

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA



PONTIFICIA
UNIVERSIDAD
CATÓLICA
DEL PERÚ

**DISEÑO E IMPLEMENTACIÓN DE UN MÓDULO DE
RECONOCIMIENTO DE NÚMEROS MANUSCRITOS**

Tesis para optar el Título de Ingeniero Electrónico, que presenta el
bachiller:

EDUART RUDOLF GARRIDO ROJAS

ASESOR: Ing. Pedro Moisés Crisóstomo Romero

Lima, Marzo del 2010

RESUMEN

Los sistemas de reconocimiento óptico de caracteres (OCR) constituyen un tema activo de investigación. El principal problema es el porcentaje de efectividad que se obtiene en el reconocimiento de caracteres manuscritos, esto es debido a la naturaleza de estos caracteres, la tipografía, la resolución de la imagen, iluminación y otros factores.

Los sistemas que reconocen números manuscritos tienen diversas aplicaciones (reconocimiento de cantidades numéricas sobre cheques bancarios, facturas, planillas de conteo de votos electorales, encuestas, etc.), para los cuales se necesita que la tasa de acierto al reconocer los números sea muy elevada (reconocimiento mayor 90% y error cercano 0%). Para lograr esto se implementó un módulo de reconocimiento basado en redes neuronales haciendo uso del toolbox de MATLAB.

El módulo de reconocimiento tiene las siguientes etapas: una primera etapa es la segmentación que se encarga de separar la imagen de valor numérico en dígitos individuales, la segunda etapa se encarga de la normalización para obtener muestras parecidas para lo cual se realiza corrección de la inclinación, espesor del trazo y tamaño, y una última etapa es la clasificación para lo cual se usó la red neuronal de topología “backpropagation”.

Luego de las pruebas respectivas se obtuvo un reconocimiento de 95.9% con un porcentaje de error 0.8%, estas pruebas se realizaron con números de buena escritura. Se realizó otras pruebas con una base de datos de números manuscritos (MNIST, base de datos del National Institute of Standards and Technology) con la cual se obtuvo un reconocimiento del 90.11% y error 3.67%.

Con esto se puede concluir que la metodología desarrollada es buena cuyas etapas se pueden optimizar para obtener mejores resultados.

ÍNDICE

<u>INTRODUCCIÓN</u>	1
<u>CAPÍTULO 1: LECTURA AUTOMÁTICA DE NÚMEROS</u>	
1.1 Usos de los sistemas de reconocimiento de números manuscritos	2
1.2 Características de los sistemas de reconocimiento de números manuscritos	2
1.3 Declaración del marco problemático	3
<u>CAPÍTULO 2: TÉCNICAS EMPLEADAS EN EL RECONOCIMIENTO DE NÚMEROS MANUSCRITOS</u>	
2.1 Estado del arte	5
2.1.1 Presentación del asunto de estudio	5
2.1.2 Estado de la investigación	6
2.1.2.1 Mejoramiento	6
2.1.2.2 Segmentación	7
2.1.2.4 Normalización	10
2.1.2.5 Reconocimiento	11
2.1.3 Síntesis sobre el asunto de estudio	14
2.2 Módulo de Reconocimiento:	15
2.2.1 Pre-procesamiento:	16
2.2.2 Redes Neuronales Artificiales	16
2.2.2.1 Características generales	16
2.2.2.2 Red Perceptrón	18
2.2.2.3 Red Perceptrón Multicapa(MLP)	20
2.2.2.4 Red Backpropagation	20
2.2.2.5 Modelo de Kohonen	21
2.3 Modelo Teórico	22
2.4 Definiciones operativas	24

CAPÍTULO 3: DISEÑO E IMPLEMENTACIÓN DEL SISTEMA BASADO EN REDES NEURONALES

3.1 Captura de la imagen	27
3.2 Procesamiento de la Imagen	28
3.2.1 Extracción de números	28
3.2.2 Segmentación	29
3.2.3 Normalización	33
3.2.3.1 Corrección de la inclinación	34
3.2.3.2 Cambio del tamaño	34
3.2.3.3 Normalización del grosor	35
3.3 Entrenamiento Red Neuronal Backpropagation	36
3.3.1 Regla de Aprendizaje	37
3.3.2 Estructura de la red neuronal	42
3.4 Reconocimiento	45

CAPÍTULO 4: PRUEBA Y ANÁLISIS DE LOS RESULTADOS

4.1 Entorno gráfico del módulo de reconocimiento	46
4.2 Análisis de Reconocimiento y Eficiencia	48
4.3 Distribución de aciertos en la salida	49
4.4 Comparación de resultados con otros trabajos	52
4.4.1 Pruebas con la base de datos MNIST	54

<u>CONCLUSIONES</u>	58
----------------------------	----

<u>RECOMENDACIONES</u>	60
-------------------------------	----

INTRODUCCIÓN

El constante uso de la tecnología para poder optimizar algunos procesos se presenta desde hace varios años y con ello la búsqueda de realizar sistemas que puedan hacer algunas tareas tales como las que puede realizar una persona. Dentro de este campo se encuentran los sistemas de reconocimiento automático de números manuscritos, los cuales se encargan de pasar a formato digital (código ASCII) los números que puedan ser escritos por personas [1]. Esta tarea puede resultar fácil para una persona entrenada pero para una computadora es complicado y se tiene que recurrir a una serie de técnicas que “simulan” algún proceso parecido que se realiza en el cerebro humano.

Existen algunos sistemas para reconocimiento de números manuscritos que son usados en aplicaciones específicas tales como la lectura de códigos postales y cantidades monetarias en cheques bancarios [2]. Estos sistemas pueden ser usados para el conteo de votos en actas electorales en procesos de elecciones, pero el costo de estos puede ser elevado aunque puede variar de acuerdo a la eficacia de cada uno de ellos. A pesar del notable progreso logrado en los últimos años en el área de reconocimiento automático de números manuscritos, todavía se está lejos de poder igualar la capacidad humana.

En el presente trabajo se implementa un módulo de reconocimiento basado en redes neuronales artificiales, haciendo uso del toolbox de redes neuronales del Matlab, la cual es una técnica que viene dando buenos resultados en este tipo de aplicaciones. El módulo planteado realiza un procesamiento de la imagen en la cual se realizan algunas mejoras para poder realizar luego el reconocimiento con mayor facilidad. El procesamiento de la imagen permite mejorar algunas características de la imagen original que dificultan el reconocimiento y además permite normalizar los números tanto en tamaño como en grosor.

CAPÍTULO 1: LECTURA AUTOMÁTICA DE NÚMEROS

1.1 Usos de los sistemas de reconocimiento de números manuscritos.

Los sistemas que reconocen números manuscritos pueden tener diferentes aplicaciones, en las cuales se desea automatizar y ahorrar tiempo en el paso de estos a un formato digital, el cual luego pueda ser editado o almacenado en una base de datos para un posterior uso de éstos.

Algunas aplicaciones de estos sistemas son automatizar la lectura de direcciones postales este proceso se puede dar de forma manual pero haciendo uso de estos sistemas se puede reducir el tiempo. Otras aplicaciones que se le pueden dar a estos sistemas son la digitalización de cantidades monetarias en cheques bancarios, actas de notas académicas, conteo de votos electorales.

1.2 Características de los sistemas de reconocimiento de números manuscritos.

Existen varias formas de reconocimiento de las cuales se puede distinguir reconocimiento “On Line” y “Off Line”, es decir si se tiene o no en cuenta la información temporal asociada a los trazos que componen el carácter numérico. El carácter puede ser reconocido mientras está siendo escrito o luego de haberlo sido, para el primer caso se tiene una representación del espacio temporal de los datos de entrada y en el segundo una representación del espacio iluminancia de estos datos [3]. Por lo general en este tipo de sistemas se usa la segunda opción y el carácter es obtenido mediante medios ópticos (cámaras, escáner). Lo que se busca con estos sistemas es obtener una representación simbólica lo más fiel y completa posible de una imagen que haya sido digitalizada el cual contenga los caracteres manuscritos.

La complejidad de estos sistemas puede variar en función de la calidad de los números manuscritos. Depende de la calidad de cómo se escribió el número o a la calidad en la que se encuentra el material del documento que se desea procesar. El resultado que se puede generar en estos sistemas depende bastante de lo mencionado anteriormente, dado que al tener un documento en el cual la caligrafía no sea muy legible o poco entendible por una persona entonces el sistema tiende a cometer algunos errores de reconocimiento. Los sistemas más complejos usan distintas técnicas de reconocimiento para poder mejorar el nivel de reconocimiento dado que por lo general la información de entrada de estos no es tan legible.

1.3 Declaración del marco problemático

En diversas situaciones tales como lectura de códigos postales, cheques, actas escolares, se necesita digitalizar números que hayan sido manuscritos y es necesario que esta información tenga una gran confiabilidad dado que al ser reconocidos de forma incorrecta pueden causar severos problemas como es el caso del reconocimiento de las cantidades monetarias en los cheques bancarios o en acta de notas.

El reconocimiento de números manuscritos se hace dificultoso debido a que cada persona escribe de distinta forma con lo cual se tiene que recurrir a algoritmos especializados para poder reconocerlo. Existen diversos algoritmos que son utilizados en el reconocimiento pero la eficacia de estos se ve limitada dado las características que presentan los números manuscritos tales como el tamaño, inclinación, grosor y las distintas formas en que una persona puede representar un mismo número. Se necesita de técnicas que sean lo suficientemente buenas para poder reconocer números manuscritos y cuyo nivel de reconocimiento sea confiable.

Niveles elevados de reconocimiento van asociados a mayores niveles de error con lo cual los sistemas se ven limitados entre buscar mayor reconocimiento o menor

error, para esto se establece un límite umbral en el que un número pasa a ser reconocido o no, al tener un nivel elevado de umbral la confiabilidad del sistema será mayor pero el porcentaje de reconocimiento se ve reducido es por eso que la búsqueda de un nivel óptimo de umbral se debe realizar para cada técnica con la que se realiza el reconocimiento.



CAPÍTULO 2: TÉCNICAS EMPLEADAS EN EL RECONOCIMIENTO DE NÚMEROS MANUSCRITOS.

2.1 Estado del arte.

2.1.1 Presentación del asunto de estudio

Los sistemas de reconocimiento de números manuscritos forman parte de un sistema más amplio conocido como sistema de reconocimiento óptico de caracteres (OCR) el cual transforma a archivos tipo ASCII los caracteres de una imagen digitalizada. El principal problema de estos sistemas es el porcentaje de efectividad que se obtiene en el reconocimiento de caracteres manuscritos, esto es debido a la naturaleza de estos caracteres, la tipografía, la resolución de la imagen, iluminación y otros factores.

Los sistemas que reconocen números manuscritos tienen diversas aplicaciones como automatizar la lectura de direcciones postales, cheques bancarios, formularios de impuestos, formularios de censo y lectores de texto para discapacitados visuales, entre otros. En las aplicaciones mencionadas se necesita que la tasa de acierto al reconocer los números sea muy elevada.

Estos sistemas por lo general comprenden las siguientes etapas: pre-procesamiento, segmentación, normalización y reconocimiento. Estas etapas comprenden diversos métodos los cuales, integrándose de forma eficaz, pueden hacer que el sistema logre buenos resultados, obteniéndose una tasa de reconocimiento alta con un bajo porcentaje de error.

En el **pre-procesamiento** se extraen los caracteres que se encuentran dentro de una imagen, en el caso de actas de notas los números pueden estar encerrados por cuadrados, también dentro de planillas diseñadas como es el caso de cheques

bancarios en el cual ya se tiene destinada la ubicación donde se escribe las cantidades monetarias.

Luego se debe mejorar algunos defectos en cuanto a la calidad de la tinta que se uso para escribir los números que se desean reconocer, además el escáner puede insertar niveles de grises dentro de los píxeles que deberían ser negros y pequeñas regiones negras dentro de los píxeles que deberían ser blancos.

La **segmentación** es la operación que permite descomponer una cadena de caracteres en caracteres individuales. Constituye una de las mayores dificultades dado que si los caracteres se encuentran conectados se obtienen menores tasas de reconocimiento a diferencia de encontrarse aislados [4].

La **normalización** consiste en obtener muestras similares en cuanto a características de tamaño y grosor. También se realiza una corrección de la inclinación. Este proceso ayuda a mejorar la eficacia del sistema para reconocer números manuscritos.

La **etapa de reconocimiento** se encarga de procesar los caracteres normalizados y luego identificarlos.

Se presentan las diferentes partes que componen un sistema de reconocimiento de números manuscritos [5]. Con todas las etapas mencionadas se busca incrementar el porcentaje de reconocimiento así como la disminución del error.

2.1.2 Estado de la investigación.

2.1.2.1. Mejoramiento:

Los números pueden estar escritos en distintas partes dentro de un documento el cual necesita ser procesado. Algunos de estos documentos pueden tener algún

formato o posición especial en el cual se escriben los números, como por ejemplo: actas de notas escolares en el cual las notas van en unos recuadros; en los cheques bancarios se tiene en la parte superior derecha un recuadro donde se coloca la cantidad monetaria[1][2]. Como la mayoría de estos documentos tiene un lugar establecido donde se escriben los números lo que se realiza para extraer los caracteres numéricos del documento es definir el tipo de formato del documento que se desea procesar, para luego con esta información, conocer las zonas de las imágenes donde se encuentran los números.

Luego de extraer la región donde se encuentran los números se debe pasar la imagen que se encuentra en escala de grises a una imagen binaria (0,1), la idea general de este método es convertir los píxeles por encima de un cierto nivel de gris en primer plano y los píxeles por debajo del nivel en un segundo plano. El método más simple consiste en usar un nivel de umbral. Esta técnica no considera la diferencia entre caracteres en una imagen determinada o entre distintas imágenes, además el contraste y el brillo en una imagen puede variar. Dependiendo de la calidad de la imagen, el fondo puede ser muy oscuro o claro. Otro método es usar los píxeles de algún lugar cercano a los bordes de la imagen el cual se supone representa a un píxel del fondo y con este crear un valor umbral, pero este método ignora el plano principal y puede dar lugar a un resultado inexacto. Un método práctico es utilizar un histograma de los valores de los píxel en la imagen, debe haber un gran pico que indica el valor general de los píxeles del fondo y otros picos pequeños que indica el valor de los píxeles que corresponden a los números.

2.1.2.2. Segmentación:

La segmentación es una operación que busca descomponer una imagen en una secuencia de imágenes individuales en la que cada imagen individual contenga el símbolo de un carácter manuscrito. Los métodos de segmentación usados en el reconocimiento de números manuscritos se pueden clasificar en 3 grupos [1]. El

primer método analiza la imagen sin utilizar la clase específica de la forma de información, consiste en dividir la imagen en sub-imágenes las cuales luego serán clasificadas. El segundo método consiste en usar una ventana movable de ancho variable la cual proporciona una secuencia de posibles caracteres segmentados lo cual luego es corroborado en la etapa de reconocimiento. El último método se basa en aplicar distintos algoritmos de segmentación buscando aquel que de buenos resultados luego de pasar estos caracteres segmentados por la etapa de reconocimiento.

- **Segmentación explícita:**

Este es uno de los métodos más comunes el cual busca los espacios blancos entre cada carácter [6], pero este método queda muy limitado a situaciones específicas dado que los caracteres pueden no presentar espacios en blancos o tal vez estos no sean muy notorios en cuyos casos no podría realizarse la segmentación.

Otro método para lograr la disección se hace realizando proyecciones verticales o también llamado “histograma vertical”, la imagen en donde se contienen los caracteres se puede considerar como una matriz en la cual los píxeles del carácter son de color negro y los del fondo son blancos. El método consiste en sumar los píxeles negros de cada columna y en donde se obtenga una baja concentración de píxeles negros entonces se asume que ahí hay una separación entre carácter. Después de realizarse las pruebas respectivas se comprobó que dicho método es bastante bueno pero presenta un alto porcentaje de error cuando los caracteres se encuentran conectados o traslapados. Una variante a este mismo tipo de método es realizar restas entre columnas adyacentes, lo que genera picos de proyecciones, identificando de esta forma las operaciones pero este caso también falla cuando se encuentran caracteres conectados.

