeniería de Telecomunicación y Electrónica perteneciente a la Universidad de Las Palmas de Gran Canaria.

Restricciones:

El usuario no podrá realizar ingeniería inversa, de compilación o desensamblado del proyecto. El usuario podrá transferir el programa a un tercero bajo la autorización del autor a los tutores o de la Escuela de Ingeniería de Telecomunicación y Electrónica de la Universidad de Las Palmas de Gran Canaria siempre que no posea copia del mismo. Asimismo, la copia transferida podrá incluir posibles actualizaciones retener material escrito adicional que las pueda acompañar.

Garantía:

La autora y los tutores garantizan que las muestras y ficheros asociados al proyecto funcionarán correctamente en el momento de su uso. Análogamente, se garantiza que el soporte en el cual estén grabados los distintos programas no contendrá defectos en el momento de la adquisición del mismo.

La única excepción de lo dispuesto en el párrafo anterior es que los programas están creados sin garantías de ninguna clase. La autora y los tutores no aseguran, garantizan o

realizan ninguna declaración respecto al uso o los resultados derivados de la utilización de los ficheros o de la documentación asociada al proyecto.

Limitación de responsabilidad:

En ningún caso serán la autora, los tutores o la Escuela de Ingeniería de Telecomunicación y Electrónica de la Universidad de Las Palmas de Gran Canaria responsables de los perjuicios directos, indirectos, incidentales o consiguientes, gastos, lucro cesante, pérdida de ahorros, interrupción de negocios, pérdida de información comercial o de negocio, o cualquier otra pérdida que resulte del uso o de la incapacidad de usar los ficheros o la documentación del proyecto. El usuario conoce y acepta que los derechos de licencia reflejan esta asignación de riesgo como el resto de cláusulas y restricciones. Asimismo, la autora y los tutores de este proyecto rechazan cualquier otra garantía que no haya sido indicada anteriormente.

Otros:

En el supuesto de que cualquier disposición de esta licencia sea declarada total o parcialmente inválida, la cláusula afectada será modificada convenientemente de manera que sea ejecutable una vez modificada, plenamente eficaz, permaneciendo el resto de este contrato en vigencia y regido por las leyes de España.

Finalmente el usuario acepta la jurisdicción exclusiva de los tribunales de este país en relación con cualquier disputa que pudiera derivarse de la presente licencia.

Presupuesto

10.1. Declaración Jurada

Doña Nayara Rodríguez Rodríguez, autora del presente Proyecto de Fin de Carrera,

DECLARA QUE:

El proyecto Fin de Carrera con título "Detección e identificación de múltiples tipos de escritura en imágenes de texto", realizado a petición de la Escuela de Ingeniería de Telecomunicación y Electrónica de la Universidad de Las Palmas de Gran Canaria y en un periodo de 12 meses, tiene un coste total de **SESENTA Y NUEVE MIL SETENTA Y NUEVE EUROS Y VEINTICUATRO CÉNTIMOS** correspondiente a la suma de las cantidades consignadas a los apartados considerados a continuación.

Firmando la presente para que así conste a los efectos oportunos.

Autora del Proyecto:

Nayara Rodríguez Rodríguez

Las Palmas de Gran Canaria, 8 de Junio de 2015

10.2. Desglose del presupuesto

El presupuesto del proyecto realizado se ha obtenido según los precios de mercado actual, y de las indicaciones del COIT (Colegio Oficial de Ingenieros de Telecomunicación) y de la AEIT (Asociación Española de Ingenieros de Telecomunicación), a efectos de visado.

Dicho informe alberga la estimación de los gastos generados durante su realización y se fundamenta sobre los siguientes conceptos, detallados a continuación:

1. Amortización del inmovilizado material.
   - 1.1. Amortización del material hardware empleado.
   - 1.2. Amortización del material software empleado.
2. Trabajo tarifado por tiempo empleado.
3. Gastos por desplazamientos realizados.
4. Redacción del proyecto.
5. Derechos de visado del COIT.
6. Gastos de tramitación y envío.
7. Aplicación de impuestos.

