

UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA



TESIS DE GRADO

**REDES NEURONALES PARA LA INTERPRETACIÓN DEL LENGUAJE
DE SEÑAS**

PARA OPTAR AL TÍTULO DE LICENCIATURA EN INFORMÁTICA
MENCIÓN: INGENIERÍA DE SISTEMAS INFORMÁTICOS

POR : MARIO JUSTINO ARGOLLO RAFAEL
TUTOR METODOLÓGICO: M. Sc. MIGUEL COTAÑA MIER
ASESOR : M. Sc. ROSA FLORES MORALES

LA PAZ – BOLIVIA
2013



**UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA**



LA CARRERA DE INFORMÁTICA DE LA FACULTAD DE CIENCIAS PURAS Y NATURALES PERTENECIENTE A LA UNIVERSIDAD MAYOR DE SAN ANDRÉS AUTORIZA EL USO DE LA INFORMACIÓN CONTENIDA EN ESTE DOCUMENTO SI LOS PROPÓSITOS SON ESTRICTAMENTE ACADÉMICOS.

LICENCIA DE USO

El usuario está autorizado a:

- a) visualizar el documento mediante el uso de un ordenador o dispositivo móvil.
- b) copiar, almacenar o imprimir si ha de ser de uso exclusivamente personal y privado.
- c) copiar textualmente parte(s) de su contenido mencionando la fuente y/o haciendo la referencia correspondiente respetando normas de redacción e investigación.

El usuario no puede publicar, distribuir o realizar emisión o exhibición alguna de este material, sin la autorización correspondiente.

TODOS LOS DERECHOS RESERVADOS. EL USO NO AUTORIZADO DE LOS CONTENIDOS PUBLICADOS EN ESTE SITIO DERIVARA EN EL INICIO DE ACCIONES LEGALES CONTEMPLADOS EN LA LEY DE DERECHOS DE AUTOR.

*A mi Madre
y a la memoria de mi padre
su amor incondicional es la fuerza que
guía mi vida.*

Agradecimientos

A mi familia, por el amor que me brindan y por ser el soporte de mi vida.

Mi mayor gratitud a M. Sc. Rosa Flores Morales, por su paciencia, comprensión, apoyo, y confianza, de quien recibí el rumbo para alcanzar mis objetivos.

A mi tutor M. Sc. Miguel Cotaña Mier, por el apoyo y la ayuda prestada a la realización de este trabajo

A el Licenciado Juan Cayoja Cortez, por su motivación y sus palabras de aliento.

Agradezco a mi compañero Lotar por sus sabios consejos y su desinteresada ayuda.

Y en especial a la persona que inspiro, el desarrollo de esta investigación, que con tan solo su mirada expresa todo lo que no puede decir con la boca.

RESUMEN

Los diferentes idiomas existentes alrededor del mundo no siempre cumplen con el objetivo de la comunicación, entre estos idiomas existe uno en particular que puede lograr que dos personas de regiones e idiomas diferentes lleguen a ser capaces de comunicarse fluidamente, este idioma se diseñó para las personas que sufren sordera o invalidez del habla. Aunque el lenguaje de señas permite la comunicación entre personas con estas incapacidades, no siempre estas personas se pueden comunicar con su ambiente social, porque la mayoría de las personas que no sufren estas incapacidades ignoran el lenguaje de señas.

Este trabajo presenta un sistema que tiene el objetivo de comunicar a las personas sordas o mudas con personas sin estas incapacidades a través de la interpretación del lenguaje de señas. Para lograr este objetivo se utilizó métodos de proceso de imágenes y redes neuronales que son parte del IDE de desarrollo de Matlab. El prototipo del sistema captura la imagen de la seña hecha con un guante verde, entonces se procedió con el tratamiento de la imagen hasta obtener los arreglos de la matriz y el vector característico relevante. Con los resultados obtenidos se procedió a interpretar la seña y la reproducción oral subsecuente.

Finalmente, con el prototipo desarrollado se realizaron experimentos, éstos demostraron la fiabilidad del prototipo, obteniendo resultados satisfactorios bajo ciertas condiciones controladas de iluminación y distancia.

Palabras clave: Lenguaje de señas, redes neuronales, tratamiento de imágenes.

ABSTRACT

The different existing languages around the world do not always fulfill the objective of communication, among these languages it exists one in particular that can accomplish that two people from different regions and dissimilar languages become able to communicate fluently among them, this language was designed for people that suffer deafness or speechless disabilities. Although, sign language allows communication between people with those disabilities, it is not always able to communicate these people with their social environment, because most of the people that do not suffer these disabilities ignore sign language.

This work presents a system that has the objective of communicate deaf or speechless people with people without those disabilities through the interpretation of sign language. To accomplish this objective it was utilized image processing methods and neural networks that are part of Matlab development IDE. The prototype of the system gets the image of the sign made with a green glove, and then it proceeds with the image treatment until to obtain the matrix arrangements and the relevant characteristic vector. With the obtained results it proceeded to interpret the sign and the subsequent oral reproduction.

Finally, with the developed prototype it performed experiments; these proved the reliability of the prototype, obtaining satisfactory results under certain controlled conditions of illumination and distance.

Keywords: Sign Language, neural networks, image processing.

ÍNDICE

CAPÍTULO I MARCO REFERENCIAL

	Página
1.1. INTRODUCCIÓN.....	2
1.2. ANTECEDENTES.....	3
1.3. PLANTEAMIENTO DEL PROBLEMA.....	5
1.3.1 FORMULACIÓN DEL PROBLEMA.....	6
1.4. HIPOTESIS.....	6
1.5. OBJETIVOS.....	6
1.5.1. Objetivo general.....	6
1.5.2. Objetivos específicos.....	6
1.6. JUSTIFICACIÓN.....	7
1.6.1. Justificación Social.....	7
1.6.2. Justificación Técnica.....	7
1.7. LIMITES Y ALCANCE.....	8
1.8. DISEÑO METODOLÓGICO.....	8
1.7.1. Método de investigación.....	8

CAPITULO II MARCO TEÓRICO

2.1. INTRODUCCIÓN.....	12
2.2. LENGUAJE DE SEÑAS.....	12
2.2.1. Origen de las lenguas de señas.....	12
2.2.2. Clasificación de las lenguas de señas.....	13

2.2.3 Dactilología o alfabeto gestual.....	14
2.3 REDES NEURONALES.....	16
2.3.1. Funcionamiento.....	16
2.3.2, Diseño y programación de una RNA.....	17
2.3.3 Estructura.....	17
2.3.4. Ventajas.....	18
2.3.5. Tipologías de las redes neuronales.....	19
2.4. TRATAMIENTO DE IMÁGENES.....	21
2.5. CROMA.....	33
2.6. CONVOLUCIÓN.....	34
2.6.1. Convolución Gaussiana.....	34
2.7. MATLAB.....	35
2.8. RED BACKPROPAGATION UTILIZANDO MATLAB.....	39
2.9. TOOLBOX DE AUDIO EN MATLAB.....	44

CAPITULO III
ANÁLISIS Y DESARROLLO

3.1. INTRODUCCIÓN.....	47
3.2. ANÁLISIS PARA EL DESARROLLO DEL PROTOTIPO.....	47
3.3. ETAPA 1: Generación y entrenamiento de Redes Neuronales.....	48
3.3.1. Arquitectura y generación de la red neuronal.....	48
3.3.1.1. Red Neuronal.....	49
3.3.2. Entrenamiento de la red neuronal.....	56
3.3.3. Simulación.....	57
3.4. ETAPA 2: captura y pre-procesamiento.....	58
3.4.1. Extracción de áreas de interés.....	58
3.4.2. Eliminación de ruido.....	61

3.4.3. Recorte.....	62
3.5. ETAPA 3: procesamiento.....	63
3.6. ETAPA 4: reconocimiento.....	64
3.7. ETAPA 5: reproducción de sonido.....	65
3.8. MÓDULO DE ENSEÑANZA DEL LENGUAJE DE SEÑAS.....	66

CAPITULO IV
PRUEBAS Y RESULTADOS

4.1. INTRODUCCIÓN.....	69
4.2. EVALUACIÓN.....	69
4.2.1. Casos experimentales.....	69
4.2. ANÁLISIS DE RESULTADOS.....	83

CAPITULO V
CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES.....	87
5.2. RECOMENDACIONES.....	88

BIBLIOGRAFIA

BIBLIOGRAFIA.....	90
-------------------	----

ÍNDICE DE FIGURAS

	Página
Fig. 1 Traductor en-línea de lenguajes de señas.....	4
Fig. 2 Alfabeto dactilológico.....	14
Fig. 3 Croma.....	33
Fig. 4 Formula de la convolución gaussiana.....	35
Fig. 5 Representación de una imagen a escala de grises en MatLab.	36
Fig. 6 Representación de una imagen a color RGB en MatLab.....	37
Fig. 7 Representación de arreglos RGB en Matlab.....	37
Fig. 8 Función de transferencia log-sigmoidal.....	40
Fig. 9 Función de transferencia tan-sigmoidal.....	41
Fig. 10 Función de transferencia lineal.....	41
Fig. 11 Gráfica de intensidad de zonas de la imagen.....	49
Fig. 12 Grafica experimento 1.....	50
Fig. 13 Grafica experimento 2.....	51
Fig. 14 Primera red neuronal, experimento 3.....	53
Fig. 15 Segunda red neuronal, experimento 3.....	53
Fig. 16 Tercera red neuronal, experimento 3.....	53
Fig. 17 Criterio de la raíz.....	55
Fig. 18 Objeto 2, 3 y 4 del gui de Entrenamiento.....	56
Fig. 19 Front-end de entrenamiento red neuronal.....	57
Fig. 20 Captura de imagen.....	59
Fig. 21 Algoritmo croma.....	60
Fig. 22 Captura de objeto.....	60
Fig. 23 Eliminación de ruido.....	61
Fig. 24 Recorte de objeto.....	62

Fig. 25 Extracción de característica.....	64
Fig. 26 Reproducción de sonido.....	65
Fig. 27 Enseñanza del lenguaje de señas.....	67
Fig. 28 Escasa iluminación, guante de lana.....	70
Fig. 29 Experimento 2.....	71
Fig. 30 Captura de la seña con guante de lana.....	72
Fig. 31 Captura de la seña con guante de goma.....	73
Fig. 32 Experimento 5, excesiva iluminación, guante de lana.....	74
Fig. 33 Experimento 6, excesiva iluminación, guante de goma.....	75
Fig. 34 Experimento 7.....	76
Fig. 35 Experimento 8.....	77
Fig. 36 Experimento 9.....	78
Fig. 37 Experimento 9.....	79
Fig. 38 Experimento 11.....	80
Fig. 39 Experimento 12.....	81
Fig. 40 Experimento 13.....	82
Fig. 41 Experimento 14.....	83

ÍNDICE DE TABLAS

	Página
Tabla. 1. Formatos y extensiones soportados por Matlab.....	38
Tabla. 2. Respuesta experimento 1.....	50
Tabla. 3. Respuesta experimento 2.....	52
Tabla. 4. Respuesta experimento 3.....	54



INTRODUCCIÓN

CAPITULO 1

MARCO REFERENCIAL

1.1. INTRODUCCIÓN

La comunicación es el pilar fundamental para la existencia de la sociedad, ya que facilita la transmisión de los conocimientos, costumbres y valores culturales, sin embargo, por distintas razones no todas las personas gozan de las mismas facultades.

Tanto la sordera como la invalidez del habla son obstáculos que excluyen a un grupo social impidiéndole comunicar sus emociones ideas e inquietudes. Ante esto surge el lenguaje de señas como un medio que el ser humano utiliza para poder integrar a las personas que no pueden comunicarse de forma oral, este lenguaje facilita el desenvolvimiento cotidiano de dichas personas.

El artículo del periódico el Deber “Personas sordas exigen trabajo y más intérpretes” (Roxana Escobar N, 2010), se refirió a las personas con discapacidad auditiva:

“La comunidad de las personas sordas cerró su semana mundial convencida de que, con la puesta en marcha de la ley contra el racismo y la discriminación, nunca más permitirán que se los llame sordomudos, sino personas sordas. El desafío es superar la mayor barrera que significa comunicarse sin perder su identidad, (hacerlo a través del lenguaje de señas), igualdad de condiciones para acceder a la educación escolar y superior y al trabajo. “Nos llaman sordomudos, pero nosotros tenemos identidad y cultura propia; nos comunicamos a través de la vista y de señas; nuestros ojos son nuestros oídos”. Otro de los puntos que se discutieron en el encuentro es la necesidad de contar con intérpretes profesionales y reconocidos, ya que la mayor barrera es que no pueden comunicarse con la población oyente. Tampoco informarse, porque en la

mayoría de los canales de televisión no emiten las noticias en lenguaje de señas y el celular lo usan sólo para enviar mensajes de textos. La limitante de la comunicación es un obstáculo a la hora de acceder a un empleo. Sólo ocupan puestos de trabajo inferior y no jerárquico”.

En el presente proyecto se desarrolló un software, con el que se interprete el lenguaje de señas para la posterior reproducción oral. El software facilitará el aprendizaje del lenguaje de señas en hogares y/o en aulas escolares. Para este fin se usaron técnicas de procesamiento de imágenes y redes neuronales, para la construcción del prototipo, el software matemático de desarrollo integrado (IDE), Matlab.

Al final del trabajo se evaluó el desempeño del prototipo, realizando pruebas experimentales. Las respuestas a las señas dan como resultado una letra reproducida en audio y en texto indicando la letra a la que corresponde la seña

1.2. ANTECEDENTES

La búsqueda de dar mejores condiciones a personas con sordera o invalidez del habla lleva a modificar muchos aspectos de la sociedad, en países como Colombia y Argentina se plantean políticas para lograr una sociedad más justa para estas personas. En Bolivia, a partir la “**LEY GENERAL PARA PERSONAS CON DISCAPACIDAD**” se trata de dar mejores condiciones de vida a las personas con diferentes tipos de discapacidad.

En diferentes países se desarrollan proyectos referentes a la enseñanza del lenguaje de señas, tales son los casos de:

- **Maya | Software para convertir texto a lenguaje de signos**

“Maya es la primera traductora simultánea en lengua de signos desarrollada en 3D por una empresa cordobesa, Xul, capaz de reproducir 1.600 signos. Este software denominado Textosign está pensado para su integración en páginas web facilitando la accesibilidad a un millón de personas sordas en España, de las que 400.000 no saben leer debido a esta discapacidad. Recientemente Maya ha sido exportada al móvil convirtiéndose también en el primer diccionario de lenguaje de signos del mundo con avatar en tres dimensiones.” (Aprende lengua de signos, 2012)



Fig. 1. Traductor en-línea de lenguajes de señas

Fuente: Recorte, **Maya | Software para convertir texto a lenguaje de signos**,
Mayasoft, 2012

- **Traductor de español a la lengua de señas¹**

“Sistema on-line de la página oficial de Fundación Hetah (Herramientas tecnológicas para ayuda humanitaria). Este sistema de traducción es elaborado para mejorar la comunicación entre las personas sin limitaciones y las personas

¹ Portal de lenguajes de Colombia Traductor en-linea de lenguas de señas..Fundación Hetah
<<http://hetah.net/>>

con limitaciones auditivas, mediante Iris, nuestra intérprete animada. Se ingrese una frase en español en el cuadro de texto, luego se da clic en el botón traducir o de enter, a continuación Iris hará la traducción en lenguaje gestual. También se puede navegar a través de la animación por medio de los controles de velocidad y reproducción. Este aplicativo realiza una búsqueda de señas analizando la frase en su estructura gramatical, haciendo énfasis en el país origen y realizando el deletreo de las palabras en caso de no encontrar equivalencia” (Portal de lenguajes de Colombia, 2013).

Estas aplicaciones se encargan de buscar palabras que son escritas en un campo determinado y luego las representan en lenguaje de señas con un avatar que se encarga de realizar el movimiento de las manos.

1.3. PLANTEAMIENTO DEL PROBLEMA

Debido a que nuestra vida está llena de ruidos y estrés no se nos pasa por la mente que puede existir un mundo paralelo al nuestro, sin sonidos. La sociedad permite pasar a las personas sordas, totalmente desapercibidas. El lenguaje de señas ayuda a las personas sordas a llevar una vida con comunicación, ya que es el medio con el que podemos comunicarnos para poder transmitirles pensamientos.

Este lenguaje que podría incluir en mayor medida a las personas con sordera o invalidez del habla, es conocida por un número muy reducido de individuos sin estas discapacidades de nuestra sociedad, lo que hace difícil la comunicación con las personas sordas.

1.3.1. FORMULACIÓN DEL PROBLEMA

¿Las técnicas informáticas de reconocimiento de patrones, podrán traducir a sonidos el lenguaje de señas y de este modo lograr comunicar a una persona sorda o con invalidez del habla de forma comprensiva, con personas que no tienen esta discapacidad?

1.4. HIPÓTESIS.

Aplicando las redes neuronales y el tratamiento de imágenes digitales en el reconocimiento de caracteres se podrá realizar la interpretación del lenguaje de señas al lenguaje hablado.

1.5. OBJETIVOS.

1.5.1. OBJETIVO GENERAL

Desarrollar un prototipo que realice la interpretación del lenguaje de señas al lenguaje oral.

1.5.2. OBJETIVOS ESPECÍFICOS

- Capturar las señales del lenguaje mediante un medio digital.
- Realizar un tratamiento digital de la imagen.
- Implementar una red neuronal para la representación de las imágenes mediante patrones.

- Representar las imágenes con palabras, realizando así la interpretación del lenguaje de señas.

1.6. JUSTIFICACIÓN

1.6.1 Justificación Social

El presente trabajo es de carácter social, ya que se pretende implementar un software que permita mejorar la integración a la sociedad de las personas sordas o con invalidez del habla, y que al mismo tiempo será de mucha ayuda para promover el aprendizaje del lenguaje de señas en ambientes como el hogar y las aulas escolares.

Ya que el lenguaje de señas usado por las personas con discapacidad no es muy conocido por las personas comunes, dificultando así la comunicación y el libre desenvolvimiento de actividades cotidianas entre ambos grupos, este ha sido el motivo paraa realizar el prototipo del software para la comunicación a fin de facilitar la comunicación, y de hecho la interrelación, de estas personas.