- **Segmentación implícita:**

Esta estrategia de segmentación está basada en reconocimiento [1]. Este método busca la imagen para las componentes que coinciden con las clases predefinidas. El principio básico de este método es utilizar una ventana móvil de ancho variable para proporcionar secuencias de posibles segmentaciones las cuales son luego comparadas con los caracteres numéricos que están almacenados en la memoria, se guardan todas las coincidencias producidas y el residuo de cada ancho de ventana, que fueron pasados a través de la ventana, son comparados con los caracteres almacenados en la memoria. Los nuevos porcentajes de coincidencia son almacenados y de la combinación de los dos porcentajes más altos se obtienen como resultado la segmentación correcta.

El método de atención selectiva toma Redes Neuronales aún más en el manejo del problema de la segmentación, en este método al reconocer un número manuscrito no se toma interés a algunos píxeles que pueden ser usados por dos caracteres numéricos. Este método efectúa una comparación con los objetos almacenados en memoria y de no haber una diferencia considerable se arroja un resultado de coincidencia.

- **Segmentación mixta:**

Esta técnica es un caso general de segmentación la cual hace uso de las técnicas descritas. Una de los métodos usados para segmentar números manuscritos que se encuentran conectados es el “Hirbrid Drop Fall” el cual consiste en aplicar diversos algoritmos de separación para obtener el camino de separación correcto [2]. Estos algoritmos se basan en la analogía de una gota de ácido que va cayendo por acción de la gravedad y escurre por el borde de los caracteres, pero cuando no se puede desplazar empieza a fundir la tinta y crea un corte de separación. Este algoritmo se puede aplicar en distintas direcciones y con pequeñas variantes, lo que da lugar a diversas maneras de separar los caracteres conectados. Se debe analizar sistemáticamente cada posibilidad de separación hasta encontrar la solución óptima. En principio, si un camino de separación no es

válido entonces los segmentos resultantes no son dígitos por lo que no podrán ser reconocidos entonces se procede con otro algoritmo y así en forma sucesiva.

2.1.2.4 Normalización:

Cuando se tienen números manuscritos estos tienen una diversidad de formas y características, tales como números de distinto tamaño, ancho y inclinación; además estos números pueden tener distintos grosores dado que algunos pueden ser escritos con un plumón y otros con un lápiz o lapicero en el cual el grosor del número sería distinto.

Para poder tener una alta eficiencia en el reconocimiento de estos se necesita que estos números estén normalizados en tamaño, grosor y inclinación.

La corrección de la inclinación se realiza calculando el ángulo de inclinación aplicando momentos de inercia y luego se realiza una traslación de los píxeles en su misma fila, esta traslación se realiza de acuerdo al ángulo de inclinación [7] [8]. Otro método consiste en ir rotando al carácter numérico con un cierto ángulo hasta que se encuentre el mínimo ancho.

La normalización del grosor consiste en 2 partes: un algoritmo para reducir el grosor y otro para aumentarlo, esto se realiza con la finalidad de tener los números con un grosor estándar.

El proceso de reducción del grosor consiste en eliminar los píxeles del borde del número sin perder su forma original. Las condiciones exactas que determinan si un punto se puede borrar están relacionadas con el concepto de *punto simple* (un píxel negro P del borde de la imagen se considera simple si el conjunto de los vecinos en negro de P tienen exactamente una componente conexa que es adyacente a P) y punto final (un punto es final si tiene exactamente un vecino negro; un punto final no es más que un punto extremo de la imagen). Es decir, un

punto del borde de cada componente se puede eliminar si es simple y no es final. El borrado de puntos debe seguir un esquema de barridos sucesivos para que la imagen siga teniendo las mismas proporciones que la original y conseguir así que no quede deformada. El borrado de cada rastreo debe hacerse en paralelo, es decir, señalar los píxeles que se borrarán para eliminar todos a la vez.

El aumento del grosor es de forma similar a la del adelgazamiento solo que esta vez se tiene que ir rellenando de píxeles por el contorno de la imagen hasta llegar al grosor deseado.

2.1.2.5 Reconocimiento:

En los sistemas de reconocimiento de números manuscritos se necesita una alta exactitud en los resultados obtenidos dado que la aplicación de estos se ve limitada por el grado de exactitud que se pueda obtener, se desea valores de reconocimiento mayores al 90% y error cercano al 0%. Algunas investigaciones realizadas en este campo [9][10] obtuvieron resultados cuyas tasas de error están alrededor del 5% y la tasa de acierto está alrededor del 70 al 90%. En algunas aplicaciones se requiere que el error sea cercano al 0% como es el caso de los cheques bancarios. Existen diversos métodos para el reconocimiento de números manuscritos:

- **Métodos geométricos o estadísticos:**

Este método es conocido con el nombre de los “k vecinos más próximos” (knn, k nearest neighbours) [11]. Este método consiste en tener una base de datos en el cual se tiene un conjunto de muestras los cuales sirven como prototipo y además se conoce su clase (conjunto de números de muestra) y cuando se tiene un nuevo objeto cuya clase no conocemos (número a reconocer) se busca dentro del conjunto de prototipos de “k” más parecidos al nuevo objeto. Luego a este se le asigna la clase más numerosa entre los “k” objetos prototipos seleccionados.

- **Métodos estructurales:**

Estos métodos se basan en la idea de que una estructura se puede dividir en partes, y en consecuencia, estas partes pueden ser descritas por sus características y la relación entre las diferentes partes. Estos métodos pasan por un adelgazamiento de los números extraídos de la imagen, de manera que pasan por una detección de contornos interiores y externos para detectar una serie de puntos singulares, de estos puntos se extrae una información topológica. Posteriormente, se pasan a un vector, que permite representar una descripción en forma de cadena de símbolos o de gráficos. El problema es cómo elegir las características y las relaciones de tal manera que se obtenga la descripción correcta y se pueda identificar de forma exclusiva a cada uno de los posibles caracteres en el reconocimiento conjunto.

- **Métodos basados en redes neuronales:**

Las redes neuronales artificiales ofrecen alto grado de exactitud en el reconocimiento de números manuscritos [9]. Mediante esta herramienta se trata de simular la capacidad que tiene el cerebro humano en cuanto al aprendizaje y de poder procesar información incompleta o que no es precisa. Las redes neuronales además de ofrecer altos grados de exactitud generalmente son muy rápidas. Para el reconocimiento de números manuscritos se hace uso de diversos tipos de redes de múltiples capas tales como: redes multicapa (Multilayer Perceptron), redes de base radial (RBF: Radial Basis Function Network), redes ART (adaptive Resonance Theory models) y Neocognitron [12][13].

Las redes neuronales deben encontrar una representación interna las cuales permiten generar las salidas deseadas durante el entrenamiento, y posteriormente durante el funcionamiento ser capaces de generar salidas para entradas que no le fueron mostradas durante el aprendizaje pero que se asemejan a algunas de las que si le fueron mostradas. Para esta técnica es necesario disponer de grandes bases de datos de aprendizaje.

- **Métodos Markovianos:**

Es un proceso estocástico generado por dos mecanismos interrelacionados: una cadena de Markov con un número finito de estados y un conjunto de funciones aleatorias, cada una de ellas está asociada con un estado [1][14]. En instantes discretos de tiempo, el proceso asume algún estado y una observación es generada por la función aleatoria correspondiente al estado actual. La cadena de Markov subyacente a continuación cambia los estados de acuerdo a sus probabilidades de transición. El trabajo es construir un modelo que explica y caracteriza la aparición de símbolos observados. La generación correspondiente a un símbolo puede ser caracterizado discreto o continuo. Salidas discretas pueden ser caracteres de un alfabeto finito o vectores cuantizados de un libro de códigos, mientras que las salidas continuas están representadas por muestras de una forma de onda continua. En la generación de una palabra o un carácter, el sistema pasa de un estado a otro, cada estado emite una salida acorde algunas probabilidades hasta que la palabra entera o carácter sea generado.

- **Métodos Mixtos:**

Este método consiste en hacer uso de algunos métodos de los ya descritos anteriormente con la finalidad de poder tomar una mejor decisión al momento de reconocer un carácter numérico. Se puede combinar estas técnicas en paralelo de tal forma que se pueda comparar la respuesta de estos y luego arrojar una única respuesta la cual tendrá un mayor nivel de acierto.

En la tabla 2.1 puede observarse los resultados obtenidos por algunos autores en el reconocimiento de números manuscritos, a través del uso de diferentes métodos basados en redes neuronales. Los marcados con “(*)” usaron la base de datos de la Universidad de Concordia y el marcado con “(**)” hizo uso de la base de datos del National Institute of Standards and Technology (NIST). Algunos de los datos mostrados en la tabla 1 fueron extraídos de [9].

Tabla 2.1. Tabla de resultados de algunos autores en el reconocimiento de números manuscritos, utilizando distintos reconocedores basados en redes neuronales [9][10]

Autores	% Reconocidos	% Error	% Rechazados	Patrones entrenamiento	Patrones prueba
(*) Nadal (1988)	86.05	2.25	11.70	4000	2000
(*) Krzyzak (1990)	86.40	1.00	12.60	4000	2000
(*) Krzyzak (1990)	94.85	5.15	0.00	4000	2000
Le Cun (1990)	92.00	2.00	6.00	7291	2007
Le Cun (1990)	96.40	3.40	0.20	7291	2007
(*) Mai (1990)	92.95	2.15	4.90	4000	2000
(*) Suen (1990)	93.05	0.00	6.95	4000	2000
Knerr (1992)	90.30	1.00	8.70	7200	1800
Lemarie (1993)	97.97	2.03	0.00	8783	7394
(*) Kim & Lee (1994)	95.40	4.60	0.00	4000	2000
(*) Lee (1995)	97.80	2.20	0.00	4000	2000
(*) Cho (1997)	96.05	3.95	0.00	4000	2000
(*) Leticia (2003)	91.90	8.10	0.00	4000	2000
(**) R. palacios (2003)	84.00	5.50	10.50	60089	28000

2.1.3 Síntesis sobre el asunto de estudio

El reconocimiento de números manuscritos es el proceso en el cual estos caracteres numéricos son transformados a tipo ASCII para su posterior utilización en el cual se necesita tener esta información en forma digitalizada.

El reconocer caracteres manuscritos es dificultoso debido a que estos no tienen un tipo de fuente definido como es el caso de los caracteres impresos, además existen distintas formas de representar un número y un variable estilo de escritura debido a que las personas no tienen un estándar de forma de escritura y esto lleva a que los sistemas tengan mayor dificultad al reconocer estos caracteres. Existen sistemas que son dependientes de escritor y otros que son independientes, este último tipo de sistema consigue menor tasa de acierto que el primero pero la forma

de escritura también puede variar dado que dependen de algunos factores que pueden hacer que el escritor escriba de forma distinta.

Para poder tener una mayor tasa al reconocer un carácter numérico se necesita que este sea acondicionado, tales etapas son pre-procesamiento, segmentación, normalización. En estas etapas se obtienen muestras de características similares en cuanto a tamaño y grosor con lo cual se facilita la etapa de reconocimiento. Estas etapas por lo general se usan con un lazo de realimentación dado que las primeras etapas pueden dar mejores resultados teniendo la información de las etapas posteriores. Con este tipo de realimentación mencionado se tiene un mejor acondicionamiento y la eficacia del reconocedor aumenta y el error se ve disminuido.

Existen diversas técnicas para poder reconocer los números manuscritos una de las técnicas que mejor resultados a dado son las redes neuronales [2]. Estas redes neuronales para un buen funcionamiento necesitan ser previamente entrenadas y contar con una buena etapa de acondicionamiento anterior a la de reconocimiento. La base de datos para entrenar la red neuronal debe ser un conjunto adecuado de muestras que contengan muestras de una diversidad de formas de escrituras de números.

2.2 Módulo de Reconocimiento:

El primer paso del módulo de reconocimiento es un proceso especial de pre-procesamiento en el cual se realiza la extracción de los caracteres de forma aislados y se realiza algunas mejoras a estas para poder luego ser reconocidas con mayor facilidad.

El conjunto de caracteres es inicialmente segmentado para poder obtener cada carácter de forma aislada luego se realiza una normalización de estos y se corrige los efectos que pueda haber tenido en cuanto a inclinación, grosor y tamaño [15].

2.2.1 Pre-procesamiento:

Los datos pueden presentar patrones de diferentes características las cuales pueden variar en posición, tamaño y grosor. Para poder ser reconocido cada carácter debe ser reconocido de forma aislada para lo cual se debe realizar un primer paso que es el de extraerlos por separado, luego se procede a corregir ciertos efectos de la escritura manuscrita tales como inclinación, grosor y tamaño. Para corregir estos efectos se realiza una normalización, los caracteres al finalizar este proceso tienen características parecidas en cuanto al tamaño, grosor e inclinación.

2.2.2 Redes Neuronales Artificiales:

Mediante esta técnica se pretende imitar el proceso del cerebro humano. Basados en la experiencia de los procesos llevados a cabo en el cerebro, e inspirados en su funcionamiento es que se desarrolla las Redes Neuronales Artificiales (RNA) [12][13]. Las RNA se utilizan para aprender estrategias de solución basadas en ejemplos de comportamiento típico de patrones; estos sistemas no requieren que la tarea a ejecutar se programe, ellos generalizan y aprenden de la experiencia.

Una RNA es un sistema compuesto de muchos elementos procesadores simples conectados en paralelo, cuya función es determinada por la estructura de la red, la fuerza en las conexiones y el procesamiento realizado por los elementos en los nodos.

2.2.2.1 Características generales:

Para la comprensión y empleo de las redes neuronales se tiene que tener en consideración algunas características propias de estas:

- **Pesos:**

Las redes neuronales pueden tener factores de peso adaptable o fijo. Los pesos fijos son definidos al momento de la construcción de la red y los pesos adaptables son los que se van ir ajustando de acuerdo a las reglas de aprendizaje de la RNA.

- **Aprendizaje:**

El aprendizaje puede ser de tres tipos: Supervisado, no supervisado e híbrido [12].

Es supervisado cuando la información es presentada a la red en forma de patrones de entrada y los resultados de entrada son conocidos previamente. Se le proporciona a la red tanto la salida como la entrada correcta, luego se verifica la salida de la red comparando esta con la salida esperada. En base al error obtenido en la comparación se ajustan los pesos para minimizar esta diferencia.

Es no supervisado cuando la información es presentada a la red bajo forma de patrones de entrada y los resultados no son conocidos previamente, pues no se requiere de la salida correcta a cada patrón de entrada. Se proporciona solo los estímulos y la red ajusta los pesos de sus interconexiones utilizando únicamente los estímulos y salida calculada por la red.

El paradigma híbrido combina los dos tipos de aprendizaje antes mencionados, para lo cual estos aprendizajes se aplican en distintas capas de la red en la cual algunos pesos se ajustan con el aprendizaje supervisado y otros con el no supervisado.

- **Fase de operación:**

Se presentan dos fases de operación: Entrenamiento y recuperación de lo aprendido. En la primera fase se realiza el ingreso de estímulos de entrada y salida a la RNA para que esta pueda ajustar sus pesos de interconexión y

minimice el error en la salida que calcula. En la segunda fase la red solamente calcula la respectiva salida. El aprendizaje de la red disminuye a medida que aumenta el número de veces que se usa.

- **No son algorítmicas:**

Las RNA no son construidas y programadas para que se dediquen solo a seguir un conjunto de instrucciones sino que estas pueden generar sus propias reglas para poder aprender de su propia experiencia. Las RNA aprenden a generalizar a partir de su entrenamiento y por ello no requieren de diversos algoritmos que se encuentren previamente programados.

- **Necesitan un patrón:**

Las RNA para poder reconocer necesitan que el conjunto de muestras de entrada tengan algún tipo de patrón de lo contrario no podrán reconocer nada. Dado esta característica es que las RNA son muy buenas resolviendo problemas de asociación, evaluación y reconocimiento de patrones.

