

PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO  
FACULTAD DE INGENIERÍA  
ESCUELA DE INGENIERÍA INFORMÁTICA

**SISTEMA DE RECONOCIMIENTO GESTUAL DE  
LENGUA DE SEÑAS CHILENA MEDIANTE CÁMARA  
DIGITAL**

**CARLOS GUILLERMO GONZÁLEZ RIVEROS  
FRANCISCO JAVIER YIMES INOSTROZA**

INFORME FINAL DEL PROYECTO  
PARA OPTAR AL TÍTULO PROFESIONAL DE  
INGENIERO DE EJECUCIÓN EN INFORMÁTICA

NOVIEMBRE 2016

Pontificia Universidad Católica de Valparaíso  
Facultad de Ingeniería  
Escuela de Ingeniería Informática

**SISTEMA DE RECONOCIMIENTO GESTUAL DE  
LENGUA DE SEÑAS CHILENA MEDIANTE CÁMARA  
DIGITAL**

**CARLOS GUILLERMO GONZÁLEZ RIVEROS  
FRANCISCO JAVIER YIMES INOSTROZA**

Profesor Guía: **Claudio Cubillos Figueroa**

Profesor Co-Referente: **Iván Mercado Bermúdez**

Carrera: **Ingeniería de Ejecución Informática**

Noviembre 2016

## Resumen

Los gestos expresados con las manos son frecuentemente utilizados debido a su gran nivel de expresividad. Para el reconocimiento de gestos con las manos es necesario percibir posiciones tanto en ángulos como en rotaciones. Para ello, es necesario especificar los puntos de inicio y fin tanto en espacio como en tiempo del patrón gestual y enseguida, segmentar el gesto relevante e interpretar la información. Sin embargo, es posible encontrarse con ambigüedad en la segmentación, una constante variabilidad en el espacio-tiempo y esto sumado a la complejidad morfológica que posee la mano, se convierte en un interesante problema a resolver, sin embargo, la tecnología y las investigaciones sobre algoritmos para perfeccionar el reconocimiento gestual ha ido en aumento, de tal manera que la misma tecnología se ha ido desarrollando a la par para seguir perfeccionando el reconocimiento gestual.

Estas técnicas de procesamiento de imágenes que nos permiten analizar fotogramas y/o imágenes estáticas reales, para adquirir información importante y resolver problemas específicos son parte del estudio de la visión artificial o en su rama más general parte de la inteligencia artificial.

*Palabras-claves: Gestos, Patrón, Segmentar, Visión Artificial, Inteligencia Artificial, Algoritmos, Procesamiento de Imágenes, Reconocimiento Gestual, Espacio-Tiempo y Tecnología.*

## Abstract

Expressed gestures with hands are often used because of its high level of expression. For the recognition of hand gestures it is necessary to perceive both positions and rotation angles. To do this, you must specify the start and end points in both space and time and then gestural pattern, segment the gesture and interpreting relevant information. However, it is possible to find ambiguity in segmentation, constant variability in space-time and this added to the morphological complexity that the hand has, becomes an interesting problem to solve. However, technology and research algorithms to refine the gesture recognition has increased, so that the same technology has been developed in tandem to further improve the gesture recognition.

These imaging techniques that allow us to analyze frames and / or actual static images, to acquire important information and resolve specific problems are part of the study of artificial vision or in its most general branch of artificial intelligence.

*Keywords: Gestures, Pattern, Segment, Computer Vision, Artificial Intelligence, Algorithms, Image Processing, Gesture Recognition, Space-Time, and Technology.*

# Índice

<b>Dedicatoria.....</b>	<b>v</b>
<b>Glosario de Términos.....</b>	<b>vi</b>
<b>Lista de Abreviaturas o Siglas .....</b>	<b>vii</b>
<b>Lista de Figuras .....</b>	<b>viii</b>
<b>Lista de Tablas .....</b>	<b>ix</b>
<b>Introducción .....</b>	<b>1</b>
<b>1. Planteamiento del Tema .....</b>	<b>2</b>
1.1 Problemática y Motivación .....	2
1.2 Definición de Objetivos .....	2
1.2.1 Objetivo general .....	2
1.2.2 Objetivos específicos.....	2
1.3 Planificación del Proyecto .....	3
1.3.1 Carta Gantt .....	3
1.4 Estructura del Documento .....	5
1.5 Metodología de Desarrollo .....	5
1.5.1 Metodología Rapid Application Development .....	5
1.6 Estudio de Factibilidad .....	6
1.6.1 Factibilidad Técnica .....	6
1.6.1.1 Hardware .....	6
1.6.1.2 Software.....	7
1.6.2 Factibilidad Operativa.....	7
1.6.3 Factibilidad Económica.....	7
1.6.4 Factibilidad Legal.....	7
1.7 Análisis de Riesgo .....	8
<b>2. Estado del Arte .....</b>	<b>10</b>
2.1 Reconocimiento Gestual .....	10
2.1.1 Clasificación.....	10
2.1.2 Aplicaciones .....	11
2.2 Análisis de Gestos.....	11
2.2.1 Detección de Características .....	11
2.2.2 Estimación de Parámetros .....	12
2.3 Sistemas de Reconocimiento Gestual .....	12
2.3.1 LeapMotion .....	12

2.3.2	Wisee.....	13
2.3.3	Kinect .....	13
2.3.4	Openarch .....	14
2.4	Desarrollo Gestual .....	14
2.4.1	OpenCV.....	14
<b>3.</b>	<b>Marco Teórico .....</b>	<b>16</b>
3.1	Lengua de Señas .....	16
3.2	Imágenes Digitales.....	16
3.2.1	Tipo de Imágenes Digitales.....	17
3.2.1.1	Imágenes de Mapa Bits .....	17
3.2.1.2	Imágenes Vectoriales .....	17
3.2.2	Colores .....	18
3.2.2.1	Espacio RGB .....	18
3.2.2.2	Espacio HSI.....	19
3.3	Preprocesamiento de Imágenes.....	19
3.3.1	Manipulación de Contraste.....	19
3.3.2	Eliminación de Ruido.....	20
3.3.3	Realce de Bordes.....	20
3.4	Extracción de Características.....	21
3.4.1	Detección de Bordes.....	21
3.4.2	Texturas.....	22
3.4.3	Detección de Movimiento .....	23
3.5	Segmentación de Imágenes.....	23
3.6	Máquinas de Aprendizaje (Machine Learning) .....	24
3.6.1	Más allá de las Líneas .....	24
3.6.2	Machine Learning y OpenCV .....	25
<b>4.</b>	<b>Análisis y Diseño.....</b>	<b>26</b>
4.1	Análisis Requerimientos del Proyecto .....	26
4.1.1	Requerimientos Funcionales .....	26
4.1.2	Requerimientos No Funcionales .....	26
4.2	Diagrama de Casos de Uso .....	26
4.2.1	Caso de Uso General .....	27
4.2.2	Casos de Uso Específicos.....	27
4.2.2.1	Caso de Uso Detectar Mano.....	28

4.2.3	Caso de Uso Guardar Gesto .....	28
4.2.4	Caso de Uso Entrenar Gesto.....	29
4.2.5	Caso de Uso Reconocer Gesto .....	30
<b>5.</b>	<b>Desarrollo de Sistema.....</b>	<b>32</b>
5.1	Arquitectura para el Reconocimiento en Android .....	32
5.2	Detección de la Mano .....	32
5.2.1	Detección del Fondo y Mano por Separados .....	32
5.2.2	Imagen Binaria .....	33
5.2.3	Detección de Contornos y Estructura de la Mano.....	33
5.2.4	Enumeración de los dedos de la mano .....	33
5.3	Entrenamiento de Gestos .....	34
5.4	Reconocimiento de Gestos.....	35
5.5	Implementación .....	35
5.5.1	Android.....	35
5.5.1.1	Sistema Android.....	35
5.5.2	OpenCV para Android .....	35
5.5.2.1	OpenCV Manager.....	36
5.6	Visualización .....	36
<b>6.</b>	<b>Pruebas y Resultados .....</b>	<b>39</b>
6.1	Capturas Realizadas .....	39
6.1.1	Resultados Caso de Estudio N°1 Ambiente Controlado .....	39
6.1.2	Resultados Caso de Estudio N°2 Ambiente no Controlado .....	39
6.1.3	Resultados Caso de Estudio N°3 Ambiente bajo luz artificial .....	40
<b>7.</b>	<b>Conclusión.....</b>	<b>41</b>
<b>8.</b>	<b>Referencias Bibliográficas .....</b>	<b>42</b>

# Dedicatoria

A nuestras familias, quienes a lo largo de toda nuestras vidas nos han apoyado y motivado en nuestra formación académica, creyeron en nosotros en todo momento y no dudaron de nuestras habilidades.

## **Glosario de Términos**

**Lengua:** Se llama "lengua" al conjunto o sistema de formas o signos orales y escritos que sirven para la comunicación entre las personas de una misma comunidad lingüística.

**Lenguaje:** El lenguaje es el medio de comunicación de los seres humanos, quienes utilizamos signos orales y escritos, sonidos y gestos que poseen un significado que les hemos atribuido.

**Fotograma:** Secuencia de imágenes captadas por cámaras de video digital.

**E-Learning:** educación y capacitación a través de Internet. Este tipo de enseñanza online permite la interacción del usuario con el material mediante la utilización de diversas herramientas informáticas.

**Segmentar:** En fotografía, es el proceso de dividir una imagen digital en varias partes (grupos de píxeles) u objetos.

**Píxel:** Unidad básica de una imagen digitalizada en pantalla a base de puntos de color o en escala de grises.

**2D:** "2D" es sinónimo de "dos dimensiones". Las formas 2D incluyen cuadrados y círculos.

**3D:** "3D" significa tridimensional. Las formas 3D tienen profundidad, e incluyen cubos y esferas.

**Punto o Grano:** Es el elemento más simple y a la vez más complejo de la imagen. Es el elemento más utilizado en el plano del diseño. Morfológica y compositivamente, el punto no es la representación geométrica de ese concepto, sino que tiene una dimensión relativa y variable pudiendo adoptar infinitas formas.

**Luminancia:** Resultado que se obtiene al dividir la intensidad luminosa de una superficie entre su área aparente para un observador alejado de ella.

## **Lista de Abreviaturas o Siglas**

MHI: Motion History Image.

D.O.S: Domestic Operation System.

RGB: Red Green Blue.

HSI: Hue Saturation Intensity.

USB: Universal Serial Bus.