1.6.2. Justificación Técnica

Las redes neuronales han tenido importantes aplicaciones como el reconocimiento de patrones, la predicción del mercado financiero, controladores para robots y muchas otras más.

Por esta razón se realizó un estudio profundo sobre las redes neuronales ya que tienen gran flexibilidad con la información de entrada y es posible tener respuestas en tiempo real.

1.7. LÍMITES Y ALCANCE

Los alcances y límites que se tendrán en el presente proyecto, serán los siguientes:

Realizar el prototipo basado en la captura de las imágenes de las señas del lenguaje, realizadas por la persona con discapacidad, estas imágenes serán procesadas para obtener los patrones de características de las imágenes.

- Se utilizaran redes neuronales para interpretar las imágenes procesadas.
- La interpretación del lenguaje de señas sin movimiento se realiza con imágenes tomadas por una Webcam en un ambiente controlado.
- Las señas se restringen al alfabeto dactilológico (sin movimiento).

1.8. DISEÑO METODOLÓGICO

1.8.1. Método de investigación

Para el desarrollo del proyecto se utilizará el Método sistémico.

El sistemismo que Bunge propone, establece que todo es un sistema o un componente de un sistema y define que para conocer un sistema, sea este físico, químico, biológico, psicológico o social, es conveniente aplicar el enfoque CESM. Este enfoque requiere de la construcción de un modelo que consiste en la descripción de: la composición (*C*), el entorno (*E*), la estructura (*S*) y el mecanismo (*M*) del sistema (Bunge, 2004).

- **La composición** de un sistema es la colección de sus partes (protones, neutrones y electrones en el sistema atómico; personas, empresas, clubes y barra de amigos en el sistema social) y se las llama componente.
- **El entorno** es la colección de cosas que modifican a los componentes del sistema o que resultan modificados por ellos, pero que no pertenecen a la composición (fotones que excitan al átomo de interés y el trigo que el hombre convierte en pan).
- **La estructura** es la colección de relaciones o vínculos que establecen los componentes. Los vínculos que se dan entre los componentes de un sistema constituyen la endoestructura, mientras que los establecidos entre los componentes y elementos del entorno conforman la exoestructura del sistema.
- **El mecanismo** es la colección de procesos que se dan dentro de un sistema y que lo hacen cambiar en algún aspecto (el mecanismo de radiación electromagnética de un átomo es un proceso en el que un electrón cambia de estado de energía, el comercio es un mecanismo económico de los sistemas sociales humanos). Más precisamente, si bien el conocimiento de un sistema concreto radica en la descripción de los cuatro aspectos mencionados, la explicación científica del comportamiento del mismo la brinda la descripción de su(s) mecanismo(s), es decir de los procesos de los cuales resultan la emergencia, la estabilidad, el cambio y la desintegración de un sistema.

Esto permite entender los sistemas en términos de acciones individuales y éstas en términos del contexto social, de igual manera, es posible estudiar los rasgos sistémicos como la cohesión, el equilibrio, el progreso y sus opuestos.

Mario Bunge asimismo considera que:

“Un enfoque es una forma de mirar las cosas y manejarlas, y está compuesto por un cuerpo de conocimientos existentes junto con un conjunto de problemas (problemática), un conjunto de objetivos y un conjunto de métodos (metódica) para enfrentar tales problemas, por la generalidad del enfoque sistémico, plantea que los únicos elementos que se pueden presentar en su conocimiento existente, son ciertas hipótesis filosóficas, las cuales consisten básicamente en un principio ontológico y su contraparte epistemológica. El primero es el principio de que toda cosa concreta es sistema o un componente de él. Su compañero epistemológico es la norma de que todo sistema debe estudiarse en su propio nivel, así como descomponerse en sus componentes interactuantes”. (Bunge, 1999, p.370).

La problemática del enfoque sistémico es la totalidad de problemas que se pueden plantear respecto de cualquier sistema, sus objetivos son describir, entender, predecir y controlar y su método incluye tanto el análisis como la síntesis, la generalización y la sistematización, el modelo matemático y la comprobación empírica, por lo que es aplicable a todas las disciplinas científicas y aplica los aspectos positivos del atomismo atendiendo los componentes individuales del sistema, así como los del holismo atendiendo la totalidad.



MARCO TEÓRICO

CAPITULO II

MARCO TEÓRICO

2.1. INTRODUCCIÓN

En el capítulo presente se realizarán todos los conceptos de las herramientas y teorías que se aplican para el desarrollo de la tesis.

2.2. LENGUAJE DE SEÑAS

El lenguaje de señas, o lengua de signos, es una lengua natural de expresión y configuración gesto-espacial y percepción visual (o incluso táctil por ciertas personas con sordo ceguera), gracias a la cual las personas sordas pueden establecer un canal de información básica para la relación con su entorno social, ya sea conformado por sordos u oyentes. Mientras que con el lenguaje oral la comunicación se establece en un canal vocal-auditivo, el lenguaje de señas lo hace por un canal gesto-viso-espacial, estos lenguajes son naturales y tienen estructuras gramaticales perfectamente definidas, e independientes de las lenguas orales con las que cohabitan (EcuRed).

2.2.1. Origen de las lenguas de señas

Aun cuando las lenguas de señas sean actualmente usadas casi exclusivamente entre las personas con sordera, su origen es tan antiguo como las lenguas orales o incluso más, en la historia de la aparición de la Humanidad, y también han sido y siguen siendo (EcuRed).

2.2.2 Clasificación de las lenguas de señas

Las lenguas de señas modernas al igual que las lenguas orales están sujetas al proceso universal de cambio lingüístico que hace que evolucionen con el tiempo y eventualmente una misma lengua puede evolucionar en lugares diferentes hacia variedades diferentes. De hecho, muchas de las lenguas modernas de señas pueden ser clasificadas en familias de lenguas de señas (Wikipedia):

- **Lenguas originadas en la antigua lengua de señas francesa.** Estas lenguas se remontan a las formas estandarizadas de lenguas de señas usadas en Italia y Francia desde el siglo XVII y sobre todo a partir del siglo XVIII en la educación de los sordos.
- **Lenguas de señas ibéricas,** que muestran similitudes con la antigua lengua de señas francesa pero cuyo origen no es bien conocido. Existen tres lenguas diferentes con cierta inteligibilidad mutua, la lengua de signos española (LSE), la lengua de signos catalana (LSC) y la Lengua de signos valenciana (LSCV)
- **Lenguas originadas en la lengua de señas británica (BSL),** que se diversificó durante el siglo XIX dando lugar a la lengua de señas australiana o Auslan, la lengua de señas de Nueva Zelanda (NZSL) y la lengua de señas de Irlanda del Norte.
- **Lenguas originadas en la lengua de señas alemana (DGS),** que se considera relacionada con la lengua de señas de la Suiza alemana (DSGS), la lengua de señas austriaca (ÖGS) y probablemente la lengua de señas israelí (ISL).
- **Lenguas originadas en la antigua lengua de señas de Kent,** usada durante el siglo XVII, que dio lugar a la lengua de señas usada en Martha's Vineyard

(Massachusetts) y que influyó de manera importante en la lengua de señas americana.

2.2.3 Dactilología o alfabeto gestual

Con frecuencia, el sordo en su comunicación recurre a la dactilología, que es una escritura en el espacio: cada letra del alfabeto latino es representada por un gesto realizado con la mano y los dedos. (González, 1991)



Fig. 2. Alfabeto dactilológico

Fuente: E. Buysens (1978: 38)

La dactilología es, según la clasificación de E. Buysens (1978: 38), una seña sustitutiva de segundo grado ya que recodifica la escritura, que es, a su vez, seña sustitutiva de primer grado con respecto a la lengua oral. (González, 1991)

Los casos en que el sordo utiliza el alfabeto manual son los siguientes:

- a) En la designación de algunos conceptos nuevos, que en una primera fase, no cuentan con un signo convenido. Muchos de los signos creados a partir de la dactilología son sustituidos por otros más convencionales; otros se pierden cuando el concepto deja de tener actualidad y no es funcional en el sistema social de los sordos. (González, 1991)
- b) Para precisar algunos significados en sus relaciones con los oyentes, sobre todo, cuando se trata de expresar antropónimos y topónimos en la comunicación con oyentes, ya que entre los sordos los nombres de personas y los nombres de lugares se expresan con signos propios de su lenguaje. (González, 1991)

Por tanto, el alfabeto gestual es, sobre todo, un recurso en la comunicación entre sordos y oyentes, en la que, por otra parte, se observa la constante actividad de extracodificación que realiza el sordo. Normalmente, le basta percibir las dos o tres primeras letras para captar lo que el emisor desean comunicarle. (González, 1991)

En el lenguaje de signos hay pocas expresiones de origen dactilológico si exceptuamos algunas realizaciones de palabras muy breves, que, por la rapidez de su articulación manual, se consideran casi convencionales, como SÍ, Y, O, TÉ. Otras expresiones de tipo dactilológico se refieren a ciertos conceptos muy nuevos, como ya se ha indicado. (González, 1991)

Algunos signos gestuales que designan ciudades reflejan su procedencia de la dactilología. Conservan tan sólo la articulación de una o dos letras manuales; por ejemplo, el signo SS para San Sebastián, V para Valladolid, CC para Cáceres. Tales realizaciones ponen de manifiesto, por otra parte, la influencia del entorno visual, en este caso, del sistema de matrícula de los automóviles. (González, 1991)

Ello es consecuencia de la influencia lógica del lenguaje verbal sobre el lenguaje de signos utilizado por minorías de sordos, que viven inmersos dentro de la sociedad de oyentes. (González, 1991)

2.3. REDES NEURONALES

Las redes de neuronas artificiales (denominadas habitualmente como RNA o en inglés como: "ANN") son un paradigma de aprendizaje y procesamiento automático inspirado en la forma en que funciona el sistema nervioso de los animales. Se trata de un sistema de interconexión de neuronas en una red que colabora para producir un estímulo de salida. En inteligencia artificial es frecuente referirse a ellas como redes de neuronas redes neuronales (Kiwix, 2011).

2.3.1. Funcionamiento

Unas redes neuronales consisten en simular las propiedades observadas en los sistemas neuronales biológicos a través de modelos matemáticos recreados mediante mecanismos artificiales (como un circuito integrado, un ordenador o un conjunto de válvulas). El objetivo es conseguir que las máquinas den respuestas similares a las que es capaz de dar el cerebro que se caracterizan por su generalización y su robustez (Kiwix, 2011).

Una red neuronal se compone de unidades llamadas neuronas. Cada neurona recibe una serie de entradas a través de interconexiones y emite una salida. Esta salida viene dada por tres funciones:

La función de propagación (también conocida como función de excitación), que por lo general consiste en el sumatorio de cada entrada multiplicada por el peso de su interconexión (valor neto). Si el peso es positivo, la conexión se denomina excitatoria; si es negativo, se denomina inhibitoria (Kiwix, 2011).

Una función de activación, que modifica a la anterior. Puede no excediendo este caso la salida la misma función de propagación (Kiwix, 2011).

La función de transferencia, que se aplica al valor devuelto por la función de activación se utiliza para acotar la salida de la neurona y generalmente viene dada por la interpretación que queramos darle a dichas salidas. Algunas de las más utilizadas son la función sigmoidea (para obtener valores en el intervalo $[0,1]$) y la tangente hiperbólica (para obtener valores en el intervalo $[-1,1]$) (Kiwix, 2011).

2.3.2. Diseño y programación de una RNA

Con un paradigma convencional de programación en ingeniería del software, el objetivo del programador es modelar matemáticamente (con distintos grados de formalismo) el problema en cuestión y posteriormente formular una solución (programa) mediante un algoritmo codificado que tenga una serie de propiedades que permitan resolver dicho problema. En contraposición, la aproximación basada en las RNA parte de un conjunto de datos de entrada suficientemente significativo y el objetivo es conseguir que la red aprenda automáticamente las propiedades deseadas. En este sentido, el diseño de la red tiene menos que ver con cuestiones como los flujos de datos y la detección de condiciones, y más que ver con cuestiones tales como la selección del modelo de red, la de las variables a incorporar y el pre-procesamiento de la información que formará el conjunto de entrenamiento. Asimismo, el proceso por el que los parámetros de la red se adecuan a la resolución de cada problema no se denomina genéricamente programación sino que se suele denominar entrenamiento (Kiwix, 2011).

2.3.3. Estructura

La mayoría de los científicos coinciden en que una RNA es muy diferente en términos de estructura de un cerebro animal. Al igual que el cerebro, una RNA se compone de un

conjunto masivamente paralelo de unidades de proceso muy simples y es en las conexiones entre estas unidades donde reside la inteligencia de la red. Sin embargo, en términos de escala, un cerebro es muchísimo mayor que cualquier RNA creada hasta la actualidad, y las neuronas artificiales también son más simples que su contrapartida animal (Kiwix, 2011).

Biológicamente, un cerebro aprende mediante la reorganización de las conexiones sinápticas entre las neuronas que lo componen. De la misma manera, las RNA tienen un gran número de procesadores virtuales interconectados que de forma simplificada simulan la funcionalidad de las neuronas biológicas. En esta simulación, la reorganización de las conexiones sinápticas biológicas se modela mediante un mecanismo de pesos, que son ajustados durante la fase de aprendizaje. En una RNA entrenada, el conjunto de los pesos determina el conocimiento de esa RNA y tiene la propiedad de resolver el problema para el que la RNA ha sido entrenada (Kiwix, 2011).

Por otra parte, en una RNA, además de los pesos y las conexiones, cada neurona tiene asociada una función matemática denominada función de transferencia (Kiwix, 2011).

Dicha función genera la señal de salida de la neurona a partir de las señales de entrada. La entrada de la función es la suma de todas las señales de entrada por el peso asociado a la conexión de entrada de la señal. Algunos ejemplos de entradas son la función escalón de Heaviside, la lineal o mixta, la sigmoidea y la función gaussiana, recordando que la función de transferencia es la relación entre la señal de salida y la entrada (Kiwix, 2011).

2.3.4. Ventajas

Las redes neuronales artificiales (RNA) tienen ventajas debido a que están basadas en la estructura del sistema nervioso, principalmente el cerebro (UNAM, 2007).

- **Aprendizaje:** Las RNA tienen la habilidad de aprender mediante una etapa que se llama etapa de aprendizaje. Esta consiste en proporcionar a la RNA datos como entrada a su vez que se le indica cuál es la salida esperada (UNAM, 2007).
- **Auto organización:** Una RNA crea su propia representación de la información en su interior, descargando al usuario de esto (UNAM, 2007).
- **Tolerancia a fallos:** Debido a que una RNA almacena la información de forma redundante, ésta puede seguir respondiendo de manera aceptable aun si se daña parcialmente (UNAM, 2007).
- **Flexibilidad:** Una RNA puede manejar cambios no importantes en la información de entrada, como señales con ruido u otros cambios en la entrada (UNAM, 2007).
- **Tiempo real:** La estructura de una RNA es paralela, por lo cual si esto es implementado con computadoras o en dispositivos electrónicos especiales, se pueden obtener respuestas en tiempo real (UNAM, 2007).

2.3.5. Tipologías de las redes neuronales

a) Topología 1

- Una primera clasificación de las redes neuronales que se suele hacer es en función del patrón de conexiones que presentan. Así se definen tres tipos básicos de redes (UNAM, 2007):
- Dos tipos de redes de propagación hacia delante o acíclicas en la que todas las señales van desde la capa de entrada hacia la salida sin existir ciclos, ni conexiones entre neuronas de la misma capa (UNAM, 2007).

- Monocapa: Sólo existe una capa de neuronas de entrada y otra de salida. Por ejemplo el perceptrón y Adaline (UNAM, 2007).
- Multicapa: Existe un conjunto de capas intermedias entre la capa de entrada y la de salida (capas ocultas). Por ejemplo, el perceptrón multicapa (UNAM, 2007).

Las redes recurrentes que presentan al menos un ciclo cerrado de activación neuronal. Por ejemplo la red de Elman, la red de Hopfiel y la máquina de Bolzman (UNAM, 2007).

b) Aprendizaje

Una segunda clasificación que se suele hacer es en función del tipo de aprendizaje de que es capaz (si necesita o no un conjunto de entrenamiento supervisado). Para cada tipo de aprendizaje se encuentran varios modelos propuestos (UNAM, 2007):

- Aprendizaje supervisado: necesitan un conjunto de datos de entrada previamente clasificado o cuya respuesta objetivo se conoce. Ejemplos de este tipo de redes son el perceptrón (simple y multicapa), la red Adaline y la memoria asociativa bidireccional (UNAM, 2007).
- Aprendizaje no supervisado o auto-organizado: no necesitan de tal conjunto previo. Ejemplos de estas redes son las memorias asociativas, redes de Hopfiel, máquina de Bolzman y las redes de Kohonen (UNAM, 2007).

- Redes híbridas: son un enfoque mixto en el que se utiliza una función de mejora para facilitar la convergencia. Un ejemplo de estas redes son las funciones de base radial (RBF) (UNAM, 2007).
- Aprendizaje reforzado: se sitúa a medio camino entre el supervisado y el auto-organizado (UNAM, 2007).

c) Tipo de entrada

Finalmente también se pueden clasificar las redes neuronales según sean capaces de procesar información de distinto tipo en:

- Redes analógicas: procesan datos de entrada con valores continuos y, habitualmente acotados. Ejemplos son la red de Hopfield, de Kohonen y redes de aprendizaje competitivo (UNAM, 2007).
- Redes discretas: procesan datos de entrada de naturaleza discreta; habitualmente valores lógicos booleanos. Ejemplos de este tipo de redes son las máquinas de Bolzman y Cauchy (UNAM, 2007).

2.4. TRATAMIENTO DE IMÁGENES

Todo lo concerniente a la imagen tratada mediante un ordenador ha cobrado una enorme importancia en los últimos tiempos. La proliferación de equipos y medios de captura han hecho habitual este fenómeno (Grimaldos, 2005).

En la sociedad actual también tienen una enorme importancia el lenguaje de la imagen. Los mensajes de contenido icónico y visual se utilizan frecuentemente para transmitir todo tipo de información, desde las señalizaciones de grandes espacios públicos hasta el formato ágil y certero de la publicidad (Grimaldos, 2005).

Sin embargo esta proximidad de mensajes gráficos que pretenden captar la atención, el manejo habitual de aparatos de captura o la obtención de imágenes a través de la red, no siempre traducen en un mejor conocimiento del fenómeno. En la mayoría de los casos suele ocurrir paradójicamente lo contrario (Grimaldos, 2005).

