



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

**DEPARTAMENTO DE CIENCIAS DE LA ENERGÍA Y
MECÁNICA**

CARRERA DE INGENIERÍA EN MECATRÓNICA

**TRABAJO DE TITULACIÓN, PREVIO A LA OBTENCIÓN DEL
TÍTULO DE INGENIERO MECATRÓNICO**

**TEMA: INVESTIGACIÓN DE UN SISTEMA DE VISIÓN
ARTIFICIAL PARA LA INTERPRETACIÓN DEL ALFABETO Y
NÚMEROS DEL UNO AL DIEZ EN LENGUAJE DE SEÑAS
ECUATORIANA MEDIANTE LA UTILIZACIÓN DE SOFTWARE
LIBRE.**

AUTOR: JAVIER PATRICIO PICHUCHO RODRÍGUEZ

DIRECTOR: ING. PATRICIA CONSTANTE PRÓCEL, M.SC.

LATACUNGA

2017



**DEPARTAMENTO DE CIENCIAS DE LA ENERGÍA Y MECÁNICA
CARRERA DE INGENIERÍA EN MECATRÓNICA**

CERTIFICACIÓN

Certifico que el trabajo de titulación, “**INVESTIGACIÓN DE UN SISTEMA DE VISIÓN ARTIFICIAL PARA LA INTERPRETACIÓN DEL ALFABETO Y NÚMEROS DEL UNO AL DIEZ EN LENGUAJE DE SEÑAS ECUATORIANA MEDIANTE LA UTILIZACIÓN DE SOFTWARE LIBRE.**” realizado por **JAVIER PATRICIO PICHUCHO RODRÍGUEZ**, ha sido revisado en su totalidad y analizado por el software anti-plagio, el mismo cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, por lo tanto me permito acreditarlo y autorizar al señor **JAVIER PATRICIO PICHUCHO RODRÍGUEZ** para que lo sustente públicamente.

Latacunga, 19 de mayo del 2017

Atentamente,



Ing. Patricia Constante, M.Sc.



**DEPARTAMENTO DE CIENCIAS DE LA ENERGÍA Y MECÁNICA
CARRERA DE INGENIERÍA EN MECATRÓNICA**

AUTORÍA DE RESPONSABILIDAD

Yo, **JAVIER PATRICIO PICHUCHO RODRÍGUEZ**, con cédula de identidad N° 1722341292, declaro que este trabajo de titulación "**INVESTIGACIÓN DE UN SISTEMA DE VISIÓN ARTIFICIAL PARA LA INTERPRETACIÓN DEL ALFABETO Y NÚMEROS DEL UNO AL DIEZ EN LENGUAJE DE SEÑAS ECUATORIANA MEDIANTE LA UTILIZACIÓN DE SOFTWARE LIBRE.**" ha sido desarrollado considerando los métodos de investigación existentes, así como también se ha respetado los derechos intelectuales de terceros considerándose en las citas bibliográficas.

Consecuentemente declaro que este trabajo es de mi autoría, en virtud de ello me declaro responsable del contenido, veracidad y alcance de la investigación mencionada.

Latacunga, 19 de mayo del 2017

Javier Patricio Pichucho Rodríguez

C.C.: 1722341292



**DEPARTAMENTO DE CIENCIAS DE LA ENERGÍA Y MECÁNICA
CARRERA DE INGENIERÍA EN MECATRÓNICA**

AUTORIZACIÓN

Yo, **JAVIER PATRICIO PICHUCHO RODRÍGUEZ**, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar en la biblioteca Virtual de la institución el presente trabajo de titulación “**INVESTIGACIÓN DE UN SISTEMA DE VISIÓN ARTIFICIAL PARA LA INTERPRETACIÓN DEL ALFABETO Y NÚMEROS DEL UNO AL DIEZ EN LENGUAJE DE SEÑAS ECUATORIANA MEDIANTE LA UTILIZACIÓN DE SOFTWARE LIBRE.**” cuyo contenido, ideas y criterios son de mi autoría y responsabilidad.

Latacunga, 19 de mayo del 2017

Javier Patricio Pichucho Rodríguez

C.C.: 1722341292

DEDICATORIA

A Dios por permitirme cada día ver un nuevo amanecer y mostrarme lo maravilloso que es la vida, por acompañarme y guiarme a cada paso que doy.

A mi madre querida Mariana que, con su cariño, amor y sobre todo su fortaleza me enseñó que con esfuerzo y dedicación todo se puede lograr. Por acompañarme en cada momento bueno o malo de la vida por estar siempre junto a mi respaldando cada decisión que tome y empujándome a cumplir todas las metas propuestas.

A mi padre Eduardo por tantos consejos que supo darme, por motivarme a seguir mis sueños, por enseñarme que la vida es de los valientes.

A mi bella esposa Karito que con su amor y su sola presencia me llena de mucha felicidad.

A mis tan amados Danna Nico y Emmita por ser la alegría de nuestro hogar y sobre todo por ser la fuerza que me permite seguir cada día hacia adelante sin importar los obstáculos que se presenten.

A mis tan queridos hermanos Eytel y Erick y a mi hermana Vale que siempre que la necesite estuvo allí para brindar su ayuda.

A mis suegros Don Marcelo y Doña Amelia que me han brindado un espacio en su familia y han sabido acogerme con mucho cariño.

A mis cuñaditas Dany, Karlita y Melita que con sus locuras más de una vez me han sacado una sonrisa.

Javier Pichucho

AGRADECIMIENTO

A Dios por darme la fuerza para poder culminar con esta etapa y poder llegar a cumplir la meta planteada.

A mi madre querida por tantas noches que me acompañó con su presencia durante todo el caminar estudiantil, por sus enseñanzas por su esfuerzo, por su paciencia, por su cariño ya que gracias a ella esta meta se llegó a cumplir.

A mi esposa amada la compañera de mis aventuras, por su amor, su paciencia y su apoyo incondicional.

A mi familia que siempre estuvo allí en cada momento brindándome su apoyo

A la Familia mecatrónica maestros, compañeros por brindarme tan valiosas enseñanzas y vivencias.

Javier Pichucho

ÍNDICE DE CONTENIDOS

CARÁTULA.....	I
CERTIFICACIÓN.....	II
AUTORÍA DE RESPONSABILIDAD	III
AUTORIZACIÓN	IV
DEDICATORIA.....	V
AGRADECIMIENTO.....	VI
ÍNDICE DE CONTENIDOS.....	VII
ÍNDICE DE TABLAS	XII
ÍNDICE DE FIGURAS.....	XIII
ABSTRACT	XVII

CAPÍTULO I

ASPECTOS GENERALES

1.1.	INTRODUCCIÓN.....	1
1.2.	ANTECEDENTES	2
1.3.	PLANTEAMIENTO DEL PROBLEMA.....	3
1.4.	JUSTIFICACIÓN E IMPORTANCIA	3
1.5.	DISCAPACIDAD AUDITIVA.....	4
1.5.1.	<i>Definición.</i>	4
1.5.2.	<i>Discapacidad Auditiva Características y Problemática.</i>	5
1.6.	PERSONA SORDA	5
1.6.1.	<i>Cultura Sorda</i>	5
1.7.	Lenguaje y Lengua	6
1.7.1.	<i>Lengua de Señas</i>	6
1.7.2.	<i>Lengua de Señas Ecuatoriana (LSEC)</i>	7
1.8.	SOFTWARE	9
1.9.	SOFTWARE LIBRE	9
1.10.	LIBRERÍAS PARA VISIÓN ARTIFICIAL.....	9
1.10.1.	<i>LabVIEW</i>	9

1.10.2.	<i>OpenCV (OPEN SOURCE COMPUTER VISION)</i>	11
1.10.3.	<i>MATLAB</i>	12
1.11.	VISIÓN ARTIFICIAL.	12
1.11.1.	<i>Etapas y niveles de la Visión Artificial</i>	13
1.12.	TRATAMIENTO DE IMÁGENES DIGITALES	13
1.12.1.	<i>Operaciones aritméticas</i>	14
1.12.2.	<i>Binarización por Umbral</i>	15
1.12.3.	<i>Ampliación y reducción de imágenes</i>	16
1.12.4.	<i>Operador detector de bordes Canny</i>	17
1.12.5.	<i>Región de interés</i>	19
1.13.	REDES NEURONALES ARTIFICIALES (RNA).....	19
1.13.1.	<i>Elementos de las Redes Neuronales Artificiales</i>	19
1.13.2.	<i>Redes de propagación hacia atrás (backpropagation)</i>	20
1.13.3.	<i>Funciones de activación</i>	20
1.13.4.	<i>Etapas de aprendizaje</i>	22
1.13.5.	<i>Aplicaciones de las Redes Neuronales Artificiales</i>	23
1.14.	RECONOCIMIENTO ÓPTICO DE CARACTERES.....	24
1.15.	ESTUDIO DEL LENGUAJE DE SEÑAS INTERPRETADO MEDIANTE EL USO DE VISIÓN ARTIFICIAL.....	25
1.16.	MATERIAL DIDÁCTICO UTILIZADO EN LA TRADUCCIÓN DEL LENGUAJE DE SEÑAS EN ECUADOR.	27

CAPÍTULO II

VISIÓN ARTIFICIAL Y TRATAMIENTO DE IMÁGENES

2.1.	SOFTWARE	31
2.2.	ADQUISICIÓN DE LA IMAGEN.....	32
2.3.	PROCESAMIENTO Y SEGMENTACIÓN DE LA IMAGEN.....	34
2.3.1.	<i>Segmentación área de interés</i>	34
2.3.2.	<i>Sustracción de fondo</i>	35
2.3.3.	<i>Operaciones morfológicas</i>	37
2.3.4.	<i>Detección de contornos</i>	39
2.3.5.	<i>Casco convexo</i>	40
2.3.6.	<i>Filtro de tamaño</i>	41
2.3.7.	<i>Detección de bordes</i>	42

2.4.	DESCRIPCIÓN Y EXTRACCIÓN DE CARACTERÍSTICAS.	44
2.4.1.	<i>Reducción de la dimensión de datos</i>	45
2.4.2.	<i>Descriptores de contorno</i>	46

CAPÍTULO III

REDES NEURONALES ARTIFICIALES

3.1.	RED NEURONAL ARTIFICIAL PARA CLASIFICACIÓN DE NÚMEROS	49
3.1.1.	<i>Clasificador</i>	49
3.2.	RED NEURONAL ARTIFICIAL PARA CLASIFICACIÓN DE LAS LETRAS DEL ALFABETO.....	54
3.2.1.	<i>Clasificador</i>	54
3.3.	RECONOCIMIENTO E INTERPRETACIÓN.....	59

CAPÍTULO IV

INTERFAZ GRÁFICA DE USUARIO (GUI);ERROR! MARCADOR NO DEFINIDO.

4.1.	DESCRIPCIÓN GENERAL.	61
4.2.	DESARROLLO DE LA INTERFAZ GRÁFICA.	62
4.2.1.	<i>Instalación de OpenCV y Qt Creator</i>	63
4.2.2.	<i>Ambiente de programación de Qt Creator</i>	63
4.2.3.	<i>Implementación de la GUI</i>	64
4.3.	FUNCIONAMIENTO PRINCIPAL	68
4.3.1.	<i>Modo aprender letras y números en LSEC</i>	68
4.3.2.	<i>Modo prueba números y letras en LSEC</i>	70

CAPÍTULO V

PRUEBAS Y RESULTADOS

5.1.	PRUEBAS DEL SISTEMA DE VISIÓN ARTIFICIAL Y DEL TRATAMIENTO DE IMÁGENES	72
5.1.1.	<i>Prueba de ambientes</i>	72
5.1.1.A.	Definición de los objetivos de la prueba	73
5.1.1.B.	Diseño de la prueba de ambientes	73
5.1.1.C.	Planificación de la prueba de ambientes.....	73

5.1.1.D.	Prototipo y banco de pruebas	73
5.1.1.E.	Interpretación y validación de los resultados de la prueba.....	74
5.1.2.	<i>Incidencia de la luz en el proyecto.</i>	76
5.1.3.	<i>Prueba de tono de piel y tamaño de la mano del usuario.</i>	78
5.1.3.A.	Definición de los objetivos de la prueba.....	78
5.1.3.B.	Diseño de la prueba de tono de piel y tamaño de la mano del usuario	78
5.1.3.C.	Planificación de la prueba de tono de piel y tamaño de la mano del usuario	79
5.1.3.D.	Prototipo y banco de pruebas	79
5.1.3.E.	Interpretación y validación de los resultados de la prueba.....	79
5.2.	PRUEBAS DE LOS CLASIFICADORES	81
5.2.1.	<i>Prueba del clasificador encargado del reconocimiento de los números en LSEC.</i>	81
5.2.1.A.	Definición de los objetivos de la prueba	81
5.2.1.B.	Diseño de la prueba de salida del clasificador encargado del reconocimiento de los números en LSEC	82
5.2.1.C.	Planificación de la prueba de salida del clasificador encargado del reconocimiento de los números en LSEC	82
5.2.1.D.	Prototipo y banco de pruebas.....	82
5.2.1.E.	Interpretación y validación de los resultados de la prueba	83
5.2.2.	<i>Pruebas del clasificador encargado del reconocimiento de las letras del abecedario en LSEC</i>	84
5.2.2.A.	Definición de los objetivos de la prueba.....	84
5.2.2.B.	Diseño de la prueba de salida del clasificador encargado del reconocimiento de los números en LSEC	84
5.2.2.C.	Planificación de la prueba de salida del clasificador encargado del reconocimiento de los números en LSEC	84
5.2.2.D.	Prototipo y banco de pruebas	85
5.2.2.E.	Interpretación y validación de los resultados de la prueba.....	85
5.3.	PRUEBA A LA INTERFAZ GRÁFICA DE USUARIO (SISTEMA COMPLETO)..	86
5.3.1.	<i>Definición de los objetivos de la prueba</i>	87
5.3.2.	<i>Diseño de la prueba de la interfaz gráfica de usuario (sistema completo)</i>	87

5.3.3.	<i>Planificación de la prueba de la interfaz gráfica de usuario</i>	87
5.3.4.	<i>Prototipo y banco de pruebas</i>	88
5.3.5.	<i>Interpretación y validación de los resultados de la prueba</i>	89
5.4.	ALCANCES Y LIMITACIONES.....	92
5.4.1.	<i>Alcances del sistema</i>	92
5.4.2.	<i>Limitaciones</i>	93
5.5.	VALIDACIÓN DE LA HIPÓTESIS	93
5.6.	COSTOS	96

CAPITULO VI

CONCLUSIONES Y RECOMENDACIONES

6.1.	CONCLUSIONES.....	97
6.2.	RECOMENDACIONES	99

ANEXOS

Anexo A: Toma de muestras para el entrenamiento de las RNA

Anexo B: Pruebas del sistema de Visión Artificial

Anexo C: Imágenes de la prueba de números y letras del alfabeto en LSEC

Anexo D: Distribución Chi - cuadrado

Anexo E: Código Implementado

Anexo F: Manual de usuario

ÍNDICE DE TABLAS

Tabla 1:	Bloques utilizados en la adquisición de una imagen a través de una webcam en LabVIEW.....	10
Tabla 2	Módulos de OpenCV.....	11
Tabla 3	Funciones de activación para una red neuronal	21
Tabla 4	Especificaciones técnicas del computador utilizado	31
Tabla 5	Comparación de software	32
Tabla 6	Verificación del error en la clasificación de los números en LSEC.....	54
Tabla 7	Verificación del error en la clasificación de las letras del abecedario en LSEC	57
Tabla 8	Prueba de ambientes no controlados	75
Tabla 9	Lista de errores en la detección de la mano del usuario	77
Tabla 10	Pruebas del tono y tamaño de mano	80
Tabla 11	Resultados del reconocimiento de los números en LSEC	83
Tabla 12	Resultados del reconocimiento de las letras del abecedario en LSEC.....	86
Tabla 13	Tiempo de familiarización con el programa desarrollado	89
Tabla 14	Tiempo de aprendizaje y porcentaje obtenido en la prueba desarrollada	91
Tabla 15	Frecuencias de valores observados.....	94
Tabla 16	Frecuencias de valores esperados	95
Tabla 17	Chi-Cuadrado calculado.....	95
Tabla 18	Costos de implementación del programa.....	96

ÍNDICE DE FIGURAS

Figura 1	Abecedario en Lengua de Señas Ecuatoriana.....	8
Figura 2	Números Naturales en Lengua de Señas Ecuatoriana	8
Figura 3	Entorno de programación en LabVIEW	10
Figura 4	OpenCV en la IDE de Qt Creator.....	11
Figura 5	Entorno de programación para MATLAB	12
Figura 6	suma de imágenes en escala de grises.....	14
Figura 7	Resta de imágenes	15
Figura 8	Binarización de una imagen.....	15
Figura 9	Escalado de imagen	16
Figura 10	Canny de una imagen.....	18
Figura 11	ROI de una imagen.....	19
Figura 12	Esquema de red neuronal.....	20
Figura 13	Modelo matemático para una neurona artificial	21
Figura 14	Etapas fundamentales para el sistema de visión artificial.....	30
Figura 15	Acer cristal eye webcam	33
Figura 16	Imagen obtenida a través de la webcam	33
Figura 17	Algoritmo para la adquisición de imágenes en OpenCV	34
Figura 18	Procesamiento y segmentación de la imagen	34
Figura 19	Extracción de la ROI.....	35
Figura 20	Código para la extracción del ROI	35
Figura 21	Sustracción de fondo	36
Figura 22	Código para sustracción de fondo	36
Figura 23	Erosión de la imagen original.....	37
Figura 24	Dilatación de la imagen original	37
Figura 25	Apertura en la imagen original	38
Figura 26	Cierre de la imagen original	38
Figura 27	Operaciones morfológicas a la imagen original	39
Figura 28	Códigos para operaciones morfológicas.....	39
Figura 29	Contorno de la mano	39
Figura 30	Código para la búsqueda y dibujo de los contornos de la imagen	40

Figura 31	Casco convexo alrededor de la mano del usuario	40
Figura 32	Código para casco convexo.....	41
Figura 33	Filtro de tamaño para obtención de la mano.....	41
Figura 34	Código para el filtro de tamaño.....	42
Figura 35	Máscara de la mano	42
Figura 36	Código para encontrar una máscara.....	43
Figura 37	Detección de bordes.....	43
Figura 38	Realce de características e inversión de color.....	44
Figura 39	Recorte de imagen.....	45
Figura 40	Reducción de imagen	46
Figura 41	Códigos de cadena	47
Figura 42	Función sigmoide estándar.....	48
Figura 43	Modelo para creación de clasificadores.....	49
Figura 44	Carpetas que contienen las imágenes de entrada.....	50
Figura 45	Información de las imágenes de entrada en 0 y 1	50
Figura 46	Red Neuronal Artificial para clasificación de números	51
Figura 47	Datos utilizados para el entrenamiento de la RNA.	51
Figura 48	Código para la función de activación	52
Figura 49	Código utilizado para el entrenamiento de la RNA	52
Figura 50	Entrenamiento de la RNA	53
Figura 51	Comprobación del entrenamiento de la RNA.....	53
Figura 52	Carpetas que contienen las imágenes de entrada.....	55
Figura 53	Información de las imágenes de entrada en 0 y 1	55
Figura 54	RNA para clasificación de las letras del alfabeto	56
Figura 55	Parámetros para el entrenamiento de la RNA.	56
Figura 56	Iteraciones para el entrenamiento de la RNA	58
Figura 57	Comprobación del entrenamiento de la red neuronal artificial ...	58
Figura 58	Reconocimiento del número 1 en LSEC	59
Figura 59	Reconocimiento de la letra g en LSEC	60
Figura 60	Funcionamiento de la GUI	62
Figura 61	Asistente para la creación de nuevas aplicaciones	64
Figura 62	Pantalla para creación de GUI en Qt Creator	64
Figura 63	Código para la captura de frames.....	65
Figura 64	Código para la impresión de frames en la GUI	65

Figura 65	Impresión de frames en la GUI	65
Figura 66	Código para la utilización de botones de radio	66
Figura 67	Combinación entre menús de la GUI	66
Figura 68	Código para la utilización de botones	67
Figura 69	Botones utilizados en la GUI.....	67
Figura 70	Código para imprimir texto en GUI.....	67
Figura 71	Impresión de texto en el Widget	68
Figura 72	Bosquejo interfaz gráfica	68
Figura 73	Modo aprender números en LSEC	69
Figura 74	Aprendizaje de números en LSEC de manera correcta.....	69
Figura 75	Aprendizaje de números en LSEC de manera incorrecta	69
Figura 76	Modo prueba letras en LSEC.....	70
Figura 77	Realización de la letra en LSEC de manera correcta	70
Figura 78	Realización de la letra en LSEC de manera incorrecta.....	71
Figura 79	Impresión del porcentaje de aciertos y error	71
Figura 80	Banco de pruebas.....	74
Figura 81	Ambiente propicio para el programa	74
Figura 82	Perdida de datos en la imagen	75
Figura 83	Reconocimiento de la mano de un niño.....	80
Figura 84	Reconocimiento de la mano de una persona adulta mayor	80
Figura 85	Banco de pruebas de los números en LSEC	83
Figura 86	Banco de pruebas letras del abecedario en LSEC	85
Figura 87	Banco de pruebas para la Interfaz gráfica de usuario.....	89
Figura 88	Usuario dando la prueba de los números en LSEC	90
Figura 89	Usuario realizando la prueba del alfabeto en LSEC	90

RESUMEN

El presente proyecto de investigación brinda una herramienta interactiva para el aprendizaje del alfabeto y números del uno al diez en Lenguaje de Señas Ecuatoriana (LSEC) sin importar el entorno donde sea ejecutado, ni el usuario, quien puede ser niño o niña, adolescente, adulto o adulto mayor. Para el desarrollo del sistema se hace uso de elementos de Inteligencia Artificial (IA) como la Visión Artificial (VA) para la adquisición y tratamiento de las imágenes digitales, y las Redes Neuronales Artificiales (RNA) para la creación de dos clasificadores: uno encargado del reconocimiento de los números y el otro encargado del reconocimiento de las letras del alfabeto en LSEC. Las muestras utilizadas en el entrenamiento de los clasificadores fueron tomadas a personas que pertenecen a la FENASEC y al INAL debido a que conocen bien el Lenguaje de Señas Ecuatoriana. Se desarrolla una interfaz gráfica de usuario que contiene dos menús uno para el aprendizaje y otro para la evaluación de conocimientos. El presente proyecto brinda la facilidad al usuario para acceder a la interfaz desarrollada mediante la utilización de software libre, por lo que se propone su libre difusión, y así contribuir a la inclusión de personas con discapacidad auditiva.

PALABRAS CLAVE:

- **LENGUAJE DE SEÑAS - ECUADOR**
- **VISIÓN ARTIFICIAL**
- **REDES NEURONALES ARTIFICIALES**
- **LIBRERÍAS OPENCV**

ABSTRACT

This research project provides an interactive tool for learning the alphabet and numbers from one to ten in Ecuadorian Sign Language (LSEC) Regardless of the setting where it is used, or the user who can be a boy or a girl, teenager, adult, or older adult. Artificial Intelligence (AI) elements have been used for the system development such as Artificial Vision (VA) for the acquisition and treatment of digital images, and Artificial Neural Networks (ANN) for the creation of two classifiers: one in charge of number recognition and the other one in charge of alphabet recognition LSEC. The samples used in the training of the classifiers were taken from people who belong to FENASEC and INAL because they know well Ecuadorian Sign Language. A graphical user interface was developed that contains two menus, one for learning and another for the assessment of learning. This provides the user facility to access the interface, which was developed with the use of free software, its free diffusion is proposed in order to contribute to social inclusion of people with hearing disabilities.

KEYWORDS:

- **Sign language - ECUADOR**
- **Artificial Vision**
- **Artificial Neural Networks**
- **Bookstores OPENCV**

CAPÍTULO I

ASPECTOS GENERALES

1.1. Introducción

La sociedad ecuatoriana hace ya varios años ha venido interesándose en los grupos de atención prioritaria y personas con discapacidad. Dentro del grupo de individuos con discapacidad se encuentran aquellas que poseen discapacidad auditiva, y se puede referir que la población en general no ha recibido una adecuada sensibilización sobre las potencialidades y derechos de las personas con este tipo de discapacidad por lo que en ocasiones han sido discriminadas en todos los niveles y aisladas de sujetos denominados “normales o sin discapacidad”.

Las personas que poseen discapacidad auditiva alrededor del mundo, como en el país requieren de la atención y el apoyo de toda herramienta tecnológica que ayude a incrementar sus capacidades y aporte con su progreso tanto social, lingüístico y psicológico. Los estereotipos sobre las personas sordas han generado el bajo involucramiento en instituciones educativas, menores oportunidades laborales y por ende se han visto perjudicados en su economía.

Para cooperar con el progreso de las personas que muestran discapacidad auditiva es conveniente que la sociedad tenga la oportunidad de poder comunicarse con ellos y de alguna forma contribuir con la atención de sus inquietudes, requerimientos y seguir fomentando la defensa de sus derechos. La comunicación entre personas sordas y personas oyentes es indispensable ya que se convive en una sociedad pluricultural y solidaria, que puede adoptar su rasgo identitario como lo es el lenguaje de señas, el cual es utilizado como canal para comunicarse con su entorno. Este modelo de lenguaje se debe propagar o difundir y lo que es fundamental aprenderlo.

La comunidad de sordos necesita más inclusión social en todos sus niveles y que la población oyente conozca su lenguaje de señas, su forma de vivir, su cultura, dejar a un lado toda clase de estereotipos. Desarrollar la comprensión ayudará a mejorar la comunicación y además aportará en el desarrollo de sus relaciones sociales.

1.2. Antecedentes

El Consejo Nacional para la Igualdad de Discapacidades (CONADIS) en conjunto con el Ministerio de Salud Pública del Ecuador en su registro nacional de discapacidades de febrero del 2017 muestra que en Ecuador 418.001 personas cuentan con una discapacidad de las cuales un 12.81% de estas personas poseen discapacidad auditiva y 1.34% de personas discapacidad de lenguaje (Ministerio de Salud Pública, 2017). El artículo 70 de la Ley Orgánica de Discapacidades del Ecuador del año 2012 reconoce a la Lengua de Señas Ecuatoriana como lenguaje y medio de comunicación de las personas con discapacidad auditiva dentro del país.

Con el avance tecnológico se pueden desarrollar nuevas herramientas de aprendizaje de toda índole orientado a facilitar la enseñanza de las personas, pero que están fuera del alcance económico de muchas de ellas, podemos citar por ejemplo la utilización del guante SignLanguageGlove el cual traduce el lenguaje de señas en voz y texto pero que actualmente se encuentra en desarrollo, en EEUU tiene un costo de 380 dólares sin contar mano de obra; o la iniciativa de la utilización de MotionSavvy cual es creada por Leap Motion para el reconocimiento de gestos a través de una Tablet aunque está en su fase beta se presenta como un producto innovador pero con costo elevado ya que el precio oscilaría entre 600 dólares y actualización de software mensual con un costo de 20 dólares adicionales.

En La Universidad De Las Fuerzas Armadas ESPE tanto en su Matriz como en su Extensión en Latacunga se ha diseñado y construido anteriormente prototipos de reconocimiento e instrucción del lenguaje de señas.

1.3. Planteamiento del problema

Un niño o niña al nacer con problemas de discapacidad auditiva en un hogar de personas oyentes que desconocen el lenguaje de señas es un problema para él ya que la comunicación y aprendizaje del niño o niña no será la adecuada sino hasta el ingreso a una entidad educativa especializada en personas con discapacidades, en sus primeras etapas de vida el niño o niña no podrá comunicarse ni entender de buena manera a las demás personas que lo rodean ya que el aprendizaje del lenguaje de señas de manera empírica no le permite socializar con el mundo externo sino únicamente en un entorno familiar.

Cabe mencionar que el lenguaje de señas no es universal, es por esto que Ecuador reconoce que posee su lengua de señas particular el mismo que es utilizado para la comunicación entre personas sorda dentro del territorio ecuatoriano.