10.2.1. Amortización del inmovilizado material.

Se trata de la corrección de valor por la depreciación del inmovilizado material del proyecto realizada de acuerdo con un plan sistemático definido con anterioridad.

Normas de valoración

El inmovilizado material se ha registrado a su coste de adquisición. No se han capitalizado ningún tipo de gastos financieros. No se han incurrido en gastos de reparación, conservación o mantenimiento.

A esta categoría pertenece la amortización del hardware y del software empleado en la realización del Proyecto de Final de Carrera. El sistema de amortización empleado ha seguido el método lineal, que distribuye el coste de los activos entre los años de vida útil de

los mismos de forma constante. Asimismo, dichos recursos están ubicados en la categoría de "Útiles, herramientas y equipos para tratamiento de la información, sistemas y programas informáticos" de las tablas de amortización propias del I.R.P.F.

La amortización se practicará elemento por elemento, si bien cuando se trate de elementos patrimoniales integrados en el mismo grupo de la tabla de amortización, la amortización podrá practicarse sobre el conjunto de ellos, siempre que en todo momento pueda conocerse la parte de la amortización correspondiente a cada elemento patrimonial.

Asimismo, los elementos de inmovilizado material nuevos cuyo valor unitario no exceda de 601,01 euros, podrán amortizarse libremente, hasta el límite de 3.005,06 euros anuales.

Según la tabla "*Útiles, herramientas y equipos para tratamiento de la información, sistemas y programas informáticos*", el número de años mínimos y máximos de amortización en esta categoría es de entre 2.5 y 5 años, por lo que para este proyecto, se ha estipulado en 3 años para cada uno de los elementos de tipo hardware empleados.

Teniendo en cuenta que la duración del proyecto ha sido aproximadamente de 1 año, y sabiendo que el cálculo del coste de amortización se constituye en un periodo de 3 años, los costes de amortización de la mayoría de los recursos utilizados se calcularán para el primer año.

Finalmente, la cuota de amortización anual será calculada haciendo uso de la siguiente ecuación:

$$Cuota\ de\ amortización\ anual = \frac{Valor\ de\ adquisición - Valor\ residual}{Nº\ de\ años\ de\ vida\ útil} \qquad \text{(Ec. 18)}$$

Donde el valor residual es el valor de cada uno de los elementos en cuestión después de su vida útil, teniendo en cuenta los índices de depreciación actual.

10.2.1.1. **Amortización del material hardware**

Puesto que la elaboración del proyecto ha precisado de 12 meses de trabajo y el cálculo del coste de amortización se estipula en un periodo de tres años, los costes serán calculados como los derivados del tiempo de utilización que se ha requerido por cada uno de los elementos hardware. Asimismo, los elementos de esta categoría que se han empleado en el proyecto han sido: ordenador portátil, ordenador de sobremesa y disco duro externo.

En la siguiente tabla se muestra la relación de cada uno de los elementos hardware con su valor de adquisición, su valor residual y el coste de amortización finalmente obtenido.

Elementos Hardware	*Coste*	*Valor residual (3 años)*	*Amortización*
Ordenador de sobremesa (laboratorio GPDS). Procesador i7, 8GB RAM, 500GB, monitor tipo LED.	1.150 €	0 €	383,33 € (1 año)
Ordenador portátil Asus F550D A10, 6GB RAM, 1TB.	599 €	0 €	199,67 € (1 año)
Escáner de documentos A4 Epson WorkForce DS-5500N	749 €	0 €	53,5 € (6 meses)
Disco duro externo 500GB	70 €	0 €	17,5 € (10 meses)
TOTAL			**654** €

Tabla 40. Precios y coste de la amortización del hardware.

Por consiguiente, el coste total del hardware empleado en el proyecto asciende a la cantidad de SEISCIENTOS CINCUENTA Y CUATRO EUROS.