Nº: Número.

SVM: Máquina de Soporte Vectorial.

## Lista de Figuras

Figura 1.1 Gráfico Plan de Trabajo. [Elab. Propia].....	4
Figura 1.2 Metodología RAD. [Elab. Propia] .....	6
Figura 2.1 Ejemplo de Reconocimiento Gestual.....	10
Figura 2.2 Proceso de Reconocimiento Gestual. [Elab. Propia] .....	11
Figura 2.3 Dispositivo LeapMotion. [21].....	12
Figura 2.4 Detección de Gestos Wisee. [23] .....	13
Figura 2.5 Reconocimiento Kinect [Elab. Propia] .....	13
Figura 2.6 Openarch en Acción. [Elab. Propia] .....	14
Figura 3.1 Alfabeto Lengua de Señas Chilenas [10].....	16
Figura 3.2 Dos rejillas de 5x5 píxeles [5].....	17
Figura 3.3 Representación tridimensional del espacio RGB.....	18
Figura 3.4 HSI: (a) Tono, (b) Saturación, (c) Intensidad. [6].....	19
Figura 3.5 (a) Imagen Original, (b) Aumento de Contraste. [2].....	20
Figura 3.6 Diferencia entre imagen sin ruido y con ruido. [9] .....	20
Figura 3.7 Realce de bordes, (a) imagen original, (b) imagen con filtro gaussiano para eliminar ruido, (c) realce de bordes sin filtro gaussiano, (d) realce de bordes con filtro gaussiano.....	21
Figura 3.8 (a) Imagen Real, (b) Imagen Binarizada. [13] .....	22
Figura 3.9 Propiedades de textura: (a) Suavidad, (b) Rugosidad y (c) Regularidad .....	22
Figura 3.10 Independencia del tono o el color en la textura de una imagen digital. ....	23
Figura 3.11 Representación Gráfica ML. ....	24
Figura 3.12 Representación Tridimensional ML.....	25
Figura 3.13 Ejemplo código de SVM usando OpenCV . ....	25
Figura 4.1 Diagrama de caso de uso general. ....	27
Figura 4.2 Diagrama de Caso de Uso Detectar Mano. ....	28
Figura 4.3 Diagrama de Caso de Uso Guardar Gesto. ....	29
Figura 4.4 Diagrama de Caso de Uso Entrenar Gesto.....	30
Figura 4.5 Diagrama de Caso de Uso Reconocer Gesto.....	31
Figura 5.1 Arquitectura para el Reconocimiento de la mano en Android. ....	32
Figura 5.2 Detección de Fondo y Detección de la Mano. ....	33
Figura 5.3 Imagen Binarizada .....	33
Figura 5.4 Mano detectada con extracción de características, enumerada de 0 a 4. ....	33
Figura 5.5 Entrenamiento Letra B. ....	34
Figura 5.6 Entrenamiento Letra C. ....	34
Figura 5.7 Entrenamiento Letra D.....	34
Figura 5.8 Inicio de la aplicación. ....	34
Figura 5.9 Interfaz de Reconocimiento. ....	34
Figura 5.10 Captar Fondo, binarizar imagen captada y Obtener Características. ....	34
Figura 5.11 Reconocimiento de gesto almacenado y entrenado .....	34
Figura 6.1 Letras del Alfabeto A - B - C - D.....	39
Figura 6.2 Falsos Positivos.....	39

## Lista de Tablas

Tabla 1.1 Carta Gantt Proyecto. [Elab. Propia] .....	4
Tabla 1.2 Tabla de Riesgos. [Elab. Propia] .....	9
Tabla 4.1 Especificación de Caso de Uso: Detectar Mano.....	27
Tabla 4.2 Especificación de Caso de Uso: Guardar Gesto. ....	28
Tabla 4.3 Especificación de Caso de Uso: Entrenar Gesto .....	29
Tabla 4.4 Especificación de Caso de Uso: Reconocer Gesto .....	30
Tabla 6.1 Estudio N°1, 35 Pruebas en Ambiente Controlado. ....	39
Tabla 6.2 Estudio N°2, 35 Pruebas en Ambiente no Controlado. ....	40
Tabla 6.3 Estudio N°1, 35 Pruebas en Ambiente bajo luz artificial. ....	40

# Introducción

La comunicación gestual ha sido, desde que se tiene conocimiento, el más importante medio de comunicación, no sólo para las personas, sino que para muchas especies animales. Cuando dos personas que hablan distintos idiomas se tratan de comunicar, utilizan gestos para facilitar la comunicación; desde gestos faciales hasta corporales, podemos entender lo que quiere expresar otra persona sin necesidad del uso de palabras; un simple movimiento de la boca, ojos, cejas, manos o cualquier extremidad, representa algo en específico. Por ejemplo, cuando queremos indicar en qué dirección se encuentra un lugar, apuntamos con la mano para referirnos a dónde se debe dirigir la persona.

Es de esta manera que las personas con dificultad auditiva logran comunicarse, estandarizando significados gestuales de manera amplia y así logrando expresar y compartir tanta información como la comunicación oral. Esta forma de comunicación gestual se denomina Lengua de Señas.

En este proyecto se pretende generar un avance en los estudios relacionados con estas tecnologías y generar un sistema que permita el reconocimiento de la Lengua de Señas, y de esta forma poder ayudar y generar inclusión no solo a personas no oyentes, sino que también a aquellos que no pueden usar instrumentos como el mouse o el teclado, pero que pueden expresarse a través de gestos.

La motivación principal es la inclusión mediante el avance tecnológico, lograr facilitar tareas cotidianas e incluso en el mundo laboral para aquellas personas que presentan una discapacidad. Un ejemplo claro de esta problemática es la de un trabajador sordo que debe comunicarse cara a cara con su cliente.

Lense® no solo pretende ser un traductor de la lengua de señas, sino también una herramienta de aprendizaje con la cual todos podamos culturizarnos con esta lengua, y así, generar una real inclusión.

# **1. Planteamiento del Tema**

## **1.1 Problemática y Motivación**

Las personas con discapacidad auditiva utilizan la lengua de señas para comunicarse, lengua que solo ellos y un grupo reducido adicional de personas dominan. Por esta razón, es que, hasta el día de hoy estas personas no pueden comunicarse con oyentes, y sumándole además que son un porcentaje menor en la población, por lo tanto, se puede entender que estas personas deben sentir excluidas, por gran parte de la población.

La única forma de poder interactuar con este grupo de personas no oyentes, que no es menor, es aprendiendo la lengua de señas a través de cursos presenciales y/o cursos E-Learning.

La motivación que nos lleva a la realización de este proyecto es, principalmente, el hecho de poder ayudar a personas que tienen problemas para comunicarse y se ven un tanto excluidos por la sociedad.

Tomando en cuenta que la lengua de señas se basa en la utilización de gestos como comunicación, y que las tecnologías de reconocimiento de gestos pertenecientes a la lengua de señas se encuentran aún en investigación y desarrollo un tanto inmaduro, se encontró una nueva motivación para contribuir y constituir una base sólida de conocimientos para el posterior desarrollo de tecnologías que permitan utilizar el reconocimiento gestual en diferentes ámbitos de progreso.

A pesar del inmaduro desarrollo tecnológico respecto al reconocimiento gestual, algunos avances se han logrado hoy en día que han dado mucho de qué hablar, como por ejemplo la llegada de la “Kinect”, la cual ha dado un importante empuje al reconocimiento gestual. Además, se han construido importantes librerías de softwares tanto libres como privadas que han ayudado bastante a este tipo de reconocimiento.

Mediante este proyecto se pretende aportar una contribución a los avances en este campo creando y desarrollando una nueva forma de aplicar el reconocimiento gestual a través una cámara digital por medio de dispositivos móviles smartphone o teléfonos inteligentes que permitan que una persona con capacidad limitada y/o una persona de su entorno se comuniquen entre sí, sin la necesidad de saber la lengua de señas.

## **1.2 Definición de Objetivos**

### **1.2.1 Objetivo general**

Desarrollar un sistema de reconocimiento gestual, implementando algoritmos de procesamiento y segmentación de imágenes, utilizando librerías de OpenCV, por medio de un lenguaje de programación, a través de una cámara digital.

### **1.2.2 Objetivos específicos**

Los objetivos específicos del actual proyecto son:

- Conocer y aprender sobre el uso de OpenCV, implementado los algoritmos de reconocimiento de patrones.

- Aprender el uso de máquinas de soporte vectorial para el entrenamiento y autoaprendizaje de un software de inteligencia artificial, y en nuestro caso, del reconocimiento de patrones, proporcionado por OpenCV.
- Implementar una máquina de aprendizaje a nuestro sistema de software.
- Entrenamiento mediante el software de aprendizaje con letras del alfabeto de lengua de señas.
- Lograr una tasa de error menor al 20% en las pruebas realizadas.

## **1.3 Planificación del Proyecto**

El proceso de gestión de un proyecto de software comienza con un conjunto de actividades que, globalmente, se denominan planificación del proyecto. No podemos pedir exactitud a la fase de planificación, es solo una idea de cómo van a transcurrir las cosas. Hay que planificar el trabajo, los recursos humanos y la tecnología. Planificar es estimar.

Es por eso que hemos desarrollado una carta Gantt para la correcta elaboración de este avance de proyecto dos, en donde se definieron a grandes rasgos las tareas que se realizaron, no se entró en detalles de la investigación, se tomaron los puntos más relevantes e importantes a desarrollarse a lo largo de este avance.

### **1.3.1 Carta Gantt**

A continuación, se presenta en la tabla 1.1 la Carta Gantt donde se especifican cada una de las etapas del presente proyecto y además de su respectivo grafico de plan de trabajo.