Otro inconveniente radica en la dificultad para acceder a instrumentos tecnológicos que ayuden con esta tarea, por su costo y casi nulo acceso ya que la tecnología no existe en el país e importarla causaría complicaciones debido a que muchos de los interpretes o traductores de lenguaje de señas están desarrollados para los países de fabricación y como ya se ha mencionado el lenguaje de señas no es universal.

1.4. Justificación e importancia

La investigación busca facilitar la enseñanza del Lenguaje de Señas Ecuatoriana a todo individuo mediante la utilización de herramientas tecnológicas que ofrezcan el aprendizaje interactivo y que este enfocado a las necesidades del Plan Nacional del Buen Vivir 2013-2017 que busca la inclusión social de todos los miembros que forman el Ecuador, el proyecto se orienta a mejorar las condiciones de vida de personas con discapacidad y es parte fundamental de uno de sus objetivos el cual señala que todo individuo posee derechos igualitarios implicando la consolidación de

políticas de igualdad que eviten la exclusión y fomenten la convivencia social y política entre los miembros del país (Secretaría Nacional de Planificación y Desarrollo, 2013).

Con el diseño e implementación de la interfaz se busca incrementar el número de personas que conocen el Lenguaje de Señas Ecuatoriana. Mecatrónica realiza aplicaciones que integran tecnologías orientadas a solucionar problemas de la localidad y de la sociedad en general.

Con anterioridad se han desarrollado proyectos similares, sin embargo, no han solucionado problemas de iluminación se han implementado en entornos controlados que limitan la eficiencia de la aplicación, es por eso que el proyecto busca crear un sistema capaz de interactuar con el usuario indistintamente del entorno que lo rodee.

La utilización de guantes traductores del lenguaje de señas presenta como principal inconveniente su difícil adaptación a diferentes usuarios ya que las manos de un niño o niña no poseen las mismas dimensiones que las de una persona adulta. Un intérprete humanoide es muy amigable para la enseñanza del lenguaje de señas, pero muy costoso limitando su adquisición.

La importancia del proyecto que se presenta radica en la facilidad que posee el usuario para acceder a la interfaz ya que se propone su libre difusión.

1.5. Discapacidad Auditiva.

1.5.1. Definición.

La Organización Mundial de la Salud define a las personas con discapacidad auditiva como aquellos individuos cuya agudeza auditiva es insuficiente para aprender su propia lengua, beneficiarse de la enseñanza escolar y participar en actividades normales de acuerdo a su edad (Bueno & Limiñana, 2002).

1.5.2. Discapacidad Auditiva Características y Problemática.

La discapacidad auditiva puede afectar varias funciones que son propias de un ser humano como son: el lenguaje, su desarrollo intelectual, el desarrollo afectivo y su comportamiento puede variar constantemente dependiendo del grado de discapacidad auditiva que una persona posea.

De acuerdo con Sarrionandia & Minguillón (1995) existen varias necesidades especiales en el área educativa para personas con discapacidad auditiva entre ellas las siguientes:

- Necesidad de requerir estrategias visuales y aprovechar otros canales para la comunicación (gestos auditivos, tacto).
- Necesidad de un método lingüístico de representación.
- Necesidad de fortalecer la autoestima y la identidad.
- Necesidad de adjudicar/integrar un código comunicativo conveniente.
- Necesidad de aprender un lenguaje diferente al mayoritario.

1.6. Persona Sorda

La persona Sorda es definida como aquel individuo visual que integra el lenguaje basándose en la visión (dactilología, señas, lectura de labios, grafía) y que de acuerdo a su percepción logra almacenar información en su memoria, mucha de las veces se vale de códigos como lo son: códigos fonológicos, manual-kinestésicos, viso espaciales, etc (CONADIS, CORDICOM, & FENASEC, 2015).

1.6.1. Cultura Sorda

La cultura Sorda nace de un proceso histórico formativo y social de un definido grupo de la colectividad dentro del ambiente cultural, con particularidades propias que se han ido adoptando y creando para satisfacer necesidades de la comunidad Sorda; cada comunidad Sorda es una

agrupación cultural que posee una lengua de señas y un patrimonio común(Riotordo, 2012).

1.7. Lenguaje y Lengua

Se define el lenguaje como una habilidad exclusivamente humana constituido por códigos los cuales nos permiten manifestar nuestras ideas sobre el entorno que nos rodea de manera clara y precisa a través de un método de signos utilizados para la comunicación(Pardo, 2010).

En cambio, la lengua es la fracción social del lenguaje, constituido de un sistema de signos particulares de una comunidad, creada como consecuencia de la necesidad de transmitir un mensaje entre individuos de un mismo agrupamiento.

1.7.1. Lengua de Señas

De acuerdo con Alarcón, Fonseca, & Rosas, (2012) La lengua de señas, es un lenguaje innato de los individuos que tienen discapacidad auditiva, este lenguaje está colmado de manifestaciones visuales mediante la cual se puede establecer un canal de comunicación con su ambiente social, ya sea estructurado por otros individuos sordos o por alguna persona que entienda la lengua de señas. El medio comunicativo empleado en el lenguaje de señas es el gesto-viso-espacial. Una singularidad de esta lengua es que a cada individuo se le concede un signo particular y característico para su identificación y no es necesario deletrear su nombre en signos.

Hay que tomar en cuenta que la lengua de señas posee una estructura gramatical diferente a la de la lengua oral, tiene como característica principal la configuración de las manos, su orientación, su ubicación espacial así como sus movimientos, existen también componentes no manuales como es el movimiento facial, labial y lingual, es importante indicar que el lenguaje de señas no es universal, aunque la comunicación entre personas sordas de diferentes comunidades presenta menos inconvenientes que el lenguaje oral.

1.7.2. Lengua de Señas Ecuatoriana (LSEC)

En Ecuador las discapacidades han existido a lo largo de la historia formando parte de la estructura de la sociedad, viviendo diferentes realidades atravesando todo tipo de abuso, maltrato hasta la inclusión que hoy se vive, se reconoce que una persona con discapacidad auditiva es un integrante de la comunidad que posee una lengua propia para poder comunicarse, una lengua que utiliza un lenguaje visual que está lleno de movimientos manuales, corporales y faciales. La lengua de señas no es un lenguaje universal, Ecuador como varios países del mundo cuenta con su propia lengua de señas y es utilizada en todo el territorio patrio es importante mencionar que sus variaciones en comparación con otros lenguajes de señas son importantes es por eso que es único y propio de los ecuatorianos (CONADIS et al., 2015).

a) Abecedario y números naturales en lengua de Señas Ecuatoriana

El proyecto se basa en la creación de una interfaz ejecutable que impulse el aprendizaje del alfabeto y números naturales del 1 al 10 en Lengua de Señas Ecuatoriana por tal motivo se debe conocer las formas de la mano para crear las diferentes letras y números, la figura 1 muestra el alfabeto en LSEC.

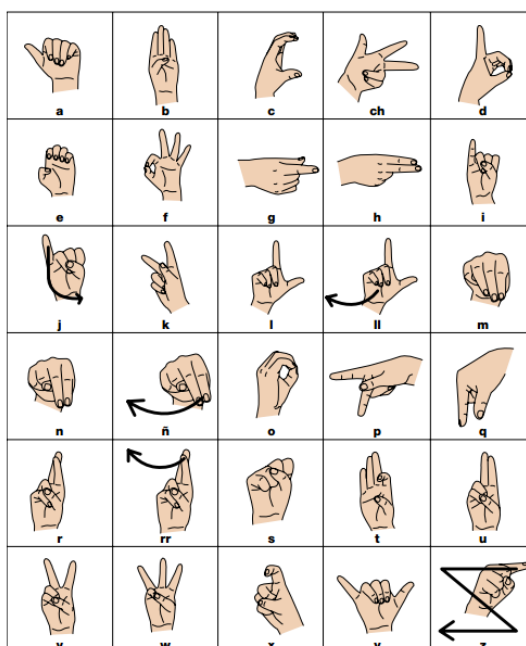


Figura 1 Abecedario en Lengua de Señas Ecuatoriana

Fuente: (FENASEC, 2012)

La figura 2 muestra los números naturales hasta el número doce, pero para la investigación únicamente se utilizan hasta el numeral diez.

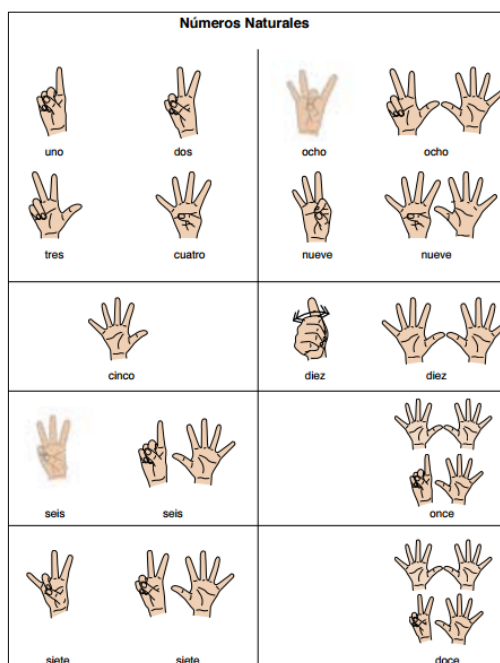


Figura 2 Números Naturales en Lengua de Señas Ecuatoriana

Fuente: (FENASEC, 2012)

1.8. Software

De acuerdo con Segunda Cohorte del Doctorado en Seguridad Estratégica (2014) software es el conjunto de programas de cómputo, procedimientos , reglas, documentación y datos afiliados que forman parte de las acciones de un sistema de computación que es empleado por una persona para ejecutar procedimientos informáticos.

1.9. Software libre

Designación del software que respeta la libertad de los usuarios sobre su beneficio de usar, ejecutar, modificar, copiar, estudiar hasta de mejorar un programa con el propósito de adaptar a las necesidades del consumidor, hay que tomar en cuenta que no es lo mismo software libre que software gratuito ya que muchas de las veces este puede ser distribuido comercialmente sin dejar su carácter de libre (Valencia, 2012).

1.10. Librerías para visión artificial

En el desarrollo de aplicaciones que utilicen visión artificial es importante elegir una librería como base ya que esta será la responsable de facilitar el cumplimiento de los requerimientos del usuario, existen librerías que poseen licencia de orden libre como de orden comercial, a continuación se presentan las librerías más empleadas en el ámbito de la visión artificial (Blanco Garcia-Moreno, 2013).

1.10.1. LabVIEW

LabVIEW es un ambiente de desarrollo integrado y proyectado específicamente para ingenieros e investigadores que desarrollan sistemas de medidas y control. Su lenguaje de programación es gráfico, posee un IP integrado para estudio de datos y procesamiento de señales, así como una arquitectura abierta que admite la integración de cualquier dispositivo de

hardware y cualquier enfoque de software, su programación es orientada mediante gráficos (National Instruments Corporation, 2017).

LabVIEW en su entorno de programación ofrece módulos de visión artificial los cuales poseen un costo y están dirigidos a la adquisición y tratamiento de las imágenes ofreciendo soluciones para monitoreo y seguridad entre los módulos que ofrece este software encontramos el NI Vision Builder for Automated Inspection y el NI Vision Development Module. La figura 3 muestra el entorno de programación de LabVIEW.

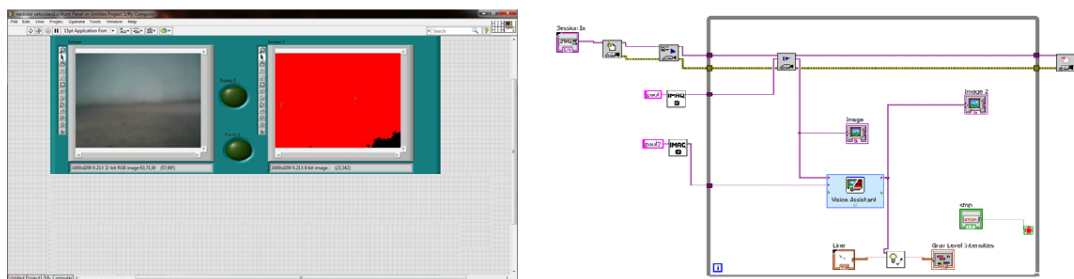






Figura 3 Entorno de programación en LabVIEW

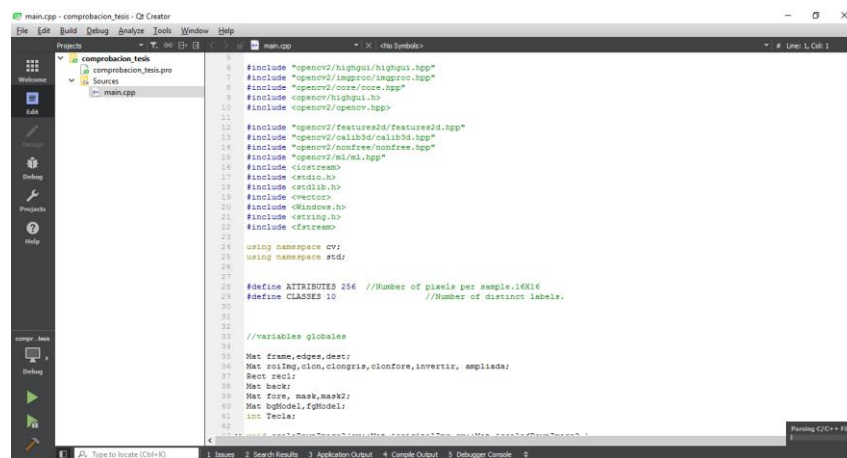
Los principales bloques utilizados en la adquisición de una imagen a través de una cámara web en el entorno de desarrollo de LabVIEW se muestran en la tabla 1.

Tabla 1
Bloques utilizados en la adquisición de una imagen a través de una webcam en LabVIEW

	Abre la cámara
	Configura el software
	Inicia la adquisición de la imagen
	Cierra la cámara

1.10.2. OpenCV (OPEN SOURCE COMPUTER VISION)

OpenCV es una librería orientada al procesamiento visual, visión por computador en tiempo real, posee licencia abierta BSD y por lo tanto está libre para su uso tanto académico como comercial. Posee interfaces de C, C++, Java y Python, posee afinidad con ciertos sistemas operativos como son Linux, Windows, Mac OS, iOS y Android (Itseez, 2016) en la figura 4 se muestra el empleo de la librería de OpenCV.



```

1 #include <opencv2/opencv.hpp>
2
3 #include <opencv2/highgui/highgui.hpp>
4 #include <opencv2/imgproc/imgproc.hpp>
5 #include <opencv2/core/core.hpp>
6 #include <opencv2/highgui.h>
7 #include <opencv2/opencv.hpp>
8
9 #include <opencv2/features2d/features2d.hpp>
10 #include <opencv2/calib3d/calib3d.hpp>
11 #include <opencv2/nonfree/nonfree.hpp>
12 #include <opencv2/ml/ml.hpp>
13 #include <iostream>
14 #include <stdio.h>
15 #include <stdlib.h>
16 #include <vector>
17 #include <windows.h>
18 #include <string.h>
19 #include <fstream>
20 using namespace cv;
21 using namespace std;
22
23 #define ATTRIBUTES 256 //Number of pixels per sample.16x16
24 #define CLASSES 10 //Number of distinct labels.
25
26 //variables globales
27
28 Mat Erase, edges, desc;
29 Mat img1p, g1on, c1ongria, c1onfore, invertis, amplada;
30 Rect rect;
31 Mat back;
32 Mat fore, mask, mask2;
33 Mat bgModel, fgModel;
34 int Text1;

```

Figura 4 OpenCV en la IDE de Qt Creator

Los principales módulos que maneja la librería de OpenCV según (Rodríguez Escudero, 2012) se enlistan en la tabla 2.

Tabla 2
Módulos de OpenCV

Core	Módulo que comprende las funciones para el procesamiento aritmético y las estructuras de datos básicas.
Highhui	Módulo que permite capturar, guardar archivos de video o de imagen, así como utilizar la interfaz gráfica de usuario
Imgproc	Módulo que permite el procesamiento de imágenes mediante filtros por ejemplo el Gaussiano, transformadas entre otras.
Video	Contiene la consideración de movimiento, extracción de fondos y algunas funciones para el seguimiento de objetos.
Objdetect	Encargado de la detección de objetos como caras, personas, etc.

1.10.3. MATLAB

MATLAB proviene de dos términos MATriz y LABoratorio es un ambiente de computación que permite el desarrollo de aplicaciones que contengan análisis numérico, procesamiento de señales e interfaz de visualización gráfica, posee un lenguaje de programación en C aunque en sus inicios utilizaba un lenguaje denominado FORTRAN es compatible con sistemas operativos como Macintosh, Windows y UNIX, posee licencia privativa (García Santillán, 2008)

MATLAB trabaja mediante toolboxes los cuales permiten que el programa realice operaciones de utilidad en todas las áreas de la ingeniería entre ellos está el procesamiento de imágenes, redes neuronales, lógica difusa, procesamiento de señales, etc (García Santillán, 2008). En la figura 5 se puede visualizar el entorno de programación en MATLAB, Image Processing Toolbox es la herramienta que utiliza MATLAB para el desarrollo de aplicaciones con visión artificial.

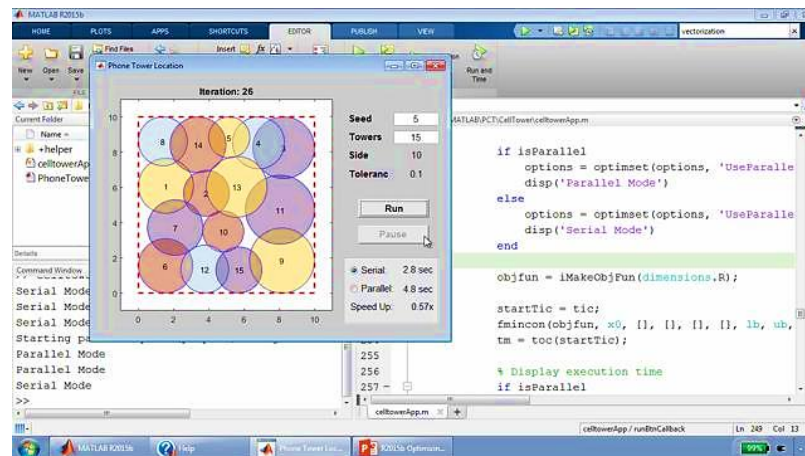


Figura 5 Entorno de programación para MATLAB

Fuente: (MathWorks, 2012)

1.11. Visión Artificial.

La Visión Artificial intenta capturar e interpretar la información visual del entorno que rodea a un sistema inteligente con el fin de extraer características cruciales para asimilar y preonosticar acontecimientos. Se la puede definir

también como parte de la inteligencia artificial que mediante el uso de técnicas lógicas permiten la adquisición, procesamiento y análisis de información obtenida a partir de imágenes digitales (Díez, Gómez, & Martínez, 2001).

1.11.1. Etapas y niveles de la Visión Artificial

Permiten el procesamiento de imágenes tomando en consideración que en cada etapa se va reduciendo y depurando la información innecesaria para el usuario. Por lo general se consideran tres niveles (Piedra Fernández, 2008).

- **Nivel bajo** orientado a la captura o adquisición de la imagen, así como su pre-procesamiento este nivel se encarga de la obtención de la imagen, reducción de ruido, aumento del contraste.
- **Nivel Intermedio** orientado a la segmentación, la descripción, el reconocimiento este nivel se encarga de dividir a la imagen en objetos de interés de estudio, así como también permite diferenciar objetos dependiendo de su forma, tamaño o área lo cual permite identificar objetos como pueden ser botellas, monedas entre otros.
- **Nivel alto** se encarga de la interpretación, asocia un significado a un grupo de elementos conocidos.

1.12. Tratamiento de imágenes digitales

El mejoramiento y tratamiento de imágenes es empleado directamente sobre la imagen, se efectúan alteraciones con el fin de conseguir información de datos que posee la imagen y que serán de utilidad en todos los niveles de visión, por cada característica extraída se obtendrá una nueva imagen.

1.12.1. Operaciones aritméticas

De acuerdo con Esqueda & Palafox (2005) existen cuatro operaciones aritméticas principales suma, resta multiplicación y división, las operaciones son aplicables a imágenes del mismo tamaño o de la misma dimensión para la investigación se tomara mayor énfasis en las operaciones de suma y resta las cuales se describen a continuación.

- **Suma de imágenes** se realiza mediante la utilización de la ecuación 1.

$$A(x,y) = B(x,y) + C(x,y)$$

Ecuación 1 Suma de imágenes

Donde la matriz A es la imagen de salida, la matriz B es la imagen 1 y la matriz C es la imagen número 2 como se aprecia en la figura 6.

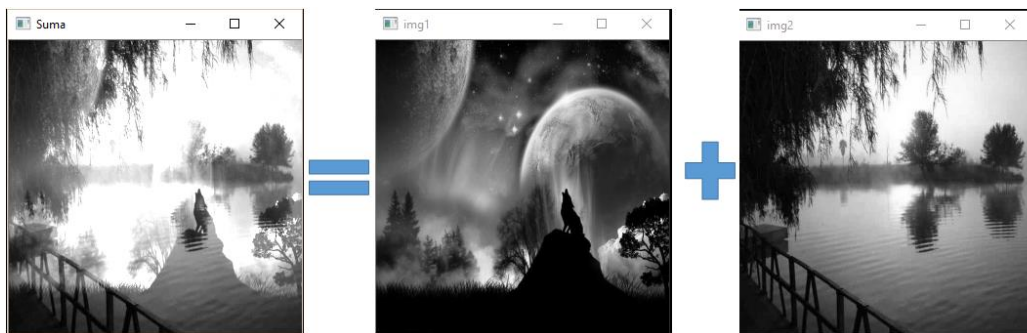


Figura 6 Suma de imágenes en escala de grises

- **Resta de imágenes** se realiza mediante la utilización de la ecuación 2.

$$A(x,y) = B(x,y) - C(x,y)$$

Ecuación 2 Resta de imágenes

Donde la matriz A es la imagen de salida, la matriz B es la imagen 1 y la matriz C es la imagen número 2 como se aprecia en la figura 7.

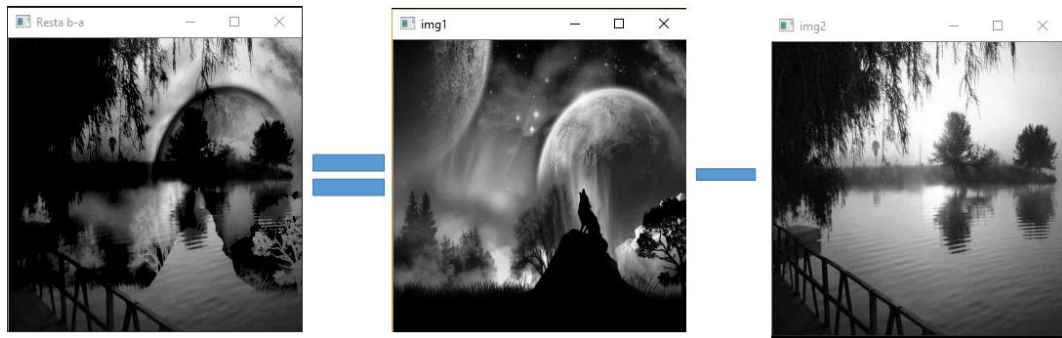


Figura 7 Resta de imágenes

1.12.2. Binarización por Umbral.

Operación puntual no lineal que permite que una imagen de entrada transforme sus píxeles para que tomen valores de 0 y 1, 0 y 255 o negro y blanco teniendo en cuenta un valor de umbral, el umbral es utilizado para separar un objeto de interés de una imagen, esto se logra ya que los píxeles menores al umbral tomarán valores de 0 y se mostrarán en color negro caso contrario tomarán valores de 1 y se mostrarán en color blanco como se puede observar en la figura 8.

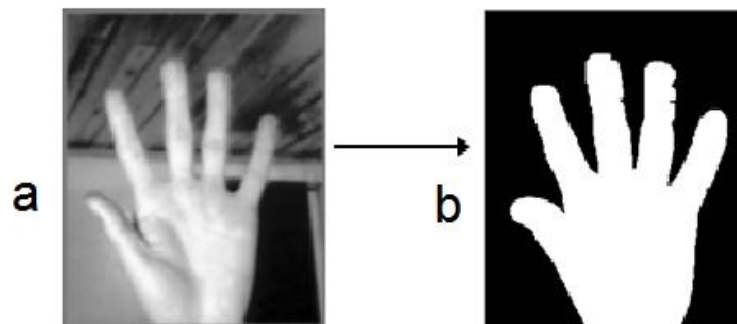


Figura 8 Binarización de una imagen

a) Imagen en escala de grises b) Imagen binarizada

De acuerdo con Alegre, Pajares, & de la Escalera (2016) la operación matemática que define a la binarización por umbral viene dada por la ecuación 3 siendo “x” e “y” las coordenadas de los píxeles en la imagen, $h(x,y)$ el valor de la intensidad de pixel de nuestra imagen binarizada,

mientras que $g(x,y)$ es el valor de la intensidad de pixel de nuestra imagen original y T es el valor de umbral.

$$h(x,y) = \begin{cases} 1 & \text{si } g(x,y) > T \\ 0 & \text{si } g(x,y) \leq T \end{cases}$$

Ecuación 3 Binarización por umbral

1.12.3. Ampliación y reducción de imágenes

De acuerdo con Alegre et al. (2016) al realizar una ampliación o reducción se varía el tamaño de una imagen inicial, la variación se la puede realizar a lo largo de las coordenadas del eje x como de las coordenadas del eje y , se interpreta mediante coeficientes de escala E_x y E_y los cuales si toman valores menores a la unidad son considerados como reducción, pero si toman valores mayores son considerados como ampliación, la transformación está dada por la ecuación 4.

$$X = E_x i, \quad y = E_y j$$

Ecuación 4 Coeficientes para el escalado de una imagen

En la figura 9 se aprecia la reducción y la ampliación de una imagen original utilizando los coeficientes de escalado.



Figura 9 Escalado de imagen

a) Imagen original b) Imagen ampliada c) Imagen reducida

1.12.4. Operador detector de bordes Canny

Canny es un detector de bordes que ofrece mejores resultado pero requiere mayor coste y complejidad computacional que otros operadores, el algoritmo se basa en el cambio de intensidad es decir el cambio brusco en la primera derivada o en su pendiente (Cauich, 2012).

De acuerdo con Cauich (2012) Canny se basa en tres criterios primordiales

- **De detección**, evita eliminar bordes significativos, así como proporcionar bordes falsos.
- **De localización**, la distancia entre la posición real y la posición localizada del borde debe ser mínima.
- **De respuesta única**, detectar un único punto de borde verdadero.

El método consta de tres etapas que serán descritas a continuación:

- **Obtener el gradiente** primero realizar un suavizado y eliminación de ruido de la imagen mediante la utilización de un filtro gaussiano a continuación se muestra un kernel igual a 5 para realizar el filtro

$$K = \frac{1}{159} \begin{pmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{pmatrix}$$

Ecuación 5 Máscara de convolución para filtro gaussiano

Una vez realizado el suavizado se encuentra la intensidad del gradiente en magnitud mediante la ecuación 6.

$$G = \sqrt{G_x^2 + G_y^2}$$

Ecuación 6 Intensidad del gradiente en magnitud

La dirección de la intensidad del gradiente se la encuentra mediante la ecuación 7.

$$\Theta = \arctan \left(\frac{G_y}{G_x} \right)$$

Ecuación 7 Intensidad del gradiente en dirección

Donde arctan significa el arco tangente del gradiente en dirección “x” como en dirección “y”.

- **Supresión no máxima al resultado del gradiente**, en este punto se eliminan los pixeles que no van a ser considerados como borde tomando en consideración la dirección y magnitud del gradiente, como imagen de salida se consigue una con los bordes adelgazados.
- **Histéresis de umbral a la supresión no máxima**, se realiza mediante la utilización de dos umbrales tal que el segundo sea más grande que el primero, los cuales permiten la eliminación de bordes que no se necesitan y eliminar líneas que se toman como si fueran reales, pero q no lo son.

