



REPÚBLICA DEL ECUADOR

Escuela Politécnica Nacional

" E SCIENTIA HOMINIS SALUS "

La versión digital de esta tesis está protegida por la Ley de Derechos de Autor del Ecuador.

Los derechos de autor han sido entregados a la "ESCUELA POLITÉCNICA NACIONAL" bajo el libre consentimiento del (los) autor(es).

Al consultar esta tesis deberá acatar con las disposiciones de la Ley y las siguientes condiciones de uso:

- Cualquier uso que haga de estos documentos o imágenes deben ser sólo para efectos de investigación o estudio académico, y usted no puede ponerlos a disposición de otra persona.
- Usted deberá reconocer el derecho del autor a ser identificado y citado como el autor de esta tesis.
- No se podrá obtener ningún beneficio comercial y las obras derivadas tienen que estar bajo los mismos términos de licencia que el trabajo original.

El Libre Acceso a la información, promueve el reconocimiento de la originalidad de las ideas de los demás, respetando las normas de presentación y de citación de autores con el fin de no incurrir en actos ilegítimos de copiar y hacer pasar como propias las creaciones de terceras personas.

Respeto hacia sí mismo y hacia los demás.

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

DISEÑO E IMPLEMENTACION DE UN SISTEMA TRADUCTOR DE LENGUAJE DE SEÑAS DE MANOS A UN LENGUAJE DE TEXTO MEDIANTE VISION ARTIFICIAL EN UN AMBIENTE CONTROLADO

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIEROS EN
ELECTRÓNICA Y CONTROL**

CHIGUANO RODRIGUEZ EDWIN FABIAN

edwinchiguano3@hotmail.com

MORENO DIAZ NATHALY VERONICA

natymd9@hotmail.com

DIRECTOR: LUIS ANIBAL CORRALES PAUCAR, PhD

luis.corrales@epn.edu.ec

Quito, Junio 2011

DECLARACIÓN

Nosotros, Edwin Fabián Chiguano Rodríguez y Nathaly Verónica Moreno Díaz declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

Edwin Fabian Chiguano Rodriguez

Nathaly Verónica Moreno Díaz

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Edwin Fabián Chiguano Rodríguez y Nathaly Verónica Moreno Díaz

Luis Aníbal Corrales Paucar, PhD.

DIRECTOR DEL PROYECTO

AGRADECIMIENTOS

Agradezco a todas las personas que ayudaron de forma directa o indirecta en la realización de este proyecto, en especial a mi compañera de tesis Nathaly. Sin ti no lo habríamos logrado.

A todos los miembros de mi familia Galo, Rosalba, Luis, Rosalba, Josselyn, Emily y Mayra por el cariño y confianza que me tienen.

A los amigos del Observatorio Astronómico de Quito por darme la oportunidad de trabajar y crecer profesionalmente.

A mis amigos de toda la vida del Panecillo por acompañarme y darme ánimos a salir adelante a lo largo de mi corta vida.

A Luis Corrales PhD. por la ayuda y el tiempo que dedicó a la culminación de este proyecto.

Finalmente quiero agradecer a todos mis profesores de la universidad, por el conocimiento impartido a lo largo de estos años.

Edwin

AGRADECIMIENTO

Gracias a mi Dios que nunca me dejó sola, que me tendió su mano para levantarme después de una caída. Gracias por permitirme llegar hasta aquí rodeada de todas las personas que quiero.

Gracias a Jorge y Marieta por darme la mejor herencia que un padre puede dar a un hijo, la profesión. Gracias por todas sus enseñanzas, sus valores y sus sacrificios que me han permitido llegar a cumplir esta meta. Todo este esfuerzo es por ustedes, para que estén muy orgullosos de su hija.

Gracias Freddy y Lila porque cuando los momentos se volvían aun más difíciles siempre pude contar con su apoyo y confianza, que me dio la fuerza necesaria para seguir adelante. Han sido y serán para mí como unos padres a los que quiero y respeto.

Gracias a Jorge, María, Fredicito, Martincito, Rosario y Teresita por siempre estar al pendiente de mí y de mis cosas, con todos sus detalles por pequeños que fueran me ayudaron y me dieron el ánimo para empezar un nuevo día. Gracias.

Gracias Edwin por nunca dejarme atrás, por siempre llevarme a tu lado a pesar de todo, por caminar por los dos cuando yo no podía más, aun cuando la carga era pesada. Gracias por todas las veces que me ayudaste y por todo el trabajo puesto en este proyecto, sin ti no hubiera sido posible.

Nathaly

DEDICATORIA

A mi padre Galo por el apoyo incondicional que me brinda todos los días de mi vida, a mi madre Rosalba por la paciencia que me ha tenido a lo largo de mi carrera, a mi hermano Luis por todo el apoyo, amistad y confianza que me brinda, a mis hermanas Rosalba y Josselyn.

A los amigos de la carrera con los cuales logramos superar todos los problemas que se presentaron en todo este camino.

Finalmente quiero dedicar este esfuerzo a mi gran amor Nathaly por estar a mi lado en las buenas y en las malas, y sobre todo por la infinita paciencia que me tiene.

Edwin

DEDICATORIA

Este trabajo lo dedico a mis padres Jorge y Marieta, porque nunca dejaron de apoyarme, confiar en mí y darme su amor incondicional.

A mis Hermanos Jorge y María Caridad y a mis primitos Freddy y Martín para que tengan en este trabajo una prueba de que con responsabilidad, lucha y perseverancia todo se puede lograr.

A mis tíos Freddy y Lila por su ejemplo, apoyo e invaluable consejos.

A mi abuelita Rosario y a Teresita que con sus pequeños detalles hicieron las cosas más simples.

A mi amor, compañero y confidente Edwin por su inagotable perseverancia y responsabilidad que en los momentos difíciles nos permitieron salir adelante.

A todos mis amigos y amigas por su amistad y cariño en todos estos años de estudio.

Nathaly

CONTENIDO

RESUMEN	vi
PRESENTACION	viii
CAPÍTULO 1	1
FUNDAMENTO TEÓRICO	1
1.1 DESCRIPCIÓN DEL PROBLEMA	1
1.2 INTRODUCCIÓN AL LENGUAJE DE SEÑAS	1
1.3 VISIÓN ARTIFICIAL	3
1.3.2 ELEMENTOS DE UN SISTEMA DE VISION ARTIFICIAL	4
1.3.2.1 ILUMINACIÓN	4
1.3.2.2 SENSOR DE IMAGEN	7
1.3.2.3 TARJETAS DE CAPTURA	7
1.3.2.4 ALGORITMOS DE ANÁLISIS DE IMAGEN	7
1.3.2.5 COMPUTADORA O MÓDULO DE PROCESO	8
1.3.2.6 SISTEMA DE RESPUESTA EN TIEMPO REAL	8
1.3.3 ETAPAS DE UN SISTEMA DE VISIÓN ARTIFICIAL	8
1.3.4 COMPARACIÓN ENTRE LOS SISTEMAS DE VISIÓN ARTIFICIALES Y HUMANOS	9
1.4 PROCESAMIENTO DIGITAL DE IMÁGENES	10
1.4.1 DEFINICIÓN DE UNA IMAGEN	10
1.4.1.1 TIPOS DE IMÁGENES	11
1.4.1.1.1 IMAGEN A COLOR	11
1.4.1.1.2 IMAGEN EN ESCALA DE GRISES	12
1.4.1.1.3 IMAGEN BINARIA	12
1.4.1.1.4 HISTOGRAMA DE UNA IMAGEN	13
1.4.2 PRE-PROCESAMIENTO DIGITAL DE IMÁGENES	14
1.4.2.1 FILTROS EN EL DOMINIO ESPACIAL	15
1.4.2.1.1 FILTROS LINEALES	15
1.4.2.1.2 FILTROS EN EL DOMINIO DE LA FRECUENCIA	17

1.4.3	SEGMENTACIÓN	18
1.4.3.1	SEGMENTACIÓN BASADA EN DETECCIÓN DE BORDES	19
1.4.3.2	SEGMENTACIÓN BASADA EN MORFOLOGÍA	19
1.4.3.2.1	DILATACIÓN	20
1.4.3.2.2	EROSIÓN	21
1.4.3.2.3	APERTURA	22
1.4.3.2.4	CIERRE	22
1.4.3.2.5	DETECCIÓN DE BORDES	23
1.4.3.2.6	MÉTODOS PARA DETECCIÓN DE FLANCOS:	25
	DETECCIÓN DE BORDES SIMPLE	
1.4.4	CLASIFICACION DE IMÁGENES	25
1.4.4.1	VECINO MÁS CERCANO	26
1.4.4.2	MÍNIMA DISTANCIA	27
1.4.4.3	K-VECINO MÁS CERCANO	27
	CAPÍTULO 2	29
	DISEÑO Y CONSTRUCCION DEL HARDWARE	29
2.1	INTRODUCCIÓN	29
2.2	SELECCIÓN DEL AMBIENTE CONTROLADO	29
2.2.1	AMBIENTE CONTROLADO DE ILUMINACIÓN	29
2.2.2	AMBIENTE CONTROLADO POR CONTRASTE	31
2.2.3	SELECCIÓN DE LA MEJOR OPCIÓN	32
2.2.4	CONSTRUCCIÓN DEL AMBIENTE CONTROLADO	32
2.2.4.1	EL FONDO	32
2.2.4.2	LA ILUMINACIÓN	33
2.2.4.3	LA CÁMARA	34
2.2.4.3.1	CÁMARA A COLOR O MONOCROMÁTICA	34
2.2.4.3.2	CÁMARA DIGITAL O ANALÓGICA	34
2.2.4.3.3	TIPO DE SALIDA DE VIDEO	34
2.2.4.3.4	FORMATO DEL VIDEO	34
2.2.4.3.5	RESOLUCIÓN	34

2.2.4.3.6 CARACTERÍSTICAS DEL SENSOR	35
2.2.4.3.7 VELOCIDAD DEL OBTURADOR	37
2.2.4.4 SELECCIÓN DE LA CÁMARA PARA LA APLICACIÓN	38
2.2.4.5 MONTAJE DE LA CÁMARA	39
2.2.4.6 HARDWARE IMPLEMENTADO	40
CAPÍTULO 3	44
DESARROLLO DEL SOFTWARE EN LABVIEW	44
3.1 INTRODUCCIÓN	44
3.2 DESCRIPCIÓN DEL PROGRAMA	44
3.2.1 PANTALLA PRINCIPAL	44
3.2.2 TRADUCTOR	45
3.2.2.1 ADQUISICIÓN	45
3.2.2.2 DETECCIÓN DE LA MANO EN LA IMAGEN	49
3.2.2.3 CAMBIO A ESCALA DE GRISES Y MEJORAMIENTO DEL CONTRASTE Y BRILLO	51
3.2.2.4 SEGMENTACIÓN	51
3.2.2.5 RECORTADO DEL OBJETO DE INTERÉS	52
3.2.2.5.1 CONSTRUCCIÓN DEL ROI	53
3.2.2.6 GENERACIÓN DE BASES DE DATOS	55
3.2.2.6.1 INTRODUCCIÓN AL MÓDULO NATIONAL INSTRUMENTS VISION ASSISTANT	55
➤ VENTANA DE REFERENCIA	56
➤ VENTANA DE FUNCIONES PARA PROCESAMIENTO	56
➤ IMAGEN	57
➤ COLOR	57
➤ ESCALA DE GRISES	58
➤ BINARIZACIÓN	58
➤ MACHINE VISION	58
➤ IDENTIFICACIÓN	58
➤ VENTANA SCRIPT	59

➤ VENTANA DE PROCESAMIENTO	59
3.2.2.6.2 GENERACIÓN DE UN ARCHIVO DE CLASIFICACIÓN (BASE DE DATOS)	59
➤ CREAR UN NUEVO ARCHIVO DE CLASIFICACIÓN	59
➤ CONFIGURACIÓN DE PARÁMETROS	60
➤ AÑADIR UNA CLASE	64
➤ AÑADIR UNA MUESTRA	64
➤ ENTRENAR LA CLASIFICACIÓN	66
➤ GUARDAR ARCHIVO DE CLASIFICACIÓN (.CLF)	66
➤ EDITAR CLASIFICADOR	67
3.2.2.7 CLASIFICACIÓN	68
3.2.3 ENTRENADOR	72
3.3 DIAGRAMAS DE FLUJO	74
3.3.1 DIAGRAMA DE FLUJO PRINCIPAL	74
3.3.2 DIAGRAMA DE FLUJO DE LA DETECCIÓN	75
3.3.3 DIAGRAMA DE FLUJO DE SEGMENTACIÓN	76
3.3.4 DIAGRAMA DE FLUJO DE CLASIFICACIÓN	78
3.3.5 DIAGRAMA DE FLUJO DEL ENTRENADOR	81
CAPÍTULO 4	84
PRUEBAS Y RESULTADOS	84
4.1 INTRODUCCIÓN	84
4.2 PRUEBAS REALIZADAS PARA LA SELECCIÓN DEL MÉTODO DE CLASIFICACIÓN	84
4.2.1 K-VECINO MÁS CERCANO	84
4.2.2 VECINO MÁS CERCANO	86
4.2.3 DISTANCIA MINIMA	89
4.2.4 COMPARACIÓN ENTRE MÉTODOS	91
4.3 PRUEBAS DEL FUNCIONAMIENTO DEL TRADUCTOR	93
4.4 PRUEBAS DEL FUNCIONAMIENTO DEL ENTRENADOR	94
4.5 COMPARACIÓN CON TRABAJOS SIMILARES	94

CAPÍTULO 5	95
CONCLUSIONES Y RECOMENDACIONES	95
5.1 INTRODUCCIÓN	95
5.2 CONCLUSIONES	95
5.3 RECOMENDACIONES	96
REFERENCIAS BIBLIOGRAFICAS	99
ANEXOS	102

RESUMEN

En el Ecuador existe una gran cantidad de personas con algún tipo de discapacidad física y/o mental que limita de alguna manera el desarrollo de una vida con normalidad. Entre las discapacidades que existen, nos enfocamos en las discapacidades auditiva y vocal que limitan especialmente el área de la comunicación con el resto de personas, ya que para comunicarse se requiere de medios específicos como el escrito u otros poco conocidos como el lenguaje de señas.

Con el objetivo de mejorar el nivel de vida de las personas con discapacidad auditiva y/o visual se ha desarrollado un sistema traductor de lenguaje de señas a texto usando visión artificial, para que personas que manejan el lenguaje de señas tengan la posibilidad de comunicarse con personas que no tienen conocimiento de este lenguaje. El proyecto también ofrece a las personas que no tienen conocimiento de este lenguaje, la opción de un entrenador para que aprendan cada uno de los símbolos de este lenguaje y adquieran la habilidad necesaria para utilizar la aplicación. Este proyecto se desarrolló en varias etapas que se describen brevemente a continuación: Etapa de adquisición donde la imagen se toma con un fondo negro para lograr un mejor contraste y con iluminación detrás de la cámara para que incida directamente sobre la mano para que tenga bien definidas sus características, con la menor cantidad de ruido (sombras). En el procesamiento digital de imágenes se aplicaron algunos filtros y operaciones morfológicas para resaltar las características de la imagen y eliminar información innecesaria como ruido. Con la ayuda del Vision Assistant de Labview se elaboraron las bases de datos, para llevar a cabo la comparación con la imagen recortada y de esta manera asignar la clase (letra) correspondiente a cada imagen para después formar el texto que se muestra en forma escrita en la pantalla o a su vez se puede enviar a un documento de Word.

Además, es posible reproducir mediante voz el texto formado con la ayuda del reproductor de Windows.

Los resultados obtenidos después de realizar las pruebas correspondientes, muestran que el proyecto funciona con un margen de error del 17,85% que en

comparación a otros proyectos similares con porcentajes de error del 14,6% y del 31,25% muestran que la aplicación funciona dentro de los parámetros aceptables.

PRESENTACIÓN

En este proyecto se implementa un sistema de traducción de señales de manos a texto mediante visión artificial.

Este sistema interpreta las señales de manos que el usuario realiza y las convierte a texto y a voz, el sistema también tiene un módulo entrenador, que permite a las personas que no conocen el lenguaje de señas aprenderlo. De esta forma se logra comunicar a las personas que usan este lenguaje con el mundo exterior y viceversa.

En el primer capítulo de este proyecto se realiza una introducción a cada una de las partes que componen un sistema de visión artificial, procesamiento y análisis de imágenes, así como también los métodos de clasificación usados en este proyecto.

En el segundo capítulo se definen algunos ambientes controlados para visión artificial, se explica las ventajas y desventajas de éstos, y se justifica la selección del ambiente controlado para este proyecto.

En el tercer capítulo se muestra cada uno de los algoritmos usados dentro del programa de visión artificial desarrollado en lenguaje gráfico, también se explica el desarrollo de las bases de datos que contiene la información de cada una de las señales de manos.

En el cuarto capítulo se muestra las pruebas realizadas al sistema de traducción con diferentes usuarios, y los resultados obtenidos.

En el quinto capítulo se expresan las conclusiones y recomendaciones una vez que se terminó este proyecto.

CAPÍTULO 1

FUNDAMENTO TEÓRICO

1.1 DESCRIPCIÓN DEL PROBLEMA

Actualmente en el Ecuador existe una gran cantidad de personas que padecen alguna discapacidad sea esta: visual, auditiva, física, intelectual o varias de ellas a la vez.

Según el Consejo Nacional de Discapacidades (CONADIS), del total de la población del Ecuador, el 13.2 % son personas con algún tipo de discapacidad (1`600.000 personas), y se puede señalar que en el país existen aproximadamente [1]:

- 592.000 personas con discapacidad por deficiencias físicas.
- 432.000 personas con discapacidad por deficiencias mentales y psicológicas.
- 363.000 personas con discapacidad por deficiencias visuales; y,
- 213.000 personas con discapacidad por deficiencias auditivas y del lenguaje.

En la provincia de Pichincha se tiene: 6.005 personas con discapacidad auditiva, 4.413 con discapacidad visual y 558 con discapacidad de lenguaje [1].

Las personas con este tipo de discapacidades han encontrado en el lenguaje de señas un medio de expresión y comunicación que sigue siendo limitado ya que no todas las personas lo conocen. Con el presente proyecto se busca mejorar el nivel de vida de la población con discapacidad auditiva y/o vocal específicamente, que de una u otra manera se ven limitados para comunicarse con el resto de personas.

1.2 INTRODUCCIÓN AL LENGUAJE DE SEÑAS

El lenguaje de señas, lenguaje dactilológico o lengua de signos es una forma de expresar caracteres, palabras o frases mediante signos usando principalmente las manos con el fin de comunicarse sin la necesidad de un sistema de signos vocales.

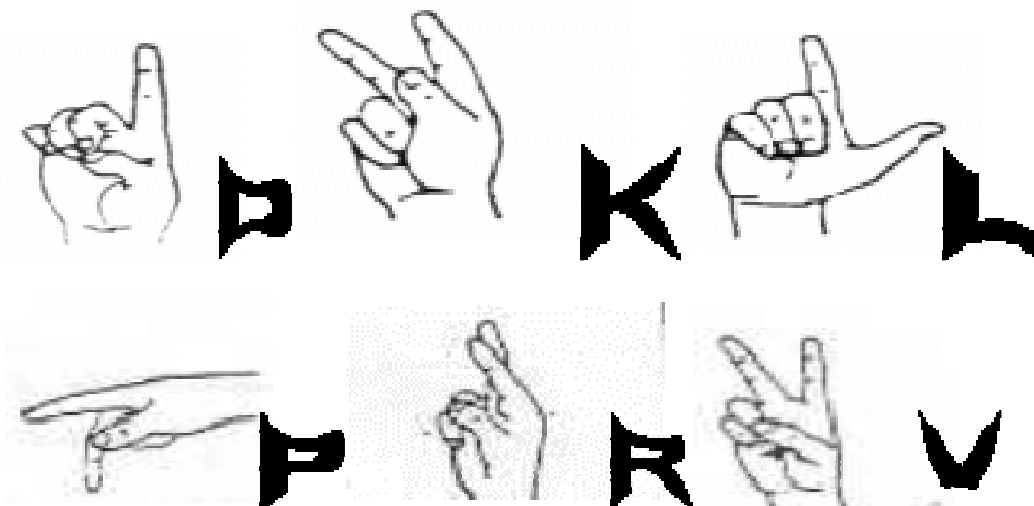


Figura 1.1 Alfabeto de Señas Ecuatoriano (Tomado de [2]).

En términos formales “la dactilología proviene del griego "*daktilos*" -dedos-, y "*logia*" -ciencia-, “ciencia de los dedos” es un sistema de comunicación que transmite información mediante el uso de los dedos de la mano. Este sistema forma parte auxiliar de la queirología de los lenguajes de señas y también se utiliza en sistemas visuales artificiales de información (árbitros deportivos, señales militares o marinas, etc.), en forma de cálculo aritmético manual, etc.” [3].

El lenguaje de señas permite que personas con discapacidad visual, auditiva o verbal se comuniquen con personas que tienen las mismas limitaciones o con personas que dentro de los márgenes se consideren normales. La ejecución de los signos de este lenguaje varía un poco dependiendo de la discapacidad, para una persona que carece del oído y/o del habla las señas se las realiza en el aire; en cambio para una persona que carece del sentido de la vista las señas se las realiza sobre la palma de la mano.

Existen varios tipos de lenguaje de signos, varían dependiendo del país, estado, provincia e incluso ciudad, por lo que no existe un lenguaje de señas universal. Países como: Venezuela, México, España, Estados Unidos, Argentina, Perú, Ecuador y otros tienen su propio lenguaje de señas. Ecuador menciona la lengua de señas ecuatoriana en el artículo 53 de la Constitución Política de 1998 [4].

Al igual que otros lenguajes poseen todas las propiedades y complejidades de cualquier lengua oral [4]:

- Poseen una fonología abstracta, llamada en este caso queirología, analizable en términos formales en rasgos de posición, orientación, configuración, en un modo análogo a como son analizados los fonemas de las lenguas orales. Además la realización de cada signo está sujeto al mismo tipo de variedad que los sonidos de las lenguas orales (variación dialectal, asimilación, cambio lingüístico).
- Poseen una sintaxis que obedece los mismos principios generales que las otras lenguas naturales, y tienen algunos mecanismos de formación de palabra productivos que permiten afirmar la existencia de procesos morfológicos.
- La adquisición de una lengua de señas por parte de bebés (sordos u oyentes) sigue un proceso paralelo a la adquisición de una lengua oral por parte de un niño oyente.
- Existen comunidades estables de hablantes, cuya lengua presenta tanto variaciones dialectales, modismos propios de cada comunidad, y está sujeto al mismo tipo de cambio lingüístico universalmente detectado en todas las lenguas naturales (las lenguas artificiales carecen de estas características).
- Las lenguas de señas, al igual que las orales, se organizan por unidades elementales sin significado propio (lexemas).

1.3 VISIÓN ARTIFICIAL

La visión artificial que también se la conoce como visión de computador o visión técnica, es una rama de estudio de la inteligencia artificial. La visión artificial tiene como propósito programar un computador para que pueda interpretar la información que contiene una imagen.

1.3.1 OBJETIVOS DE LA VISIÓN ARTIFICIAL

Los objetivos de un sistema de visión artificial, dado los requerimientos de automatización de procesos, se están enfocando para [5]:

- La cuantificación, localización, detección, de objetos dentro de una imagen.

- Reconocimiento y seguimiento de un objeto dentro de una imagen.
- Reconocimiento y clasificación de los objetos de una imagen.
- Navegación automática de robots móviles.
- Construcción de posturas tridimensionales de humanos.

1.3.2 ELEMENTOS DE UN SISTEMA DE VISIÓN ARTIFICIAL

Un sistema de visión artificial se conforma de los siguientes elementos [6]:

1.3.2.1 Iluminación

El éxito de un sistema de visión artificial depende mucho más de un buen sistema de iluminación antes que del algoritmo de extracción de la información de la imagen.

Un buen sistema de iluminación reduce altamente el costo del preprocesamiento de la imagen, puesto que una imagen no es más que una representación de la información de la luz, por lo que la información de la escena será función de [6]:

- Las fuentes de luz.
- Las características de la superficie (reflexión, rugosidad).
- Las relaciones espaciales (distancias y ángulos) entre sensores, superficies y fuentes.

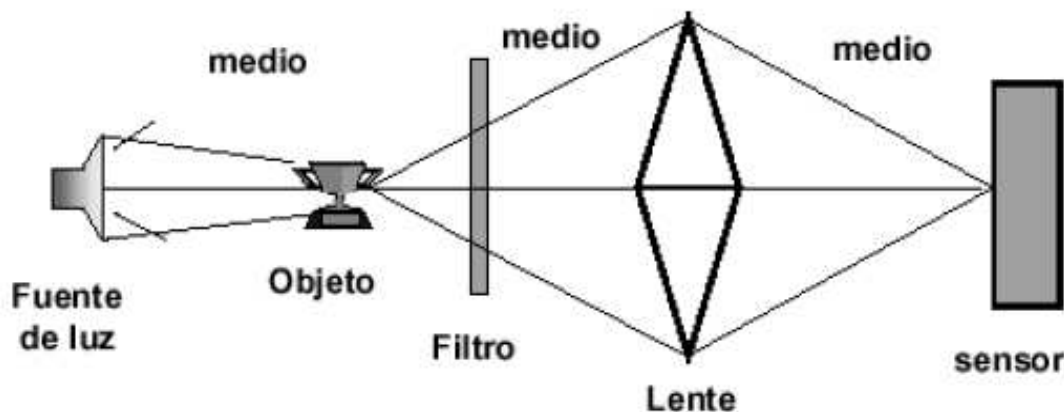


Figura 1.2 Representación de la información de la escena (Tomado de [7]).

Existen varias técnicas de iluminación que de acuerdo a la aplicación, se las puede dividir en grupos por ejemplo:

1.3.2.1.1 *Direccionales*

Se utiliza en la localización y reconocimiento de piezas, inspección de superficies, seguimiento de soldadura o pegamento.



Figura 1.3 Iluminación Direccional (Tomado de [6]).

1.3.2.1.2 *Difusas*

Este tipo de iluminación busca que los haces luminosos lleguen al objeto desde todas direcciones y no solo desde la fuente luminosa, de esta forma se eliminan las sombras. Se usa en superficies suaves.

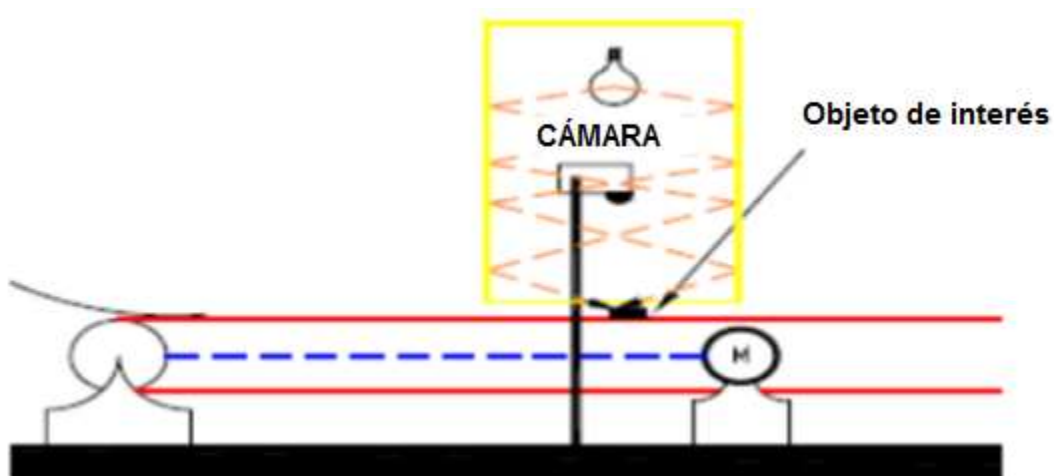


Figura 1.4 Iluminación Difusa (Tomado de [6]).

1.3.2.1.3 A contra luz

En este caso se coloca la fuente de iluminación detrás del objeto de análisis, se usa para localización y análisis dimensional de piezas, se debe tener en cuenta que se puede deslumbrar a la cámara lo que causa la pérdida de información en la imagen.

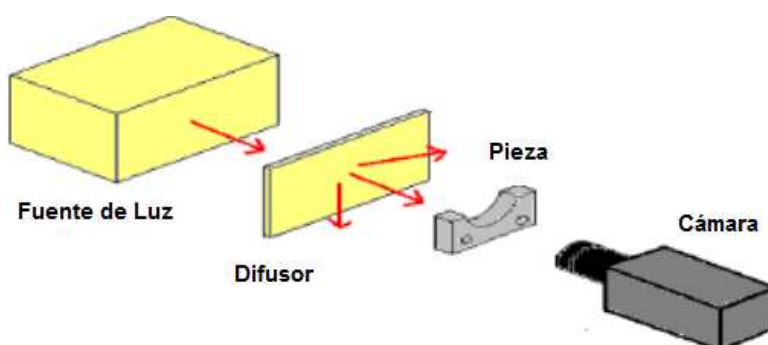


Figura 1.5 Iluminación a Contra Luz (Tomado de [6]).

1.3.2.1.4 Estructuradas

Este tipo de iluminación se sirve de la proyección de puntos franjas o rejillas sobre la superficie de trabajo. En función de cómo se deforme este patrón de luz se detecta las singularidades del objeto.

Las ventajas de este método es que el patrón de luz es constante y conocido, analizando los cambios en este patrón se obtiene las características tridimensionales del objeto.

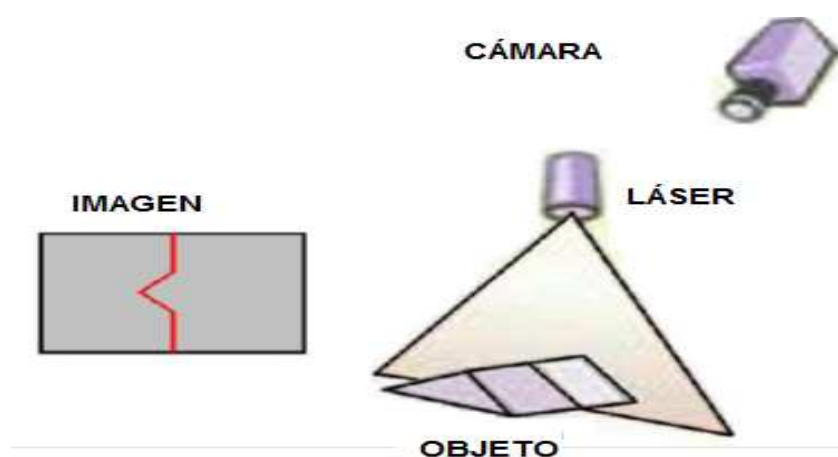


Figura 1.6 Iluminación Estructurada (Tomado de [6]).

1.3.2.2 Sensor de imagen

Los sensores que se usan en visión artificial son las cámaras en estado sólido, que pueden ser dispositivos de cargas acopladas CCD, o CMOS.

Las cámaras en estado sólido se basan en la entrada de luz sobre un elemento sensitivo, que es un semiconductor. Los sensores están en arreglos rectangulares donde cada elemento se denomina foto celda o foto sensor.

Un fotosensor es un convertidor fotoeléctrico debido a que se carga con un nivel de energía proporcional a la cantidad de luz que incide sobre él durante un tiempo de integración.

La energía captada en el arreglo de fotosensores puede ser transmitida directamente fuera del arreglo (salida digital) o procesada en tiempo variable dentro de una señal de video (salida analógica).

1.3.2.3 Tarjetas de captura

Estas tarjetas son las que se encargan de transferir la imagen desde la cámara a la memoria de la computadora, para que realice el procesamiento de las imágenes.

Los principales aspectos que se deben tener en cuenta en una tarjeta de adquisición de datos son: la velocidad de transmisión, el formato de los datos, la profundidad del píxel, la capacidad de captura por disparo, la capacidad de controlar los parámetros de la cámara en tiempo real.

1.3.2.4 Algoritmos de análisis de imagen

Es la parte inteligente del sistema. Su función consiste en aplicar las diferentes transformaciones, filtros que sean necesarios para la extracción de información de las imágenes adquiridas, para que se tomen las decisiones para las que fue diseñado.

1.3.2.5 Computadora o módulo de proceso

Es el sistema encargado de ejecutar el algoritmo de análisis en las imágenes adquiridas por el sensor, para extraer la información de interés. Éste es el cerebro del sistema de visión artificial.

1.3.2.6 Sistema de respuesta en tiempo real

El sistema de visión artificial con la información extraída puede tomar decisiones sobre el sistema productivo con la finalidad de mejorar el rendimiento de la producción.

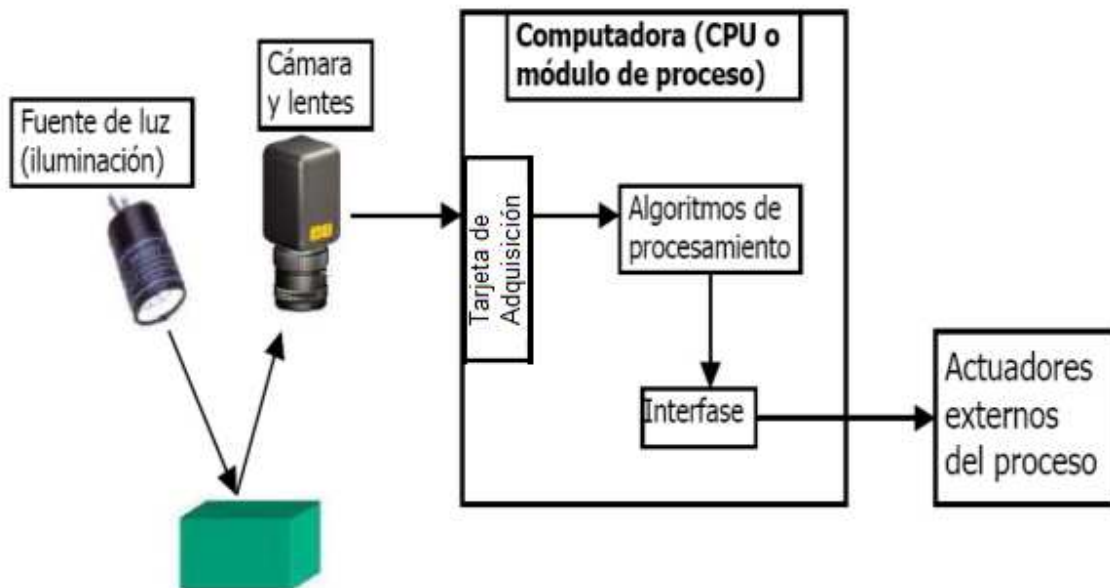


Figura 1.7 Elementos de un sistema de visión artificial (Tomado de [8]).

1.3.3 ETAPAS DE UN SISTEMA DE VISIÓN ARTIFICIAL

Todos los sistemas de visión artificial tienen una estructura específica, pero se puede generalizar en los pasos que se nombran a continuación [6]:

- Construcción del sistema de formación de imágenes
- Pre-procesamiento
- Segmentación
- Representación y descripción
- Reconocimiento e interpretación

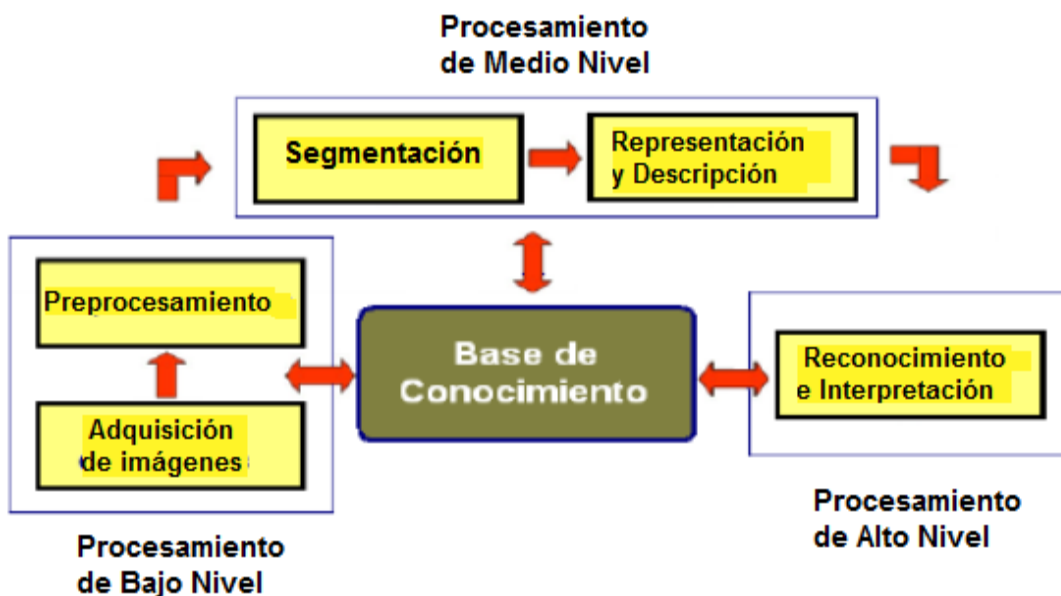


Figura 1.8 Etapas de un sistema de visión artificial (Tomada de [6]).