Tabla 1.1 Carta Gantt Proyecto. [Elab. Propia]

Actividades	Fecha de Inicio	Duración
Requerimientos Funcionales	07-03-2016	4
Requerimientos No Funcionales	12-03-2016	2
Diagrama de Casos de Uso General	15-03-2016	4
Diagrama de Casos de Uso Específicos	20-03-2016	3
Diseño de Interfaz	24-03-2016	2
Documentación	27-03-2016	3
Investigación Sobre Machine Learning	31-03-2016	5
Investigación Machine Learning en OpenCV	06-04-2016	5
Investigación Entrenamiento de Máquina	12-04-2016	5
Investigación Entrenamiento en OpenCV	18-04-2016	5
Modelo de Procesos	24-04-2016	1
Documentación	26-04-2016	2
Desarrollo de Layouts	29-04-2016	1
Implementación Librería OpenCV	01-05-2016	3
Implementación Machine Learning	05-05-2016	5
Implementación Algoritmos de Entrenamiento	11-05-2016	6
Implementación Algoritmo de Reconocimiento de Imagen	18-05-2016	6
Implementación Algoritmo de Reconocimiento de Patrones	25-05-2016	6
Pruebas Unitarias	01-06-2016	1
Pruebas Integrales	03-06-2016	1
Documentación	05-06-2016	2
Análisis de Errores	08-06-2016	1
Conclusión y Resultados	10-06-2016	2

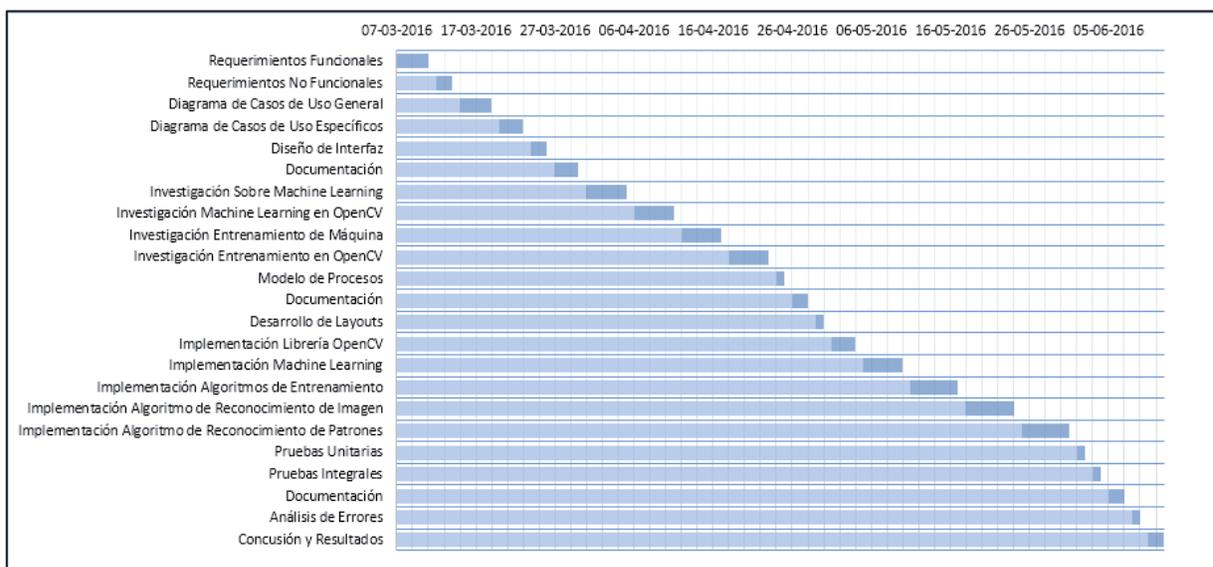


Figura 1.1 Gráfico Plan de Trabajo. [Elab. Propia]

## 1.4 Estructura del Documento

El presente documento se divide en 8 capítulos, los cuales serán descritos de manera breve y precisa a continuación:

- Capítulo 1, Planteamiento del tema: En este capítulo, se describirá la motivación, se establecerán y definirán objetivos, junto con la metodología de desarrollo, el estudio de factibilidad y finalmente la planificación del proyecto.
- Capítulo 2, Estado del arte: En este capítulo, se verán las tecnologías que ya se han desarrollado sobre reconocimiento gestual, se definirán el concepto de reconocimiento gestual y las alternativas que se tienen para el desarrollo del proyecto.
- Capítulo 3, Marco teórico: En este capítulo, se podrá ver en primera instancia la definición de la lengua de señas chilenas, y se verán los principales aspectos del reconocimiento de imágenes digitales.
- Capítulo 4, Análisis y Diseño: En este capítulo, se verán las representaciones técnicas del software, llámese diagrama de casos de uso, especificaciones de los casos de uso y diseños de interfaces. En el análisis se verán los requerimientos del software tanto funcional como no funcional.
- Capítulo 5, Desarrollo del Sistema: En este capítulo, se puede observar el desarrollo del presente proyecto y las herramientas utilizadas para el reconocimiento de la mano y el alfabeto.
- Capítulo 6, Pruebas y Resultados: En este capítulo, se realizaran pruebas en distintos ambientes para evaluar la calidad de reconocimiento del algoritmo de entrenamiento propuesto, abordando 3 casos de estudio, un ambiente controlado bajo las mejores condiciones de luz, en un ambiente normal, y otro caso bajo luz artificial sin controlar el ambiente.
- Capítulo 7, Conclusiones: En este capítulo, se presentarán las conclusiones a las que se ha llegado tras el marco de investigación de reconocimiento gestual.
- Capítulo 8, Referencias: En este último capítulo se encuentran todas aquellas referencias, de las cuales se tomaron ciertos apuntes que permitieron que el proyecto fuera abordado de manera exitosa a lo largo de todo el período de desarrollo.

## 1.5 Metodología de Desarrollo

El desarrollo de software no es una tarea fácil. Prueba de ello es que existen numerosas propuestas metodológicas que inciden en distintas dimensiones del proceso de desarrollo. Por una parte, tenemos aquellas propuestas más tradicionales que se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, y las herramientas y notaciones que se usarán. Estas propuestas han demostrado ser efectivas y necesarias en un gran número de proyectos, pero también han presentado problemas en muchos otros.

Una posible mejora es incluir en los procesos de desarrollo más actividades, más artefactos y más restricciones, basándose en los puntos débiles detectados. Sin embargo, el resultado final sería un proceso de desarrollo más complejo que puede incluso limitar la propia habilidad del equipo para llevar a cabo el proyecto.

### 1.5.1 Metodología Rapid Application Development

El desarrollo rápido de aplicaciones (RAD) es una metodología de desarrollo de software, que implica el desarrollo iterativo y la construcción de prototipos. El desarrollo rápido de aplicaciones es un término originalmente utilizado para describir un proceso de desarrollo de software introducido por James Martin en 1991.

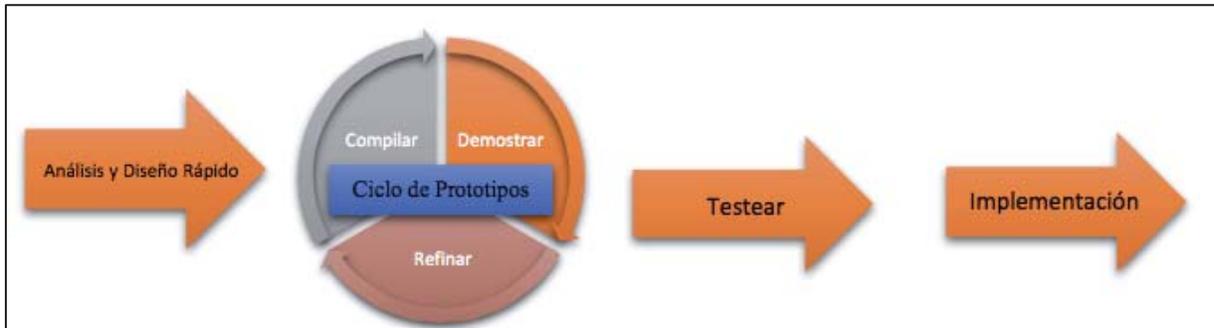


Figura 1.2 Metodología RAD. [Elab. Propia]

Si bien se tiene la intención de desarrollar un software, pero que a su vez se desea llevar a cabo en un dispositivo móvil, es por eso que se ha decidido trabajar con esta metodología y no con otra, ya que para que el reconocimiento gestual se pueda lograr se deberá someter a varias pruebas de imagen, diseño de la aplicación y entre muchos otros factores que harán modificar el proyecto a medida que se vayan logrando ciertos requerimientos propuestos en la etapa de análisis del proyecto.

## 1.6 Estudio de Factibilidad

El estudio de factibilidad es el encargado de determinar la infraestructura tecnológica y la capacidad técnica que implica la implantación del sistema dentro de un proyecto, así como también los costos, beneficios y el grado de riesgo que se pueda tener al momento de implementar esta nueva tecnología en sistemas móviles. El proyecto debe cumplir con ciertos requisitos técnico necesarios para el correcto desarrollo de la aplicación o software, los cuales pueden ser determinantes para el correcto funcionamiento del sistema que se quiere implementar. Este caso en particular, son de gran importancia los requerimientos tecnológicos para realizar pruebas en tiempo real, que permitan implementar el software óptimamente para el usuario.

### 1.6.1 Factibilidad Técnica

La factibilidad técnica es aquella necesidad, disponibilidad y capacidad del hardware como software sean suficientes para el desarrollo de la aplicación o software.

#### 1.6.1.1 Hardware

La herramienta de desarrollo utilizada para su correcto uso, requiere un computador de prestaciones normales en conjunto con un teléfono Smartphone de gama alta para la captación, procesamiento e interpretación de una señal de video continua desde la cámara digital y que trabaje bajo el sistema operativo Android o IOS. Los requerimientos específicos mínimos de hardware del Smartphone son los siguientes [32]:

- Velocidad CPU 2.1GHz Octa-Core.
- Resolución 2560 x 1440 (Quad HD).

- Cámara Trasera Principal CMOS 16.0 MP.
- Sensores: Acelerómetro, Barómetro, Fingerprint, Giroscopio, Geomagnético, Hall, HR, Luminosidad, Proximidad.
- Entrada USB.
- Memoria de Almacenamiento Mín. 4GB.

En el actual desarrollo de la aplicación se utilizará un computador personal (Notebook) con los requerimientos suficientes para ejecutar y alojar el entorno de desarrollo. En cuanto a la captura del video digital, se requiere de una cámara digital que tenga una óptica especial que capture la escena de forma completa y con gran apertura.

### **1.6.1.2 Software**

En cuanto a las necesidades de software, se utiliza el IDE para desarrollo de aplicaciones móviles, Android Studio, desarrollado por Google, el cual nos permite programar en lenguaje Java, Android, XML en conjunto con las librerías de procesamientos de imágenes de OpenCV, las cuales son de código abierto y de libre acceso, las cuales permiten ir probando de manera agilizada el funcionamiento del software.

### **1.6.2 Factibilidad Operativa**

La factibilidad operativa son todas aquellas posibles condiciones que permitirán tener un aplicativo en todas las condiciones ideales, permitiendo el correcto funcionamiento sobre todo a la hora de hacer el reconocimiento y/o proceso de segmentación de imagen.