En la figura 10 se puede observar cómo actúa el operador Canny con un kernel igual a 3 y con valores de umbral menor igual a 60 y valor de umbral mayor igual a 100.

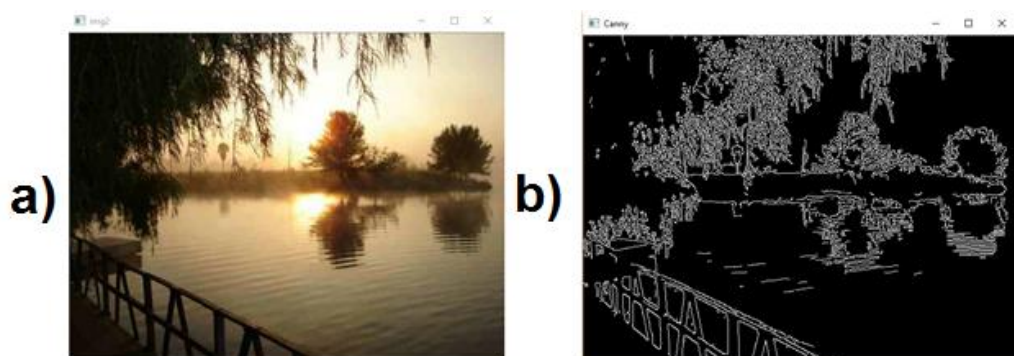


Figura 10 Canny de una imagen

a) Imagen original b) Imagen con Canny

1.12.5. Región de interés

La región de interés de una imagen se la conoce como ROI lo cual proviene del inglés (Region Of Interest), la ROI es una región de la imagen en el cual se desea aplicar mayor énfasis para su procesamiento, es utilizado principalmente para la reducción de coste computacional ya que el procesamiento se enfoca en un área reducida de la imagen, es generalmente especificada sobre un rectángulo (Brahmbhatt, 2013) la figura 11 presenta una ROI.

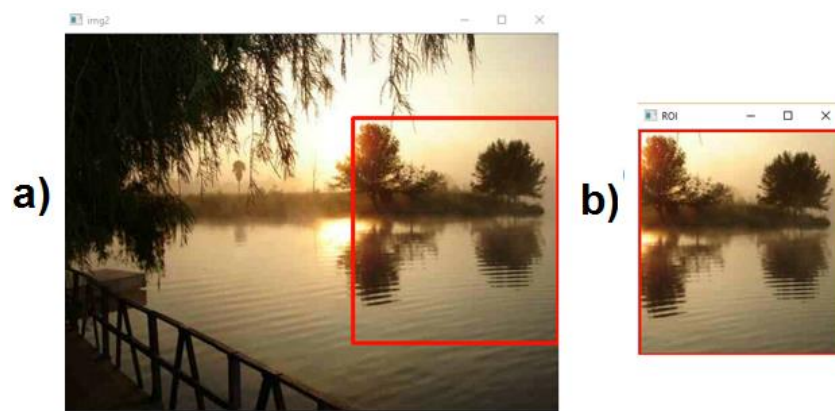


Figura 11 ROI de una imagen

a) Imagen original b) ROI

1.13. Redes Neuronales Artificiales (RNA).

Las RNA tratan de emular la conducta del cerebro humano, tienen una estructura configurada por varias neuronas artificiales las cuales tienen como objetivo extraer conocimiento a partir de un conjunto de datos, las RNA satisfacen necesidades de memorización, aprendizaje y clasificación de patrones a través de la experiencia tomada en la fase de entrenamiento.

1.13.1. Elementos de las Redes Neuronales Artificiales

Las RNA están compuestas de neuronas artificiales interconectadas integrando tres capas las mismas que son la capa de entrada, la capa oculta

y la capa de salida como se observa en la figura 12, es importante recalcar que la capa oculta puede estar compuesta de varias capas (Matich, 2001).

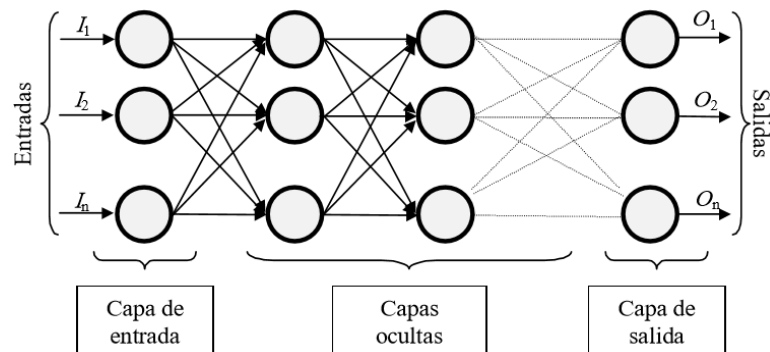


Figura 12 Esquema de red neuronal

Fuente: (Matich, 2001)

La ecuación 8 la misma que propone Sucar & Gómez (2010) para representar una red neuronal artificial se muestra a continuación.

$$S = g(\sum_i W_i E_i)$$

Ecuación 8 Red neuronal artificial

Siendo S la salida, W_i los pesos participantes y E_i las entradas.

1.13.2. Redes de propagación hacia atrás (backpropagation)

La retropropagación o backpropagation manifiesta que el error es propagado hacia atrás desde la capa de salida hacia la capa de entrada pasando por las capas ocultas permitiendo que los pesos de las conexiones entre neuronas cambien durante el entrenamiento influyendo directamente sobre la entrada global así como en su salida (Matich, 2001).

1.13.3. Funciones de activación

Las neuronas artificiales poseen un estado de activación el mismo que es imitado a partir del funcionamiento de las neuronas biológicas, siguen una regla tal que si excede un umbral se activaran las neuronas caso opuesto no,

los umbrales pueden tomar rangos de (0 a 1), (-1 a 1), es decir las neuronas permanecerán inactivas en 0 o -1 y se activaran cuando tomen un valor de 1 (Matich, 2001).

De acuerdo con Aguilar J. (2015) las funciones de activación que son comúnmente utilizadas en Redes Neuronales Artificiales son la escalón, semi-lineal y la sigmoidal las mismas que se presentan en la tabla 3.

Tabla 3
Funciones de activación para una red neuronal

Función de activación	Descripción
Función escalón	$\Phi(x) = \begin{cases} 1 & \text{si } x \geq 0 \\ -1 & \text{si } x < 0 \end{cases}$
Función semi-lineal	$\Phi(x) = \begin{cases} -1 & \text{si } x < -c \\ 1 & \text{si } x > c \\ ax & \text{en otro caso} \end{cases}$
Función sigmoidal	$\Phi(x) = \frac{1}{1 + e^{-\alpha x}}$

Para Constante & Gordón (2015) la salida de una neurona artificial viene dada por el siguiente modelo matemático representado en la figura 13

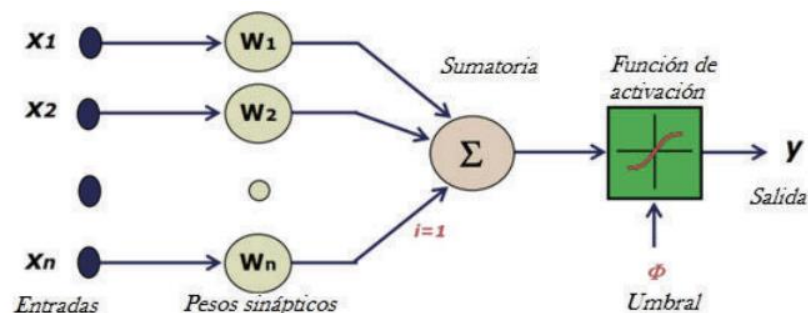


Figura 13 Modelo matemático para una neurona artificial

Fuente:(Constante & Gordón, 2015)

El mismo está definido mediante la ecuación 9.

$$Y = f(X_i W_i + \Phi)$$

Ecuación 9 Salida para una neurona artificial

1.13.4. Etapa de aprendizaje

La RNA es utilizada para la resolución de problemas, pero para esto debe pasar por una fase de aprendizaje la cual mediante la modificación de los pesos en respuesta a la información de entrada que se brinde a través del entorno logra ser entrenada, esto comprende que existirán cambios como la modificación, destrucción y la creación de enlaces entre neuronas, la creación de un enlace se produce cuando el valor de su peso es distinto de cero, en cambio un enlace se destruye cuando el valor de su peso pasa a ser cero, los pesos de los enlaces sufrirán varias modificaciones durante el proceso de aprendizaje, pero si estos pesos llegan a estabilizarse la red ha completado su aprendizaje (Matich, 2001).

Reglas de aprendizaje

De acuerdo con Flórez López & Fernández (2008) existen dos tipos de reglas de aprendizaje para el reajuste de los pesos: el aprendizaje supervisado y el aprendizaje no supervisado, cuya principal diferencia radica en la presencia o no de un agente supervisor que dirige el proceso de aprendizaje.

Aprendizaje supervisado

Consta de un agente externo el cual dirige el proceso de entrenamiento de tal forma que comprueba la salida de la RNA y compara con la salida deseada y si existe algún error va ajustando los pesos de forma interactiva hasta obtener la salida anhelada con el menor porcentaje de error posible (Flórez López & Fernández, 2008).

De acuerdo con Matich (2001) existen tres tipos de aprendizaje supervisado los cuales son enunciados a continuación:

- **Por corrección de error.** Se basa en ajustar los pesos de los enlaces de la red en función del error cometido en la salida.

- **Por refuerzo.** Este tipo de aprendizaje se basa en un mecanismo probabilístico ya que el agente supervisor no expresa que desea obtener como salida y únicamente se limita a actuar como un crítico ya que indica a la red si obtuvo éxito o fracaso y de acuerdo a esto se van ajustando los pesos de la red.
- **Estocástico.** Se basa en el cambio aleatorio del valor de los pesos de los enlaces de la red, valorando su resultado a partir de la salida deseada y de una repartición probabilística.

Aprendizaje no supervisado

Se basa en la experiencia observa como cambiará una red con la utilización de diferentes patrones de entrenamiento (Escobar, 2014). Es decir, para este tipo de aprendizaje no será necesario la comparación de una respuesta deseada con la respuesta que brinde la RNA ya que no se conoce cuál será la respuesta correcta, en si las neuronas de la red deben auto-organizarse para captar la periodicidad de los datos de entrada sin ninguna clase de ayuda externa (Gestal Pose, 2009).

1.13.5. Aplicaciones de las Redes Neuronales Artificiales

Las aplicaciones de las RNA son muy variadas ya que suelen ser utilizadas en aplicaciones comerciales como militares, de acuerdo con Matich (2001) estas son algunas de las aplicaciones comerciales más conocidas:

Biología

- Aprender más referente al cerebro y otros sistemas.
- Adquisición de modelos de la retina.

Medio ambiente

- Análisis de patrones.

- Pronostico del tiempo

Empresa

- Obtención de bases de datos
- Temporización de los semáforos obteniendo mayor optimización en el flujo del transito
- Reconocimiento de caracteres escritos
- Automatización y control
- Control de calidad

Medicina

- Monitorización de pacientes en cirugías
- Reconocimiento de bacterias en la sangre
- Pronóstico de rechazo a los medicamentos

Militares

- Vigilancia ininterrumpida
- Armas inteligentes
- Organización de señales de radar
- Búsqueda e identificación de patrones en tiro al blanco

1.14. Reconocimiento Óptico de Caracteres

Se lo conoce comúnmente como OCR ya que proviene del idioma inglés (Optical Character Recognition), es parte de investigaciones en el campo de la inteligencia artificial, su propósito principal es el reconocimiento de patrones en forma de caracteres, funcionan separando los posibles caracteres de una planilla y relacionándolos con patrones de caracteres ya conocidos hasta localizar uno que tenga el mayor parecido (Navarro et al., 2006).

1.15. Estudio del lenguaje de señas interpretado mediante el uso de visión artificial.

Hace ya varios años el estudio de la visión artificial ha sido utilizado para la interpretación y reconocimiento del lenguaje de señas a nivel mundial dando un paso importante en la inclusión social para la población que sufre de discapacidad auditiva, contribuyendo con esto a su desarrollo social y permitiendo brindar nuevos modelos de investigación para solucionar problemas de desconocimiento y mala interpretación del lenguaje de señas a continuación se describen algunos de los trabajos que se han desarrollado con el paso del tiempo.

Dolphio Technologies empresa que en el 2014 empieza a desarrollar SignAll un prototipo para un sistema de visión artificial el cual se encarga de la traducción del lenguaje de señas americano, consta de un sensor de movimiento Kinect de Microsoft, tres cámaras web normales, sistema de iluminación, fondo controlado y un software encargado de la traducción, el método que la empresa propone es capturar imágenes de todos los ángulos de la persona y mediante algoritmos realizados en un computador traducirlos a texto e imprimirlos en una ventana del ordenador. Lo importante del método es que busca capturar cada detalle de la persona y detectar cada uno de sus movimientos con el fin de que la traducción sea lo más acertada, el prototipo va cambiando de acuerdo al avance tecnológico y es preciso recalcar que aún se encuentra en proceso de mejora.

El trabajo presentado por Escudeiro et al. (2015) se enfatiza en la traducción bidireccional del lenguaje de señas portugués con ayuda de la visión artificial, para la traducción del lenguaje de señas utilizan dos tecnologías el sensor de movimiento Kinect de Microsoft y los guantes de sensores 5DT los cuales mediante la aplicación de algoritmos realizados en Opencv logran la traducción a texto de las señas realizadas y captadas por el sensor Kinect y los guantes, es importante recalcar que el método utilizado para el reconocimiento de los signos es Support Vector Machine con una eficacia mayor al 90% en la traducción. La conversión de la lengua

portuguesa al lenguaje de señas portuguesa se la realiza con la utilización de un avatar 3D desarrollado en el software Blender.

El proyecto presentado por López (2009) es capaz de reconocer el abecedario en lenguaje de señas ASL, para ello se utiliza la visión artificial su funcionamiento se basa en la captura de la imagen, la segmentación, uso de la apertura y cierre para el tratamiento morfológico, se extraen los datos más relevantes e importantes para la investigación los mismos que serán comparados con una base de datos ya establecida y con esto se obtiene el resultado de cada clase, es importante señalar que el algoritmo de comparación utilizada en el diseño es el DTW (Dinamic Time Warping), la eficacia del proyecto es del 68.75%, el manejo de la iluminación es importante para el autor ya que no se opera en ambientes que cambien constantemente.

La investigación expuesta por Arshad & Alam (2015) se basa en el reconocimiento del lenguaje de señas Bengali mediante el análisis de las yemas de los dedos, el algoritmo presentado en un principio realiza la captura de la imagen que contiene la mano en un ambiente controlado, al obtener la imagen se realiza la umbralización de la misma, una vez obtenida la forma de la mano se buscan las yemas de los dedos y el centroide de la imagen, con esto se obtiene un vector que contienen los ángulos y las distancias de cada dedo para luego con esto usar una RNA encargada del aprendizaje de cada signo del alfabeto Bengali. La eficiencia del proyecto presentado es del 88.69% el cual es muy alto en comparación con otros trabajos realizados para la interpretación del lenguaje de señas bengali.

Lo más relevante del proyecto de Cui (2009) es la utilización del método AdaBoost de Viola y Jones para la creación del clasificador en su etapa de entrenamiento en RNA, la función principal del algoritmo es obtener una imagen a través de la cámara web crear una máscara para separar la mano del entorno que lo rodea mediante el uso del HSV y contornos, una vez obtenida la mano se toma varias muestras para utilizarlas en la etapa de entrenamiento de la RNA. El autor no especifica el porcentaje de efectividad del método usado y las pruebas realizadas son únicamente con cinco

números mostrando un alto porcentaje de reconocimiento en la comprobación del procedimiento empleado.

1.16. Material didáctico utilizado en la traducción del lenguaje de señas en Ecuador.

La investigación en el Ecuador ha creado tecnologías que son de gran utilidad para el desarrollo de las personas con discapacidad entre estas las de discapacidad auditiva, se aborda comúnmente el problema del lenguaje de señas tomando como punto principal su aprendizaje, a continuación, se detallan algunos de los proyectos realizados en el país.

La investigación realizada por Vintimilla (2014) busca brindar una herramienta de ayuda a personas con discapacidad auditiva es desarrollada en Android con la ayuda de JAVA y se utiliza en un entorno controlado, el funcionamiento principal se basa en dos menús la de traducción del abecedario en lenguaje de señas y la del aprendizaje de nuevas características de las letras del abecedario mediante la generación de pesos para las RNA utilizadas en el proyecto. En un principio se realiza la captura de la imagen mediante la cámara de cualquier dispositivo que posea android, se aplica una conversión de la imagen a escala de grises, se produce la binarización de la imagen para obtener únicamente 0 y 1, la imagen cambiará de tamaño a una de 40*40 pixeles para que sea más sencilla de aprender por parte de la RNA creada para el proyecto no se brinda información acerca del porcentaje de eficiencia pero se da a conocer que de 10 muestras 5 se reconocen con exactitud algo muy importante para el proyecto es que el dispositivo debe estar conectado al internet porque su base de datos está almacenado ahí.

El proyecto de Chiguanco & Moreno (2011) se encarga de la traducción del lenguaje de señas a lenguaje de texto mediante el uso de visión artificial, se utiliza el software LabVIEW con sus herramientas para visión artificial, la primera etapa del proyecto se basa en la adquisición de la imagen luego de esto se busca detectar la mano del usuario y una vez obtenido la mano se

aplica el tratamiento para mejorarla cambiándola a escala de grises y con esto binarizarla, como se aplica en un entorno controlado no necesita de operaciones morfológicas, una vez obtenida la mano se crea una base de datos que contengan las letras del abecedario y para esto se utiliza el método del K-Vecino más cercano y con una muestra de 6 imágenes para cada letra las mismas que son utilizadas en la fase de entrenamiento, la traducción propuesta por el proyecto es de letra en letra y únicamente se limita a la utilización de una mano, hay que considerar que la aplicación desarrollada permite que el usuario vaya aprendiendo las letras del abecedario ya que posee dos menús uno de aprendizaje y otro de traducción, el porcentaje de erro que posee es de 17.85%.

El trabajo de Chacón Mosquera (2013) presenta el desarrollo de una interfaz para el reconocimiento del Lenguaje de Señas Ecuatoriana, la interfaz se ayuda del guante AcceleGlove el cual consta de seis acelerómetros cuya finalidad es brindar información en forma de datos para el reconocimiento de cada una de las letras del abecedario y enviar dichos datos a la interfaz para su reconocimiento se debe mencionar que no todas las señas van a ser interpretadas ya que reconoce únicamente una mano la izquierda, la interfaz consta de audio y está diseñada para interactuar de manera amigable con el usuario, cada letra posee una imagen de referencia para su aprendizaje.

El trabajo realizado por Larriva Calle (2011) se basa especialmente en el diseño de una interfaz que sea amigable y de gran ayuda para el aprendizaje del Lenguaje de Señas Ecuatoriana por parte de niños y de adultos, para el proyecto se manejan más de 600 palabras que son de uso cotidiano, se utilizan los entornos de acuerdo al tema que se desea aprender por ejemplo el hogar con sus respectivos ambientes comedor, sala, dormitorio y baño es un trabajo desarrollado con el auspicio del gobierno provincial de Pichincha y busca la libre difusión entre sus habitantes. No se brindan muchos detalles del programa utilizado para la creación de los personajes y ambientes en tercera dimensión, así como de los resultados obtenidos.

El proyecto realizado por Sivilapa Aguilera (2014) busca reconocer un cierto número de letras en el lenguaje de señas ecuatoriana ayudado de la visión artificial así como de sus herramientas, la aplicación está desarrollada en OpenCV y Qt creator, el algoritmo de reconocimiento del Lenguaje de Señas Ecuatoriana se basa en la adquisición de la imagen para luego ser umbralizada, encontrar la ubicación del rostro de la persona para ser eliminado, buscar el color de la piel de la mano para realizar el seguimiento, encontrar bordes, el polígono aproximado y luego encontrar la ROI para ser comparado con el clasificador, es importante mencionar que para la construcción del clasificador se utiliza la técnica del SVM (Máquina de Soporte Vectorial) y el KNN (K-vecino más cercano) estos dos algoritmos combinados, mejorando así en la disminución del error a un 3.33%, es utilizado en entornos controlados, el autor muestra que en poca iluminación el sistema tiene falencias.

CAPÍTULO II

VISIÓN ARTIFICIAL Y TRATAMIENTO DE IMÁGENES

El capítulo II se basa en el diseño del sistema de visión artificial, así como del tratamiento de imágenes, para esto se requiere un software que posea ciertos requerimientos como es el tipo de licencia, procesamiento de la información, uso de herramientas de visión artificial y el manejo de RNA.

El diseño del sistema de visión artificial se apoyará en las etapas primordiales para el proceso de visión artificial, tomando en consideración que el proyecto va orientado a satisfacer necesidades del aprendizaje del Lenguaje de Señas Ecuatoriana en entornos no controlados y con beneficiarios distintos. Como se observa en la figura 14 el sistema se basa en la adquisición de las imágenes a través de una cámara web tomando énfasis en la mano del usuario, para mediante técnicas apropiadas del tratamiento de imágenes extraer las características más relevantes necesarias para la implementación de una RNA que se encargue de la interpretación del alfabeto y de los números del uno al diez en Lenguaje de Señas Ecuatoriana.

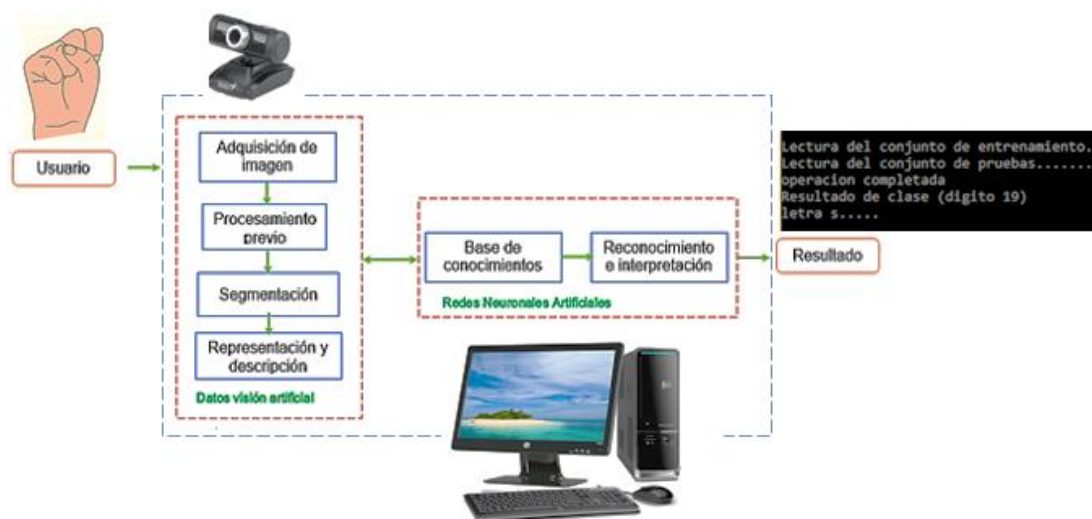


Figura 14 Etapas fundamentales para el sistema de visión artificial

Se plantea el desarrollo de una interfaz gráfica que ayude con el aprendizaje de los números del uno al diez, así como del abecedario en Lenguaje de

Señas Ecuatoriana, que sea eficaz y de libre distribución, para el propósito planteado se utiliza un computador Acer Aspire 5742G con las características que se enuncian en la tabla 4.

Tabla 4
Especificaciones técnicas del computador utilizado

Procesador	Intel Core i5 480M (2670 MHz)
Memoria RAM	8.00 GB
Sistema operativo	Windows 10 Pro *64 bits
Cámara	Acer cristal eye
Tarjeta de video	NVIDIA GeForce GT 520M (512MB)
Pantalla	LED 15.6" (1366x768) táctil
Almacenamiento	HDD 500GB (5400rpm)

2.1. Software

Existen gran variedad de programas que proporcionan herramientas para la creación de aplicaciones que utilizan tratamiento digital de imágenes y que además permiten la creación y entrenamiento de RNA. En la tabla 5 se observa la comparación entre varios softwares como lo es Matlab, LabVIEW y OpenCV. Es importante acotar que se busca un software que posea licencia libre pero la mayor parte de ellos aún se encuentran en desarrollo o no cuentan con las mismas prestaciones que OpenCV ofrece por tanto se realiza una comparación considerando los beneficios que brindan los distintos tipos de software con licencia privativa para probar la robusteza de OpenCV en comparación con dichos softwares. Como se mencionó anteriormente el software seleccionado es OpenCV gracias a que presenta gran cantidad de librerías que permiten el desarrollo de un sin número de aplicaciones completas de visión artificial. Es importante mencionar que OpenCV es compatible con todo tipo de cámaras web y gracias a su lenguaje de programación presenta mayor rapidez en el procesamiento de la información.

Tabla 5
Comparación de software

Característica	OpenCV	Matlab	LabVIEW
Tipo de licencia	Abierta	Privativa	Privativa
Lenguaje de programación	C++	C y FORTRAN	Gráfica
Procesamiento de información	Rápida	Regular	Regular
Toolkit para redes neuronales	Muy buena	Buena	buena
Herramientas para visión artificial	Muy buena	Buena	buena
Compatibilidad con cámaras web	Todas	Muy limitada	Limitada

2.2. Adquisición de la Imagen.

El primer paso en el diseño del sistema de visión artificial para la interpretación del alfabeto y números del uno al diez en Lenguaje de Señas Ecuatoriana es la captura o adquisición digitalizada de la imagen, la cual es realizada mediante la utilización de una webcam acer cristal eye que se puede visualizar en la figura 15, la webcam utilizada posee las siguientes especificaciones:

- Resolución 3.0 MP
- Rango de enfoque >26.6cm
- Tipo de sensor SOC
- Resolución de pixel 1280*1024
- Tamaño de pixel 2 μ m*2 μ m
- Tamaño de imagen 2608 μ m*2072 μ m

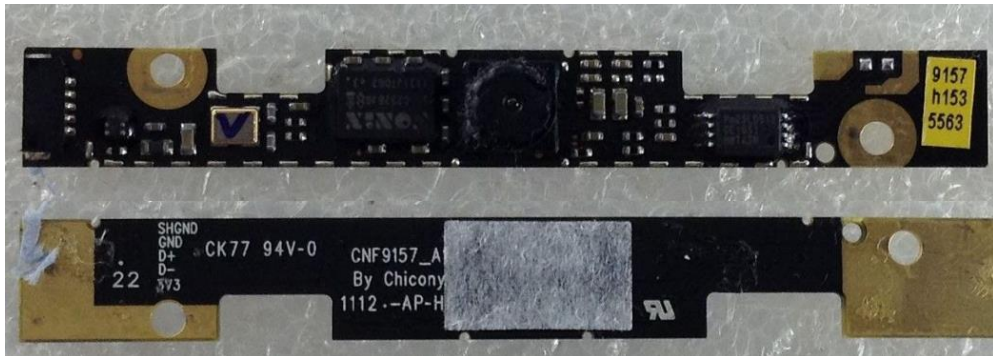


Figura 15 Acer cristal eye webcam

Fuente: (ebay, 2016)

Al inicializar la webcam del ordenador mediante el uso del software OpenCV se obtiene una imagen digitalizada similar a la mostrada en la figura 16, esta imagen posee una resolución de 640*480 píxeles.