10.2.1.2. Amortización del material software

De forma análoga, en este apartado se procederá a calcular la amortización del inmovilizado material de tipo software del proyecto.

Elementos Software	*Coste*	*Valor residual (3 años)*	*Amortización*
Paquete Microsoft Office 365 Pro Plus.	155 €	0 €	51,67 € (12 meses)

Sistema operativo Microsoft Windows 7. Versión Profesional, 64 bits.	135 €	0 €	27 € (12 meses)
Mathworks MATLAB. Versión r2007a académica.	3900 €	0 €	1300 € (12 meses)
TOTAL			**1378,67 €**

Tabla 41. Precios y coste de la amortización del software.

Por tanto, el coste total del software empleado ha sido de MIL TRESCIENTOS SETENTA Y OCHO EUROS Y SESENTA Y SIETE CÉNTIMOS.

10.2.2. Trabajo tarifado por tiempo empleado.

Normas de valoración

La aproximación del importe de las horas empleadas en la realización del presente proyecto con respecto a los honorarios finales estimados, está indicada en la siguiente fórmula de recomendación del COIT:

$$H = 74{,}88 \cdot H_N \cdot C_T + 96{,}72 \cdot H_E \cdot C_T \quad euros \qquad \text{(Ec. 19)}$$

En donde:

- H son los honorarios por tiempo.
- H_N son las horas trabajadas dentro de la jornada laboral.
- H_E son las horas especiales trabajadas.
- C_T es un factor de corrección en función del número de horas trabajadas.

Asimismo, el valor del factor de correlación según el número de horas trabajadas vendrá dado por la siguiente tabla:

HORAS TRABAJADAS	FACTOR DE CORRECCIÓN C_T
Hasta 36 horas	1,0
De 36 a 72 horas	0,90
De 72 a 108 horas	0,80
De 108 a 144 horas	0,70
De 144 a 180 horas	0,65
De 180 a 360 horas	0,60
De 360 a 540 horas	0,55
De 540 a 720 horas	0,55
De 720 horas a 1080 horas	0,45
Más de 1080 horas	0,40

Tabla 42. Factor de corrección del COIT según las horas trabajadas.

Para la realización del presente proyecto, se estima que se ha trabajado 1920 horas laborales (8 horas x 5 días x 4 semanas x 12 meses) y 192 horas especiales (4 horas x 4 semanas x 12 meses). Como el número de horas trabajadas es superior a 1080 horas, se aplicará un factor de corrección de 0,40.

De igual forma, habrá que descontar un número de horas asociadas a los periodos vacacionales o festivos de Navidades, Carnaval y Semana Santa que han afectado a la jornada laboral anteriormente definida. Como consecuencia, y teniendo en cuenta que dichos periodos vacacionales han acumulado un total de 168 horas para este caso, la cantidad total de horas efectivas trabajadas en jornada laboral ha sido de: 1920 – 168 = 1752 horas.

Por lo tanto, los honorarios asociados al total de horas trabajas en este proyecto han sido:

$$H = 74{,}88 \cdot 1752 \cdot 0{,}40 + 96{,}72 \cdot 192 \cdot 0{,}40 = 52475{,}90 + 7428{,}10 = 59904{,}00\ € \qquad \text{(Ec. 20)}$$

Obteniéndose una tarifa final por tiempo de ejecución de CINCUENTA Y NUEVE MIL NOVECIENTOS CUATRO EUROS.

Siendo el desglose por tiempo y fase del proyecto el que se muestra en la tabla:

DESCRIPCIÓN	TIEMPO	COSTE/MES	IMPORTE
Documentación	1 mes	4992 €	4992 €
Diseño	2 meses	4992 €	9984 €
Desarrollo	9 meses	4992 €	44928€
TOTAL	**12 meses**		**59904 €**

Tabla 43. Honorarios del proyecto según tiempo y fase del mismo.