- La cámara debe mantenerse estable para hacer un enfoque óptimo a la mano sin que tome ningún otro elemento de condiciones similares a la mano.
- El fondo debe contrastar con la mano tratando de elegir colores altamente distantes, así como por ejemplo si la mano es blanca, es de esperar que el fondo sea negro.
- Las condiciones de iluminación deben ser altas, ya que si son bajas se provoca un mayor ruido, evitar realizar sombras, ideal un enfoque de luz directo sobre la mano.
- Para el manejo de la tasa de error se recomienda enfocar directamente la mano sin ningún otro factor que implique que el algoritmo detecte falsos positivos.

### **1.6.3 Factibilidad Económica**

La factibilidad económica es aquella condición que sirve para ver si es posible el desarrollo del sistema o aplicativo con respecto a los gastos que este conlleva. El hardware necesario para poder trabajar en el desarrollo del prototipo preliminar, es totalmente por parte del ente desarrollador, por lo cual no se incurre a gastos tanto como en sistema de desarrollo como Notebook o el dispositivo móvil con cámara integrada.

En cuanto a los gastos incurridos en el software necesario para el desarrollo de la aplicación, el IDE de Android Studio es totalmente gratuito además de las librerías que son de código abierto especializadas en el tratamiento de imágenes. En definitiva, no se incurre en gastos extras por parte de los desarrolladores.

### **1.6.4 Factibilidad Legal**

Es de mucha importancia, para el desarrollo del proyecto, y que, de acuerdo a todos los estudios de factibilidad, todo se encuentre en orden y sea permitido por el marco de la ley de proyectos informáticos de Chile. Cabe mencionar que todas y cada uno de las herramientas,

licencias, herramientas de procesador de texto entre otras se encuentran efectivizadas y de acorde a la ley 19.233 de delitos informáticos.

## **1.7 Análisis de Riesgo**

El presente proyecto presenta una planificación definida a grandes rasgos en las cartas Gantt y que todo resultara como lo esperado con una carga positiva, sin retrasos y cumpliendo los plazos establecidos, pero no contempla los retrasos que podrían ocurrir al llevar a la practica un proyecto de desarrollo de software, muchas veces se cree que todo es posible, pero pasa que no todo lo que se puede imaginar se encuentra en desarrollo, ya sea una funcionalidad o que las tecnologías necesarias no sean las más óptimas para el trabajo que se está realizando.

A continuación, se ve una tabla la cual está compuesta por ID de riesgo, medidas de impacto y todos aquellos riesgos que sean de alta criticidad para su desarrollo normal, estas medidas de impacto que se dividen en 4 categorías:

- **Bajo Impacto:** que no afecta mayormente al desarrollo del sistema, siendo ésta categoría mayormente enfocada en la gestión del proyecto. Llámese Redacción de informe, recopilación de información, desarrollo de Gantt, etc.
- **Medio Impacto:** afecta medianamente el desarrollo del proyecto, siendo un hito importante el cálculo ideal de horas de desarrollo de aplicativos dentro del sistema.
- **Alto Impacto:** que afecta directamente al proyecto, como el mal uso o el mal aprendizaje de las herramientas que se utilizaran para su desarrollo, tanto el IDE, los lenguajes y/o problemas técnicos como el deterioro del hardware de desarrollo.
- **Crítico:** afecta al tanto al proyecto como al desarrollador, en este caso al alumno para un buen final de entrega de proyecto ya sea en avances como en presentaciones finales, fallando el lado técnico, personal y/o lógico del desarrollador y la mala comunicación entre el profesor guía y el alumnado.

Tabla 1.2 Tabla de Riesgos. [Elab. Propia]

ID	Impacto	Probabilidad	Riesgo	Consecuencia
001	Medio	40%	Mal estimación de tiempo.	Atraso en entregas.
002	Critico	20%	Mal uso de herramientas.	Cambio de Objetivos
003	Critico	40%	Mal estudio de factibilidad.	Cambio de Objetivos
004	Alto	20%	Poco dominio de las herramientas.	Cambio de Objetivos, Atraso en entregas.
005	Alto	10%	Problemas de comunicación con el profesor.	Atraso en entregas de proyecto, desmotivación.
006	Critico	30%	Mal planteamiento de los requerimientos.	Proyecto no cumple las expectativas.
007	Critico	20%	Falta de herramientas	Cambio de Objetivos, Atraso en entregas.
008	Critico	40%	Subestimación de los límites del proyecto.	Cambio de Objetivos, Atraso en entregas.

## 2. Estado del Arte

### 2.1 Reconocimiento Gestual

El gesto en sí es un movimiento corporal con significado, que puede ser realizado con los dedos, las manos, los brazos, la cabeza, con todo el tronco; o incluso con una variación de la expresión de la cara o mediante el movimiento de ojos. Por sí mismo constituye una parte importante de la comunicación natural cotidiana, en muchas ocasiones como refuerzo al lenguaje hablado y, cuando éste no es posible, como la primera alternativa de comunicación. [8]

El reconocimiento gestual es un tema de la visión por computador o bien de la visión artificial, área de la inteligencia artificial, además de la tecnología del lenguaje que tiene como objetivo reconocer e interpretar gestos y/o patrones humanos a través de algoritmos matemáticos aplicados a la imagen que se reconoce mediante ciertas técnicas de visión, por medio de dispositivos de entrada como cámaras, webcams, o dispositivos móviles.

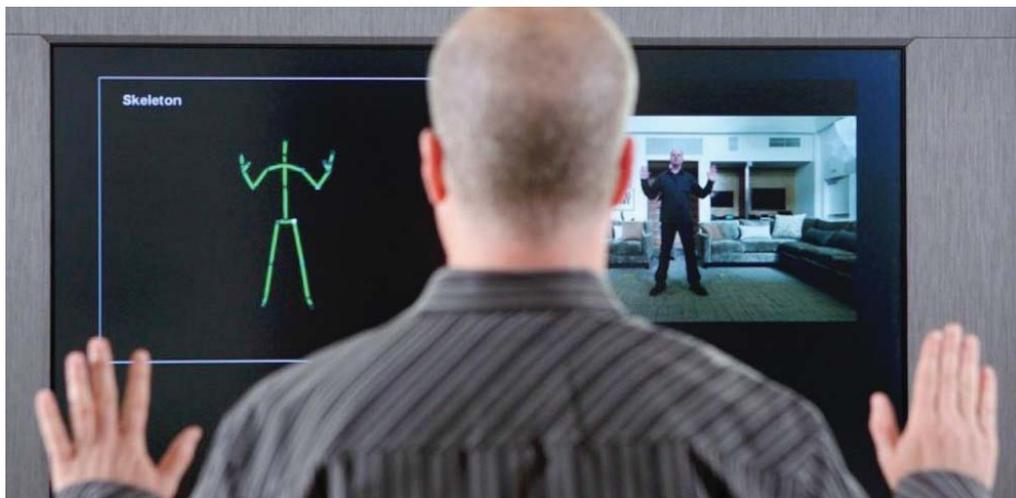


Figura 2.1 Ejemplo de Reconocimiento Gestual.

Como se puede ver en la Figura 2.1, no solo se trata de reconocer manos o cara, sino toda actividad corporal que produzca un cambio significativo a la imagen, pero como nuestro proyecto está centrado en el reconocimiento del alfabeto de lengua de señas chilena, éste está enfocado en el rescate de la forma y movimiento de manos.

#### 2.1.1 Clasificación

Los gestos se pueden clasificar como gestos estáticos y dinámicos:

- Estáticos: El usuario asume cierta postura, o configuración de su cuerpo o de la cara, fija.
- Dinámicos: En los que la información está codificada en el movimiento, a través de la evolución temporal de las posturas, o bien una mezcla de ambos, como, por ejemplo, la lengua de señas. [8]

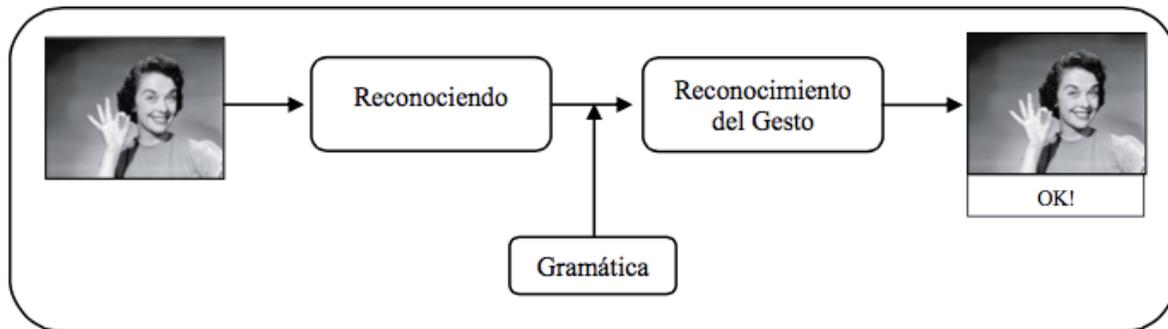


Figura 2.2 Proceso de Reconocimiento Gestual. [Elab. Propia]

## 2.1.2 Aplicaciones

Las aplicaciones posibles son numerosas y sólo están limitadas por la imaginación. Éstas podrían influir, no sólo específicamente en la industria de la informática, sino también cambiando la forma en la que las personas se comunican, se transportan o son atendidas por sistemas automáticos, pues dichos sistemas serían más conscientes de la situación física de sus usuarios. Esencialmente el reconocimiento de gestos puede ser extremadamente útil para:

- Reconocimiento de la lengua de señas.
- Navegar o realizar manipulaciones en entornos de realidad virtual o realidad aumentada.
- Computación afectiva. Las expresiones faciales podrían ser usadas por el ordenador para interpretar estados emocionales del usuario.
- Controladores virtuales, manejo de acciones como el apagar la luz en un espacio realizando señas que se reconozcan a través de señales de onda.
- Nuevas tecnologías de ocio y juegos de tipo inmersivo. El estado de los espectadores, posición o postura podría ser captado e integrado en la acción.

## 2.2 Análisis de Gestos

El objetivo de este paso es estimar los parámetros del modelo usando mediciones de las imágenes de video. En este paso se usan dos tareas: la detección de características, en donde se busca extraer características relevantes de la imagen, y la estimación de parámetros, usando las características para calcular los parámetros del modelo [26]. La importancia de este paso es inmensa, teniendo en cuenta que un error en la extracción de características implica la obtención de una mala clasificación.