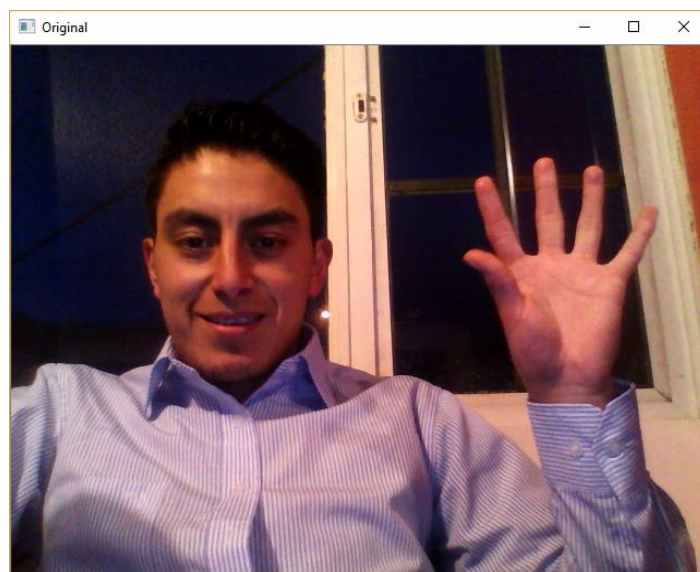


Figura 16 Imagen obtenida a través de la webcam

En la figura 17 se muestra el algoritmo utilizado en el software OpenCV para la adquisición de imágenes en forma de video.

```

Mat frame; // definimos la variable frame como imagen de tipo Mat

//Programa Principal
int main(int argc, char** argv )
{

    VideoCapture cap = VideoCapture(0); //captura del video desde la webcam

    namedWindow("webcam");//crear ventana

    while(1)
    {
        cap >> frame;//captura del video desde la webcam
        //para almacenarlo en la variable frame

        imshow("webcam", frame);//Impresión de ventana

        if(waitKey(30) >= 0) break;//Tiempo de espera

    }
    cap.release();
    return 0;
}

```

Figura 17 Algoritmo para la adquisición de imágenes en OpenCV

2.3. Procesamiento y Segmentación de la imagen.

Para la investigación propuesta se requiere extraer únicamente la mano del usuario por lo que se ha optado por aplicar la segmentación como primera etapa para luego utilizar el procesamiento de la imagen y con esto extraer la información relevante que sea de utilidad para el proyecto. se utiliza la secuencia descrita por el diagrama de bloques presentado en la Figura 18.

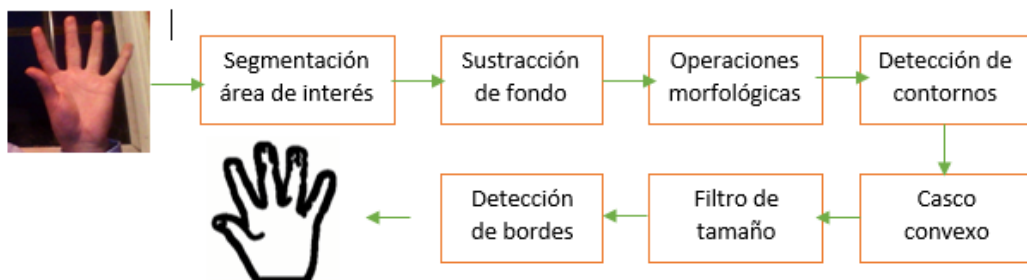


Figura 18 Procesamiento y segmentación de la imagen

2.3.1. Segmentación área de interés.

La segmentación es una etapa que permite excluir objetos que no tienen importancia en una escena y se basa en los principios de semejanza y discontinuidad. En la imagen capturada por la webcam se busca un área de interés para ser analizada que se conoce comúnmente como ROI la cual

permitirá separar un cierto número de píxeles de la imagen para convertirlos en un cuerpo entero, al aplicar una ROI en la imagen se obtiene otra representación, pero con los datos que son importantes para la investigación como se puede observar en la figura 19.

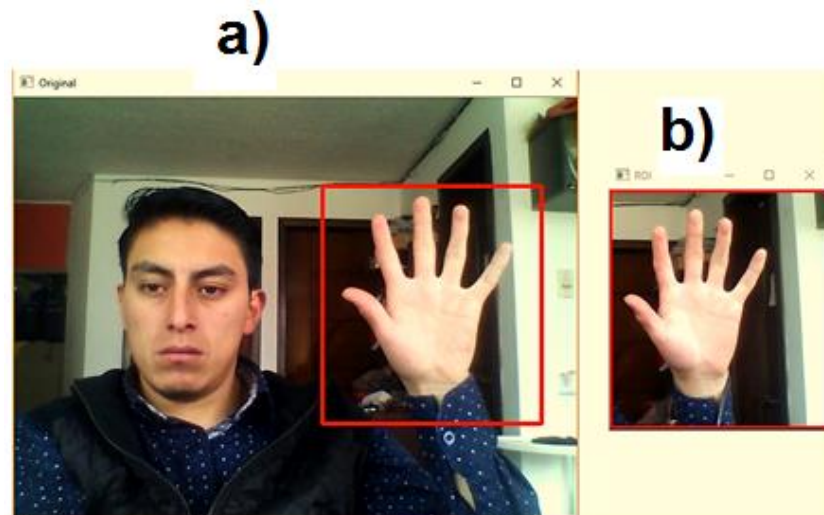


Figura 19 Extracción de la ROI

a) Imagen original b) ROI

Para la extracción de la ROI de la imagen original se utiliza el algoritmo presentado en la figura 20.

```
Rect rec1;// definimos la variable rec1
Mat roiImg;//definimos la variable roiImg como imagen de tipo Mat
rec1 = Rect(350,100,250,270);// dimensiones del rectángulo
rectangle(frame,rec1,Scalar( 0, 20, 255 ),3,8);// dibujo del rectángulo en la imagen
roiImg = frame(rec1);// extracción del ROI
imshow("ROI", roiImg);//Impresión de una ventana con la ROI
imshow("webcam", frame);//Impresión de ventana
```

Figura 20 Código para la extracción del ROI

2.3.2. Sustracción de fondo.

El fin de la sustracción de fondo es obtener una imagen en primer plano la cual goza de un movimiento, el principio de la sustracción de fondo se

basa en restar las tramas actuales con un fondo que posee la parte inmóvil de la escena como se puede observar en la figura 21.

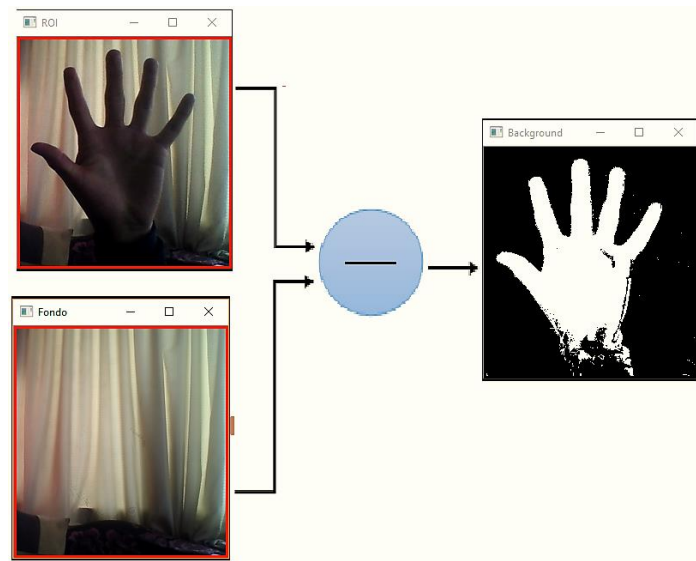


Figura 21 Sustracción de fondo

Una vez que se ha obtenido la sustracción de fondo es importante que este se acondicione a los cambios del medio y se vaya actualizando cada cierto tiempo para obtener la imagen de primer plano en este caso la mano del usuario con el menor porcentaje de ruido posible es por eso que se utiliza un método de mezclas Gausianas el algoritmo utilizado en la sustracción de fondo se observa en la figura 22.

```

Mat fore; // definimos la variable fore como imagen de tipo Mat
Mat back; // definimos la variable back como imagen de tipo Mat
//Programa Principal
int main(int argc, char** argv )
{
    VideoCapture cap = VideoCapture(0); //captura del video desde la webcam

    cv::BackgroundSubtractorMOG2 bg; //sustracción de fondo

    bg.set ("nmixtures",3);//filtro gaussiano
    bg.setBool("detectShadows", false);// no detección de sombras

    namedWindow("webcam");//crear ventana

    while(1)
    {
        cap >> frame;//captura del video desde la webcam
        //para almacenarlo en la variable frame

        //backgroundsubtractor mog

        bg.operator()(frame, fore); // parte móvil de la escena
        bg.getBackgroundImage (back);// parte inmóvil de la escena

        imshow("Móvil", fore);//Impresión de una ventana
        //con la parte móvil de la escena
    }
}

```

Figura 22 Código para sustracción de fondo

2.3.3. Operaciones morfológicas.

Las operaciones morfológicas normalmente son aplicadas en imágenes binarias, cuya finalidad es realzar características importantes de una imagen mientras elimina el ruido. La imagen de la mano del usuario separada de la escena será limpiada del ruido aplicando las operaciones morfológicas a continuación se enuncian algunas de estas operaciones.

- **Erosión:** elimina ruido, pero a su vez adelgaza el área de estudio es decir reduce las regiones de la imagen con valor 1 o conocido como color blanco (figura 23).

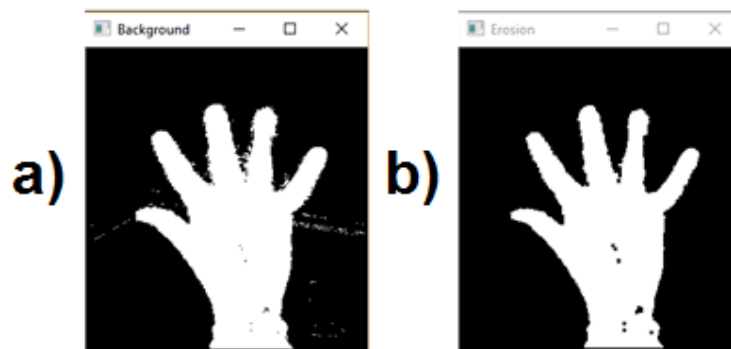


Figura 23 Erosión de la imagen original

a) Imagen con sustracción de fondo b) Imagen erosionada

- **Dilatación:** no elimina el ruido, incrementa las zonas de la imagen con valor 1 (figura 24).

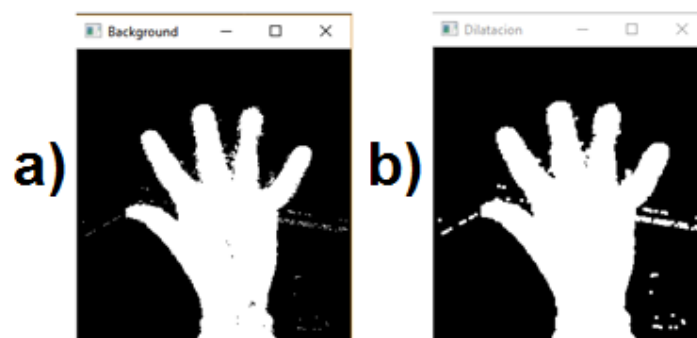


Figura 24 Dilatación de la imagen original

a) Imagen con sustracción de fondo b) Imagen dilatada

- **Apertura:** consiste en aplicar erosión y luego dilatación en una imagen (figura 25).

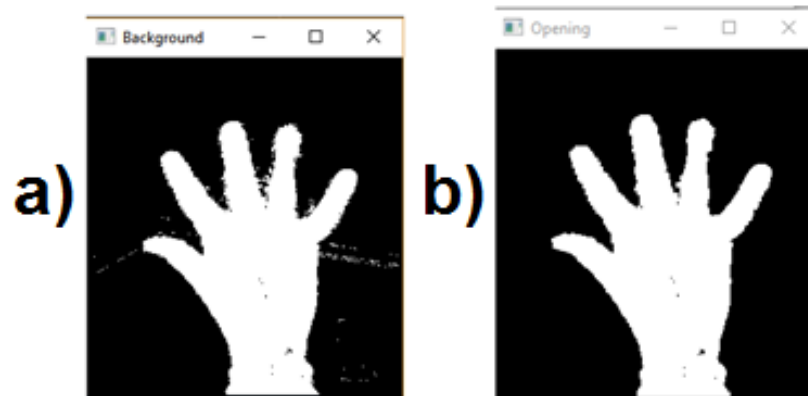


Figura 25 Apertura en la imagen original

a) Imagen con sustracción de fondo b) Imagen aplicando cierre

- **Cierre:** consiste aplicar dilatación y luego erosión a una imagen (figura 26).

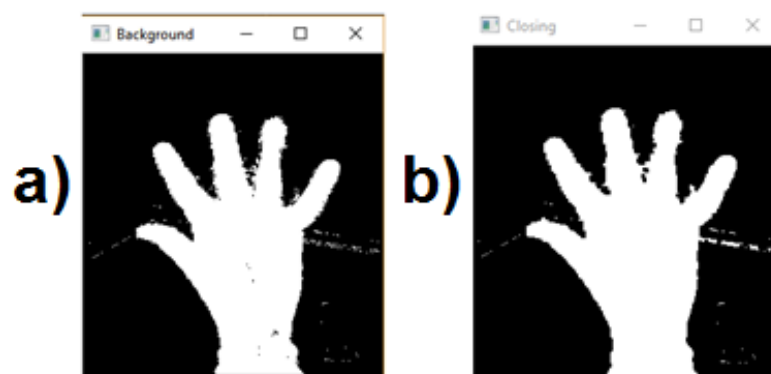


Figura 26 Cierre de la imagen original

a) Imagen con sustracción de fondo b) Imagen aplicada apertura

Aplicando operaciones morfológicas a la imagen segmentada se obtiene una imagen más limpia con un porcentaje de ruido mucho menor como se aprecia en la figura 27.



Figura 27 Operaciones morfológicas a la imagen original

a) Imagen con sustracción de fondo b) Imagen limpia

Los códigos que permiten realizar operaciones morfológicas en el software OpenCV se observan en la figura 28.

```
morphologyEx (fore, fore, cv::MORPH_OPEN, cv::Mat ()); //operación morfológica de apertura
morphologyEx (fore, fore, cv::MORPH_CLOSE, cv::Mat ()); //operación morfológica de cierre
```

Figura 28 Códigos para operaciones morfológicas

2.3.4. Detección de contornos.

Se entiende por contorno aquella curva que combina todos los puntos continuos alrededor de una frontera comúnmente se utilizan imágenes binarizadas para lograr mejores resultados. En la imagen sin ruido se procede a encontrar los contornos que poseen el borde exterior del objeto de estudio, proporcionando información de su forma como se puede visualizar en la figura 29 se encierra en el contorno la mano del usuario.



Figura 29 Contorno de la mano

En la figura 30 se visualiza el código que permite la búsqueda y representación de los contornos de la imagen.

```
vector<vector<Point> > contours;//vector de puntos para los contornos
vector<Vec4i> hierarchy; // vector de salida opcional

//busqueda de contornos en la imagen
findContours( fore, contours, hierarchy, CV_RETR_TREE, CV_CHAIN_APPROX_SIMPLE, Point(0,0));

for( int i = 0; i < contours.size(); i++ )
{
// dibujo del contorno en la imagen original
Scalar color = Scalar( rng.uniform(0, 255), rng.uniform(0,255), rng.uniform(0,255) );
drawContours( roiImg, contours, -1, color, 2, 8, hierarchy, 0 );
}
```

Figura 30 Código para la búsqueda y dibujo de los contornos de la imagen

2.3.5. Casco convexo

Lo que busca el casco convexo es encontrar el contorno que encierra una región pero que se adapta a esta por completo. El casco convexo es utilizado para formar la máscara que contendrá la imagen de la mano, se puede visualizar esta representación en la figura 31.



Figura 31 Casco convexo alrededor de la mano del usuario

El código para la creación de un casco convexo se observa en la figura 32.

```

vector<vector<Point> > hulls(1); // vector de puntos para el casco convexo
for( int i = 0; i < contours.size(); i++ )
{
convexHull(Mat(contours[0]),hulls[0],false); // casco convexo
// dibujo del casco convexo en la imagen original
Scalar color = Scalar( rng.uniform(0, 255), rng.uniform(0,255), rng.uniform(0,255) );
drawContours(roiImg,hulls,-1,cv::Scalar(0,255,0),2);// dibujo del
}

```

Figura 32 Código para casco convexo

2.3.6. Filtro de tamaño

Para el etiquetado de la imagen se procede a utilizar un filtro de objeto de acuerdo a su tamaño con el fin de eliminar un porcentaje de ruido que pueda aparecer en la imagen, así como también la reducción del número de objetos presentes en la misma ya que existen cuerpos de determinados tamaños que no interesan en la escena, se utiliza un área de contorno para eliminar pixeles de menor tamaño de la escena ya que si poseen un valor menor a esta se descartan, aplicando el principio del filtro de tamaño a la imagen de la figura 32 se logra limpiar y obtener un mejor resultado en la obtención del contorno y del casco convexo como se aprecia en la figura 33.

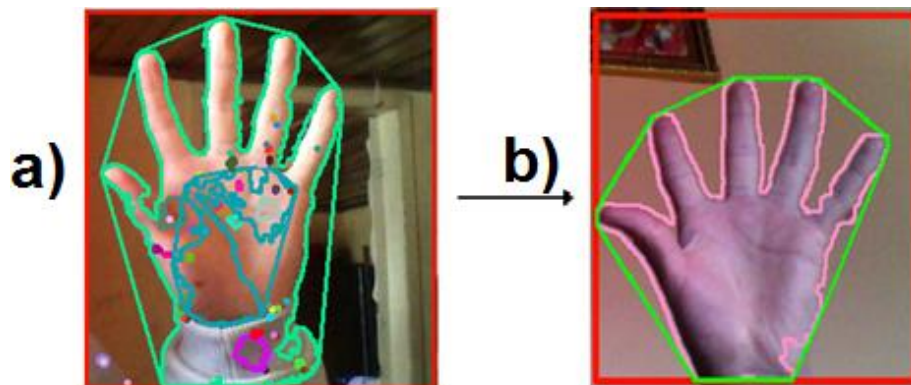


Figura 33 Filtro de tamaño para obtención de la mano

a) Imagen sin filtro de tamaño b) Imagen con filtro de tamaño

Para la realización del filtro de tamaño se toma en cuenta el área que servirá como umbral en la imagen el código empleado en el filtro de tamaño se muestra en la figura 34.

```

for( int i = 0; i < contours.size(); i++ )
{
    if(contourArea(contours[i])>=3000)
    {
        tcontours.push_back(contours[i]);
        convexHull(Mat(tcontours[0]),hulls[0],false);// casco convexo
        // dibujo del contorno y casco convexo en la imagen original
        Scalar color = Scalar( rng.uniform(0, 255), rng.uniform(0,255), rng.uniform(0,255)
        drawContours( roiImg, tcontours, -1, color, 2, 8, hierarchy, 0 );
        drawContours(roiImg,hulls,-1,cv::Scalar(0,255,0),2);
    }
}

```

Figura 34 Código para el filtro de tamaño

2.3.7. Detección de bordes

Tiene como finalidad reconocer los cambios drásticos en el brillo de una imagen o dicho en otras palabras buscar las discontinuidades que esta posea, la detección de bordes se realiza siguiendo la siguiente secuencia.

- **Máscara.** Es el resultado de aplicar una operación AND entre dos imágenes una que posea sus valores de pixeles en todo su contorno iguales a 0 (negro) y la otra que posea la información de la imagen a recortar, para el caso de la investigación se obtiene una máscara que contiene su perímetro negro pero el contorno de la mano rellena y se realiza la operación AND con la imagen de la ROI transformada a escala de grises como se observa en la figura 35.

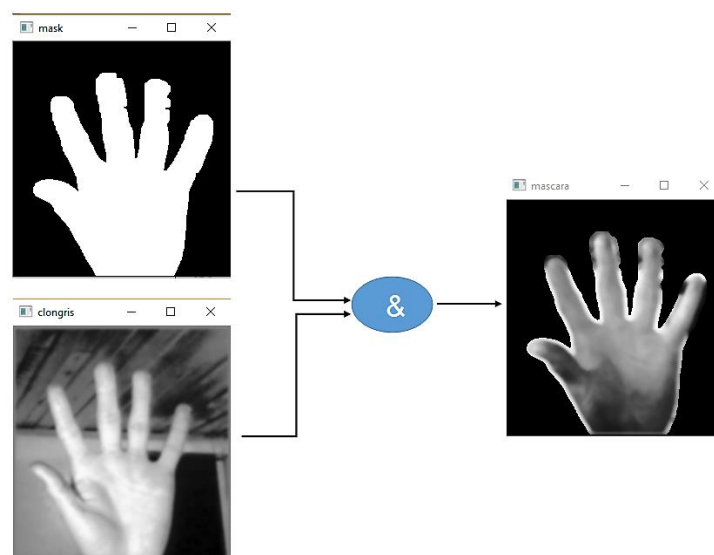


Figura 35 Máscara de la mano

La máscara se ha obtenido a partir del código mostrado en la figura 36.

```

Mat mask;// definimos la variable mask como imagen de tipo Mat
clon=roiImg.clone(); // clon de la ROI
cvtColor(clon, clongris, CV_BGR2GRAY);// transformacion a escala de grises
mask = Mat::zeros(clongris.rows,clongris.cols,CV_8U);//utilizacion de un solo canal RGB
for( int i = 0; i < contours.size(); i++ )
{
    if(contourArea(contours[i])>=3000)
    {
        tcontours.push_back(contours[i]);
        //dibujo del contorno y relleno de la mascara en color negro
        drawContours( mask, tcontours,-1,255, CV_FILLED, 8);
    }
}
mask = mask&clongris;

```

Figura 36 Código para encontrar una máscara

- **Bordes.** Una vez obtenida la mano separada del entorno se procede a encontrar bordes, tomando en cuenta un umbral tanto máximo como mínimo, para detectar la mayor cantidad de particularidades relevantes para la investigación, se hace uso de las herramientas de OpenCV y se ha utilizado un detector Canny el cual está ya acondicionado para la detección de bordes de manera eficiente como se puede visualizar en la figura 37 recordando que con los bordes estamos buscando la diferencia de características entre un pixel y otro es decir se busca la pendiente.

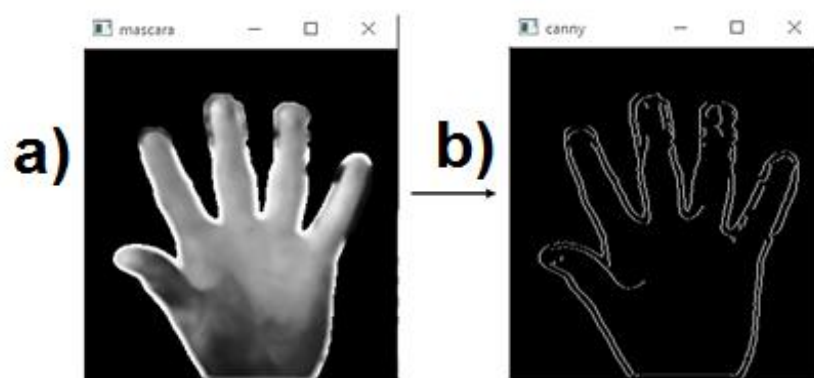


Figura 37 Detección de bordes

a) Máscara b) Canny

- **Inverso e identificación de contornos.** Se requiere invertir los pixeles de la imagen que contienen los bordes y realzar los mismos para esto se utiliza la búsqueda de contornos, así como de la

inversión de color, aplicando esto a la imagen original se halla otra igual a la mostrada en la figura 38.

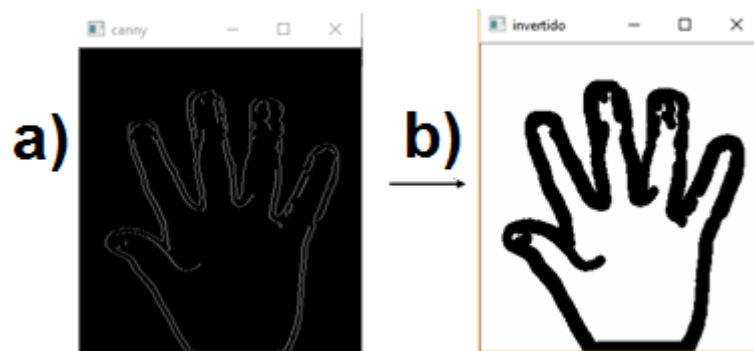


Figura 38 Realce de características e inversión de color

a) Canny b) Inversión de color y búsqueda de contornos

La imagen final que se ha obtenido de la detección de bordes es la que se utilizará en la fase de descripción y extracción de características, como también servirá para la etapa de reconocimiento del mismo.

2.4. Descripción y Extracción de características.

Para el reconocimiento de un objeto se debe tener un modelo o descripción del mismo. Los descriptores deben ser indiferentes a los cambios de orientación, forma y localización del objeto, pero deben ser capaces de distinguir un objeto de otro, esto se logra gracias a la extracción de características las mismas que permiten disminuir el exceso de información y representar de mejor manera al patrón original a través de un vector de características.

De acuerdo con (Sobrado Malpartida, 2011) se deben considerar ciertos parámetros en la extracción de características como son:

- **La fiabilidad** es decir objetos de la misma clase presentan dispersiones pequeñas.
- **La discriminación** cada clase posee un vector de características diferente.

- **La invariancia** no se afecta frente a modificaciones que sufra el objeto como puede ser su escalado, su rotación y su traslación.
- **La dimensionalidad** este aspecto es muy importante al momento de trabajar con redes neuronales artificiales ya que al obtener un vector de características con una dimensionalidad reducida a comparación con la del patrón original beneficia a la RNA ya que el tiempo de entrenamiento será mucho menor.

2.4.1. Reducción de la dimensión de datos

En muchos casos los clasificadores dependen de la dimensionalidad de los patrones llevando una relación directa con el coste computacional es por esto que se deben clasificar los datos ya que existirán algunos que no brinden la información necesaria para la investigación, la reducción de datos se realiza mediante dos procesos que se dan a conocer a continuación:

- **Recorte de imagen** se realiza un corte de la imagen principal tomando especialmente el área que contenga información importante para la investigación en este caso únicamente el área que contiene la mano del usuario como se observa en la figura 39.

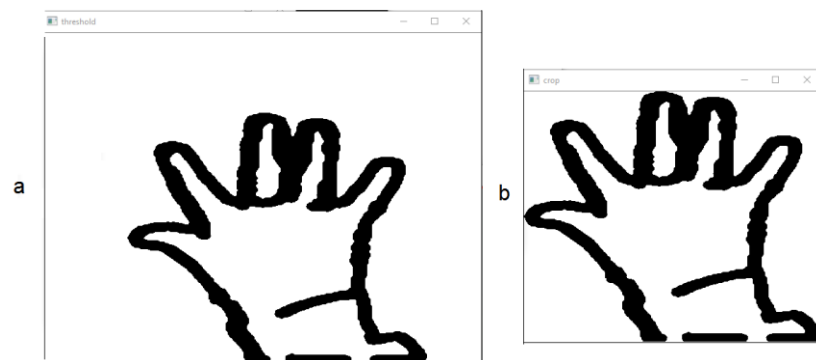


Figura 39 Recorte de imagen

a) Imagen principal b) Imagen recortada

- **Reducción de la imagen** la reducción de la imagen es algo muy importante para el rendimiento computacional y para el

reconocimiento de cada clase ya que al tener una imagen con menor cantidad de píxeles el almacenamiento e interpretación de la misma será de manera rápida, se requiere escalar la imagen a una que no distorsione la información pero que brinde los datos necesarios para su identificación (figura 40).

