U

P

T

Reconocimiento del Lenguaje de Señas usando análisis Wavelet y Redes Neuronales Artificiales

Por

Oscar Morales Alvarez

Tesis sometida como requisito parcial para obtener el grado de

MAESTRO EN COMPUTACIÓN ÓPTICA

en la

UNIVERSIDAD POLITÉCNICA DE TULANCINGO

Noviembre 2012
Tulancingo de Bravo, Hidalgo.

Supervisada por:

Dr. José Francisco Solís Villarreal
Dra. Carina Toxqui Quitl

Dedicatoria

A Dios y a mi familia por su gran apoyo.

Agradecimientos

Al Laboratorio de Óptica y Visión por Computadora de la Universidad Politécnica de Tulancingo por las facilidades para llevar a cabo esta Tesis.

A la Universidad Politécnica de Tulancingo por el apoyo otorgado a través de una beca académica de Investigación y Posgrado.

Al Espacio Común de Educación Superior Tecnológica (ECSEST) por el apoyo otorgado a través de una beca del Programa de Movilidad Estudiantil con el proyecto "Reconocimiento del Lenguaje de Señas".

A los Drs. Gerardo Téllez Reyes y César Santiago Tepantlán por todo el apoyo otorgado para la realización de esta Tesis.

A los Drs. Francisco Solís Villarreal y Carina Toxqui Quitl por su invaluable asesoria, apoyo y dirección durante el desarrollo de esta Tesis.

A los Drs. Alfonso Padilla Vívanco, Juan Carlos Valdiviezo Navarro y José Humberto Arroyo Nuñéz por sus valuables sugerencias y aportaciones que ayudaron a mejorar esta Tesis.

Prefacio

El lenguaje de señas es el medio de comunicación más importante para las personas sordomudas, sin embargo, la comunicación oral es la que predomina en el intercambio de información cara a cara entre los individuos. Por lo que, las personas que únicamente pueden comunicarse por medio del lenguaje de señas se ven en la necesidad de requerir un traductor, limitando así sus posibilidades de comunicación efectiva.

Los sistemas de visión por computadora ofrecen herramientas que permiten la posibilidad del reconocimiento parcial del lenguaje de señas, ya sea el reconocimiento de signos de manera estática (imágenes) o de forma automática (video). Un sistema de visión por computadora llevará a cabo los siguientes procesos: a) adquisición y captura de imágenes digitales, b) filtrado y segmentación de imágenes, c) extracción de descriptores y d) clasificación. Los sistemas diseñados para el reconocimiento automático del lenguaje de señas deben de satisfacer además de lo anterior, el tiempo y costo computacional de analizar volúmenes significativos de información.

Esta tesis está orientada al diseño de un sistema de visión por computadora para el reconocimiento del lenguaje de señas americano. Está principalmente enfocada al reconocimiento estático de los dígitos y signos del alfabeto. En el capítulo uno se muestran los antecedentes, objetivos, el estado del arte y las aportaciones. En el capítulo dos se abordan diferentes técnicas de procesamiento digital de imágenes para el análisis, segmentación, mejoramiento y la extracción de descriptores, incluyendo la transformación wavelet para la compresión de datos. El capítulo tres describe el modelo de la red neuronal artificial que es el clasificador principal en la resolución de este problema. Las cuatro bases de datos generadas en este trabajo se especifican dentro del capítulo cuatro. La extracción de descriptores en una y dos dimensiones, sus métodos de clasificación y resultados, además del diseño de un prototipo, se abordan en el capítulo cinco. Las conclusiones generales y el trabajo a futuro se plantean en el capítulo seis.

Resumen

Se propone un método de visión por computadora que permita identificar los signos del lenguaje de señas americano, sin la necesidad de marcadores o dispositivos más allá de un par de cámaras y un equipo de computo, lo que en algún momento facilitaría el uso de la aplicación. Se utilizan tres métodos para la extracción de características, la transformación wavelet Haar así como la correlación y la extracción de pendientes. Esta última, es una propuesta que no ha sido estudiada anteriormente y que convierte la información del contorno de las manos en un vector descriptor de las mismas, al igual que el método de las firmas de correlación, ambos métodos permiten clasificación por correlación y redes neuronales artificiales. Se lograrón resultados del 89.6 % de clasificación para los dígitos y 85.5 % para las letras, ambos casos estáticos, del lenguaje de señas americano. La transformación wavelet Haar fue utilizada para la compresión de la información, buscando la reducción del costo computacional en el momento de analizar volúmenes significativos de información.

Abstract

A computer vision system for the recognition of the american sign language signs is proposed. This system identifies signs without markers, special background or an special electronic device beyond a couple of cameras and a computer equipment, wich at some point facilitate the use of the application. Three methods are used for feature extraction, the Haar wavelet transform in 1D and 2D, correlation and slope extraction. This latter latter is a proposal that has not been studied previously and which converts the contour information of the hands in a descriptor vector of the same, as the method of correlation of signatures, both methods enable correlation and classification by artificial neural networks. Results were achieved over 89.6 % classification for digits and 85.5 % for the letters, both static for the signs of the American Sign Language. The Haar wavelet transformation was used for compression of information, searching the computational cost reduction when analyzing the significant information data volumes.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

El reconocimiento del lenguaje de señas es un área de investigación que ha tenido un crecimiento significativo en los últimos años. El lenguaje de señas no es universal, es decir, cada país o región maneja su propio lenguaje, por ejemplo el lenguaje de señas americano es diferente al chino y al taiwanés. Los lenguajes de señas cambian entre regiones o países debido a la cultura, usos y costumbres usados por las diferentes poblaciones. Analizar las expresiones del lenguaje de señas es una tarea muy compleja debido al hecho de que involucra movimientos de la mano, brazos, el cuerpo y expresiones faciales o gestuales. Las manos por ejemplo, pueden generar una gran cantidad de formas o movimientos debido al número de grados de libertad que ella posee [1].

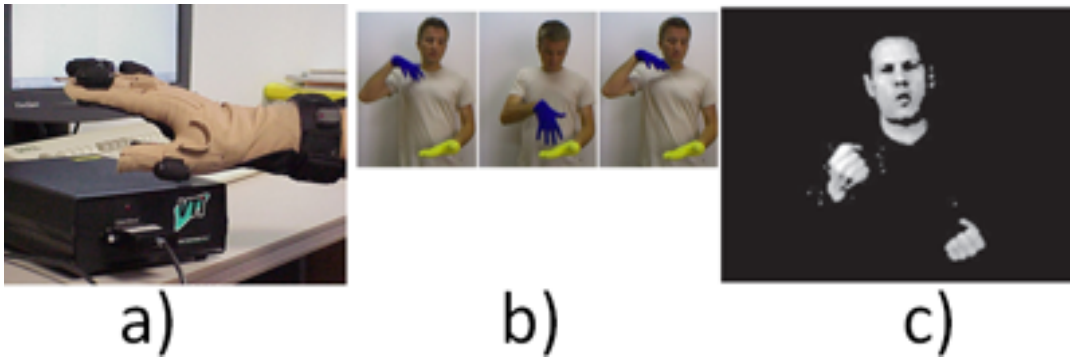

figura 1.1: a) Guante con acelerómetro, b) Marcadores, c) Fondo y ropa especial. Todas estas técnicas, favorecen la segmentación de las manos.

El reconocimiento del lenguaje de señas tiene como objetivo principal traducir el lenguaje de señas a texto, de tal forma que este proceso incite a que la comunicación entre la población con sordera y sin sordera sea conveniente. Existen estrategias para el reconocimiento del lenguaje de señas, uno es con el uso de guantes o dispositivos electrónicos (Figura 1.1) o marcadores, por ejemplo el color de la ropa del interlocutor o intérprete generalmente debe ser oscura para que contraste con las manos (marrón, azul, verde, negro, burdeos, etc.), lisa y sin estampados ni dibujos. No debe llevar puestas joyas (anillos, collares, pulseras, pendientes

llamativos), pañuelos, ni tarjetas identificativas que puedan producir reflejos, lo que restringe en mucho el uso de estos sistemas. En el caso del uso de dispositivos electrónicos como guantes con acelerómetros, la persona sordomuda está atada a este sistema debido a la gran cantidad de cables que estos sistemas generalmente requieren y a la poca movilidad que ofrecen, pues una persona que domina el lenguaje de señas sabe expresarse adecuadamente y requiere de la libertad de movimientos que este tipo de métodos no ofrecen [1].

Por otro lado existe la estrategia de utilizar sistemas de visión para el reconocimiento del lenguaje de señas [7], que finalmente es más complejo por el ángulo de visión y la posición que requieren las manos, los brazos, el cuerpo y la cara para la interpretación, resulta más comodo para el interlocutor expresarse por este medio. En este trabajo se propone un sistema de visón sin el uso de dispositivo alguno (guantes, ropa especial, iluminación, entre otros factores) que permita la obtención de información suficiente para realizar una traducción del signo al símbolo correspondiente de los números o del alfabeto del lenguaje de señas. En este sistema, el interlocutor realiza las letras en el aire frente a un sistema de cámara(s) que permite la segmentación de la información por medio de la visión infrarroja.

1.1. Objetivos

El objetivo principal de esta tesis es el desarrollo de un sistema de visión por computadora para el reconocimiento del lenguaje de señas americano. Para ello se llevaron a cabo las siguientes tareas:

1. *a*) La implementación de dos sistemas para la adquisición de imágenes del Lenguaje de Señas Americano, en el espectro visible e infrarrojo.

   b) Generación de cuatro bases de datos, una de digitos y tres del alfabeto del Lenguaje de Señas Americano estático.

   c) La búsqueda e implementación de algoritmos de preprocesamiento para la segmentación, mejoramiento y compresión de la información en imágenes dígitales.

   d) Implementación de algoritmos basados en análisis de pendientes, correlación, transformación wavelet y redes neuronales artificiales para la extracción de características de signos del Lenguaje de Señas Americano estático.

   e) Desarrollo en MATLAB de una interfaz para el reconocimiento dinámico de los dígitos.

1.2. Justificación

El lenguaje de señas es el mejor medio de comunicación para una persona incapacitada del oído, es adquirir una forma de expresar sus necesidades, pensamientos y comprender las

expresiones de los demás. No todas las personas con las que convive una persona con tal incapacidad conocen el lenguaje a señas, imposibilitando una comunicación completa. Se propone un sistema de visión por computadora, que realice el reconocimiento del lenguaje a través del procesamiento de grandes volúmenes de información, sin necesidad de ningún dispositivo físico como ropa especial o dispositivos electrónicos, que impidan el movimiento libre del interlocutor. El reconocimiento por métodos ópticos de visión, es mayormente favorable al de la utilización de dispositivos físicos; la utilización del menor número de dispositivos complejos facilita el uso de un sistema. El reconocimiento de patrones en la óptica computacional es un campo muy amplio abierto a la investigación y mediante la utilización de sistemas inteligentes es posible automatizar esta tarea.

1.3. Antecedentes

Aún cuando hoy en día las lenguas de señas se utilizan casi exclusivamente entre personas con sordera, su origen es tan antiguo como el de las lenguas orales o incluso más en la historia de la humanidad; también han sido y siguen siendo empleadas por comunidades de oyentes. Aunque existen referencias a culturas antiguas que utilizaban este tipo de lenguaje, no existen referencias documentales sobre estas lenguas antes del siglo XVII [3]. Los datos que se poseen tratan sobre los sistemas y métodos educativos para personas sordas. En el año 1620 Juan de Pablo Bonet publica su *Reducción de las letras y Arte* para enseñar a hablar a los mudos, considerado como el primer tratado moderno de Fonética y Logopedia, en el que se proponía un método de enseñanza oral de los sordos mediante el uso de señas alfabéticas configuradas unimanualmente; esto permitió divulgar así en toda Europa, y después en todo el mundo, el alfabeto manual, útil para mejorar la comunicación de los sordos y mudos. En 1817 Gallaudet fundó la primera escuela de la nación para las personas sordas, en los Estados Unidos de Norteamérica se convirtió en el primer maestro sordo de lengua de señas de los Estados Unidos [2] [3] [4].

Actualmente existe una variedad muy extensa de sistemas de comunicación de personas sordomudas, como la de configuraciones de la mano donde cada letra del alfabeto corresponde con una forma de la mano. Otro es el trazado de letras en el que se transmite el mensaje representando cada letra del alfabeto por su correspondiente grafía de la escritura con el dedo sobre una superficie; además, el sistema dactilológico que consiste en deletrear el mensaje apoyando cada una de las letras sobre la palma de la mano, (este más usado por personas sordociegas), sin embargo, el más conocido y utilizado es el sistema de lenguaje de señas tradicional, que aunque varía de región en región, todos convergen en un gran número de signos como los números y la letras [2] [5] [6].

1.4. Estado del arte

Existen numerosos artículos, reseñas y revistas para la solución de la comunicación y traducción entre personas que utilizan el lenguaje de señas y personas que no lo conocen.

Este reto de reconocimiento y clasificación siempre es interdisciplinario, pues no existe una rama especializada para resolver este problema. Los artículos más comúnes utilizan ropa de colores especiales para la segmentación de las manos, y/o también, un fondo especial que facilite la extracción de la mano de la imagen y un preprocesado por medio de filtros para la obtención de sus descriptores [7] [8] [9] [10]. Después de la segmentación del objeto en cuestión, estos métodos suelen utilizar redes neuronales artificiales para la clasificación de los signos adquiridos; en el caso de signos dinámicos, es decir, que lleven movimiento, como es el caso de la letra 'J' en el alfabeto del lenguaje de señas americano, utilizan el modelo oculto de Markov (Figura 1.2) que es un modelo estadístico en el que se asume que el sistema a modelar es un proceso de parámetros desconocidos, cuyo objetivo es el de determinar estos mismos (u ocultos, de ahí el nombre) de una cadena de datos a partir de los parámetros observables. [4], [8]

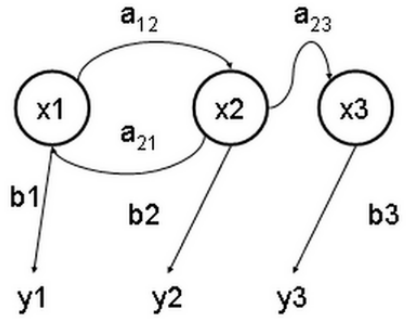

figura 1.2: Ejemplo de transición de estados en un modelo oculto de Markov donde: x representa los estados ocultos, y las salidas observables, a las probabilidades de transición y b las probabilidades de salida.

Existen otros métodos que utilizan sistemas de cámaras mucho más complejos [7], que buscan por medio de la fusión de imágenes o la adquisición de datos independientes de cada una de estas, la extracción y clasificación de información. Mientras más elevado es el número de dispositivos empleados, se elevan los resultados, pero aún así la resolución total del problema no es del 100 % [7]. Aunque la mayoría de los proyectos que intentan el reconocimiento del lenguaje de señas utilizan redes neuronales artificiales como clasificador, los métodos de correlación, lógica difusa y estadísticos suelen ser frecuentes también en este tipo de investigaciones, además del filtrado y segmentación de las imágenes [5] [6] [10] [11]. Las bases de datos de imágenes del lenguaje de señas suelen ser delimitadas, pues una base de datos con la que pudiera ser comparada alguna palabra es tan extensa como extensa es la cantidad de palabras de un idioma.

Dentro de los dígitos numéricos y alfabéticos del Lenguaje de Señas (Figura 1.3) cabe mencionar que algunos símbolos tales como la "J ", "Q " y la "Z " requieren de movimiento, por lo tanto no puedieron ser registrados en su totalidad en una sola imagen, por lo que se omitió su captura, aunque se espera que una vez dominado el caso estático, este tipo de dígitos con movimiento sean capturados y procesados en conjunto con los demás.

figura 1.3: Alfabeto y números del Lenguaje de Señas Americano

1.5. Aportaciones

En este proyecto de tésis se tienen las siguientes aportaciones:

1. *a*) Un sistema de visión por computadora para la adquisición de imágenes sin las siguientes restricciones: 1) Dispositivos físicos, como ropa y fondos especiales, marcadores y/o guantes; así como 2) dispositivos electrónicos como guantes con acelerómetros, cables, entre otros.

   b) Cuatro bases de datos de dígitos y signos del Lenguaje de Señas Americano estático.

   c) Un nuevo método, basado en el análisis de pendientes para la descripción de objetos en una imagen. Este, aunado con la transformación wavelet para la compresión de información y redes neuronales para la clasificación, se logró un 99.9 % de reconocimiento de los dígitos.

   d) Un nuevo método, basado en algoritmos de correlación, transformación wavelet , redes neuronales artificiales, cádenas de código y convolución para el reconocimiento de los signos del alfabeto del Lenguaje de Señas Americano. Con éste se logró una clasificación correcta del 85.5 %

   e) Una interfaz para el control de la cámara IR, la adquisición y el preprocesado de imágenes, así como el reconocimiento dinámico de dígitos.

Bibliografía

[1] M. Gonzáles "Lenguaje de signos". Confederación Nacional de Sordos de España, ISBN: 84-604-2047-7, *Fundación ONCE* (1992)

[2] P. Gómez, E. Romero, "La sordoceguera Un análisis multidisciplinar", ISBN: 84-484-0142-5, *Organización Nacional de Ciegos Españoles ONCE* (2004)

[3] A. Gascón "Historia de la lengua de signos", Memorias de la conferencia "La problemática de las personas sordas", Universidad Complutense de Madrid (2004)

[4] M.I.T, "Media Laboratory Perceptual Computing Section Technical Report No. 466", IEEE PAMI '98, M.I.T. Press (1996).

[5] O. Al-Jarrah, A. Halawani, Recognition of gestures in Arabic sign language using neuro-fuzzy systems". Artificial Intelligence Vol. 133 Elsevier (2001)

[6] D. Rozado, B. Rodriguez, P. Varona ". Extending the bioinspired hierarchical temporal memory paradigm for sign language recognition"Neurocomputing, Vol. 79, Elsevier (2004).

[7] Q. Wang, X. Chen, L.-G. Zhang, C. Wang, W. Gao, "Viewpoint invariant sign language recognition". Computer Vision and Image Understanding, Vol. 108, Science Direct (2007)

[8] F. Jiang, W. Gao, H. Yao, D. Zhao, X. Chen "Synthetic data generation technique in Signer-independent sign language recognition", Pattern Recognition Letters, Vol. 30 Elsevier (2009)

[9] H. Brashear,V. Henderson, K.-H. Park, H. Hamilton, S. Lee, T. "Starner American Sign Language Recognition in Game Development for Deaf Children"Pattern Recognition (2006)

[10] O. Arana, T.Burgerb, A.Caplierc, L. Akaruna ". A belief-based sequential fusion approach for fusing manual signs and non-manual signals"Pattern Recognition Vol. 42 Elsevier (2009)

[11] Q. Munib, M. Habeeb, B. Takruri, H. A. Al-Malik, ". American sign language (ASL) recognition based on Hough transform and neural networks". Expert Systems with Applications, Vol. 32 Science Direct (2007)

Capítulo 2

Procesamiento Digital de Imágenes

2.1. Introducción

En este capítulo se estudiarán los métodos analíticos para el análisis y procesamiento de señales e imágenes. En las secciones tres y cuatro se expone el proceso de adquisición de una imagen digital a través del muestreo y cuantización de la imagen continua. La delimitación de regiones y los contornos de objetos en la imagen y la obtención de sus cadenas de código se describen en la sección cuatro, en la sección cinco se describen los contornos usando una relación de pendientes, en la sección seis se analizan algoritmos basados en correlación y convolución 1D y 2D de señales e imágenes para medir la similitud entre dos funciones, el mejoramiento, la eliminación de ruido y búsqueda de objetos en la imagen. Un análisis multiresolución basado en transformada Wavelet Haar para la compresión de información es descrito en la sección 7. Finalmente, en la última sección se escriben las conclusiones.

2.2. Imágenes de Radiometría

El arreglo de sensores en una CCD (charge-coupled device) empieza con el sensado de la energía electromagnética proveniente de una fuente como el sol, este proceso se ejemplifica en la Figura 2.1. La radiometría es el estudio de la detección y medición de la radiación electromagnética. En un sistema formador de imágenes óptico digital (como es el caso de una cámara térmica), la radiometría describe la intensidad de energía en la señal que la cámara "sensa"para producir una imagen final. El brillo y contraste en la imagen final, no pueden ser calculados exactamente sin un modelado de radiometría apropiado para el sistema [1].

2.2.1. Ondas electromagnéticas

La energía que es capturada por la cámara es una forma de radiación electromagnética, una onda autopropagante, (es decir, que viaja la energía y no la materia) compuestas de las oscilaciónes de campos eléctricos y magnéticos generados por la aceleración de particulas. La

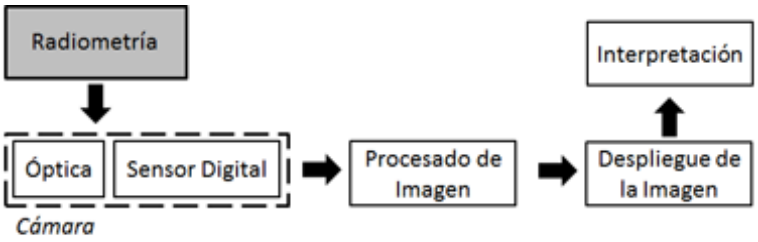

figura 2.1: Modelo de radiometría que describe la luz que entra por un sistema óptico digital.

longitud de onda λ multiplicada por la frecuencia de la onda f, medida en ciclos por segundo o Hertz (Hz), obtiene la velocidad a la cual la onda se propaga. Para ondas electromagnéticas en el vacío, la relación entre longitud de onda y la frecuencia esta dada por,[1]

$$c = \lambda f, \tag{2.1}$$

donde $c = 2{,}9979\text{x}10^8$ m/s, es la velocidad de las ondas electromagnéticas en el vacío. Esto aplica para todas las ondas, sin importar la longitud de onda. Pero si la onda entra en un medio que no es el vacío, la velocidad decrementa y la longitud de onda se incrementa.

La radiación electromagnética también tiene las propiedades de particulas, o fotones, donde cada fotón tiene la energía dada por,[1]

$$E_{fotón} = hf = \frac{hc}{\lambda}, \tag{2.2}$$

donde $h = 6{,}6261\text{x}10^{-34}$ J·s (segundos · joule) es la constante de Planck. Note que la energía del fotón es proporcional a la frecuencia de la onda e inversamente proporcional a la longitud de onda. La radiación electromagnética se clasifican deacuerdo a la longitud de onda como rayos gamma, rayos x, UV (ultravioleta), visible, IR (infrarrojo), microondas u ondas de radio como se muestra en la Figura 2.2. Las ondas de longitud de onda λ más cortas contienen mucho más energía por fotón a comparación de las ondas cuya longitud de onda λ es más larga, por tanto son más peligrosas para los humanos. Centraremos nuestra atención a las cámaras desarrolladas para obtener imágenes en el espectro visible ($0{,}4 - 0{,}8\mu$m) del espectro electromagnético.

Una onda electromagnética propagándose en el tiempo, puede ser descrita matemáticamente por la función coseno,[1]

$$E(x,t) = A\cos\left[2\pi\frac{x-ct}{\lambda} - \varphi\right] = A\cos\left[2\pi\left(\frac{x}{\lambda} - ft\right) - \varphi\right], \tag{2.3}$$

donde A es la amplitud, λ es la longitud de onda, f es la frecuencia y φ es la fase. En el contexto de las ondas del espectro visible, la amplitud determina el brillo y la frecuencia determina el color, esto es mucho más conveniente para representar matemáticamente una onda propagandose por,

$$E(x,t) = Ae^{2\pi\left(\frac{x}{\lambda} - vt\right) - \varphi} = Ae^{i(kx-\omega t)-\varphi}, \tag{2.4}$$

donde $k = 2\pi/\lambda$ y $\omega = 2\pi f$. Ésta función es referenciada a las ondas coseno y seno dadas por la ecuación de Euler.

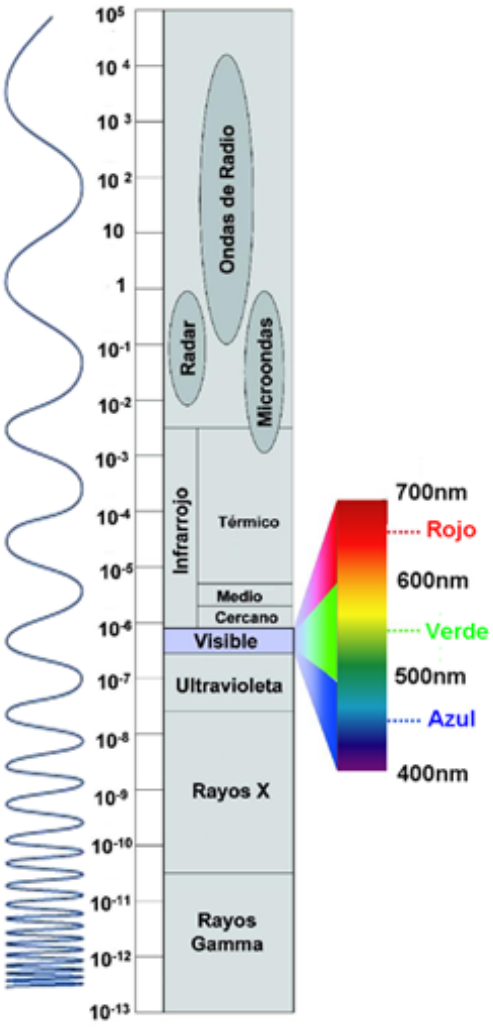

figura 2.2: Espectro electromagnético.

2.2.2. Radiación de cuerpos negros

El modelado de la energía de la radiación electromagnética proveniente de una fuente emisora, puede ser simplificado por el modelado para cuerpos negros (M_{BB}). Los cuerpos negros son objetos ideales que perfectamente absorben y re-irradian toda la radiación electromagnética basada en la temperatura del objeto. En 1900, Max Planck derivó una expresión para la radiación espectral excitante basado en la cuantización de los estados energéticos por la vibración de los electrones en las moléculas, esta expresión está dada por,

$$M_{BB}(\lambda, T) = \frac{2\pi hc^2}{\lambda^5} \frac{1}{e^{\frac{hc}{\lambda kBT}} - 1} \left(\frac{W}{\mathrm{m}^2 \cdot \mathrm{m}} \right), \tag{2.5}$$

donde $kB = 1{,}3807\text{x}10^{-23} J/K$ es la constante de Boltzmann, $h = 6{,}6256\text{x}10^{-34} J/s$ es la constante de Planck y T es la temperatura del cuerpo negro en grados Kelvin (K). Para los sistemas de formación de imágenes en el espectro visible, las unidades son típicamente escogidas en micras (μm) en el denominador, ésto debido a que λ es típicamente medida en micras para fuentes de luz visible, dando como resultado,

$$M_{BB}(\lambda, T) = \frac{3{,}7414x10^8}{\lambda^5} \frac{1}{e^{\frac{14386}{\lambda T}} - 1} = \left(\frac{W}{\mathrm{m}^2 \cdot \mu\mathrm{m}} \right), \tag{2.6}$$

para λ en μm y T en grados Kelvin. La radiación excitatante (radiación por unidad de área del espectro electromagnético) describe la fuerza o flujo [energía por unidad de tiempo, medida en watts (W)] emitida desde una superficie por la longitud de onda de la luz. El total de exitación que proviene de un cuerpo negro (M_{BB}) puede ser calculado por la integración de la ecuación de Planck sobre todas las longitudes de onda, dando la ecuación Stefan-Boltzmann:

$$M_{BB}(T) = \int_0^\infty \frac{2\pi hc^2}{\lambda^5} \frac{1}{e^{\frac{hc}{\lambda kBT}} - 1} d\lambda = \frac{2\pi^5 k^4}{15c^2h^3} T^4 = \sigma T^4 \left(\frac{W}{\mathrm{m}^2} \right), \tag{2.7}$$

donde σ es la constante de Stefan-Boltzman $=5{,}6704\text{x}10^{-8} W/\mathrm{m}^2/K^4$. El sol puede ser aproximado como un cuerpo negro a una temperatura de 6000° K, una aproximación a la temperatura del sol y 300° K, la temperatura aproximada de la tierra. La curva de la Figura 2.3 ilustra el desplazamiento del espectro electromagnético a una curva de longitud de onda λ más larga para bajas temperaturas. La longitud de onda donde el pico de la excitación espectral ocurre, puede ser calculado por el desplazamiento obtenido por la ley de Wien,

$$\lambda_{\text{máx}} = \frac{2898}{T} (\mu m). \tag{2.8}$$

Para el modelo de cuerpo negro del sol, el pico está en $\lambda_{\text{máx}} = 0.50\mu$m, y para la tierra $\lambda_{\text{máx}} = 9.7\mu$m. Notese que el pico de la exitación espectral para la tierra (y el cuerpo humano) esta en el espectro infrarrojo, mientras que el pico del sol se encuentra en el espectro visible. Así que no es una casualidad que la sensitividad del ojo humano este centrada muy cerca

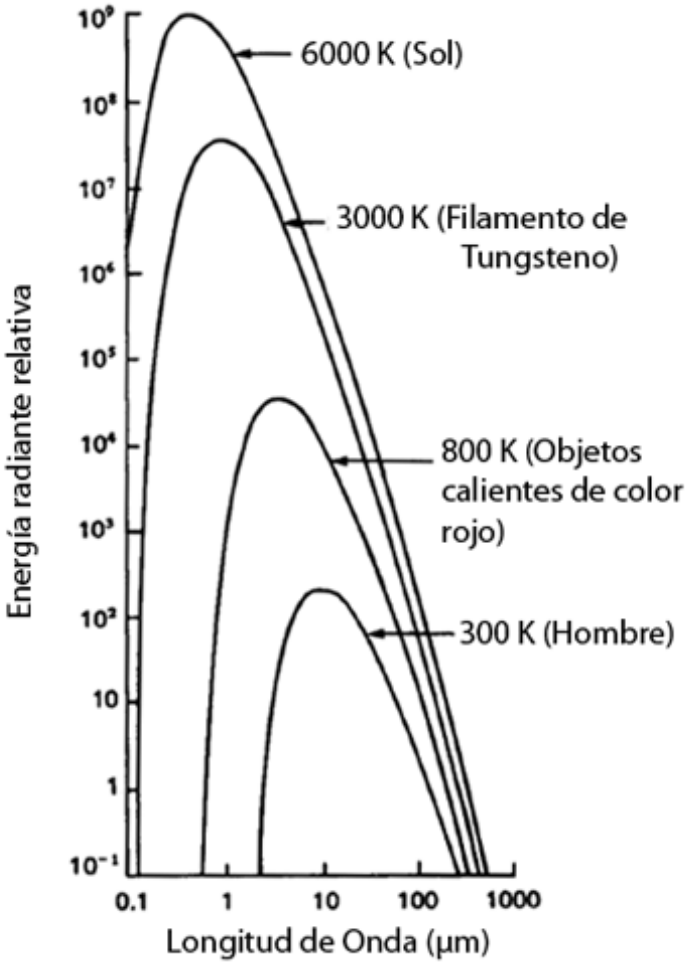

figura 2.3: Curva de emisión de radicación de cuerpos negros para diferentes objetos.

del pico de excitación espectral del sol, que es aproximadamente tres millones de veces más grande que el pico de exitación espectral de la tierra.

En la vida real, no hay cuerpos negros ideales, pero en cambio, los objetos emiten radiaciones más pequeñas que las del modelo para los cuerpos negros. La emisividad (proporción de radiación térmica emitida por una superficie u objeto debida a una diferencia de temperatura determinada) representada por la letra ε de un objeto, se puede comparar a la radiación del espectro electromagnético con base al modelo de cuerpos negros, y es dado por

$$\varepsilon(\lambda) = \frac{M(\lambda, T)}{M_{BB}(\lambda, T)}. \tag{2.9}$$

Cuando la emisividad no depende de la longitud de onda, el objeto es llamado cuerpo gris. La emisividad de un objeto debe ser conocida si la temperatura del objeto puede ser

calculada de la medida de excitación del objeto.

El modelo de cuerpos negros para el cálculo de la excitancia espectral del sol es una buena aproximación para la medición de los datos. Los bulbos de luz incandecente generan radiación por medio del calor (Filamentos de Tungsteno en la Figura 2.3), los filamentos de alta resistencia, ralentizan los electrones para crear luz, así que el modelo de cuerpos negros, usualmente es una buena aproximación para las temperaturas típicas del modelo cercano a los 3000° Kelvin [1].

El cálculo de la radiación espectral, a partir de una fuente puntual, puede ser simplificado si la superficie de un objeto se asume como una fuente lambertiana: es decir, la radiancia aparente del objeto es la misma sin importar el ángulo de visión, como se muestra en la Figura 2.4.

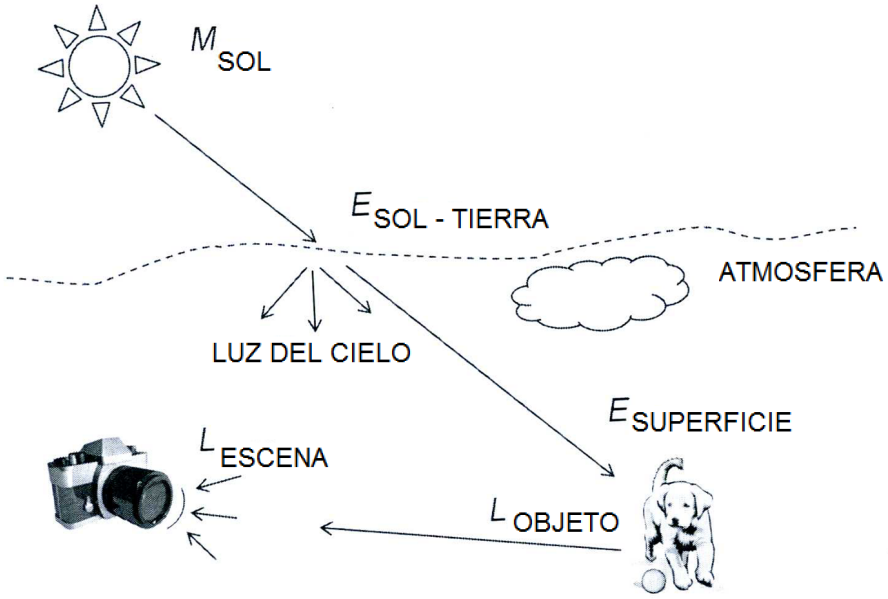

figura 2.4: Esquema de una escena Lambertiana.

En el esquema de una escena Lambertiana mostrado en la Figura 2.4, la luz del sol es dispersada por la atmósfera, esta misma luz incide sobre un objeto y es reflejada hacia el sistema óptico de una cámara. La luz que refleja el objeto y la radiancia aparente del objeto es la misma sin importar el ángulo de visión..

2.3. Muestreo y cuantización

El muestreo es el proceso de convertir una señal (por ejemplo, una función continua en el espacio) en una secuencia numérica (una función discreta en el tiempo o en el espacio), como se muestra en la Figura 2.5.