1.3.4 COMPARACIÓN ENTRE LOS SISTEMAS DE VISIÓN ARTIFICIALES Y HUMANOS

Todos los sistemas de visión artificial procuran trabajar de manera similar al sistema de visión de los humanos:

Fuente de iluminación: el sistema de visión de un ser humano de igual forma que un sistema de visión artificial trabaja cabalmente dentro de condiciones mínimas de luz para que puedan interpretar de manera correcta la información que contiene la escena.

Sensores de imagen: el trabajo que realizan el iris, el cristalino, la pupila, la retina, los conos, los bastones y todos los componentes del globo ocular en el ser humano, en un sistema de visión artificial lo realizan las cámaras que poseen un lente, sensores fotoeléctricos en un arreglo CCD, un objetivo, un diafragma y demás componentes de la cámara.

Sistema de adquisición: la adquisición en el ser humano la realiza el nervio óptico que se encarga de llevar la información proporcionada por los conos y los bastones hacia el cerebro en forma de pulsos eléctricos; el sistema de visión artificial posee un sistema de adquisición que transforma la información eléctrica analógica de la naturaleza en una señal digital.

Aplicación de procesado: mientras que el cerebro puede interpretar la información de los objetos dentro de una imagen y tomar una decisión de acuerdo a sus necesidades, un sistema de visión artificial para tomar decisiones de los objetos depende del programa de aplicación que se encuentre ejecutando un computador.

1.4 PROCESAMIENTO DIGITAL DE IMÁGENES

1.4.1 DEFINICIÓN DE UNA IMAGEN [9]

Una imagen digital está formada por millones de píxeles ordenados en líneas y columnas. El píxel es una unidad de información pero no corresponde con un tamaño concreto. Un píxel puede ser muy pequeño en unidades de mm o muy grande en unidades de cm.



Figura 1.9 Una imagen digital consta de píxeles yuxtapuestos (Tomada de [9]).

El color que tiene cada píxel se obtiene de la combinación de tres elementos luminóforos: uno para el rojo, otro para el azul y otro para el verde. Esta distribución de colores se llama RGB (Red, Green, Blue). La combinación de estos elementos puede reproducir casi cualquier color del espectro visible que son más de 16 millones de colores.

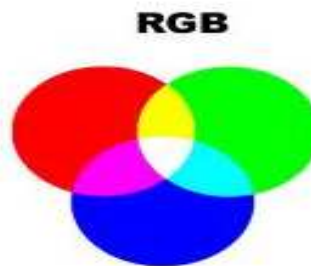


Figura 1.10 Esquema del espacio RGB (Tomado de [9]).

1.4.1.1 Tipos de imágenes

1.4.1.1.1 Imagen a color

Una imagen a color está formada de tres diferentes planos en un solo espacio, por ejemplo el espacio RGB (red, green, blue). Cada pixel está formado por tres valores entre 0 y 255 de cada uno de los planos (red, green, blue). En todos los modelos de imagen de color, un valor adicional de 8 bits es dejado vacante de ahí el nombre 4*8 o 32 bits. Existen otros espacios para representar una imagen a color por ejemplo: HSL, CMY.

✓ *Espacio RGB*

Este espacio es conocido como el sistema, o espacio de color aditivo. Todos los colores que se pueden crear en este espacio es la mezcla de estos tres colores, cuando ningún color está presente se percibe el color negro. Este sistema es usado en monitores televisores, proyectores de video, y todo sistema que utilizan combinaciones que fosforecen en el rojo, verde y azul

✓ *Espacio CMY*

En este espacio los colores que se pueden crear son la mezcla substractiva de los colores cian, magenta y amarillo, que son los colores secundarios de la luz o bien los colores primarios de los pigmentos. La mezcla de los colores CMY da el negro, pero en baja calidad, por lo que se incluye un cuarto color K que representa el negro. Recientemente se ha demostrado que este espacio es más preciso para mezclas de pigmentos.

✓ *Espacio RYB*

En este espacio los colores primarios son el rojo, amarillo y azul el resto de colores puros (color materia) pueden ser creados mezclando estos colores. Este espacio RYB es utilizado en general en conceptos de arte y pintura tradicionales, aunque este espacio es usado para la representación de pigmentos el modelo RYB no representa con precisión los colores que debería resultar de mezclar los 3 colores RYB primarios.

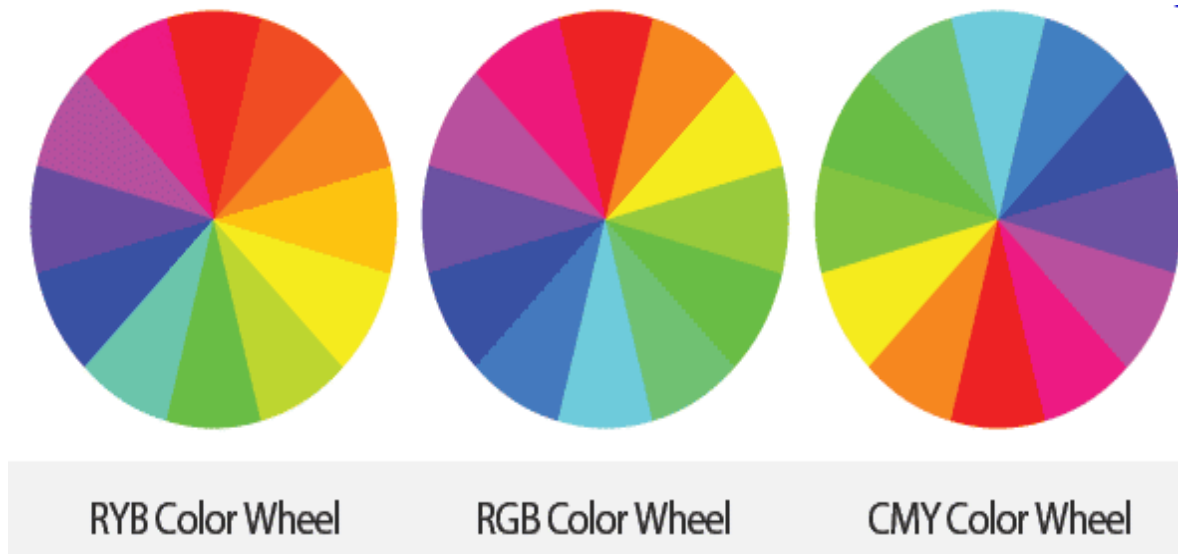


Figura 1.11 Espacios RYB, RGB, CMY (Tomado de [10]).

1.4.1.1.2 *Imagen en escala de grises*

Una imagen en escala de grises es una de un solo plano en el que se representa la luminancia de cada pixel y se la coloca dentro de una escala que depende del tamaño de la imagen. Por ejemplo, una imagen de grises de 8 bits, cada pixel tendrá un valor entre 0 y 255, en una imagen en escala de grises de 16 bits, cada pixel tendrá un valor entre 0 y 65535.

1.4.1.1.3 *Imagen binaria*

Una imagen binaria es una función $f[m, n]$ en donde cada pixel toma valores de cero o uno que son representados con dos colores cualesquiera los más comunes que se usa son el blanco y el negro, el valor que tome cada pixel de uno o cero dependerá de su nivel de gris.

Para convertir a una imagen binaria se utiliza la umbralización, la que consiste en definir dos límites uno inferior y otro superior, entonces los pixeles con el nivel de gris entre los límites se representa con un uno y los pixeles fuera de los límites se los representa con cero.

La imagen binaria es la manera más sencilla de representar una imagen digital, pero en contraste es la que más información posee para los sistemas de visión artificial.

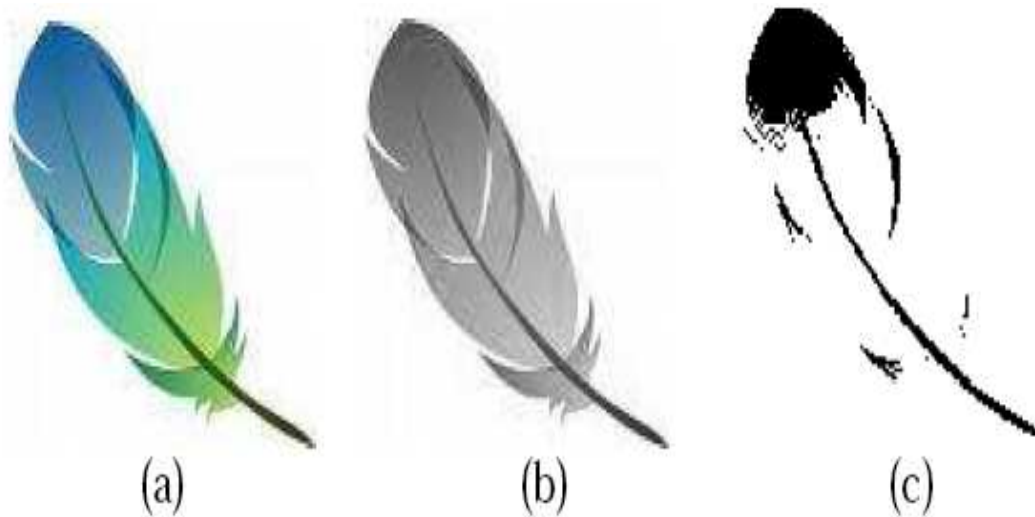


Figura 1.12 Tipos de imágenes digitales; (a) RGB; (b) Escala de grises; (c) Binaria (Tomado de [11]).

1.4.1.1.4 Histograma de una imagen

El histograma de una imagen es una curva donde se representa la frecuencia con la que aparece cada nivel de gris en una imagen. Se representa en el eje x el nivel de gris (de 0 a 255 en una imagen en escala de grises) y en el eje y el porcentaje de cada nivel de gris, tomando en cuenta que en una imagen en escala de grises el valor cero representa el color negro y el 255 representa el blanco.

Es una función discreta $h(r_k) = n_k$ donde r_k es el nivel de gris k -ésimo y n_k es el número de píxeles en la imagen con nivel de gris r_k .

Para normalizar el histograma se divide cada uno de sus valores entre el número total de píxeles en la imagen, n , el histograma normalizado se obtiene de $p(r_k) = n_k/n$ para $k = 0, 1, 2, \dots, L-1$. La suma de las componentes normalizados del histograma es igual a 1.

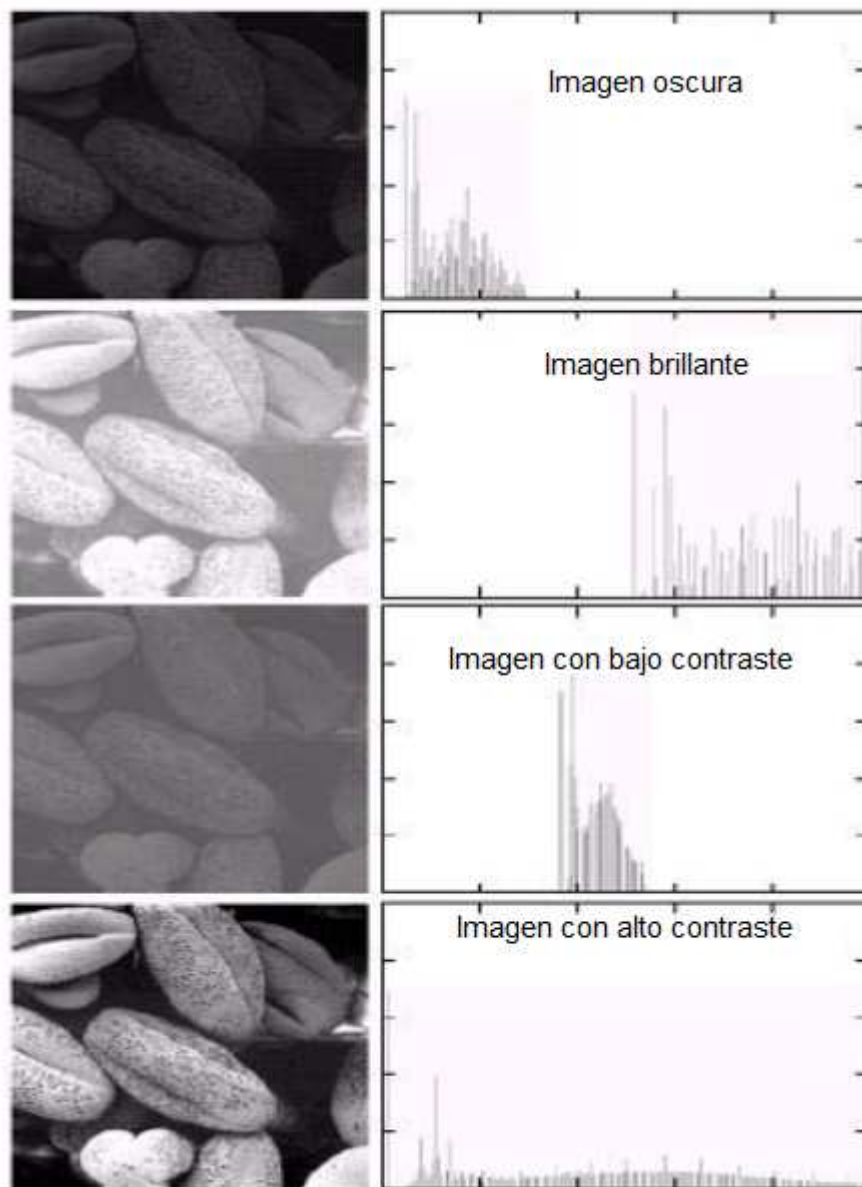


Figura 1.13 Ejemplo del histograma para 4 tipos de imágenes (Tomado de [6]).

El histograma permite analizar donde se encuentra la mayor cantidad de información de la imagen, esta información se usa para operaciones posteriores como por ejemplo, la binarización, compresión de la imagen.

1.4.2 PRE-PROCESAMIENTO DIGITAL DE IMÁGENES

La etapa de procesamiento digital de imágenes tiene como propósito el mejoramiento de una imagen; es decir, eliminar el ruido, atenuar imperfecciones, mejorar el contraste (luminosidad) y el realce de características, a través de filtros aplicados a la imagen. En esta etapa se incluye la conversión de la imagen para

llevar a cabo el procesamiento. Generalmente se trabaja con imágenes en escala de grises de 8 o 16 bits de profundidad. Estas operaciones se las puede resumir en los grupos siguientes [6]:

- Algebraicas entre imágenes.
- Mejoramiento de relación de señal-ruido (denoising).
- Atenuación de imperfecciones en la adquisición debido a la función de transferencia del sistema de captación de imágenes (deconvolución).
- Mejora de contraste o de optimización de la distribución de intensidad (enhancement).
- Realce de características específicas, como por ejemplo los bordes de la imagen.

Los filtros que se aplican son de dos tipos: filtros en el dominio espacial y filtros en el dominio de la frecuencia.

1.4.2.1 Filtros en el dominio espacial

Los filtros en el dominio espacial se aplican directamente sobre los píxeles de una imagen. Toman la información de cada uno de los píxeles próximos al píxel objetivo que está siendo analizado, para recolectar información importante y aplicar el filtro a dicho píxel, de esta manera se consigue el mejoramiento de la imagen.

Estos filtros se clasifican en:

- Filtros lineales (filtros basados en kernels o máscaras de convolución).
- Filtros no lineales.

1.4.2.1.1 Filtros lineales

Estos filtros se usan para suavizar, transformar y remover el ruido en una imagen en escala de grises, cambian el valor de un píxel en base a un algoritmo que usa los valores de los píxeles próximos. Este algoritmo se basa en una matriz de

Kernell o máscara en donde cada elemento de la matriz es un coeficiente. Esta máscara se mueve por toda la imagen para que se aplique a todos los pixeles.

“El resultado del filtrado espacial lineal en un pixel es la suma de los productos de los coeficientes de la máscara por los valores de los pixeles del vecindario”. El resultado de un filtrado espacial lineal aplicado a un pixel es [12]:

$$P_{i,j} = \frac{\sum(K_{a,b} \times P_{a,b})}{N}$$

Donde “a” está entre $i-1$ e $i+1$ y “b” está entre $j-1$ e $j+1$ y “N” es el factor de normalización que toma el valor mayor entre 1 y $\sum K_{a,b}$.

Tipos de filtros espaciales lineales

Filtro Pasa Altos.- En este filtro el Kernell tiene coeficientes negativos, permite el aumento del brillo.

Filtro Pasa Bajos.- Si tiene coeficientes positivos, es un *filtro* de suavizado.



Figura 1.14 Ejemplo de Filtro Pasa Bajos (Tomado de [12]).

Realce de bordes por desplazamiento y diferencia.- Consiste en hacer una copia de la imagen original y moverla para localizar y realzar los bordes. Se usan Kernels: horizontales, verticales y horizontales/verticales (diagonales).

Realce de Bordes mediante Laplace.- Este filtro realza los bordes en todas las direcciones, resalta la intensidad de luz alrededor de un pixel; se aplica la segunda derivada para mejores resultados.

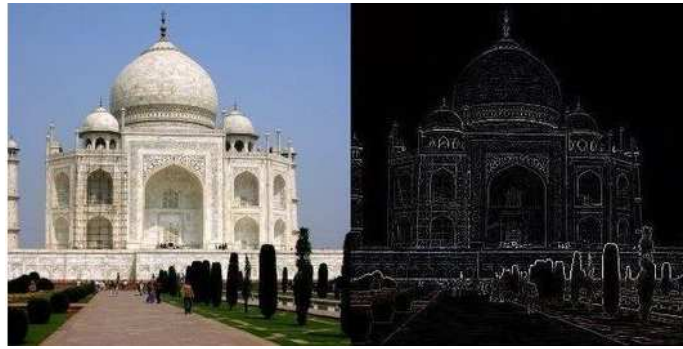


Figura 1.15 Ejemplo de Filtro Laplaciano (Tomado de [12]).

Resalte de bordes con gradiente direccional.- Emplea la intensidad de los pixeles vecinos para resaltar los bordes en determinada dirección con mayor precisión.

Detección de bordes y filtros de contorno.- Son conocidos como Prewitt y Sobel, son usados especialmente para detectar formas ya que resalta los contornos basándose, al igual que los anteriores, en la intensidad de los pixeles vecinos.

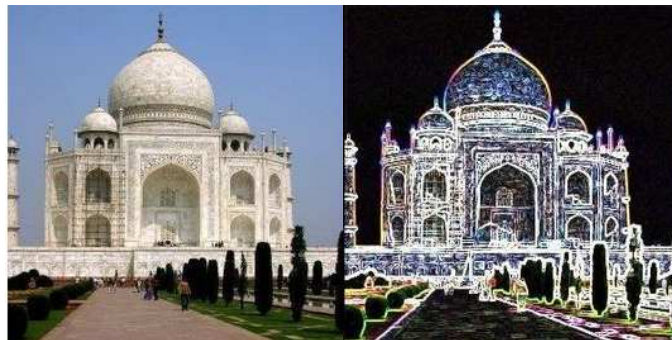


Figura 1.16 Ejemplo de Filtro Sobel (Tomado de [12]).

1.4.2.1.2 Filtros en el dominio de la frecuencia

Estos filtros que trabajan en el dominio de la frecuencia se aplican sobre la Transformada de Fourier de la imagen, por lo cual ésta se modifica mediante el teorema de la convolución.



Figura 1.17 Etapas del procesamiento de imágenes en el dominio de la frecuencia (Tomado de [12]).

Existen tres tipos de filtros en el dominio de la frecuencia:

Filtro pasa bajos.- Se encarga de filtrar altas frecuencias que equivalen a cambios bruscos de intensidad, es decir elimina el ruido. Este filtro es equivalente al filtro de suavizado en el dominio espacial.

Filtro pasa alto.- Este filtra las bajas frecuencias que se encuentran especialmente en los bordes de una imagen, por lo es usado para la detección de bordes además que refuerza el contraste de la imagen.

Filtro pasa banda.- Sirve para filtrar frecuencias muy altas o muy bajas.

1.4.3 SEGMENTACIÓN

Segmentar una imagen es un proceso que consiste en separar la imagen digital en regiones en las que se resalte alguna característica, por ejemplo, el brillo, la forma, el color. Este proceso se realiza con el fin de facilitar un posterior análisis o reconocimiento automático, como por ejemplo clasificación de piezas, o señales de manos en el presente proyecto.

“La segmentación debe verse como un proceso que a partir de una imagen, produce otra en la que cada pixel tiene asociada una etiqueta distintiva del objeto al que pertenece. Así, una vez segmentada una imagen, se podría formar una lista de objetos consistentes en las agrupaciones de los píxeles que tengan la misma etiqueta [13].”

El proceso de segmentación de una imagen depende mucho del problema que se desee resolver. En el presente proyecto, la segmentación consiste en aislar la forma de la mano del ambiente extrayendo una imagen de la forma para ser clasificada.

Las técnicas de segmentación más desarrolladas en la actualidad son [6]:

- Segmentación basada en detección de bordes.
- Segmentación basada en morfología.
- Segmentación basada en la transformada de Hough.

1.4.3.1 Segmentación basada en detección de bordes

Este tipo de segmentación agrupa un gran número de técnicas que usan la información por las fronteras de los objetos que aparecen en una imagen.

Puesto que se desea encontrar los objetos individuales presentes en una imagen, parece lógico que si se encuentra las fronteras de tales objetos con el fondo se podría segmentar los objetos de la escena general. La segmentación por bordes en su mayoría es implementada en base a filtros espaciales como por ejemplo: filtro de gradiente, filtro, laplaciano, etc.



Figura 1.18 Segmentación basada en detección de bordes (Tomada de [6]).

1.4.3.2 Segmentación basada en morfología

“La segmentación basada en morfología se basa en la teoría del conjunto, el objetivo de las transformaciones morfológicas es la extracción de estructuras geométricas en los conjuntos sobre los que se opera, mediante la utilización de otro conjunto de forma conocida, al que se denomina elemento estructurante. Este elemento se elige a priori, de acuerdo con la morfología sobre la que se va a intersecar y en función de las formas que se desea extraer” [6].

Es importante tener muy claro que es una imagen binaria dado que la segmentación por morfología se aplica a este tipo de imágenes.

Para este tipo de morfología se debe tener en cuenta la diferencia entre vecindad y conectividad.

Vecindad.- La vecindad indica si hay pixeles de valor uno interconectados o adyacentes, de tal forma que se pueda definir una agrupación de pixeles o blob. Existen la vecindad de 4 y de 8, la vecindad de 4 considera pixeles conectados en direcciones perpendiculares al pixel central, la vecindad de 8 considera pixeles ubicados en las diagonales.

Conectividad.- La conectividad define la manera que los pixeles están interconectados en una imagen, esta conectividad establece una conexión entre los blobs y así formar una imagen.

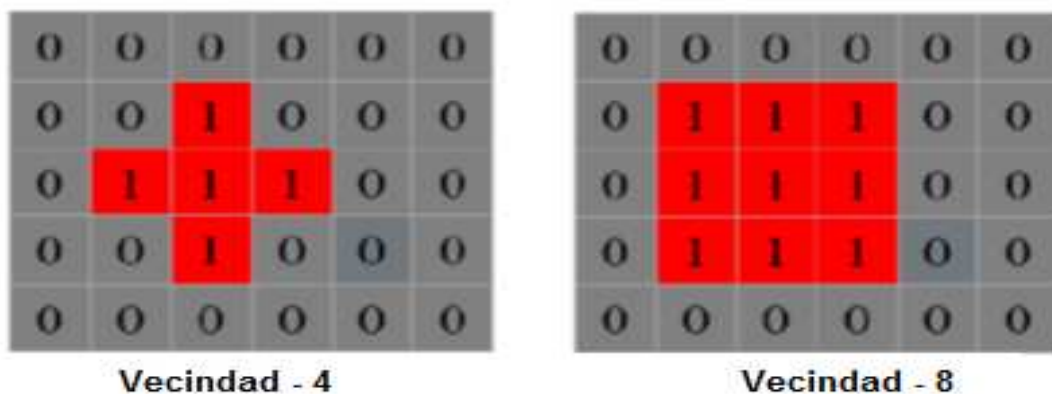


Figura 1.19 Tipos de vecindad (Tomado de [6]).

Una vez aclarados estos conceptos se procede a enumerar las operaciones existentes dentro de este tipo de segmentación:

1.4.3.2.1 Dilatación

Al dilatar una imagen lo que se logra es aumentar el tamaño de los objetos contenidos en ésta. La dilatación es una operación que aumenta capas a los objetos o blobs aun los que tengan agujeros. La fórmula que expresa la dilatación es la siguiente [5]:

$$A \oplus B = \{c \in E^N | c = a + b \text{ para todo } a \in A \text{ y } b \in B\}$$

El resultado es la suma de A más B negado conforme se desplaza en cada uno de los pixeles de la imagen.

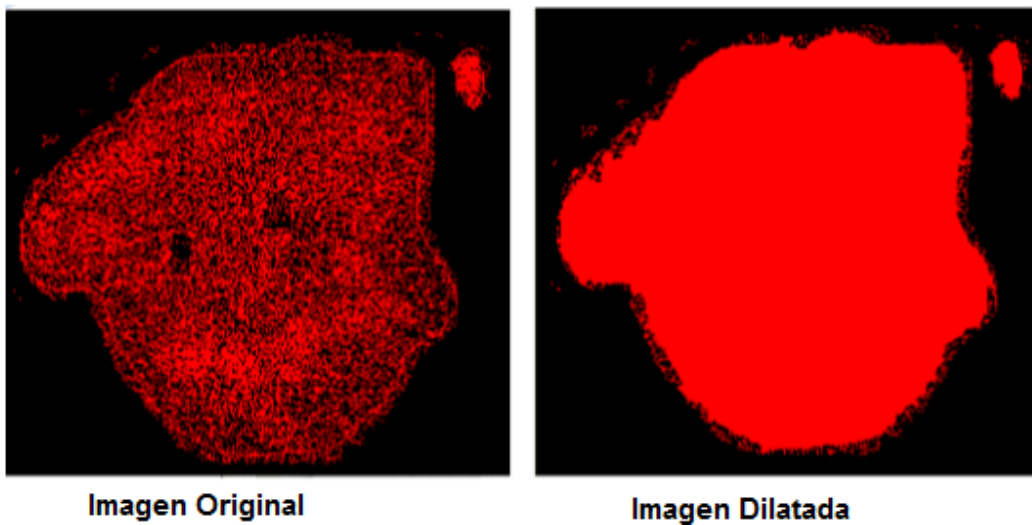


Figura 1.20 Ejemplo de la dilatación de una imagen (Tomado de [6]).

1.4.3.2.2 Erosión

La erosión es la operación que reduce el tamaño de los blobs dentro de una imagen. La erosión es una operación dual a la dilatación, pero no la inversa; es decir, que si se aplica la dilatación a una imagen y luego se aplica la erosión no se obtiene la imagen original. La ecuación es la siguiente [6]:

$$A \ominus B = \{x \in E^N \mid x + b \in A \text{ para todo } b \in B\}$$

El resultado de la erosión es un conjunto de píxeles tales que B movido z, esté contenido en A; es decir, se resta A menos B conforme se desplaza, con esto se logra remover partículas pequeñas fuera de los blobs. Esta operación se la usa para eliminar el ruido de la imagen binaria.

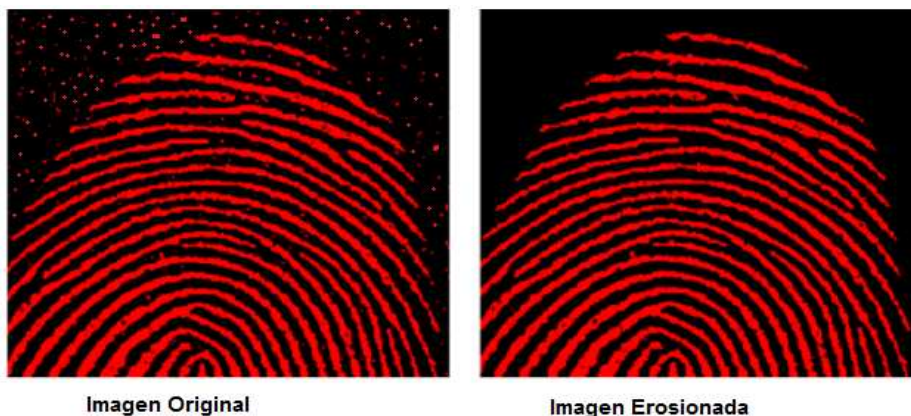


Figura 1.21 Operación de Erosión (Tomado de [6]).

1.4.3.2.3 Apertura

Esta operación consiste en realizar en una imagen una erosión y después una dilatación. Como se explicó antes, la erosión no es la inversa a la dilatación por tal motivo la imagen resultante no es la original.

Esta operación se usa para descomponer objetos en elementos más simples, eliminar salientes estrechos, separar objetos que no están demasiado pegados y aumentar los agujeros de los objetos. Esta operación puede estar formada de un número dispar de erosiones y dilataciones produciendo resultados diferentes. La ecuación es la siguiente [6].

$$A \circ B = (A(-)B) \oplus B$$

A continuación se puede visualizar un ejemplo de apertura de una imagen:

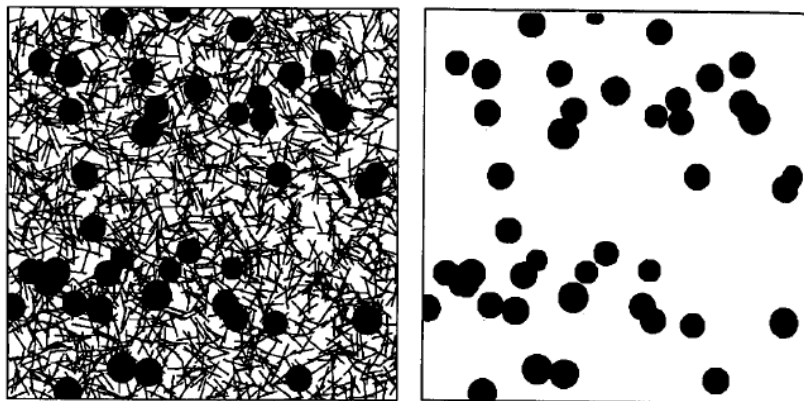


Figura 1.22 Apertura de una imagen (Tomado de [14]).

1.4.3.2.4 Cierre

Esta operación hace exactamente lo contrario que la apertura, primero realiza la dilatación y después una erosión. Esta operación suaviza los contornos de los objetos y remueve agujeros; pero la desventaja es que esta operación no elimina partículas pequeñas y salientes de las imágenes. La ecuación es la siguiente [6].

$$A \cdot B = (A \oplus B)(-)B$$

Ejemplo de una imagen a la que se le aplicó la operación de cierre [6].

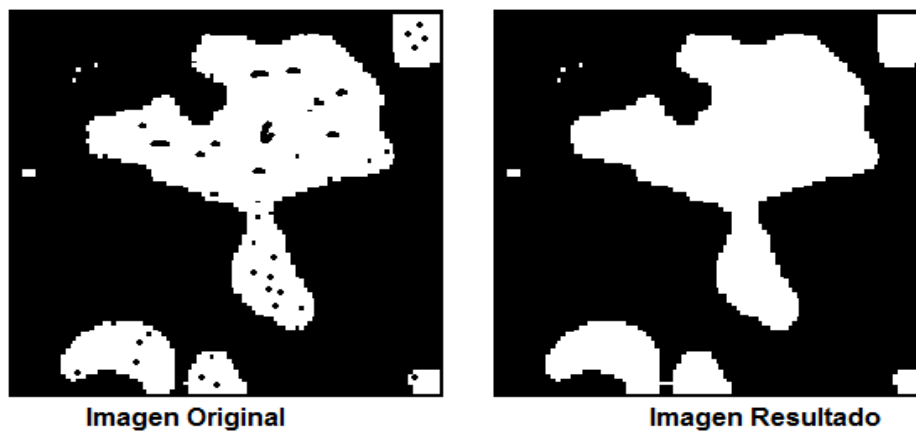


Figura 1.23 Operación de cierre (Tomado de [14]).

1.4.3.2.5 Detección de bordes

Un borde es un cambio significativo en los valores de la escala de grises entre los píxeles adyacentes en una imagen. La región de búsqueda puede tomar la forma de una línea, el perímetro de un círculo o de una elipse, la frontera de un rectángulo o un polígono, o de una región a mano alzada. El programa analiza los valores de los píxeles a lo largo del perfil para detectar cambios significativos en la intensidad.

En la siguiente figura se ilustra un modelo común que se utiliza para caracterizar un borde.

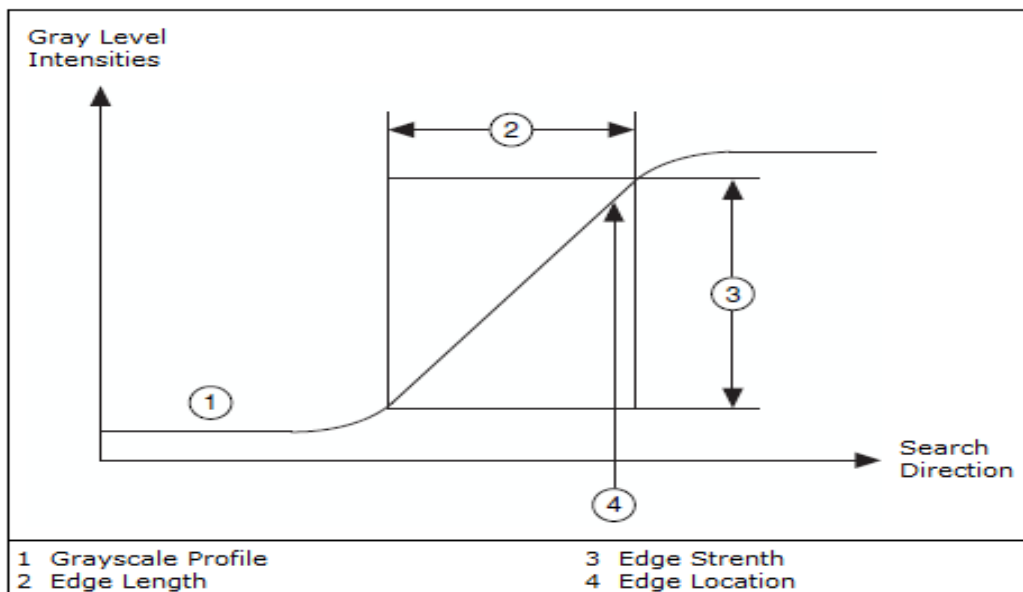


Figura 1.24 Modelo común para detección de un borde (Tomado de [15]).

Este modelo depende principalmente de los siguientes parámetros:

La fuerza del borde.- También se la llama el contraste de los bordes, y es la mínima diferencia en los valores de escala de grises entre el fondo y el borde. La fuerza del borde puede variar por las siguientes razones:

- Las condiciones de iluminación; y,
- Objetos con diferentes características en la escala de grises.

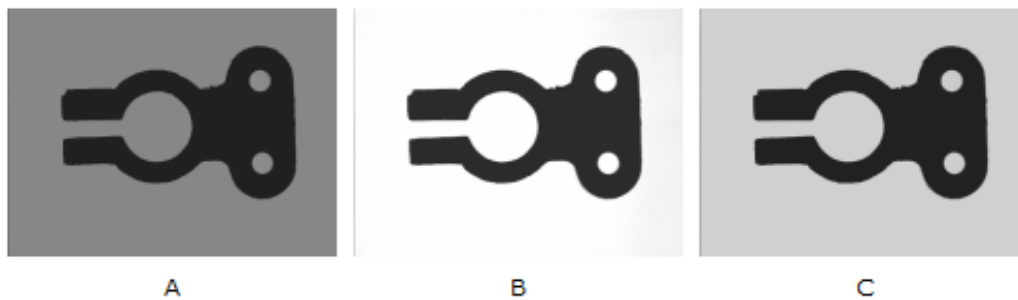


Figura 1.25 Factores de iluminación en detección de bordes (Tomado de [15]).