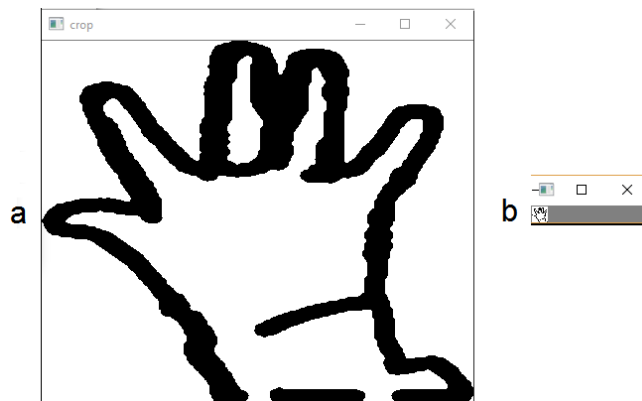


Figura 40 Reducción de imagen

a) Imagen recortada b) Imagen reducida

2.4.2. Descriptores de contorno

Poseen información binaria del contorno de la imagen entre los principales descriptores de contornos encontramos los códigos de cadena, representación polar, aproximaciones poligonales, esqueletización entre otros.

- **Códigos de cadena.** Para él proyecto se adopta el método de los códigos de cadena ya que estos permiten seguir el contorno de una imagen a través de todas sus direcciones y encontrar una estructura de datos que simboliza el borde del área a ser analizada, en la investigación se obtendrá una cadena de 256 datos ya que la imagen a ser codificada posee 256 píxeles, una vez aplicado el principio de códigos de cadena se obtiene un documento de texto que contiene la información de la imagen pero en un dialecto de fácil interpretación por parte del clasificador. Se aprecia en la figura 41 la cadena de caracteres con

valores de 1 y 0 a partir de una imagen que contiene únicamente el contorno y ciertas características importantes de la mano.

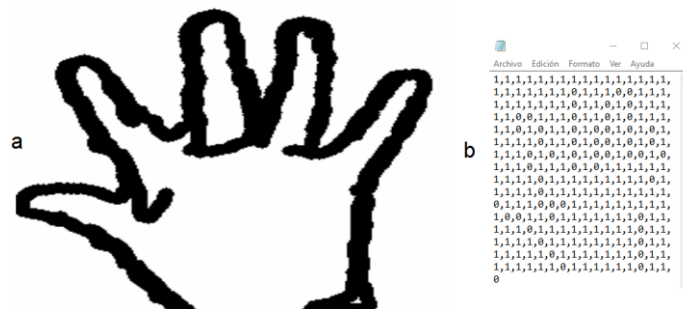


Figura 41 Códigos de cadena

a) Imagen binarizada b) Estructura de datos

Es importante mencionar que el reconocimiento tanto de los números como del alfabeto en Lenguaje de Señas Ecuatoriana se la realiza mediante la utilización de RNA específicamente utilizando el reconocimiento de caracteres ópticos los mismo que fueron parte principal en la construcción del clasificador y que se detalla en el capítulo III.

CAPÍTULO III

REDES NEURONALES ARTIFICIALES

El capítulo III se basa en el diseño de las Redes Neuronales Artificiales encargadas del aprendizaje de los números del uno al diez, así como del alfabeto en Lenguaje de Señas Ecuatoriana, se crean dos clasificadores uno encargado del reconocimiento de los números del uno al diez y el otro encargado del alfabeto, cada red neuronal artificial consta de la misma topología, pero se diferencian únicamente en sus salidas y en su capa oculta.

El software de OpenCV ofrece grandes prestaciones a la hora de realizar aplicaciones que requieran del uso de Redes Neuronales Artificiales para esto utiliza perceptrones multicapa la cual es simbolizada por las siglas MLP que provienen del inglés, multi-layer perceptrons, MLP consta de capa de entrada, capa de salida y una o más capas ocultas (opencv dev team, 2014).

La función de activación que maneja OpenCV de manera óptima para Redes Neuronales Artificiales es la sigmoide simétrica la misma que está representada por la ecuación 10.

$$f(x) = \frac{\beta * (1 - e^{-\alpha x})}{1 + e^{-\alpha x}}$$

Ecuación 10 Función sigmoide simétrica

A continuación, se muestra en la figura 42 la función sigmoide estándar con valores de $\beta=1$ y $\alpha=1$.

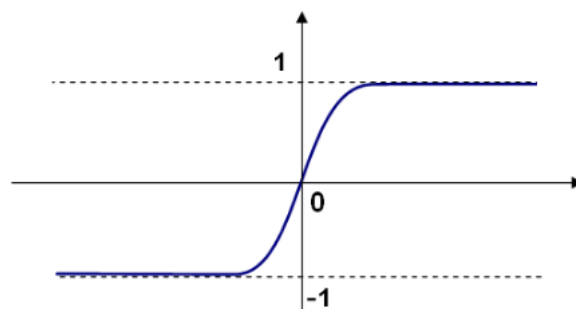


Figura 42 Función sigmoide estándar

Es primordial indicar que las muestras para el entrenamiento que se utilizan en cada uno de los clasificadores fueron tomadas a partir de personas que conocen muy bien la Lengua de Señas Ecuatoriana las mismas que pertenecen al FENASEC, así como también son parte del Instituto Nacional de Audición y Lenguaje - INAL

3.1. Red Neuronal Artificial para clasificación de números

Para la clasificación de los números del uno al diez en Lenguaje de Señas Ecuatoriana se requiere seguir una secuencia de pasos tanto para el procesamiento, el entrenamiento y el reconocimiento de cada uno de los caracteres que conforman cada una de las clases, para ello se crea un clasificador que contenga la información de los números en LSEC.

3.1.1. Clasificador

Para la creación del clasificador encargado de la interpretación de los números del uno al diez en Lenguaje de Señas Ecuatoriana se toma como referencia el modelo creado por Raj S. (2013) el cual se basa en tres módulos el de pre-procesamiento, el de entrenamiento y el de clasificación, el mismo que se puede observar en la figura 43.

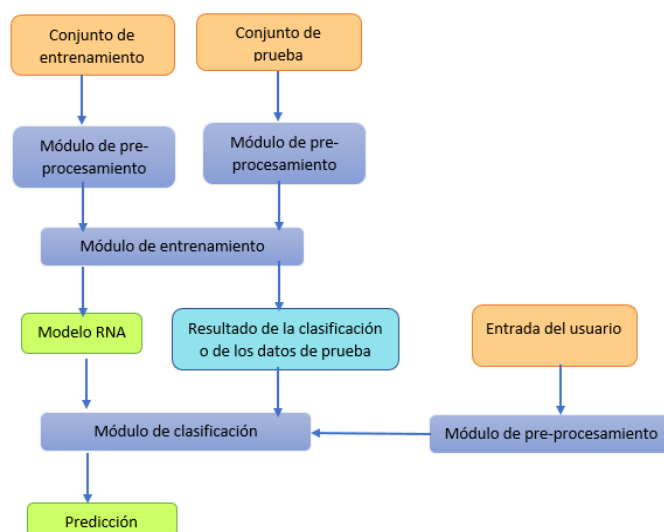


Figura 43 Modelo para creación de clasificadores

Fuente:(Raj S., 2013)

Etapa de pre-procesamiento

El módulo de pre-procesamiento es el encargado de convertir las imágenes de entrada en un conjunto de datos con formato que la red neuronal artificial pueda entender con claridad para esto se utiliza la secuencia descrita en el apartado 2.4 dando como resultado dos archivos de texto uno que servirá para el entrenamiento y otro como conjunto de pruebas para la validación de la RNA, en la figura 44 se observan las carpetas que contienen el conjunto de datos utilizados para el entrenamiento de la red neuronal artificial.

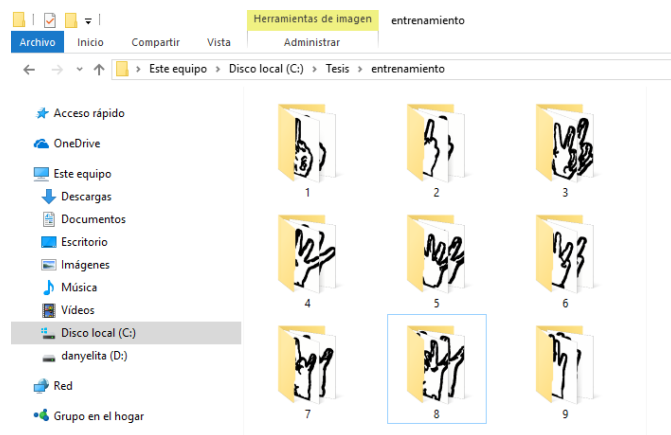


Figura 44 Carpetas que contienen las imágenes de entrada

Los documentos que contienen la información de las imágenes de entrada una vez convertidas en valores de 0 y 1 son representados en la figura 45. Es importante mencionar que estos documentos son utilizados en la etapa de entrenamiento del clasificador.

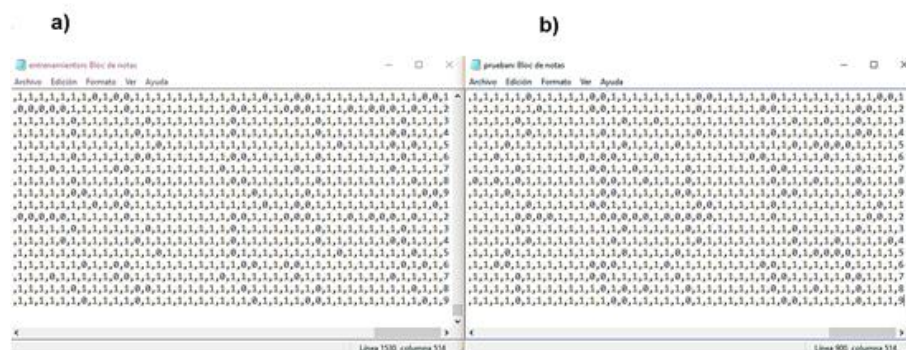


Figura 45 Información de las imágenes de entrada en 0 y 1

a) Documento para el entrenamiento b) Documento para prueba

Etapa de entrenamiento

Para la clasificación de los números se utiliza una red neuronal artificial del tipo perceptrón multicapa cuya estructura se muestra en la figura 46, compuesta por una sola capa oculta y con una función de activación sigmoide simétrica, posee 256 entradas que provienen de una ROI de 16*16 pixeles en blanco y negro que contiene el contorno de la mano formando el número en Lenguaje de Señas Ecuatoriana. 16 neuronas forman la capa oculta y 10 neuronas como salida las mismas que serán tomadas como las clases uno para cada número. El modelo de la red neuronal es tomada del trabajo expuesto por Raj S. (2013).

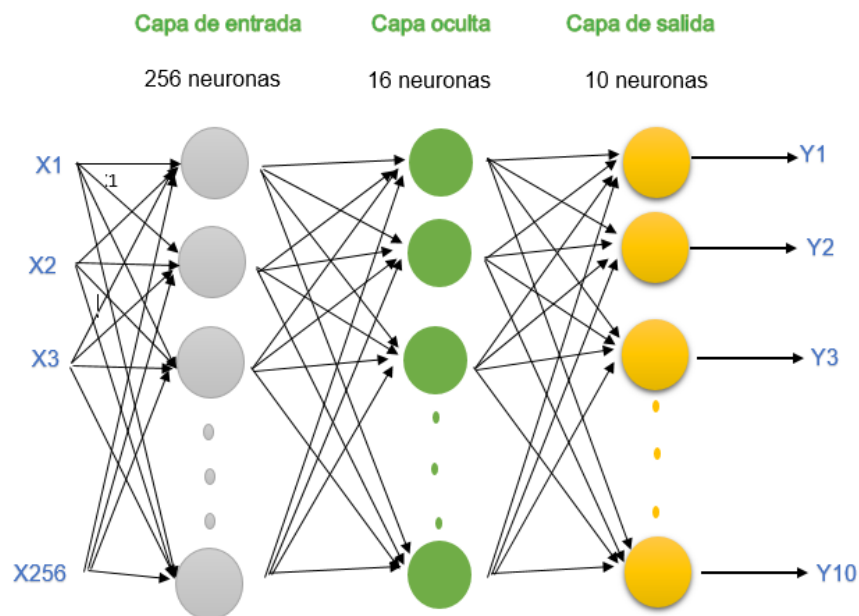


Figura 46 Red neuronal artificial para clasificación de números

Se crea un conjunto de entrenamiento de 1530 muestra, 170 muestras por dígito y un conjunto para pruebas de 900 muestras para validación de la red (figura 47).

```
#define TRAINING_SAMPLES 1530 //Número de muestras en la formación datos
#define ATTRIBUTES 256 //Número de pixeles por muestra.16X16
#define TEST_SAMPLES 900 //Número de muestras en el conjunto de datos de prueba
#define CLASSES 10 //Número de etiquetas distintas clases.
```

Figura 47 Datos utilizados para el entrenamiento de la RNA.

La salida de la red neuronal artificial pretende reconocer cada número en Lenguaje de Señas Ecuatoriana delimitado del 1 al 10, de las 10 neuronas tomadas en la capa de salida cada una tendrá información de cada número o de cada clase.

La función de activación que se utiliza para el entrenamiento de la red neuronal artificial es la sigmoidea simétrica con coeficientes que OpenCV maneja con eficiencia esto se observa en la figura 48.

```
CvANN_MLP nnetwork(layers, CvANN_MLP::SIGMOID_SYM,0.6,1);
```

Figura 48 Código para la función de activación

El entrenamiento de la red neuronal artificial cumple con los siguientes parámetros: se termina el entrenamiento después de 1000 iteraciones o si existe un cambio muy pequeño en la red es decir los pesos tendrán valores por debajo de 0.00001, se utiliza la retropropagación con coeficientes estándares que brinda el software OpenCV (figura 49).

```
CvANN_MLP_TrainParams params(
    // terminar el entrenamiento después de 1000
    // iteraciones o un cambio muy pequeño en la red
    // pesos por debajo del valor especificado
    cvTermCriteria(CV_TERMCRIT_ITER+CV_TERMCRIT_EPS, 1000, 0.000001),
    // uso de retropropagación para el entrenamiento
    CvANN_MLP_TrainParams::BACKPROP,
    // coeficientes para el entrenamiento de retropropagación
    0.1,
    0.1);
```

Figura 49 Código utilizado para el entrenamiento de la RNA

Para que la RNA se entrene son necesarias 791 (figura 50), a partir de este punto los valores de los pesos serán inferiores al valor ingresado como parámetro de entrenamiento.

```

Seleccionar C:\Qt\Tools\QtCreator\bin\qtcreator_process_stub.exe
Using training dataset
Training iterations: 791
Testing Sample 0 -> class result (digit 1)
Testing Sample 1 -> class result (digit 2)
Testing Sample 2 -> class result (digit 3)
Testing Sample 3 -> class result (digit 4)
Testing Sample 4 -> class result (digit 5)
Testing Sample 5 -> class result (digit 6)
Testing Sample 6 -> class result (digit 7)
Testing Sample 7 -> class result (digit 8)
Testing Sample 8 -> class result (digit 9)
Testing Sample 9 -> class result (digit 1)
Testing Sample 10 -> class result (digit 2)
Testing Sample 11 -> class result (digit 3)
Testing Sample 12 -> class result (digit 4)
Testing Sample 13 -> class result (digit 5)
Testing Sample 14 -> class result (digit 6)
Testing Sample 15 -> class result (digit 7)
Testing Sample 16 -> class result (digit 8)
Testing Sample 17 -> class result (digit 9)
Testing Sample 18 -> class result (digit 1)
Testing Sample 19 -> class result (digit 2)
Testing Sample 20 -> class result (digit 3)
Testing Sample 21 -> class result (digit 4)
Testing Sample 22 -> class result (digit 5)
Testing Sample 23 -> class result (digit 6)
Testing Sample 24 -> class result (digit 7)
Testing Sample 25 -> class result (digit 8)

```

Figura 50 Entrenamiento de la RNA

La figura 51 muestra la comprobación de la RNA y se hace uso del test de prueba dado como resultado un excelente entrenamiento con el reconocimiento de las 900 muestras sin error alguno.

```

C:\Qt\Tools\QtCreator\bin\qtcreator_process_stub.exe
Testing Sample 895 -> class result (digit 5)
Testing Sample 896 -> class result (digit 6)
Testing Sample 897 -> class result (digit 7)
Testing Sample 898 -> class result (digit 8)
Testing Sample 899 -> class result (digit 9)

Results on the testing dataset
Correct classification: 900 (100%)
Wrong classifications: 0 (0%)

```

	1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	0	0	0	0
1	0	100	0	0	0	0	0	0	0
2	0	0	100	0	0	0	0	0	0
3	0	0	0	100	0	0	0	0	0
4	0	0	0	0	100	0	0	0	0
5	0	0	0	0	0	100	0	0	0
6	0	0	0	0	0	0	100	0	0
7	0	0	0	0	0	0	0	100	0
8	0	0	0	0	0	0	0	0	100
9	0	0	0	0	0	0	0	0	0

```

Press <RETURN> to close this window...

```

Figura 51 Comprobación del entrenamiento de la RNA

Para confirmar que las 16 neuronas tomadas en la capa oculta son las que requiere la red neuronal se realiza una prueba con 100 imágenes que contienen los números en LSEC las mismas que la RNA nunca observó en su etapa de entrenamiento, verificando que el reconocimiento sea aceptable es decir sobrepase de un 86% de exactitud ya que todo programa posee error se busca que este sea razonable, la tabla 6 muestra la verificación de la RNA.

Tabla 6
Verificación del error en la clasificación de los números en LSEC

N°- de neuronas en la capa oculta	Muestra	Tipo	imágenes reconocidas	% de clasificación	%error
16	100	números	92	92 %	8 %

De la muestra de 100 imágenes 92 son reconocidas con efectividad representando una clasificación del 92% y un error del 8%.

3.2. Red Neuronal Artificial para clasificación de las letras del Alfabeto.

Para la clasificación de las letras del alfabeto en Lenguaje de Señas Ecuatoriana se requiere seguir la misma secuencia de pasos que se utiliza para formar el clasificador de números tanto para el procesamiento, el entrenamiento y el reconocimiento de cada uno de los caracteres que conforman cada una de las clases.

3.2.1. Clasificador

Para la creación del clasificador encargado de la interpretación de las letras del abecedario en Lenguaje de Señas Ecuatoriana se toma como referencia el mismo modelo creado por Raj S. (2013) usado en la clasificación de los números.

Etapas de pre-procesamiento

Se emplea la secuencia descrita en el apartado 2.4 que da como resultado dos archivos de texto uno que servirá para el entrenamiento y otro como conjunto de pruebas para la validación de la red neuronal artificial, en la figura 52 se observan las carpetas que contienen el conjunto de datos utilizados para el entrenamiento de la RNA.

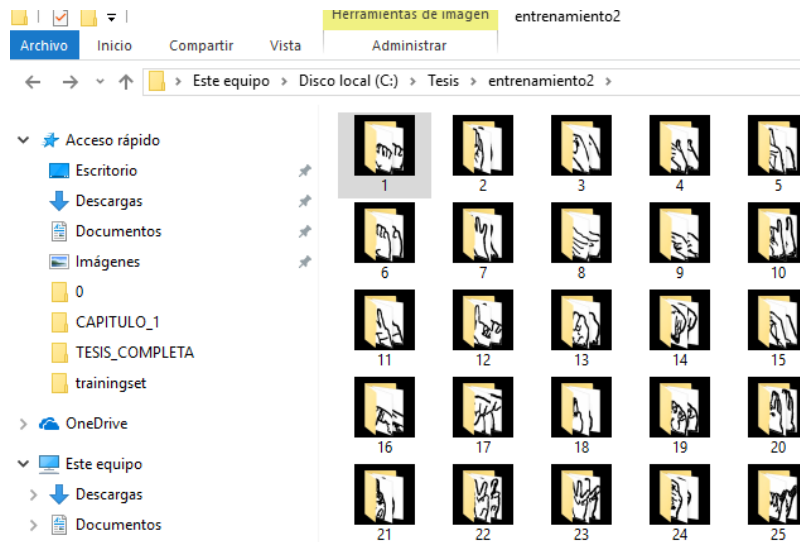


Figura 52 Carpetas que contienen las imágenes de entrada

Los documentos que contienen la información de las imágenes de entrada una vez convertidas en valores de 0 y 1 se visualizan en la figura 53. Estos datos son utilizados para el entrenamiento del clasificador.

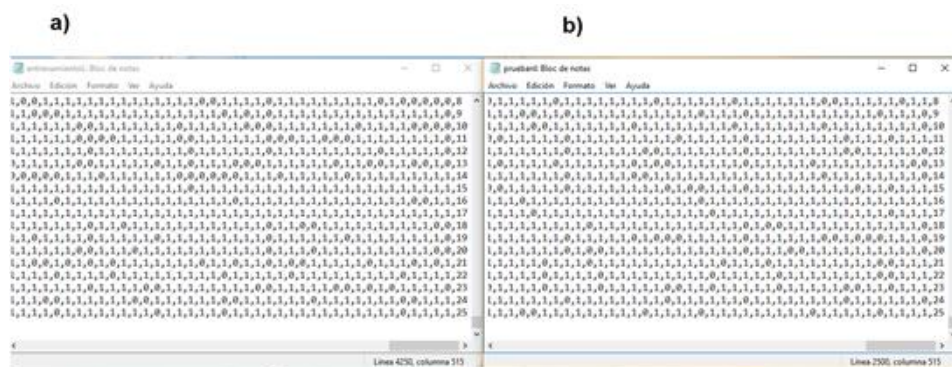


Figura 53 Información de las imágenes de entrada en 0 y 1

a) Documento para el entrenamiento b) Documento para prueba

Etapa de entrenamiento

Para la clasificación de las letras del abecedario se utiliza una RNA del tipo perceptrón multicapa cuya estructura se muestra en la figura 54, compuesta por una sola capa oculta y con una función de activación sigmoide simétrica, posee 256 entradas que provienen de la ROI de 16*16 pixeles que contienen el contorno de la mano formando la letra del

abecedario en Lenguaje de Señas Ecuatoriana. n neuronas forman la capa oculta y 26 neuronas como salida las mismas que serán tomadas como las clases uno para cada letra. El modelo de la RNA es tomada del trabajo expuesto por Raj S. (2013).

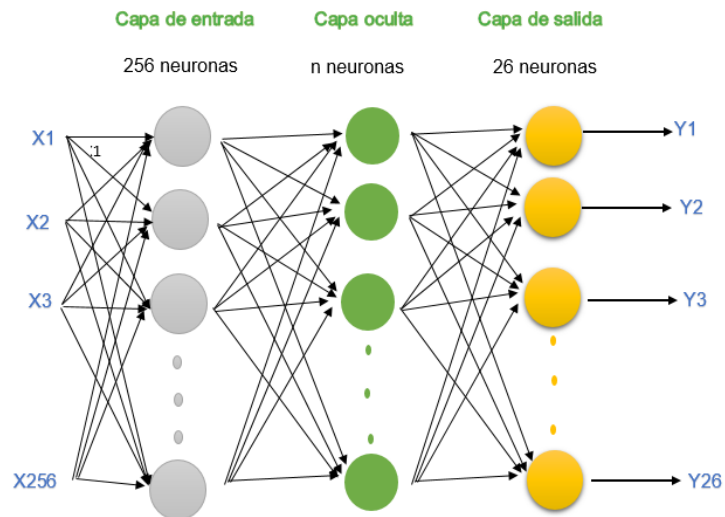


Figura 54 RNA para clasificación de las letras del alfabeto

Se crea un conjunto de entrenamiento de 4250 muestra, 170 muestras por dígito y un conjunto para pruebas de 2500 muestras para validación de la red como se observa en la figura 55.

```
#define TRAINING_SAMPLES 4250 //Número de muestras en la formación datos
#define ATTRIBUTES 256 //Número de pixeles por muestra.16X16
#define TEST_SAMPLES 2500 //Número de muestras en el conjunto de datos de prueba
#define CLASSES 26 //Número de etiquetas distintas clases.
```

Figura 55 Parámetros para el entrenamiento de la RNA.

La salida de la RNA pretende reconocer cada letra del alfabeto en Lenguaje de Señas Ecuatoriana, de las 26 neuronas tomadas en la capa de salida cada una tendrá información de cada letra del abecedario o de cada clase.

Al igual que para el entrenamiento de los números para las letras se utiliza la función de activación sigmoidea simétrica con los coeficientes que OpenCV maneja (figura 48).

El entrenamiento de la red neuronal artificial cumple con los mismos parámetros que se utilizaron en el entrenamiento de los números, estos se pueden observar en la figura 48.

Al no conocer el número de neuronas que se utilizan en la capa oculta se emplea el método de prueba y error para localizar el número necesario para que la RNA sea lo bastante robusta y el error sea aceptable es decir no sobrepase de un 10%, se utilizan 200 imágenes que contienen las letras del alfabeto en Lenguaje de Señas Ecuatoriana las mismas que la red nunca analizo en su etapa de entrenamiento, se observa en la tabla 7 el empleo del método.

Tabla 7
Verificación del error en la clasificación de las letras del abecedario en LSEC

N°- de neuronas en la capa oculta	Muestra	Tipo	imágenes reconocidas	% de clasificación	%error
16	200		194	97 %	3 %
25	200	Letras	196	98 %	2 %
34	200		197	98,5 %	1,5 %

En la tabla 7 se aprecia claramente que la RNA al utilizar 34 neuronas en su capa oculta presenta menor porcentaje de error en comparación con sus otros dos competidores no se utilizan mayor número de neuronas en esta capa ya que se corre con el riesgo de que la RNA se sobreentrene y entregue resultado erróneos, al utilizar 34 neuronas en la capa oculta se obtiene que para el entrenamiento son necesarias 912 iteraciones (figura 56), a partir de este punto los valores de los pesos serán inferiores al valor ingresado como parámetro de entrenamiento.

```

C:\Qt\Tools\QtCreator\bin\qtcreator_process_stub.exe
Using training dataset
Training iterations: 912
Testing Sample 0 -> class result (digit 1)
Testing Sample 1 -> class result (digit 2)
Testing Sample 2 -> class result (digit 3)
Testing Sample 3 -> class result (digit 4)
Testing Sample 4 -> class result (digit 5)
Testing Sample 5 -> class result (digit 6)
Testing Sample 6 -> class result (digit 7)
Testing Sample 7 -> class result (digit 8)
Testing Sample 8 -> class result (digit 9)
Testing Sample 9 -> class result (digit 10)
Testing Sample 10 -> class result (digit 11)
Testing Sample 11 -> class result (digit 12)
Testing Sample 12 -> class result (digit 13)
Testing Sample 13 -> class result (digit 14)
Testing Sample 14 -> class result (digit 15)
Testing Sample 15 -> class result (digit 16)
Testing Sample 16 -> class result (digit 17)
Testing Sample 17 -> class result (digit 18)
Testing Sample 18 -> class result (digit 19)
Testing Sample 19 -> class result (digit 20)
Testing Sample 20 -> class result (digit 21)
Testing Sample 21 -> class result (digit 22)
Testing Sample 22 -> class result (digit 23)
Testing Sample 23 -> class result (digit 24)
Testing Sample 24 -> class result (digit 25)
Testing Sample 25 -> class result (digit 1)

```

Figura 56 Iteraciones para el entrenamiento de la RNA

La figura 57 presenta los resultados de la comprobación de RNA, haciendo uso del test de prueba dado como resultado un entrenamiento magnifico con el reconocimiento de las 2500 muestras sin error alguno.

```

C:\Qt\Tools\QtCreator\bin\qtcreator_process_stub.exe
Testing Sample 2481 -> class result (digit 7)
Testing Sample 2482 -> class result (digit 8)
Testing Sample 2483 -> class result (digit 9)
Testing Sample 2484 -> class result (digit 10)
Testing Sample 2485 -> class result (digit 11)
Testing Sample 2486 -> class result (digit 12)
Testing Sample 2487 -> class result (digit 13)
Testing Sample 2488 -> class result (digit 14)
Testing Sample 2489 -> class result (digit 15)
Testing Sample 2490 -> class result (digit 16)
Testing Sample 2491 -> class result (digit 17)
Testing Sample 2492 -> class result (digit 18)
Testing Sample 2493 -> class result (digit 19)
Testing Sample 2494 -> class result (digit 20)
Testing Sample 2495 -> class result (digit 21)
Testing Sample 2496 -> class result (digit 22)
Testing Sample 2497 -> class result (digit 23)
Testing Sample 2498 -> class result (digit 24)
Testing Sample 2499 -> class result (digit 25)

Results on the testing dataset
Correct classification: 2500 (100%)
Wrong classifications: 0 (0%)

```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0

Figura 57 Comprobación del entrenamiento de la red neuronal artificial

La base de datos formada luego del entrenamiento es almacenada en un documento del tipo XML que es utilizado como plantilla para el reconocimiento ya sea de los números o de las letras del alfabeto en LSEC. Es importante recordar que se crean dos clasificadores uno encargo del reconocimiento de los números y el otro de las letras, pero no de los dos, debido a que pueden

existir confusiones en ciertos números con algunas letras como es el caso del número 6 con la letra w o el número 2 con la letra v.