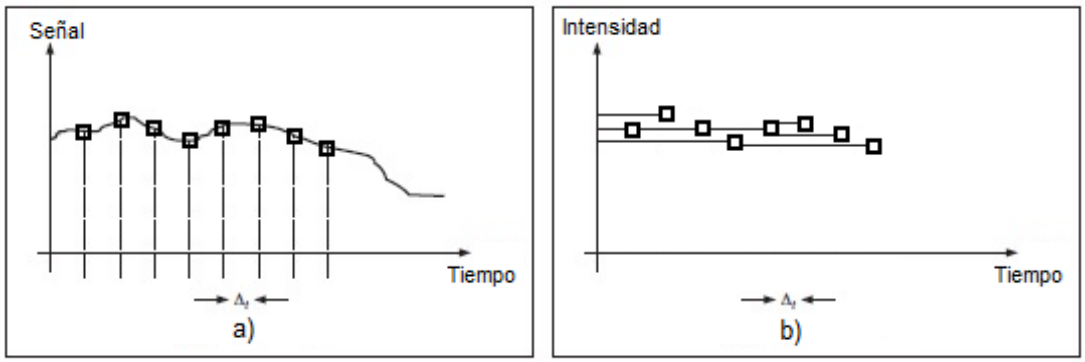

figura 2.5: Digitalización de una señal, a) Muestreo y b) cuantización.

El teorema de muestreo o criterio de Nyquist [2], señala que la reconstrucción (aproximadamente) exacta de una señal continua en el tiempo en banda base a partir de sus muestras es posible si la señal es limitada en banda y la frecuencia de muestreo $fmuestra$ es mayor a dos veces el ancho de banda de la señal $fmax$, tal como se muestra en la Figura 2.6.

El proceso de muestreo sobre una señal continua que varía en el tiempo (o en el espacio como en una imagen u otra variable independiente en cualquier otra aplicación) es realizado midiendo simplemente los valores de la señal continua cada t unidades de tiempo (o espacio), llamado intervalo de muestreo. El resultado de este proceso es una imágen digital, compuesta por una secuencia de números, llamadas muestras y son una representación de la imagen original.

Normalmente, el proceso de adquisición de la imagen se realiza usando una matriz de sensores. El número de sensores dentro la matriz establece los límites del muestreo en ambas direcciones. La digitalización de la amplitud o cuantización la realiza cada sensor asignando un valor discreto a ciertos intervalos de amplitudes continuas. Como se muestra en la Figura 2.7. La Figura 2.8 es un ejemplo de este proceso, claramente se ve que la calidad de la imagen está determinada en gran medida por el número de muestras (resolución de la matriz) y los niveles discretos de gris usados durante el muestreo y cuantización respectivamente.

2.4. Imagen Digital

Una imagen puede ser definida como una función de dos dimensiones $f(x, y)$ donde x y y son las coordenadas espaciales (plano) y la amplitud de la función f en algún par de

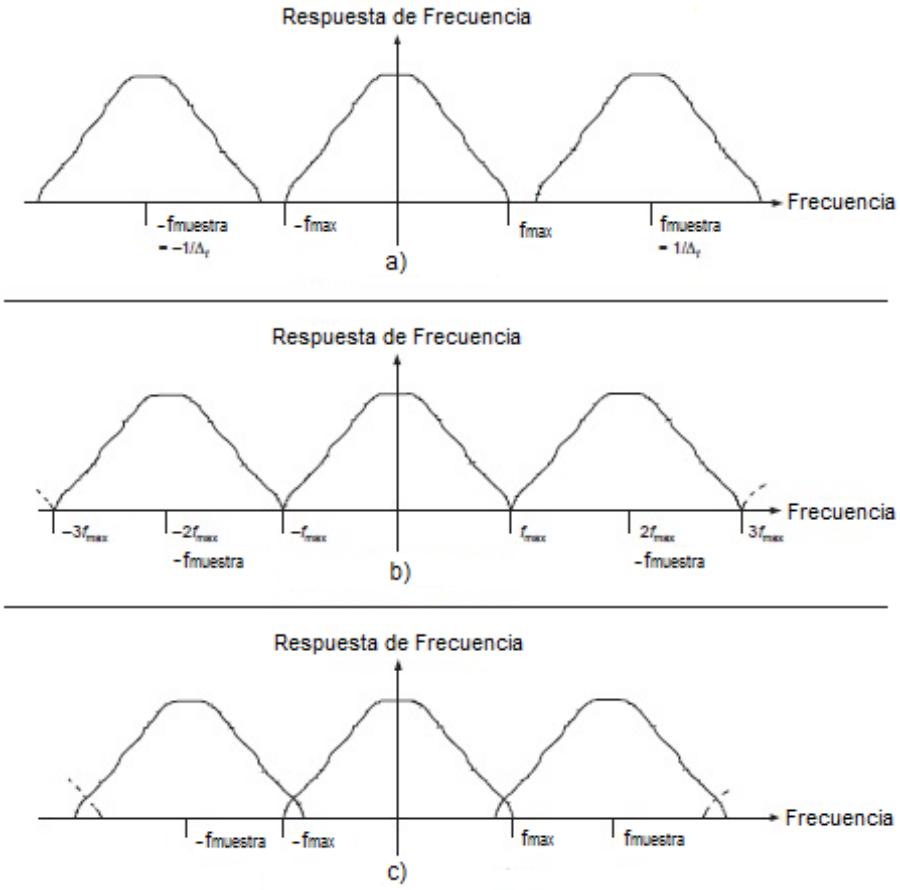

figura 2.6: Espectro a) Sobremuestreado, b) Muestreado por el teorema de Nyquist y c) Submuestreado.

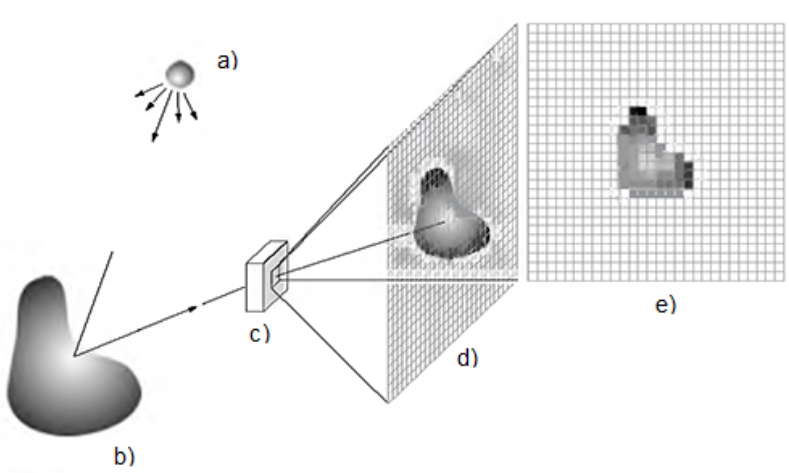

figura 2.7: Un ejemplo del proceso de adquisición de imágenes. (a)Fuente de iluminación ("Energía") (b) Elemento de una imagen (c) Sistema de adquisición de imágenes (d) Proyección de la escena en sobre el plano imagen (e) Imagen digitalizada

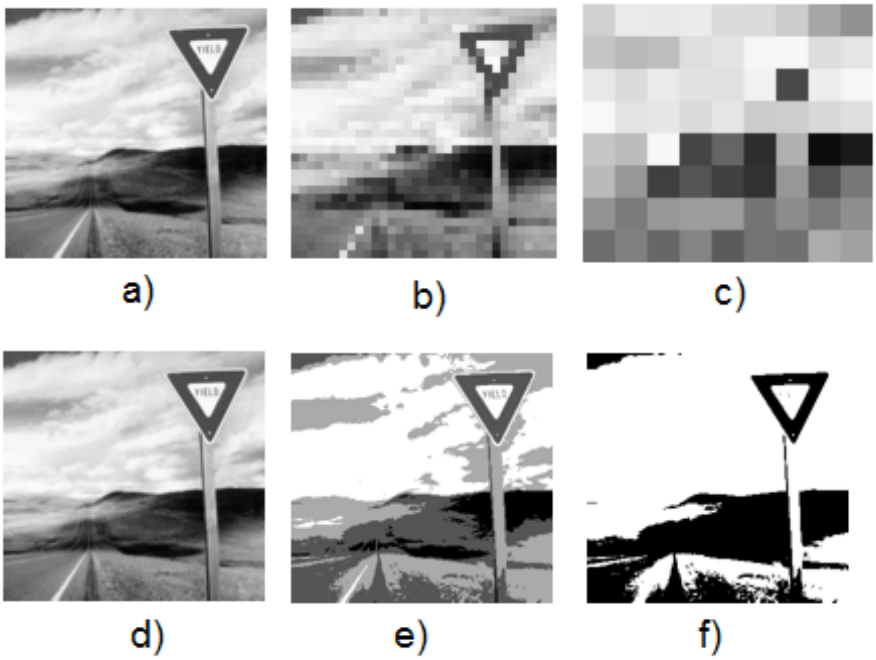

figura 2.8: Ejemplo de cambios en la resolución espacial. Sobremuestreo de una imagen a a) 512 x 512, b) 128 x 128, c) 32 x 32 pixeles a partir de una imagen de 1024 x 1024 pixeles. Ejemplo de cuantización usando d) 256, e) 64 y f) 2 niveles de intensidad.

coordenadas (x, y) es llamada intensidad o nivel de gris de la imagen en ese punto como se muestra en la Figura 2.9 b). Cuando x, y y los valores de la amplitud de la función f, son cantidades discretas finitas, a dicha imagen se le llama imagen digital [3].

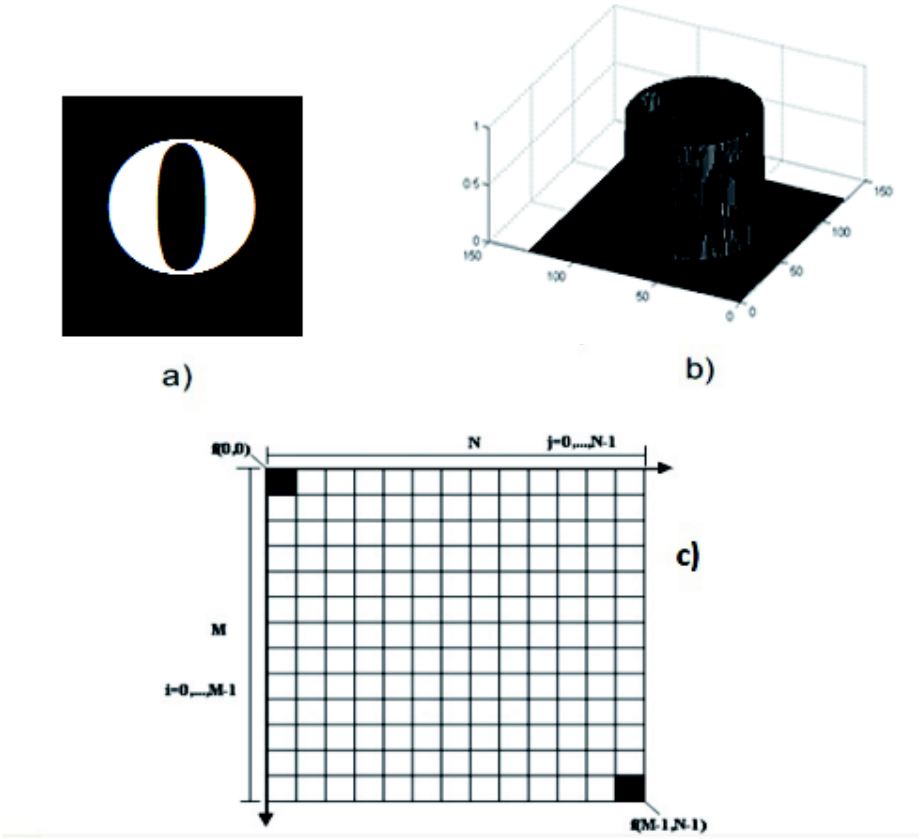

figura 2.9: Imagen digital a) como mapa de intensidades, b) como función bidimensional y c) como matriz de datos

Una imagen digital está compuesta de un número finito de elementos y cada uno tiene una posición y valor particulares. A estos elementos se les llama elementos de la imagen o píxeles; siendo este último el término más usado para denotar los elementos de una imagen digital.

Las imágenes digitales a color consisten regularmente de una composición de tres imágenes separadas en escala de grises, cada una representa los tres colores primarios; rojo, verde y azul; comúnmente conocido como RGB. Otros espacios de color descritos en el Apéndice A, que usan 3 vectores para colores son HSV (Tono, Saturación, Valor de Intensidad). Algunos espacios de color como el CMYK (Cyan, Magenta, Yellow, blacK) son usados para procesos de impresión.

2.5. Regiones y Fronteras

En las imágenes aparecen ciertas áreas o zonas caracterizadas por el hecho de que constituyen agrupaciones de píxeles conectados entre sí, pero además de la conexión, dichos píxeles presentan propiedades o características comunes llamadas regiones [2].

En esta sección, se estudian los descriptores de límite para definir bordes o contornos de una región y cadenas de código para obtener las coordenadas de los píxeles que forman un contorno.

2.5.1. Vecinos de un píxel

Descriptores de límite

Una región usualmente describe el contenido (o los puntos interiores) que son rodeados por un límite (o perímetro) el cual es llamado contorno de región. La forma del contorno es generalmente referida a su silueta. Un punto (o píxel) puede ser arbitrariamente elegido como límite si es parte de la región y hay al menos un píxel en su vecindario que no es parte de la región.

El límite por sí mismo es usualmente encontrado por el seguimiento del contorno: primero se encuentra un punto en el contorno y posteriormente avanza alrededor del contorno ya sea en dirección de las manecillas del reloj o en sentido contrario, encontrando así el punto del contorno más cercano.[2]

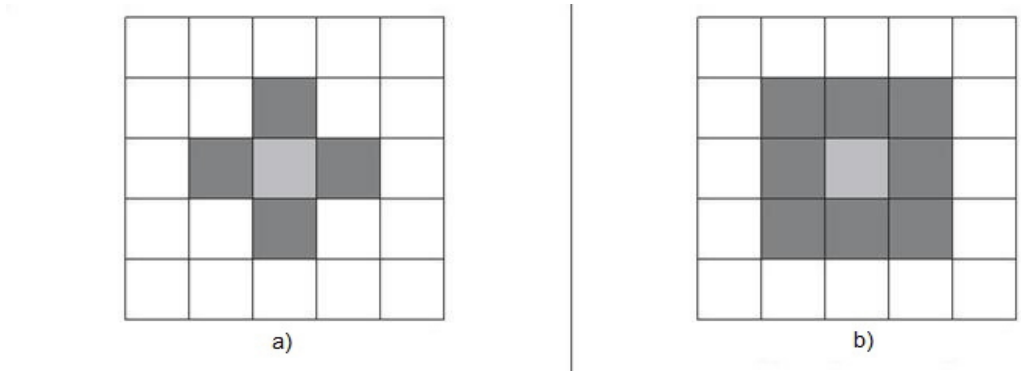

figura 2.10: Principales tipos de conectividad. a) Conectividad 4 b) Conectividad 8.

Con el fin de definir los puntos del interior de la región y los puntos del límite de la misma, se consideran las relaciones de vecindad entre píxeles. Estas relaciones son descritas por reglas de conectividad.[3] Las dos reglas para imágenes binarias más comunes son:

1. *a*) Vecindad de 4 caminos (o de vecindad 4) cuyas coordenadas son:
   $(x+1, y), (x-1, y), (x, y+1), (x, y-1)$

b) Vecindad de 8 caminos (o de vecindad 8) cuyas coordenadas son:
$(x+1, y), (x-1, y), (x, y+1), (x, y-1), (x+1, y+1), (x-1, y-1), (x-1, y+1), (x+1, y-1)$

Los tipos de conectividad mostrados en la Figura 2.10 son:

1. *a*) Conectividad 4

   Dos píxeles p y q tienen conectividad 4 si y solo si q es un vecino 4 de p y tienen valores en V donde $V = \{1\}$.

   $(p, q)_4 \Leftrightarrow q \in N_4(p) \wedge f(p), f(q) \in V$

   b) Conectividad 8

   Dos píxeles p y q tienen conectividad 8 si y solo si q es un vecino 8 de p y tienen valores en V donde $V = \{1\}$.

   $(p, q)_8 \Leftrightarrow q \in N_8(p) \wedge f(p), f(q) \in V$

Un límite y una región pueden definirse usando ambos tipos de conectividad, cabe mencionar que ambos son siempre complementarios. Esto es, que si los píxeles del límite son conectividad 4, entonces los píxeles de la región están conectados por conectividad 8 y viceversa. Esta relación puede verse en el ejemplo mostrado en la Figura 2.11. En el ejemplo, el límite es mostrado en gris obscuro y la región en gris claro.

Se puede observar claramente que para un límite diagonal, la conectividad 4 da un límite escalera, mientras que la conectividad 8 da una línea diagonal formada por los puntos en las esquinas del vecindario. Tenga en cuenta que todos los píxeles que forman la región en la Figura 2.11(b) tienen conectividad 8, mientras que los píxeles de la Figura 2.11(c) tiene cuatro vías de conectividad. Esto es complementario a los píxeles en la frontera.

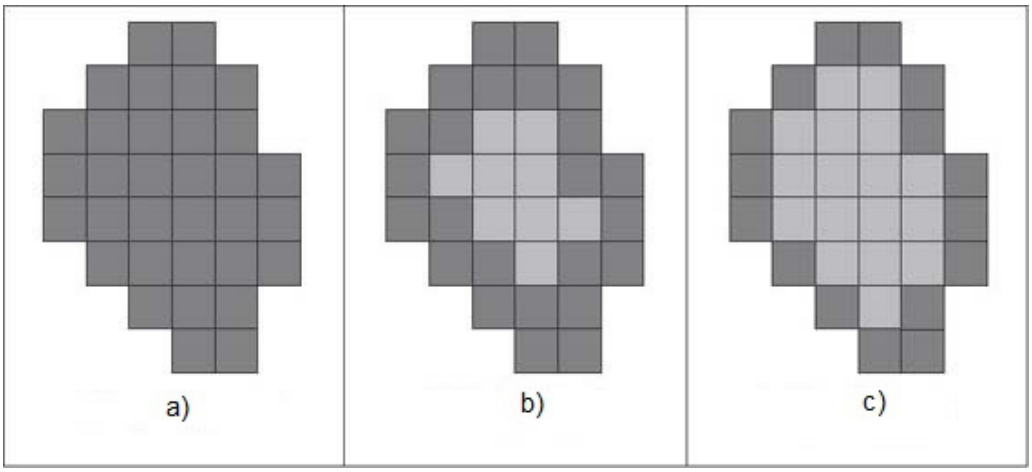

figura 2.11: Límites y regiones. a) Región Original, b) Limite y región de conectividad 4 y c) Limite y región de conectividad 8.

2.5.2. Cadena de código

Con el fin de obtener la representación de un contorno, se puede simplemente almacenar las coordenadas de una secuencia de píxeles en la imagen. Por otra parte, sólo se puede almacenar la posición relativa entre píxeles consecutivos. Esta es la idea básica detrás de la cadena de código [2].

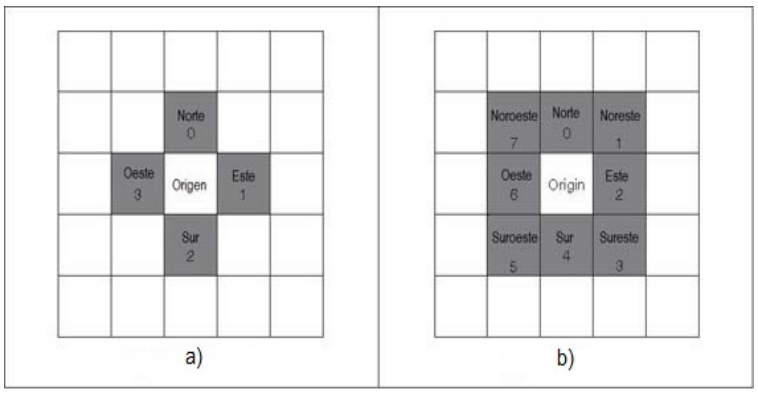

figura 2.12: Conectividad en la cadena de códigos. a) Conectividad 4 b) Conectividad 8.

La cadena de código es de hecho una de las técnicas más antiguas en la visión por computadora, originalmente introducida en los alrededores de 1960 [2]. Esencialmente, el conjunto de píxeles del límite de una silueta es traducido a un conjunto de conexiones entre ellos. Dado un borde completo, es decir un conjunto de puntos conectados, comenzando desde un píxel, se debe ser capaz de determinar la dirección en la cual se encuentra el siguiente píxel.

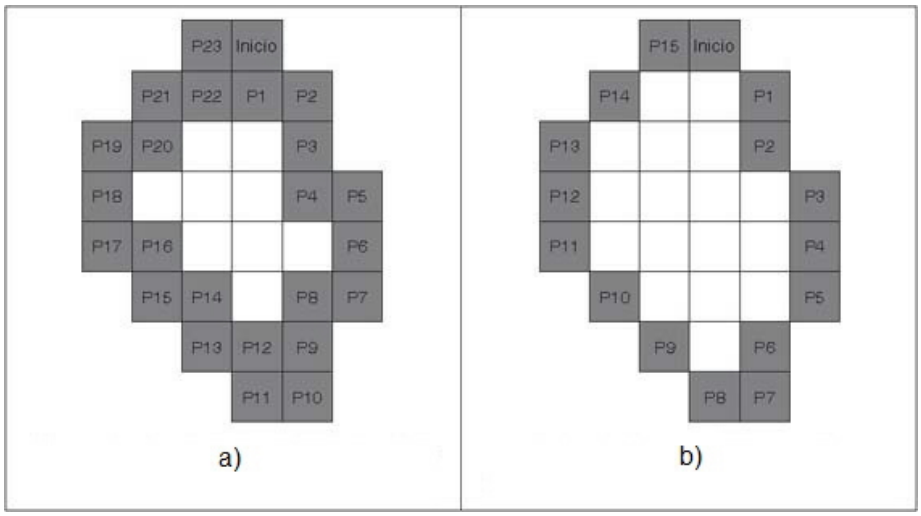

figura 2.13: Cadenas de código por diferentes métodos de conectividad. a) Cadena de código para la conectividad 4 y b) Cadena de código para la conectividad 8. Ejemplo de conectividad 4 código={2,1,2,2,1,2,2,3,2,2,3,0,3,0,3,0,3,0,0,1,0,1,0,1},

A saber, el siguiente píxel es uno de los puntos adyacentes en una de las direcciones de la conectividad. Por lo tanto, la cadena de código es formada por las concatenaciones de los números que designan la dirección del siguiente píxel. Esto es, dado un píxel, la dirección sucesiva de un píxel al próximo se convierte en un elemento de la cadena de código. Esto es repetido para cada punto hasta que el punto inicial es alcanzado, que es precisamente cuando la figura de objeto es completada y cerrada.

La conectividad 4 y 8 puede ser asignada como muestra la Figura 2.12 Por ejemplo, la cadena de código para la región en la Figura 2.11(a) es mostrada en la Figura 2.13. La Figura 2.13(a) muestra la cadena de código para la conectividad 4. En este caso, la dirección proveniente del punto inicial al siguiente punto es sur (i.e. código 2), así que el primer elemento de la cadena de código que escribe la silueta es un 2.

La dirección desde el punto P1 al siguiente, P2, es el este (código 1) así que el siguiente elemento en el código es 1. El siguiente punto después de P2 es P3 que se encuentra al sur dando un código 2. Esta codificación se repite hasta el P23 que está conectado hacia el este hasta el punto de partida, por lo que el último elemento (el elemento P24) del código es 1.

Nótese que la longitud del código es más corta para la conectividad 8, que para la conectividad 4 dada por el número de puntos del contorno.

Si bien se pueden almacenar las coordenadas de una secuencia de píxeles en la imagen o almacenar la posición relativa entre píxeles consecutivos, la relación que se encuentra en la Figura 2.13 b) es la siguiente,

$$código = \{3, 4, 3, 4, 4, 5, 4, 6, 7, 7, 7, 0, 0, 1, 1, 2\} \tag{2.10}$$

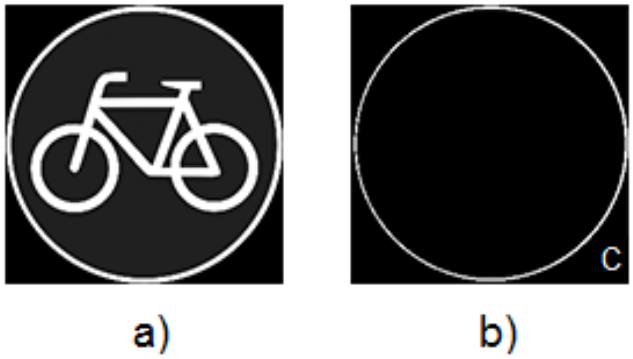

figura 2.14: Ejemplo de extracción de bordes de la imagen a) por medio de la cadena de código. El resultado es b) un conjunto C de pixeles que pertenecen al contorno. C.

en terminos de coordenadas píxel es,

$$código = \{(X_0, Y_0), (X_1, Y_1), ..., (X_k, Y_k)\} \tag{2.11}$$

donde $(X_k, Y_k) \in C, i = 0 \ldots k$ y C es el conjunto de coordenadas del píxel que conforman al contorno de un objeto en la imagen, que se ejemplifica en la Figura 2.14

2.5.3. Algoritmo de relleno por difusión (Floodfill)

El algoritmo de relleno por difusión (o en inglés floodfill), determina el área formada por elementos contiguos en una matriz multidimensional. Este algoritmo se utiliza para el relleno del interior de un contorno C y requiere de tres parámetros: un píxel inicial, un valor para sustituir y otro valor de relleno. El algoritmo rastrea todos los píxeles p que sean del valor seleccionado $f(p)$, y a la vez contiguos entre sí y con el inicial $N_4(p)$ o $N_8(p)$, y los sustituye por el valor de relleno.

Hay muchas maneras en las que el algoritmo de relleno por difusión puede ser estructurado, pero todas ellas hacen uso de tipos de datos tales como la cola o la pila o bien una cadena de código, explícita o implícitamente [3]. Una implementación del algoritmo de relleno por difusión basada en la cadena de código se define de la siguiente manera (para un arreglo bidimensional):

Flood-fill (píxel inicial, valor de relleno, valor de remplazo):
1. Si el valor de un píxel es distinto del que se pretende sustituir, se termina el algoritmo.
2. Remplaza el valor de un píxel por el propuesto.
3. Se ejecuta de nuevo el algoritmo, usando el píxel situado a la izquierda del presente, y los mismos parámetros de valor.

Se ejecuta de nuevo el algoritmo, usando el píxel situado a la derecha del presente, y los mismos parámetros de valor.

Se ejecuta de nuevo el algoritmo, usando el píxel situado inmediatamente superior al presente, y los mismos parámetros de valor.

Se ejecuta de nuevo el algoritmo, usando el píxel situado inmediatamente inferior al presente, y los mismos parámetros de valor.

4. Fin del algoritmo.

o bien,

$$\text{Si } [f(p) \vee f(q) \wedge N_4(p,q)] \text{ entonces } [N_4(p,q)]\exists[V_{remplazo}] \vee [N_8(p,q)]\exists[V_{remplazo}] \tag{2.12}$$

de lo contrario

$$f(p) = V_{relleno} \tag{2.13}$$
$$q = p \tag{2.14}$$

Este algoritmo esta basado en el método de la obtención de la cadena de código, debido a que su avance puede ser por medio de la conectividad 4 y 8, en este caso, se utiliza el ocho. El resultado de aplicar este algoritmo a un conjunto de puntos cerrados C se muestra en la Figura 2.15.

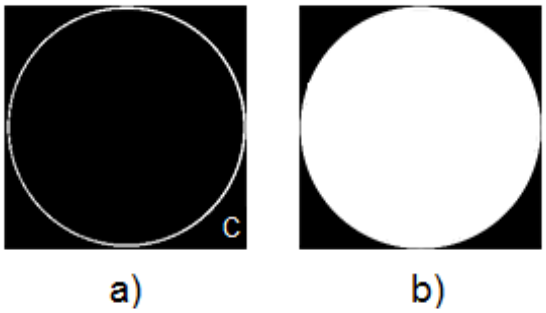

figura 2.15: a) Conjunto de puntos que conforman un controno cerrado C, b) Resultado de aplicar el algoritmo Floodfill con $N_8(p,q)$.

2.6. Descriptores de contorno

Los descriptores de contorno son aquellos algoritmos capaces de describir el contenido de varios tipos de información multimedia para poderlos buscar y clasificar. Éstos tienen un buen conocimiento de los objetos y eventos presentes en una imagen y permiten la búsqueda rápida y eficiente de los objetos.

Sea C el conjunto de píxeles que conforman el contorno de un objeto en la imagen. La pendiente de una función entre dos puntos, $P1(X_k, Y_k)$ y $P2(X_{k+\Delta}, Y_{k+\Delta})$ está dada por,

$$m = \frac{Y_{k+\Delta} - Y_k}{X_{k+\Delta} - X_k} \tag{2.15}$$

si y sólo si

$$k + \Delta x = \Delta_{fijo} \vee k + \Delta y = \Delta_{fijo}$$

donde Δx,Δy y Δ_{fijo} son los incrementos. En la Figura 2.16 se muestra un ejemplo del seguimiento del contorno usando la Ecuacion 2.15.

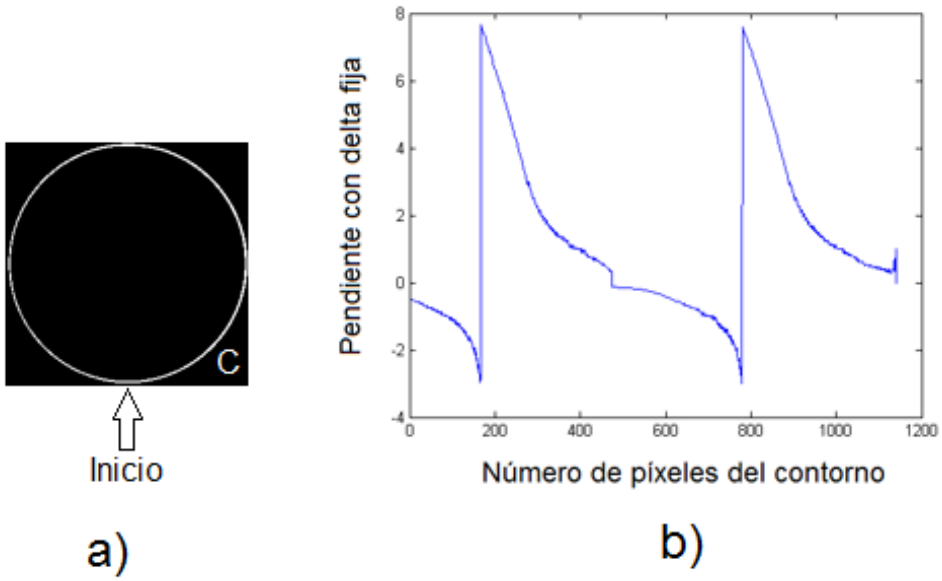

figura 2.16: Descriptor del contorno de un círculo por medio de la pendiente. a) Contorno de una figura (2D) b) Gráfica (1D) del seguimiento del contorno C. Comenzando desde el pixel Inicio con dirección de las manecillas del reloj.

2.7. Transformaciones espaciales: convolución y correlación

Las tranformaciones matemáticas que se describen en esta sección tienen como objetivo favorecer o detectar alguna caraterística presente en la imagen original, o bien, eliminar alguna otra asociada con ruido o que este ocultando información en la imagen [4].

En esta sección se analizan las transformaciones en el dominio del espacio: convolución y correlación.

2.7.1. Convolución

La convolución discreta de dos señales $f(x)$ y $h(x)$ de tamaño M en el dominio del espacio es denotada por,

$$g(x) = f(x) * h(x) = \frac{1}{M} \sum_{i=1}^{M} f(i)h(x-i) \tag{2.16}$$

en dos dimensiones se presenta como,

$$g(x,y) = f(x,y) * h(x,y) = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} f(i,j)h(x-i,y-j) \tag{2.17}$$

donde MxN es el tamaño en píxeles de una imagen y $h(x)$ es un filtro o máscara de tamaño mxn.

Filtraje pasa bajas

El resultado de aplicar un filtro pasa bajas $h(x,y)$ a una imagen $f(x,y)$ es simplemente el promedio de los píxeles contenidos en el vecindario (o entorno) de la máscara utilizada. Sea $f(x,y)$ una función imagen sobre la cual se convoluciona un filtro promedio definido por,

$$h(x,y) = \frac{1}{mn} \begin{bmatrix} 1 & \cdot & \cdot & \cdot & 1 \\ \cdot & \cdot & & & \\ \cdot & & \cdot & & \\ \cdot & & & \cdot & \\ 1 & & & & 1 \end{bmatrix} \tag{2.18}$$

donde mxn es el tamaño en pixeles de la máscara. El efecto que produce el filtro promedio sobre los pixeles de la imagen es el de suavizado, debido a que cada píxel de la imagen filtrada es el promedio de sus vecinos.

Este tipo de filtro suele ser utilizado para eliminar ruido dentro de la imagen, tomando en cuenta que el tamaño del filtro (máscara) esta directamente relacionado con el tamaño del ruido a tratar. Este tipo de filtros también reducen el contraste en la imagen. En la Figura 2.17 se muestra claramente el resultado del filtraje pasa bajas.

Filtraje pasa altas

El objetivo principal de estos filtros es resaltar los detalles delicados de una imagen (por ejemplo, los bordes) o realzar los detalles de una imagen borrosa. Los filtros pasa altas que estudiaremos aquí están basados en la primera y segunda derivada. Las derivadas de una función discreta se definen en términos de diferencias. La derivada parcial de primer orden de una imagen $f(x,y)$ es la diferencia,

figura 2.17: Ejemplo de filtrado pasa bajas por medio de convolución. a) Imagen original $f(x, y)$ b) Resultado de la convolución con un filtro $f(x, y)$ de 9x9 *píxeles*.

$$\frac{\partial f}{\partial x} = f(x+1, y) - f(x, y), \tag{2.19}$$

$$\frac{\partial f}{\partial y} = f(x, y+1) - f(x, y), \tag{2.20}$$

de la misma manera se define la derivada de segundo orden como,

$$\frac{\partial^2 f}{\partial x^2} = f(x+1, y) - f(x, y) + f(x-1, y) - f(x, y), \tag{2.21}$$

$$\frac{\partial^2 f}{\partial x^2} = f(x+1, y) + f(x-1, y) - 2f(x, y), \tag{2.22}$$

$$\frac{\partial^2 f}{\partial y^2} = f(x, y+1) - f(x, y) + f(x, y-1) - f(x, y), \tag{2.23}$$

$$\frac{\partial^2 f}{\partial y^2} = f(x, y+1) + f(x, y-1) - 2f(x, y). \tag{2.24}$$

El Laplaciano para una función imagen está definido por,

$$\bigtriangledown^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y). \tag{2.25}$$

A partir de esto, el filtro Laplaciano tiene los siguientes coeficientes [4],

En la Figura 2.18 se muestra un ejemplo del filtrado pasa altas, donde se puede apreciar el realce de los bordes y orillas.