2.2.2.2 Red Perceptrón:

Es la base de la arquitectura de las redes neuronales artificiales. El perceptrón trabaja con funciones de activación. En la Figura 2.1 se observa el modelo de un perceptrón, en esta se puede observar como se hace la semejanza con las partes de las que se compone una neurona biológica (axón, sinapsis, dendritas, cuerpo).

El cuerpo de la neurona se representa como sumador lineal seguida de una función de transferencia (no lineal) que es la función de activación, esta utiliza la suma de estímulos para determinar la salida de la neurona. En el perceptrón las entradas son multiplicadas por los pesos W correspondiente, los resultados son sumados y luego evaluados contra el valor de umbral θ , si el valor es mayor al máximo el perceptrón se activa.

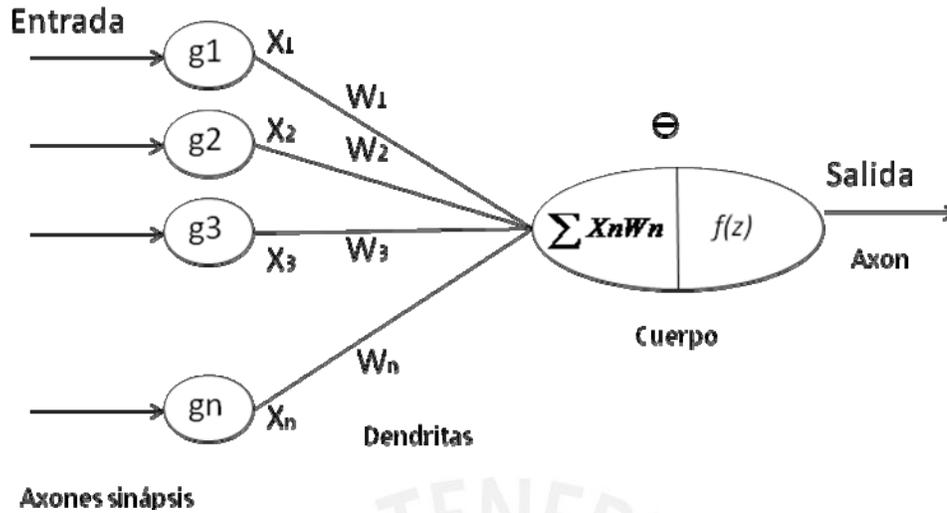


Figura 2.1. Modelo de un Perceptrón [12]

La única neurona de salida del Perceptrón realiza la suma ponderada de las entradas, resta el umbral y pasa el resultado a una función de transferencia de tipo escalón. La regla de decisión es responder +1 si el patrón presentado pertenece a una clase, o -1 si el patrón pertenece a otra clase, la salida depende de la entrada neta.

La red tipo Perceptrón emplea principalmente dos funciones de transferencia, *hardlim* con salidas 1, 0 o *hardlims* con salidas 1, -1. El uso depende del valor de salida que se espera para la red, es decir si la salida de la red es unipolar o bipolar; sin embargo la función *hardlims* es preferida sobre la *hardlim*, ya que el tener un cero multiplicando algunas de los valores resultantes del producto de las entradas por el vector de pesos, ocasiona que estos no se actualicen y que el aprendizaje sea más lento.

Una técnica utilizada para analizar el comportamiento de redes como el Perceptrón es presentar en un mapa las regiones de decisión creadas en el espacio multidimensional de entradas de la red, en estas regiones se visualiza qué patrones pertenecen a una clase y cuáles a otra. El Perceptrón separa las regiones por un hiperplano cuya ecuación queda determinada por los pesos de las conexiones y el valor umbral de la función de activación de la neurona, en este

caso los valores de los pesos pueden fijarse o adaptarse empleando diferentes algoritmos de entrenamiento.

2.2.2.3 Red Perceptrón Multicapa (MLP):

Un Perceptrón multicapa es una red con alimentación hacia delante, compuesta de varias capas de neuronas entre la entrada y la salida de la misma, esta red permite establecer regiones de decisión mucho más complejas que las de dos semiplanos, como lo hace el Perceptrón de un solo nivel.

2.2.2.4 Red Backpropagation:

En esta red se interconectan varias unidades de procesamiento en capas, las neuronas de cada capa se conectan entre sí. Cada neurona de una capa proporciona una entrada a cada una de las neuronas de la siguiente capa.

En esta red se realizan dos procesos, uno hacia adelante y otro hacia atrás [12]. En el primero se ingresa una señal de entrenamiento para lo cual la red arroja un resultado que luego es comparado con la salida deseada produciéndose luego una señal de error por cada neurona de la capa de salida. En el segundo, la señal de error es propagada hacia atrás de tal forma que se vaya realizando el ajuste de los pesos de las neuronas de las capas anteriores en función de la razón de aprendizaje, el gradiente local y la entrada de la neurona específica. Este proceso se da repetidamente para cada patrón de entrada hasta que se obtenga un mínimo de error.

La importancia de este proceso consiste en que, a medida que se entrena la red, las neuronas de las capas intermedias se organizan a sí mismas de tal modo que las distintas neuronas aprenden a reconocer distintas características del espacio total de entrada. Después del entrenamiento, cuando se les presente un patrón arbitrario de entrada que contenga ruido o que esté incompleto, las neuronas de la

capa oculta de la red responderán con una salida activa si la nueva entrada contiene un patrón que se asemeje a aquella característica que las neuronas individuales hayan aprendido a reconocer durante su entrenamiento. Y a la inversa, las unidades de las capas ocultas tienen una tendencia a inhibir su salida si el patrón de entrada no contiene la característica para reconocer, para la cual fueron entrenadas.

2.2.2.5 Modelo de Kohonen:

Esta red se basa en la capacidad que tiene el cerebro humano de formar mapas característicos de la información recibida del exterior.

Este tipo de red contiene solamente una capa de neuronas y una capa de entrada, que se ramifica para todos los nodos. Pertenece a la categoría de redes competitivas o mapas de auto organización, es decir tiene un aprendizaje no supervisado [16].

El aprendizaje en el modelo de Kohonen es de tipo Off-line, por lo que se distingue una etapa de aprendizaje y otra de funcionamiento. En la etapa de aprendizaje se fijan los valores de las conexiones entre la capa de entrada y la salida. Esta red utiliza un aprendizaje no supervisado de tipo competitivo, las neuronas de la capa de salida compiten por activarse y sólo una de ellas permanece activa ante una determinada información de entrada a la red, los pesos de las conexiones se ajustan en función de la neurona que haya resultado vencedora.

Durante la etapa de entrenamiento, se presenta a la red un conjunto de informaciones de entrada (vectores de entrenamiento) para que ésta establezca en función de la semejanza entre los datos las diferentes categorías (una por neurona de salida), que servirían durante la fase de funcionamiento para realizar clasificaciones de nuevos datos que se presenten a la red. Los valores finales de los pesos de las conexiones entre cada neurona de la capa de salida con las de entrada se corresponderán con los valores de los componentes del vector de

aprendizaje que consigue activar la neurona correspondiente. En el caso de existir más patrones de entrenamiento que neuronas de salida, más de uno deberá asociarse con la misma neurona, es decir pertenecerán a la misma clase.

En este modelo el aprendizaje no concluye después de presentarle una vez todos los patrones de entrada, sino que habrá que repetir el proceso varias veces para refinar el mapa topológico de salida, de tal forma que cuantas más veces se presenten los datos, tanto más se reducirán las zonas de neuronas que se deben activar ante entradas parecidas, consiguiendo que la red pueda realizar una clasificación más selectiva.

2.3 Modelo Teórico

El reconocimiento de dígitos manuscritos es una tarea importante en el análisis automático de documentos. A pesar del notable progreso logrado en los últimos años en el área de reconocimiento automático de números manuscritos, todavía se está lejos de poder igualar la capacidad humana de reconocimiento.

En estos sistemas se busca tener una tasa alta de reconocimiento y un bajo porcentaje de error lo que se busca en este trabajo es probar una de las técnicas que están dando mejores resultados y realizar la etapa de acondicionamiento que colaboren con esta para poder tener un mejor resultado y luego analizar los resultados para decidir si es conveniente usar una técnica específica o es conveniente hacer uso de otras técnicas. También se podrá usar más de una técnica en paralelo dándole una mayor robustez al sistema.

Los números manuscritos pueden estar escritos de diversos tamaños y de diversas características en cuanto a inclinación y grosor. Estas características hacen que el reconocimiento sea dificultoso por este motivo se realiza una etapa de acondicionamiento (pre-procesamiento, segmentación, normalización). En estas etapas se busca obtener muestras de características parecidas en cuanto a inclinación grosor y tamaño.

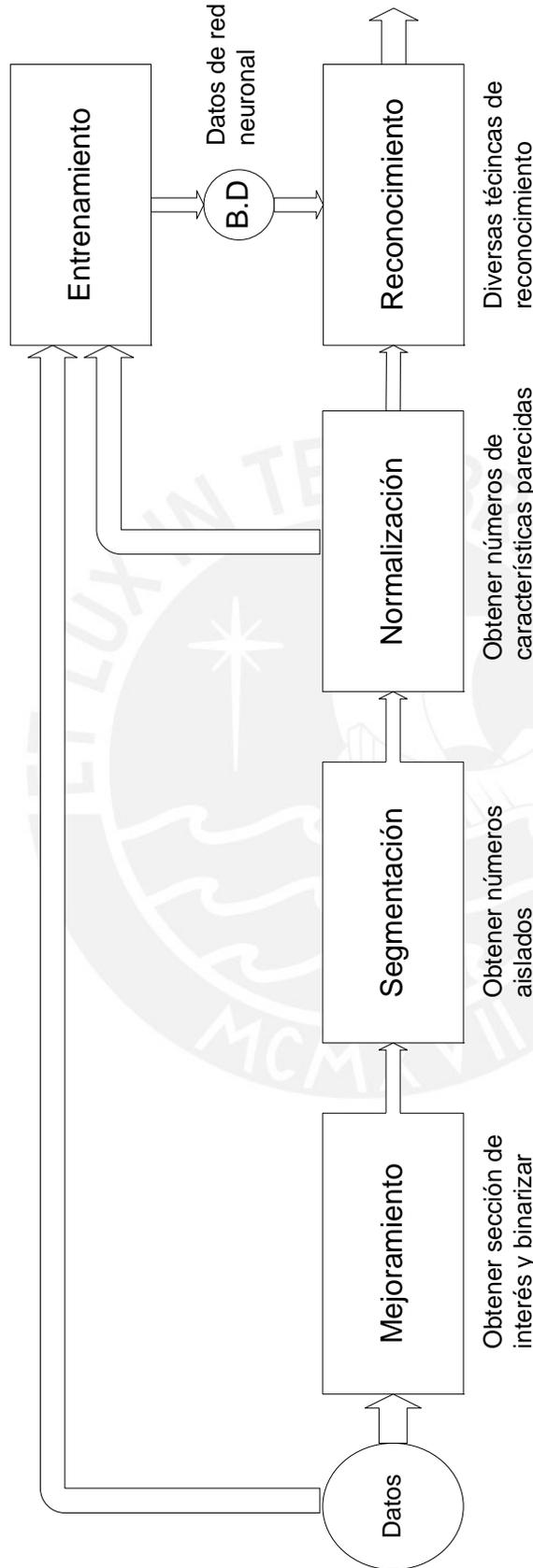


Figura 2.2 Representación gráfica del modelo teórico

El propósito del sistema es poder realizar el reconocimiento de los números manuscritos y en base a los resultados obtenidos encontrar la técnica que mejor resultados se obtenga en el reconocimiento. Para ello el sistema comprende de diversos etapas en la cual inicialmente se hace uso de un escáner para obtener en formato digital los números manuscritos; se realiza un mejoramiento de la imagen original; se extraen los números de forma aislada (segmentación); luego se corrigen ciertos aspectos de la escritura manuscrita tanto en el grosor y la inclinación (normalización); por último se procede a reconocer el carácter numérico haciendo uso de distintas técnicas.

2.4 Definiciones operativas

Para evaluar al sistema se tiene que tener en cuenta los aspectos de digitalización y de procesamiento.

Digitalización:

- **Resolución Espacial:** La resolución es la relación entre las dimensiones digitales (píxeles) y las físicas (tamaño real). La calidad de una imagen, también depende de la resolución que tenga el dispositivo que la capta. El número de píxeles que contenga una imagen dependen de cuántos píxeles utilice el sensor CCD de la cámara o el escáner para captar la imagen. El obtener una imagen con más detalle o mejor calidad visual se entiende como tener mejor resolución.

El número de celdillas o unidades mínimas de impresión que una máquina es capaz de realizar como máximo en un espacio dado es lo que se llama "resolución". A mayor número de resolución, mayor nitidez de dibujo. La resolución se suele medir en puntos por pulgadas (abreviado ppp) y que estas son siempre lineales, es decir, que están formadas por una fila de puntos de impresión colocados en línea, uno detrás de otro. Así, hablamos de una

impresora con 300 puntos por pulgada (una pulgada equivale a unos 2,54 centímetros).

- **Distorsión de imagen:** Alteración de la imagen en la cual esta resulta ser ilegible por algunas alteraciones que esta puede haber sufrido debido a posibles cambios en la posición de la imagen o cambio de las escalas del tamaño original.
- **Ruido de fondo:** Puede considerarse como imágenes que no tienen significado. Esta la decoloración o manchas que pueden surgir por el paso del tiempo, también puede ser debido al contacto de la superficie donde se escribió con algún elemento que desprenden color como las tintas de lapicero o también se puede dar por uso de lápiz.
- **Ruido en la imagen:** El dispositivo que obtiene la imagen puede insertar niveles de grises dentro de los píxeles que deberían ser negros y pequeñas regiones negras dentro de los píxeles que deberían ser blanco.

Procesamiento:

- **Tiempo de reconocimiento:** Es el tiempo promedio que demora el sistema en dar resultados, este tiempo está directamente relacionado a la cantidad de operaciones que se tienen que realizar.
- **Exactitud:** Se refiere a que tan cerca del valor real se encuentra el valor medido. En términos estadístico, la exactitud está relacionada con el sesgo de una estimación. Cuanto menor es el sesgo más exacto es una estimación. En el reconocimiento de caracteres la exactitud está ligada a cuan parecido o cuan alto es el porcentaje de coincidencia.

- **Error:** Es la inexactitud o equivocación en dar una determinada respuesta. El error es visto como el porcentaje de resultados incorrectos que se dan dentro de un conjunto de muestras a analizar. Sistemas de bajo porcentaje de error van relacionados con menores porcentajes de reconocimiento dado que ponen mayores restricciones en la emisión de un resultado. Este tipo de sistemas llegan a tener un porcentaje de error menor al 5% [2][3].

Los factores importantes a analizar son la exactitud y el error dado que se busca obtener un método que logre buenos resultados en cuanto a estas variables. Se tienen que analizar y probar métodos usados en estos tipos de sistema y encontrar el un mejor resultados arroje o en todo caso hacer un sistema robusto que se base en clasificadores en paralelo (distintas técnicas).



CAPÍTULO 3:

DISEÑO E IMPLEMENTACIÓN DEL SISTEMA BASADO EN REDES

NEURONALES

En este capítulo se detalla los pasos realizados para poder reconocer un número. Estos pasos comprenden el obtener la imagen mediante un escáner convencional, luego se debe realizar una mejora de estas, luego extraer estos números de forma aislada (segmentación), finalmente se debe realizar una normalización de las características físicas (tamaño, grosor, inclinación) para luego poder ser reconocidos.

3.1 Captura de la imagen

Para la captura de la imagen se usó un escáner, se trabajó con una resolución de 200dpi (puntos por pulgada). Para este trabajo se utilizó el formato BMP.