10.2.3. Gastos por desplazamientos realizados.

Normas de valoración

En la ejecución del presente proyecto ha sido necesario el desplazamiento, en una media de 3-4 veces por semana durante once de los doce meses de duración del mismo. El itinerario seguido en todos los casos fue: Telde - Campus Universitario de Tafira (incluido retorno), con el objetivo de realizar diversas funciones del proyecto así como otras gestiones.

El desplazamiento se ha llevado a cabo tanto mediante vehículo propio como transporte público. Considerando que la distancia recorrida asciende a 20 kilómetros, el coste unitario es de aproximadamente 2 euros.

Teniendo en cuenta además que el período de ejecución del PFC ha sido de aproximadamente 12 meses además de las consideraciones anteriormente mencionadas, se pueden calcular los gastos de desplazamiento como:

$$G = días \cdot semanas \cdot precios = 5 \cdot 48 \cdot 2 = \mathbf{480}\ € \qquad \text{(Ec. 21)}$$

10.2.4. Costes del material fungible

Normas de valoración

El material fungible es aquel que no puede hacerse el uso adecuado a su naturaleza sin que éste se consuma en el proceso. Con respecto al proyecto realizado, los materiales fungibles empleados han sido:

CONCEPTO	COSTE
Papel de impresión	10,00 € (2 x 5€)
Tinta de impresión a color	25,90 €
Tinta de impresión negra	37,80 € (2 x 18,90€)
Encuadernación	10.00 € (2 x 5€)
DVD-R de 4,7GB	2,00 €
TOTAL	**81,70 €**

Tabla 44. Costes del material fungible del proyecto.

Ascendiendo entonces a OCHENTA Y UN EUROS CON SETENTA CÉNTIMOS.

10.2.5. Cuotas de redacción

El coste asociado a la redacción del proyecto se ha calculado según la siguiente expresión propuesta por el COIT:

$$R = 0{,}07 \cdot P \cdot C_h \qquad \text{(Ec. 22)}$$

Donde P es el presupuesto del proyecto obtenido y C_h un coeficiente de ponderación en función del presupuesto estipulado por el COIT, que varía según el valor de P. Dicho valor se ha calculado sumando los costes de las secciones anteriores correspondientes a la amortización del inmovilizado material, tanto hardware como software, al trabajo tarifado por tiempo empleado y los gastos por desplazamiento.

Así, el valor de P asociado a este proyecto ha sido:

CONCEPTO	COSTE
Amortización del material hardware	654 €
Amortización del material software	1378,67 €
Trabajo tarifado por tiempo empleado	59904 €
Gastos por desplazamiento	480 €

TOTAL	62416,67 €

Tabla 45. Presupuesto P para el cálculo del coste de redacción del proyecto.

Que según el COIT tiene un coeficiente de ponderación de 0,4 pues es el valor fijado para proyectos cuyo presupuesto se encuentra entre los 30.050€ y los 90.150€, que es nuestro caso.

Teniendo en cuenta el presupuesto calculado en la tabla anterior, se tiene que:

$$R = 0{,}07 \cdot P \cdot C_h = 0{,}07 \cdot \ 62416{,}67 \cdot 0{,}4 = \mathbf{1747{,}67}\ € \qquad \text{(Ec. 23)}$$

Por lo tanto, el importe final de redacción del proyecto asciende a la cantidad de MIL SETECIENTOS CUARENTA Y SIETE Y SESENTA Y SIETE CÉNTIMOS.

10.3. Derechos de visado

El COIT establece que para la redacción de proyectos y trabajos en general, los derechos de visado se calculan de acuerdo con la siguiente expresión:

$$V = 0{,}006 \times P \times C \qquad \text{(Ec. 24)}$$

Donde P representa el presupuesto total y C es el coeficiente reductor en función de dicho presupuesto.