La distancia del borde.- Es la distancia en la que la diferencia de píxeles en la escala de grises entre el borde y el fondo debe ocurrir.

Ubicación del borde.- Este parámetro describe la posición x, y de un borde en la imagen.

Polaridad del borde.- Este parámetro define si un borde sube o baja, un borde que sube se caracteriza por un aumento en los valores de escala de grises al cruzar el borde, un borde que desciende se caracteriza por una disminución en los valores de escala de grises al cruzar el borde.

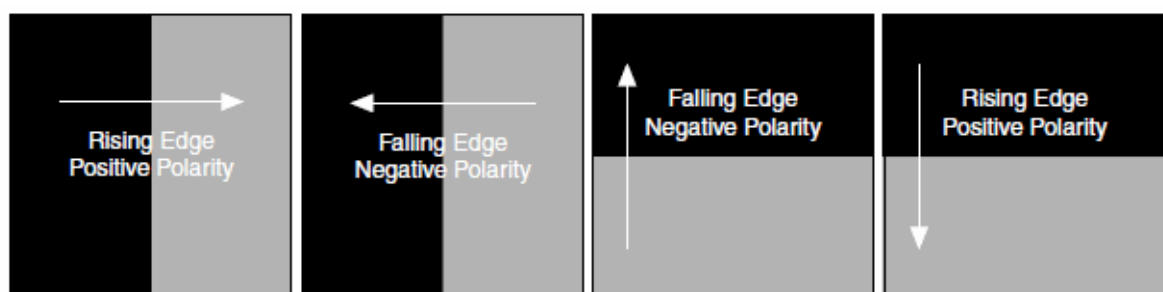


Figura 1.26 Sentidos de orientaciones posibles para detección de bordes (Tomado de [15]).

1.4.3.2.6 Métodos para detección de flancos: Detección de bordes simple [15]

Se utiliza el valor de un pixel en cualquier punto a lo largo del perfil de píxeles para definir la resistencia de los bordes en ese punto, se analiza pixel a pixel desde el principio hasta el final. Un flanco de subida se detecta en el primer punto en el que el valor del pixel es mayor que el valor de umbral, más un valor de histéresis. Una vez que se detecta el flanco de subida se busca el flanco de bajada, un flanco de bajada se detecta cuando el valor del pixel cae por debajo del valor de umbral especificado, este método funciona bien cuando la imagen presenta un nivel de ruido bajo.

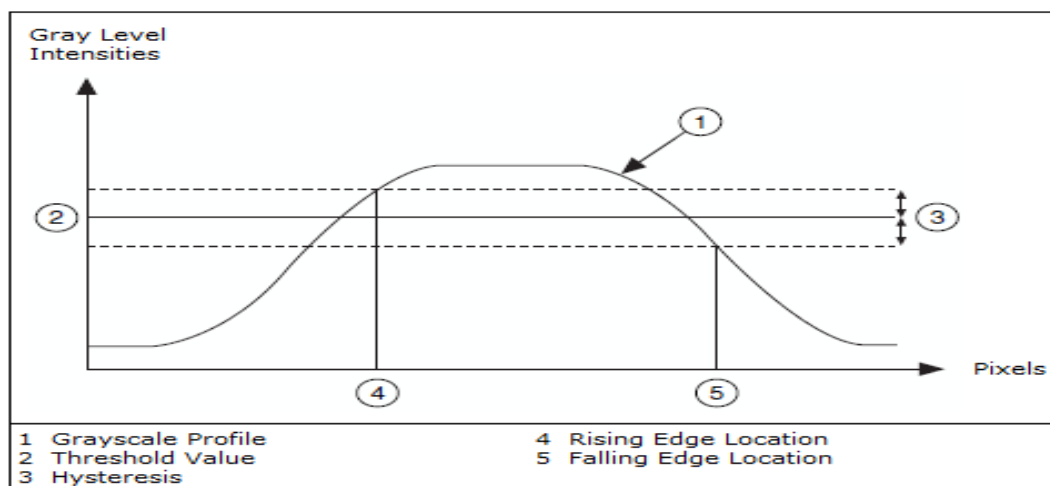


Figura 1.27 Limite de histéresis característico de un borde (Tomado de [15]).

1.4.4 CLASIFICACION DE IMÁGENES

La clasificación de imágenes se realiza mediante algoritmos de clasificación basados en una función de distancia. Permite asignar a una imagen, o conjunto de imágenes, una clase, mediante la comparación de características de las imágenes con las características de otras tomadas como muestra. Para llevar a cabo un proceso de clasificación se siguen las siguientes etapas [15]:

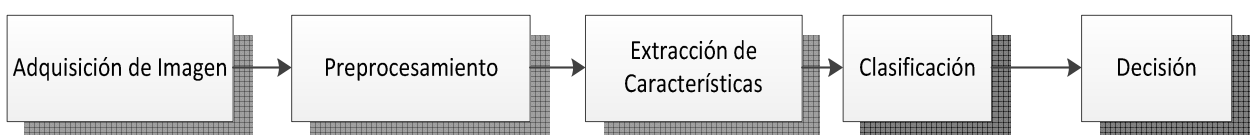


Figura 1.28 Diagrama funcional de las fases de clasificación (Tomado de [15]).

Para la clasificación de imágenes basadas en la definición de proximidad se tienen tres algoritmos:

- Vecino más cercano
- K -vecino más cercano
- Mínima Distancia

1.4.4.1 Vecino más Cercano [15]

Este método consiste en tomar una imagen X y asignarla a determinada clase C_j mediante la medida de la distancia de la imagen a clasificar a la muestra más cercana que representa a la clase.

$$X \in \text{Class } C_j, \text{ si } d(X, C_j) = \min_i d(X, C_i)$$

En algunas aplicaciones de clasificación, una clase puede ser representada por varias muestras que no están en el mismo grupo, como se muestra en la Figura 1.29.

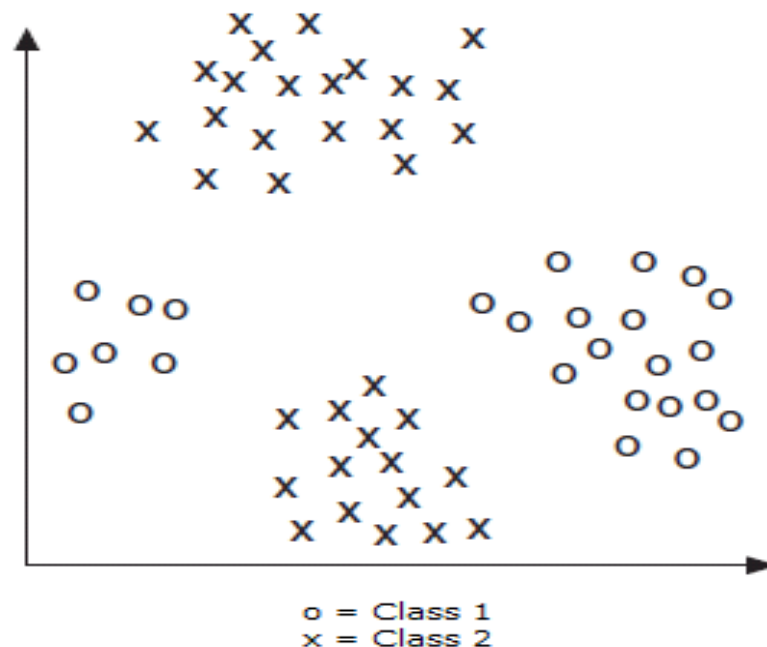


Figura 1.29 Ejemplo de Clasificación (Tomado de [15]).

En estas aplicaciones, el clasificador del vecino más cercano es más eficaz que el clasificador de la mínima distancia media.

1.4.4.2 Mínima Distancia [15]

Consiste en crear un vector de muestras (prototipo) para cada clase. Este método calcula la distancia euclidea entre el patrón desconocido y los vectores prototipo, y selecciona la distancia más pequeña para asignar una clase.

La distancia de la imagen de entrada X al centro de cada clase viene dada por:

$$X \in \text{Class } C_j, \text{ si } d(X, M_j) = \min_i d(X, M_i)$$

1.4.4.3 K-Vecino más Cercano [16]

Este es un método de clasificación no paramétrico, se basa en la estimación de la densidad de probabilidad o, directamente, en la probabilidad que X pertenezca a determinada clase.

Este algoritmo es más robusto frente al ruido que el algoritmo del vecino más cercano. Consiste en un sistema de votación que depende del número k de muestras que se tomen para realizar la clasificación.

Se toma un número k de las muestras más cercanas, de estas k muestras cercanas se cuentan las de mayor número que pertenezcan a una misma clase y de acuerdo a esto se asigna la clase.

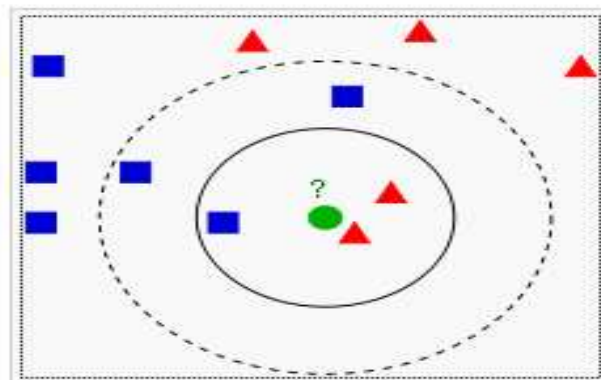


Figura 1.30 Ejemplo del Algoritmo k -vecino más cercano (Tomado de [16]).

De la Figura 1.30 se observa que para el círculo de menor radio, con $k = 3$, el círculo es clasificado con los triángulos ya que estos tiene el mayor número de muestras ($m = 2$); en cambio para el círculo de mayor radio con $k = 5$, el círculo es

clasificado con los cuadrados, porque corresponde al mayor número de muestras ($m = 3$).

En este capítulo se ha abordado los conceptos generales sobre visión artificial enfocados al preprocesamiento, procesamiento y reconocimiento de imágenes con el fin de establecer un conocimiento previo, facilitando el entendimiento de la aplicación realizada.

CAPÍTULO 2

DISEÑO Y CONSTRUCCION DEL HARDWARE

2.1 INTRODUCCIÓN

En este capítulo se estudia cada uno de los componentes del sistema de visión artificial, así como también se escoge y se justifica los componentes para la construcción del sistema.

2.2 SELECCIÓN DEL AMBIENTE CONTROLADO

Por definición, un ambiente es controlado cuando cumple con ciertas condiciones predeterminadas. Para el caso presente, se define como aquel que permita mejorar el desempeño del sistema de visión artificial, para controlar factores que podrían afectar el sistema, por ejemplo:

- Efectos de la iluminación (sombras, deslumbramiento, etc.);
- Distancia entre el objeto de interés y el lente de la cámara;
- Objetos externos que aparezcan en la imagen y confundan al sistema.

Un ambiente controlado para desarrollar un sistema de visión artificial, es el que permite extraer fácilmente las características de la imagen, logrando que el sistema sea más rápido y exacto. Con este objetivo se busca un ambiente tal que el sistema solo se ocupe en analizar los objetos de interés, dejando de lado aspectos como el mejoramiento de la imagen, el cual es un proceso largo y que requiere algoritmos complejos. Además, el hardware se haría más simple haciendo que el sistema resulte más barato y asequible.

A continuación se enumeran los ambientes controlados que se estudiaron para el desarrollo del proyecto.

2.2.1 AMBIENTE CONTROLADO DE ILUMINACIÓN

Este ambiente controlado es un espacio reducido con iluminación predominantemente controlada, con paredes recubiertas de un color que favorezca la iluminación. Un ambiente como éste se logra con una caja cuyo

interior se encuentra cubierto de papel blanco y el fondo donde se va a ubicar la mano es de color negro. Para iluminar este espacio se usa una luminaria detrás de la cámara.

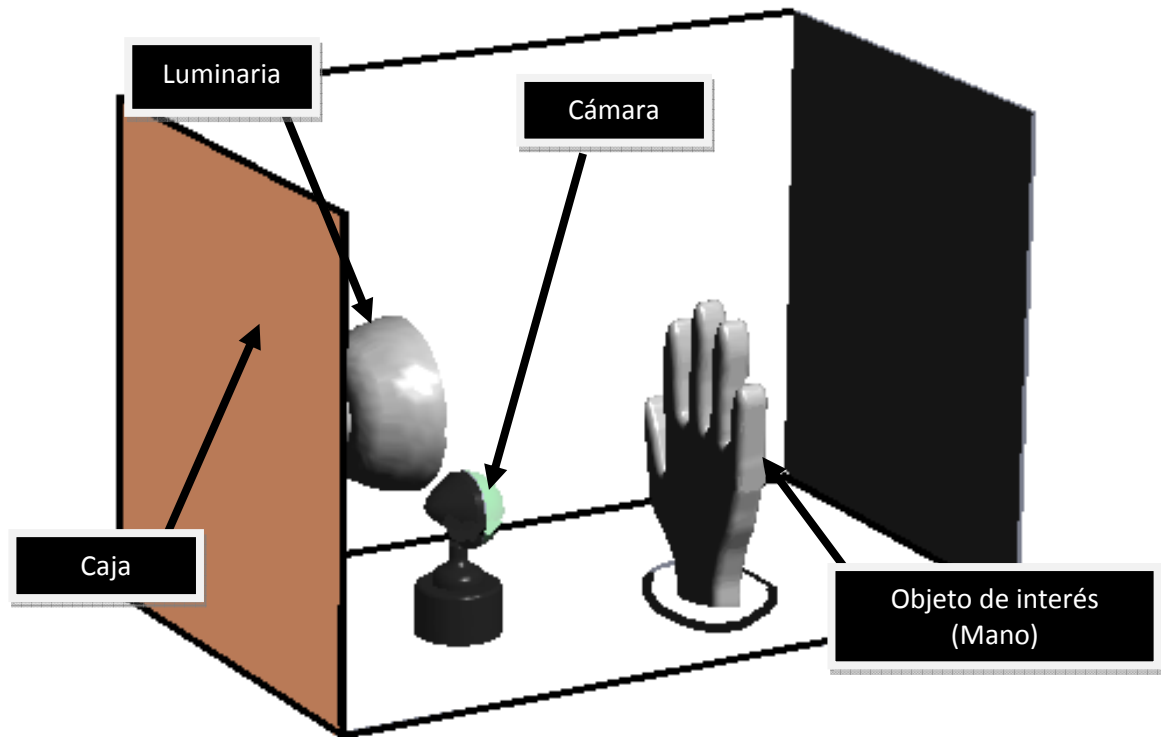


Figura 2.1 Ambiente controlado de iluminación.

Ventajas

- Al tener una iluminación controlada se evita la formación de sombras alrededor de la mano, lo que permite apreciar mayor cantidad de detalles de la imagen.
- Con el uso de un fondo negro se logra un mejoramiento significativo en el contraste y se evita que aparezcan objetos extraños en el fondo; es decir, se aísla la mano.

Desventajas

- Una de las principales desventajas es el espacio limitado del que se dispone para realizar los diferentes símbolos con la mano.
- Este sistema es poco práctico y requiere un mayor espacio físico para su colocación debido al tamaño de la caja.

2.2.2 AMBIENTE CONTROLADO POR CONTRASTE

En este ambiente controlado se recurre a una buena iluminación y un fondo de un color tal que resalte el objeto de interés, en este caso negro. En este sistema la iluminación debe provenir de dos lugares diferentes: la primera de una lámpara en el techo del cuarto y la segunda de una luminaria que está ubicada detrás de la cámara. Esta iluminación secundaria se usa para eliminar las sombras que genera la iluminación principal como se observa en la Figura 2.2.

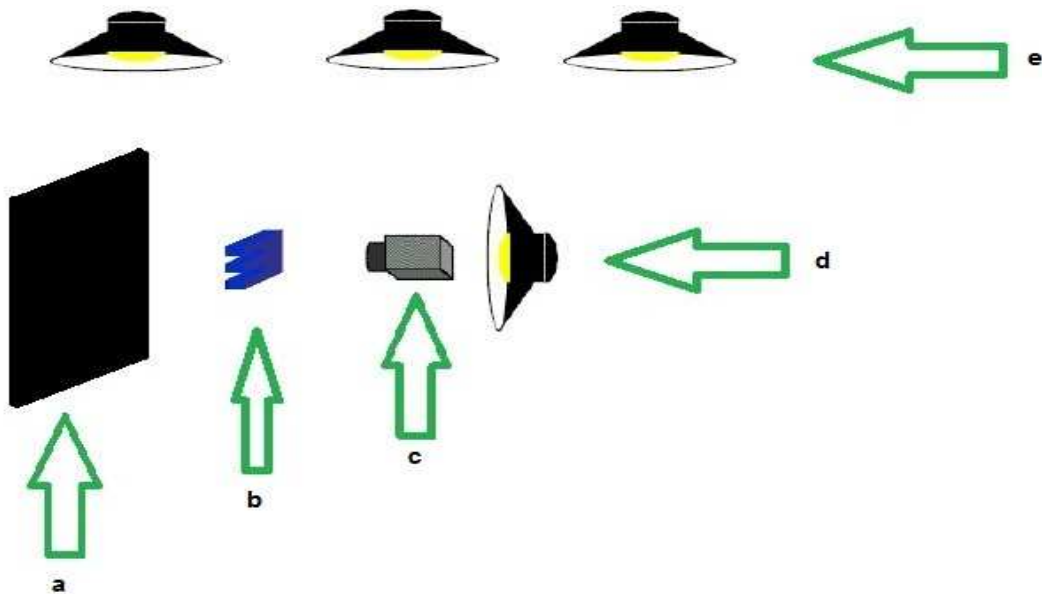


Figura 2.2 Ambiente controlado por contraste.

En la Figura 2.2 se observa los siguientes elementos:

- a. Fondo negro para contrastar el objeto de interés;
- b. Objeto de interés (mano);
- c. Cámara;
- d. Iluminación secundaria ubicada detrás de la cámara;
- e. Iluminación principal existente en el lugar de trabajo.

Ventajas

- Ofrece menos restricciones de uso en comparación al anterior.
- Hardware fácilmente transportable.
- Ofrece más libertad de movimiento al usuario.

Desventajas

- Requiere mayor procesamiento de la imagen, lo que hace lento al sistema.
- Requiere de un sistema de iluminación existente en lugar de trabajo.
- Costos de hardware más elevado que el anterior.

2.2.3 SELECCIÓN DE LA MEJOR OPCIÓN

De los sistemas antes mencionados se optó por el Ambiente Controlado por contraste debido a la flexibilidad que brinda, ya que se puede transportar y montar fácilmente.

Si bien es cierto requiere un mayor procesamiento de la imagen y tiene un mayor costo, pero por otro lado presenta mayor flexibilidad en cuanto al tipo de ambiente en el que puede trabajar, volviéndolo un sistema práctico.

2.2.4 CONSTRUCCIÓN DEL AMBIENTE CONTROLADO

La construcción de este ambiente se llevo a cabo en tres partes: el fondo, la iluminación y el montaje de la cámara.

2.2.4.1 El fondo

El fondo que se usa en el proyecto consiste de un pliego de fomix de color negro mate, el cual se adhiere a una superficie vertical.

Se opta por el fomix porque al ser de color mate no genera brillo en su superficie, lo que favorece al correcto funcionamiento del sistema.

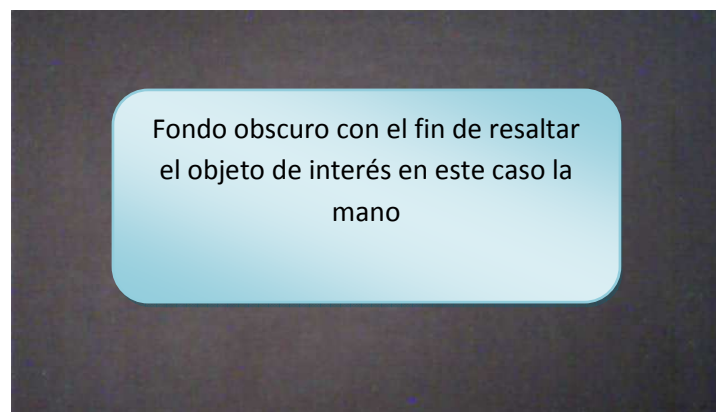


Figura 2.3 Fondo negro mate de fomix.

2.2.4.2 La iluminación

Esta iluminación consiste en una luminaria con una lámpara ubicada detrás de la cámara. Se analizaron tres tipos de fuentes de luz: lámpara incandescente, lámpara fluorescente y leds de alta luminosidad.

La lámpara incandescente genera la cantidad de luz necesaria sobre el área de trabajo, pero a su vez genera una gran cantidad de calor sobre la cámara.

Los leds no generan la cantidad de luz necesaria sobre el área de trabajo, además de que requieren una fuente de corriente continua para su funcionamiento.

La lámpara fluorescente genera la cantidad de luz necesaria sobre el área de trabajo, además de que genera poco calor sobre la cámara.

Como hoy en día es muy común encontrar en el mercado lámparas ahorradoras fluorescentes, se optó por elegir este tipo de lámpara para la luminaria.

La luminaria se ubica en la parte trasera a la cámara para evitar el deslumbramiento de la misma. La forma de la luminaria es tal que dirige la luz directamente sobre la mano.

La luminaria secundaria está formada por: una boquilla, un foco ahorrador que funciona a 110Vac y un difusor. La luminaria debe estar respaldada por una buena iluminación existente en el lugar donde se desee usar el sistema.



Figura 2.4 Iluminación secundaria.

2.2.4.3 La cámara

El primer factor a considerar en un sistema de visión artificial es la cámara que se va a utilizar, para esto se debe tener en cuenta los siguientes factores.

2.2.4.3.1 *Cámara a color o monocromática.*

Este factor depende mucho de la aplicación que se va a realizar, del presupuesto y del tiempo de procesamiento que requiere la aplicación. Una cámara a color es más cara que una monocromática además que requiere más algoritmos de procesamiento que una monocromática afectando la respuesta del sistema en tiempo real.

2.2.4.3.2 *Cámara digital o analógica.*

Las cámaras digitales son más rápidas que las analógicas, porque transmiten el dato directamente a la tarjeta de adquisición. Además las digitales pueden implementar bits de corrección de errores. En los dos tipos de cámaras se debe tener en cuenta si el cable de conexión es apantallado para evitar ruido que dañe la calidad de la imagen.

2.2.4.3.3 *Tipo de salida de video.*

Existen tres formatos: analógicos, digital monocromático y digital RGB. Las cámaras digitales RGB poseen los colores fundamentales por separado, lo que permite realizar el análisis de cada color.

2.2.4.3.4 *Formato del video.*

Para seleccionar esta característica de la cámara, se verifica el formato con el que trabaja la tarjeta de video. Existen dos formatos: el europeo PAL (50Hz) y el americano NTCS (60HZ).

2.2.4.3.5 *Resolución.*

Una cámara con mayor resolución permite distinguir los detalles más pequeños de la imagen, pero esto implica un mayor tiempo de procesamiento de la imagen como también mayor costo.

2.2.4.3.6 *Características del sensor.*

La tecnología que se usa para fabricar los sensores de cámaras digitales son: los CCD (*Charge Couple Device*) y los CMOS (*Complementary Metal Oxide Semiconductor*). Ambos tipos de sensores están formados por semiconductores de metal-óxido (*MOS*) y están distribuidos en forma de matriz.

La función de estos sensores es de acumular una carga eléctrica en cada una de las celdas de la matriz (píxeles). Esta carga eléctrica depende de la cantidad de luz que incida sobre el mismo, cuanto más luz incida sobre el píxel mayor será la carga que éste adquiera, es decir, la carga que adquiere el píxel es directamente proporcional a la cantidad de luz.

Aunque las dos clases de sensores los CCD y CMOS funcionan de manera similar, existen diferencias entre las dos tecnologías.

Sensor CCD

El sensor CCD convierte las cargas eléctricas de las celdas de la matriz en voltajes y entrega una señal analógica en la salida. En esta clase de sensor se hace una lectura de cada uno de los valores correspondiente a cada una de las celdas. Entonces, es esta información la que un convertidor analógico-digital traduce en forma de datos. La estructura interna del sensor es muy simple, pero se tiene como inconveniente la necesidad de un chip adicional que se encargue del tratamiento de la información proporcionada por el sensor, lo que provoca que el equipo sea de gran tamaño y más costoso.

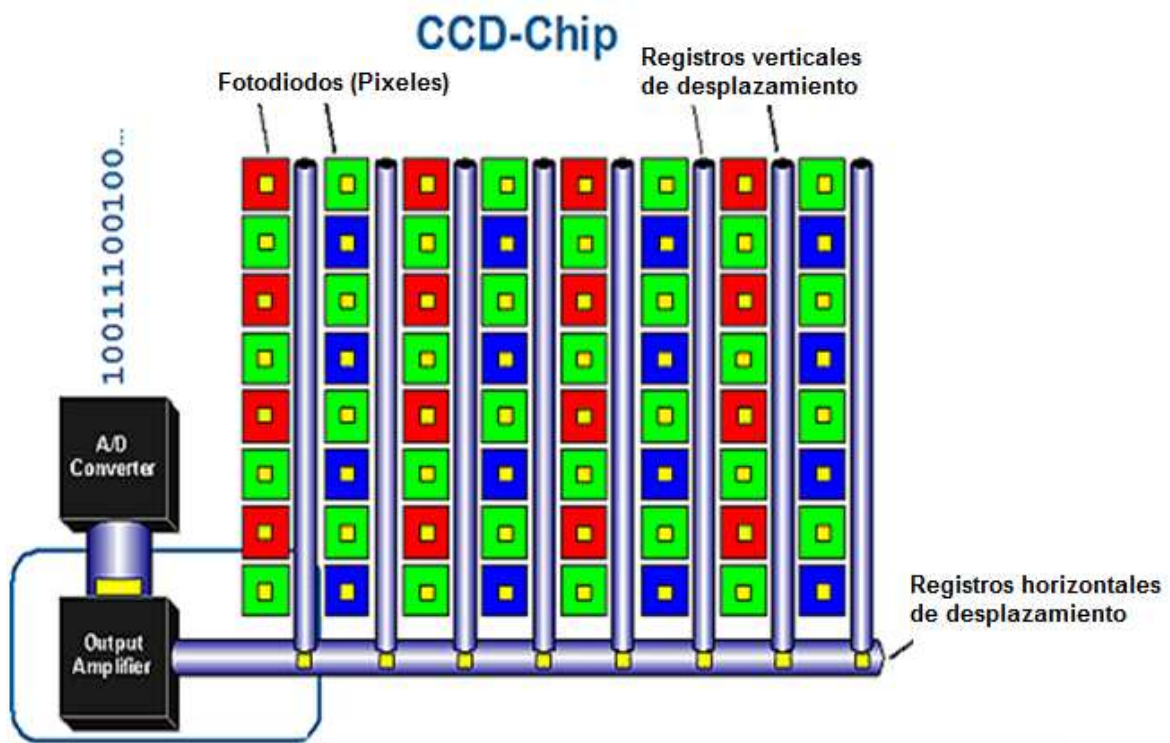


Figura 2.5 Estructura interna del sensor CCD (Tomado de [17]).

El sensor CCD con respecto al sensor CMOS presenta las siguientes ventajas:

- Tolera mejor los extremos de la luz;
- Mejor control del ruido;
- Respuesta uniforme de los píxeles sometidos a un mismo nivel de excitación.

Sensor CMOS

En este sensor cada celda (pixel) es independiente. La principal diferencia es que aquí la digitalización de los píxeles se realiza internamente en unos transistores que lleva cada celda, por lo que todo el trabajo se lleva a cabo dentro del sensor y no se hace necesario un chip externo que realice esta función, con esto se consigue equipos más baratos y más pequeños.

La velocidad del CMOS es mayor a la del CCD debido a que todo el proceso se realiza dentro del propio sensor. Esta es una de las principales razones por las cuales estos sensores se utilizan en cámaras de alta velocidad de hasta 1000 fotogramas por segundo.

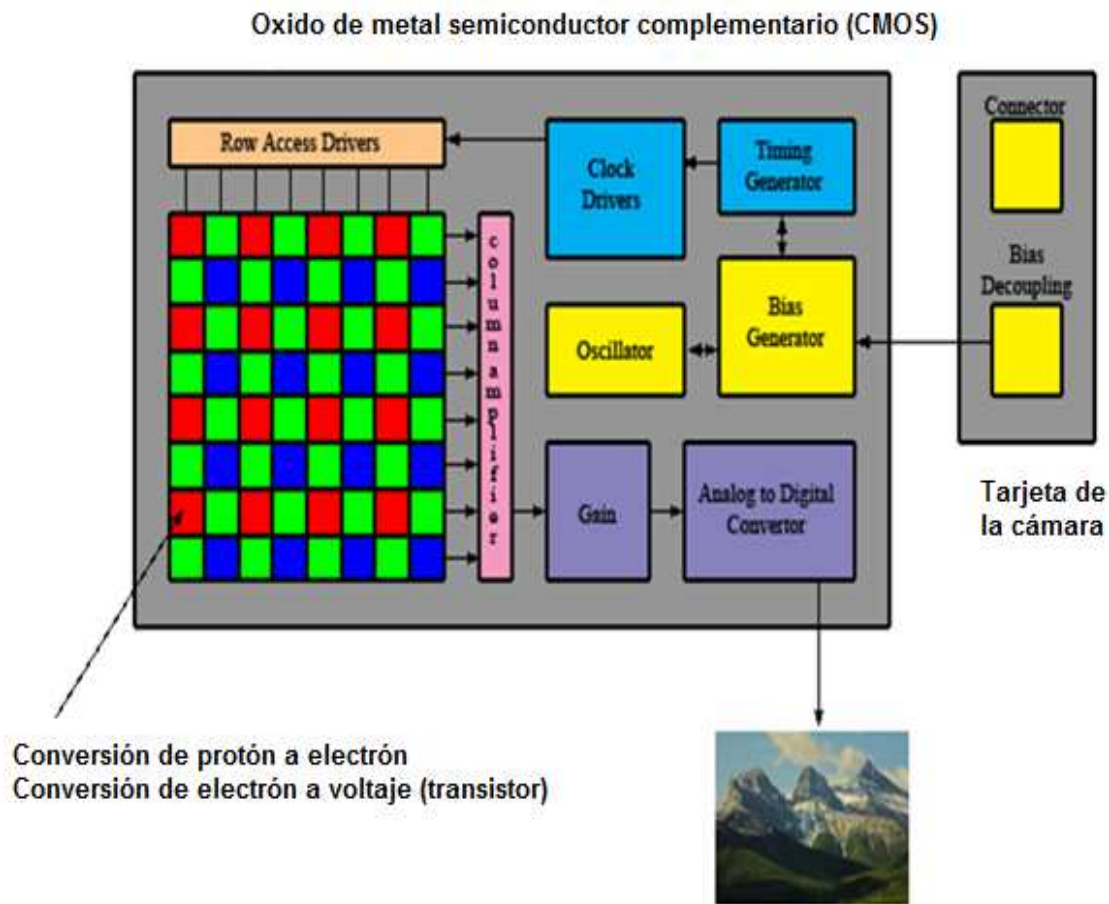


Figura 2.6 Estructura interna del sensor CMOS (Tomado de [17]).

El sensor CMOS con respecto al sensor CCD presenta las siguientes ventajas:

- Son más sensibles a la luz, por lo que en condiciones pobres de iluminación se comportan mucho mejor.
- Control del fenómeno “blooming”. Este fenómeno se produce cuando un píxel se satura por la luz que incide sobre él y a continuación empieza a saturar a los que están a su alrededor

2.2.4.3.7 *Velocidad del obturador.*

Este factor es importante para aplicaciones con objetos en movimiento, esto depende de que tan rápida sea la aplicación.

Analizando cada uno de los factores anteriores se concluye que la aplicación define las características de la cámara, pero se debe tener en cuenta el factor económico porque mejores características en una cámara implica mayor costo [7].

2.2.4.4 Selección de la cámara para la aplicación

Como el software que se utiliza es LabVIEW 2009 y éste tiene una librería completa para leer cámaras USB, se escogió este tipo de conexión para la cámara digital. Además de una librería para leer imágenes en el plano RGB.

Ya que se busca una aplicación de bajo costo, se escoge una cámara de 30 fotogramas por segundo (fps) que son las más comunes en el mercado. Para escoger el tamaño de la imagen se tomó en cuenta el área de trabajo del ambiente controlado y el tiempo de procesamiento.

Tomando en cuenta estos parámetros se optó por la cámara web Genius Messenger 310, que ofrece las siguientes características:



Figura 2.7 Cámara Messenger 310 (Tomada de [21]).

- Cámara digital USB.
- Trabaja en el plano RGB.
- Tamaño de imagen 480*640 (requerido).
- Toma 30 fotogramas por segundo.
- Sensor CMOS

Se debe tener en cuenta que esta cámara tiene autocontraste lo que afecta al programa de procesamiento de la imagen; por esta razón, esta función se debe cancelar. Esto se explicara con más detalle en el Capítulo 3.

2.2.4.5 Montaje de la cámara

La cámara se encuentra colocada sobre un trípode de altura y giro regulable, para poder ajustarla a la altura del usuario y realizar el encuadre respectivo.

El objetivo de montar la cámara en un trípode es por la facilidad de adaptación a la altura del usuario, además que permite girar la cámara facilitando el encuadre de la mano.

La cámara se encuentra montada delante de la fuente de iluminación secundaria, para evitar el deslumbramiento de la misma.



Figura 2.8 Montaje de cámara

Características del trípode:

- Altura trípode extendido 57".
- Peso 2.2 Lbs.
- Altura trípode plegado 21.45"
- Diámetro de las patas 20mm.
- Posee un medidor de nivel.
- Material de construcción aluminio.

2.2.4.6 Hardware implementado



Figura 2.9 Montaje de la cámara en trípode.



Figura 2.10 Partes principales del hardware.



Figura 2.11 Iluminación secundaria y computadora.



Figura 2.12 Partes transportables del hardware.

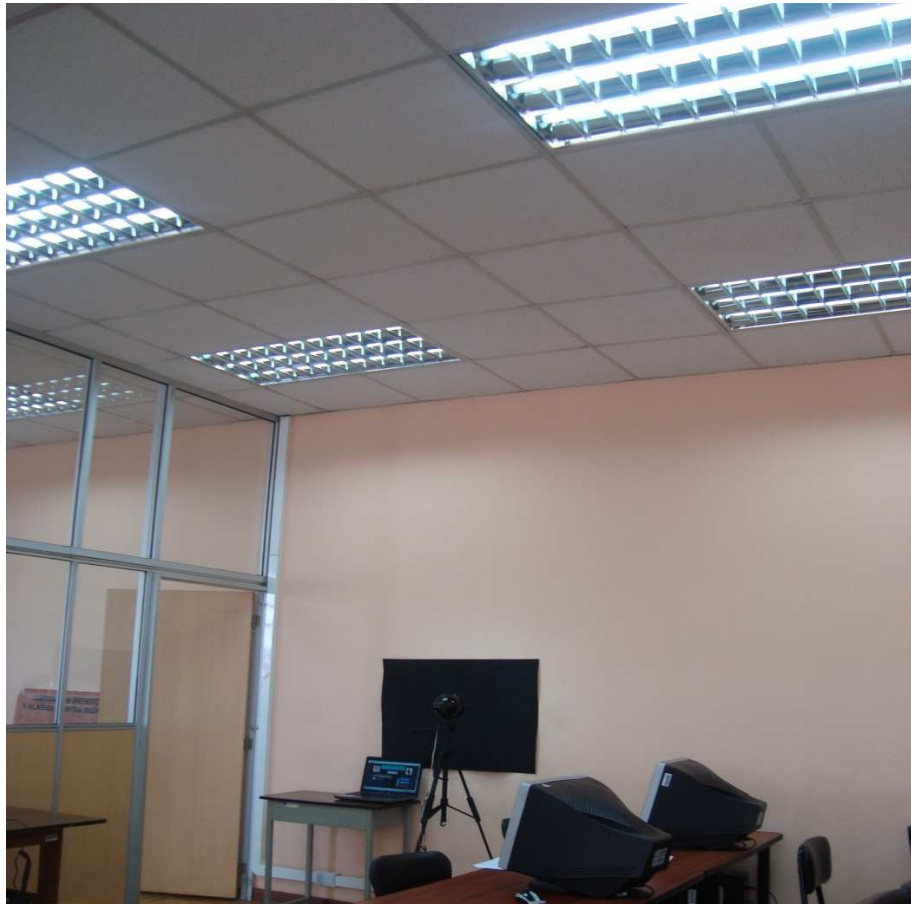


Figura 2.13 Sistema completo.