3.3. Reconocimiento e interpretación.

De acuerdo con (Gómez Correa & Trejos Largo, 2014) la clasificación es la fase de incitación del conocimiento a través de información suministrada a una máquina para que la aprenda, posteriormente esta información aprendida es utilizada para la interpretación y reconocimiento de las clases o categorías a la cual pertenece o no un objeto.

Para la investigación la interpretación se efectúa mediante la comparación de la imagen procesada para encontrar las características más importantes versus una base de datos que contenga información acerca de los números del 1 al 10 y las letras del alfabeto en Lenguaje de Señas Ecuatoriana, la interpretación es el último paso del sistema de visión artificial, la figura 58 muestra cómo se reconoce o se identifica el numeral 1

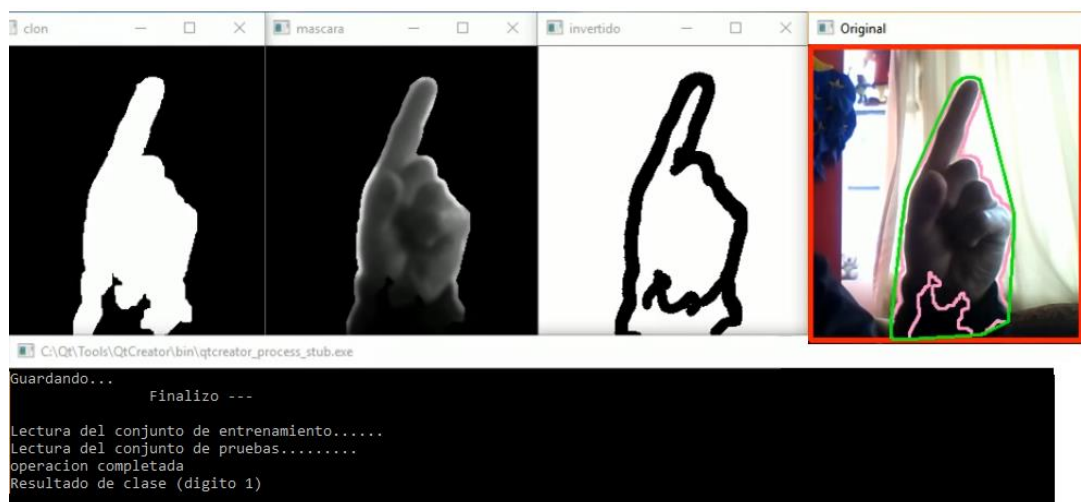


Figura 58 Reconocimiento del número 1 en LSEC

En la figura 59 se puede observar el reconocimiento de una de las letras del alfabeto en Lenguaje de Señas Ecuatoriano la cual es la letra g.

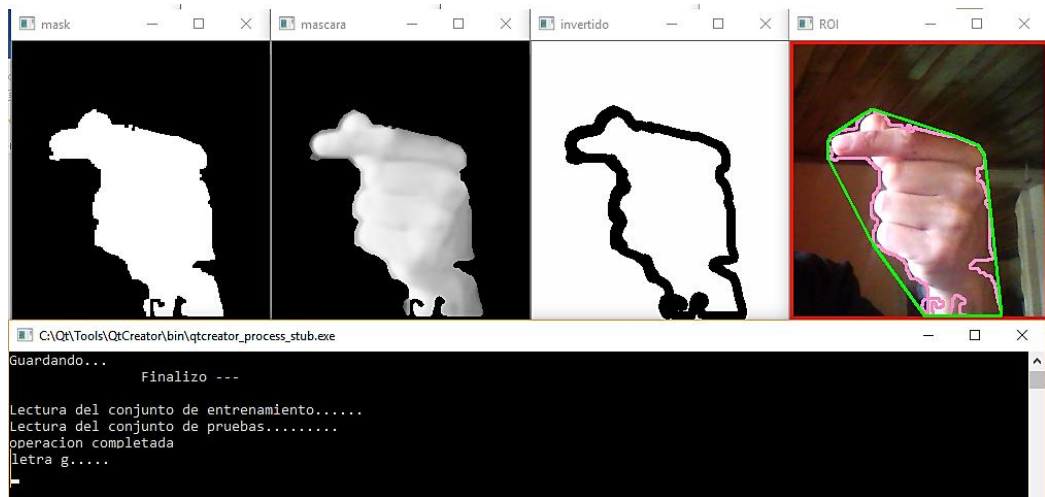


Figura 59 Reconocimiento de la letra g en LSEC

CAPÍTULO IV

INTERFAZ GRÁFICA DE USUARIO (GUI)

El capítulo IV se basa en el diseño e implementación de la parte gráfica del sistema de visión artificial encargado de la interpretación de los números del uno al diez, así como de las letras del abecedario en Lenguaje de Señas Ecuatoriana, en el capítulo se detallan los procesos de implementación de la interfaz gráfica buscando que la interacción con el usuario sea de forma sencilla y se logre un aprendizaje exitoso.

4.1. Descripción general.

Tras haber desarrollado la parte del código fuente del programa que permite cumplir con todas las operaciones deseadas con la imagen de entrada para ser reconocida como un número o una letra del alfabeto en LSEC, se propone el diseño de una interfaz gráfica que permita al usuario interactuar de manera intuitiva y sencilla con el programa.

Se busca que la GUI tenga ciertos parámetros para su utilización como son la facilidad de manejo, estética agradable, inclusión de iconos, inclusión de botones, accesibilidad simple y sobre todo intuitivo.

La figura 60 muestra el esquema que seguirá la interfaz gráfica a implementar, se accederá a la GUI a partir de un ejecutable, una vez dentro se despliega la ventana principal la cual consta de un widget principal el cual permite la visualización de las imágenes que son recibidas a partir de la cámara web, botones los cuales permiten la captura de la imagen para ser comparada con la base de datos obtenidos en el capítulo anterior, así como seleccionar entre varios tipos de menús como es el aprender y el modo test o prueba, también estos son útiles para la elección de la utilización de números o letras del abecedario y con esto brindar una respuesta de la realización de manera correcta o incorrecta ya sea del número o de la letra en LSEC.

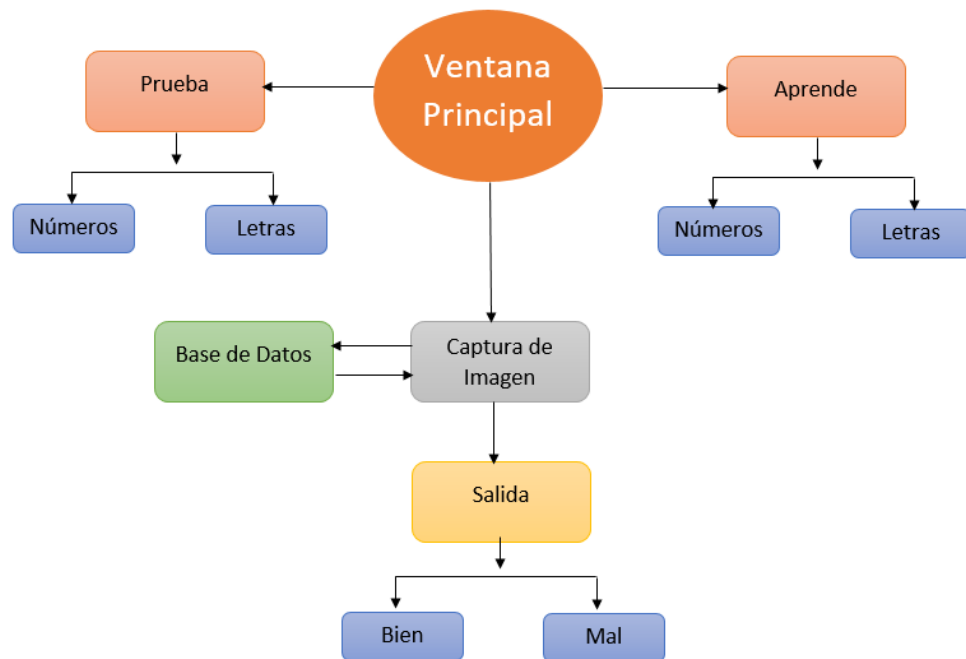


Figura 60 Funcionamiento de la GUI

4.2. Desarrollo de la interfaz gráfica.

La interfaz gráfica va ser desarrollada en Qt la misma que es una multiplataforma que utiliza código abierto e incluye herramientas gráficas que apresuran el desarrollo, se basa en lenguaje de programación C++, se utiliza uno de sus entornos de desarrollo integrado (IDE) como es el Qt Creator el mismo que es compatible con las librerías de OpenCV, Qt permite el desarrollo de infinidad de aplicaciones orientados a la industria, así como del hogar.

Qt Creator presenta varias ventajas por la cual es un software que posee grandes prestaciones para el desarrollo de la GUI de cualquier aplicación de acuerdo con García (2015) existen varias ventajas que ofrece el software como son los siguientes:

- Sencillo de utilizar
- Gratis
- Código abierto
- Mejora continua
- Multiplataforma compatible con Linux, Windows y Mac

4.2.1. Instalación de OpenCV y Qt Creator

La instalación de OpenCV así como de Qt Creator y todos sus componentes necesarios para que este pueda ser utilizado en un computador se realiza mediante el seguimiento secuencial de pasos descrito por Mendoza (2015) en su video donde muestra cómo realizar una correcta instalación del conjunto de paquetes que son necesarios para el uso adecuado de Qt con OpenCV.

4.2.2. Ambiente de programación de Qt Creator

Esta IDE de acuerdo con García (2015) presenta varias características que ayudan en el desarrollo de aplicaciones como son: soporta lenguaje de programación en C++, sintaxis con diferente coloración para mayor entendimiento, ayuda en el cierre de paréntesis y símbolos, autocompletado de código en forma predictiva, así como depurados línea a línea, puesta en circulación, localiza y permite remover errores del programa.

Al crear un nuevo proyecto el programa ayuda en la selección de ciertos parámetros para la creación de dichas aplicaciones como es el tipo de proyecto, la ubicación, el compilador y las clases que se utilizarán (figura 61).

De acuerdo con García (2015) una vez configurado el tipo de proyecto a utilizar se obtienen cuatro archivos uno de proyecto del tipo (.pro) que contiene todas la librerías necesarias para él programa, uno de cabecera del tipo (.h) que contiene por lo general subrutinas como también las variables, uno de fuentes del tipo (.ccp) que contiene las rutinas así como procedimiento de programación y por ultimo uno de formas del tipo (.ui) el cual contiene todas la herramientas gráficas para la creación de la GUI de cualquier proyecto.

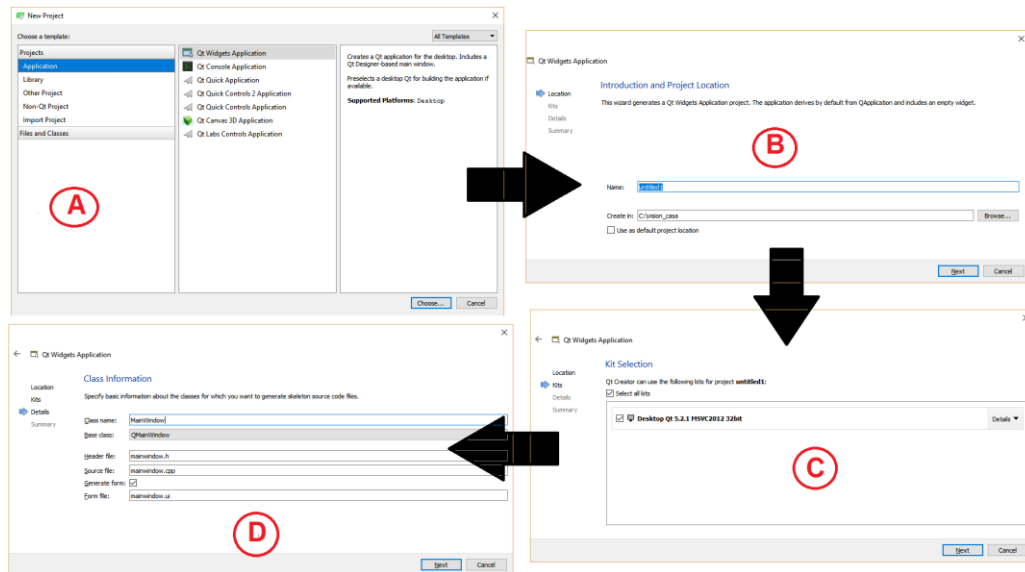


Figura 61 Asistente para la creación de nuevas aplicaciones
 a) tipo de proyecto b) ubicación c) compilador d) clases

La pantalla principal en la que se trabaja en la IDE de Qt Creator se puede observar en la figura 62, pero hay que tener presente que se requiere de la utilización de un lenguaje de programación tipo c++ para que las herramientas gráficas de la GUI sean funcionales.

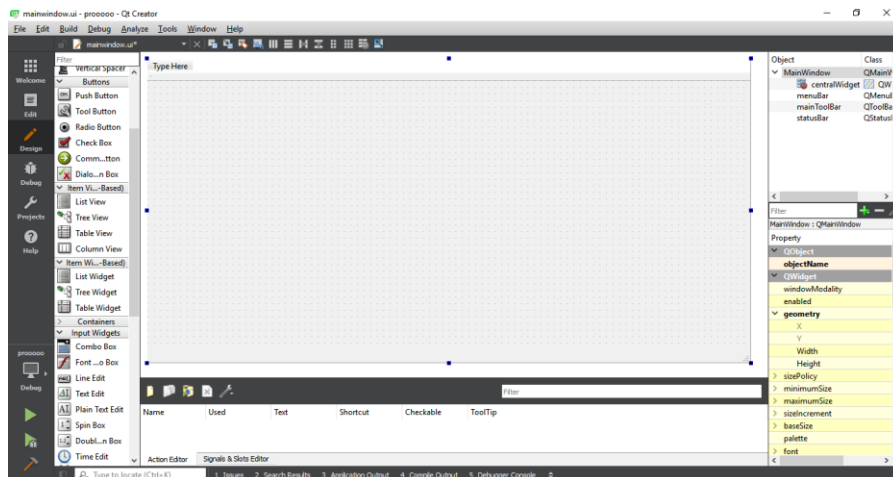


Figura 62 Pantalla para creación de GUI en Qt Creator

4.2.3. Implementación de la GUI

Para la implementación de la GUI se utilizan herramientas propias de Qt Creator las mismas que logran la captura de la imagen a través de una cámara

web utilizando el código que se puede visualizar en la figura 63 el cual contiene un objeto del tipo QTimer el cual permite la captura de los frames cada cierto tiempo.

```
cap=VideoCapture(0);
if(!cap.isOpened())
{
    ui->plainTextEdit->appendPlainText("camara no conectada");
    return;
}
bg.setInt("nmixtures",2);
bg.setBool("detectShadows", false);
timer = new QTimer(this);
connect(timer,SIGNAL(timeout()),this,SLOT(principal()));
timer->start(35);
```

Figura 63 Código para la captura de frames

Para imprimir los frames en la GUI se requiere de la utilización de un widget del tipo label o etiqueta, así como de la creación de un objeto que contenga la imagen como es el QImage y un mapa de pixeles que permitan imprimir los datos en el widget para esto se utiliza QPixmap. La impresión de los frames en el widget se obtiene mediante la utilización del código que se visualiza en la figura 64.

```
QImage image = QImage((const uchar*)frame.data, frame.cols,
                    frame.rows,frame.step, QImage::Format_RGB888).rgbSwapped();
ui->label->setPixmap(QPixmap::fromImage(image));
```

Figura 64 Código para la impresión de frames en la GUI

La figura 65 muestra la captura e impresión de los frames de video en la GUI.

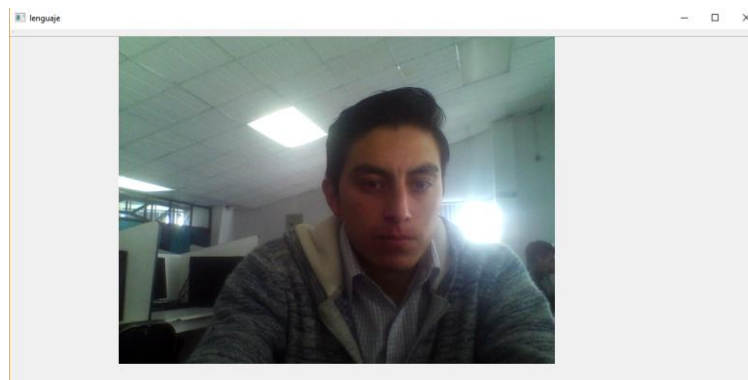


Figura 65 Impresión de frames en la GUI

Para la creación del menú prueba, así como del menú aprende se requiere de la utilización de botones de radio o conocidos por su nombre en inglés de Radio Button, estos también son necesarios para la elección entre números o letras ya sea en el menú aprende o en el menú prueba en la figura 66 se visualiza el código empleado para la utilización de los botones de radio.

```
void MainWindow::on_radioButton_clicked()
{ }

void MainWindow::on_radioButton_2_clicked()
{ }

void MainWindow::on_radioButton_3_clicked()
{ }

void MainWindow::on_radioButton_4_clicked()
{ }
```

Figura 66 Código para la utilización de botones de radio

La figura 67 muestra las posibles combinaciones que existirán entre el menú aprende (números o letras) y el menú prueba (letras o números).

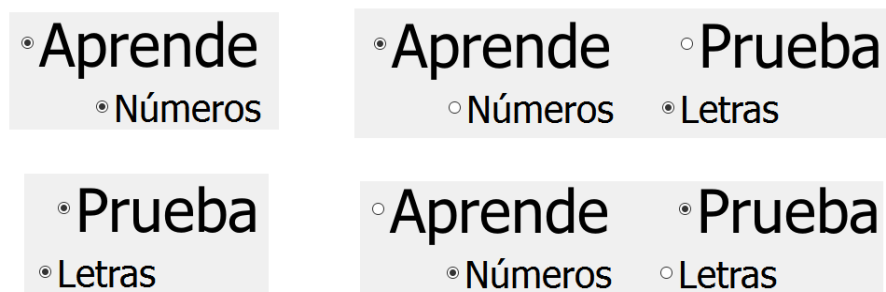


Figura 67 Combinación entre menús de la GUI

Para la captura y comparación de la imagen principal con la base de datos, así como para cerrar el programa o para finalizar en el modo prueba, hasta para resetear las variables se requieren de la utilización de botones, los mismos que para su implementación utilizan el código observado la figura 68.

```

void MainWindow::on_pushButton_clicked()
{ }

void MainWindow::on_pushButton_3_clicked()
{ }

void MainWindow::on_pushButton_4_clicked()
{ }

```

Figura 68 Código para la utilización de botones

En la figura 69 se muestran los botones que se utilizan en la GUI de la aplicación cada uno cumple con distintos requerimientos como es la captura, el reseteo, finalización del modo prueba y salir de la aplicación.

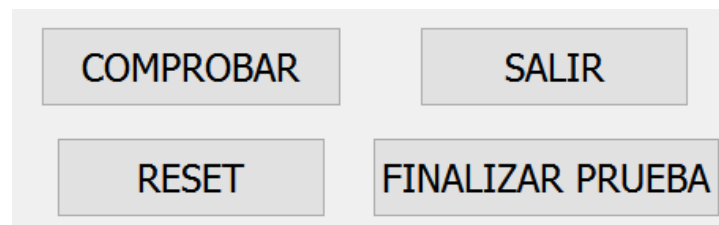


Figura 69 Botones utilizados en la GUI

Como se observa tanto los botones como los radio botones actúan como un condicional "If" es decir si son activados entonces cumplen una sentencia en tanto que no se activen permanecen sin realizar ningún trabajo.

Para la impresión de texto se utiliza un plainText Edit el cual permite visualizar y editar cualquier tipo de texto ya sea este un carácter o una cadena de caracteres, se utiliza el código que se visualiza en la figura 70 para la impresión de texto.

```

ui->plainTextEdit->appendPlainText("camara no conectada");
return;

```

Figura 70 Código para imprimir texto en GUI

En la figura 71 se observa la impresión de texto en el widget.

Cámara no conectada

Figura 71 Impresión de texto en el Widget

Obtenidas las herramientas principales para la creación de un widget, se utilizan para la creación de la interfaz gráfica de usuario encargada de la interpretación y enseñanza del Lenguaje de Señas Ecuatoriana, son necesarias varias imágenes para cumplir con el objetivo, en la figura 72 se muestra el bosquejo de la interfaz culminada con cada una de sus partes, se toma en cuenta que la GUI es orientada a la enseñanza por tanto se realiza un diseño que capte la atención del usuario y el aprendizaje sea de manera interactiva y emocionante.

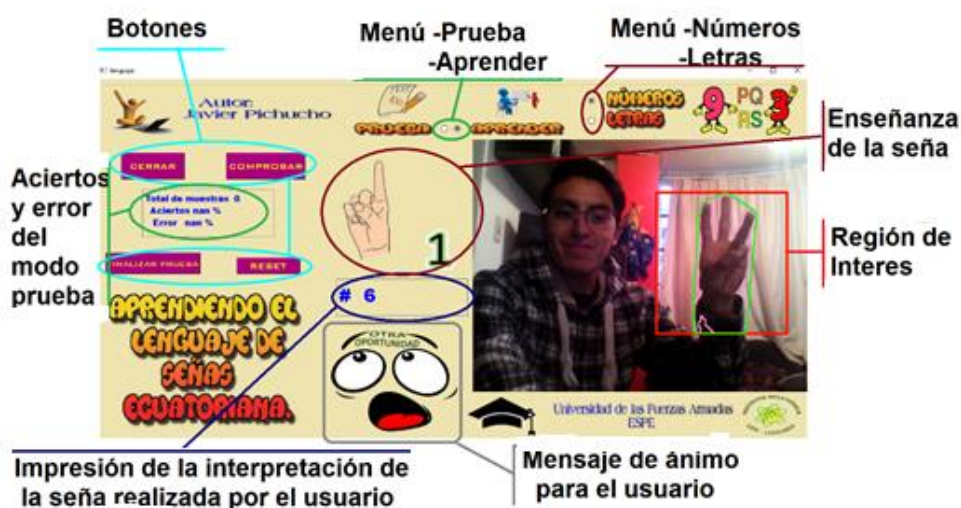


Figura 72 Bosquejo interfaz gráfica

4.3. Funcionamiento principal

4.3.1. Modo aprender letras y números en LSEC

Se selecciona de entre las opciones el ítem del modo aprender y se elige entre números o letras, como se observa en la figura 73 se elige números.



Figura 73 Modo aprender números en LSEC

Una vez seleccionada la opción de aprendizaje se procede a través del botón comprobar a la enseñanza de cada uno de los números en LSEC, cada vez que se realiza bien el número este pasara al siguiente y se emite el mensaje “MUY BIEN” como se observa en la figura 74.

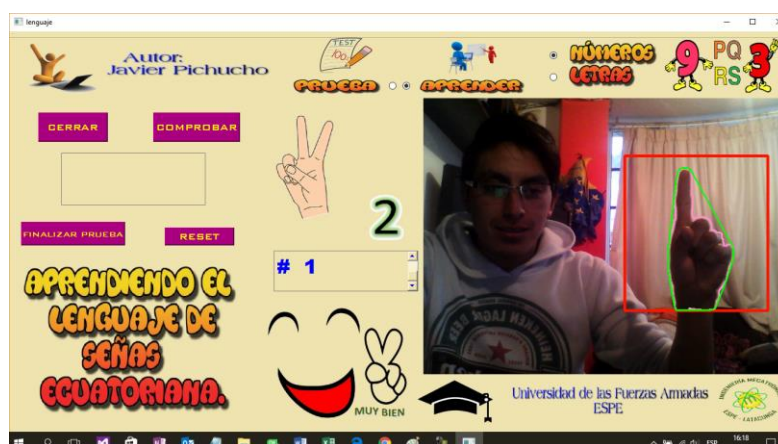


Figura 74 Aprendizaje de números en LSEC de manera correcta

Si el número es realizado de manera incorrecta este permanecerá estático hasta que se lo realice bien y se emite un mensaje de ánimo al usuario el mismo que es “OTRA OPORTUNIDAD” como se observa en la figura 75.

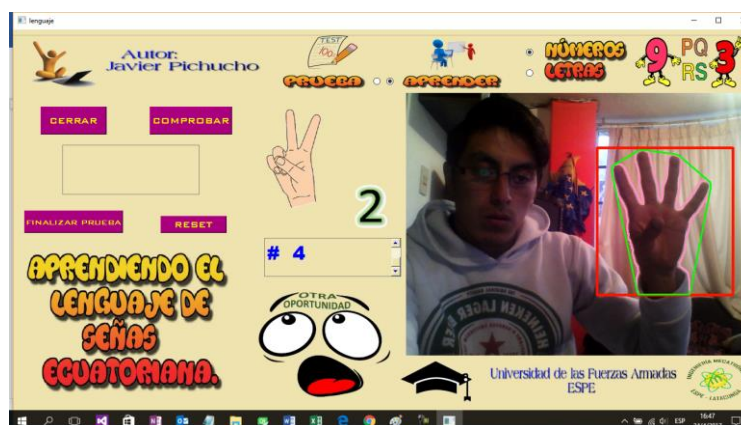


Figura 75 Aprendizaje de números en LSEC de manera incorrecta

Una vez realizados todos los números del 1 al 10 estos se repetirán nuevamente en un lazo infinito hasta que el usuario tome la decisión de aprender ahora las letras o realizar ya sea la prueba de letras o números, para el aprendizaje de las letras en LSEC se repite la secuencia descrita para el aprendizaje de los números.

4.3.2. Modo prueba números y letras en LSEC

Se selecciona de entre las opciones el ítem del modo prueba y se elige entre números o letras, como se observa en la figura 76 se elige letras.



Figura 76 Modo prueba letras en LSEC

Una vez seleccionada la opción de prueba se procede a través del botón comprobar la toma de cada una de las letras del abecedario en LSEC de forma aleatoria, cada vez que se realiza bien la letra se emite el mensaje “MUY BIEN” (figura 77) y un contador interno guardara la respuesta.

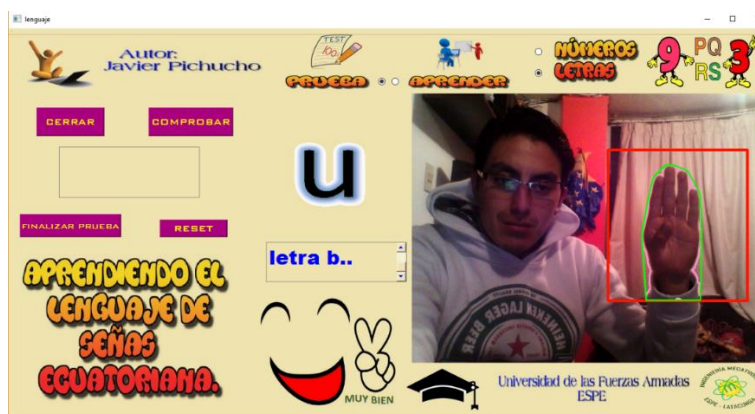


Figura 77 Realización de la letra en LSEC de manera correcta

Si la letra es realizada de manera incorrecta siguen cambiando las letras y se emite un mensaje al usuario el mismo que es “MAL” (figura 78), las respuestas se siguen guardando en él contador.

