Universidad Autónoma del Estado de México
Facultad de Ingeniería
Maestría en Ciencias de la Ingeniería

Reconocimiento de notación matemática escrita a mano fuera de línea

TESIS

QUE PARA OBTENER EL TÍTULO DE:
Maestro en Ciencias de la Ingeniería

PRESENTA:

Carlos Ramírez Piña

Tutores Academicos:
Dr. José Antonio Hernández Servín
Dra. Rosa María Valdovinos Rosas

Tutores Adjuntos:
Dr. José Raymundo Marcial Romero
Dr. Marco Antonio Ramos Corchado

Toluca, México, 2017

Índice general

Índice de figuras

Página

Resumen

El reconocimiento automático de expresiones matemáticas es uno de los problemas de reconocimiento de patrones, debido a que las matemáticas representan una fuente valiosa de información en muchos áreas de investigación.

La escritura de expresiones matemáticas a mano es un medio de comunicación utilizado para la transmisión de información y conocimiento, con la cual se pueden generar de una manera sencilla escritos que contienen notación matemática. Este proceso puede volverse tedioso al ser escrito en lenguaje de composición tipográfica que pueda ser procesada por una computadora, tales como LaTeX, MathML, entre otros.

En los sistemas de reconocimiento de expresiones matemáticas existen dos métodos diferentes a saber: fuera de línea y en línea.

En esta tesis, se estudia el desempeño de un sistema fuera de línea en donde se describen los pasos básicos para lograr una mejor precisión en el reconocimiento, las cuales están divididas en dos pasos principales: reconocimiento de los símbolos de las ecuaciones matemáticas y el análisis de la estructura en que están compuestos. Con el fin de convertir una expresión matemática escrita a mano en una expresión equivalente en un sistema de procesador de texto, tal como TeX.

La fase principal se basan en el reconocimiento de símbolos, de esté proceso se realiza la propuesta de un algoritmo para la extracción de características de símbolos escritos a mano aislados, esta idea involucra los conceptos básicos de las técnicas de Wavelet y Zonas, esta fase representa uno de los elementos más importantes de un sistema reconocimiento matemático.

Con el propósito de validar la propuesta, se realizaron pruebas sobre una base de datos de tamaño mediano de símbolos matemáticos aislados que incluyen (letras mayúsculas y minúsculas del alfabeto latín y del alfabeto griego, dígitos y símbolos matemáticos) para obtener el desempeño de la nueva técnica híbrida en comparación con otros algoritmos de extracción de

características y analizar qué método es el más efectivo para cada clasificador y para cada tipo de símbolo. Los resultados muestran que el nuevo modelo se comporta significativamente mejor que el resto de los algoritmos probados, independientemente de la categoría de símbolos.

Capítulo 1

Introducción

El reconocimiento de expresiones matemáticas se basan en la forma en como se realiza la digitalización y la forma de analizar, mientras ciertos sistemas realizan el reconocimiento al final de la digitalización, otros mientras se escribe y se digitaliza comienzan el proceso del reconocimiento, cada proceso proporciona un enfoque diferente en el reconocimiento de patrones.

El estudio de estos enfoques a adquirido interés en las últimas décadas por investigadores en sistemas OCR (Optical Character Recognition).

En trabajos previos usualmente han dividido el reconocimiento en fases [7, 25], las cuales son el reconocimiento de símbolos y el análisis estructural. La etapa de reconocimiento de símbolos comprende un conjunto de procesos que se aplican a la imagen de entrada: pre-procesamiento, segmentación para aislar símbolos, extracción de características y clasificación. Por otro lado, el análisis estructural determina las relaciones entre los símbolos reconocidos para construir una estructura completa que representa la expresión matemática.

El alcance de este trabajo se centra en el campo específico del reconocimiento de símbolos matemáticos aislados. A partir de las diferentes operaciones incluidas en esta etapa, la extracción de características es uno de los elementos más importantes de un sistema de reconocimiento matemático, porque proporciona el conjunto de características utilizadas para describir cada símbolo con precisión.

1.1. Organización de la tesis

En base en el reglamento de los Estudios Avanzados de la Universidad Autónoma del Estado de México, la presente tesis está desarrollada en la modalidad de tesis por artículo especializado a ser publicado y está compuesta por los siguientes capítulos:

Capítulo 2 - Protocolo de Investigación: Presenta la versión del protocolo de tesis registrada ante la Secretaria de Estudios Avanzados de la Universidad Autónoma del Estado de México.
Capítulo 3 - Artículos: Se presentan los principales hallazgos realizados durante el trabajo de investigación:

- A simplified feature vector obtained by wavelets method for fast and accurate recognition of handwritten characters off-line.

  Este artículo fue aceptado y publicado en CEUR-workshop proceedings (LANMR 2016), Puebla, México. Este artículo presenta un algoritmo para la extracción simplificada de características basado en un método Wavelet para el reconocimiento fuera de línea del caracteres escritos a mano e incluye la comparación entre la técnica presentada con las propuestas por otros autores.

- A hybrid feature extraction method for recognition of offline handwritten mathematical symbols.

  Este artículo es enviado a revisión y posible publicacíon en la revista Journal: Expert Systems With Applications. Este artículo presenta un esquema de extracción de características para el problema del reconocimiento de símbolos escritos a mano de notación matemática. Es un modelo híbrido que involucra las ideas básicas de las técnicas de Wavelet y Zonas para definir los vectores de características con propiedades estadísticas y geométricas de los símbolos, con el objetivo de superar algunas limitaciones de los algoritmos individuales utilizados.

Capítulo 4 - Conclusiones y trabajo a futuro: Se presentan las conclusiones obtenidas en el desarrollo de esta investigación con las líneas adicionales a seguir en esté caso de estudio presentadas como trabajo a futuro.

Capítulo 2

Protocolo de Investigación

2.1. Antecedentes y Estado del Arte

El estudio del reconocimiento de caracteres escritos que contienen expresiones matemáticas se remonta a 1960, cuando Anderson [5] propuso un esquema al reconocimiento de expresiones escritas a mano. Este esquema se divide, en el reconocimiento de expresiones aritméticas y matrices, de tal modo que se puede generar escritos con fórmulas, notación matemática, entre otros. de una manera rápida mediante la escritura a mano. No obstante aún no es fácil transformar expresiones matemáticas escritas a mano de manera automática en una representación que pueda ser procesada por una computadora. La mayoría de las dificultades encontradas en el reconocimiento de tipografías de texto están presentes en la baja calidad del escaneo y la gran variedad de tipos de fuentes usadas en los documentos [20].

El Reconocimiento Óptico de Carácter (OCR) es una técnica usada para transformar diferentes tipos de documentos (el escaneo de un documento o la escritura a mano digitalizado por algún dispositivo) a un texto procesable por la computadora. El proceso general para OCR consiste en la digitalización, recupera los símbolos de la escritura y los convierte en una cadena de caracteres que pueden ser procesados por una computadora. El reconocimiento de símbolos específicos como puede ser la notación matemática se convirtió en un importante problema de reconocimiento de patrones [21].

El reconocimiento de expresiones matemáticas escritas a mano se puede realizar de dos formas, fuera de línea y en línea [18]. La diferencia en las dos categorías antes mencionadas se basa en la manera en cómo se produce y se analiza la escritura mano. Por un lado, para el reconocimiento fuera de línea los datos son considerados cómo una representación estática de texto, porque no se puede describir el orden en el que fueron impresos o escritos a mano, en general, los datos originales se escribieron, antes de que el proceso de reconocimiento se lleve a cabo.

Por otro lado, en el reconocimiento de expresiones matemáticas escritas a mano en línea, los datos originales son puntos y trazos. Los puntos tienen información de trazos obtenidos por el estilo de escritura en una tableta digital [21]. Y son normalmente almacenados en intervalos de tiempo regular. Los trazos son secuencias de puntos generados entre el lápiz abajo y lápiz arriba de un lápiz óptico. Los datos en línea también son conocidos como tinta digital, que es una representación *dinámica* de la escritura a mano. Contrario al caso de fuera de línea, el proceso de reconocimiento de datos en línea se lleva a cabo durante la escritura o inmediatamente después de ser finalizada [20].

En este proyecto se busca realizar la verificación de expresiones matemáticas escritas a mano fuera de línea mediante una metodología de OCR.

Aunque el reconocimiento de texto escrito a mano ha llegado a un punto de madurez, el reconocimiento de expresiones matemáticas sigue siendo aún un caso de estudio. A continuación se incluyen algunos estudios realizados al respecto.

Lee y Wang [13] presentan un sistema para la segmentación y la interpretación de expresiones matemáticas en los documentos fuera de línea. El sistema se divide en siguientes etapas: segmentación de caracter, extracción de características, reconocimiento de caracteres, formación de expresiones matemáticas, corrección de errores.

Chan y Yeung en [7] revisan algunos métodos para el reconocimiento de expresiones matemáticas, de las etapas de reconocimiento de símbolo y el análisis estructural, para datos en línea y fuera de línea, en el reconocimiento de símbolos se revisan métodos como la red neuronal, y otras aproximaciones estadísticas. En el análisis estructural, algunos métodos analizan las expresiones matemáticas usando reglas sintácticas explícitas mientras que otros obtienen la estructura sin análisis sintáctico. En ambas etapas se hace énfasis en las similitudes y diferencias de los sistemas.

Ernesto Tapia en [20] aborda el tema de reconocimiento de la escritura a mano en línea de expresiones matemáticas, en sistemas en los que se introducen los datos a través de dispositivos basados en lápiz digital, como tabletas digitales, pizarras contacto sensible, o Tablet PC. En particular, se desarrolla un reconocedor de expresiones matemáticas escritas a mano en línea. Para ello, estudia los diferentes métodos de clasificadores para el reconocimiento de los símbolos más utilizados en la notación matemática. También se enfoca en el desarrollo de un método robusto para el análisis estructural de la notación matemática.

Garain y Bidyut B. Chaudhuri [9] han estudiado los pasos principales para el reconocimiento óptico de caracteres de documentos científicos. Identifican problemas del reconocimiento de páginas completas que contienen expresiones específicas y que no llegan a reconocerlas con exactitud por las siguientes razones:

- La mayoría de los estudios se han ocupado de expresiones aisladas y por lo tanto, el problema para hacer frente a toda una página, que contienen expresiones matemáticas no se ha estudiado.

- El no contar con un conjunto de datos representativos sobre los métodos propuestos al ser utilizados en una situación real, en lugar de ser probados con un conjunto de datos limitados o con condiciones controladas.

Mencionan que es necesario una mayor investigación en la identificación de zonas de expresión en imágenes de documentos y en un método unificado para la evaluación del desempeño de sistemas en el reconocimiento de expresiones.

Christopher Malon [14] se enfoca en el reconocimiento de un carácter individual en la fase del análisis estructural. Demuestran la eficiencia de SVM (Support Vector Machine) en un problema multi-clase, con múltiples símbolos similares y muchas clases con pocos datos de formación. Muchos de los errores que quedan después de la aplicación de SVM, son la representación de caracteres que son indistinguibles sin información contextual (tales como el tamaño relativo de los caracteres), o que representan imágenes de carácter degradados.

Muhammad Imran Razzak [16] menciona la importancia del pre-procesamiento como fase crucial para el éxito de los sistemas de reconocimiento de caracteres eficientes. Se propone una nueva técnica para procesamiento previo tanto en línea como fuera de línea; se utiliza este método para eliminar las variaciones y aumentar la eficiencia del sistema de reconocimiento.

Anshul Gupta, Manisha Srivastava y Chitralekha Mahanta [10] realizan un estudio de varias técnicas de clasificación de características basadas en la escritura a mano en línea para el reconocimiento de caracteres. El método propuesto consiste en la segmentación de una palabra escrita a mano. Tres combinaciones de descriptores de Fourier se utilizan en paralelo como vectores de características, utilizando SVM como el clasificador. El procesamiento posterior se lleva a cabo empleando el léxico para verificar la validez de la palabra original.

Gurpreet Singh y Manoj Sachan [19] presentan una nueva técnica para el reconocimiento de caracteres escritos a mano fuera de línea, utilizando una Red Neuronal Perceptrón Multicapa (MLP) debido a su capacidad de ejecución en paralelo para hacer frente a la complejidad asociada con caracteres.

Fotini Simistira [18] hace la suposición de que los símbolos que pertenecen a la expresión matemática han sido reconocidos correctamente. Utiliza un clasificador SVM probabilístico para reconocer las relaciones espaciales entre dos símbolos matemáticos o sub-expresiones y después emplear un algoritmo basado en el método de análisis sintáctico que determina si una cadena puede ser generada por una gramática libre de contexto y si es posible, cómo ser generada (CYK) para analizar la expresión matemática y mostrar los resultados en páginas web haciendo uso de MathML(lenguaje de marcado basado en XML, cuyo objetivo es expresar notación matemática de forma que distintas máquinas puedan entenderla).

Francisco Alvaro Muñoz [4] propone un método para reconocer cualquier tipo de fórmula.Para ello, desarrolla el marco estadístico formal que deriva varias distribuciones de probabilidad. Explora algunas de las aplicaciones del reconocimiento de expresiones matemáticas, como por ejemplo la traducción a código TEX, mediante un demo de una pagina web para el reconocimiento de expresiones matemáticas escritas fuera de linea, el cual está limitado a un cierto conjunto de símbolos que puede reconocer.

En este proyecto se busca realizar la verificación de expresiones matemáticas escritas a mano fuera de línea mediante una metodología de OCR.

2.2. Planteamiento del problema

Las expresiones escritas a mano se pueden digitalizar o generar mediante la computadora, de este modo se podrán analizar el documento para que el equipo reconozca las expresiones directamente por medio de una imagen. Esto se puede realizar a través del reconocimiento óptico de caracteres (OCR).

Los caracteres y símbolos posiblemente son de diferentes tamaños. Todos los caracteres y los símbolos, cuando se agrupan correctamente, forman una estructura jerárquica interna. Sin embargo, la propia agrupación de los símbolos en una expresión matemática no es trivial. Por ejemplo, los dígitos 3, 8, y 1 tienen su propio significado como tal, pero si son del mismo tamaño y se encuentran en la misma línea, pueden representar el valor entero 381 [7].

También hay dos tipos de operadores a saber, operadores explícitos y operadores implícitos [21]. Los primeros son operadores de símbolos. Por ejemplo, la expresión $a + b$ es una relación explícita, la *suma* entre a y b está dado por el símbolo $+$. Mientras que los segundos son operadores espaciales. Ejemplo de una relación implícita sería en la expresión x^2. En este caso la expresión representa la relación *potencia* entre x y 2, una variación en la localización resulta en una relación muy diferente, por ejemplo x_2, que representa la relación subíndice.