	$y-1$	y	$y+1$
$x-1$	0	1	0
x	1	−4	1
$x+1$	0	1	0

Tabla 2.1: Máscara Laplaciana

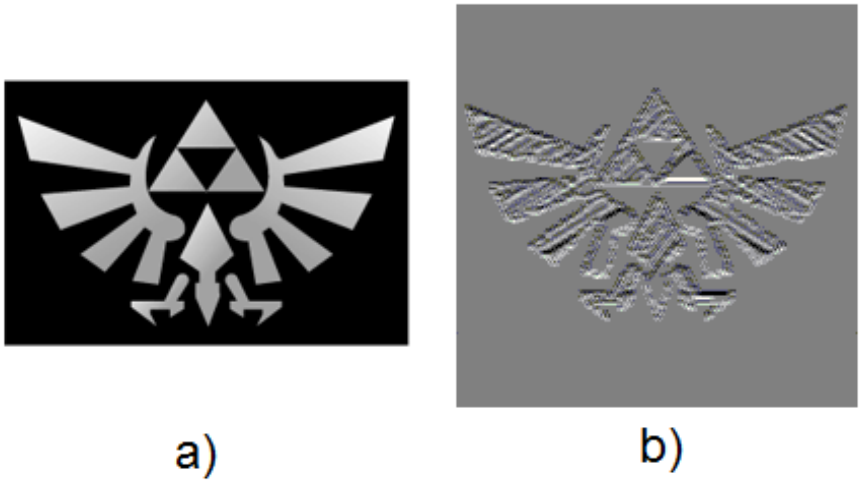

figura 2.18: Ejemplo de una imagen convolucionada por un filtro Laplaciano de la Tabla 2.1.

2.7.2. Correlación

La correlación cruzada entre dos señales $f(x)$ y $h(x)$ está definida como [4];

$$g(x) = f(x) \circ h(x) = \frac{1}{M} \sum_{i=1}^{M} f(i)h(x+i). \tag{2.26}$$

Sean,

$$f(x) = \begin{cases} 1 & 0 < x \leq 1 \\ 2 & 1 < x \leq 3 \\ 1 & 3 < x \leq 4 \\ 0 & \text{en cualquier otro caso} \end{cases}, \tag{2.27}$$

$$h(x) = \begin{cases} 2 & -1 < x < 1 \\ 0 & \text{en cualquier otro caso} \end{cases}, \tag{2.28}$$

Entonces el resultado de correlación de $f(x)$ y $h(x)$ es:

$-1 \leq x < 0$

$$g(x) = \int_0^{x+1} 2 \cdot 1di = 2(x+1)$$

$0 \leq x < 1$

$$g(x) = \int_0^1 2 \cdot di + \int_1^{x+1} 2 \cdot 2di = 2 + 4x$$

$1 \leq x < 2$

$$g(x) = \int_{x-1}^1 2 \cdot 1di + \int_1^{x+1} 2 \cdot 2di = 4 + 2x$$

$2 \leq x < 3$

$$g(x) = \int_{x-1}^3 2 \cdot 2di + \int_3^{x+1} 2 \cdot 1di = 12 - 2x$$

$3 \leq x < 4$

$$g(x) = \int_{x-1}^3 2 \cdot 2di + \int_3^4 2 \cdot 1di = 18 - 4x$$

$4 \leq x < 5$

$$g(x) = \int_{x-1}^4 2 \cdot 1di = 2(5-x)$$

En la Figura 2.19 puede verse la correlación entre dos señales. Cuando las dos funciones son iguales se le denomina autocorrelación. El funcionamiento es parecido al de la convolución, la diferencia se encuentra en un cambio de signos.

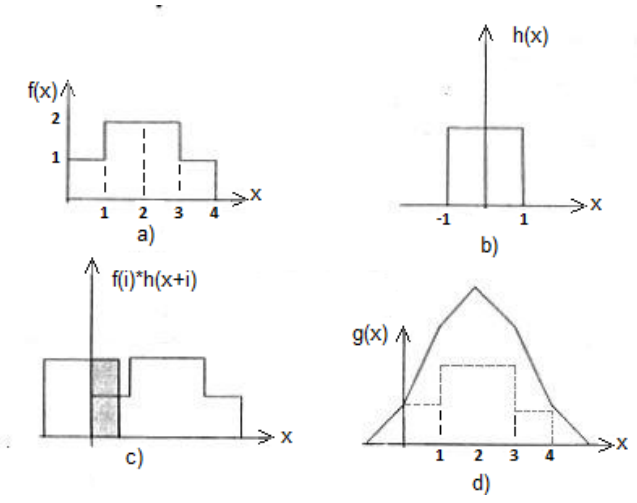

figura 2.19: Correlación de dos señales a) $f(x)$ b) $h(x)$ c) $g(x)$ d) El valor de correlación está asociado al valor del área de intersección de las dos señales.

2.7.3. Coeficientes de correlación lineal de Pearson

El coeficiente de correlación es una medida de asociación entre dos funciones y se define como [5],

$$corr(f,h) = \frac{N\sum f(x)h(x) - \sum f(x)\sum h(x)}{\sqrt{\left[N\sum f(x)^2 - (\sum f(x))^2\right]\left[N\sum h(x)^2 - (\sum h(x))^2\right]}}, \tag{2.29}$$

donde N es el tamaño de los vectores $f(x)$ y $h(x)$. El rango de resultados de la correlación se encuentran entre los valores 0 y 1, donde 1 es el valor de correlación máximo (las funciones son iguales), 0 es el valor minímo (las funciones son entramente diferentes) y los valores intermedios representan la similitud existente entre ambas funciones.

2.7.4. Correlación de imágenes

Correlación lógica

Sea $f(x,y)$ y $h(x,y)$ dos funciones imagen binarias y del mismo tamaño, la correlación lógica se define como:

$$g(x,y) = \begin{cases} 1 & f(x,y) == h(x,y) \\ 0 & \text{en cualquier otro caso} \end{cases},$$

y por tanto el grado de correlación es,

$$corr(f,h) = \frac{1}{MN}\sum_{x=1}^{M}\sum_{y=1}^{N} g(x,y) \tag{2.30}$$

Es importante tener en cuenta que para que esta correlación lógica funcione adecuadamente, ambas imágenes deben tener exactamente los dos mismos valores (0 y 255 en el caso anterior) y que ademas sean del mismo tamaño, pues el valor de cada píxel perteneciente a determinada posición es comparado con el píxel de la otra imagen en la misma posición. [3]

La correlación en dos dimensiones se presenta como,

$$corr(f,h) = g(x,y) = h(x,y) \circ f(x,y) = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N} f(i,j)h(x+i,y+j), \tag{2.31}$$

donde $f(x,y)$ es una función imagen.

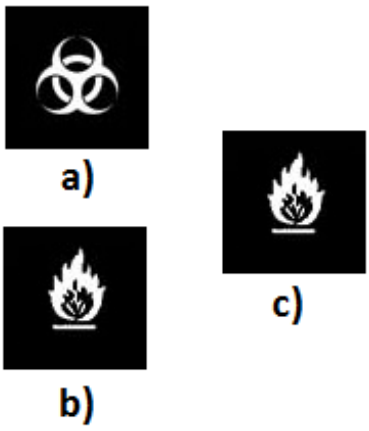

figura 2.20: Correlación lógica. La correlación lógica entre a) y c) es de 0.90 y mientras que entre b) y c) es de 1.

Observando el resultado anterior puede deducirse que una posible aplicación de la correlación es la localización de un patrón dentro de una imagen, ya que el valor donde la correlación es máxima corresponde a las coordenadas donde ese patrón se encuentra. Se tiene la correlación normalizada, que además evita posibles cambios en los niveles de gris entre el objeto que se busca y el objeto presente en la imagen.

$$corr(f,h) = g(x,y) = \frac{\sum_i \sum_j (f(x+i,y+j) - \overline{f})(h(i,j) - \overline{h})}{\sqrt{\sum_i \sum_j (f(x+i,y+j) - \overline{f})^2 \sum_i \sum_j (h(i,j) - \overline{h})^2}}, \tag{2.32}$$

En la Figura 2.21 puede verse el uso de la correlación normalizada. En este caso se trata de tres imágenes diferentes dentro de la misma imagen. Puede observarse que la correlación no es inmune a cambios en la escala, ni a la rotación del objeto.

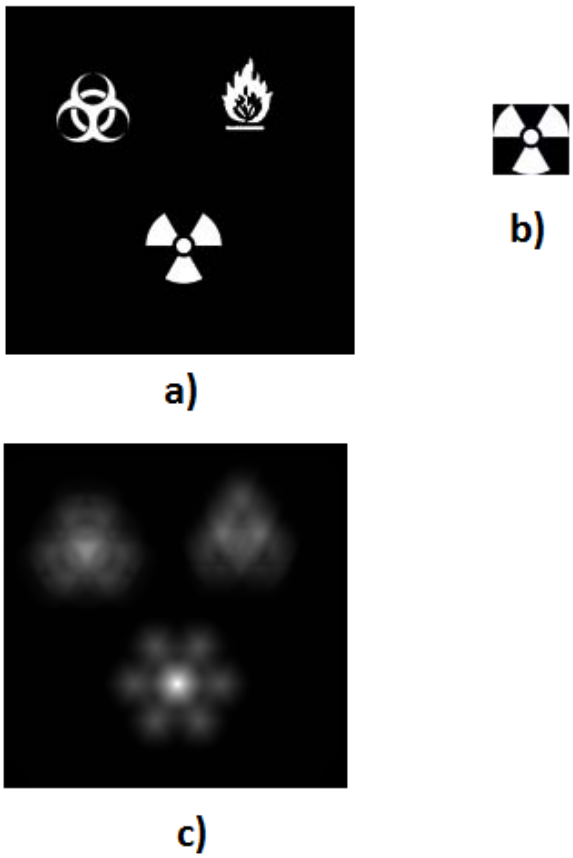

figura 2.21: Detección de un objeto mediante correlación. a) Imagen b) Plantilla c) Muestra el punto donde la mejor correlación tuvo lugar.

2.8. Transformada Wavelet

2.8.1. Transformación Wavelet en 1D

La transformada Haar descompone una señal discreta en sucesivas subseñales de la mitad de su tamaño. La primera como resultado de promedios normalizados y la segunda por diferencias. Por lo que, también puede verse como filtraje pasa bandas y pasa altas.

Sea f una señal discreta $f = \{f_1, f_2, ..., f_N\}$ normalizada, donde $N \in \mathbb{Z}^+$, sea $a^1 = (a_1, a_2, ..., a_{\frac{N}{2}})$ la primera subseñal calculada a partir de f, llamada primer tendencia. Su primer valor, a_1, es calculado tomando el promedio del primer par de valores de $f' = \frac{(f_1+f_2)}{2}$ multiplicado por $\sqrt{2}$ para preservar la energía de la señal; de esta manera y sucesivamente se obtiene la formula general para los valores a_m como [6],

$$a^m = \frac{f_{2m-1} + f_{2m}}{2}\sqrt{2} \tag{2.33}$$

para $m = 1, 2, ..., \frac{N}{2}$.

La otra subseñal llamada la primera fluctuación, es denotado por $d^1 = (d_1, d_2, ..., d_{\frac{N}{2}})$, y es calculada al tomar la diferencia entre pares de valores, definiendo así la función general como,

$$d^m = \frac{f_{2m-1} - f_{2m}}{2}\sqrt{2} \tag{2.34}$$

La transformada Haar es llevada a cabo en varios niveles. El primer nivel es el mapeo H, compuesto por la primer tendencia a^1 y la primera fluctuación d^1, definido por,

$$f \overset{H_1}{\rightarrow} (a^1|d^1). \tag{2.35}$$

El nivel de transformación 2 de f es a partir de a^1, usando las Ec. (2.33) y (2.34) queda,

$$f \overset{H_2}{\rightarrow} (a^2|d^2|d^1). \tag{2.36}$$

El nivel de transformación Haar de nivel k, es,

$$f \overset{H_k}{\rightarrow} (a^\gamma|d^\gamma|d^{\gamma-1}|...|d^1). \tag{2.37}$$

Un ejemplo de este proceso se muestra a continuación,

$$\begin{gathered} f = (6, 4, 5, 7, 1, 3, 8, 2) \\ a^1 = (5\sqrt{2}, 6\sqrt{2}, 2\sqrt{2}, 5\sqrt{2}) \\ d^1 = (2\sqrt{2}, -2\sqrt{2}, -2\sqrt{2}, 6\sqrt{2}) \end{gathered} \tag{2.38}$$

2.8.2. Transformada Wavelet en 2D

A continuación se describe un método para el cálculo de la Transformada Wavelet 2D a partir de matrices de transformación. El conjunto de funciones $\{\Psi_n(x)\}$ que buscamos son del tipo medible, con soporte compacto y cuadráticamente integrables sobre la recta real. Este conjunto se denota como $L^2(\Re)$, por lo que $\forall \Psi_n(x) \in L^2(\Re)$ se cumple que

$$\int_{-\infty}^{\infty} |\Psi_n(x)|^2 \, dx < \infty \tag{2.39}$$

En el análisis wavelet, se genera un conjunto de funciones base, por dilatación y traslación de una sola función prototipo, $\Psi_n(x)$, la cual llamamos Wavelet Madre.

Esta es alguna función oscilatoria, usualmente centrada en el origen, que tiende rápidamente a cero cuando $|x| \to \infty$ como se observa en la gráfica de la Figura 2.22

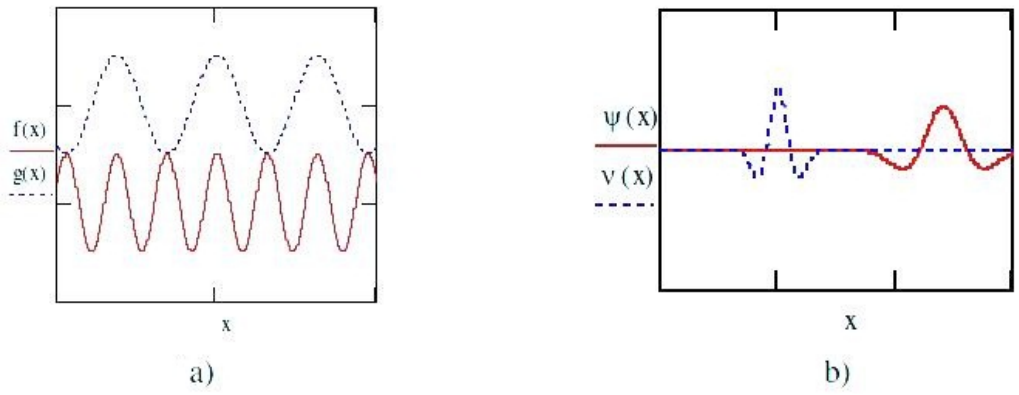

figura 2.22: a) Ondas Sinusoidales con soporte infinito b) Wavelets con soporte compacto.

Se tiene entonces que, el conjunto de funciones Wavelet base $\{\Psi_{a,b}(x)\}$ pueden ser generadas por traslaciones b y escalamientos a del Wavelet Madre $\Psi(x)$ como se muestra en siguiente expresión:

$$\Psi_{a,b}(x) = \frac{1}{\sqrt{a}} \Psi\left(\frac{x-b}{a}\right), \tag{2.40}$$

donde $a > 0$ y $b \in \Re$.

Normalmente el Wavelet Madre $\Psi(x)$, esta centrado en el origen, por lo que, el wavelet $\Psi_{a,b}(x)$ está centrado en $x = b$, como se observa en la Figura 2.23. Este wavelet particular está dado por,

$$\Psi_{a,b}(x) = \frac{2}{\sqrt{3a\sqrt{\pi}}}\left[1 - [\frac{x-b}{a}]^2\right] e^{\frac{-[\frac{x-b}{a}]^2}{2}}. \tag{2.41}$$

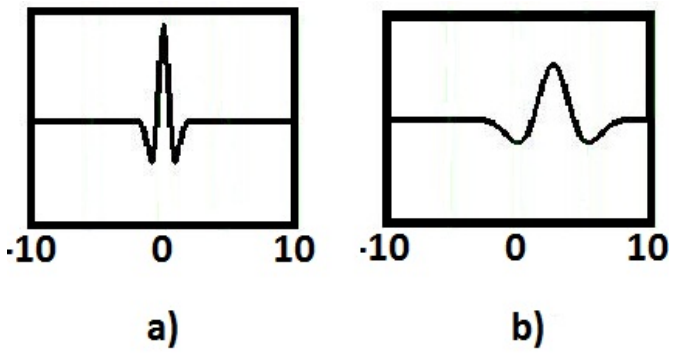

figura 2.23: (a) Wavelet Madre en $b = 0$ y $a = 1$. (b) Wavelet con escala $a = 3$ y traslación $b = 5$

2.8.3. Wavelets Bivalentes

Como en series de Fourier, una forma de calcular la transformada wavelet es vía una expansión en series wavelet, donde ahora el Wavelet Madre, una vez escalado y trasladado, forma un conjunto de funciones base, pero ahora sus escalamientos serán binarios (factores de dos, 2^p) y sus translaciones bivalentes. Una traslación bivalente es un desplazamiento $b = q^{2^p}$, el cual es un múltiplo entero q del factor de escala binario, por lo que el ancho del wavelet, $a = 2^p$, también lo es. Sustituyendo los nuevos valores para a y b en la Ecuación (2.40), resulta que un wavelet bivalente queda expresado como,

$$\Psi_{p,q}(x) = \frac{1}{\sqrt{2^p}}\Psi\left[\frac{x - q^{2^p}}{2^p}\right],$$

es decir,

$$\Psi_{p,q}(x) = 2^{-\frac{p}{2}}\Psi(2^{-p}x - q), \tag{2.42}$$

ó

$$\Psi_{p,q}(x) = 2^{\frac{p}{2}}\Psi(2^p x - q), \tag{2.43}$$

donde $-\infty < p, q < \infty$ son números enteros. El entero p determina la dilatación o escalamiento, mientras que q especifica la traslación.

Se tiene aquí la base de la transformada wavelet discreta. Si f es una función discreta muestreada en N valores, con N en potencias de 2, y si $\Psi_k(x)$ es un wavelet bivalente,

entonces podremos calcular la transformada wavelet discreta usando versiones discretas de las funciones anteriores.

2.8.4. Transformada Haar

Alfred Haar, un matemático Húngaro, descubre una base de funciones que se reconocen actualmente como las primeras wavelets. Consisten en un breve impulso positivo seguido de un breve impulso negativo.

El más simple ejemplo de una familia de funciones apropiadas para el análisis multiresolución en el espacio de funciones cuadráticamente integrables sobre la línea real, están dadas por la familia de funciones Haar. Tales funciones constituyen una base ortogonal. Por lo que la transformada Haar discreta se beneficia de las propiedades de transformaciones ortogonales. La inversa de una transformación ortogonal es también particularmente fácil de implementar porque ésta es la transpuesta de una transformación directa [6].

La transformada wavelet Haar es una transformada lineal y separable basada en la función escalón

$$g(x) = \begin{cases} 1 & \text{Si } 0 \leq x \leq 1 \\ 0 & \textit{en otra parte} \end{cases} \tag{2.44}$$

y en la función Haar, expresada como:

$$h(x) = \begin{cases} 1 & \text{Si } 0 \leq x \leq \frac{1}{2} \\ -1 & \text{Si } \frac{1}{2} \leq x \leq 1 \\ 0 & \textit{en otra parte} \end{cases} \tag{2.45}$$

Para la formulación de una matriz de transformación Haar, de tamaño NxN, con $N = 2^n$, se hace uso de las funciones Haar, $h_k(x)$ las cuales están definidas sobre un intervalo continuo y cerrado [0,1]. Por lo que, para determinar una base ortonormal se define [7],

$$h_k(x) \equiv 2^{\frac{p}{2}} h(2^p x - q) \tag{2.46}$$

donde el entero k puede ser generado únicamente como en las ecuaciones (2.48) y (2.49).

Se tiene la función Wavelet Madre,

$$\Psi_k(x) = 2^{\frac{p}{2}} \Psi(2^p x - q), \tag{2.47}$$

donde p y q son funciones de k, como se muestra en la siguiente relación [7],

$$k = 2^p + q - 1, \tag{2.48}$$

con

$$k = 1, ..., N-1,\ p = 0, 1, ..., n-1 \text{ y } q = 1, ..., 2^p, \tag{2.49}$$

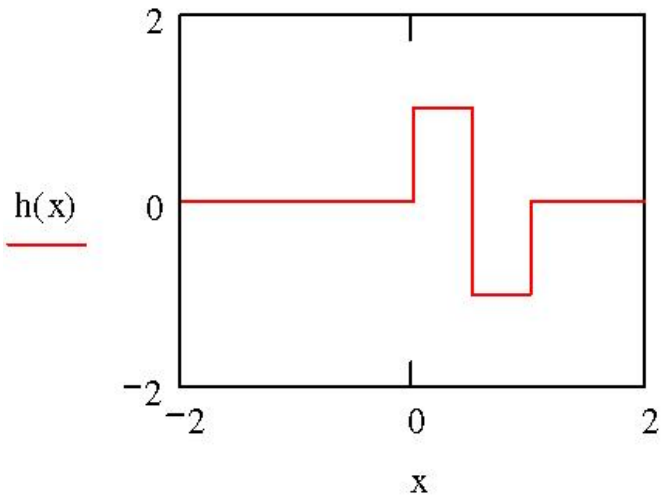

figura 2.24: Función Haar

donde $N = 2^n$, y n un número entero positivo.

Para p y q fijas, la función $h_k(x)$ es una translación q de la función $h(2^p x)$ y esta a su vez es una dilatación 2^p de la función wavelet Haar $h(x)$. Por lo que, la infinita familia de funciones $h_k(x)$ junto con la función escalón $g(x)$ constituyen una base ortogonal, para el conjunto de funciones cuadráticamente integrables sobre el intervalo unidad. Con esto, el conjunto ortogonal de funciones Haar está definido como:

$$h_0(x) = \frac{1}{\sqrt{N}} \qquad \text{(función escalón)}$$

$$h_k(x) = \frac{1}{\sqrt{N}} \begin{bmatrix} 2^{\frac{p}{2}} & \text{Si } \frac{(q-1)}{2^p} \leq x < \frac{(q-\frac{1}{2})}{2^p} \\ -2^{\frac{p}{2}} & \text{Si } \frac{(q-\frac{1}{2})}{2^p} \leq x < \frac{q}{2^p} \\ 0 & otra\_parte \end{bmatrix} \qquad \text{(funciónes Haar)} \tag{2.50}$$

En la Figura 2.25 se muestran un conjunto de funciones base wavelet Haar, $\{h_k(x_i)\}$ ortonormales, siendo k el índice de las funciones .

El wavelet Haar básico es continuamente reducido por potencias de dos, esto es, cada función $h_k(x)$ está definida en un intervalo 2^{-p} . Además cada wavelet reducido es transladado por incrementos igual a su ancho, por lo que el conjunto de wavelets h_k a cualquier escala cubre completamente el intervalo [0,1].

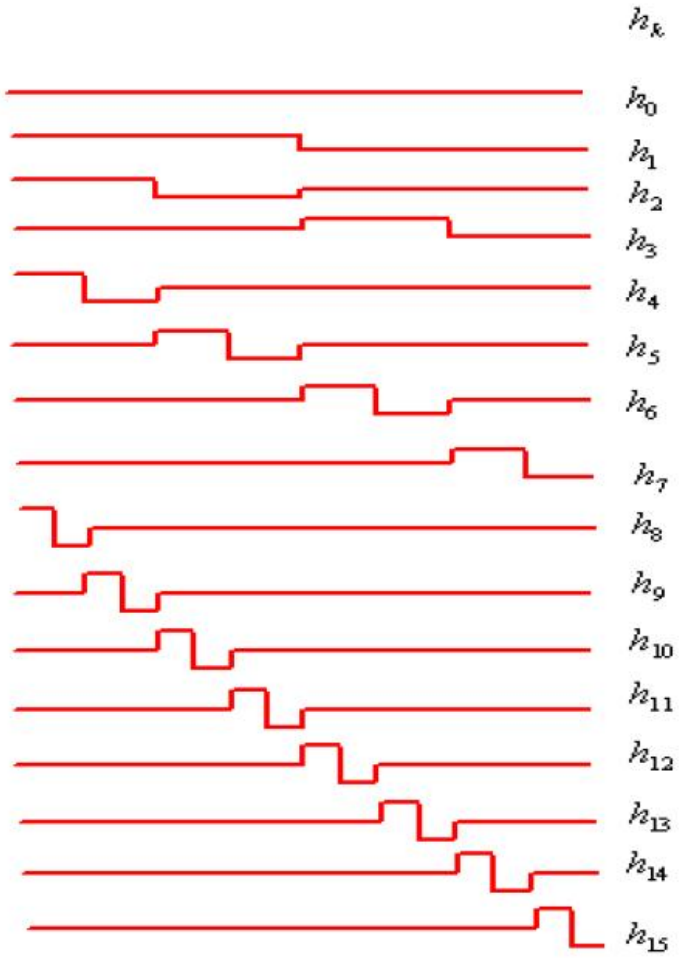

figura 2.25: Funciones base de la Transformada Haar,H_N,para $N = 16$.

El cálculo de la transformada Haar de $A_{i,j}$, da como resultado una matriz de coeficientes $C_{i,j}$, de tamaño MxN dados por:

$$C_{i,j} = H_m[A_{i,j}]H_n^T \tag{2.51}$$

Un ejemplo de la implementación de la transformada wavelet Haar para imágenes digitales se muestra en la Figura 2.26, donde se presenta la imagen digital de "Lena" y su transformada wavelet Haar.

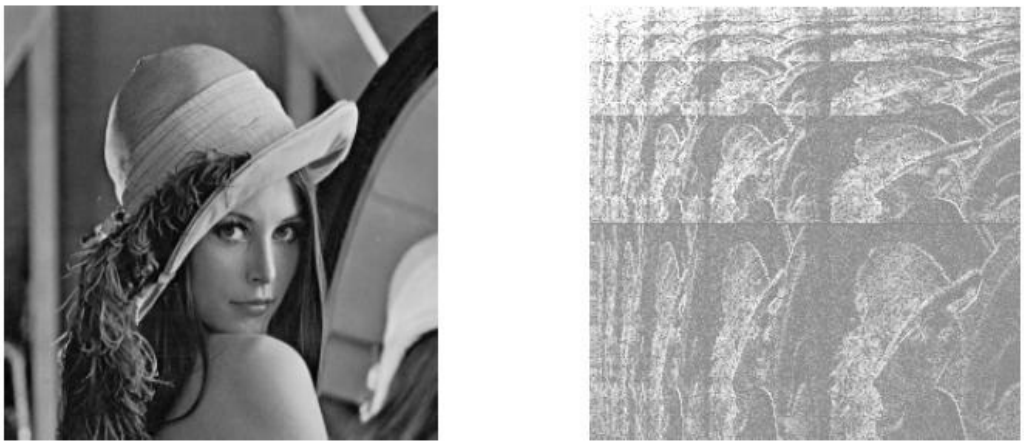

figura 2.26: (a) Imagen $A_{i,j}$, de la modelo "Lena" (b) Transformada Wavelet Haar $C_{i,j}$, de "Lena"

La imagen mostrada en la Figura 2.26 b) es la matriz de datos de la transformada, ahora con coeficientes $C_{i,j}$, y con un tamaño igual al de la imagen original, 2^l por 2^k. A diferencia de la original, esta matriz es particionada en una rejilla multiresolución con $(l+1)$ por $(k+1)$ subregiones, donde cada subregión corresponde a una particular escala o resolución sobre los ejes x y y.

2.9. Conclusiones

En este capítulo se estudió el proceso de formación de imágenes, desde la detección y medición de la radiación electromagnética, haciendo énfasis en el espectro visible e infrarrojo, hasta el muestreo y cuantización de imágenes contínuas que generan imágenes digitales.

Para el análisis y procesamiento de estas imágenes se revisarón técnicas para la delimitación, reducción y descripción de regiones. Lo anterior, nos permitirá, extraer y describir objetos de la imágen para su posterior reconocimiento y clasificación.

Durante el proceso de adquisición de imágenes, generalmente se tiene ruido no deseable. Es por ello que se estudió la convolución para el filtraje y realce de imágenes usando filtros basados en promedio y derivadas.

La correlación 1D de dos funciones fue usada para determinar el grado de similitud entre ellas; y la correlación en 2D estudiada fue usada para: a) ubicar un objeto en la imágen y b) comparar distribuciones entre imágenes. Finalmente, se estudia la transformada wavelet en 1D y 2D para la compresión de información en resoluciones en términos de potencias de 2.

Bibliografía

[1] R. D. Fiete, "Modeling the Imaging Chain of Digital Cameras", *Tutorial Texts in Optical Enginering* SPIE Press (2010).

[2] M. S. Nixon, A. S. Aguado, "Feature Extraction and Image Processing", British Library Cataloguing in Publication Data (2002).

[3] R. C. Gonzales, R. E. Woods, "Digital Image Processing", *Second Edition*, Prentice Hall (2002).

[4] Escalera "Vision por Computadora", *Tal Edicion*, Prentice Hall (2002).

[5] GOVINDEN, Lincoyán, "Introducción a la Estadística", McGraw Hill. Interamericana Editores. S.A., (1985).

[6] C. Toxqui, A. Padilla, E. Sánchez, "Fusión Digital de Imágenes usando Transformada Wavelet"Tesis de Licenciatura (2004).

[7] J. S. Walker ". A primer on wavelets and their scientific applications". Chapman and Hall /CRC (1999)

Capítulo 3

Redes Neuronales Artificiales (RNA)

3.1. Introducción

La teoria y el diseño de la redes neuronales artificiales han tenido un avance significativo durante los pasados 20 años. Gran parte de este progreso tiene una influencia directa en el procesamiento de señales. En particular, la naturaleza no lineal de las redes neuronales, así como su capacidad para aprender en un entorno supervisado, así como sin supervisión, de igual manera la propiedad de aproximación universal a las redes neurales hacen muy adecuado para la solución de problemas de procesamiento de señales.

Desde la perspectiva del procesamiento de señales es imperativo desarrollar una comprensión adecuada de las estructuras base de las redes neuronales así como del impacto que tienen los algoritmos de procesamiento de señales y sus aplicaciones. El reto en el campo de revisión de paradigmas de redes neuronales es identificar aquellas estructuras de redes neuronales que se han aplicado con éxito para resolver problemas del mundo real de las que siguen en fase de desarrollo o tienen dificultades para la ampliación y aplicación para resolver problemas reales. Cuando se trata con aplicaciones de procesamiento de señales, es fundamental comprender la naturaleza del problema para realizar la formulación, de modo que el paradigma de redes neuronales más apropiado pueda ser aplicado. Además, también es importante evaluar el impacto de las redes neuronales en el rendimiento, robustez y coste-efectividad de los sistemas de procesamiento de señales y desarrollar metodologías para la integración de las redes neuronales con otros algoritmos de procesamiento. [1]

Otra cuestión importante es la forma de evaluación de los paradigmas de las redes neuronales, los algoritmos de aprendizaje y las estructuras de redes neuronales para identificar aquellas que hacen o no hacen su función de manera fiable en la solución de problemas del procesamiento de señales. En esta sección, se describe un modelo de preceptrón simple y multicapa que servirá posteriormente para la clasificación general de patrones. El tipo de entrenamiento de la red neuronal es supervisado, y como funciones de activación se utilizan la función escalón y sigmoidal.

3.2. El Perceptrón Simple

3.2.1. Estructura de la neurona

Una neurona es una célula viva y, como tal, contiene los mismos elementos que forman parte de todas las células biológicas. Además contiene elementos característicos que la diferencian. En general una neurona consta de un cuerpo celular más o menos esférico, de 5 a 10 micras de diámetro del que salen una rama principal, el axón, y varias ramas más cortas, llamadas dendritas. A su vez, el axón puede producir ramas en torno a su punto de arranque, y con frecuencia se ramifica extensamente cerca de su extremo.

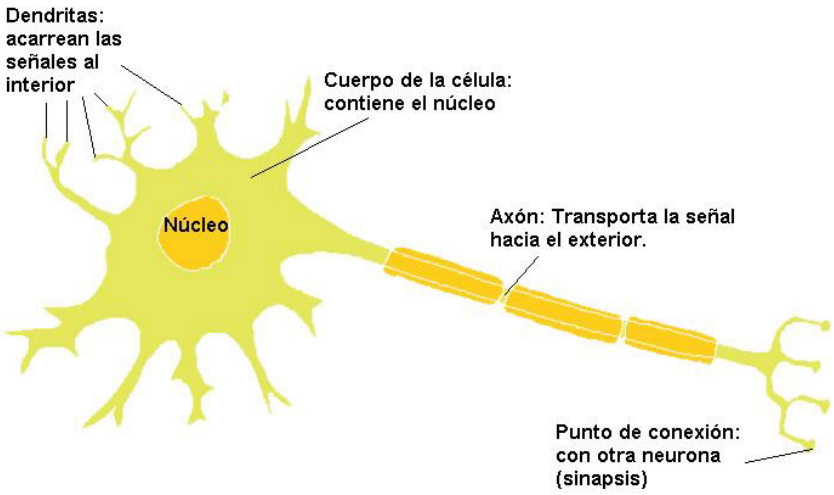

figura 3.1: Forma general de una neurona

Una de las características que diferencian a las neuronas del resto de las células vivas es su capacidad de comunicarse. En términos generales las dendritas y el cuerpo celular reciben señales de entrada; el cuerpo celular las combina e integra y emite las señales de salida. El axón transporta esas señales a los terminales axónicos, que se encargan de distribuir información a un nuevo conjunto de neuronas. Por lo general, una neurona recibe información de miles de otras neuronas y, a su vez, envía información a miles de neuronas más. Se calcula que en el cerebro humano existen del orden de 10^{15} conexiones.

La sinapsis o articulación interneuronal corresponde a las estructuras que permiten el paso del impulso nervioso desde una célula nerviosa a otra. Básicamente, la información recibida a través las dendritas es acarreada hacia el interior de la neurona, este impulso es procesado y posteriormente reflejado y transmitido por el axón hacia el exterior nuevamente (hacia otra neurona) por medio de la dentritas. La sinapsis se realiza cuando las dendritas de una neurona realizan un intercambio de impulsos nerviosos con otra neurona.

3.2.2. El Perceptrón

El Perceptrón fue el primer modelo de red neuronal artificial (RNA) desarrollado por Rosenblatt en 1958 [2]. Esta RNA despertó un enorme interés en los años 60, debido a su capacidad para aprender a reconocer patrones sencillos, un perceptrón, formado por varias neuronas lineales para recibir las entradas a la red y una neurona de salida, es capaz de clasificar diferenciando a una de las dos clases.

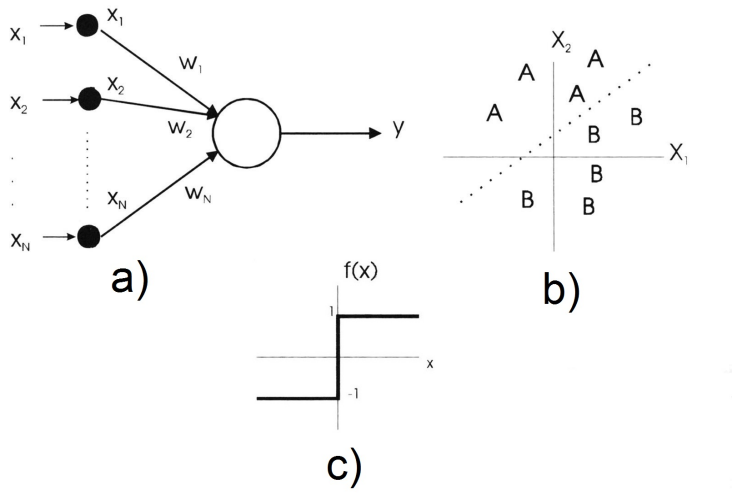

figura 3.2: El Perceptrón. a) Representación gráfica del perceptrón, b) El perceptrón puede clasificar un vector analógico de entrada en dos clases A y B, c) Función de activación.