Para recolectar las muestras de los números se realizó planillas con unas cuadrículas similares a las de una acta de notas en las cuales se escribieron los números. Los números que fueron escritos son de diversos tamaños y algunos se encuentran inclinados para cuyos casos se tuvo que realizar algunas mejoras previas de la imagen antes de ser reconocidas.

1	2	3	4	5	6	7	8	9	0
1	2	3	4	5	6	7	8	9	0
1	2	3	4	5	6	7	8	9	0
1	2	3	4	5	6	7	8	9	0

Figura 3.1. Muestra de formato usado para recolectar números

Los números con los que se trabajó son de buena escritura y de un solo tipo de escritura tal como los que se muestra en la Figura 3.1. Estos números no presentan mayores problemas tales como borrones o números incompletos.

3.2 Procesamiento de la Imagen

3.2.1 Extracción de números

Las imágenes se encuentran dentro de unas cuadrillas las cuales tienen que ser eliminadas para poder tener solo los números y poder realizar el procesamiento a estos. Para poder eliminarlas se realiza la suma de píxeles en forma vertical y horizontal, dado que las cuadrillas son de color negro entonces el valor de la suma generada en dichos casos es mucho menor al del resto. De acuerdo a los resultados se genera una imagen que forma la máscara de las cuadrillas, la cual es usada para poder eliminar las cuadrillas de la imagen.

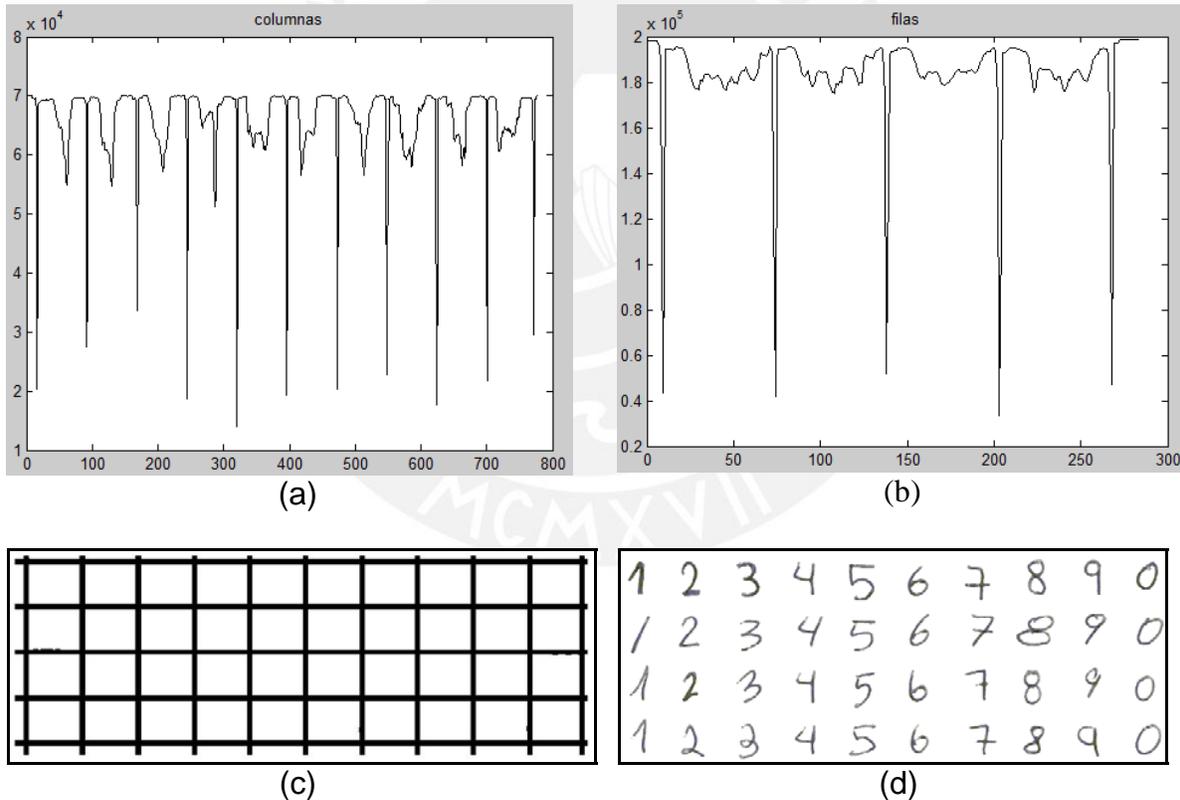


Figura 3.2. (a) suma de píxeles verticales, (b) suma de píxeles horizontales, (c) máscara de la cuadrilla, (d) imagen sin cuadrilla

En la figura 3.2 se puede observar que se tiene una imagen con cuadrillas formadas por 11 líneas verticales y 5 horizontales, al realizar la suma de píxeles horizontales y verticales se obtiene 11 y 5 picos mínimos respectivamente. Basados en estas cantidades de picos y la cantidad de píxeles de la imagen, se realiza la máscara de la cuadrilla para luego poder eliminarlo.

Las imágenes pueden ser escaneadas en escala de grises o colores (modelo RGB), en caso se tenga la imagen en modelo RGB es necesario hacer un conversión a escala de grises. Se trabaja en escala de grises dado que la información de color no es relevante para el proceso de reconocimiento. La fórmula usada para esta transformación es:

$$I = 0.299 * R + 0.587 * G + 0.114 * B \quad (3.1)$$

Luego de tener la imagen en escala de grises se procede a eliminar la cuadrilla para luego extraer los números.

3.2.2 Segmentación

Este proceso comprende en recortar zonas que contengan los números y así poder obtener cada número de forma aislada. Para obtener cada número por separado se realiza un etiquetado de los píxeles que se encuentran conexos. Para ubicar cada dígito dentro de la imagen se hace un recorrido de la imagen y se va etiquetando a cada uno dentro de una vecindad de 8 píxeles, de este modo se va recorriendo hasta encontrar todos los vecinos y a todos estos se les da una misma etiqueta. Del modo anterior se procede con el siguiente grupo, cada grupo va corresponder a un dígito dentro de la imagen [17, Cap.9].

Antes de realizar el proceso se requiere realizar algunas mejoras a la imagen que se encuentra en escala de grises.

Primero se aplica un filtrado promediador, máscara de 3×3 ($h = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$), el cual suaviza los cambios de tonalidad que se dan al escribir con lapicero. Se puede notar que en el trazo hay partes más oscuras que otras y es por ello que se debe suavizar estos cambios producidos por escribir con lapicero.

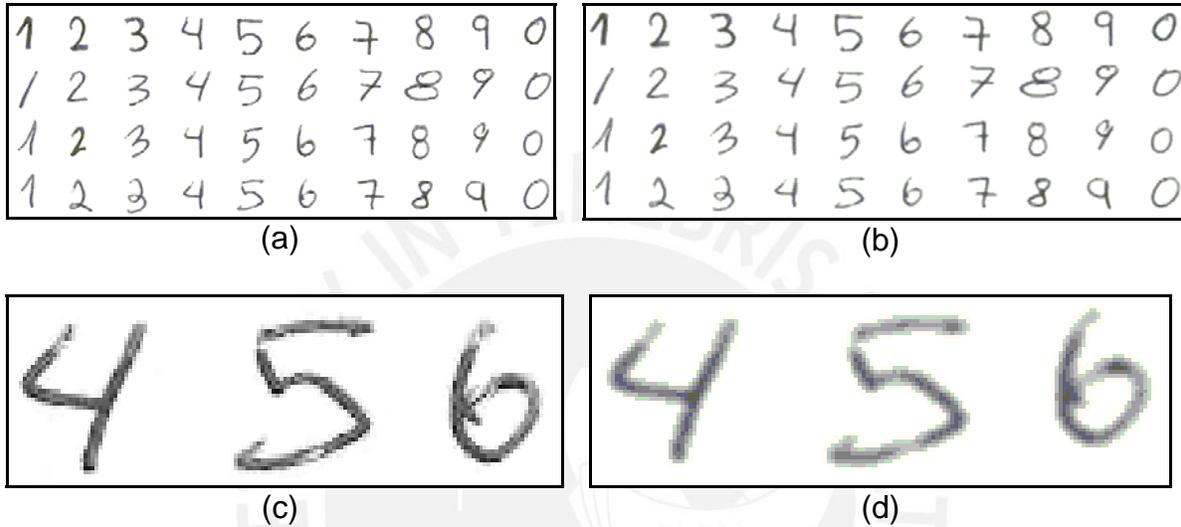


Figura 3.3. (a) Imagen en escala de grises (b) resultado del filtro promediador empleando máscara de 3×3 , (c) ampliación de imagen original, (d) ampliación de imagen filtrada

En la figura 3.3 se muestra el efecto que se tiene al aplicar un filtro promediador, se observa un suavizado en los cambios de tonalidad que se da dentro de la escritura, de esta forma al pasar este filtro ayuda a poder mejorar estos cambios de tonalidad, de no realizarse este suavizado se puede tener algunos inconvenientes en el reconocimiento tal como obtener un número cortado luego del proceso de umbralización.

Luego se realiza un ajuste del contraste usando “Histogram stretching”, el cual consiste en distribuir los píxeles con mayor frecuencia de aparición dentro de todo el rango del histograma. Por lo tanto, es una operación que consiste en modificar el histograma de tal forma que se realice lo mejor posible una distribución de las

intensidades en la escala de valores disponible. Esto equivale a ampliar el histograma de modo que el valor de intensidad más bajo es cero y el más alto es el valor máximo. De esta manera, si los valores del histograma son muy similares entre sí, el estiramiento permitirá proporcionar una mejor distribución con la finalidad de que los píxeles claros sean aún más claros y los oscuros más cerca del negro.

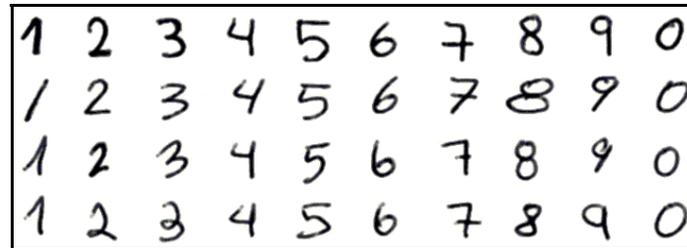


Figura 3.4. Aumento del nivel de brillo

Luego se convierte la imagen a una imagen binaria (blanco y negro). Se realiza una umbralización y mediante el algoritmo de Otsu, el cual es una técnica que se viene utilizando en este tipo de aplicaciones, se calcula el umbral óptimo [18]. Cuando se tiene un histograma "bimodal" se puede reconocer dos picos claramente con lo cual se podría elegir como umbral la media entre los dos picos del histograma o el valor más bajo entre ambos picos pero esto puede estar propenso a errores es por ello que se opta por el método de Otsu.

Este método calcula un umbral maximizando la separación de las clases resultantes dadas por el histograma de la imagen en escala de grises. La imagen se considera como una función bidimensional de la intensidad del nivel de gris, y contiene N píxeles cuyos niveles de gris se encuentran entre 1 y L . El número de píxeles con nivel de gris i se denota como f_i , y la probabilidad de ocurrencia del nivel de gris i en la imagen está dada por:

$$p_i = \frac{f_i}{N} \tag{3.2}$$

En el caso de la binarización de una imagen, los píxeles se dividen en dos clases, C_1 con niveles de gris $[1, \dots, t]$ y C_2 con niveles de gris $[t+1, \dots, L]$. La probabilidad de la ocurrencia de clases está dada por las siguientes expresiones:

$$\omega_0(t) = \sum_{i=1}^t p_i \quad \omega_1(t) = \sum_{i=t+1}^L p_i \quad (3.3)$$

La media para la clase C_1 y la clase C_2 es:

$$u_1 = \sum_{i=1}^t \frac{i \cdot p_i}{\omega_1(t)} \quad u_2 = \sum_{i=t+1}^L \frac{i \cdot p_i}{\omega_2(t)} \quad (3.4)$$

Sea u_t la intensidad media de toda la imagen. Se demuestra:

$$\omega_1 \cdot u_1 + \omega_2 \cdot u_2 = u_t \quad \omega_1 + \omega_2 = 1 \quad (3.5)$$

Usando análisis discriminante, Otsu definió la variancia entre clases de una imagen umbralizada como:

$$\sigma_B^2 = \omega_1 \cdot (u_1 - u_t)^2 + \omega_2 \cdot (u_2 - u_t)^2 \quad (3.6)$$

Para una umbralización de dos niveles, Otsu verificó que el umbral óptimo t^* se elige de manera que σ_B^2 sea máxima; esto es:

$$t^* = \text{Max}_t \{ \sigma_B^2(t) \} \quad 1 \leq t \leq L \quad (3.7)$$

Utilizando el umbral encontrado con el método de Otsu se realiza la umbralización

$$g(x, y) = \begin{cases} 1 & \Leftrightarrow f(x, y) < t^* \\ 0 & \Leftrightarrow f(x, y) \geq t^* \end{cases} \quad (3.8)$$

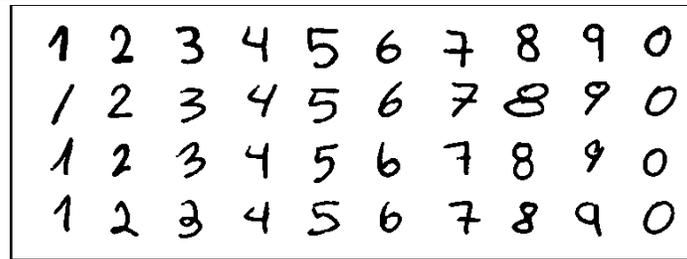


Figura 3.5. Resultado de la umbralización
Luego de tener la imagen en binario se procede a extraer cada dígito por separado (segmentación), se realiza la técnica del etiquetado con una vecindad de 8. Para ello se utilizan dos punteros, un puntero de inicio y otro de parada los cuales sirven para etiquetar los objetos. Esta técnica es muy útil cuando se tiene caracteres de distintos tamaños o que tienen alguna inclinación en cuyos casos no es posible realizar la segmentación usando solo la suma de líneas verticales como horizontales.

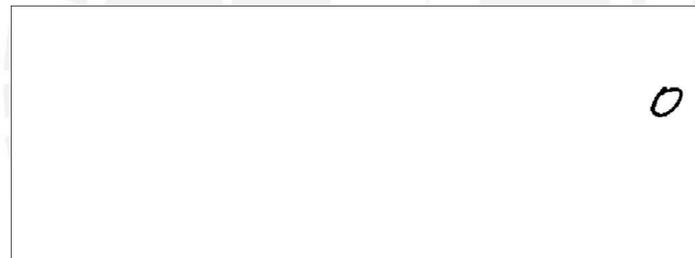


Figura 3.6. Resultado de la segmentación

3.2.3 Normalización

Los números al ser escritos por distintas personas presentan algunas características distintas en cuanto al estilo, tamaño, grosor e inclinación, por estos motivos se debe realizar una normalización para obtener muestras de características parecidas que puedan facilitar el reconocimiento de los caracteres [16].

3.2.3.1 Corrección de la inclinación

La primera operación de normalización es la corrección de la inclinación, lo cual se realiza calculando el ángulo de inclinación; está basado en la idea que si el carácter numérico es rotado con distintos ángulos, y se va tomando en cuenta el ancho del número en cada rotación, se llega a obtener un mínimo ancho lo cual podría significar que se obtuvo la mejor corrección de la inclinación[12]. Luego se realiza una traslación de los píxeles en su misma fila, esta traslación se realiza de acuerdo al ángulo de inclinación [19]. Para realizar la corrección de la inclinación se hace uso de la siguiente ecuación:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & -\tan \beta \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \quad (3.9)$$

Esta forma de corregir la inclinación no serviría en el caso que el ancho del número fuera mayor al de su altura. Un número puede presentar inclinación en solo parte de su trazo que debería ser vertical en cuyo caso si se realiza la rotación, los trazos horizontales que se encuentran correctamente, tendrán una ligera rotación.

3.2.3.2 Cambio del tamaño

Se tiene que tener los números en un tamaño estándar debido a que cada píxel del número corresponde a cada neurona de entrada de la red neuronal, esta cantidad de neuronas es fija.