Se necesita un análisis contextual para eliminar la ambigüedad de las funciones de símbolos matemáticos. Por ejemplo, una línea horizontal puede actuar como una línea de fracción, símbolo resta, lo cual confunde el orden, el alcance, e incluso la presencia de operaciones [**?**].

Derivado de lo anterior, surge la siguiente pregunta de investigación.

¿Cómo reconocer la escritura a mano de expresiones algebraicas (binomios, trinomios, identidades, series matemáticas, entre otros) a través de reconocimiento de patrones?

2.3. Meta de Ingeniería

Implementar un programa de reconocimiento de patrones, que permita la verificación de la escritura a mano de expresiones matemáticas y su traducción a código $\TeX$.

2.4. Justificación

Las expresiones matemáticas constituyen una parte esencial en la mayoría de las disciplinas científicas y de Ingeniería, la captura de estas expresiones matemáticas en una computadora son más difíciles que un texto plano, debido a que las expresiones matemáticas consisten en símbolos especiales, letras griegas, letras en latín, caracteres y dígitos.

Por la gran cantidad de elementos que constituyen a las expresiones matemáticas, al ir capturando en la computadora, se convierte en un proceso tedioso y tardado. Por tal motivo el reconocimiento de expresiones matemáticas escritas a mano, facilita este proceso y ayuda al usuario al realizar este tipo de escritos.

Al utilizar un sistema TEX éste es capaz de componer diferentes documentos de tipografía de alta calidad, es útil para formar expresiones matemáticas complejas, con la alineación apropiada de todos los elementos para esto es necesario conocer su sintaxis por ejemplo en texto matemático los símbolos se indican mediante un nombre de "palabra de control", que comienza con una barra invertida , por ejemplo, $\backslash alpha$ y $\backslash Omega$ produce α y Ω. Los subíndices se indican con un guión bajo _ por ejemplo a_2 es a_2, Agrupación se indica con llaves { } [23].

Al proponer una metodología de OCR para el reconocimiento de expresiones matemáticas escritas a mano y generar un código TEX ya no es necesario el conocer la sintaxis del sistema TEX, por lo cual reduce el tiempo de captura y facilita la escritura de documentos que requieran notación matemática. El reconocimiento de la escritura a mano de expresiones matemáticas requiere la elaboración de una base de datos de símbolos que contengan los reconcidos por el sistema TEX para solucionar la falta de disponibilidad de un conjunto de datos representativos, sobre los métodos propuestos son comprobados en un conjunto de datos limitado [9].

2.5. Objetivo general

Reconocer expresiones matemáticas escritas a mano mediante la aplicación de una metodología de OCR, para su traducción a código TEX.

2.6. Objetivos particulares

- Identificar métodos para el preprocesamiento de la imagen digital que contengan las expresiones matemáticas escritas a mano [15].

- Seleccionar los métodos de segmentación útiles para el reconocimiento de expresiones matemáticas para agrupar, o aislar los datos con el fin de clasificar unidades que representan un símbolo independiente [6, 7, 17].

- Elegir los métodos que sean necesarios para el procesamiento de símbolos independientes.

- Revisar y seleccionar los métodos adecuados para la normalización de símbolos con el fin de reducir inclinaciones en estilos de escritura, la orientación de la línea de base y el tamaño de los caracteres, palabras y componentes.

- Aplicar los métodos útiles para la extracción de características de los símbolos independientes.

- Analizar los métodos de clasificadores de símbolos para obtener la etiqueta del símbolo [2, 10, 14].

- Realizar el reconocimiento de las expresiones matemáticas escritas a mano para la transformación automática a un código en TEX.

2.7. Propuesta

La metodología propuesta para el logro del objetivo planteado es de la siguiente manera:

1. Revisión bibliográfica. Se accederá a fuentes de consulta formales para el estudio del estado del arte relacionado a los diferentes temas involucrados en el desarrollo del proyecto. Tales como: OCR, tratamiento de imágenes, clasificadores de símbolos, código TEX.

2. Con base en [11, 20, 21] se proponen las fases del reconocimiento de la escritura a mano fuera de línea de expresiones matemáticas (Figura 2.1).

    - Reconocimiento de símbolos.

        - Obtención de una imagen que contenga las expresiones matemáticas escritas a mano en documentos planos (hojas en blanco, pizarrones, entre otros). En este paso se transforma una expresión matemática escrita a mano en una imagen digital con los formatos más comunes, como JPG, PNG o la creación de una imagen a través de una interfaz de usuario utilizando el mouse como el dispositivo de entrada.

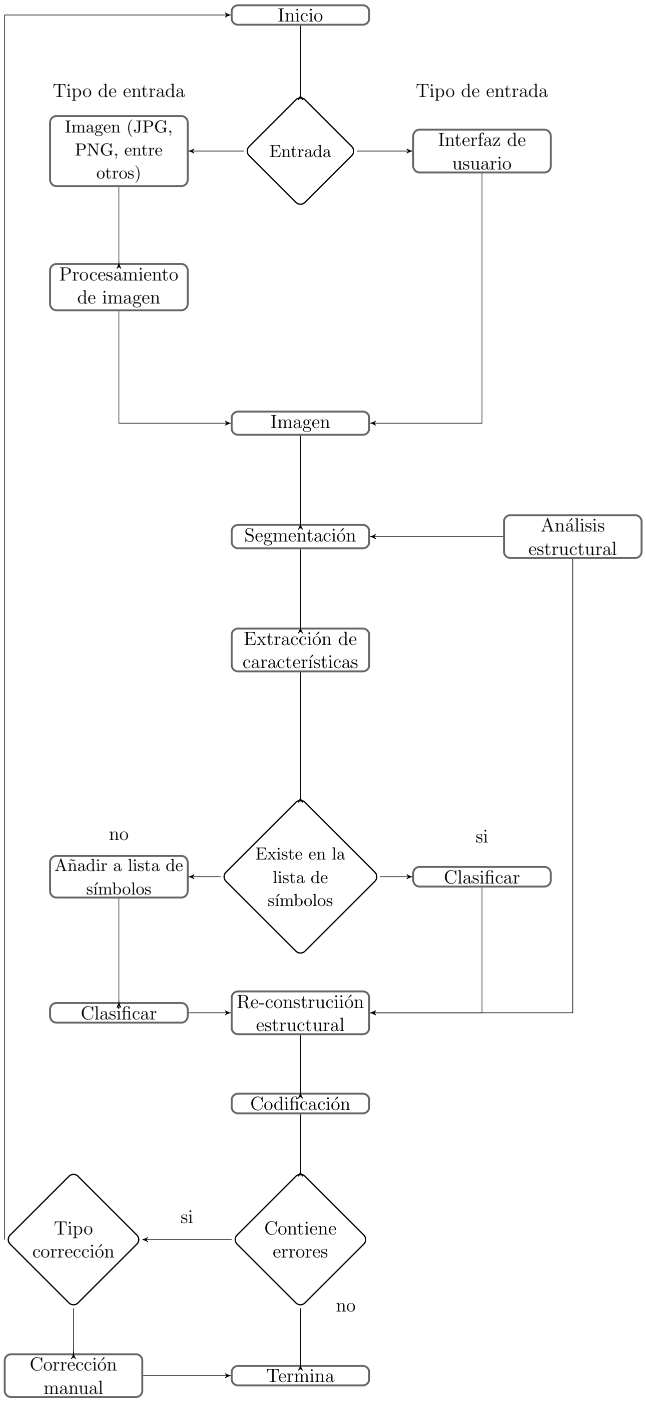


Figura 2.1: Metodología propuesta.

    - Procesamiento de la imagen. Es necesaria para realizar un preprocesamiento a la digitalización de las expresiones matemáticas escritas a mano, esto elimina ruido y reduce la cantidad de información a ser procesada por ejemplo convertirla en una imagen binaria [15].
    - Segmentación. Es el proceso de descomponer, agrupar, o aislar los datos con el fin de clasificar unidades que representan un símbolo independiente [12].
    - Extracción de características. Significa derivar medidas y las características de los datos que son útiles al hacer predicciones. Es común que los métodos de extracción de características se basan en la invariancia, reconstrucción y distorsiones esperadas [1, 8].
    - Clasificador de símbolo.
        - Lista de Símbolos. Una vez que se realiza la extracción de características se generará una BD que será el conjunto de entrenamiento para el clasificador.
        - Clasificador. Sirve para obtener la etiqueta del símbolo. Al formar las expresiones con las etiquetas del símbolo que da como resultado el clasificador, como pueden ser: SVM [2, 18, 20], Red Neuronal [3, 10, 19, 20].

- Análisis estructural.
    - Re-contrucción estructural. Con la lista de símbolos reconocidos se agrupan para representar una expresión y poder ser procesada por un programa de computadora, como ejemplo se puede utililizar un árbol de expansión mínimo [22, 24].
    - Codificación. Utilizando los resultados de la re-construcción estructural para obtener un resultado final, este resultado final puede ser un conjunto de caracteres que se representarán en un código $\TeX$.
    - Corrección de errores. Esto con el fin de verificar que el resultado del reconocimiento de expresiones matemáticas concuerda con lo escrito a mano.

- Presentación de los resultados.

2.8. Plan de trabajo

En el siguiente apartado se desglosan las actividades que se realizarán cada semestre.

Agosto 2015 - Enero 2016:

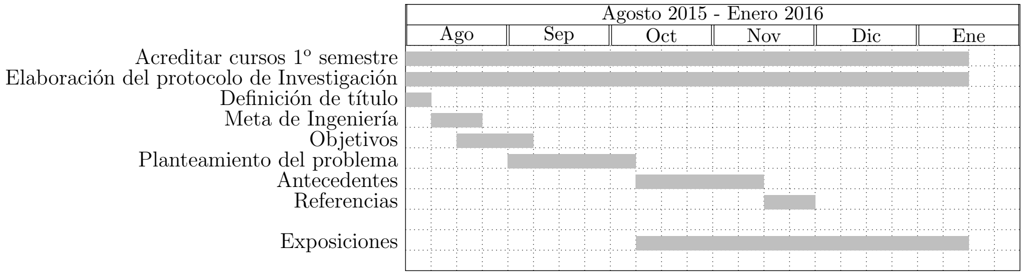


Figura 2.2: Cronograma de actividades periodo Agosto 2015 - Enero 2016

Febrero 2016 - Julio 2016:

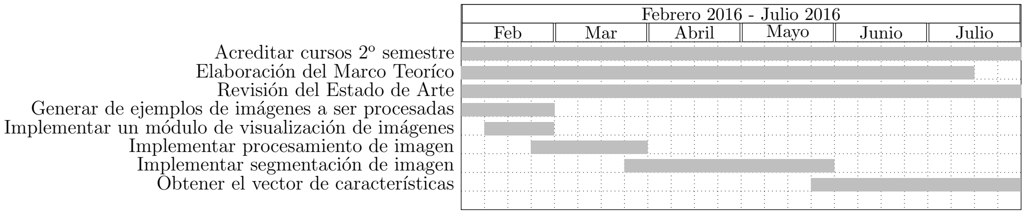


Figura 2.3: Cronograma de actividades periodo Febrero 2016 - Julio 2016

Agosto 2016 - Enero 2017:

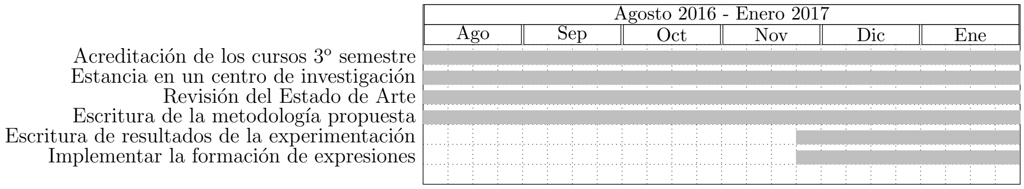


Figura 2.4: Cronograma de actividades periodo Agosto 2016 - Enero 2017

Febrero 2017 - Julio 2017:

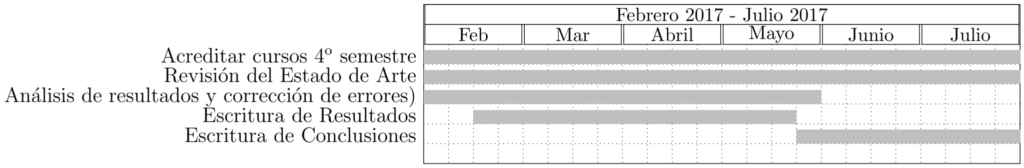


Figura 2.5: Cronograma de actividades periodo Febrero 2017 - Julio 2017

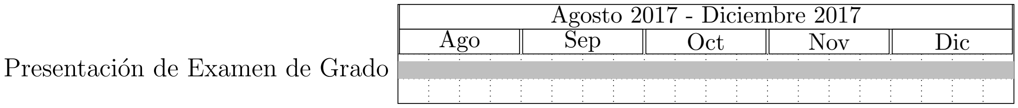


Figura 2.6: Cronograma de actividades periodo Agosto 2017 - Diciembre 2017

2.9. Resultados esperados

Durante el periodo de estudios de Maestría se espera obtener las siguientes contribuciones:

- Obtener el grado en tiempo y forma.
- Escritura de tesis de maestría.
- Publicación de un artículo científico con los principales hallazgos encontrados.
- Presentación de resultados obtenidos en un congreso de prestigio en el área.

Bibliografía

[1] Francisco Alvaro, J-A Sanchez, and J-M Benedi. Offline Features for Classifying Handwritten Math Symbols with Recurrent Neural Networks. pages 2944–2949, 2014.

[2] Francisco Álvaro and Joan Andreu Sánchez. Comparing several techniques for offline recognition of printed mathematical symbols. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 1953–1956. IEEE, 2010.

[3] Francisco Álvaro, Joan-Andreu Sánchez, and José-Miguel Benedí. Recognition of on-line handwritten mathematical expressions using 2d stochastic context-free grammars and hidden markov models. *Pattern Recognition Letters*, 35:58 – 67, 2014. Frontiers in Handwriting Processing.

[4] Francisco Álvaro Muñoz. *Mathematical Expression Recognition based on Probabilistic Grammars*. PhD thesis, 2015.

[5] Robert H Anderson. Syntax-directed recognition of hand-printed two-dimensional mathematics. *Symposium on Interactive Systems for Experimental Applied Mathematics: Proceedings of the Association for Computing Machinery Inc. Symposium*, pages 436–459, 1967.

[6] Richard G Casey and Eric Lecolinet. A survey of methods and strategies in character segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 18(7):690–706, 1996.

[7] Kam-Fai Chan and Dit-Yan Yeung. Mathematical expression recognition: a survey. *International Journal on Document Analysis and Recognition*, 3(1):3–15, 2000.

