

UNIVERSIDAD NACIONAL DE INGENIERÍA
FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA



**MÉTODO DE RECONOCIMIENTO DE PATRONES BASADO EN
PROCESAMIENTO DIGITAL DE IMÁGENES Y REDES NEURONALES
ORIENTADO AL REGISTRO AUTOMÁTICO DE EVALUACIONES
ACADÉMICAS**

TESIS

**PARA OPTAR EL GRADO ACADÉMICO DE MAESTRO EN CIENCIAS CON
MENCIÓN EN TELEMÁTICA**

ELABORADO POR

CHRISTIAN CARLOS DEL CARPIO DAMIÁN

ASESOR

Dr. GUILLERMO KEMPER VASQUEZ

LIMA – PERÚ

2015

DEDICATORIA

A mis queridos padres, por cultivar en mí los más nobles sentimientos.

A mi esposa Alicia, por su apoyo y comprensión.

A mi asesor, por brindarme su confianza y dedicación.

A todos los docentes, compañeros y amigos que me apoyaron en el desarrollo del presente trabajo, GRACIAS.

ÍNDICE DE CONTENIDO

	Página
INTRODUCCIÓN	1
CAPÍTULOS	
I ASPECTOS INTRODUCTORIOS	4
1.1 Antecedentes bibliográficos	4
1.2 Descripción de la realidad problemática	8
1.3 Formulación del problema	8
1.4 Justificación e importancia de la investigación	8
1.5 Objetivos	9
1.5.1 Objetivo general	9
1.5.2 Objetivos específicos	9
1.6 Hipótesis	10
1.6.1 Hipótesis principal	10
1.6.2 Hipótesis secundaria	10
1.7 Variables e indicadores	10
1.7.1 Variable independiente	10
1.7.2 Variable dependiente	10
1.7.3 Indicadores	10
1.8 Unidad de análisis	11
1.9 Tipo y nivel de investigación	11
1.10 Periodo de análisis	11
1.11 Fuentes de información e instrumentos utilizados	11
1.12 Técnicas de recolección y procesamiento de datos	11
II MARCO TEÓRICO	
2.1 Imagen digital	12
2.2 Histograma	12

2.3 Vecindad de un píxel	13
2.4 Modelos de color	14
2.4.1 Modelo RGB	14
2.5 Imagen binaria	15
2.6 Imagen en escala de grises	15
2.7 Imagen a color	16
2.8 Etiquetado	16
2.9 Distancia entre píxeles	23
2.9.1 Distancia euclidiana	23
2.10 Operaciones aritméticas lógicas entre píxeles	23
2.10.1 Negación	23
2.10.2 Adición	24
2.10.3 Substracción	24
2.11 Binarización de una imagen	25
2.12 Rotación de una imagen	25
2.13 Inclinación de una imagen	25
2.13.1 Estimación del ángulo de inclinación	26
2.13.2 Corrección del ángulo de inclinación	27
2.14 Operaciones morfológicas	27
2.14.1 Transformaciones morfológicas	28
2.14.1.1 Dilatación binaria	28
2.14.1.2 Erosión binaria	29
2.14.1.3 Apertura	30
2.14.1.4 Cierre	31
2.15 Esqueletización	31
2.16 Segmentación de imágenes	34
2.16.1 Segmentación basada en umbralización	34
2.17 Descriptores de imagen	34
2.17.1 Momentos	35
2.17.1.1 Área	35
2.17.1.2 Centro de gravedad	35
2.17.2 Coeficiente de correlación	36
2.18 Redes neuronales	36
2.18.1 Definición	37
2.18.2 Funciones de activación	38
2.18.3 Características de las redes neuronales	38

2.18.4 Métodos de aprendizaje	39
2.18.4.1 Aprendizaje supervisado	39
2.18.4.2 Aprendizaje no supervisado	39
2.18.5 Generalización	39
2.18.6 Perceptrón	40
2.18.7 Red perceptrón multicapa (MLP)	40
2.18.7.1 Algoritmo BACPROPAGATION	41
2.18.7.2 Mínimos locales y mínimos globales	45
2.18 Validación cruzada	46
III DESCRIPCIÓN DEL SISTEMA PROPUESTO	
3.1 Descripción del hardware	48
3.2 Descripción del software desarrollado	51
3.3 Adquisición de la imagen	53
3.4 Descripción del cuadernillo de evaluación	53
3.5 Selección de las áreas de procesamiento	56
3.6 Pre procesamiento	60
3.6.1 Pre procesamiento del código del alumno en marcas	60
3.6.2 Pre procesamiento de la nota del alumno en marcas	65
3.6.3 Pre procesamiento de la nota del alumno en números	70
3.7 Etapa de reconocimiento	77
3.7.1 Etapa de detección del código del alumno en marcas	78
3.7.2 Etapa de detección de la nota del alumno en marcas	84
3.7.3 Etapa de detección de nota del alumno en números	85
3.7.3.1 Detección mediante correlación	87
3.7.3.2 Detección mediante redes neuronales	89
IV ANÁLISIS Y PRESENTACIÓN DE RESULTADOS	
4.1 Descripción del escenario de las pruebas	92
4.2 Resultados de imágenes obtenidas	92
4.3 Métodos de validación empleados y resultados obtenidos	103
4.4 Tiempo de procesamiento	105
4.5 Software desarrollado	106
4.6 Costo del trabajo de tesis	109

CONCLUSIONES	110
RECOMENDACIONES PARA TRABAJOS FUTUROS	111
BIBLIOGRAFÍA	112

ÍNDICE DE FIGURAS

	Página
FIGURAS	
Figura 1.1 Sistema Feedback	4
Figura 1.2 Lectora Datawin 200s de OMRForms	5
Figura 2.1 Imagen digital representada mediante una matriz	12
Figura 2.2 Histograma de una imagen	13
Figura 2.3 Cubo de color RGB	15
Figura 2.4 Imagen binaria	15
Figura 2.5 Imagen a escala de grises	16
Figura 2.6 Imagen a color	16
Figura 2.7 Matriz binaria de 5x5	17
Figura 2.8 Los primeros diez pasos en el proceso de etiquetado	19
Figura 2.9 Los diez pasos siguientes en el proceso de etiquetado	21
Figura 2.10 Los últimos cinco pasos en el proceso de etiquetado	22
Figura 2.11 Resultado final del proceso de etiquetado	23
Figura 2.12 Operación NOT	23
Figura 2.13 Operación adición	24
Figura 2.14 Substracción de imágenes binarias	24
Figura 2.15 Número circunscrito por un rectángulo inclinado	26
Figura 2.16 Algunos elementos estructurales	28
Figura 2.17 Ejemplo de dilatación	29
Figura 2.18 Ejemplo de erosión	30
Figura 2.19 Ejemplo de operación de apertura	30
Figura 2.20 Ejemplo de operación de cierre	31
Figura 2.21 Un patrón en forma de T, y su respectivo esqueleto	32
Figura 2.22 El esqueleto de un rectángulo definido en términos de círculos bitangenciales	32
Figura 2.23 Imágenes a correlacionar	36
Figura 2.24 Esquema de una neurona biológica	37
Figura 2.25 Neurona artificial	37

Figura 2.26 Arquitectura del Perceptrón multicapa	41
Figura 2.27 Mínimos locales y Mínimos globales	45
Figura 2.28 Punto de parada basado en la regla de validación cruzada	47
Figura 3.1 Bloques de la etapa física	49
Figura 3.2 Fotos del recinto diseñado	49
Figura 3.3 Estructura mecánica del equipo	50
Figura 3.4 Diagrama de arreglo de LEDs	50
Figura 3.5 Diagrama de flujo del sistema general	52
Figura 3.6 Características de la cámara web utilizada	53
Figura 3.7 Cuadernillo de evaluación actual	54
Figura 3.8 Cuadernillo de evaluación diseñado	55
Figura 3.9 Cuadrícula para colocar el código del alumno en marcas	56
Figura 3.10 Áreas de interés del cuadernillo a procesar	57
Figura 3.11 Ejemplo del código de alumno	58
Figura 3.12 Cuadrícula para colocar la nota del alumno en marcas	58
Figura 3.13 Ejemplo de la nota de la alumno en “marcas”	59
Figura 3.14 Cuadrícula donde se escribe los números	59
Figura 3.15 Ejemplo del número manuscrito	60
Figura 3.16 Imagen en formato RGB del área “código del alumno en marca” y su imagen binarizada resultante	61
Figura 3.17 Imagen resultante del proceso de etiquetado	62
Figura 3.18 Componente “R” de la imagen del área código del alumno en marcas	62
Figura 3.19 Imagen I_{1RB} resultante	63
Figura 3.20 Proceso de obtención de las “Marcas” seleccionadas	63
Figura 3.21 Distancias de cada “marca”	65
Figura 3.22 Imagen en formato RGB del área “nota del alumno en marca” y su imagen binarizada resultante	66
Figura 3.23 Imagen resultante del proceso de dilatación	66
Figura 3.24 Imagen resultante	67
Figura 3.25 Componente “R” de la imagen del área nota del alumno en marcas	67
Figura 3.26 Imagen MS2 resultante	68
Figura 3.27 Proceso de obtención de las marcas seleccionadas	68
Figura 3.28 Distancias de cada marca	69
Figura 3.29 a) Imagen RGB, b) Componente R, c) Componente G, d) componente B	71

Figura 3.30 a) Componente G, b) Imagen binarizada obtenida de la componente G	71
Figura 3.31 Imagen binarizada del área donde se encuentran los números	72
Figura 3.32 Imagen binarizada del área donde se encuentran los números	72
Figura 3.33 Extracción de los números a procesar	73
Figura 3.34 a) Imagen sin rotar, b) Imagen rotada	74
Figura 3.35 Imagen de la derecha con corrección de inclinación	74
Figura 3.36 b) Imagen dilatada, c) máscara que se utilizó para el proceso de dilatación	75
Figura 3.37 a) Imagen original, b) imagen adelgazada. C) Imagen resultante de la dilatación de b)	76
Figura 3.38 a) Imagen NF, b) imagen obtenida	76
Figura 3.39 Ejemplo de reconocimiento RN1	80
Figura 3.40 Diagrama de flujo de obtención del código del alumno	81
Figura 3.41 (continuación) Diagrama de flujo de obtención del código del alumno	82
Figura 3.42 (continuación) Diagrama de flujo de obtención del código del alumno	83
Figura 3.43 Diagrama de flujo de reconocimiento de la nota en marcas	84
Figura 3.44 Ejemplo de reconocimiento RN2	85
Figura 3.45 Número de época vs. Error cuadrático medio	86
Figura 3.46 Gráfica de regresión de cada conjunto de datos	87
Figura 3.47 Imagen referencial de cada dígito de dimensión 7x5	88
Figura 3.48 Diagrama de flujo del pre procesamiento de los números	88
Figura 3.49 Ejemplo de reconocimiento del número manuscrito	90
Figura 3.50 Diagrama de bloques de reconocimiento del número manuscrito	91
Figura 4.1 Ejemplo 1 de resultado de distancia obtenida del código	93
Figura 4.2 Ejemplo 2 de resultado de distancia obtenida del código	94
Figura 4.3 Ejemplo 1 de resultado de distancia obtenida de la nota	95
Figura 4.4 Ejemplo 2 de resultado de distancia obtenida de la nota	96
Figura 4.5 Números escritos por el docente	98
Figura 4.6 (continuación) Números escritos por el docente	99
Figura 4.7 Resultados obtenidos para el proceso de normalización de grosor	100
Figura 4.8 (continuación) Resultados obtenidos para el proceso de normalización de grosor	101
Figura 4.9 Resultados de dígito obtenido de 7x5	102

Figura 4.10 Plantilla de número (7x5) para el proceso de correlación	103
Figura 4.11 Interfaz visual del software desarrollado	106
Figura 4.12 Interfaz visual de selección de áreas	107
Figura 4.13 Pantalla de resultados obtenidos	108

ÍNDICE DE TABLAS

	Página
Tabla 4.1 Resultados de reconocimiento de los números manuscritos mediante el uso de redes neuronales y correlación	104
Tabla 4.2 Resultados de reconocimiento de los códigos y notas mediante marcas haciendo uso de redes neuronales	105
Tabla 4.3 Matriz de confusión del reconocimiento de los números manuscritos mediante la RN	105
Tabla 4.4 Materiales para la tesis y costos	109

RESUMEN

En el presente trabajo se desarrolla un método de reconocimiento de patrones basado en procesamiento digital de imágenes y redes neuronales orientadas al registro automático de evaluaciones académicas. El sistema está conformado por una parte de hardware y una parte de software. La etapa de hardware consiste de un recinto que contiene los cuadernillos de evaluación a registrar. Dicho recinto contiene una bandeja de alimentación donde se colocan los cuadernillos. El recinto cuenta con una adecuada iluminación provista de LEDs (Light - Emitting Diode) de alta luminosidad. Una vez obtenida la imagen, el cuadernillo es retirado del compartimiento hacia una bandeja de salida.

La etapa de software inicialmente adquiere la imagen de un cuadernillo académico posicionado en el compartimiento del recinto a través de una cámara web. Luego se identifican en el cuadernillo tres áreas de interés, que él usuario inicialmente configuró, estas son: código del alumno en marcas, nota del alumno en números y nota del alumno en marcas. Mediante el uso de algoritmos de procesamiento de imágenes las áreas de interés son filtradas, umbralizadas, segmentadas y parametrizadas. Luego haciendo uso de algoritmos de redes neuronales se lleva a cabo el reconocimiento del código del alumno y de la nota expresada en marcas. Asimismo, se reconoce la nota del profesor expresada en números escritos manualmente. El sistema propuesto responde a la necesidad de incrementar la eficiencia del proceso de registro de notas del departamento de coordinación académica de la Facultad de Ingeniería y Arquitectura de la Universidad de San Martín de Porres.

ABSTRACT

In this paper a method of pattern recognition based on digital image processing and neural networks oriented to automatic registration of academic assessments is developed. The system consists of a hardware stage and a software stage. The hardware stage consists of an enclosure containing the assessment booklets to register. Said chamber contains a feed tray where the assessment booklets are placed. The enclosure has an adequate lighting provided with high brightness LEDs (Light - Emitting Diode). Once the image has been obtained, the booklet is removed from the bin onto an output tray.

The software stage initially acquires the image of an assessment booklet, positioned in the compartment enclosure, via a webcam. Then, three areas of interest that the user has initially configured are identified in the booklet, these are: the student's code, expressed in marks, the student's grade, expressed in numbers, and the student's grade, expressed in marks. By using image processing algorithms, these areas of interest are filtered, thresholded, segmented and parameterized. Then using neural network algorithms is carried out the recognition of the student's code and the student's grade, expressed in marks. Likewise, the student's grade in handwritten numbers is recognized. The proposed system answers to the need to increase the efficiency of the grades registration process in the academic coordination department of the Faculty of Engineering and Architecture at the University of San Martin de Porres.

INTRODUCCIÓN

En una institución académica que alberga gran cantidad de alumnado, el registro de notas, de las distintas evaluaciones de los cursos que se llevan durante el ciclo académico, resulta complejo y demanda mucho tiempo.

A diferencia de los múltiples sistemas que existen en el mercado sobre un corrector automático de evaluaciones de opción múltiple y posterior registro de estas evaluaciones, el sistema propuesto no corrige los exámenes, estos son corregidos por el docente del curso.

Se han realizado modificaciones al cuadernillo de evaluaciones, pero se ha tratado que sea lo mínimo posible. Adicionalmente, se realiza la detección de marcas ópticas para poder obtener el código del alumno y registro de su nota también, lo cual representa una segunda validación de manera que pueda ser confiable el sistema propuesto. Una vez obtenido, el código y nota del estudiante estos son almacenados en un archivo excel.

Asimismo, no se utiliza un lector óptico de "marcas" para la lectura de estas, sino una cámara WEB que captura la imagen del cuadernillo para procesarla y procesar los datos del alumno y su respectiva nota.

Para el desarrollo de esta tesis se utilizan o emplean técnicas de procesamiento digital de imágenes y redes neuronales para realizar el pre procesamiento, obtención de los descriptores y posterior reconocimiento de los datos que se obtienen. El trabajo se centra en el reconocimiento de los números manuscritos y adicionalmente, se realiza también el reconocimiento de las marcas ópticas.

El sistema tiene la capacidad de reconocer números manuscritos que se escriben sobre una cuadrícula acondicionada para tal fin. Los números deben ser escritos con tinta líquida o seca de color rojo, azul o negro para un adecuado reconocimiento.

Los dígitos deben ser escritos de manera continua sin presentar discontinuidades, es decir, el dígito debe formar un solo objeto.

En el caso de las marcas ópticas, estas deben ser llenadas correctamente, es decir, se debe sombrear toda la marca dentro del área correspondiente. Pueden ser llenados con tinta líquida o seca de color rojo, azul y negro. En el caso de la cuadrícula para el llenado del código, solo debe existir una marca por columna. Y en el caso de la cuadrícula para el llenado de la nota, solo una marca.

Tanto para la cuadrícula de los números manuscritos como en la cuadrícula de las marcas ópticas, en el caso de equivocarse se puede utilizar un corrector ortográfico, teniendo en cuenta lo mencionado anteriormente.

La tasa de error máxima que se espera alcanzar es igual o menor al 5% para el caso del reconocimiento de los dígitos manuscritos y menor o igual al 5% para el caso de reconocimiento de las marcas ópticas. Se estima que el tiempo procesamiento para cada cuadernillo será de tres (03) segundos.

Para la captura de las imágenes se está utilizando una webcam para probar el método desarrollado en el peor caso. Se estima que con una cámara profesional para la adquisición de imágenes la tasa de error mejorara.

Como resultado del trabajo, se espera una tasa de acierto superior al 95% en el reconocimiento de los números manuscritos, esto hace que el sistema sea confiable, sobre todo si esta aplicación se va a utilizar en instituciones educativas para el manejo de sus notas de evaluaciones.

En el capítulo I se presentan los antecedentes y descripción del problema, en el capítulo II, se explican los diversos conceptos y técnicas empleadas en el procesamiento digital de imágenes, así como también, el uso de redes neuronales artificiales; para que el lector pueda tener un conocimiento de lo que se ha aplicado en el proyecto, lo cual se verá acondicionado al proyecto en los capítulos III y IV.

En el Capítulo III se describe el sistema propuesto, detallando las etapas que lo componen para que el lector aprecie como se aplica la teoría y como el método propuesto constituye un aporte científico través de una tesis de maestría.

Finalmente en el capítulo IV se presentan los resultados y los métodos de validación empleados, basados en el sistema descrito en el capítulo III; se describe también, el software de simulación que se implementó.

Después del último capítulo se consignan las conclusiones obtenidas del trabajo de investigación, se presentan las recomendaciones de trabajos futuros y las fuentes de información empleadas.

CAPÍTULO I ANTECEDENTES Y DESCRIPCIÓN DEL PROBLEMA

1.1 Antecedentes bibliográficos

En la actualidad, no existe un sistema similar al que se propone, los que se conocen o emplean son sistemas de corrección automáticos de evaluaciones de opción múltiple y posterior registro de evaluaciones. Los formularios de evaluación son adquiridos a través de un lector óptico o escáner de papel. Así, se tiene productos en el mercado como Feedback [10], que se muestra en la Figura 1.1, sistema que a través de un sensor óptico lee las fichas y realiza la corrección y almacenamiento automático de las notas; también se dispone de la lectora DataWin 200S (Figura 1.2), Omrforms, Sekonic SR 11000 OMR Scanner [21], y muchos productos similares, que leen fichas ópticas.

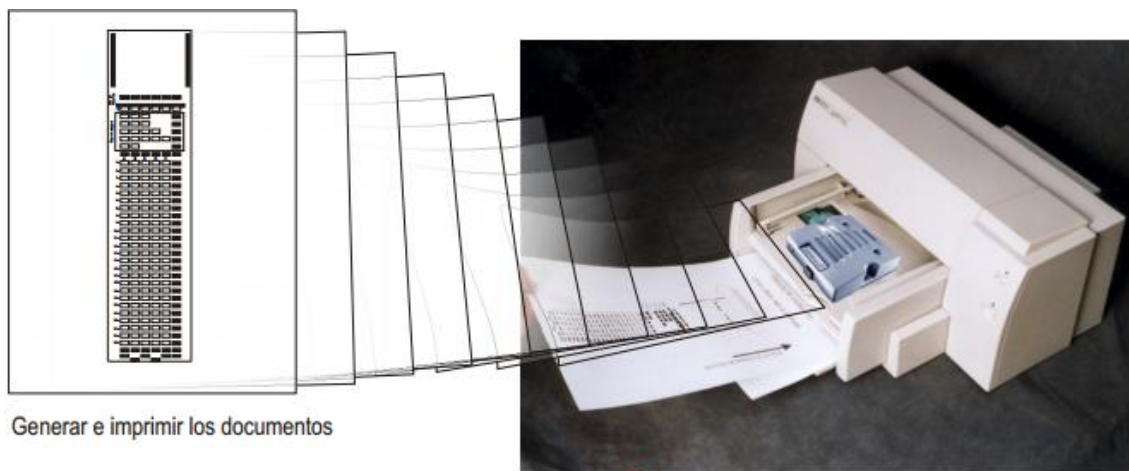


Figura 1.1 Sistema Feedback



Figura 1.2 Lectora Datawin 200s de OMRForms

Todos estos productos utilizan lectores ópticos de marcas (Optical Mark Reader, OMR por sus siglas en ingles). Un lector de marcas cuenta con pequeñas células fotoeléctricas que emiten luz a través de un canal de fibra óptica que detecta si en una posición determinada de un impreso hay una marca o no. La sensibilidad puede llegar a detectar una simple mota de polvo. Estos lectores son rápidos pero la gran desventaja es que solo sirve para exámenes de opción múltiple.

Por otro lado, en el campo del reconocimiento de números manuscritos existen diversos estudios y aplicaciones, entre ellos se hallan:

Según F. Kimura y M. Shridhar en su publicación en la Sociedad de Reconocimiento de Patrones, titulada “Handwritten numerical recognition based on multiple algorithms”, [16] comentan sobre el reconocimiento de números escritos a mano mediante la mezcla de dos algoritmos. Uno de los algoritmos emplea una función discriminante cuadrática modificada, en la que se utiliza la dirección de las características espaciales del número, y el segundo algoritmo utiliza las características derivativas del perfil del número, logrando tasas de rechazo por debajo del 4%. Esta tasa va a depender mucho del hardware utilizado para la captura de los datos así como la implementación adecuada de los algoritmos propuestos.

K. Batuwita y G. Bandara, en su artículo del Instituto de Ingeniería Eléctrica y Electrónica, (IEEE por sus siglas en inglés) titulado “Fuzzy Recognition of Offline Handwritten Numeric Characters” [1] proponen el reconocimiento de números escritos a mano en imágenes digitales utilizando lógica difusa, debido a que con un sistema difuso

obtienen flexibilidad, eficiencia y capacidad de adaptación para el reconocimiento de aquellos. Las imágenes son adquiridas con un escáner de papel. El porcentaje de acierto de este método varía, para algunos números el acierto va hasta el 50% como para otros hasta el 100%. Justamente esta tasa de acierto variable, es la desventaja de dicho método.

M. Hanmandlu, K. Muraly y S. Chakraborty, en su artículo IEEE “Fuzzy Logic Based Handwritten Character Recognition”, [13] presentan métodos de reconocimiento de números manuscritos utilizando redes neuronales y lógica difusa, y realizan una comparación entre ambas. Para el reconocimiento del número manuscrito dividen el número en partes más pequeñas y extraen características de estas y mediante el uso de lógica difusa y redes neuronales realizan la detección. Para el entrenamiento y validez de su estudio, emplean una base de datos pre existente de IEEE [15]. La tasa de aceptación varía entre el 90% y 100% sobre estos números almacenados. La desventaja que presenta este estudio es que el algoritmo de adelgazamiento que se utiliza para el pre procesamiento del número no es muy eficiente.

J. Wang, J. Yang, S. Li, Q Dai y Jianxing Xie, en su artículo IEEE “Number Image Recognition Based on Neural Network Ensemble”, [24] presentan un estudio que efectúa el reconocimiento de números manuscritos en imágenes, en el que se emplean un conjunto de redes neuronales, mediante la implementación del “*Bagging algorithm*”. El resultado de este estudio arroja un acierto de reconocimiento del 93.64%. Algunos números como el 6, 8 y 9 presentan error de reconocimiento por su similar forma y tamaño.

E. Valveny y A. López, en su artículo “Numeral recognition for quality control of surgical sachets” [23] describen una aplicación de técnicas de OCR (Optical Character Recognition) para el control de calidad en la producción industrial. El propósito del sistema es verificar la correcta impresión numérica de la información en sobres con material quirúrgico. Los números se imprimen en una superficie de aluminio, cubierto por una película de plástico transparente, que puede producir sombras o reflejos en la imagen. Las principales dificultades para el reconocimiento de caracteres surgen de la baja resolución de adquisición, el ruido, la impresión pesada o ligera, y los diferentes patrones de impresión. El sistema debe operar con una tasa de error inferior a 0,1% y con el mínimo de tiempo de cálculo.

M. Gumah, E. Schneider, and A. Aburas, en su publicación IEEE "Handwriting Recognition System Using Fast Wavelets Transform" [12], proponen el reconocimiento óptico de caracteres usando la transformada rápida de Wavelets (FWT) para parametrizar las características de las imágenes que se utilizarán directamente para reconocer. En este caso, los tipos de caracteres que se utilizan son árabes. Con el uso de la técnica propuesta, se logró una precisión del 94.18%, en promedio.

Xiang Fang y A. T. Alouani, en su publicación IEEE "Unconstrained handwritten numeral recognition using fuzzy rule-based classifier" [8] proponen el reconocimiento de números escritos a mano basado en reglas Fuzzy, analizando el esqueleto del numeral escrito a mano. En una imagen esqueletizada, la información de los puntos finales, puntos de unión y ramas posibilitan identificar los números escritos a mano. Las reglas Fuzzy hacen hincapié en la estructura de las ramas del esqueleto, pero no en la misma rama individual. De esta manera, las estructuras significativas se pueden mantener. Una forma difusa se define con tres parámetros: la orientación, la curvatura y peso longitud. Cada parámetro se ha definido por un conjunto difuso. Asimismo, en ese trabajo, no se mencionan tasas de acierto, solo indica que respondió satisfactoriamente.

También, en el artículo IEEE "A Comparative Study of Fourier Descriptors and Hu's Seven Moment Invariants for Image Recognition", [5] se realiza una evaluación y comparación entre los descriptores de Fourier y los 7 momentos invariantes de Hu, para el reconocimiento de caracteres a base de imágenes con diferentes resoluciones espaciales. Los descriptores de Fourier y los 7 momentos invariantes de Hu tienen la propiedad de invariancia sobre las transformaciones de la imagen, incluyendo el cambio de escala, traslación y rotación. Sin embargo, existen umbrales de resolución espacial para ambos. En este estudio, el motor de reconocimiento de imágenes, para los descriptores de Fourier, con vectores de características compuestas por los primeros 10 elementos de la serie, la resolución espacial no debe ser inferior a 64×64 para lograr el reconocimiento del 100%. En el caso de los siete Momentos Invariantes de Hu, la resolución espacial mínima debe de ser 128×128 . Así mismo, cabe mencionar que esta comparación se utilizó en números impresos y no manuscritos.

1.2 Descripción de la realidad problemática

En instituciones académicas que albergan gran cantidad de alumnado, el registro de notas en el sistema, que se realiza de manera manual, de las distintas evaluaciones de los cursos que se llevan a cabo durante el ciclo académico, resulta monótono y demanda mucho tiempo. Esta tarea resulta aún más preocupante si dichos cursos son masivos y tener que destinar personal por la gran cantidad de evaluaciones por registrar y que son propensos a cometer errores al realizar dicha tarea por el cansancio visual.

En la Facultad de Ingeniería y Arquitectura (FIA), de la Universidad de San Martín de Porres (USMP), por ejemplo en el ciclo académico 2014 - 01 se registraron más de 21000 evaluaciones, solo en cursos de evaluación continua (estudios generales); y para dicha tarea la facultad destina a una secretaria que es la que se encarga de ingresar al sistema las notas.

1.3 Formulación del problema

Hay que tener cuenta, que los datos que se están registrando son notas de evaluaciones, algo con lo cual se debe de tener mucho cuidado, y que debido a la gran cantidad de datos a registrar se pueden cometer errores por cansancio y esto provocaría reclamos por parte de los alumnos. Asimismo, como las notas la registra una secretaria y no un coordinador académico las notas podrían ser propensas a manipulaciones.

1.4 Justificación e importancia de la investigación

Es un problema planteado por el Departamento Académico de la Facultad de Ingeniería y Arquitectura (FIA) de la Universidad de San Martín de Porres (USMP). Se justifica porque es una solución innovadora que permitirá automatizar el proceso de registro de notas en el sistema de notas de la Facultad de Ingeniería y Arquitectura de la USMP, así mismo este sistema se podría también implementar en las distintas facultades de la universidad y siendo más ambiciosos se podría dar en cualquier institución educativa del país.

Porque las soluciones actuales comerciales que utilizan el método OMR (Lectura Óptica de Marcas) solo permiten el registro de evaluaciones que contengan preguntas de elección. Los estudiantes marcan sus respuestas, u otra información personal, llenando los círculos marcados en una hoja pre-impresa.

Se justifica, porque el costo estimado de la implementación del sistema es de aproximadamente el 20% de un equipo comercial.

Porque aplica conocimientos adquiridos en el estudio de la maestría y así mismo la solución propuesta amerita definir un método basado en el estado del arte pero optimizado y acondicionado a las necesidades de instituciones educativas que todavía registran de manera manual las evaluaciones a través de cuadernillos.

De acuerdo a los artículos revisados en los antecedentes de solución, las técnicas más utilizadas para el reconocimiento de números manuscritos son el procesamiento de imágenes y el uso de inteligencia artificial con énfasis en las redes neuronales. Debido a esto es que en el presente trabajo se hace uso de dichas técnicas.

Porque esto permitirá automatizar el proceso de registro de evaluaciones, y el personal que se dedicaba a dicha tarea puede ser derivado a otras tareas, optimizando de esta manera el tiempo de trabajo del personal.

1.5 Objetivos

1.5.1 Objetivo general

Proponer y desarrollar un método robusto y confiable que permita el reconocimiento de números manuscritos en evaluaciones escritas, basado en procesamiento de imágenes e inteligencia artificial.

1.5.2 Objetivos específicos

- Diseñar una nueva carátula de cuadernillo de evaluación de acuerdo con los requerimientos del sistema, procurando realizar mínimas modificaciones en el cuadernillo de evaluación existente.
- Entrevistarse con el responsable de registros académicos de la Facultad de Ingeniería y Arquitectura de la Universidad de San Martín de Porres para conocer la problemática de las necesidades.
- Implementar algoritmos de procesamiento digital de imágenes para limpiar, esqueletizar, segmentar la imagen, y que sean computacionalmente eficientes.
- Reconocer la nota del estudiante en números manuscritos con una tasa de acierto por encima del 95%.

- Reconocer el código y nota del alumno en marcas con una tasa de acierto por encima del 98%.
- Desarrollar de una interfaz visual en MATLAB que permita el fácil manejo de la aplicación.
- Reconocer los números manuscritos mediante el uso de redes neuronales y correlación.

1.6 Hipótesis

1.6.1 Hipótesis Principal:

Si se aplica el método basado en procesamiento digital de imágenes y redes neuronales para el reconocimiento de números manuscritos, entonces será posible mejorar la eficiencia en términos de tasa de error de reconocimiento.

1.6.2 Hipótesis Secundaria:

"La aplicación eficiente de técnicas de procesamiento digital de imágenes y redes neuronales permitirá la automatización del proceso de registro de evaluaciones académicas".

1.7 Variables e indicadores

1.7.1 Variable independiente

La aplicación del método de reconocimiento de números manuscritos basado en procesamiento digital de imágenes y redes neuronales.

1.7.2 Variable dependiente

Tasa de verdadero reconocimiento del número manuscrito.

1.7.3 Indicadores

Tasa de error obtenida mediante la medición del número de imágenes reconocidas de manera correcta dividida entre el número total de imágenes que ingresan al sistema.

Matriz de confusión obtenida para el reconocimiento de los números manuscritos mediante el uso de redes neuronales.

1.8 Unidad de análisis

El número manuscrito.

1.9 Tipo y nivel de investigación

Es una investigación de tipo experimental, debido a que el método de reconocimiento propuesto se está validando a través distintas muestras (números manuscritos, escritos por distintos docentes) y el sistema los está reconociendo adecuadamente teniendo una tasa de error muy baja, comprobando así el funcionamiento de la solución propuesta.

Así mismo, el nivel de investigación es el de una maestría en ciencias debido a que se están aplicando conocimientos adquiridos en la maestría como es el procesamiento digital de señales y redes neuronales.

1.10 Periodo de análisis

Los datos se obtuvieron durante los semestres académicos 2014 - 01 y 2014 - 02 en la Facultad de Ingeniería y Arquitectura de la Universidad de San Martín de Porres.

1.11 Fuentes de información e instrumentos utilizados

Las fuentes de información son artículos IEEE, libros relacionados a los temas de procesamiento digital de imágenes y redes neuronales. Todos estos están indicados en las fuentes bibliográficas.

1.12 Técnicas de recolección y procesamiento de datos

Los datos (números manuscritos) se obtuvieron a través de una ficha que se diseñó, en donde los docentes escribían la nota y así mismo rellenaban las marcas.

El procesamiento de los datos se realizó a través del software que se diseñó en donde se implementó el método propuesto de reconocimiento de numerales y marcas.

CAPÍTULO II MARCO TEÓRICO

2.1 Imagen digital

Una imagen es representada de forma digital como una matriz de $M \times N$ elementos, siendo el primero, el número de filas y el segundo el número de columnas. Cada uno de los elementos en la matriz $f[x][y]$ representa un píxel. Cada píxel tiene un valor numérico asociado que representa el valor de intensidad de la imagen. Siendo el píxel la menor unidad homogénea que forma una imagen digital.