Figura 2.14 Posición de la mano en el hardware

En este capítulo se llega a la conclusión de que la iluminación es muy importante en el sistema, porque el desempeño del sistema traductor depende mucho de ésta. El hardware está diseñado de la forma más simple posible, pero con la idea de que posea un gran rendimiento, cumpliendo las características del sistema de visión artificial. Éste solo necesita una toma de corriente alterna a 110vac para la luminaria y una computadora con un puerto USB donde debe estar cargada con la aplicación.

CAPÍTULO 3

DESARROLLO DEL SOFTWARE EN LABVIEW

3.1 INTRODUCCIÓN

Existen una gran cantidad de software que permite realizar aplicaciones de visión artificial como por ejemplo: Matlab, Labview, Opencv, VisualBasic, etc. Para la realización de este proyecto se utilizó el software de Labview con los toolkits de visión artificial, porque permite realizar aplicaciones de forma simple gracias a su programación gráfica amigable con el usuario. Además se adapta fácilmente al hardware (cámaras USB), que se está utilizando y posee una aplicación dedicada al reconocimiento de imágenes, que es el objetivo de este proyecto.

3.2 DESCRIPCIÓN DEL PROGRAMA

El programa en Labview se desarrolló en dos módulos, un traductor y un entrenador. Cada módulo está programado en forma secuencial, dividido en etapas a fin de lograr los mejores resultados y cumplir con los objetivos del proyecto.

3.2.1 PANTALLA PRINCIPAL

Como se observa en la Figura 3.1 la pantalla principal contiene la presentación del proyecto con su información básica como: el nombre de la Escuela, Carrera, Título del proyecto, Autores y Manual de Usuario. También presenta las opciones para acceder a cualquiera de los dos módulos y la opción *Stop* para detener el programa.



Figura 3.1 Pantalla principal del programa.

3.2.2 TRADUCTOR

En el traductor se encierra el objetivo del proyecto, ya que éste es el encargado de tomar la señal que forma la mano y traducirla a su correspondiente letra, e ir formando el texto que se desea comunicar.

Este módulo consta de las siguientes etapas: adquisición, detección, cambio a escala de grises, mejoramiento del contraste, segmentación, recortado del objeto de interés, clasificación y formación del texto.

3.2.2.1 Adquisición

En esta etapa se configura los parámetros de la cámara con la que se está trabajando, estos son: tamaño de la imagen, brillo, contraste, colores y el foco.

Es importante eliminar la función de autocontraste que posee la cámara, con esto se elimina el ruido que afecta a la etapa de Detección y se reduce el tiempo de ejecución de la aplicación. Para eliminar esta función se debe usar el módulo "Measurement & Automation", en el que se visualiza las cámaras conectadas al computador (Figura 3.2), dando la posibilidad de variar los parámetros de adquisición de cada cámara.

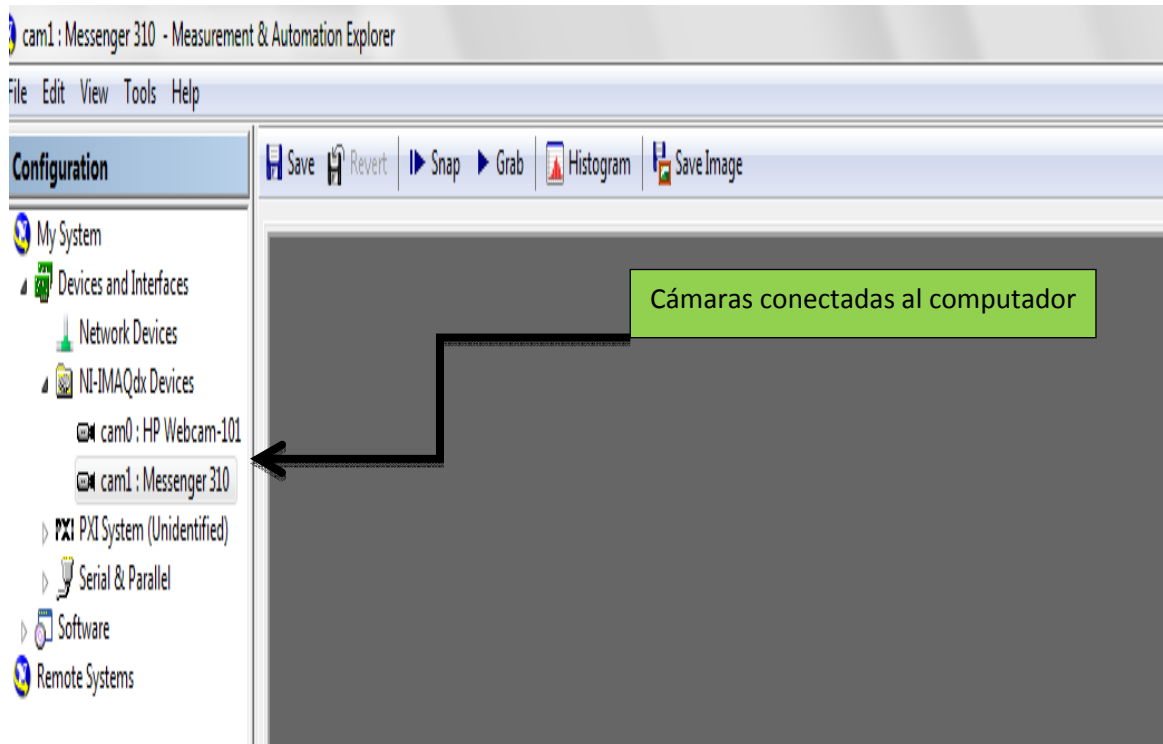


Figura 3.2 Cámaras conectadas al computador.

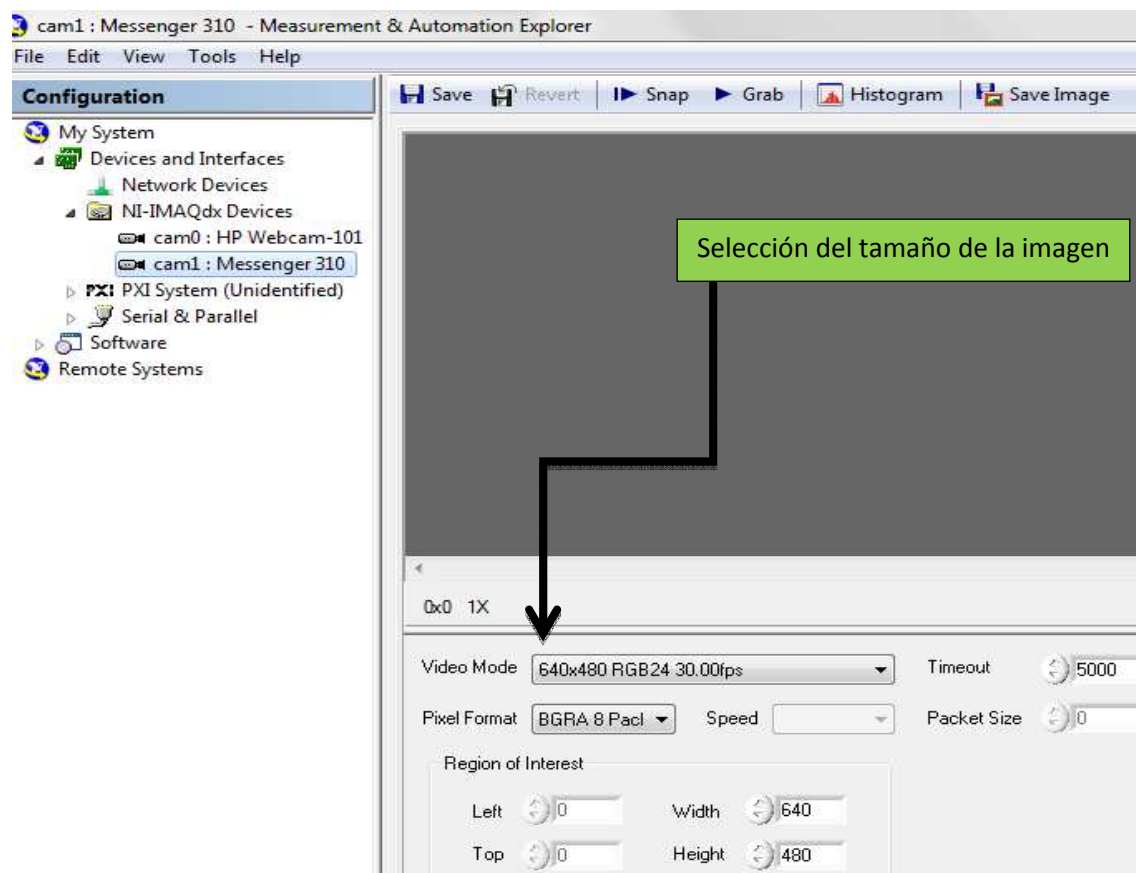


Figura 3.3 Selección del tamaño de la imagen.

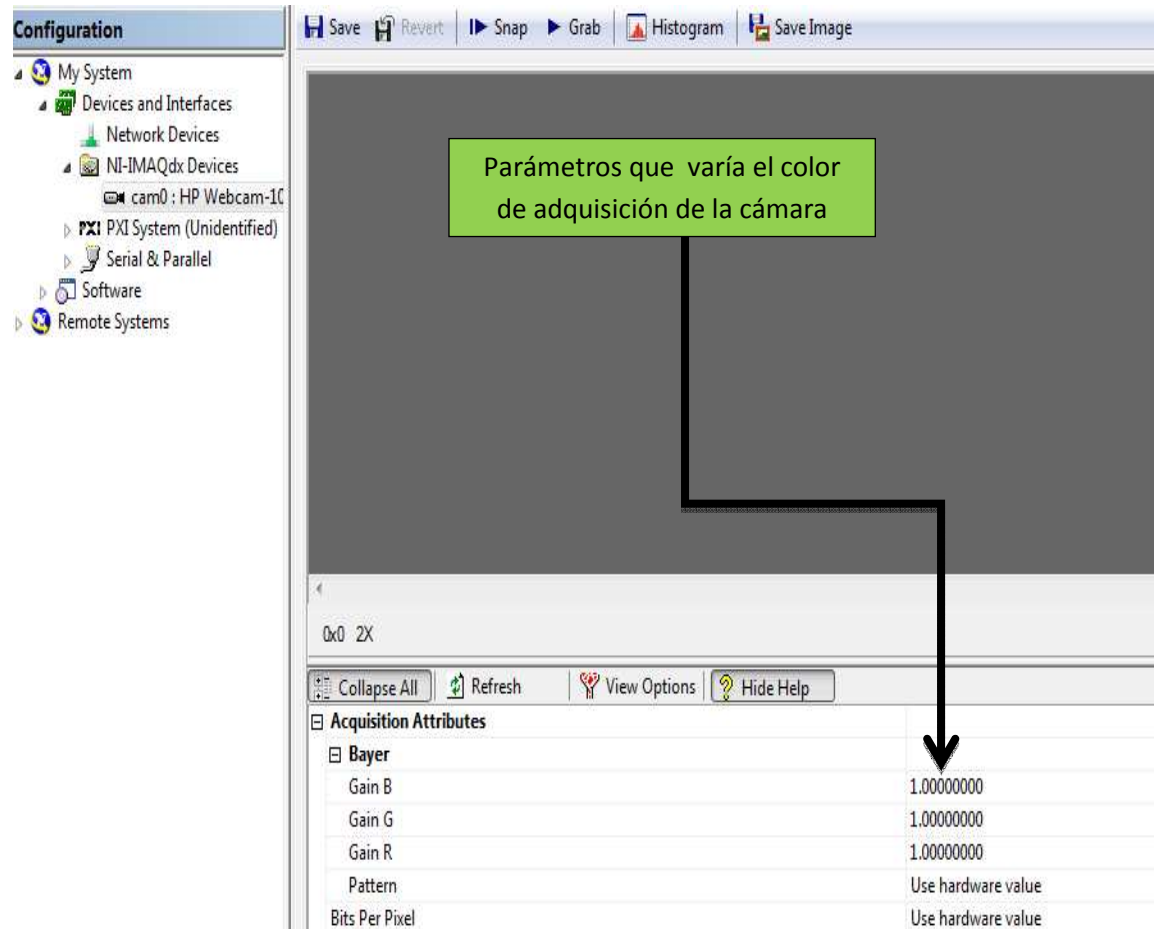


Figura 3.4 Valores de ganancia de los colores RGB.

En la Figura 3.3 se observa el tamaño de la imagen que la cámara adquiere.

Ajustando a los valores de la Figura 3.4, la imagen que la cámara adquiere es al parecer monocromática, pero en realidad está adquiriendo una imagen RGB32 bits con ganancias de colores bajos.

Para evitar el autocontraste se ajusta los parámetros “Gamma” y el “Contraste”, a los valores que se muestran en la Figura 3.5.

El factor brillo no se altera porque de éste depende la adquisición de la imagen, al variar esta imagen lo que se afecta es la claridad de la imagen.

Otro factor que se debe ajustar es el balance de sombras automático de la cámara. Los valores a los que se ajustó se presentan en la Figura 3.6.

Collapse All		Refresh	View Options	Hide Help
Acquisition Attributes				
Bayer				
Gain B				1.00000000
Gain G				1.00000000
Gain R				1.00000000
Pattern				Use hardware value
Bits Per Pixel				Use hardware value
Height				480
Ignore First Frame				<input type="checkbox"/>
Overwrite Mode				Get Newest
Payload Size				921600
Pixel Format				BGRA 8 Packed
Pixel Signedness				Signed
Receive Timestamp Mode				None
Shift Pixel Bits				<input type="checkbox"/>
Swap Pixel Bytes				<input type="checkbox"/>
Timeout				5000
Vertical Mirror				<input type="checkbox"/>
Video Mode				640x480 RGB24 30.00fps
Camera Attributes				
Backlight Compensation				
Mode				Manual
Value				<input checked="" type="checkbox"/>
Brightness				
Mode				Manual
Value				107
Color Enable				
Mode				Manual
Value				<input type="checkbox"/>
Contrast				
Mode				Manual
Value				0.00000000
Exposure				
Mode				Auto
Value				1.00000000
Gamma				
Mode				Manual
Value				0.00000000
Hue				
Mode				Manual
Value				0.00000000

Figura 3.5 Valores de “Contrast” y “Gamma” para evitar autocontraste.

Collapse All		Refresh	View Options	Hide Help
[-] Saturation	Mode	Manual		
	Value	0		
[-] Sharpness	Mode	Manual		
	Value	0		
[-] White Balance	Mode	Auto		
	Value	0		
[-] Zoom	Mode	Manual		
	Value	100		
[-] Camera Information	Base Address	0x0		
	Bus Type	DirectShow		
	Model Name			
	Serial Number High	0x1B266817		
	Serial Number Low	0x42545319		
	Vendor Name	Messenger 310		
[-] Status Information	Acq In Progress	<input type="checkbox"/>		
	Last Buffer Count	0		
	Last Buffer Number	4294967292		
	Lost Buffer Count	0		

Figura 3.6 Valores establecidos para evitar ajustes automáticos de sombras.

3.2.2.2 Detección de la mano en la imagen

Con la finalidad que el sistema trabaje únicamente cuando esté presente un objeto (mano) dentro del área de trabajo, se realiza un análisis estadístico de los píxeles dentro del ambiente controlado. Esta operación se realiza constantemente hasta que el usuario comienza a usar la aplicación.

Este análisis se hace con la función "lineprofile.vi". Esta función proporciona el valor de los píxeles dentro del área de trabajo, y con estos valores se realiza el cálculo de la media y se proporciona una gráfica donde se muestra la distribución de estos píxeles.

Este análisis se realiza en la imagen en color en el espacio RGB32 bits, por esta razón se tiene un conjunto de valores para cada color. Una vez establecidos los valores de los píxeles del área de trabajo se realiza la supervisión de los mismos, con el objetivo de verificar las variaciones y arrancar el siguiente paso del programa.

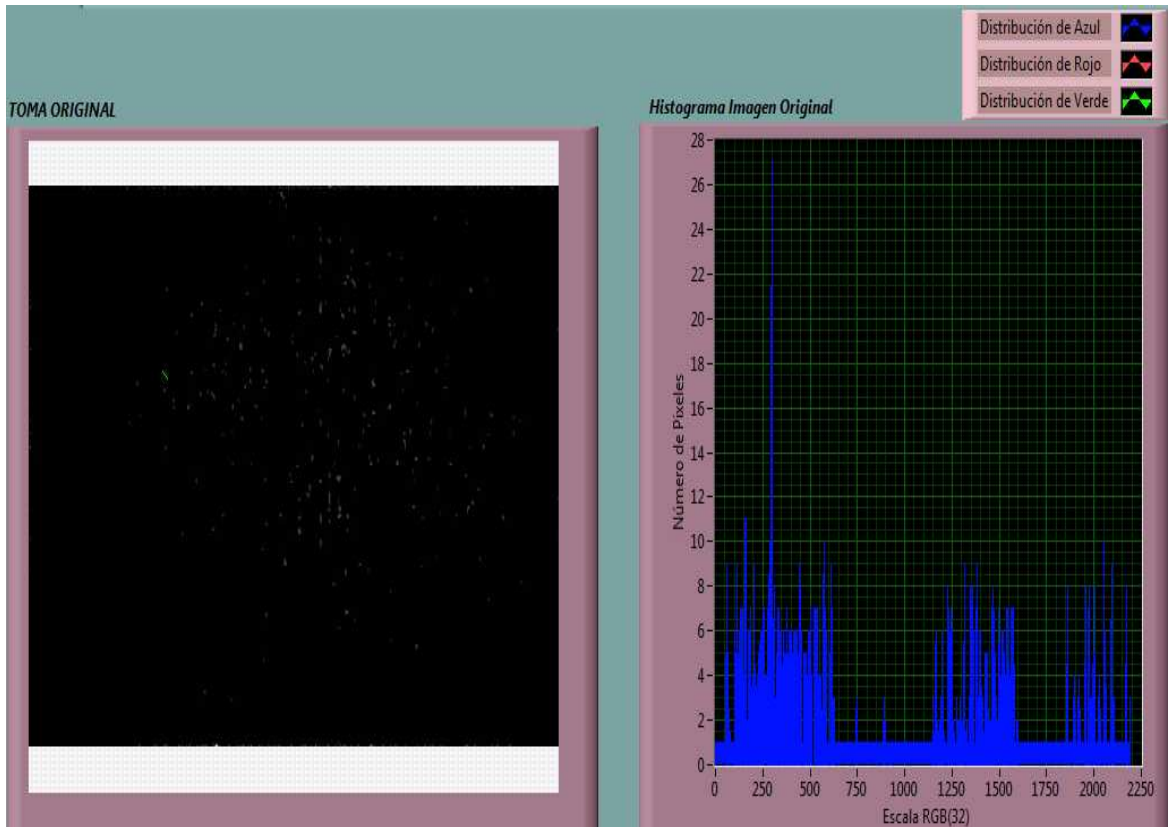


Figura 3.7 Imagen e histograma del fondo (Sin mano).

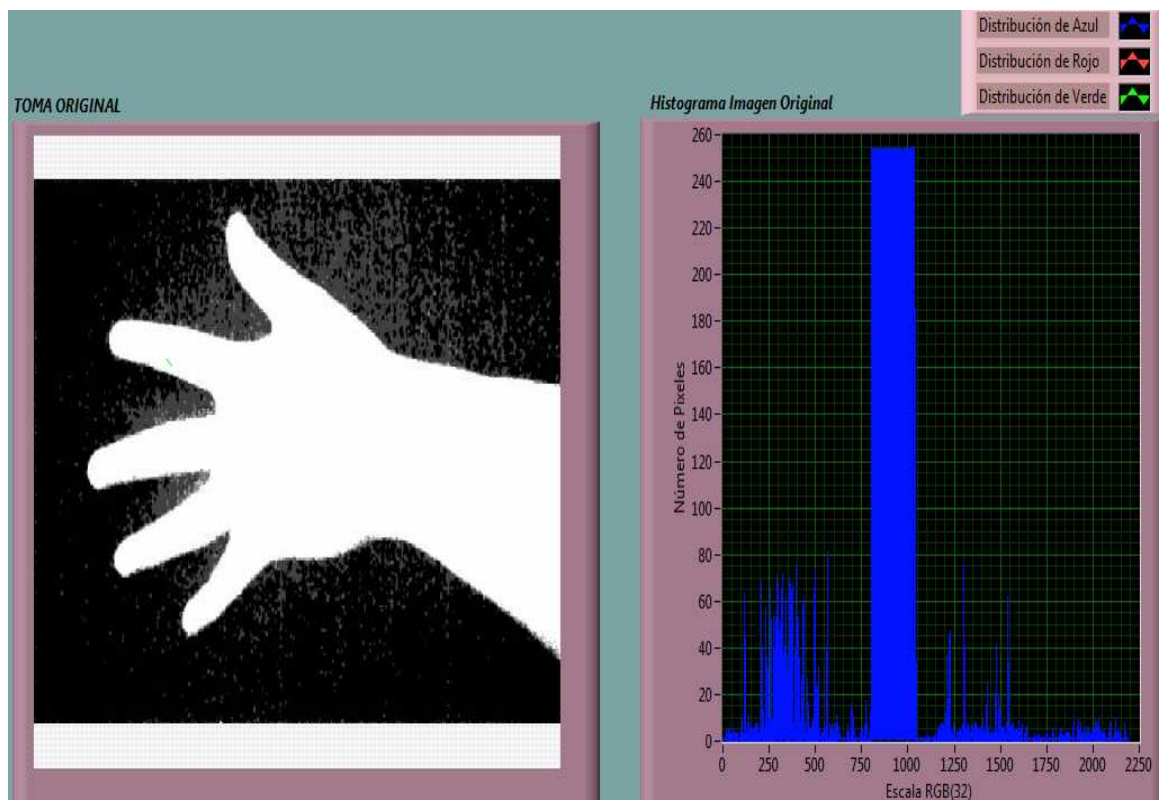


Figura 3.8 Imagen e histograma con presencia de mano.

3.2.2.3 Cambio a escala de grises y mejoramiento del contraste y brillo

Como la aplicación no requiere de un análisis del color de la imagen, se hace una conversión de la imagen a escala de grises de ocho bits con la función “*ExtractSingleColorPlane.VI*”; es decir, los píxeles de la imagen pasan a tener un valor de entre 0 y 255. Después de esto se realiza una calibración del contraste y brillo de la imagen para eliminar el ruido presente en la imagen, como las sombras, y resaltar las características del objeto de interés, en este caso la señal realizada con la mano. La calibración del contraste se realiza teniendo en cuenta la cantidad de luz presente en el lugar de trabajo.

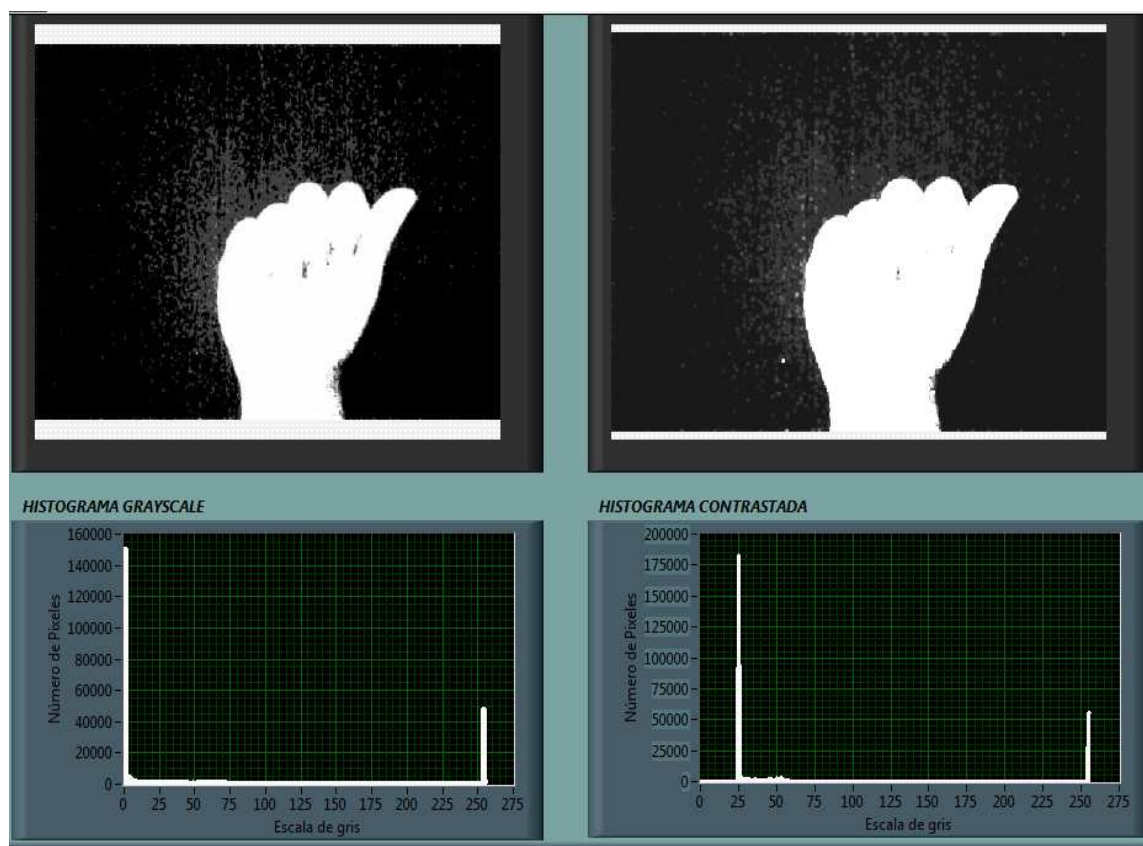


Figura 3.9 Imagen en escala de grises y calibración del contraste.

3.2.2.4 Segmentación

Como la segmentación de la imagen se basa en la morfología de la señal de la mano, se realiza un proceso de binarización en el cual se resalta más la forma del objeto de interés. Para realizar la binarización se toma en cuenta que existe variación de luz en la imagen, por esta razón se usó la binarización automática con la función “*IMAQ AutoBtherhold2*”, que se adapta fácilmente a los cambios de

iluminación. Para purificar la binarización se realiza operaciones morfológicas como por ejemplo: la dilatación para eliminar el ruido de la imagen, filtros basados en morfología para eliminar pequeñas partículas y finalmente se realiza una ecualización de la imagen para que ésta pueda ser usada en la siguiente etapa. No se realiza la operación “close” para llenar agujeros porque algunas señales de manos dependen de los agujeros que forman las señales.



Figura 3.10 Imagen resultante del proceso de binarización.

3.2.2.5 Recortado del objeto de interés

Una vez realizado el proceso de segmentación de la imagen, se lleva a cabo el seguimiento y recortado de la mano dentro del espacio de trabajo. Con estas operaciones se mejora el proceso de clasificación porque se identifica el objeto de interés en cualquier lugar del espacio de trabajo y se reduce el tamaño de la imagen a clasificarse.

Para recortar la imagen se usa la detección de bordes con el propósito de encontrar los flancos de subida y bajada en la orientación derecha e izquierda, y también en orientación superior e inferior. De esta forma se encuentra los puntos máximos y mínimos en los dos sentidos de orientación que sirve para localizar el objeto de interés dentro de la imagen.

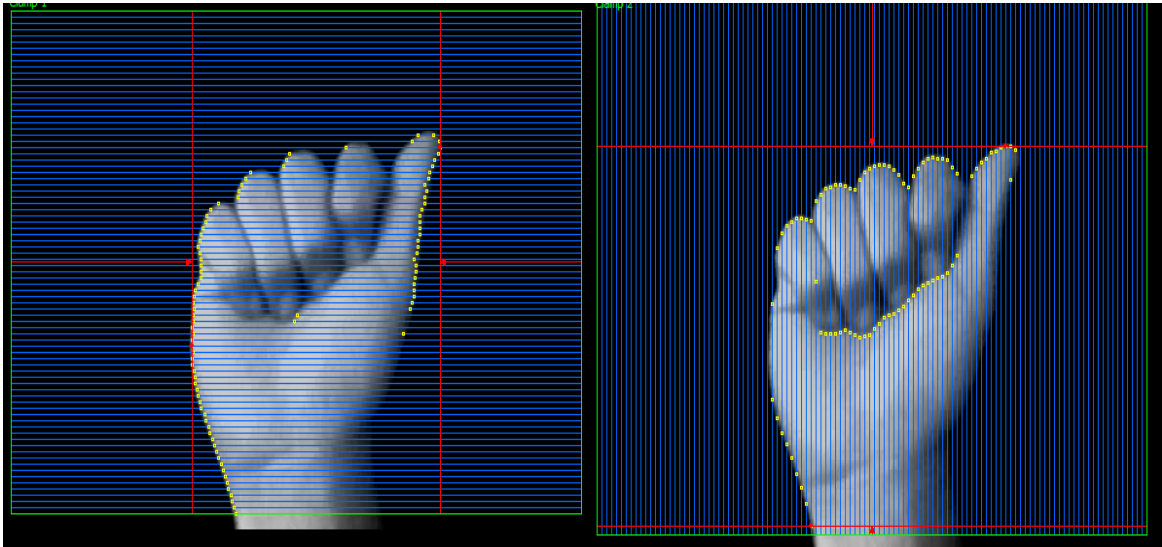


Figura 3.11 Detección de bordes en los dos sentidos.

3.2.2.5.1 Construcción del ROI

Una vez que se detectan los bordes del objeto de interés, en este caso la mano, se obtiene los puntos máximos en cada lado del objeto de interés como se muestra en la Figura 3.12.

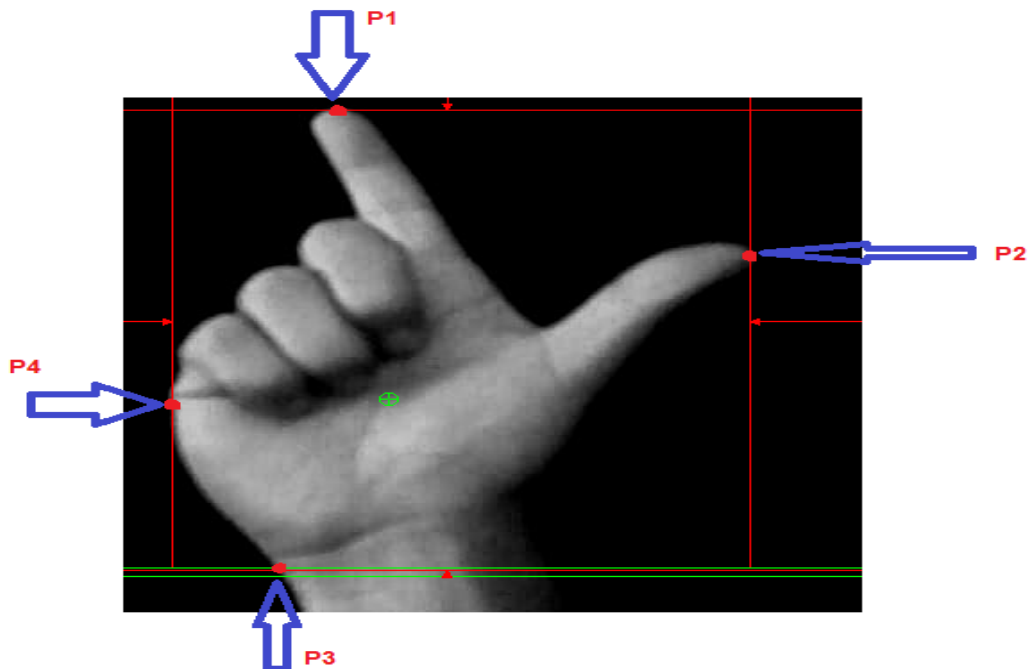


Figura 3.12 Puntos extremos del objeto de interés.

Cada uno de los puntos de la imagen P1, P2, P3 y P4 tienen componentes tanto en el eje "x" como en el eje "y". Con estas componentes se calculan las

coordenadas de los puntos P11, P22, P33 y P44 que son los límites de la región de interés.

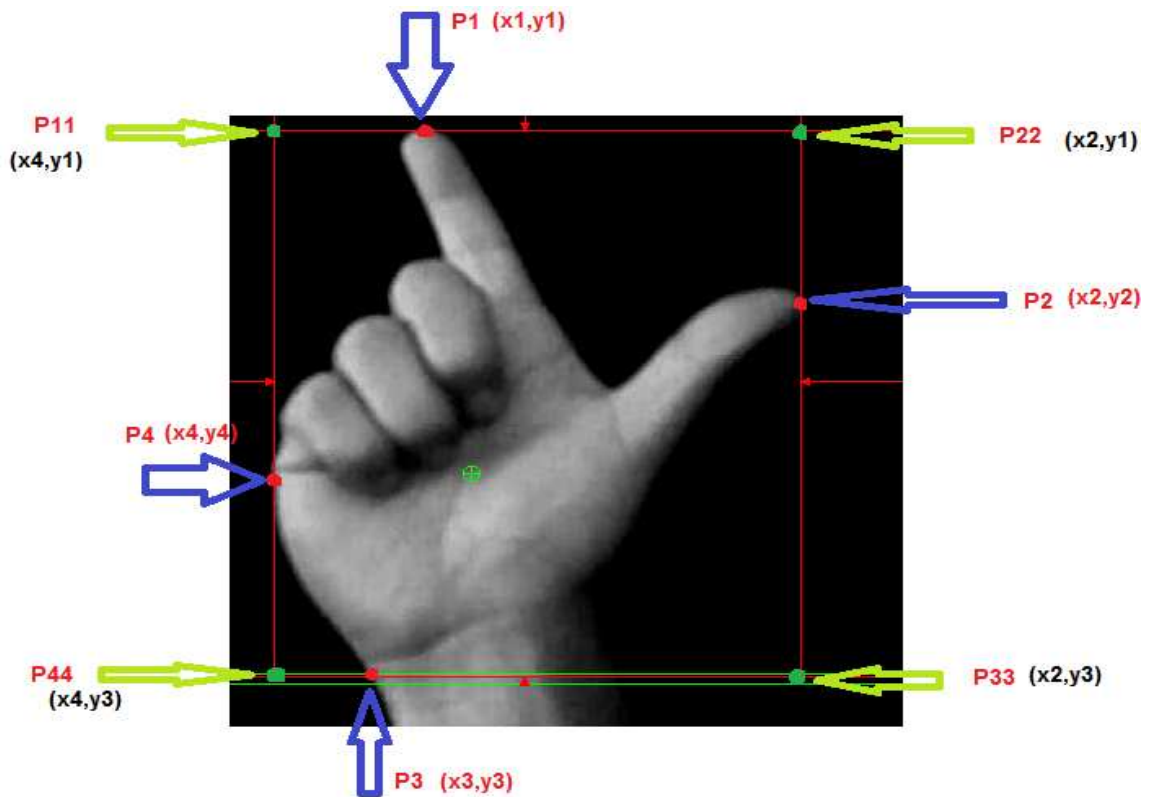


Figura 3.13 Puntos límites del ROI con sus coordenadas.

Para el cálculo de las coordenadas de los puntos de la región de interés se debe hacer lo siguiente: Para obtener las coordenadas del punto P11 se debe tener en cuenta que este tiene la misma componente en el eje "y" que el punto P1, y la misma componente en el eje de las "x" que el punto P4. Tomando en cuenta estas igualdades se obtiene las coordenadas de los puntos restantes.

P1=(x1,y1)	Puntos de interés	P11=(x4,y1)
P2=(x2,y2)		P22=(x2,y1)
P3=(x3,y3)		P33=(x2,y3)
P4=(x4,y4)		P44=(x4,y3)

Tabla 3.1 Coordenadas de los puntos del objeto de interés.

Una vez que se calculan estos puntos, se procede a cortar el objeto de interés del resto de la imagen, enviando solo el objeto de interés al proceso de clasificación.