[8] Robin Clark, Quik Kung, and Anton Van Wyk. System for the recognition of online handwritten mathematical expressions. In *EUROCON, 2013 IEEE*, pages 2029–2035. IEEE, 2013.

[9] Utpal Garain and Bidyut B Chaudhuri. OCR of printed mathematical expressions. In *Digital Document Processing*, pages 235–259. Springer, 2007.

[10] Anshul Gupta, Manisha Srivastava, and Chitralekha Mahanta. Offline handwritten character recognition using neural network. In *Computer Applications and Industrial Electronics (ICCAIE), 2011 IEEE International Conference on*, pages 102–107. IEEE, 2011.

[11] Ming-Hu Ha, Xue-Dong Tian, and Na Li. Structural analysis of printed mathematical expressions based on combined strategy. In *Machine Learning and Cybernetics, 2006 International Conference on*, pages 3354–3358. IEEE, 2006.

[12] Widad Jakjoud and Azzeddine Lazrek. Segmentation method of offline mathematical symbols. In *Multimedia Computing and Systems (ICMCS), 2011 International Conference on*, pages 1–7. IEEE, 2011.

[13] Hsi-Jian Lee and Jiumn-Shine Wang. Design of a mathematical expression understanding system. *Pattern Recognition Letters*, 18(3):289–298, 1997.

[14] Christopher Malon, Seiichi Uchida, and Masakazu Suzuki. Mathematical symbol recognition with support vector machines. volume 29, pages 1326–1332. Elsevier, 2008.

[15] Sonali G Oval and RV Dagade. RECOGNITION OF HANDWRITTEN DEVANAGARI WORDS USING NEURAL NETWORK. *International Journal of Computer Engineering and Applications*, Volume IX, 2015.

[16] Muhammad Imran Razzak, Syed Afaq Hussain, Muhammad Sher, and Zeeshan Shafi Khan. Combining offline and online preprocessing for online urdu character recognition. In *Proceedings of the international multiconference of engineers and computer scientists*, volume 1, pages 18–20, 2009.

[17] Jungpil Shin. On-line handwriting character recognition using stroke information. In *Developments in Applied Artificial Intelligence*, pages 703–714. Springer, 2002.

[18] Fotini Simistira, Vassilis Katsouros, and George Carayannis. Recognition of online handwritten mathematical formulas using probabilistic {SVMs} and stochastic context free grammars. *Pattern Recognition Letters*, 53:85 – 92, 2015.

[19] Gurpreet Singh and Manoj Sachan. Multi-layer perceptron (MLP) neural network technique for offline handwritten Gurmukhi character recognition. In *Computational Intelligence and Computing Research (ICCIC), 2014 IEEE International Conference on*, pages 1–5. IEEE, 2014.

[20] Ernesto Tapia. *Understanding mathematics: A system for the recognition of on-line handwritten mathematical expressions.* PhD thesis, Freie Universität Berlin, Germany, 2005.

[21] Ernesto Tapia. A survey on recognition of on-line handwritten mathematical notation. In *Technical Report B-07-01.* Freie Universität Berlin, Germany, 2007.

[22] Ernesto Tapia and Raúl Rojas. Recognition of on-line handwritten mathematical expressions using a minimum spanning tree construction and symbol dominance. In *Graphics Recognition. Recent Advances and Perspectives*, pages 329–340. Springer, 2004.

[23] Herbert Voß. *Typesetting mathematics with LATEX.* UIT Cambridge Ltd., 2010.

[24] Qi Xiangwei, Pan Weimin, Wang Yang, et al. The study of structure analysis strategy in handwritten recognition of general mathematical expression. In *Information Technology and Applications, 2009. IFITA'09. International Forum on*, volume 2, pages 101–107. IEEE, 2009.

[25] Richard Zanibbi, Dorothea Blostein, and James R Cordy. Recognizing mathematical expressions using tree transformation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(11):1455–1467, 2002.

Capítulo 3

Artículos

En este capítulo se presentan los artículos que reportan la investigación realizada sobre el desarrollo del reconocimiento de notación matemática escrito a mano fuera de línea.

3.1. A simplified feature vector obtained by wavelets method for fast and accurate recognition of handwritten characters off-line

Artículo aceptado y publicado en: Latin American Workshop on Logic/Languages, Algorithms and New Methods of Reasoning 2016 (LANMR 2016), Puebla, México: `http://ceur-ws.org/Vol-1659/paper12.pdf`

A simplified feature vector obtained by wavelets method for fast and accurate recognition of handwritten characters off-line

Carlos Ramírez Piña[1], Vianney Muñoz-Jiménez[1], Rosa Maria Valdovinos Rosas[1], and J. A. Hernández Servín[1] ⋆

Facultad de Ingeniería,
Universidad Autónoma del Estado de México,
Toluca, Estado de México.
{cramirezp,vmunozj,rvaldovinosr,xoseahernandez}@uaemex.mx

Abstract. This paper presents an algorithm for simplified features extraction based on a wavelet method for off-line recognition of handwritten character. The proposal is applied to a set of 3250 handwritten symbols, which include the digits and the upper and lowercase character of English alphabet. The effectiveness of our algorithm is tested by comparison against the descriptors *FKI* and *Wavelets* using the Nearest Neighbour rule as classifier. The classification is measured in percentage of overall Accuracy and the processing time obtained by each methods.

1 Introduction

The study of character recognition is divided into off-line and on-line methods mainly [1]. The difference between them lies on how handwriting is done and analyzed. For the off-line recognition, the data are taken to be a static representation of text, since it can not be establish the order on which they were produced by a machine or handwritten [2]. On the other hand, in the on-line recognition, the original data are glyphs and points, which are normally storage on regular intervals of time [3].

This paper is focused on the off-line recognition of handwritten characters. The study is based on descriptors such as FKI [4] and discrete wavelets [5]. The dataset used in this work have been generated by [6] which includes digits and characters ($0 - 9$, A-Z, a-z). Our proposal was compared with the descriptors FKI and the discrete wavelet, in accuracy and processing time terms using the Nearest Neighbour rule *1-NN* as classifier.

1.1 The FKI offline features

The FKI algorithm was proposed by [4] which obtain a set of geometric features that has been used in handwriting recognition. That is, given a binary image

⋆ Corresponding author

$S(x, y)$ of size $M \times N$, the method computes nine geometrical features c_i where $i \in \{1, ..., 9\}$ for each entry column x such that $1 \leq x \leq M$. This is done on each column of the image, thus the method obtain $9N$ features in total. The authors also have features such as number of black and white pixels and their transitions, centre of gravity and second order moments.

1.2 Wavelets Descriptors

The wavelets are transformations which decompose an image into multi-resolution descriptions localized in space and frequency domain providing a smaller frames of the images. The frequency domain analyse different variations that has been successfully used in many image processing applications [7].

The DWT decompose the image S into wavelet blocks, an average image of smaller size than the original for a factor of two, and three more images containing the gradients and contours of itself, according to the following definitions:

$$W_g(j, m, n) = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} S(x, y) g^i_{jmn}(x, y) \tag{1}$$

$$W^i_h(j, m, n) = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} S(x, y) h^i_{jmn}(x, y) \tag{2}$$

where g is $g(x) = \begin{cases} 1 & x \in [0, \frac{1}{2}] \\ -1 & x \in [\frac{1}{2}, 1] \end{cases}$ and h belongs to the *Daubechies* family of mother wavelets; where as before $i \in \{H, V, D\}$. The wavelet blocks will be denoted by $A_j = W_g(j, m, n)$, $H_j = W^H_h(j, m, n)$, $V_j = W^V_h(j, m, n)$ and $D_j = W^D_h(j, m, n)$ where j is an index that indicates level of decomposition of the image (see Figure 1 (b)).

Frequency domain analysis is the background of representation of the feature vector. Different textural and statistical values are also computed which enrich the feature vector, like mean (μ) and standard deviation (σ) [5]. The type of entropies in the reference, which we have also implemented for comparison to our proposal, are like *shannon*, *Log energy*, *threshold*, *sure* and *norm*, which are computed on approximation the A_j coefficient block, as illustrated in Figure 1 (a).

2 Our Proposal

The main objective of the proposal method is to obtain an strategy which combine feature extraction methods in handwritten characters off-line and the recognition process of these characters in an accurate way. For that, segmentation and binarization methods were used before the actual feature extraction.

2.1 Binarization and segmentation

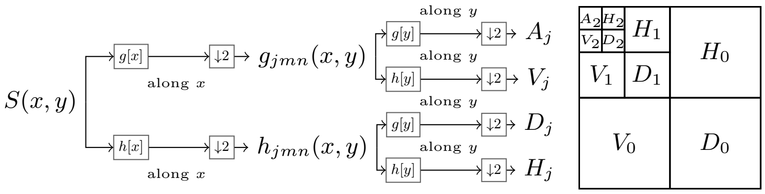
A pre-processing to the image is applied before feature extraction in order to eliminating noise of the image. In this way, firstly the images are converted into a binary type by analysing their histogram in a gray scale, in order to determine the optimal cut threshold. On a second stage, the symbol image is segmented extracting pixels corresponding to the symbol only. Finally, the symbol image are resized to a fixed size of 120 × 120. The size has been fixed in order to get optimal results when the wavelet transform is applied.

2.2 Feature extraction by a simplified vector feature using wavelets method

Feature extraction in the context of image processing, specifically in handwriting character recognition, is based on two types [8]; structured and statistical methods. The first one, are derived from the probability distributions of pixels, e.g. zones, first and second moments, projection and direction histograms. The second one, are based on topological and geometrical properties of the object under study.

The Wavelet transformation is used to compress an image by transforming it into the frequency domain [9]. In order to accomplish this, the image are represented using a set of basic functions produced by translation and scale up of a mother function. Let $S(x, y)$ be an input image, where x, y represent indexes, whereas $S(x, y)$ is the pixel value. In this paper, a 2D wavelet transform is used, the scaling of $S(x, y)$ is given by the functions g and h.

Coefficients wavelet analysis are obtained from three blocks; it was observed that wavelet coefficient of the third block are features of the input image, that is, it maintains representative information of the symbol. The wavelet transformation for the third state generate four images of size 15 × 15, A_2, H_2, V_2 and D_2 with 17 features correspondlly. The information from the approximation coefficients A_2 in third block keeps the information of the input image and the other four coefficients obtained represent 12% of the original image size and 25% of the size of the A_0 coefficient.



(a) (b)

Fig. 1: (a) Block diagram for calculating the DWT , (b) Wavelet decomposition indicating the block coefficients, A_j, etc.

For each coefficient obtained, were calculated the median, entropy and standard deviation; additionally five entropy wavelets are also calculated: Shannon, Log energy, Threshold, sure and norm; with this in mind we are reducing an amount of 77% the statistical measures as compared with the original method.

The Algorithm 1 represent the feature extraction of the vector formed by 21 features proposed for this study.

Algorithm 1 Simplified vector feature using Wavelet method

```
Require: Gray scale input image
Ensure: Set of 21 features
1: Convert image to binary type
2: Apply the wavelet transform to obtain the coefficients of the third block $A_2$, $H_2$,
   $V_2$, $D_2$ thus obtainig four features.
3: Calculate the mean ($\mu$), standard deviation ($\sigma$), entropy ($E$) thus giving 12 features
4: Calculate the entropies shannon, Log energy, threshold, sure, norm from $A_2$ thus
   generating five features at this stage.
5: Repeat steps 1 to 4, for each symbol image in order to form its feature vector.
```

3 Tools and Methods

3.1 Data set

The results here reported correspond to the experiments over the data set generated by [6], which includes digits $0-9$ with 10 classes and 527 feature vectors, the uppercase characters $A - Z$ form 26 classes and 1402 feature vectors, the lowercase characters $a - z$ with 26 classes and 1321 feature vectors.

For the data, the 10-fold cross-validation method was employed to estimate the classification error: 80% of the available patterns were for training purposes and 20% for the test set. On the other hand, we use as base classifier the 1-NN rule, expressed as [10]:

$$\delta_E(V_1, V_2) = \sqrt{\sum_{j=1}^{e}(V_1[j] - V_2[j])^2} \tag{3}$$

Where δ_E is the euclidean distance between vectors V_1 test feature and V_2 training feature .

3.2 The configuration of the method

The experiments were carried out datasets with different dimension of the feature vector, depending on the method used. That is:

- The FKI method, obtain nine features by column that containing the image, therefore the feature vector will have nine features by the number of columns that containing the image.
- Wavelts method obtain 55 features. The vector dimension is computed by the matrix of A_0, which generates $(\frac{x}{2})$ $(\frac{y}{2})$ features, where, x and y are the original image size, plus 54 features which represent the statistical averages.
- The Simplified vector features using Wavelet method obtain a vector with 21 features. That is, the whole of the features is $(\frac{x}{8})$ $(\frac{y}{8})$ plus 17 features which represent the statistical averages.

4 Results and Discussion

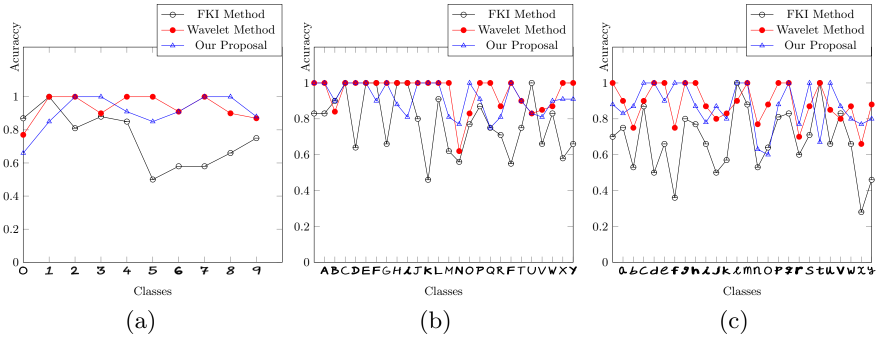
In this paper, we study two descriptor methods: FKI and Wavelets, in comparison with our wavelets method for recognition of handwritten characters off-line, in *Accuracy* and processing time terms. The *Accuracy* is obtained as follow:

$$Accuracy = 1 - \frac{M_e}{M}\,, \tag{4}$$

where M_e is the number of misclassified samples and M is the number of training samples.

4.1 Classifier performance

Figure 2, shows the 1-NN classification resuslts for each feature selection method here studied. The y axis represents the *Accuracy*, x axis correspond to the class. Some comments about these results are: First, it is clear that the recognition obtained from each method is not uniform by each class, however, for the digit dataset our method proposed shows a uniform behaviour with an average accuracy of 93.8%. On the other hand, the upper case dataset Wavelet method shows an uniform behaviour having an average accuracy of 93.0%. Finally, with the lower case dataset, Wavelet method abtain an average accuracy of 88.0%.