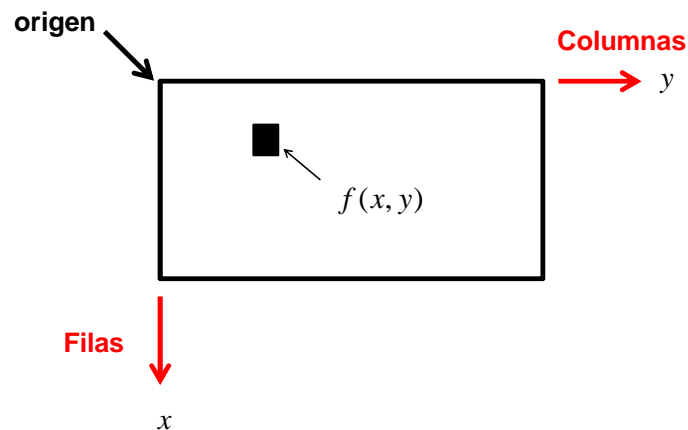


Figura 2.1 Imagen digital representada mediante una matriz

2.2 Histograma

El histograma de una imagen es el conteo o estadística de los valores de píxeles que conforman una determinada imagen [11]. En el caso del histograma de una imagen de niveles de grises, se define como una gráfica, en que a cada valor posible de nivel de gris entre 0 y 255, se le asigna la proporción de píxeles en la imagen con dicho valor.

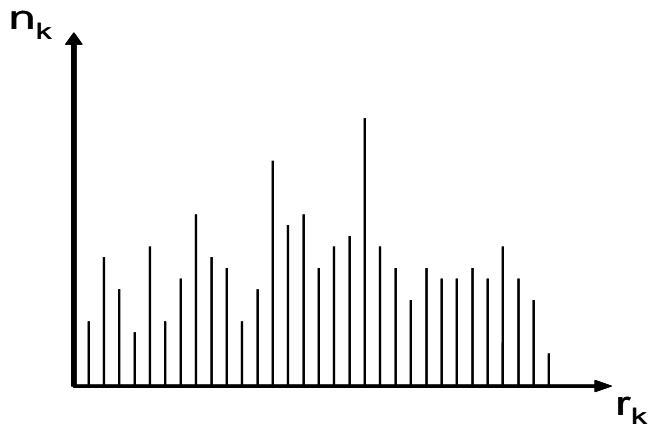


Figura 2.2 Histograma de una imagen

$$h[r_k] = n_k \quad (2.1)$$

Donde:

r_k es el nivel de gris

n_k es el número de pixeles de la imagen que tiene valor k

2.3 Vecindad de un píxel

Sea $I(x_0, y_0)$ un píxel que pertenece a una imagen $I(x, y)$, (donde “x” representa a las filas y “y” a las columnas), que tiene cuatro vecinos horizontales y verticales. Estos son denominados los 4 vecinos del píxel $I(x_0, y_0)$. Sus coordenadas son:

$$\{I(x_0 - 1, y_0), I(x_0, y_0 - 1), I(x_0, y_0 + 1), I(x_0 + 1, y_0)\} \quad (2.2)$$

También se tiene a los cuatro vecinos en diagonal del píxel $I(x_0, y_0)$. Sus coordenadas son las siguientes:

$$\{I(x_0 - 1, y_0 - 1), I(x_0 - 1, y_0 + 1), I(x_0 + 1, y_0 - 1), I(x_0 + 1, y_0 + 1)\} \quad (2.3)$$

Estos puntos junto a los cuatro vecinos de píxel $I(x_0, y_0)$ se denominan los ocho vecinos del píxel $I(x_0, y_0)$. Algunos de estos 8 vecinos caen fuera de la imagen si (x_0, y_0) está en el borde de la misma [11].

2.4 Modelos de color

El propósito de un modelo de color es facilitar la especificación de colores en alguna forma estándar. En esencia, un modelo de color es una especificación de un sistema coordinado de tres dimensiones y de un sub espacio dentro de tal sistema, donde cada color se representa por un punto.

Los modelos de color que se emplean frecuentemente en el procesamiento digital de imágenes son el RGB, el YIQ, el HSI y el YCbCr. [11]. En el presente trabajo se utiliza el modelo de color RGB.

2.4.1 Modelo RGB

El modelo de color RGB es dependiente del dispositivo y suele ser usado por las cámaras de video, monitores de televisión y computadoras.

No es necesaria la unión de todas las longitudes del espectro visible para obtener el blanco, ya que si mezclamos solo rojo, verde y azul obtendremos el mismo resultado. Es por esto, que son denominados colores primarios, porque la suma de los tres produce el color blanco. Todos los colores del espectro pueden ser obtenidos a partir de ellos.

Todos los colores que se visualizan en el monitor están en función de las cantidades de rojo, verde y azul. Por ello, para representar un color, en el modelo RGB, se le asigna un valor entre 0 y 255 (notación decimal) o entre 00 y FF (notación hexadecimal) para cada uno de los componentes rojo, verde y azul que lo forman. Los valores más altos de RGB corresponden a una cantidad mayor de luz blanca. Por consiguiente, cuanto más alto, en conjunto, son los valores de R, G, y B, más claros son los colores.

De esta forma, un color cualquiera vendrá representado, en el sistema RGB, mediante la sintaxis decimal (R, G, B) o mediante la sintaxis hexadecimal #RRGGBB. El color rojo puro, por ejemplo, se especificará como (255,0,0) en notación RGB decimal y #FF0000 en notación RGB hexadecimal, mientras que el color rosa claro, dado en notación decimal por (252,165,253) se corresponde con el color hexadecimal #FCA5FD.

El modelo de color RGB se puede representar por un sistema de coordenadas cartesianas como lo muestra la Figura 2.3.

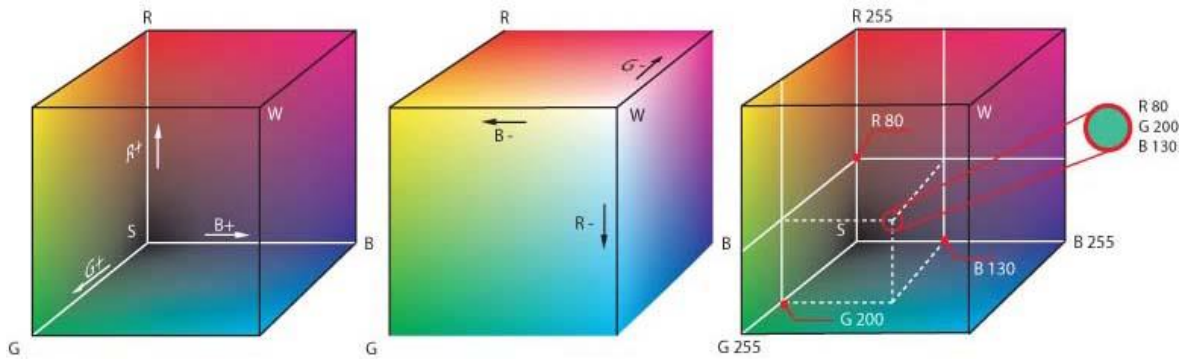


Figura 2.3 Cubo de color RGB

2.5 Imagen binaria

En una imagen, en que solo se permiten dos niveles de intensidad, (normalmente blanco y negro) se dice que son imágenes binarias. Esto se muestra en la Figura 2.4



Figura 2.4 Imagen binaria

2.6 Imagen en escala de grises

En una imagen en escala de grises, cada punto de la imagen se almacena en un byte, donde su valor numérico representa su tono, que puede oscilar entre el blanco (255) y el negro (0). Esto quiere decir que es una imagen donde existen 256 tonos de gris (de 0 a 255, ambos inclusive). Es decir, la profundidad de color es el número de bits que define a cada píxel, que determinan el máximo número de colores que pueden tener. Esto se muestra en la Figura 2.5.

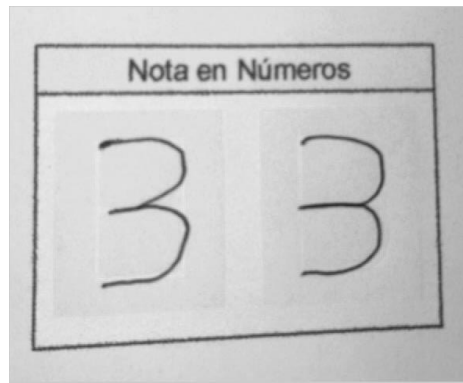


Figura 2.5 Imagen a escala de grises

2.7 Imagen a color

Una imagen a color, como se muestra en la Figura 2.6, donde se usan 256 niveles para representar la intensidad de cada uno de los tres colores primarios, de esta manera, se obtiene más de 16 millones de colores ($256 \times 256 \times 256$).



Figura 2.6 Imagen a color

2.8 Etiquetado

El etiquetado, es un proceso el cual tiene como objetivo, en una imagen binaria, la asignación de una etiqueta a todos los píxeles que sean vecinos.

Teniendo en cuenta que una imagen binaria está compuesta por unos y ceros y que el valor del píxel de interés es igual a "1", se procede al siguiente procesamiento.

Supóngase que se hiciera un barrido de la imagen píxel a píxel, de izquierda a derecha y de arriba abajo. Sea "p" el píxel en cada paso del proceso de barrido. La secuencia de barrido asegura que cuando se esté en el píxel "p" de coordenadas (x, y), los píxeles "r" de coordenadas (x, y - 1), "s" de coordenadas (x - 1, y + 1), "t" de coordenadas (x - 1, y) y "u" de coordenadas (x - 1, y - 1) ya han sido encontrados y etiquetados si fuesen unos.

De lo anteriormente descrito, se considera el procedimiento siguiente: si el valor de “p” es 0, simplemente se continúa hasta la siguiente posición de barrido. Si el valor de “p” es 1, se examinan los píxeles “r”, “s”, “t” y “u”. Si todos estos son 0, se asigna una nueva etiqueta a “p”. Si solo uno de los píxeles es 1, se asigna su etiqueta a “p”. Si todos los píxeles son 1 y tienen la misma etiqueta, se asigna esta etiqueta a “p”. Si todos son 1 y tienen etiquetas diferentes se asigna a “p” la etiqueta más antigua, y se hace referencia que esta es equivalente a las otras tres, es decir, los píxeles “r”, “s”, “t” y “u” están conectados a través de “p”.

Si todos los píxeles son 1 y solo algunos de estos tuvieran etiquetas iguales y el resto diferentes, se asignaría a “p” la etiqueta más antigua y se hace la referencia que esta etiqueta es equivalente a las demás de ese entorno. Para el caso en el que solo 2 o 3 píxeles sean 1 y tengas etiquetas iguales o diferentes, igualmente se asignaría a “p” la etiqueta más antigua y se hace la referencia de etiquetas equivalentes. Al final del barrido, todos los píxeles con valor igual a 1 han sido etiquetados, aunque algunas de estas etiquetas puedan ser equivalentes. Todo lo que se tiene que hacer ahora es clasificar todos los pares de etiquetas equivalentes y agruparlas en una sola, por ejemplo, si las etiquetas de “r” y de “t” son equivalentes con la etiqueta de “p”, los píxeles “r” y “t” toman el valor de la etiqueta de “p”, de esta manera, se asigna una sola etiqueta para los pares de etiquetas que sean equivalentes.

Para ilustrar mejor este proceso, se tomará como ejemplo una matriz binaria de 5 x 5 como se muestra en la Figura 2.7.

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

Figura 2.7 Matriz binaria de 5x5

Ahora se procederá a recorrer la matriz y realizar el proceso de etiquetado como se indica líneas arriba.

Como la matriz es de 5 x 5, entonces se realizará 25 iteraciones. En la Figura 2.8 se muestran los primeros 10 pasos. En la Figura 2.10 se muestran los 10 pasos siguientes. Y en la Figura 2.11 se muestran los últimos 5 pasos.

Las etiquetas que se asignarán tendrán el valor de "A" en delante.

Los píxeles que están dentro de la máscara de color rojo ya han sido evaluados y etiquetados si fuese el caso; el círculo azul indica el píxel que se está evaluando.

A continuación se describe cada paso.

Paso 1: el primer píxel a evaluar tiene el valor de "0". Por lo tanto se pasa al siguiente píxel.

Paso 2: el segundo píxel a evaluar tiene valor de "1". Los píxeles anteriormente evaluados (que se encuentran dentro de la máscara) no tienen etiquetas. Por lo tanto se le asigna la etiqueta "A".

Paso 3: el tercer píxel a evaluar tiene valor de "0". Por lo tanto se pasa al siguiente píxel.

Paso 4: el cuarto píxel a evaluar tiene valor de "1". Los píxeles anteriormente evaluados no tienen etiquetas. Por lo tanto se le asigna la etiqueta "B".

Paso 5: el quinto píxel a evaluar tiene valor de "1". Un píxel anteriormente evaluado tiene el valor de "B". Por lo tanto se le asigna a este píxel la etiqueta "B".

Paso 6: el sexto píxel a evaluar tiene el valor de "0". Por lo tanto se pasa al siguiente píxel.

Paso 7: el séptimo píxel a evaluar tiene valor de "1". Un píxel anteriormente evaluado tiene el valor de "A". Por lo tanto se le asigna a este píxel la etiqueta "A".

Paso 8: el octavo píxel a evaluar tiene valor de "1". Tres píxeles anteriormente evaluados han sido etiquetados. Dos con el valor de "A" y uno con el valor de "B". Por lo tanto se le asigna a este píxel la etiqueta más antigua, que es "A". Así mismo se hace referencia que la etiqueta "A" es equivalente a la etiqueta "B".

Paso 9: el noveno píxel a evaluar tiene valor de "1". Tres píxeles anteriormente evaluados han sido etiquetados. Dos con el valor de "B" y uno con el valor de "A". Por lo tanto se le asigna a este píxel la etiqueta más antigua, que es "A". Así mismo se hace referencia que la etiqueta "B" es equivalente a la etiqueta "A".

Paso 10: el décimo píxel a evaluar tiene el valor de "0". Por lo tanto se pasa al siguiente píxel.

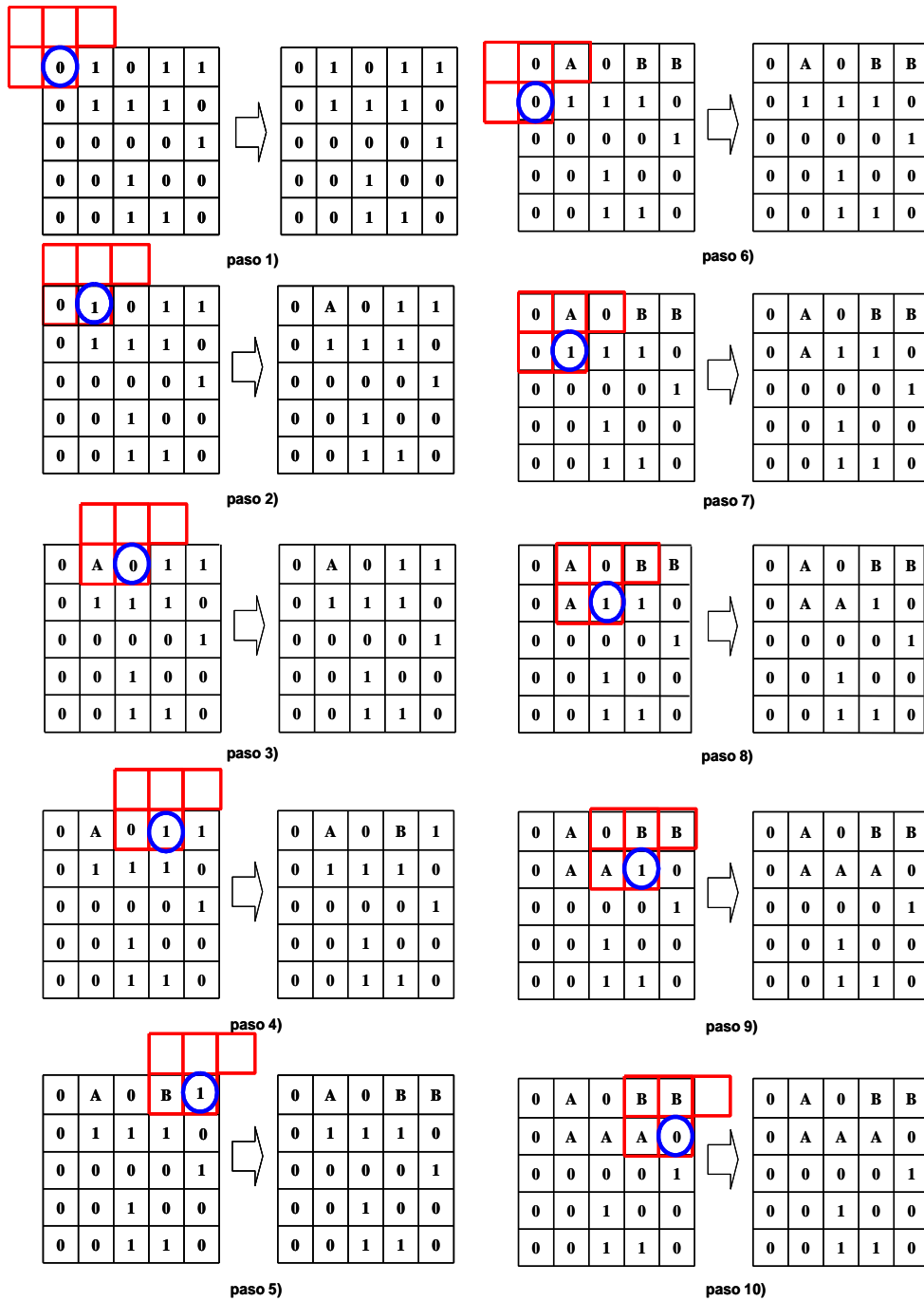


Figura 2.8. Los primeros diez pasos en el proceso de etiquetado

Paso 11: el décimo primer píxel a evaluar tiene el valor de "0". Por lo tanto se pasa al siguiente píxel.

Paso 12: el décimo segundo píxel a evaluar tiene el valor de "0". Por lo tanto se pasa al siguiente píxel.

Paso 13: el décimo tercer píxel a evaluar tiene el valor de “0”. Por lo tanto se pasa al siguiente píxel.

Paso 14: el décimo cuarto píxel a evaluar tiene el valor de “0”. Por lo tanto se pasa al siguiente píxel.

Paso 15: el décimo quinto píxel a evaluar tiene valor de “1”. Un píxel anteriormente evaluado tiene el valor de “A”. Por lo tanto se le asigna a este píxel la etiqueta “A”.

Paso 16: el décimo sexto píxel a evaluar tiene el valor de “0”. Por lo tanto se pasa al siguiente píxel.

Paso 17: el décimo séptimo píxel a evaluar tiene el valor de “0”. Por lo tanto se pasa al siguiente píxel.

Paso 18: el décimo octavo píxel a evaluar tiene valor de “1”. Los píxeles anteriormente evaluados no tienen etiquetas. Por lo tanto se le asigna la etiqueta “C”.

Paso 19: el décimo noveno píxel a evaluar tiene el valor de “0”. Por lo tanto se pasa al siguiente píxel.

Paso 20: el vigésimo píxel a evaluar tiene el valor de “0”. Por lo tanto se pasa al siguiente píxel.

Paso 21: el vigésimo primer píxel a evaluar tiene el valor de “0”. Por lo tanto se pasa al siguiente píxel.

Paso 22: el vigésimo segundo píxel a evaluar tiene el valor de “0”. Por lo tanto se pasa al siguiente píxel.

Paso 23: el vigésimo tercer píxel a evaluar tiene valor de “1”. Un píxel anteriormente evaluado tiene el valor de “C”. Por lo tanto se le asigna a este píxel la etiqueta “C”.

Paso 24: el vigésimo cuarto píxel a evaluar tiene valor de “1”. Dos píxeles anteriormente evaluados tienen el valor de “C”. Por lo tanto se le asigna a este píxel la etiqueta “C”.

Paso 25: el vigésimo quinto píxel a evaluar tiene valor de “0”.

Ahora se procede a reemplazar las etiquetas equivalentes. En este ejemplo solo la etiqueta “B”, es equivalente a la etiqueta “A”. Por lo tanto las etiquetas con valor “B” pasan a tener el valor a “A”. La Figura 2.12 muestra el resultado final del proceso.

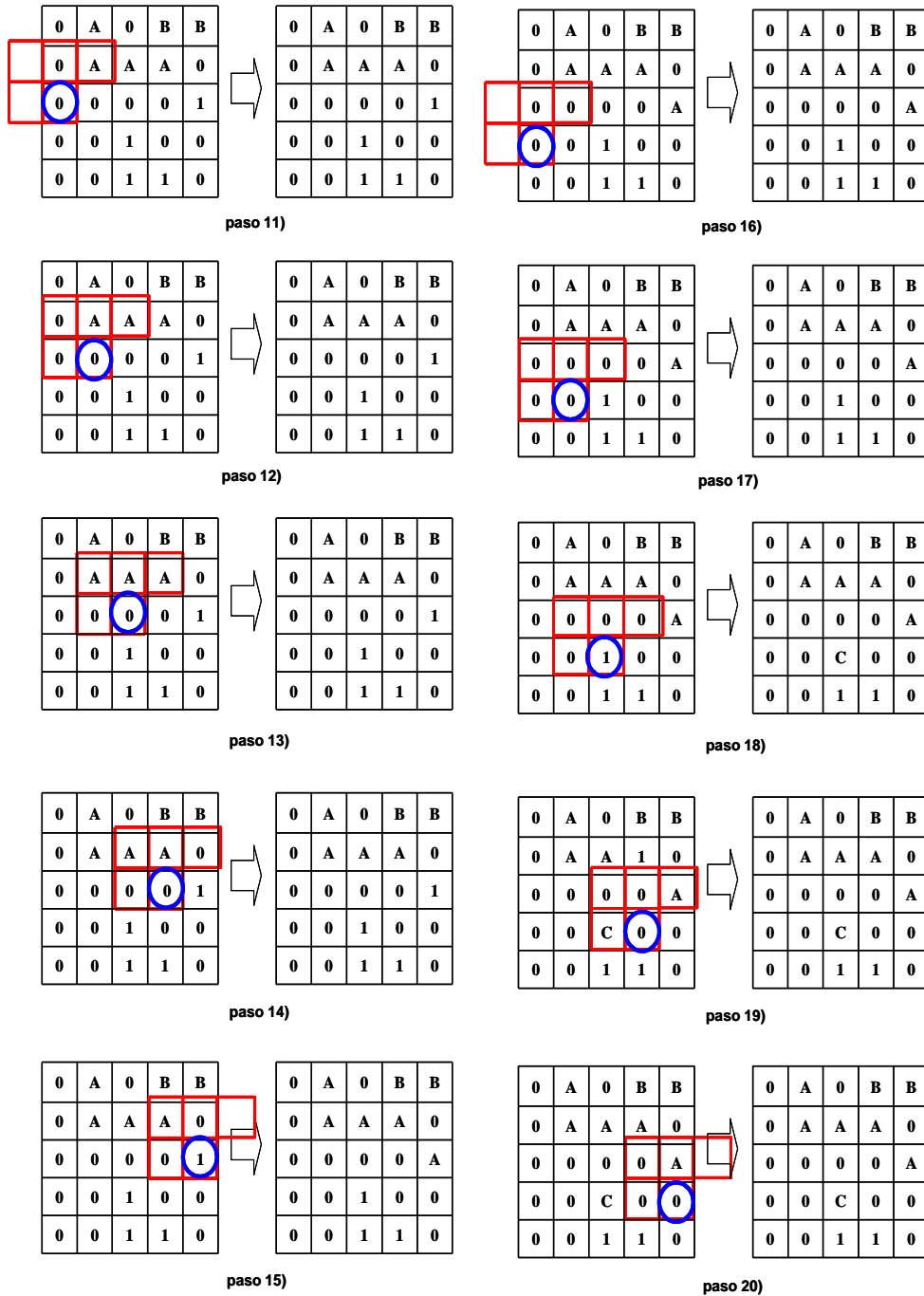
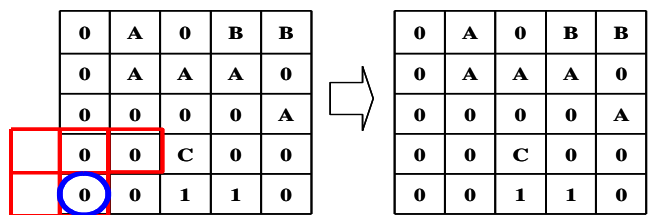
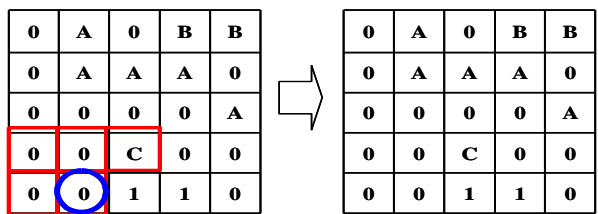


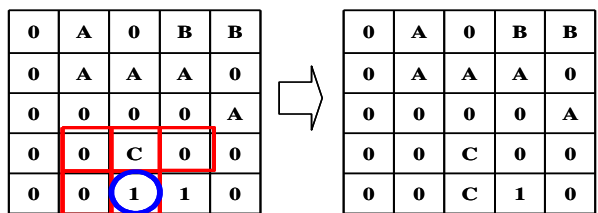
Figura 2.9 Los diez pasos siguientes en el proceso de etiquetado



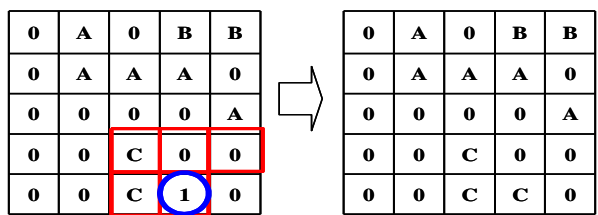
paso 21)



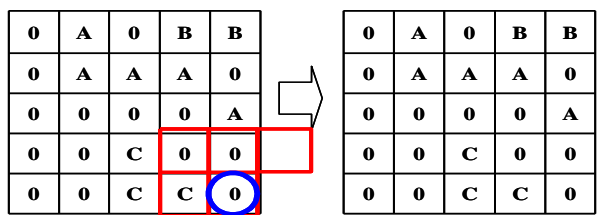
paso 22)



paso 23)



paso 24)



paso 25)

Figura 2.10 Los últimos cinco pasos en el proceso de etiquetado

0	A	0	A	A
0	A	A	A	0
0	0	0	0	A
0	0	C	0	0
0	0	C	C	0

Figura 2.11. Resultado final del proceso de etiquetado

2.9 Distancia entre pixeles

Sea que $p(x_0, y_0)$ y $q(x_1, y_1)$ son dos pixeles distintos, y “d” es una función de la distancia.

2.9.1 Distancia euclidiana

La distancia euclidiana entre los pixeles “p” y “q” es la raíz cuadrada de la diferencia de las coordenadas al cuadrado [6].

$$d(p, q) = \sqrt{(x_0 - x_1)^2 + (y_0 - y_1)^2} \quad (2.4)$$

2.10 Operaciones aritméticas lógicas entre pixeles

Las operaciones aritmético lógicas, son operaciones aplicadas directamente sobre los pixeles. Considérese los pixeles “p” y “q”.

2.10.1 Negación

Se aplica la operación lógica NOT (NOT p) en los pixeles de una imagen binaria. En la figura 2.12 se aprecia la operación resultante.

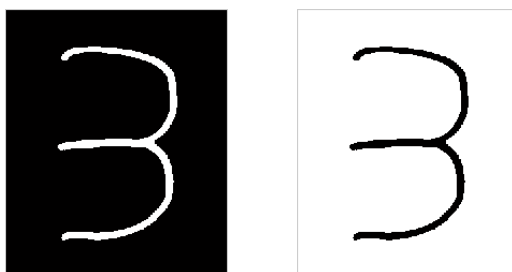


Figura 2.12 Operación NOT

2.10.2 Adición

La suma de dos imágenes $f(x, y)$ y $h(x, y)$, es expresada como se indica en la siguiente ecuación.

$$g(x, y) = |f(x, y) + h(x, y)| \quad (2.5)$$

Como ejemplo de este proceso se muestra en la Figura 2.13 la suma de dos imágenes.

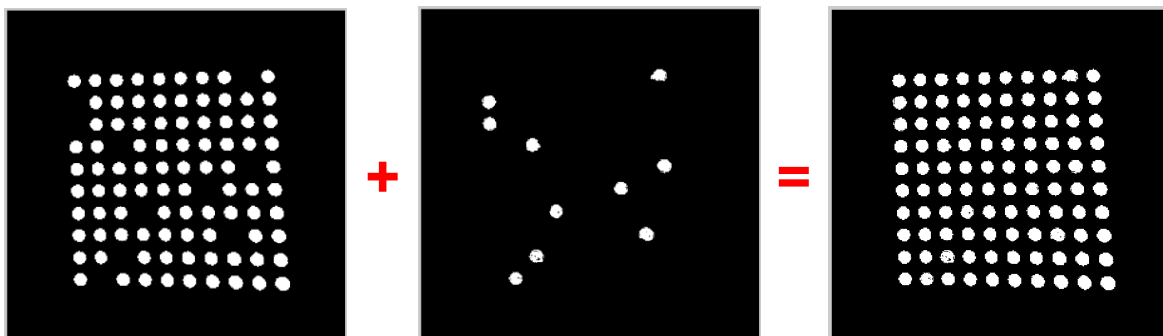


Figura 2.13 Suma de dos imágenes binarias

2.10.3 Substracción

La diferencia entre dos imágenes $f(x, y)$ y $h(x, y)$, es expresada como se indica en la siguiente ecuación.

$$g(x, y) = |f(x, y) - h(x, y)| \quad (2.6)$$

Como ejemplo de este proceso se muestra en la Figura 2.14 la resta de dos imágenes.

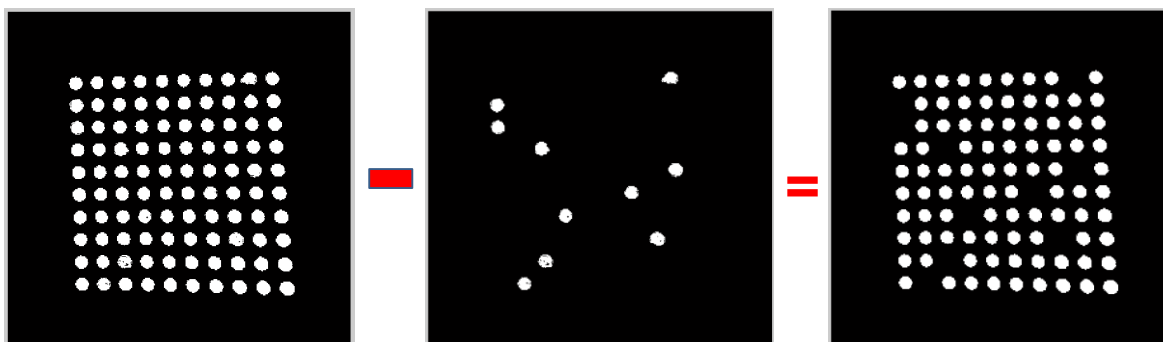


Figura 2.14 Substracción de Imágenes binarias

2.11 Binarización de una imagen

La binarización es una técnica que permite convertir imágenes con niveles de gris, en una imagen binaria (blanco y negro). De acuerdo a la técnica, los valores del píxel, en la imagen de entrada que son menores a un cierto umbral pre especificado, son convertidos a blanco, mientras que los píxeles con valores mayores al umbral, son convertidos a negro. Esto se muestra en la Ecuación 2.7. Así mismo este proceso puede darse de manera inversa.

$$I_{Binaria} = \begin{cases} 255 & ; I(x, y) \geq umbral \\ 0 & ; I(x, y) < umbral \end{cases} \quad (2.7)$$

Donde:

$I_{Binaria}$ imagen binaria

I imagen en escala de grises

2.12 Rotación de una imagen

Dado un píxel $I_{MB}(x_0, y_0)$ y se rota θ grados, así las coordenadas x_0' y y_0' del nuevo píxel serán [6]:

$$\begin{bmatrix} x_0' \\ y_0' \end{bmatrix} = \begin{bmatrix} \cos \theta & -\text{sen} \theta \\ \text{sen} \theta & \cos \theta \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \quad (2.8)$$

Por lo tanto, la versión girada de la imagen principal será:

$$I_{MB}(x', y') = I_{MB}(x \cdot \cos \theta - y \cdot \text{sen} \theta, x \cdot \text{sen} \theta + y \cdot \cos \theta) \quad (2.9)$$

2.13 Inclinación de una imagen

La inclinación de caracteres que se encuentran normalmente en la escritura cursiva, se denomina sesgo [22]. Se puede apreciar en la Figura 2.15

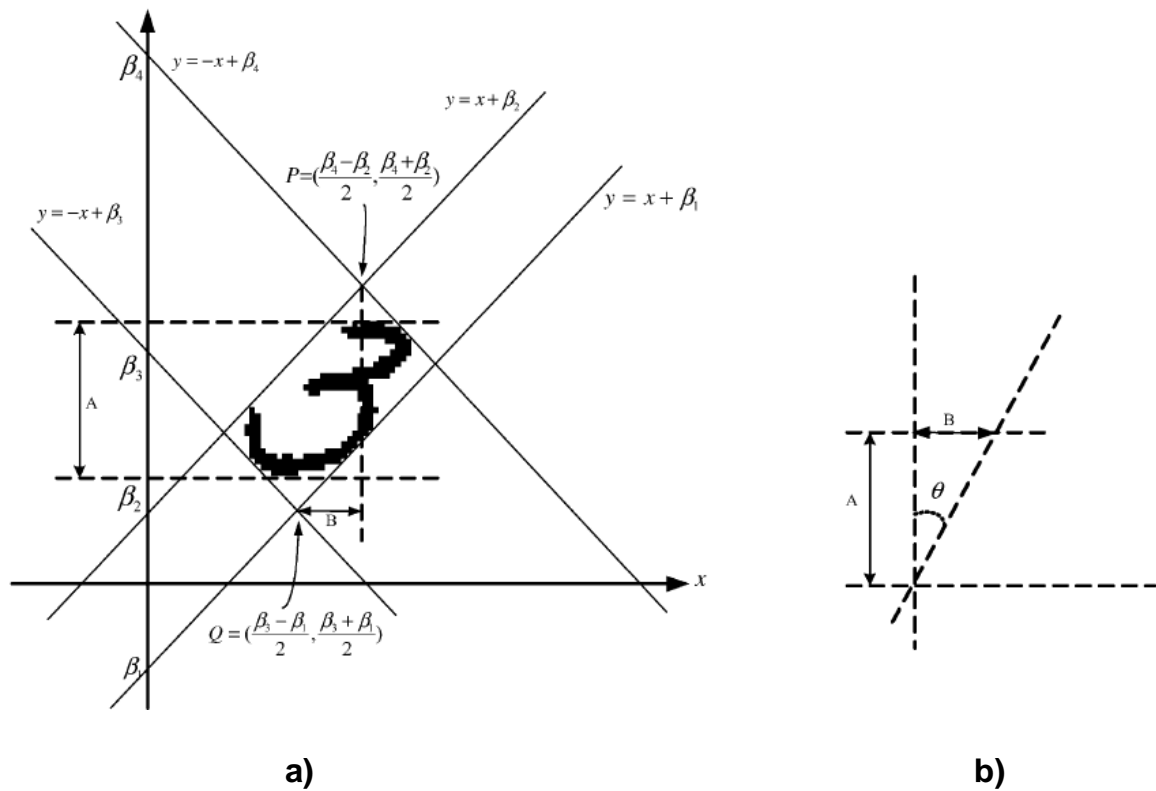


Figura 2.15 a) Número circunscrito por un rectángulo inclinado; b) estimación del ángulo de inclinación para el número de (a). [6]

2.13.1 Estimación del ángulo de inclinación

Un carácter está circunscrito por un rectángulo inclinado, como se muestra en la figura 2.15a. En otras palabras, cada carácter o dígito está rodeado por las siguientes cuatro líneas:

$$y = x + \beta_1 \quad (2.10)$$

$$y = x + \beta_2 \quad (2.11)$$

$$y = -x + \beta_3 \quad (2.12)$$

$$y = -x + \beta_4 \quad (2.13)$$

Donde $b_2 > b_1$ y $b_4 > b_3$, respectivamente

Como se aprecia en la Figura 2.15b, el ángulo de inclinación se puede calcular como

$$\theta = \arctan(B/A)$$

$A = \text{altura del dígito/carácter}$

(2.14)

$$B = \frac{\beta_4 + \beta_1 - \beta_3 - \beta_2}{2}$$
(2.15)

En correspondencia con la orientación de la inclinación hacia la izquierda o hacia la derecha, θ puede tener valores positivos o negativos, respectivamente.