El valor de una neurona de salida está definida por,

$$net = \sum_{i=1}^{N} w_i x_i - \theta, \tag{3.1}$$

$$y = f\left[net\right], \tag{3.2}$$

donde w_i y x_i son vectores de entrada y pesos respectivamente, θ es el bias y f es la función de activación de la neurona.

Para el caso de dos entradas podemos expresar la frontera de decisión como,

$$x_2 = \frac{-w_1}{w_2} x_1 + \frac{\theta}{w_2}. \tag{3.3}$$

En la Figura 3.2 se muestra un esquema representativo del perceprón simple, la función de activación y la gráfica de la frontera de decisión. Cabe señalar que es importante considerar el valor del bias θ para evitar que la frontera de decisión atraviese siempre por el origen.

Si las entradas son linealmente separables, el perceptrón ajusta la frontera de decisión dentro de la zona en un número k de épocas, de lo contrario, el perceptrón no converge a la solución como es el caso de la operación OR-Exclusivo (Figura 3.3).

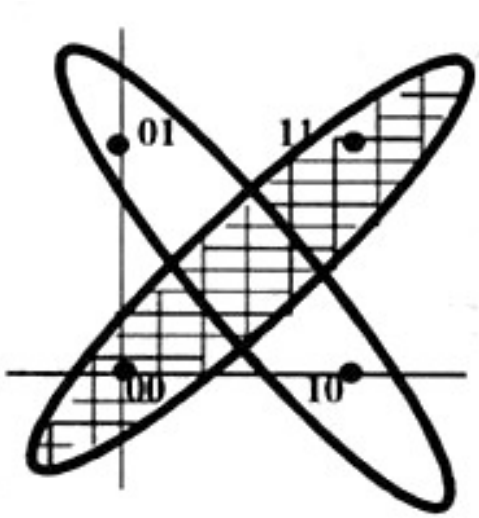

figura 3.3: Función XOR. No es posible obtener una recta que separe las dos clases

3.2.3. Función de activación

En el modelo biológico de una neurona (ver Figura 3.1), el axón cumple con la función de enviar el impulso de salida a miles de neuronas con las que se esta comunicando, este impulso se ve limitado por una resistencia natural de la célula, si este impulso no fuera acotado la neurona corre el riesgo de dañar significativamente la pared del axón. En un perceptrón, esta dinámica se modela por medio de la función de activación quien a su vez toma la decisión inhibitoria o exitatoria de la neurona, es decir la decisión de enviar o no un impulso de salida.

Una función de activación cumple generalmente (aunque no necesariamente) con las siguientes propiedades:

1. La decisión inhibitoria/excitatoria se encuentra centrada en el origen.

2. Los valores de salida están acotados frecuentemente en el intervalo $[-1, 1]$ ó $[0, 1]$, de tal forma que cuando el valor absoluto de la integración sea muy grande, el valor de salida o de activación permanece pequeño o constante.

Existen varias funciones de activación en la literatura, las más comunes son: función escalón, lineal y mixta, sigmoidal y gaussiana. Es de suma importancia elegir la función adecuada para cada problema, por lo general se elige una misma función de activación para todas las neuronas del modelo de RNA, comunmente las neuronas artificales toman el nombre de la función de activación con la que estan trabajando. Para los propositos de este trabajo, vamos a describir las funciónes de activación escalón y sigmoidal.

Función de activación escalón

La forma más fácil de definir la activación de una neurona es considerar que ésta es binaria. La función de transferencia escalón se asocia a neuronas binarias en las cuales cuando la suma de las entradas es mayor o igual que el umbral de la neurona, la activación es 1; si es menor, la activación es 0 (ó -1). En la Figura 3.4, en ambos casos se ha tomado que el umbral es cero; en caso de que no lo fuera, el escalón quedaría desplazado.

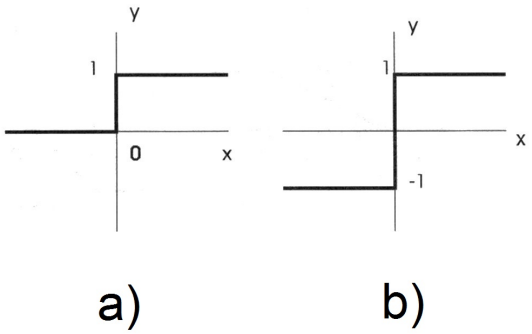

figura 3.4: Función de transferencia escalón. a) $f(x) = 1$ si ≥ 0 y 0 si $x \leq 0$ y en b) $f(x) = 1$ si ≥ 0 y -1 si $x \leq 0$.

Por otro lado, las redes formadas por este tipo de neuronas son fáciles de implementar en hardware, pero a menudo sus capacidades están limitadas.

Función de activación sigmoidal

La importancia de la función sigmoidal (o cualquier otra función similar) es que su derivada es siempre positiva y cercana a cero para los valores grandes positivos o negativos; además, la derivada toma su valor máximo cuando x es 0. Esto hace que se puedan utilizar las reglas de aprendizaje definidas para las funciones escalón, con la ventaja, respecto a esta función, de que la derivada está definida en todo el intervalo. La función escalón no tiene definida la derivada en el punto de transición, y esto no ayuda a los métodos de aprendizaje en los cuales se usan derivadas. La función que se muestra en la Figura 3.5 es $f(x) = \frac{1}{(1+e^{-x})}$.

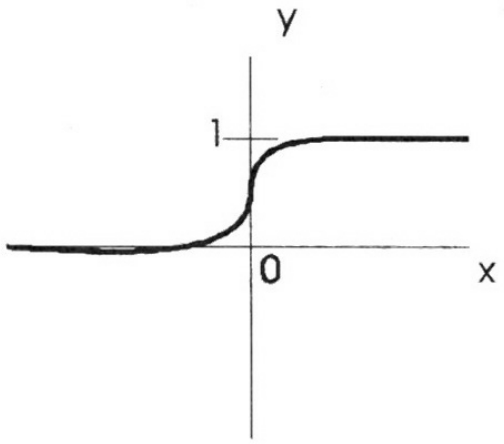

figura 3.5: Función de activación sigmoidal.

3.2.4. Regla de aprendizaje del Perceptrón

El algoritmo de aprendizaje del Perceptrón es de tipo supervisado, es decir que por cada ciclo de entrenamiento al modelo se le presenta un patron de entrada y un patron de salida (deseado). Para que el modelo pueda asociar los patrones de entrada con los de salida se necesita de una regla de aprendizaje cuyo funcionamiento consiste en modificar los pesos del perceptrón para ajustar la frontera de decisión con la separación de las clases, es decir, que si los patrones son linealmente separables, el perceptrón puede encontrar la solución.

Para el caso del perceptrón es común encontrar la siguiente regla de aprendizaje:

$$\begin{aligned} w_i(t+1) &= w_i(t) + \alpha[e(t)x_i(t)] \\ e(t) &= d(t) - y(t) \\ (0 &\leq i \leq N) \end{aligned} \tag{3.4}$$

donde
t = número de época.
$d(t)$ = valor deseado de salida.
$y(t)$ = valor de salida de la función de activación.
$e(t)$ = error.
$x_i(t)$ = elemento i del patrón de entrada.
$w_i(t)$ = peso correspondiente al elemento i del patrón de entrada.

α es conocido como el factor de aprendizaje, es ajustado experimentalmente en un intervalo sugerido de $(0, 0{,}25)$.

Con esta regla de aprendizaje el perceptrón es capaz de ubicar la frontera de decisión en cualquier punto del espacio de soluciones, la frontera de decisión en un ciclo completo de entrenamiento se va a modificar t veces, siendo cada t una época. Si en una época el error es 0, resulta evidente que el peso no va a ser modificado. El factor de aprendizaje α es un parametro de velocidad de aprendizaje, si el valor α es muy pequeño es posible que el entrenamiento necesite un gran número de épocas, por otro lado, si α es muy grande aumentan las probabilidades de no convergencia.

A continuación se presenta el algoritmo de convergencia de ajuste de pesos para realizar el aprendizaje de un Perceptrón con N elementos de entrada y un único elemento procesal de salida:

1. Inicialización de los pesos y del umbral

   Inicialmente se asignan valores aleatorios pequeños a cada uno de los pesos w_i y al umbral $w_0 = \theta$.

2. Presentación de un nuevo par (Entrada, Salida esperada)

   Presentar un nuevo patrón de entrada $X^p = (x_1, x_2, ..., x_N)$ junto con la salida esperada $d(t)$.

3. Cálculo de la salida actual

$$\begin{aligned} net &= \sum_i w_i(t)x_i(t) - \theta \\ y(t) &= f(net) \end{aligned}$$

   donde net es el valor de integración y $y(t)$ es la función de activación.

4. Adaptación de los pesos

$$w_i(t+1) = w_i(t) + \alpha[e(t)x_i(t)]$$

5. Volver al paso 2

   Este proceso se repite hasta que el error que se produce para cada uno de los patrones es 0 o bien menor que un valor preestablecido.

Ejemplo

Para un patrón como el siguiente,

Patrón	x_2	x_1	OR
1	0	0	0
2	1	0	1
3	0	1	1
4	1	1	1

utilizando la funcion net de la Ecuación 3.1 y y es el valor de la función de activacion escalón

acotada entre $[0, 1]$ descrita en la Figura 3.4 a), cuya regla de aprendizaje hace referencia a la Ecuacion 3.4 se obtiene,

Patrón	w_1	w_2	θ	net	y	$error$
1	0.1	-0.2	0.3	0.3	1	-1
2	0.1	0.8	-0.7	-0.9	0	1
3	0.1	0.8	0.3	0.4	1	0
4	0.1	0.8	0.3	1.2	1	0
1	0.1	0.8	-0.7	0.3	1	-1
2	0.1	0.8	-0.7	0.1	1	0
3	1.1	0.8	0.3	-0.6	0	1
4	1.1	0.8	0.3	2.2	1	0
1	1.1	0.8	-0.7	0.3	1	-1
2	1.1	0.8	-0.7	0.1	1	0
3	1.1	0.8	-0.7	0.4	1	0
4	1.1	0.8	-0.7	1.2	1	0
1	1.1	0.8	-0.7	-0.7	0	0

Fronteras de decisión para $-0{,}5 < X_2 < 1{,}5$

$$a) X_1 = -\frac{-0{,}2}{0{,}1} * X_2 - \frac{0{,}3}{0{,}1} \quad b) X_1 = -\frac{-0{,}2}{0{,}1} * X_2 - \frac{-0{,}7}{0{,}1} \quad c) X_1 = -\frac{0{,}8}{0{,}1} * X_2 - \frac{-0{,}3}{0{,}1}$$
$$d) X_1 = -\frac{0{,}8}{0{,}1} * X_2 - \frac{-0{,}7}{0{,}1} \quad e) X_1 = -\frac{0{,}8}{1{,}1} * X_2 - \frac{0{,}3}{1{,}1} \quad f) X_1 = -\frac{0{,}8}{1{,}1} * X_2 - \frac{-0{,}7}{1{,}1}$$

A continuación en la Figura 3.6 se muestran las gráficas de las fronteras de decisión según el caso,

3.2.5. Limitaciones del perceptrón simple

En 1969 Minsky y Papert publicarón un libro [3], donde señalaron las grandes limitaciones de las capacidades del perceptrón. Mostraron que por ejemplo, el perceptrón no puede resolver el OR Exclusivo. De hecho, hay un gran número de problemas de clasificación que

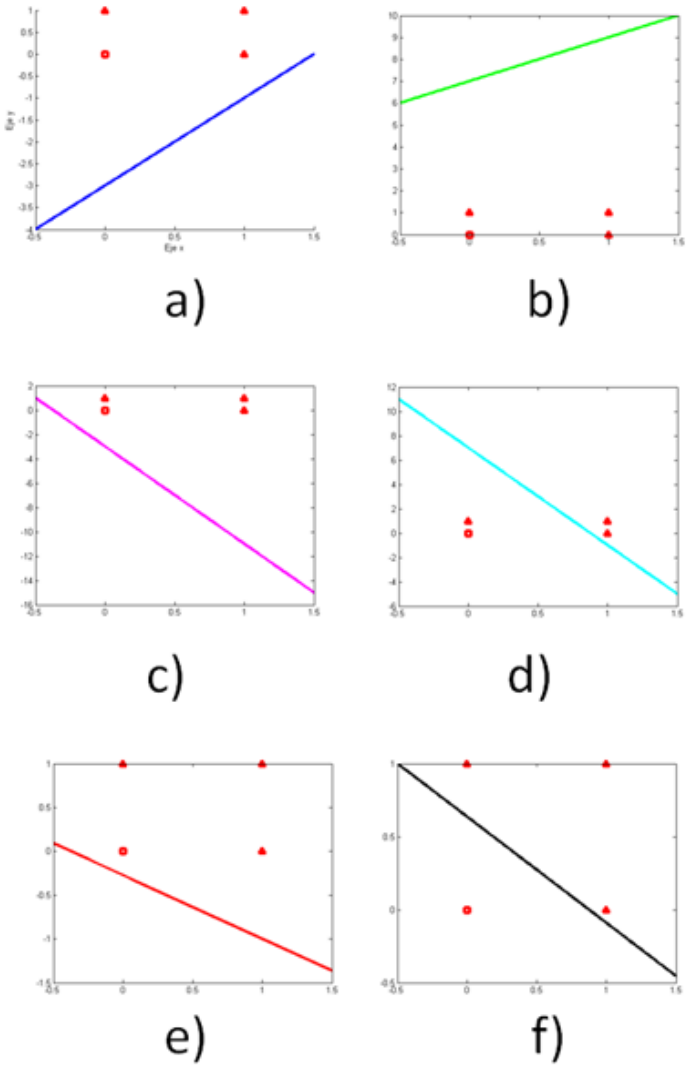

figura 3.6: Gráfica de la fronteras de decisión. a) Datos iniciales, b) 1er cambio, c) 2do cambio, d) 3er cambio, e) 4to cambio, f) 5to cambio, g) 6to cambio. Notese que los cambios son extensos debido a que $\alpha = 1$.

el perceptrón simple no puede resolver. De modo que para un perceptrón simple con un número extenso de entradas, el número de problemas que puede clasificar es una fracción de la totalidad de los problemas que pueden ser formulados.

Específicamente, una neurona con entradas binarias puede tener 2^n diferentes patrones de entrada. Como cada patron de entrada puede producir 2 salidas binarias diferentes, por lo tanto 2^{2^n} funciones diferentes de n variables. El número de problemas linealmente separables de n entradas binarias es una pequeña fracción de 2^{2^n} como se muestra en la Tabla 3.1, [4].

No. de entradas n	2^{2^n}	No. de problemas linealmente separables
1	4	4
2	16	14 (todas excepto XOR, XNOR)
3	256	104
4	65 K	1.9 K
5	4.3 x 10^9	95 K
.	.	.
.	.	.
.	.	.
$n > 7$	x	$< x^{1/3}$

Tabla 3.1: Número de problemas binarios linealmente separables. (Basado en P. P. Wasserman: Neural Computing Theory and Practice 1989 International Thomson Computer Press.) [4].

3.3. El perceptrón multicapa (MLP)

Para superar las limitaciones señaladas por Minsky y Paper, que detuvieron de forma casi total la investigación en las redes neuronales fue necesario ir más allá de las redes de una sola capa [4].

Un perceptrón multicapa Figura (3.7) es una red compuesta de varias capas de neuronas entre la entrada y salida de la misma. Esta red permite establecer regiones de decisión mucho más complejas que las de dos semiplanos, como hace un perceptrón de dos capas.

Las capacidades del perceptrón con dos, tres y cuatro capas y con una única neurona en la capa de salida, se ilustran en la Figura 3.8. En la segunda columna se muestra el tipo de región de decisión que se puede formar con cada una de las configuraciones. En la siguiente columna se indica el tipo de región de decisión que se formaría para el problema del XOR. En las dos últimas columnas se muestran las regiones formadas para resolver el problema de clases con regiones mezcladas y las formas de regiones más generales para cada uno de los casos.

El perceptrón básico de dos capas sólo puede establecer dos regiones separadas por una frontera lineal en el espacio de patrones de entrada. Un Perceptrón con tres niveles de neuronas puede formar cualquier región convexa en este espacio. Las regiones convexas se forman

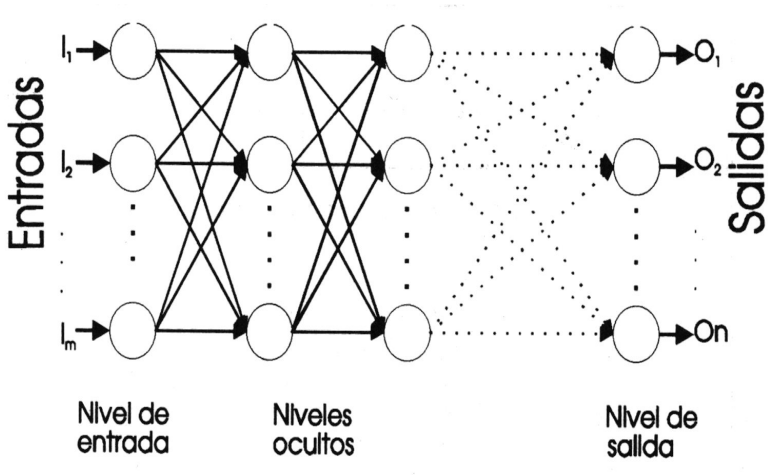

figura 3.7: Esquema del perceptrón multicapa.

mediante la intersección entre las regiones formadas por cada neurona de la segunda capa. Cada uno de estos elementos se comporta como un perceptrón simple, activándose su salida para los patrones de un lado del hiperplano.

Este análisis nos introduce en el problema de la selección del número de neuronas ocultas de un Perceptrón de tres capas. En general, este número deberá ser lo suficientemente grande como para que se forme una región lo suficientemente compleja para la resolución del problema. Sin embargo, tampoco es conveniente que el número de nodos sea tan grande que la estimación de los pesos no sea fiable para el conjunto de patrones de entrada disponibles.

3.3.1. Red con alimentación hacia atras (backpropagation)

El método de retropropagación del error, también conocido como del gradiente descendente y conocido en general como el método backpropagation (propagación del error hacia atrás), está basado en la generalización de la regla delta y, a pesar de sus propias limitaciones, ha ampliado de forma considerable el rango de aplicaciones de las redes neuronales.

A esta arquitectura de red neuronal artificial se le conoce también como red con conexiones hacia adelante, en la que todas las señales neuronales se propagan hacia la siguiente capa hasta la salida de la red, es decir que no existen conexiones hacia las capas anteriores ni conexiones entre las neuronas de la misma capa.

En 1986, Rumelhart, Hinton y Williams [3], basándose en los trabajos de otros investigadores [5] y [6] formalizaron un método para que una red neuronal aprendiera la asociación que existe entre los patrones de entrada y las clases correspondientes, utilizando más niveles de neuronas que los que utilizó Rosenblatt para desarrollar el perceptrón.

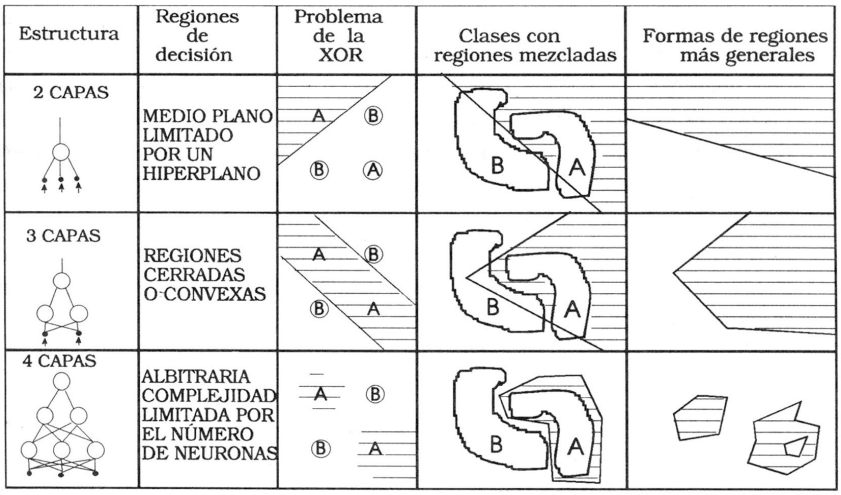

figura 3.8: Distintas formas de regiones generadas por un perceptrón multicapa. [2]

El algoritmo de propagación hacia atrás, es una regla de aprendizaje que se puede aplicar en modelos de redes con más de dos capas de neuronas. Una característica importante de este algoritmo es la representación interna del conocimiento que es capaz de organizar en la capa intermedia de las neuronas para conseguir cualquier correspondencia entre la entrada y la salida de la red.

3.3.2. Algoritmo de aprendizaje

Paso 1

Inicializar los pesos de la red con valores pequeños aleatorios.

Paso 2

Presentar un patrón de entrada $X_p : [x_{p1}, x_{p2}, ..., x_{pN}]$, y especificar la salida deseada que debe generar la red: $[d_1, d_2, ..., d_M]$, (si la red se utiliza como un clasificador, todas las salidas deseadas serán cero, salvo una que será la de la clase a la que pertenece el patrón de entrada).

Paso 3

Calcular la salida actual de la red, para ello se presentan las entradas a la red y vamos calculando la salida que presenta cada capa hasta llegar a la capa de salida, ésta será la salida de la red $[y_1, y_2, ..., y_M]$ como se muestra a continuación.

Se calculan las entradas netas para las neuronas ocultas procedentes de las neuronas de entrada.

Para una neurona j oculta:

$$net^h_{pj} = \sum_{i=1}^{N} w^h_{ji} x_{pi} + \theta^h_j, \tag{3.5}$$

en donde el índice h se refiere a la magnitud desde la capa oculta (*hidden*); el subíndice p, al *p-ésimo* vector de entrenamiento y j a la *j-ésima* neurona oculta. El término θ puede ser opcional, pues actua como una entrada más.

-Se calculan las salidas de las neuronas ocultas:

$$y_{pj} = f_j^h(net_{pj}^h) \tag{3.6}$$

-Se realizan los mismos cálculos para obtener las salidas de las neuronas de salida (capa o: *output*)

$$net_{pk}^o = \sum_{j=1}^{L} w_{kj}^o y_{pj} + \theta_k^o, \tag{3.7}$$

$$y_{pk} = f_k^o(net_{pk}^o) \tag{3.8}$$

Paso 4
Calcular los términos de error para todas las neuronas.

Si la neurona k es una neurona de la capa de salida, el valor de la delta es,

$$\delta_{pk}^o = (d_{pk} - y_{pk}) f_k^o(net_{pk}^o), \tag{3.9}$$

donde δ_{pk}^o es el error que se produce en la capa de salida y d_{pk} es el error de que se produce en una neurona.

La función f, como se citó anteriormente, debe cumplir el requisito de ser derivable. En general, disponemos de dos formas de función de salida que nos pueden servir, la función lineal de salida ($f_k(net_{jk}) = net_{jk}$) y la función sigmoidal representada en la Figura 3.5 y definida por la expresión,

$$f_k(net_{jk}) = \frac{1}{1 + e^{-net_{jk}}}. \tag{3.10}$$

La selección de la función de salida depende de la forma en que se decida representar los datos de salida. Si se desea que las neuronas de salida sean binarias, se utiliza la función sigmoidal, puesto que esta función es casi biestable y, además, derivable. En otros casos es tan aplicable una función como otra.

Para la función lineal, tenemos: $f_k^o = 1$, mientras que la derivada de una función f sigmoidal es,

$$f_k^{o\prime} = f_k^o(1 - f_k^o) = y_{pk}(1 - y_{pk}), \tag{3.11}$$

por lo que los términos de error para las neuronas de salida quedan,

$$\delta_{pk}^o = (d_{pk} - y_{pk}), \tag{3.12}$$

para la salida lineal, y

$$\delta_{pk}^{o} = (d_{pk} - y_{pk})y_{pk}(1 - y_{pk}), \tag{3.13}$$

para la salida sigmoidal. Donde $f_k^o = y(x)$ y

$$\begin{aligned}
y(x) &= \frac{1}{1+e^{-x}} \\
y'(x) &= \frac{d(1+e^{-x})^{-1}}{dx} \\
y'(x) &= -1(1+e^{-x})^{-2} \cdot \frac{d(1+e^{-x})}{dx} \\
y'(x) &= \frac{-1}{(1+e^{-x})^2} \cdot e^{-x} \cdot \frac{d(-x)}{dx} \\
y'(x) &= \frac{1}{1+e^{-x}} \left(\frac{e^{-x}}{(1+e^{-x})} \right) \\
y'(x) &= \frac{1}{1+e^{-x}} \left(1 - \frac{1}{1+e^{-x}} \right) \\
y'(x) &= y(x)(1 - y(x))
\end{aligned}$$

Si la neurona j no es de salida, entonces la derivada parcial del error no puede ser evaluada directamente. Por tanto, se obtiene el desarrollo a partir de valores que son conocidos y otros que pueden ser evaluados.

La expresión obtenida en este caso es:

$$\delta_{pj}^{h} = f_k^h(net_{pj}^h) \sum_k \delta_{pk}^o w_{kj}^o, \tag{3.14}$$

donde observamos que el error en las capas ocultas depende de todos los términos de error de la capa de salida. De aquí surge el término de propagación hacia atrás. En particular, para la función sigmoidal:

$$\delta_{pj}^{h} = x_{pi}(1 - x_{pi}) \sum_k \delta_{pk}^o w_{kj}^o, \tag{3.15}$$

donde k se refiere a todas las neuronas de la capa superior a la de la neurona j. Así, el error que se produce en una neurona oculta es proporcional a la suma de los errores conocidos que se producen en las neuronas a las que está conectada la salida de ésta, multiplicado cada uno de ellos por el peso de la conexión. Los umbrales internos de las neuronas se adaptan de forma similar, considerando que están conectados con pesos desde entradas auxiliares de valor constante.

Paso 5

Actualización de los pesos

Para ello, utilizamos el algoritmo recursivo, comenzando por las neuronas de salida y trabajando hacia atrás hasta llegar a la capa de entrada, ajustando los pesos de la forma siguiente:

Para los pesos de las neuronas de la capa de salida,

$$w_{kj}^{o}(t+1) = w_{kj}^{o}(t) + \Delta w_{kj}^{o}(t+1), \tag{3.16}$$

$$\Delta w_{kj}^{o}(t+1) = \alpha \delta_{pk}^{o} y_{pj}; \tag{3.17}$$

y para los pesos de las neuronas de la capa oculta,

$$w_{ji}^{h}(t+1) = w_{ji}^{h}(t) + \Delta w_{ji}^{h}(t+1), \tag{3.18}$$

$$\Delta w_{ji}^{h}(t+1) = \alpha \delta_{ji}^{h} x_{pi}; \tag{3.19}$$

En ambos casos, para acelerar el proceso de aprendizaje, se puede añadir un término momento de valor $\beta(w_{kj}^{o}(t) - w_{kj}^{o}(t-1))$ en el caso de la neurona de salida y $\beta(w_{ji}^{h}(t) - w_{ji}^{h}(t-1))$ cuando se trata de una neurona oculta.

Paso 6

El proceso se repite hasta que el término de error

$$E_p = \frac{1}{2} \sum_{k=1}^{M} \delta_{pk}^{2}, \tag{3.20}$$

resulta aceptablemente pequeño para cada uno de los patrones aprendidos.

Consideraciones sobre el algoritmo de aprendizaje

El algoritmo de backpropagation encuentra un valor mínimo de error (local o global) mediante la aplicación de pasos descendentes (gradiente descendente). Cada punto de la superficie de la función de error correspondiente a un conjunto de valores de los pesos de la red. Con el gradiente descendente, siempre que se realiza un cambio en todos los pesos de la red, se asegura el descenso por la superficie del error hasta encontrar el valle más cercano, lo que puede hacer que el proceso de aprendizaje se detenga en un mínimo local de error.

Por tanto, uno de los problemas que presenta este algoritmo de entrenamiento de redes multicapa es que busca minimizar la función de error, pudiendo caer en un mínimo local o en algún punto estacionario, con lo cual no se llega a encontrar el mínimo global de la función del error. Sin embargo, debe tenerse en cuenta que no tiene por que alcanzarse el mínimo global en todas las aplicaciones, sino que puede ser suficiente con un error mínimo preestablecido.

Control de la convergencia

En las técnicas de gradiente decreciente es conveniente avanzar por la superficie de error con incrementos pequeños de los pesos. Esto se debe a que tenemos una información local de la superficie y no se sabe lo lejos o lo cerca que se está del punto mínimo. Con incrementos grandes, se corre el riesgo de pasar por encima del punto mínimo sin conseguir estacionarse en él. Con incrementos pequeños, aunque se tarde más en llegar, se evita que ocurra esto.

El elegir un incremento de paso adecuado influye en la velocidad con la que converge el algoritmo. Esta velocidad se controla a través de la constante de proporcionalidad o tasa de aprendizaje α. Normalmente, α debe ser un número pequeño (del orden de 0.05 a 0.25), para asegurar que la red llegue a asentarse en una solución. Un valor pequeño de α significa que la red tendrá que hacer un gran número de iteraciones. Si esa constante es muy grande, los cambios de pesos son muy grandes, avanzando muy rápidamente por la superficie de error, con el riesgo de saltar el mínimo y estar oscilando alrededor de él, pero sin poder alcanzarlo.

Lo habitual es aumentar el valor de α a medida que disminuye el error de la red durante la fase de aprendizaje. Así, aceleramos la convergencia aunque sin llegar nunca a valores de α demasiado grandes, que hicieran que la red oscilase alejándose demasiado del valor mínimo. Otra forma de incrementar la velocidad de convergencia consiste en añadir un término momento que consistente en sumar una fracción del anterior cambio cuando se calcula el valor del cambio de peso actual. Este término adicional tiende a mantener los cambios de peso en la misma direccción.

Un último aspecto a tener en cuenta es la posibilidad de convergencia hacia alguno de los mínimos locales que pueden existir en la superficie de error del espacio de pesos. En el desarrollo matemático que se ha realizado para llegar al algoritmo de retropropagación, no se asegura en ningun momento que el mínimo que se encuentre sea global. Una vez que la red se asienta en un mínimo, sea local o global, cesa el aprendizaje, aunque el error siga siendo demasiado alto, si se ha alcanzado un mínimo local. En todo los casos si la solución es admisible desde el punto de vista del error, no importa si el mínimo es local o global o si se ha detenido en algún momento previo a alcanzar un verdadero mínimo.

En la práctica, si una red deja de aprender antes de llegar a una solución aceptable, se realiza un cambio en el número de neuronas ocultas o en los parámetros de aprendizaje o, simplemente, se vuelve a empezar con un conjunto distinto de pesos originales y se suele resolver el problema.

3.3.3. La regla delta generalizada

La regla propuesta por Widrow en 1960 (regla delta) ha sido extendida a redes con capas intermedias (regla delta generalizada) con conexiones hacia adelante (feedforward) y cuyas células tienen funciones de activación continua (lineales o sigmoidales), dando lugar al algoritmo de retropropagación (backpropagation). Estas funciones continuas son no decrecientes y

derivables. La función sigmoidal pertenece a este tipo de funciones, a diferencia de la función escalón que se utiliza en el perceptrón.

Este algoritmo utiliza también una función o superficie de error asociado a la red, buscando el estado estable de mínima energía o de mínimo a través del camino descendente de la superficie del error. Por ello, realimenta el error del sistema para realizar la modificación de los pesos en un valor proporcional al gradiente decreciente de dicha función de error.

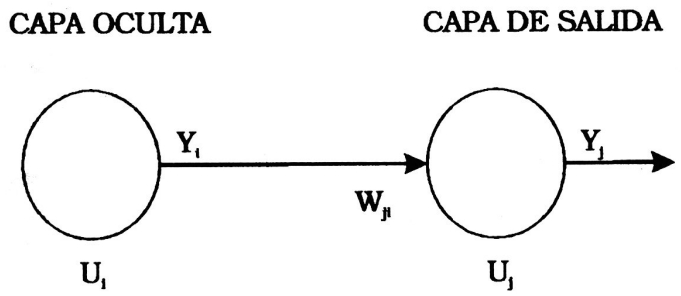

figura 3.9: Conexión entre una neurona de una capa oculta con una neurona de salida

La regla delta generalizada para ajustar los pesos es exactamente la misma que la regla delta utilizada en el Perceptrón; es decir, los pesos se actualizan de forma proporcional a la delta, o diferencia entre la salida deseada y la obtenida (δ = sal. deseada - sal. obtenida).

Dada una neurona (unidad U_i) y la salida que produce (Fig. 3.9), el cambio que se produce en el peso de la conexión que une la salida de dicha neurona con la unidad U_i (w_{ji}) para un patrón de aprendizaje p determinado es:

$$\Delta w_{ji}(t+1) = \alpha \delta_{pj} y_{pi}$$

En donde el subíndice p se refiere al patrón de aprendizaje concreto y α es la constante o tasa de aprendizaje.

El punto en el que difieren la regla delta generalizada de la regla delta es en el valor concreto de δ. Por otro lado, en las redes multinivel, a diferencia de las redes sin neuronas ocultas, en principio no se puede conocer la salida deseada de las neuronas de las capas ocultas para poder determinar los pesos en función del error cometido. Sin embargo, inicialmente sí podemos conocer la salida deseada de las neuronas de salida, entonces definimos,

$$\delta_{pj} = (d_{pj} - y_{pj}) \cdot f'(net_j)$$

donde d_{pj} es la salida deseada de la neurona j para el patrón p y net_j es la entrada neta que recibe la neurona j.

Esta fórmula es como la de la regla delta, excepto en lo que se refiere a la derivada de la función de transferencia. Este término representa la modificación que hay que realizar en la

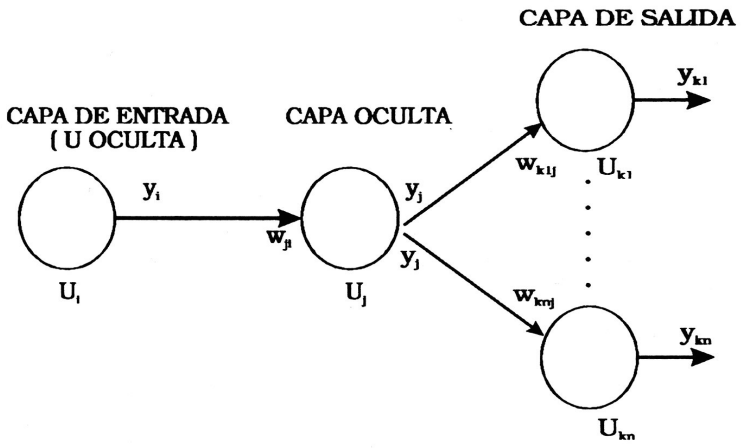

figura 3.10: Conexiones entre neuronas de la capa oculta con la capa de salida

entrada que recibe la neurona j. En el caso de que dicha neurona no sea de salida, el error que se produce estará en función del error que se cometa en las neuronas que reciban como entrada la salida de dicha neurona. Esto es lo que se denomina procedimiento de propagación del error hacia atrás.