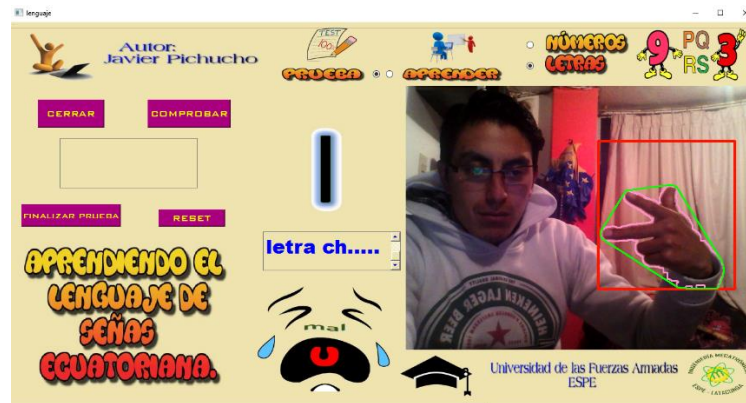


Figura 78 Realización de la letra en LSEC de manera incorrecta

Una vez realizadas todas las pruebas que el usuario decida desarrollar se pulsa el botón finalizar prueba y el programa emite un mensaje dando a conocer el número de muestras que se han desarrollado, así como el porcentaje de aciertos y el porcentaje de error que se obtuvo como se puede visualizar en la figura 79, para la realización de la prueba en el modo números en LSEC se siguen los mismos lineamientos descritos en el modo prueba para las letras.



Figura 79 Impresión del porcentaje de aciertos y error

CAPÍTULO V

PRUEBAS Y RESULTADOS

El capítulo V agrupa todos los protocolos de pruebas establecidos para analizar el grado de cumplimiento de los parámetros propuestos en un inicio en todas las etapas del sistema de visión artificial como de la implementación de la interfaz gráfica de usuario.

5.1. Pruebas del sistema de Visión Artificial y del tratamiento de imágenes

Se requiere comprobar que los rangos tomados tanto en la etapa de adquisición como del tratamiento de las imágenes sean los adecuados para obtener los detalles necesarios para la interpretación del lenguaje de señas ecuatoriana a partir de los clasificadores implementados.

Se realizará una prueba para verificar cómo reacciona el programa en diferentes ambientes si la adquisición y el tratamiento es la adecuada para cada imagen y observar como la información va evolucionando, verificando si existen pérdidas o se mantienen los datos de acuerdo al ambiente que se considera, se verificará la incidencia de la intensidad de la luz en la adquisición y tratamiento de las imágenes, un punto importante es comprobar si el programa reconocerá de entre varios usuarios la mano sin importar el tono de piel o el tamaño que esta tenga.

5.1.1. Prueba de ambientes

Se requiere definir para las pruebas de ambientes los objetivos, la planificación, el banco de ensayos o de pruebas, así como la forma de validación e interpretación de los resultados a obtener.

5.1.1.A. Definición de los objetivos de la prueba

Se requiere comprobar que el sistema de visión artificial para la interpretación del alfabeto y números del uno al diez en Lenguaje de Señas Ecuatoriana mediante la utilización de software libre funcionará en diferentes ambientes sin importar mucho el entorno que lo rodea, así como la iluminación que este tenga.

5.1.1.B. Diseño de la prueba de ambientes

Para ejecutar las pruebas de ambientes para el sistema se consideran 10 entornos que no están controlados y los más usuales dentro y fuera de una casa para observar cómo reaccionará el programa tanto en la adquisición, tratamiento y reconocimiento de la mano del usuario.

5.1.1.C. Planificación de la prueba de ambientes

Para la ejecución de la prueba es necesario obtener un banco de imágenes que contengan tanto la mano del usuario como los 10 ambientes comunes dentro y fuera de un hogar con la ayuda de la cámara web del computador.

Para la toma de las muestras a partir de la cámara web y el programa implementado se procede de la siguiente forma.

- Encender el computador
- Ejecutar el programa para la adquisición de muestras.
- Guardar las muestras.
- Finalizar el programa.

5.1.1.D. Prototipo y banco de pruebas

Para la realización de las pruebas de ambiente son necesarios ciertos elementos los cuales se enlistan a continuación

- Mano del usuario
- Computador
- Programa del sistema de la interpretación del alfabeto y números del uno al diez en Lenguaje de Señas Ecuatoriana

En la figura 80 se observan los elementos utilizados para la realización de las pruebas de ambientes.



Figura 80 Banco de pruebas

5.1.1.E. Interpretación y validación de los resultados de la prueba

Una vez realizadas las pruebas del funcionamiento del programa en varios ambientes se muestran los resultados obtenidos, en la figura 81 se presenta uno de los ambientes donde el programa funciona satisfactoriamente.

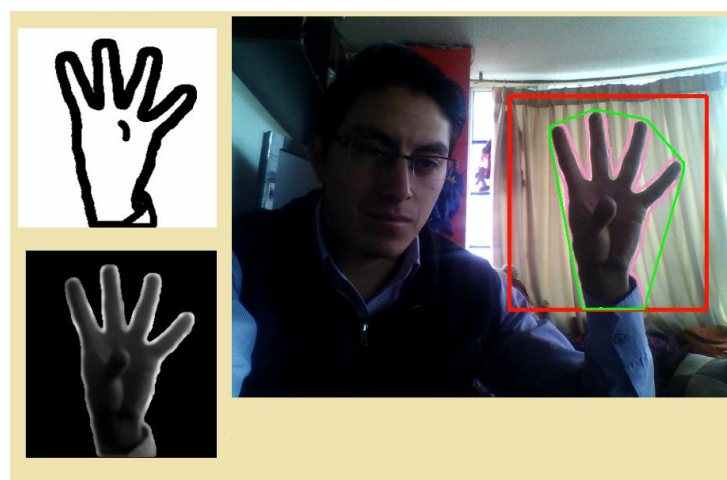


Figura 81 Ambiente propicio para el programa

Como se observa en la figura 81 la adquisición, el tratamiento y el reconocimiento se realiza de manera óptima, los datos enviados a los clasificadores son limpios y no existe presencia de ruido alguno. En la figura 82 se contempla la adquisición de manera no tan eficiente por ende existirá la presencia de ruido y no será reconocido de buena manera por parte del clasificador.

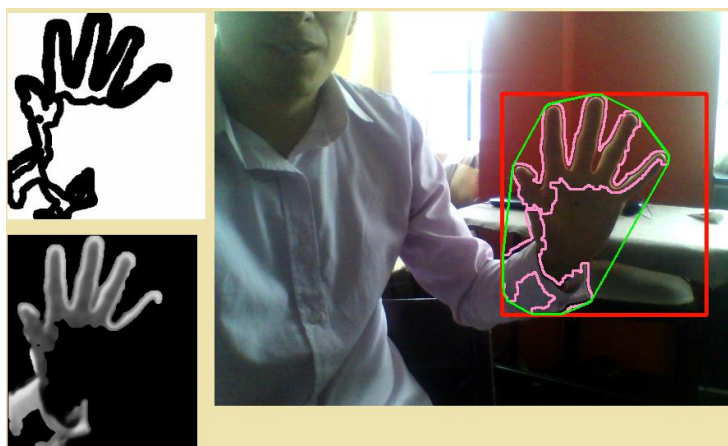


Figura 82 Perdida de datos en la imagen

Las demás muestras tomadas se presentan en el Anexo B. En la tabla 8 se indica cómo reacciona el programa de acuerdo al ambiente tomado en cada muestra.

Tabla 8
Prueba de ambientes no controlados

Muestra	Reconocimiento de la mano	Tratamiento de la imagen
Ambiente 1 (Sala)	OK	OK
Ambiente 2 (Comedor)	OK	OK
Ambiente 3 (Dormitorio)	MAL	MAL
Ambiente 4 (Cocina)	OK	OK
Ambiente 5 (Cuarto de Estudio)	OK	OK
Ambiente 6 (Sala toma nocturna)	OK	OK
Ambiente 7 (Dormitorio)	MAL	MAL
Ambiente 8 (Sin presencia de Luz)	OK	OK
Ambiente 9 (Terraza)	MAL	MAL
Ambiente 10 (Oficina)	OK	OK

Como se observa en la tabla 8 siete de los diez ambientes funcionan con acierto para calcular el error se hace uso de la siguiente formula

$$ep = \frac{(Tm - Mc) * 100}{Tm}$$

Ecuación 11 Calculo del porcentaje de error

Donde:

ep = Error porcentual [%]

Tm = Total de muestras

Mc = Muestras correctas

Entonces utilizando los valores obtenidos se tiene que

$$ep = \frac{(10 - 7) * 100}{10}$$

$$ep = 30\%$$

El error porcentual obtenido para el funcionamiento del sistema en diferentes ambientes es del 30%, por lo que el porcentaje de confiabilidad es del 70% tomando como base las pruebas realizadas con las muestras presentadas en la tabla 8.

5.1.2. Incidencia de la luz en el proyecto.

Se requiere comprobar que afectación provocará el cambio de iluminación sobre los datos que se envían a los clasificadores a partir de la captura y tratamiento de las imágenes.

La luminosidad en varios proyectos de Visión artificial es muy importante ya que para un correcto funcionamiento se requiere de entornos controlados donde no cambie constantemente la luz es decir permanezca estática, para la investigación el entorno siempre variará no permanece constante por tal

motivo el programa se implementa buscando compensar ciertos cambios en el ambiente, pero como todo equipo sufre defectos bajo ciertos parámetros produciendo errores los mismos que serán enlistados en la tabla 9.

Tabla 9
Lista de errores en la detección de la mano del usuario

Imagen	Error
	<p>Cambio brusco en la iluminación el ambiente cambia de un día soleado a un día con lluvia, el fondo es susceptible a los cambios en el ambiente debido a que existe una puerta de vidrio cerca y no permite detectar de manera óptima la mano del usuario por tanto el tratamiento de la imagen es malo.</p>
	<p>El fondo es una ventana el exceso de luz hace que el programa reconozca detalles de la ventana y permite que algunos de los datos pertenecientes a la mano se pierdan.</p>
	<p>El fondo de color blanco permite notar más claramente los cambios de iluminación y más aún si este se encuentra bien alumbrado por parte de ventanas o puertas es ahí cuando se producen errores en la detección correcta de la mano.</p>

CONTINÚA





Al encontrarse al aire libre tanto el programa como el usuario son susceptibles a los cambios en la iluminación así sean estos los más insignificantes.

Como se observa de acuerdo a lo detallado en la tabla 9 la mayor deficiencia que presenta el programa es cuando existan cambios extremos en la iluminación del entorno este es un factor que se puede controlar muy poco ya que el cambio climático ambiental es repentino, el programa busca adaptarse al entorno, pero al existir estos cambios provocan desperfectos en la detección del contorno de la mano existiendo defectos en el tratamiento de la imagen y en el reconocimiento de la letra o número.

5.1.3. Prueba de tono de piel y tamaño de la mano del usuario.

Se requiere definir para las pruebas de tono de piel y tamaño de la mano del usuario los objetivos, la planificación, el banco de ensayos o de pruebas, así como la forma de validación e interpretación de los resultados a obtener.

5.1.3.A. Definición de los objetivos de la prueba

Se busca comprobar que el sistema visión artificial para la interpretación del alfabeto y números del uno al diez en Lenguaje de Señas Ecuatoriana mediante la utilización de software libre funcionará sin importar el usuario ya sea este niño o niña, adolescente, adulto o adulto mayor, sin importar su tonalidad de piel y el tamaño de sus manos.

5.1.3.B. Diseño de la prueba de tono de piel y tamaño de la mano del usuario

Para ejecutar las pruebas de tono de piel y tamaño de la mano del usuario para el sistema se consideran 10 personas de diferente contextura, cultura y

edad para observar cómo reaccionará el programa tanto en la adquisición, tratamiento y reconocimiento de la mano.

5.1.3.C. Planificación de la prueba de tono de piel y tamaño de la mano del usuario

Para la ejecución de la prueba es necesario obtener 10 imágenes que contengan la mano del usuario siendo este niño o niña, adolescente, adulto y adulto mayor con la ayuda de la cámara web del computador.

Para la toma de las muestras a partir de la cámara web y el programa implementado se procede de la forma descrita en el apartado 5.1.1.C.

5.1.3.D. Prototipo y banco de pruebas

Para la realización de las pruebas de ambiente son necesarios ciertos elementos los cuales se enlistan en el apartado 5.1.1.D. Se utiliza el mismo banco de pruebas que se empleó para la prueba de ambientes la misma que se muestra en la figura 80.

5.1.3.E. Interpretación y validación de los resultados de la prueba

Una vez realizadas las pruebas del funcionamiento del programa con diferentes usuarios se muestran los resultados obtenidos, en la figura 83 se observa una muestra de la toma de imagen a un menor de edad de aproximadamente 4 años con tonalidad de piel clara.

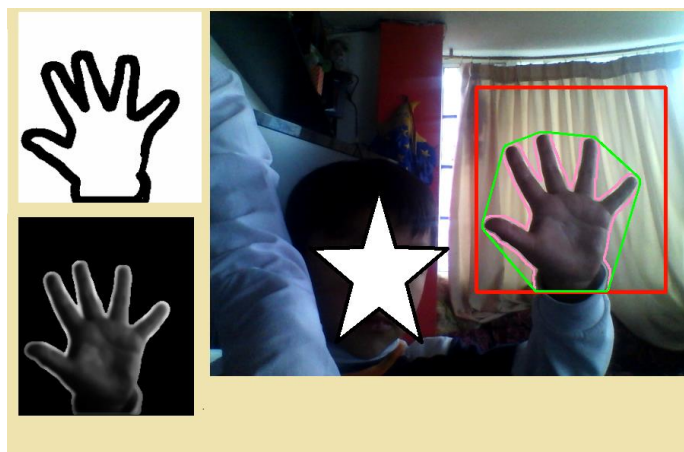


Figura 83 Reconocimiento de la mano de un niño

En la figura 84 se muestra la toma de imagen a una persona adulta mayor con tonalidad de piel oscura.

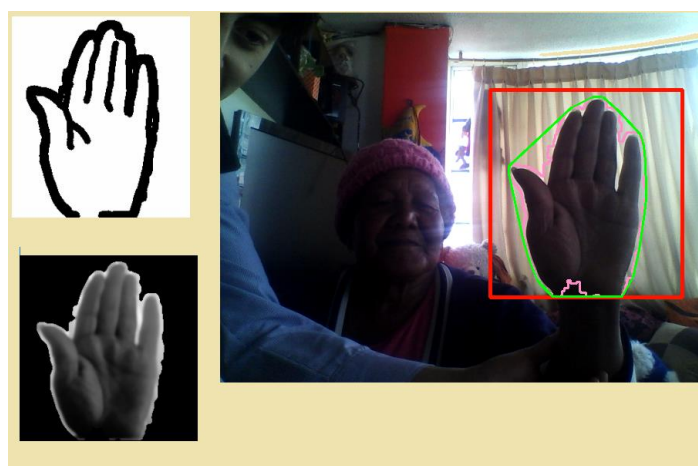


Figura 84 Reconocimiento de la mano de una persona adulta mayor

Las demás muestras tomadas se presentan en el Anexo B. En la tabla 10 se manifiestan los resultados obtenidos al utilizar el programa con diferentes tipos de usuarios.

Tabla 10
Pruebas del tono y tamaño de mano

Tipo	Reconocimiento de la mano	Tono de piel	Tamaño de la mano	Tratamiento de la imagen
Muestra 1 (2 años)	OK	OK	OK	OK

CONTINÚA 

Muestra 2 (4 años)	OK	OK	OK	OK
Muestra 3 (11 años)	OK	OK	OK	OK
Muestra 4 (13 años)	OK	OK	OK	OK
Muestra 5 (16 años)	OK	OK	OK	OK
Muestra 6 (27 años)	OK	OK	OK	OK
Muestra 7 (31 años)	OK	OK	OK	OK
Muestra 8 (48 años)	OK	OK	OK	OK
Muestra 9 (55 años)	OK	OK	OK	OK
Muestra10 (73 años)	OK	OK	OK	OK

Como se observa en las figuras 83 y 84 así como en la tabla 10 la adquisición, el tratamiento y el reconocimiento se realiza de forma eficiente sin importar tanto la tonalidad de la piel, el tamaño ni el usuario dando un excelente porcentaje de efectividad por parte del programa al aplicar la prueba prevista dando resultados positivos para el sistema.

5.2. Pruebas de los clasificadores

Se requiere comprobar que los clasificadores creados tanto para la interpretación de los números, así como de las letras del abecedario en lenguaje de señas ecuatoriana sean capaces de reconocer cada uno de los patrones para los cuales fueron entrenados.

5.2.1. Prueba del clasificador encargado del reconocimiento de los números en LSEC

Se requiere definir para la prueba del clasificador encargado del reconocimiento de los números en LSEC los objetivos, la planificación, el banco de ensayos o de pruebas, así como la forma de interpretación y validación de los resultados a obtener.

5.2.1.A. Definición de los objetivos de la prueba

Se requiere comprobar que el clasificador encargado del reconocimiento de los números en LSEC permite una correcta clasificación brindando una efectividad mayor al 90%.

5.2.1.B. Diseño de la prueba de salida del clasificador encargado del reconocimiento de los números en LSEC

Para ejecutar la prueba del clasificador encargado del reconocimiento de los números en LSEC se crea un conjunto de verificación de 270 muestra, 30 muestras por cada número, las muestras son tomadas a personas que conocen muy bien el Lenguaje de Señas Ecuatoriana.

5.2.1.C. Planificación de la prueba de salida del clasificador encargado del reconocimiento de los números en LSEC

Para la ejecución de la prueba es necesario obtener las muestras de la mano realizando los diferentes números y almacenarlos en una carpeta para su posterior reconocimiento y clasificación esto se logra mediante la cámara web del computador.

Para la toma de las muestras, así como de la interpretación y clasificación de cada una se procede de la siguiente forma.

- Encender el computador
- Ejecutar el programa para la adquisición de muestras
- Ejecutar el programa para comprobación del clasificador encargado del reconocimiento de los números en LSEC.
- Interpretar cada una de las muestras.
- Finalizar el programa.
- Apagar el computador.

5.2.1.D. Prototipo y banco de pruebas

Para la realización de las pruebas de clasificación de los números en LSEC son necesarios ciertos elementos los cuales se enlistan a continuación

- Usuarios
- Computador
- Programa del sistema de la interpretación del alfabeto y números del uno al diez en Lenguaje de Señas Ecuatoriana

En la figura 85 se observan los elementos utilizados para la realización de las pruebas de clasificación de los números en LSEC.



Figura 85 Banco de pruebas de los números en LSEC

5.2.1.E. Interpretación y validación de los resultados de la prueba

El sistema una vez capturada la imagen busca interpretarla a través del clasificador buscando el número al que pertenece. En la tabla 11 se presentan los resultados una vez realizadas las pruebas al clasificador encargado del reconocimiento de los números en LSEC.

Tabla 11
Resultados del reconocimiento de los números en LSEC

Número	Aciertos	Fallas	% Aciertos	% Error
1	26	4	86,67	13,33
2	30	0	100	0
3	27	3	90	10
4	30	0	100	0
5	30	0	100	0
6	26	4	86,67	13,33
7	25	5	83,33	16,67
8	30	0	100,00	0,00
9	25	5	83,33	16,67
Promedio			92,22	7,78

Como se observa en la tabla 11 el error promediado de todas las pruebas realizadas que se obtiene es del 7.78%, por lo que el porcentaje de confiabilidad de reconocimiento y clasificación de los números en LSEC es del 92.22% dando como resultado un clasificador confiable.

5.2.2. Pruebas del clasificador encargado del reconocimiento de las letras del abecedario en LSEC

Se requiere definir para la prueba del clasificador encargado del reconocimiento de las letras del abecedario en LSEC los objetivos, la planificación, el banco de ensayos o de pruebas, así como la forma de interpretación y validación de los resultados a obtener.

5.2.2.A. Definición de los objetivos de la prueba

Se requiere comprobar que el clasificador encargado del reconocimiento de las letras del abecedario en LSEC permite una correcta clasificación brindando una efectividad mayor al 90%.

5.2.2.B. Diseño de la prueba de salida del clasificador encargado del reconocimiento de los números en LSEC

Para ejecutar la prueba del clasificador encargado del reconocimiento de las letras del abecedario en LSEC se crea un conjunto de verificación de 750 muestra, 30 muestras por cada letra, las muestras son tomadas a personas que conocen muy bien el Lenguaje de Señas Ecuatoriana.

5.2.2.C. Planificación de la prueba de salida del clasificador encargado del reconocimiento de los números en LSEC

Para la ejecución de la prueba es necesario obtener las muestras de la mano realizando las diferentes letras del abecedario y almacenarlas en una carpeta para su posterior reconocimiento y clasificación esto se logra mediante la ayuda de la cámara web del computador.

Para la toma de las muestras, así como de la interpretación y clasificación de cada una de las letras del abecedario se procede de la forma descrita en el apartado 5.2.1.C. con el cambio de utilización del programa para comprobación del clasificador encargado del reconocimiento de las letras del abecedario en LSEC

5.2.2.D. *Prototipo y banco de pruebas*

Para la realización de las pruebas de clasificación de las letras del abecedario en LSEC son necesarios ciertos elementos los cuales se enlistan en el apartado 5.2.1.D. En la figura 86 se observan los elementos utilizados para la realización de las pruebas de clasificación de las letras del abecedario en LSEC.



Figura 86 Banco de pruebas letras del abecedario en LSEC

5.2.2.E. *Interpretación y validación de los resultados de la prueba*

El sistema una vez capturada la imagen busca interpretarla a través del clasificador buscando la letra del alfabeto a la que pertenece. En la tabla 12 se presentan los resultados una vez realizadas las pruebas al clasificador encargado del reconocimiento de las letras del abecedario en LSEC.

Tabla 12
Resultados del reconocimiento de las letras del abecedario en LSEC

Letra	Aciertos	Fallas	% Aciertos	% Error
A	28	2	93,33	6,67
B	30	0	100,00	0,00
C	29	1	96,67	3,33
Ch	30	0	100,00	0,00
D	28	2	93,33	6,67
E	28	2	93,33	6,67
F	28	2	93,33	6,67
G	28	2	93,33	6,67
H	25	5	83,33	16,67
I	30	0	100,00	0,00
K	29	1	96,67	3,33
L	30	0	100,00	0,00
M	27	3	90,00	10,00
N	26	4	86,67	13,33
O	27	3	90,00	10,00
P	30	0	100,00	0,00
Q	28	2	93,33	6,67
R	30	0	100,00	0,00
S	29	1	96,67	3,33
T	30	0	100,00	0,00
U	28	2	93,33	6,67
V	30	0	100,00	0,00
W	30	0	100,00	0,00
X	25	5	83,33	16,67
Y	29	1	96,67	3,33
	Promedio		94,93	5,07

Como se observa en la tabla 12 el error promediado de todas las pruebas realizadas que se obtiene es del 5.07%, por lo que el porcentaje de confiabilidad de reconocimiento y clasificación de las letras del abecedario en LSEC es del 94.93% dando como resultado un clasificador confiable.

5.3. Prueba a la interfaz gráfica de usuario (Sistema completo).

Se requiere comprobar que la interfaz de usuario se adapta bien al código desarrollado en C++ y además que su diseño sea adecuado y tenga gran

accesibilidad, así como cumple con la enseñanza tanto de los números como de las letras del abecedario en Lenguaje de Señas Ecuatoriana.

5.3.1. Definición de los objetivos de la prueba

Se requiere comprobar que la interfaz gráfica se adapta a cualquier usuario y brinda gran accesibilidad para el aprendizaje de los números y letras del abecedario en Lenguaje de Señas Ecuatoriana

5.3.2. Diseño de la prueba de la interfaz gráfica de usuario (sistema completo)

Para ejecutar la prueba de la interfaz gráfica de usuario se consideran 10 personas con edades comprendidas entre los 13 y los 55 años los mismos que no poseen ningún conocimiento previo del Lenguaje de Señas Ecuatoriana este conocimiento será adquirido a través de la interfaz gráfica planteada.

5.3.3. Planificación de la prueba de la interfaz gráfica de usuario (sistema completo)

Para la ejecución de la prueba se requiere de la predisposición de las 10 personas para el aprendizaje de los números y letras del abecedario en Lenguaje de Señas Ecuatoriana, y estar dispuestos para su posterior evaluación.

Para el aprendizaje de los números y de las letras en Lenguaje de Señas Ecuatoriana se procede de la siguiente forma.

- Elegir un ambiente adecuado para el aprendizaje
- Ejecutar el programa utilizado en la interpretación del alfabeto y números del uno al diez en Lenguaje de Señas Ecuatoriana.
- Familiarización con el programa
- Elegir el modo aprende tanto para números como para el alfabeto

- Aprendizaje

Una vez culminado el aprendizaje tanto del alfabeto como de los números del uno al diez en Lenguaje de Señas Ecuatoriana se procede a la toma de la prueba la misma que se aplica en el mismo programa teniendo como base 10 números y 20 letras por separado, es importante recalcar que el porcentaje de aceptación para aprobar la prueba es del 70% o más para la toma de prueba se procede de la siguiente forma.

- Elegir el modo prueba para números
- Realizar 10 interacciones
- Finalizar la prueba.
- Observar los resultados obtenidos
- Elegir el modo prueba para letras
- Realizar 20 interacciones
- Finalizar la prueba.
- Observar los resultados obtenidos
- Finalizar el programa
- Apagar el computador.

5.3.4. Prototipo y banco de pruebas

Para la realización de las pruebas de la interfaz gráfica de usuario (sistema completo) son necesarios ciertos elementos los cuales se enlistan a continuación

- Usuarios
- Computador
- Programa del sistema para la interpretación del alfabeto y números del uno al diez en Lenguaje de Señas Ecuatoriana

En la figura 87 se observan los elementos utilizados para la realización de las pruebas.

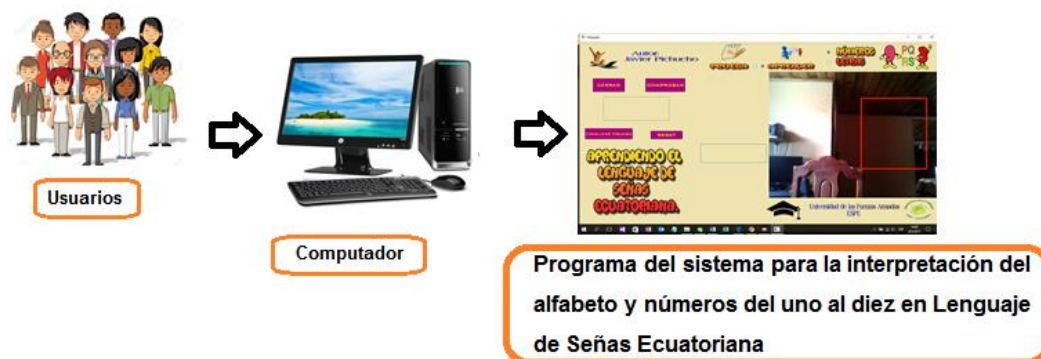


Figura 87 Banco de pruebas para la Interfaz gráfica de usuario

5.3.5. Interpretación y validación de los resultados de la prueba

Una vez realizado las pruebas al funcionamiento de la interfaz gráfica de usuario se muestran los resultados obtenidos, en la tabla 13 se puede observar el tiempo que se demora cada usuario en familiarizarse con la interfaz gráfica desarrollada.

Tabla 13
Tiempo de familiarización con el programa desarrollado

Usuario	Tiempo de Familiarización
Segundo	3 minutos
Carolina	2 minutos
Daniela	2 minutos
Jairo	3 minutos
Melany	3 minutos
Karla	2 minutos
Marcelo	3 minutos
María	3 minutos
Erick	2 minutos
Alfredo	3 minutos
Promedio	2.6 minutos

Como se puede contemplar en la tabla 13 el tiempo promedio utilizado para la familiarización con el programa desarrollado por parte de los usuarios fue de 2.6 minutos siendo el tiempo máximo 3 minutos y el tiempo mínimo de 2 minutos con estos rangos de valores se puede deducir que el

programa es de fácil entendimiento y presenta gran accesibilidad en cuanto a su manejo.