2.13.2 Corrección del ángulo de inclinación

Una vez obtenido la estimación del ángulo de inclinación (θ), una transformación horizontal se aplica a todos los píxeles de la imagen de los dígitos con el fin de desplazar hacia la izquierda o hacia la derecha (dependiendo del signo de θ). Las expresiones de transformación se dan a continuación:

$$x' = x - y \cdot \tan(\theta)$$
(2.16)

$$y' = y$$
(2.17)

En estas ecuaciones, “x” e “y” son las coordenadas originales horizontales y verticales de los píxeles de la imagen; x' e y' son las correspondientes coordenadas transformadas, respectivamente.

2.14 Operaciones morfológicas

Son procesos que se basan en el análisis topológico de las estructuras de los objetos de una imagen. Las operaciones morfológicas se aplican sobre imágenes binarias y se centran en la extracción de estructuras de un conjunto (objeto dentro de una imagen); para lo cual se requiere de otro conjunto, llamado elemento estructurante (ver Figura 2.16), el cual puede ser considerado como una máscara de filtraje.

A continuación, algunas formas típicas de los elementos estructurantes:

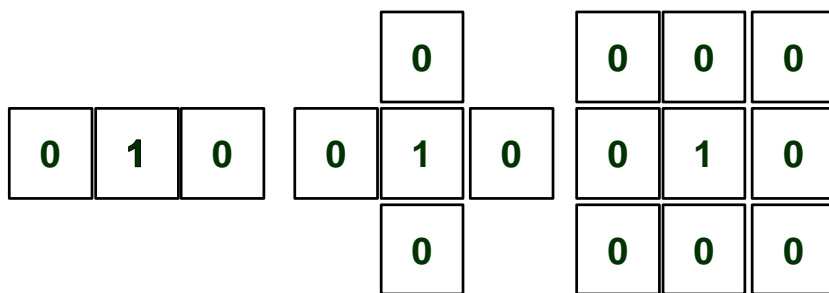


Figura 2.16 Algunos elementos estructurales

Los operadores morfológicos tratan a las imágenes como conjuntos de puntos; por lo que la teoría de conjuntos es usada comúnmente en una descripción matemática de estos conceptos. De esta manera, un objeto perteneciente a una imagen, constituye un conjunto y el fondo de la imagen representa su complemento.

Las operaciones morfológicas se suelen utilizar para eliminar puntos aislados que no pertenecen al objeto de interés, y que es parte de la imagen que se está analizando.

2.14.1 Transformaciones morfológicas

Algunas de las transformaciones morfológicas son: erosión, dilatación, apertura y cierre. Teniendo la imagen A y el elemento estructural B, (ambos imágenes binarias con fondo blanco) se tiene las siguientes operaciones.

2.14.1.1 Dilatación binaria

Dado dos conjuntos A y B de Z^2 , la dilatación [11], denotada por $A \oplus B$, se define como

$$A \oplus B = \{x | (\bar{B})_x \cap A \neq \emptyset\} \quad (2.18)$$

Tener en cuenta que, para la intersección sólo se considera los píxeles negros de A y B.

Esta transformación sirve para aumentar el contorno de los objetos y unir líneas discontinuas de estos, producidas por un filtrado, etc. La máscara, dependiendo de la forma de ésta, se puede utilizar para realizar la dilatación en la dirección o direcciones que más interesen. En la Figura 2.17 se aprecia la operación de dilatación.

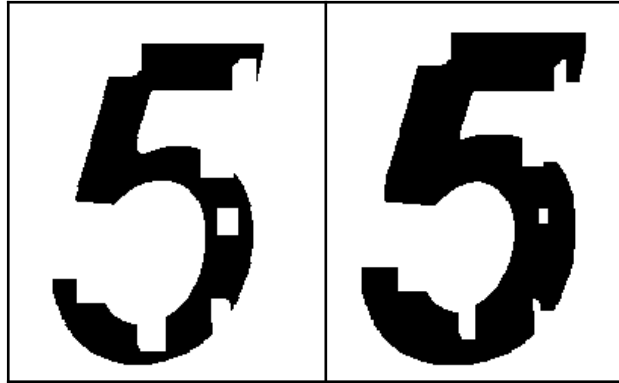


Figura 2.17 Ejemplo de dilatación. En la imagen del lado izquierdo se aprecia una imagen con orificios, en la imagen del lado derecho se tiene la imagen de la izquierda dilatada.

2.14.1.2 Erosión binaria

Dado dos conjuntos A y B de Z^2 , la erosión [11], denotada por $A \ominus B$, se define como

$$A \ominus B = \{x \mid B_x \subseteq A\} \quad (2.19)$$

Tener en cuenta que, para la condición $B_x \subseteq A$, sólo se considera los píxeles negros de A y B .

La erosión es la función dual de la dilatación (pero no es la inversa, es decir, si se hace una erosión y luego una dilatación no se queda la imagen como estaba). Dada una imagen A se erosiona por B cuando para todos los puntos x tales que B , trasladado por x , está contenido en A .

Si la dilatación expandía los bordes y contornos de los objetos, la erosión reduce los contornos de los objetos. Se utiliza para separar objetos que están unidos por una pequeña parte de los contornos.

La máscara, dependiendo de la forma de esta, se puede utilizar para realizar la erosión en la dirección o direcciones que interesen. En la Figura 2.18 se aprecia la operación de erosión.

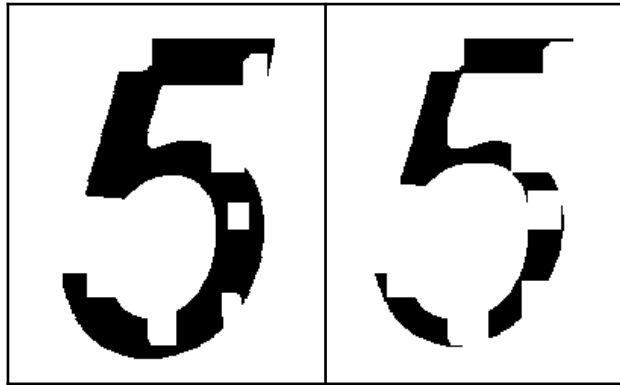


Figura 2.18 Ejemplo de erosión. En la imagen del lado izquierdo se aprecia una imagen con orificios, en la imagen del lado derecho se tiene la imagen de la izquierda erosionada.

2.14.1.3 Apertura

La apertura de A por un elemento estructural B [11] se define como la erosión de A por B, seguido de la dilatación del resultado por B.

$$A \circ B = (A \ominus B) \oplus B \quad (2.20)$$

Las aperturas pueden estar formadas por un número de erosiones y dilataciones diferentes, produciendo resultados muy dispares. En la Figura 2.19 se aprecia la operación de apertura.

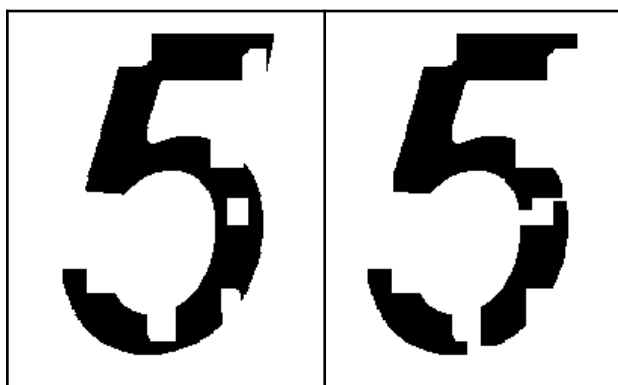


Figura 2.19 Ejemplo de operación de apertura.

2.14.1.4 Cierre

La operación de cierre o clausura de A por un elemento B [11] consiste en realizar la dilatación de A por B, seguido de la erosión del resultado por B.

$$A \bullet B = (A \oplus B) \ominus B \quad (2.21)$$

En la Figura 2.20 se aprecia la operación de cierre.

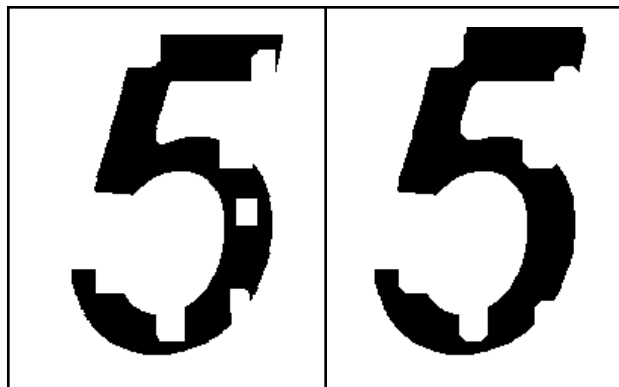


Figura 2.20 Ejemplo de operación de cierre

La combinación de las operaciones de apertura y cierre pueden servir para múltiples operaciones: filtrado, segmentación, etc.; simplemente escogiendo distintos tipos de máscaras y tamaños.

2.15 Esqueletización

La esqueletización es el proceso de quitar de un patrón la mayor cantidad de píxeles posibles sin afectar la forma general del patrón. En otras palabras, después que los píxeles hayan sido retirados, el patrón debe ser reconocido.

El esqueleto obtenido debe de ser tan delgado como sea posible, estar conectado y centrado. Cuando las condiciones anteriores se cumplan, el algoritmo debe detenerse.

En la Figura 2.21 se muestra un patrón y su esqueleto



Figura 2.21 Un patrón en forma de T, y su respectivo esqueleto

Otra manera de pensar sobre los esqueletos es como el lugar donde convergen los centros de círculos bitangenciales y entran de manera completa dentro de la región considerada.

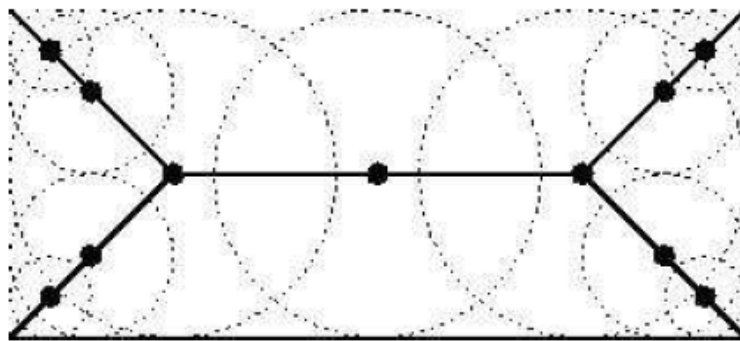


Figura 2.22 El esqueleto de un rectángulo definido en términos de círculos bitangenciales [2]

El termino transformación de eje medio (MAT) y esqueletización son comúnmente usados de manera indistinguible. El esqueleto es un imagen binaria que simplemente muestra el esqueleto de por sí, mientras que MAT es una imagen de niveles de grises, donde cada punto del esqueleto tiene una intensidad, la cual representa una distancia de los límites del objeto original [2].

Existen diversas técnicas de esqueletización [18], en el presente trabajo, se utiliza una modificación realizada por Zhang y Wang [27].

Defínase los siguientes términos:

Teniendo en cuenta la ecuación 2.22, donde se muestra el vecindario de un punto P.

$$\begin{array}{ccc}
P_7 & P_0 & P_1 \\
P_6 & P & P_2 \\
P_5 & P_4 & P_3
\end{array} \tag{2.22}$$

$B_8(P)$ es el número de vecinos marcados de P usando la representación o denominación 8 - conectividad y es utilizado para determinar si un punto es "final".

$B_8(P)$ se define como

$$B_8(P) = \sum_{k=0}^7 P_k \tag{2.23}$$

El algoritmo de Zhang y Suen (1984) [26], permite obtener el esqueleto de un objeto o el contorno adelgazado mediante la aplicación de iteraciones compuestas por dos fases:

a. Primera fase

$$2 \leq B_8(P) \leq 6 \tag{2.24}$$

$$A(P) = 1 \tag{2.25}$$

$$P_0 P_2 P_4 = 0 \tag{2.26}$$

$$P_2 P_4 P_6 = 0 \tag{2.27}$$

b. Segunda fase

$$2 \leq B_8(P) \leq 6 \tag{2.28}$$

$$A(P) = 1 \tag{2.29}$$

$$P_0 P_2 P_6 = 0 \tag{2.30}$$

$$P_0 P_4 P_6 = 0 \tag{2.31}$$

Donde: $A(P)$ es el número de transiciones de puntos no marcados a puntos marcados (transiciones de "0" a "1") en el conjunto ordenado P_0, P_1, \dots, P_7 ,

Este algoritmo no permite obtener siempre un esqueleto o contorno unitario, es decir, compuesto por líneas de un punto de grosor.

El punto P no puede ser borrado por dicho algoritmo porque no se verifica que $A(P)=1$.

Por otro lado, el algoritmo de Wang y Zhang (1989) [25], es un método iterativo compuesto por dos fases en las que se eliminan los puntos marcados que cumplan las siguientes condiciones

a. Fase sudeste

$$1 < B_8(P) < 7 \quad (2.32)$$

$$A(P) = 1 \vee C(P) = 1 \quad (2.33)$$

$$P_2 + P_4 P_0 P_6 = 0 \quad (2.34)$$

b. Fase noroeste

$$1 < B_8(P) < 7 \quad (2.35)$$

$$A(P) = 1 \vee C(P) = 1 \quad (2.36)$$

$$P_0 + P_6 P_2 P_4 = 0 \quad (2.37)$$

Donde $C(p)$ se define como:

$$C(P) = \begin{cases} 1 & \text{Si } P_{k-1} + P_k + P_{k+1} + P_{k+4} = 0 \\ & \wedge P_{k+3} = 1 \wedge P_{k+5} = 1 \\ & \wedge (k = 1 \vee k = 3) \\ 0 & \text{En cualquier otro caso} \end{cases} \quad (2.38)$$

2.16 Segmentación de imágenes

La segmentación de imágenes se define como la partición de una imagen en unidades significativas o regiones (no traslapadas) que presentan alguna característica similar (por ejemplo, nivel de intensidad). Esta región es comúnmente llamada objeto [11].

2.16.1 Segmentación basada en umbralización

La umbralización es una de las técnicas de segmentación más simple que existe, dado que realiza un análisis sobre el histograma. Después de seleccionar el valor del umbral U , cualquier valor de pixel que cumpla con $f(x, y) > U$, serán etiquetados como objeto; caso contrario será considerados como parte del fondo.

2.17 Descriptores de imagen

Los momentos y las funciones de los momentos se emplean como características de patrón en numerosas aplicaciones para reconocer patrones en imágenes de dos

dimensiones. Estas características del patrón extraen propiedades globales de la imagen, como la zona de forma, el centro de la masa, etc.

2.17.1 Momentos

La geometría de una región plana se basa en el tamaño, la posición, la orientación y la forma. Todas estas medidas están relacionadas con una familia de parámetros llamada momentos [19].

Los momentos de orden $p + q$ de una imagen $f(x, y)$ son

$$M_{pq} = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} x^p y^q f(x, y) \quad (2.39)$$

2.17.1.1 Área

Esta medida solo tiene significado válido en imágenes binarias. Se tomará la siguiente convención [19]:

$f(x,y) = 1$ si el píxel forma parte del objeto

$f(x,y) = 0$ si el píxel forma parte del fondo

Área es el número de píxeles para los cuales $f(x, y)$, es decir,

$$A = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} f(x, y) \quad (2.40)$$

Se puede expresar en función del momento de orden cero, $M(0,0)$

2.17.1.2 Centro de gravedad

La definición física es el punto de un objeto que tiene la misma cantidad de objeto en cualquier dirección. En el caso de las imágenes se utiliza como punto de referencia del objeto y viene dado por [19].

$$\bar{x} = \frac{\sum_{x=0}^{N-1} \sum_{y=0}^{M-1} x f(x, y)}{\sum_{x=0}^{N-1} \sum_{y=0}^{M-1} f(x, y)} \quad (2.41)$$

$$\bar{y} = \frac{\sum_{x=0}^{N-1} \sum_{y=0}^{M-1} yf(x, y)}{\sum_{x=0}^{N-1} \sum_{y=0}^{M-1} f(x, y)} \quad (2.42)$$

Se puede expresar en función de los momentos

$$\bar{x} = M(1,0) / M(0,0) \quad (2.43)$$

$$\bar{y} = M(0,1) / M(0,0) \quad (2.44)$$

2.17.2 Coeficiente de correlación

El coeficiente de correlación es una medida normalizada [0; 1], que indica en términos cuantitativos la correspondencia entre dos imágenes. [11]

$$r = \frac{\sum_m \sum_n (A_{mn} - \bar{A})(B_{mn} - \bar{B})}{\sqrt{\left(\sum_m \sum_n (A_{mn} - \bar{A})^2 \right) \left(\sum_m \sum_n (B_{mn} - \bar{B})^2 \right)}} \quad (2.45)$$

En donde: A es la imagen base, B es la imagen a comparar y \bar{A} y \bar{B} son las medias de A y B respectivamente.

En la Figura 2.23 se tiene un ejemplo de dos imágenes a correlacionar; la imagen base (imagen de la izquierda) y la imagen a comparar (imagen de la derecha).

Dependiendo de su grado de correspondencia se obtendrá un coeficiente de correlación.



Figura 2.23 Imágenes a correlacionar. La imagen (izquierda) y la imagen a comparar (derecha)

2.18 Redes neuronales

Para definir las Redes Neuronales (RN) se deben tener claras algunas ideas y varios conceptos. Se puede empezar por mostrar el modelo biológico de una neurona (Figura 2.24), y describir su funcionamiento como un elemento simple que cuando un

umbral es superado por un estímulo de una serie de entradas, se genera una salida [14]. Esto se puede apreciar en la Figura 2.25

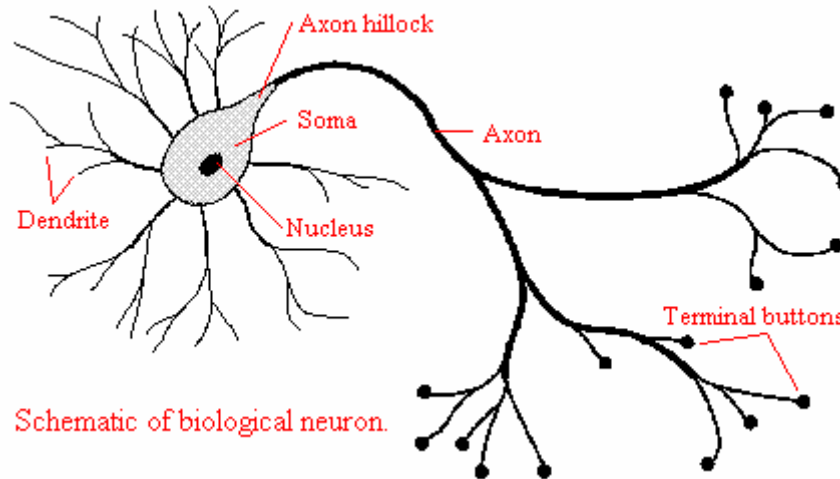


Figura 2.24 Esquema de una neurona biológica

2.18.1 Definición

Una vez definido el modelo biológico, se puede hablar del modelo artificial o modelo de una neurona artificial (unidad). La neurona artificial estándar es un elemento de procesamiento que calcula una salida multiplicando su vector de entradas por un vector de pesos y este resultado es aplicado a una función de activación, la cual será definida posteriormente. [14]

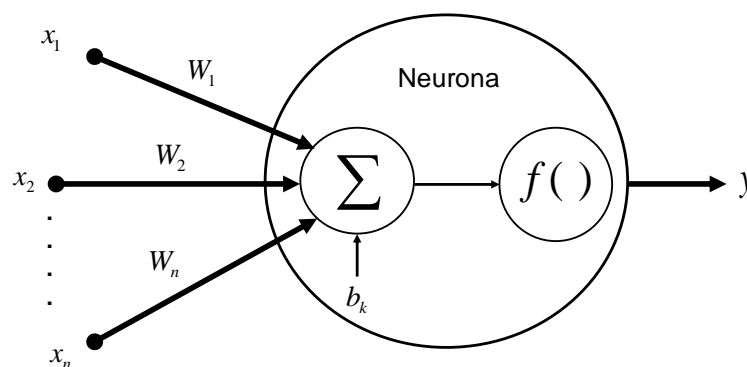


Figura 2.25 Neurona artificial

El modelo de una unidad, se representa por la siguiente ecuación:

$$y = f\left(\sum_{k=1}^n x_k w_k + b_k\right) \quad (2.46)$$

Donde:

x : vector de entradas
 w : vector de pesos
 b : bias (fija nivel de umbral)
 $f(\cdot)$: función de activación
 y : salida

2.18.2 Funciones de activación

Las funciones de activación hacen las veces de la función umbral de la neurona biológica. Dentro de las más conocidas se consideran a las funciones de activación lineal, tangente hiperbólico, sigmoide, signo, etc. [14]

Función sigmoide

$$f_{(a)} = \frac{1}{1 + e^{-\beta a}} \quad (2.47)$$

Función lineal

$$f_{(a)} = a \quad (2.48)$$

Función tangente hiperbólico

$$f_{(a)} = \frac{e^a - e^{-a}}{e^a + e^{-a}} \quad (2.49)$$

2.18.3 Características de las redes neuronales

Algunas de las características más importantes de las redes neuronales pueden ser:

- a. Son elementos que pueden tener comportamiento no-lineal.
- b. Son adaptables.
- c. Las respuestas están basadas en evidencia.
- d. Usan contexto, es decir, a más información, la respuesta es más precisa.
- e. Son tolerantes a fallas, o sea, que la falla es degradada.
- f. Su diseño y análisis es uniforme.
- g. Tienen analogía neurobiológica.
- h. Su procesamiento es masivamente paralelo, distribuido y realizado con operaciones individuales simples.

- i. Pueden tener algoritmos de auto aprendizaje
- j. Tienen capacidad de generalizar.

2.18.4 Métodos de aprendizaje

En una red neuronal es necesario definir un procedimiento por el cual las conexiones de la red varían para proporcionar la salida deseada (algoritmo de aprendizaje). Los métodos de aprendizaje se pueden dividir en las siguientes categorías [14]

La primera gran división en los métodos de aprendizaje es entre algoritmos supervisados y no supervisados

2.18.4.1 Aprendizaje Supervisado

El aprendizaje supervisado presenta a la red las salidas que debe proporcionar ante las señales que se le presentan. Se observa la salida de la red y se determina la diferencia entre esta y la señal deseada. Posteriormente, los pesos de la red son modificados de acuerdo con el error cometido. Este aprendizaje admite dos variantes: aprendizaje por refuerzo o aprendizaje por corrección. En el aprendizaje por refuerzo solo conocemos si la salida de la red se corresponde o no con la señal deseada, es decir, nuestra información es de tipo booleana (verdadero o falso). En el aprendizaje por corrección conocemos la magnitud del error y esta determina la magnitud en el cambio de los pesos.

2.18.4.2 Aprendizaje no-supervisado

En los algoritmos no supervisados no se conoce la señal que debe dar la red neuronal (señal deseada). La señal en este caso se organiza ella misma agrupando, según sus características, las diferentes señales de entrada. Estos sistemas proporcionan un método de clasificación de las diferentes entradas mediante técnicas de agrupamiento o clustering.

2.18.5 Generalización

La generalización es una medida de cómo se debería comportar la red neuronal una vez terminado su entrenamiento frente a datos de entrada nuevos. La generalización es función principalmente de:

- a. Cantidad de datos para el entrenamiento (patrones)
- b. Calidad de la muestra de los patrones de entrenamiento
- c. Complejidad del problema o de la relación entre variable de entrada y salida

d. Tamaño de la red

2.18.6 Perceptrón

La capa más simple de una sola neurona es el Perceptrón. Este se representa como un sumador lineal seguido 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.[9]

$$y_{(x)} = \Phi \left(\sum_{i=1}^p x_i w_i - \theta \right) \quad (2.50)$$

Donde $\Phi(\cdot)$ es la función de activación

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 signo. 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 algunos 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.

2.18.7 Red Perceptrón multicapa (MLP):

Una red Perceptrón multicapa es una red con alimentación hacia adelante, esta consiste de una capa de entrada, una o más capas ocultas y una capa de salida. Los cálculos se realizan en las capas ocultas y en la capa de salida. La señal de entrada se propaga a través de la red hacia adelante capa por capa. La red Perceptrón multicapa es entrenado usando el algoritmo Backpropagation. Este es un algoritmo de aprendizaje supervisado de corrección de errores. [14]

La arquitectura del Perceptrón Multicapa con dos capas ocultas se muestra en la Figura 2.26.

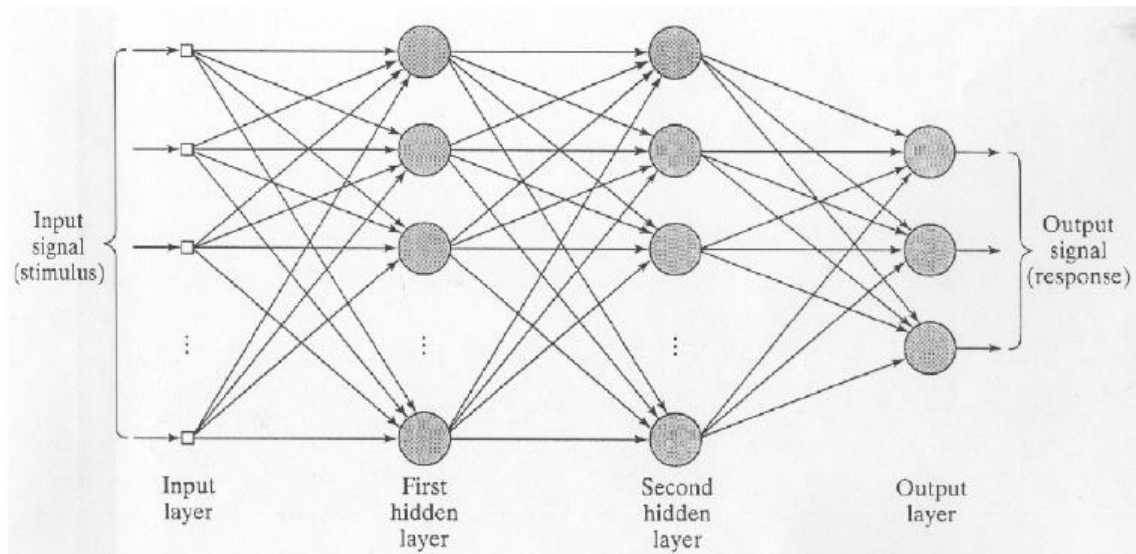


Figura 2.26 Arquitectura del Perceptrón Multicapa con dos capas ocultas [14]

2.18.7.1 Algoritmo 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 a través de las diferentes capas, uno hacia adelante y otro hacia atrás.

En el primer proceso, la respuesta de la red a una entrada es calculada, aquí todos los pesos sinápticos se mantienen fijos.

En el segundo, los pesos son ajustados mediante una corrección de errores. La señal de error se propaga hacia atrás a través de la red. Después del ajuste, la salida de la red debería haberse acercado más a la respuesta deseada en un sentido estadístico.

El algoritmo de aprendizaje de Backpropagation es una técnica de optimización que procura minimizar una función de costo. [14]

Los valores iniciales de los pesos y biases se pueden elegir de una distribución uniforme con media cero a menos que se tenga una información previa disponible.

$$\begin{pmatrix} \text{Corrección} \\ \text{del peso} \\ \Delta\omega_{ij}(n) \end{pmatrix} = \begin{pmatrix} \text{Parámetro} \\ \text{de aprendizaje} \\ \eta \end{pmatrix} \begin{pmatrix} \text{Gradiente} \\ \text{local} \\ \delta_j(n) \end{pmatrix} \begin{pmatrix} \text{Señal de salida} \\ \text{de la neurona } j \\ y_j(n) \end{pmatrix} \quad (2.51)$$

La gradiente local está dada por

$$\delta_j(n) = e_j(n)\varphi'_j(v_j(n)) \quad (2.52)$$

Donde $v_j(n)$ denota el cuerpo local de la neurona j en la iteración n . Es la suma ponderada de las entradas más el umbral de la neurona. $\varphi_j(\cdot)$ es la función de activación, y $e_j(n)$ es el error de la señal a la salida de la neurona j para la iteración n .

Cuando la neurona j es la capa de salida. En la capa de oculta, la gradiente local es

$$\delta_j(n) = \varphi'_j(v_j(n)) \sum_k \delta_k(n) \omega_{kj}(n) \quad (2.53)$$

Donde ω_{ji} es el peso que conecta la salida de la neurona i a la entrada de la neurona j en la iteración n .

El error de backpropagation es calculado de forma recursiva a partir de las gradientes locales de la capa siguiente.

Las dos etapas de cálculo:

Al aplicar el algoritmo Backpropagation se distinguen dos etapas de cálculo:

Etapas hacia adelante

Los pesos no se cambian en esta etapa. La señal de la función que aparece en la salida de la neurona j se calcula como:

$$y_j(n) = \varphi(v_j(n)) \quad (2.54)$$

Aquí el cuerpo local de la neurona $v_j(n)$ que denota a la neurona j es:

$$v_j(n) = \sum_{i=0}^m \omega_{ji}(n) y_i(n) \quad (2.55)$$

En la primera capa oculta, $m = m_0$ y es el número de señales de entradas $x_i(n), i = 1, \dots, m_0$, y en la ecuación 2.56

$$y_i(n) = x_i(n) \quad (2.56)$$

En la capa de salida, $m = m_L$ y es el número de salidas en la ecuación 2.50. Las salidas (los componentes del vector de salida) se denotan por

$$y_j(n) = o_j(n) \quad (2.57)$$

Donde $o_j(n)$ denota el j ésimo elemento del vector de salida global.

Estas salidas son comparadas con las respectivas respuestas deseadas $d_j(n)$, modificando la señal de error $e_j(n)$. Donde $d_j(n)$ es la respuesta deseada de la neurona j . En la etapa hacia adelante, el cálculo comienza de la primera capa oculta y termina en la última capa.

Etapas hacia atrás

En la etapa hacia atrás, el cálculo comienza en la última capa y finaliza en la primera capa oculta.

La gradiente local δ es calculada por cada neurona haciendo pasar la señal de error a través de la red capa por capa.

La regla delta de la ecuación 2.51 es usada para actualizar los pesos sinápticos. Los pesos actualizados se calculan de manera recursiva capa por capa. Después de esto, el próximo vector (entrada) a entrenar es ingresado a la red.

A continuación se aplicará el Algoritmo de Backpropagation a una red neuronal que tenga funciones de activación sigmoideas en la capa oculta y funciones lineales en la capa de salida. El sistema a ser aprendido tiene dos entradas y una salida. Se sabe que se cuenta con 4 patrones de entrenamiento:


```

ve = [0.4 0.8 1.3 -1.3
      0.7 0.9 1.8 -0.9];
vs = [0.1 0.3 0.6 0.2];

```

Solución:

% Algoritmo de entrenamiento del backpropagation utilizado en un MLP de capa
 %funciones de activación de la capa oculta sigmoides y de salida lineales.

clear all; close all

% Inicializando los datos

```

ve = [ 0.4 0.8 1.3 -1.3
      0.7 0.9 1.8 -0.9 ];
vs = [ 0.1 0.3 0.6 0.2 ];
[entradas,patrones] = size(ve);
[salidas, patrones] = size(vs);
u_ocultas = 6;

```

```

W1 = 0.1*ones(u_ocultas,entradas+1);% Inicializando la matriz de pesos con 0.1
W2 = 0.1*ones(salidas,u_ocultas+1); % Inicializando la matriz de pesos con 0.1

```

```

maxciclos = 200;           % Cantidad máxima de ciclos
Error_Deseado = 0.03;    % Mínimo Error deseado
fa = 0.5;                 % Factor de aprendizaje (lr)
sy = size(vs);
terminos = sy(1,1)*sy(1,2); % Cantidad de términos en una salida del
                             %entrenamiento
X = [ ones(1,patrones) ve ]; % Aumento del vector ve con el bias.