Según esto, en el caso de que U_j *no sea una neurona de salida* ver Figura 3.10, el error que se produce está en función del error que se comete en las neuronas que reciben como entrada la salida de U_i:

$$\delta_{pj} = (\sum_k \delta_{pk} w_{ij}) \cdot f'(net_j)$$

donde el rango de k cubre todas aquellas neuronas a las que está conectada la salida de U_i. De esta forma, el error que se produce en una neurona oculta es la suma de los errores que se producen en las neuronas a las que está conectada la salida de ésta, multiplicando cada uno de ellos por el peso de la conexión.

3.3.4. Adición de un momento en la regla delta generalizada

El método de retropropagación del error, requiere un importante número de cálculos para lograr el ajuste de los pesos de la red. En la implementación del algoritmo, se toma una amplitud de paso que viene dada por la tasa de aprendizaje α. A mayor tasa de aprendizaje, mayor es la modificación de los pesos en cada iteración, con lo que el aprendizaje será más rápido, pero, por otro lado, puede dar lugar a oscilaciones. [8][9][10] En la literatura se sugiere que para filtrar estas oscilaciones se añada en la expresión del incremento de los pesos un término (momento), β, de manera que dicha expresión quede:

$$w_{ij}(t+1) = w_{ij}(t) + \alpha\delta_{pj}y_{pi} + \beta(w_{ij}(t) - w_{ji}(t-1) = \Delta w_{ji}(t+1) = \alpha\delta_{pj}y_{pi} + \beta\Delta w_{ji}(t)$$

donde β es una constante (momento) que determina el efecto en $t+1$ del cambio de los pesos en el instante t.

Con este momento se consigue la convergencia de la red en menor número de iteraciones, ya que si en t el incremento de un peso era positivo y en $N-1$ también, entonces el descenso por la superficie de error en $t+1$ es mayor. Sin embargo, si en t el incremento era positivo y en $N-1$ es negativo, el paso que se da en $t+1$ es más pequeño, lo cual es adecuado, ya que eso significa que se ha pasado por un mínimo y que los pasos deben ser menores para alcanzarlo.

Resumiendo, el algoritmo backpropagation queda finalmente:

$$\begin{aligned} w_{ji}(t+1) &= w_{ji}(t) + [\Delta w_{ji}(t+1)] \\ w_{ji}(t+1) &= w_{ji}(t) + [\alpha \delta_{pj} y_{pi} + \beta \Delta w_{ji}(t)] \end{aligned}$$

donde:

$$\delta_{pj} = (d_{pj} - y_{pi}) f'(net_j)$$

si U_j es una neurona de salida y

$$\delta_{pj} = (\sum_k \delta_{pk} w_{kj}) f'(net_j)$$

si U_j no es una neurona de salida.

3.4. Conclusiones

En este capítulo se estudiaron las técnicas, métodos y algoritmos de dos tipos de perceptrón, el simple y el multicapa. El perceptrón simple es un modelo simple de neurona con una regla de aprendizaje basada en la corrección del error y tiene la capacidad de aprender a reconocer patrones, está constituido por un conjunto de sensores que reciben los patrones de entrada a reconocer o clasificar y una neurona de salida que se ocupa de clasificar a los patrones de entrada en dos clases, pero a pesar de estas características tiene limitantes, por ejemplo, si se le alimenta con un número extenso de entradas, el número de problemas que puede clasificar es una fracción de la totalidad de los problemas que pueden ser formulados, por esta razón el perceptrón multicapa se convierte en una herramienta más efectiva, como su nombre lo dice el perceptrón multicapa es una red neuronal artificial (RNA) formada por múltiples capas, esto le permite resolver problemas que no son linealmente separables, lo cual es la principal limitación del perceptrón simple, y aunque su tiempo de aprendizaje es bastante lento, el perceptrón multicapa es realmente efectivo cuando las instancias están representadas por muchos pares atributo-valor.

Bibliografía

[1] H. Hu, J. Hwang, "Handbook of Neural Network Signal Processing", *CRC PRESS* (2002).

[2] R. Hilera, J. Martínez, Redes Neuronales Artificiales Fundamentos, Modelos y Aplicaciones", *Addison Wesley Iberoamericana* Alafaomega ra-ma (2000).

[3] Minsky, Papert, "Perceptrons"*MIT Press* (1969).

[4] D. Graupe, "Principles of Artificial Neural Networks"2nd Edition, *World Scientific* (2007).

[5] J. Werbos. "Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences"*PhD thesis*, Harvard University (1974)

[6] E. Parker, "Notes on Multilayer, Feedforward Neural Networks"*Projects in Machine Learning* (2006)

[7] B. Widrow and S. D. Stearns, .ªdaptive Signal Processing"*New Jersey: Prentice-Hall, Inc.,* (1985).

[8] D. Rumelhart, J. McClelland, "Parallel Distributed Processing: Explorations in the Microstructure of Cognition"(1986)

[9] D. Joksimović "Neural networks and geographic information systems"(1993)

[10] Williams, "Evolutionary artificial neural networks, (1986).

Capítulo 4

Sistemas de Adquisicion de imágenes para la generación de la base de datos del Lenguaje de Señas UPT.

4.1. Introducción

En este capítulo se describen dos sistemas óptico-digitales para la adquisición de las imágenes que conforman las bases de datos del lenguaje de señas. En la sección 2 se describe el funcionamiento de una cámara térmica así como la radiación que detecta, cercana a las 10μm, que es una longitud de onda cercana a la que irradia el cuerpo humano a través del calor. Con la cámara térmica se generó la primera base de digitos del 1 al 9. El rango de temperatura se ajusta automáticamente por la cámara y el desplegado de la imagen es con mapas de color arcoiris. También se describe el preprocesamiento de la información en el espacio de color HSI (Tono, Saturación, Intensidad) para segmentación de la región de la mano.

La cámara usada para la adquisición de imágenes en el visible es descrita en la sección 4. En la sección 5 se describe un sistema de dos cámaras, visible e infrarroja, para la adquisición de la base de datos de los signos del alfabeto. Con el sistema propuesto se segmenta la información de las imágenes de la cámara visible con las imágenes de la cámara térmica, para así obtener sólo la región de la mano. En la sección 5 se muestran las series de imágenes adquiridas y preprocesadas que conforman las dos bases de datos finales: la base de A) Digitos y B) Alfabeto del lenguaje de señas. Las conclusiones de este capítulo se encuentran al final del mismo.

4.2. Cámara FLIR Thermacam P65

Tal como se muestra en la Figura 4.1, "la energía infrarroja a) que proviene de un objeto se hace converger, por medio de un sistema óptico b), sobre un detector de infrarrojo c). El detector envía la información a un arreglo de sensores d) para que se registre la imagen. El conjunto de tarjetas electrónicas utiliza los datos que provienen del detector para crear una imagen e) que puede verse en el visor o en un monitor de vídeo o pantalla LCD. La termografía infrarroja es el arte de transformar una imagen infrarroja en radiométrica, lo que permite leer los valores de temperatura a partir de la imagen. Para hacerlo, la cámara infrarroja cuenta con algoritmos complejos"[1]. [1]

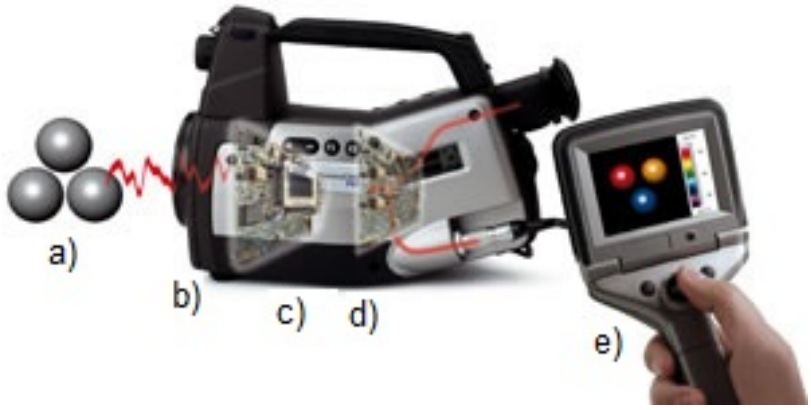

figura 4.1: Funcionamiento de una cámara térmica.

4.2.1. Especificaciones Técnicas

En el laboratorio de Óptica y Sistemas de Visión de la Universidad Politécnica de Tulancingo se cuenta con una cámara FLIR Thermacam P65, Figura 4.2, para la adquisición de imágenes térmicas en formatos JPEG. En la tabla 4.1 se enumeran sus caraterísticas técnicas [1]. La idea de utilizar esta cámara térmica, es la de captar la radiación que emite la temperatura del cuerpo humano. Si bien existen muchas cámaras de este tipo en el mercado, ya sea de mayor o menor costo y calidad, esta cámara se encuentra a disponibilidad dentro de la universidad, pero cualquier cámara térmica que cumpla con los requisitos de sensar el espectro térmico del cuerpo humano, la conexión a una computadora y de preferencia, la utilización de diferentes mapas de color que permitan la rápida segmentación puede funcionar.

figura 4.2: Cámara FLIR Thermacam P65

	Cámara FLIR Thermacam P65
1	Rango de visión termal desde -40°C hasta+2,000°C
2	Visión del objeto tanto en el espectro IR como en el visible
3	Rango Espectral de 7500nm a 13000nm
4	Almacenamiento de imágenes radiométricas en formato JPEG
5	Calidad de imagen de 320 X 240 a 640 X 480 píxeles
6	Autoenfoque
7	Salida NTSC/PAL IEEE - 1394 FireWire DV-output

Tabla 4.1: Ficha Técnica de la cámara térmica FLIR Thermacam P65

4.2.2. Mapas de color de la cámara térmica

Las cámaras que captan fotografía o videos de luz infrarroja (cámaras térmicas), no captan las señales en el espectro visible, sino que más bien tienden a ser imágenes monocromáticas, porque utilizan un sólo tipo de sensor que percibe un rango de longitudes de onda en particular, donde las areás más calientes de un cuerpo en color blanco y las más frías en color negro, con matices grises entre los grados de temperatura intermedios entre los límites térmicos. Sólo aparecen fotografías en tonos de color cuando estas imágenes son iluminadas con colores predominates típicos de nuestra visión, estos colores se asignan a los píxeles de manera indirecta, haciendo uso de una tabla llamada *mapa de colores*. La temperatura humana promedio es de aproximadamente 37°C, [2] la curva de máxima emisión en el rango espectral se encuentra en el infrarrojo cercano con una longitud de onda cercana a las 10 micras (10,000 nm). En la Figura 4.3 se muestran algunas imágenes adquiridas con la cámara térmica, que corresponden a a) Espectro visible,

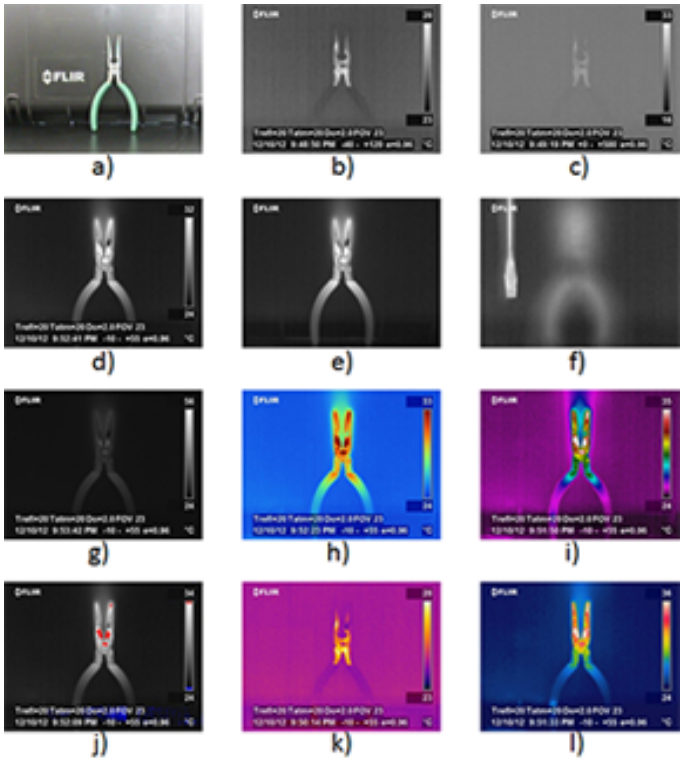

figura 4.3: Imágenes de una pinza tomadas por la cámara térmica FLIR Thermacam P65.

b) con rango entre 23 y 28 $°C$, c) con rango entre 16 y 33 $°C$, d) con display, e)sin display, f) enfocando un destornillador en un plano cercano, g) con el mapa de color escala de grises, h) con el mapa de color arcoiris, i) con el mapa de color arcoiris hc, j) con el mapa de color saturación, k) con el mapa de color blue-red blended, l) con el mapa de color hierro.

Los mapas de color son elegidos manualmente, el rango de temperaturas para sensar puede ser automático, donde el objeto más caliente de la escena es el máximo, o bien puede sensar de forma manual, manteniendo la temperatura a la que sea ajustada, también posee puertos de salidas para los datos como el firewire 1394, tarjeta flash y vcr, puede guardar imágenes o videos e inclusive etiquetas de voz.

4.2.3. Interfaz para la adquisición de imágenes de la cámara térmica

La programación de una interfaz para la comunicación dinámica entre la computadora y la cámara térmica fue de gran ayuda, los sistemas FLIR son soportados por el sistema operativo Windows y por consecuencia es posible que MATLAB interactúe con este dispositivo. MATLAB es un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio (lenguaje M) [3], sobre el cual se diseñó y desarrolló esta interfaz (Figura 4.4), capaz de capturar y procesar las imágenes de la cámara térmica en tiempo real.

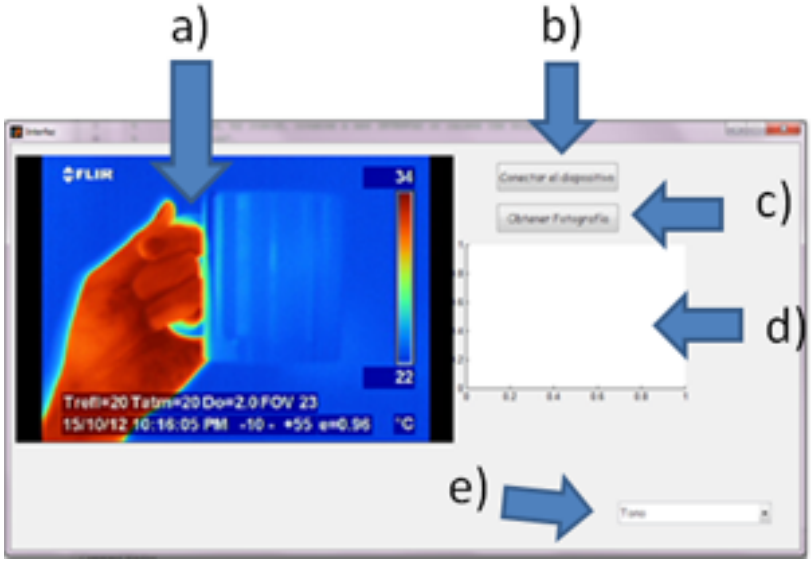

figura 4.4: Aplicación programada en MATLAB para la adquisición de imágenes.

En la Figura 4.4, se muestran los componentes de la interfaz, donde a) es un cuadro de desplegado que muestra las imágenes de la cámara térmica en tiempo real, b) es un botón que sirve para establecer y parametrizar la conexión con la cámara térmica, c) es un botón que permite obtener y guardar la imagen del cuadro de desplegado a), d) es un cuadro de desplegado más pequeño en el cual se puede ver la última imagen capturada y e) es una caja de combinaciones que permite modificar las imágenes que se muestran del cuadro de desplegado a) aplicando transformaciones en el espacio de color pudiendo convertir la imagen

RGB que provee la cámara a solo R,G o B o también transformarla a HSI y mostrar solo H,S o I; además de otras transformaciones espaciales como el binarizado y el filtrado con sobel o canny. Las imágenes capturadas por esta aplicación son guardadas en la misma carpeta donde se ubica la aplicación y bajo el nombre de una numeración progresiva.

4.3. Cámara HITACHI KP-F120CL

4.3.1. Especificaciones Técnicas

En el laboratorio de óptica y sistemas de visión, también se cuenta con una cámara HITACHI, Figura 4.5, para la adquisición de imágenes en formatos JPEG. En la tabla 4.2 se enumeran sus caraterísticas técnicas en el visible [4]

figura 4.5: Cámara HITACHI KP-F120CL

	Cámara HITACHI KP-F120CL
1	Rango Espectral de 400nm a 1000nm (Figura 4.6)
2	Tomas de 15, 30, 60 o 120 cuadros/seg.
3	Visión del objeto tanto en el espectro IR como en el visible
4	Calidad de imagen de 1392 (H) x 1040 (V) píxeles
5	Salida Canal único de 8 bit

Tabla 4.2: Ficha técnica de la cámara HITACHI KP-F120CL

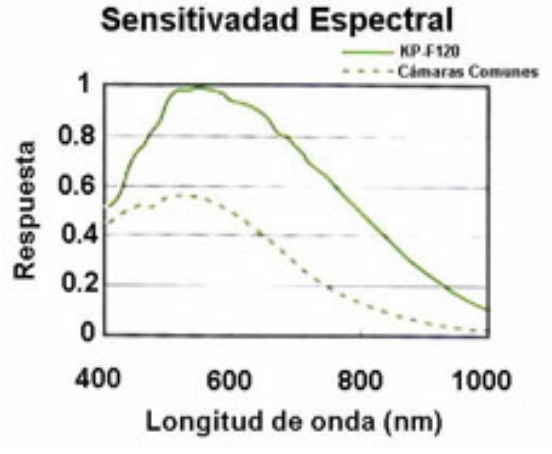

figura 4.6: Gráfica de la sensitividad espectral de la cámara HITACHI KP-F120

4.3.2. Lente zoom de la cámara visible

La lente zoom que se utilizó para la adquisición de imágenes con la cámara HITACHI KP-F120CL fue una lente para televisión Edmund Optics 6X Manual Zoom Video Lens (8-48 mm FL) de la Figura 4.7 cuyas características se enumeran en la tabla 4.3 [5]

figura 4.7: Lente para televisión Edmund Optics 6X Manual Zoom Video Lens (8-48 mm FL).

	Lente zoom Edmund optics 6X Manual Zoom Video Lens (8-48 mm FL) Modelo NT53-152
1	Amplificación Primaria PMAG 6X
2	Formato de sensor máxima de la cámara 1/2"
3	Apertura (f/#) F1.2 - 16C
4	Longitud Focal FL (mm) 8.0 - 48.0
5	Campo de visión, 1/2"Sensor (°) 44.6 - 8.0
6	Distancia de Trabajo (mm) 1200 - ∞

Tabla 4.3: Ficha Técnica de la lente zoom Edmund optics.

4.3.3. Tarjeta de adquisición de imágenes para la cámara visible

Es una tarjeta EPIX modelo PIXCI CL1 Figura 4.8, incluye un software llamado XCAP-Lite, que es un programa cuyas características incluyen el control de la cámara, impresiones de lineas y columnas de píxeles, captura, desplegado y guardado de una imagen o una secuencia de ellas, pero también puede ser utilizado por otros programas como IDEA Software, que es un ambiente de desarrollo de aplicaciones y control de cámaras que funciona por medio de un conjunto completo de funciones de llamadas de hardware independientes. Se compone de un Software Development Kit (SDK) y una biblioteca de herramientas. El IDEA SDK proporciona funciones C y controles ActiveX para que los programadores tengan la opción de utilizar Microsoft Visual C / C + +, Visual Basic entre otros lenguajes. Las características se enumeran en la Tabla 4.4, [6].

	Tarjeta EPIX modelo PIXCI CL1.
1	Bus Base de Enlace para Cámara PCI Frame Grabber
2	Hasta 4.096 píxeles por línea y 65.535 líneas por cuadro
3	Tasa de secuencia de captura
4	Las imágenes almacenadas en la memoria de la placa base
5	132 Mbytes por segundo transferencias de ráfaga
6	Compatible con Windows y Linux, 32 y 64-bit

Tabla 4.4: Ficha Técnica de la tarjeta EPIX modelo PIXCI CL1.

figura 4.8: Tarjeta EPIX modelo PIXCI CL1.

4.3.4. Interfaz para la adquisición de imágenes de la cámara visible

Se programó una aplicación, que fue desarrollada en el lenguaje de programación Visual Basic y es capaz de reconocer la tarjeta de adquisición mostrada en al Figura 4.8 y obtener y grabar una imagen o una secuencia de imágenes a una frecuencia determinada. La velocidad de cuadros por segundo que se utilizó para la obtención de las imágenes con esta aplicación fue de 16 imágenes por segundo.

En la Figura 4.9 se muestran los elementos que conforman a) cuadro para el despliegue de la Imagen de la cámara en tiempo real, b) caja de texto para nombrar la secuencia de imágenes, c) caja de texto para determinar el número de imágenes a tomar, d) caja de texto para determinar el número de frames capturados por segundo, e) Botón para comenzar el proceso de adquisición y f) Salir del programa.

Las imágenes capturadas por esta aplicación son guardadas en la misma carpeta donde se ubica la aplicación y bajo el nombre especificado en la caja de texto b) concatenado con una numeración progresiva.

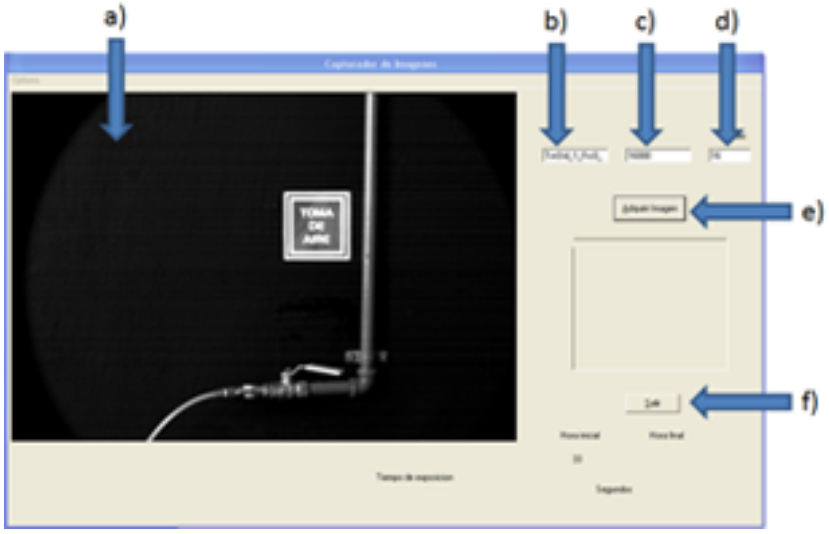

figura 4.9: Aplicación programada para la adquisición de imágenes con la cámara HITACHI.

4.4. Sistema de adquisición de Imágenes digitales en el espectro Infrarrojo y Visible

Dos cámaras proporcionan dos proyecciones distintas de una misma escena, que están relacionadas entre sí mediante la geometría de la Figura 4.10. En este sistema se captura un punto X en el espacio real, que se proyecta en el espacio imagen por los puntos x y $x\prime$, donde las coordenadas en los planos imagen estan relacionadas por,

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = e \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} tx \\ ty \end{bmatrix}, \tag{4.1}$$

donde e es un factor de escalamiento y tx, ty desplazamientos en los ejes respectivos.

El objetivo de este sistema es segmentar la información de la mano, sin necesidad de un fondo especial, marcadores o guantes que realcen la zona de interés.

En la Figura 4.11 se muestra el ejemplo de una imágen infrarroja y una visible adquiridas con el sistema propuesto. Puede observarse claramente, que las imágenes adquiridas son diferentes en escala, posición y energía.

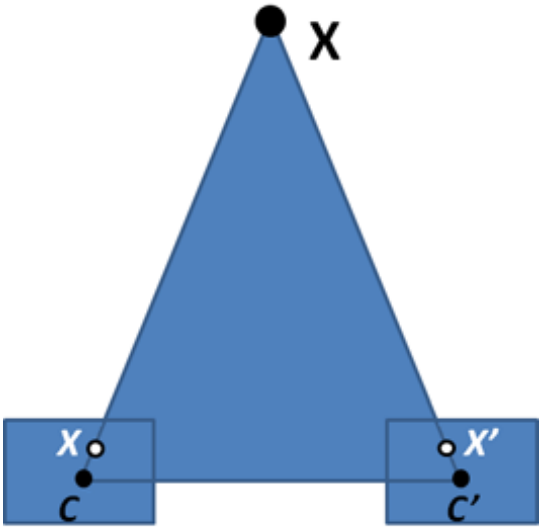

figura 4.10: Esquema del sistema de dos cámaras con centro en C y C'.

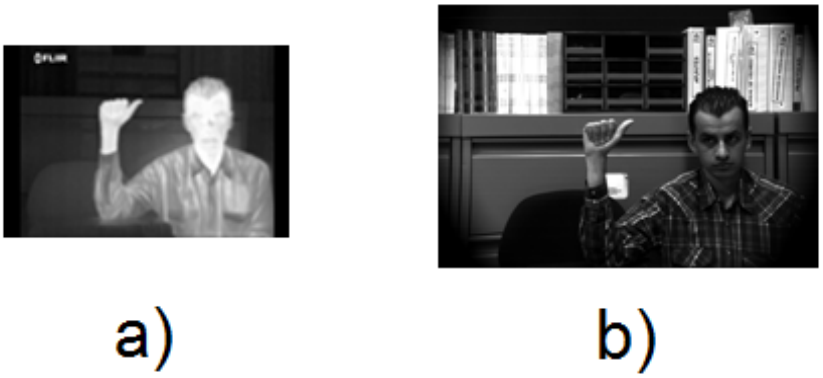

figura 4.11: Imágenes obtenidas por el sistema compuesto a) por la cámara térmica y b) por la cámara en el espectro visible de una misma escena.

4.5. Base de Datos UPT de Imágenes del Lenguaje de Señas

4.5.1. Imágenes en IR de los números del Lenguaje de Señas

Usando la cámara térmica para la adquisición de imágenes en el espectro del infrarrojo lejano, descrita en la sección 3.2, con rango de temperatura automático y los mapas de color arcoíris y escala de grises mostrados en la Figura 4.12, se generó una base de datos de los números del Lenguaje de Señas, desde el número uno hasta el número nueve como se muestra en la Figura 4.13; en esta imagen se aprecian las imágenes capturadas por la cámara térmica con un mapa de color arcoiris. Se capturaron 3 diferentes versiones de esta serie de números de diez personas diferentes.

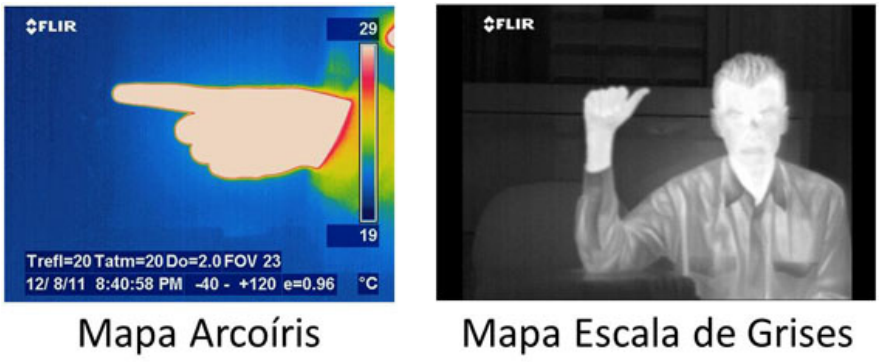

figura 4.12: Mapas de color de la cámara térmica utilizados.

La información de los datos que se buscó extraer de las imágenes que se muestran en la Figura 4.13 no se encuentran de forma procesable, lo que indica que deben ser sometidas a un pre-procesamiento. Claramente, la mayor cantidad de información útil se encuentra en las manos, es decir, en el color de la mano que arrojó el mapa de colores utilizado para la obtención de las muestras, por tanto, fue necesario hacer transformaciones al espacio de color HSI (Hue, Saturation, Intensity). Las imágenes que proporciona la cámara térmica se encuentran en el espacio de color RGB (Red, Green, Blue), espacio de color que la mayoría de las cámaras utilizan para grabar imágenes, de forma que la imagen con un tamaño original de 640x480 píxeles son recortadas para eliminar información de la escena que no es necesaria, posteriormente es transformada del espacio RGB al espacio HSI obteniendo nuevamente tres matrices en escala de grises, dichas matrices fueron binarizadas intentando así facilitar la extracción de información. Este proceso se muestra en la Figura 4.14.

A simple vista las tres matrices resultantes de la conversión del espacio de color de la Figura 4.14 parecieran ser completamente iguales, pero una imagen mejor detallada, Figura 4.15, muestra lo contrario. Obsérvese que las matrices de Saturación e Intensidad presentan gran cantidad de ruido en la silueta dibujada y que la matriz de Tono no lo hace.

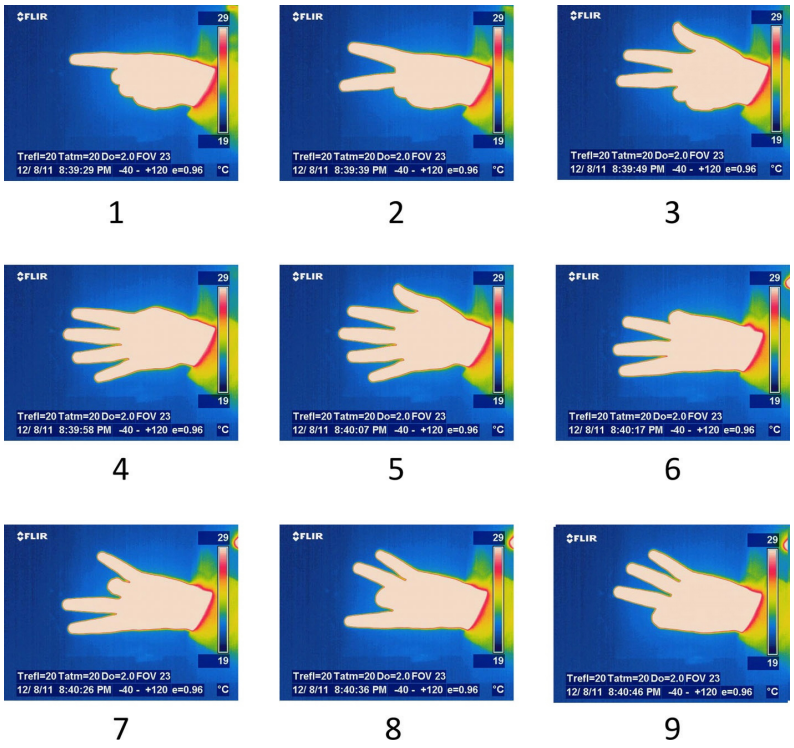

figura 4.13: Imágenes de los dígitos númericos del Lenguaje de Señas obtenidas de la cámara térmica en el mapa de color arcoiris. Version 1 de los números del 1 al 9. El número de cuadro representa el dígito representado por la mano.

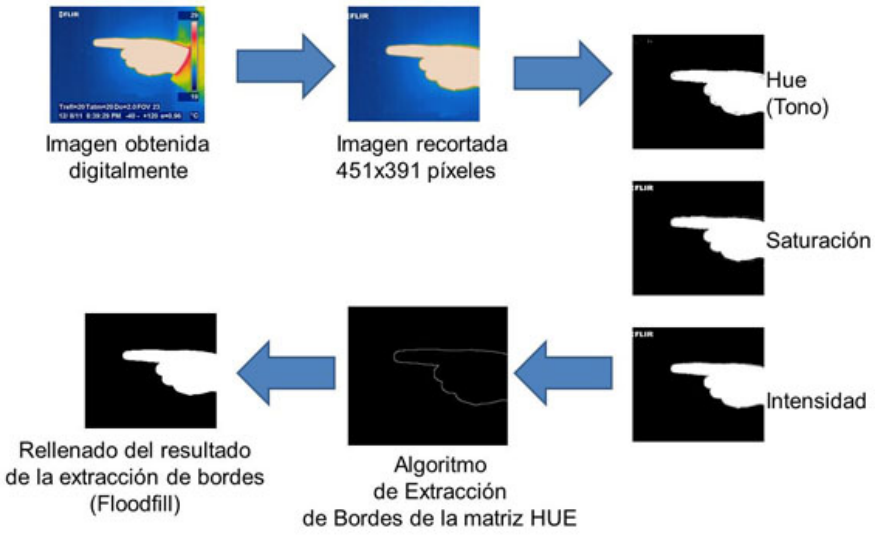

figura 4.14: Preprocesado de las imágenes digitales para la segmentación de la información de interés.

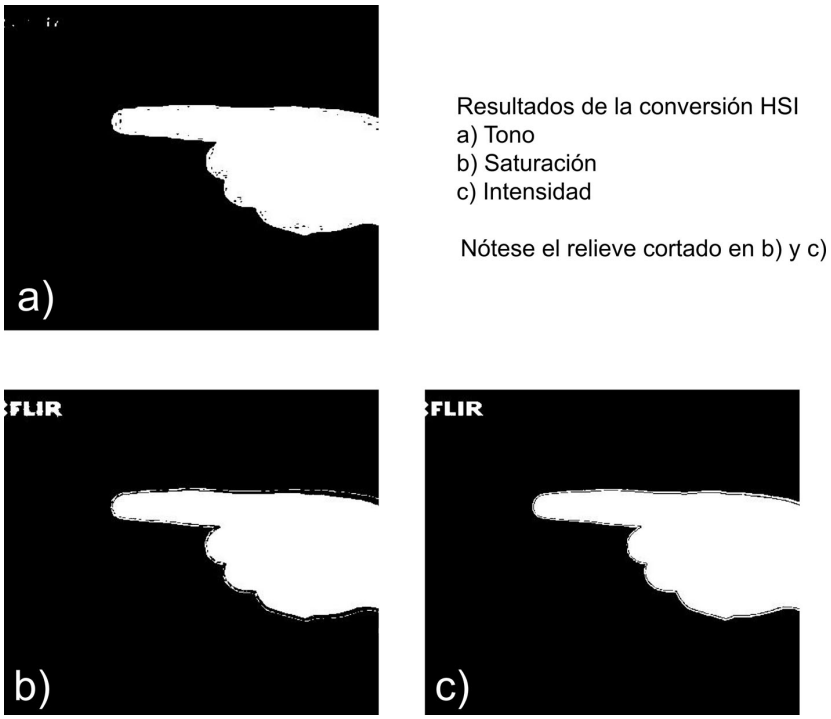

figura 4.15: Resultados de la conversión HSI a) Tono b) Saturación c) Intensidad

De las tres matrices resultantes, la de tono es la mejor candidata para su tratamiento, así que se le aplica un algoritmo para la extracción de bordes descrito en la sección 2.5.2, con el cual obtenemos un conjunto de posiciones en coordenadas cartesianas, que al gráficar nos arroja una imagen como la que muestra en el algoritmo de extracción de bordes de la imagen 4.14. Posteriormente, este contorno es rellenado con un algoritmo de relleno llamado floodfill, cuyo objetivo es el de cambiar los valores de intensidad de los píxeles que se encuentran dentro del contorno para así poder obtener una imagen solida de la mano. Este preprocesamiento fue aplicado a las tres versiones de un sujeto, resultando en una base de imágenes como la que se muestra en la Figura 4.16.