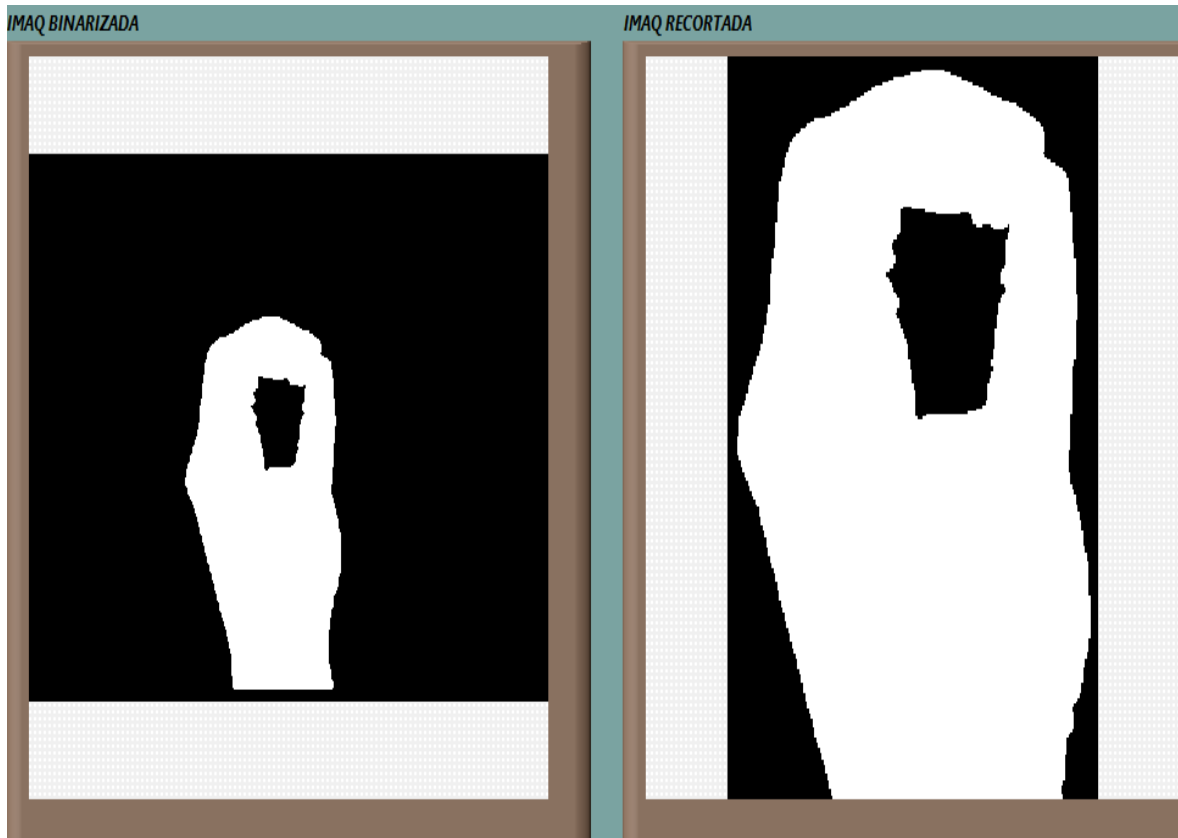


Figura 3.14 Imagen binarizada y recortada.

3.2.2.6 Generación de Bases de Datos

Para realizar el proceso de clasificación primero se debe crear las bases de datos, que en otras palabras son los vectores de características de cada clase. Este proceso se explica a continuación.

3.2.2.6.1 Introducción al módulo National Instruments Vision Assistant

Esta herramienta del Labview permite hacer prototipos y pruebas para aplicaciones de procesamiento de imágenes. Permite ir desarrollando paso a paso un algoritmo de procesamiento para después probarlo con varias imágenes y de esa manera comprobar su correcto funcionamiento.

En este proyecto se toman seis muestras por cada letra del alfabeto, estas muestras corresponden a personas con diferente color de piel y tamaño de la mano. Usando el “*Vision Asistant*” y las muestras almacenadas se genera las bases de datos que contienen los vectores característicos de cada clase. Para este proyecto cada letra del alfabeto corresponde a una clase.

Estos vectores de clase son los que servirán para el reconocimiento y clasificación de las señales de manos. A continuación se hace una introducción al uso del módulo “*Vision Assistant*”.

En la pantalla principal del “*Vision Assistant*” se diferencian las siguientes ventanas:

- Ventana de referencia (1).
- Ventana de Funciones para Procesamiento (2).
- Ventana Script (3).
- Ventana de Procesamiento (4).

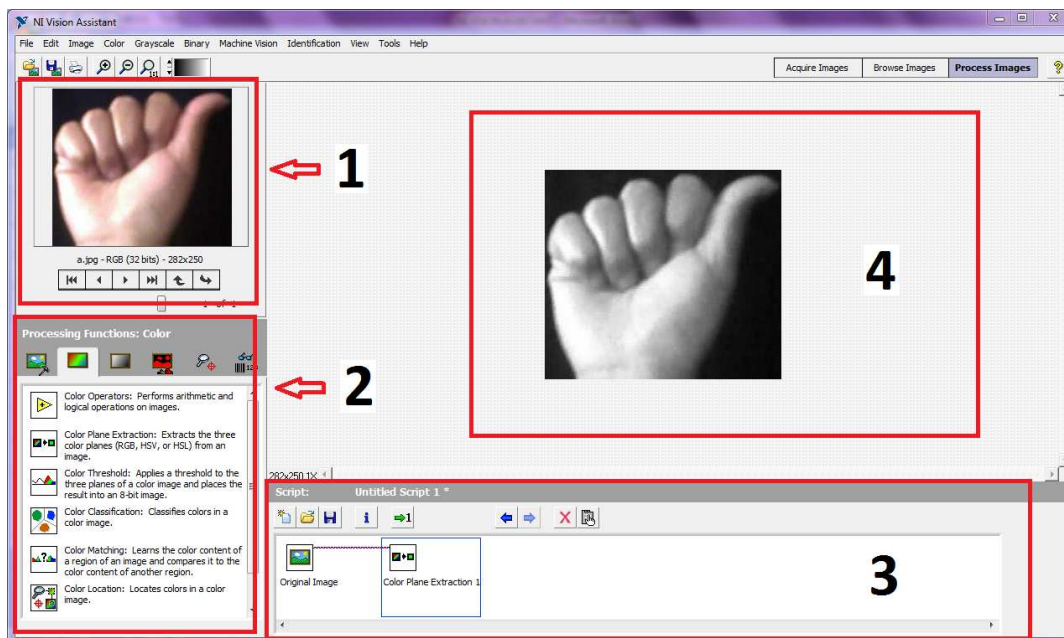


Figura 3.15 Ventana Principal del “*Vision Assistant*” (Tomado de [18]).

✓ *Ventana de Referencia*

En esta ventana se visualiza la imagen previamente adquirida mediante la cámara y que va a ser procesada, en este caso es una imagen en color RGB de 32 bits.

✓ *Ventana de Funciones para Procesamiento*

En esta ventana existen seis paletas de funciones que pueden ser usadas dentro del procesamiento de imágenes. Estas paletas son: Imagen, Color, Escala de Grises, Binarización, Machine Vision e Identificación.

- *Imagen*

Aquí se encuentran bloques para determinar las propiedades de una imagen, entre las más importantes se tiene: el histograma, brillo que permite variar el contraste y perfil de línea que permite saber la distribución de la escala de grises a lo largo de una línea de píxeles de una imagen.

Para efectos de nuestro estudio se utiliza la imagen original sin analizar ninguna de sus propiedades.

- *Color*

Los bloques de esta paleta permiten realizar operaciones entre imágenes, conversiones entre espacios de color, etc. Entre las más importantes se tiene: operaciones en color (suma, resta, multiplicación y división), extracción de planos color, clasificación en color y localización de colores en una imagen en color.

Como en este proyecto se necesita realizar la conversión de una imagen de color RBG de 32 bits a escala de grises se utiliza la función (*Color Plane Extraction*) y se selecciona la opción “*Intensity*”.

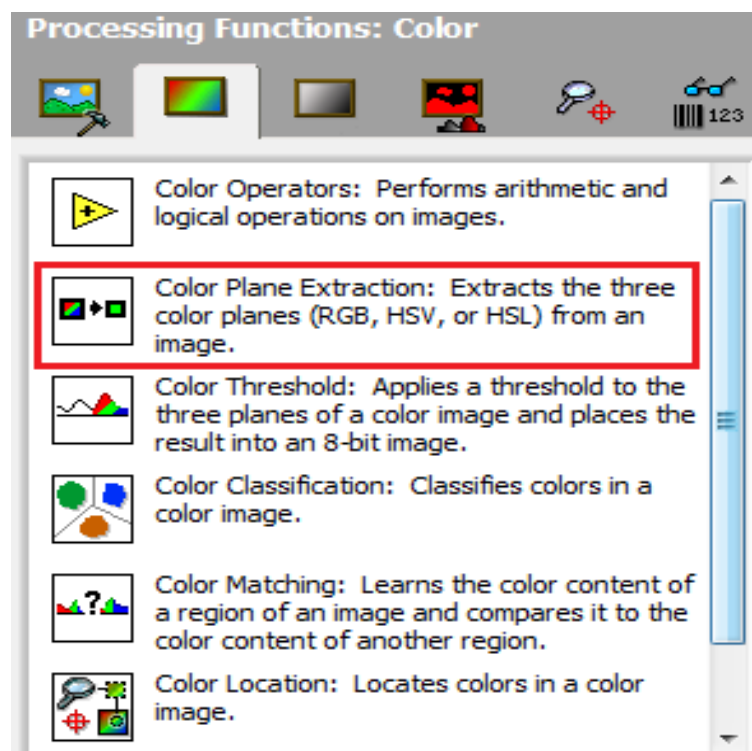


Figura 3.16 Menú de Color.

- *Escala de Grises*

En esta paleta se tienen opciones para: mejorar el contraste y el brillo con la función Lookup Table, aplicar filtros espaciales como: Sobel, Prewitt, Gaussianos, etc. y filtros en el dominio de la frecuencia.

- *Binarización*

Con estos bloques se asigna un valor de 1L a las áreas claras y de 0L a las áreas oscuras de la imagen, de esta manera se resalta la forma del área de interés en la imagen.

- *Machine Vision*

Mediante estos bloques se puede hacer la detección de bordes, medir la distancia entre los bordes, buscar cualquier forma de una imagen de acuerdo a una muestra o buscar una forma geométrica.

- *Identificación*

De esta paleta la opción que interesa es la de generar una clasificación de partículas (*Particle Classification*) (Figura 3.17), para desarrollar la base de datos que luego servirá para la identificación y clasificación de las señales de manos. La descripción para crear las bases de datos se explica más adelante.

Se entiende por partícula a una región continua de píxeles no nulos en una imagen.

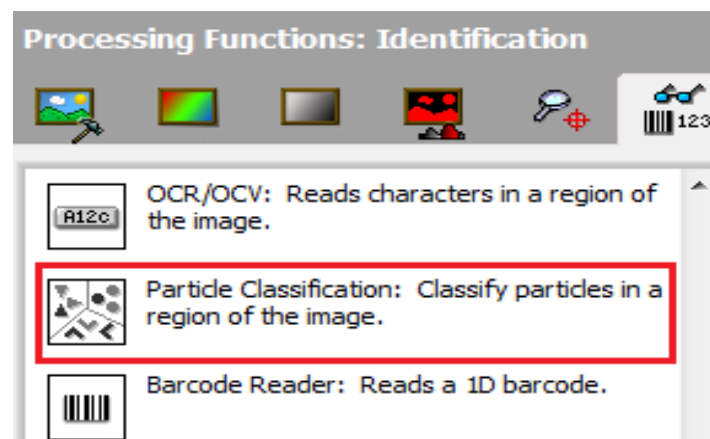


Figura 3.17 Menú de Clasificación.

✓ *Ventana Script*

En esta ventana se graba secuencialmente cada uno de los pasos para llegar a tener un algoritmo de procesamiento de imágenes, en este caso se realiza la conversión a escala de grises de la imagen de entrada. Como se puede observar en la Figura 3.18 los bloques que aparecen son los mismos que si se trabaja en el diagrama de bloques del Labview.

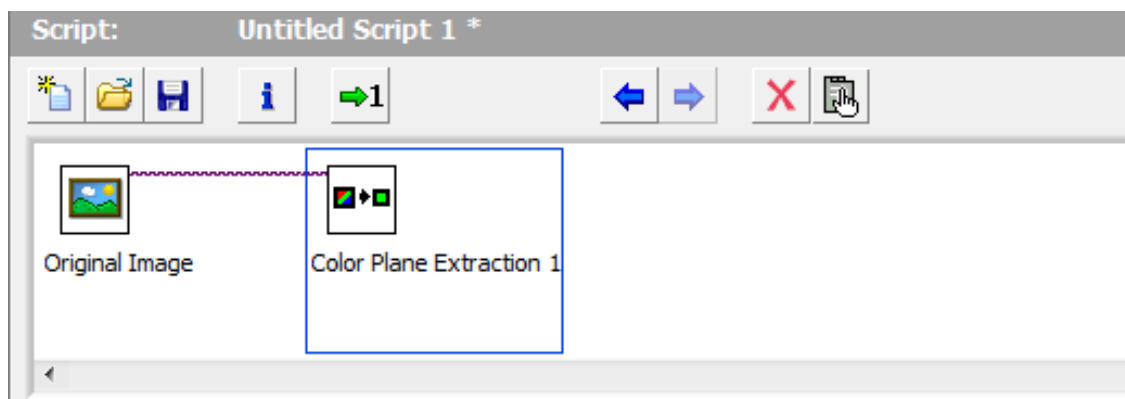


Figura 3.18 Ventana Script.

✓ *Ventana de Procesamiento*

En esta ventana se visualiza la imagen resultante luego que se aplica el procesamiento de imágenes.

3.2.2.6.2 Generación de un Archivo de Clasificación (Base de Datos)

En esta etapa se crean los vectores característicos de cada clase; es decir, se genera el archivo de clasificación. Para esto se toma la imagen fuente, se convierte a escala de grises y se crea una “*Particle Classification*”, tal y como se describió en la sección anterior. Partiendo de esto, se siguen a continuación los siguientes pasos para generar el archivo de clasificación:

✓ *Crear un nuevo archivo de clasificación*

Una vez que se abre la ventana (esquina inferior izquierda), se dibuja el ROI (Region of Interest) que consiste en dibujar un cuadro alrededor del objeto del estudio de tal manera que contenga toda la mano. Así como se observa en la Figura 3.19, luego de esto se selecciona la opción de Nuevo Archivo de Clasificación como se observa en la Figura 3.20.



Figura 3.19 ROI.

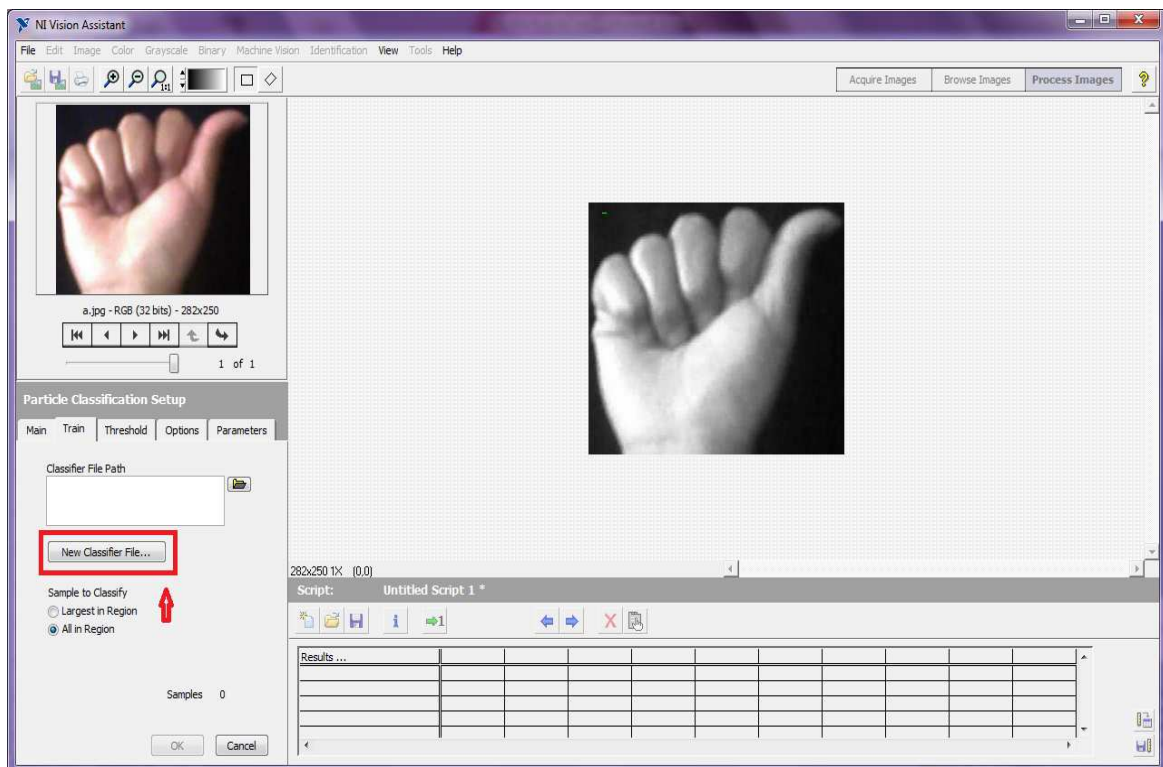


Figura 3.20 Opción para crear un nuevo archivo de clasificación.

✓ *Configuración de Parámetros*

A continuación se abre la ventana “*NI Particle Classification Training Interface*”, en esta ventana se van a generar los vectores característicos de cada clase en base a una fase de entrenamiento.

Esta ventana ofrece las opciones de “*Preprocessing*”, “*Engine Options*”, “*Particle Classifier Options*”; para configurar parámetros como: el umbral, el método de clasificación, -si se quiere que la imagen sea reconocida independientemente de la escala y si se desea reconocer con la opción “espejo”.

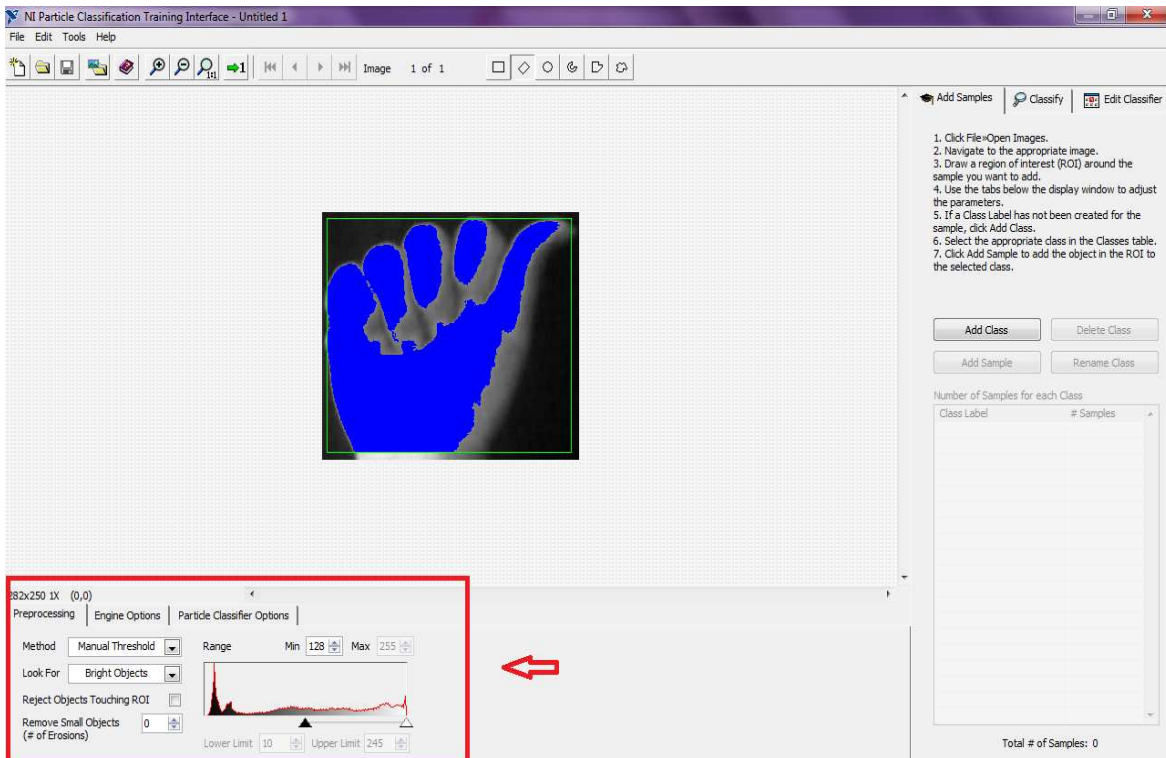


Figura 3.21 Configuración de parámetros de clasificación.

En la primera pestaña de “*Preprocessing*” se selecciona el método para el umbral (Threshold), si se busca un objeto brillante, oscuro o en grises y si se desea remover pequeños objetos. El umbral es el área del histograma (Figura 3.21), definida por un máximo y un mínimo dentro de los cuales se encuentra la mayor cantidad de información de la imagen.

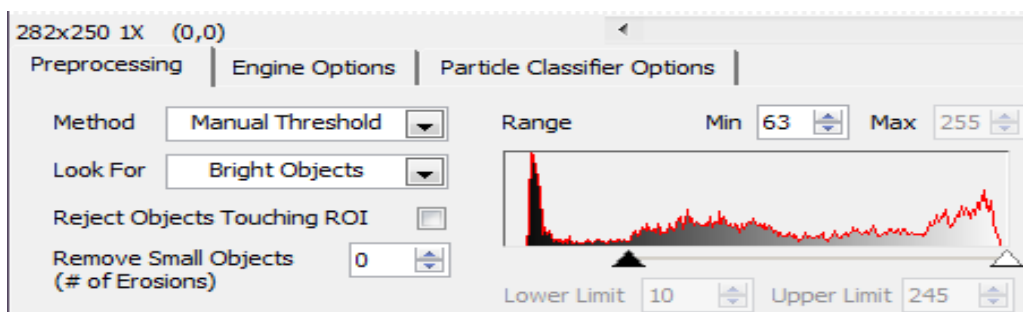


Figura 3.22 Histograma con límite inferior de 63 y un máximo de 255.

Para definir el umbral se tienen métodos automáticos y uno manual, en este caso se optó por el método manual para seleccionar de acuerdo a cada muestra el umbral correspondiente. La selección del límite inferior se la realiza en función al color azul que se sobrepone sobre la mano una vez dibujado el ROI, tratando que el color azul cubra la mayor cantidad de la mano.



Figura 3.23 Selección del umbral con límite inferior adecuado.

La ventaja de un fondo negro es que se logra que la mano contraste, por lo que el objeto a buscar se considera brillante para obtener mejores resultados a la hora del reconocimiento. Y ya que se tiene un ambiente controlado, en este caso no se requiere eliminar objetos indeseados.

En la pestaña de “*Engine Options*” se muestran los métodos para realizar la clasificación: Vecino más Cercano, *K*-Vecino más Cercano y Distancia Mínima de la Media.

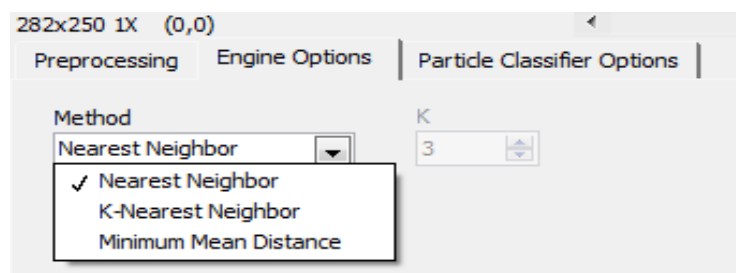


Figura 3.24 Selección del método de Clasificación.

Para determinar cuál método es el más apropiado para la aplicación se realizaron archivos de clasificación empleando los tres métodos, con el fin de realizar pruebas y determinar ya en la práctica, cuál arroja mejores resultados.

En este proyecto se maneja seis muestras por cada letra; es decir, que influyen varios factores a la hora de la clasificación. Estos son: tamaño de la mano entre muestras, similitud entre clases, conocimiento y habilidad para realizar los símbolos.

Con la influencia de estos factores se presentan pequeñas diferencias en la misma letra, a pesar que en su forma son parecidas, por tal motivo el método de clasificación a utilizar es el del *K*-Vecino más Cercano, las pruebas que justifican esta elección se presentan en el Capítulo 4. Este método se basa en un sistema de votación el cual toma *k* muestras más cercanas y en base al mayor número de coincidencias asigna la clase, además que es más tolerante al ruido.

El “*Vission Assistant*” ofrece tres opciones para el cálculo de la distancia entre las muestras de las bases de datos y la muestra a ser clasificada, estas opciones son: “*Euclidean*”, “*Sum*”, “*Maximum*”. Estos métodos se presentan en la Figura 3.25.

Distancia Euclideana (Euclidean).- Es menos sensible a pequeñas variaciones entre muestras, y se usa cuando se requiere clasificar muestras con pequeñas variaciones en la misma clase. Su ecuación es la siguiente:

$$d(X, Y) = \sqrt{\sum_{i=1}^n (X_i - Y_i)^2}$$

Distancia Suma (Sum).- Es conocida también como la Métrica Manhattan, es la más usada en procesos de clasificación y es la que viene seleccionada por default en el “*Vission Assistant*”. Se refiere al promedio, se usa en formas definidas. Su ecuación es la siguiente:

$$d(X, Y) = \sum_{i=1}^n |X_i - Y_i|$$

Distancia Máxima (Maximum).- Es el más sensible a pequeñas variaciones entre muestras, se usa cuando se requiere clasificar muestras con pequeñas variaciones en diferentes clases. Su ecuación es la siguiente:

$$d(X, Y) = \max_i |X_i - Y_i|$$

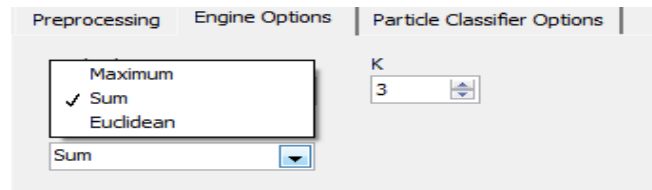


Figura 3.25 Métodos de cálculo de la distancia

En la pestaña “*Particle Classifier Options*” no se seleccionó ninguna opción, debido a que se busca que la clasificación sea independiente de la escala porque se tienen diferentes tamaños de manos, además que no se escogió la opción “espejo”.

✓ *Añadir una Clase*

Una vez configurados los parámetros se procede a crear la clase usando la opción “*Add Class*”, aquí se pide asignar un nombre a la clase en el caso del ejemplo se asigna la letra “A”, como se muestra en la Figura 3.26.

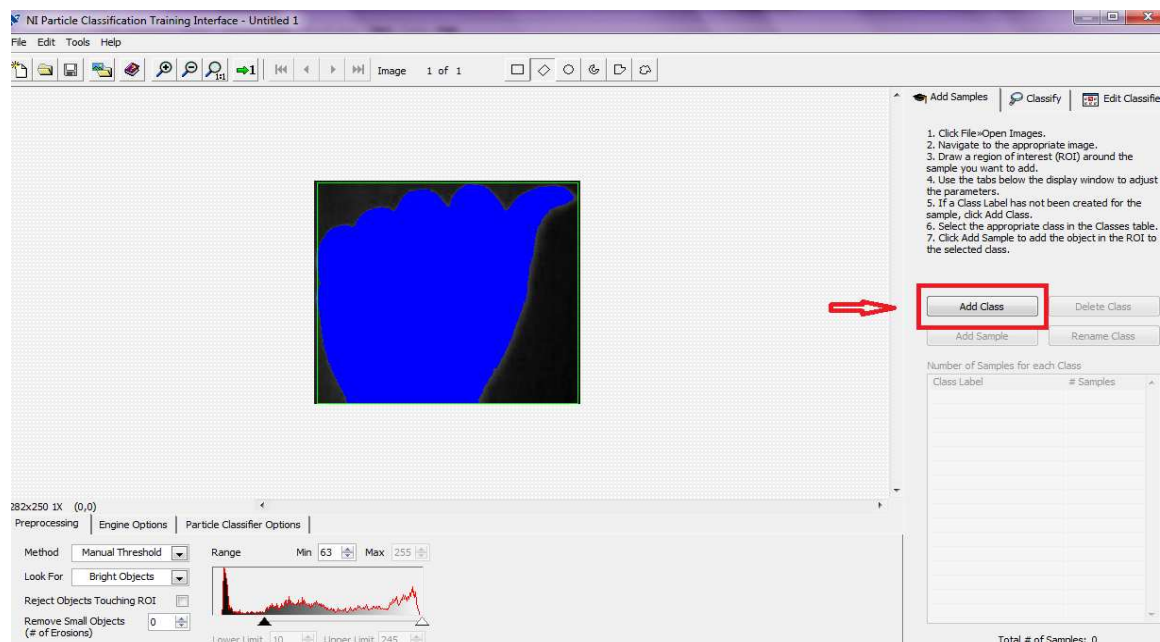


Figura 3.26 Añadir nueva clase.

✓ *Añadir una Muestra*

Luego de creada la clase se deben incluir las muestras dentro de la clase, para esto se selecciona la opción “*Add Sample*”. Luego de añadir la primera muestra se sigue el procedimiento anterior: abrir la imagen que corresponda a la misma

letra en este caso “A”, dibujar el ROI, definir el umbral, seleccionar el método y luego seleccionar la opción “Add Sample”, como se observa en la Figura 3.27.

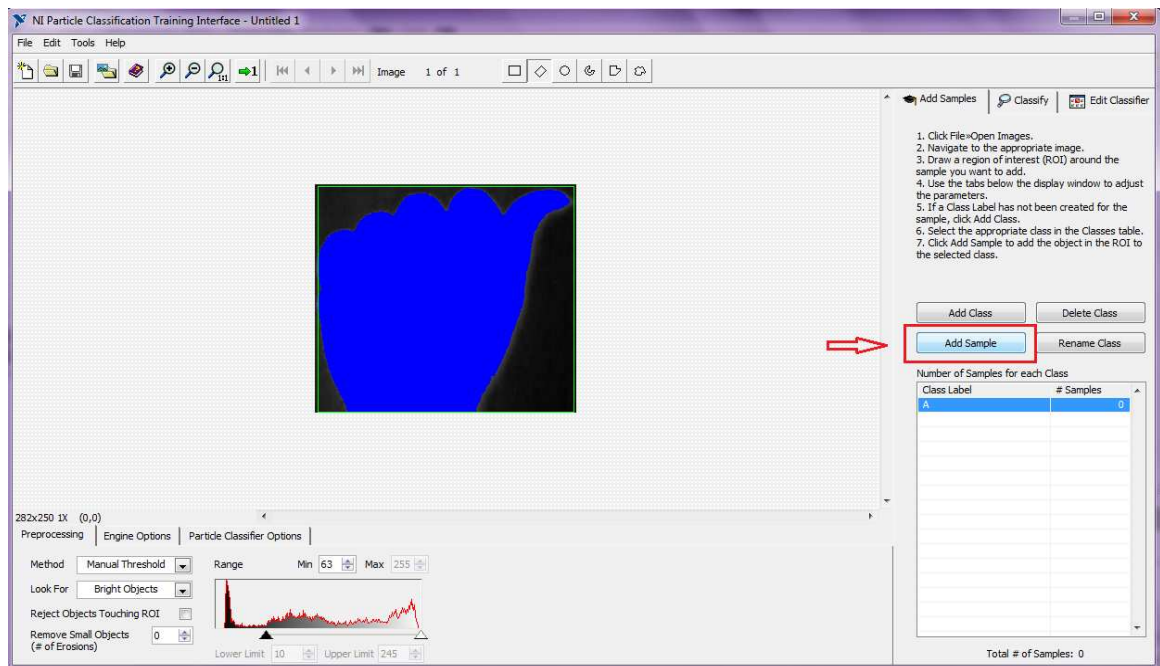


Figura 3.27 Añadir una muestra a una clase.

Como se mencionó anteriormente para cada letra se tienen seis fotografías (muestras); es decir, para cada clase se tienen seis muestras. En la Figura 3.28 se observa el número de muestras por cada clase y el número total de muestras.

Number of Samples for each Class	
Class Label	# Samples
G	6
H	6
I	6
K	6
L	6
M	6
N	6
O	6
P	6
Q	6
R	6
S	6
T	6
U	6
V	6
W	6
X	6
Y	6
-	6

Total # of Samples: 151

Figura 3.28 Número de muestras por clase.

✓ *Entrenar la Clasificación*

Una vez que se ha terminado de añadir a cada clase las seis muestras correspondientes se procede con la etapa de entrenamiento. En la pestaña “Classify” se tiene la opción “Train Classifier”, que sirve para entrenar la clasificación. Hay que tomar en consideración que se debe aplicar un método de clasificación en cada fase de entrenamiento; es decir, no se puede entrenar una base de datos con métodos de clasificación diferentes.

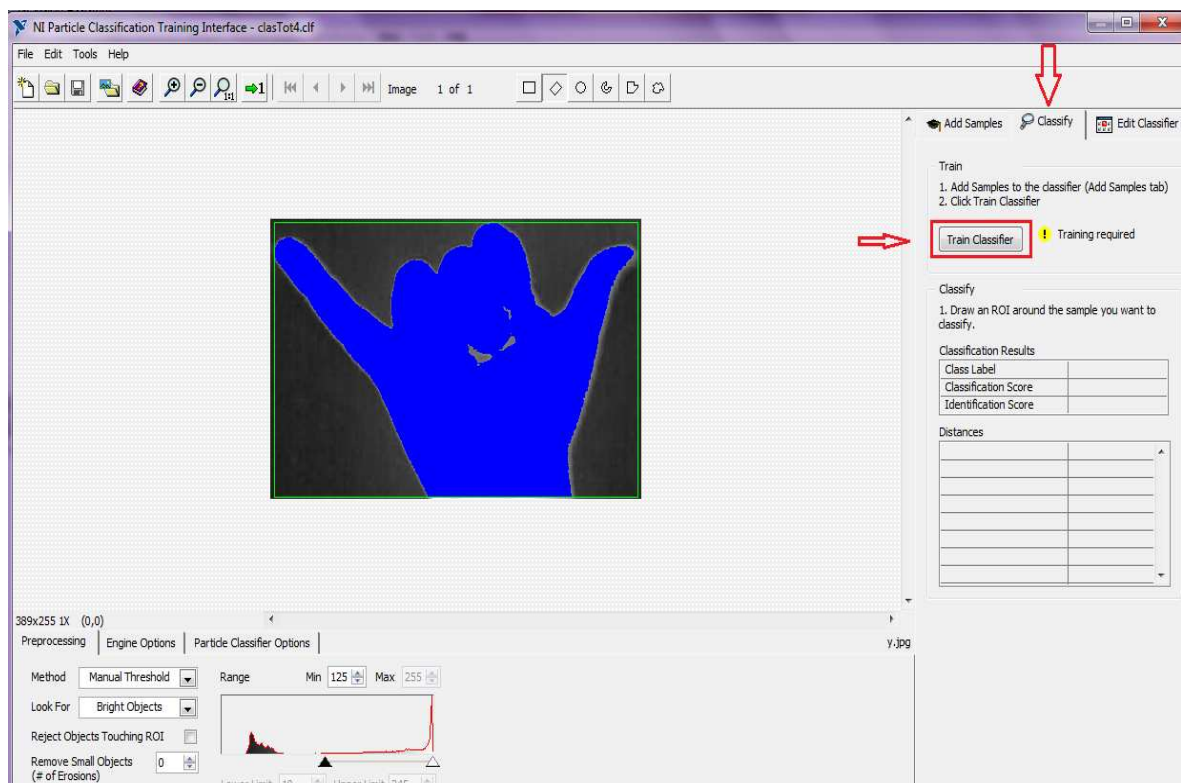


Figura 3.29 Fase de entrenamiento.

✓ *Guardar Archivo de Clasificación (.clf)*

Después de la fase de entrenamiento se guarda el archivo de clasificación, de preferencia en el directorio donde están las fotos usadas para crear el archivo. El archivo que se genera es de extensión .clf, como se observa en la Figura 3.30.