```

```

for i = 1:maxciclos
    h = sigmoide(W1*X);
    H = [ ones(1,patrones) h];
    e = yd - sigmoide(W2*H);
    Error(i) = sum(sum(e.^2))/terminos;
    ciclos(i) = i;

    if Error(i) < Error_Deseado; break; end
    y = sigmoide(W2*H);
    delta2 = y.*(1-y).*e;
    del_W2 = 2*fa* delta2*H';
    W2 = W2 + del_W2;
    delta1 = h.*(1-h).*(W2(:,2:u_ocultas+1))*delta2);
    del_W1 = 2*fa*delta1*X';
    W1 = W1 + del_W1;

```

end

% Gráfica del Error por los Ciclos

```

figure(1)
plot(ciclos,Error);
title('Entrenamiento de Backpropagation');
xlabel('Ciclos');ylabel('Media del Error Cuadrático')
grid on; zoom on

```

```
if i < 200; fprintf('El Error Deseado se ha alcanzado en %i ciclos.',i);end
```

Sabiendo que la función sigmoide es:

```
function [y] = sigmoide(x);  
%SIGMOIDE o Logistic, Función de Activación  
% [y] = sigmoide(x)  
% x : la entrada  
% y : la función sigmoide de cada una de las entradas  
%  
% Devuelve el resultado de aplicar la función  
% sigmoide a la entrada x.  
y = 1./(1+exp(-x));
```

2.18.7.2 Mínimos locales y mínimos globales

Los mínimos locales son mínimos de un rango determinado de una función, mientras que el mínimo global es el mínimo de la función, se puede decir que es el mínimo de todos los mínimos locales. [14]

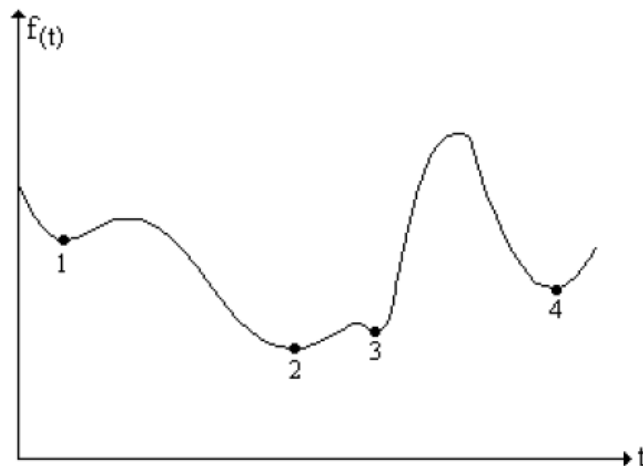


Figura 2.27 Mínimos locales y mínimos globales

En la Figura 2.27 se puede apreciar que los puntos 1, 3 y 4 son mínimos locales pero que a su vez el punto 2 es el mínimo global.

Cuando se entrena una red neuronal se cae en un mínimo local, lo más probable es que nunca se llegue al error mínimo deseado. Este problema de caer en un mínimo local se puede solucionar adicionando neuronas a la capa oculta, y a su vez para llegar más rápido a dicho mínimo se puede utilizar el factor de aprendizaje adaptivo.

Para que el factor de aprendizaje sea adaptivo se debe colocar la condición de que si el error actual es mayor que el anterior, no se deben modificar los parámetros y el factor de aprendizaje se multiplica por un número sustancialmente menor que cero. Si el error actual es menor que el anterior, se deben modificar los parámetros y el factor de aprendizaje se multiplica por un número ligeramente mayor que cero.

2.19 Validación cruzada

Se espera que una red MLP entrenada con el algoritmo de Backpropagation aprenda lo suficiente del pasado para generalizar para el futuro. El proceso de aprendizaje equivale a una elección de parametrización de la red para un conjunto de datos. Se trata de un problema de selección de modelo: escoger la mejor opción dentro de un conjunto de modelos candidatos. [14]

Una herramienta estadística útil para la selección del modelo se llama validación cruzada. Esta consiste en dividir los datos disponibles aleatoriamente en dos grupos: Subconjunto de aprendizaje, usado para seleccionar el modelo y Subconjunto de validación, usado para validar al modelo.

La idea es validar al modelo con una data diferente de la data utilizada para entrenar al modelo. El uso de la validación cruzada es útil particularmente cuando se diseña una red neuronal con una buena generalización. Por ejemplo se utiliza la validación cruzada para determinar en el Perceptrón multicapa cual es el mejor número de neuronas ocultas y cuando el entrenamiento se detiene.

Según Kearns (1996), debido a que no se puede determinar si una red neuronal ha generalizado bien o no, es conveniente validar o comprobar su entrenamiento. Para ello, del total de patrones disponibles para el entrenamiento se deben reservar de un 10 a 20 % para la validación cruzada, los dos grupos de patrones, el de entrenamiento y el de validación deben estar distribuidos normalmente.

En La Figura 2.28 se muestra las dos curvas, uno perteneciente a las mediciones en el subconjunto de estimación y la otra relacionada con el subconjunto de validación. Normalmente, el modelo no va tan bien en el subconjunto de validación como lo hace en el subconjunto de estimación, en la que se basó su diseño. La curva de aprendizaje de estimación disminuye monotónicamente para un número cada vez mayor de épocas de la

manera habitual. En contraste, la curva de aprendizaje de validación disminuye monotónicamente a un mínimo, y de ahí comienza a incrementarse de manera continua.

El entrenamiento se debe detener cuando el error, en el conjunto de validación, comienza a crecer. En ese punto, se llega a la mejor generalización

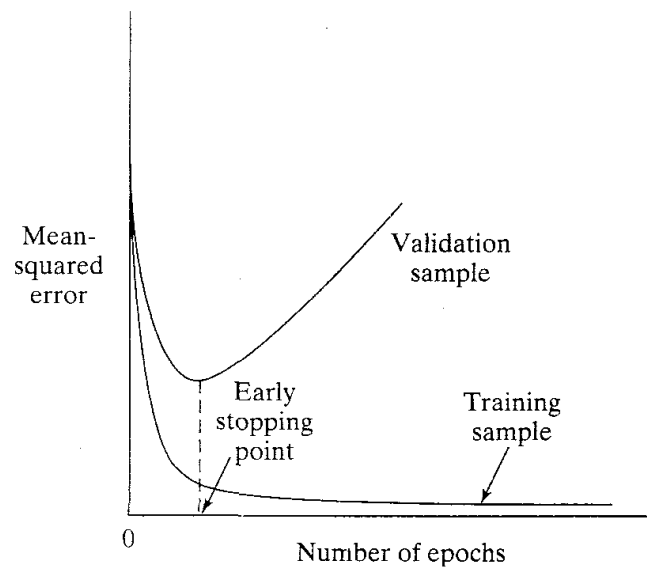


Figura 2.28 Punto de parada basado en la regla de validación cruzada [14]

CAPÍTULO III

DESCRIPCIÓN DEL SISTEMA PROPUESTO

3.1 Descripción del hardware

La etapa del hardware (Figura 3.1) consiste en un recinto que contiene al cuadernillo de evaluación a registrar. Dicho recinto contiene una cámara WEB y una adecuada iluminación provista por un arreglo de LEDs de alta luminosidad. Asimismo, se cuenta con un computador donde se encuentra instalado el software desarrollado para realizar el reconocimiento de la nota y código del alumno. Posteriormente, estos datos podrán ser almacenados.

El recinto fue diseñado por alumnos de pregrado de la Escuela de Ingeniería Electrónica de la Universidad de San Martín de Porres. En la Figura 3.2 se muestran fotos del recinto prototipo diseñado.

El recinto tiene una capacidad de carga de 40 cuadernillos, es decir, puede procesar en bloques de 40 cuarenta cuadernillos, que es el número promedio de alumnos por salón. El recinto tiene una bandeja inicial donde se cargan los cuadernillos y se toma la imagen del cuadernillo para procesar. Tiene un separador de cuadernillos, es decir, una vez procesado el cuadernillo este separador se encarga de enviar el cuadernillo procesado a la bandeja final, que es donde se almacenan todos los cuadernillos procesados. En la Figura 3.3, a la izquierda, se muestra la estructura mecánica del equipo, y a la derecha se muestra el diseño mecánico en el "SketchUp 8" de Google.

En la Figura 3.4, se muestra el diagrama de arreglo de LEDs. que se encuentra dentro del recinto. Esto permite tener una iluminación uniforme. La iluminación es suministrada por 6 LEDs de color blanco de 3.5 watts de potencia cada uno.

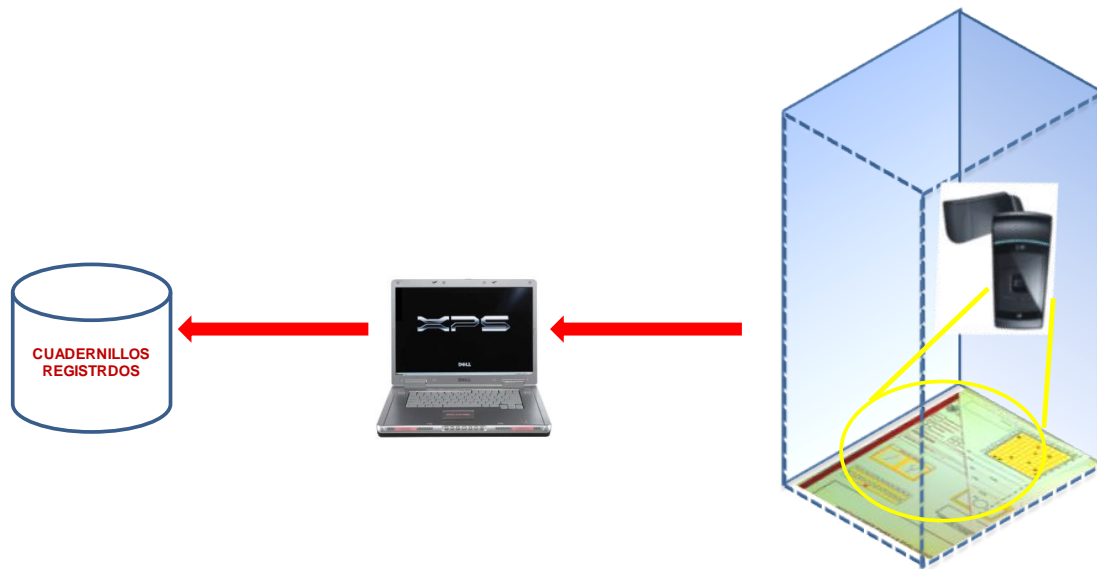


Figura 3.1 Bloques de la etapa física

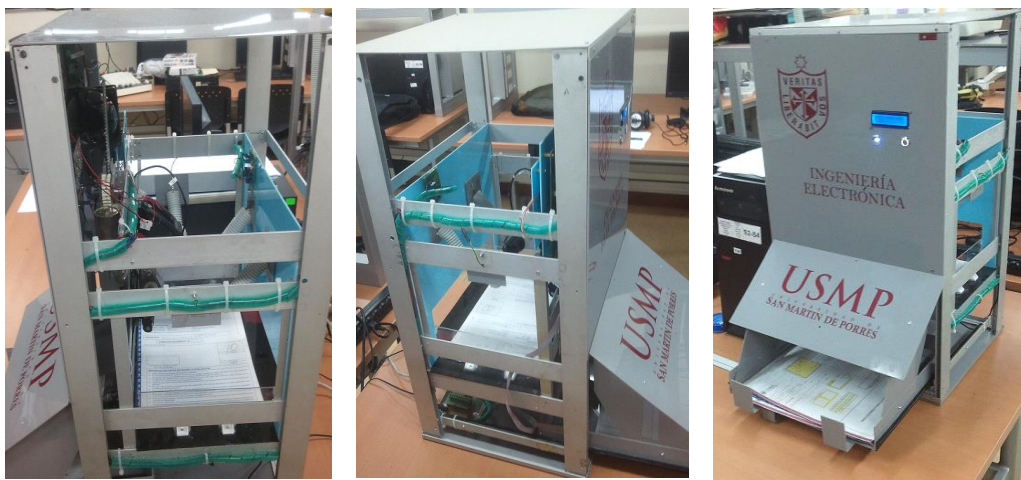


Figura 3.2 Fotos del recinto diseñado

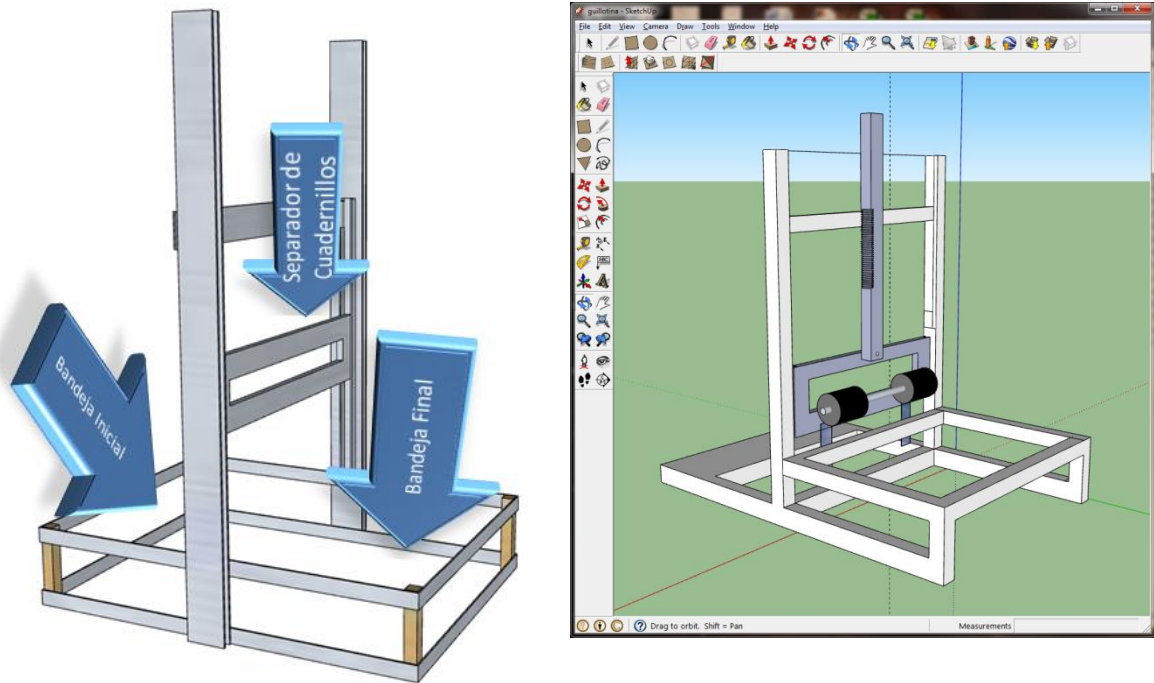


Figura 3.4 Estructura mecánica del equipo.

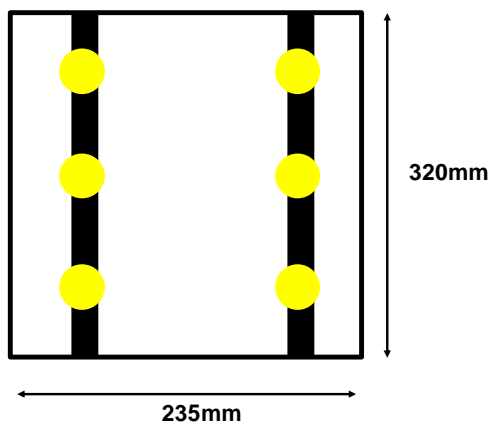


Figura 3.4 Diagrama de arreglo de LEDs

3.2 Descripción del software desarrollado

Los algoritmos se encuentran implementados en MATLAB. En el software se usa procesamiento de imágenes y redes neuronales. La parte de procesamiento digital de imágenes involucra algoritmos que tienen la función del pre-procesamiento, obtención de los parámetros de reconocimiento de las imágenes adquiridas y la parte de redes neuronales para la validación de dichas imágenes.

Finalmente se muestra en pantalla, el código del alumno y su nota correspondiente y estos pueden ser almacenados en un archivo o base de datos.

El sistema inicia adquiriendo la imagen del cuadernillo de evaluación, y de este extrae tres (3) áreas a procesar. Estas son código y nota del alumno en “marcas” y la nota del alumno en números manuscritos; estas imágenes pasan por un pre procesamiento, que deja a las imágenes listas para la etapa de reconocimiento y, posteriormente el almacenamiento de los datos. En la Figura 3.5, se muestra el diagrama de flujo general del sistema.

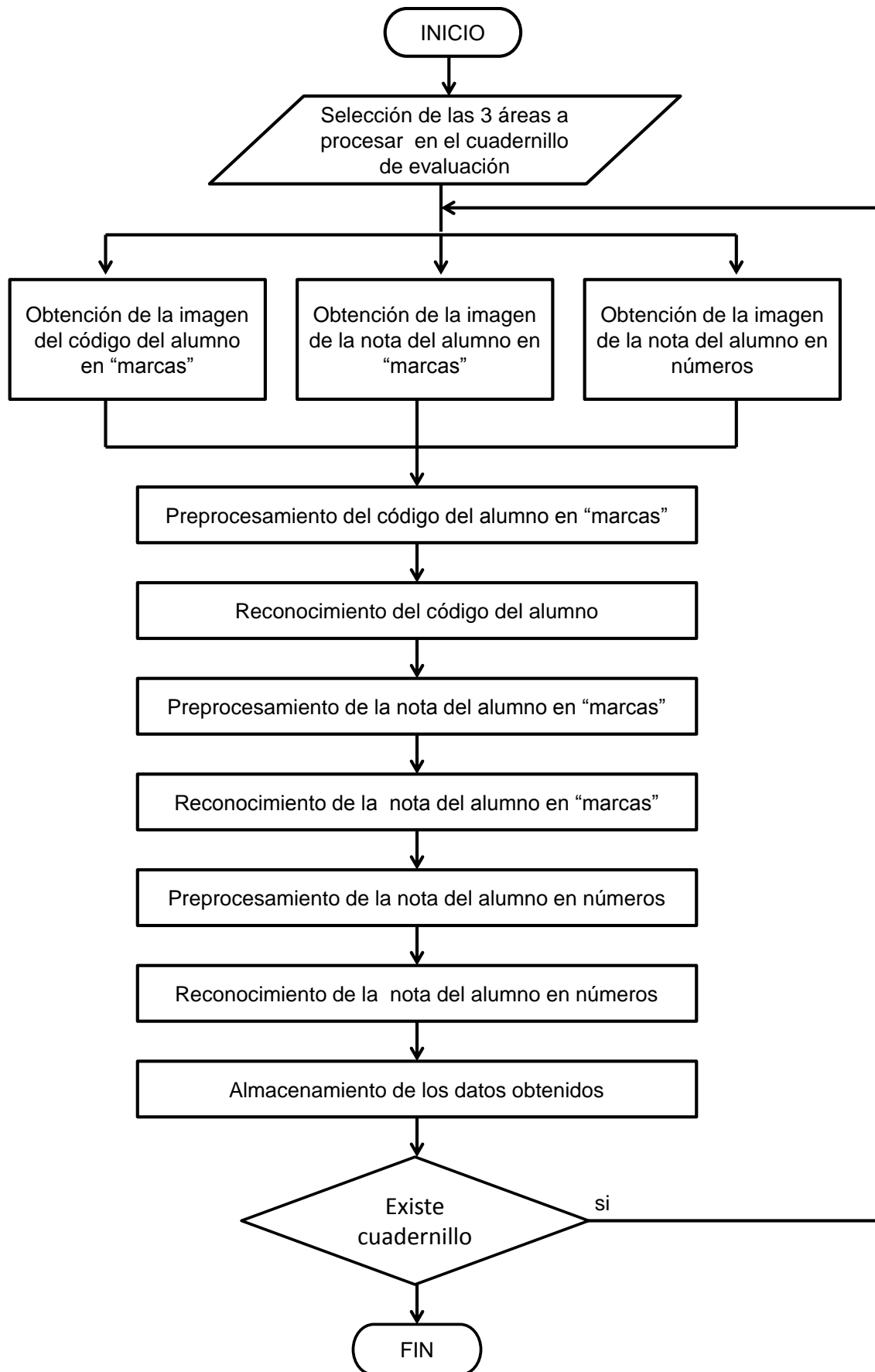


Figura 3.5 Diagrama de flujo del sistema general

3.3 Adquisición de la imagen

La imagen que se obtiene, proviene de una cámara Web Microsoft LifeCam Studio (en formato RGB), la imagen tiene una resolución de 720 x 1280 (filas x columnas) píxeles. Sus características se muestran en la Figura 3.6.



Principales Características

- Sensor CMOS de alta definición 1080p HD.
- Resolución del sensor: 1920x1080.
- 75 ° Campo de visión diagonal.
- Bandeja digital, inclinación digital y zoom digital 3x.
- Enfoque automático de 0,1 m a ≥ 10 m.
- Ajuste de imagen Automático con control manual.
- Hasta 30 cuadros por segundo.

Figura 3.6 Características de la cámara web utilizada

3.4 Descripción del cuadernillo de evaluación

Uno de los objetivos que se planteó en el presente trabajo es desarrollar una nueva carátula del cuadernillo de evaluación que se ajuste a los requerimientos particulares necesarios para la detección. Se procuró que esta nueva propuesta tenga los menores cambios posibles con respecto al cuadernillo usado, contiene los mismos campos, el mismo tamaño, etc. Por este motivo se tuvo bastante interacción con el coordinador académico de la Facultad de Ingeniería y Arquitectura de la Universidad de San Martín de Porres para que evaluara las propuestas. En la Figura 3.7 se muestra el cuadernillo actual que se utiliza para las distintas evaluaciones.

USMP
UNIVERSIDAD
SAN MARTÍN DE PORRES

FACULTAD DE INGENIERÍA Y ARQUITECTURA

APELLIDOS: _____
 NOMBRES: _____
 CÓDIGO: _____
 SECCIÓN: _____ AULA: _____
 TURNO: _____

PRACTICA CALIFICADA N°

ESCUELA PROFESIONAL:

<input type="checkbox"/> 01 COMP. Y SIST.	<input type="checkbox"/> 03 INDUSTRIAL	<input type="checkbox"/> 10 ARQUITECTURA
<input type="checkbox"/> 02 ELECTRÓNICA	<input type="checkbox"/> 07 CIVIL	<input type="checkbox"/> 11 IND. ALIMENTARIAS
		<input type="checkbox"/> 14 AERONÁUTICA

NOTA: Marcar con una x la escuela a la que pertenece

Firma
Nes.

ASIGNATURA: _____ CICLO: _____ FECHA: / / _____

SEMESTRE ACADÉMICO: 20 ____ - ____

(Llevar sólo hasta esta línea)

NOMBRE DEL PROFESOR QUE CALIFICÓ LA PRUEBA

En números:

En letras:

NOTA


ADVERTENCIAS ANTES DE INICIAR LA EVALUACIÓN

1. Utilice únicamente lapicero de tinta AZUL o NEGRA.
2. En la esquina superior derecha de la presente página, escriba los datos solicitados con letra impresa, coloque su rúbrica y el NÚMERO de orden que le asignan en la lista de asistencia.
3. No escriba su nombre, firma, marca o cualquier señal en otro lugar del cuadernillo; caso contrario se le calificará con CERO (00) de nota, sin derecho a reclamo.
4. Utilice la columna en blanco que aparece en el cuadernillo, denominada ZONA PARA CÁLCULOS a modo de borrador.
5. No use hojas sueltas ni arranque hojas del cuadernillo. Todo trabajo debe figurar en este cuadernillo, caso contrario se anulará la prueba.
6. Evite realizar borrones, pues las manchas, roturas o el uso de correctores líquidos, invalidarán su respuesta, así sea ésta correcta. La buena presentación influye en la calificación.
7. El examen calificado lo podrá recoger en las aulas de clase o según el cronograma de entrega de prácticas; caso contrario lo recogerá en Mesa de Partes sin lugar a reclamo.
8. Si hubiera algún reclamo, deberá realizarlo al momento de la entrega de prácticas en clase y/o de acuerdo al cronograma de entrega.
9. Para el caso de reclamo, el docente o la persona encargada recibirá el cuadernillo con los datos solicitados en la parte posterior del mismo, previamente llenados por el alumno, y lo derivará a la coordinación Académica, que posteriormente, y a través de Mesa de Partes, dará a conocer si el reclamo procede o no.
10. El incumplimiento de cualquier punto señalado anteriormente, influirá negativamente en su calificación.

ATENCIÓN A LA INDICACIÓN EN LA PARTE POSTERIOR

Figura 3.7 Cuadernillo de evaluación actual

En la Figura 3.8 se muestra la carátula del cuadernillo de evaluación diseñado.



USMP
UNIVERSIDAD
SAN MARTÍN DE PORRÉS

FACULTAD DE
INGENIERÍA Y ARQUITECTURA

EXAMEN FINAL PRÁCTICA CALIFICADA Nº

EXAMEN PARCIAL PRÁCTICA CALIFICADA Nº

ESCUELA PROFESIONAL:

01 SISTEMAS 03 INDUSTRIAL 10 ARQUITECTURA
 02 ELEC TRÓNICA 04 CIVIL 05 ALIMENTARIAS

NOTA: Marcar con una x la escuela a la que pertenece

ASIGNATURA:

SEMESTRE ACADÉMICO: 20__ -- __

FECHA: / /

C																				
0																				
1																				
2																				
3																				
4																				
5																				
6																				
7																				
8																				
9																				

Ap:

Nom:

Sec: Turno:

Aula:

Firma

Nro.

Nota en Números

--	--

Nota en Letras

--

00	01	02	03	04	05	06	07	08	09	10
N	11	12	13	14	15	16	17	18	19	20

NOMBRE DEL PROFESOR QUE
CALIFICÓ LA PRUEBA

INDICACIONES PARA LA EVALUACIÓN

ATENCIÓN A LA INDICACIÓN EN LA PARTE POSTERIOR

Figura 3.8 Cuadernillo de evaluación diseñado

3.5 Selección de las áreas de procesamiento

Al cargar el sistema se seleccionan tres (3) áreas de procesamiento o de interés. De esta manera, se concentra el procesamiento en las áreas específicas. Dichas áreas son: “nota del alumno en marcas”, “código del alumno en marcas” y “nota del alumno en números manuscritos”.

Esta selección solo se realiza una vez al inicio del proceso y se guardan las posiciones de dichas áreas. El cuadernillo de evaluación donde se encuentran las áreas de interés se muestran en la Figura 3.10.

En la Figura 3.9, se muestra la cuadrícula donde el alumno coloca su código. En la fila superior, el alumno coloca en números su código que está compuesto de 10 dígitos y sombrea cada marca blanca que corresponde según el dígito. En la Figura 3.11, se muestra un ejemplo de esto:

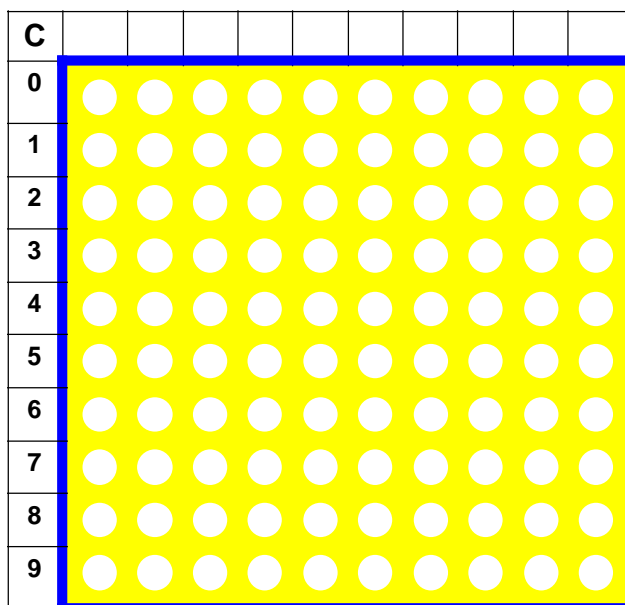


Figura 3.9. Cuadrícula para colocar el código del alumno en marcas

Nota del alumno en números **Código del alumno en "marca"**

USMP FACULTAD DE INGENIERÍA Y ARQUITECTURA

EXAMEN FINAL EXAMEN PARCIAL PRÁCTICA CALIFICADA N°

ESCUELA PROFESIONAL:

01 SISTEMAS 03 INDUSTRIAL 10 ARQUITECTURA
 02 ELÉCTRICA 04 CIVIL 05 ALIMENTARIAS

NOTA: Marcar con una x la escuela a la que pertenece

ASIGNATURA:

SEMESTRE ACADÉMICO: 20 ____ / ____ FECHA: / /

Ap: Nom: Sec: Turno: Aula:

Firma No. ○

Nota en Números

Nota en Letras

NOMBRE DEL PROFESOR QUE CALIFICÓ LA PRUEBA

INDICACIONES PARA LA EVALUACIÓN

ATENCIÓN A LA INDICACIÓN EN LA PARTE POSTERIOR

Nota del alumno en "marca"

Figura 3.10 Áreas de interés del cuadernillo a procesar

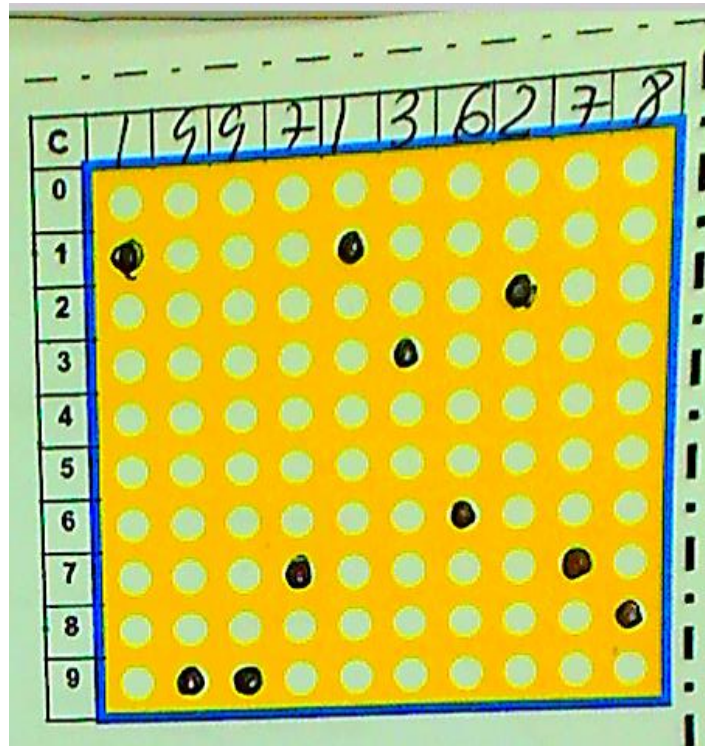


Figura 3.11 Ejemplo del código del alumno

00	01	02	03	04	05	06	07	08	09	10
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
N	11	12	13	14	15	16	17	18	19	20
<input type="checkbox"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Figura 3.12 Cuadrícula para colocar la nota del alumno en marcas

En la Figura 3.12, se muestra la cuadrícula donde el docente que califica el examen coloca la nota en “marca” correspondiente. En este caso, solo se sombrea la marca blanca que corresponde según la nota. Cabe resaltar que la nota será un número entero. En la Figura 3.13, se muestra un ejemplo de esto.

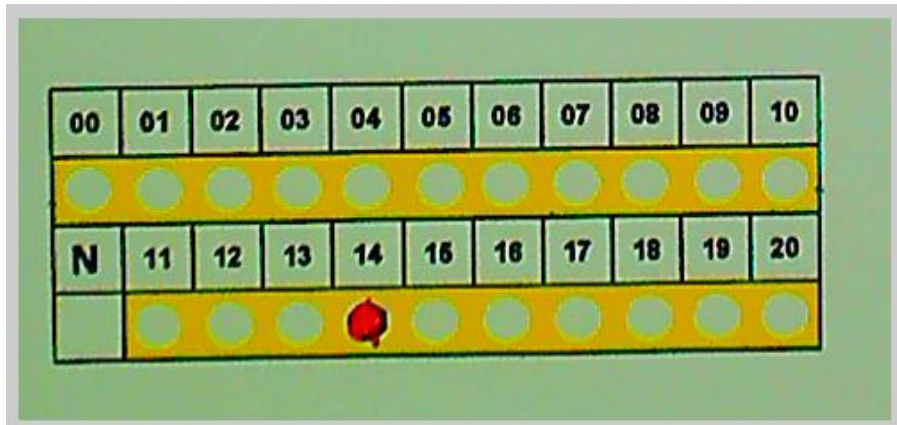


Figura 3.13 Ejemplo de la nota del alumno

En la Figura 3.14, se muestra la cuadrícula donde se coloca la nota del alumno en números manuscritos. Algo importante que resaltar, como se indicó al inicio, es que los números son escritos sobre una cuadrícula acondicionada. Esto permitió hacer menos cambiante al número y permitió caracterizarlo mejor. El número debe estar escrito sobre el “ocho” de color amarillo. Un ejemplo de esto se muestra en la Figura 3.15

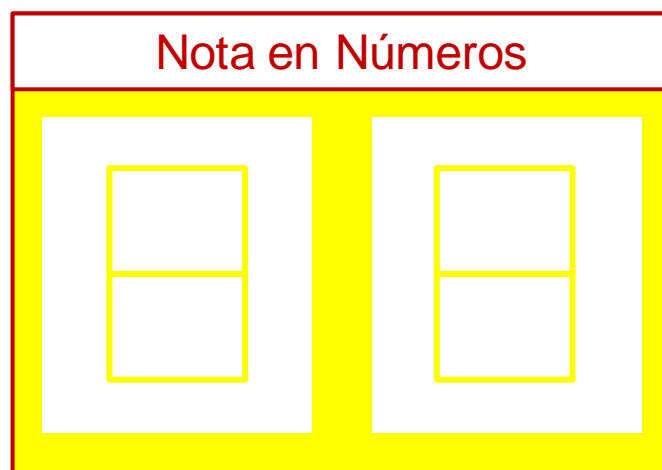


Figura 3.14 Cuadrícula donde se escribe los números

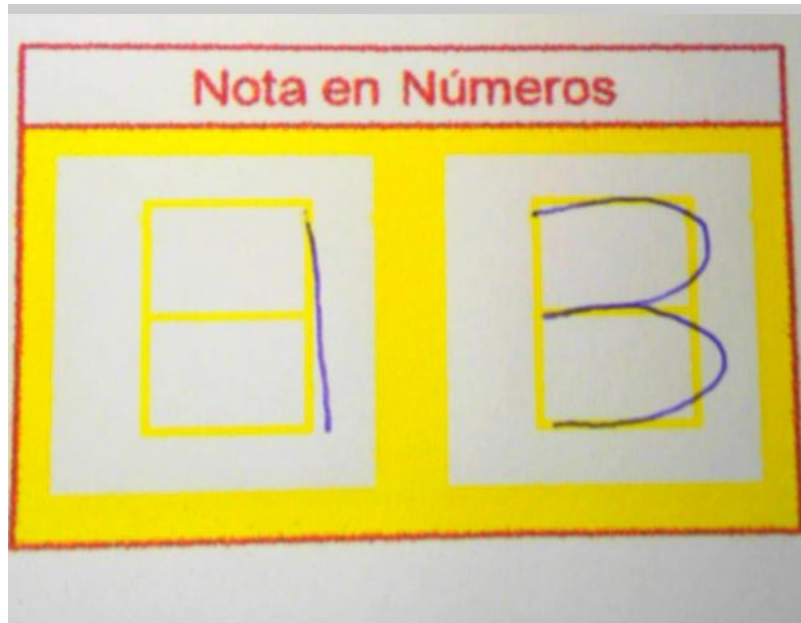


Figura 3.15 Ejemplo del número manuscrito

Cabe resaltar que tanto las marcas como los números manuscritos se pueden sombrear y escribir, respectivamente, con lapiceros de tinta azul, negro o rojo.