- **Imágenes digitales**

Una imagen digital, también llamada gráfico digital, es una representación bidimensional de una imagen utilizando bits (unos y ceros). Dependiendo de si la resolución de la imagen es estática o dinámica, puede tratarse de un gráfico rasterizado o de un gráfico vectorial. A menos que se indique lo contrario en general por imagen digital se entiende gráfico rasterizado (Grimaldos, 2005).

- **Obtención:**

Las imágenes digitales se pueden obtener de varias formas:

Por medio de dispositivos de conversión analógica-digital como los escáneres y las cámaras digitales (Grimaldos, 2005).

Directamente mediante programas informáticos, como por ejemplo realizando dibujos con el ratón (informática) o mediante un programa de renderización 2D.

Las imágenes digitales se pueden modificar mediante filtros, añadir o suprimir elementos, modificar su tamaño, etc. y almacenarse en un dispositivo de grabación de datos como por ejemplo un disco duro (Grimaldos, 2005).

- **Visualización**

Para visualizar una imagen digital se requiere un programa de visualización que convierta la información binaria contenida en un archivo en puntos de color perceptibles por el ojo humano, llamados píxeles y un dispositivo que permita su

renderización, típicamente una pantalla de ordenador. En Internet los formatos de imágenes digitales más extendidos son (Grimaldos, 2005):

JPG, GIF y PNG para imágenes rasterizadas.

SVG para gráficos vectoriales, formato estándar del W3C (World Wide Web Consortium) (Grimaldos, 2005).

- **Estructura**

La mayoría de formatos de imágenes digitales están compuestos por una cabecera que contiene atributos (dimensiones de la imagen, tipo de codificación, etc.), seguida de los datos de la imagen en sí misma. La estructura de los atributos y de los datos de la imagen es distinta en cada formato (Grimaldos, 2005).

Además, los formatos actuales añaden a menudo una zona de meta datos que sirve para precisar información adicional sobre la imagen, como por ejemplo: la fecha, la hora y el lugar donde se tomó la imagen las características físicas de la fotografía (fotosensibilidad ISO, velocidad de obturación, flash, etc.) (Grimaldos, 2005).

Estos meta datos se utilizan muy a menudo en el formato Exif (extensión del formato JPG), que es el formato más utilizado en las cámaras digitales (Grimaldos, 2005).

- **Color**

Al descomponer la luz se ve que está compuesto por siete colores, Básicamente los colores se descomponen en dos tipos: primarios o colores luz y complementarios o colores pigmento (Grimaldos, 2005).

- **Primarios O Colores Luz**

Son los tres colores principales de la composición de la luz, son también llamados los RGB (Red, Green, Blue) (Grimaldos, 2005).

Estos colores son sustractivos, es decir que al interponerse uno contra otro reducen o cambian su color es por eso que la suma de estos tres colores luz Rojo Verde y Azul da origen al Blanco (LUZ) (Grimaldos, 2005).

Ejemplo:

Rojo + Verde + Azul = Blanco (LUZ)

Rojo + Azul = Amarillo

Cuando el ojo humano percibe un color, los colores que realmente están actuando son los que le hacen falta a este para convertirse en luz (blanco). Así por ejemplo cuando se observa el color rojo, en realidad los colores luz que actúan son Verde y Azul; esto nos enseña que el efecto que producen los colores luz es opuesto a la realidad. Los monitores de computador son máquinas RGB y reproducen el color en 16.7 millones. Usa 3 canales que contienen 24 bit por pixel (8x3) (Grimaldos, 2005).

- **Colores Complementarios**

Estos son los llamados colores pigmento o de impresión, son los que se manejan usualmente, y se forman a partir de las combinaciones de los colores luz, estos son: azul, rojo y amarillo en su forma original en donde se llaman CYAN, MAGENTA y AMARILLO; también son llamados los colores puros (Grimaldos, 2005).

Del CYAN MAGENTA y YELLOW nace la sigla CMYK con la que se maneja la imagen de impresión; la letra K adicional representa el color negro que haría falta para completar una gama completa de colores (Grimaldos, 2005).

Los colores CMYK son usados para impresión y separación de colores. Usan 4 canales que contienen 32 bit por pixel (8 x 4) (Grimaldos, 2005).

- **Manejo del color por el computador**

El software gráfico manipula imágenes utilizando uno o varios canales los cuales representan información acerca de uno de los elementos del color en la imagen. El uso de canales hace posible manipular imágenes en sofisticadas formas. El usuario puede ajustar un solo elemento de color en una imagen, comparar el color en las imágenes separadas analizando y ajustando sus respectivos canales y usar canales para adicionar detalle a una imagen que fácilmente puede modificar y remover después (Herrera, 2010).

Para el manejo de imagen se hace importante conocer los diferentes tipos en los que el computador almacena y transporta imágenes (Herrera, 2010).

Un concepto muy importante en los mapas de bits es la profundidad de color: la cantidad de colores que puede tener una imagen por cada píxel. La denominación se suele hacer por medio de bits, pues se refiere a cuantos bits (unos o ceros) por píxel utiliza la imagen para representar los colores: 2 bits, 8 bits, 16 bits (Herrera, 2010).

- **Tipos de imagen**

- **Bitmap**

Está compuesto por un solo bit de color por píxel (blanco o negro) y requiere la mínima cantidad de memoria de todos los modos de imagen. Porque tiene pocas opciones de edición, se convierte a imagen GRAY SCALE y luego a BITMAP otra vez para exportarlo.

- **Gray scale**

Están constituidas por 8 bit de información por píxel y usan de 0 a 255 escalas de grises simulando gradación de color. Este es un modo de un solo canal para convertir en alta calidad imágenes blanco y negro.

- **Duotono**

El modo de Dúo tono se utiliza para incrementar el rango de grises en las imágenes Grayscale. Es usado para mono tonos - dúo tonos – trítonos o tetra tonos, pero aún no es policromía. Son esencialmente imágenes en escala de grises de un solo canal (8 bits por píxel).

- **Indexed**

Indexed color son imágenes que usan un solo canal (8 bits x píxel) y tienen una tabla de 256 colores. Estas imágenes son útiles a la hora de editar la tabla de color o cuando una paleta de colores limitada se va a exportar. Por ejemplo para aplicaciones multimedia.

- **Rgb**

Las imágenes RGB usan tres colores para reproducir 16.7 millones de colores en la pantalla de un computador. Para muchas imágenes y

scanners este es el modo para editar y pintar. Las imágenes RGB usan 3 canales de imagen contienen 24 bit x píxel.

- **Cmyk**

Las imágenes CMYK consisten en 4 colores usados para impresión y separación de colores. Estos son 4 canales de imagen; contienen 32 bit por píxel (8 x 4). Este modo permite trabajar directamente con imágenes CMYK que han sido escaneadas o importadas de sistemas superiores. Porque este modo usa la mayor cantidad de memoria de todos los tipos de imagen, es generalmente lenta para edición de imagen.

- **Hsl**

HSL usa tres canales; tono, saturación y luces. No es compatible con los modos RGB ni CMYK, no existe muestra de composición para estas imágenes. Las imágenes HSL son generalmente usadas para aplicaciones científicas.

- **Hsb**

Las imágenes HSB usan 3 canales: Tono, saturación y brillo. No es compatible con los modos RGB ni CMYK, no existe muestra de composición para estas imágenes. Las imágenes HSB al igual que las HSL son generalmente usadas para aplicaciones científicas.

- **Multichannel**

Adicionando un canal a la escala de grises GrayScale o borrando un canal desde un RGB, CMYK, HSL o HSB se crea una imagen MULTICHANNEL. Imágenes en modo MULTICHANNEL se pueden convertir a imágenes GrayScale.

- **Iluminación**

La iluminación existente en el entorno no siempre es la adecuada ya que se obtienen imágenes con bajo contraste, es decir, poco o demasiado iluminadas. Esto conlleva a problemas como sombras no deseadas, que afectan directamente la complejidad del algoritmo.

Para definir claramente el tipo de iluminación es necesario identificar las propiedades de los objetos que intervienen en la escena, pueden ocurrir tres situaciones cuando un haz luminoso incide sobre un material; que se refleje, se absorba o se transmita a través de él.

Podemos clasificar la iluminación en:

- Iluminación direccional: Esta técnica consiste en aplicar una iluminación orientada al objeto usando un haz altamente direccional. Es óptima en aplicaciones que buscan el reconocimiento de objetos ya que define claramente las distintas regiones de una imagen. Este tipo es de las más utilizadas debido a su uso fácil.
- Iluminación difusa: Con este tipo de iluminación se intenta que los haces luminosos incidan sobre el objeto desde todas las direcciones. Se utiliza cuando se van a tomar imágenes desde diversos puntos de vista, por lo que no se pueden favorecer a unas zonas más que otras.
- Iluminación a contra luz: Consiste en iluminar el objeto por detrás, de forma que la fuente luminosa, el objeto y la cámara estén alineados. Esta técnica genera imágenes prácticamente con sólo dos niveles de gris, es decir, imágenes binarias.

- Iluminación oblicua: Puede considerarse un caso particular de la iluminación direccional. El objetivo principal es la creación de sombras encargadas de aumentar el contraste de las partes tridimensionales. Este tipo de iluminación es utilizado para generar sombras sobre objetos cuyo contraste es pequeño respecto al fondo.

- **Segmentación de imágenes**

La segmentación de imágenes tiene su origen en numerosos estudios psicológicos que indican la preferencia de los humanos por agrupar regiones visuales en términos de proximidad, similitud y continuidad, para constituir un conjunto de unidades significativas. Estas propiedades no son fáciles de cumplir y sin embargo son fundamentales para una buena segmentación.

En general la segmentación es una de las tareas más significativas en el procesamiento de imágenes. Este paso del proceso determina el eventual éxito o fracaso del análisis de la imagen.

Los algoritmos de segmentación de imágenes tienen tres formas comunes: métodos basados en bordes, técnicas basadas en regiones y técnicas de umbral.

- **Segmentación por bordes**

Se centran en la detección de contornos. Delimitan el borde de un objeto y segmentan los píxeles dentro del contorno como pertenecientes a ese objeto. Su desventaja consiste en conectar contornos separados o incompletos, lo que los hace susceptibles a fallas.

- **Segmentación por regiones**

En esta aproximación todos los píxeles que correspondan a un objeto se agrupan juntos y son marcados para indicar que pertenecen a una región. Los píxeles son

asignados a regiones según algún criterio que los distingue del resto de la imagen. Un criterio muy estricto puede provocar fragmentación mientras uno poco estricto ocasiona uniones indeseadas.

- **Segmentación por umbral**

Esta técnica segmenta la imagen píxel por píxel, es decir, no toman en consideración el valor de los píxeles vecinos para el proceso. Si el valor de un píxel está dentro de un rango de valores especificado para un objeto el píxel es segmentado. Son efectivas cuando los objetos y el fondo de la imagen tienen rangos de valores diferentes y existe un contraste marcado entre ellos. Como la información de los píxeles vecinos es ignorada, las fronteras de regiones borrosas pueden ocasionar problemas.

La operación conocida como "umbral simple" consiste en utilizar cero para todos los píxeles cuyo nivel de gris se encuentra por debajo de un cierto valor (llamado el umbral) y el valor máximo para todos los píxeles con un valor mayor.

- **Filtrados espacial**

Los filtros espaciales tienen como objetivo modificar la contribución de determinados rangos de frecuencias a la formación de la imagen. El término espacial se refiere al hecho de que el filtro se aplica directamente a la imagen y no a una transformada de la misma, es decir, el nivel de gris de un píxel se obtiene directamente en función del valor de sus vecinos.

Los filtros espaciales pueden clasificarse basándose en su linealidad: filtros lineales y filtros no lineales. A su vez los filtros lineales pueden clasificarse según las frecuencias que dejen pasar: los filtros paso bajo atenúan o eliminan las componentes de alta frecuencia a la vez que dejan inalteradas las bajas frecuencias; los filtros paso alto atenúan o eliminan las componentes de baja

frecuencia con lo que agudizan las componentes de alta frecuencia; los filtros paso banda eliminan regiones elegidas de frecuencias intermedias.

La forma de operar de los filtros lineales es por medio de la utilización de máscaras que recorren toda la imagen centrando las operaciones sobre los píxeles que se encuadran en la región de la imagen original que coincide con la máscara y el resultado se obtiene mediante una computación (suma de convolución) entre los píxeles originales y los diferentes coeficientes de las máscaras.

Los filtros espaciales no lineales también operan sobre entornos. Sin embargo, su operación se basa directamente en los valores de los píxeles en el entorno en consideración. Unos ejemplos de filtros no lineales habituales son los filtros mínimo, máximo y de mediana que son conocidos como filtros de rango. El filtro de mediana tiene un efecto de difuminado de la imagen, y permite realizar una eliminación de ruido de forma eficaz, mientras que el filtro de máximo se emplea para buscar los puntos más brillantes de una imagen produciendo un efecto de erosión, y el filtro de mínimo se emplea con el objetivo contrario, buscar los puntos más oscuros de una imagen produciendo un efecto de dilatación.

Otra clasificación de los filtros espaciales puede hacerse basándose en su finalidad, y así tenemos los filtros de realce (Sharpening) para eliminar zonas borrosas o filtros de suavizado (Smoothing) para difuminar la imagen. También tenemos los filtros diferenciales que se componen de varios tipos de máscaras (Laplaciano, Prewitt, Sobel, etc.), y se utilizan para la detección de bordes. El proceso de detección de bordes se basa en realizar un incremento del contraste en las zonas donde hay una mayor diferencia entre las intensidades, y en una reducción de éste donde no tenemos variación de intensidad.

- **Dilatación y Erosión**

El filtrado de mínimo y de máximo es muy similar al filtrado de mediana, junto con éste forman los filtros de rango. En vez de sustituir el píxel del centro por el valor medio, se utiliza el valor mínimo o máximo de la ventana. Estos filtros son eficaces en eliminar el ruido impulsivo extremo. El filtro mínimo elimina los puntos blancos (valor 255) mientras que el filtro máximo elimina los puntos oscuros (valor 0).

Con el filtrado de mínimo, un píxel se representa con el píxel más oscuro de su vecindad, por lo que la intensidad total de la imagen de salida será reducida. Éste filtro tiende a "ensanchar" las líneas negras de la imagen, por esta razón también es conocido como filtro de dilatación.

Con el filtrado de máximo se aumenta la intensidad total de la imagen de salida. Este filtro también es conocido como filtro de erosión, debido a la propiedad que posee de "adelgazar" líneas. Si nuestra imagen posee líneas negras, al elegir el valor máximo de la vecindad de cada píxel, los valores más oscuros serán sustituidos por valores más altos con la consiguiente reducción de los píxeles cercanos al negro.

Otra aplicación posible del filtrado de mínimo y máximo es la eliminación de ruido. Estos filtros son eficaces en eliminar el ruido impulsivo extremo. El filtro de mínimo elimina los puntos blancos (valor 255) mientras que el filtro máximo elimina los puntos oscuros (valor 0). Aunque los filtros mínimos y máximos fallan en la eliminación del ruido impulsivo mezclado, pues los filtros mínimo y máximo tienden a realzar los puntos negativos y positivos, respectivamente. Una cascada de filtros máximos y mínimos puede quitar este ruido mezclado. Un filtro mínimo seguido de un filtro máximo se conoce como filtro de abertura.

2.5. CROMA

El croma o inserción croma (del inglés chroma key) es una técnica audiovisual utilizada ampliamente tanto en el cine y televisión como en fotografía, que consiste en la sustitución de un fondo por otro mediante un equipo especializado o un ordenador. Esto se hace debido a que es demasiado costoso e inviable el recorte del fondo o personajes para completar cuadro a cuadro mediante rotoscopia². (Serrano, 2012)

Para realizar estas operaciones con facilidad y buen resultado, se utiliza el Chroma key o “clave de color”. Básicamente consiste en un fondo de color sólido y uniforme y el objeto que se desea recortar o cambiar de fondo, dejando lo demás para tarea del ordenador. Para realizarla operación correctamente se debe escoger un fondo y después un objeto o persona para colocar en una situación posterior a ese hipotético fondo. (Serrano, 2012)



Fig. 3. Croma

Fuente: (Serrano, 2012)

² El rotoscopio es un dispositivo que permite a los animadores diseñar imágenes para películas de animación. Puede ser usado para animar siguiendo una referencia filmada en vivo. Pudiendo ser considerado un precursor de la técnica de captura de movimiento digital.

El color debe ser rojo, verde, o azul. Habitualmente se usa verde o azul y no rojo, ya que es el componente de color más importante de la piel humana. Aunque el color más habitual del fondo o ciclorama era el azul, el verde se ha impuesto al ser la ropa azul mucho más común que la verde. (Serrano, 2012)

2.6. CONVOLUCIÓN

En matemáticas y en particular en Análisis funcional, una convolución es un operador matemático que transforma dos funciones f y g en una tercera función que en cierto sentido representa la magnitud en la que se superponen, f y una versión trasladada e invertida de g . Una convolución es un tipo de cierto promedio en movimiento, como se puede observar si una de las funciones la tomamos como la función característica de un intervalo (Academic, 2008).

La convolución de dos señales se define como un operador matemático ($*$) entre dos funciones $f(x)$ y $g(x)$, escrita como $h(x) = f(x) * g(x)$, donde $h(x)$ representa cierta señal generada. El valor que toma $h(x)$ en cada punto es la integral del producto de $f(x)$ con la función de filtro $g(x)$ del punto original (Academic, 2008).

2.6.1. Convolución Gaussiana

La convolución Gaussiana se caracteriza por su operador el cual tiene forma de campana de Gauss. En el procesamiento de imágenes, esta convolución es usada para filtrar el ruido presente en una imagen a través del suavizado del tono de los píxeles con respecto a sus vecinos. La intensidad del suavizado se determinará a través del tamaño de la máscara del filtro utilizada (función de filtro). La distribución Gaussiana bidimensional tiene la forma (Academic, 2008).