Para crear un nuevo archivo con las mismas muestras pero diferente método, no es necesario volver a cargar cada una de las fotos sino que simplemente se cambia el método en la opción de “Engine Options” y se vuelve a entrenar para luego guardarlo.

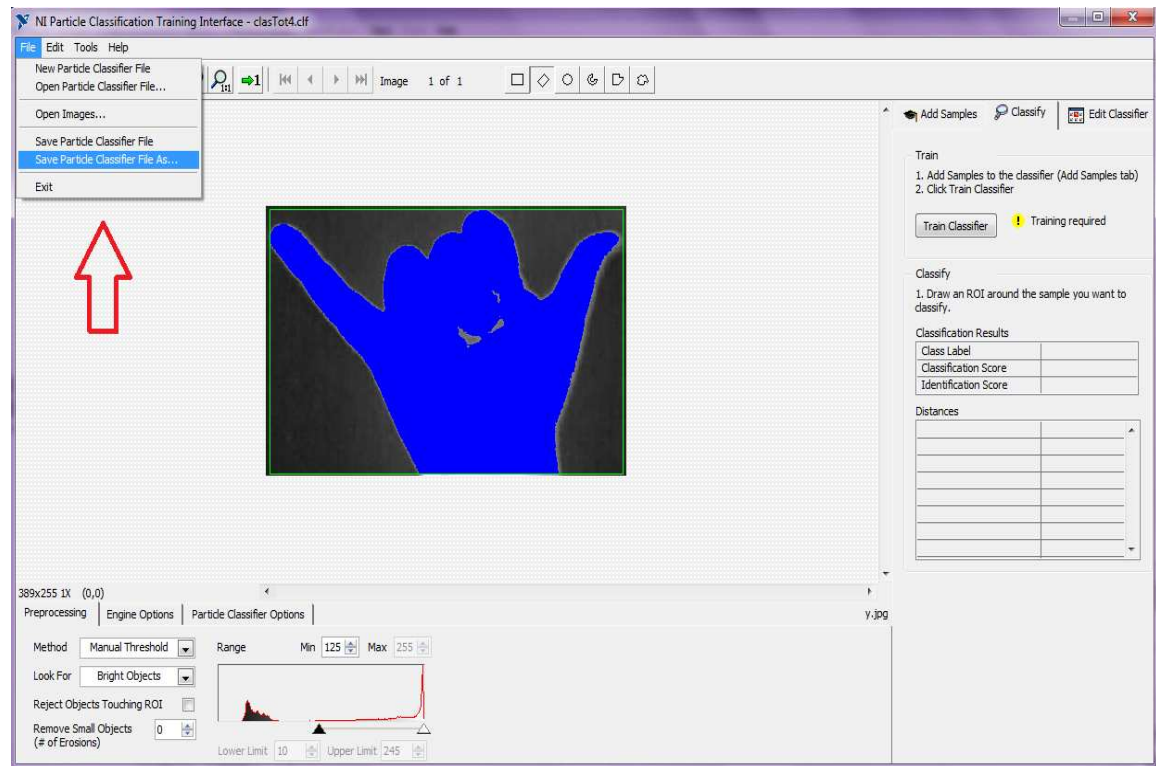


Figura 3.30 Guardar un archivo de clasificación.

✓ *Editar Clasificador*

En esta ventana se puede visualizar todas las muestras en su respectiva clase. Además es posible editar una muestra; es decir, se puede borrar la muestra o cambiarla a una clase diferente mediante las opciones “Delete” y “Relabel”, como se observa en la Figura 3.31. Para realizar cualquiera de las dos acciones se selecciona la muestra o muestras que se quieren editar y se aplica la opción que se necesite.

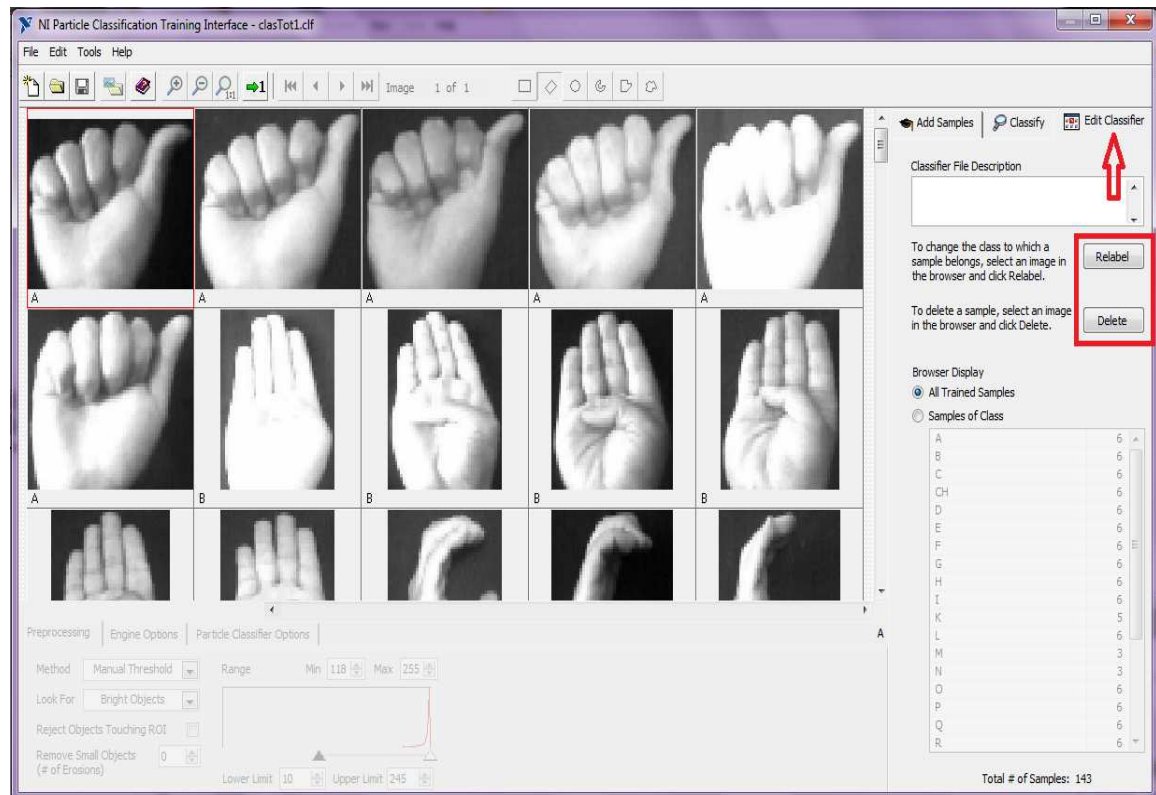


Figura 3.31 Ventana para editar clasificación.

3.2.2.7 Clasificación

En esta etapa de la aplicación se realiza la clasificación de la imagen basada en los archivos generados en el “*Visión Assistant*” (bases de datos) que contiene los vectores característicos de cada clase. Para realizar el proceso de clasificación se usa la paleta que se presenta en la Figura 3.32.

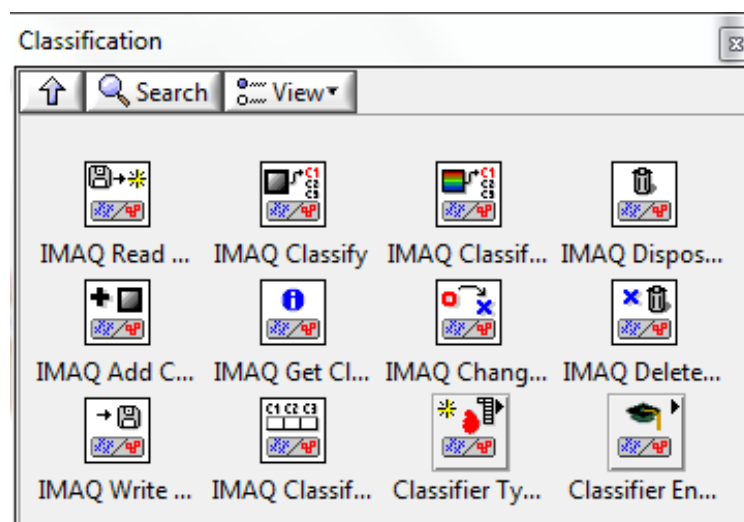


Figura 3.32 Paleta de Clasificación en Labview.

El código para realizar el proceso de clasificación en Labview es el que se muestra en la Figura 3.33.

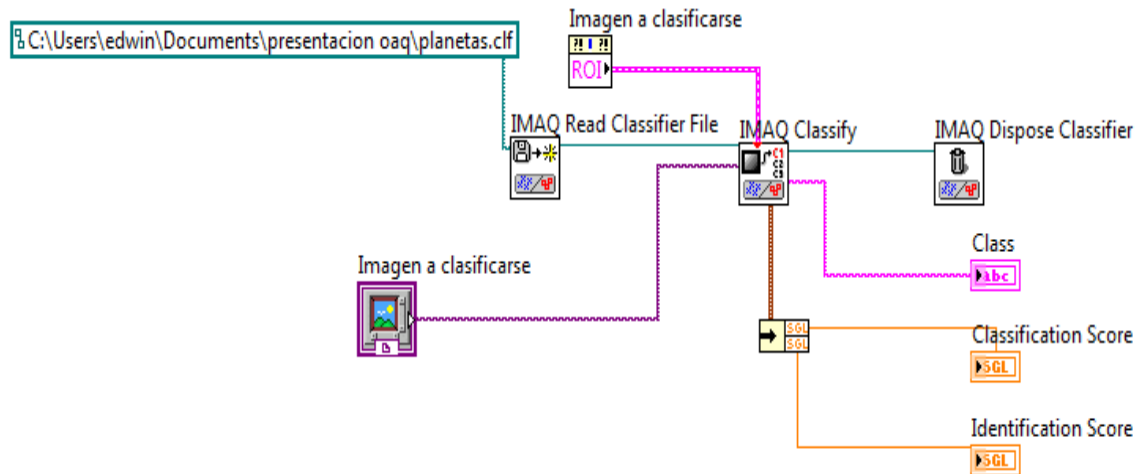


Figura 3.33 Código para realizar la clasificación de una imagen en Labview.

Con las bases de datos generadas en el “*Visión Assistant*” y con el código de la Figura 3.33, se realiza la clasificación de la imagen. Esta clasificación da como resultado un clúster que contiene los siguientes elementos:

- Class;
- Classification score;
- Identification score.

La “*Class*” es un dato tipo string que es el nombre de la clase a la que pertenece la señal de la mano.

El “*Classification score*” es un valor numérico de 0 a 1000 (siendo 1000 el valor más alto de acierto). Este valor representa que tan acertada es la clase asignada a la muestra ingresada.

El “*Identification score*” es un valor numérico el cual indica la similitud entre la entrada y clase asignada. Este dato se usa solo cuando no se puede tomar una decisión con el valor que proporciona el “*classification score*”.

Para mejorar el proceso de clasificación se utiliza el método “*K-Nearest Neighbor Classifier*” con las tres formas de calcular la distancia entre los vectores

característicos de cada clase; es decir, el proceso de clasificación utiliza tres bases de datos.

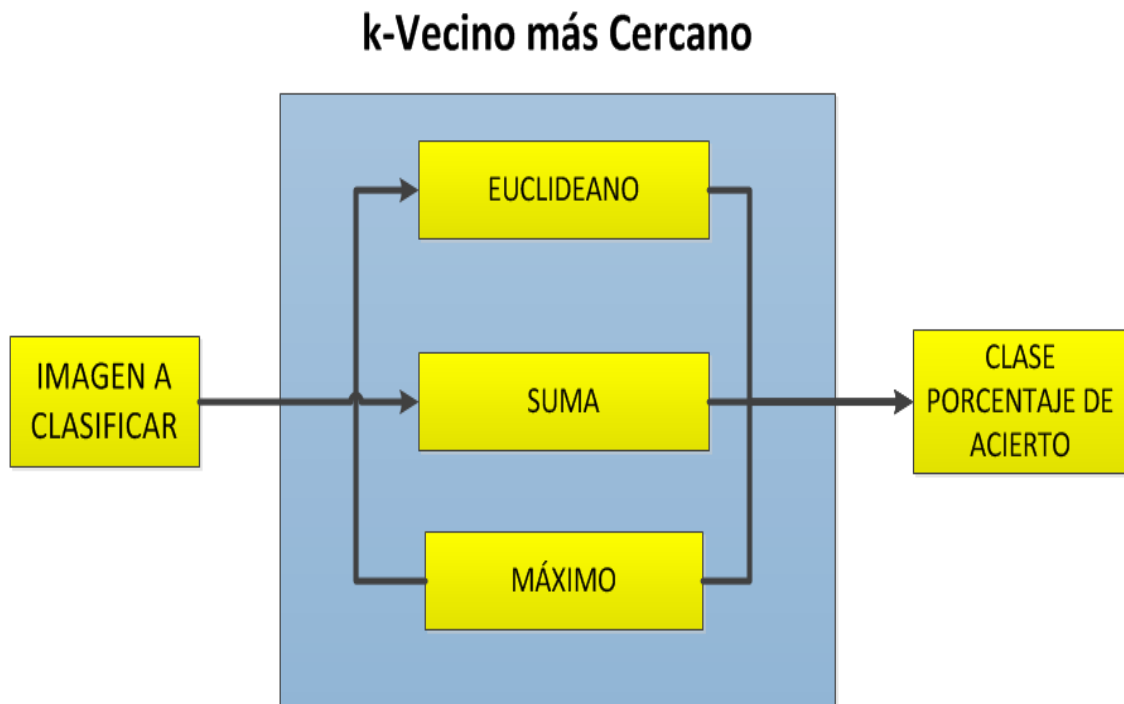


Figura 3.34 Diagrama de bloques del proceso de clasificación.

Como se observa en la Figura 3.34 el proceso de clasificación se compone de tres subprocesos, cada uno de éstos da como resultado una clase y un factor de acierto. Para aceptar la clasificación de una imagen como correcta, el valor del “*Classification score*” debe ser de 700 o mayor. Hay que tener en cuenta que el método con el que se generó la base de datos define la forma como se va a calcular la distancia entre los vectores de las clases y la imagen a clasificarse.

Si el valor del “*classification score*” es superior al límite aceptable de 700 se acepta la clasificación y se guarda el resultado de la “class”. Este valor se sobrepone a la imagen (letra en verde ver Figura 3.35) y se envía hacia el algoritmo que se encarga de formar el texto de la traducción.

Al escoger el valor de 700 para aceptar una clasificación, se logra eliminar clasificaciones equivocadas con rangos bajos.

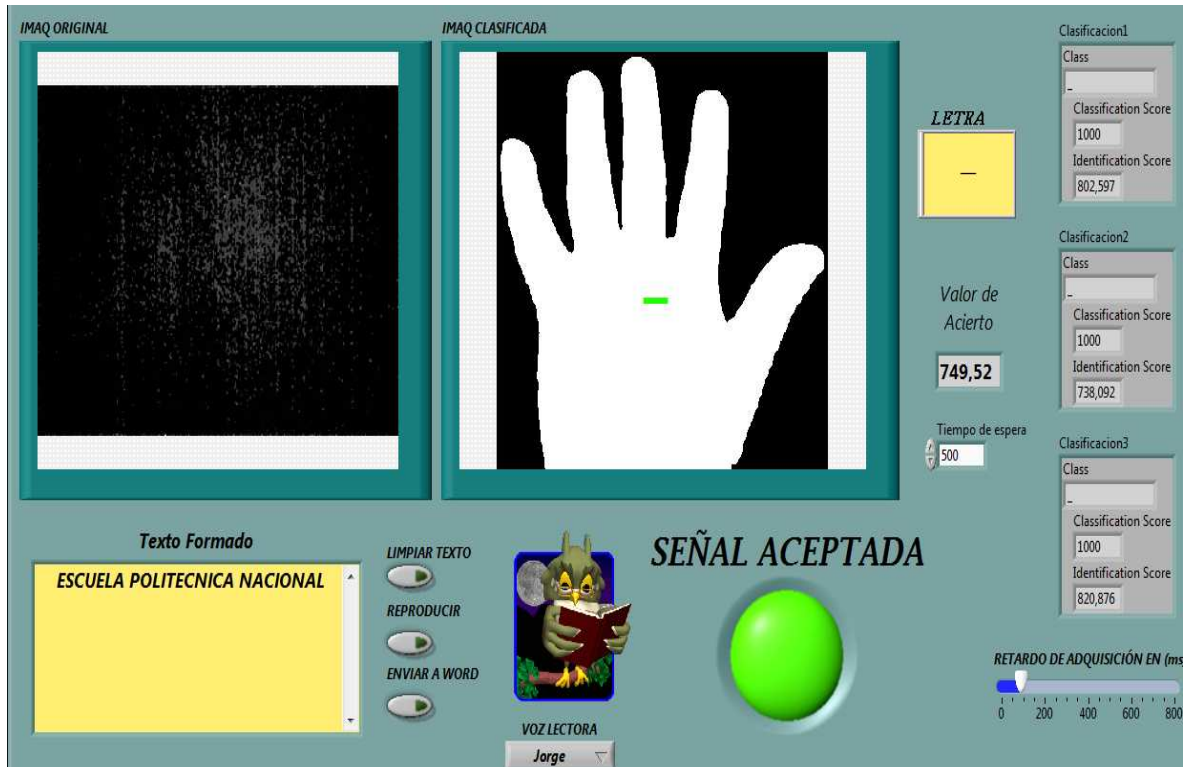


Figura 3.35 Imagen clasificada y texto formado.

Para unir los resultados de la clasificación y formar un texto se toma en cuenta las siguientes consideraciones:

- Que el primer string debe ser el último del texto.
- Evitar errores de repetición.
- Que los strings deben concatenarse para formar una palabra.

Hay que tener en cuenta que esto limita la escritura de palabras que tienen letras repetidas por ejemplo: carro, perro, etc. Pero se logra evitar errores graves como una cadena de caracteres completamente irreconocibles.

Para evitar el error que provoca el cambio de la forma de la mano entre señal y señal se implementó un algoritmo. Este extrae el promedio de las clasificaciones aceptadas en un periodo de tiempo, que se puede ajustar a la habilidad del usuario. Con esto se logra evitar errores que se producen en el transcurso del cambio de una señal a otra.



Figura 3.36 Panel frontal del Vi que ensambla el texto.

Para que el usuario note que el sistema reconoce la letra que está haciendo, se implementa una realimentación luminosa, con esto se hace que el sistema sea amigable con el usuario.

3.2.3 ENTRENADOR

Para ayudar al usuario a obtener el conocimiento y habilidad necesarios para usar el lenguaje de signos, se implementa un proceso de entrenamiento que tiene dos modos de operación: automático y manual. En modo automático el software genera aleatoriamente la letra con su correspondiente signo, para que el usuario imite la posición de la mano que se muestra en la imagen.

Una vez que el usuario realizó correctamente el signo que el software generó, se activa el indicador de señal aceptada y se genera automáticamente otro signo. En caso de que el usuario no logre realizar el signo correctamente, tiene la posibilidad de cambiar a un nuevo signo con el botón “*Genera*”. El panel frontal del entrenador se observa en la Figura 3.37.

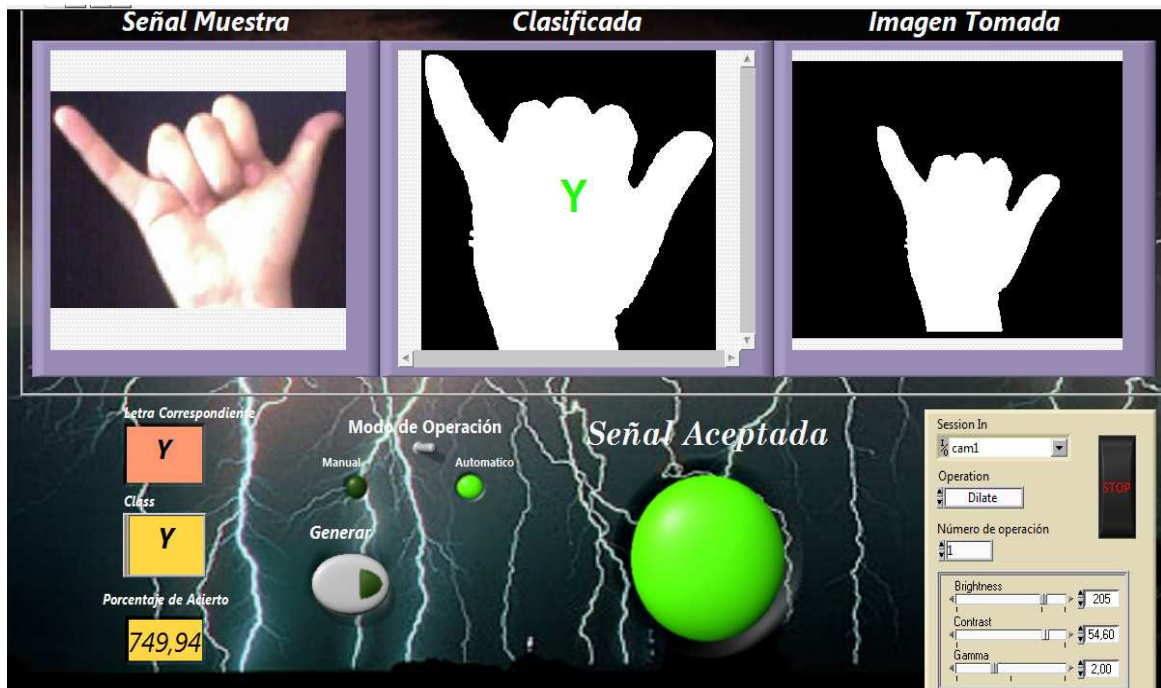


Figura 3.37 Panel frontal del entrenador en modo automático.

En modo manual el software permite al usuario elegir la letra que desea practicar. Aparece la imagen del signo escogido y el sistema de reconocimiento funciona de igual manera que en el modo automático.

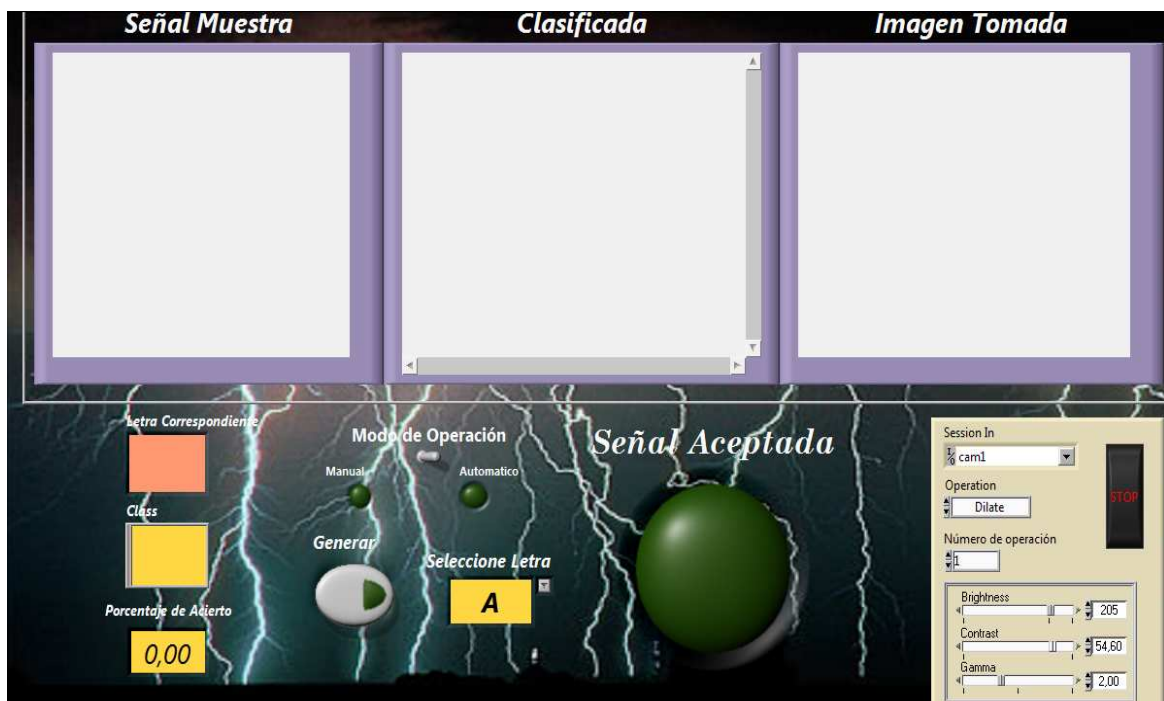


Figura 3.38 Panel frontal del entrenador en modo manual.

Para calibrar el sistema existe el panel de configuraciones donde es posible adaptar el sistema a los niveles de iluminación donde se esté utilizando.

Para detener el sistema existe un botón de “STOP” con el cual regresa al panel principal y se puede escoger entre el traductor y el entrenador.

3.3 DIAGRAMAS DE FLUJO

3.3.1 DIAGRAMA DE FLUJO PRINCIPAL

En el siguiente diagrama de flujo se explica el algoritmo que ejecuta el sistema del traductor.

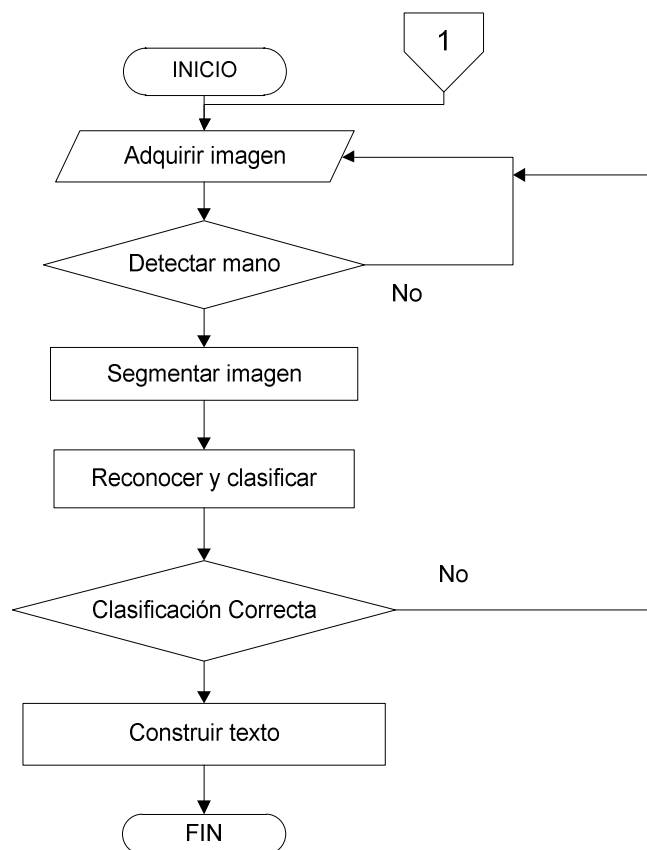


Figura 3.39 Diagrama de flujo principal.

A continuación se procede a explicar con lenguaje estructurado el diagrama de flujo de la Figura 3.39.

Adquirir imagen

Escoger cámara a utilizar en el proceso, en el icono “IMAQdx Session”;

Ajustar parámetros focales de imagen;

Ajustar el sistema de iluminación;

Mostrar imagen adquirida.

Fin de tarea

Las demás subrutinas del diagrama de flujo principal tienen un grado alto de complejidad, por esta razón se realiza un diagrama de flujo para cada una de estas subrutinas. Estos diagramas se presentan a continuación:

3.3.2 DIAGRAMA DE FLUJO DE LA SUBROUTINA “DETECTAR MANO”

En el siguiente diagrama de flujo se explica la lógica que realiza el programa para detectar si la mano del usuario se encuentra dentro del área de trabajo

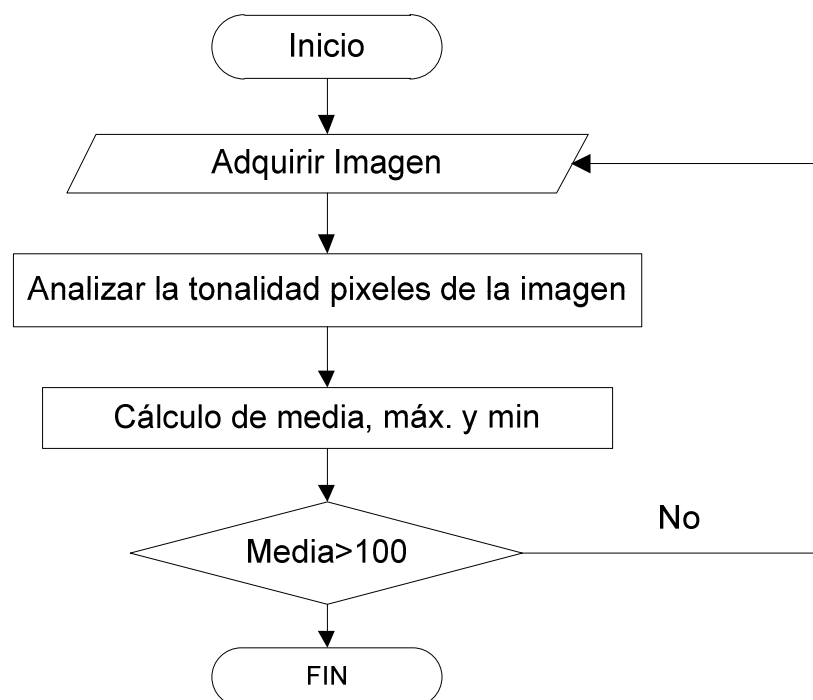


Figura 3.40 Diagrama de la subrutina “detectar mano”.

A continuación se procede a explicar con lenguaje estructurado el diagrama de flujo de la Figura 3.40.

Detectar mano

Analizar el histograma de la imagen.

Calcular en base a tonalidad de los pixeles su valor de media, máximo y mínimo.

Abrir valores de la media, máximo y mínimo obtenidos del ambiente controlado.

Si valores calculados son mayores a los del ambiente controlado.

Existe mano en la imagen

Pasar a segmentar la imagen

Caso contrario

Regresar a adquisición de imagen.

Fin de tarea

3.3.3 DIAGRAMA DE FLUJO DE LA SUBROUTINA “SEGMENTAR IMAGEN”

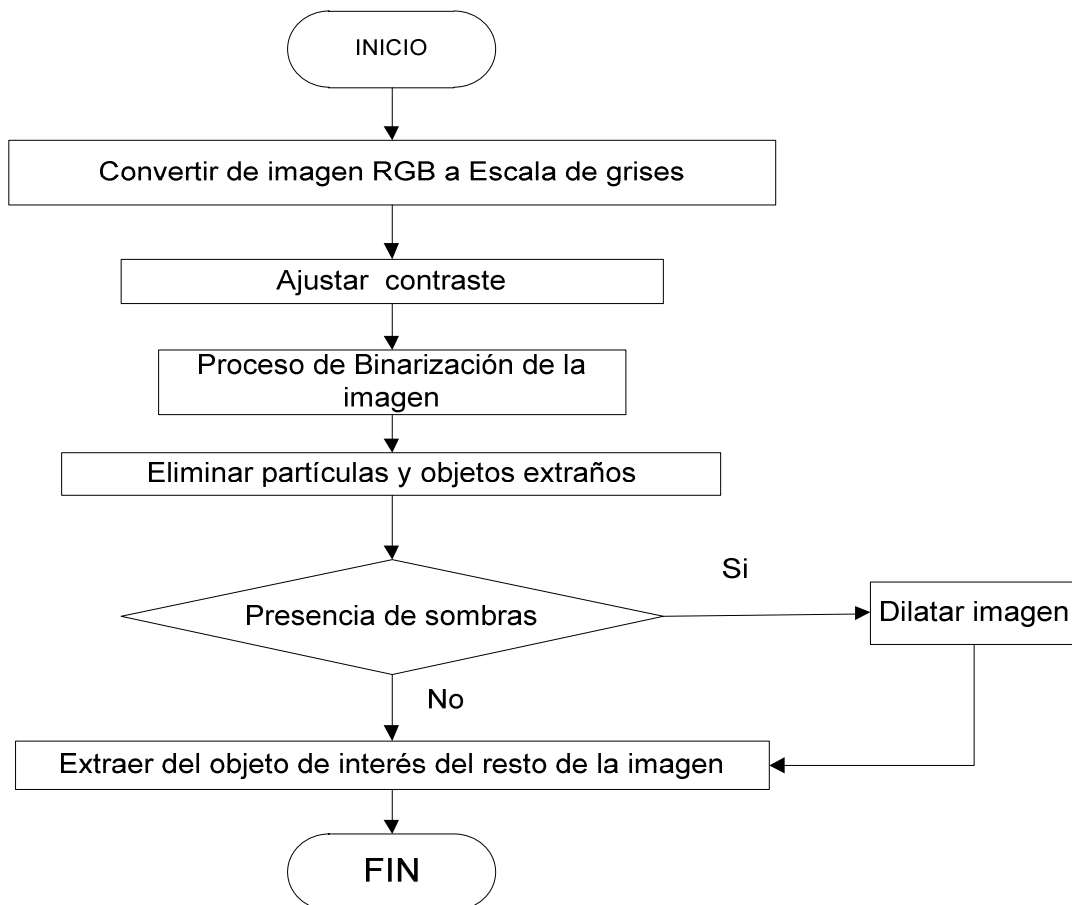


Figura 3.41 Diagrama de flujo de la subrutina “segmentar imagen”.

A continuación se procede a explicar con lenguaje estructurado el diagrama de flujo de la Figura 3.41.

Segmentar imagen

Convertir imagen de RGB a Escala de grises.

Ajustar el contraste de la imagen para resaltar el objeto de interés.

Binarizar la imagen de acuerdo a la iluminación del ambiente.

Eliminar partículas de los bordes del objeto de interés.

Si existen sombras en la objeto de interés

Dilatar hasta que se eliminen.

Extraer el objeto de interés del resto de la imagen para hacer más rápido el proceso de clasificación.

Fin de tarea

3.3.4 DIAGRAMA DE FLUJO DE CLASIFICACIÓN

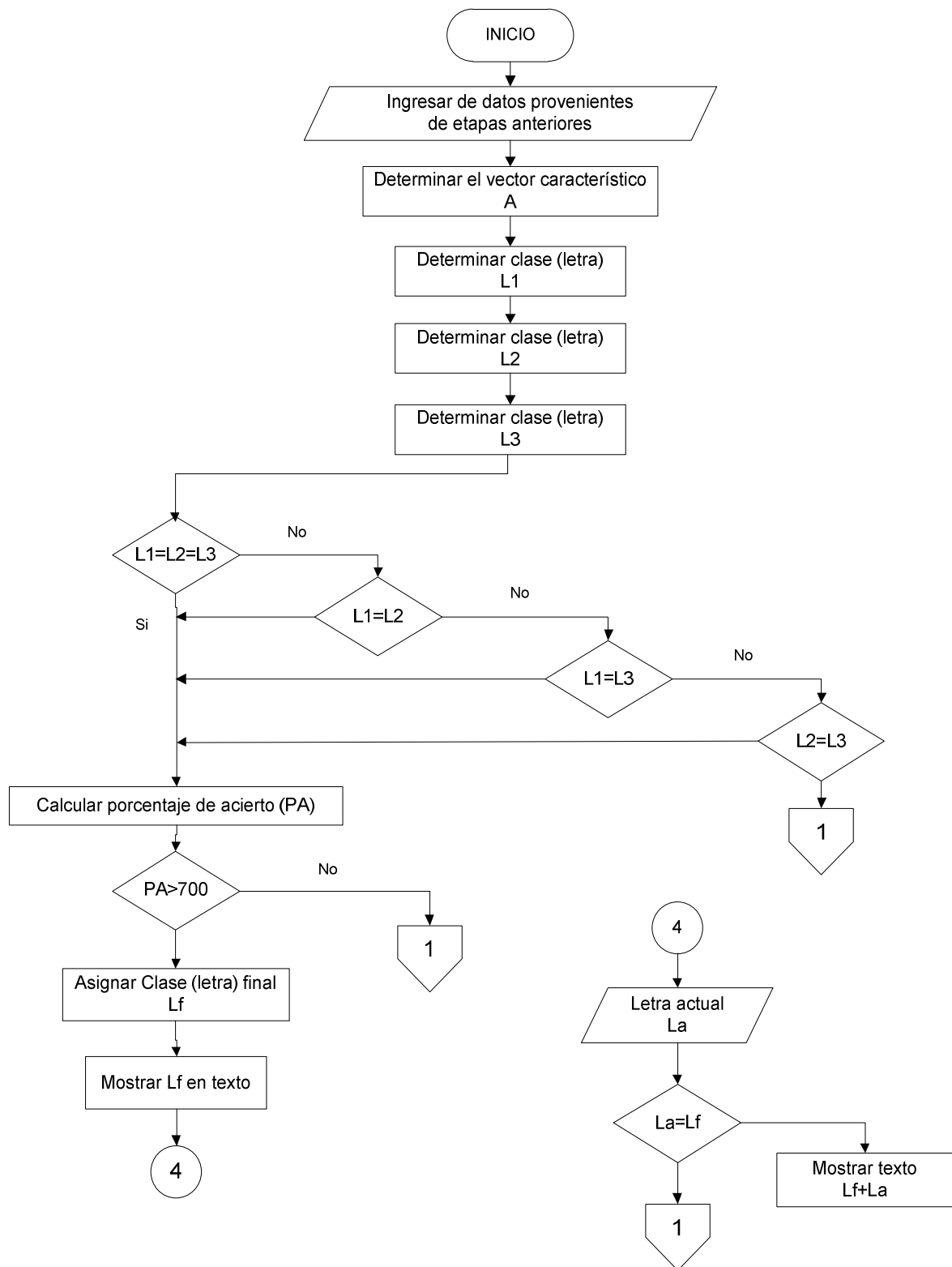


Figura 3.42 Diagrama de flujo de clasificación.