### 2.2.1 Detección de Características

En la detección de características el objetivo es encontrar y señalar los aspectos claves de la imagen que serán utilizados en la estimación de los parámetros del modelo gestual escogido. Aquí se dan dos procesos: la localización de la persona que realiza los gestos y, una vez completado este proceso, se puede proceder a la detección de las características deseadas [26].

Con la localización se busca detectar a la persona que ejecuta los gestos, diferenciándola del resto de la imagen (o también llamado ambiente). Para ello, se utilizan diferentes modelos y

métodos de rastreo [30], en los que por lo general se ejecutan dos pasos: la segmentación de la figura-fondo y la correspondencia temporal.

## 2.2.2 Estimación de Parámetros

La estimación de parámetros es la última etapa de la fase de análisis de gestos, depende de dos factores: las características seleccionadas, y los parámetros requeridos por el modelo de reconocimiento [26]. Los modelos basados en la apariencia generalmente se usan para describir o identificar acciones gestuales. La estimación de parámetros de estos modelos usa una descripción compacta de la imagen o de la secuencia de imágenes. Existen varias formas de definir parámetros para estos modelos; un primer método es seleccionar un conjunto de fotogramas o frames importantes visualmente como los parámetros del modelo (por ejemplo, los frames que contengan el objeto por reconocer) [26]. Otro método es utilizar la acumulación de información espacio-temporal en una secuencia de imágenes y transformarla en una sola imagen 2D; este método se denomina motion history image (MHI) [31]. Esa imagen puede ser fácilmente parametrizada utilizando técnicas de descripción de imágenes en 2D como la descripción de momentos geométricos o la descomposición por valores propios [26]. La mayor ventaja de los enfoques basados en la apariencia es la simplicidad en la etapa de estimación de parámetros; sin embargo, esta simplicidad puede hacer que en algunos casos se pierda precisión en la información espacial de las imágenes [26].

## 2.3 Sistemas de Reconocimiento Gestual

### 2.3.1 LeapMotion

Fundado en el 2010 por Michael Buckwald, David Holz, en San Francisco, California, EE.UU.

El controlador LeapMotion es un pequeño dispositivo USB que está diseñado para ser colocado en un escritorio físico, mirando hacia arriba. Utiliza dos cámaras de infrarrojos monocromática y tres infrarrojos LEDs, el dispositivo observa un área aproximadamente semiesférica, a una distancia de aproximadamente 1 metro. Los LEDs generan patrones con luz Infrarroja y las cámaras generan cerca de 300 fotogramas por segundo, que se envían a través de un cable USB al ordenador central, donde es analizada por el software del controlador de LeapMotion usando "Matemáticas Complejas". Motion Controller resalta un área de observación más pequeña y de mayor resolución a diferencia del producto Kinect, que es más adecuado para todo el cuerpo, seguimiento en un espacio del tamaño de una sala de estar. Algunas de las tareas a realizar son: navegar por un sitio web, el uso de gestos de pellizcar para hacer zoom en los mapas, dibujos de alta precisión, y la manipulación de visualizaciones de datos 3D complejas. [21]



Figura 2.3 Dispositivo LeapMotion. [21]

### 2.3.2 Wisee

Wisee es una novedosa interfaz de interacción que aprovecha las señales inalámbricas en curso en el medio ambiente, para permitir la detección y el reconocimiento de gestos humanos. Dado que las señales inalámbricas no requieren de líneas de visión y pueden atravesar las paredes. Wisee permite que en toda la casa sea controlada sin una gran cantidad de fuentes inalámbricas por ejemplo un router wifi y un par de dispositivos móviles en alguna habitación. Wisee es el primer sistema que permite reconocer gestos en la línea de visión, sin línea de visión y a través de la pared de distintos escenarios, no así otros sistemas de reconocimiento de gestos como Kinect, LeapMotion.

Wisee no requiere ninguna infraestructura de cámaras ni de instrumentación de dispositivos para los usuarios. Hasta el momento Wisee detecta 9 gestos con un promedio del 94% de precisión. [23]

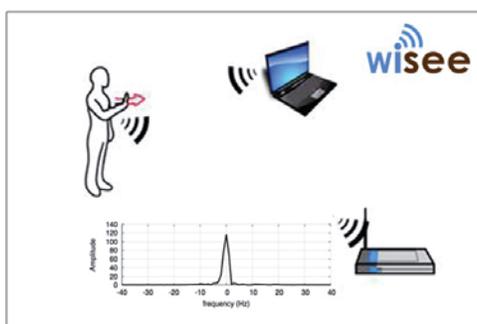


Figura 2.4 Detección de Gestos Wisee. [23]

### 2.3.3 Kinect

Kinect (originalmente conocido por el nombre en clave Project Natal), es un controlador de juego libre y entretenimiento creado por Alex Kipman, desarrollado por Microsoft para la videoconsola Xbox 360, y desde junio del 2011 para PC a través de Windows 7 y Windows 8. Kinect permite a los usuarios controlar e interactuar con la consola sin necesidad de tener contacto físico con un controlador de videojuegos tradicional, mediante una interfaz natural de usuario que reconoce gestos, comandos de voz, objetos e imágenes. El dispositivo tiene como objetivo primordial aumentar el uso de la Xbox 360, más allá de la base de jugadores que posee en la actualidad. Kinect fue lanzado en Norteamérica el 4 de noviembre de 2010. [24]

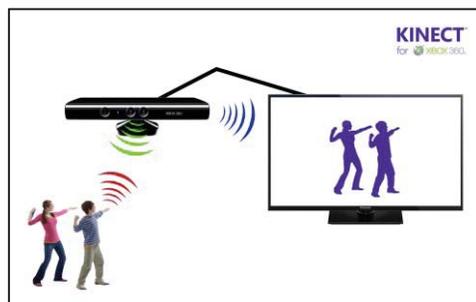


Figura 2.5 Reconocimiento Kinect [Elab. Propia]

### 2.3.4 Openarch

Openarch es un prototipo real de una vivienda inteligente. La primera vivienda diseñada desde el inicio para incorporar una capa digital que conecta la casa y sus elementos a Internet. Sus habitantes participan y se incorporan a una nueva vida conectada y digital. Es flexible y gracias a su capacidad de transformación se adapta a cualquier condición que requiera el usuario.

La capa digital a la que a partir de ahora llamaremos D.O.S (domestic operating system) incluye una serie de elementos que permiten a los usuarios estar conectados con cualquier persona o espacio, controlar los elementos de la casa mediante el movimiento del cuerpo, realizar conferencias desde la casa, conocer el consumo eléctrico en cada instante, activar cualquier electrodoméstico desde el trabajo, compartir en video y en directo las recetas de cocina con el resto del mundo, crear tu propio plató de TV en el salón, etc. [22]

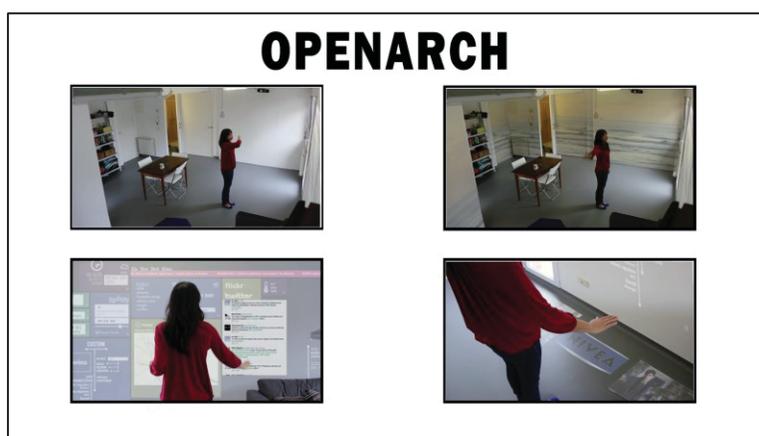


Figura 2.6 Openarch en Acción. [Elab. Propia]

## 2.4 Desarrollo Gestual

Se ha realizado una investigación sobre las librerías existentes que pudieran facilitar en todo lo posible la tarea de reconocimiento de movimiento gestual. Se realizó una búsqueda generalizada, considerando cualquier plataforma, valorando las ventajas e inconvenientes de cada una de ellas.

Dando como resultado OpenCV, una de las librerías más completas que existen hoy en día.

### 2.4.1 OpenCV

OpenCV (Open Source Computer Vision Library) es una fuente abierta de visión por computador y la biblioteca de software de aprendizaje automático. OpenCV fue construido para proporcionar una infraestructura común para aplicaciones de visión por computador y acelerar el uso de la percepción de la máquina en los productos comerciales.

La biblioteca cuenta con más de 2500 algoritmos optimizados, que incluye un amplio conjunto de clásico y algoritmos de visión por computador y aprendizaje automático.

Estos algoritmos se pueden utilizar para detectar y reconocer las caras, identificar objetos, clasificar las acciones humanas en videos, movimientos de cámara pista, objetos de la pista en movimiento, extraer modelos 3D de objetos, producen nubes de puntos 3D de cámaras estéreo,

unir imágenes juntos para producir una alta resolución imagen de toda una escena, encontrar imágenes similares de una base de datos de imágenes, eliminar los ojos rojos de las imágenes tomadas con flash, seguir los movimientos de los ojos, reconocer el paisaje y establecer marcadores para cubrirás de realidad aumentada, etc.

Cuenta con interfaces de C ++, C, Python, Java y MATLAB y es compatible con Windows, Linux, Android y Mac OS. OpenCV se inclina principalmente hacia las aplicaciones de visión en tiempo real, además de ser rápida, eficiente y de código abierto. [25]

### 3. Marco Teórico

#### 3.1 Lengua de Señas

La lengua de señas es la lengua natural de las personas sordas, se basa en movimientos y expresiones a través de ojos, rostro, boca, cuerpo y principalmente manos, siendo esta última nuestra principal base a estudiar.

Las personas sordas nacen con la capacidad biológica del lenguaje intacta. Sin embargo, su limitación auditiva impide que puedan apropiarse del idioma o lengua oral que se habla a su alrededor.