(a) (b) (c)

Fig. 2: Overall Accuracy (a) Digits dataset, (b) Uppercase characters dataset, (c) Lowercase characters dataset.

In order to identify the statistic significance between the methods, the Table 1, shows the average accuracy for each dataset, bold values represent the best results. For that, the rank of each method was calculated as follows [11]: For each dataset, the method with the best accuracy receives rank 1, and the worst receives rank 3. If there is a tie, the ranks are shared. Thus the overall rank of a method is the averaged rank of this method across the data set used. The results shown that the highest rank is obtained by the Wavelet method and the method with lowest rank is the FKI method.

Table 1: Overall Accuracy Performance

	FKI		Wavelet		Our proposal	
Dataset	μ	*Rank*	μ	*Rank*	μ	*Rank*
$0 \ldots 9$	77	(3)	89	(2)	**93**	**(1)**
A ... Z	78	(3)	**93**	**(1)**	88	(2)
a ... z	67	(3)	**88**	**(1)**	86	(2)
Average accuracy	74		90		89	
Average Rank		3		1.3		1.6

To complete the analysis of statistical significance between the results, the Namenyi test is used [11] $DC = q_\alpha\sqrt{\frac{K(K+1)}{6N}}$, where q_α is critical value, K is the number of methods to compare and N is the number of training set used. The test obtains a critical difference (CD) to reject the assumptions on which the corresponding p value is less than the adjusted α. In this paper we compare three feature selection methods and analyse their behaviour on three different datasets; the corresponding value for q_a are: $q_{0.05}$ is 2.343 and for $q_{0.10}$ is 2.052. The critical difference for $q_{0.5}$ is 1.913 and for $q_{0.10}$ is 1.675.

To interpret the results it is stated that a particular method A is significantly different than B, if the overall rank $(A) + CD <$ rank(B). From results in Table 1 it is posible to identify that the behaviour of our method and the Wavelets method do not offer statistic difference, that is to say that it is competitive with the Wavelets methdo. However, comparing the resulst respect to the FKI method, the Wavelets method is significatively better (1.3 (Wavelets Rank) $+1.675(CD_{0.10}) < 3$ (FKI Rank)).

4.2 Processing time

Table 2 shows the processing time using the different methods here studied.

As Table 2 shows, with our proposal the size of feature vector has less entries, in consequence it requires less processing time compared to the others methods. If we recall the classification results from the Table 1, our algorithm proposed show better accuracy than the FKI method and clearly competes with Wavelet method, reducing execution time with the classifier 1-NN.

Table 2: Processing time

Method	Features vector	Run time(sec.)
FKI	1082	1573.63
Wavelet method	3653	5183.59
Our proposal	915	1313.74

5 Conclusions and future work

In this paper we propose a method for reducing the feature vector for handwriting recognition in comparison to the results reported by [5], in which method obtain a vector with 55 features. Our method obtain a feature vector of 21 features only, using the third moment of the wavelet transformation. This allow us to reduce processing time compared to the FKI and traditional wavelet methods. That means, our algorithm reduces the processing time from 74.65% to 16.51% and decrease in size vector from 74.87% to 15% respect to FKI and Wavelet method respectively.

The future work will be focus on the processing of the dataset generated through a simplified vector feature using Wavelet method. We are in search to improve accuracy of the classifier by using the multilayer perceptron.

Acknowledgment. This work has partially been supported by the SEP-PRODEP-3238 and 3834/2014/CIA Mexican Projects and by the Mexican Science and Technology Council (CONACYT-Mexico) through the Masters scholarship 702528.

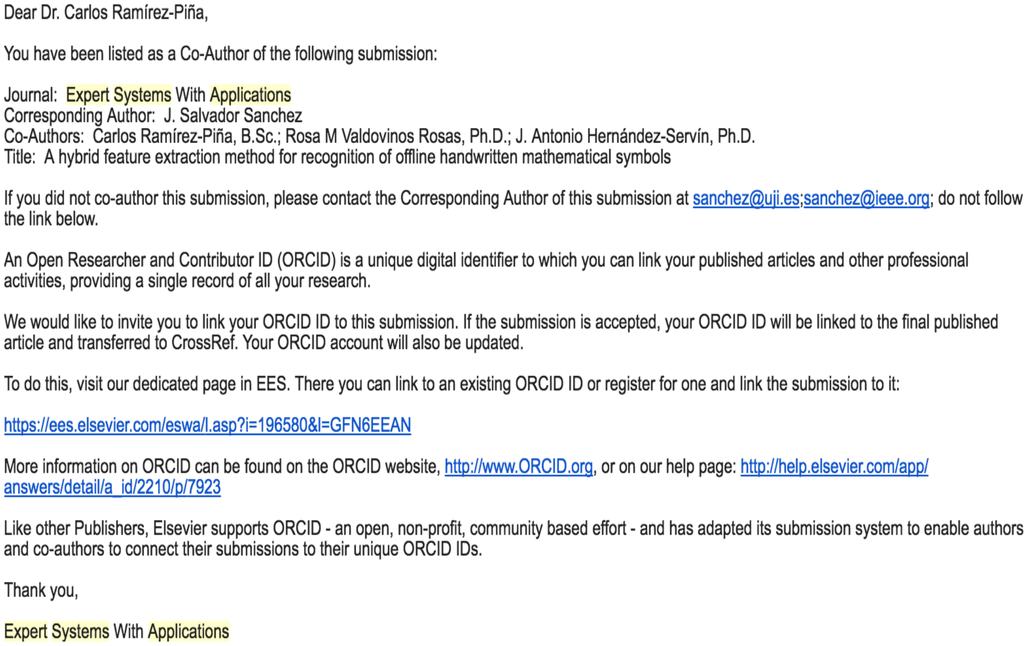
References

1. Fotini Simistira, Vassilis Katsouros, and George Carayannis. Recognition of online handwritten mathematical formulas using probabilistic {SVMs} and stochastic context free grammars. *Pattern Recognition Letters*, 53:85 – 92, 2015.
2. Ernesto Tapia. A survey on recognition of on-line handwritten mathematical notation. In *Technical Report B-07-01*. Freie Universität Berlin, Germany, 2007.
3. Ernesto Tapia. *Understanding mathematics: A system for the recognition of on-line handwritten mathematical expressions.* PhD thesis, Freie Universität Berlin, Germany, 2005.
4. F. Álvaro, J. A. Sánchez, and J. M. Benedí. Offline features for classifying handwritten math symbols with recurrent neural networks. In *Pattern Recognition (ICPR), 2014 22nd International Conference on*, pages 2944–2949, Aug 2014.
5. Sk Md Obaidullah, Chayan Halder, Nibaran Das, and Kaushik Roy. Numeral script identification from handwritten document images. *Procedia Computer Science*, 54:585–594, 2015.
6. Teófilo Emídio de Campos, Bodla Rakesh Babu, and Manik Varma. Character Recognition in Natural Images. *Proceedings of the International Conference on Computer Vision Theory and Applications, Lisbon, Portugal*, February 273–280, 2009.

7. K. B. Raja, S. Sindhu, T. D. Mahalakshmi, S. Akshatha, B. K. Nithin, M. Sarvajith, K. R. Venugopal, and L. M. Patnaik. Robust image adaptive steganography using integer wavelets. In *Communication Systems Software and Middleware and Workshops, 2008. COMSWARE 2008. 3rd International Conference on*, pages 614–621, Jan 2008.
8. Hedieh Sajedi. Handwriting recognition of digits, signs, and numerical strings in Persian. *Computers & Electrical Engineering*, 49:52–65, 2016.
9. R Colom, Rafael Gadea, A Sebastiá, Marcos Martínez, Vicente Herrero, and Vicente Arnau. Transformada Discreta Wavelet 2-D para procesamiento de video en tiempo real. *Actas de las XII Jornadas de Paralelismo*, 2010.
10. Cristina García Cambronero and Irene Gómez Moreno. Algoritmos de aprendizaje: knn & kmeans. *Intelgencía en Redes de Comunicación*, 2006.
11. Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7:1–30, 2006.

3.2. A hybrid feature extraction method for recognition of offline handwritten mathematical symbols

Este artículo se envió para su revisión y posible publicación en la revista internacional Elsevier Expert Systems With Applications.



Dear Dr. Carlos Ramírez-Piña,

You have been listed as a Co-Author of the following submission:

Journal: Expert Systems With Applications
Corresponding Author: J. Salvador Sanchez
Co-Authors: Carlos Ramírez-Piña, B.Sc.; Rosa M Valdovinos Rosas, Ph.D.; J. Antonio Hernández-Servín, Ph.D.
Title: A hybrid feature extraction method for recognition of offline handwritten mathematical symbols

If you did not co-author this submission, please contact the Corresponding Author of this submission at sanchez@uji.es;sanchez@ieee.org; do not follow the link below.

An Open Researcher and Contributor ID (ORCID) is a unique digital identifier to which you can link your published articles and other professional activities, providing a single record of all your research.

We would like to invite you to link your ORCID ID to this submission. If the submission is accepted, your ORCID ID will be linked to the final published article and transferred to CrossRef. Your ORCID account will also be updated.

To do this, visit our dedicated page in EES. There you can link to an existing ORCID ID or register for one and link the submission to it:

https://ees.elsevier.com/eswa/l.asp?i=196580&l=GFN6EEAN

More information on ORCID can be found on the ORCID website, http://www.ORCID.org, or on our help page: http://help.elsevier.com/app/answers/detail/a_id/2210/p/7923

Like other Publishers, Elsevier supports ORCID - an open, non-profit, community based effort - and has adapted its submission system to enable authors and co-authors to connect their submissions to their unique ORCID IDs.

Thank you,

Expert Systems With Applications

A hybrid feature extraction method for recognition of offline handwritten mathematical symbols

C. Ramírez-Piña[a], J.S. Sánchez[b,*], R.M. Valdovinos-Rosas[a], J.A. Hernández-Servín[a]

[a] *School of Engineering, Universidad Autónoma del Estado de México, Toluca, Mexico*
[b] *Institute of New Imaging Technologies, Dept. Computer Languages and Systems*
Universitat Jaume I, Castelló de la Plana, Spain

Abstract

This paper introduces a feature extraction scheme for the challenging problem of offline handwritten math symbol recognition. It is a hybrid model that involves the basic ideas of the wavelet and zoning techniques so as to define the feature vectors with both statistical and geometrical properties of the symbols, with the aim of overcoming some limitations of the individual algorithms used, thus yielding a further improvement of the recognition rate. Experiments over a medium-sized database of isolated math symbols investigate the performance of the new hybrid technique in comparison to other feature extraction algorithms, and analyze which method is the most effective for each classifier and for each type of symbol. The results show that the new model performs significantly better than the rest of algorithms tested, independently of the symbol category. Besides, it appears that the support vector machine and the multi-layer perceptron are especially appropriate classifiers for this hybrid feature extraction method.

Keywords: Feature extraction; Handwritten math symbol recognition

1. Introduction

Recognition of printed and handwritten mathematical expressions constitutes a research area of growing interest. In brief, it allows to transform formulas in scientific paper documents into an electronic representation (Tapia, 2005). An illustrative example of the interest in this area is

[*] Corresponding author. Tel.: +34-964-728350.
Email address: sanchez@uji.es (J.S. Sánchez)

the CROHME competitions (Mouchère et al., 2016), which provide a useful comparison between different techniques for mathematical expression recognition on a set of standardized tasks.

Although it might appear that the problem of mathematical expression recognition is equivalent to the recognition of plain text, there exist several differences that make unrealistic to apply the standard solutions of handwritten character recognition to mathematical notation. First, a line of text is one-dimensional and discrete, whereas symbols in mathematical expressions are spatially arranged into complex two-dimensional structures. Second, symbol recognition is a nontrivial problem because the vocabulary is very large (digits, Latin and Greek letters, operator symbols, opening, closing and fence symbols, binding symbols, relation symbols, etc.) with a variety of typefaces (regular, bold, italic, calligraphic) and several font sizes (subscripts, superscripts, limit expressions) (Blostein and Zanibbi, 2014). Third, mathematical handwriting may involve large operators such as matrix brackets, fraction bars or square roots. Besides, ambiguity in mathematical expressions is another problem that makes it especially difficult to define robust methods for symbol recognition; for instance, the same expression can be interpreted differently in different contexts because mathematical notation does not follow a completely formal language (Awal et al., 2010b).

Accurate recognition of a mathematical expression comprises two main steps (Chan and Yeung, 2000; Zanibbi et al., 2002): symbol recognition and structural analysis. The recognition stage translates the input image into a set of mathematical symbols present in the expression and therefore, this task is of most relevance because it determines the performance of the whole mathematical recognition system. In general, symbol recognition comprises a set of processes that are applied to the input image: pre-processing, segmentation to isolate symbols, feature extraction, and classification. On the other hand, the objective of structural analysis is to determine the relations among the symbols recognized in the previous stage in order to build a complete structure that represents the mathematical expression.

The scope of this paper focuses on the specific field of isolated mathematical symbol recognition, which is deemed to be a hard problem (Koerich et al., 2003). From the different operations included in this stage, feature extraction is one of the most important elements of a mathematical recognition system because it provides the set of features used to describe each symbol precisely. In fact, it is admitted that high recognition performance strongly depends on the quality of the features extracted from the raw image data (Trier et al., 1996).

In this work, a new feature extraction method for offline handwritten mathematical symbol recognition is proposed. This method combines the wavelet and zoning techniques in order to obtain a feature vector with both statistical and geometrical properties of the symbols, thus overcoming some limitations of those individual feature extraction algorithms and further improving the recognition rate. Efficiency of the new method is then investigated by comparing its performance with that of four well-known feature extraction techniques when using six classifiers of different nature over a medium-sized database of isolated mathematical symbols. The aim of this paper therefore is four-fold:

1. To introduce a new feature extraction method based on the wavelet and zoning algorithms;
2. to analyze the performance of this method for recognition of offline handwritten mathematical symbols;
3. to compare the new scheme with four well-known feature extraction techniques;
4. to explore how each algorithm performs depending on the different types of symbols.

From now on, the paper is structured as follows. Section 2 provides a brief review of a collection of related works. Section 3 outlines the feature extraction techniques used in this study. Section 4 introduces the new combined method for feature extraction and presents the methodology we have adopted for recognizing mathematical symbols. Section 5 describes the database and the set-up of the experiments carried out. Section 6 discusses the experimental results. Finally, Section 7 remarks the main findings and discusses possible avenues for future research.