El presupuesto total se obtiene de la suma de las secciones anteriores correspondientes al trabajo tarifado por tiempo empleado, la amortización del inmovilizado material y la redacción del proyecto, como se observa en la tabla:

CONCEPTO	COSTE
Amortización del material hardware	654 €
Amortización del material software	1378,67 €
Trabajo tarifado por tiempo empleado	59904 €

Gastos por desplazamiento	480 €
Redacción del proyecto	1747,67 €
TOTAL	**64164,34 €**

Tabla 46. Presupuesto P para el cálculo del visado del proyecto.

En función del presupuesto obtenido, se extrae el valor del coeficiente reductor del presupuesto C, que según el COIT, para presupuestos de más de 30.050€ y menos de 90.150€ viene definido con un valor de 0,80.

Por consiguiente:

$$V = 0{,}006 \times 64164{,}34 \times 0{,}8 = 307{,}99\ € \qquad \text{(Ec. 25)}$$

Finalmente, los costes por derecho de visado del proyecto ascienden a TRESCIENTOS SIETE EUROS Y NOVENTA Y NUEVE CÉNTIMOS.

10.3.1. Gastos de tramitación y envío

Los gastos de tramitación y envío son fijos y se estipulan por el COIT en 6,01€ por cada documento en un visado digital.

En consecuencia, los gastos de tramitación y envío ascienden a SEIS EUROS Y UN CÉNTIMO.

10.3.2. Presupuesto antes de impuestos

Sumando todos los conceptos calculados hasta el momento, se obtiene el total del presupuesto previo a la aplicación de impuestos, como se muestra a continuación.

CONCEPTO	COSTE
Amortización del material hardware	654 €
Amortización del material software	1378,67 €
Trabajo tarifado por tiempo empleado	59904 €
Gastos por desplazamiento	480 €
Redacción del proyecto	1747,67 €
Material fungible	81,70 €
Derechos de visado del COIT	307, 99 €
Gastos de tramitación y envío	6,01 €
TOTAL	**64560,04 €**

Tabla 47. Presupuesto sin impuestos.

El presupuesto calculado antes de impuestos asciende a SESENTA Y CUATRO MIL QUINIENTOS SESENTA EUROS Y CUATRO CÉNTIMOS.

10.3.3. Presupuesto después de impuestos

Añadiendo un 7% de IGIC sobre el presupuesto hallado en el apartado anterior se tiene:

TOTAL SIN IGIC	**64.560,04 €**
IGIC (7%)	4.519,20 €
TOTAL	**69.079,24 €**

Tabla 48. Presupuesto con impuestos.

El presupuesto total incluyendo los impuestos, asciende a la cantidad de SESENTA Y NUEVE MIL SETENTA Y NUEVE EUROS Y VEINTICUATRO CÉNTIMOS.

Las Palmas de Gran Canaria, a 8 de Junio de 2015

Fdo: Nayara Rodríguez Rodríguez

Anexo J. Contenido del DVD-R

11.1. Introducción

Junto con esta memoria se adjunta un DVD-R en el que se recopila el trabajo realizado a lo largo de este Proyecto de Final de Carrera.

El contenido de este DVD-R es el siguiente:

- Memoria en formato PDF.
- Funciones y scripts implementadas en Matlab.
- Bases de datos elaboradas para este proyecto.

11.2. Descripción del contenido

Se ha organizado el contenido en tres carpetas: Memoria, 1erEstudio y 2oEstudio.

- ***Memoria***: Documentación de la memoria del presente proyecto dividida en capítulos. También se incluye la bibliografía, pliego de condiciones, el presupuesto y los anexos.
- ***Estudio_Laboratorio:*** Esta carpeta contiene la base de datos y los programas creados durante el estudio de laboratorio.
- ***Estudio_Realista:*** Aquí se encuentra la base de datos realista de los últimos trabajos, la cual contiene todas las imágenes impresas y manuscritas a nivel de documentos, líneas y palabras. Esta carpeta también cuenta con la programación usada en esta fase del proyecto.