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Fig. 4. Formula de la convolución gaussiana

Fuente: (Academic, 2008)

2.7. MATLAB

La implementación de algoritmos en visión por computador resulta muy costosa en tiempo ya que se requiere de la manipulación de punteros, gestión de memoria, etc. Hacerlo en cualquier lenguaje (que por sus características compartidas de alto y bajo nivel lo hacen el más apropiado para la implementación de algoritmos de visión computacional) supondría la inversión de tiempo y sin la seguridad de que lo que se desea implementar funcionará. Además utilizar cualquier lenguaje para el periodo de prueba exige un tiempo normal de corrección de errores debidos al proceso de implementación del algoritmo, es decir errores de programación. Todos estos problemas pueden ser resueltos si la implementación de prueba es realizada en MATLAB utilizando su toolbox³ de procesamiento de imágenes con ello el tiempo de implementación se convierte en el mínimo con la confianza de utilizar algoritmos científicamente probados y robustos (Javier García de Jalón, 2005).

- **Procesamiento de imágenes en Matlab**

El toolbox de procesamiento de imágenes contiene un conjunto de funciones de los algoritmos más conocidos para trabajar con imágenes binarias, transformaciones geométricas, morfología y manipulación de color que junto con las funciones ya integradas en matlab permite realizar análisis y transformaciones

³ Cajas de herramientas, módulo de matlab especializado en una rama, existen toolboxes especializados para redes neuronales, captura de video y muchos otros

de imágenes en el dominio de la frecuencia (transformada de Fourier y Wavelets) (Javier García de Jalón, 2005).

En matlab una imagen a escala de grises es representada por medio de una matriz bidimensional de $m \times n$ elementos en donde n representa el número de píxeles de ancho y m el número de píxeles de largo. El elemento v_{11} corresponde al elemento de la esquina superior izquierda (ver fig. 3), donde cada elemento de la matriz de la imagen tiene un valor de 0 (negro) a 255 (blanco) (Javier García de Jalón, 2005).

Las imágenes de color RGB es representada por una matriz tridimensional $m \times n \times p$, donde m y n tienen el mismo valor que para el caso de las imágenes de escala de grises mientras presenta el plano, que para RGB que puede ser 1 para el rojo, 2 para el verde y 3 para el azul. Las figuras 5, 6 y 7 muestran detalles de estos conceptos (Javier García de Jalón, 2005).

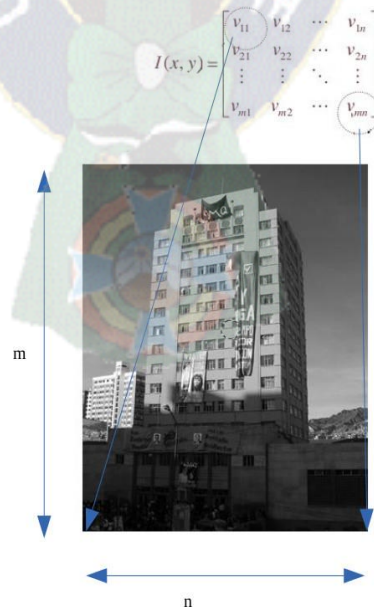


Fig. 5. Representación de una imagen a escala de grises en Matlab.

Fuente: Elaboración propia

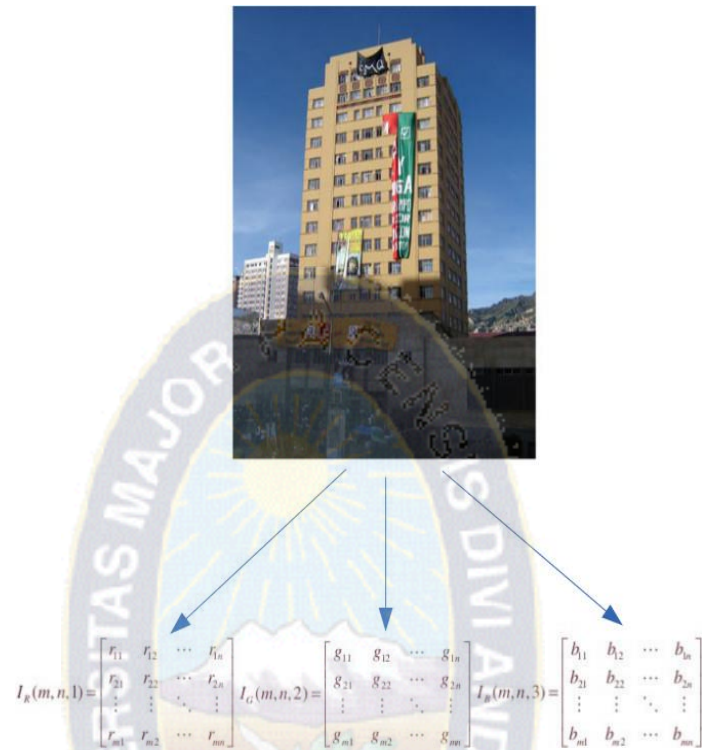


Fig. 6. Representación de una imagen a escala RGB en Matlab.

Fuente: Elaboración propia

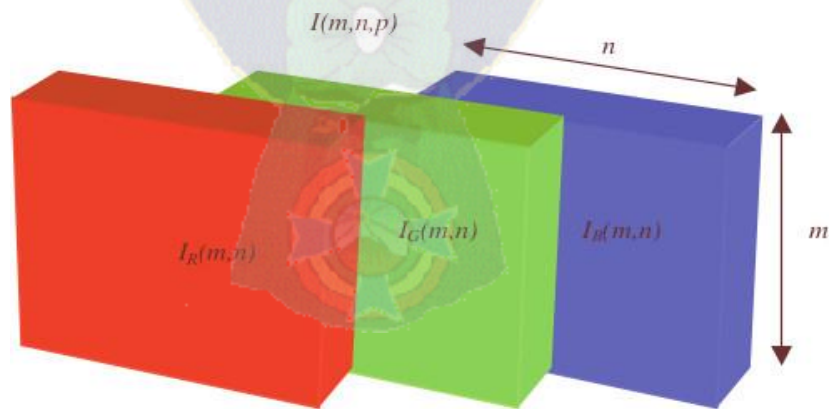


Fig. 7. Representación de los arreglos RGB en Matlab

Fuente: Elaboración propia

- **Lectura de imágenes a través de archivo**

Para realizar la lectura de una imagen en MatLab se utiliza la función *imread*, cuya sintaxis es *imread* ('nombre_del_archivo.extencion ') Donde nombre del archivo es una cadena de caracteres conteniendo el nombre completo de la imagen con su respectiva extensión (Javier García de Jalón, 2005).

Formato	Extensión
TIFF	.tiff
JPEG	.jpg
GIF	.gif
BMP	.bmp
PNG	.png
XWD	.xwd

Tabla. 1. Formatos y extensiones soportadas por Matlab.

Fuente: Elaboración propia

Para introducir una imagen guardada en un archivo con alguno de los formatos especificados en la tabla anterior solo tiene que usarse la función *imread* y asignar su resultado a una variable que representará a la imagen (de acuerdo a la estructura, para representar escala de grises y para RGB). De tal forma que si se quisiera introducir la imagen contenida en el archivo *data.jpg* a una variable para su procesamiento en matlab, entonces se tendría que escribir en línea de comandos: `>>image=imread('data.jpg')` (Javier García de Jalón, 2005).

Este comando pasara la imagen en una hipermatriz⁴ que contendrá los pixeles de la escala RGB (Javier García de Jalón, 2005).

⁴ Matriz de más de dos dimensiones

- **Escritura de imágenes**

Para poder grabar el contenido de una imagen en un archivo se utiliza la función `imwrite` (variable, 'nombre del archivo'), en donde variable representa la variable que contiene a la imagen y nombre del archivo el nombre del archivo con su respectiva extensión de acuerdo a la tabla (Javier García de Jalón, 2005).

Suponiendo que la variable `image2` contiene la imagen que nos interesa grabar en el archivo `dato2.jpg` tendríamos que escribir:

```
>> imwrite (image2, 'data2.jpg') ;
```

Después que realizamos un procesamiento con la imagen, es necesario desplegar el resultado obtenido, la función `imshow` (variable) permite desplegar la imagen en una ventana en el ambiente de trabajo de matlab. Si la variable a desplegar por ejemplo, es `face` al escribir en la línea de comandos (Javier García de Jalón, 2005):

```
>> imshow (face);
```

2.8. RED BACKPROPAGATION UTILIZANDO MATLAB

En matlab se pueden resolver problemas específicos sobre entrenamiento de redes neuronales esto es posible utilizando funciones del toolbox de matlab. Por lo general se utilizan cuatro pasos principales en el proceso de entrenamiento de una red neuronal (Basogain, 2010).

- Reunir los datos para el entrenamiento.
- Crear el objeto red.
- Entrenar la red.
- Simular la red

- **Reunir los datos para el entrenamiento.**

Se reúnen los datos para el entrenamiento dependiendo de cómo y para qué va a ser utilizada la red. Estos datos pueden ser presentados como matrices, vectores, números simple, aleatorios (Basogain, 2010).

- **Crear el objeto red.**

Para explicar la creación del objeto red es necesario conocer las diferentes funciones de transferencia y de entrenamiento existentes ya que éstas se utilizan en la creación del objeto red (Reyes, 2006).

- **Funciones de transferencia**

Las redes multicapas generalmente utilizan la función de transferencia log-sigmoidal (logsig) (Reyes, 2006).

La función log-sigmoidal (logsig) genera salidas entre 0 y 1 como las entradas de las neuronas van de negativo a positivo infinito (Basogain, 2010).

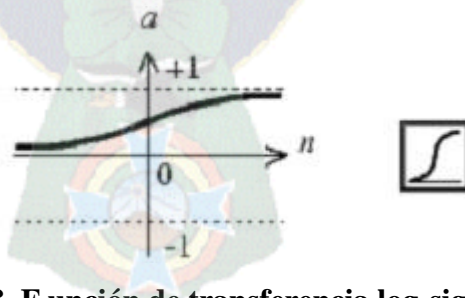


Fig. 8. Función de transferencia log-sigmoidal

Fuente: (Basogain, 2010)

Alternativamente las redes pueden usar también la función de transferencia tan-sigmoidal (tansig) (Basogain, 2010).

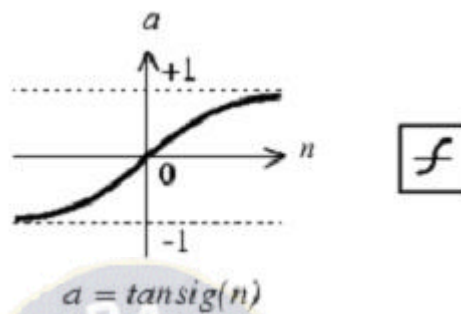


Fig. 9. Función de transferencia tan-sigmoidal

Fuente: (Basogain, 2010)

Ocasionalmente, la función de transferencia lineal (purelin) la cual es utilizada en la red Backpropagation (Basogain, 2010).

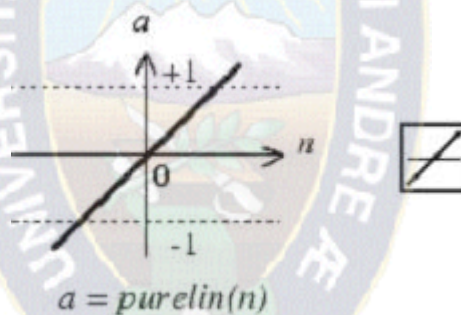


Fig. 10. Función de transferencia lineal

Fuente: (Basogain, 2010)

Si la última capa de la red tiene una neurona sigmoideal, luego la salida de la red es limitada a un rango pequeño. Si la salida lineal de la neurona es usada en la salida de la red puede tomar cualquier valor (Basogain, 2010).

En Backpropagation es importante tener habilitado para calcular la derivada de cualquier función de transferencia usada. Cualquiera de las funciones de transferencia sea ésta logsig, tansig o purelin puede ser llamada para calcular su propia derivada (Basogain, 2010).

Para calcular la derivada de la función de transferencia se llama a la función de transferencia con el string 'dn' (Basogain, 2010).

$$Dn = \text{tansig}('dn', n, a)$$

Las tres funciones de transferencia nombradas anteriormente son las funciones más comunes utilizadas para Backpropagation (Basogain, 2010).

- **Algoritmos de entrenamiento**

Es necesario también conocer con anterioridad los algoritmos de entrenamiento existentes ya que esto también se utiliza en la creación de la red neuronal (Basogain, 2010).

Existen diferentes algoritmos de entrenamiento para las redes feedforward. Estos algoritmos utilizan la gradiente en la función de activación para determinar el ajuste de los pesos (Basogain, 2010).

Existe dos caminos diferentes en los cuales el algoritmo de gradiente descendente puede ser implementada: modo incremental y el modo batch. En el modo incremental, la gradiente es calculada y los pesos son actualizados antes de que cada entrada sea aplicada a la red. En el modo batch, todas las entradas son aplicadas a la red antes de que los pesos sean actualizados. Los diferentes algoritmos existentes son (Basogain, 2010):

- Batch training (train)
- Batch Gradient Descent (traingd)
- Batch Gradient Descent with momentum (traingdm)
- Conjugate gradient (traingcf, traingcp, traingcb, traingscg)
- Quasi-Newton (trainbfg, trainoss)
- Levenberg- Marquardt (trainlm)

El algoritmo de entrenamiento básico de backpropagation es en el cual los pesos se mueven en dirección negativa a la gradiente (Basogain, 2010).

- **Creación de la red neuronal (newff)**

El primer paso para la creación de una red feedforward es crear el objeto red (net). La función de matlab newff crea una red feedforward. Esta requiere de cuatro entradas y retorna el objeto red. La primera entrada es R por 2 de la matriz o los valores máximos y mínimos para cada uno de los R elementos de la entrada de los vectores. La segunda entrada es una matriz que contiene los tamaños de cada capa tanto de la de entrada como la de salida. La tercera entrada es un conjunto de células que contienen los nombres de las funciones de transferencia a ser utilizadas en cada capa. Y la última capa contiene el nombre de la función o algoritmo de entrenamiento a ser utilizada en el entrenamiento de la red (Reyes, 2006).

```
Net = newff(P, [tam_capas], {func_transf}, 'func_entren');
```

Por ejemplo, el siguiente comando crea una red de dos capas. Hay una entrada del vector con dos elementos. Los valores para el primer elemento del vector de entrada oscila entre -1 y 2, y los valores del segundo elemento del vector de entrada oscila entre 0 y 5. Hay tres neuronas en la primera capa y una neurona en la segunda capa (de salida). La función de transferencia en la primera capa es tan-sig, y en la capa de salida la función de transferencia es lineal. La función de entrenamiento es traingd (Reyes, 2006).

```
Net = newff([-1 2; 0 5], [3,1], {'tansig', 'purelin'}, 'traingd');
```

Con este comando el objeto red ya fue creado.

- **Entrenar la red**

El entrenamiento de la red neuronal se la realiza con la función (train) la cual requiere de tres datos, el primero es el objeto red del cual se va a realizar el entrenamiento, el segundo es el vector de entrada que tendrá la red neuronal y el tercero es el vector de salida que se desea obtener del entrenamiento de la red neuronal (Reyes, 2006).

```
[net,tr]=train(net,P,T) ;
```

Con este código se realiza el entrenamiento de la red neuronal.

- **Simular la red**

La simulación de la red neuronal se la realiza con la función (sim) ésta función requiere los siguientes datos, la entrada a la red neuronal y el objeto red del cual va a realizar a simulación y esto retorna la salida esperada (Reyes, 2006).

Es necesario crear una variable en donde se almacena la respuesta de la simulación y ésta variable se la almacena en el workspace de matlab (Reyes, 2006).

```
Inter=sim(net,imagen);  
assignin('base','inter',inter);
```

2.9. TOOLBOX DE AUDIO EN MATLAB

En Matlab se tiene el toolboxes de sonido que contienen funciones de lectura de audio, (*wavread*), y de reproducción (*wavplay* o *sound*).

Wavread es una función de Matlab, del cual su objetivo es leer archivos de sonido WAVE (“wav”), su manejo es el siguiente.

```
Y=wavread(FILE)
```


Lee un archivo de onda especificado por el archivo cadena, la devolución de los datos muestreados en Y, la extensión de FILE debe tener la extensión de wav.

El rango de valores en Y depende del formato de datos especificado.

La reproducción de sonido *sound* es también por el toolbox multimedia de Matlab:
Sound(Y,FS) envía la señal en el vector Y (con la frecuencia de nuestro FS) a los altavoces en las plataformas que soporta sonido. Los valores fuera ese rango se recortan, los sonidos estereofónicos se juegan en plataformas que el apoyo que, cuando Y es una matriz $N \times 2$.





ANÁLISIS Y DESARROLLO

CAPITULO III

ANÁLISIS Y DESARROLLO

3.1. INTRODUCCIÓN

El capítulo al que damos inicio describe paso a paso los lineamientos y desarrollo del prototipo de “Redes Neuronales para la Interpretación del Lenguaje de Señas”, aplicando el método sistémico de Mario Bunge (2004).

3.2. ANÁLISIS PARA EL DESARROLLO DEL PROTOTIPO

En el análisis de los temas abordados se identificaron los componentes del prototipo.

Los componentes que identificamos son el lenguaje de señas, los lenguajes de programación existentes, proceso de imágenes y la red neuronal.

- Lenguaje de señas: al conocer la existencia de innumerables variaciones dentro del lenguaje de señas tomamos al alfabeto dactilológico universal, (Fig.2 alfabeto dactilológico), para trabajar en el prototipo ya que todos los lenguajes de señas, variantes, parten de éste.
- Lenguaje de programación: el lenguaje de programación que seleccionamos para el desarrollo es Matlab, por su capacidad de integración y la facilidad en el tratamiento de imágenes.
- Redes neuronales: Entre los distintos tipos de redes neuronales que pueden implementarse con Matlab se decidió utilizar la red neuronal perceptron multicapa de retro-propagación y el algoritmo de entrenamiento backpropagation (feed-forward backpropagation network: FFNN)

Para una mejor descripción del prototipo, el proceso se desarrolló en etapas que abarcan desde la manipulación de las imágenes hasta la respuesta del sistema, estas etapas son las siguientes:

- **ETAPA 1: generación y entrenamiento la red neuronal**
- **ETAPA 2: captura y pre-procesamiento de imagen**
- **ETAPA 3: procesamiento de la imagen**
- **ETAPA 4: reconocimiento de la seña**
- **ETAPA 5: reproducción de sonido**

3.3. ETAPA 1: generación y entrenamiento de Redes Neuronales

3.3.1. Arquitectura y generación de la red Neuronal

El primer paso para la creación del prototipo es el diseño de la red neuronal que se utiliza.