A continuación se procede a explicar con lenguaje estructurado el diagrama de flujo de la Figura 3.42.

Ingresar datos provenientes de etapas anteriores

Abrir Bases de Datos (Vectores Característicos) 4, 5 y 6.

Ingresar la imagen recortada.

Fin de tarea

Determinar el vector característico A

Tomar forma de la imagen recortada.

Asignar un vector característico.

Fin de tarea

Determinar clase (letra) L1

Comparar A con base de datos 4 (*K*-Vecino más Cercano con Euclidean).

Determinar el índice de acierto.

Asignar la clase correspondiente de acuerdo al índice de acierto.

Fin de tarea

Determinar clase (letra) L2

Comparar A con base de datos 5 (*K*-Vecino más Cercano con Sum).

Determinar el índice de acierto.

Asignar la clase correspondiente de acuerdo al índice de acierto.

Fin de tarea

Determinar clase (letra) L3

Comparar A con base de datos 6 (*K*-Vecino más Cercano con Maximum).

Determinar el índice de acierto.

Asignar la clase correspondiente de acuerdo al índice de acierto.

Fin de tarea

Calcular porcentaje de acierto final

Determinar el porcentaje del mayor índice de acierto (PA) como resultado de las comparaciones.

Fin de tarea

Asignar la letra final Lf

Si el porcentaje de acierto es correcto

Asignar la letra definitiva correspondiente

Pasar a formar el texto.

Mostrar en la ventana del texto.

Fin de tarea

Letra actual La

Asignar letra La a la nueva señal.

Comparar la letra anterior con la letra actual.

Si no son iguales

Mostrar en la ventana de texto junto a la letra anterior.

Si son iguales

Regresar al programa principal.

Fin de tarea

3.3.5 DIAGRAMA DE FLUJO DEL ENTRENADOR

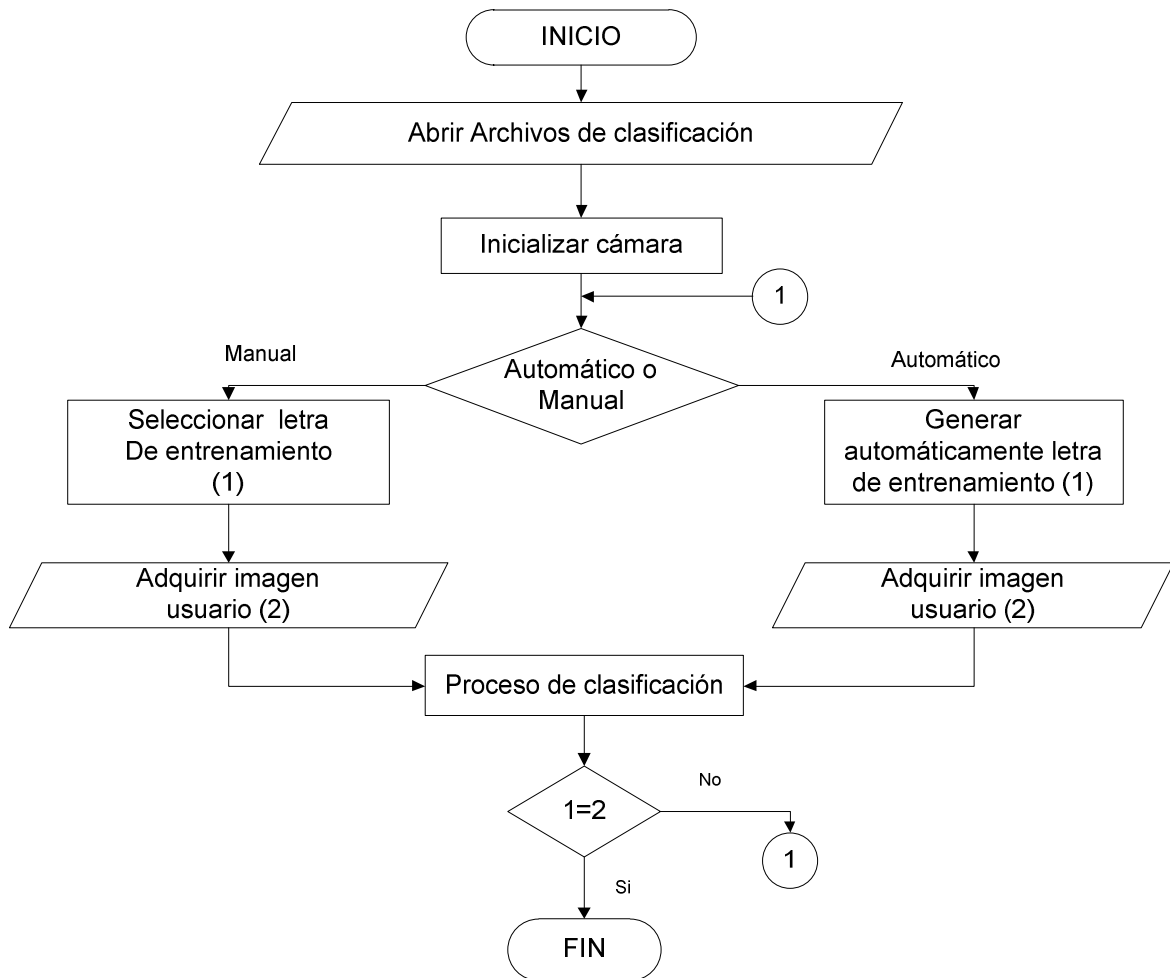


Figura 3.43 Diagrama de flujo del entrenador.

A continuación se procede a explicar con lenguaje estructurado el diagrama de flujo de la Figura 3.43.

Abrir archivos de clasificación

Buscar archivos de clasificación

Cargar en el programa.

Fin de Tarea

Inicializar cámara

Escoger cámara a usar en el proceso

Reservar espacios de memoria para las imágenes que se están adquiriendo.

Fin de Tarea

Automático o Manual

Escoger la forma de uso del entrenamiento.

Fin de Tarea

Manual, Seleccionar letra “1” de entrenamiento

Escoger manualmente la letra que se desea realizar.

Mostrar la imagen y la letra a la que corresponde.

Escoger una nueva letra a realizar.

Fin de Tarea

Automático, Generar automáticamente letra “1” de entrenamiento

Generar de forma randómica la letra con su correspondiente imagen.

Cambiar la letra al presionar el botón “Generar”.

Fin de Tarea

Adquirir imagen de usuario “2”

Adquirir la imagen de la señal que está haciendo el usuario.

Pasar al proceso de clasificación

Comparar con la imagen de entrenamiento.

Fin de Tarea

Proceso de Clasificación

Analizar la imagen que contiene la señal realizada por el usuario.

Extraer el vector de características.

Comparar con la base de datos.

Si la imagen del usuario corresponde a la imagen en el entrenamiento.

Aceptar la señal.

Generar una nueva señal.

Fin de Tarea

En este capítulo se explica detalladamente cada uno de los algoritmos que conforman el software de la aplicación, de tal manera que se entienda la lógica del programa y se pueda realizar modificaciones y mejoras en trabajos futuros.

CAPÍTULO 4

PRUEBAS Y RESULTADOS

4.1 INTRODUCCIÓN

En este capítulo se muestran los resultados obtenidos al probar el algoritmo con los diferentes métodos de clasificación, además del cálculo del error que existe en el sistema terminado.

4.2 PRUEBAS REALIZADAS PARA LA SELECCIÓN DEL MÉTODO DE CLASIFICACIÓN.

Para la selección del método de clasificación se realizó bases de datos con cada uno de los métodos: *k*-Vecino más cercano, Vecino más cercano y Distancia mínima. El cálculo de la distancia para cada uno de los métodos, se realizó por: sum, max y euclidean; lo cual generó 9 bases de datos para la clasificación. Estas pruebas se realizaron con cada una de estas bases de datos y con dos usuarios diferentes.

4.2.1 K-VECINO MÁS CERCANO

Con este método se realizaron tres archivos de clasificación basados en los tres métodos de cálculo de distancia. Los valores de clasificación obtenidos se presentan en las siguientes tablas.

K-Vecino más Cercano (índices de acierto)				
Letra	Euclidean	Maximum	Sum	Total
A	1000,00	1000,00	1000,00	749,94
B	1000,00	1000,00	1000,00	749,94
C	1000,00	1000,00	500,00	749,94
D	1000,00	1000,00	1000,00	749,94
E	500,00	500,00	500,00	500,05
F	1000,00	500,00	1000,00	749,94
G	1000,00	1000,00	1000,00	749,94
H	1000,00	1000,00	500,00	749,94
I	1000,00	1000,00	1000,00	749,94
K	1000,00	1000,00	1000,00	749,94
L	1000,00	1000,00	500,00	749,94

K-Vecino más Cercano (índices de acierto)				
Letra	Euclidean	Maximum	Sum	Total
M	1000,00	1000,00	1000,00	749,94
N	1000,00	1000,00	500,00	749,94
O	1000,00	1000,00	500,00	749,94
P	1000,00	1000,00	1000,00	749,94
Q	1000,00	1000,00	1000,00	749,94
R	1000,00	500,00	1000,00	749,94
S	500,00	1000,00	500,00	749,94
T	1000,00	1000,00	1000,00	749,94
U	500,00	500,00	1000,00	749,94
V	1000,00	1000,00	1000,00	749,94
W	500,00	No reconoce	500,00	500,05
X	500,00	500,00	1000,00	749,94
Y	1000,00	1000,00	1000,00	749,94
PROMEDIO	895,83	891,30	833,33	729,12

Tabla 4.1 Índices de acierto para k- Vecino más Cercano (Usuario 1).

K-Vecino más Cercano (índices de acierto)				
Letra	Euclidean	Maximum	Sum	Total
A	1000,00	1000,00	1000,00	749,94
B	1000,00	1000,00	1000,00	749,94
C	1000,00	1000,00	1000,00	749,94
D	1000,00	1000,00	1000,00	749,94
E	500,00	500,00	1000,00	749,94
F	1000,00	500,00	1000,00	749,94
G	1000,00	500,00	1000,00	749,94
H	1000,00	1000,00	1000,00	749,94
I	1000,00	500,00	1000,00	749,94
K	500,00	500,00	500,00	500,05
L	500,00	1000,00	500,00	749,94
M	1000,00	1000,00	1000,00	749,94
N	500,00	1000,00	1000,00	749,94
O	1000,00	1000,00	1000,00	749,94
P	1000,00	1000,00	1000,00	749,94
Q	1000,00	1000,00	1000,00	749,94
R	1000,00	500,00	1000,00	749,94
S	1000,00	1000,00	1000,00	749,94
T	1000,00	0,00	500,00	300,03
U	1000,00	500,00	1000,00	749,94
V	1000,00	1000,00	1000,00	749,94

K-Vecino más Cercano (índices de acierto)				
Letra	Euclidean	Maximum	Sum	Total
W	500,00	500,00	1000,00	749,94
X	1000,00	1000,00	1000,00	749,94
Y	1000,00	1000,00	1000,00	749,94
PROMEDIO	895,83	791,67	937,50	720,78

Tabla 4.2 Índices de acierto para k - Vecino más Cercano (Usuario 2).

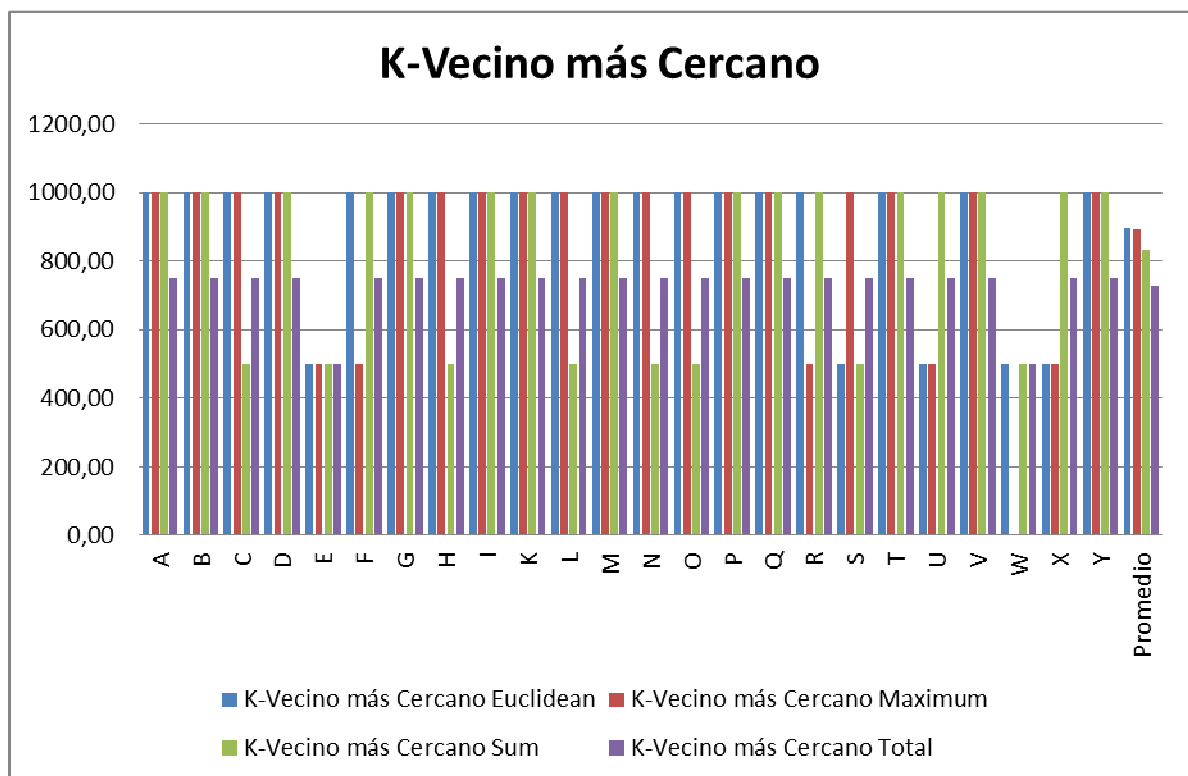


Figura 4.1 Distribución de las distancias para k -Vecino más Cercano.

De los resultados de la Tabla 4.1 y la Tabla 4.2 se observa que las bases de datos dan la clasificación correcta, con altos porcentajes de acierto en cada una de las clases sin confusión entre letras.

4.2.2 VECINO MÁS CERCANO

Con este método se realizaron tres archivos de clasificación basados en los tres métodos de cálculo de distancia. Los valores de clasificación obtenidos se presentan en las siguientes tablas.

Vecino Más Cercano (índices de acierto)				
Letra	Euclidean	Maximum	Sum	Total
A	604,00	550,13	526,84	601,76
B	855,71	823,98	860,57	750,06
C	52,69	52,18	98,87	249,24
D	489,19	440,60	470,37	469,24
E	508,69	365,40	609,29	610,36
F	416,67	467,71	364,71	407,39
G	587,43	515,33	627,48	627,46
H	No reconoce			
I	610,71	580,19	727,20	717,61
K	18,11	151,17	227,86	250,30
L	148,07	340,64	140,32	249,66
M	492,50	554,09	434,60	545,19
N	480,95	412,22	508,30	487,49
O	855,89	908,00	809,38	749,08
P	456,14	387,83	541,65	498,36
Q	311,73	51,44	225,43	250,20
R	503,76	602,23	523,99	599,85
S	300,91	304,04	247,94	298,74
T	386,02	481,19	448,22	434,74
U	150,48	159,87	82,12	249,61
V	288,11	379,66	167,56	301,98
W	335,57	164,60	484,35	332,73
X	300,48	72,40	419,93	349,26
Y	177,98	203,94	203,26	250,69
PROMEDIO	405,73	389,95	423,92	447,00

Tabla 4.3 Índices de acierto para Vecino más Cercano (Usuario 1).

Vecino Más Cercano (índices de acierto)				
Letra	Euclidean	Maximum	Sum	Total
A	332,80	424,79	284,85	329,92
B	736,12	733,97	723,83	736,24
C	358,03	218,41	370,01	360,83
D	858,40	859,11	840,77	750,04
E	331,06	530,64	149,77	334,24
F	561,34	421,74	628,73	628,81
G	675,48	694,85	669,09	691,89
H	470,66	393,50	443,97	433,53
I	173,66	211,98	94,01	249,05
K	748,92	729,81	589,55	748,30
L	141,53	344,99	120,54	250,23

Vecino Más Cercano (índices de acierto)				
Letra	Euclidean	Maximum	Sum	Total
M	244,27	93,48	3,40	250,39
N	576,87	497,68	592,44	589,48
O	569,04	719,81	439,65	690,87
P	101,75	76,64	167,57	249,92
Q	25,01	192,81	58,93	249,35
R	513,07	531,84	507,85	530,82
S	No reconoce			
T	165,87	71,13	194,93	250,26
U	610,23	606,46	605,25	500,00
V	582,70	540,04	422,28	579,55
W	No reconoce			
X	607,83	564,26	619,57	618,88
Y	463,68	499,12	358,89	451,94
PROMEDIO	535,76	531,69	489,23	535,41

Tabla 4.4 Índices de acierto para Vecino más Cercano (Usuario 2).

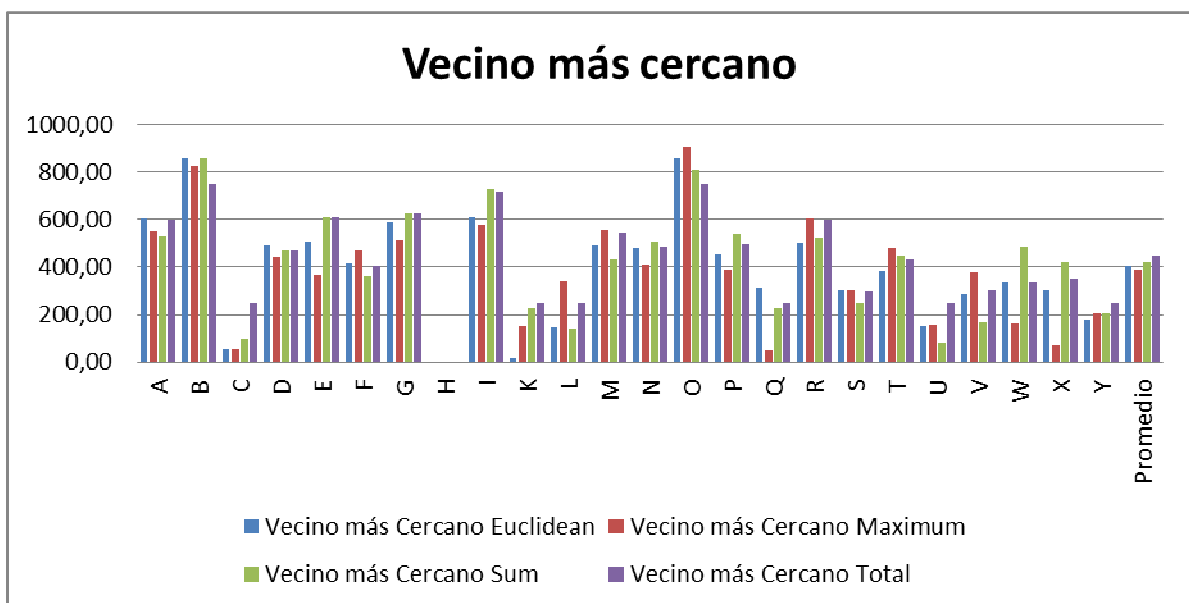


Figura 4.2 Distribución de las distancias para Vecino más Cercano.

Del análisis de la Tabla 4.3 y de la Tabla 4.4 se concluye que con este método los resultados de la clasificación son medianamente bajos, además de que algunas letras no se reconocen correctamente lo que ocasiona confusión en la interpretación.

4.2.3 DISTANCIA MINIMA

Con este método se realizaron tres archivos de clasificación basados en los tres métodos de cálculo de distancia. Los valores obtenidos con este método se detallan en las siguientes tablas.

Distancia Mínima (índices de acierto)				
Letra	Euclidean	Maximum	Sum	Total
A	596,13	611,14	597,03	609,54
B	88,69	26,12	15,80	252,39
C	469,51	544,74	461,47	513,03
D	248,35	149,64	156,80	250,16
E	446,16	462,25	404,50	439,16
F	508,74	341,87	513,64	518,57
G	323,62	387,96	315,52	332,52
H	34,43	10,04	64,58	251,32
I	844,50	831,69	849,62	749,94
K	173,91	57,55	233,78	249,94
L	230,65	32,56	346,46	249,67
M	241,11	152,49	256,71	250,30
N	59,56	46,66	10,68	249,63
O	48,10	351,08	84,12	249,63
P	602,23	716,16	506,27	706,65
Q	534,70	459,46	539,70	538,75
R	175,81	272,74	22,27	251,71
S	382,70	337,67	344,22	350,67
T	103,30	33,55	130,63	251,10
U	652,45	576,97	689,22	682,32
V	187,54	130,18	395,62	250,10
W	425,08	306,66	415,18	418,54
X	223,01	307,18	36,18	249,22
Y	331,74	214,78	346,75	333,57
PROMEDIO	330,50	306,71	322,36	383,27

Tabla 4.5 Índices de acierto para Distancia Mínima (Usuario 1).

Distancia Mínima (índices de acierto)				
Letra	Euclidean	Maximum	Sum	Total
A	131,15	9,78	268,96	250,60
B	523,94	480,65	501,95	522,88
C	551,53	571,82	526,13	568,79
D	271,78	90,39	139,70	249,20

Distancia Mínima (índices de acierto)				
Letra	Euclidean	Maximum	Sum	Total
E	488,61	501,57	483,58	490,37
F	423,68	162,27	544,26	423,64
G	465,72	345,86	573,81	536,40
H	365,32	136,24	474,99	363,07
I	159,20	0,00	274,31	249,97
K	695,74	720,57	563,06	718,39
L	174,26	232,05	143,23	249,82
M	309,14	541,28	115,52	328,65
N	409,25	165,58	466,64	412,20
O	577,84	658,29	561,61	659,67
P	526,90	618,99	358,43	611,56
Q	18,81	0,00	80,05	250,12
R	636,73	691,81	578,10	679,28
S	39,28	61,15	90,52	250,16
T	148,41	51,79	129,24	249,76
U	456,32	163,63	587,29	450,54
V	99,44	269,52	188,40	250,84
W	663,79	706,68	671,75	700,97
X	467,67	454,46	491,72	468,21
Y	539,56	580,33	479,30	577,18
PROMEDIO	381,00	342,28	387,19	438,01

Tabla 4.6 Índices de acierto para Distancia Mínima (Usuario 2).

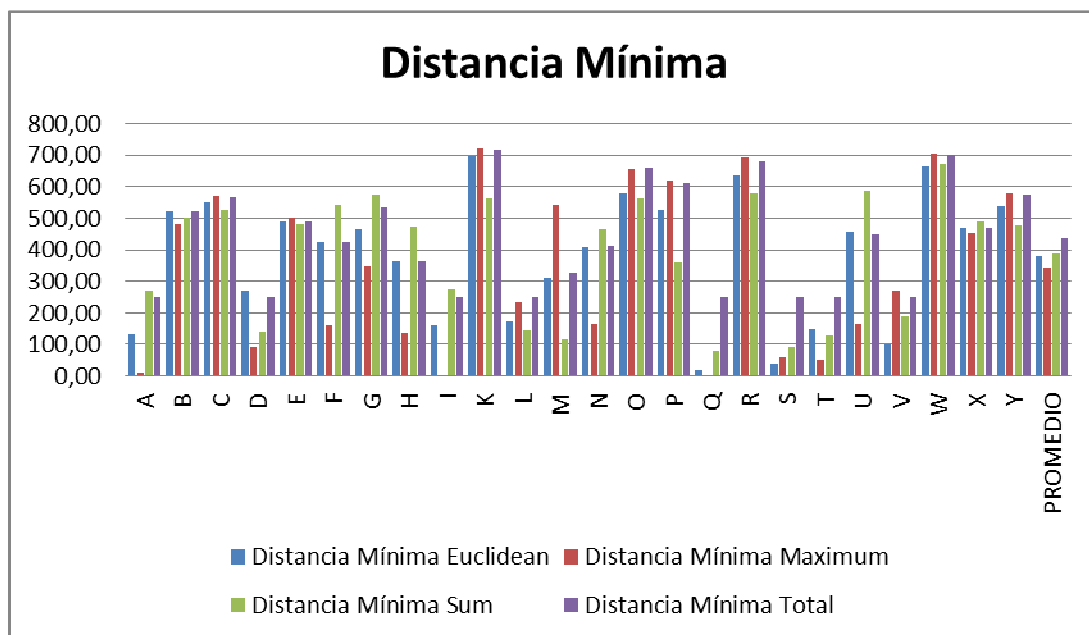


Figura 4.3 Distribución de las distancias para Distancia Mínima.

Del análisis de la Tabla 4.5 y la Tabla 4.6 se puede concluir que con este método se reconoce y asigna correctamente la clase a cada una de las letras, pero el índice de acierto es muy bajo, lo que ocasiona que se introduzcan falsas clasificaciones, mientras se cambia de un símbolo a otro.

4.2.4 COMPARACIÓN ENTRE MÉTODOS

Para realizar la comparación entre los tres métodos se tomaron los promedios de clasificación obtenidos anteriormente.

Métodos (índices de acierto)			
Letra	Vecino cercano	k-Vecino más Cercano	Mínima distancia
A	375,4	749,94	750,12
B	750,42	749,94	250,07
C	249,58	749,94	293,33
D	554,39	749,94	746,43
E	722,36	500,05	250,1
F	650,71	749,94	505,93
G	250,1	749,94	373,9
H	249,05	749,94	250,07
I	749,96	749,94	749,87
K	250,7	749,94	No reconoce
L	628,58	749,94	458,64
M	250,3	749,94	250,79
N	284,01	749,94	270,16
O	709,52	749,94	249,03
P	591,72	749,94	233,2
Q	No reconoce	749,94	249,03
R	431,69	749,94	551,22
S	657,08	749,94	750,18
T	750,3	749,94	250,74
U	749,88	749,94	750,17
V	No reconoce	749,94	249,4
W	750,31	500,05	738,22
X	735,93	749,94	288,08
Y	750,54	749,94	717,7
PROMEDIO	549,66	729,11	442,45

Tabla 4.7 Valores de clasificación para cada método (Usuario 1).

Métodos (índices de acierto)			
Letra	Vecino cercano	k-Vecino más cercano	Mínima distancia
A	517,18	749,94	525,01
B	750,1	749,94	250,22
C	250,87	749,94	558,15
D	250,4	749,94	250,02
E	250,19	749,94	355,49
F	749,9	749,94	644,96
G	440,06	749,94	250
H	433,56	749,94	390,47
I	377,56	749,94	750,32
K	276,39	500,05	249,13
L	250,01	749,94	374,77
M	No reconoce	749,94	No reconoce
N	250,27	749,94	250,09
O	749,87	749,94	250,01
P	439,84	749,94	627,3
Q	No reconoce	749,94	248,32
R	750,49	749,94	501,02
S	422,65	749,94	250,04
T	437,92	300,03	No reconoce
U	402,21	749,94	750,55
V	750,25	749,94	340,44
W	396,11	749,94	750,05
X	636,77	749,94	388,4
Y	704,14	749,94	572,66
PROMEDIO	476,67	720,78	433,06

Tabla 4.8 Valores de clasificación para cada método (Usuario 2).

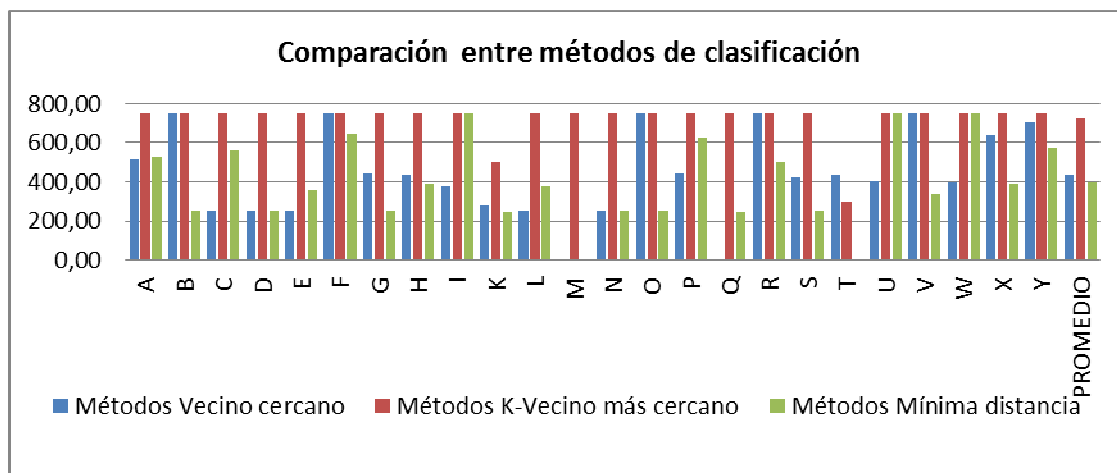


Figura 4.4 Distribución de barras de los tres métodos.

Para la selección del método se consideró dos factores: el reconocimiento de la letra y el índice de acierto de la clasificación, para esto se analizó los datos de la Tabla 4.7 y de la Tabla 4.8. Por esta razón se concluye que el mejor método para esta aplicación es el *k*-vecino más cercano, porque el promedio del porcentaje de acierto es mayor que los otros dos métodos, además que identifica correctamente cada una de las letras.

4.3 PRUEBAS DEL FUNCIONAMIENTO DEL TRADUCTOR

Para probar el funcionamiento del traductor se hicieron varias veces las señales correspondientes al texto “**ESCUELA POLITECNICA NACIONAL**”. De los resultados obtenidos se calculó el error teniendo en cuenta la cantidad de letras en la frase, como la frase está compuesta de 28 letras el error por letra es del 3,57%. Estas pruebas fueron realizadas por dos usuarios: Usuario A que tiene conocimiento del lenguaje de señas y el Usuario B que tiene un limitado conocimiento de este lenguaje. Los resultados que se obtuvieron se presentan a continuación.

RESULTADO DE LA PRUEBA	ERROR	NOMBRE
<p><i>Texto Formado</i></p> <p>ESCUELA POLITECNICA NACIONAML</p>	Prueba 1 Error 3,57 %	USUARIO A
<p><i>Texto Formado</i></p> <p>ESWCUELA POLITECNICA NACIONAL</p>	Prueba 2 Error 3,57%	
<p><i>Texto Formado</i></p> <p>ESCUELA POLITECNICA NACIONAL</p>	Prueba 3 Error 0%	
<p><i>Texto Formado</i></p> <p>ESWCUELA POLITEWCTNIWCA NSACIONAL</p>	Prueba 1 Error 17,85%	USUARIO B
<p><i>Texto Formado</i></p> <p>ESCUELA CPOLITECNMXIXCA NAWCIONAL</p>	Prueba 2 Error 17,85%	
<p><i>Texto Formado</i></p> <p>ESCUEMPLA POLITECNICA NACAIONAL</p>	Prueba 3 Error 7,14%	

Tabla 4.9 Cálculo de errores en el traductor.

Con esta prueba se concluye que el funcionamiento del traductor depende mucho de la habilidad y la velocidad que el usuario tenga en el uso del lenguaje de señas. Como se observa en la Tabla 4.9 a medida que el usuario se familiariza con el uso de la aplicación obtiene mejores resultados en su traducción.

4.4 PRUEBAS DEL FUNCIONAMIENTO DEL ENTRENADOR

El entrenador se probó con tres usuarios diferentes que no tenían ningún conocimiento previo del lenguaje de señas. Los resultados que se obtuvieron se detallan a continuación:

USUARIO	RESULTADOS
Niña de 13 años	Fallo letras "M" y "R"
Señorita 1 de 17 años	Fallo letra "S"
Señorita 2 de 17 años	Fallo letras "H" y "S"

Tabla 4.10 Errores en el entrenador.

De estos resultados se puede concluir que esta herramienta es de utilidad para aquellos usuarios que no tienen conocimiento del lenguaje de señas, ayudándoles en el aprendizaje de este lenguaje.

4.5 COMPARACIÓN CON TRABAJOS SIMILARES

Al realizar una investigación sobre el tema, se encontró dos trabajos que usan visión artificial para el reconocimiento del lenguaje de señas: (1) el sistema que se realizó en la Universidad Autónoma de Madrid que tiene un error del 14.6% [19]; y, (2) el sistema que se realizó en la Universidad de Barcelona que tiene un error del 31.25% [20]. Al comparar el error del sistema que es del 17.85 % se observa que está dentro de los límites definidos por los trabajos anteriores.

En este capítulo se realizaron las pruebas a cada una de las partes del sistema terminado, con la finalidad de justificar el uso de cada una de estas partes, así como también de los métodos de reconocimiento empleados en esta aplicación. Con estas pruebas se concluye que el sistema funciona correctamente logrando cumplir los objetivos propuestos al principio del proyecto.

CAPÍTULO 5

CONCLUSIONES Y RECOMENDACIONES

5.1 INTRODUCCIÓN

Una vez desarrollado el hardware del proyecto que incluye todos los elementos de un sistema de visión artificial y el software del sistema traductor de señales de manos mediante visión artificial, y de los resultados de las pruebas realizadas es posible extraer las siguientes conclusiones.

5.2 CONCLUSIONES

- Un sistema de iluminación correctamente seleccionado, reduce altamente el proceso de análisis de la imagen ya que una imagen es el reflejo de la luz. De esto se obtiene un mejor resultado dentro de los límites de la aplicación reduciendo la complejidad del sistema y su tiempo de ejecución.
- El uso de Labview para el desarrollo del software permitió realizar una programación sencilla y ordenada, además que se pudo realizar una interfaz de fácil uso y amigable con el usuario.
- Del buen desempeño del algoritmo de preprocesamiento de la imagen depende el proceso de clasificación, porque el preprocesamiento resalta las características de la imagen que el clasificador necesita.
- Realizar el proceso de segmentación morfológica de la imagen antes de ingresarla en la base de datos, permite obtener vectores de características definidos en cada clase, lo cual evita errores al momento de la clasificación.
- Labview es un programa muy completo ya que su toolkit de Visión es una herramienta que contiene todas las operaciones en cuanto al procesamiento de imágenes se refiere; esto permitió desarrollar totalmente la aplicación sin necesidad de software adicional.

- Del conocimiento y habilidad que el usuario tiene para realizar los símbolos, depende el reconocimiento y, por tanto la asignación de la clase con un porcentaje de error bajo.
- Como el usuario requiere facilidad y comodidad a la hora de usar la aplicación, se construyó un hardware que ofrezca facilidades de adaptación a diferentes usuarios, además que es fácilmente transportable y montable.
- Se investigaron los métodos de clasificación de Vecino más Cercano, *k*-Vecino más Cercano y Mínima Distancia; de los cuales se concluye: para aplicaciones donde existe ruido el método el vecino más cercano es poco efectivo, en aplicaciones con mucho ruido y cuyos patrones tiene variaciones muy pequeñas entre si es mejor usar el método de la mínima distancia. Por lo tanto para esta aplicación con presencia de ruido y cuyos patrones en su mayoría están bien diferenciados se usa el método del *k*-Vecino más Cercano.