Al igual que los idiomas orales, existen diferentes lenguas de señas. Esto debido a que como ha pasado con los demás idiomas, las lenguas de señas han surgido de acuerdo a los procesos de educación o de influencia de unos países sobre otros al igual que a las características propias de cada región, así por ejemplo existe una lengua de señas colombiana, una lengua de señas argentina, una lengua de señas española, una lengua de señas americana, que no son iguales. Lo mismo que sucede con los idiomas orales, (español, francés, inglés de estados unidos, inglés de Inglaterra, portugués, etc.) las lenguas de señas también presenta diferencias. [4]

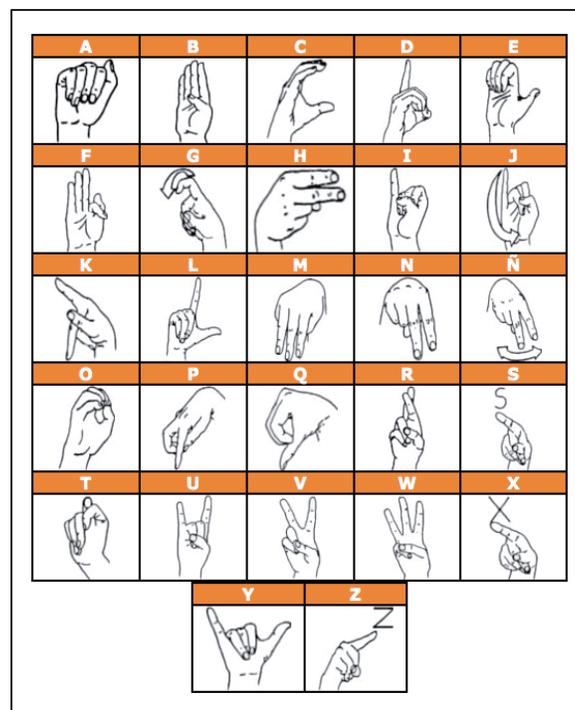


Figura 3.1 Alfabeto Lengua de Señas Chilenas [10]

#### 3.2 Imágenes Digitales

La imagen digital está formada por un conjunto definido de puntos llamados píxeles. La imagen tradicional, sobre material posible, también está formada por puntos o granos; la gran diferencia es que, en la fotografía tradicional, los granos son irregulares y están situados de forma aleatoria, mientras que en la digital forman una matriz con filas y columnas.

### 3.2.1 Tipo de Imágenes Digitales

Por la forma de manejar los datos en un archivo de imagen, se puede hablar de dos modos principales para manipular la información que integra una imagen digital. Estos modos son las imágenes de mapa de bits y las imágenes vectoriales. Dado que cada uno se adapta mejor a un tipo de imagen, antes de conocer los diferentes espacios, debe conocerse el funcionamiento tanto de imágenes vectoriales como de imágenes de mapa de bits.

#### 3.2.1.1 Imágenes de Mapa Bits

Las imágenes de mapa de bits están formadas por una rejilla de celdas. A cada una de estas celdas, que se denominan píxeles, se le asigna un valor de color y luminancia propios. Por esto, cuando vemos todo el conjunto de celdas, tenemos la ilusión de una imagen de tono continuo.

El pixel es una unidad de información, no una unidad de medida, ya que no se corresponde con un tamaño concreto. Un píxel puede ser muy pequeño (0.1 mm) o muy grande (1 cm). Cuando creamos una imagen de mapa de bits se genera una rejilla específica de píxeles; por esto, al modificar su tamaño, transformamos, a su vez, la distribución y coloración de los píxeles, por lo que los objetos, dentro de la imagen, suelen deformarse. Así los objetos pierden o ganan algunos de los píxeles que los definen, como se muestra en la figura 3.2.

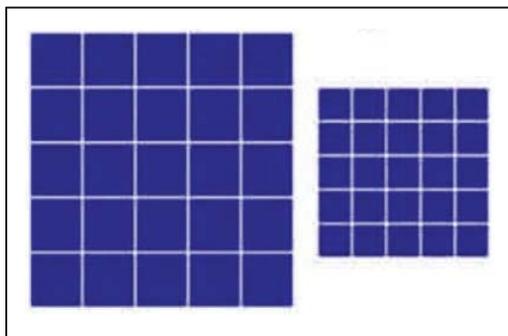


Figura 3.2 Dos rejillas de 5x5 píxeles [5]

Gracias a esta característica, que siempre hay que tener en cuenta, las imágenes de mapa de bits se crean con un tamaño determinado y pierden calidad si se modifican sus dimensiones.

#### 3.2.1.2 Imágenes Vectoriales

Los llamados gráficos orientados a objetos son las imágenes vectoriales. Su tamaño es mucho más reducido, en comparación con los mapas de bits, porque el modo como organizan la información de una imagen es más simple que en aquellos. Dicha simplicidad radica en generar los objetos que conforman una imagen a través de trazos geométricos determinados por cálculos y fórmulas matemáticas. De manera tal que los gráficos vectoriales se visualizan a partir de las coordenadas de una línea guardadas como referencia, mismas que forman los objetos a partir de la definición matemática de los puntos y líneas rectas o curvas.

Mientras que en un mapa de bits los objetos se definen píxel por píxel, por lo que no pueden manipularse individualmente, caso contrario sucede con las imágenes vectoriales, con ellas cada objeto, dentro de una imagen, puede modificarse sin que se alteren los demás. En un gráfico de vectores se delimitan por la posición de los puntos inicial y final y por la trayectoria

de la línea que los une. Como la información se guarda matemáticamente, esto hace posible que las imágenes vectoriales sean independientes de la resolución del monitor, ya que, al contrario de los mapas de bits, no dependen de los píxeles. Por lo tanto, tienen la mínima o máxima resolución que permita el formato en que se almacenen. Asimismo, al aumentar o reducir la resolución de un gráfico vectorial, tampoco se pierde definición en la imagen, porque la computadora solo tiene que redefinir las coordenadas y vectores que la imagen contenga para redimensionar los objetos. [5]

### 3.2.2 Colores

Aunque el color es una de las características más importantes que definen a los objetos, hasta hace poco tiempo se le prestaba poca atención debido al coste computacional y a la memoria necesaria para procesar imágenes de color.

Para obtener una imagen hay que transformar primero los parámetros cromáticos en eléctricos y representar los colores, lo cual puede hacerse de distintas maneras, dando lugar a los distintos espacios de color.

Un espacio de color es un método por el que se puede especificar, crear o visualizar cualquier color. Dependiendo del tipo de sensor y aplicación se han desarrollado diversos espacios de colores.

#### 3.2.2.1 Espacio RGB

El espacio RGB se basa en la combinación de tres señales de luminancia cromática distinta: rojo, verde y azul (Red, Green, Blue). La manera más intuitiva de conseguir un color concreto es determinar la cantidad de color rojo, verde y azul que se necesita combinar, para ello se realiza la suma aritmética de las componentes:  $X = R + G + B$ . [3]

Si tomamos en cuenta que cada una de las componentes tiene variaciones de tonalidad que van desde el 0 al 255 (28), entonces el cubo posee  $(28)^3 = 16,777,216$  colores representados vectorialmente; de esta manera podemos distinguir los extremos como el rojo (255,0,0), verde (0,255,0), azul (0,0,255), negro (0,0,0) y blanco (255,255,255).

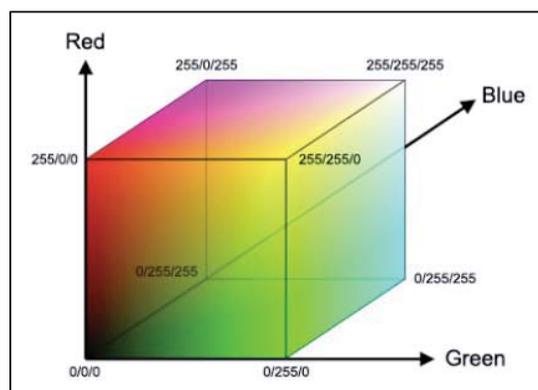


Figura 3.3 Representación tridimensional del espacio RGB.

### 3.2.2.2 Espacio HSI

El espacio de color HSI se basa en el modo de percibir los color que tenemos los humanos. Dicho sistema caracteriza el color en términos de tinte (Hue), saturación (Saturation) y brillo (Intensity).

El tono está asociado con la longitud de onda dominante en una mezcla de ondas luminosas. Así el tono representa el color dominante tal y como lo percibimos; cuando decimos que un objeto es rojo, verde o café, estamos indicando su tono en la Figura 3.4 (a) Tono.

La saturación se refiere a la cantidad de luz blanca mezclada con el color dominante. La saturación es un atributo que nos diferencia un color intenso de uno pálido. Cada uno de los colores primarios tiene su mayor valor de saturación antes de ser mezclados con otros. Así el azul cielos es muy claro (menos saturado), mientras que el azul marino es más opaco (más saturado). Otro ejemplo es el color rosa (rojo y blanco) que está menos saturado; mientras que el color rojo está totalmente saturado en la Figura 3.4 (b) Saturación.

La intensidad representa la iluminación percibida. Da la sensación de que algún objeto refleja más o menos luz. Este atributo lo podemos ver claramente en un televisor en blanco y negro, en la Figura 3.4 (c) Intensidad. [6]

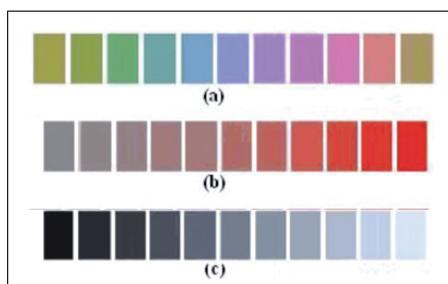


Figura 3.4 HSI: (a) Tono, (b) Saturación, (c) Intensidad. [6]

## 3.3 Preprocesamiento de Imágenes

El preprocesamiento de imágenes comprende aquellos algoritmos cuya finalidad es conseguir una mejora en la apariencia de la imagen original. Esta mejora en la apariencia consiste en resaltar determinadas características de la imagen o en eliminar aquello que las oculta. No se trata de mejorar la imagen de manera que quede lo más ideal posible, sino que pueda ser analizada de la forma más simple.

### 3.3.1 Manipulación de Contraste

El contraste muestra las variaciones locales del brillo, incrementa el cambio de luminosidad entre las zonas más oscuras o más claras de una fotografía, simulando a su vez, un mejor enfoque y claridad de imagen [1]. Su manipulación busca mejorar unas zonas en perjuicio de otras, y así distanciar más los píxeles con valores bajos o a la inversa. En una imagen en escala de grises, consiste en aumentar el rango dinámico de los niveles de gris de la imagen [2].