Se realiza un cambio de tamaño, obteniéndose un tamaño de imagen estándar de 40*28 píxeles y un segundo cambio a 20*14, que tiene una buena resolución. Para realizar este cambio de tamaño se realizó una interpolación bilineal [19]

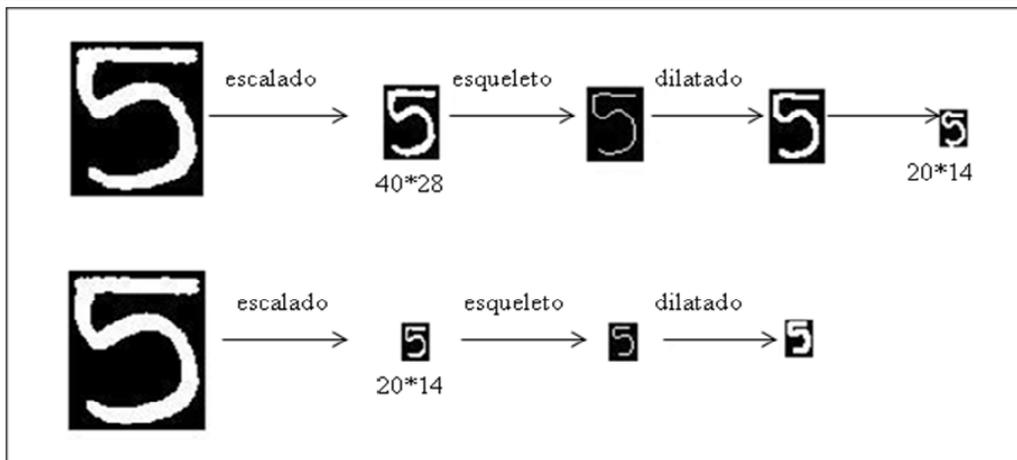


Figura 3.7. Ejemplos de cambio de tamaño

En la figura 3.7 se observa dos formas de realizar el cambio de tamaño de los números, se puede observar que la calidad de la primera imagen quedó mejor que la de la segunda, debido a esto se optó por la primera forma de escalado.

3.2.3.3 Normalización del grosor

La normalización del grosor consiste de 2 partes: El primero es un algoritmo para reducir el grosor y luego otro para aumentarlo, esto se realiza con la finalidad de tener los números con un grosor estándar.

El adelgazamiento consiste en ir borrando sucesivamente los puntos del borde del número, de forma que se preserve su topología. Para realizar el adelgazamiento se hace uso del método de eliminación en paralelo de puntos simples [20]. El borrado de puntos debe seguir un esquema de barridos sucesivos para que la imagen siga teniendo las mismas proporciones que la original y conseguir así que no quede deformada. El borrado de cada rastreo debe hacerse en paralelo, es decir, señalar los píxeles que se borrarán para eliminar todos a la vez.

Luego de realizar el adelgazamiento se procede a aumentar el grosor, para lo cual se procede de forma similar a la del adelgazamiento solo que esta vez se tiene que ir rellenando de píxeles por el contorno de la imagen hasta llegar al grosor deseado.

Cuando la imagen pasa por el proceso de normalización primero se procede a corregir la inclinación. Esta corrección ayuda a obtener números más parecidos entre sí para cualquier estilo de escritura. Luego de corregir la inclinación se procede a cambiar el tamaño de la imagen a una de 40*28, la cual tiene buena resolución, la normalización del grosor se realiza obteniendo el esqueleto y luego se dilata para obtener un trazo uniforme de 3 píxeles de grosor. Finalmente se realiza un segundo escalado para reducirlo a 20*14, esta imagen final luego será procesada por las redes neuronales.

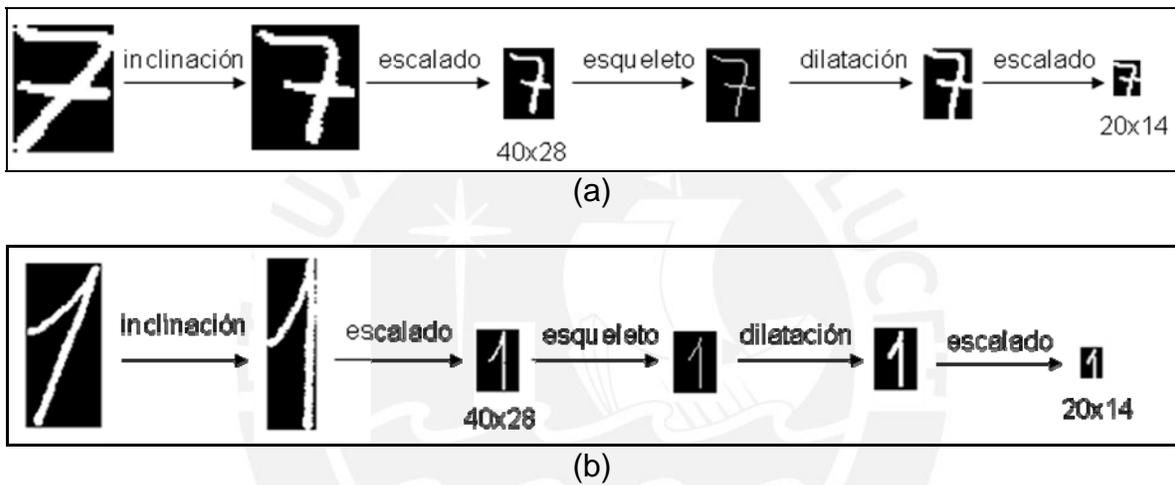


Figura 3.8. (a) y (b) Etapas de normalización

En la Figura 3.8 se muestra a dos números a los cuales se realizó la corrección mediante dos métodos (traslación de píxeles y rotación), se puede notar que de haberse aplicado rotación a la primera, esta hubiera quedado con sus trazos horizontales inclinados; en el caso de la segunda figura hubiera quedado de forma similar el aplicar cualquiera de las dos técnicas. Para este trabajo se propuso usar la primera forma de corrección de inclinación.

3.3 Entrenamiento Red Neuronal Backpropagation

El término backpropagation se refiere al método de entrenamiento para calcular el error en una red *feed-forward* (de aprendizaje supervisado). El entrenamiento de esta red no necesita de complejos algoritmos a diferencia de otro tipo de redes

neuronales. El uso de este tipo de redes neuronales a dado buenos resultados en este tipo de aplicaciones [9]. Para realizar la implementación de la red neuronal mencionada se hizo uso de del toolbox de Matlab, en este trabajo se detallará como se compone la red neuronal utilizada.

La estructura de una red típica es como la que se muestra en la Figura 3.9, esta estructura de red representa a una que cuenta con dos capas (una capa oculta y una capa de salida), lo cual equivale a tener dos capas de tipo perceptrón en cascada; la salida de la primera red, cada capa puede tener diferente número de neuronas, e incluso distinta función de transferencia.

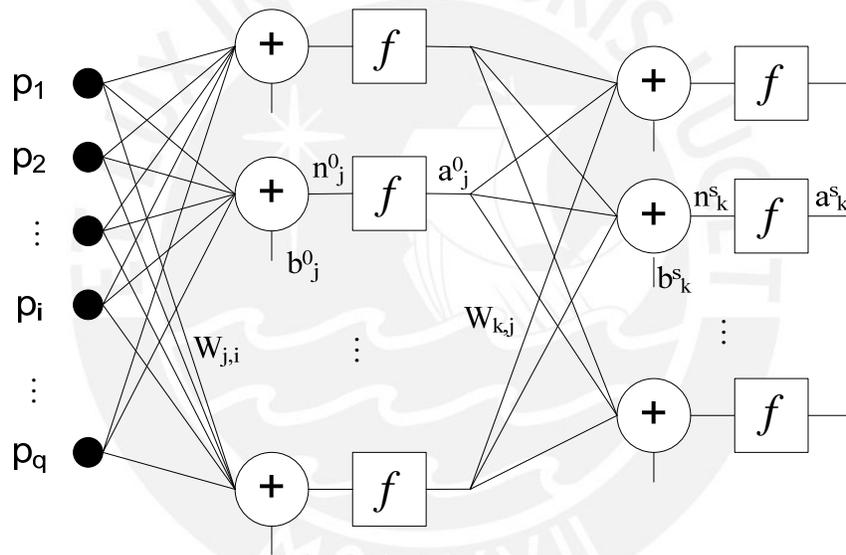


Figura 3.9. Estructura de una red neuronal

El desarrollo de este algoritmo proporciona un método eficiente para entrenar este tipo de redes. Su importancia radica en su capacidad de auto adaptar los pesos de las neuronas intermedias para aprender la relación que existe entre el conjunto de vectores o patrones de entrada y su correspondiente salida, y poder aplicar esa relación después del entrenamiento a nuevos vectores de entrada imperfectos o con ruido.

Esta capacidad se conoce como generalización. La red debe encontrar una representación interna que le permita generar las salidas deseadas durante la etapa de entrenamiento, y posteriormente durante el funcionamiento ser capaz de generar salidas para entradas que no le fueron mostradas durante el aprendizaje pero que se asemejan a algunas de las que si le fueron mostradas.

3.3.1 Regla de Aprendizaje

El aprendizaje se realiza mediante un entrenamiento. Antes de poder realizarlo se debe tener definida la topología de la red: cantidad de neuronas en la capa de entrada, cantidad de capas ocultas y cantidad de neuronas en cada una de ellas, número de neuronas en la capa de salida y funciones de transferencia requeridas en cada capa. Luego de haber definido la topología se debe de asignar valores iniciales a cada uno de los parámetros que conforma la red.

Es importante resaltar que no existe una técnica para determinar el número de capas ocultas, ni el número de neuronas que debe contener cada una de ellas para un problema específico, esta elección se determina en base a pruebas.

La deducción matemática de este procedimiento se realizará para una red con una capa de entrada, una capa oculta y una capa de salida [12].

Para la deducción se debe tener en cuenta lo siguiente:

- q : Equivale al número de componentes del vector de entrada.
- m : Número de neuronas de la capa oculta
- l : Número de neuronas de la capa de salida

Para iniciar el entrenamiento se le presenta a la red un patrón de entrenamiento, el cual tiene q componentes.

$$P = \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_i \\ \vdots \\ P_q \end{bmatrix} \quad (3.10)$$

Cuando se le presenta a la red un patrón de entrenamiento, este se propaga a través de las conexiones existentes produciendo una entrada neta n en cada una de las neuronas de la siguiente capa, la entrada neta a la neurona j de la siguiente capa debido a la presencia de un patrón de entrenamiento en la entrada esta dada por la (ec.3.11), nótese que la entrada neta es el valor justo antes de pasar por la función de transferencia

$$n_j^0 = \sum_{i=1}^q W_{ji}^0 P_i + b_j^0 \quad (3.11)$$

W_{ji}^0 : Peso que une la componente i de la entrada con la neurona j de la capa oculta

P_i : Componente i del vector p que contiene el patrón de entrenamiento de q componentes

b_j^0 : Ganancia de la neurona j de la capa oculta

Donde el superíndice (0) representa la capa a la que pertenece cada parámetro, en este caso la capa oculta.

Cada una de las neuronas de la capa oculta tiene como salida a_j^0

$$a_j^0 = f^0 \left(\sum_{i=1}^q W_{ji}^0 P_i + b_j^0 \right) \quad (3.12)$$

f^0 : Función de transferencia de las neuronas de la capa oculta

Las salidas a_j^0 de las neuronas de la capa oculta (de l componentes) son las entradas a los pesos de conexión de la capa de salida.

$$n_k^s = \sum_{j=1}^m W_{kj}^s a_j^0 + b_k^s \quad (3.13)$$

W_{kj}^s : Peso que une la neurona j de la capa oculta con la neurona k de la capa de salida, la cual cuenta con s neuronas

a_j^0 : Salida de la neurona j de la capa oculta, la cual cuenta con m neuronas.

b_k^s : Ganancia de la neurona k de la capa de salida.

n_k^s : Entrada neta a la neurona k de la capa de salida

La red produce una salida final descrita por la ecuación:

$$a_k^s = f^s \left(\sum_{j=1}^m W_{kj}^s a_j^0 + b_k^s \right) \quad (3.14)$$

f^s : Función de transferencia de las neuronas de la capa de salida

La salida de la red de cada neurona a_k^s se compara con la salida deseada t_k para calcular el error en cada unidad de salida.

$$\delta_k = (t_k - a_k^s) \quad (3.15)$$

El error debido a cada patrón p propagado está dado por:

$$ep^2 = \frac{1}{2} \sum_{k=1}^s (\delta_k)^2 \quad (3.16)$$

ep^2 : Error medio cuadrático para cada patrón de entrada p

δ_k : Error en la neurona k de la capa de salida con l neuronas

Este proceso se repite para el número total de patrones de entrenamiento (r), para un proceso de aprendizaje exitoso el objetivo del algoritmo es actualizar todos los pesos y ganancias de la red minimizando el error medio cuadrático total.

$$e^2 = \sum_{p=1}^r ep^2 \quad (3.17)$$

e^2 : Error total en el proceso de aprendizaje en una iteración luego de haber presentado a la red los r patrones de entrenamiento

El error que genera una red neuronal en función de sus pesos, genera un espacio de n dimensiones, donde n es el número de pesos de conexión de la red, al evaluar el gradiente del error en un punto de esta superficie se obtendrá la

dirección en la cual la función del error tendrá un mayor crecimiento, como el objetivo del proceso de aprendizaje es minimizar el error debe tomarse la dirección negativa del gradiente para obtener el mayor decremento del error y de esta forma su minimización, condición requerida para realizar la actualización de la matriz de pesos en el algoritmo Backpropagation:

$$W_{k+1} = W_k - \alpha \nabla e p^2 \quad (3.18)$$

El gradiente negativo de $e p^2$ se denotará como $-\nabla e p^2$ y se calcula como la derivada del error respecto a todos los pesos de la red.

Las funciones de transferencia f utilizadas son las siguientes:

Logsig

$$f(n) = \frac{1}{1 + e^{-n}} \quad (3.19)$$

Tansig:

$$f(n) = \frac{e^n - e^{-n}}{e^n + e^{-n}} \quad (3.20)$$

Error para las neuronas de la capa de salida:

$$\delta_k^s = (t_k - a_k^s) f'^s(n_k^s) \quad (3.21)$$

Este algoritmo se denomina Backpropagation o de propagación inversa debido a que el error se propaga de manera inversa al funcionamiento normal de la red, de esta forma, el algoritmo encuentra el error en el proceso de aprendizaje desde las capas más internas hasta llegar a la entrada; con base en el cálculo de este error se actualizan los pesos y ganancias de cada capa.

Error para las neuronas de la capa oculta:

$$\delta_j^0 = f'^0(n_j^0) * \sum_{k=1}^l \delta_k^s \cdot W_{kj}^s \quad (3.22)$$

Luego de encontrar el valor del gradiente del error se procede a actualizar los pesos de todas las capas empezando por la de salida.

$$W_{kj}(t+1) = W_{kj}(t) - 2\alpha\delta_k^s \quad (3.23)$$

$$b_k(t+1) = b_k(t) - 2\alpha\delta_k^s \quad (3.24)$$

α : Rata de aprendizaje que varía entre 0 y 1 dependiendo de las características del problema a solucionar.

Luego de actualizar los pesos y ganancias de la capa de salida se procede a actualizar los pesos y ganancias de la capa oculta.

$$W_{ji}(t+1) = W_{ji}(t) - 2\alpha\delta_j^0 p_i \quad (3.25)$$

$$b_j(t+1) = b_j(t) - 2\alpha\delta_j^0 \quad (3.26)$$

En el aprendizaje se usa la técnica de la gradiente descendiente en cual es conveniente avanzar por la superficie de error con incremento pequeño de los pasos; esto se debe a que solo se tiene 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 del punto mínimo, con incrementos pequeños aunque se tarde más en llegar, se evita que esto ocurra. El elegir un incremento adecuado influye en la velocidad de convergencia del algoritmo, esta velocidad se controla a través de la rata de aprendizaje α .