2. Related works

A plethora of approaches to offline and online recognition of mathematical notation have been proposed in the past (Fateman et al., 1996; Garain and Chaudhuri, 2004; Garcia and Coüasnon, 2002; Liu et al., 2003; Mahmoud, 2008; Mayora-Ibarra and Curatelli, 1998; Tapia, 2005), and it still remains a challenging area in pattern recognition as revealed by the large number of papers published in recent years (Álvaro et al., 2014a,b; Awal et al., 2010a; Hu and Zanibbi, 2011; MacLean and Labahn, 2013; Naik and Metkewar, 2015; Nguyen et al., 2015; Sajedi, 2016; Saroui, 2015; Zhu et al., 2013).

Mayora-Ibarra and Curatelli (1998) proposed the use of a holographic associative memory for recognizing handwritten variations of the ten digits.

Each digit was represented as a feature vector constructed by dividing each character into 16 equal-sized squares, each one used to extract seven different features for recognition: smoothing, $+45$ and -45 degrees inclination tendency, $+60$ and -60 degrees inclination tendency, horizontal orientation tendency, and vertical orientation tendency.

Liu et al. (2003) presented the results of handwritten digit recognition on the CENPARMI, CEDAR and MNIST databases using state-of-the-art feature extraction and classification techniques. The features included chain-code feature, gradient feature with Sobel and Kirsh operators, profile structure feature, and peripheral direction contributivity; the classifiers tested were the nearest neighbor rule, three neural networks, the learning vector quantization model, a quadratic discriminant function and two configurations of support vector machine.

The online handwritten mathematical expression recognition system designed by Garain and Chaudhuri (2004) employed a multi-classifier approach, in which wide variations in shape and size of the large number of symbols were captured to form the feature vectors; in the first level a nearest neighbor classifier was for feature template matching, whereas the second level consisted of a hidden Markov model for classification.

Hanmandlu and Ramana-Murthy (2007) proposed the recognition of handwritten numerals by representing them in the form of exponential membership functions, which performed as a fuzzy model; their parameters were derived from features consisting of normalized distances obtained by dividing each image into a number of boxes.

Keshari and Watt (2007) combined online and offline feature extraction to obtain a hybrid system based on support vector machines in order to improve the recognition results. The online feature vector consisted of coordinates of each point on the stroke, sines and cosines of the angles made by the line segments on the stroke, sines and cosines of the turning angles between line segments and the center of gravity of the symbol. The offline feature vector was formed by the intensity (gray level) of each pixel in the image.

Mahmoud (2008) utilized hidden Markov models for the recognition of offline handwritten Arabic (Indian) numerals by extracting angle-span, distance-span, horizontal-span and vertical-span features. Vamvakas et al. (2010) proposed a feature extraction technique based on recursive subdivisions of the image in such a way that the resulting sub-images at each iteration had approximately the same number of foreground pixels; afterwards, a two-stage classification scheme based on the level of granularity of the feature extraction method was applied. Experiments were carried out over the MNIST and CEDAR handwritten digit databases.

Jou and Lee (2009) implemented a tree-like classifier for handwritten numeral recognition based on simplified structural classification and fuzzy memberships; they first extracted five kinds of primitive segments for each image, and then a fuzzy membership function was employed to estimate the likelihood of those primitives being close to the two vertical boundaries of the image.

The system developed by Hu and Zanibbi (2011) was based on a hidden Markov model using four-dimensional online local features and a new feature corresponding to the normalized distance to stroke edge in order to recognize isolated online handwritten mathematical symbols.

Vuong et al. (2010) extended the conventional elastic matching algorithm (Chan and Yeung, 1998) for symbol recognition. Apart from the Euclidean distance between points, the extended elastic matching scheme incorporated slope and curvature information during its matching process, which consisted of calculating the minimum distance between the template symbol and input symbol with dynamic programming.

Álvaro et al. (2014a) studied various sets of offline features for handwritten math symbol classification, comparing the performance of the PRHLT method proposed by Toselli et al. (2004), the FKI algorithm, a technique based on polar histograms (Su et al., 2013) and the vertical repositioning scheme (Dreuw et al., 2009).

Rajashekararadhya and Ranjan (2008) proposed an efficient zone-based feature extraction algorithm for handwritten numeral recognition of four popular south Indian scripts. They divided the binary image of a character into 50 zones and extracted two features from each zone, using the nearest neighbor classifier and a feed-forward back-propagation neural network for subsequent classification and recognition.

Sung et al. (2006) introduced a new algorithm for recognizing offline handwritten numerals by means of hierarchical Gabor features and a Bayesian neural network that encoded the dependency between features. Those hierarchical Gabor features were extracted by maximizing Fisher's linear discriminant in such a way that they were able to represent different levels of information.

The online handwritten digit recognition system proposed by Kherallah et al. (2008) utilized a feature extraction technique based on the Beta-elliptic representation consisting of a combination between geometry and kinematics in handwriting generation movements. Through this method of modeling, the dimension of the feature vector depends on stroke number of trajectory, which may result in high dimensionality.

3. Feature extraction techniques

As already pointed out, feature extraction plays an important role in any pattern recognition application because generating a good feature set allows to represent the underlying characteristics of each problem class and discriminate between them correctly. This section reviews the feature extraction algorithms used in the experiments, which correspond to four well-known methods that have extensively been applied to online and offline handwritten character recognition.

Feature extraction methods can be classified into two major categories: statistical and structural (Arica and Yarman-Vural, 2001). In the statistical approach, a character image is represented using a set of d features that are derived from the statistical distributions of pixels and can be considered as a point in d-dimensional feature space. In the structural category, various local and global properties of the character can be represented by geometrical and topological characteristics. It is worth remarking that structural and statistical features are deemed to be complementary in the sense that they emphasize different properties of the characters (Blostein and Zanibbi, 2014).

3.1. The FKI algorithm

Given a binary image I of size $M \times N$, the FKI algorithm (Marti and Bunke, 2001) computes a set of nine geometrical values for each image column y, thus obtaining 9-dimensional vectors $v(y) = [v_1(y), \ldots, v_9(y)]$. The algorithm uses a sliding window of size 1, moving from the very left of the image to the very right, to calculate the geometrical values reported in Table 1.

Therefore, as we are computing nine values for each column of the image, the dimension (d) of the feature vector given by the FKI algorithm is equal to $9N$.

3.2. The wavelet method

Wavelet transform is a multi-resolution signal decomposition tool that provides a a representation of an image at different levels of resolution. The present work utilizes 3-level Daubechies discrete wavelet transform, which recursively decomposes an input image I of size $M \times N$ into one low-frequency component (a thumbnail of the input image) and three high-frequency components for each level of decomposition j as illustrated in Fig. 1. The contour of the image is in the low-frequency sub-band and contains the approximation or scale coefficients (A_j), whereas the high-frequency sub-band includes

Table 1: Values calculated by the FKI algorithm

Number of black pixels in the window	$v_1(y) = \sum_{x=1}^{M} I(x,y)$
Center of gravity of the window	$v_2(y) = \frac{1}{M} \sum_{x=1}^{M} xI(x,y)$
Second order moment of the window	$v_3(y) = \frac{1}{M^2} \sum_{x=1}^{M} x^2 I(x,y)$
Position of the upper contour in the window	$v_4(y) = \min\{x \mid I(x,y) = \text{black}\}$
Position of the lower contour in the window	$v_5(y) = \max\{x \mid I(x,y) = \text{black}\}$
Orientation of the upper contour in the window	$v_6(y) = \frac{v_4(y+1)-v_4(y-1)}{2}$
Orientation of the lower contour in the window	$v_7(y) = \frac{v_5(y+1)-v_5(y-1)}{2}$
Number of black-white transitions in vertical direction	$v_8(y) = NT_{black->white} I(x,y);\ 1 \leq x \leq M$
Number of black pixels between the upper and lower contours	$v_9(y) = \sum_{v_4(y)<x<v_5yx)}^{M} I(x,y)$

the so-called detail coefficients H_j (horizontal), V_j (vertical) and D_j (diagonal).

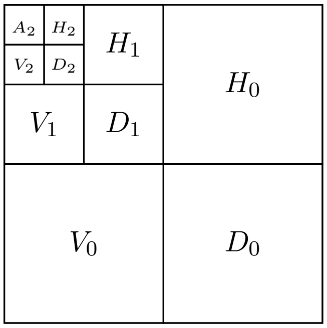


Figure 1: 3-level Daubechies wavelet decomposition

The input image is fed into two filters h and g, which produce the approximation coefficient A_j and the three detail coefficients H_j, V_j and D_j, which are all down-sampled by a factor of 2. Since images are two-dimensional structures, these filterings and sub-sampling are first applied along the rows of the image and then along the columns of the transformed image. As depicted in Fig. 1, the result of these operations is a transformed image with four distinct bands: the upper left band corresponds to a down-sampled by a factor of two version of the original image, the bottom left band tends to preserve localized horizontal features, the upper right band tends to preserve localized vertical features, and the bottom right band tends to isolate localized high-frequency point features in the image. As shown in Fig. 2, additional levels of decomposition can be applied only to the upper left band of the transformed image at the previous level in order to extract lower frequency features in the image.

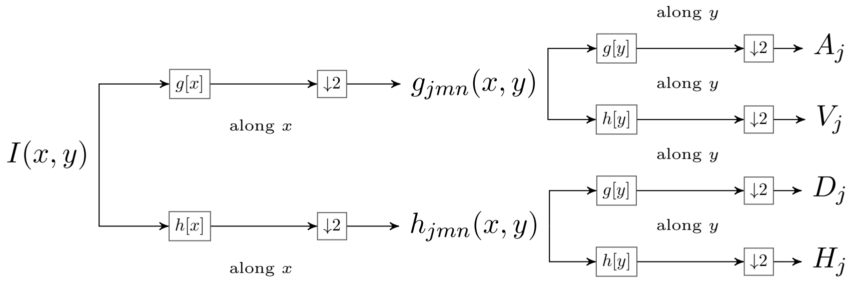


Figure 2: Block diagram of the two-dimensional discrete wavelet transform

Frequency domain analysis is the background of representation of the feature vector (with a size of $\frac{M}{2}\frac{N}{2}$), but some textural and statistical values are also computed to enhance the feature vector (Obaidullah et al., 2015). In particular, entropy, mean and standard deviation are computed on the gray-scale, binary and twelve sub-band images. Analogously, the Shannon entropy, the ‘log energy’ entropy, the threshold entropy, the sure entropy and the norm entropy are also calculated on the approximation coefficient sub-band. Thus the total number of features will be $(\frac{M}{2}\frac{N}{2}) + 54$. Table 2 reports the 54 values computed to enrich the feature vector constructed with the 3-level Daubechies discrete wavelet transform. Note that the first three values (v_1, v_2, v_3) correspond to a texture analysis on the gray-scale and binary images.

3.3. The zoning technique

The zone-based feature extraction algorithm (Bokser, 1992) used in this paper follows the foundations of the procedure proposed by Ashoka et al. (2012). It splits a binary image I of size $M \times N$ into a number of squared, non-overlapping zones or patches of a predefined size $m \times n$ (see Fig. 3(b)). For each zone Z_i, two values are calculated to build up the feature vector: one is the density of black pixels and the second corresponds to the normalized coordinate distance of black pixels.

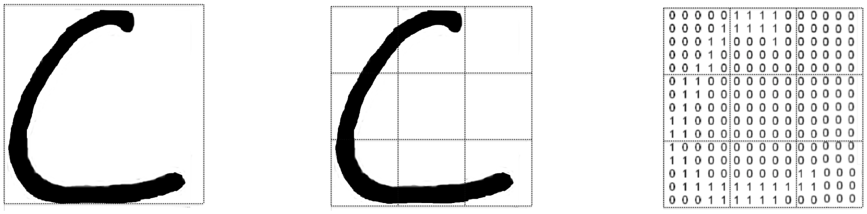
Table 2: Values computed to build up the feature vector with the 3-level Daubechies discrete wavelet transform

	Description
v_1 to v_3	Entropy, mean and standard deviation of the gray-scale and binary images.
v_4 to v_{15}	Entropy, mean and standard deviation of the coefficients A_0, H_0, V_0, D_0 corresponding to the Daubechies wavelet decomposition at first level
v_{16} to v_{20}	Shannon entropy, 'log energy' entropy, threshold entropy, sure entropy and norm entropy computed on approximation coefficient A_0
v_{21} to v_{32}	Entropy, mean and standard deviation of the coefficients A_1, H_1, V_1, D_1 corresponding to the Daubechies wavelet decomposition at second level
v_{33} to v_{37}	Shannon entropy, 'log energy' entropy, threshold entropy, sure entropy and norm entropy computed on approximation coefficient A_1
v_{38} to v_{50}	Entropy, mean and standard deviation of the coefficients A_2, H_2, V_2, D_2 corresponding to the Daubechies wavelet decomposition at third level
v_{51} to v_{54}	Shannon entropy, 'log energy' entropy, threshold entropy, sure entropy and norm entropy computed on approximation coefficient A_2

Firstly, a grid L of size $M \times N$ is superimposed on the image (see Fig. 3(c)), where the (x, y)-th element of L will be assigned to 1 if the pixel $I(x, y)$ is black and 0 otherwise. Then the density of black pixels in a zone Z_i can be computed as follows:

$$v_1(Z_i) = \frac{1}{mn} \sum_{l(x,y) \in L} l(x, y) \tag{1}$$

where mn is the total number of pixels in Z_i and $l(x, y)$ denotes the value of the (x, y)-th element of L.



0	0	0	0	0	1	1	1	1	0	0	0	0	0	0
0	0	0	0	1	1	1	1	1	0	0	0	0	0	0
0	0	0	1	1	0	0	0	1	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	0	0	0	0	0	0	0	1	1	0	0	0
0	1	1	1	1	1	1	1	1	1	1	1	0	0	0
0	0	0	1	1	1	1	1	1	0	0	0	0	0	0

a) Original binary image b) Splitting into zones of size 5 × 5 c) Superposition of a grid

Figure 3: An example of zoning applied to a binary image

For the second value, consider the bottom left corner of each grid as the absolute origin $(0, 0)$ and compute the coordinate distance of the j-th pixel in zone Z_i at location (x, y) as:

$$\delta_j(Z_i) = \sqrt{x^2 + y^2} \tag{2}$$

Then the normalized coordinate distance of black pixels can be obtained by dividing the sum of coordinate distances of black pixels (i.e., elements of the grid L whose value is equal to 1) by the sum of coordinate distances of all pixels in zone Z_i:

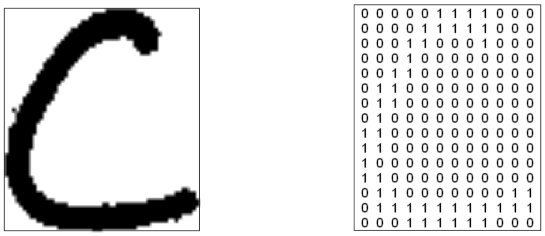
$$v_2(Z_i) = \frac{\displaystyle\sum_{j \in Black(Z_i)} \delta_j(Z_i)}{\displaystyle\sum_{j=1}^{mn} \delta_j(Z_i)} \tag{3}$$

where $Black(Z_i)$ denotes the set of black pixels in zone Z_i.