Previo a la creación de la red neuronal se cargaron los datos de las imágenes de entrenamiento, tomados en un ambiente controlado con el fondo de color blanco, para luego extraer las características importantes y las relevantes de cada una de ellas (Fig.11 Grafica de intensidad de las zonas de la imagen), con las que se crearon vectores de patrones.

Con los vectores anteriores se obtuvo las variables de entrada y las salidas esperadas para la creación de la red neuronal.

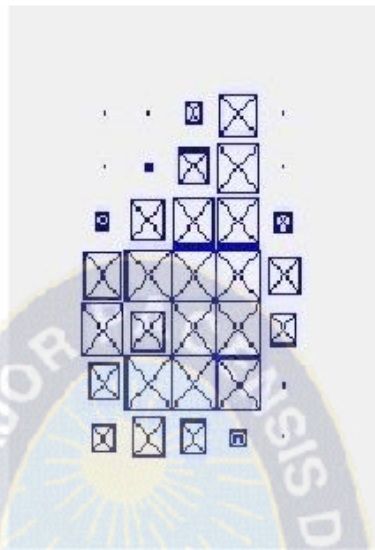


Fig. 11. Gráfica de intensidad de zonas de la imagen

Fuente: Elaboración propia

3.3.1.1. Red Neuronal

- Los primeros datos de ingreso, experimentados en la red neuronal, fueron de 7 por 4 haciendo un total de 28 elementos de entrada y 7 de salidas esperadas.

Se generó una sola capa oculta con 10 neuronas, la figura 12 ilustra la convergencia que logró el entrenamiento de la red neuronal con respecto a la respuesta de errores esperados y la cantidad de iteración realizadas. El nivel de confianza logrado para las señales es detallada en la tabla 2.

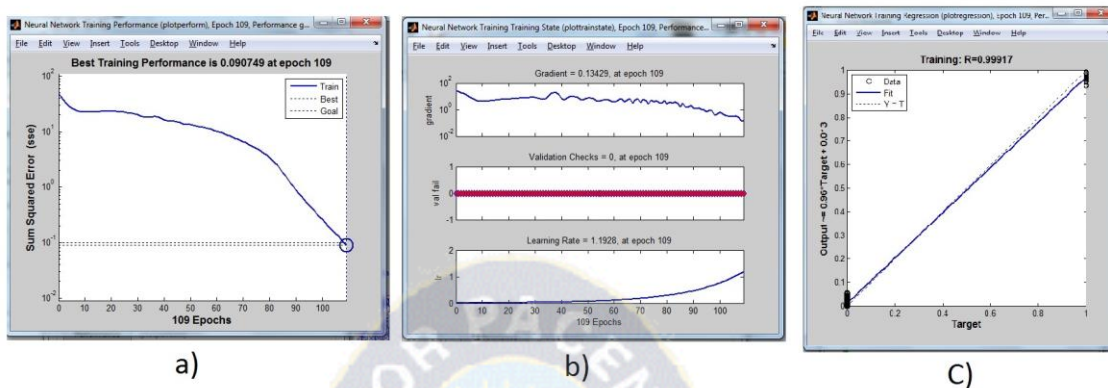


Fig. 12. Grafica experimento 1

Fuente: Elaboración propia

Seña	Nivel de confianza	Respuesta
A	97.6228 %	IDENTIFICADA
B	97.9917 %	IDENTIFICADA
C	97.6865 %	IDENTIFICADA
D	98.1431 %	IDENTIFICADA
E	98.6445 %	IDENTIFICADA
F	97.1282 %	IDENTIFICADA
G	96.8543 %	IDENTIFICADA

Tabla. 2. Respuesta experimento 1

Fuente: Elaboración propia

- En el segundo caso el número de señas que se busca interpretar es de 24, este número corresponde a las señas sin movimiento, del alfabeto dactilológico y es el número de salidas que debe tener la red neuronal del prototipo. Los datos de entrada fueron de 24 por 13, lo que nos dio un total de 312 elementos de entrada para la red neuronal. En la figura 13 se detalla la convergencia que logró el entrenamiento de la red neuronal con respecto a la respuesta de errores esperados

y la cantidad de iteración realizadas. El nivel de confianza logrado es detallada en la tabla 3.

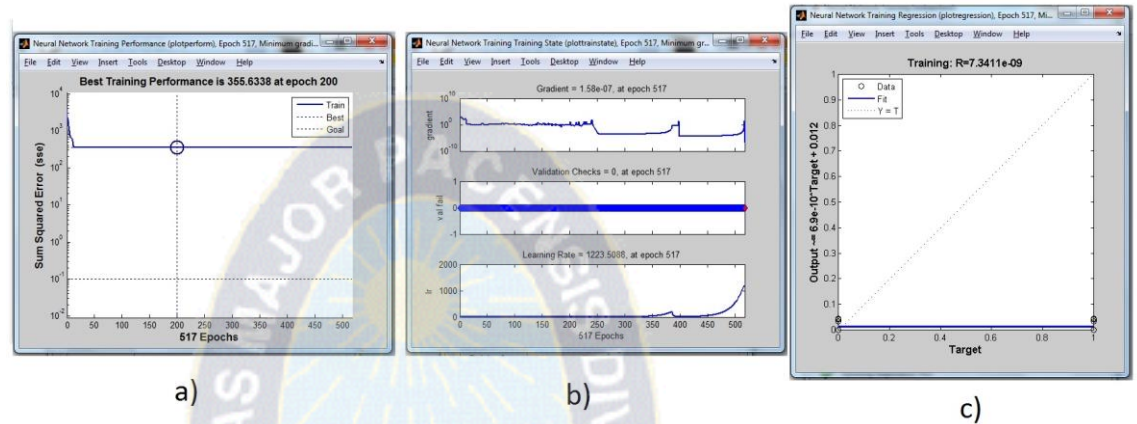


Fig. 13. Grafica experimento 2

Fuente: Elaboración propia

Seña	Nivel de confianza	Respuesta
A	5.0313 %	NO IDENTIFICADA
B	5.0313 %	NO IDENTIFICADA
C	5.0313 %	NO IDENTIFICADA
D	5.0313 %	NO IDENTIFICADA
E	5.0313 %	NO IDENTIFICADA
F	5.0313 %	NO IDENTIFICADA
G	5.0313 %	NO IDENTIFICADA
H	5.0313 %	NO IDENTIFICADA
I	5.0313 %	NO IDENTIFICADA
K	5.0313 %	NO IDENTIFICADA
L	5.0313 %	NO IDENTIFICADA
M	5.0313 %	NO IDENTIFICADA
N	5.0313 %	NO IDENTIFICADA

O	5.0313 %	NO IDENTIFICADA
P	5.0313 %	NO IDENTIFICADA
Q	5.0313 %	NO IDENTIFICADA
R	5.0313 %	NO IDENTIFICADA
S	5.0313 %	NO IDENTIFICADA
T	5.0313 %	NO IDENTIFICADA
U	5.0313 %	NO IDENTIFICADA
V	5.0313 %	NO IDENTIFICADA
W	5.0313 %	NO IDENTIFICADA
X	5.0313 %	NO IDENTIFICADA
Y	5.0313 %	NO IDENTIFICADA

Tabla. 3. Respuesta experimento 2

Fuente: Elaboración propia

- Al realizar la experimentación para el desarrollo de las redes neuronales generamos tres redes neuronales donde se dividieron las salidas esperadas en un número de 8. Los datos de entrada son de 8 por 4 en un total de 32 elementos para cada red neuronal. Las figuras 14, 15 y 16, dan detalle de las respuestas del entrenamiento de las redes neuronales y la tabla 4, resume los resultados obtenidos de las pruebas de entrenamiento aplicadas a las 3 redes neuronales.

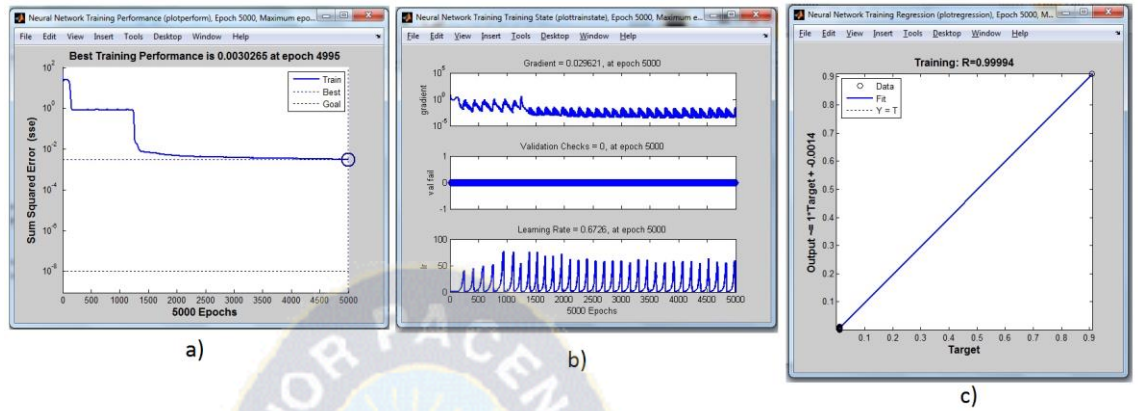


Fig. 14. Primera red neuronal, experimento 3

Fuente: Elaboración propia

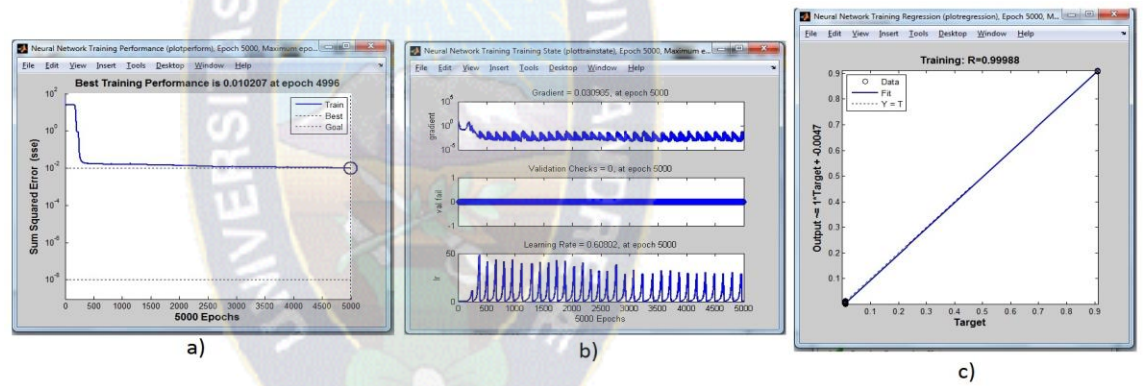


Fig. 15. Segunda red neuronal, experimento 3

Fuente: Elaboración propia

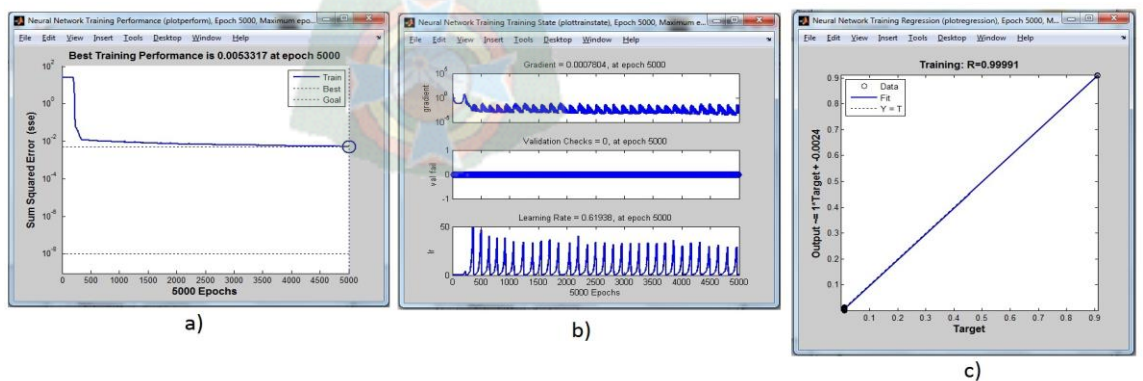


Fig. 16. Tercera red neuronal, experimento 3

Fuente: Elaboración propia

Seña	Nivel de confianza	Respuesta
A	91.0027 %	IDENTIFICADA
B	91.0207 %	IDENTIFICADA
C	90.9783 %	IDENTIFICADA
D	90.9303 %	IDENTIFICADA
E	90.99 %	IDENTIFICADA
F	91.0032 %	IDENTIFICADA
G	91.0017 %	IDENTIFICADA
H	90.8479 %	IDENTIFICADA
I	90.9205 %	IDENTIFICADA
K	91.0052 %	IDENTIFICADA
L	91.0225 %	IDENTIFICADA
M	90.9756 %	IDENTIFICADA
N	90.9337 %	IDENTIFICADA
O	90.9958 %	IDENTIFICADA
P	91.0107 %	IDENTIFICADA
Q	91.0003 %	IDENTIFICADA
R	91.0052 %	IDENTIFICADA
S	90.7935 %	IDENTIFICADA
T	91.0043 %	IDENTIFICADA
U	90.9971 %	IDENTIFICADA
V	91.011 %	IDENTIFICADA
W	91.0134 %	IDENTIFICADA
X	90.9972 %	IDENTIFICADA
Y	91.0031 %	IDENTIFICADA

Tabla. 4. Respuesta experimento 3

Fuente: Elaboración propia

Para cada experimento existe una variación del número de capas ocultas y neuronas, dependiendo de la cantidad de datos de entrada de las redes neuronales.

- El experimento 1 se realizó con 10 neuronas en una única capa oculta, y las funciones de activación, tangente sigmoideal “tansig” y logarítmica sigmoideal “logsig”, la primera designada para la capa oculta y la segunda para la capa de salida.
- Para el experimento 2 se implementaron dos capas ocultas con 19 y 8 neuronas respectivamente, las funciones de activación “tansig” para ambas capas ocultas y “logsig” para la capa de salida.
- El experimento 3 utiliza tres redes neuronales con una estructura idéntica, una única capa oculta conteniendo 15 neuronas y al igual que en el experimento 1 las funciones “tansig” para la capa oculta y logsig para la de salida.

El criterio de la raíz, (ver figura 17), fue utilizada para establecer las capas y las neuronas.

$$\begin{aligned}
 \mathbf{h}_i &= \sqrt{n + 1} & n &= \text{Número de elementos de entrada de la red neuronal} \\
 \mathbf{h}_1 &= \sqrt{n + 1} & h &= \text{Numero de neuronas de la capa oculta} \\
 \mathbf{h}_2 &= \sqrt{h1 + 1} & i &= \text{Numero de caps ocultas}
 \end{aligned}$$

Fig. 17. Criterio de la raíz

Fuente: Elaboración propia

La línea de comando:

```
newff(minmax(seniales),[S1,S3,S2],{'tansig','logsig','logsig'},'traingdx')
```

Genera la red neuronal para el prototipo como se explica en el capítulo 2, página 42.

El vector de características importantes es la variable *seniales* y el vector de salidas esperadas es *S2*.

3.3.2. Entrenamiento de Red Neuronal

Existen dos tipos de entrenamiento, el incremental y por lotes. En nuestro caso utilizo el entrenamiento por lotes. La forma más elemental de realizar este entrenamiento es con tres parámetros y es la forma como se realizó el entrenamiento de la red neuronal, los parámetros que mandamos son: nombre de la red neuronal, el vector de entradas y el vector de salidas deseadas, ingresándolos en ese orden.

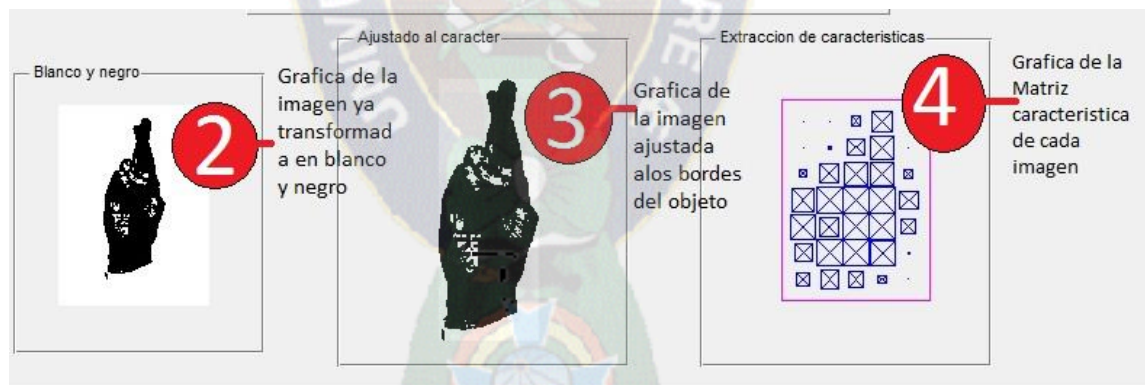


Fig.18. Objeto 2, 3 y 4 del gui de Entrenamiento

Fuente: Elaboración propia

La función *train*, parte del toolbox de redes neuronales de Matlab, es la que realiza el entrenamiento de la red neuronal como se describe en el capítulo 2, página 43.

3.3.1. Simulación

Para verificar el funcionamiento de la Red Neuronal, simulamos el comportamiento con la función "sim", esta función fue descrita en el capítulo 2.



Fig. 19. Front-end de entrenamiento red neuronal

Fuente: Elaboración propia

Todos los puntos descritos en esta etapa, son reunidos en el módulo *entrenamiento*. Este módulo también consta de un front-end de entrenamiento del prototipo que contiene cuatro espacios donde se exponen las imágenes del entrenamiento de datos y la simulación. En la figura 19 apreciamos la imagen de la ventana generada para el entrenador de la red neuronal.

3.4. ETAPA 2: captura y pre-procesamiento

Las técnicas de pre-procesado pretenden mejorar o realzar las propiedades de la imagen para facilitar las operaciones en las siguientes etapas.

Estas técnicas de pre-procesado se pueden dividir, en general, en varias facetas, para el prototipo se utilizó las siguientes:

- Realce o aumento del contraste (enhancement)
- Suavizado o eliminación del ruido (denoising)

La etapa de pre-procesamiento del sistema consiste en la manipulación de las imágenes que se desean traducir para obtener una representación adecuada de la señal para la etapa de procesamiento. La finalidad de esta etapa es eliminar toda aquella información innecesaria para el proceso de traducción y brindarle a la etapa de procesamiento el conjunto de datos de interés para la clasificación de cada una de las imágenes.