3.3.2 Estructura de la red neuronal

La red neuronal usada es la "Backpropagation". La red está compuesta por una capa oculta y una capa de salida. La función de transferencia usada para la capa oculta es la "Tangente Sigmoidal Hiperbólica" y para la capa de salida es la "Sigmoidal Logarítmico". La función de la capa oculta nos devuelve valores en el rango de -1 a 1, la función de la capa se salida arroja valores entre 0 y 1.

Para este trabajo se consideran las cifras arábicas del 0 al 9. El tamaño de la muestra para cada carácter a ser reconocidos es de 20 de alto y 14 de ancho, por lo que se tiene 280 píxeles, este tamaño tendrán todas las muestras a ser reconocidas.

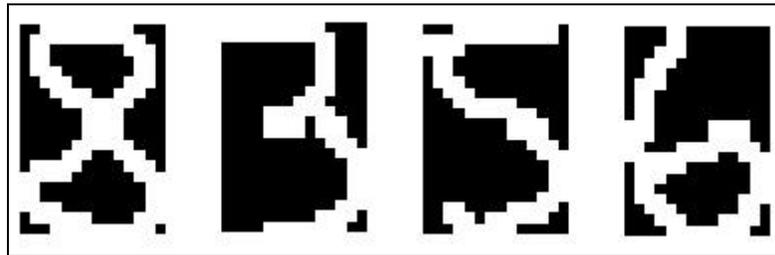


Figura 3.10. Números de 20*14 píxeles listos para ser reconocidos

La red neural recibe un vector como entrada, es necesario que los 280 píxeles de la imagen se encuentren en un solo vector en el cual las filas sean concatenadas una después de la otra. Para la capa oculta se utilizó 150 neuronas y en la capa de salida se tiene 10, donde cada salida corresponde a los números del 0 al 9.

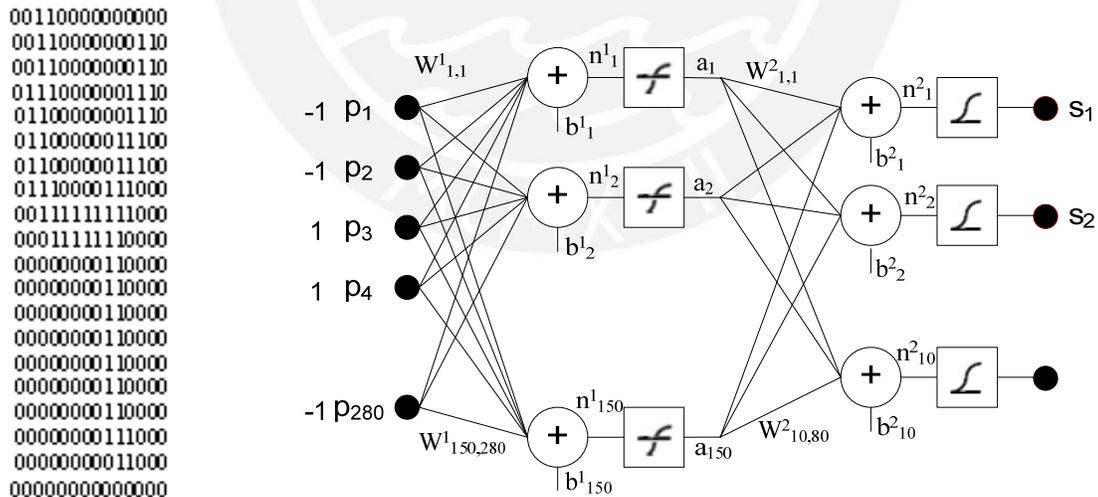


Figura 3.11. Esquema de la Red Neuronal

El algoritmo de entrenamiento usado en esta red fue: "Reslient Backpropagation", cuya función de entrenamiento es llamada "Trainrp". La función de entrenamiento

fue elegida dado que cuando se usa un algoritmo de pasos descendientes ocurren problemas cuando el gradiente empieza a tomar valores muy pequeños y por lo tanto no habrán cambios representativos en los pesos y las ganancias, así se encuentran bastante lejos de sus valores óptimos, En el algoritmo mencionado solamente el signo de la derivada es utilizado para determinar la dirección de actualización de los parámetros, la magnitud de las derivadas no tiene efecto en la actualización.

En la Figura 3.12 se tiene la gráfica del proceso de entrenamiento, se puede notar que el error empieza en un valor elevado y conforme avanzan las épocas de entrenamiento el error va disminuyendo hasta que el error sea constante. El entrenamiento termina cuando se llega a un error mínimo (idealmente puede ser cero) que es establecido por el programador, o cuando se llega a una cantidad de épocas determinada, en este caso el entrenamiento culminó cuando se llegó a la cantidad de épocas establecidas y se puede notar que el error permanecía constante después de varias iteraciones, por lo cual este error no tiene tendencia a seguir disminuyendo.

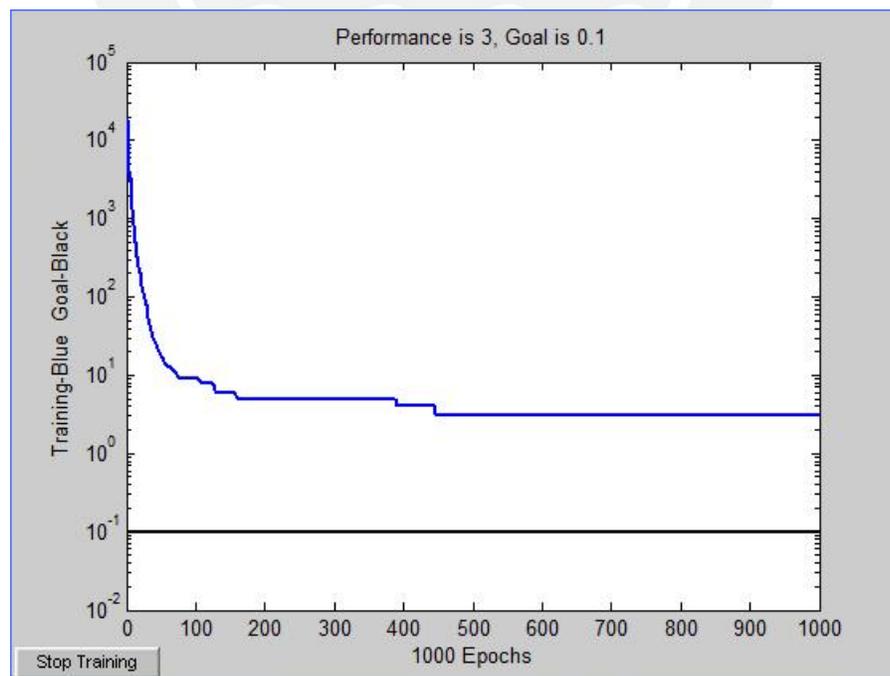


Figura 3.12. Gráfica de entrenamiento de la Red Neuronal

3.4 Reconocimiento del Número

El reconocimiento del carácter consiste en clasificar dentro de una gama de posibilidades a un carácter como tal, lo que permitirá generar una salida ASCII.

El reconocimiento se realiza en base a redes neuronales artificiales debido a que son muy rápidas y ofrecen buenos resultados en cuanto a exactitud. Son especialmente efectivas para predecir eventos en donde la red tiene una buena base de datos de ejemplos a ser analizados.

Este reconocimiento se logra luego de que se produce una coincidencia entre un carácter numérico almacenado en memoria con la imagen obtenida. Para ser considerado una coincidencia el porcentaje de coincidencia debe ser lo suficientemente alto.

Luego del entrenamiento de la red neuronal se obtiene los valores para las variables internas de la red neuronal, como pesos y constantes. El reconocimiento del número se hace de manera similar al reconocimiento solo que en este paso no se tiene que ajustar el valor de las variables internas y el valor que se obtiene al salida de la red es comparado con un valor umbral de 0.85 (escala del 0 al 1) para poder establecer si el número fue reconocido

CAPÍTULO 4: PRUEBAS Y ANÁLISIS DE LOS RESULTADOS

Se utilizaron muestras de números escritos por distintas personas (150 en total) en papel con formato en cuadrículas. Dentro de cada cuadrícula se escribe un número.

Las muestras de los números no debería tocar o cruzar el borde de la cuadrícula debido a que, en el procesamiento de la imagen la cuadrilla es eliminada y entonces el número quedaría cortado e incompleto, esto afectaría cuando se desee reconocer dado que al número al ser cortado se puede tomar como dos figuras aisladas. Los números fueron escritos por distintas personas.

Se obtuvieron en total 6000 muestras de números de los cuales se usó 4000 para el entrenamiento de la red neuronal y los 2000 números restantes se usaron para poder realizar pruebas de reconocimiento.

Dentro de la muestra se encontró números que se encuentran escritos de distintas formas, pero se realizaron las pruebas con los números para los cuales se limitó este trabajo. Los números eran de buena escritura, escritos con calma por las personas y debido a esto no se presentaban problemas de posibles borrones o números de escritura confusa.

Adicional a las pruebas realizadas con las muestras recolectadas, se realizaron otras pruebas con la base de datos MNIST (9104 muestras), para ver el rendimiento de la estructura de la red neuronal propuesta.

4.1 Entorno gráfico del módulo de reconocimiento

Se realizó una interfaz gráfica para poder realizar algunas pruebas y demostraciones de cómo se realiza el reconocimiento de números manuscritos, esta interfaz fue desarrollada haciendo uso del software Matlab.

La interfaz cuenta con una opción para poder abrir un archivo que contenga números manuscritos que hayan sido previamente escaneados, con la opción cuadrícula se quita los cuadrados, en caso tuviera, que sirvieron como referencia para escribir los números; luego se pulsa seleccionar para poder seleccionar un número haciendo uso del "Mouse". Después de seleccionarlo se puede ver los procesos principales que se realizan antes de poder ser reconocidos, como la etapa de normalización. Por último con el botón "reconocer" se realiza el reconocimiento de la muestra seleccionada.

En la Figura 4.1. Se puede observar el entorno gráfico del módulo de reconocimiento propuesto en este trabajo, se puede observar las opciones con las que se cuenta, lo cual sirve para poder mostrar los procesos que se realizan para poder reconocer los números.

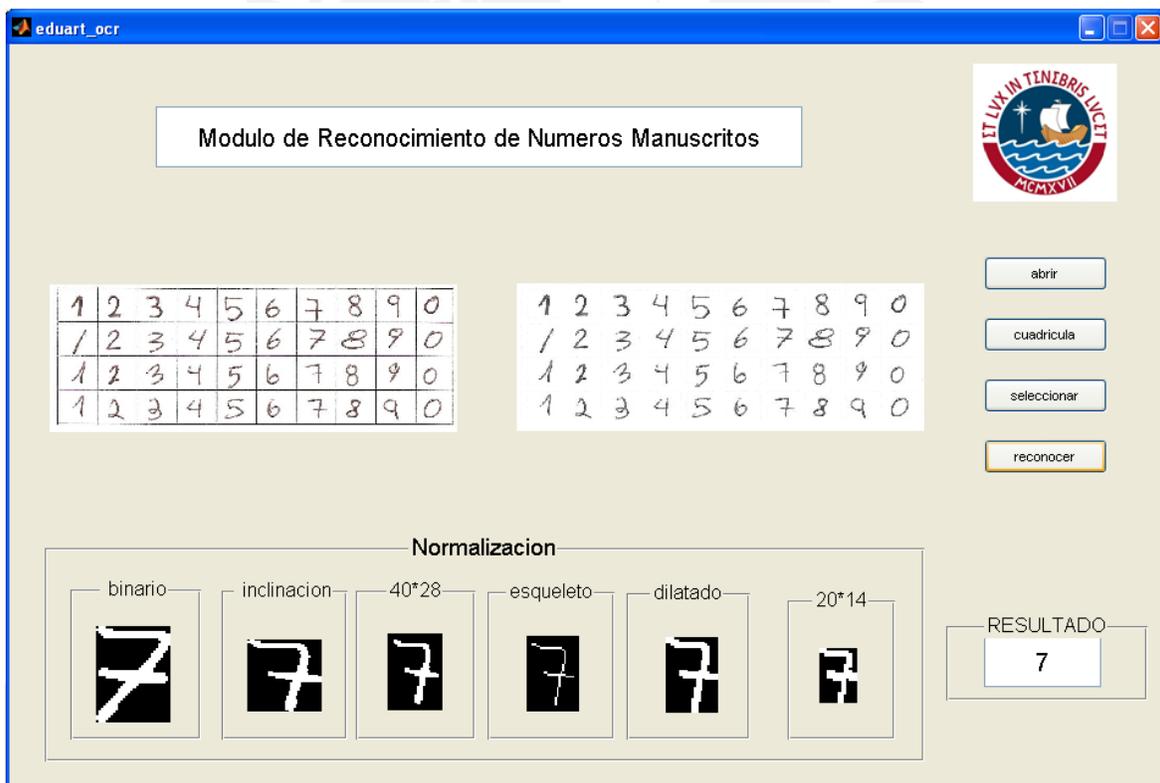


Figura 4.1. Aciertos a la salida de la red

4.2 Análisis de Reconocimiento y Eficiencia

Para realizar el análisis de los resultados se realiza la revisión de estos en cuanto al nivel de reconocimiento y eficiencia que se obtienen dado que se puede tener un alto nivel de reconocimiento pero que puede estar asociado a bajo niveles de eficiencia.

En la Figura 4.2 se muestra los niveles de reconocimiento que se obtuvieron para cada carácter numérico (200 muestras por carácter), pero éstos se pueden haber reconocido incorrectamente. Esta gráfica no permite visualizar mayores detalles sobre el real funcionamiento del sistema. Se puede tener que todo el conjunto de un determinado número fue reconocido al 100% pero puede haber un porcentaje que haya sido reconocido incorrectamente, lo importante es saber el grado de eficiencia del sistema. Se requiere un análisis de eficiencia.

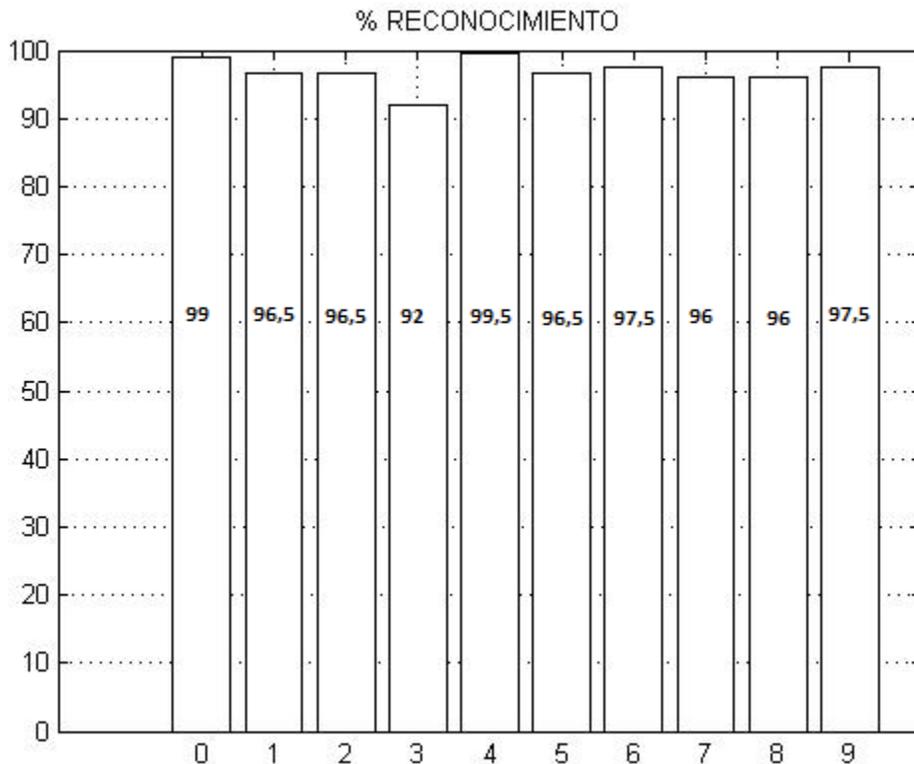


Figura 4.2. Porcentaje de éxito de reconocimiento de los números

En la Figura 4.3 se muestra una gráfica que presenta una mejor información del funcionamiento del sistema. Se muestra la eficiencia del reconocimiento y de acuerdo a esto se mide la eficiencia de la red neuronal empleada durante el reconocimiento de números. Esta gráfica permite tener una mayor visión para poder realizar algunas mejoras al sistema.