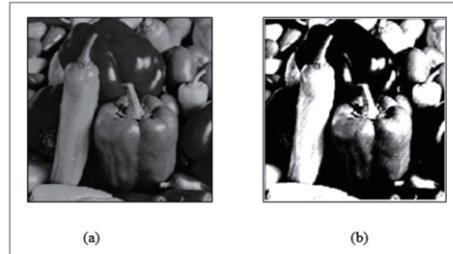


Figura 3.5 (a) Imagen Original, (b) Aumento de Contraste. [2]

### 3.3.2 Eliminación de Ruido

Todas las imágenes tienen una cierta cantidad de ruido, valores distorsionados, bien debidos al sensor de la cámara o al medio de transmisión de la señal. El ruido se manifiesta generalmente en píxeles aislados que toman un valor gris deferente al de sus vecinos. El ruido puede clasificarse en cuatro tipos:

- **Gaussiano:** Produce pequeñas variaciones en la imagen. Es debido, por ejemplo, a las diferentes ganancias en el sensor, ruido en los digitalizadores, perturbadores en la transmisión, etc.
- **Impulsional:** El valor del píxel no tiene relación con el valor ideal sino con el valor del ruido que toma valores muy altos o bajos. Se caracteriza entonces porque el píxel toma un valor máximo, causado por una saturación del sensor, o mínimo si se ha perdido la señal.
- **Frecuencial:** La imagen obtenida es la suma entre imagen ideal y otra señal, la interferencia, caracterizada por ser una senoide de frecuencia determinada.
- **Multiplicativo:** La imagen obtenida es la multiplicación de dos señales.



Figura 3.6 Diferencia entre imagen sin ruido y con ruido. [9]

### 3.3.3 Realce de Bordes

El realce de bordes transforma una imagen de manera que exhibe sólo el detalle de bordes o fronteras. Los bordes aparecen como las líneas de contorno de los objetos dentro de la imagen (Figura 3.7 (c) y Figura 3.7 (d)). Estos contornos pueden utilizarse en posteriores operaciones de análisis de imágenes para el reconocimiento de objetos o rasgos. [7]

El realce de bordes tiene un efecto opuesto a la eliminación de ruido, ya que de lo que se trata es de resaltar aquellos píxeles que representan un valor de gris distinto al de sus vecinos.

Por ello, si la imagen es ruidosa, el efecto de ruido se multiplicará; por lo que antes de resaltar los bordes, habrá que eliminar el ruido como se muestra en la Figura 3.7.

Los realces de bordes son implementados a través de filtros espaciales. Uno de los más usados es el método de desplazamiento y sustracción.

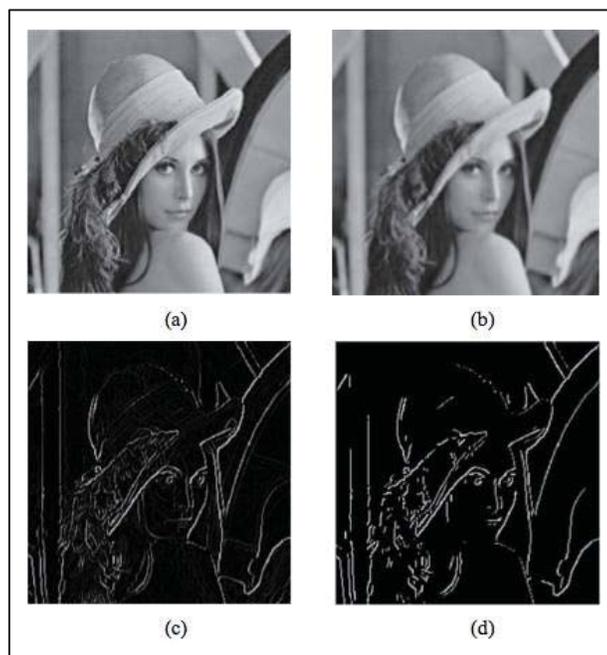


Figura 3.7 Realce de bordes, (a) imagen original, (b) imagen con filtro gaussiano para eliminar ruido, (c) realce de bordes sin filtro gaussiano, (d) realce de bordes con filtro gaussiano.

## 3.4 Extracción de Características

Una vez que la imagen ha sido procesada, se puede proceder a detectar los objetos presentes en ella; para ello se busca en la imagen cuáles son las características que definen al objeto, como los bordes, líneas, esquinas, texturas e incluso movimiento.

### 3.4.1 Detección de Bordes

Una de las características más útiles que se encuentran en una imagen, la constituyen los bordes, ya que se utilizan para definir los límites entre si y el fondo de la imagen. [3]

Los bordes de una imagen digital se pueden definir como transiciones entre dos regiones de niveles de gris significativamente distintos. Suministran una valiosa información sobre las fronteras de los objetos y puede ser utilizada para segmentar la imagen, reconocer objetos, etc. [11]

Ya hemos mencionado anteriormente que una de las técnicas para el realce de bordes es eliminando el ruido, con el fin de destacar sólo los bordes importantes.

Otra técnica para la ideal detección de bordes es binarizar la imagen, la binarización consiste en representar los píxeles, ya sea en color o escala de grises, en solo un bit (0 o 1), ya



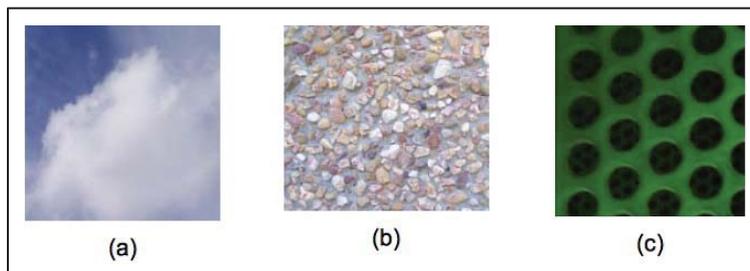
Figura 3.8 (a) Imagen Real, (b) Imagen Binarizada. [13]

sea completamente negro o completamente blanco; de esta manera los bordes quedarán completamente marcados.

### 3.4.2 Texturas

La textura es una característica importante utilizada en segmentación, identificación de objetos o regiones de interés en una imagen y obtención de forma. El uso de la textura para identificar una imagen proviene de la habilidad innata de los humanos para reconocer diferencias texturales. Por medio de la visión y el tacto, el ser humano es capaz de distinguir en forma intuitiva diversos tipos de textura.

La textura, por consiguiente, es una característica de difícil definición siendo la más extendida la siguiente: “Una textura está definida por la uniformidad, densidad, grosor, rugosidad, regularidad, intensidad y direccionalidad de medidas discretas del tono en los píxeles y de sus relaciones espaciales”. En la siguiente figura se muestran las diferencias entre una



textura suave (a) y otra rugosa (b), y entre una textura regular (c) de otra que no lo es (b).

Figura 3.9 Propiedades de textura: (a) Suavidad, (b) Rugosidad y (c) Regularidad

En una imagen digital, la textura depende de tres conceptos:

- La frecuencia de cambio de los tonos en los píxeles.
- La dirección o direcciones de cambio.
- El contraste entre un píxel y sus vecinos.

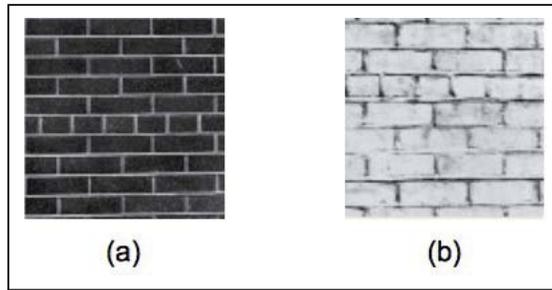


Figura 3.10 Independencia del tono o el color en la textura de una imagen digital.

Sin embargo, es independiente del tono o color de la imagen: en el ejemplo de la figura 3.10 se muestran dos imágenes de una misma textura. Aunque una sea en tonos claros y la otra en tonos oscuros, se observa que se trata del mismo tipo de textura [12].

### 3.4.3 Detección de Movimiento

Otro tipo de característica es la debida el movimiento. Puede ayudar a la segmentación ya que los puntos del mismo objeto presentan unas velocidades semejantes. Los distintos tipos de movimiento que existen son.

- Movimiento de la cámara.
- Movimiento de los objetos.
- Cambios en la iluminación.
- Cambios en la estructura, forma o tamaño del objeto.

El método más sencillo es aquel que se basa en las diferencias de imágenes. Aunque estos algoritmos son menos precisos, son mucho más simples y por lo tanto pueden procesar la imagen en poco tiempo, por lo que su utilidad es mayor. Los pasos a la hora de analizar el movimiento son tres:

- Determinar en qué zonas de la imagen existe algún objeto que esté moviéndose. En esta primera etapa no es importante la exactitud, sólo detectar las futuras zonas de interés en las que se realizará un análisis más detallado.
- En la segunda etapa, la atención sólo está en aquellas zonas donde se haya detectado actividad y se extrae información que pueda ser usada para el reconocimiento de objetos, análisis del movimiento, etc.
- La última etapa tiene el reconocimiento previo del problema en concreto. Así por ejemplo la información obtenida de un objeto pequeño que se mueve lentamente cerca de la cámara de otra igual forma, pero mayor que se mueve rápidamente pero lejos de ella es la misma [3].

## 3.5 Segmentación de Imágenes

La segmentación de imágenes divide la imagen en sus partes constituyentes hasta un nivel de subdivisión en el que se aíslan las regiones u objetos de interés. Los algoritmos de segmentación se basan en una de estas dos propiedades básicas de los valores del nivel de gris: discontinuidad o similitud entre los niveles de gris de píxeles vecinos [14].

- Similitud: Cada uno de los píxeles de un elemento tiene valores parecidos para alguna propiedad.

- Discontinuidad: Los objetos destacan del entorno y tienen, por tanto, unos bordes definidos.
- Conectividad: Los píxeles pertenecientes al mismo objeto tienen que ser contiguos, es decir, deben estar agrupados.

### 3.6 Máquinas de Aprendizaje (Machine Learning)

Machine Learning es un conjunto de técnicas de Inteligencia Artificial donde la inteligencia se obtiene a partir de ejemplos. Machine Learning se basa en dos pilares.

- El algoritmo de entrenamiento que se encarga de generar el modelo que utilizará el algoritmo de predicción.
- El algoritmo de predicción que toma el modelo generado por el algoritmo de entrenamiento para clasificar futuras entradas.