Esta etapa inicia con la captura de la imagen cada una imagen para luego realizar el procesamiento de las imágenes, este procesamiento es realizado mediante los siguientes pasos: extracción de áreas de interés, eliminación de ruido y Recorte

3.4.1. Extracción de áreas de interés

La extracción de áreas de interés, reconoce el área conformada por la imagen, esta es la encargada de extraer todas aquellas áreas de la imagen donde podrá estar presente la mano con el guante (Fig.20). Para extraer las áreas de interés se emplea el algoritmo basado en la técnica ya mencionada, Croma, con la que buscamos un área que contenga el color verde en la imagen obtenida.



Fig. 20. Captura de imagen

Fuente: Elaboración propia

Este algoritmo no toma en cuenta los pixeles que tengan colores distintos del verde.

Matlab toma las imágenes en varios formatos como explicamos en la sección, 2.4. Tratamiento de Imágenes. Para el prototipo utilizamos el formato RGB y esto facilita al algoritmo la búsqueda del color verde por medio de los canales imagen_R, imagen_G e imagen_B, que conforman la imagen capturada con la cámara, en la figura 21 presentamos las líneas de código que se encarga de realizar la consulta. El resultado de este proceso se observa en la figura 22.

```

If ((imagen(m,n,2)>imagen(m,n,1)) & (imagen(m,n,2)>
30) & ...
(imagen(m,n,2)>imagen(m,n,3)) & ...
(imagen(m,n,1)<((imagen(m,n,2)/2)*1.7)) & ...
(imagen(m,n,3)<((imagen(m,n,2)/2)*1.7)) )
platilla(m,n)=255;
else
plantilla(m,n)=0;
end

```

Fig. 21. Algoritmo croma

Fuente: Elaboración propia



Fig. 22. Captura de objeto

Fuente: Elaboración propia

3.4.2. Eliminación de ruido

En algunos casos, al realizar la plantilla se genera ruido⁵ en la imagen, su eliminación es la encargada de continuar con el Pre-Procesamiento. El resultado es una imagen binaria donde se obtiene la captura de la señal deseada.



Fig. 23. Eliminación de ruido

Fuente: Elaboración propia

Para lograr la eliminación de ruido aplicamos una convolución basada en filtros de Gauss a la imagen anterior, resultando una imagen estilizada con bordes difuminados. La finalidad de aplicar este filtro es hacer que los píxeles aislados de ruido presentes en la plantilla tomen valores considerablemente inferiores al blanco.

A partir de esta nueva imagen obtenida se procede a eliminar todas las áreas blancas excepto la de mayor tamaño. Y luego solo queda el área correspondiente a la señal de la mano. La figura 23 ilustra la eliminación de ruido.

⁵ El ruido en las imágenes es la variación aleatoria que no corresponde a la imagen, puede ser brillo o el color en las imágenes producido por la cámara web.

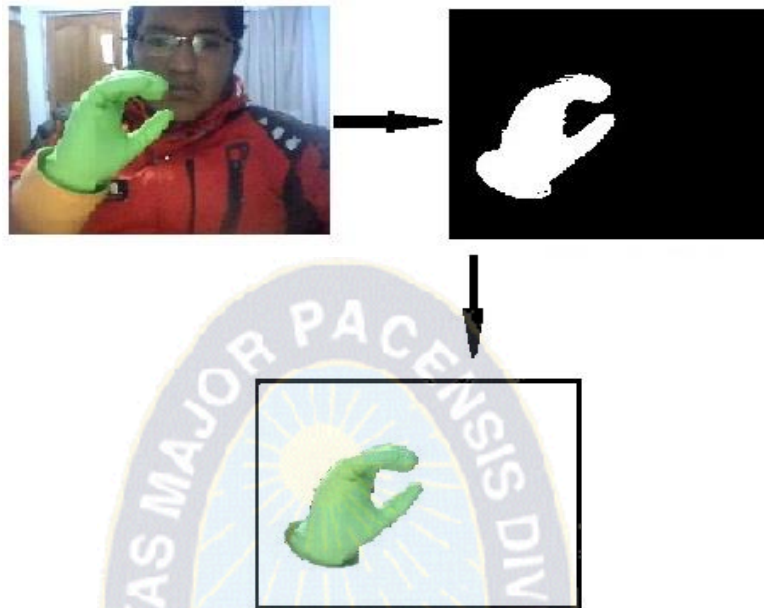


Fig. 24. Recorte de objeto
Fuente: Elaboración propia

3.4.3. Recorte

Este proceso tiene como objetivo recortar la plantilla obtenida en la etapa anterior y a partir de ésta extraer de la imagen original la seña que realiza la mano con su información interna, a diferencia de la plantilla que representa únicamente el contorno de la forma de la mano cómo podemos apreciar en la figura 24.

Para realizar el corte del área, de la mano, analizamos cada pixel de la plantilla anteriormente creada. El análisis consiste en la búsqueda de píxeles que contengan el blanco como valor, al encontrar esta coincidencia se toma las posiciones del pixel para copiar el valor del pixel de la imagen original y generar otra imagen que contiene solo la imagen de la seña. Luego la imagen obtenida es transformada a escala de grises para facilitar su manipulación, ya que al convertir la imagen RGB a escala de grises su representación pasa a ser de una matriz numérica de tres dimensiones ($N \times M \times 3$) a una

matriz de una dimensión ($N \times M$). A continuación las dos imágenes recortadas son re-escaladas a un tamaño preestablecido (100x100 px) que facilita el uso de la imagen en la etapa siguiente. El siguiente paso es generar una imagen con el guante en fondo negro. Esta imagen se logra haciendo una superposición de imágenes mediante la multiplicación entre la imagen binaria y la imagen en escala de grises obtenidas de en la etapa 1. Finalmente se obtiene la imagen del guante solo con las líneas del contorno para luego ajustar el tamaño de la imagen.

3.5. ETAPA 3: procesamiento

La etapa de procesamiento tiene como finalidad emplear las imágenes pre-procesadas para que éstos sean evaluados a través de un método de clasificación que determine si pertenece a alguna seña.

- **Extracción de características importantes o relevantes de la imagen.**

Tomamos la imagen obtenida al finalizar la etapa anterior y generamos una matriz de $n \times m$ que es representado como un vector.

Luego se encuentra el porcentaje de cada caja (100 pixeles por 100 pixeles) de la imagen que está vacío ya que negro tiene un valor de cero por lo tanto no sumara al área ocupada. Asignamos el porcentaje a una posición del vector que representa la letra.

La matriz obtenida contiene las características más relevantes de la imagen, puede ser apreciado en la figura 25.

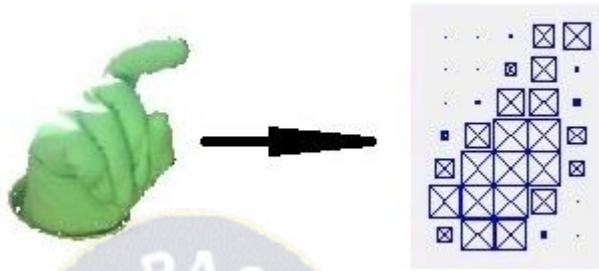


Fig. 25. Extracción de característica

Fuente: Elaboración propia

3.6. ETAPA 4: reconocimiento

Para realizar esta etapa, previamente se debe generar y entrenar a la red neuronal, esta etapa fue detallada en, 3.3. Etapa 1: Generación y entrenamiento de Redes Neuronales.

Procedemos a realizar el reconocimiento e interpretación de una imagen, para esto se utiliza la función `sim` de Matlab, la cual simula a la red, `sim` toma la matriz de características relevantes como entradas de la red (`charvec`) y el objeto red (`net`) y retorna una salida de la neurona. Para realizar el reconocimiento se crea una variable en donde se almacenará la respuesta de la misma (`result`), esta variable contiene la salida esperada. Luego con la función (`max`), hallamos el valor (`val`) y el número (`num`) de la posición dentro de (`result`).

La variable (`num`) almacena el número de posición de salida que representa el carácter en cuestión, aplicando la función `cellstr` de los toolbox de matrices, de Matlab generamos, el arreglo de `string's`, caracteres y con (`val`) encontramos el nivel de confianza. Nivel de confianza analizada para que pueda ser aceptable y con esta dar fiabilidad a la respuesta final del prototipo.

3.7. ETAPA 5: reproducción de sonido

Para realizar la traducción de la seña, realizada por la mano, al lenguaje oral accedemos a funciones de sonido, propias de Matlab, funciones descritas en el capítulo 2, página 43, las que se encargan de la lectura y reproducción de sonido.

Al encontrar la respuesta final del prototipo en la etapa anterior podemos realizar la selección de la respuesta en lenguaje oral, utilizamos la variable (num) para realizar esta selección. Los sonidos para la respuesta hablada fueron grabados en un ambiente controlado (figura 26).



Fig. 26. Reproducción de sonido

Fuente: Elaboración propia

Para realizar la lectura tomamos la variable (num) como referencia de letra a reproducir, luego realizamos la lectura y generamos dos variables (Y) y (Fs) . El vector (Y) contiene los bits de sonido y Fs contiene un valor entero que es la velocidad en la que se debe reproducir (Y).

Finalmente reproducimos el sonido seleccionado con la función sound dándole los variables (Y) y (Fs).

3.8. MÓDULO DE ENSEÑANZA DEL LENGUAJE DE SEÑAS

Este último módulo es el encargado de realizar la enseñanza del lenguaje de señas a los usuarios del prototipo.

Introducimos por teclado la letra que se desee aprender, esta letra es guardada en la variable (letra) para luego ser analizada. De la variable (letra) se calcula el tamaño, luego realizamos la búsqueda de la letra introducida en la base de imágenes, generada para el entrenamiento de la red neuronal del módulo de traducción del lenguaje de señas.

Al realizar el hallazgo de la letra se despliega la imagen y el sonido correspondiente a la letra detectada, en la figura 27 podemos apreciar la introducción de una letra y su correspondiente salida.

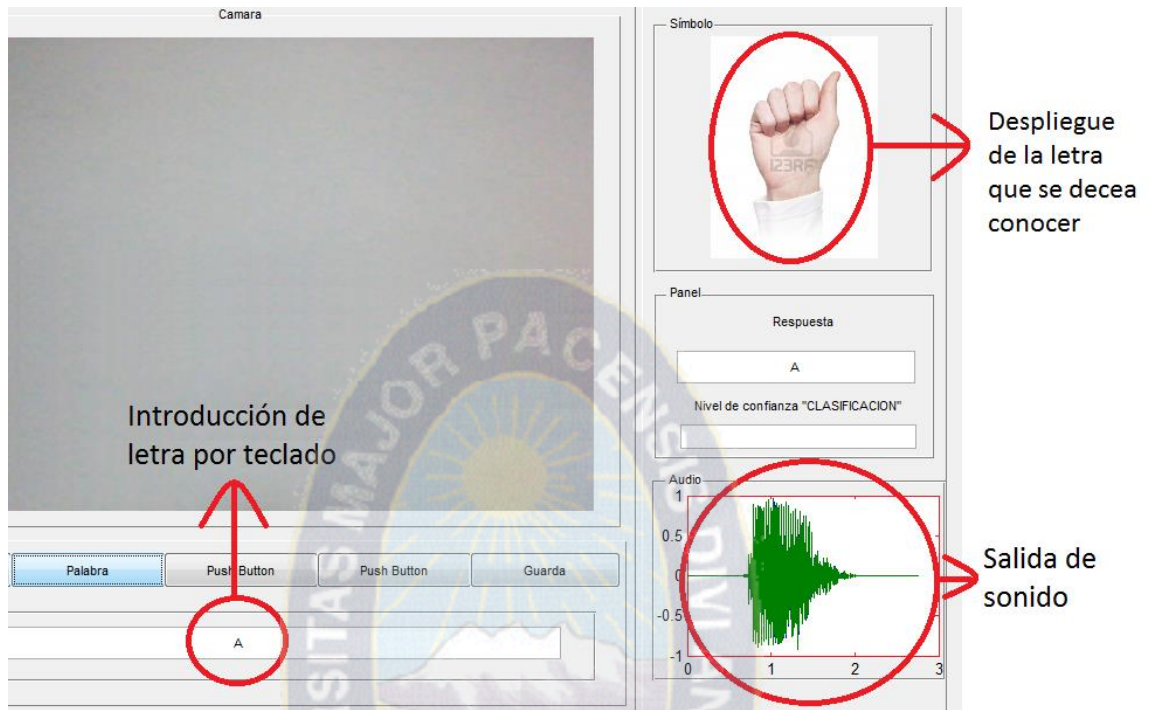


Fig. 27. Enseñanza del lenguaje de señas

Fuente: Elaboración propia



PRUEBAS Y RESULTADOS

CAPITULO IV

PRUEBAS Y RESULTADOS

4.1. INTRODUCCIÓN

En el capítulo se muestran resultados obtenidos de la evaluación del prototipo y se procederá a realizar una valoración en torno a los objetivos, alcances y aportes de la tesis.

4.2. EVALUACIÓN

Para la evaluación del prototipo se efectuaron 14 experimentos, en ambientes controlados. El desarrollo del sistema se lo realizó con el IDE Matlab 2012a, para la captura de imágenes se empleó una cámara HP Webcam, alternativamente se utilizó una cámara Genius look 110. Para la identificación de la mano en la imagen se utilizó un guante de lana de color verde y otro guante de goma también de color verde.

Los niveles de iluminación que se experimentaron fueron: escasa iluminación, iluminación estable⁶, excesiva iluminación. Las distancias cámara – usuario que se experimentaron fueron: medio metro, un metro y dos metros.

4.2.1. Casos experimentales

- **Experimento 1:**

El primer caso se realizó bajo los siguientes parámetros:

Condiciones:

Iluminación: escasa iluminación

Distancia aproximada: medio metro.

⁶ La iluminación estable será definida como: la iluminación intermedia entre excesiva y escasa iluminación.

Material:

Guante de lana

Cámara HP Webcam

Respuesta: error en la obtención de datos de la imagen capturada. La figura 28, gráfica los resultados hallados.

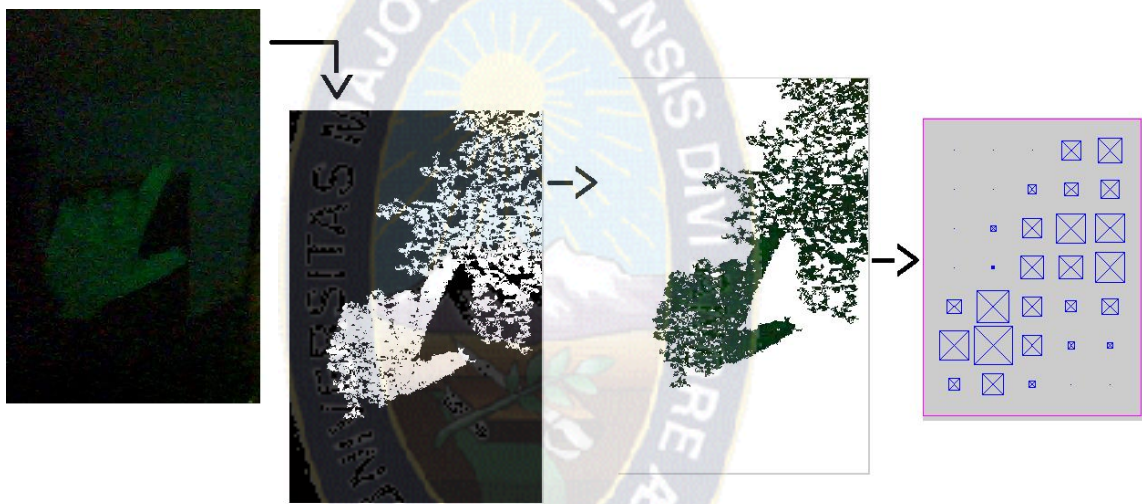


Fig. 28. Escasa iluminación, guante de lana

Fuente: Elaboración propia

- **Experimento 2:**

Este caso se realizó bajo los siguientes parámetros:

Condiciones:

Escasa Iluminación

Distancia aproximada: medio metro.

Material:

Guante de goma

Cámara HP Webcam

Respuesta: error en la obtención de datos de la imagen capturada, seña no identificada, nivel de confianza de clasificación del 72.4233%. La figura 29 gráficas los resultados hallados.

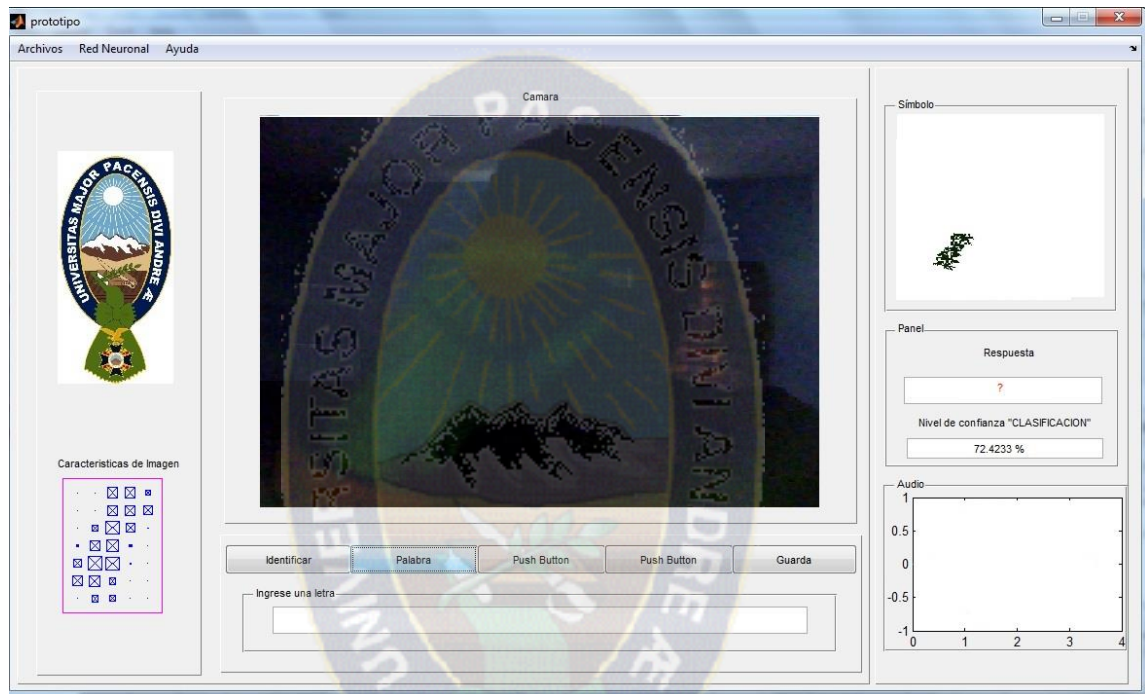


Fig. 29. Experimento 2

Fuente: Elaboración propia

- **Experimento 3:**

Los parámetros para este caso fueron:

Condiciones:

Iluminación: iluminación estable

Distancia aproximada: medio metro.