3.6 Pre procesamiento

En primer lugar, se va a describir el pre procesamiento que se realiza a las áreas de interés, “código del alumno en marcas” y “nota del alumno en marcas”, y posteriormente, se describirá el pre procesamiento de la “nota del alumno en números manuscritos”.

3.6.1 Pre procesamiento del código del alumno en marcas

La imagen del área de interés “código del alumno en marcas” se denotada como I_1 .

La etapa de pre procesamiento de la imagen del área de interés, “código del alumno en marcas” se procede aplicando la ecuación (3.1), siendo “ I_{1BD} ”, la imagen binarizada resultante. Además “ I_{1R} ”, “ I_{1G} ” y “ I_{1B} ” las imágenes componentes Rojo, Verde y Azul del formato de color RGB de la imagen I_1 respectivamente.

$$I_{1BD}(x, y) = \begin{cases} 255 & , \quad I_{1R}(x, y) \geq \mu_0 \wedge I_{1G}(x, y) \geq \mu_1 \wedge I_{1B}(x, y) \geq \mu_2 \\ 0 & , \quad \text{caso contrario} \end{cases} \quad (3.1)$$

Es importante resaltar que como las áreas son de color amarillo, el uso de niveles adecuados facilitará la extracción del área. Debido a las distintas pruebas realizadas, se encontraron los siguientes niveles adecuados $\mu_0 = 75$, $\mu_1 = 65$ y $\mu_2 = 0$.

El resultado de la conversión se muestra en la Figura 3.16

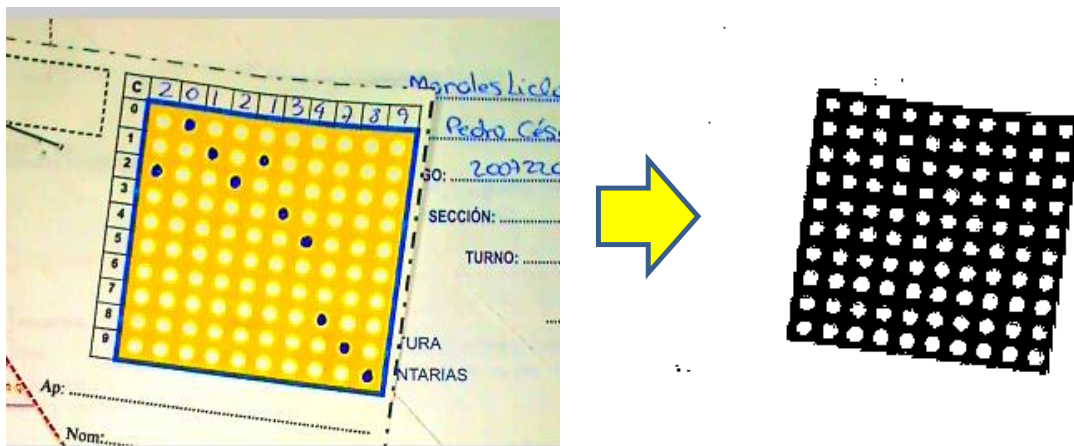


Figura 3.16 Imagen en formato RGB del área “código del alumno en marca” y su imagen binarizada resultante

A continuación, a la imagen I_{1BD} se le aplicará el algoritmo de etiquetado. Hay que tener en cuenta que dicho proceso tomará como etiqueta a los píxeles que son de color blanco, por lo tanto, en dicha imagen existen 101 etiquetas. Estas serán las 100 marcas y el fondo. La imagen que contiene las etiquetas será I_{1BE} y se muestra en la Figura 3.17

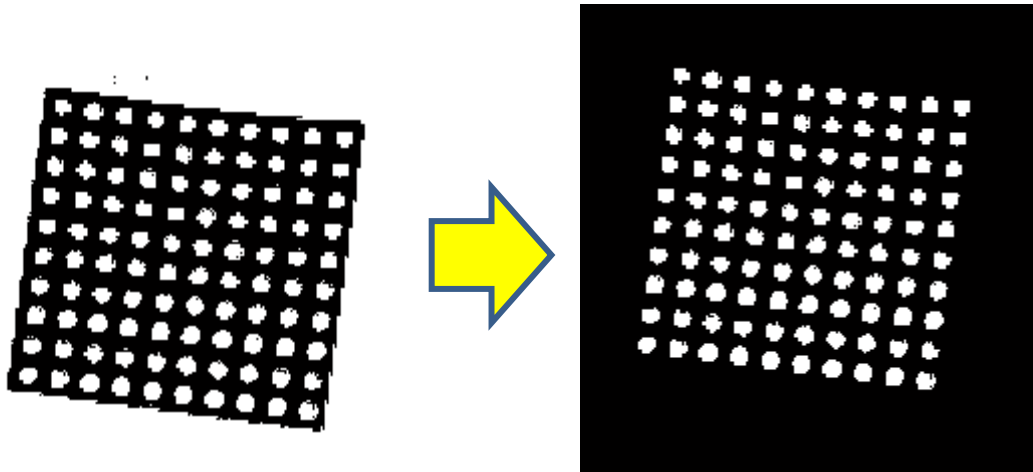


Figura 3.17 Imagen resultante del proceso de etiquetado

Como se puede apreciar en la Figura 3.17 aparecen todas las “marcas” indistintamente de que si se han sombreado o no.

Las “marcas” sombreadas, se obtendrán de la imagen I_{1R} (componente R de la imagen “código del alumno en marcas”), en donde sobresalen de las no sombreadas (Figura 3.18). Para este proceso se aplica la ecuación (3.2), siendo el valor de $\mu_3 = 171$.

$$I_{1RB}(x, y) = \begin{cases} 0 & , I_{1R}(I_{1BE}(x, y)) \geq \mu_3 \\ 255 & , I_{1R}(I_{1BE}(x, y)) < \mu_3 \end{cases} \quad (3.2)$$

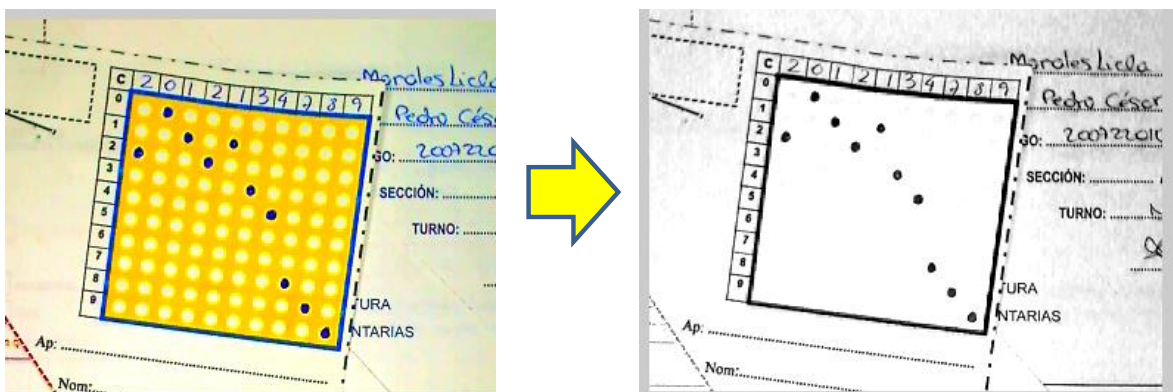


Figura 3.18 Componente “R” de la imagen del área de código del alumno en marcas

Como se puede observar en la ecuación, este proceso se aplica solo sobre las posiciones de las etiquetas de la imagen I_{1BE} . El resultado que se obtiene será una

imagen binaria que no contenga las marcas sombreadas. Esto se aprecia en las Figuras 3.19.

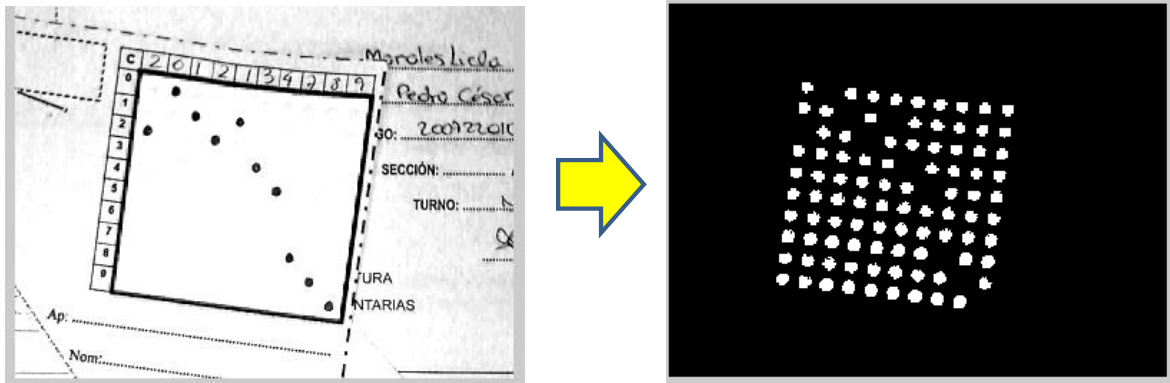


Figura 3.19 Imagen I_{1RB} resultante

Una vez obtenido la imagen I_{1BE} y I_{1RB} se aplica la operación lógica OR-EXCLUSIVA dando como resultado la imagen I_{1BC}

La imagen resultante se aprecia en la Figura 3.20

$$I_{1BC} = I_{1BE} \oplus I_{1RB} \quad (3.3)$$

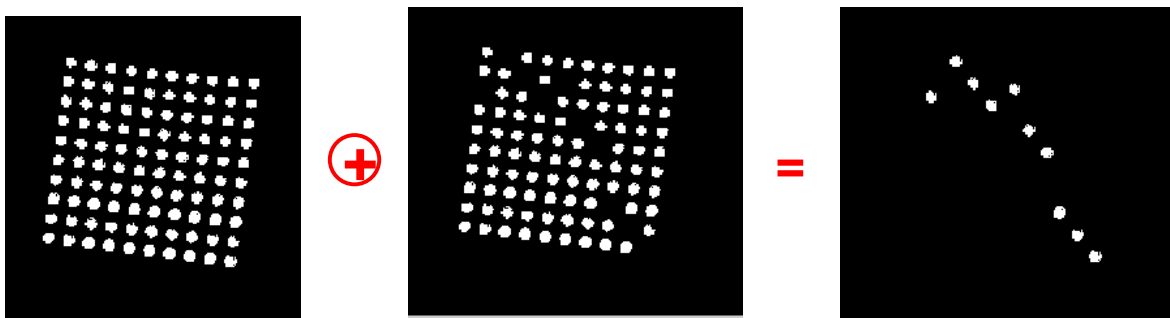


Figura 3.20 Proceso de obtención de las “Marcas” seleccionadas

Cada etiqueta que conforma la imagen I_{1BC} , es I_{1BC_k} y será la “k”-ésima etiqueta de la imagen, que correspondería a cada una de las “marcas” que formar el código. La posición (centroide) de cada una de estas “k” etiquetas se obtienen con las ecuaciones 3.7 y 3.8.

La matriz PET1 contendrá las posiciones de todas las etiquetas (“marcas”) que contiene la imagen I_{1BC} .

$$M_k(0,0) = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} I_{1BC_k}(x,y) \quad (3.4)$$

$$M_k(1,0) = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} x \cdot I_{1BC_k}(x,y) \quad (3.5)$$

$$M_k(0,1) = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} y \cdot I_{1BC_k}(x,y) \quad (3.6)$$

$$x_k = M_k(1,0) / M_k(0,0) \quad (3.7)$$

$$y_k = M_k(0,1) / M_k(0,0) \quad (3.8)$$

$$PET1C(k,1) = x_k \quad (3.9)$$

$$PET1C(k,2) = y_k$$

El descriptor que se utilizará para reconocer cada “marca seleccionada” será la distancia euclidiana (ecuación 3.10) que existe entre cada una de estas “marcas” seleccionadas con los 4 extremos que conforma el grupo de “marcas” totales como se puede apreciar en la Figura 3.21.

$$DC1_{nk} = \sqrt{(x_k - x_n)^2 + (y_k - y_n)^2} \quad (3.10)$$

Donde:

x_k, y_k : centro de gravedad de cada marca seleccionada

x_n, y_n : centro de gravedad de cada marca de referencia

$k = 1, 2, 3, \dots, 10$

$n = 1, 2, 3, 4$

Esto quiere decir que cada “marca” tendrá cuatro (04) distancias que lo caracterizan. Se llamara DC1 a la matriz que contenga las distancias del código del alumno.

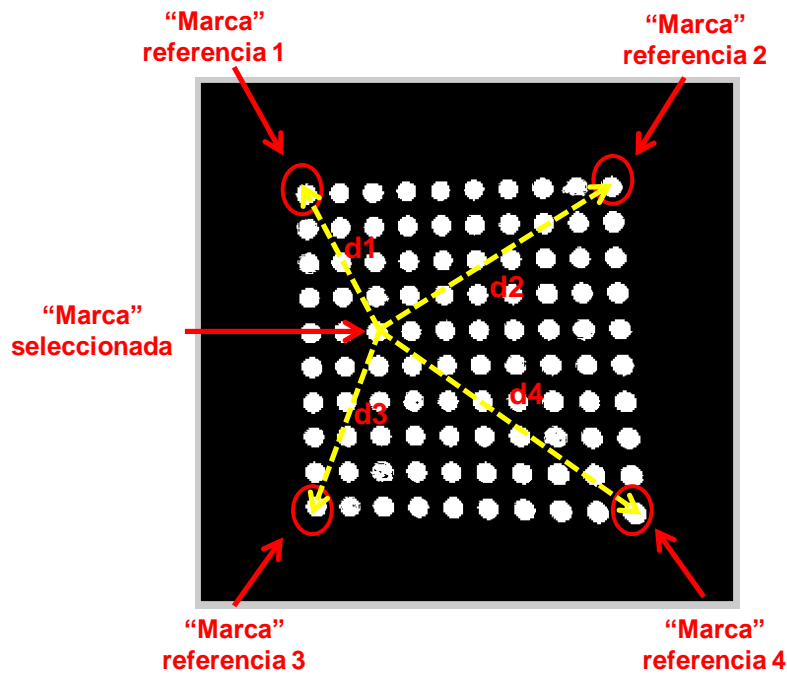


Figura 3.21 Distancias de cada "marca"

La matriz DC1 es una matriz de 4x10, debido a que son 10 marcas las que se han sombreado y cada una de estas con sus cuatro distancias correspondientes; y este será el descriptor que permitirá el reconocimiento del código del alumno.

3.6.2 Pre procesamiento de la nota del alumno en marcas

La imagen del área de interés "nota del alumno en marcas"; se denotada como I_2 .

La etapa de pre procesamiento de la imagen del área de interés, "nota del alumno en marca" se procede aplicando la ecuación (3.11), siendo " I_{2BD} ", la imagen binarizada resultante y " I_{2R} ", " I_{2G} " y " I_{2B} " las imágenes componentes Rojo, Verde y Azul del formato de color RGB de la imagen I_2 respectivamente.

$$I_{2BD}(x, y) = \begin{cases} 255 & , \quad I_{2R}(x, y) \geq \mu_0 \wedge I_{2G}(x, y) \geq \mu_1 \wedge I_{2B}(x, y) \geq \mu_2 \\ 0 & , \quad \text{caso contrario} \end{cases} \quad (3.11)$$

Es importante resaltar que como las áreas son de color amarillo facilitará la extracción del área usando niveles adecuados. Debido a las distintas pruebas realizadas se encontraron los siguientes niveles adecuados $\mu_0 = 75$, $\mu_1 = 65$ y $\mu_2 = 0$.

El resultado de la conversión se muestra en la Figura 3.22

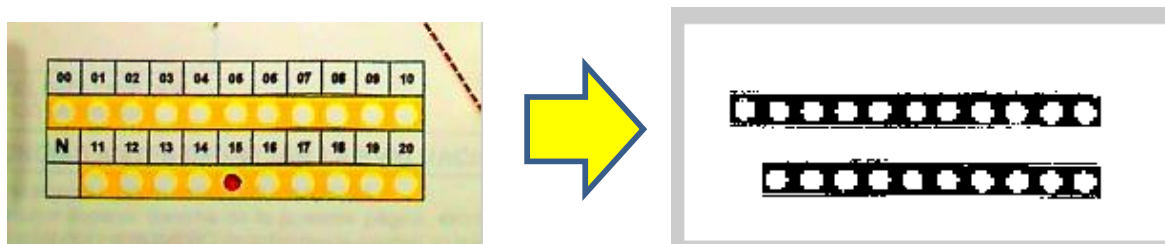


Figura 3.22 Imagen en formato RGB del área “nota del alumno en marca” y su imagen binarizada resultante

A la imagen I_{2BD} , se aplica el proceso de dilatación para poder darle a la imagen mayor consistencia. La imagen resultante será I_{2BDD} y se muestra en la Figura 3.23

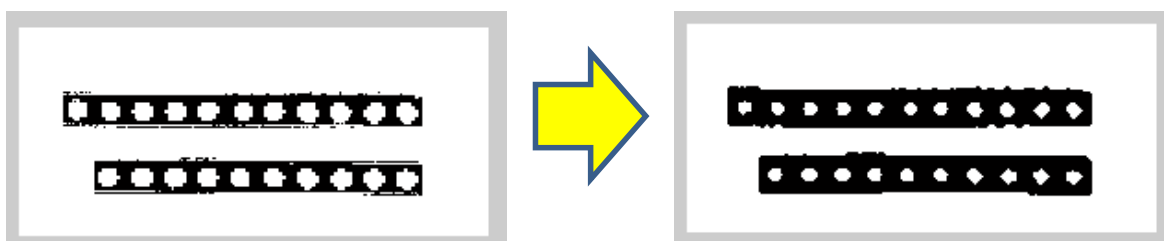


Figura 3.23 Imagen resultante del proceso de dilatación

A continuación, a la imagen I_{2BDD} se le aplicará el algoritmo de etiquetado. Hay que tener en cuenta que dicho proceso tomará como etiqueta a los pixeles que son de color blanco, por lo tanto, en dicha imagen existen 22 etiquetas. Estas serán las 21 marcas y el fondo. La imagen que contiene las etiquetas será I_{2BDE} y se muestra en la Figura 3.24.

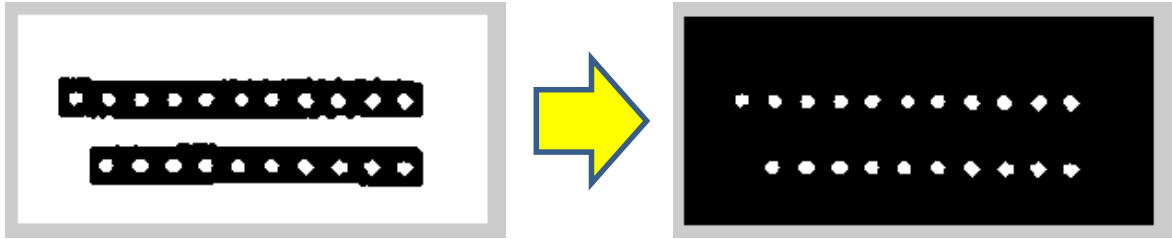


Figura 3.24 Imagen resultante

Como se puede apreciar en las Figura 3.24 aparecen todas las “marcas” indistintamente de que si se han sombreado o no.

Las “marcas” sombreadas, se obtendrán de la imagen I_{2R} , en donde sobresalen de las no sombreadas (Figura 3.25). Para este proceso se aplica la ecuación (3.12), siendo el valor de $\mu_4 = 171$.

$$I_{2RB}(x, y) = \begin{cases} 0 & , I_{2R}(I_{2BDE}(x, y)) \geq \mu_4 \\ 255 & , I_{2R}(I_{2BDE}(x, y)) < \mu_4 \end{cases} \quad (3.12)$$

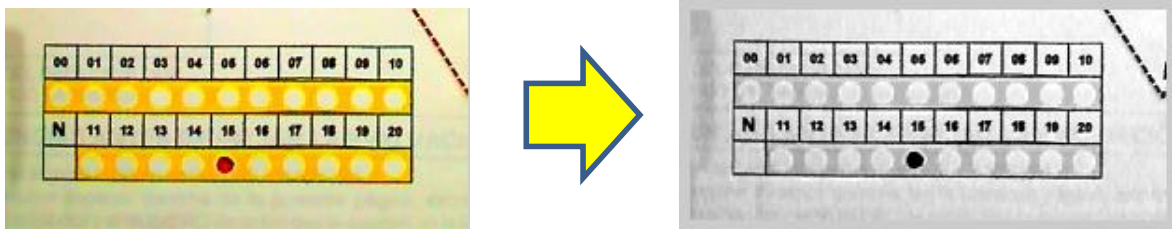


Figura 3.25 Componente “R” de la imagen del área nota del alumno en marcas

Como se puede observar en la ecuación, este proceso se aplica solo sobre las posiciones de las etiquetas de la imagen I_{2BDE} . El resultado que se obtiene será una imagen binaria que no contenga las marcas sombreadas. Esto se aprecia en las figuras 3.26

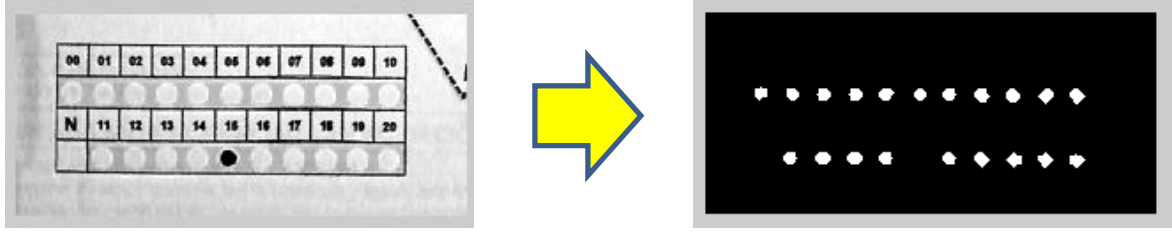


Figura 3.26 Imagen MS2 resultante

Una vez obtenido la imagen I_{2BDE} y I_{2RB} se aplica la operación lógica OR EXCLUSIVA dando como resultado la imagen I_{2BN}

La imagen resultante se aprecia en la Figura 3.27

$$I_{2BN} = I_{2BDE} \oplus I_{2RB} \quad (3.13)$$

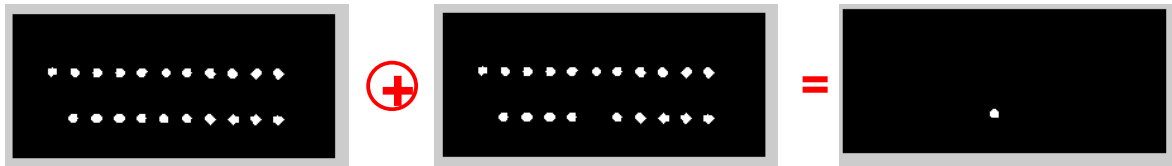


Figura 3.27: Proceso de obtención de las “Marcas” seleccionadas

La etiqueta I_{2BN_1} será la etiqueta que correspondería a la “marca” que indica la nota. La posición (centroide) de esta etiqueta se obtiene con las ecuaciones 3.17 y 3.18

La matriz PET2 contendrá las posiciones de esta etiqueta (“marcas”).

$$M_1(0,0) = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} I_{2BN_1}(x, y) \quad (3.14)$$

$$M_1(1,0) = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} x \cdot I_{2BN_1}(x, y) \quad (3.15)$$

$$M_1(0,1) = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} y \cdot I_{2BN_1}(x, y) \quad (3.16)$$

$$x_1 = M_1(1,0) / M_1(0,0) \quad (3.17)$$

$$y_1 = M_1(0,1) / M_1(0,0) \quad (3.18)$$

$$\begin{aligned} PET2N(1,1) &= x_1 \\ PET2N(1,2) &= y_1 \end{aligned} \quad (3.19)$$

El descriptor que se utilizará para reconocer cada “marca seleccionada” será la distancia euclidiana (ecuación 3.20) que existe entre esta “marca” seleccionada con los 4 extremos que conforma el grupo de “marcas” totales como se puede apreciar en la Figura 3.28.

$$DN1_{n1} = \sqrt{(x_1 - x_n)^2 + (y_1 - y_n)^2} \quad (3.20)$$

Donde:

x_1, y_1 : centro de gravedad de la marca seleccionada

x_n, y_n : centro de gravedad de cada marca de referencia

$n = 1, 2, 3, 4$

Esto quiere decir que la “marca” tendrá cuatro (04) distancias que lo caracterizan. Se llamara DN1 a la matriz que contenga las distancias de la nota del alumno en marcas.

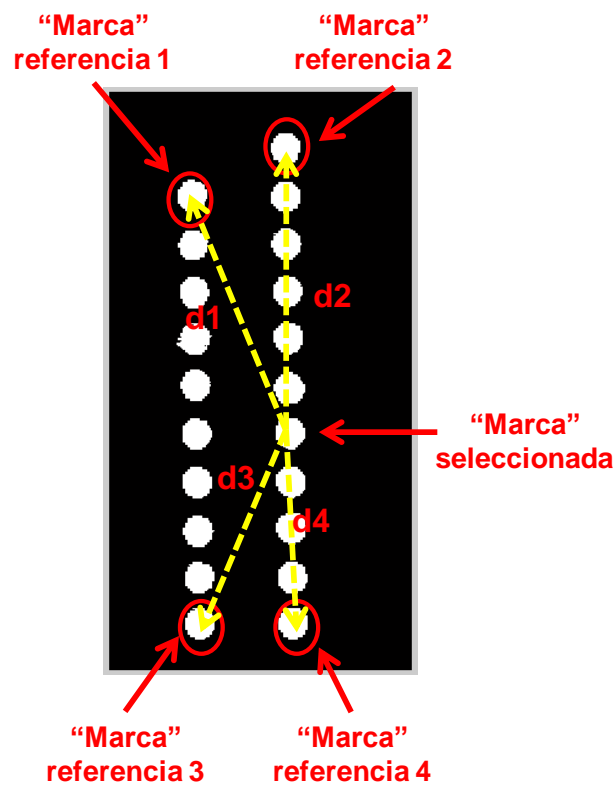


Figura 3.28 Distancias de cada “marca”

La matriz DN1 es una matriz de 4x1, debido a que solo es una marca la que se ha sombreado y esta con sus cuatro distancias correspondientes; y este será el descriptor que permitirá el reconocimiento de la nota del alumno en “marcas”.

3.6.3 Pre procesamiento de la nota del alumno en números

Ahora se va a describir el pre procesamiento del área de interés “nota del alumno en números”; dicha imagen está denotada como I_3 .

La etapa de pre procesamiento de la imagen del área de interés, “nota del alumno en números” se procede aplicando la ecuación (3.21), siendo “ I_{3BD} ”, la imagen binarizada resultante y “ I_{3G} ” la imagen componente Verde del formato de color RGB de la imagen I_3 .

$$I_{3BD}(x, y) = \begin{cases} 255 & , \quad I_{3G}(x, y) \geq \mu_7 \\ 0 & , \quad I_{3G}(x, y) < \mu_7 \end{cases} \quad (3.21)$$

Debido a las distintas pruebas realizadas se encontró el siguiente nivel adecuado $\mu_7 = 75$.

Por otra parte, se busca obtener las áreas donde se encuentran los números para poder extraerlos para su procesamiento posteriormente. Para esto, se procede a binarizar la imagen RGB aplicando la ecuación (3.22), aprovechando que el área que rodea a los números está con borde de color amarillo. El resultado obtenido se muestra en la figura 3.29.

$$I_{AP}(x, y) = \begin{cases} 0 & , \quad I_{3R}(x, y) \geq \mu_0 \wedge I_{3G}(x, y) \geq \mu_1 \wedge I_{3B}(x, y) \geq \mu_2 \\ 255 & , \quad \text{caso contrario} \end{cases} \quad (3.22)$$

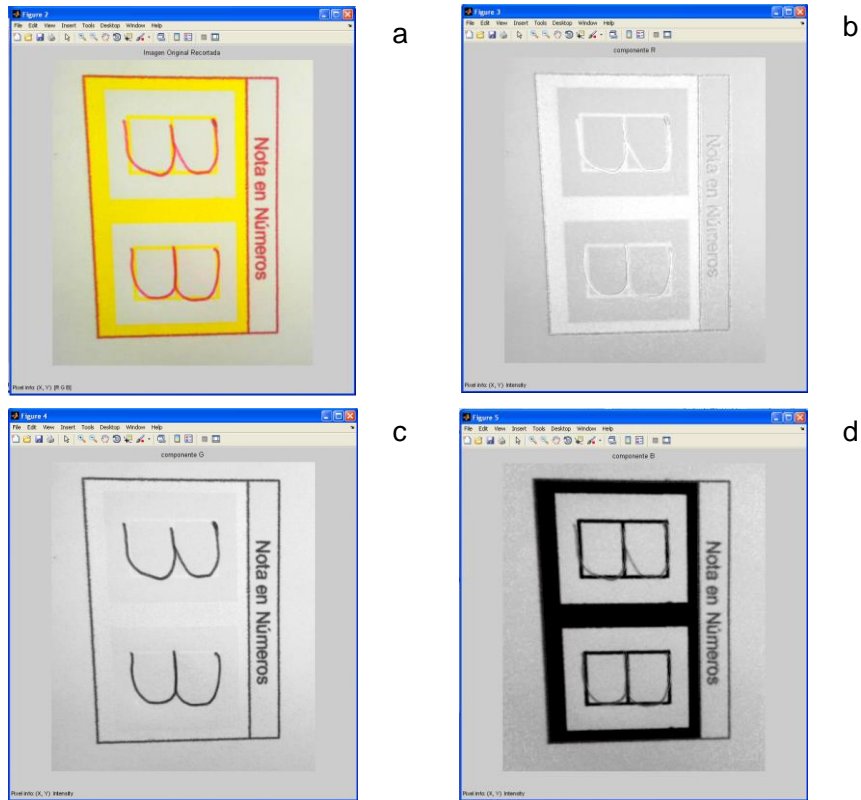


Figura 3.29 a) Imagen RGB, b) Componente R, c) Componente G, d) componente B

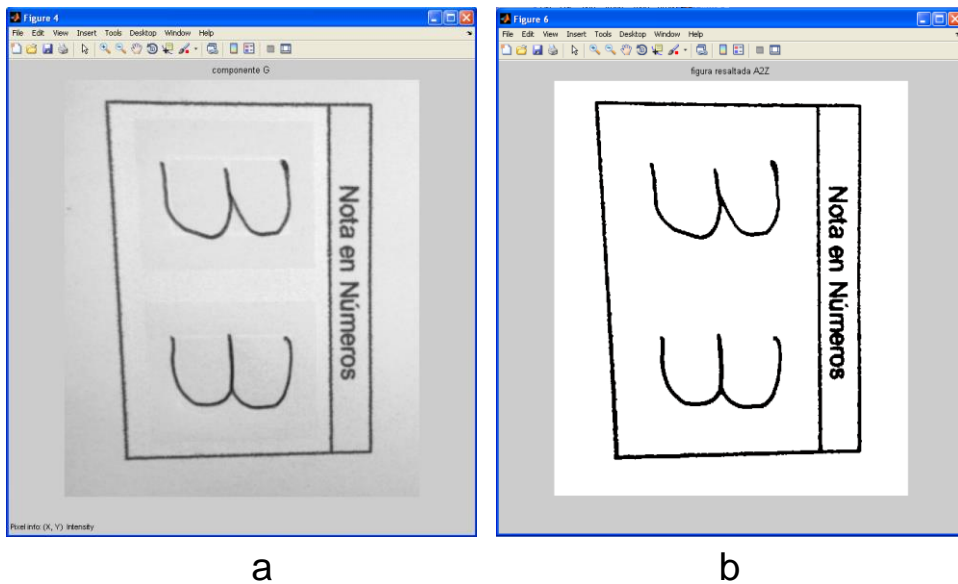


Figura 3.30 a) Componente G, b) Imagen binarizada obtenida de la componente G

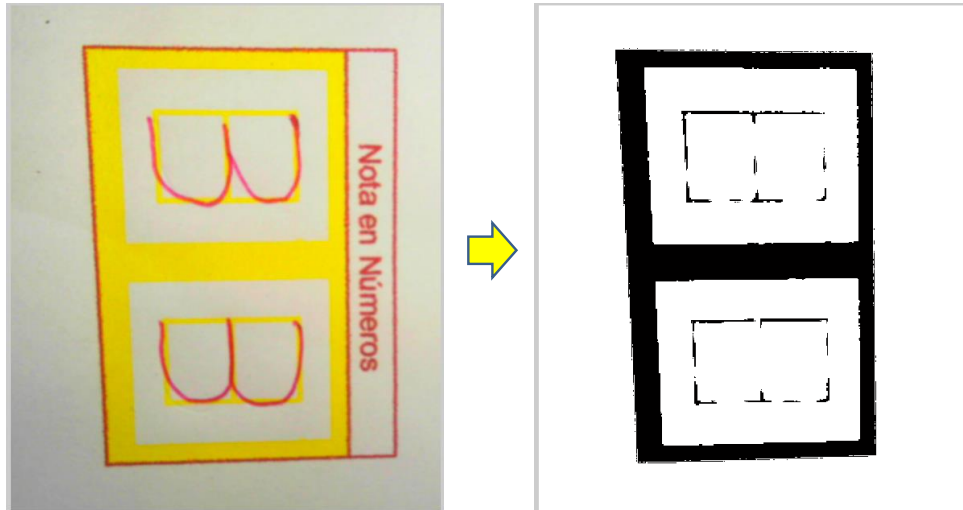


Figura 3.31 Imagen binarizada del área donde se encuentran los números

Sobre la imagen I_{AP} obtenida, se aplica un primer proceso de etiquetado teniendo como resultado la imagen que se muestra en la Figura 3.32b y a esta nuevamente se le vuelve a aplicar el proceso de etiquetado teniendo como resultado la Figura 3.32c. Esta última imagen se guarda como imagen I_{AP2} . Como se puede apreciar en esta imagen binaria se tienen las dos áreas sobre donde están los números escritos. Por lo tanto se aplica sobre esta nuevamente el proceso de etiquetado y solo se tendrá dos objetos, que representan las áreas de interés que se necesita.