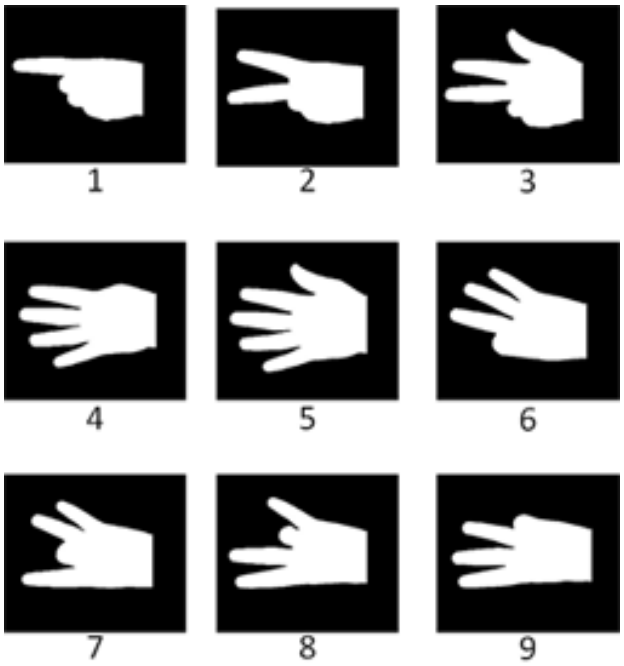

figura 4.16: Base de datos de los dígitos del lenguaje de señas. Una de las tres series de imágenes de los números del lenguaje de señas obtenida despues del preprocesamiento de las imágenes térmicas.

Este preprocesamiento se aplicó a tres diferentes versiones de cinco sujetos (Figura 4.17), para después ser procesadas para su clasificación como se abordará en el próximo capítulo.

figura 4.17: Digito 1 version 1 de los cinco sujetos.

4.5.2. Imágenes en IR y Visible para el abecedario del Lenguaje de Señas

Posteriormente, se programó una interfaz en MATLAB que conecta la cámara térmica a la computadora, Figura 4.4, realizando la función del guardado de las imágenes que recolectara la cámara térmica. También se utilizó un programa que capturara las imágenes de la cámara HITACHI en el espectro visible en escala de grises (Figura 4.9). De esta forma se capturaron tres versiones del alfabeto, recordando que se omitieron las letras que conllevan movimiento. Ambas cámaras grabaron las imágenes de una misma escena al mismo tiempo, obteniendo así las imágenes de las Figuras 4.18 de la cámara térmica y Figura 4.19 de la cámara en espectro visible.

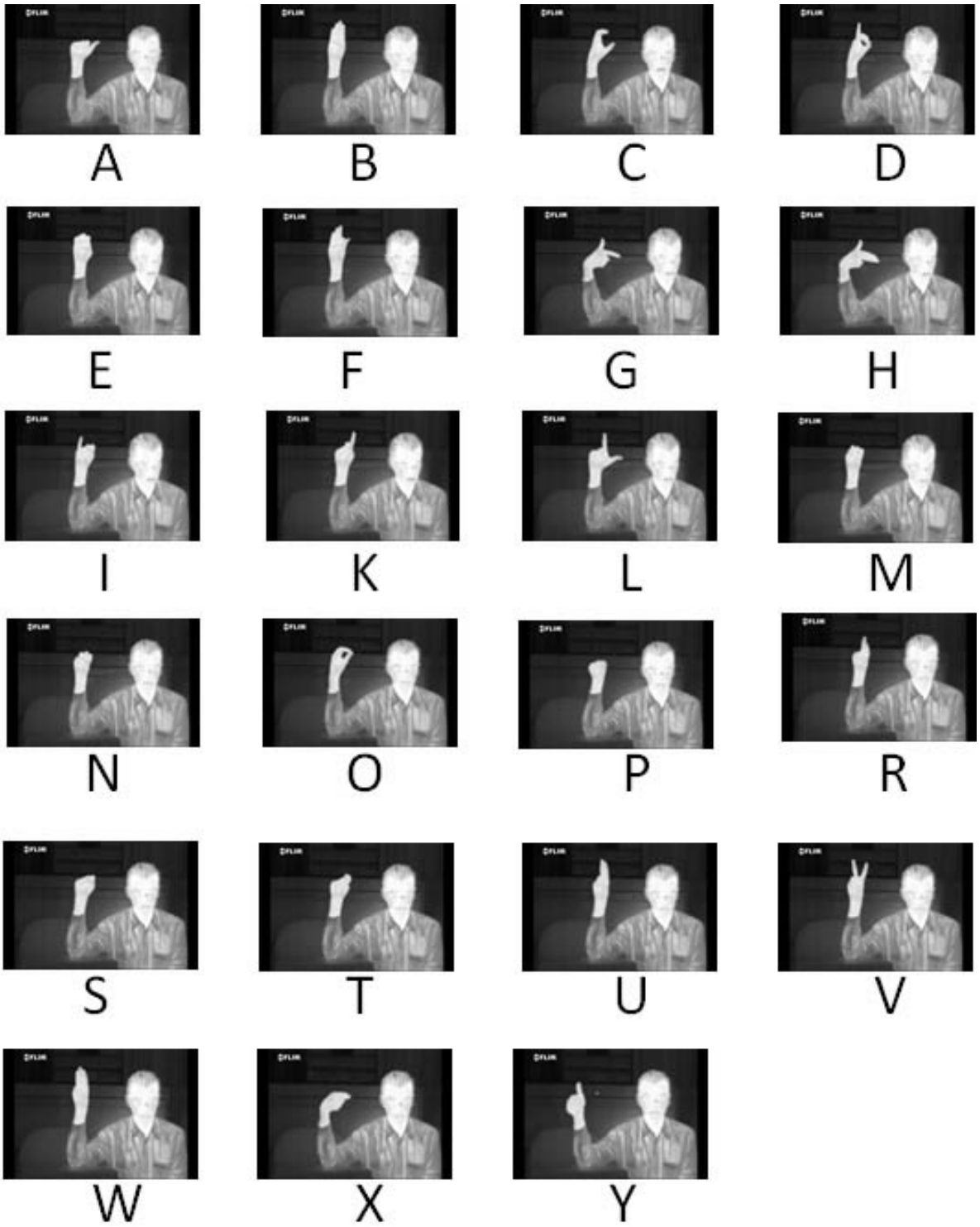

figura 4.18: Imágenes de los dígitos alfabéticos del Lenguaje de Señas obtenidas de la cámara térmica con el mapa de color escala de grises. Version 1 de las letras de la "A" a la "Y". La letra inferior del cuadro representa el símbolo representado por la mano.

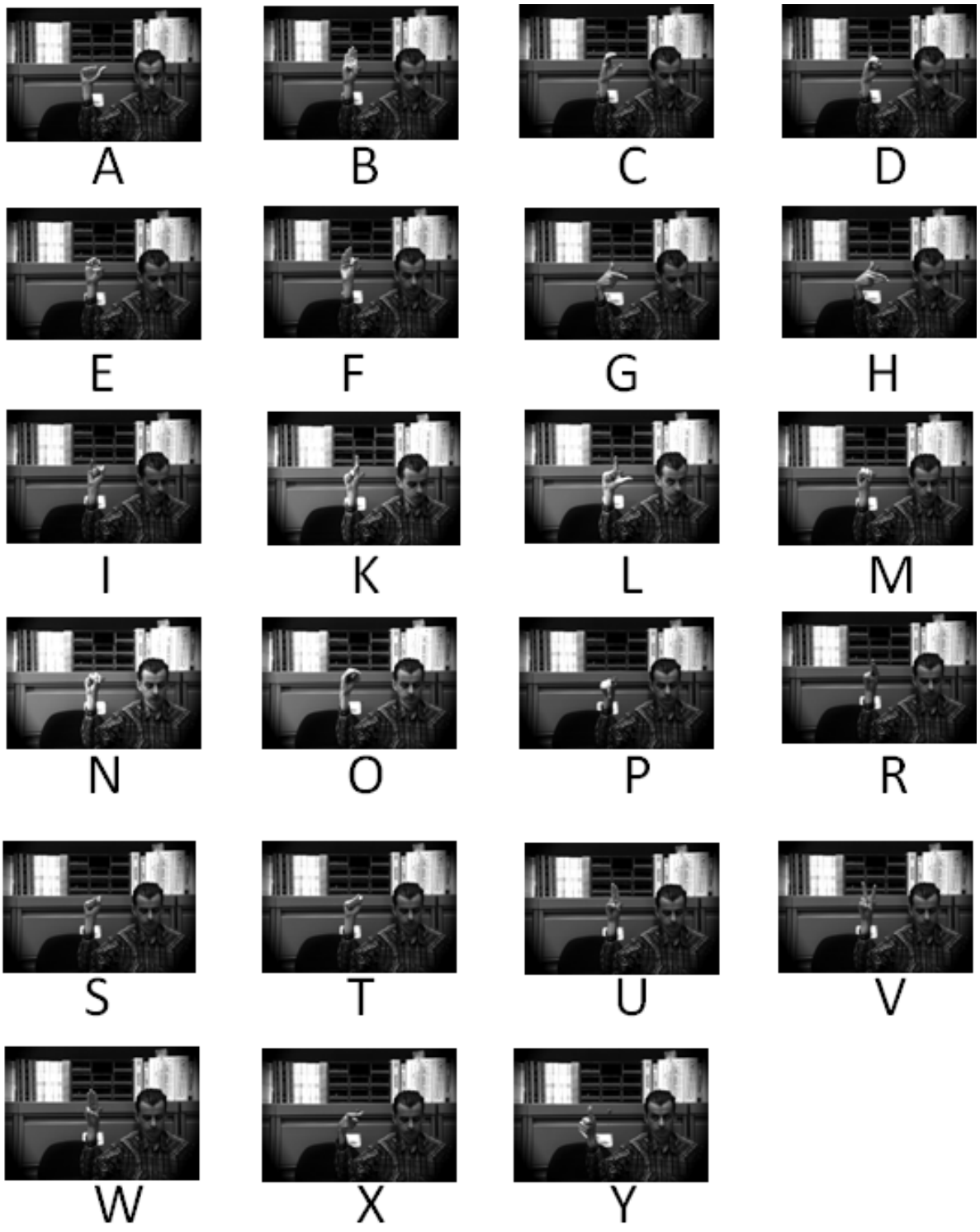

figura 4.19: Imágenes de los dígitos alfabéticos del Lenguaje de Señas obtenidas de la cámara visible. Version 1 de las letras de la "A" a la "Y". La letra inferior del cuadro representa el símbolo representado por la mano.

Una vez obtenidas estas dos series de imágenes para tres versiones del alfabeto del lenguaje de señas, se hace incapié en el uso de las imágenes térmicas debido a que esta realza la información de la región deseada. Sin embargo, se tienen cambios de amplificación y posición que son determinados experimentalmente.

La binarización de la imagen IR está dada por,

$$BinariaIR(x', y') = \begin{cases} 0 & IR(x', y') \leq umbral \\ 1 & \text{otro caso} \end{cases} \tag{4.2}$$

y el escalamiento y traslación de la imagen visible por,

$$Visible(x, y) = V(x', y'), \tag{4.3}$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = e \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} tx \\ ty \end{bmatrix}. \tag{4.4}$$

La segmentación se obtiene por,

$$Segmentada(x', y') = BinariaIR(x', y') \wedge Visible(x, y). \tag{4.5}$$

Determinación del umbral de binarización

Si bien antes mencionamos una ecuación de binarización (ec. 4.2), debemos describir como fue el proceso de selección del umbral. Para la obtención de este número, fue primeramente recortada la zona de mayor importancia en la escena, es decir, la zona que contiene la mano, como se muestra en la Fig. 4.20.

Una vez que fue obtenida esa zona, se obtuvo un histograma (el número de veces que se repiten los niveles de intensidad) cuyo resultado se muestra en la Figura 4.21.

Como puede apreciarse en la figura anterior, la cuantización de intensidad en la imagen se encuentra en el rango de [0,255] donde mayor parte de intensidades se encuentra entre el 40 y el 60 que representa el fondo de la escena y existe otro pico entre el 200 y el 250 lo que indicaría que el número umbral sería de 200, pero al inspeccionar de cerca la imagen b) de la Fig. 4.23, observamos que contiene píxeles con valores de intensidad menores a 200, por variaciones en la temperatura y para poder obtener una binarización de la mano sin pérdida de informacion se reduce este número a 180, de esta forma es como se obtiene el umbral de binarización. El resultado de este proceso se muestra en la Figura 4.22.

Usando como factor de escala $e = 0{,}3$ y el vector de traslaciones $t = (80\ 47)$, la imagen segmentada es mostrada en la Figura 4.23.

Con lo anterior fue posible generar tres bases de datos del alfabeto en niveles de gris, de contorno y binarios. Las bases de datos se muestran en las Figuras 4.24, 4.25 y 4.26, respectivamente.

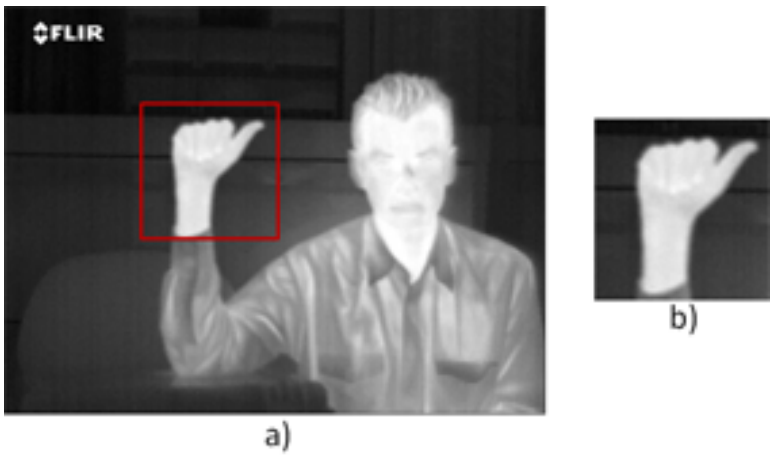

figura 4.20: a) Imagen térmica, b) Segmentación de la zona de interés con mayor información de la imagen térmica.

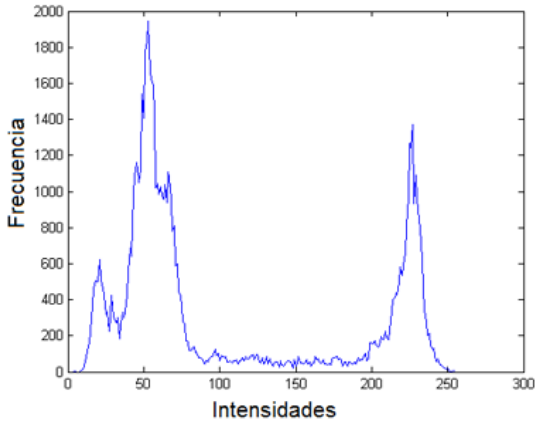

figura 4.21: Gráfica del histograma de la imagen b) de la Figura 4.20.

figura 4.22: Imagen de la cámara térmica binarizada con el umbral [180, 255] obtenido por medio del histograma.

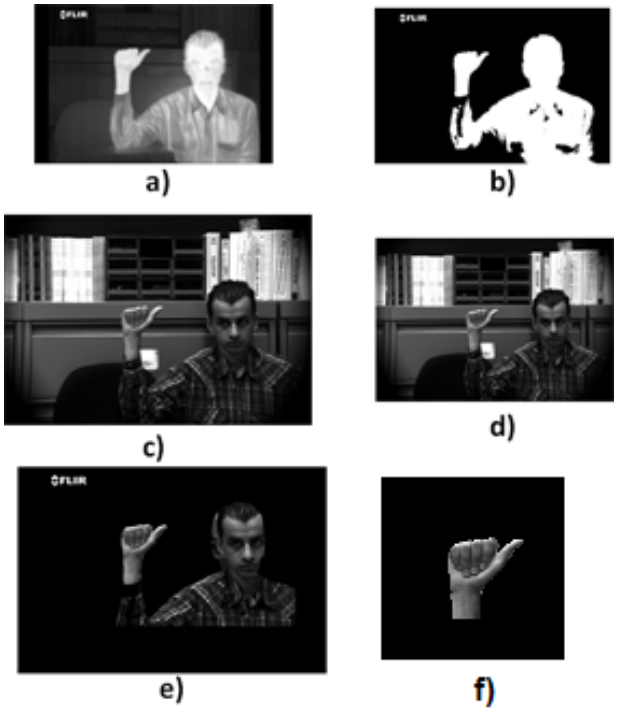

figura 4.23: a) Imagen de la cámara térmica, b) Imagen binarizada de la cámara térmica, c) Imagen de la cámara visible, d) Imagen de la cámara visible escalada $e = 0{,}3$ y trasladada $tx = 80$ y $ty = 47$, e) Imagen de la cámara visible d) segmentada por la imagen binarizada de la cámara térmica b) y f) Región con mayor información extraida de la imagen e).

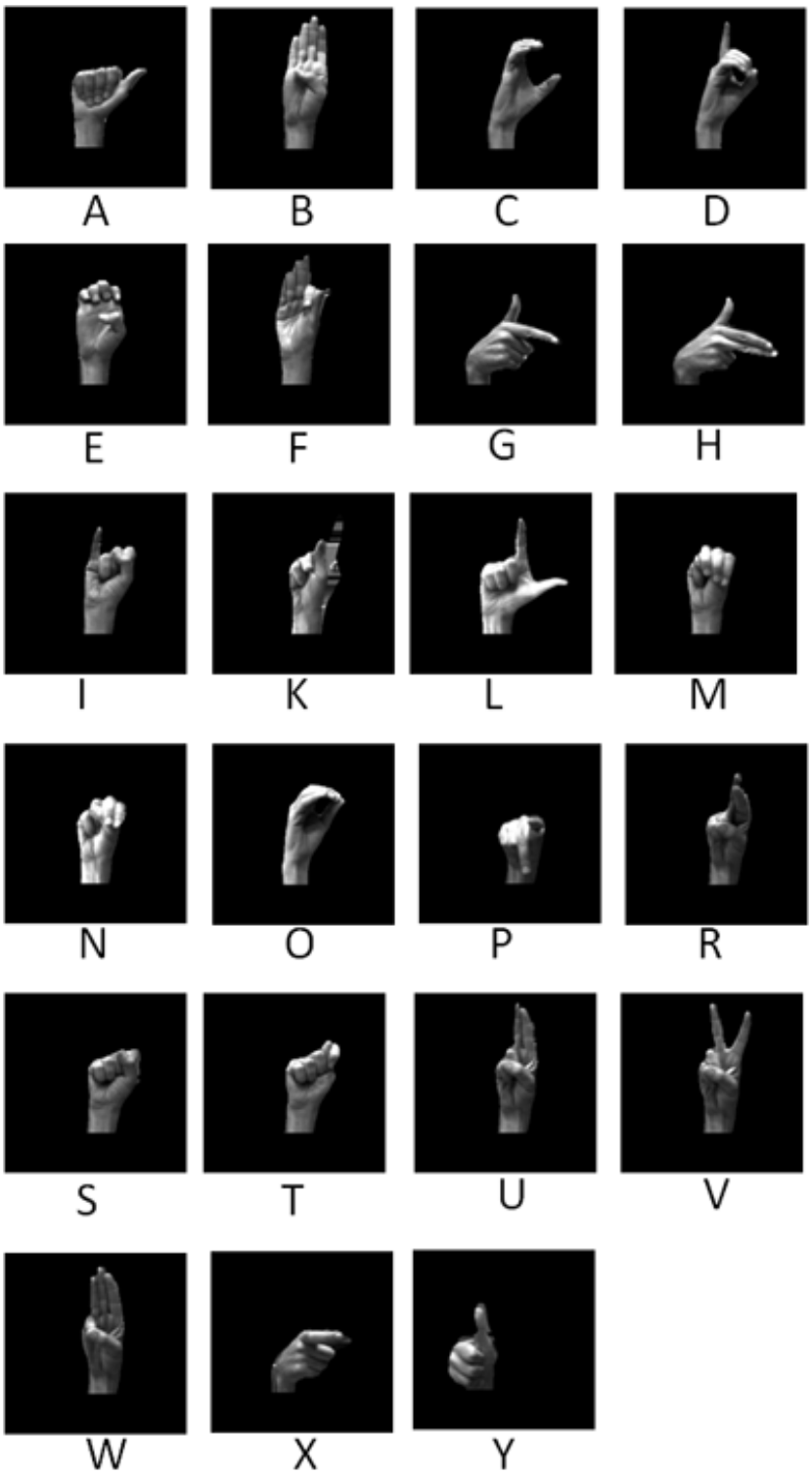

figura 4.24: Base de datos en niveles de gris del alfabeto. Imágenes de los dígitos alfabeticos del Lenguaje de Señas obtenidas de la cámara visible, segmentadas por las imágenes de la cámara térmica y recortadas. Version 1 de las letras de la "A" a la "Y". La letra inferior del cuadro representa el símbolo representado por la mano.

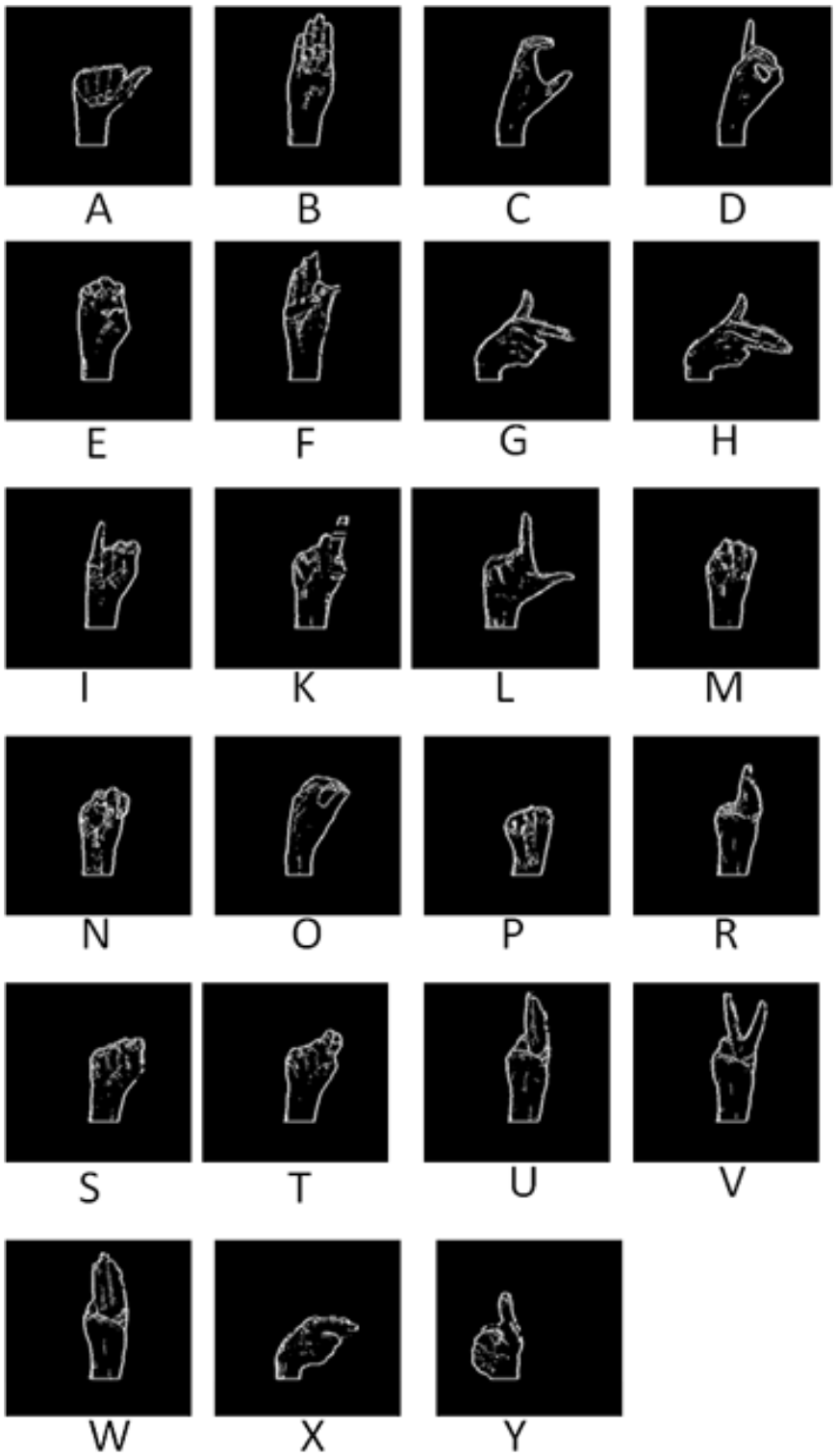

figura 4.25: Base de datos formadas por contornos a través de el filtraje del alfabeto.Imágenes de los dígitos alfabeticos del Lenguaje de Señas obtenidas de la cámara térmica, binarizadas y recortadas. Version 1 de las letras de la “A” a la “Y”. La letra inferior del cuadro representa el símbolo representado por la mano.

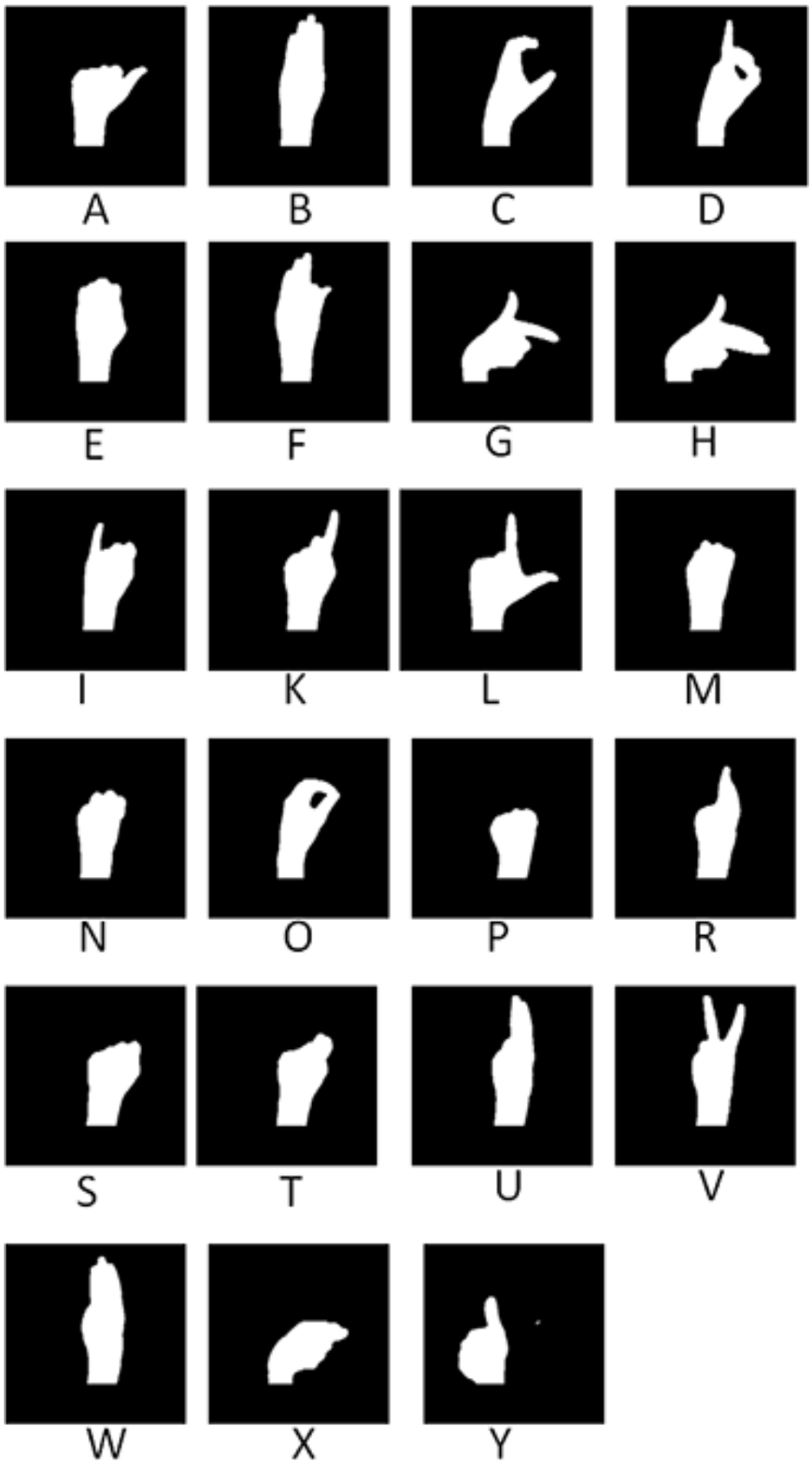

figura 4.26: Base de datos de imágenes binarias del alfabeto. Imágenes de los dígitos alfabeticos del Lenguaje de Señas obtenidas de la cámara térmica, binarizadas y recortadas. Version 1 de las letras de la la "A" a la "Y". La inferior letra del cuadro representa el símbolo representado por la mano.

4.6. Conclusiones

En este capítulo se describieron dos sistemas para la adquisición de imágenes digitales del Lenguaje de Señas Americano. Con el primer sistema se obtuvieron las imágenes de los números y con el segundo las imágenes del alfabeto.

El primer sistema hace uso de una cámara térmica con respuesta espectral de alrededor de las 10μm, que siguiendo el modelado para cuerpos negros, es cercana a la radiación del calor emitido por el cuerpo humano. Con este sistema, fueron capturadas tres versiones de los números del 1 al 9 de diez diferenetes sujetos. Las imágenes se muestran en la Figura 4.13. La segmentación de la información de la mano fue hecha en el espacio de color HSI, por medio de la matriz de tonos. La base de datos final fue pre-procesada como se muestra en la Figura 4.14.

Con el sistema de dos cámaras, IR y visible de la figura 4.10, se adquirieron 2 series de imágenes de diferentes escalas, traslaciones y energía pero de la misma escena y al mismo tiempo los signos del alfabeto del lenguaje de señas se muestran en las Figuras 4.18 y 4.19. La segmentación de la mano se logró con las imágenes de la cámara térmica. El proceso se describe en la Figura 4.23. Se generaron tres bases de datos del alfabeto que se muestran en las Figuras 4.24, 4.25 y 4.26

En el siguiente capítulo se usarán las imágenes de las Figuras 4.16 y 4.24 para el reconocimiento y clasificación de señas usando redes neuronales artificales, correlación y descriptores de pendiente.

Bibliografía

[1] Sitio Oficial de los Sistemas FLIR http://www.flir.com/MX/

[2] R. D. Fiete, "Modeling the Imaging Chain of Digital Cameras", *Tutorial Texts in Optical Enginering* SPIE Press (2010).

[3] MATLAB Forum's Programmers http://www.mathworks.com/matlabcentral/

[4] Sitio Oficial de HITACHI Kokusai Electric http://www.hitachi-kokusai.co.jp/global/index.html

[5] Sitio Oficial de Edmund Optics http://www.edmundoptics.com/

[6] Sitio Oficial de EPIX, Inc. Image Processing for Research and Industry http://www.epixinc.com/

Capítulo 5

Reconocimiento de Signos del Lenguaje de Señas

5.1. Introducción

En este capítulo se analizan los resultados de clasificación de los dígitos y señas del Lenguaje Americano. Los algoritmos para la extracción de descriptores 1D basados en valores pendiente, alrededor de un contorno y firmas de correlación son analizadas en las secciones 2 y 3. Lo anterior permite tener vectores descriptores que serán clasificados usando correlación de Pearson y redes neuronales artificiales.

La extracción de características en 2D usando la transformación wavelet Haar, se describe en la sección 3. A partir de estos descriptores, se llevó a cabo la clasificación usando un perceptrón multicapa con validación cruzada, factor de aprendizaje $\alpha = 0{,}3$ y un momentum $\beta = 0{,}2$. Los resultados de clasificación de dígitos y signos del alfabeto del Lenguaje de Señas Americano son mostrados en la sección cuatro, donde se lograrón porcentajes de correcta clasificación del 100 % de los dígitos y 85.5 % para el alfabeto para tres versiones diferentes de los digitos y del alfabeto de una misma persona. En la sección cinco se muestra una interfaz programada en MATLAB para el reconocimiento dinámico de los dígitos del 1 al 9. Finalmente, las conclusiones son expuestas en la sección seis.

5.2. Extracción de características 1D

5.2.1. Descriptores de digitos basados en Pendientes

Sea $f(x, y)$ una imagen binaria de tamaño MxM, mostrada en la Figura 5.1.

Se genera la cadena de código (CC) estudiada en la sección 2.5.2, construida por las coordenadas píxel que pertenecen al contorno del objeto y dados por,

$$CC^{P,V,D} = CC^{1,1,5} = \{(265, 446), (265, 445), (265, 444), (264, 444)...(135, 446)\} \tag{5.1}$$

donde $P = 1..,5$ es el número de personas, $V = 1..,3$ es el número de versiones y $D = 1..,9$ es la cantidad de digitos.

Aplicando la Ecuación 2.15 con $\Delta_{fijo} = 10, 20, 40$ a la cadena de código (5.1), se obtiene la pendiente entre $(P_k, P_{k+\Delta})$ por,

$$M_k^{1,1,5}(P_k, P_{k+\Delta}) = \frac{Y_{k+\Delta} - Y_k}{X_{k+\Delta} - X_k} \tag{5.2}$$

$$\text{con } k + \Delta x = \Delta_{fijo} \tag{5.3}$$

donde $k = 0...K$ para K el número máximo de elementos de todas las cadenas de código. En la Figura 5.2 se muestra un ejemplo del cálculo de la pendiente con $\Delta_{fijo} = 10, 20, 40$ para todas las cadenas $CC^{1,1,5}$.

Como se puede observar en la figura anterior, el cambio de Δ_{fijo} influye en las características de las curvas, si bien, logramos observar que la curva mejor definida para la persona 1, version 1 y digito 5, se observa cuando $\Delta_{fijo} = 10$. Se hicieron las pruebas con los demás digitos del 1 al 9 para obtener los mismos resultados, es decir, que las curvas mejor definidas para todos los casos es cuando $\Delta_{fijo} = 10$. Por lo que esta configuración se propone para el resto del procesamiento de los digitos como se muestra en la Figura 5.3.

figura 5.1: Imagen proveniente de la matriz Tono de la Figura 4.15.

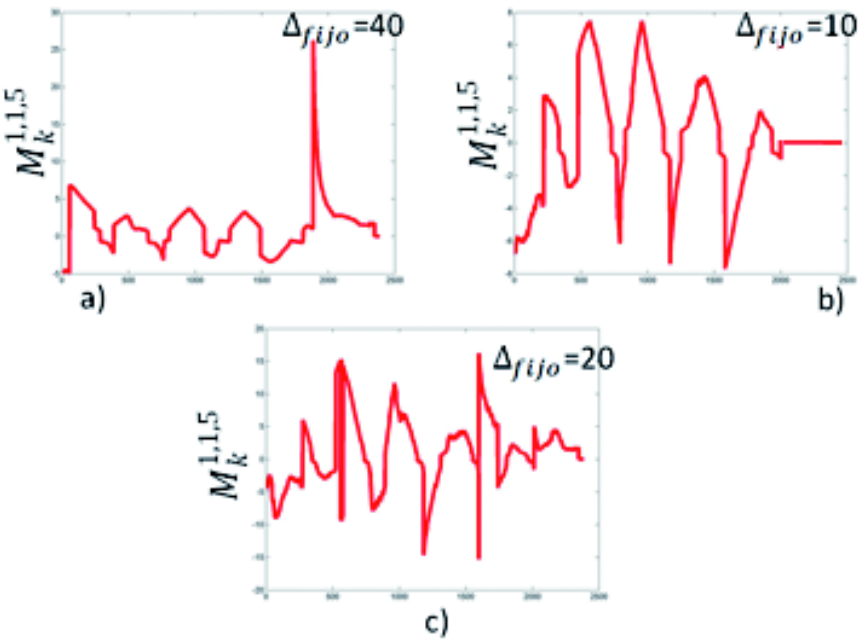

figura 5.2: Curvas de pendientes de la $C^{1,1,5}$ para a) $\Delta_{fijo} = 40$, b) $\Delta_{fijo} = 10$, c) $\Delta_{fijo} = 20$. Cada pico en la curva representa un dedo de la mano, por lo que, cuando $\Delta_{fijo} = 10$, tenemos la mejor extracción de características.