Material:

Guante de lana

Cámara HP Webcam

Respuesta: buena obtención de datos de la imagen capturada, seña identificada, nivel de confianza de clasificación del 90.9986 %. La figura 30 detalla los resultados

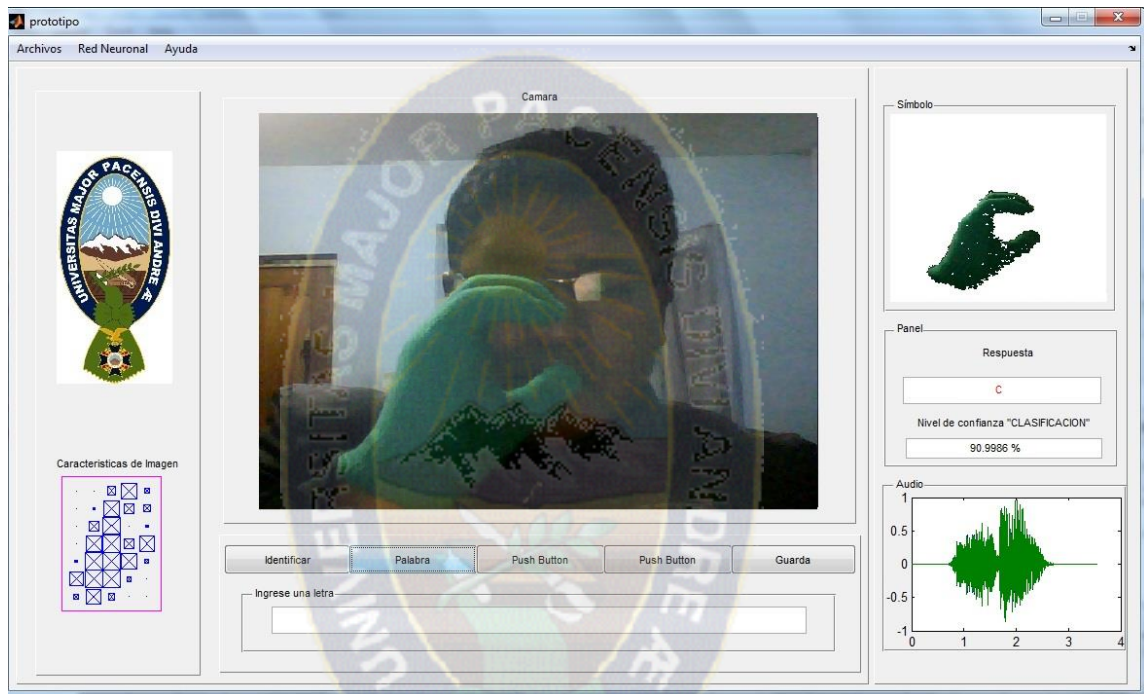


Fig. 30. Captura de la seña con guante de lana.

Fuente: Elaboración propia

- **Experimento 4:**

Los parámetros para el experimento fueron:

Condiciones:

Iluminación: iluminación estable

Distancia aproximada, medio metro

Material:

Guante de goma

Cámara HP Webcam

Respuesta: mejora en la obtención de datos de la imagen capturada, seña identificada, nivel de confianza de clasificación del 91.0095 %. La figura 31 detalla los resultados

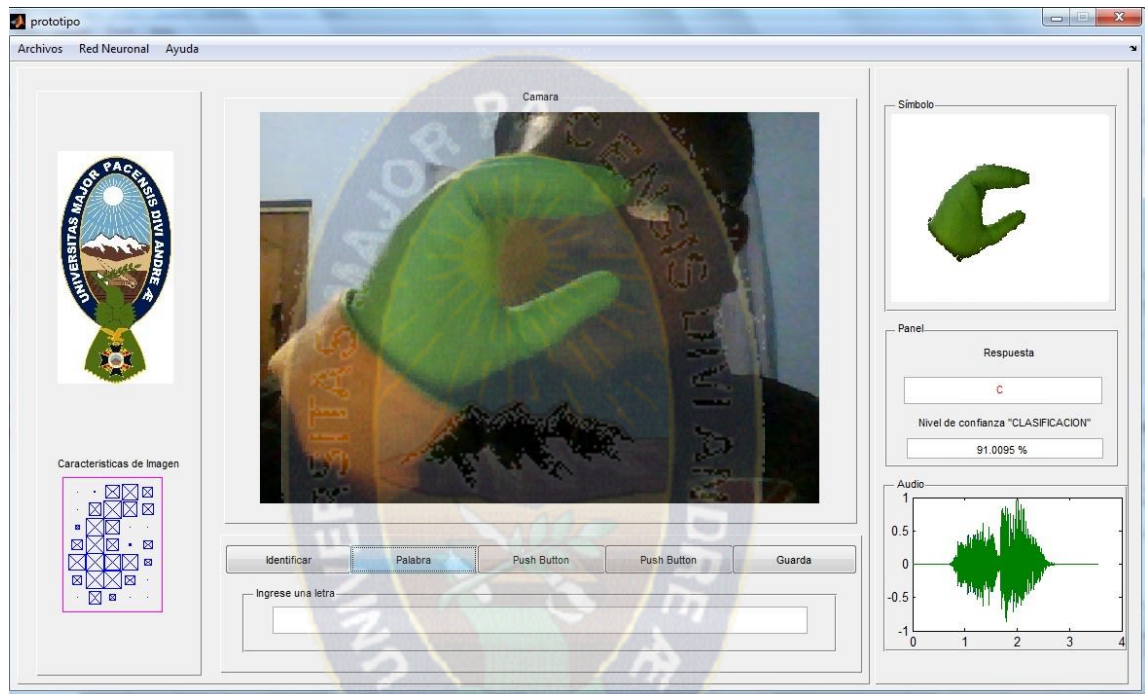


Fig. 31. Captura de la seña con guante de goma.

Fuente: Elaboración propia

- **Experimento 5:**

Este caso se realizó bajo los siguientes parámetros:

Condiciones:

Excesiva iluminación

Distancia aproximada, medio metro.

Material:

Guante de lana

Cámara HP Webcam

Respuesta: error en la obtención de datos de la imagen capturada, no detecta la seña. Nivel de confianza del 0,0%, seña no identificada. La figura 32 grafica los resultados.

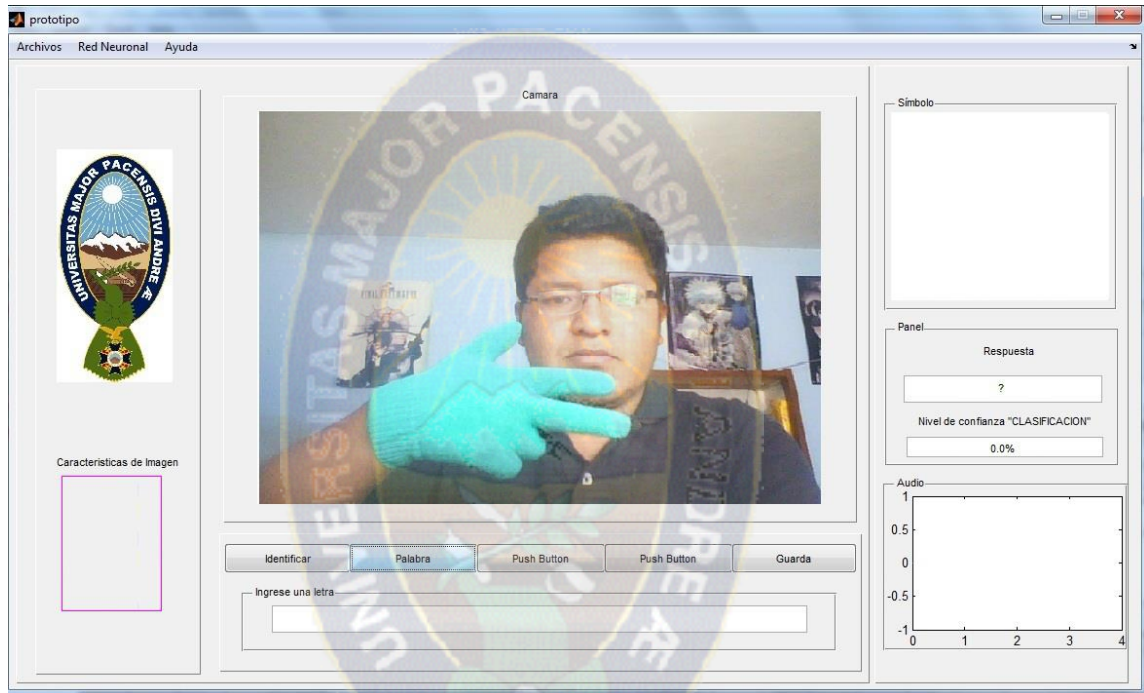


Fig. 32. Experimento 5, excesiva iluminación, guante de lana

Fuente: Elaboración propia

- **Experimento 6:**

Este caso se realizó bajo los siguientes parámetros:

Condiciones:

Excesiva iluminación

Distancia aproximada, medio metro.

Material:

Guante de goma

Cámara HP Webcam

Respuesta:

Error en la obtención de datos de la imagen capturada. Nivel de confianza de 75.3387% seña no identificada. La figura 33 es grafica los resultados hallados.



Fig. 33. Experimento 6, excesiva iluminación

Fuente: Elaboración propia

Los experimento 7 a 14, se realizaron con individuos que evaluaron las respuestas del sistema. Los experimentos se realizaron en función a la distancia entre webcam y persona que realiza la seña. La iluminación utilizada en los experimentos fue la iluminación estable.

- **Experimento 7:**

Este caso se realizó bajo el siguiente detalle.

Condiciones:

Iluminación: iluminación estable

Distancia aproximada: medio metro.

Material:

Guante de lana

Cámara HP Webcam

Respuesta: respuesta correcta con un nivel de confianza de clasificación del 87.6502%. La figura 34, señala la captura de pantalla del prototipo realizando la interpretación de la seña.



Fig. 34. Experimento 7

Fuente: Elaboración propia

- **Experimento 8:**

Condiciones:

Iluminación: iluminación estable

Distancia aproximada, medio metro.

Material:

Guante de goma

Cámara HP Webcam

Respuesta: respuesta correcta con un nivel de confianza de clasificación del 90.0849%. En la figura 35 vemos la captura de pantalla del prototipo realizando la interpretación de la seña.



Fig. 35. Experimento 8

Fuente: Elaboración propia

- **Experimento 9:**

Este caso se realizó bajo el siguiente detalle:

Condiciones:

Iluminación: iluminación estable

Distancia aproximada, un metro.

Material:

Guante de lana

Cámara HP Webcam

Respuesta: este experimento no dio respuesta correcta, el nivel de confianza de clasificación del 0.0%. En la figura 36 se aprecian los resultados obtenidos.

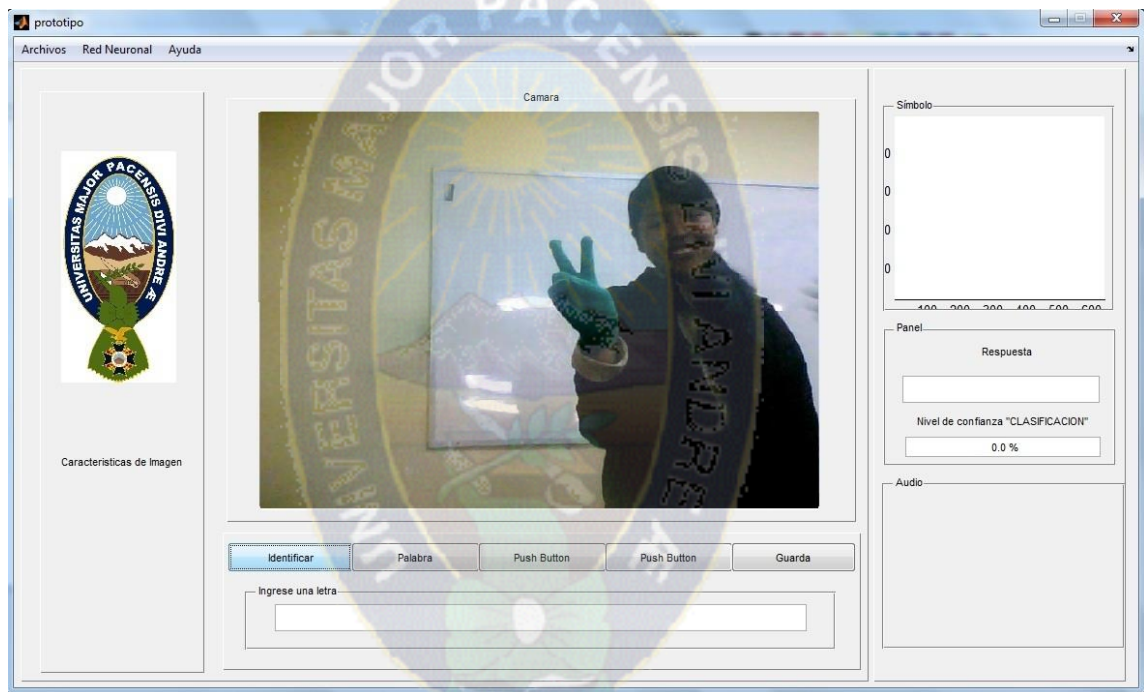


Fig. 36. Experimento 9

Fuente: Elaboración propia

- **Experimento 10:**

Este caso se realizó bajo el siguiente detalle:

Condiciones:

Iluminación: iluminación estable

Distancia aproximada, un metro.

Material:

Guante de goma

Cámara HP Webcam

Respuesta: los resultados fueron correctos, el nivel de confianza de la seña es de 88.3212%. En la figura 37 se aprecian los resultados obtenidos.

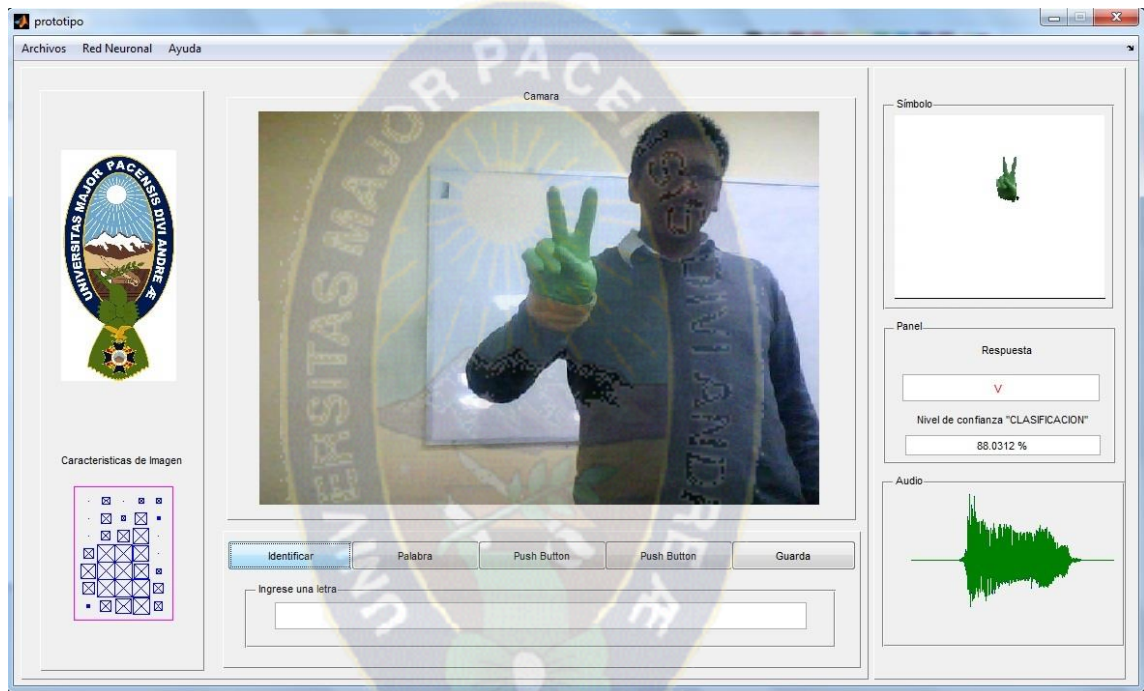


Fig. 37. Experimento 9

Fuente: Elaboración propia

- **Experimento 11:**

Este caso se realizó bajo el siguiente detalle:

Condiciones:

Iluminación: iluminación estable

Distancia aproximada, dos metros.

Material:

Guante de lana

Cámara HP Webcam

Respuesta: sin respuesta con un nivel de confianza de clasificación del 0.0%.

La figura 38 grafica los resultados hallados.