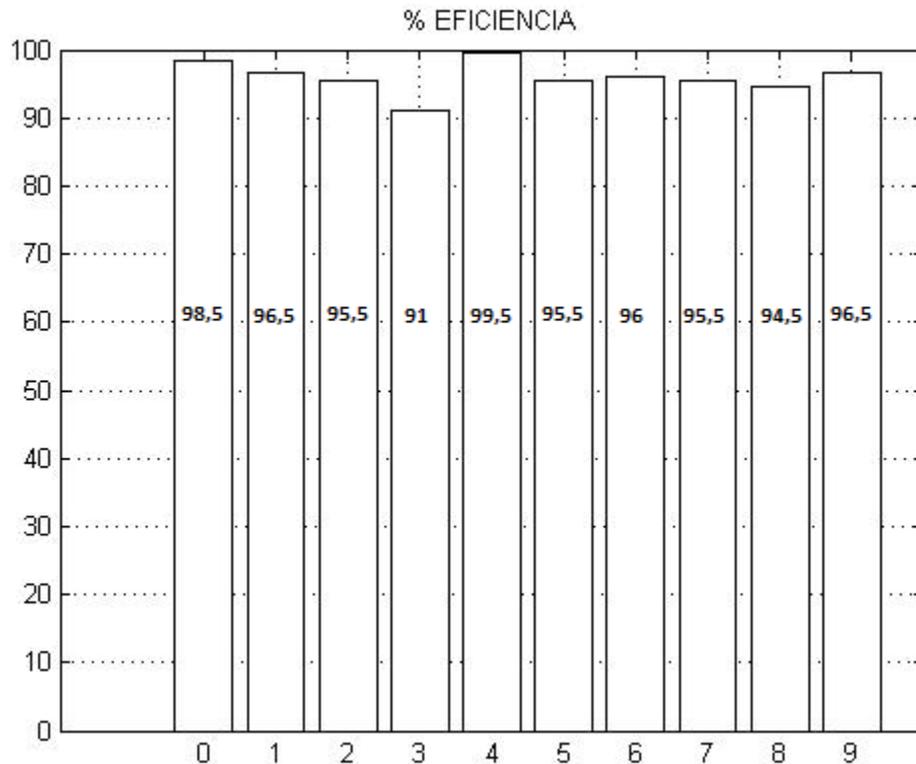


Figura 4.3. Porcentaje de eficiencia

4.3 Distribución de aciertos en la salida

Para poder analizar la eficiencia del sistema se debe realizar un análisis de acuerdo al reconocimiento para cada tipo de número. Se analiza con cual tipo de número se suele confundir el clasificador y en base a estos realizar las mejoras al sistema. Se realiza las gráficas de eficiencia en el reconocimiento de cada número y en base a estas se observa con cual número se tiende a equivocar en el reconocimiento de determinado número.

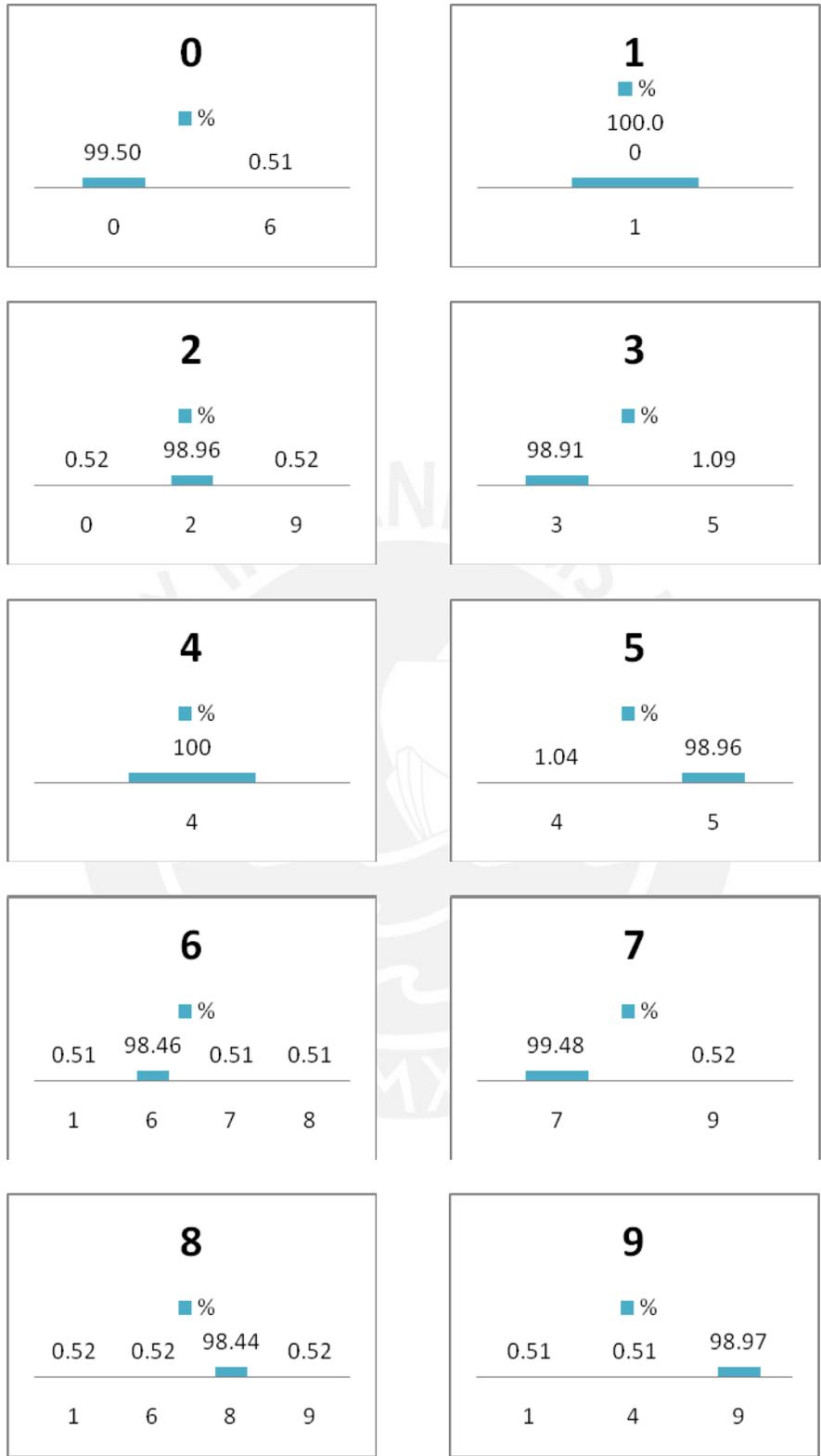


Figura 4.4. Aciertos a la salida de la red

En la Figura 4.4, se representan gráficas de eficiencia durante el proceso de reconocimiento, haciendo uso de las redes neuronales. Para un patrón de entrada la red neuronal puede dar un valor erróneo en la salida y la red no puede advertir este error dado que se activa la neurona correspondiente a un patrón de entrada específico. En estas gráficas se puede ver con que otro tipo de caracteres se dio un reconocimiento erróneo. Por ejemplo en el caso del número ocho se dio un error de reconocimiento confundiéndose este con el número 6 y el uno, teniendo en cuenta estos posibles errores se puede realizar módulos de post-proceso o aplicar otras técnicas de reconocimiento en el cual se realice un análisis entre estos tipos de números y luego llegar a una respuesta por consenso acorde a los resultados obtenidos por las distintas técnicas empleadas.

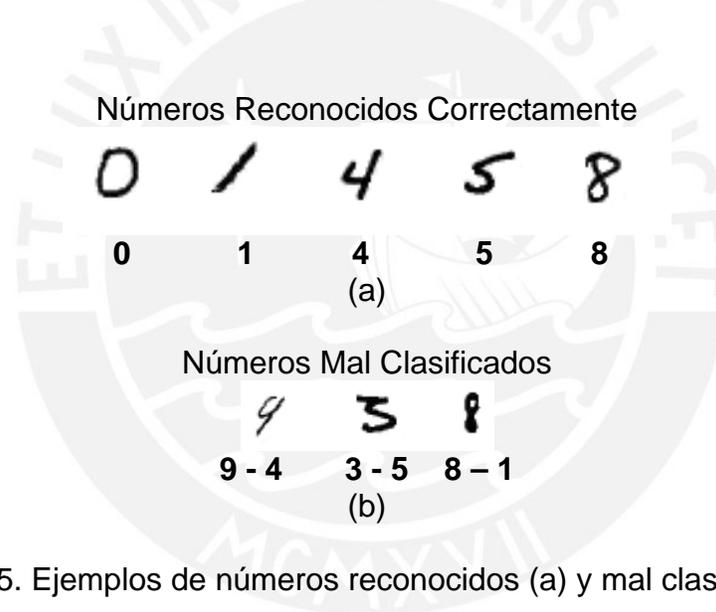


Figura 4.5. Ejemplos de números reconocidos (a) y mal clasificados (b)

En la Figura 4.5 se puede observar algunos ejemplos de los números que fueron reconocidos y los cuales fueron mal clasificados. Dentro de los mal clasificados se puede observar que los dígitos comparten la mayor cantidad de características con las que realizó la clasificación incorrecta. En el caso del ocho por ser delgado se confunde con el uno y el caso del tres la parte superior de este y la parte del cuerpo es similar a la del uno, por las características dadas anteriormente es que el clasificador reconoce a estos números de forma incorrecta.

4.4 Comparación de resultados con otros trabajos

En la Tabla 4.1 se presentan los resultados del conjunto de muestras que se usaron para el testeo con el clasificador propuesto. Por ejemplo para el caso del siete se usaron 200 patrones para el testeo de los cuales el clasificador reconoció 192 (96%) pero solo 189 (94.5%) del total fueron reconocidos correctamente, de esta forma se obtuvieron resultados para cada tipo de número que se reconoció.

Tabla 4.1. Resultado de los reconocimientos de números manuscritos

Números	cantidad de patrones	% Reconocidos	% Eficiencia
0	200	99.00	98.50
1	200	96.50	96.50
2	200	96.50	95.50
3	200	92.00	91.00
4	200	99.50	99.50
5	200	96.50	95.50
6	200	97.50	96.00
7	200	96.00	95.50
8	200	96.00	94.50
9	200	97.50	96.50

En la tabla 4.2, se observa el resultado final que se obtuvo para todo el conjunto de testeo (2000 muestras). Se debe tener en cuenta que la mayoría de números usados eran de escritura calmada y es por esto que en el conjunto de muestras no se encontraban números incompletos o que presenten problemas de borrones.

Tabla 4.2. Resultado total

TOTAL			
cantidad de muestras	% Reconocidos	% Error	% Rechazados
2000	95.9%	0.8%	3.3%

Se realizó el trabajo de entrenamiento y reconocimiento cuando no se aplica el método de corrección de inclinación a los caracteres, los resultados obtenidos (94.9%) no difieren mucho de lo obtenido cuando si se aplica este método

(95.9%). En la Figura 4.6, se puede observar los resultados obtenidos, se puede notar que para algunos casos el reconocimiento mejora en comparación a los resultados mostrados en la Figura 4.3.

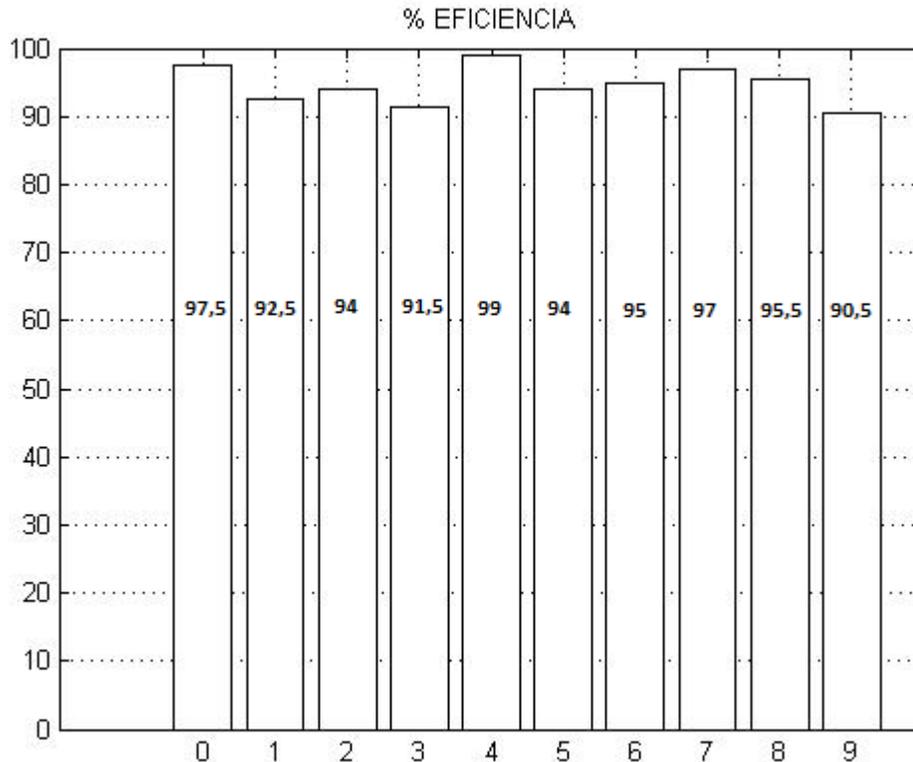


Figura 4.6. Porcentaje de eficiencia sin corrección de la inclinación.

El realizar la corrección de inclinación sería de gran apoyo cuando se encuentran números que tengan una severa inclinación, pero cuando la inclinación es leve la red neural está en capacidad de poder reconocer este número lo cual depende de la variedad que se tenga en las muestras que fueron usados en el entrenamiento.

Para realizar una buena corrección de inclinación se debe hacer uso de un método eficaz, para poder hallar el ángulo de inclinación, dado que un incorrecto análisis de esto puede ocasionar que para algunos números se realice corrección de inclinación sin necesidad de que estos requieran de ello, lo cual puede dar lugar a posibles errores en la fase de reconocimiento.

4.4.1 Pruebas con la base de datos MNIST

Se realizaron pruebas para ver el rendimiento de la estructura de la red neuronal propuesta, para estas pruebas se realizó el uso de las muestras de la base de datos del MNIST. Esta base de datos fue hecha a partir de la base de datos del NIST (National Institute of Standards and Technology), esta es usada para probar las distintas técnicas y métodos de aprendizaje de reconocimiento que se vienen realizando dado que es una muestra muy rica en cuanto a cantidad y estilo de escritura de los números.

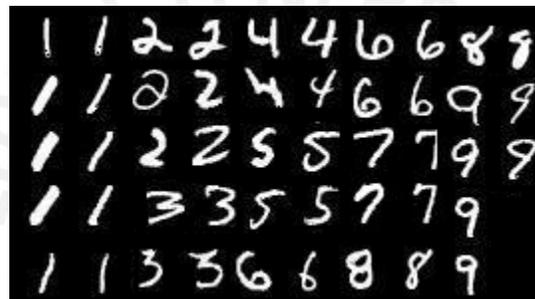


Figura 4.7. Ejemplos de muestra base de datos MNIST

En la Figura 4.7, se presenta una muestra de los números tal como se encuentra en la base de datos MNIST, las muestras con la que cuenta esta base de datos son imágenes binarias.