Una vez familiarizado el usuario con el programa se procede a la enseñanza del alfabeto y de los números del uno al diez en Lenguaje de Señas Ecuatoriana por parte del programa en el modo aprender y luego de esto, una vez que el usuario se sienta preparado para dar la prueba de aprendizaje se cambia al programa al modo prueba y se ejecuta la prueba, se muestran los resultados obtenidos a continuación en la figura 88 se observa a uno de los usuarios realizando la prueba de los números en LSEC, así como el porcentaje obtenido luego de realizar la prueba.

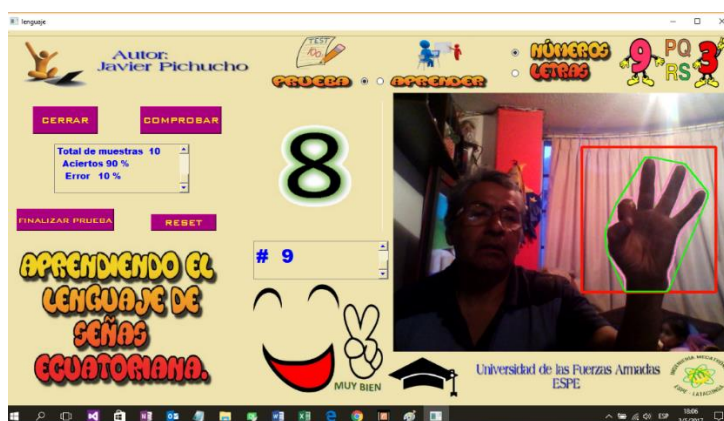


Figura 88 Usuario dando la prueba de los números en LSEC

En la figura 89 se contempla al usuario realizando la prueba del alfabeto en LSEC, así como se muestra el porcentaje obtenido luego de realizar la prueba.

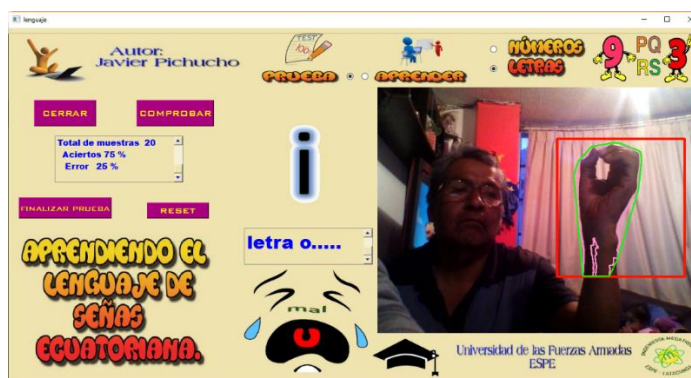


Figura 89 Usuario realizando la prueba del alfabeto en LSEC

Los resultados de las pruebas en imágenes tomadas a los diferentes usuarios se presentan en el Anexo C. En la tabla 14 se puede observar el tiempo que se demora cada usuario en aprender ya sea los números o el alfabeto en LSEC con la ayuda de la interfaz gráfica desarrollada, así como también el porcentaje obtenido luego de aplicar la prueba correspondiente. Es importante recalcar que el tiempo empleado para el aprendizaje y la toma de las pruebas a la población objeto de estudio fue alrededor de 1 semana por disponibilidad de tiempo.

Tabla 14
Tiempo de aprendizaje y porcentaje obtenido en la prueba desarrollada

Usuario	Tiempo de aprendizaje números en LSEC	Porcentaje obtenido en la evaluación de números	Tiempo de aprendizaje del alfabeto en LSEC	Porcentaje obtenido en la evaluación del alfabeto
Segundo	7 minutos	70%	3 horas 30 minutos	80%
Carolina	5 minutos	80%	2 horas 10 minutos	75%
Daniela	5 minutos	80%	2 horas 20 minutos	75%
Jairo	6 minutos	80%	2 horas 30 minutos	75%
Melany	7 minutos	80%	3 horas	70%
Karla	6 minutos	70%	3 horas	75%
Marcelo	10 minutos	90%	3 horas 20 minutos	75%
María	7 minutos	70%	3 horas 30 minutos	75%
Erick	10 minutos	80%	3 horas 10 minutos	80%
Alfredo	7 minutos	80%	4 horas	70%
Promedio	7 minutos	78%	3 horas 25 minutos	75%

Como se observa en la tabla 14 el tiempo promedio empleado para el aprendizaje de los números del uno al diez en Lenguaje de Señas Ecuatoriana fue de 7 minutos, el tiempo máximo 10 minutos y tiempo mínimo 5 minutos. Se visualiza también que luego del aprendizaje al aplicar la prueba se obtienen resultados muy buenos con un porcentaje promedio

de evaluación de 78 % es decir todos han aprendido y han superado la prueba de los números en LSEC.

Se observa también en la tabla 14 que el tiempo promedio empleado para el aprendizaje del alfabeto en Lenguaje de Señas Ecuatoriana fue de 3 horas con 25 minutos, el tiempo máximo 4 horas y tiempo mínimo 2 horas con 10 minutos. Se visualiza también que luego del aprendizaje al aplicar la prueba se obtienen resultados muy buenos con un porcentaje promedio de evaluación de 75 % es decir todos han aprendido y han superado la prueba del alfabeto en LSEC.

Es importante recalcar que el tiempo empleado para el aprendizaje y para la evaluación no es prioritario ya que se requiere comprobar que la interfaz impulsa el aprendizaje entendible de los números y del alfabeto en LSEC esto se comprueba observando la calificación obtenida luego de aplicar la prueba correspondiente ya que toda la población objeto de estudio logro pasar con un porcentaje mayor al 70%, otro de los puntos importantes que se nota al ejecutar el programa es que el usuario disfruta al realizar esta actividad ya que se convierte en un reto para él.

5.4. Alcances y limitaciones

5.4.1. Alcances del sistema

- Puede ser utilizado en varios tipos de ambientes dentro y fuera de un hogar tomando en consideración que la iluminación no cambie constantemente.
- Puede ser utilizado por cualquier tipo de usuario ya sea este niño o niña, adolescente, adulto con piel clara u oscura.
- Puede trabajar por tiempo ilimitado en un ambiente propicio para él.
- Reconocerá tanto el número o la letra que se realice con la mano siempre y cuando esta esté bien hecha.

5.4.2. Limitaciones

- El programa enseña únicamente los números del uno al diez y las letras del alfabeto en Lenguaje de Señas Ecuatoriana, las letras como los números en LSEC que requieren de movimiento no son enseñados por parte del programa los mismos que son el número 10 y las letras j ll ñ rr z.
- El programa presenta complicaciones en lugares donde la iluminación cambia constantemente.

5.5. Validación de la hipótesis

La hipótesis formulada para la realización del presente proyecto es la siguiente:

El diseño e implementación de un sistema de visión artificial permitirá la enseñanza de las letras del abecedario y números del uno al diez en Lenguaje de Señas Ecuatoriana

- **Variable independiente**

Enseñanza del alfabeto y números del uno al diez en Lenguaje de Señas Ecuatoriana

- **Variable dependiente**

Sistema de visión artificial

Para la validación de la hipótesis se utiliza la prueba de Chi-Cuadrado la misma que permite verificar si dos variables se encuentran relacionadas o no.

Lo primero que se debe realizar es fijar la hipótesis nula (H_0) y la hipótesis de alternativa (H_1)

Hipótesis nula (H₀): El sistema de visión artificial no permitirá la enseñanza del alfabeto y números del uno al diez en Lenguaje de Señas Ecuatoriana.

Hipótesis de alternativa (H₁): El sistema de visión artificial permitirá la enseñanza del alfabeto y números del uno al diez en Lenguaje de Señas Ecuatoriana.

Se realiza la tabla de frecuencia de valores observados (tabla 15) utilizando los datos de la tabla 14.

Tabla 15
Frecuencias de valores observados

	Bien >70% <80%	muy bien >80%	Total
Resultados de la evaluación de los números en LSEC	3	7	10
Resultados de la evaluación del alfabeto en LSEC	8	2	10
Total	11	9	20

Una vez encontradas las frecuencias de los valores observados se procede al cálculo de las frecuencias de valores esperados mediante la utilización de la ecuación 12 (Navidi, 2006).

$$E_{ij} = \frac{O_i * O_j}{O}$$

Ecuación 12 Cálculo de la frecuencia esperada

O_i = Frecuencia observada fila

O_j = Frecuencia observada columna

E_{ij} = Frecuencia esperada

O = Total

En la tabla 16 se observan los valores de esperados

Tabla 16
Frecuencias de valores esperados

	Bien >70% <80%	muy bien >80%
Resultados de la evaluación de los números en LSEC	5,5	4,5
Resultados de la evaluación del alfabeto en LSEC	5,5	4,5

Ahora se obtiene el valor de Chi Cuadrado calculado mediante la utilización de la ecuación 13 (Navidi, 2006).

$$X_{cal}^2 = \sum \frac{(O_{ij} - E_{ij})^2}{E_{ij}}$$

Ecuación 13 Chi-Cuadrado calculado

En la tabla 17 se presentan los resultados obtenidos al aplicar la ecuación 13 al conjunto de datos.

Tabla 17
Chi-Cuadrado calculado

	Bien >70% <80%	muy bien >80%
Resultados de la evaluación de los números en LSEC	1,1364	1,3889
Resultados de la evaluación del alfabeto en LSEC	1,1364	1,3889
Total	2,2727	2,7778
Total	5,0505	

Ahora se procede al cálculo del grado de libertad mediante la utilización de la ecuación 14.

$$V = (\text{Cantidad de filas} - 1) * (\text{Cantidad de columnas} - 1)$$

Ecuación 14 Grado de libertad

Entonces:

$$V = (2-1) * (2-1) = 1$$

Por lo general se trabaja con un valor de significancia de $p=0.05$ lo que indica un alto rendimiento del sistema de visión artificial, según los datos obtenidos de la tabla que se tiene en el Anexo D (Distribución Chi-Cuadrado) el valor tabulado para un grado de libertad y con un $p = 0.05$ es de 3.841 en cambio el valor de Chi-Cuadrado calculado es de 5.050 entonces: $X_{calc} > X_{tab}$ por tanto se acepta la hipótesis alternativa y se rechaza la hipótesis nula.

Por tal motivo queda verificada la hipótesis planteada ya que el sistema de visión artificial para la interpretación de lenguaje en señas permitirá la enseñanza del alfabeto y números del uno al diez en Lenguaje de Señas Ecuatoriana

5.6. Costos

Los costos utilizados en el desarrollo del sistema de visión artificial para la interpretación del alfabeto y números del uno al diez en Lenguaje de Señas Ecuatoriana se presentan en la tabla 15.

Tabla 18
Costos de implementación del programa

Orden	Descripción	Cantidad	Valor unitario	Valor total
1	horas de ingeniería	480	4	1920
			total	\$1920

Además de lo detallado en la tabla 15 se hizo uso de un computador Acer Aspire 5742G.

CAPÍTULO VI

Conclusiones y Recomendaciones

En este capítulo se presentan las conclusiones y recomendaciones que se recogen luego de haber culminado con el sistema de visión artificial para la interpretación del alfabeto y números del uno al diez en Lenguaje de Señas Ecuatoriana mediante la utilización de software libre.

6.1. Conclusiones

- Se ha desarrollado un sistema de visión artificial encargado de la interpretación del alfabeto y números del uno al diez en Lenguaje de Señas Ecuatoriana mediante la utilización de software libre.
- Se ha realizado una búsqueda bibliográfica adecuada permitiendo conocer los diferentes softwares que pueden ser utilizados en el desarrollo de una aplicación de Visión Artificial.
- El sistema ha sido desarrollado mediante la ayuda de las librerías de OpenCV las mismas que permitieron un adecuado tratamiento de las imágenes adquiridas a través de la cámara web posibilitando con esto que el sistema sea utilizado en varios ambientes.
- El sistema puede ser utilizado ya sea por niños o niñas, adolescentes, personas adultas, adultos mayores sin importar su cultura su lugar de origen o su contextura corporal es decir funciona para todo tipo de usuario.
- El sistema de visión artificial encargado de la interpretación del alfabeto y números del uno al diez en Lenguaje de Señas Ecuatoriana ha sido implementado gracias a la ayuda de Redes Neuronales Artificiales las mismas que son utilizadas en el aprendizaje de las características más representativas de cada letra del abecedario y de cada número comprendido entre el uno y el diez.
- Se ha conseguido entrenar dos agentes inteligentes los mismo que se los conoce en la investigación como clasificadores, cada agente se encuentra conformado por una Red Neuronal Artificial de 3 capas

se crean dos clasificadores uno encargado del reconocimiento de los números del uno al diez y el otro encargado del reconocimiento alfabeto en Lenguaje de Señas Ecuatoriana.

- Se establece un porcentaje de confiabilidad superior al 90% a los clasificadores creados a partir del entrenamiento de las Redes Neuronales Artificiales con la ayuda de imágenes tomadas de las manos formando los diferentes números como también de cada una de las letras de personas que conocen muy bien el Lenguaje de Señas Ecuatoriana como son los integrantes de la FENASEC y del INAL.
- Es un sistema que posee una interfaz gráfica de fácil entendimiento y gran accesibilidad ya que el promedio de tiempo para que un usuario se familiarice con la interfaz es de 2.6 minutos.
- Con el análisis de resultados de la investigación se confirma que el sistema enseña ya que se aplica una prueba a los diferentes usuarios que utilizaron el programa dando como resultado un porcentaje de efectividad en el aprendizaje mayor al 70%.
- El sistema desarrollado predice de forma efectiva un número o letra bien realizada caso contrario será mal interpretada por cualquier otra con la que más similitud posea.
- El sistema desarrollado brinda un ambiente interactivo para el aprendizaje tanto del alfabeto como de los números del uno al diez en Lenguaje de Señas Ecuatoriana.

6.2. Recomendaciones

- Es importante elegir el ambiente conveniente donde el sistema funcione a su máxima capacidad evitando ejecutarlo en lugares donde la iluminación cambie constantemente.
- Se recomienda utilizar computadores de gama alta a partir de procesadores Intel core i5 o superiores para asegurar la eficiencia del sistema.
- Si al ambiente elegido para el aprendizaje se le adapta iluminación fija mejorará el tratamiento de las imágenes, así como su posterior clasificación y reconocimiento se obtendrá mejores resultados por parte del programa.
- Colocar el computador a utilizar en una superficie fija procurando una vez ejecutado el programa no moverla o si se lo hace esperar un tiempo aproximado de 20 segundos para que el programa se adapte a la nueva escena y con esto evitar que se produzcan imperfecciones en el reconocimiento ya sea de la letra o número en Lengua de Señas Ecuatoriana.
- Si se utiliza una cámara web externa procurar no moverla una vez ejecutado el programa y si se lo hace esperar un tiempo aproximado de 20 segundos para que el programa se adapte a la nueva escena y con esto evitar que se produzcan imperfecciones en el reconocimiento.
- Tomarse el tiempo que necesite el usuario para el aprendizaje ya sea de los números o de las letras del abecedario en Lenguaje de Señas Ecuatoriana ya que la finalidad del programa es el aprendizaje de estos dos parámetros y lo más importante que el usuario se divierta aprendiendo.

REFERENCIAS BIBLIOGRÁFICAS

- Aguilar J., J. (2015, mayo). Introducción a las Redes Neuronales Artificiales. Recuperado el 03 de abril de 2017, a partir de http://www.flacsoandes.edu.ec/comunicacion/aaa/imagenes/publicaciones/pub_25.pdf
- Alarcón, V., Fonseca, M., & Rosas, N. (2012, julio 10). Lengua de señas: Definición [Blogspot]. Recuperado el 28 de noviembre de 2016, a partir de <http://senasupn.blogspot.com/2012/07/definicion.html>
- Alegre, E., Pajares, G., & de la Escalera, A. (Eds.). (2016). *Conceptos y Métodos en Visión por Computador*. España. Recuperado el 22 de marzo de 2017 a partir de <http://intranet.ceautomatica.es/sites/default/files/upload/8/files/ConceptosyMetodosenVxC.pdf>
- Arshad, S., & Alam, N. (2015). An Automated Bengali Sign Language Recognition System Based on Fingertip Finder Algorithm. Recuperado el 03 de marzo de 2017, a partir de http://www.academia.edu/14445087/An_Automated_Bengali_Sign_Language_Recognition_System_Based_on_Fingertip_Finder_Algorithm
- Blanco Garcia-Moreno, J. L. (2013, abril 10). La mirada del Golem: Librerías de visión artificial. Recuperado el 20 de marzo de 2017, a partir de <http://miradadelgolem.blogspot.com/2013/04/librerias-de-vision-artificial.html>
- Brahmbhatt, S. (2013). *Practical OpenCV*. Apress.
- Bueno, A. B., & Limiñana, A. R. (2002). *Guía didáctica de psicología de la intervención social*. Editorial Club Universitario.

- Cauich, C. (2012). Detección de Bordes Visión Artificial. Recuperado el 27 de marzo de 2017, a partir de http://www.academia.edu/24190825/Detecci%C3%B3n_de_Bordes_Visi%C3%B3n_Artificial_13_03_2012
- Chacón Mosquera, E. F. (2013). Desarrollo de una interfáz para el reconocimiento automático del lenguaje de signos. Recuperado el 03 de marzo de 2017, a partir de <http://repositorio.espe.edu.ec/jspui/handle/21000/8827>
- Chiguano, F. E., & Moreno, N. (2011, junio 16). *Diseño e implementación de un sistema traductor de lenguaje de señas de manos a un lenguaje de texto mediante visión artificial en un ambiente controlado* (Thesis). QUITO/EPN/2011. Recuperado el 03 de marzo de 2017, a partir de <http://bibdigital.epn.edu.ec/handle/15000/3915>
- CONADIS, CORDICOM, & FENASEC. (2015). MANUAL PRÁCTICO PARA INTÉRPRETES EN LENGUA DE SEÑAS ECUATORIANA. Recuperado el 17 de marzo de 2017, a partir de http://www.cordicom.gob.ec/wp-content/uploads/downloads/2016/04/Manual_de_interprete_de_lengua_de_senas_ecuatoriana.pdf
- Constante, P., & Gordón, A. (2015, agosto 12). *Diseño e implementación de un sistema de visión artificial para clasificación de al menos tres tipos de frutas* (Thesis). Quito, 2015. Recuperado el 03 de abril de 2017, a partir de <http://bibdigital.epn.edu.ec/handle/15000/11368>
- Cui, D. (2009). *Hand Detection and Gesture Recognition using ASL Gestures*. MASSEY, Auckland. Recuperado el 03 de marzo de 2017 a partir de

http://www.massey.ac.nz/~albarcza/ResearchFiles/DaKuanCui_PGDi_p_2009.pdf

- Díez, R. P., Gómez, A. G., & Martínez, N. de A. (2001). *Introducción a la inteligencia artificial: sistemas expertos, redes neuronales artificiales y computación evolutiva*. Universidad de Oviedo.
- ebay. (2016, julio 12). Acer 5742 PEW71 Webcam Camera Original Genuine 9157H1535563 CNF9157 CK77 94V-0. Recuperado el 7 de marzo de 2017, a partir de <http://www.ebay.es/itm/Acer-5742-PEW71-Webcam-Camera-Original-Genuine-9157H1535563-CNF9157-CK77-94V-0-/331906775699>
- Escobar, R. (2014). Redes neuronales, procesos cognoscitivos y análisis de la conducta. *Conductual*, 2(1), 23–43.
- Escudeiro, P., Escudeiro, N., Reis, R., Lopes, J., Norberto, M., Baltasar, A. B., ... Bidarra, J. (2015). Virtual Sign – A Real Time Bidirectional Translator of Portuguese Sign Language. *Procedia Computer Science*, 67, 252–262. <https://doi.org/10.1016/j.procs.2015.09.269>
- Esqueda, J. J., & Palafox, L. E. (2005). *Fundamentos para el procesamiento de imágenes*. UABC.
- FENASEC. (2012). Glosario Básico de Lengua de Seña Ecuatoriana. Recuperado el 07 de marzo de 2017, a partir de <https://dl.dropboxusercontent.com/u/89448753/Glosario%20B%C3%A1sico%20de%20Lengua%20de%20Se%C3%B1as.pdf>
- Flórez López, R., & Fernández, J. M. (2008). *Las Redes Neuronales Artificiales*. Netbiblo.

- García Posada, M. (2015). Diseño de prototipo de recogida automatizada de bolos mediante brazo robótico y visión artificial. Recuperado el 21 de marzo de 2017, a partir de <http://digibuo.uniovi.es/dspace/handle/10651/29906>
- García Santillán, I. D. (2008). *Visión artificial y procesamiento digital de imágenes usando Matlab*. Ibarra. Recuperado el 03 de abril de 2017, a partir de <https://es.scribd.com/doc/226655280/Vision-Artificial-y-Procesamiento-Digital-de-Imagenes-Usando-Matlab>
- Gestal Pose, M. (2009, diciembre). Introducción a las Redes Neuronales Artificiales. Recuperado el 03 de abril de 2017, a partir de <http://sabia.tic.udc.es/mgestal/cv/RNATutorial/TutorialRNA.pdf>
- Gómez Correa, A. M., & Trejos Largo, L. F. (2014). Análisis de vídeo para la segmentación y clasificación de jugadores en juegos de fútbol. Recuperado el 15 de marzo de 2017, a partir de <http://repositorio.utp.edu.co/dspace/handle/11059/4709>
- Itseez. (2016). OpenCV | OpenCV. Recuperado el 29 de noviembre de 2016, a partir de <http://opencv.org/>
- Larriva Calle, D. (2011). Diseño de Software Educativo incluyente. Recuperado 03 de marzo de 2017, a partir de <http://dspace.uazuay.edu.ec:8080/handle/datos/2960>
- López, E., Antón. (2009, julio 20). *Reconocimiento automático de lenguaje de signos: Lenguaje ASL*. Universitat de Barcelona, Barcelona. Recuperado el 01 de marzo de 2017 a partir de <http://www.maia.ub.es/~sergio/linked/enrique09.pdf>

- MathWorks. (2012). Thumbnail.11.640.360.jpg (640x360). Recuperado el 21 de marzo de 2017, a partir de https://www.mathworks.com/content/dam/mathworks/videos/o/4484672947001/_jcr_content/renditions/Thumbnail.11.640.360.jpg
- Matich, D. J. (2001, marzo). Redes Neuronales: Conceptos Básicos y Aplicaciones. Recuperado el 31 de marzo de 2017, a partir de https://www.frro.utn.edu.ar/repositorio/catedras/quimica/5_anio/orientadora1/monograis/matich-redesneuronales.pdf
- Mendoza, D. (2015). *OpenCV con Qt y C++*. Recuperado el 19 de abril de 2017 a partir de <https://www.youtube.com/watch?v=ZfIQ-ujcib4>
- Ministerio de Salud Pública. (2017, febrero). Estadísticas | CONADIS. Recuperado el 17 de marzo de 2017, a partir de <http://www.consejodiscapacidades.gob.ec/estadistica/index.html>
- National Instruments Corporation. (2017). ¿Qué puede hacer con LabVIEW? - National Instruments. Recuperado el 21 de marzo de 2017, a partir de <http://www.ni.com/labview/why/esa/>
- Navarro, F. M., Corrales, J. D., Villar, M. L. G., Torre, A. S. D. L., Cifredo, E. A. P., & Olivas, L. M. (2006). *Técnicos de soporte informático de la comunidad de castilla y león. Temario volumen i*. MAD-Eduforma.
- Navidi, W. (2006). *Estadística para ingenieros y para científicos* (Pimera). México: McGraw-Hill.
- opencv dev team. (2014, noviembre 10). Redes Neuronales - OpenCV documentación 3.0.0-dev. Recuperado el 5 de abril de 2017, a partir de http://docs.opencv.org/3.0-beta/modules/ml/doc/neural_networks.html

- Pardo, J. (2010). INVESTIGACIÓN CIENTÍFICA: LENGUAJE, LENGUA Y HABLA. Recuperado el 23 de enero de 2017, a partir de <http://gp6cia2010.blogspot.com/p/blog-page.html>
- Piedra Fernández, J. A. (2008). *Aplicación de los sistemas neurodifusos a la interpretación automática de imágenes de satélite*. Universidad Almería.
- Raj S., N. (2013, agosto 23). OCR using Artificial Neural Network (OpenCV) – Part 1 |. Recuperado el 5 de abril de 2017, a partir de <http://archive.is/0J4Vm>
- Riotordo, A. (2012, octubre 3). ¿Cultura Sorda? - A.P.A.S.U. Recuperado el 23 de enero de 2017, a partir de <http://www.apasu.org.uy/b-sordos/d-cultura-sorda>
- Rodríguez Escudero, A. A. (2012, agosto 3). Imagen Digital: Módulos de la librería OpenCV. Recuperado el 21 de marzo de 2017, a partir de <http://digitimagen.blogspot.com/2012/08/modulos-de-la-libreria-opencv.html>
- Sarrionandia, G. E., & Minguillón, C. G. (1995). *Las necesidades educativas especiales del niño con deficiencia auditiva*. Ministerio de Educación.
- Secretaría Nacional de Planificación y Desarrollo. (2013). Objetivos Nacionales para el Buen Vivir - Plan Nacional 2013 - 2017. Recuperado el 17 de marzo de 2017, a partir de <http://www.buenvivir.gob.ec/objetivos-nacionales-para-el-buen-vivir>
- Segunda Cohorte del Doctorado en Seguridad Estratégica. (2014). *Seguridad de la Información: Revista de la Segunda Cohorte del Doctorado en*

Seguridad Estratégica. Segunda Cohorte Doctorado en Seguridad Estratégica.

Sivisapa Aguilera, L. G. (2014). *Visión artificial para el reconocimiento del lenguaje de señas*. Recuperado el 03 de marzo de 2017, a partir de <http://dspace.unl.edu.ec//handle/123456789/14233>

Sobrado Malpartida, E. Á. (2011). *Sistema de visión artificial para el reconocimiento y manipulación de objetos utilizando un brazo robot*. Pontificia Universidad Católica del Perú. Recuperado el 14 de marzo de 2017, a partir de <http://tesis.pucp.edu.pe/repositorio/handle/123456789/68>

Sucar, L. E., & Gómez, G. (2010). *Visión Computacional*. Recuperado el 31 de marzo de 2017, a partir de <http://ccc.inaoep.mx/~esucar/Libros/vision-sucar-gomez.pdf>

Valencia, G. (2012, abril). EL SOFTWARE LIBRE EN EL ECUADOR Lu 8-12: CONCEPTO [Blogspot]. Recuperado el 29 de noviembre de 2016, a partir de <http://andreitamedina.blogspot.com/2012/04/concepto.html>

Vintimilla, M. (2014). *Desarrollo e implementación de una aplicación que traduzca el abecedario y los números del uno al diez del lenguaje de señas a texto para ayuda de discapacitados auditivos mediante dispositivos móviles Android*. Recuperado el 03 de marzo de 2017, a partir de <http://repositorio.espe.edu.ec/jspui/handle/21000/8873>

ANEXOS



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

**DEPARTAMENTO DE CIENCIAS DE LA ENERGÍA
MECÁNICA**

CARRERA DE INGENIERÍA MECATRÓNICA

CERTIFICACIÓN

Se certifica que el presente trabajo fue desarrollado por el señor: **JAVIER PATRICIO PICHUCHO RODRÍGUEZ**.

En la ciudad de Latacunga, a **los 19 días del mes de mayo del 2017**.

Ing. Patricia Constante, M.Sc.

DIRECTOR DEL PROYECTO

Aprobado por:

Ing. Vicente Hallo

DIRECTOR DE CARRERA

Dr. Rodrigo vaca

SECRETARIO ACADÉMICO