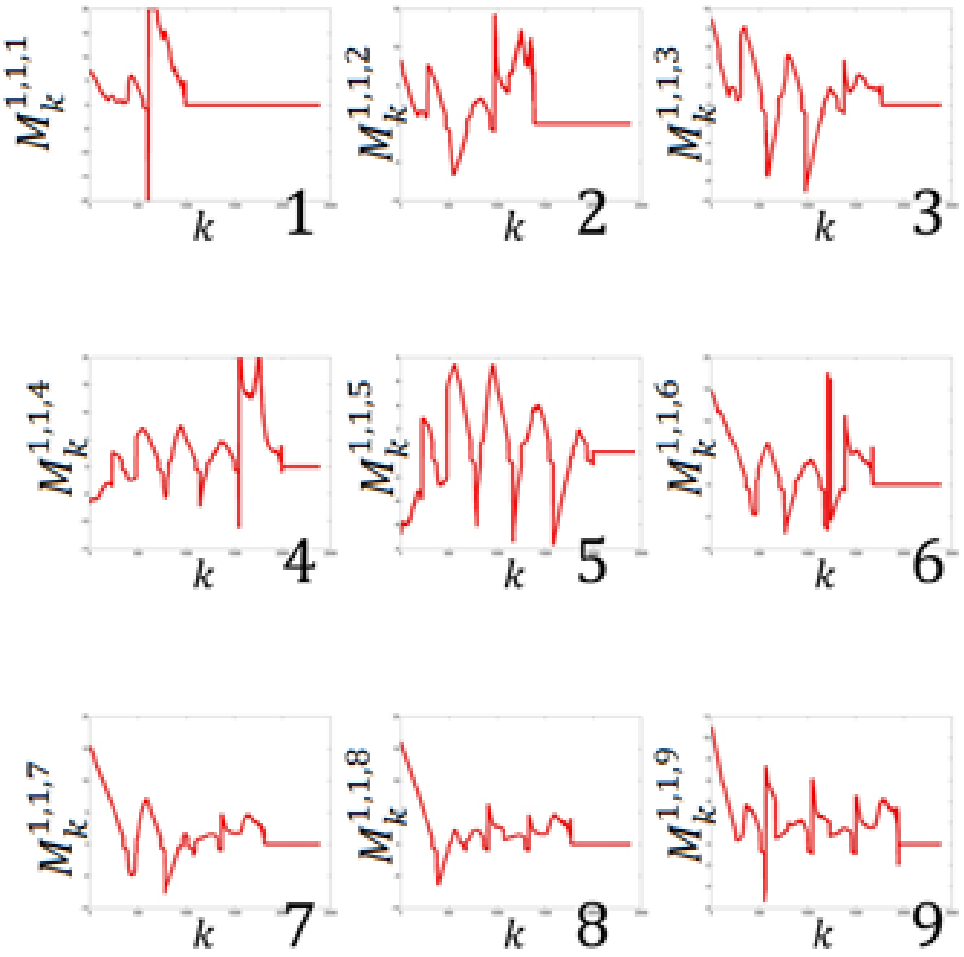

figura 5.3: Descriptores 1D de la base de datos de la Figura 4.16.

5.2.2. Descriptores de dígitos basados en firmas de correlación

Sea $f(x, y)$ una imagen digital binaria y centrada con la técnica de momentos descrita en el Apéndice D. Usando la base rotatoria de la Figura 5.4, se adquirierón un conjunto de n-imágenes, rotadas analógicamente de $[0°, 330°]$, como se muestran en la Figura 5.5.

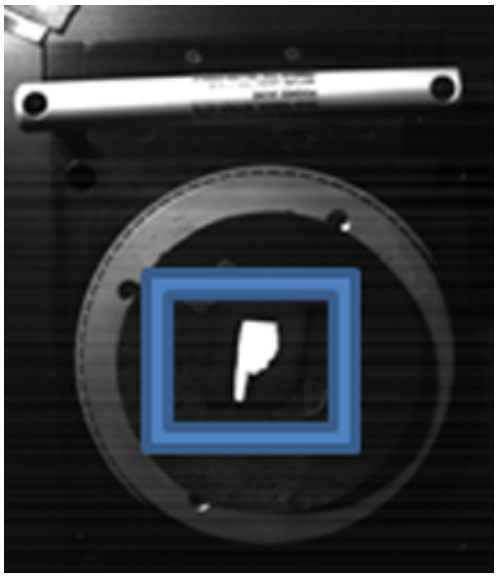

figura 5.4: Imagen de uno de los dígitos númericos del lenguaje de señas sobre una base rotatoria.

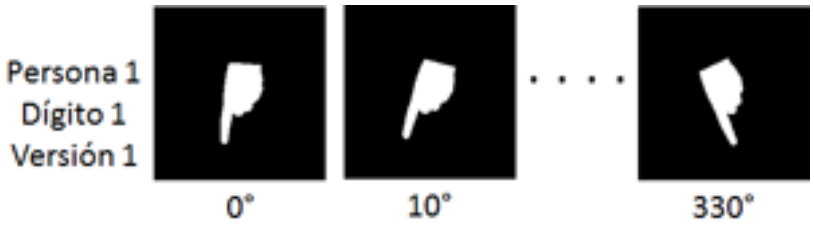

figura 5.5: Secuencia de imagenes capturadas con la cámara HITACHI, recortadas, binarizadas, centradas y giradas analógicamente.

Se calcula la correlación de $f(x, y)$ con cada una de las versiones rotadas $h_i(x, y)$ usando la (ec. 2.30), para n imágenes como,

$$corr_i(f_1, h_i) = \frac{1}{MN} \sum_{x=1}^{M} \sum_{y=1}^{N} g_i(x, y) \tag{5.4}$$

En la Figura 5.6 se muestran las gráficas de correlación de la imágen f con cada una de sus versiones rotadas h_i con un número n de muestras diferentes. A esta curva se le denomina firma de correlación [1].

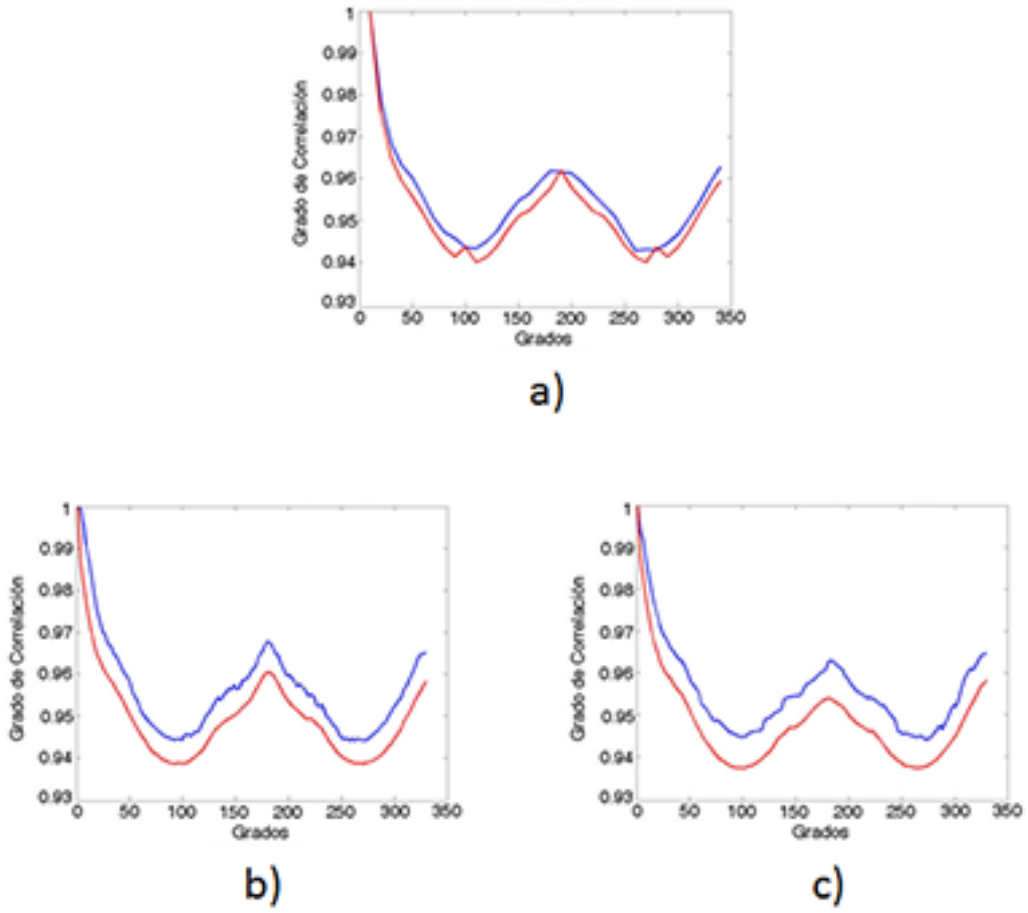

figura 5.6: Firmas de correlación (Grado de correlación VS Grados Girados). Las firmas de correlación del dígito 1 versiones 1,2 y 3 respectivamente. Las firmas de color azul son obtenidas de forma analógica con la base rotatoria y las de color rojo con giros digitales.

De la Figura 5.6 nótese que la firma a) no está tan definida como b) y c) debido a que las muestras fueron tomadas cada 10°, sin embargo en b) y c) las muestras sucedierón cada 2°, lo que provoca que un mayor número de elementos sean gráficados y por lo tanto una mejor definición de la firma. Nótese también que las tres comparten similaridades en la forma. Es importante tomar en cuenta que la diferencia entre las firmas analógicas (azules) y las digitales (rojas) son extremedamente parecidas, apenas 0.01 grados de diferencia entre una y otra, lo que indica que no existe mucha diferencia para el uso de estas en la clasificación, por lo que los resultados serán casi indistintos sea de forma analógica o de forma digital.

En la Figura 5.7 se muestran las firmas de correlación digital de los dígitos del 1 al 9 correspondientes a la versión 2 de la persona 1, con muestras cada 2°.

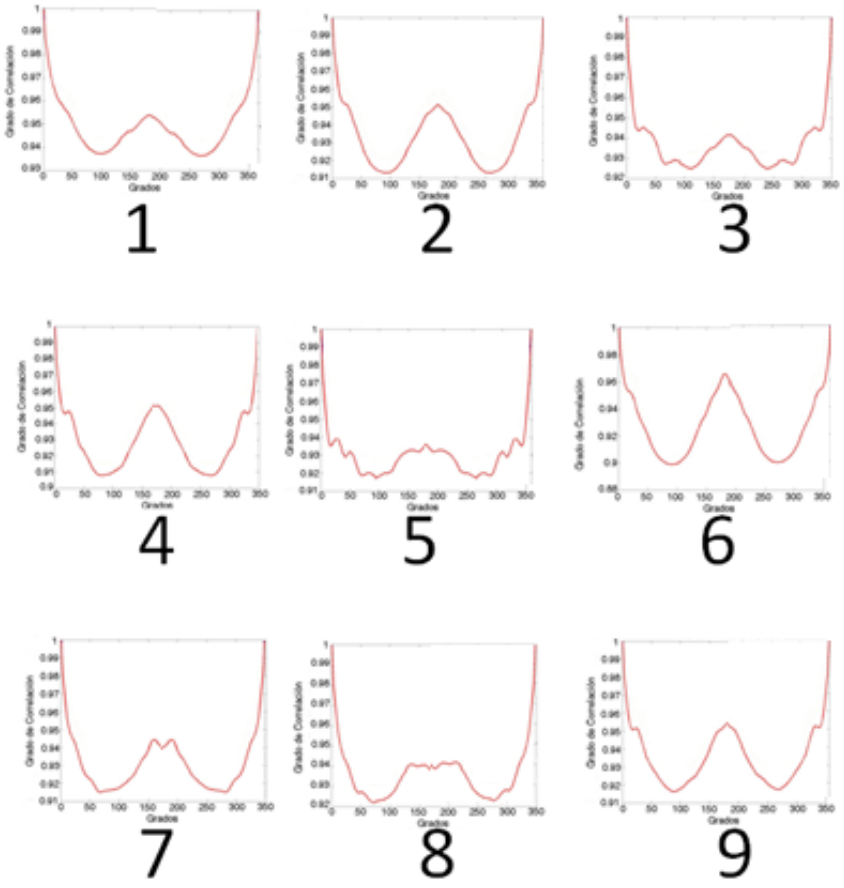

figura 5.7: Firmas de correlación digital de la versión 2 de la persona 1, cada número representa la firma de correlación del digito.

5.3. Extracción de características 2D

5.3.1. Descriptores de Signos del Alfabeto basados en Transformada Wavelet Haar

Sea $f(x, y)$ una imagen binaria de tamaño $2^k x 2^k$, donde $k = \{\mathbb{Z}\}$, se calcula su transformada wavelet Haar, usando la Ec. 2.51. El resultado es una rejilla multiresolución $\delta^{x,y}$.

Los descriptores de la imagen se forman con las sub-imágenes $\delta^{8,8}$,$\delta^{7,7}$,$\delta^{6,6}$ y $\delta^{5,5}$. Un ejemplo de estas imágenes se muestra en la Figura 5.8. Cada sub-imagen $\delta^{x,y}$, se transforma en un vector descriptor de tamaño $\left(\frac{N}{2^x}\right)\left(\frac{N}{2^y}\right)$.

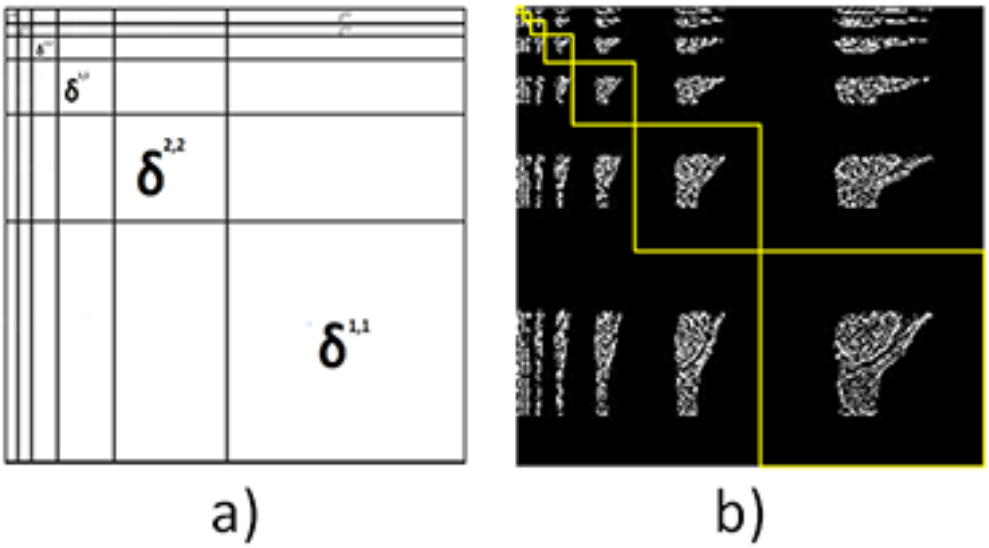

figura 5.8: a) Rejilla multiresolución Wavelet Haar, b) Transformada Wavelet Haar del Dígito 1 de la Figura 4.24.

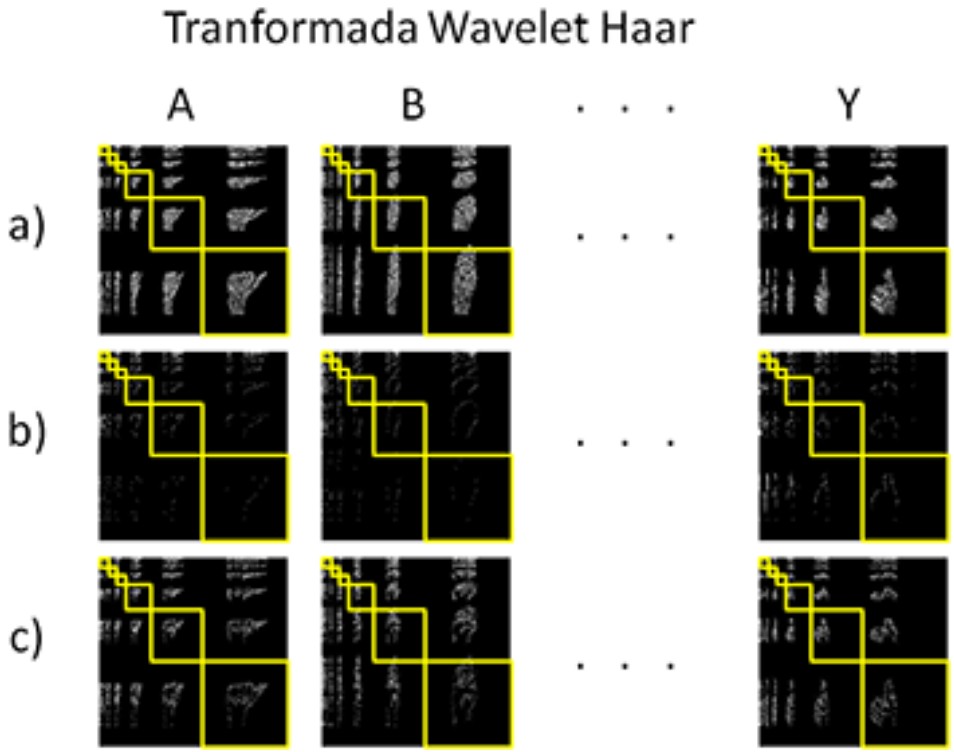

figura 5.9: Transformada Wavelet Haar para todas las imágenes de las bases de datos del alfabeto: a)Figura 4.24, b) Figura 4.25 y c) Figura 4.26.

5.4. Clasificadores y sus resultados de clasificación

5.4.1. Correlación

Clasificación a partir de los descriptores de pendientes

Procesando la base de datos de dígitos $d = 1, 2, ..., D$ con versiones $v = 1, 2, ..., V$ para la persona $p = 1$, se obtuvieron los vectores descriptores $M_k^{p,v,d}$ de la Figura 5.3.

Usando la Ec. (2.32) del correlador de Pearson como,

$$c(d) = corr_d = (M_k^{1,VTest,DTest}, M_k^{1,v,d}), \tag{5.5}$$

Es posible comparar los descriptores de un dígito $dTest$ con las versiones v de los dígitos d. El dígito será clasificado correctamente cuando,

$$clasifica(d, v) = \begin{cases} 1 & si\ máx(c(d, v)) = c(dTest, vTest) \\ 0 & \text{otro caso} \end{cases}, \tag{5.6}$$

Por lo tanto, el porcentaje de clasificación es,

$$Porcentaje\ de\ Clasificación = \frac{1}{V \cdot D} \sum_{dTest=1}^{D} \sum_{d=1}^{D} \sum_{v=1}^{V} Clasifica(d, v). \tag{5.7}$$

Un ejemplo de clasificación se muestra en la Figura 5.10, donde se muestran las gráficas correspondientes a las correlaciones entre el Dígito 1 Versión 1 contra los demás dígitos de la Versión 2, nótese como la correlación más alta siempre se da en el dígito correspondiente al Dígito 1 consigo mismo (barra azul claro), a excepción del caso f) donde el dígito 1 se confunde con el dígito 8. Este mismo método se usó para la clasificación de la firmas de correlación.

Esto resultaría en,

$$clasifica(d, v) = \{1, 1, 1, 1, 1, 0, 1, 1, 1\}, \tag{5.8}$$

donde el porcentaje de clasificación se calcula como,

$$Porcentaje\ de\ Clasificación = \frac{1}{1 \cdot 9} \sum \{1, 1, 1, 1, 1, 0, 1, 1, 1\} \tag{5.9}$$

$$.Porcentaje\ de\ Clasificación = \frac{8}{9} = 0{,}888^{-} = 88\,\%. \tag{5.10}$$

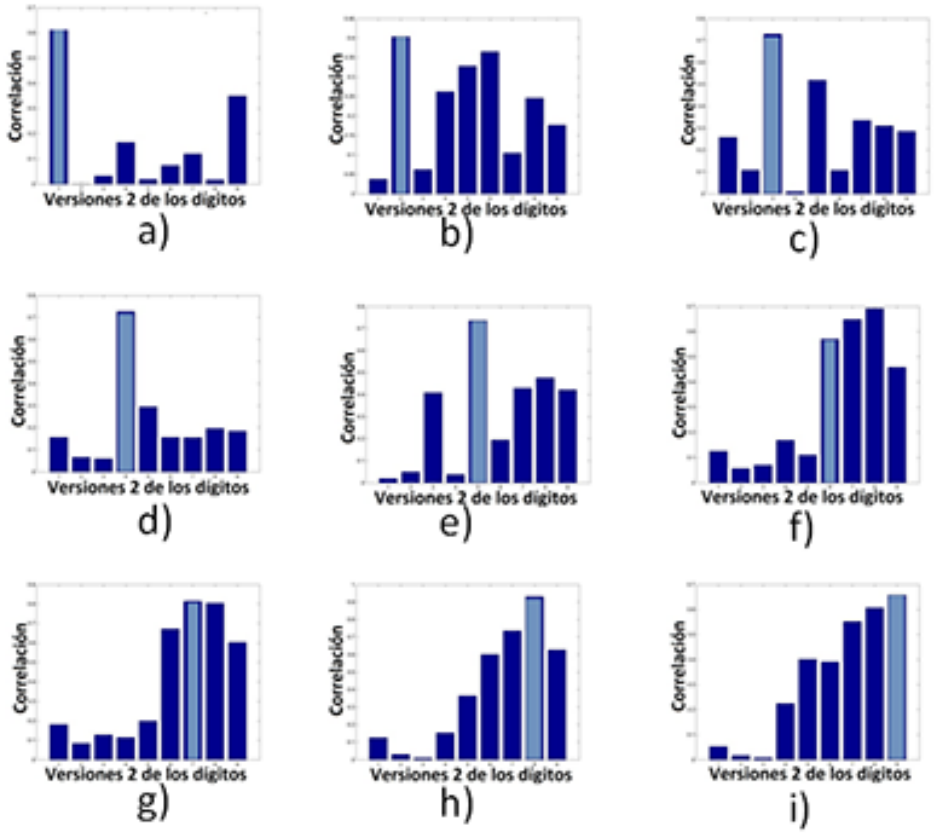

figura 5.10: Gráficas de Correlación. Dígito 1 Versión 1 contra las Versiones2 de a) Dígito 1, b) Dígito 2, c) Dígito 3,d) Dígito 4,e) Dígito 5 ,f) Dígito 6, g) Dígito 7, h) Dígito 8, i) Dígito 9.

5.4.2. Redes Neuronales Artificales

Para la clasificación de los elementos de las bases de datos del alfabeto y los dígitos se programó un perceptrón multicapa que fué utilizado en las primeras pruebas, pero debido al tiempo, fué utilizado un programa llamado WEKA (Waikato Environment for Knowledge Analysis - Entorno para Análisis del Conocimiento de la Universidad de Waikato) que es una plataforma de software para aprendizaje automático y minería de datos escrito en Java y desarrollado en la Universidad de Waikato. Weka es un software libre distribuido bajo licencia GNU-GPL. En este programa se eligieron los parámetros, perceptrón multicapa con una capa oculta, un factor de aprendizaje $\alpha = 0{,}3$ y un momentum $\beta = 0{,}2$ por medio de una validación cruzada para 3 versiones de los dígitos del Lenguaje de Señas, estos son los resultados.

5.4.3. Clasificación a partir de firmas de correlación

En la Figura 5.7 se muestran las firmas de correlación digital de los dígitos del 1 al 9 correspondientes a la version 2, la versión 1 y 3 se incluyen también para la clasificación con redes neuronales artificales como vectores descriptores de los dígitos del alfabeto del lenguaje de señas. A continuación se muestra la matriz de confusión y el porcentaje de clasificación en Tabla 5.5.

Matriz de Confusión

a	b	c	d	e	f	g	h	i	clasificado como
3	0	0	0	0	0	0	0	0	|a=1
0	3	0	0	0	0	0	0	0	|b=2
0	0	3	0	0	0	0	0	0	|c=3
0	0	0	1	0	1	1	0	0	|d=4
0	0	0	1	2	0	0	0	0	|e=5
0	0	0	0	0	3	0	0	0	|f=6
0	0	0	0	0	0	2	0	1	|g=7
0	0	0	0	0	0	0	3	0	|h=8
0	0	0	1	0	1	0	0	1	|i=9
									Porcentaje de Clasificación
									77.7778 %

Tabla 5.1: Matriz de Confusión para la V1 V2 y V3 de las firmas de correlación como vectores descriptores del sujeto 1.

Matriz de Confusión

a	b	c	d	e	f	g	h	i	clasificado como
3	0	0	0	0	0	0	0	0	|a=1
0	3	0	0	0	0	0	0	0	|b=2
0	0	3	0	0	0	0	0	0	|c=3
0	0	0	3	0	0	0	0	0	|d=4
0	0	0	0	3	0	0	0	0	|e=5
0	0	0	0	0	2	0	0	1	|f=6
0	0	0	0	0	0	3	0	0	|g=7
0	0	0	0	0	0	0	3	0	|h=8
0	0	0	0	0	0	0	0	3	|i=9
									Porcentaje de Clasificación
									96.2963 %

Tabla 5.2: Matriz de Confusión para la V1 V2 y V3 de las firmas de correlación como vectores descriptores del sujeto 2.

Matriz de Confusión

a	b	c	d	e	f	g	h	i	clasificado como
3	0	0	0	0	0	0	0	0	|a=1
0	3	0	0	0	0	0	0	0	|b=2
0	1	1	0	1	0	0	0	0	|c=3
0	0	0	2	0	0	0	0	1	|d=4
0	0	0	0	3	0	0	0	0	|e=5
0	0	0	0	0	2	0	0	1	|f=6
0	0	1	0	0	0	2	0	0	|g=7
0	0	1	0	0	0	1	1	0	|h=8
0	1	0	0	0	0	0	0	2	|i=9
									Porcentaje de Clasificación
									70.3704 %

Tabla 5.3: Matriz de Confusión para la V1 V2 y V3 de las firmas de correlación como vectores descriptores del sujeto 3.

Matriz de Confusión

a	b	c	d	e	f	g	h	i	clasificado como
3	0	0	0	0	0	0	0	0	|a=1
0	3	0	0	0	0	0	0	0	|b=2
0	0	3	0	0	0	0	0	0	|c=3
0	0	0	3	0	0	0	0	0	|d=4
0	0	0	0	3	0	0	0	0	|e=5
0	0	0	0	0	3	0	0	0	|f=6
0	0	0	0	0	0	3	0	0	|g=7
0	0	0	0	0	0	0	3	0	|h=8
0	0	0	0	0	0	0	0	3	|i=9
									Porcentaje de Clasificación
									100 %

Tabla 5.4: Matriz de Confusión para la V1 V2 y V3 de las firmas de correlación como vectores descriptores del sujeto 4.

Matriz de Confusión

a	b	c	d	e	f	g	h	i	clasificado como
3	0	0	0	0	0	0	0	0	|a=1
0	3	0	0	0	0	0	0	0	|b=2
0	0	1	0	0	0	2	0	0	|c=3
0	0	0	2	1	0	0	0	0	|d=4
0	0	1	0	2	0	0	0	0	|e=5
1	0	0	0	0	2	0	0	0	|f=6
0	0	2	0	0	0	0	0	1	|g=7
0	0	0	0	0	0	1	2	0	|h=8
0	0	0	0	0	0	1	0	2	|i=9
									Porcentaje de Clasificación
									62.96 %

Tabla 5.5: Matriz de Confusión para la V1 V2 y V3 de las firmas de correlación como vectores descriptores del sujeto 5.

Sujeto	Porcentaje de clasificación sujeto 1
1	77.7778 %
2	96.2963 %
3	70.3704 %
4	100 %
5	62.96 %
Promedio	81.4809 %

Tabla 5.6: Tabla del promedio de porcentajes de clasificación para la base de datos de la Figura 4.25.

5.4.4. Clasificación de descriptores basados en pendientes para los dígitos del 1 al 9

Sea $M_k^{persona,version,digito}$ un vector descriptor propuesto de la Figura 5.3, sobre el cual se calcula la transformada wavelet haar 1D, de nvel γ de la Ec. (2.37),

$$M_k^{persona,version,digito} \underset{\rightarrow}{H_\gamma} (a^\gamma|d^\gamma|d^{\gamma-1}|...|d^1).$$

El primer vector promedio o de tendendencia de nivel de transformación γ es usado como vector de entrada en un perceptrón multicapa descrito en la sección 3.4.1 definido como $X^\phi = a_\gamma^{persona,version,digito}$. En general, el conjunto de entradas consta de una persona $p = 1$, tres versiones $v = 3$, y nueve dígitos $d = 9$, esto es, $\phi = (persona)(version)(digito) = 27$ entradas y nueve clases $clases = 9$ en el perceptrón multicapa. En la Tabla 5.11, se muestra la matriz de confusión y debajo el porcentaje de clasificación con validación cruzada.

Matriz de Confusión

a	b	c	d	e	f	g	h	i	clasificado como
3	0	0	0	0	0	0	0	0	|a=1
0	3	0	0	0	0	0	0	0	|b=2
0	0	3	0	0	0	0	0	0	|c=3
0	0	0	3	0	0	0	0	0	|d=4
0	0	0	0	3	0	0	0	0	|e=5
0	0	0	0	0	3	0	0	0	|f=6
0	0	0	0	0	0	3	0	0	|g=7
0	0	0	0	0	0	0	3	0	|h=8
0	0	0	0	0	0	0	0	3	|i=9
									Porcentaje de Clasificación
									100 %

Tabla 5.7: Matriz de Confusión para el sujeto 1 de la relación de pendientes con transformada wavelet nivel 7 como vectores descriptores.

Matriz de Confusión

a	b	c	d	e	f	g	h	i	clasificado como
3	0	0	0	0	0	0	0	0	|a=1
0	3	0	0	0	0	0	0	0	|b=2
0	0	3	0	0	0	0	0	0	|c=3
0	0	0	3	0	0	0	0	0	|d=4
0	0	0	0	3	0	0	0	0	|e=5
0	0	0	0	0	1	2	0	0	|f=6
0	0	0	0	0	0	3	0	0	|g=7
0	0	0	0	0	0	1	2	0	|h=8
0	0	0	0	0	0	0	0	3	|i=9
									Porcentaje de Clasificación
									88.8889 %

Tabla 5.8: Matriz de Confusión para el sujeto 2 de la relación de pendientes con transformada wavelet nivel 7 como vectores descriptores.

Matriz de Confusión

a	b	c	d	e	f	g	h	i	clasificado como
2	0	0	0	0	0	0	0	1	|a=1
0	3	0	0	0	0	0	0	0	|b=2
0	0	3	0	0	0	0	0	0	|c=3
0	0	0	3	0	0	0	0	0	|d=4
0	0	0	0	3	0	0	0	0	|e=5
0	0	1	0	0	1	1	0	0	|f=6
0	0	0	0	0	0	3	0	0	|g=7
0	0	0	0	0	0	1	2	0	|h=8
0	0	0	0	0	1	0	0	2	|i=9
									Porcentaje de Clasificación
									81.4815 %

Tabla 5.9: Matriz de Confusión para el sujeto 3 de la relación de pendientes con transformada wavelet nivel 7 como vectores descriptores.

Matriz de Confusión

a	b	c	d	e	f	g	h	i	clasificado como
3	0	0	0	0	0	0	0	0	|a=1
0	3	0	0	0	0	0	0	0	|b=2
0	0	3	0	0	0	0	0	0	|c=3
0	0	0	2	0	0	1	0	0	|d=4
0	0	0	0	3	0	0	0	0	|e=5
0	0	0	0	0	3	0	0	0	|f=6
0	0	0	0	0	0	3	0	0	|g=7
0	0	0	1	0	0	0	2	0	|h=8
0	1	1	0	0	0	0	0	1	|i=9
									Porcentaje de Clasificación
									85.1852 %

Tabla 5.10: Matriz de Confusión para el sujeto 4 de la relación de pendientes con transformada wavelet nivel 7 como vectores descriptores.

Matriz de Confusión

a	b	c	d	e	f	g	h	i	clasificado como
3	0	0	0	0	0	0	0	0	|a=1
0	3	0	0	0	0	0	0	0	|b=2
0	0	3	0	0	0	0	0	0	|c=3
0	0	0	3	0	0	0	0	0	|d=4
0	0	0	0	3	0	0	0	0	|e=5
0	0	0	0	0	1	2	0	0	|f=6
0	0	0	0	0	0	3	0	0	|g=7
0	0	0	0	0	0	0	3	0	|h=8
0	0	0	0	0	0	0	0	3	|i=9
									Porcentaje de Clasificación
									92.5926 %

Tabla 5.11: Matriz de Confusión para el sujeto 5 de la relación de pendientes con transformada wavelet nivel 7 como vectores descriptores.

Sujeto	Porcentaje de clasificación sujeto 1
1	100 %
2	88.8889 %
3	81.4815 %
4	85.1852 %
5	92.5926 %
Promedio	89.6296 %

Tabla 5.12: Tabla del promedio de porcentajes de clasificación de la relación de pendientes para la base de datos de la Figura 4.25.

5.4.5. Clasificación de descriptores basados en wavelets para los dígitos estáticos del alfabeto del lenguaje de señas

Antes de iniciar la transformación wavelet, las imágenes fueron preprocesadas para obtener la mayor información posible, de modo que se extrajo el objeto y después fue reescalado para que tuviera una medida adecuada para la transformación, tal como se muestra en la Figura 5.11, posteriormente se aplica la transformación wavelet para después extraer los niveles del 5 al 8. El procedimiento se encuentra a continuación.

Sea $\delta^{x,y}$ una imagen de tamaño $\left(\frac{N}{2^x}\right)\left(\frac{N}{2^y}\right)$ sobre la cual se extrae sólo el dígito de interés, para reescalarlo a una matriz descriptor de tamaño $2^k x 2^k$ como se muestra en la Figura 5.11. Finalmente es convertido a un vector $X^{\phi} = \delta_{\gamma}^{persona,version,digito}$.