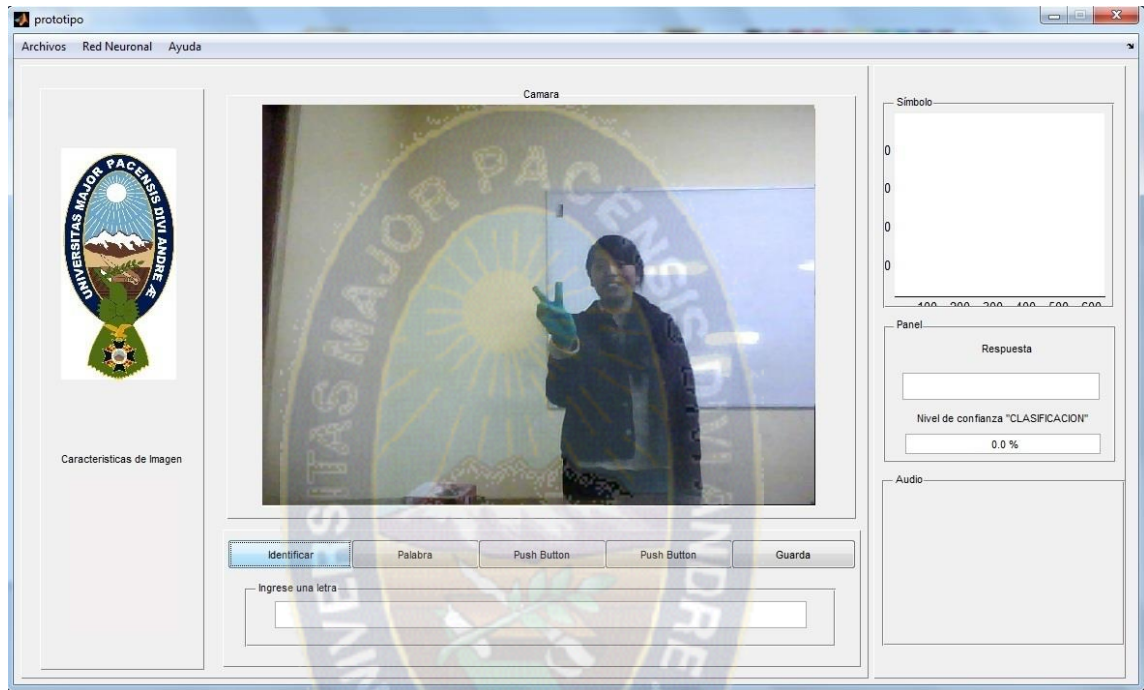


Fig. 38. Experimento 11

Fuente: Elaboración propia

- **Experimento 12:**

Este caso se realizó bajo el siguiente detalle:

Condiciones:

Iluminación: iluminación estable

Distancia aproximada, dos metros.

Material:

Guante de goma

Cámara HP Webcam

Respuesta: respuesta errónea, con un nivel de confianza de clasificación del 85.0006%. La figura 39 grafica los resultados hallados

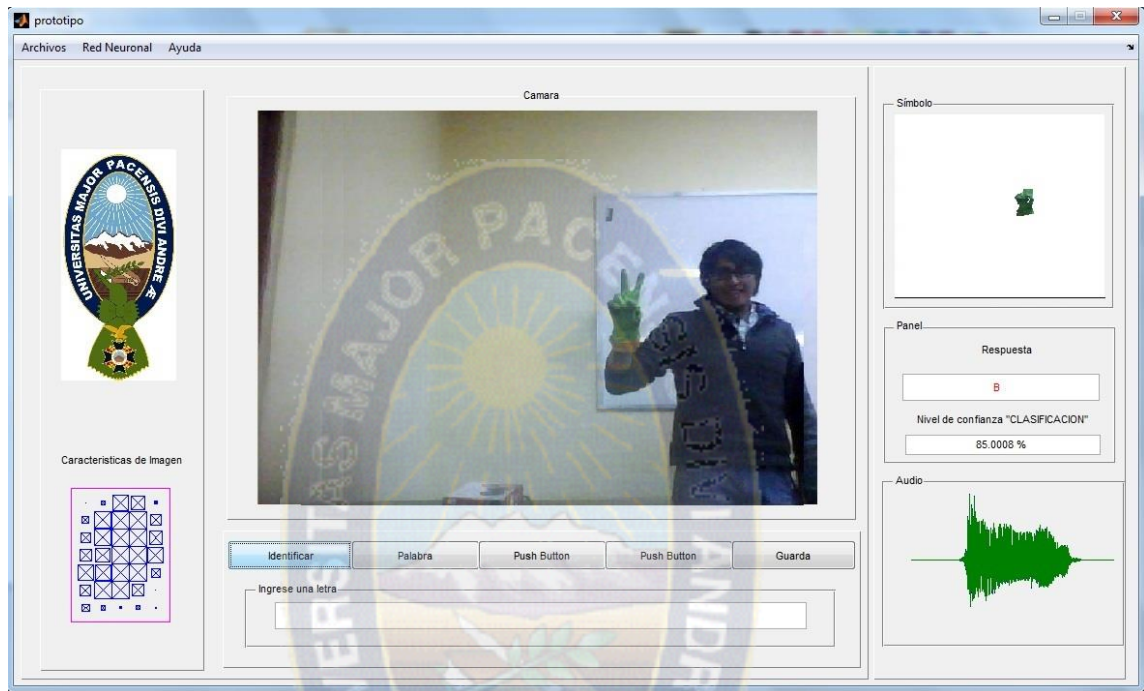


Fig. 39. Experimento 12

Fuente: Elaboración propia

- **Experimento 13:**

Este caso se realizó bajo el siguiente detalle:

Condiciones:

Iluminación: iluminación estable

Distancia aproximada, un metro.

Material:

Guante de lana

Cámara look 110 Genius

Respuesta: sin respuesta, con un nivel de confianza de clasificación del 0.0302%.

La figura 40 grafica los resultados hallados

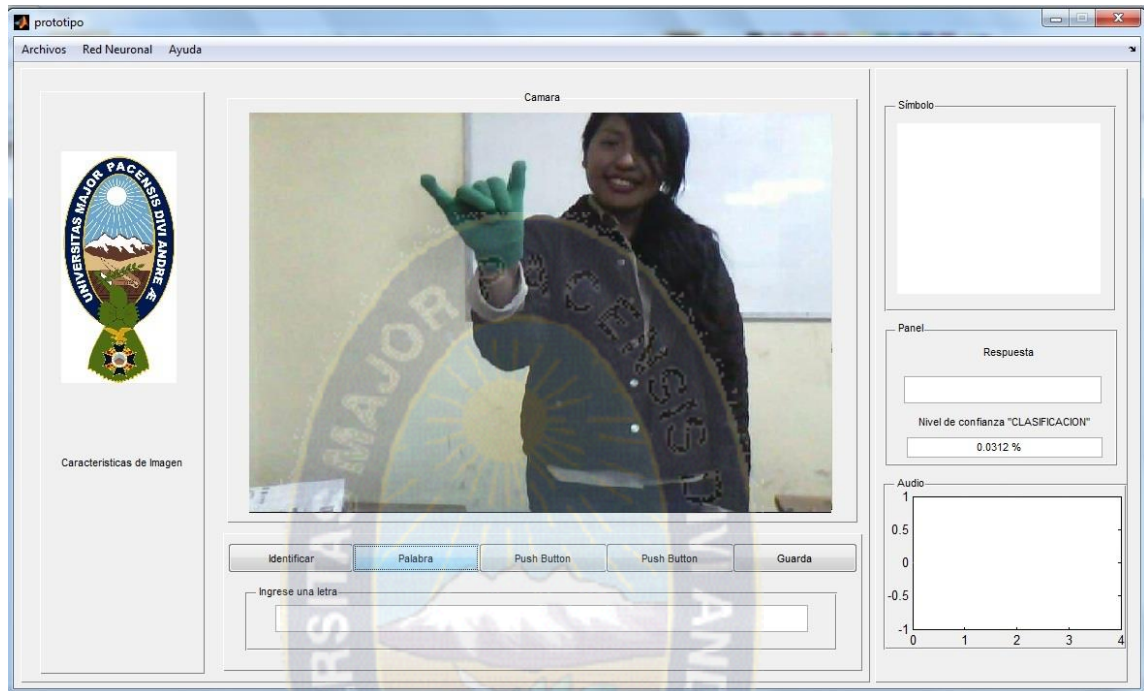


Fig. 40. Experimento 13

Fuente: Elaboración propia

- **Experimento 14:**

Este caso se realizó bajo el siguiente detalle:

Condiciones:

Iluminación: iluminación estable

Distancia aproximada, un metro.

Material:

Guante de goma

Cámara look 110 Genius

Respuesta: respuesta correcta con un nivel de confianza de clasificación del 89.0812%. La figura 41 grafica los resultados hallados

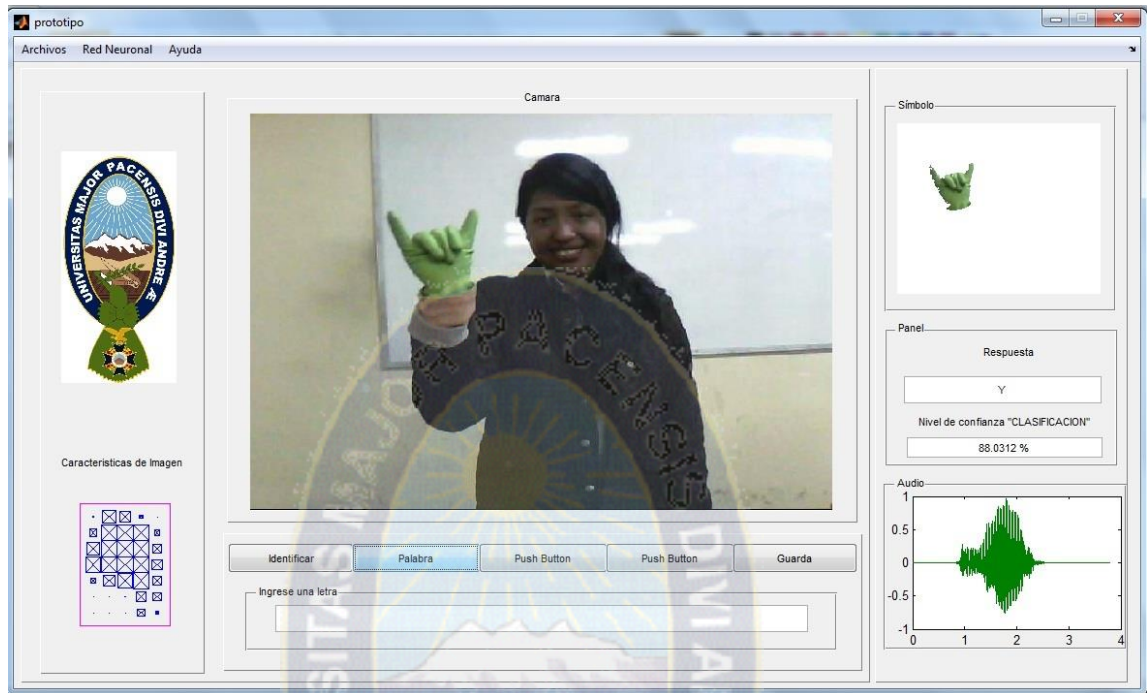


Fig. 41. Experimento 14

Fuente: Elaboración propia

4.2. ANÁLISIS DE RESULTADOS

Los experimentos realizados permitieron la construcción y la mejora del prototipo. De los experimentos se obtuvo lo siguiente.

- Uno de los factores con mayor influencia tiene para la interpretación de la seña es la iluminación del ambiente, ya que la inadecuada iluminación, distorsiona la calidad de la imagen con respecto al color de los objetos presentes en la captura o la no detección de dichos objetos. La distorsión ocasiona que la seña no sea debidamente reconocida en experimentos 1 y 4 se puede apreciar la mala obtención de datos y la distorsión que causada por la falta de iluminación y excesiva iluminación respectivamente. En el experimento 2, la falta de iluminación no permite la detección de la seña.

- En los experimento 3 y 4 se obtienen mejores respuestas, el material utilizado no afecta en demasía a la obtención de datos. La iluminación media mejora la calidad de imagen obtenida.
- Los experimentos 7 y 8 a una distancia aproximada de medio metro tiene respuestas acertadas, el experimento realizado con el guante de goma supera al realizado con el guante de lana, en el nivel de confianza de clasificación del 2.4347%.
- Los experimentos 9 y 10 fueron realizado a una distancia aproximada de un metro, el experimento con el guante de lana (experimento 9) no da una respuesta, la influencia de la iluminación es mucho mayor. El experimento con el guante de goma (experimento 10), tiene respuesta acertada con una disminución del nivel de confianza.
- Los experimentos 11 y 12 realizados a una distancia aproximada de dos metros, que dan los siguientes resultados, el experimento realizado con el guante de lana (experimento 11) no tiene respuesta, no es detectada la seña. El experimento con el guante de goma (experimento 12), tiene respuesta errada, no adquiere datos correctos de la seña confundiendo la seña realizada.
- Para los experimentos 13 y 14 se cambió la cámara HP Webcam por una Genius look-110, con una distancia aproximada de un metro. El experimento 13, realizado con el guante de lana no dio una respuesta. El experimento 14 respondió con un nivel de confianza menor a la del experimento 10.
- La escasa o excesiva iluminación ocasionan que el prototipo obtenga datos erróneos para la interpretación de la seña.

- Con el guante de goma se mejora en gran medida la captura de la imagen, además que la influencia de la iluminación fue no es muy importante. Mientras que en los resultados obtenidos en la captura de la imagen de la seña con el guante de lana la iluminación tiene una mayor influencia si es escasa o si es excesiva.
- La iluminación con luces blancas mejoró la obtención de datos de la seña realizada por la mano.





CONCLUSIONES Y RECOMENDACIONES

CAPITULO V

CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

De acuerdo a los resultados logrados se llega a las siguientes conclusiones.

- De particular relevancia es la iluminación del ambiente que se tiene al realizar la captura de la imagen, ya que una excesiva o una escasa iluminación genera distorsión, que conlleva la pérdida de datos o la deformación de la seña.
- La distancia ideal entre la webcam y la persona que realiza la seña, debe ser menor o igual a un metro, puesto que en los experimentos se demuestra que a una mayor distancia se tiene mayor distorsión con la subsecuente pérdida de datos.
- Las capturas de las imágenes de las señas con los mejores resultados fueron realizadas con el guante de goma.
- En las capturas de las imágenes de las señas que se realizaron con el guante de lana, la iluminación del ambiente tuvo una mayor influencia, ocasionando la pérdida de datos y un mayor error en las respuestas del prototipo.
- Con respecto al objetivo general, se logró realizar el prototipo que realiza la interpretación de las señas al lenguaje oral.
- Se cumplen los objetivos específicos, puesto que para cumplir con la interpretación del lenguaje de señas se realizaron:

- ✓ La captura de la imagen de la seña con una cámara digital es evidente
- ✓ El tratamiento de la imagen mediante métodos informáticos
- ✓ El entrenamiento de las redes neuronales para que posteriormente se realice la clasificación de la seña

Finalmente fue posible la interpretación oral de la captura digital de la seña con el módulo de selección de sonido.

5.2. RECOMENDACIONES

A continuación se mencionan los que podrían ser trabajos futuros a partir del proyecto de tesis expuesto.

- ✓ Se propone el desarrollo de una versión del prototipo para tecnologías móviles, haciendo que el sistema sea portable.
- ✓ Podría ampliarse la cantidad de señas interpretadas, ya que en el lenguaje de señas existen varias palabras realizadas con señas estáticas, esto implicaría la ampliación del número de entradas de las redes neuronales.
- ✓ Para ampliar la interpretación a señas con movimiento de manos, se podría trabajar con mayor cantidad de imágenes capturadas, por cada seña realizada.
- ✓ Para que el sistema, responda en tiempo real, se podría realizar un módulo que identifique las señas en video.



BIBLIOGRAFÍA

BIBLIOGRAFÍA

Referencias escritas

- BasogainOlabe, X. (2010). *Redes neuronales artificiales y sus aplicaciones*. Bilbao: Escuela Superior de Ingeniería de Bilbao, Dpto. Ingeniería de Sistemas y Automática.
- Bunge, M. (2004). *Emergencia y Convergencia, Novedad cualitativa y Unidad del Conocimiento*. Barcelona - España: Gedisa.
- Germán Sergio Monroy Alvarado (2008) . Más allá del conocimiento: un enfoque sistémico: UAM-X, México, Diciembre
- Grimaldos, J. J. (2005). Tratamiento digital de imágenes, Un recurso didáctico. En J. J. Grimaldos, *Tratamiento digital de imágenes*. Publicación digital.
- González, M. Á. (1991). *Lenguaje de Signos*. Valladolid.
- Herrera, L. S. (2010). Manejador de Imágenes "¿Que es una imagen digital?". En S. A. Herrera, *Manejador de Imágenes*. Publicación web.
- Javier García de Jalón, J. R. (2005). *Aprenda Matlab 7.0, como si estuviera en primero*. Madrid: Universidad politécnica de Madrid, Publicación digital.
- Olabe, X. B. (s.f.). *Redes neuronales artificiales y sus aplicaciones*. Bilbao: Escuela Superior de ingeniería de Bilbao, Dpto. Ingeniería de Sistemas Automática.
- Reyes, M. A. (2006). *Algoritmo Backpropagation para Redes Neuronales*:. Ciudad de México: INSTITUTO POLITÉCNICO NACIONAL.
- Señas, A. I. Traductor en-linea de lenguaje de señas. *Maya / Software para convertir texto a lenguaje de signos*. Maya soft, España.

Referencias electrónicas

Academic. (2008). *Convolucion*. Recuperado el Septiembre de 2013, de sitio web de Academin:
http://enciclopedia_universal.esacademic.com/13093/Convoluci%C3%B3n

EcuRed. (s.f.). *EcuRed*. Recuperado el Agosto de 2013, de Lenguaje de signos:
http://www.ecured.cu/index.php/Lenguaje_de_signos

Josep Antequera Baiget. (2013). *La teoría de Sistemas*,
Recuperado de: www.eumed.net

Kiwix. (2011). *Kiwix*. Recuperado el Septiembre de 2013, de www.kiwix.com:
www.kiwix.com/red_neuronal_artificial

Serrano, C. (2012). *El blog de Carlos*. Recuperado el 14 de Octubre de 2013, de
<http://carloserrano10colegioesclavas.wordpress.com/>:
<http://carloserrano10colegioesclavas.wordpress.com/articulos-destacados/>

UNAM. (2007). *UNAM*. Recuperado el Septiembre de 2013, de <http://www.fi-b.unam.mx>: <http://www.fi-b.unam.mx/Profesores/AbelHerrera/RedesNeuronales/homeRedes.html>

Wikipedia. (s.f.). *www.wikipedia.com*. Recuperado el 24 de Agosto de 2013, de
http://es.wikipedia.org/wiki/Lengua_de_se%C3%B1as

Wikipedia. (2013). *Mario Bunge*.
Recuperado de: [http://es.wikipedia.org/wiki/Mario%Bunge](http://es.wikipedia.org/wiki/Mario%20Bunge)