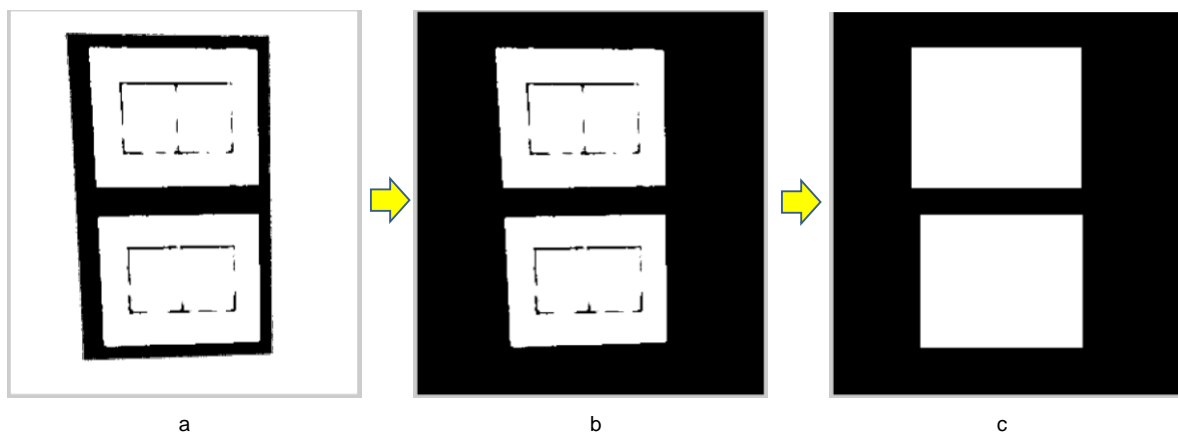


Figura 3.32 Imagen binarizada del área donde se encuentran los números

Conociendo las píxeles que ocupan estas áreas de interés. Se extrae de la imagen I_{3BD} los objetos, teniendo como resultado lo que se muestra en la Figura 3.33a y 3.33b. Como se puede apreciar en dichas Figuras ya se tiene solo a los números para

poder procesarlos. El primer dígito se almacenara como imagen I_{3XB1} y al segundo dígito I_{3XB2} .

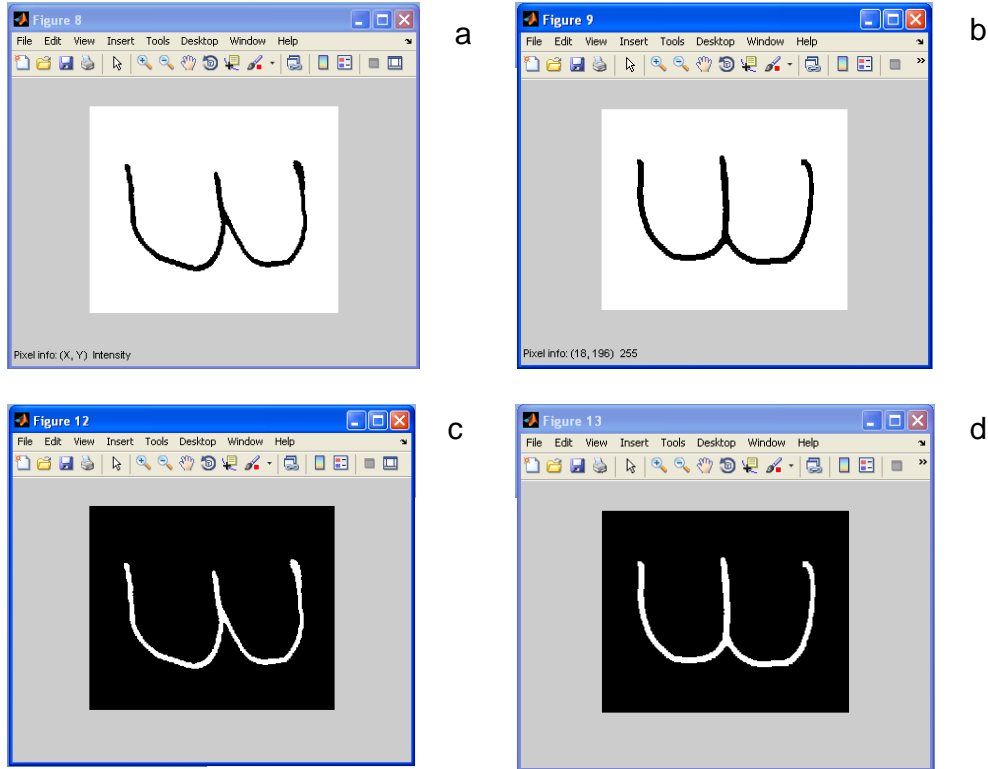


Figura 3.33 Extracción de los números a procesar. a) I_{3XB1} , b) I_{3XB2} , c) I_{3XBN1} , d) I_{3XBN2}

Posteriormente a cada imagen I_{3XB1} y I_{3XB2} se le aplica la operación NOT teniendo como resultado las Figuras 3.33c y 3.33d.

$$\begin{aligned} I_{3XBN1}(x, y) &= NOT(I_{3XB1}(x, y)) \\ I_{3XBN2}(x, y) &= NOT(I_{3XB2}(x, y)) \end{aligned} \quad (3.23)$$

Los procesos que se realizan posteriormente son por igual a ambos dígitos. Por lo tanto, aquí se va a proceder explicar el procedimiento que se realiza a un dígito, la imagen I_{3XBN1} .

Teniendo el dígito binarizado como lo muestra la figura 3.34a, se procedió a rotarlo. El resultado de la operación se muestra en la figura 3.34b.

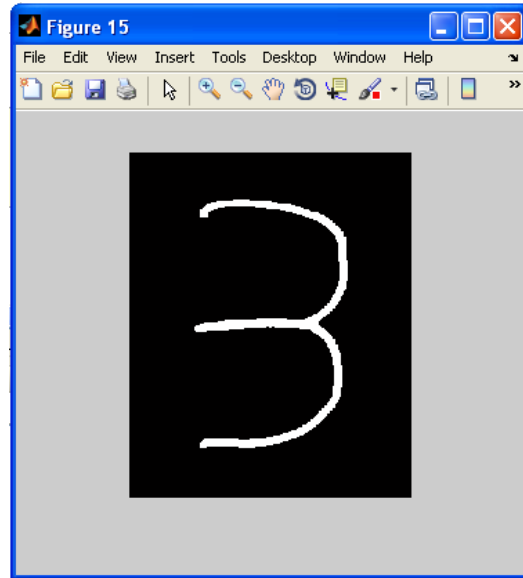
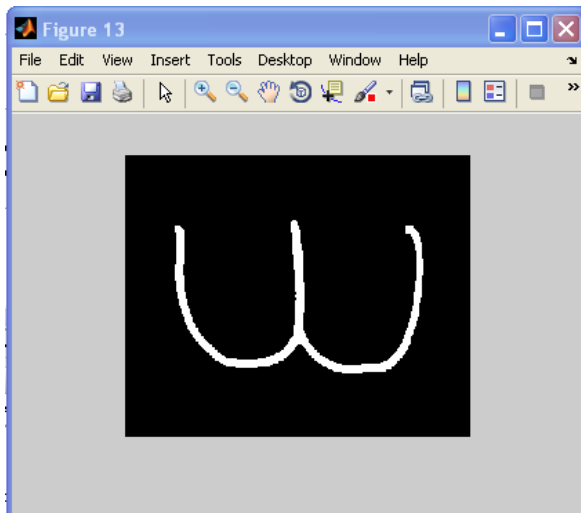


Figura 3.34 a) Imagen sin rotar, b) Imagen rotada

$$I_{3XBR1}(x, y) = I_{3XB1}(x \cdot \cos \theta - y \cdot \text{sen} \theta, x \cdot \text{sen} \theta + y \cdot \cos \theta) \quad (3.24)$$

A continuación se procede a corregir la posible inclinación que pudiera tener el dígito. Esto se realiza para poder normalizar el dígito. El imagen resultante de este proceso se muestra en la Figura 3.35b y la imagen resultante se denota por I_{3XBC1} .

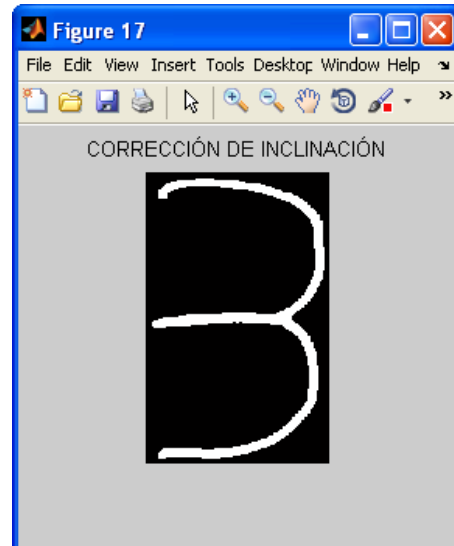
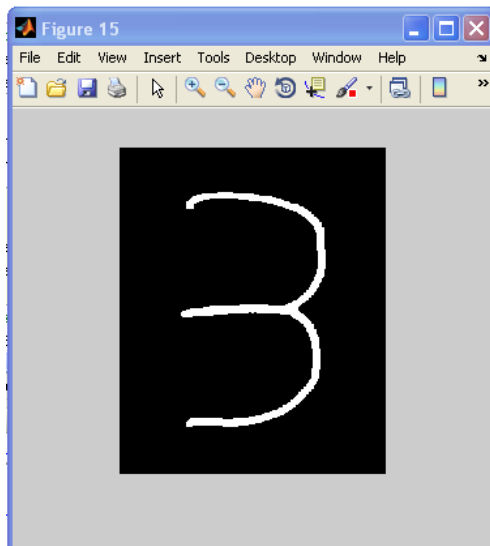


Figura 3.35 Imagen de la derecha con corrección de inclinación.

Posteriormente se dilata la imagen para poder “fijar” la imagen debido a que por un trazo tenue la imagen pueda partirse. Esto se realiza con una máscara I_M de 3x3 que se muestra en la Figura 3.36c.

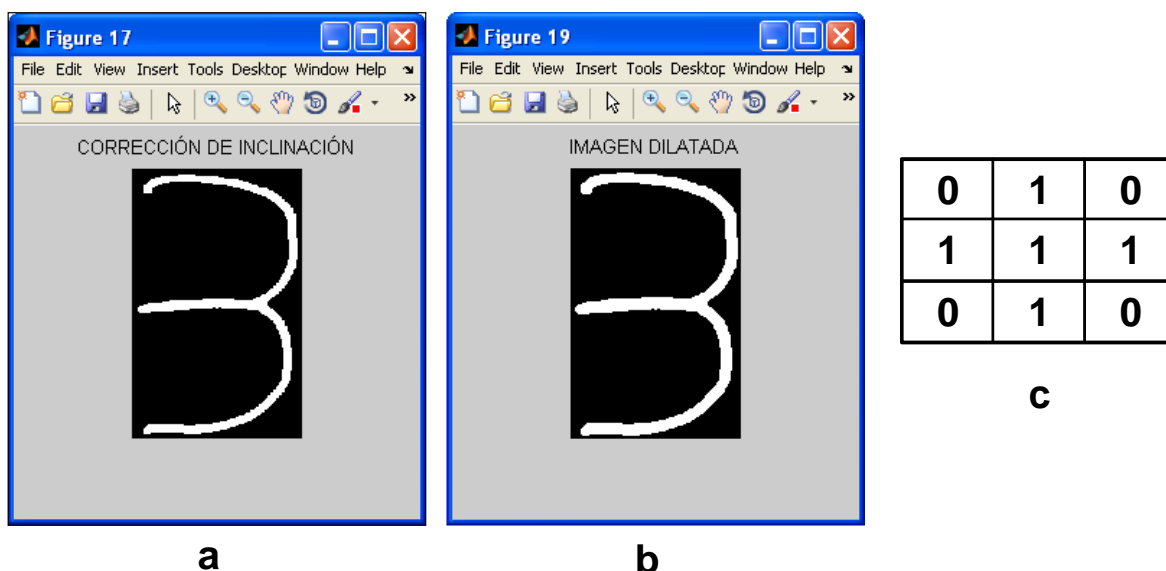


Figura 3.36 b) Imagen dilatada, c) máscara que se utilizó para el proceso de dilatación

$$I_{3XBD1} = \bigcup_{b_i \in I_M} I_{3XBC1_{b_i}} \quad (3.25)$$

Posteriormente para poder uniformizar el grosor de la imagen, se procede primero a esqueletizar la imagen y posteriormente a dilatarla, esto permitirá lograr lo requerido. Es importante, que los números no varíen mucho, que sean lo más estándar posible para poder realizar un reconocimiento adecuado.

En la Figura 3.37a se puede apreciar a la imagen resultante de una primera dilatación, luego en la Figura 3.37b la imagen ha sido adelgazada, y posteriormente dilatada nuevamente como se ve en la Figura 3.37c. El resultado obtenido es de un grosor más uniforme. Este proceso se llamará proceso de normalización de grosor de la imagen. Esta imagen resultante se denota como I_{3F1} , y el segundo dígito se denota como I_{3F2} .

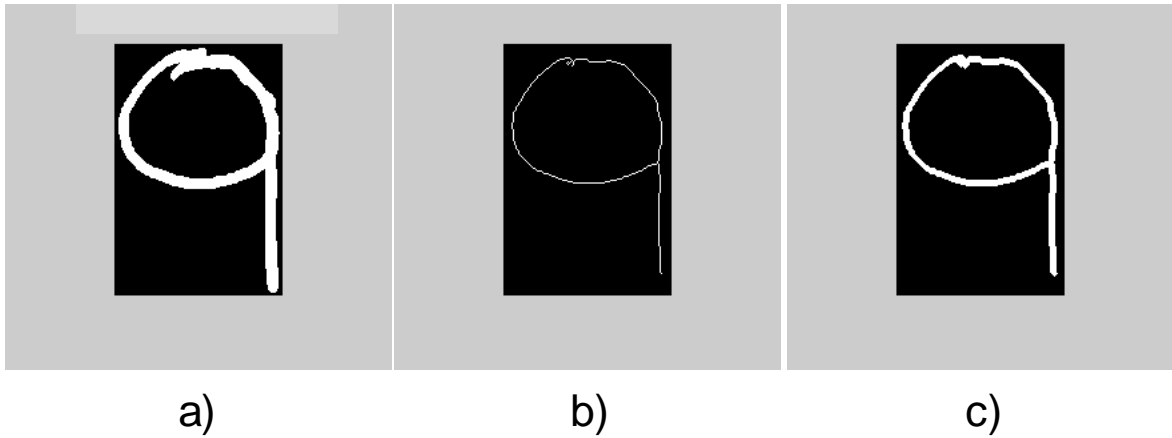


Figura 3.37. a) Imagen original, b) Imagen adelgazada a), c) Imagen resultante de la dilatación de b)

Ahora se procede a normalizar el tamaño de la imagen, para que todas las imágenes tengan el mismo tamaño y estandarizar más el número. Por lo tanto con la imagen final resultante del proceso anterior se procede a dividir dicha imagen en 35 áreas iguales (7 filas x 5 columnas) donde cada una de estas áreas contiene píxeles blancos y negros.

Si cada área contiene más del 15% de píxeles blancos entonces dicha área toma el valor de uno, caso contrario toma el valor de cero. Este proceso da origen a una nueva imagen como se puede apreciar en la Figura 3.38. Dicha imagen del dígito es mucho más uniforme y con esta se procede a ingresar al proceso de detección. Las imágenes para cada dígito serán NC1 y NC2.

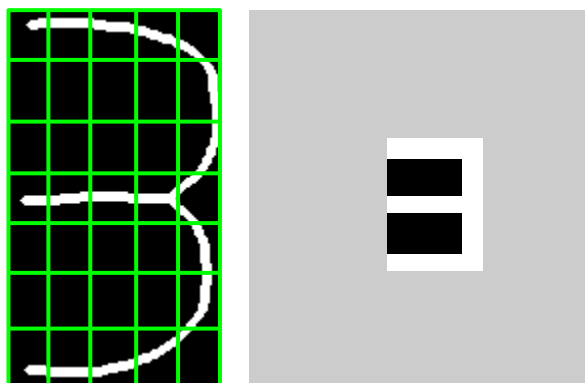


Figura 3.38. a) Imagen NF, b) imagen obtenida

3.7 Etapa de Reconocimiento

Para la etapa de reconocimiento tanto del código y nota del alumno en marcas como de los números en manuscritos se hizo uso de redes neuronales. Adicionalmente, para los números escritos a mano se utilizó correlación para tener una segunda manera de reconocimiento.

Así mismo para la implementación de la red neuronal se utilizó el toolbox de redes neuronales de MATLAB. El número de capas ocultas y el número de entradas y salidas de cada red se detallaran posteriormente.

Por lo general, una red neuronal se compone por tres tipos de capas. Capa de entrada, capa oculta y capa de salida. Si los datos son linealmente separables no hay necesidad de utilizar capa oculta. Pero en el caso de problemas que no son linealmente separables, dependiendo de la complejidad pueden tener una, dos y hasta tres capas ocultas. También, el número de neuronas que se debe tener en cada capa oculta puede ser calculado. Si el número de neuronas es menor en comparación con la complejidad de los datos del problema, puede producirse un "Underfitting", que no permite detectar adecuadamente las señales en un conjunto de datos complicados. Si existen más neuronas de las necesarias se puede producir un "Overfitting" que no permitirá también un adecuado reconocimiento. [17]

Por lo general, existen algunas reglas que se utilizan para determinar el número de neuronas en las capas ocultas. Estas reglas se indican a continuación:

El número de neuronas ocultas está relacionado con el número de neuronas de entrada [3]. En particular, suele aplicarse la regla 2×1 , de forma que el número de neuronas ocultas no puede ser superior al doble del número de variables de entrada.

El número de neuronas de la capa oculta es $2/3$ (o 70% a 90%) del número de neuronas de la capa de entrada. Si esto no es suficiente entonces se aumenta el número de neuronas de la capa de salida. [4]

Existe también la regla de la pirámide geométrica que se basa en la suposición de que el número de neuronas de la capa oculta ha de ser inferior al total de las variables de entrada, pero superior al número de variables de salida [20]. Se considera que el número

de neuronas en cada capa sigue una progresión geométrica, tal que, para una red de una única capa oculta, el número de neuronas intermedias debe ser próximo a $\sqrt{N \cdot M}$, siendo N el número de variables de entrada y M el total de neuronas de salida.

Si la red tuviera dos capas ocultas, el número de neuronas de la primera capa oculta debería ser próximo a $H_1 = M \cdot r^2$, y el número de neuronas de la segunda capa oculta debería ser próximo a $H_2 = M \cdot r$, siendo $r = \sqrt[3]{N/M}$.

Para el reconocimiento mediante el uso de redes neuronales se usó el algoritmo de entrenamiento de gradiente conjugado escalado "Scaled conjugate gradient", cuya función de entrenamiento es llamada "Trainscg" [7]. Un algoritmo que no requiere búsqueda lineal. Este algoritmo se utilizó para las tres redes que se implementó.

3.7.1 Etapa de detección de código del alumno en marcas

La red neuronal que se utilizó para la detección de la "marca" dentro de la cuadrícula código del alumno en marcas, se llamará RN1 y esta tiene una capa de entrada con 4 entradas, primera capa oculta con 30 neuronas, segunda capa oculta con 25 neuronas y una capa de salida con 20 salidas. En ambas capas ocultas se utilizó la función de transferencia sigmoideal.

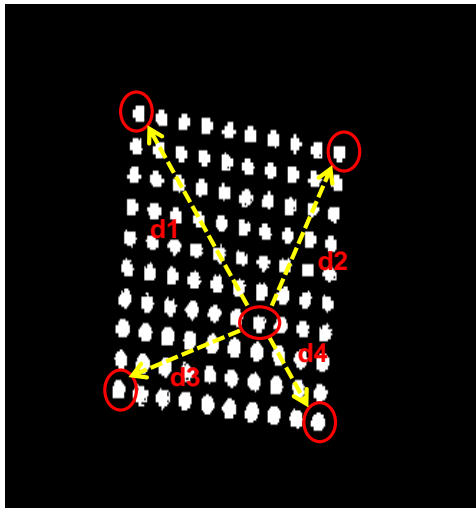
Las cuatro entradas serán las distancias con que se caracterizaron a cada marca y las 20 salidas indicarán en qué fila (los primeros 10 valores) y columna (los 10 valores restantes) se encuentra la marca detectada.

En la Figura 3.39 se muestra un ejemplo. Como se aprecia para los siguientes valores de entrada se obtiene que la marca detectada se encuentre en la fila 6 y columna 3.

El código de alumno está compuesto por 10 dígitos, por lo tanto, la cuadrícula del código del alumno que contiene 100 marcas, tendrá 10 de estas sombreadas como se muestra la Figura 3.11. Entonces, para el reconocimiento se necesita la matriz donde se encuentre las distancias de estas 10 marcas. Estas, se obtuvieron en la etapa de pre procesamiento y se guardaron en la matriz MP1. Por lo tanto, para obtener el código se realiza el siguiente procedimiento que se detalla en el diagrama de flujo que se muestra en la figura 3.40.

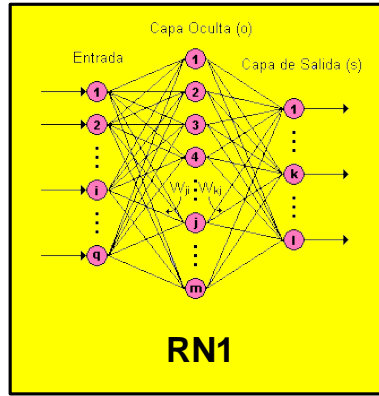
Como se muestra en el diagrama, el código del alumno está formado por C1, C2, C3, C4, C5, C6, C7, C8, C9, y C10 en ese orden. Por lo tanto, solo puede existir una marca por columna pero pueden existir varias marcas en una misma fila.

En caso que un cuadernillo no se haya marcado que el código del alumno la red devolverá el valor "0000000000".



a)

d1	0.33945
d2	0.39755
d3	0.62079
d4	0.68502



F I L A	1	0
	2	0
	3	0
	4	0
	5	0
	6	1
	7	0
	8	0
	9	0
	10	0
C O L U M N A	1	0
	2	0
	3	1
	4	0
	5	0
	6	0
	7	0
	8	0
	9	0
	10	0

b)

Figura 3.39 Ejemplo de reconocimiento de RN1

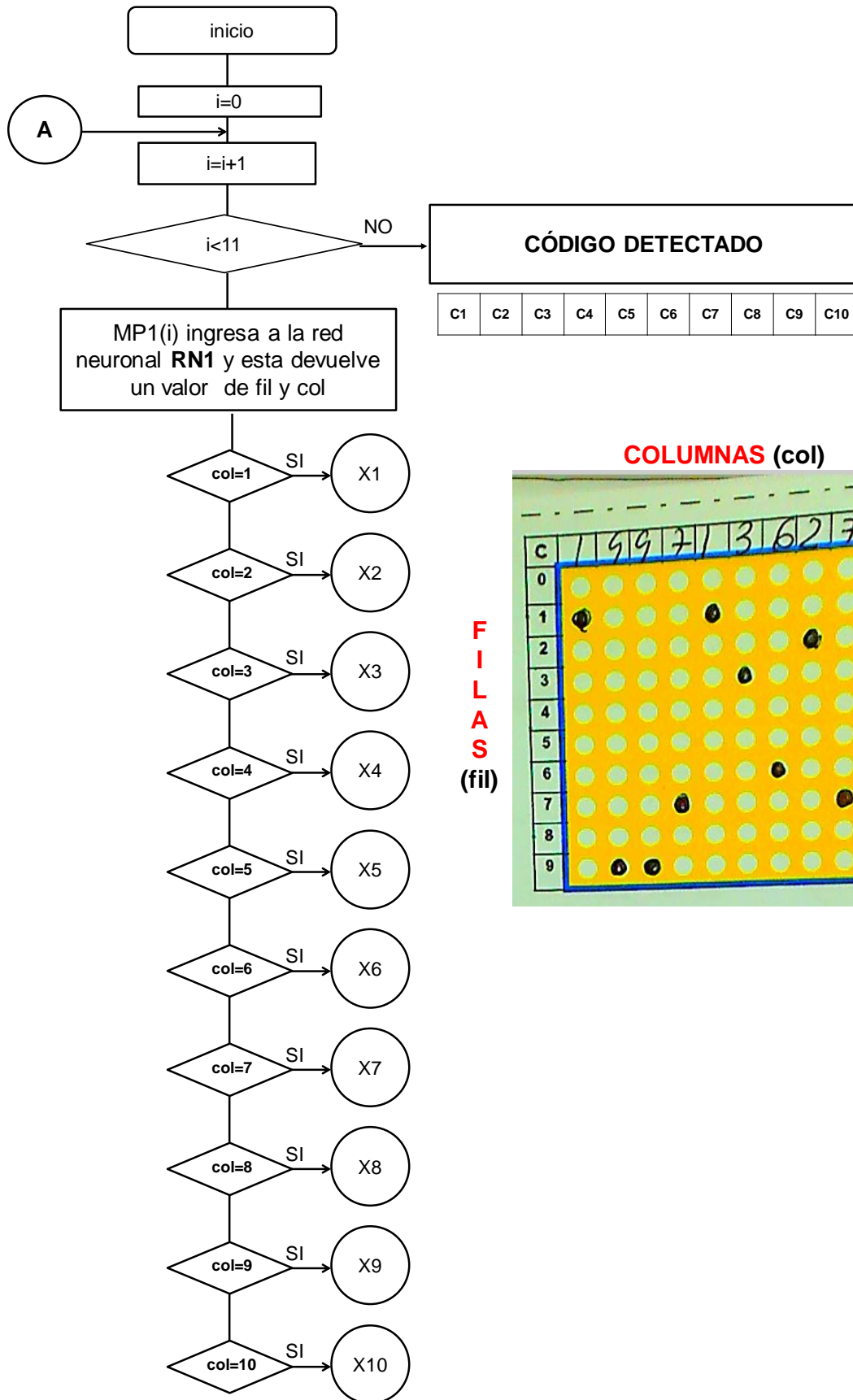


Figura 3.40 Diagrama de flujo de obtención de código del alumno

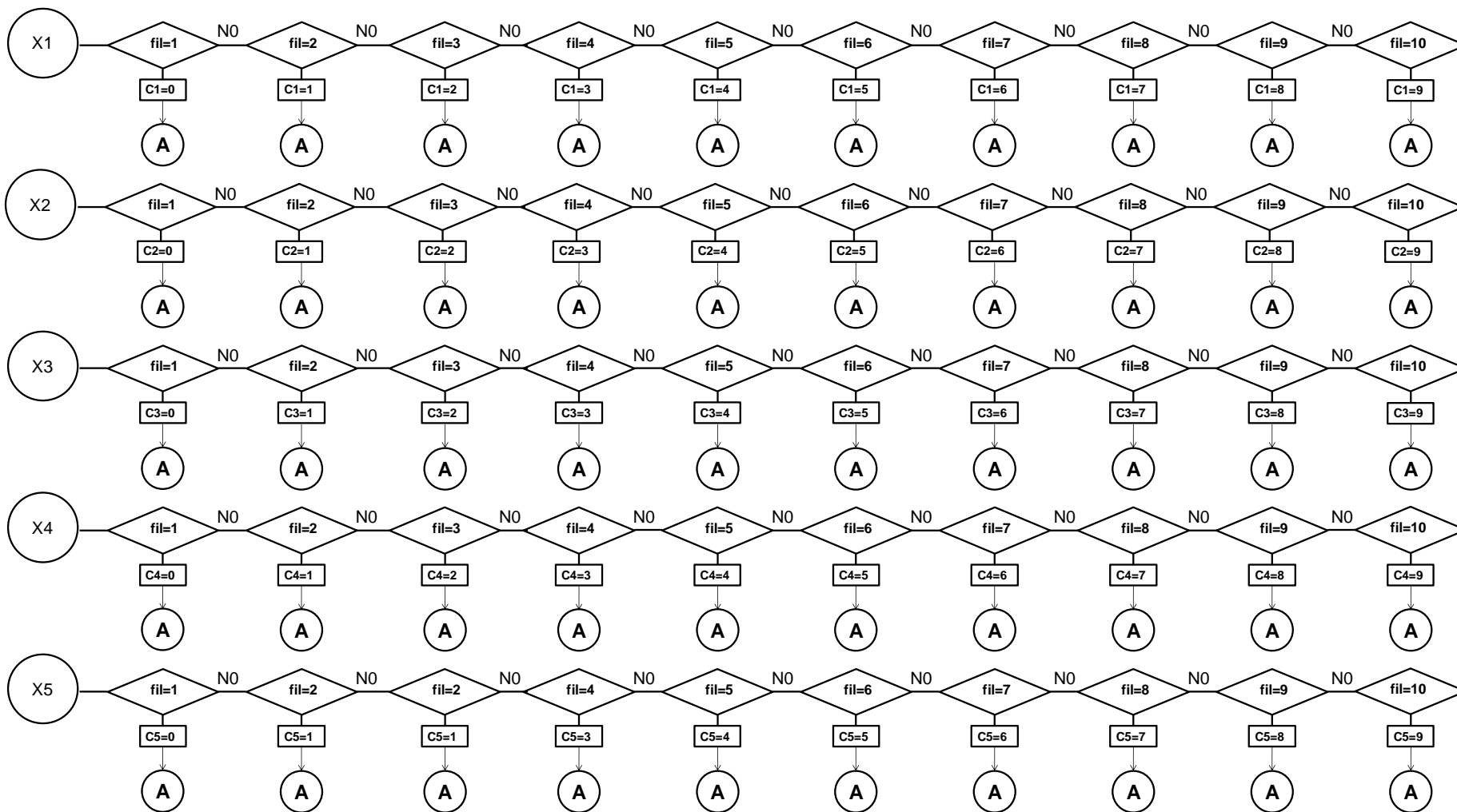


Figura 3.41 continuación Diagrama de flujo de obtención de código del alumno

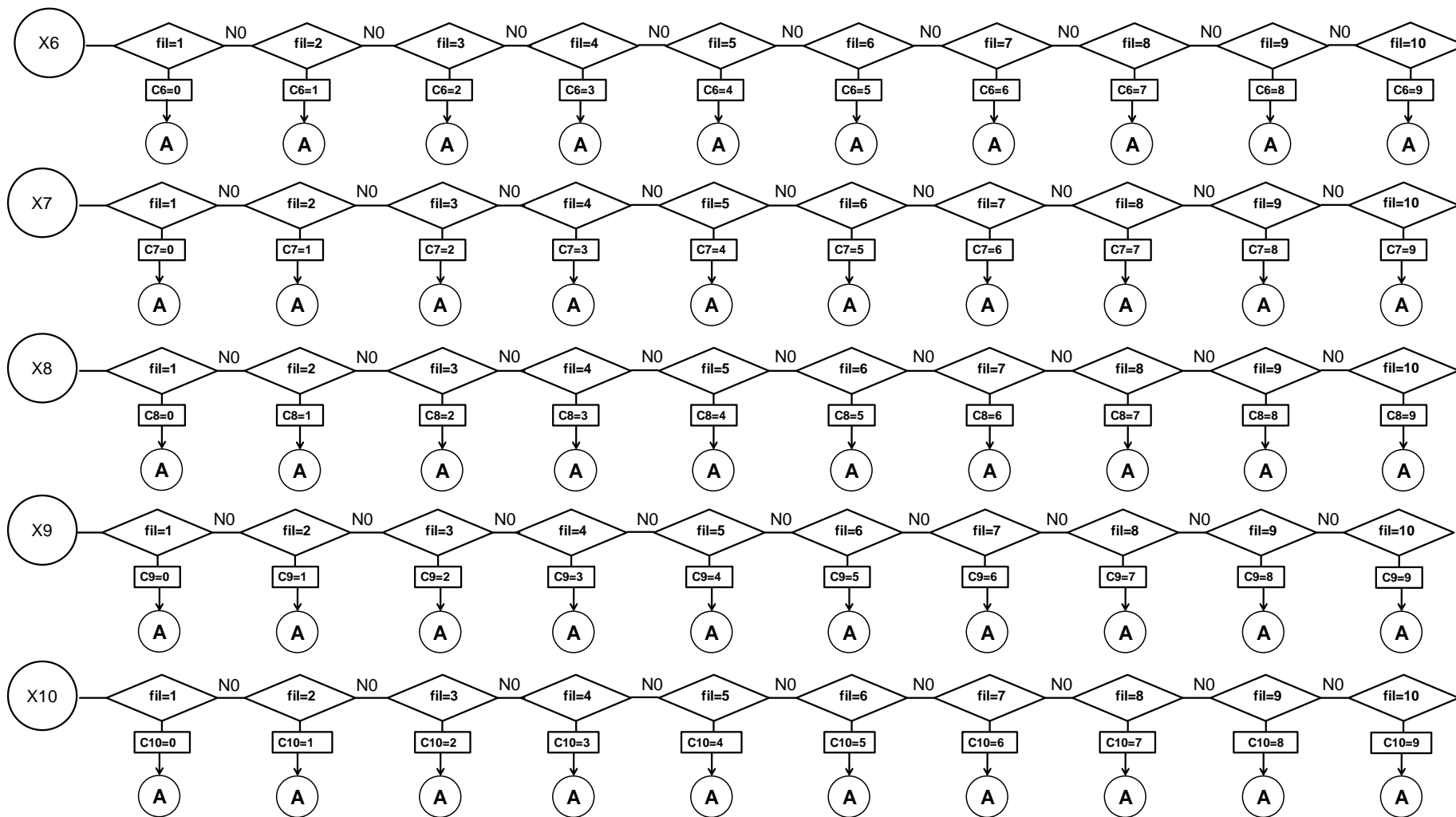


Figura 3.42 continuación Diagrama de flujo de obtención de código del alumno

3.7.2 Etapa de detección de la nota del alumno en marcas

La red neuronal que se utilizó para la detección de la “marca” dentro de la cuadrícula nota del alumno en marcas, se llamará RN2 y esta tiene una capa de entrada con 4 entradas, primera capa oculta con 35 neuronas, segunda capa oculta con 25 neuronas y una capa de salida con 21 salidas. Las cuatro entradas serán las distancias con que se caracterizó a la marca y las 21 salidas indicarán qué nota fue la que se marcó. En ambas capas ocultas se utilizó la función de transferencia sigmoideal.