Después de terminado este trabajo y de la experiencia adquirida es posible dar las recomendaciones siguientes:

5.3 RECOMENDACIONES

- Para poder añadir a esta aplicación las letras que poseen movimiento se podría usar el módulo RIO de la National Instrument, el cual permite trabajar el procesamiento de imágenes en tiempo real y tiene la capacidad de usar cámaras más rápidas como las Smart, lo que permitiría usar un algoritmo para el reconocimiento de trayectorias en tiempo real.
- El reconocimiento de formas se puede usar en otros procesos en los cuales se necesite reconocer formas definidas. Esto se lograría creando bases de datos con las formas que se desee reconocer, para lograr una automatización en estos procesos.
- Para empezar a desarrollar una aplicación similar se recomienda estudiar los métodos de clasificación de imágenes existentes, sus ventajas y desventajas, para seleccionar el método correcto para la aplicación que se desee realizar.

- Al momento de escoger el ambiente donde se va a realizar un sistema de visión artificial se debe tener en cuenta: los factores externos e internos que afectan al sistema, por ejemplo, la velocidad de la aplicación para diseñar correctamente cada uno de los elementos del sistema de visión artificial.
- En un trabajo futuro para mejorar el rendimiento del traductor, se recomienda añadir un algoritmo de predicción de texto para aumentar la velocidad de la aplicación.
- Para obtener mejores resultados en esta aplicación, se deben crear las bases de datos dentro del Vision Assistant en base a un preprocesamiento y procesamiento dentro del mismo paquete, porque así se resalta la morfología de la mano, permitiendo crear vectores de características con valores más definidos. Al emplear estas bases de datos en el reconocimiento se logra altos índices de acierto, lo que a su vez favorece a la asignación de clase con menos porcentaje de error (confusión entre letras).
- Al usar el método del k -Vecino más Cercano se recomienda poner un número de k menor al número de muestras de cada clase, de esta forma se logra que el sistema compare las k muestras más parecidas a la imagen que adquiere la cámara. Por lo tanto asigna una clase con alto porcentaje de acierto reduciendo así el error.
- Un factor importante al momento de elegir la cámara es si esta tiene o no la función del autocontraste debido a que esta función no permite al sistema funcionar correctamente, porque varía la intensidad de los píxeles. Si la cámara posee esta característica, se la debe deshabilitar para que este parámetro no afecte al sistema.
- Para el sistema de adquisición de imagen se usó una cámara web de 30fps que, por sí misma introduce un retardo en la adquisición limitando así la adquisición en tiempo real, además que el procesamiento de cada imagen retarda aún más la respuesta del sistema. Para mejorar esta latencia en la

aplicación se recomienda usar cámaras especializadas en visión artificial que adquieren imágenes a mayor velocidad.

- Se recomienda continuar mejorando este trabajo y herramienta hasta que se lo pueda ofrecer a la comunidad y así contribuir a la solución de problemas que afecten a personas que necesitan ayuda, y no cuentan con soluciones a precios razonables.

REFERENCIAS BIBLIOGRAFICAS

- [1] Consejo Nacional de Discapacidades, (2011) CONADIS, [Online], Disponible:
<http://www.conadis.gob.ec/estadisticas.htm>
- [2] Sitio de Sordos, (2011) [Online], Disponible:
http://www.sitiodesordos.com.ar/alfabeto_ecua.htm
- [3] Wikipedia, (2010) [Online], Disponible:
<http://es.wikipedia.org/wiki/Dactilolog%C3%ADa>
- [4] F. Ugarte, (2011) Infociegos, [Online], Disponible:
http://www.infociegos.com/espanol/index.php?option=com_content&task=view&id=29&Itemid=57#3
- [5] Wikipedia, (2010) [Online], Disponible:
http://es.wikipedia.org/wiki/Visi%C3%B3n_artificial
- [6] P. AREVALO, Presentación curso “Visión Artificial”, Viaenterprise. Ecuador, 2010
- [7] U. Salesiana, (2007) Dspace, [Online], Disponible:
<http://dspace.ups.edu.ec/bitstream/123456789/220/2/Capitulo%201.pdf>
- [8] U. Quilmes, (2005) “Aspectos de un sistema de visión artificial”, [Online] Disponible:
<http://iaci.unq.edu.ar/materias/vision/archivos/apuntes/Aspectos%20de%20un%20Proyecto%20de%20Visi%C3%B3n%20Artificial.pdf>
- [9] N. Fernández, (2011) Slideshare, [Online], Disponible:
<http://www.slideshare.net/maixu61/caractersticas-de-la-imagen-digital>
- [10] O. James, (2010) tutorial9, [Online], Disponible:
<http://www.tutorial9.net/articles/design/simple-practical-color-Theory/>
- [11] Google doc, (2010) “Fundamentos de color” [Online], Disponible:

<https://docs.google.com/fileview?id=0B8U6WB->

[Bi2HYWZIMjVjOTYtYTIzOS00ODIxLWI0N2EtMTk1NTZhODgzNTgw&hl=es&pli=1](https://docs.google.com/fileview?id=0B8U6WB-Bi2HYWZIMjVjOTYtYTIzOS00ODIxLWI0N2EtMTk1NTZhODgzNTgw&hl=es&pli=1)

[12] Wikipedia, (2011) [Online], Disponible:

http://es.wikipedia.org/wiki/Procesamiento_digital_de_im%C3%A1genes

[13] J. Velez, (2003) Escet, [Online]. Disponible:

<http://www.escet.urjc.es/~visionc/VisionPorComputador.pdf>

[14] A. Gonzales, F. Martínez, A. Pernia, F. Alba, M. Castejón, J. Ordieres, y E. Vergara, “Técnicas y algoritmos básicos de visión artificial”, 1ra ed., Universidad de la Rioja Ed. Madrid: 2006,

[15] National Instrument “Vision Concepts Help” Junio 2009.

[16] Wikipedia, (2011) [Online], Disponible: <http://es.wikipedia.org/wiki/Knn>

[17] Weblogs. SL, (2011) Xatakafoto, [Online]. Disponible:

<http://www.xatakafoto.com/camaras/sensores-con-tecnologia-ccd-vs-cmos>

[18] National Instrument “Vision Assitant ” Junio 2009.

[19] J. Molina, “Reconocimiento de gestos mediante técnicas de visión artificial”, Universidad Autónoma de Madrid, Madrid, 1998.

[20] E.López, “Reconocimiento automático de lenguaje de signos: Lenguaje ASL”, Universidad de Barcelona, Barcelona, Julio-2009.

[21] Compunoa, (2011) Compunoa, [Online]. Disponible:

<http://www.compunoa.com/webcam-genius-messenger-p-562.html>

ANEXO A
MANUAL DE USUARIO
TRADUCTOR DE SEÑALES DE MANOS A TEXTO
MEDIANTE VISIÓN ARTIFICIAL EN UN AMBIENTE
CONTROLADO



Figura A.1 Sistema Traductor de Señales de Manos a Texto.

A.1 DESCRIPCION GENERAL

Mediante una cámara se captan las imágenes de señales de mano, que luego pasan por las etapas de pre-procesamiento, procesamiento y clasificación. Para finalmente llegar a formar el texto. Se tiene además la opción de un entrenador para que el usuario adquiera la habilidad necesaria para usar la aplicación.

Este sistema consta de un fondo de color negro, el trípode, la cámara y la luminaria que corresponden al hardware y el software que consiste en el programa desarrollado en Labview 2009 incluyendo los Toolkits de: Visión Artificial, Fuzzy Logic y Report Generation.

A.2 CONSIDERACIONES GENERALES

El tamaño de la imagen que toma la cámara debe ser de 640*480.

Antes de iniciar con esta aplicación es importante eliminar la función de autocontraste que posee la cámara, si fuera este el caso.

Asegurarse que el fondo sea oscuro, de tal manera que contraste con la mano, mejorando de esta forma la calidad de la imagen.

Ajustar la altura del trípode de acuerdo al usuario de forma que el usuario pueda realizar las señales fácilmente.

Cuidar que la mano este encuadrada en la imagen, para lo que se debe alejar o acercar el trípode. Se obtiene mejores resultados calibrando con la mano abierta.

Dependiendo de las condiciones ambientales, calibrar el brillo y contraste de la imagen, de tal manera que se forme la menor cantidad de ruido (sombras en la mano y su alrededor).

A.3 INSTALACIÓN

Colocar el fondo oscuro a una altura conveniente para el usuario.

Posicionar el conjunto de la cámara y la luminaria en su lugar correspondiente sobre el trípode.

Calibrar la altura del trípode de acuerdo al usuario.

Conectar la lámpara a la alimentación de 110V y la cámara al puerto USB del computador e instalar el driver correspondiente.

Arrancar el programa desarrollado para la aplicación.

A.4 OPERACIÓN

Correr el programa desde el VI Principal. En la pantalla principal seleccionar la opción: traductor o entrenador según se requiera.



Figura A.2 Pantalla principal, selección de opciones (entrenador o traductor).

Luego de haber seleccionado una opción, se debe ajustar la distancia de la cámara a la mano. Esto se hace abriendo la mano completamente y observando que en la pantalla del programa, donde se ve la imagen que adquiere la cámara, este la mano completa abarcando todo el cuadro.



Figura A.3 Calibración de la distancia de la cámara a la mano.

A.5 Entrenador

Esta opción permite al usuario aprender el signo que corresponde a cada letra. Primero se debe seleccionar el dispositivo (cámara), luego ajustar los parámetros de brillo, contraste y a fin de mejorar la imagen evitando ruido, seleccionar una opción dentro del menú *Operation* y ajustar con el menú de *Número de operación*. Se puede operar de dos formas: automático y manual.

En el modo manual se selecciona la letra que se quiere aprender y en el modo automático el programa genera una letra aleatoriamente, en ambos casos se emite una señal (led) que indica que el signo fue hecho correctamente. En caso de que el signo no se logre realizar adecuadamente es posible cambiar a una nueva letra, esto ocurre en ambos modos.



Figura A.4 Ventana del Entrenador.

A.6 Traductor

En esta opción se traduce las señales de manos a texto, para cumplir esta función contiene varios menús en los que se configuran los parámetros de acuerdo al ambiente de trabajo y al usuario para que la aplicación funcione correctamente. Estos menús son:

Adquisición

En esta pestaña se hace la selección de la cámara y se observa la imagen que la cámara está adquiriendo.

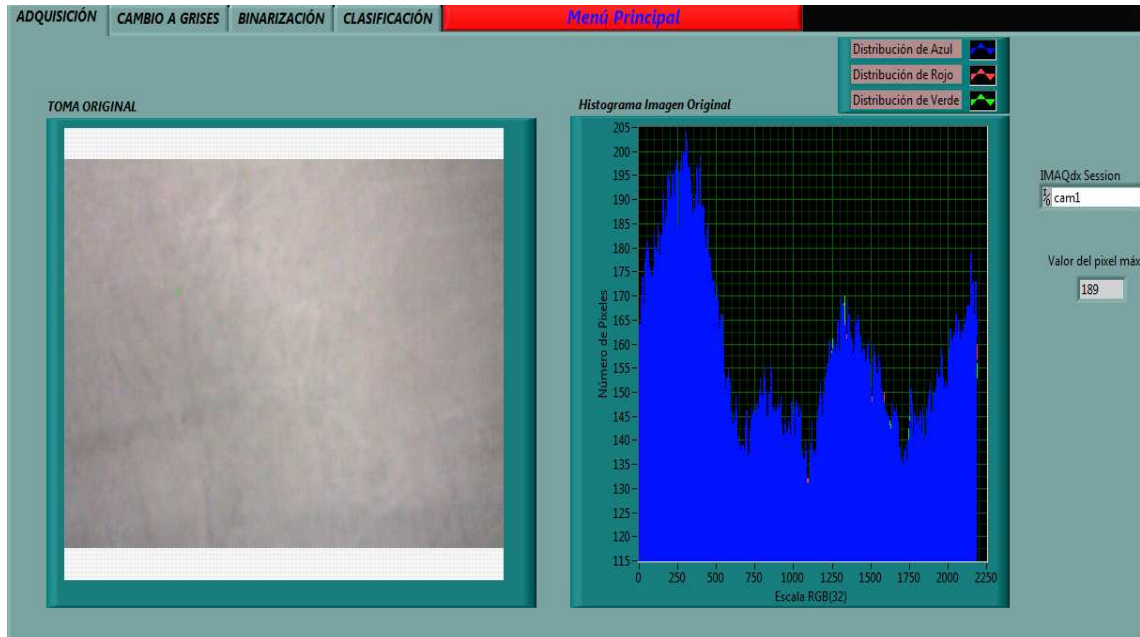


Figura A.5 Pestaña de adquisición.

Cambio a grises

En esta pestaña se observa la imagen transformada a escala de grises y se tiene una ventana para variar parámetros del contraste de la imagen (brillo, contraste, gamma) para resaltar las características del objeto de interés.

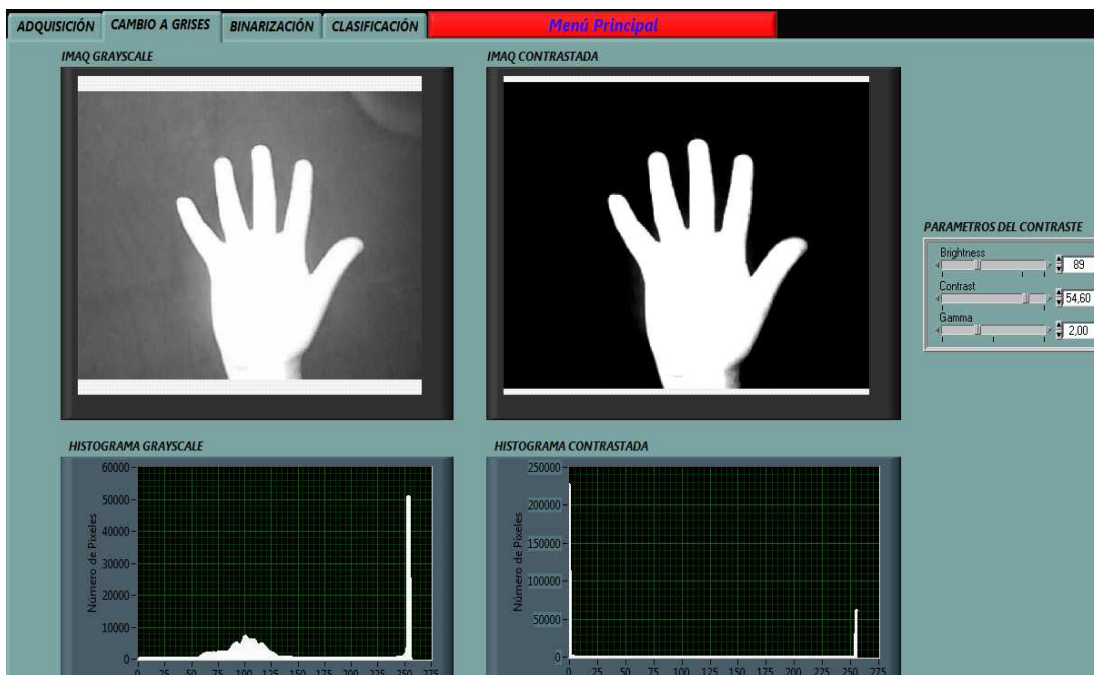


Figura A.6 Pestaña de cambio a escala de grises.

Binarización

Con esta opción se observa la imagen en blanco y negro (0 y 1), y se puede aplicar a la imagen una operación morfológica para eliminar sombras (ruido) y facilitar el reconocimiento.

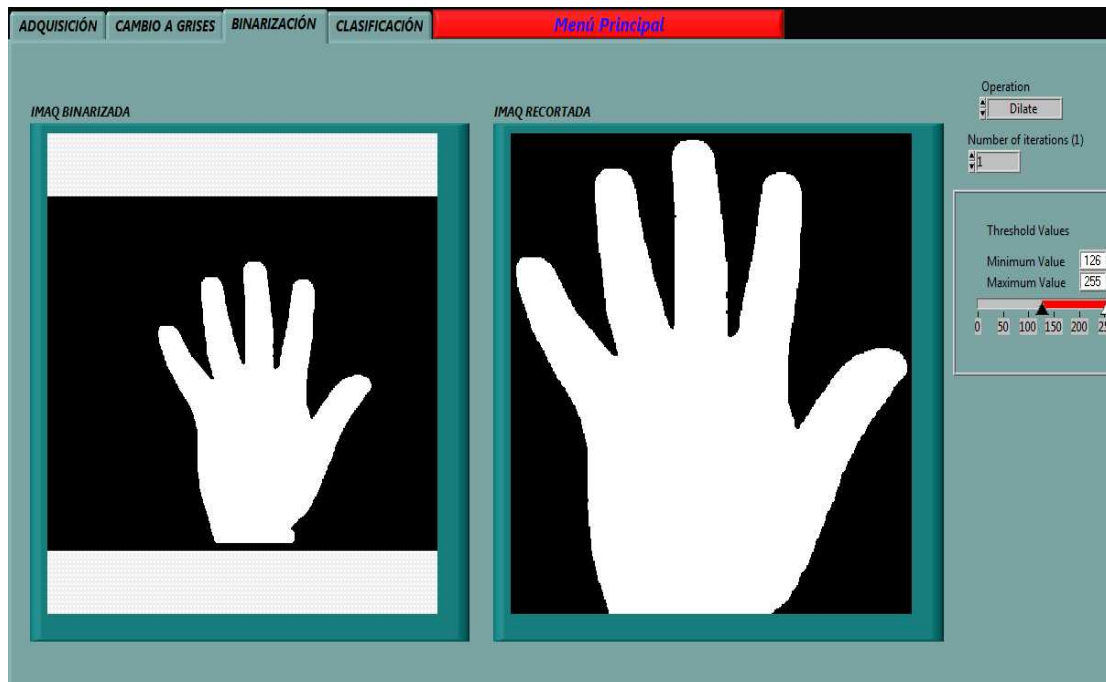


Figura A.7 Pestaña de binarización.

Clasificación

Por último, en este menú se visualiza la clase (letra) que se asigna al símbolo que está realizando, es posible configurar el retardo de adquisición en milisegundos y el tiempo de espera en milisegundos que debe realizarse cada señal para ser aceptada.

El retardo se usa para hacer más rápida o lenta la ejecución del programa, esta velocidad depende de la habilidad y rapidez del usuario al hacer los signos. En cambio el tiempo de espera sirve para evitar que el programa acepte como señales correctas a los signos que se van formando mientras se cambia de un símbolo a otro.



Figura A.8 Pestaña de clasificación.

ANEXO B

HARDWARE

B.1 CÁMARA MESSENGER 310

La cámara tiene las siguientes características:

- Micrófono incorporado
- Multifunción Clipping Base Design
- Giratorio 360 grados
- Una mayor flexibilidad de visualización
- Sensor de imagen tipo 300k píxeles CMOS sensor de imagen
- Lente de enfoque manual
- Interfaz Compatible con USB1.1/1.0
- Formato de imagenes JPEG / WMV
- Resolución de video 640 x 480 píxeles / hasta 30fps
- Compatible con Intel Pentium III 800MHz MMX o más rápido
- Compatible con Microsoft Windows Vista, XP, 2000, Me, 98SE o superior
- Espacio de memoria requerido 128 MB de RAM
- Disponible USB 1.1 / 1.0
- 60 MB de espacio libre en disco duro (260 MB recomendado)
- DirectX 9,0 arriba

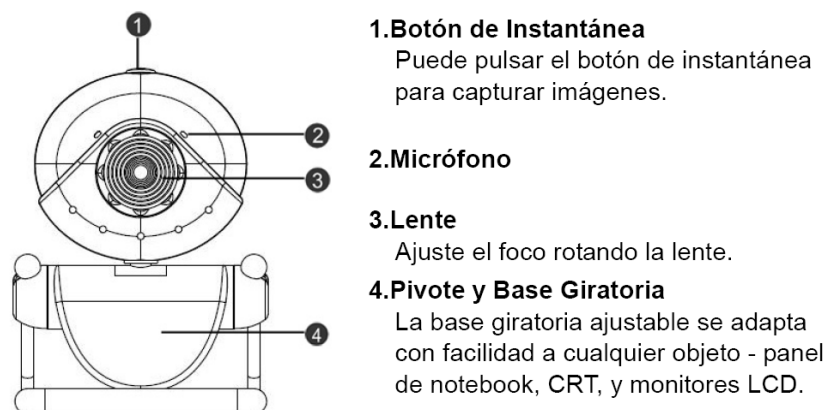


Figura B.1 Partes de Messenger 310.

B.2 TRÍPODE VIVITAR VPT-2457

El Trípode Profesional Vivitar 57" cuenta con un cabezal de 3 vías, un nivel de burbuja, pies de goma ajustable, gancho de accesorios, y montura de rápida liberación. Con una columna central con una llave de cerradura, junto con tres cerraduras de sección del canal en cada pierna del trípode.

Características:

- Diámetro de la pierna : 20mm
- Altura máxima: 57"
- Altura del trípode plegado 21.45"
- Dimensiones: 4.00"x23.75"x3.75"
- Peso: 1.90lbs



Figura B.2 Trípode VIVITAR VPT-2457.

B.3 LUMINARIA Y LÁMPARA

Lámpara



Figura B.3 Luminaria y foco

Características

- Marca: BASCO
- Datos eléctricos: 120vac/11w//60Hz
- Duración: 8000 horas

B.4 DETALLE DE COSTOS

Con la finalidad de dar un valor al trabajo realizado en este proyecto y los dispositivos utilizados, a continuación se presentan los detalles de costos del proyecto.

La realización del proyecto duro nueve meses: los tres primeros meses se dedicaron a la investigación de soluciones y los seis meses restantes a la implementación de la solución. Se trabajó solo en días laborables a razón de 5 horas diarias; por tal motivo se establece un costo de ingeniería de 1200 dólares americanos.

Detalle de costos			
Descripción	Cantidad	Precio unitario	Subtotal
Cámara web	1	15	15
Trípode	1	40	40
Lámpara y foco	1	25	25
Varios		20	20

Detalle de costos			
Descripción	Cantidad	Precio unitario	Subtotal
Ingeniería		1200	1200
	Total	1300	1300

Tabla B.1 Detalle de costos del proyecto.

Se debe indicar que para la realización de este proyecto no se compró la licencia de Labview, debido a que ésta fue proporcionada por la Escuela Politécnica Nacional.

El detalle de costos esta actualizado al presente año y se realizó en dólares.

ANEXO C

C.1 SUB-VIS REALIZADOS EN LABVIEW

En esta parte se explicara el funcionamiento, las entradas y salidas de cada uno de los Sub-VI que se usaron dentro del programa principal. Estos VI fueron creados dentro de la interfaz de Labview 2009.

C.1.1 Detector2.vi

En este VI se realiza el monitoreo de la existencia o no de la mano dentro del lugar de trabajo. Tiene como entrada la imagen fuente a color, y proporciona tres salidas: la imagen de entrada, una gráfica de la distribución de los pixeles y el valor máximo de los pixeles. Este último valor se usa como comparación para saber si existe o no existe la mano dentro del área de trabajo.

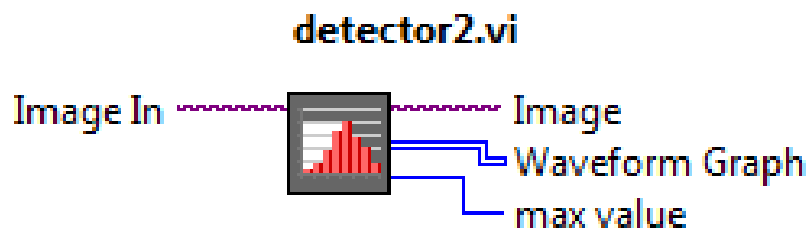


Figura C.1 Detector2.vi

C.1.2 Coordenadas-copia.vi

Este VI tiene la función de cambiar la imagen de entrada a escala de grises, mejorar el contraste, binarizar la imagen, aplicar las operaciones morfológicas y filtros morfológicos, y detectar los bordes del objeto de interés para después realizar el cortado de dicho objeto del resto de la imagen. Tiene como entrada la imagen fuente, la operación morfológica que se desee realizar y el número de veces que se desea realizar esta operación. Como salida tiene la imagen en escala de grises, la imagen contrastada, la imagen binaria, la imagen recortada, los puntos donde se encuentra el rectángulo de corte, y el rango en la escala de grises donde se encuentra el objeto de interés.

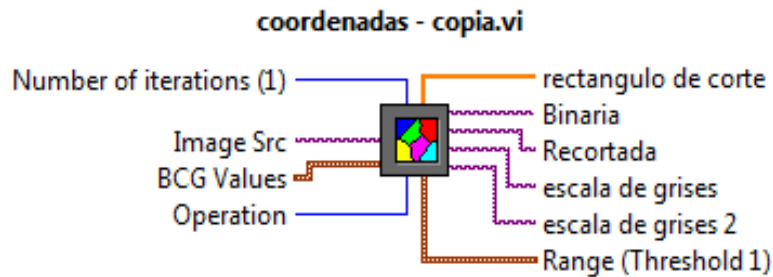


Figura C.2 Coordenadas.copia.vi

C.1.3 Clasificacionsubvi.vi

Este VI es el que realiza la clasificación de la imagen de entrada por el método escogido, dando como respuesta la clase a la que pertenece y el índice de acierto.

Tiene como entrada la imagen recortada, y como salida tiene: la clase, el índice de acierto, y el resultado de cada una de las variantes del método de clasificación. Internamente el VI supervisa la clasificación y proporciona una salida booleana que es la que indica si la clasificación se acepta o se rechaza.

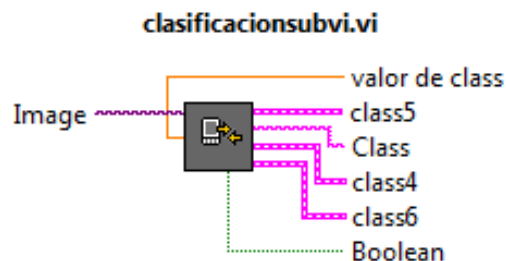


Figura C.3 Clasificacionsubvi.vi.

C.1.4 Filtro de cambio.vi

Este VI tiene la función de filtrar las falsas clasificaciones que resultan entre el cambio de una señal de la mano a otra. Para esto el VI saca un promedio de las imágenes de entrada en un lapso de tiempo y toma la decisión de aceptar o no la imagen clasificada. Tiene como entrada la imagen clasificada y el número de muestras que se desea tomar para realizar el promedio y una salida booleana que indica si se acepta o no la imagen clasificada.

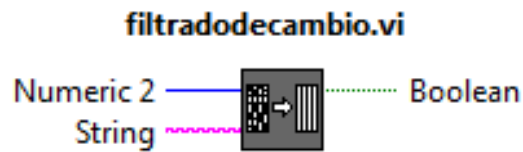


Figura C.4 Filtrodecambio.vi.

C.1.5 Error de repetición.vi

Este VI realiza la función de unir la frase letra a letra evitando los errores de repetición, además realiza la función de invertir el texto porque la primera letra que entra debe ser la última. Este vi acepta una letra válida siempre y cuando ésta sea diferente a la anterior. Tiene como entrada la letra de entrada y una entrada booleana que sirve para borrar el texto y una salida que es el texto corregido.

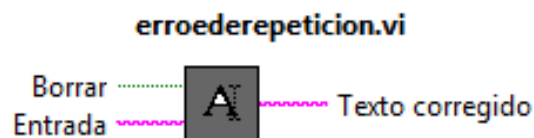


Figura C.5 Errorderepeticion.vi.

C.1.6 Text2Speech.vi

Este VI reproduce en audio el texto ensamblado, para esto tiene como entrada la dirección del ActiveX donde está el reproductor de voz de Windows, la entrada donde se selecciona la voz con la que se desea reproducir el texto y la entrada del texto que se desea reproducir. Tiene una salida virtual que es el audio reproducido que se escucha en los altavoces de la computadora y la copia de la dirección donde se encuentra el ActiveX.

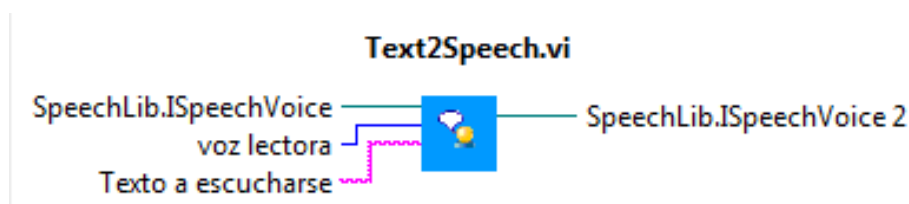


Figura C.6 Text2Speech.vi.

C.1.7 Reportaaword.vi

Este VI tiene como función la de enviar el texto traducido a un documento de Word con el objetivo de que el usuario revise la traducción y la imprima, o sea enviada por email según la necesidad del usuario. Tiene como entrada el texto de la traducción y una entrada booleana. Dependiendo del estado de la entrada booleana se realiza una operación, *false* no realiza ninguna operación y *true* envía el texto a Word. Tiene una salida virtual que es el documento de Word con la traducción, la fecha de la traducción y los nombres de los autores del proyecto.

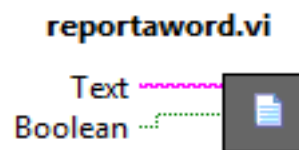


Figura C.7 Reportword.vi.

C.1.8 Logic.vi

Este VI tiene como función analizar los resultados de la clasificación del método seleccionado y realizar un análisis de estos valores y proporcionar un valor adecuado a la salida. Tiene tres entradas que son los valores resultantes de los cálculos de las distancias y una salida que es el valor del análisis de los valores de entrada. Este VI se usa cuando los tres resultados de clasificación son los mismos.

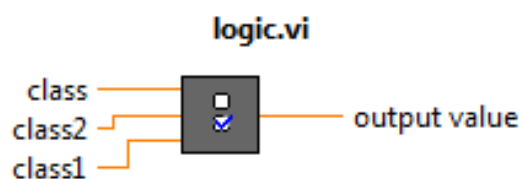


Figura C.8 Logic.vi.

C.1.9 Fuzz2.vi

Este Vi tiene como objetivo realizar el análisis de los valores de la clasificación del método seleccionado y realizar un análisis de estos valores y proporcionar un valor analizado a la salida. Tiene dos entradas que son los cálculos de las distancias de las clasificaciones y una salida que es el resultado del análisis de las entradas. Este VI se usa cuando dos de las tres clasificaciones son iguales.



Figura C.9 Fuzz2.vi

C.1.10 Open Word.vi

Este VI tiene la función de abrir un archivo de Word de una ubicación específica. Tiene una entrada booleana que permite ejecutar o no este VI, tiene una salida virtual que es el documento de Word.

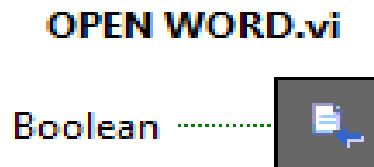


Figura C.10 Open Word.vi

C.2 Subvis realizados en Visión Assistant

También se crearon Sub'Vis utilizando la herramienta Visión Assistant. Estos VI se crearon dentro de esta aplicación por el elevado grado de dificultad que conlleva realizar la función que realizan.

Para la creación de estos VI's primero se realiza un script dentro de Visión Assistant y luego se los pasa a Labview para su uso. Cuando el programa principal utiliza estos Vis, el programa llama a la herramienta Visión Assitant dentro de Labview.

C.2.1 Detector de pixeles en los márgenes del área de trabajo

Este VI realiza el análisis de los pixeles existentes dentro del lugar del trabajo y proporciona el histograma de los pixeles sobre las líneas trazadas.

Tiene como entrada la imagen fuente, la imagen destino, la región de interés y una entrada de error. Tiene como salida el histograma, la imagen de salida, y el error de salida.

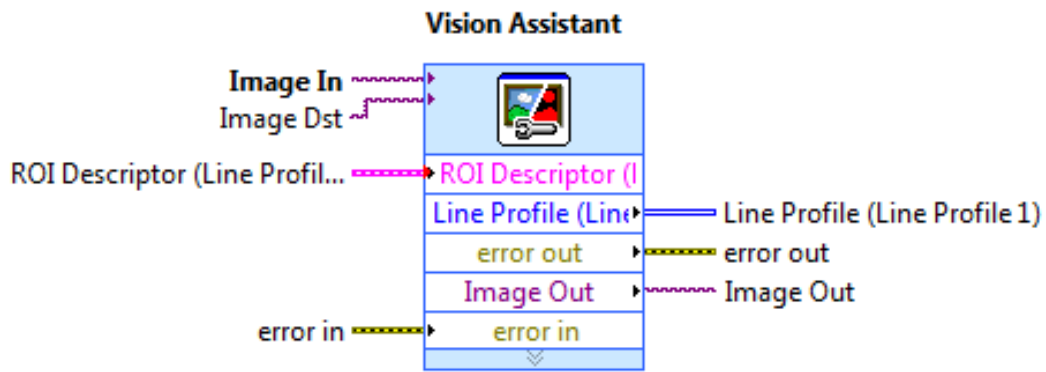


Figura C.10 Detector de pixeles en los bordes 1.0.

C.2.2 Filtro de frecuencia

Este VI tiene como función realizar un filtro pasa bajos de frecuencia a la imagen binarizada, realizando un suavizado a los bordes del objeto de interés. Tiene como entrada la imagen fuente, la imagen destino, la frecuencia de corte y el error de entrada. Como salida tiene la imagen respuesta y el error de la salida.

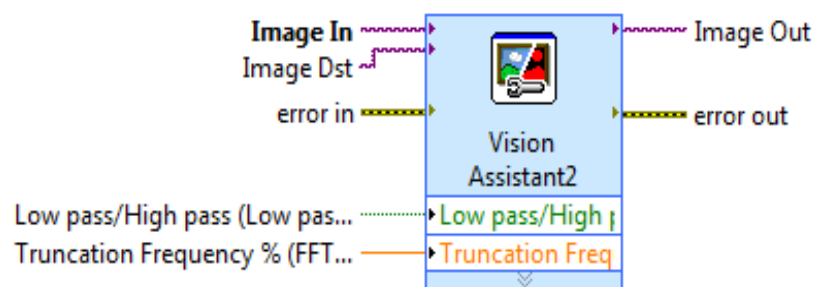


Figura C.11 Filtro pasa bajos.

C.2.3 Detector de flancos

Este VI tiene como función detectar los flancos del objeto de interés, para luego calcular los puntos donde se encuentra este objeto y proporcionar la localización de éste dentro de la imagen. Tiene como entrada la imagen fuente, la imagen destino, el error de entrada y las regiones de interés donde se desee realizar la búsqueda del objeto de interés, como salida tiene la imagen de salida, las coordenadas de los flancos encontrados, el centroide del objeto encontrado y el error de salida.

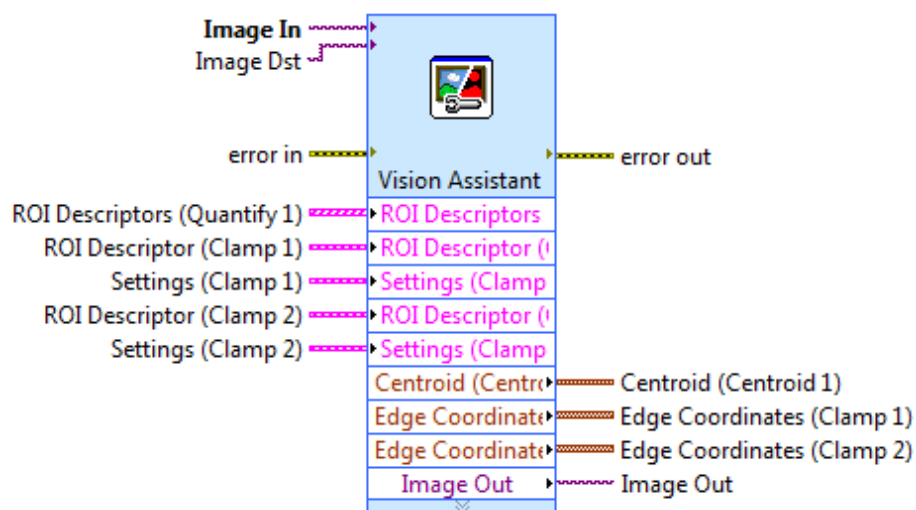


Figura C.12 Detector de Flancos.