Con la finalidad de revisar el rendimiento de la red neuronal propuesta para este trabajo, se realizaron pruebas teniendo en cuenta solo muestras de la base de datos del MNIST tanto para el entrenamiento y el reconocimiento. En la Tabla 4.3, Se puede ver los resultados obtenidos en el reconocimiento de los números, para hacer esta prueba se usó 6000 muestras para el entrenamiento y 3104 para el reconocimiento, se puede observar la cantidad de muestras usadas para el reconocimiento y las usadas para el entrenamiento.

Tabla 4.3. Resultado de los reconocimientos de números manuscritos usando la base de datos MNIST

Números	Cantidad Entrenar	Cantidad Reconocer	% Reconocidos	% Error	% Rechazados
0	635	317	99.37	0.63	0.00
1	714	397	97.99	1.01	1.00
2	581	305	83.61	7.21	9.18
3	640	354	87.57	4.24	8.19
4	558	293	94.20	1.02	4.78
5	502	274	86.86	4.74	8.4
6	448	217	86.18	1.84	11.98
7	706	342	88.89	4.09	7.02
8	664	328	86.28	8.23	5.49
9	552	277	91.70	3.61	4.69

Tabla 4.4. Resultado total de pruebas con base de datos MNIST

TOTAL			
cantidad de muestras	% Reconocidos	% Error	% Rechazados
8026	90.11	3.67	6.22

En la muestra de esta base de datos, varios de estos números tienen una escritura diferente a los utilizados para este trabajo pero se puede notar que la red neuronal es capaz de poder identificarlos y cumplir con el trabajo de reconocimiento con un alto porcentaje de eficiencia.

Se realizó pruebas de reconocimiento teniendo en cuenta las muestras de la base de datos MNIST (muestras para entrenamiento) y las muestras que se recaudó para este trabajo (muestras para reconocimiento). En la Figura 4.8, se puede ver el resultado obtenido de la prueba mencionada, se observa que el porcentaje de reconocimiento eficiente promedio (68,50%) es menor al obtenido cuando se realiza el entrenamiento con las muestras recolectadas (95,90%). Este resultado es debido a que ambas muestras no siguen el mismo pre-procesamiento, en el caso de las muestra MNIST estas ya se encuentran en binario con lo cual no se

tiene información del proceso previo al cual fueron sometidas por lo cual no siguen el mismo proceso que las muestras obtenidas para este trabajo. Estas muestras son muy variadas en cuanto a tipo y forma, se requiere de otros pasos antes del reconocimiento, similares a los realizados para los números que se recaudó, dado que se encuentran en binario no se puede hacer el mismo proceso para poder corregir los defectos de la uniformidad del trazo y obtener imágenes en binario.

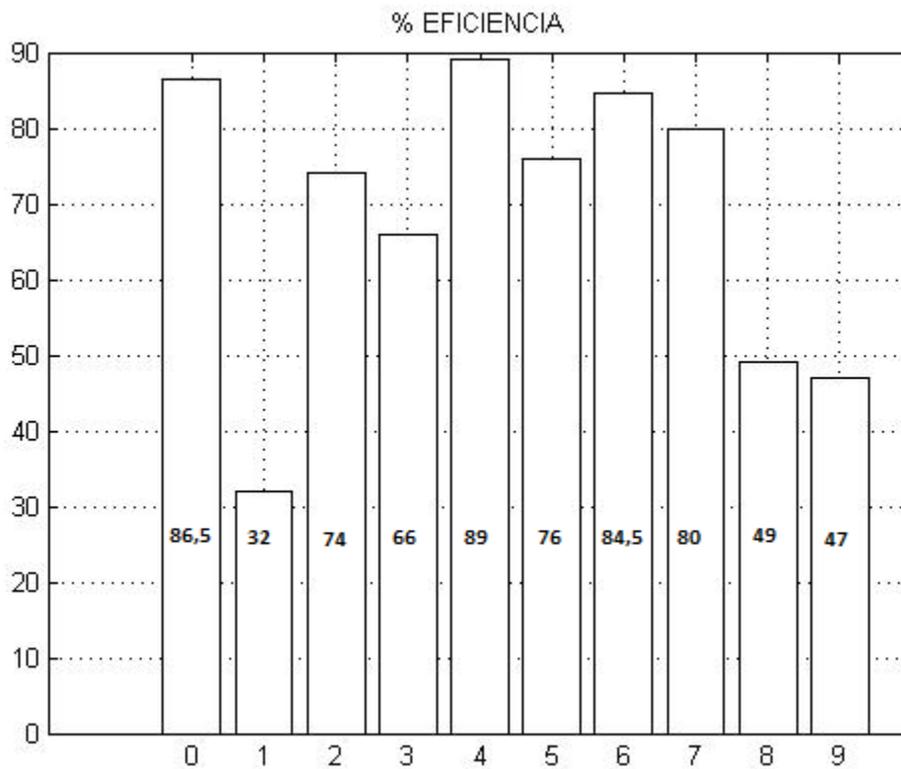


Figura 4.8. Porcentaje de eficiencia sin corrección de la inclinación.

En la Tabla 4.5 se puede observar los resultados obtenidos por otros autores en el reconocimiento de dígitos manuscritos a través del uso de diferentes métodos basados en redes neuronales. Algunos, indicados con “(*)”, han utilizado para el entrenamiento y testeo, la base de datos de la Universidad de Concordia. El marcado con con “(**)” hizo uso de la base de datos del National Institute of Standards and Technology (NIST).

Tabla 4.5. Tabla de resultados de algunos autores en el reconocimiento de números manuscritos, utilizando distintos reconocedores basados en redes neuronales

Autores	% Reconocidos	% Error	% Rechazados	Patrones entrenamiento	Patrones Prueba
(*) Nadal (1988)	86.05	2.25	11.70	4000	2000
(*) Krzyzak (1990)	86.40	1.00	12.60	4000	2000
(*) Krzyzak (1990)	94.85	5.15	0.00	4000	2000
Le Cun (1990)	92.00	2.00	6.00	7291	2007
Le Cun (1990)	96.40	3.40	0.20	7291	2007
(*) Mai (1990)	92.95	2.15	4.90	4000	2000
(*) Suen (1990)	93.05	0.00	6.95	4000	2000
Knerr (1992)	90.30	1.00	8.70	7200	1800
Lemarie (1993)	97.97	2.03	0.00	8783	7394
(*) Kim & Lee (1994)	95.40	4.60	0.00	4000	2000
(*) Lee (1995)	97.80	2.20	0.00	4000	2000
(*) Cho (1997)	96.05	3.95	0.00	4000	2000
(*) Leticia (2003)	91.90	8.10	0.00	4000	2000
(**) R. palacios (2003)	84.00	5.50	10.50	60089	28000

En la tabla anterior se puede notar que en la mayoría de casos se llegaron a valores de reconocimiento superiores al 90% pero el porcentaje de error oscila alrededor del 3%. Estos resultados son buenos pero para algunas aplicaciones es imprescindible tener un error cercano de 0% dado que los números al ser rechazados pueden ser digitalizados manualmente, en cambio si se tiene algún reconocimiento erróneo esto dificultaría el trabajo de corrección dado que difícilmente se encuentra cuales sean los dígitos erróneos.

CONCLUSIONES

- Se obtuvo un reconocimiento, para las muestras recolectadas (2000 muestras), de 95.9%, con error de 0.8% y un 3.3% no pudieron ser reconocidos. Estos números fueron escritos de manera calmada por lo cual no se tuvo problemas con posibles borrones o números incompletos.
- Se realizó pruebas, para la estructura de la red, con la base de datos del MNIST y se obtuvo un total de reconocimiento del 90.11%, error 3.67% y un 6.22% no pudo ser reconocido. Esta base de datos tiene un mayor número y más variada muestras los cuales presentan diversas formas, pero se logra un alto grado de reconocimiento haciendo uso de la red neuronal propuesta.
- Debido a tener números que pueden ser escritos con lápiz o lapicero, fue necesario realizar unas mejoras previas a la imagen debido a que al usar lapicero el trazo no es parejo. Con un filtro promediador se puede mejorar estos efectos y se puede luego realizar una buena segmentación.
- La etapa de normalización sirvió notablemente para poder aumentar los niveles de reconocimiento, la etapa de corrección de la inclinación no es preponderante para los resultados dado que la red neuronal puede lograr reconocer los números que tengan cierto grado de inclinación.
- El número de neuronas en la entrada de la red fue adecuado (280 neuronas), el cual se corresponde a los 280 píxeles de una imagen de 20*14, con este tamaño se logra tener la representación del número original con las mismas características y además con esa cantidad de neuronas se obtuvieron buenos resultados.
- El uso de redes neuronales permite que se pueda obtener altos porcentajes de reconocimiento lo cual depende bastante de las etapas previas que se realicen

- En la clasificación, el sistema reconocedor siempre da una respuesta, es decir, siempre asocia el patrón de entrada a una clase con la cual el patrón comparte el mayor número de características. Esta característica quizás no puede ser considerada importante para el caso de reconocer números manuscritos pero si puede llegar a ser fundamental cuando se quiera hacer uso de redes neuronales para problemas en el cual se tenga que tomar decisiones.



RECOMENDACIONES

- Para que el sistema de mejores resultados, se debe mejorar las etapas previas al reconocimiento para poder tener una imagen de buena calidad a la entrada de la red neuronal. Los niveles de error que se presentan en el reconocimiento son atribuidos a los trabajos de segmentación.
- Se debe tener en consideración una adecuada resolución de la imagen a la entrada dado que esta condiciona la cantidad de información que se puede perder de la imagen original en caso la resolución sea muy baja.
- Para poder tener un clasificador más confiable se debe de tener una cantidad de muestras bastante variada para poder realizar el entrenamiento y así tener un clasificador más robusto para poder tener un mayor rango de tipos de números que puedan ser reconocidos.
- Para poder evitar posibles errores de clasificación o tomar una decisión, en caso el reconocedor tenga algunos posibles valores de reconocimiento, es necesario realizar algunas etapas de post-procesamiento en el cual se realicen algunas consideraciones para poder diferenciar entre las posibles respuestas que pueda dar el clasificador.
- Para poder tener un clasificador más confiable se puede realizar el reconocimiento usando clasificadores en paralelo (mejor si están basados en distintas técnicas), con esto es posible rechazar un dígito cuando no se obtiene un determinado nivel de acuerdo entre los resultados.
- Para obtener mayor porcentaje de acierto al reconocer se puede realizar un trabajo conjunto entre el bloque de segmentación y el de reconocimiento de tal forma que se tenga distintos algoritmos de segmentación y en el caso que no se puede reconocer el número se aplique otro algoritmo y así sucesivamente



BIBLIOGRAFIA

- [1] Arica, N.; Yarman-Vural, F.T.
“An overview of character recognition focused on off-line handwriting”, IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews
- [2] R,Palacios and A. Gupta
2008, “A system for processing hand written bank checks automatically”, submitted to Image and Vision Computing, Vol.26, No.10, Página(s) 1297-1313
- [3] R. Plamondon, S.N. Srihari,
2000, “On-Line and Off-Line Handwriting Recognition: A Comprehensive Survey”, IEEE Trans. On Pattern Analysis and Machine Intelligence, Vol.22, No.1, Página(s) 63-84
- [4] Hadar I, Avi-itzhak, Thanh A.Diep, Harry Garland
1995, “High Accuracy Optical Character Recognition Using Networks with Centroid Dithering”, IEEE Transaction on pattern analysis and machine intelligence, Vol 17, No.2, Página(s) 218-224
- [5] Anshu Sinha
1999, “An improved recognition module for the identification of handwritten digits”, MS Thesis, Massachusetts Institute of Technology.
- [6] Richard G. Casey, Erci Lecolinet
1996, “A Survey of Methods and Strategies in Character Segmentation”, IEEE Transactions on pattern analysis and machine intelligence, Vol 18, No. 7, Página(s) 690-705.
- [7] Peter Salavík and Venu Govindaraju.
2001, “Equivalence of Different Methods for Slant and Skew Corrections in Word Recognition Applications”, IEEE Transactions on pattern analysis and machine intelligence, Vol. 23, No. 3
- [8] Uchida, S.; Taira, E.; Sakoe, H.
2001, “Nonuniform Slant Correction Using Dynamic Programming”, Sixth International Conference on Document Analysis and Recognition, Página (s) 434-438
- [9] Michael D. Garris, Charles L. Wilson, James L. Blue
1998, “Neural Network-Based Systems for Handprint OCR Applications”, IEEE Transactions on pattern analysis and machine intelligence, Vol 7, No. 8, Página(s) 1097-1111.

- [10] Leticia María Seijas
2003, "Reconocimiento de dígitos manuscritos mediante redes neuronales: una técnica híbrida", Jornada Argentina de informática e investigación operativa, Departamento de computación- Facultad de Cs. Exactas y Naturales – Universidad de Buenos Aires
- [11] Georgios N. Sazaklis
1997, "Geometric methods for optical character recognition", State University of New York at Stony Brook
- [12] Fausett, Laurence
1994, "Fundamentals of neural networks: architectures, algorithms, and applications", Englewood Cliffs, NJ: Prentice-Hall.
- [13] Anil K. Jain, Jianchang Mao
1996, "Artificial Neural Networks: A Tutorial", IEEE Computer and System, Vol.29, No.3, Página(s) 31-44
- [14] M. A. Mohamed, P. Gader,
2000 "Generalized Hidden Markov Models – Part II: Application to Handwritten Word Recognition", *IEEE Trans. Fuzzy Systems*, Vol.8, No.1, Página(s) 82-95
- [15] Rafael palacios, Amar Gupta, Patrick S. P. Wang
2003, "Feedback-based architecture for reading courtesy amounts on checks", *Journal of Electronic Imaging*, Vol.12, No.1, Página(s) 194-202
- [16] Bailing Zhang, Minyue Fu, Hong Yan, and Marwan A. Jabri
1999, "Handwritten Digit Recognition by Adaptive-Subspace Self-Organizing Map (ASSOM)", *IEEE Transactions on Neural Networks*, Vol.10, No.4
- [17] Gonzalez, Rafael C.
2004, "Digital image processing using MATLAB".
Pearson Prentice Hall.
- [18] Otsu, N. A
1979, "Threshold Selection Method from Gray-Level Histogram". *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 9, No.1
- [19] P. Slavík, V. Govindaraju
2001, "Equivalence of Different Methods for Slant and Skew Corrections in Word Recognition Applications", *IEEE Transactions on pattern analysis and machine intelligence*, Vol. 23, No. 3, Página(s) 323-326

- [20] L. Lam, S. W. Lee, C. Y. Suen,
1992, "Thinning Methodologies- A Comprehensive Survey", IEEE Trans.
Pattern Recognition and Machine Intelligence, Vol.14, Pagina(s) 869-885
- [21] M.Blumenstein and B. Verma.
1998, "A Neural Based Segmentation and Recognition Technique for
Handwritten Words", IEEE World Congress on Computational Intelligence.
Vol 3, No.4, Página(s) 1738 - 1742
- [22] Jibu Punnoose.
1999, "An Improved Segmentation Module for Identification of Handwritten
Numerals", MS Thesis, Massachusetts Institute of Technology.
- [23] R. Plamondon, S. N. Srihari.
2000, "On-line and Off-line Handwriting Recognition: A Comprehensive
Survey", IEEE Transactions on pattern analysis and machine intelligence,
Vol. 22, No.1, Página(s) 63-84
- [24] González, Rafael C.
1996, Tratamiento digital de imágenes.
Pearson Prentice Hall.
- [25] Breuel, T.M.
2001, "Segmentation of handprinted letter strings using a dynamic
programming algorithm", Sixth International Conference on Document
Analysis and Recognition, Pagina(s) 821 – 826
- [26] Suzete E.N. Correia, Joao M. de Carvalho and Robert Sabourin.
2002, "On the Performance of Wavelets for Handwritten Numerals
Recognition". 16th International Conference on Pattern Recognition, Vol. 3,
Página(s) 127-130