In this case, the total number of features given by the zoning algorithm will be $2(\frac{M}{m}\frac{N}{n})$.

3.4. The binarization algorithm

The binarization technique for feature extraction aims to minimize the useless information that can be present in an image (Choudhary et al., 2013). It is assumed that a binary image I has black pixels, which correspond to the characters, and white pixels for the background. Thus, as shown in Fig. 4, we can represent the image by a matrix $\mathbf{W} = [\mathbf{w_{xy}}]_{\mathbf{M \times N}}$ where the (x, y)-th component of $\mathbf{W}$ will be assigned to 1 if the pixel $I(x, y)$ is black, and to 0 for a white pixel.



0	0	0	0	0	1	1	1	1	0	0	0
0	0	0	0	1	1	1	1	1	0	0	0
0	0	0	1	1	0	0	0	1	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0	0	0	0	0
0	1	1	0	0	0	0	0	0	0	0	0
0	1	1	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0
0	1	1	0	0	0	0	0	0	0	1	1
0	1	1	1	1	1	1	1	1	1	1	1
0	0	0	1	1	1	1	1	1	0	0	0

a) Original binary image

b) Matrix representation

Figure 4: An example of the binarization technique (Choudhary et al., 2013)

This matrix $\mathbf{W}$ can be then reshaped in a row first manner to a column vector of size $M \times N \times 1$, which leads to a feature space of dimensionality $d = MN$.

4. Methodology

This section presents the proposed methodology, which follows the phases of a standard image recognition system. As shown in Fig. 5, the handwritten character recognition system consists of five primary stages (Gonzalez and Woods, 2008): image acquisition, pre-processing, segmentation, feature extraction, and recognition. Apart from describing the specific tasks performed at each stage, we will also introduce a new method for feature extraction, which is the result of hybridizing the foundations of wavelet and zoning algorithms.

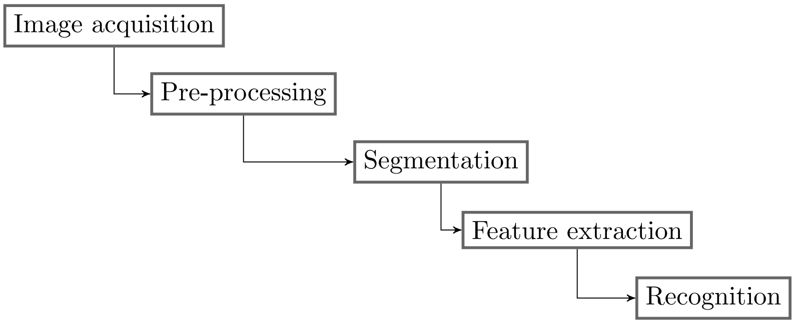
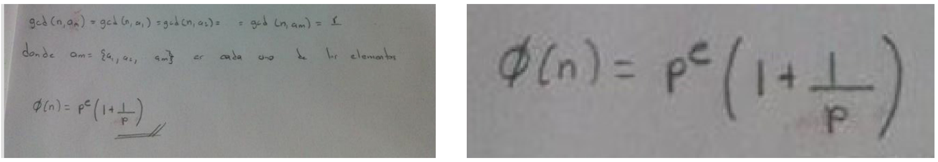


Figure 5: Block diagram of the handwritten character recognition system

4.1. Image acquisition

The images of symbols were obtained by scanning documents written by a pool of writers. The documents consisted not only of mathematical expressions, but also plain text and graphics (see Fig. 6(a)). Therefore, the region of interest that contained the handwritten mathematical expression was selected manually in order to reduce the amount of information and get an image formed only by a set of mathematical symbols (see Figura 6(b)).


gcd (n, an) = gcd (n, a1) = gcd (n, a2) = = gcd (n, am) = 1
donde am = {a1, a2, am} es cada uno de los elementos
ϕ(n) = pc(1 + 1/p)
ϕ(n) = pc(1 + 1/p)

a) Part of an original document

b) The mathematical expression selected

Figure 6: A piece of a handwritten document

4.2. Pre-processing

The goal of pre-processing is to enhance the quality of an image by means of a series of operations that are necessary to bring the input data into an acceptable form for further processes. The pre-processing stage may involve different tasks to be performed on the image, such as noise reduction, binarization, skew correction, slant and slope removal, edge detection, dilation and filling, smoothing, normalization, etc.

In this work, the only pre-processing operations that were carried out were noise reduction and binarization. Noise generated by shaded areas and dots was filtered using the non-linear median filtering technique

4.2.1. Binarization

The binarization process converts a gray-scale image into binary using a global thresholding method. For this purpose, those pixels in the image whose gray-levels are above a predefined threshold are set to white color and those below it are set to black color. However, the most critical element in this method is to select a proper threshold because it mainly depends on the image and on the particular conditions of the application in hand.

The algorithm that selects a threshold T for partitioning pixels into foreground and background pixels is based on the idea of Otsu's thresholding method, which assumes that the distribution of the pixel intensities is bimodal: dark pixel intensities (corresponding to the object or character) can be separated from light pixel intensities (corresponding to the background) in the gray-scale histogram.

We propose to analyze the histogram and determine the range of pixel intensities whose frequency values do not present variations greater than a constant value ψ, as shown in Figure 7(b). Thus the optimal threshold T can be calculated as follows:

$$T = \frac{\tau_{min} + \tau_{max}}{2} + \psi \tag{4}$$

where τ_{min} and τ_{max} represent the lowest and highest values of the range selected, respectively.

When the values of τ_{min} and/or τ_{max} in the analysis of the histogram are greater than a certain cut-off value of pixel intensity (in our experiments, this value was set to 100), the optimal threshold T is calculated as,

$$T = \frac{\tau_{max}}{2} + \psi \tag{5}$$

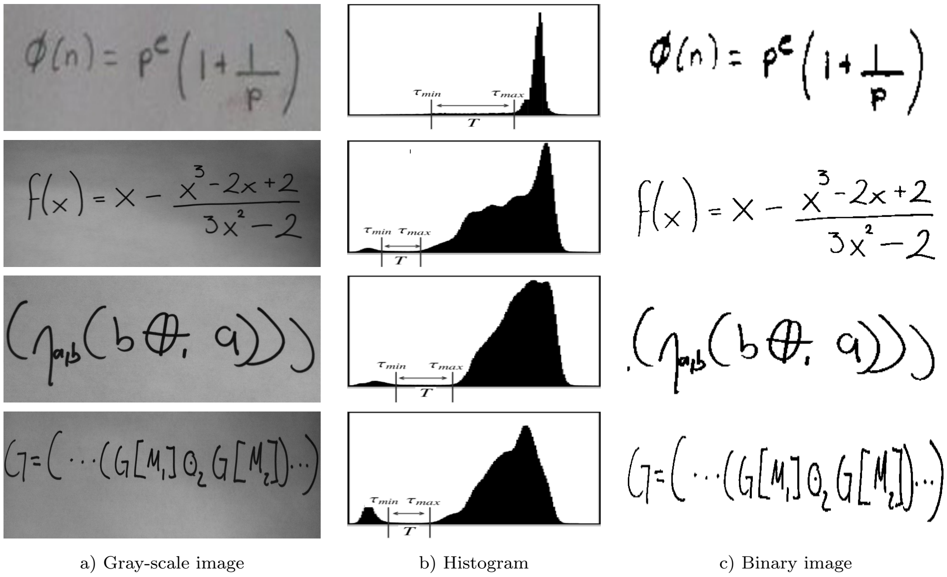


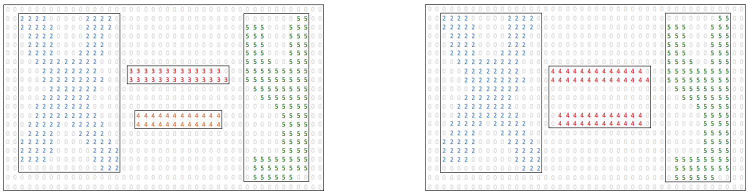
Figure 7: Some examples of the thresholding method

4.3. Segmentation

In the segmentation stage, the binary image that contains a mathematical expression is decomposed into sub-images, each one with an individual symbol that will be given to the next process. Unfortunately, this is a non-trivial task because of unconstrained handwritten expressions, overlapping and touching components, different symbol sizes, varied skew angles of symbols and identification of spatial relations of symbols within mathematical expressions (Simistira et al., 2014).

In this work, we chose a segmentation method based on labeling connected regions of the binary image, which correspond to symbols. To label a region, the algorithm starts from the first foreground pixel found and then, it propagates to any of the pixel's 4-neighbors. Each already visited pixel cannot be visited again, and after the entire connected region is labeled, its pixels are assigned a region number, and the procedure continues to search for the next connected region. Afterwards, each connected region, which is labeled with a region number, is enclosed by a bounding box (Fig. 8(a)).

The coordinates of these bounding boxes help to describe the relationships between the input symbols and distinguish independent symbols from those symbols composed of two or more strokes. In order to check whether two or more boxes correspond to the same symbol, we analyze some characteristics of the boxes, such as length, height, distance between boxes, and size. Boxes complying with these characteristics will be re-labeled, indicating that they belong to the same symbol (Fig. 8(b)).



a) Labeling of connected regions

b) Re-labeling

Figure 8: An example of the segmentation process

Finally, each symbol image was resized to a fixed size of 120 × 120 and converted to *.png* format so as to make all the images ready for the next stage.

4.4. Feature extraction by combining wavelets and zoning

In this section, we introduce a new method, hereafter called c-WZ, to extract discriminant features from the binary image to build up a feature vector using the bases of the discrete wavelet transform and the zoning technique in a combined manner. The rationale behind this strategy is to employ both statistical and geometrical characteristics of the image as recommended by several researchers (Blostein and Zanibbi, 2014) because the use of complementary information may lead to higher recognition rates.

To this end, the algorithms consists of two parts: the wavelet-based stage and the zoning-based stage. The resulting feature vectors are then joined to form the c-WZ feature vector with both statistical and geometrical characteristics of a symbol.

Firstly, the 3-level Daubechies discrete wavelet transform decomposes the binary image I in order to obtain the coefficients of the third block, which correspond to those with the most representative geometrical characteristics of the image. The approximation coefficient A_2 represents a thumbnail of I, whereas the detail coefficients H_2, V_2 and D_2 contain characteristics related to the contour of the symbol. Each of these coefficients is of size $\frac{M}{m_w}\frac{N}{n_w}$ with $m_w = n_w = 8$, leading to a total amount of 900 features. Next step consists of calculating the mean, the standard deviation and the entropy for the coefficients A_2, H_2, V_2 and D_2, which gives 12 more features. In addition, the Shannon entropy, the 'log energy' entropy, the threshold entropy, the sure entropy and the norm entropy are calculated for the approximation coefficient A_2. Therefore, the wavelet-based stage of the c-WZ algorithm produces a feature vector with 917 textural and statistical values as a result of the frequency domain analysis.

Next, the zoning-based part of the c-WZ algorithm will aim at endowing the feature vector with additional statistical characteristics of the input image. Similar to the procedure described in Section 3.3, the image I is divided into a number of squared zones of size $m_z \times n_z$ and two values are calculated for each zone Z_i: the total number of black pixels (instead of the density of black pixels as done in the standard zoning technique) and the normalized coordinate distance of black pixels. This produces a feature vector of size $2(\frac{M}{m_z}\frac{N}{n_z})$ with $m_z = n_z = 15$, that is, 128 values.

Finally, the feature vectors that result from both stages (wavelet and zoning) are now appended to build up the feature vector of the hybrid c-WZ algorithm, whose implementation is summarized in Algorithm 1.

Algorithm 1 c-WZ

```
Input: Binary image, $I$
Output: Feature vector, $[f_1, f_2, \ldots, f_d]$
Decompose $I$ using the 3-level Daubechies discrete wavelet transform
Obtain the coefficients of the third block ($A_2$, $H_2$, $V_2$, $D_2$)
for $A_2$, $H_2$, $V_2$, $D_2$ do
    Calculate the mean
    Calculate the standard deviation
    Calculate the entropy
end for
Calculate the Shannon entropy, the ‘log energy’ entropy, the threshold entropy, the sure entropy and the norm entropy for the approximation coefficient $A_2$
Split $I$ into zones of size $m_z \times n_z$
for all $Z_i$ do
    Compute the number of black pixels
    Compute the normalized coordinate distance of black pixels
end for
```

5. Experiments

The aim of the experiments here carried out is to compare the feature extraction method proposed in Section 4 (c-WZ) with FKI, wavelet, zoning and binarization, which correspond to four algorithms extensively applied to the recognition of handwritten characters. Six standard classification models with the parameter values reported in Table 3 have been used to analyze the performance of the feature extraction techniques: the nearest neighbor (1-NN) rule, the naive Bayes classifier (NBC), a Bayesian network (BN), a multi-layer perceptron (MLP), a support vector machine (SVM) with a linear kernel, and the C4.5 decision tree. The WEKA (Hall et al., 2009) and KEEL (Alcalá-Fdez et al., 2011) data mining and knowledge discovery software tools have been chosen to conduct the experiments.

Table 3: Parameter settings for the classifiers

1-NN	Euclidean distance
BN	Initial count for estimating the conditional probability tables = 0.5; Naive Bayes network used as the initial structure; Bayesian Dirichlet score to evaluate the structure learned
MLP	Learning rate = 0.3; Momentum = 0.2; Training time = 500; Hidden layers = (features + classes)/2
SVM	Linear kernel; Complexity = 1.0: Tolerance = 0.001; Epsilon = 1.0E-12
C4.5	Confidence factor = 0.25; Minimum number of examples per leaf = 2; Pruning by means of the subtree raising approach

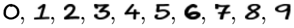
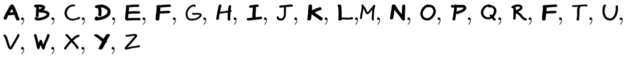
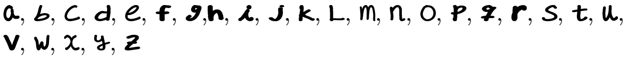
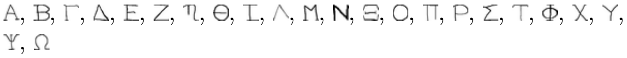
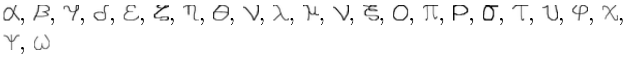
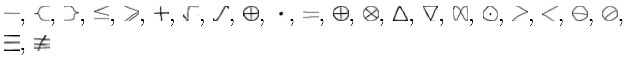
The 10-fold cross-validation method has been adopted for the experimental design because it provides some advantages over other resampling methods, such as bootstrap with a high computational cost or re-substitution with a biased behavior (Kim, 2009; Ounpraseuth et al., 2012). The original data set has been randomly divided into ten stratified segments or folds of equal (or nearly equal) size; for each fold, nine blocks have been used to fit the model, and the remaining portion has been held out for evaluation as an independent test set.