En la Figura 3.44, se muestra un ejemplo. Como se aprecia para los siguientes valores de entrada se obtiene que la marca detectada sea 12.

La nota del alumno está compuesta por un solo valor, por lo tanto, la cuadrícula de nota del alumno que contiene 21 marcas, tendrá solo una de estas sombreada como se muestra la Figura 3.13. Entonces para el reconocimiento se necesita la matriz donde se encuentren las distancias de esta marca. Estas se obtuvieron en la etapa de pre procesamiento y se guardaron en la matriz MP2. Por lo tanto, para obtener la nota se realiza el siguiente procedimiento que se detalla en el diagrama de flujo que se muestra en la Figura 3.43.

En caso que un cuadernillo no se haya marcado que la nota del alumno la red devolverá el valor “99”.

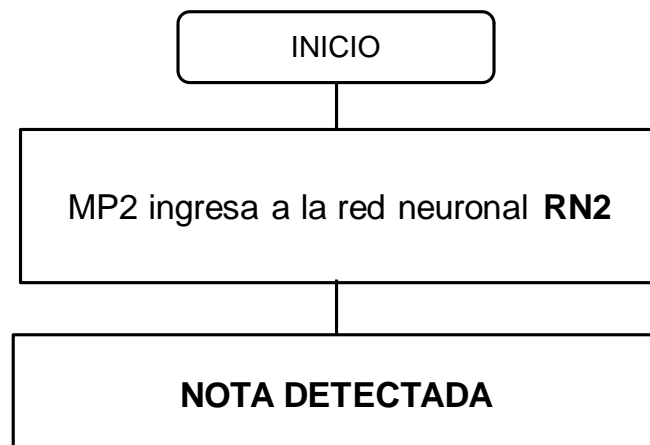
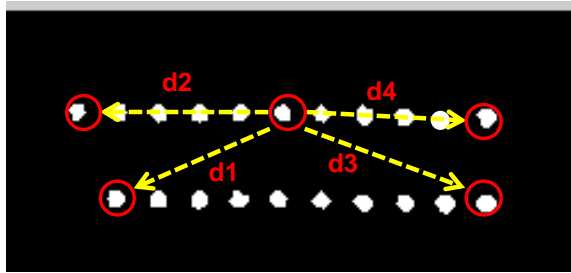
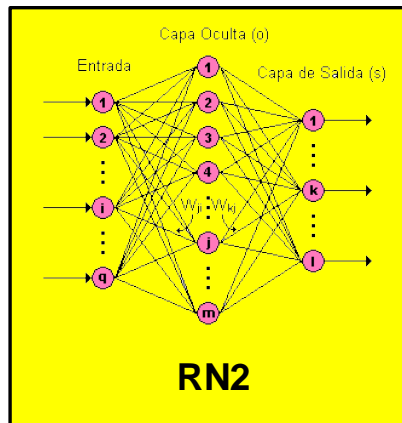


Figura 3.43 Diagrama de flujo de reconocimiento de la nota en marcas



a)

d1	0.4426
d2	0.4918
d3	0.5246
d4	0.4863



0	0
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	0
11	0
12	1
13	0
14	0
15	0
16	0
17	0
18	0
19	0
20	0

b)

Figura 3.44 Ejemplo de reconocimiento RN2

3.7.3 Etapa de detección de nota del alumno en números

Para la detección de la nota del alumno en números manuscritos se utilizaron redes neuronales y correlación.

Para poder ingresar las imágenes NC1 y NC2, obtenidas en la etapa de pre procesamiento a la red neuronal, se concatena cada columna que conforma la imagen desde la primera columna hasta la columna 5, formando de esta manera una matriz de una sola columna de 35 filas. Esta operación se realiza para los dos dígitos procesados y se obtiene la matriz ND1 para el primer dígito y ND2 para el segundo. Tiene dos capas ocultas de 37 y 8 neuronas respectivamente. En ambas capas ocultas se utilizó la función

de activación sigmoideal. En la tabla 3.1 se muestra el criterio que se utilizó para determinar el número de neuronas por cada capa oculta.

Número de neuronas en Capa oculta 1	Número de neuronas en Capa oculta 2	Número de épocas	Error cuadrático medio alcanzado	
35	5	638	0.00047055	
36	5	208	0.00075632	
37	5	624	0.00002206	
38	5	903	0.000036694	
39	5	795	0.000064541	
40	5	282	0.000035817	
41	5	1095	0.00071354	
42	5	745	0.00020951	
43	5	313	0.00020285	
44	5	872	0.00091625	
37	4	929	0.000054779	
37	8	848	0.000018864	Elegida
37	9	330	0.00027428	

Tabla 3.1 Prueba para escoger el número de neuronas adecuado para cada capa oculta.

En la Figura 3.45 se muestra la gráfica de número de épocas versus error cuadrático medio para los tres grupos de datos, con el número de neuronas en las capas ocultas escogido.

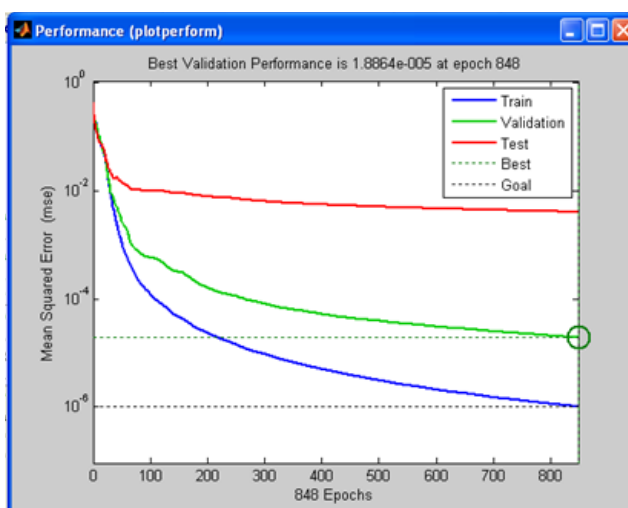


Figura 3.45 Número de época vs. Error cuadrático medio

En la Figura 3.46 se muestra la gráfica regresión de cada conjunto de datos

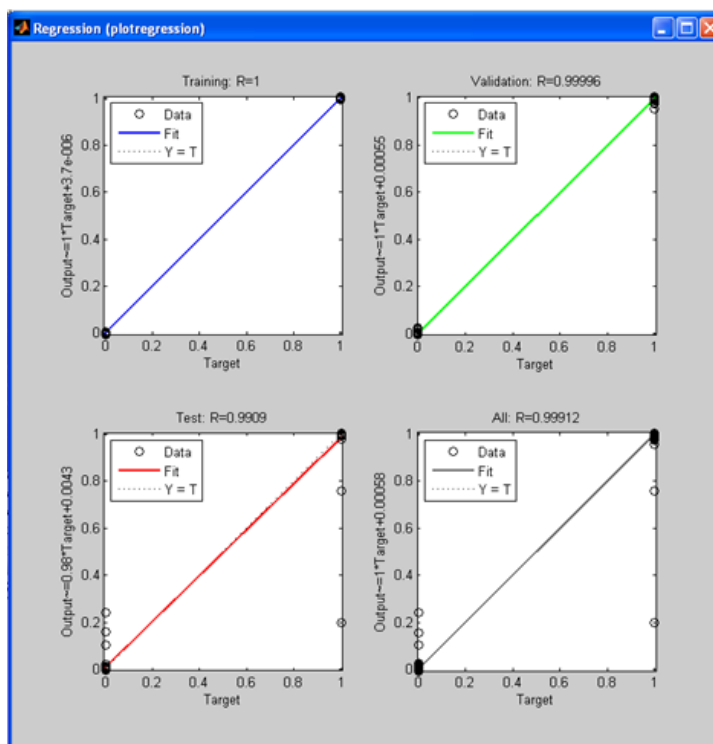


Figura 3.46 Gráfica de regresión de cada conjunto de datos

En caso que un cuadernillo no se haya escrito la nota del alumno la red devolverá el valor “99”.

3.7.3.1 Detección mediante correlación

Para la detección mediante el uso de la correlación se utilizaron unas plantillas que se correlacionan con las imágenes NC1 y NC2. Por lo tanto, se correlaciona la imagen NC1 con todas las plantillas de los dígitos I_{PK} y se elige el resultado r_k con mayor índice de correlación, que corresponderá al “k” número reconocido.

$$rk = \frac{\sum_m \sum_n (NC_{mn} - \overline{NC})(I_{Pkmm} - \overline{I_{Pk}})}{\sqrt{\left(\sum_m \sum_n (NC_{mn} - \overline{NC})^2 \right) \left(\sum_m \sum_n (I_{Pkmm} - \overline{I_{Pk}})^2 \right)}} \quad (3.26)$$

En donde: NC es la imagen base que se obtiene del pre procesamiento,

I_{PK} son los números a comparar, siendo el valor de $K=0, 1, 2, \dots, 9$ que solo dígitos a comparar.

\overline{NC} y $\overline{I_{PK}}$ son las medias de NC y I_{PK} respectivamente.

La misma operación se realiza con NC2. Después con los dos resultados obtenidos se forma la nota detectada conformada por los dos dígitos. En la figura 3.48, se muestra un diagrama de flujo.

En la figura 3.47, se muestran las imágenes virtuales referenciales de dimensión 7x5 (filas, columnas) de cada dígito con la que se correlacionan los resultados obtenidos en la etapa de pre procesamiento.

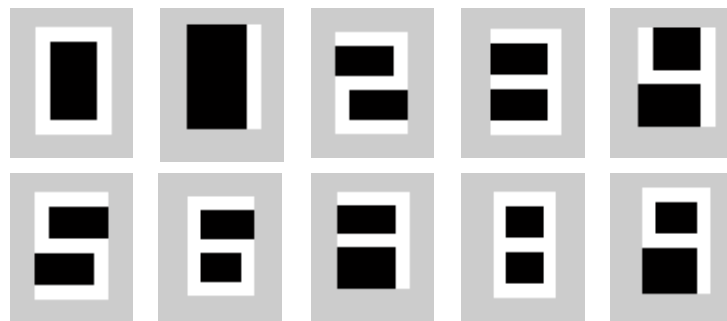


Figura 3.47 Imagen referencial de cada dígito de dimensión 7x5 para el proceso de correlación

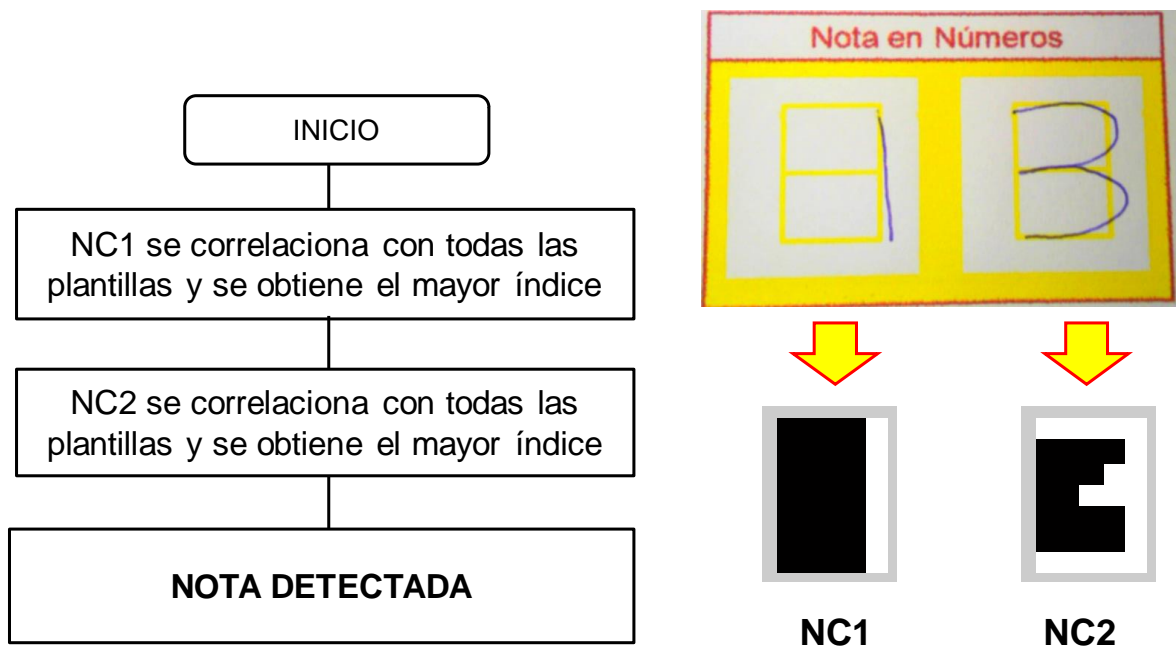


Figura 3.48 Diagrama de flujo de reconocimiento de la nota en números

3.7.3.2 Detección mediante redes neuronales

La red neuronal que se utilizó para la detección de la nota del alumno en números dentro de la cuadrícula nota del alumno, se llamará RN3 y esta tiene 35 entradas y 4 salidas. Las 35 entradas serán parámetros con los que se caracterizó al número y las 4 salidas indicarán qué número fue.

En la Figura 3.49, se muestra un ejemplo. Como se aprecia el número tres presenta un vector columna de 35 valores de entrada y se obtiene a la salida de la red el valor "0011" en binario que significa el número tres. Esta codificación en binario permitirá una mayor robustez en el resultado.

La nota del alumno está compuesta por dos (02) dígitos, por lo tanto, la cuadrícula de nota del alumno que contiene 2 números, como se muestra la Figura 3.14. Entonces, para el reconocimiento de la nota se necesitan las matrices donde se encuentren sus patrones de cada número. Estos se obtuvieron en la etapa de pre procesamiento y se guardaron como matriz ND1 y ND2.

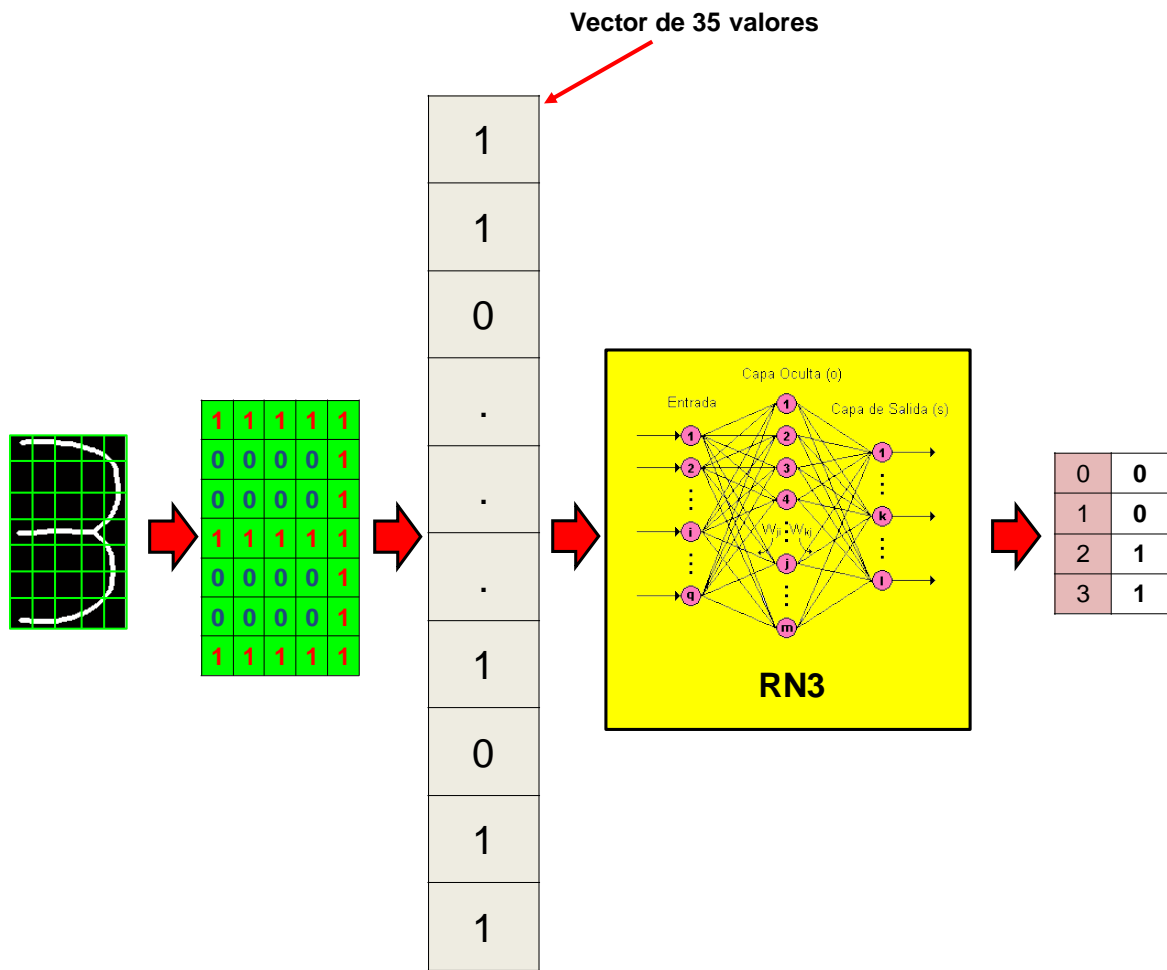


Figura 3.49 Ejemplo de reconocimiento del número manuscrito

Para obtener la nota se realiza el siguiente procedimiento que se aplica en el diagrama de flujo que se muestra en la Figura 3.50.

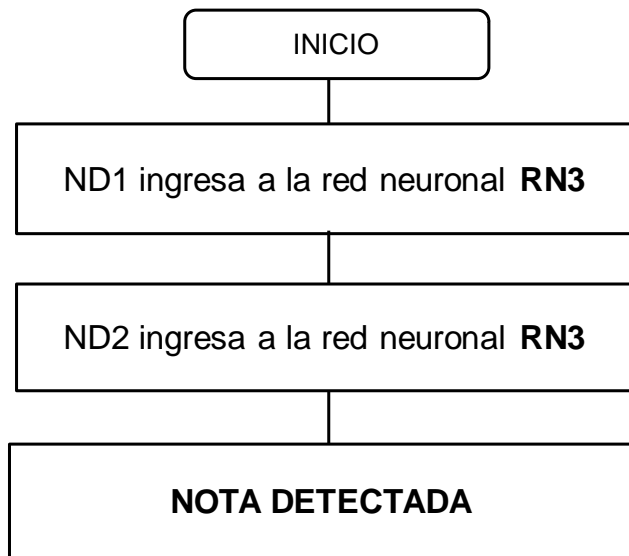


Figura 3.50 Diagrama de bloques de reconocimiento del número manuscrito

CAPÍTULO IV

ANÁLISIS Y PRESENTACIÓN DE RESULTADOS

4.1 Descripción del escenario de las pruebas

Las pruebas se han realizado sobre el recinto mostrado en la Figura 3.2. Dicho recinto cuenta con una iluminación provista de 6 indicadores Leds de alta intensidad, de 3.5 vatios cada uno, dispuestos como lo indica la Figura 3.4. Esto permite que se tenga una iluminación constante y homogénea dentro del recinto sobre el cuadernillo.

Asimismo, la cámara web al momento de obtener la imagen mantenía una distancia constante sobre el cuadernillo, es igual a 25cm. Esto permite que todas las tomas sean uniformes.

Las muestras de los cuadernillos escritos se obtuvieron de 50 docentes pertenecientes a la Facultad de Ingeniería y Arquitectura de la Universidad de San Martín de Porres, los cuales, cada uno de ellos, llenaron cinco (05) cuadernillos, dos dígitos por cuadernillo, del cero al nueve.

4.2 Resultados de imágenes obtenidas

En las Figuras 4.1 y 4.2, se muestran dos resultados que se obtuvo en la detección del código del alumno en marcas. Solo se hace referencia a una sola marca para que se pueda visualizar. En las Figuras 4.1a y 4.2a, se tiene la imagen; en las figuras 4.1b y 4.2b, se obtienen las marcas seleccionadas, y en las Figuras 4.1c y 4.2c, se tiene las distancias de dicha marca a los 4 puntos de referencia y en 4.1d y 4.2d, se muestran las distancias correspondientes.

En las Figuras 4.3 y 4.4, se muestran dos resultados que se obtuvo en la detección de la nota del alumno. En este caso es solo una sola marca. En 4.3a y 4.4a, se tiene la imagen, en 4.3b y 4.4b, se obtiene la marca seleccionada, en 4.3c y 4.4c, se

tienen las distancias de dicha marca a los 4 puntos de referencia y en las figuras 4.3d y 4.4d, se muestran las distancias correspondientes.

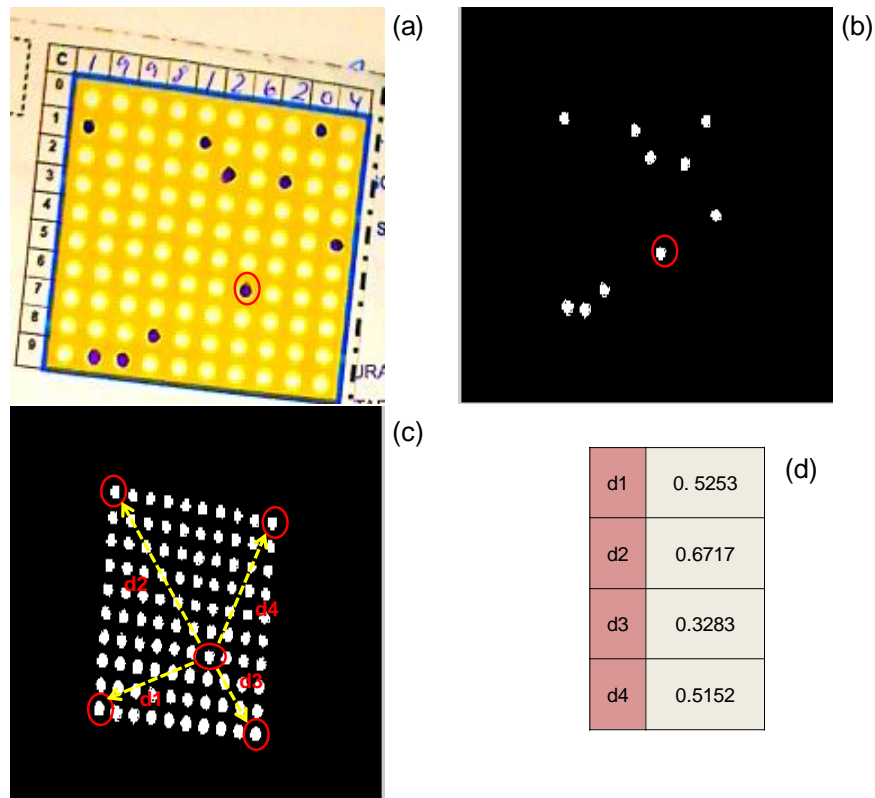


Figura 4.1 Ejemplo 1: (a) Imagen del código del alumno seleccionando una marca en particular, que se encuentra marcado de rojo. (b) imagen binarizada de las marcas, indicando la marca particular de rojo. (c) Imagen de la distancia de la marca, marcado de rojo con los cuatro puntos de referencia. (d) Obtención de las cuatro distancias de la marca.

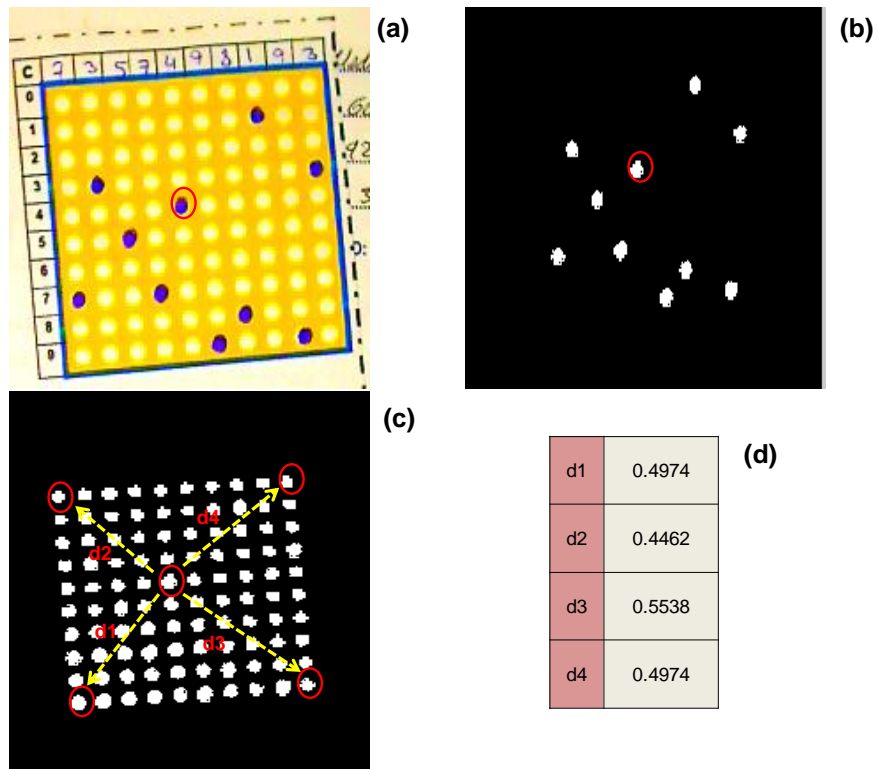


Figura 4.2: Ejemplo 2: (a) Imagen del código del alumno seleccionando una marca en particular, que se encuentra marcado de rojo. (b) imagen binarizada de las marcas, indicando la marca particular de rojo. (c) Imagen de la distancia de la marca, marcado de rojo con los cuatro puntos de referencia. (d) Obtención de las cuatro distancias de la marca.

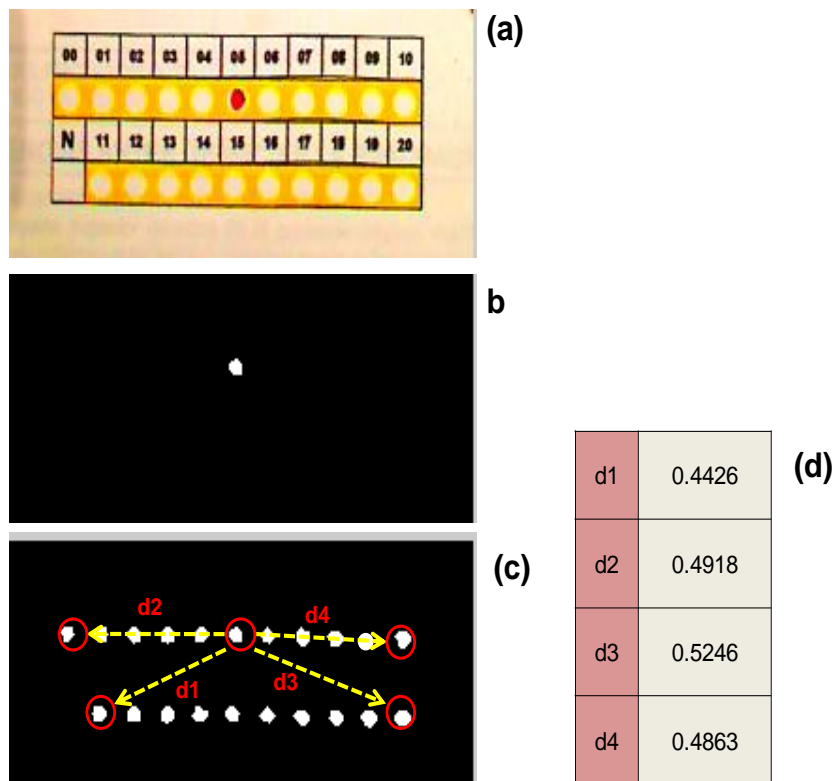


Figura 4.3 Ejemplo 1: (a) Imagen de la nota del alumno. (b) imagen binarizada de la marca. (c) Imagen de la distancia de la marca, marcado de rojo con los cuatro puntos de referencia. (d) Obtención de las cuatro distancias de la marca.

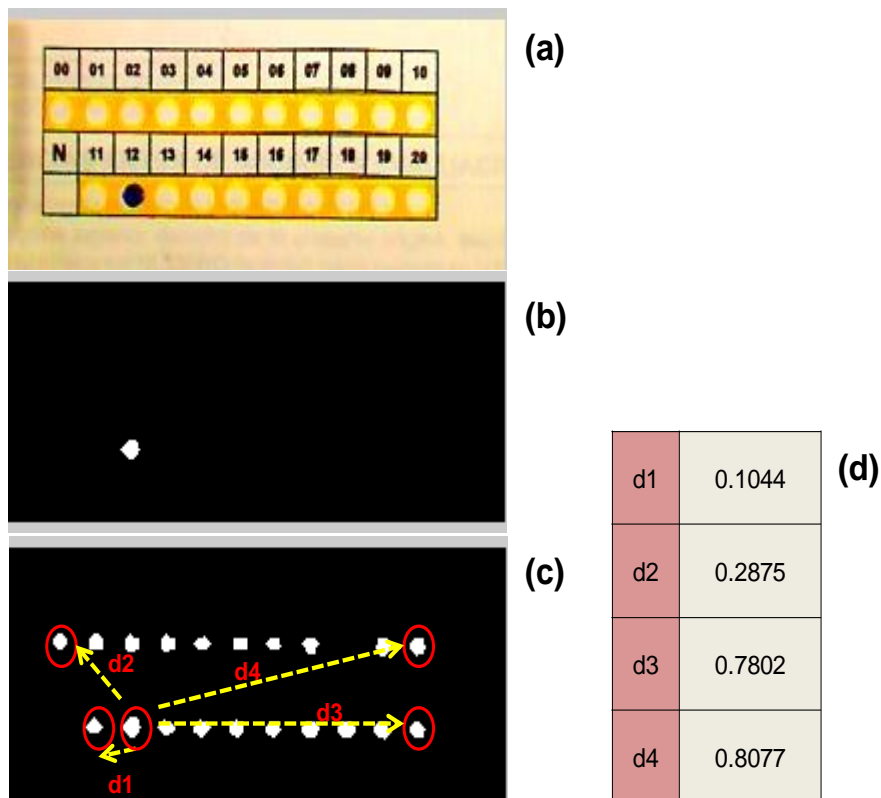


Figura 4.4 Ejemplo 2: (a) Imagen de la nota del alumno. (b) imagen binarizada de la marca. (c) Imagen de la distancia de la marca, marcado de rojo con los cuatro puntos de referencia. (d) Obtención de las cuatro distancias de la marca.

Es importante que los números por escribir por el docente mantengan cierta forma. En las Figuras 4.5 y 4.6, se observa la manera de escribir cada número. Asimismo, se visualiza que si bien existe una guía a seguir para escribir el número, no necesariamente debe ser ceñida a esta.

Se busca que el docente tenga comodidad o naturalidad al momento escribir los números.

Los números que contienen curvas como el cero, dos, tres, cinco, seis, ocho y nueve son los más complicados de que se ajusten a la guía. Por el contrario, los números uno, cuatro y siete son los más fáciles de ajustar debido a que están conformados por líneas rectas.

En las Figuras 4.7 y 4.8, se muestran números resultantes del proceso de normalización del grosor. Se puede apreciar que se logra en todos los dígitos esta uniformidad. También se puede apreciar que no necesariamente los dígitos mantienen un

mismo tamaño, la resolución de estos varían entre 195 y 235 píxeles por fila y entre 120 y 165 píxeles por columna para todos los dígitos a excepción del número uno. La resolución del número uno oscila entre 195 y 235 píxeles por fila y entre 15 y 23 píxeles por columna.

Como se mencionó en el Capítulo III el resultado final del pre procesamiento de los números manuscritos, es una imagen que debe tener una resolución de 7 filas por 5 columnas. En la Figura 4.10 se muestra la forma ideal que debería tener cada dígito. Sin embargo, el dígito resultante del pre procesamiento puede tener la forma que se muestran en la Figura 4.9. Estas son algunas formas que podrían tener los distintos números.