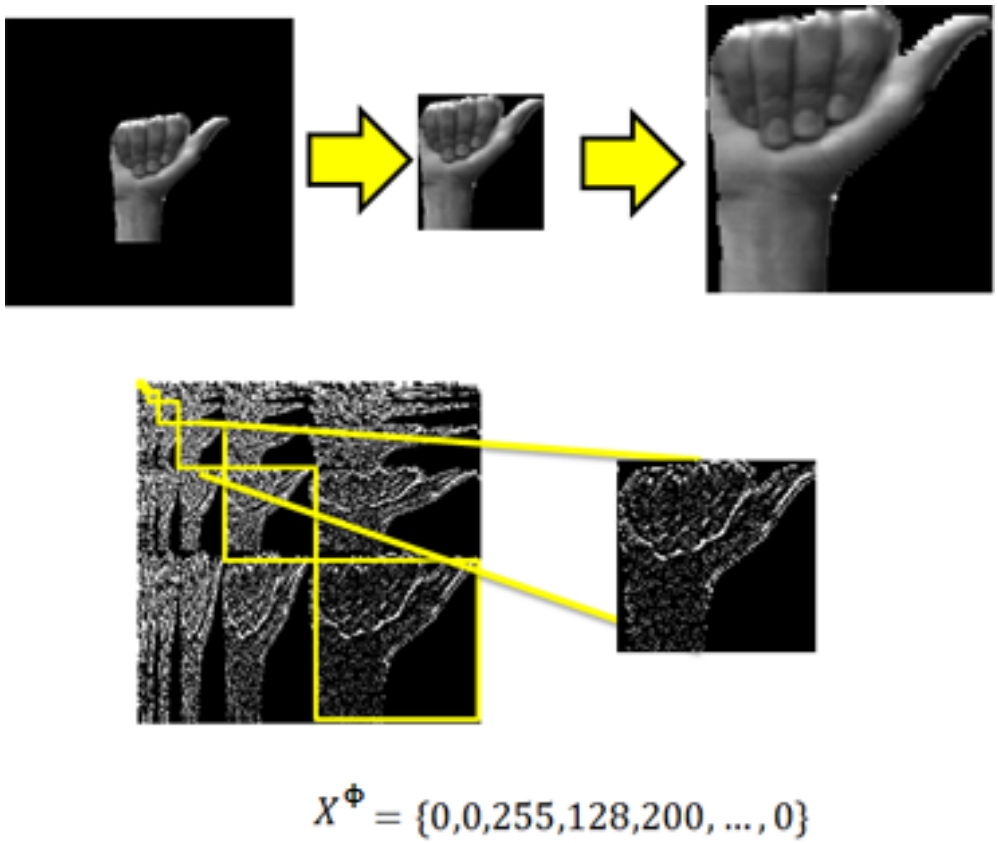

figura 5.11: Preprocesado de 8 niveles de transformación wavelet Haar. Los valores de la imagen recortada se convierten en un vector descriptor.

El conjunto de entradas consta de las bases de datos de las Figuras 4.24, 4.25 y 4.26, con $p = 1$, $v = 3$, y $d = 23$, esto es, $\phi = (persona)(version)(digito) = 69$ entradas y $clases = 23$ en el perceptrón multicapa para cada base. Adelante se muestran las matrices de confusión y el porcentaje de clasificación con validación cruzada para cada base de imágenes.

Para la base de datos filtrada de las Figuras 4.25, 4.24 y 4.26 respectivamente, los porcentajes de clasificación son,

Nivel de transformación Wavelet	Porcentaje de clasificación
5	60.8696 %
6	73.913 %
7	76.8116 %
8	60.8696 %

Tabla 5.13: Tabla de porcentajes de clasificación para la base de datos de la Figura 4.25.

Nivel de transformación Wavelet	Porcentaje de clasificación
5	75.3623 %
6	69.5652 %
7	84.058 %
8	68.1159 %

Tabla 5.14: Tabla de porcentajes de clasificación para la base de datos de la Figura 4.24.

Nivel de transformación Wavelet	Porcentaje de clasificación
5	75.3623 %
6	85.50752 %
7	85.5072 %
8	76.8116 %

Tabla 5.15: Tabla de porcentajes de clasificación para la base de datos de la Figura 4.26.

A continuación se muestran las matrices de confusión para los Niveles 7 de las Figuras 4.25 y 4.24 que resultó obtener el porcentaje de clasificación más alto, y la matriz de confusión del Nivel 6 para 4.26.

Matriz de Confusión

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	clasificado como
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	| a = A
0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	| b = B
0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	| c = C
0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	| d = D
0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	| e = E
0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	| f = F
0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	| g = G
0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	| h = H
0	0	0	0	0	1	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	| i = I
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	1	| j = K
0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	| k = L
0	0	0	0	0	0	0	0	0	0	0	2	1	0	0	0	0	0	0	0	0	0	0	| l = M
0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	| m = N
0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	| n = O
0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	| o = P
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	| p = R
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0	0	0	0	| q = S
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	0	0	0	0	0	| r = T
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	1	| s = U
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	| t = V
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	| u = W
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	2	0	| v = X
0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	1	0	0	0	0	| w = Y
																							Porcentaje de Clasificación
																							76.8116 %

Tabla 5.16: Tabla de porcentajes de clasificación para la base de datos de la Figura 4.25. Notese como las mayores discrepancias, es decir, los dígitos del alfabeto más dificles de clasificar son la 'M' y la 'N', en segundo lugar se encuentran la 'T' y la 'S', esto sucede para todos los casos.

Matriz de Confusión

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	clasificado como
2	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	| a = A
0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	| b = B
0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	| c = C
0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	| d = D
0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	| e = E
0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	| f = F
0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	| g = G
0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	| h = H
0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	| i = I
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	0	0	| j = K
0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	| k = L
0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	| l = M
0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	1	0	0	0	0	0	0	| m = N
0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	| n = O
0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	1	0	0	| o = P
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	1	0	0	0	0	| p = R
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	1	0	0	0	0	| q = S
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	| r = T
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	2	0	0	0	| s = U
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	| t = V
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	| u = W
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	| v = X
0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	| w = Y
																							Porcentaje de Clasificación
																							84.058 %

Tabla 5.17: Tabla de porcentajes de clasificación para la base de datos de la Figura 4.24. Notese como las mayores discrepancias, es decir, los dígitos del alfabeto más dificles de clasificar son la 'M' y la 'N', en segundo lugar se encuentran la 'T' y la 'S', esto sucede para todos los casos.

Matriz de Confusión

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	clasificado como
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	| a = A
0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	| b = B
0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	| c = C
0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	| d = D
0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	| e = E
0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	| f = F
0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	| g = G
0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	| h = H
0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	1	0	0	0	0	0	0	0	| i = I
0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	1	0	0	0	| j = K
0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	| k = L
0	0	0	0	0	0	0	0	0	0	0	2	1	0	0	0	0	0	0	0	0	0	0	| l = M
0	0	0	0	0	0	0	0	0	0	0	1	2	0	0	0	0	0	0	0	0	0	0	| m = N
0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	1	0	| n = O
0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	| o = P
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	| p = R
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	| q = S
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	| r = T
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	| s = U
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	2	0	0	0	| t = V
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	| u = W
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	2	0	| v = X
0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	| w = Y
																							Porcentaje de Clasificación
																							85.5072 %

Tabla 5.18: Tabla de porcentajes de clasificación para la base de datos de la Figura 4.26. Notese como las mayores discrepancias, es decir, los dígitos del alfabeto más dificles de clasificar son la 'M' y la 'N', en segundo lugar se encuentran la 'T' y la 'S', esto sucede para todos los casos.

Claramente, la base de datos que mejores resultados ha generado, ha sido la de las imagenes binarias solidas de la Figura 4.26, por este motivo, se aumenta la base de datos de las imágenes binarias solidas conformado un total de tres sujetos con tres versiones diferentes del mismo alfabeto, una muestra de estas imágenes se presenta en la Figura 5.12. A continuación se muestran los resultados de clasificación de los niveles wavelet 5, 6, 7 y 8 para tres sujetos diferentes y la tabla de sus promedios.

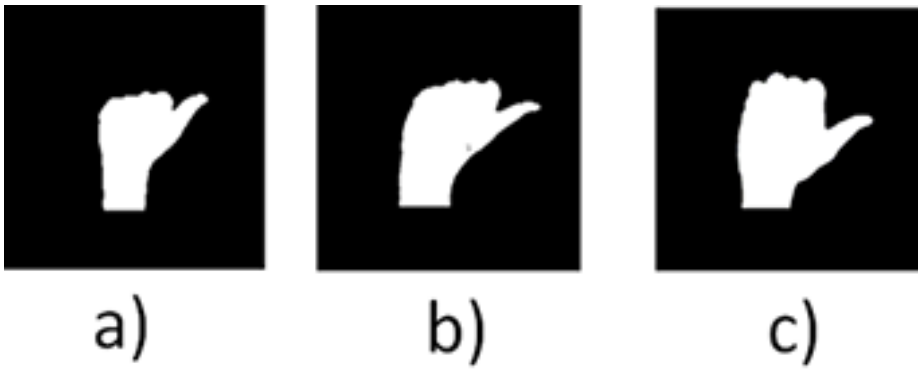

figura 5.12: Versiones 1 de la letra A del a) sujeto 1 b) sujeto 2 c) sujeto 3.

Nivel de transformación Wavelet	Porcentaje de clasificación sujeto 1
5	75.3623 %
6	85.50752 %
7	85.5072 %
8	76.8116 %

Tabla 5.19: Tabla de porcentajes de clasificación del sujeto uno para la base de datos de la Figura 4.25.

Nivel de transformación Wavelet	Porcentaje de clasificación sujeto 1
5	73.913 %
6	86.9565 %
7	85.5072 %
8	69.5652 %

Tabla 5.20: Tabla de porcentajes de clasificación del sujeto dos para la base de datos de la Figura 4.25.

Nivel de transformación Wavelet	Porcentaje de clasificación sujeto 1
5	68.1159 %
6	78.2609 %
7	82.6087 %
8	60.8696 %

Tabla 5.21: Tabla de porcentajes de clasificación del sujeto tres para la base de datos de la Figura 4.25.

Nivel de transformación Wavelet	Porcentaje de clasificación sujeto 1
5	72.4637 %
6	83.5750 %
7	84.5410 %
8	69.0821 %

Tabla 5.22: Tabla del promedio de porcentajes de los sujetos de clasificación para la base de datos de la Figura 4.26.

Un dato extra se muestra en la tabla 5.23, donde se muestran los resultados de clasificación cuando uno de los sujetos utiliza un anillo y un reloj (a) de la Figura 5.13); notése que en comparativa con los demás resultados, el uso de objetos que obstruyan la visibilidad de la cámara térmica afecta fuertemente el resultado de clasificación.

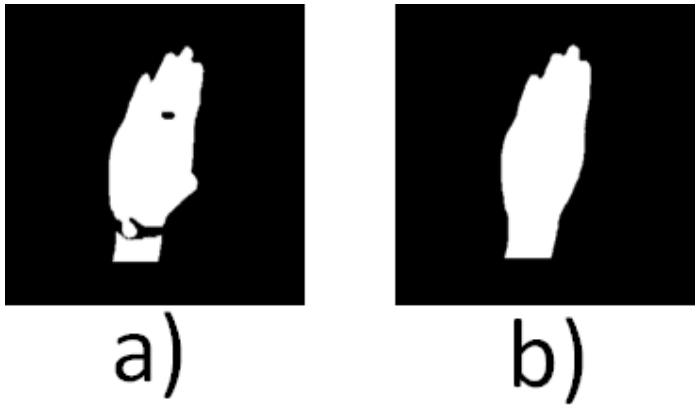

figura 5.13: Letra B del alfabeto del lenguaje de señas; a) imagen con ruido b) imagen ideal.

Nivel de transformación Wavelet	Porcentaje de clasificación sujeto 1
5	37.6812 %
6	33.3333 %
7	26.087 %
8	13.0435 %

Tabla 5.23: Tabla de porcentajes de clasificación de un sujeto extra para la base de datos de la Figura 4.26 con ruido.

5.5. Interfaz para el reconocimiento de digitos

Se programó una interfaz (que aún se encuentra en etapa de desarrollo) para la comunicación dinámica entre la computadora y la cámara térmica en el lenguaje MATLAB, cuyo entorno contiene un apartado para la creación de aplicaciones orientadas a objetos, sobre el cual se diseñó y desarrolló esta interfaz Figura 5.14 capaz de procesar las imágenes de la cámara térmica en tiempo real. Esta interfaz realiza un recorte de la mano cuya finalidad es la de realizar una firma de correlación. Esta interfaz además puede de realizar una correlación lógica (Ec. 2.30) entre una base de datos para los dígitos del lenguaje de señas del 1 al 9 y la imagen recibida de la cámara térmica, siendo así capaz de reconocer los dígitos del 1 al 9 cuando dicha correlación es mayor a un umbral definido por el usuario. El objetivo de la aplicación además del reconocimiento dinámico, es el de familiarizar al usuario con los dígitos del lenguaje de señas.

figura 5.14: Aplicación para el reconocimiento dinámico de los dígitos del Lenguaje de Señas.

En la Figura 5.14 se muestran las partes de la interfaz: a) cuadro de desplegado de correlación, b) Imagen correspondiente al dígito de la base de datos a correlacionar (rojo),

c) Región qe comparten ambas imágenes (blanco), d) Imagen en tiempo real de la mano adquirida por la cámara térmica, e) Signo númerico del Lenguaje de Señas actual, f) Botón para la conexión entre la computadora y la cámara térmica.

5.6. Conclusiones

Se analizarón tres métodos para la extracción de características: A) Descriptores de digitos basados en Pendientes, B) Firmas de correlación y C) descriptores Wavelet. A partir de los descriptores basados en pendientes para el reconocimiento de los dígitos del 1 al 9, de la Figura 5.3, se buscó la clasificación usando el método de correlación de Pearson. Los resultados se muestran en la Figura 5.10 dando como resultado un porcentaje de clasificación del 88 %. Debido a que el tamaño de los decriptores $M^{p,v,d}$ es grande, se buscó reducir el vector de elementos usando transfomada wavelet Haar 1D de $\gamma = 6, 7$. Cada señal $a^{\gamma}(persona, version, digito)$ fue clasificada usando tanto el correlador de Pearson como un perceptrón multicapa. Los porcentajes de clasificación de los dígitos del 1 al 9 usando: correlador de Pearson y un perceptrón multicapa se resumen en la siguiente tabla.

Descriptores / Clasificación	Correlador de Pearson	Perceptrón Multicapa
Descriptores basados en pendientes	88 %	Vectores muy grandes
Firmas de correlación	88 %	77.77 %
Descriptores pendientes + TW con $\gamma = 7$	No necesario	89.6296 %

Tabla 5.24: Tabla de porcentajes de clasificación para los dígitos del Lenguaje de Señas.

A partir de los descriptores Wavelet de la Figura 5.9, y el preprocesado de la Figura 5.11 se buscó la clasificación usando un perceptrón multicapa. Los porcentajes de clasificación de los signos del alfabeto usando una Red Neuronal Artificial con validación cruzada se muestran en la siguiente tabla.

B.D. del alfabeto formada con sub-imágenes de la T.W.	Porcentaje de clasificación
A. Imágenes segmentadas en 256 niveles de gris	84 %
B. Imágenes Binarias	85.5 %
C. Imágenes filtradas de A	76.81 %

Tabla 5.25: Tabla de porcentajes de clasificación para los dígitos del alfabeto del Lenguaje de Señas. Resultados a partir de las transformaciones wavelet de nivel $\gamma = 6$.

Finalmente se programó una interfaz en MATLAB para la adquisición de imágenes de una cámara térmica, el procesado y reconocimiento de dígitos del 1 al 9 usando correlación lógica

Bibliografía

[1] A. Padilla V. Gonzalo A. Cornejo J. Báez ". Correlation based rotational signature of planar binary objects ". Applications of digital image processing XXVII *Proceedings of SPIE* Vol. 5558.

Capítulo 6

Conclusiones

Se implementó un sistema de visión por computadora para el análisis y reconocimiento de los dígitos y alfabeto del lenguaje de señas americano para el caso estático. Las imágenes digitales fueron adquiridas por sensores que captan en el visible e infrarrojo. Lo anterior con el fin de obtener una segmentación de la información sin la necesidad de ítems especiales tales como marcadores, fondos y ropa especial, cables, entre otros, que imposibiliten la libertad de movimientos del interlocutor. Con el sistema propuesto, se generaron cuatro bases de datos, una para los dígitos y otras tres para el alfabeto del LSA.

Para el análisis, segmentación, descripción y clasificación de estas señas, se propusieron y programaron diferentes algoritmos, dentro de estos, también se encuentran aquellos basados en convolución para el mejoramiento de imágenes y la eliminación de ruido.

En el capítulo dos, se analizan técnicas clásicas para el procesamiento de imágenes digitales que permiten la segmentación y delimitación de una región, con el fin de obtener la representación de un contorno a través de cadenas de código. El análisis de la silueta del contorno es propuesto como descriptor de un objeto en la imagen. Debido a la gran cantidad de datos, que estos descriptores pueden arrojar, se propuso la compresión de la información utilizando la transformación wavelet Haar de k niveles. Finalmente como algoritmo de reconocimiento, se estudió el correlador de Pearson y el correlador lógico.

En el capítulo tres se aborda el estudio de las redes neuronales artificiales para la clasificación de patrones. Dos tipos de modelos son analizados: a) El perceptrón simple, que es un método de clasificación muy popular pero limitado por la cantidad de entradas y clases que este puede manejar. Y b) el perceptrón multicapa que permite una clasificación mucho más precisa al delimitar regiones entre clases además del adecuado manejo del número de entradas que puede procesar. El tipo de entrenamiento propuesto es supervisado, con un factor de aprendizaje $\alpha = 0{,}3$ y un momentum $\beta = 0{,}2$, con una capa oculta y una función de activación tipo sigmoidal acotada entre 0 y 1.

En el capítulo cuatro se describen los sistemas de adquisición imágenes para las bases de datos de los dígitos y el alfabeto del Lenguaje de Señas. Estos sistemas tienen dos configuraciones: a) Una cámara térmica y b) Una cámara térmica y una cámara que sensa en espectro

visible. En esta última configuración, ambas cámaras captura la misma escena en el mismo momento, lo que permite una relación entre las imágenes que sirve para realizar la fusión de las mismas y así lograr una segmentación digital adecuada.

La primera base de datos mostrada en la Figura 4.16, está conformada por imágenes binarias de nueve dígitos, con tres versiones para una persona. Las tres bases de datos para los veintitrés signos del alfabeto del LSA, para el caso estático y con tres versiones para una persona, son generadas de tres formas diferentes: a) Imágenes en niveles de gris mostradas en la Figura 4.24, de la cámara en el visible pero segmentada por la cámara térmica, b) Imágenes binarias sólidas mostradas en la Figura 4.26, obtenidas de la cámara térmica, y c) Imágenes en niveles de gris de la Figura 4.25, pero procesadas para extraer únicamente los contornos.

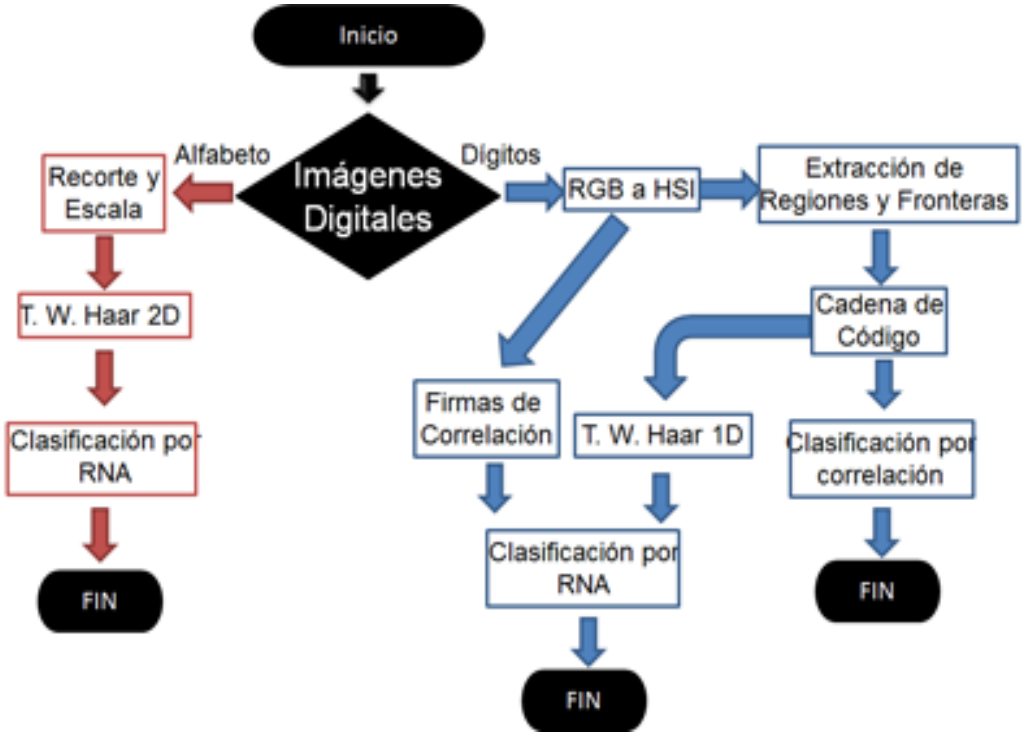

figura 6.1: Algoritmo general. A partir de las imágenes digitales, hacia la derecha (azul) se muestra el procedimiento para la obtención de la base de datos de los dígitos, su procesamiento y clasificación. De las imágenes digitales, hacia la izquierda (rojo) se muestra el procedimiento para la obtención de la base de datos del alfabeto, su procesamiento y clasificación. Ambas para los casos estáticos del Lenguaje de Señas Americano.

Los métodos y resultados de clasificación son agrupados en el capítulo cinco. Usando los descriptores basados en las pendientes de una silueta y un perceptrón multicapa se logró una clasificación del 100 %. Cabe destacar, que lo anterior se logró con treinta descriptores. Esto es que de la imagen original en 2D, se convierte a una señal 1D y después esta es comprimida

usando transformación wavelet de nivel 7. Los porcentajes de clasificación de los dígitos del 1 al 9 usando: correlador de Pearson y un perceptrón multicapa se resumen en la siguiente tabla..

Descriptores / Clasificación	Correlador de Pearson	Perceptrón Multicapa
Descriptores basados en pendientes	88 %	Vectores muy grandes
Firmas de correlación	88 %	81.4809 %
Descriptores pendientes + TW con $\gamma = 7$	No necesario	89.6296 %

Tabla 6.1: Tabla de porcentajes de clasificación para los dígitos del Lenguaje de Señas.

A partir de los descriptores Wavelet de la Figura 5.9, y el preprocesado de la Figura 5.11 se buscó la clasificación usando un perceptrón multicapa. Los porcentajes de clasificación de los signos del alfabeto usando una Red Neuronal Artificial con validación cruzada se muestran en la siguiente tabla.

B.D. del alfabeto formada con sub-imágenes de la T.W.	Porcentaje de clasificación
A. Imágenes segmentadas en 256 niveles de gris	84 %
B. Imágenes Binarias	85.5 %
C. Imágenes filtradas de A	76.81 %

Tabla 6.2: Tabla de porcentajes de clasificación para los dígitos del alfabeto del Lenguaje de Señas. Resultados a partir de las transformaciones wavelet de nivel $\gamma = 6$.

El porcentaje de clasificación más alto que es de 85.5 % se obtuvo a partir de las imágenes del alfabeto en versión binaria de la Figura 4.26, cabe señalar que todas las imágenes provienen del mismo ángulo de visión entre la cámara y la persona, existen trabajos con un porcentaje de clasificación menor a pesar de que la captura de las imágenes de los signos del alfabeto fueron obtenidas desde diferentes ángulos y posiciones [Karami 2011].

El porcentaje de clasificación del 84 % de las imágenes segmentadas en 256 niveles de gris, que es el segundo resultado más alto, es comparable al obtenido en el trabajo de [Karami 2011]. Se concluye que la base de datos más desciptiva es la que pertenece a la Figura 4.26 que corresponde a la serie de imágenes binarias solidas en el septimo nivel wavelet, por lo que se realiza el mismo experimento para tres sujetos diferentes. Finalmente se obtiene un promedio de clasificación de **84.5410 %** para los niveles wavelet siete de los tres sujetos.

Un dato extra se muestra en la siguiente tabla, donde se muestran los resultados de clasificación cuando uno de los sujetos utiliza un anillo y un reloj; notése que en comparativa con los demás resultados, el uso de objetos que obstruyan la visibilidad de la cámara térmica afecta fuertemente el resultado de clasificación.

Nivel de transformación Wavelet	Porcentaje de clasificación sujeto 1
5	37.6812 %
6	33.3333 %
7	26.087 %
8	13.0435 %

Tabla 6.3: Tabla de porcentajes de clasificación de un sujeto extra para la base de datos de la Figura 4.26 con ruido.

Otro punto importante es que el algoritmo no esta limitado a un fondo, ropa especial, o dispositivos electrónicos que ayuden a aumentar la eficiencia del algoritmo. En la siguiente tabla se muestran las ventajas y desventajas de nuestra propuesta comparada con las propuestas que se encuentran en el estado del arte.

Ventajas	Desventajas
Número de cámaras	Variación de la temperatura
No fondo ni ropa especial	Separación mano-cara
Número pequeño de descriptores	
No dispositivos electrónicos	

Tabla 6.4: Tabla de porcentajes de clasificación para los dígitos del alfabeto del Lenguaje de Señas. Resultados a partir de las transformaciones wavelet de nivel $\gamma = 6$.

Se programó una interfaz en MATLAB para la adquisición de imágenes de una cámara térmica, el procesado y reconocimiento de dígitos del 1 al 9 usando correlación lógica.

Como trabajo a futuro, se propone una caracterización del usuario previo al reconocimiento del LSA, estudiar el modelo de oculto de Markov para el reconocimiento dinámico de los signos que llevan movimiento y la paralelización de algunos algoritmos de procesamiento y preprocesamiento de información.

Apéndice A

Espacio de color HSI

La descripción RGB (del inglés Red, Green, Blue; rojo, verde, azul") de un color hace referencia a la composición del color en términos de la intensidad de los colores primarios con que se forma: el rojo, el verde y el azul. Las imágenes que capturan las cámaras dígitales son usualmente en este espacio de color compuesto por las tres matrices antes mencionadas. HSI es otro espacio de color, y para llegar a este desde RGB, el tono (H), la saturación (S) y la intensidad (I) estan dados en términos de las ecuaciones definidas como,

$$I = \frac{1}{3}(R + G + B), \tag{A.1}$$

$$S = 1 - \frac{3\,\text{mín}(R + G + B)}{R + G + b}, \tag{A.2}$$

$$H = \cos^{-1}\left[\frac{\frac{1}{2}[(R - G) + (R - B)]}{\sqrt{(R - G)^2 + (R - B)(G - B)}}\right], \tag{A.3}$$

para $0 \leq H \leq 180$

Un ejemplo se muestra a continuación,

figura A.1: a) Imagen en RGB y sus componentes correspondientes al espacio HSI, b) (H) tono, c) (S) saturación y d) (I) Intensidad.

Apéndice B

Biblioteca de Filtros

Para la obtención de la base de datos 4.25, fue realizado un filtraje pasaaltas y pasabajas, como está definido en la sección 2.7.1, por medio de convolución con los siguientes filtros,

$$filtro1 = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad filtro2 = \begin{bmatrix} 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & 1 & 0 \end{bmatrix} \quad filtro3 = \frac{1}{25}\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$filtro4 = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad filtro5 = \begin{bmatrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{bmatrix} \quad filtro6 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$filtro3 = \frac{1}{4}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

Apéndice C

Operaciones Morfológicas

C.0.1. Erosión

Erosión de la imagen A por el elemento estructural (EE) C está dada por,

$$\xi_C(A) = A \ominus C = \{x | C_x \subset A\} \tag{C.1}$$

Suponiendo que el EE contiene al origen y es simétrico, la salida de la erosión es el conjunto de puntos barridos por el centro del EE mientras se cumpla que todos los puntos de C estaban contenidos en A. La erosión elimina grupos de píxeles donde el EE no cabe como pequeñas islas y protuberancias. La erosión es antiextensiva, es decir, que reduce el tamaño del objeto. Un ejemplo de la erosión se expone en la Figura C.1.

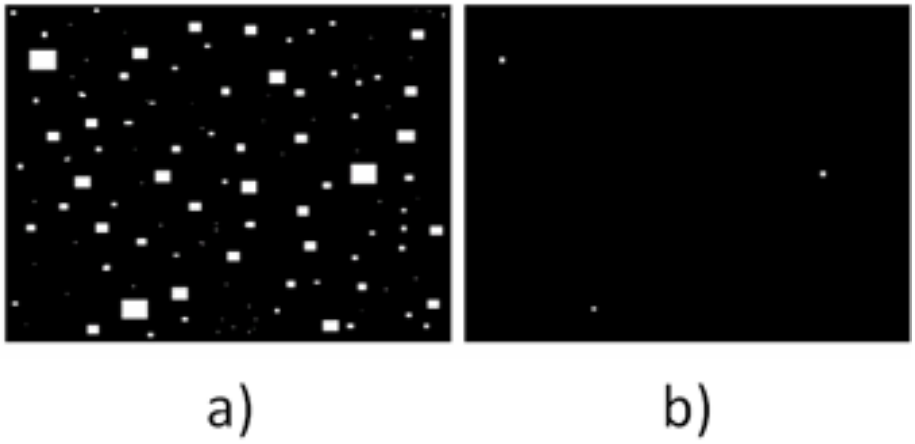

figura C.1: a) Imagen original, b) Imagen erosionada, notese que en b) los conjuntos de pequeños objetos desaparecen por completo y solo quedan fragmentos de los objetos más grandes.

C.0.2. Dilatación

Dilatación de la imagen A por el elemento estructural C está dada por,

$$\delta_c(A) = A \oplus C = \{x|(\hat{C})_x \cap A \neq 0\} \tag{C.2}$$

Suponiendo que el EE contiene al origen y es simétrico (en este caso el EE y su reflexión son iguales): la salida de la dilatación es el conjunto de puntos barridos por el centro del EE mientras algún punto de C tocaba a algún punto de A. La dilatación añade todos los puntos del fondo que tocan el borde de un objeto. La dilatación es extensiva, es decir, que rellena entrantes en los que no quepa el EE como pequeños agujeros y bahías. Un ejemplo de la dilatación se expone en la Figura C.2.

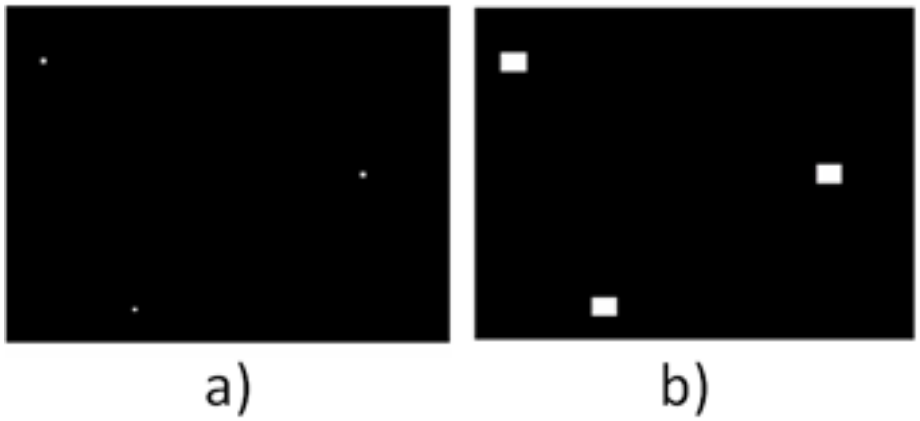

figura C.2: a) Imagen original, b) Imagen original dilatada, notese que en b) los conjuntos de pequeños objetos que aparecen en a) son expandidos convirtiendose en objetos más grandes.

Apéndice D

Momentos geométricos

Sea $f(x, y)$ una imagen digital binaria Figura D.1,

figura D.1: Imagen binaria Digito 1, Versión 1, Persona 1.

sobre la cual se calculan los momentos geométricos, $m_{p,q}$, de orden (p, q) como,

$$m_{p,q} = \sum_{x=1}^{M} \sum_{y=1}^{N} f(x, y) x^p y^q \tag{D.1}$$

donde MxN es el tamaño de la imagen. A partir de estas, es posible obtener el centroide (cx, cy) de un conjunto en la imagen como,

$$cx = \frac{m_{0,0}}{m_{1,0}} \tag{D.2}$$

$$cx = \frac{m_{0,0}}{m_{0,1}}. \tag{D.3}$$

Usando la transformación espacial,

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} \pm \begin{bmatrix} cx \\ cy \end{bmatrix}, \tag{D.4}$$

es posible desplegar el centro de la imagen al centroide del objeto. Un ejemplo se muestra en la Figura D.2.

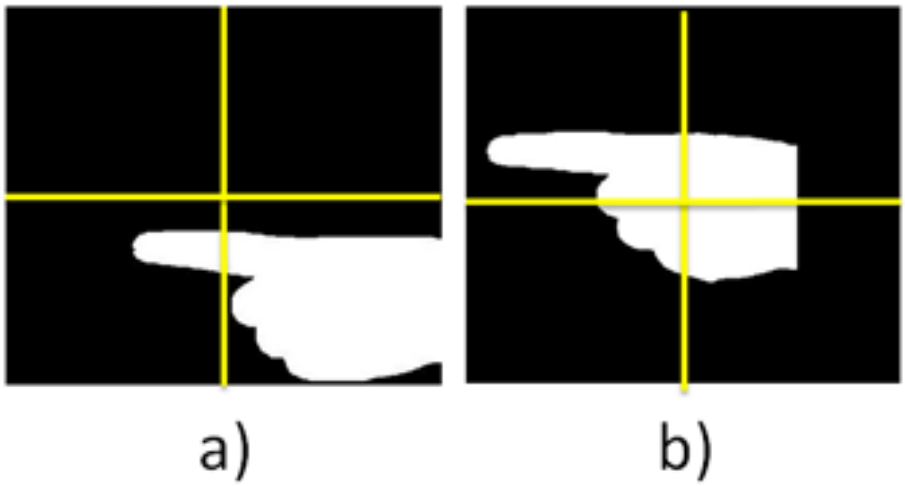

figura D.2: El centro de la imagen corresponde a la intersección de las lineas amarillas. a) Imagen binaria sin centrar, b) mima imagen binaria, pero con el centroide del objeto desplazado al centro de la escena.

Apéndice E

Trabajos derivados de la Tesis

Oscar Morales-Alvarez et al. **"Static sign language recognition using neural networks"**, **Proceedings of SPIE**, 8661 (2013) Aceptado para presentación.
Electronic Imaging Science and Technology
Information Systems and Technology
Image Processing: Machine Vision Applications VI
California (EUA)
Febrero de 2013

José Francisco Solis-Villarreal, Oscar Morales-Alvarez **Reconocimiento del Lenguaje de Señas", UAEH** Presentado
Encuentro de investigación
Hidalgo - Pachuca (México)
Octubre (2012)

Oscar Morales-Alvarez et al. **"Sign language recognition"**, **UPT**, (2012) Presentado
Coloquio Internacional de Investigación
Universidad Politécnica de Tulancingo
Tulancingo de Bravo, Hgo. (México)
Septiembre (2012)

Oscar Morales-Alvarez et al. **"Software para la enseñanza de lenguaje de señas" UPT**, (2012) Presentado
Prototipos
Universidad Politécnica de Tulancingo
Tulancingo de Bravo, Hgo. (México)
Septiembre (2012)