The classification models have been applied to the sets of samples generated by the feature extraction algorithms. The results from classifying the test samples have been averaged across the ten runs and then evaluated for significant differences using the non-parametric Friedman's ranking test and the Nemenyi's *post hoc* test (Demšar, 2006) at significance levels of $\alpha = 0.05$ and $\alpha = 0.10$.

5.1. Description of the database

The empirical analysis has been performed over the database generated by Campos et al. (2009), which includes digits (10 classes) with 527 samples, the uppercase Latin letters (up-Latin) with 26 classes and 1402 samples, and the lowercase Latin letters (low-Latin) with 26 classes and 1321 samples. Besides, by means of the methodology described in Sections 4.1-4.3, we have added the uppercase and lowercase Greek letters (up-Greek and low-Greek, respectively) with 24 classes each, and a miscellany of mathematical symbols (24 classes), all of them with 1320 samples. Putting these sets (types) of characters all together leads to a database with a total of 7210 samples of isolated symbols that belong to 134 different classes. For each set, Table 4 provides an example of the different handwritten mathematical characters that define the problem classes to be recognized in the experiments.

Table 4: Examples of the handwritten characters included in this study

Digits	
up-Latin	
low-Latin	
up-Greek	
low-Greek	
Math symbols	

Although all images were resized to a fixed size of 120 × 120 pixels, the dimension of the feature vectors depends on the particular characteristics of each feature extraction algorithm as summarized in Table 5.

Table 5: Dimension of the feature vectors obtained by the feature extraction methods

	Dimensionality	Parameters	# Features
FKI	$9N$		1080
Wavelet	$(\frac{M}{2}\frac{N}{2}) + 54$		3654
Zoning	$2(\frac{M}{m}\frac{N}{n})$	$m = n = 5$	1152
Binariz.	$M \times N$		14400
c-WZ	$4(\frac{M}{m_w}\frac{N}{n_w}) + 17 + 2(\frac{M}{m_z}\frac{N}{n_z})$	$m_w = n_w = 8$, $m_z = n_z = 15$	1045

6. Results

Table 6 reports the accuracy rates when using the feature extraction techniques with each classifier over the whole data set (7210 samples). The values for the best performing algorithm with each classification model are highlighted in bold face. As can be seen, the proposed c-WZ method achieved the highest recognition rates when using SVM, MLP and C4.5, whereas its accuracy were not too far from that of the best technique for the rest of classifiers (wavelet with BN, binarization with NBC, and zoning with 1-NN). In order to assess the statistical significance of these results, the Friedman's average rank for each algorithm was also calculated (note that the one with the lowest average rank corresponds to the best strategy), showing that the recognition rates using the c-WZ method were better than those obtained with any other feature extraction procedure.

Table 6: Accuracy rate and Friedman's average ranking over the whole data set

	BN	NBC	SVM	MLP	1-NN	C4.5	Rank
FKI	91.82	87.22	93.71	92.04	90.20	82.26	4.00
Wavelet	**92.65**	86.85	95.97	94.17	85.43	81.58	3.16
Zoning	91.50	88.05	95.76	94.14	**94.38**	83.35	2.66
Binariz.	90.58	**89.99**	95.60	93.87	93.34	81.18	3.50
c-WZ	92.51	89.20	**96.54**	**94.90**	93.15	**83.83**	**1.66**

The Nemenyi's *post hoc* test was then employed to discover whether or not there exist statistically significant differences in rank between the feature extraction methods. This test states that the performances of two or more algorithms are significantly different if their average ranks differ by at least the critical difference (CD).

The results of Nemenyi's test have been depicted by a significance diagram in Fig. 9. This plots the feature extraction algorithms against average rankings, whereby all methods have been sorted in ascending order of their ranks on the x-axis. The two horizontal lines, which are at height equal to the sum of the lowest rank and the critical difference, represent the threshold for the best performing method at each significance level ($\alpha = 0.05$ and $\alpha = 0.10$). This indicates that all algorithms above these cut lines perform significantly worse than the best technique.

The statistical comparison reveals that, despite differences in terms of accuracy rate were small in most cases, the algorithms differ significantly. The diagram shows that all the feature extraction techniques performed significantly worse than the proposed c-WZ method at $\alpha = 0.10$; at $\alpha = 0.05$, c-WZ outperformed significantly wavelet, binarization and FKI algorithms, but there was no enough evidence to conclude that zoning and c-WZ yielded equal performance.

After evaluating the performance of the feature extraction methods when all the characters were put into a unique data set, one should wonder whether these algorithms show the same behavior irrespective of the set of characters being analyzed or on the contrary, they perform differently with each

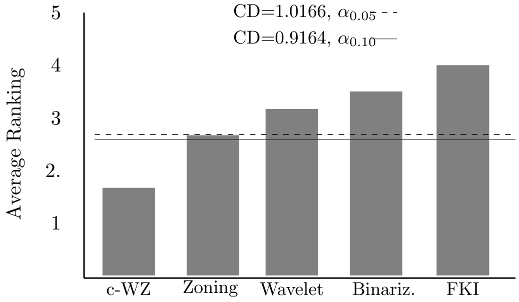


Figure 9: Significance diagram for the Nemenyi's test when using all characters

particular set. In order to outline an answer to these questions, Table 7 provides the accuracy results and the Friedman's average ranks for the feature extraction algorithms when these were applied to each individual set of characters. Bold-faced values of the accuracy highlight the best feature extraction algorithm for each classifier and each data set, whereas underlined values indicate the best performing classifier for each feature extraction technique and each set.

The only case in which the c-WZ method did not received the best Friedman's average rank corresponds to the set of digits, although it was very close to the lowest ranking assigned to the binarization approach. For all the other data sets, the algorithm proposed here showed the best overall behavior, that is, the lowest Friedman's average rank. In general, as already set out in Table 6, FKI and binarization were the feature extraction techniques with the poorest performance: FKI took the highest average rank when applied to digits, uppercase Latin letters and lowercase Latin letters, and binarization over the sets of uppercase Greek letters, lowercase Greek letters and math symbols. On the other hand, the results in Table 7 also reflect that the SVM was the classifier with the highest recognition rate independently of the feature extraction algorithm and the set of characters.

As in the case of the results for the whole data set, a Nemenyi's test at $\alpha = 0.05$ and $\alpha = 0.10$ was also applied to report any significant differences in rank between all pairs of algorithms for each individual set of characters, and then depicted by significance diagrams in Fig. 10. From these graphics, the following findings can be remarked:

- Digits (Fig. 10(a)): the binarization technique performed significantly better than wavelet and FKI at both $\alpha = 0.05$ and $\alpha = 0.10$. On the other hand, the c-WZ and zoning methods were also significantly

Table 7: Accuracy rate and Friedman's average ranking for each set of characters

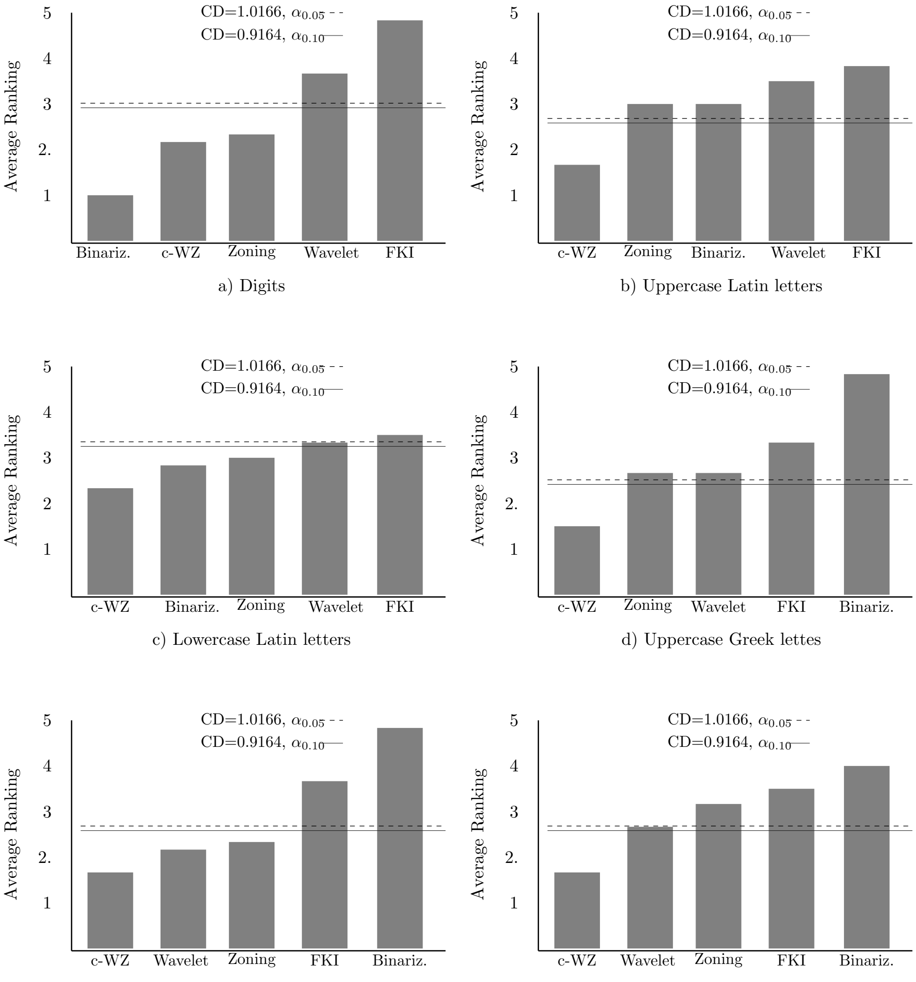
		BN	NBC	SVM	MLP	1-NN	C4.5	Rank
Digits	FKI	85.85	78.02	91.06	89.16	84.05	72.80	4.83
	Wavelet	**89.31**	79.46	95.47	93.87	83.01	73.94	3.66
	Zoning	88.85	84.32	96.31	93.95	93.11	**82.18**	2.33
	Binariz.	88.43	**84.35**	96.33	94.87	**94.24**	78.73	**2.00**
	c-WZ	89.06	83.21	**96.56**	**94.91**	91.89	77.39	2.16
up-Latin	FKI	91.44	85.37	92.32	89.22	85.40	80.36	3.83
	Wavelet	91.75	83.37	95.68	94.01	75.90	79.18	3.50
	Zoning	90.50	83.47	94.89	93.80	94.27	**81.70**	3.00
	Binariz.	88.07	**87.47**	94.93	93.37	**94.48**	80.33	3.00
	c-WZ	**91.89**	85.77	**95.77**	**94.40**	90.39	81.21	**1.66**
low-Latin	FKI	79.24	74.10	83.10	82.29	80.74	62.06	3.50
	Wavelet	**79.96**	69.36	87.64	83.97	71.81	54.82	3.33
	Zoning	76.50	73.98	86.99	83.42	**86.07**	**62.25**	3.00
	Binariz.	76.89	**77.83**	87.02	83.04	84.77	60.67	2.83
	c-WZ	78.66	75.19	**89.58**	**84.63**	82.55	60.47	**2.33**
up-Greek	FKI	98.87	97.61	99.08	97.95	98.14	95.92	3.33
	Wavelet	98.57	98.15	**99.29**	98.52	95.37	96.80	2.66
	Zoning	98.54	**99.09**	99.25	98.48	**98.56**	93.92	2.66
	Binariz.	97.36	97.46	98.81	97.50	96.00	91.63	4.83
	c-WZ	**99.05**	98.46	99.26	**98.60**	98.25	**97.36**	**1.50**
low-Greek	FKI	**99.72**	88.77	96.76	93.64	93.10	82.68	3.66
	Wavelet	96.39	90.89	97.79	94.97	86.70	84.82	2.83
	Zoning	94.86	87.61	97.20	95.38	94.28	80.46	3.33
	Binariz.	94.17	**93.53**	96.58	94.85	90.79	76.13	4.00
	c-WZ	96.45	92.68	**98.11**	**97.05**	**95.79**	**86.57**	**1.66**
Math	FKI	99.81	99.47	**99.97**	**100**	99.81	99.79	2.83
	Wavelet	**99.95**	99.92	**99.97**	99.72	99.81	**99.97**	2.50
	Zoning	99.75	99.83	99.95	99.85	**100**	99.61	3.33
	Binariz.	98.57	99.33	99.95	99.62	99.81	99.61	4.66
	c-WZ	**99.95**	**99.93**	**99.97**	99.85	**100**	**99.97**	**1.66**

better than wavelet and FKI at $\alpha = 0.10$.

- Uppercase Latin letters (Fig. 10(b)): the proposed c-WZ algorithm was significantly better than the remaining methods at both $\alpha = 0.05$ and $\alpha = 0.10$, whereas there were not significant differences between binarization, zoning, wavelet and FKI.
- Lowercase Latin letters (Fig. 10(c)): c-WZ was significantly better than FKI at both $\alpha = 0.05$ and $\alpha = 0.10$, and better than wavelet at $\alpha = 0.10$.
- Uppercase Greek letters (Fig. 10(d)): c-WZ performed significantly better than any other algorithm at both $\alpha = 0.05$ and $\alpha = 0.10$,

but there were not significant differences between binarization, zoning, wavelet and FKI.

- Lowercase Greek letters (Fig. 10(e)): c-WZ was significantly better than FKI and binarization at both $\alpha = 0.05$ and $\alpha = 0.10$.
- Math symbols (Fig. 10(f)): c-WZ performed significantly better than zoning, FKI and binarization at both $\alpha = 0.05$ and $\alpha = 0.10$, and better than wavelet at $\alpha = 0.10$.



a) Digits

b) Uppercase Latin letters

c) Lowercase Latin letters

d) Uppercase Greek lettes

e) Lowercase Greek letters

f) Math symbols

Figure 10: Significance diagram for the Nemenyi's test when using each set of characters

In summary, it is worth pointing out that the proposed c-WZ method appears to be the best feature extraction technique in almost all cases, followed by zoning and wavelet. Paradoxically, despite binarization and FKI were in general the worst methods, the former performed the best for the set of digits, as already seen in Table 7.