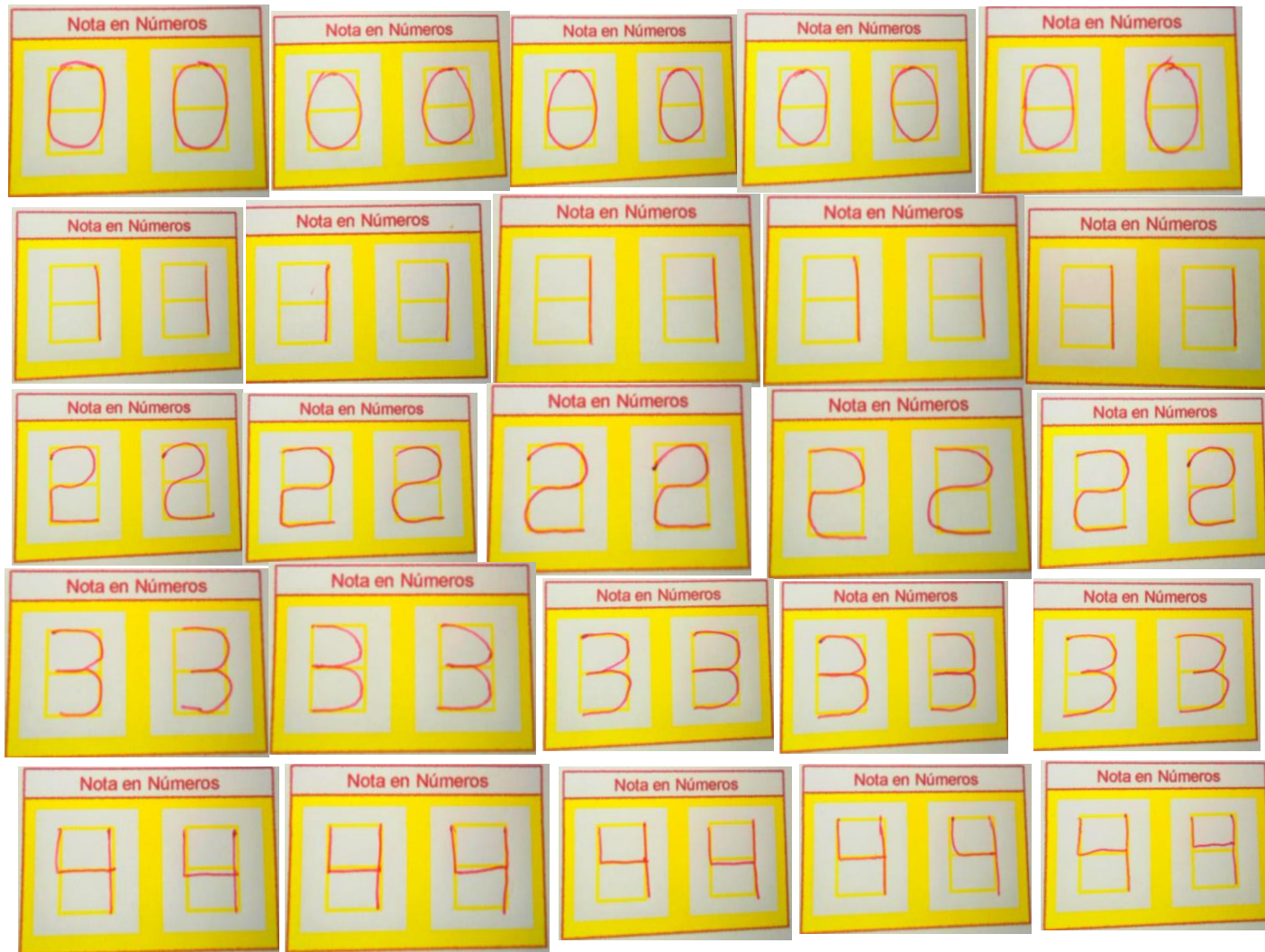


Figura 4.5 Números escritos por el docente



Figura 4.6 (continuación) Números escritos por el docente

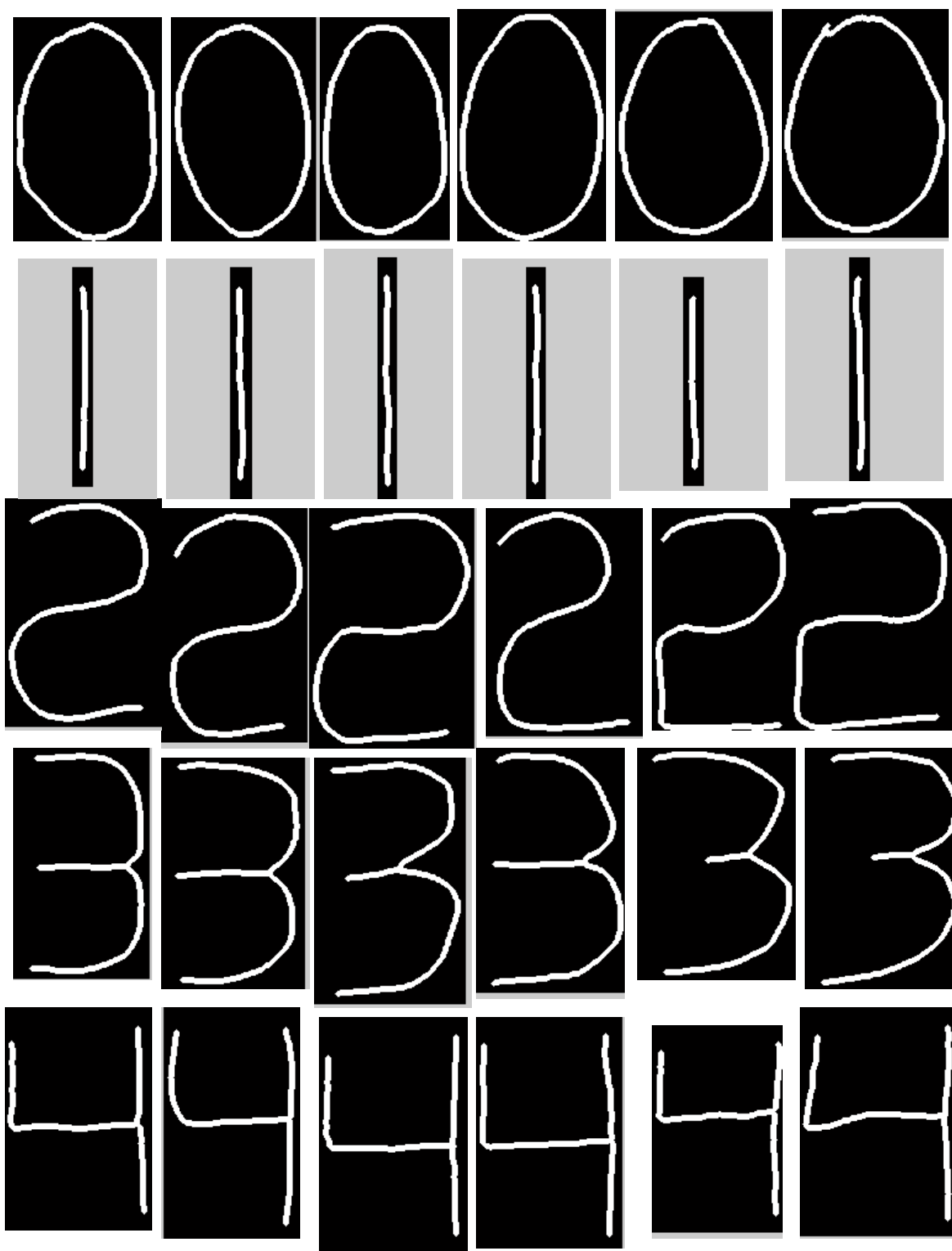


Figura 4.7 Resultados obtenidos para el proceso de normalización de grosor

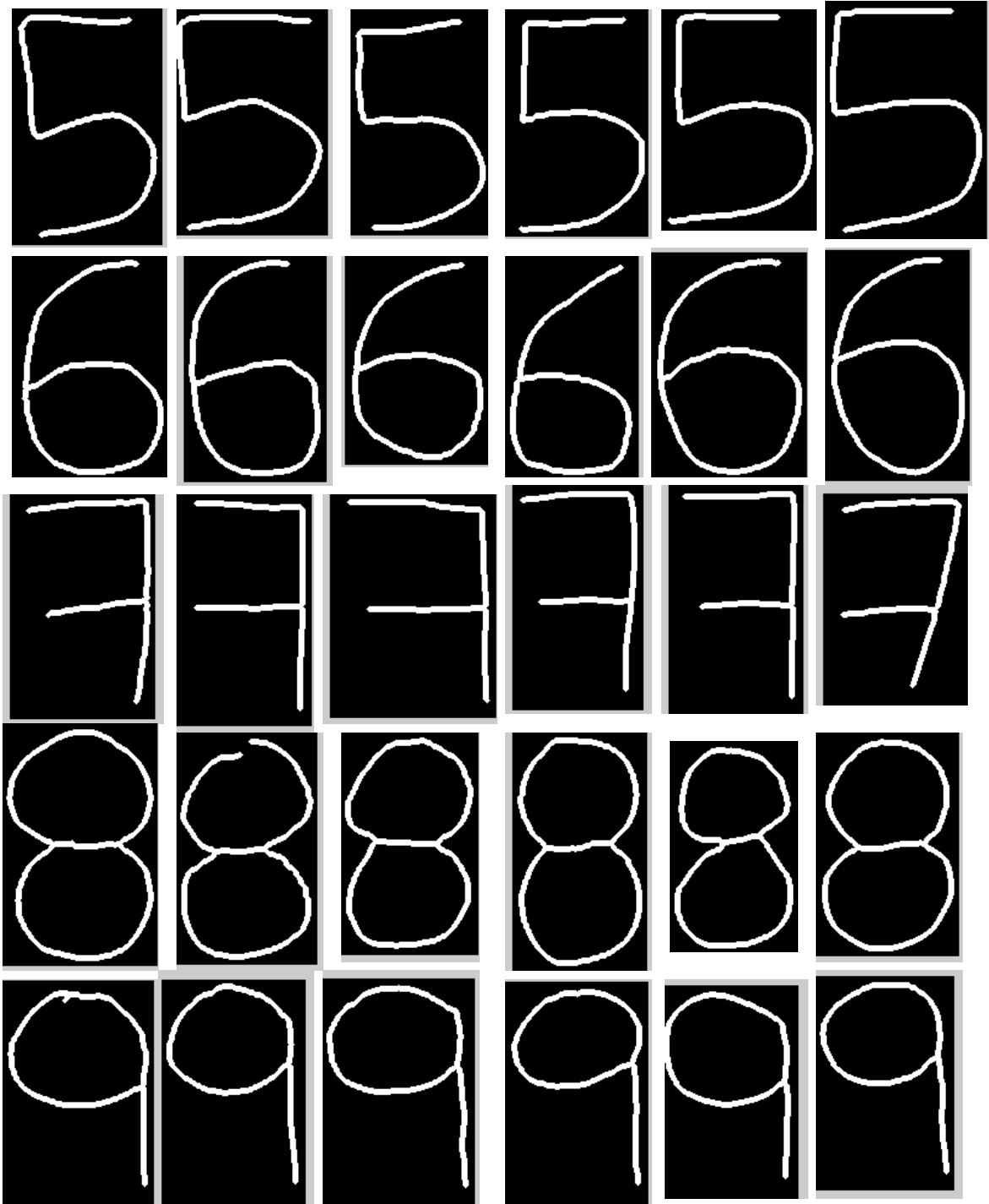


Figura 4.8 (continuación) Resultados obtenidos para el proceso de normalización de grosor

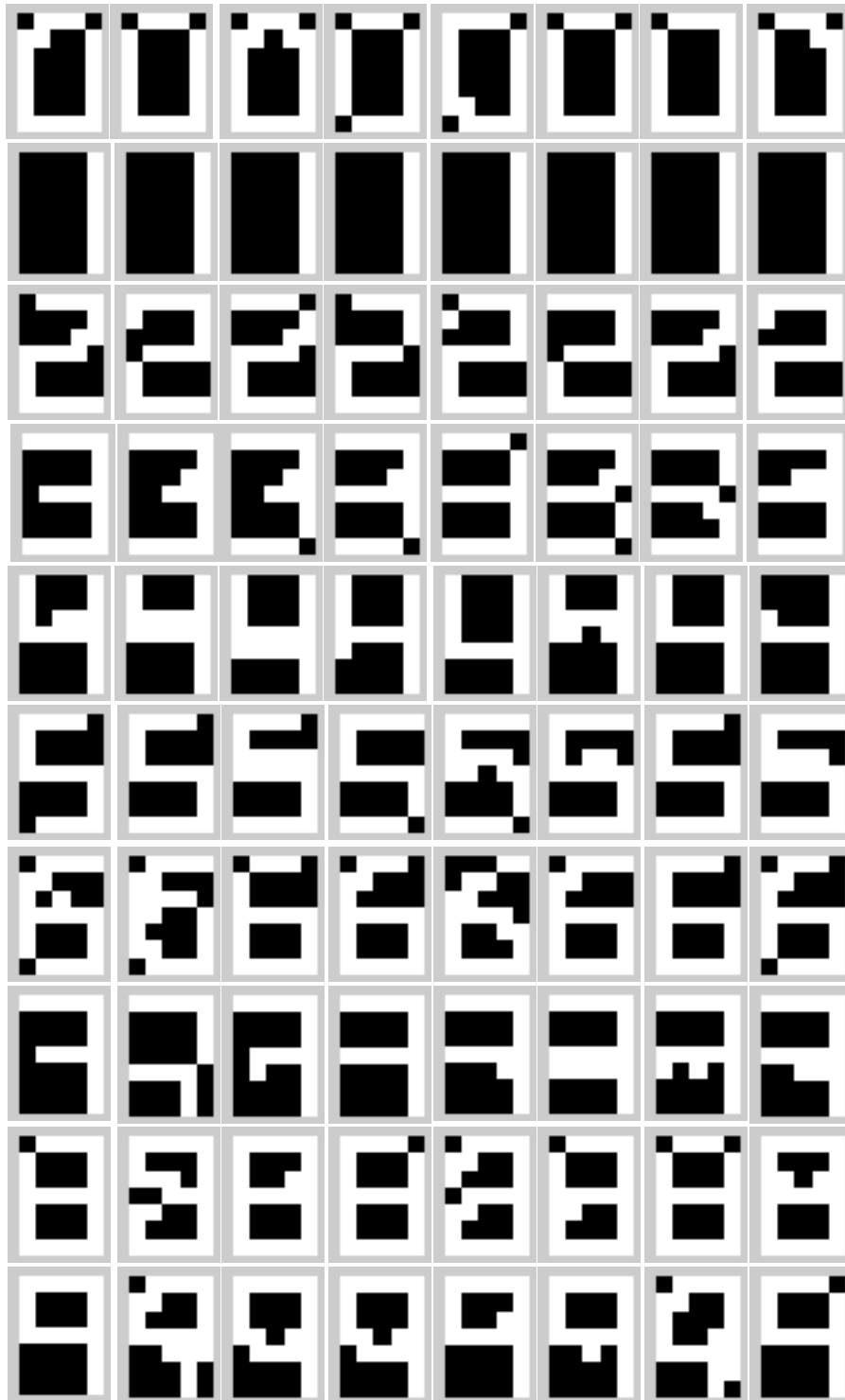


Figura 4.9 Resultados de dígito obtenido de 7x5

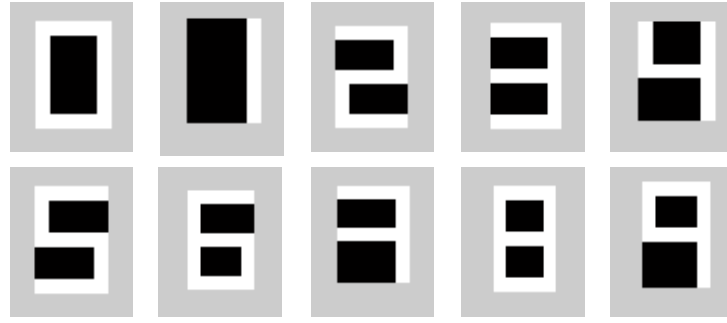


Figura 4.10 Plantilla de los números de dimensión 7x5 píxeles para el proceso de correlación

4.3 Métodos de validación empleados y resultados obtenidos

Para la correcta detección se tiene que cumplir con lo siguiente 1) el cuadernillo este posicionado correctamente; 2) que el recinto se encuentre iluminado; 3) que las marcas estén bien sombreadas, es decir, no muy tenue y dentro de la marca; 4) que el número manuscrito este de acuerdo con la plantilla; 5) que el número tenga un trazo continuo.

Para la validación de los resultados mediante el uso de redes neuronales y correlación de los números manuscritos se utilizó la ecuación (4.1). Mediante esta, se obtuvo la probabilidad de acierto del sistema propuesto.

En lo que respecta al reconocimiento de los números manuscritos mediante el uso de redes neuronales y correlación, se obtuvo un total de 50 muestras de cada dígito provistas por 50 docentes, a las cuales se les instruyó como debían colocar los números en la cuadrícula acondicionada. Estas muestras se procesaron cinco (05) veces. En la tabla 4.1, se muestran los resultados obtenidos del reconocimiento de los números manuscritos mediante correlación y redes neuronales.

$$P_{ad} = \frac{\text{Cantidad de dígitos reconocidos}}{\text{Cantidad total de dígitos} \times \text{Cantidad de veces procesados}} \times 100 \quad (4.1)$$

Algo importante que se menciona es que se presentó en el reconocimiento, mediante el índice de correlación, debido al parecido de ciertos dígitos como es el caso de los números 7 con el 9, 5 con el 6, 8 con el 9, 2 con 8; presentan un coeficiente de correlación muy parecido con una diferencia menor o igual a 0.15. Se

llegaba a reconocer el dígito correcto, pero existía dicha similitud. En el caso del reconocimiento mediante el uso de redes neuronales no se presenta esta confusión. En este caso, fue bien marcado el patrón reconocido.

Para la validación del código y nota del alumno en marcas, se utilizaron las ecuaciones (4.2) y (4.3).

En este caso, se utilizaron 50 cuadernillos debidamente sombreados en las marcas, y se procesaron 5 veces.

Tabla 4.1 Resultados de reconocimiento de los números manuscritos mediante el uso de redes neuronales y correlación

DÍGITO	CORRELACIÓN		REDES NEURONALES		UNIVERSO	N° de repeticiones	N° total de pruebas por dígito
	VERDADERO RECONOCIMIENTO	FALSO RECHAZO	VERDADERO RECONOCIMIENTO	FALSO RECHAZO			
CERO	100%	0%	100%	0%	50	5	250
UNO	100%	0%	100%	0%	50	5	250
DOS	99.6%	0.4%	100%	0%	50	5	250
TRES	100%	0%	100%	0%	50	5	250
CUATRO	100%	0%	100%	0%	50	5	250
CINCO	100%	0%	100%	0%	50	5	250
SEIS	98%	2%	98.4%	1.6%	50	5	250
SIETE	100%	0%	98.4%	1.6%	50	5	250
OCHO	96.8%	3.2%	98%	2%	50	5	250
NUEVE	100%	0%	100%	0%	50	5	250
	99.44%		99.48				

$$P_{ac} = \frac{\text{Cantidad de codigos reconocidos}}{\text{Cantidad total de codigos} \times \text{Cantidad de veces procesados}} \times 100 \quad (4.2)$$

$$P_{an} = \frac{\text{Cantidad de notas reconocidos}}{\text{Cantidad total de notas} \times \text{Cantidad de veces procesados}} \times 100 \quad (4.3)$$

Tabla 4.2 Resultados de reconocimiento de los códigos y notas mediante marcas haciendo uso de redes neuronales

	VERDADERO RECONOCIMIENTO	FALSO RECHAZO	UNIVERSO	N° DE VECES	N° total de pruebas
CÓDIGO	98.4%	1.6%	50	5	250
NOTA	98.8%	1.2%	50	5	250
	98.6%				

En el caso de las marcas, para el código y nota, el error que se presentó fue debido a que no se sombreó correctamente la marca o se sombreó fuera de la marca. En la tabla 4.2 se muestra los resultados obtenidos del reconocimiento de las marcas, tanto del código del alumno como de la nota, por medio de redes neuronales.

En la tabla 4.3 se muestra la tabla de confusión para analizar la identificación de la red neuronal de los números manuscritos. Como se puede apreciar, en el caso del número 6, el sistema en tres ocasiones lo reconoció como número 3; para el número 7 el sistema en cuatro ocasiones lo reconoció como el número 9; y así mismo para el número 8 el sistema en dos ocasiones lo reconoció como el número 9.

Tabla 4.3 Matriz de confusión del reconocimiento de los números manuscritos mediante la RN

	Cero	Uno	dos	Tres	Cuatro	Cinco	Seis	Siete	Ocho	Nueve
Cero	250	0	0	0	0	0	0	0	0	0
Uno	0	250	0	0	0	0	0	0	0	0
Dos	0	0	250	0	0	0	0	0	0	0
Tres	0	0	0	250	0	0	0	0	0	0
Cuatro	0	0	0	0	250	0	0	0	0	0
Cinco	0	0	0	0	0	250	0	0	0	0
Seis	1	0	0	0	3	0	246	0	0	0
Siete	0	0	0	0	0	0	0	246	0	4
Ocho	3	0	0	0	0	0	0	0	245	2
Nueve	0	0	0	0	0	0	0	0	0	250

4.4 Tiempo de procesamiento

El sistema tiene un tiempo de procesamiento de aproximadamente 3 segundos por cuadernillo. Un salón de clases que tuviera 85 alumnos, el sistema registraría las

notas en 255 segundos o 4.25 minutos aproximadamente. Lo cual implica que en una jornada de 8 horas el sistema podría registrar 9600 cuadernillos, es decir, en dos días de trabajo el sistema registraría las evaluaciones de todo un ciclo en la facultad.

4.5 Software desarrollado

El software se ha desarrollado en Matlab 2009, en el cual se han programado los distintos algoritmos de procesamiento digital de imágenes y se ha usado el toolbox de de redes neuronales [7]. Además se ha desarrollado una interfaz gráfica con la herramienta GUIDE de Matlab.

La interfaz visual del software se muestra en la Figura 4.11. También se tiene una viñeta que va a permitir seleccionar las tres áreas de interés que se realiza al inicio del proceso (Figura 4.12).

En la Figura 4.13, se visualiza una pantalla donde se muestran los resultados obtenidos, código del alumno y su respectiva nota. Asimismo, en un archivo Excel también se almacenan los resultados obtenidos.



Figura 4.11 Interfaz visual del software desarrollado

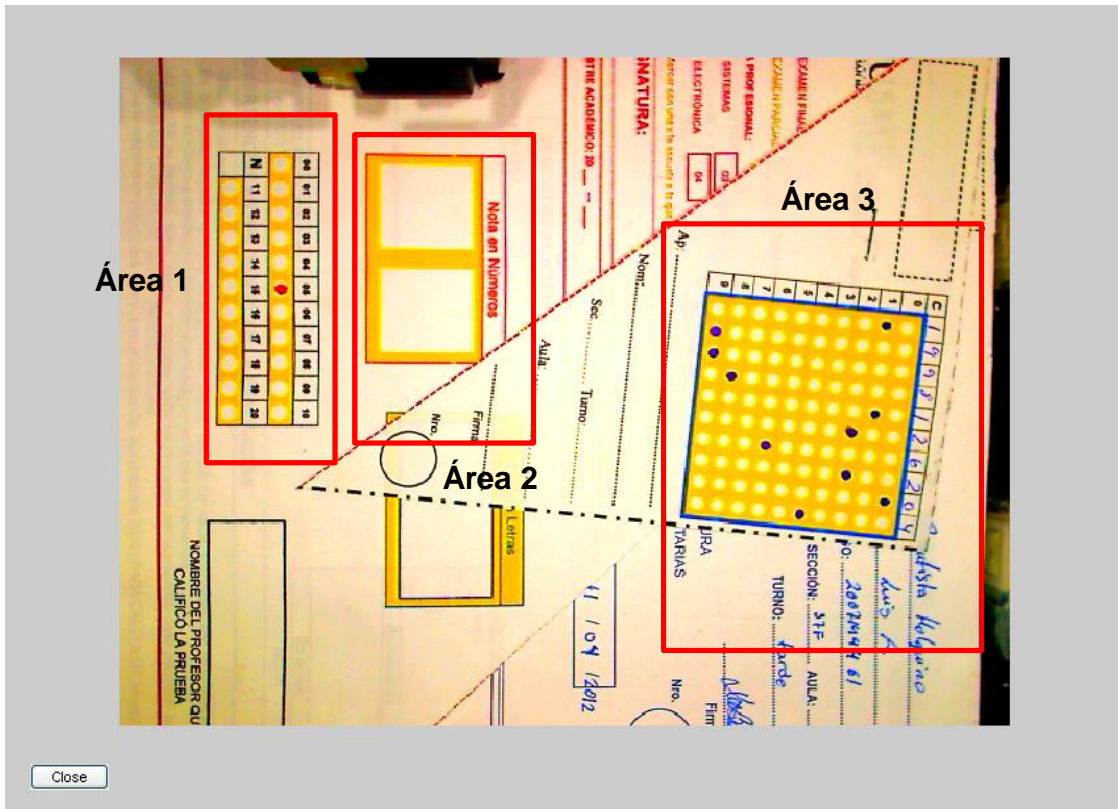


Figura 4.12 Interfaz visual de selección de áreas de interés

REGISTRO DE ALUMNOS

ALUMNOS REGISTRADOS

	CÓDIGO	NOTA
1	1998126204	5
2	7357498193	12
3	6451327185	12
4	7887666454	19
5	6544333333	16
6	9922531543	11
7	2043868468	14
8	2092464646	15
9	2110357535	20
10	2013124684	3

Limpiar

Figura 4.13 Pantalla de resultados obtenidos

4.6 Costo del trabajo de tesis

En la Tabla 4.4, se muestra el costo que tuvo el presente trabajo. En él no se toma en cuenta el costo que tuvo la licencia de MATLAB, computadora personal, y fuente de poder, debido a que estos materiales fueron provistos por la universidad. Asimismo, no se está cotizando horas de desarrollo dedicadas al trabajo. También hubo otros artículos que se usaron, pero que eran de libre distribución, así como textos y artículos.

Tabla 4.4 Materiales para la tesis y costos

NOMBRE	DESCRIPCIÓN	CANTIDAD	COSTO
Webcam	Cámara Microsoft Life Cam Studio USB Modelo: Q2F-00009	1	S/. 300
LEDs	Leds de alta intensidad de 3.5 vatios	6	S/. 50
Recinto	Recinto fabricado de madera de MDF, marco de iluminación, cableado y componentes	1	S/. 700
Fuente de poder	Fuente de poder BK Precision 1670A	1	Provisto por la universidad USMP
Millar Papel Bond	Impresión de las carátulas de los cuadernillos para las diversas pruebas	5	S/. 80
Matlab 2009	Software MATLAB	1	Provisto por la universidad USMP
PC	Desktop LENOVO THINKVISION, Modelo 9434-DN5	1	Provisto por la universidad
Papers	Artículos IEEE	6	S/. 350
TOTAL			S/. 1480

CONCLUSIONES

Se desarrolló un método eficiente de reconocimiento de patrones numéricos y de marcas matriciales basado en procesamiento digital de imágenes a color y redes neuronales Backpropagation orientado al registro automático de evaluaciones académicas numéricas.

El método fue evaluado y se obtuvo la tasa de verdadero reconocimiento para números manuscritos de 99.44%, haciendo uso del coeficiente de correlación y de 99.48 haciendo uso de redes neuronales. Hay que tener en cuenta que la cuadrícula que se implementó para la escritura del dígito ayudo mucho a que se obtenga dicho resultado. Además la tasa de verdadero reconocimiento que se logró para las marcas ópticas fue de 98.6%.

Por otro lado, se diseñó una nueva carátula del cuadernillo de evaluación que cumplió con los requerimientos del sistema, adicionalmente, como una segunda validación a la detección de los números manuscritos se detecta la nota mediante una marca óptica.

Al realizar el procesamiento de marcas ópticas mediante una webcam, se suple el uso de un lector óptico de marcas. Esto permite un ahorro considerable en un sistema de registro automático de evaluaciones debido al costo de dichos lectoras.

La cuadrícula que se implemento para los números manuscritos permitió estandarizar los símbolos reducir la variación de los dígitos, dando como resultado mejores tasas de acierto de reconocimiento.

Se logró estandarizar al número en inclinación, grosor y tamaño de manera adecuada permitiendo una correcta detección del dígito.

RECOMENDACIONES PARA TRABAJOS FUTUROS

Al sistema propuesto se le puede agregar un recinto que automatice el proceso de alimentación de los cuadernillos logrando, de esta manera, que el sistema sea más rentable.

Cambiando la cámara Web por una cámara industrial se puede lograr una mejor imagen, es decir, menos cambiante, esto debido al ruido introducido por el sensor de la cámara Web. Con esta cámara, si tomamos dos imágenes consecutivas a un dígito, estas serán ligeramente variantes, por lo tanto entraría una secuencia distinta a la red neuronal.

BIBLIOGRAFÍA

- [1] **Batuwita, K., & Bandara, G.** (2006). Fuzzy Recognition of Offline Handwritten Numeric Characters. *Cybernetics and Intelligent Systems (CIS)*, IEEE Conference on.
- [2] **Beristain, A.** (2009). Skeleton based visual pattern recognition: applications to tabletop interaction. Tesis doctoral. Universidad del país Vasco. [Fecha de acceso 15 de agosto del 2014]
- [3] **Berry, M., Linoff, G.** (1997), *Data Mining Techniques*. New York. John Wiley & Sons. PP 464.
- [4] **Boger, Z., & Guterman, H.**, (1997), Knowledge extraction from artificial neural network models. *IEEE Systems, Man, and Cybernetics Conference*, Orlando, FL, USA.
- [5] **Chen, Q., Petriu, E., & Yang, X.** (2004). A Comparative Study of Fourier Descriptors and Hu's Seven Moment Invariants for Image Recognition. *Electrical and Computer Engineering*, 2004. Canadian Conference on (Vol. 1)
- [6] **Cheriet, M., Kharma, N., Liu, C., & Suen, C.** (2007). *Character Recognition Systems – A guide for Students and Practitioners*. New Jersey. John Wiley & Sons, Inc. PP 321.
- [7] **Demuth, H., Beale, M. & Hagan, M.** (2009). *Neural Network Toolbox™ 6 User's Guide*. [Fecha de acceso 05 de enero de 2014]. URL disponible en: <http://filer.case.edu/pjt9/b378s10/nnet.pdf>.
- [8] **Fang, X., & Alouani, A.** (2002). Unconstrained handwritten numeral recognition using fuzzy rule – based classifier. *System Theory*, 2002. Proceedings of the Thirty-Fourth Southeastern Symposium on.
- [9] **Fausett, L.** (1994). *Fundamentals of neural networks: architectures, algorithms, and applications*. Englewood Cliffs, NJ. Prentice-Hall. PP 461.
- [10] **Feedback.** Sistema integral de generación, corrección automática y análisis de exámenes escritos. [Fecha de acceso 15 de junio de 2014]. URL disponible en: www.sistemafeedback.com.ar/descargas/Una_mirada_al_interior_de_feedback.pdf
- [11] **Gonzalez, R & Woods, R.** (2001). *Digital Image Processing*. 2da. Edición. EE.UU. Prentice Hall. PP 793.
- [12] **Gumah, M., Schneider, E., & Aburas, A.** (2010). Handwriting Recognition System Using Fast Wavelets Transform. *Information Technology (ITSim)*, 2010 International Symposium in (Vol.1).

- [13] **Hanmandlu, M., Murali, K., & Chakraborty, S.** (2001). Fuzzy Logic Based Handwritten Character Recognition. Proceedings. 2001 International Conference on (Volume: 3).
- [14] **Haykin, S.** (1998). Neural Networks: A Comprehensive Foundation. 2nd Edition. EE.UU.PRENTICE HALL. PP 842.
- [15] **Hull, J.** (1994). A Database For Handwritten TextRecognition Research. IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol.16, No.5, May.1994. PP 550-554
- [16] **Kimura, F., & Shridhar, M.** (1991). Handwritten numerical recognition based on multiple algorithms. Pattern Recognition Society Vol 24, N° 10, pp 969-983.
- [17] **Lawrence, S., Lee C., & Chung, A.** (1996). What Size Neural Network Gives Optimal Generalization? Convergence Properties of Backpropagation. NEC Research Institute. URL disponible en: <http://clgiles.ist.psu.edu/papers/UMD-CS-TR-3617.what.size.neural.net.to.use.pdf>
- [18] **Lam, L., Lee, S., & Suen, S.** (1992). Thinning Methodologies – A Comprehensive Survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol, 14, N°. 9, September 1992.
- [19] **Martin, M.** (2004). Descriptores de imagen. [Fecha de acceso febrero 2015]. [En línea]. URL disponible en <http://lmi.bwh.harvard.edu/papers/pdfs/2003/martin-fernandezCOURSE03e.pdf>
- [20] **Masters, T.** (1993). Practical Neural Network Recipes in C++. San Diego CA. Academic Press, Inc. PP 493.
- [21] **Sekonic.** Powerful & Versatile OMR, World's top speed of 11000 sheets/hr. [Fecha de acceso 15 de junio de 2014]. URL disponible en: www.sekonic.co.jp/english/product/omr/sr_11000/pdf/catalog_SR-11000.pdf
- [22] **Slavik, P., & Govindaraju, V.** 2002). Equivalence of different methods for slant and skew corrections in word recognition applicatios. Pattern Analysis and Machine Intelligence, IEEE Transactions on (Vol. 23).
- [23] **Valveny, E., & López, A.** (2003). Numeral recognition for quality control of surgical sachets. Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR 2003) IEEE Computer Society.
- [24] **Wang, J., Yang, J., Li, S., Dai, Q., & Xie, J.** (2007). Number Image Recognition Based on Neural Network Ensemble. Natural Computation, 2007. ICNC 2007. Third International Conference on (Volume: 1)
- [25] **Wang, P., & Zhang, Y.** (1989). A Fast and Flexible Thinning Algoritm. Computers, IEEE Transactions on (Volume: 38, Issue: 5): Pages 741-745.
- [26] **Zhang, T., & Suen, C.** (1984). A Fast Parallel Algorithm for Thinning Digital Patterns. CACM, 27 (1984), PP. 236 -239.
- [27] **Zhang, Y., & Wang, P.** (1988). A Modified Parallel thinning Algoritm. IEEE trans. 1988.