7. Conclusions and future work

In this paper, a hybrid feature extraction method for offline handwritten mathematical symbol recognition has been introduced. The bases of this model relies on the properties of statistical and geometrical characteristics of the symbol images, which have been obtained from the combined application of an extended version of discrete wavelet transform and a zone-based technique.

Experiments have revealed that the hybrid method here proposed performs significantly better than other four well-known feature extraction algorithms, both over the whole database with 7210 samples from 134 different classes and over almost each set of symbols. Besides, we have observed that the SVM and MLP classifiers can be deemed as the most appropriate recognition models to be used with the new method c-WZ. Another point of interest refers to the dimensionality of the feature space because the proposed c-WZ has led to feature vectors of size smaller than those of the remaining algorithms.

Some interesting directions for further research have emerged from this study. First, a natural extension is to develop a system that also incorporates structural analysis in order to recognize complete handwritten mathematical expressions instead of isolated symbols. Second, it would be interesting to assess the performance of c-WZ when it is applied together with deep neural networks or classifier ensembles, as well as to test more sophisticated feature extraction strategies. Finally, application of feature selection (and ranking) algorithms to reduce the dimension of the feature vectors constitutes another avenue that deserves some attention for future investigation.

Acknowledgment

This work has partially been supported by the Mexican CONACYT through the Student Support Program [CVU 702528], the Spanish Ministry of Economy [TIN2013-46522-P], the Generalitat Valenciana [PROMETEOII/2014/062] and the Mexican PRODEP [SEP-PRODEP-3238 and 3834/2014/CIA].

References

Alcalá-Fdez, J., Fernández, A., Luengo, J., Derrac, J., García, S., Sánchez, L., Herrera, F., 2011. KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. Journal of Multiple-Valued Logic and Soft Computing 17, 255–287.

Álvaro, F., Sánchez, J. A., Benedí, J. M., 2014a. Offline features for classifying handwritten math symbols with recurrent neural networks. In: Proceedings of 22nd International Conference on Pattern Recognition. Stockholm, Sweden, pp. 2944–2949.

Álvaro, F., Sánchez, J.-A., Benedí, J.-M., 2014b. Recognition of on-line handwritten mathematical expressions using 2D stochastic context-free grammars and hidden Markov models. Pattern Recognition Letters 35, 58–67.

Arica, N., Yarman-Vural, F. T., 2001. An overview of character recognition focused on off-line handwriting. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 31 (2), 216–233.

Ashoka, H., Manjaiah, D., Bera, R., 2012. Feature extraction technique for neural network based pattern recognition. International Journal on Computer Science and Engineering 4 (3), 331–339.

Awal, A. M., Mouchère, H., Viard-Gaudin, C., 2010a. Improving online handwritten mathematical expressions recognition with contextual modeling. In: Proceedings of the 12th International Conference on Frontiers in Handwriting Recognition. Kolkata, India, pp. 427–432.

Awal, A. M., Mouchère, H., Viard-Gaudin, C., 2010b. The problem of handwritten mathematical expression recognition evaluation. In: Proceedings of the 12th International Conference on Frontiers in Handwriting Recognition. Kolkata, India, pp. 646–651.

Blostein, D., Zanibbi, R., 2014. Processing mathematical notation. In: Doermann, D., Tombre, K. (Eds.), Handbook of Document Image Processing and Recognition. Springer, London, UK, pp. 679–702.

Bokser, M., 1992. Omnidocument technologies. Proceedings of the IEEE 80 (7), 1066–1078.

Campos, T. E., Babu, B. R., Varma, M., 2009. Character recognition in natural images. In: Proceedings of the International Conference on Computer Vision Theory and Applications. Lisbon, Portugal, pp. 273–280.

Chan, K.-F., Yeung, D.-Y., 1998. Elastic structural matching for on-line handwritten alphanumeric character recognition. In: Proceedings of the 14th International Conference on Pattern Recognition. Vol. 2. Brisbane, Australia, pp. 1508–1511.

Chan, K.-F., Yeung, D.-Y., 2000. Mathematical expression recognition: A survey. International Journal on Document Analysis and Recognition (1), 3–15.

Choudhary, A., Rishi, R., Ahlawat, S., 2013. Off-line handwritten character recognition using features extracted from binarization technique. AASRI Procedia 4, 306–312.

Demšar, J., 2006. Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research 7, 1–30.

Dreuw, P., Rybach, D., Gollan, C., Ney, H., 2009. Writer adaptive training and writing variant model refinement for offline Arabic handwriting recognition. In: Proceedings of the 10th International Conference on Document Analysis and Recognition. Barcelona, Spain, pp. 21–25.

Fateman, R., Tokuyasu, T., Berman, B., Mitchell, N., 1996. Optical character recognition and parsing of typeset mathematics. Journal of Visual Communication and Image Representation 7 (1), 2–15.

Garain, U., Chaudhuri, B. B., 2004. Recognition of online handwritten mathematical expressions. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 34 (6), 2366–2376.

Garcia, P., Coüasnon, B., 2002. Using a generic document recognition method for mathematical formulae recognition. In: Blostein, D., Kwon, Y.-B. (Eds.), Graphics Recognition Algorithms and Applications. Springer, Berlin, Germany, pp. 236–244.

Gonzalez, R. C., Woods, R. E., 2008. Digital Image Processing. Pearson Prentice-Hall, Upper Saddle River, NJ.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I. H., 2009. The WEKA data mining software: an update. ACM SIGKDD Explorations Newsletter 11 (1), 10–18.

Hanmandlu, M., Ramana-Murthy, O. V., 2007. Fuzzy model based recognition of handwritten numerals. Pattern Recognition 40 (6), 1840–1854.

Hu, L., Zanibbi, R., 2011. HMM-based recognition of online handwritten mathematical symbols using segmental K-means initialization and a modified pen-up/down feature. In: Proceedings of the International Conference on Document Analysis and Recognition. Beijing, China, pp. 457–462.

Jou, C., Lee, H.-C., 2009. Handwritten numeral recognition based on simplified structural classification and fuzzy memberships. Expert Systems with Applications 36 (9), 11858–11863.

Keshari, B., Watt, S., 2007. Hybrid mathematical symbol recognition using support vector machines. In: Proceedings of the 9th International Conference on Document Analysis and Recognition. Vol. 2. Kyoto, Japan, pp. 859–863.

Kherallah, M., Haddad, L., Alimi, A. M., Mitiche, A., 2008. On-line handwritten digit recognition based on trajectory and velocity modeling. Pattern Recognition Letters 29 (5), 580–594.

Kim, J.-H., 2009. Estimating classification error rate: Repeated cross-validation, repeated hold-out and bootstrap. Computational Statistics & Data Analysis 53 (11), 3735–3745.

Koerich, A. L., Sabourin, R., Suen, C. Y., 2003. Large vocabulary off-line handwriting recognition: A survey. Pattern Analysis & Applications 6 (2), 97–121.

Liu, C.-L., Nakashima, K., Sako, H., Fujisawa, H., 2003. Handwritten digit recognition: benchmarking of state-of-the-art techniques. Pattern Recognition 36 (10), 2271–2285.

MacLean, S., Labahn, G., 2013. A new approach for recognizing handwritten mathematics using relational grammars and fuzzy sets. International Journal on Document Analysis and Recognition 16 (2), 139–163.

Mahmoud, S., 2008. Recognition of writer-independent off-line handwritten Arabic (Indian) numerals using hidden Markov models. Signal Processing 88 (4), 844–857.

Marti, U.-V., Bunke, H., 2001. Using a statistical language model to improve the performance of an HMM-based cursive handwriting recognition system. International Journal of Pattern Recognition and Artificial intelligence 15 (1), 65–90.

Mayora-Ibarra, O., Curatelli, F., 1998. Handwritten digit recognition by means of a holographic associative memory. Expert Systems with Applications 15 (3–4), 399–403.

Mouchère, H., Zanibbi, R., Garain, U., Viard-Gaudin, C., 2016. Advancing the state of the art for handwritten math recognition: the CROHME competitions, 2011–2014. International Journal on Document Analysis and Recognition 19 (2), 173–189.

Naik, S., Metkewar, P., 2015. Recognizing offline handwritten mathematical expressions (ME) based on a predictive approach of segmentation using K-NN classification. International Journal of Technology 6 (3).

Nguyen, H. D., Le, A. D., Nakagawa, M., 2015. Deep neural networks for recognizing online handwritten mathematical symbols. In: Proceedings of the 3rd IAPR Asian Conference on Pattern Recognition. Kuala Lumpur, Malaysia, pp. 121–125.

Obaidullah, S. M., Halder, C., Das, N., Roy, K., 2015. Numeral script identification from handwritten document images. Procedia Computer Science 54, 585–594.

Ounpraseuth, S., Lensing, S. Y., Spencer, H. J., Kodell, R. L., 2012. Estimating misclassification error: a closer look at cross-validation based methods. BMC Research Notes 5 (656), 1–12.

Rajashekararadhya, S., Ranjan, P. V., 2008. Efficient zone based feature extration algorithm for handwritten numeral recognition of four popular south Indian scripts. Journal of Theoretical and Applied Information Technology 4 (12), 1171–1181.

Sajedi, H., 2016. Handwriting recognition of digits, signs, and numerical strings in Persian. Computers & Electrical Engineering 49, 52–65.

Saroui, S., 2015. Recognition of mathematical handwriting on whiteboards. Ph.D. thesis, School of Computer Science, University of Birmingham, UK.

Simistira, F., Papavassiliou, V., Katsouros, V., Carayannis, G., 2014. Recognition of spatial relations in mathematical formulas. In: Proceedings of the 14th International Conference on Frontiers in Handwriting Recognition. Crete Island, Greece, pp. 164–168.

Su, B., Ding, X., Peng, L., Liu, C., 2013. A novel baseline-independent feature set for Arabic handwriting recognition. In: Proceedings of the 12th International Conference on Document Analysis and Recognition. Washington, DC, pp. 1250–1254.

Sung, J., Bang, S.-Y., Choi, S., 2006. A Bayesian network classifier and hierarchical Gabor features for handwritten numeral recognition. Pattern Recognition Letters 27 (1), 66–75.

Tapia, E., 2005. Understanding mathematics: A system for the recognition of on-line handwritten mathematical expressions. Ph.D. thesis, Institute für Informatik, Freie Universität Berlin, Germany.

Toselli, A. H., Juan, A., Vidal, E., 2004. Spontaneous handwriting recognition and classification. In: Proceedings of the 17th International Conference on Pattern Recognition. Vol. 1. Cambridge, UK, pp. 433–436.

Trier, O. D., Jain, A. K., Taxt, T., 1996. Feature extraction methods for character recognition – a survey. Pattern Recognition 29 (4), 641–662.

Vamvakas, G., Gatos, B., Perantonis, S. J., 2010. Handwritten character recognition through two-stage foreground sub-sampling. Pattern Recognition 43 (8), 2807–2816.

Vuong, B.-Q., He, Y., Hui, S. C., 2010. Towards a web-based progressive handwriting recognition environment for mathematical problem solving. Expert Systems with Applications 37 (1), 886–893.

Zanibbi, R., Blostein, D., Cordy, J. R., 2002. Recognizing mathematical expressions using tree transformation. IEEE Trans. on Pattern Analysis and Machine Intelligence 24 (11), 1455–1467.

Zhu, S., Hu, L., Zanibbi, R., 2013. Rotation-robust math symbol recognition and retrieval using outer contours and image subsampling. In: Doermann, D., Tombre, K. (Eds.), Proceedings of SPIE Document Recognition and Retrieval XX. Vol. 8658. Burlingame, CA, pp. 1–12.

Capítulo 4

Conclusiones y trabajo futuro

En este capítulo se presentan las conclusiones obtenidas de este trabajo de investigación, además se presentan las posibles líneas de investigación y trabajo futuro.

4.1. Conclusiones

En esta tesis se desarrollo la metodología de reconocimiento de escritura a mano de expresiones matemáticas en donde se analizo cada una de las fases para el reconocimiento de símbolos y el análisis estructural.

- La digitalización de los documentos que contienen las expresiones matemáticas se emplearon dispositivos electrónicos como scanner, cámaras de teléfonos móviles y una interfaz de usuario donde se escriben las expresiones matemáticas.

- El proceso de reconocimiento se basa en expresiones matemáticas aisladas.

- El proceso de binarización por elección de umbral utilizando el histograma en escala de grises propuesto en esté estudio aumenta la calidad de los trazos que componen los símbolos que integran las expresiones matemáticas.

- La fase de segmentación de símbolos por etiquetado de píxeles se utiliza cuando los trazos que componen un símbolo están definidos y separados de otros símbolos. Se utiliza técnicas de análisis estructural para la re etiquetación de trazos que perteneces al mismo símbolo.

- Se propone un nuevo algoritmo para la extracción de características nombrado **c-WZ** el cual aumenta el porcentaje de precision en la clasificación de símbolos aislados, en comparación de propuestas realizadas por otros autores.

- El conjunto de datos empleado en este estudio se basa en caracteres del alfabeto latín en mayúsculas y minúsculas, alfabeto griego mayúsculas y minúsculas, dígitos, y 24 símbolos matemáticos.

- Aunque el objetivo no era medir la precision de los algoritmos de clasificación se puede concluir en base a los estudios realizados que el clasificador de máquina de vectores de soporte (SVM) y el perceptron multicapa (MLP) dan la tasa mas alta de precision utilizando el método **c-WZ**.

- La mejora en precision de reconocimiento depende en gran medida de la calidad de las características extraídas de cada uno de los símbolos, estos resultados se utilizan para la fase de análisis estructural y reconstrucción de la expresiones matemáticas.

4.2. Trabajo futuro

En esté trabajo de investigación en el reconocimiento de expresiones matemáticas se baso en la fase de reconocimientos de símbolos ya que es la base para obtener mejores resultados en el análsis estructural y en la reconstrucción de expresiones matemáticas, por esta razón se derivan nuevas direcciones adicionales para reconocimiento de expresiones matemáticas.

- Evaluar el rendimiento de c-WZ cuando se apliquen clasificadores con redes neuronales más robustas.

- Probar estrategias de extracción de características más sofisticadas.

- La aplicación de algoritmos para reducir la dimensión de los vectores de características la cual constituye otra vía que merece cierta atención para futuras investigaciones.

- Aplicaciones de algoritmos de aprendizaje para el análisis estructural y reconstrución de las expresiones matemáticas.

- Complementar la base de datos con todos los símbolos reconocidos por LaTeX.

- Implementar la codificación en código TeX del resultado de la reconstrucción de la expresiones matemáticas.

- Comprobar y verificar métodos para la corrección de errores en la codificación en TeX utilizando el contexto de las expresiones matemáticas escritas a mano.