El siguiente ejemplo es un algoritmo de categorización que se encarga de dividir dos gestos realizados con la mano. Supongamos que tenemos un conjunto de datos de ejemplo que consiste en registros de longitud, anchura, ángulo y cantidad de dedos de la mano. Con estos datos podemos dibujar un gráfico 2D utilizando la longitud y anchura como ejes. La información del gesto determina el color de cada punto: negro para el primer gesto y blanco para el segundo gesto.

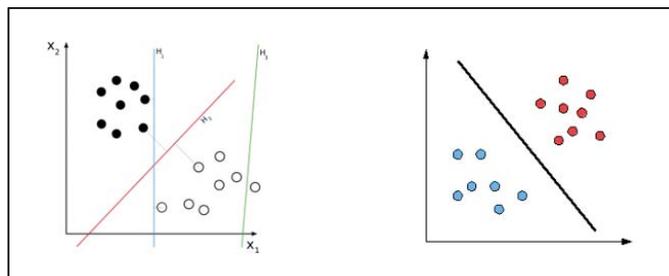


Figura 3.11 Representación Gráfica ML.

Machine Learning busca una línea muy particular que maximice la distancia entre dos ejemplos de cada clase que no tengan elementos entre ellos. Los ejemplos usados para definir esta línea se le conocen como vectores de soporte, de ahí su nombre. Como se ilustra en la imagen.

#### 3.6.1 Más allá de las Líneas

Cuando nuestro problema se puede separar usando líneas, decimos que es linealmente separable. Pero no todos los problemas lo son. En esos casos, ML contiene un 'truco' que es el de trasladar el problema a otra dimensión donde sí sea separable:

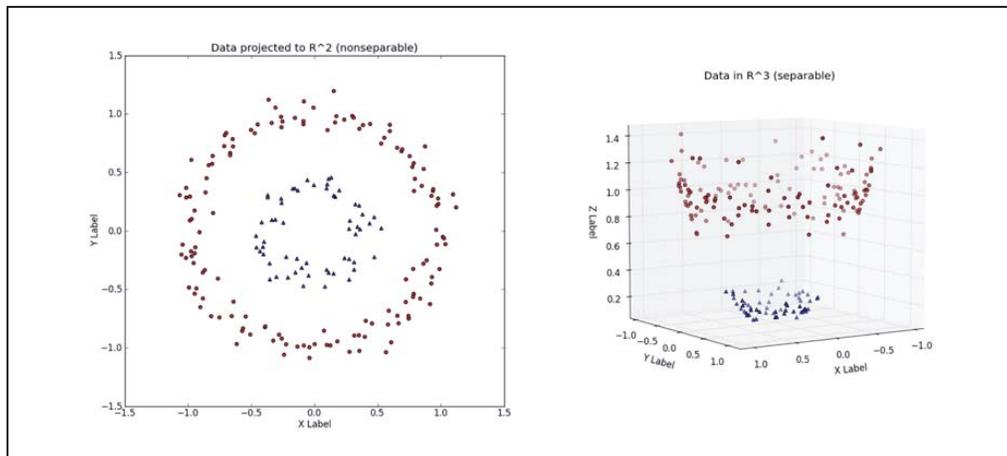


Figura 3.12 Representación Tridimensional ML.

En resumen, para resolver el problema ML se puede aumentar el número de dimensiones de nuestros ejemplos. Sin embargo, esto es literalmente un truco y ML lo maneja a través de la definición de un 'kernel' que transforma nuestros ejemplos. Algunos 'kernels' comunes son: Lineal y Polinómico. En nuestro caso, debemos diferenciar no dos, sino un conjunto mayor de señas, por lo que nuestro 'kernel' es polinómico, y es importante mencionar que son más de tres dimensiones donde ya no es posible graficar.

### 3.6.2 Machine Learning y OpenCV

OpenCV incluye una implementación de Machine Learning llamada Support Vector Machine (Máquina de Soporte Vectorial). SVM recibe parámetros, los cuales son ejemplos positivos y ejemplos negativos de los patrones a entrenar.

```

1  svm_params = dict( kernel_type = cv2.SVM_LINEAR,
2                    svm_type = cv2.SVM_C_SVC,
3                    C=2.67, gamma=5.383 )
4  svm = cv2.SVM()
5  svm.train_auto(trainData, responses, None, None, params=svm_params, k_fold=3)

```

Figura 3.13 Ejemplo código de SVM usando OpenCV .

La variable 'trainData' y response representa a los ejemplos (positivos y negativos), y la variable 'svm\_params' los parámetros de nuestra máquina. En este caso le pedimos que sea una máquina con un kernel lineal (SVM\_LINEAR), es decir no usamos el truco inherente a SVM y confiamos que caras de no caras sean linealmente separables. Le pedimos que use la implementación de OpenCV (SVM\_C\_SVC), y finalmente dos parámetros del algoritmo. Sin embargo, estos últimos parámetros no tendrán un efecto, ya que usamos la versión auto en donde el sistema adicional a calcular el modelo calcula estos parámetros.

Una vez que tenemos el modelo aprendido, podemos usarlo para predecir.

```
svm.predict(nuevo_ejemplo)
```

## **4. Análisis y Diseño**

### **4.1 Análisis Requerimientos del Proyecto**

Un requerimiento es una necesidad documentada sobre el contenido, forma o funcionalidad de un producto o servicio.

Los requerimientos son declaraciones que identifican atributos, capacidades, características y/o cualidades que necesita cumplir un sistema (o un sistema de software) para que tenga valor y utilidad para el usuario. En otras palabras, los requerimientos muestran qué elementos y funciones son necesarias para un proyecto.

#### **4.1.1 Requerimientos Funcionales**

- Implementación de algoritmo para la detección basado en imágenes.
- Implementación de algoritmo para el reconocimiento de patrones.
- Implementación de sistema de aprendizaje de máquina.
- Implementación de algoritmo de entrenamiento basado en coordenadas.
- Mostrar contorno, puntos y palma marcados en la cámara.
- Mostrar el número de dedos de 0 a 4.
- Guardar información de entrenamiento en el Smartphone.

#### **4.1.2 Requerimientos No Funcionales**

- Diseño de una interfaz sencilla y con alta usabilidad.
- Software optimizado, alto rendimiento, procesos con bajos requerimientos de hardware.
- Documentación detallada del código y sus modificaciones.

### **4.2 Diagrama de Casos de Uso**

Los casos de uso son una técnica para especificar el comportamiento de un sistema, “Un caso de uso es una secuencia de interacciones entre un sistema y alguien o algo que usa alguno de sus servicios.”

## 4.2.1 Caso de Uso General

El siguiente diagrama nos muestra el caso de uso general del sistema y como los actores se desenvuelven en él.

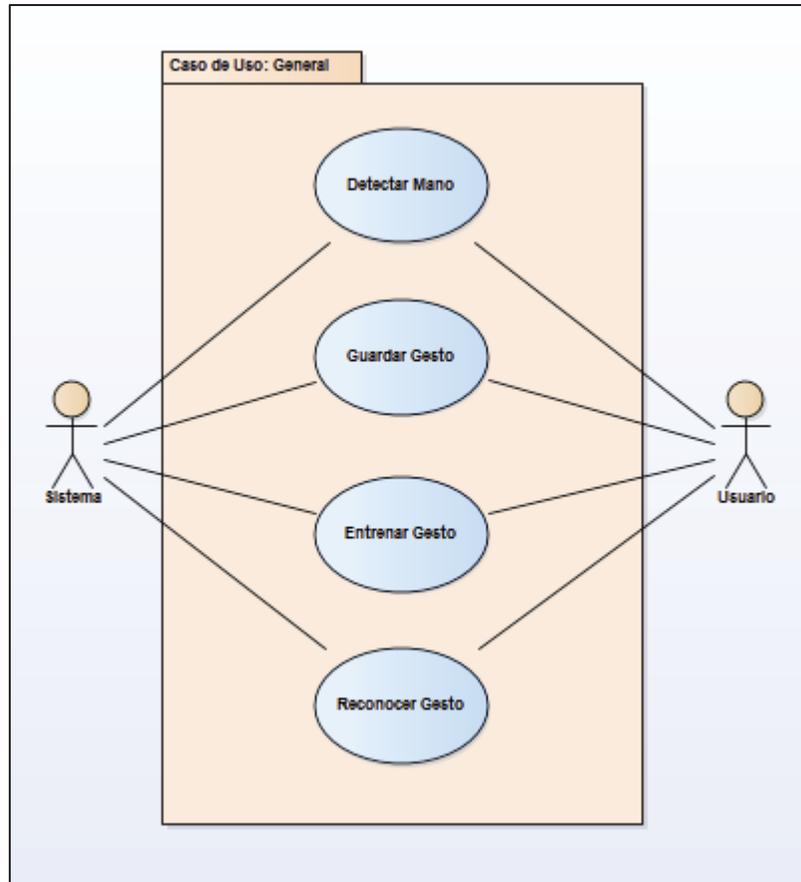


Figura 4.1 Diagrama de caso de uso general.

## 4.2.2 Casos de Uso Específicos

A continuación, se detalla cada una de las funcionalidades del Caso de Uso General, además de la especificación del Caso de Uso.

Tabla 4.1 Especificación de Caso de Uso: Detectar Mano.

<b>ESPECIFICACION DE CASO DE USO: DETECTAR MANO.</b>	
<b>Caso de Uso</b>	Detectar Mano.
<b>Actores</b>	Usuario, Sistema
<b>Objetivo</b>	Detectar la mano de manera precisa para posteriormente reconocer gestos.
<b>Pre Condición</b>	<ul style="list-style-type: none"> <li>- El usuario debe ingresar a la aplicación.</li> <li>- El usuario debe enfocar la mano del emisor de seña.</li> </ul>
<b>Alcance</b>	Desde que el usuario ingresa a la aplicación hasta que el usuario finalice el proceso de detección.
<b>CURSO NORMAL DE LOS EVENTOS</b>	
	<ol style="list-style-type: none"> <li>1. El usuario ingresa a la aplicación.</li> <li>2. El sistema inicia la cámara.</li> <li>3. El usuario selecciona resolución.</li> </ol>

4. El sistema configura la resolución seleccionada por el usuario.
5. El usuario enfoca el fondo a contrastar con la mano.
6. El sistema detecta el fondo.
7. El usuario enfoca la mano.
8. El sistema detecta la mano.
9. El sistema debe binarizar la imagen.
10. El sistema extrae características de la mano.
11. El usuario realiza o no un gesto.
12. El usuario guarda o no un gesto.

#### 4.2.2.1 Caso de Uso Detectar Mano

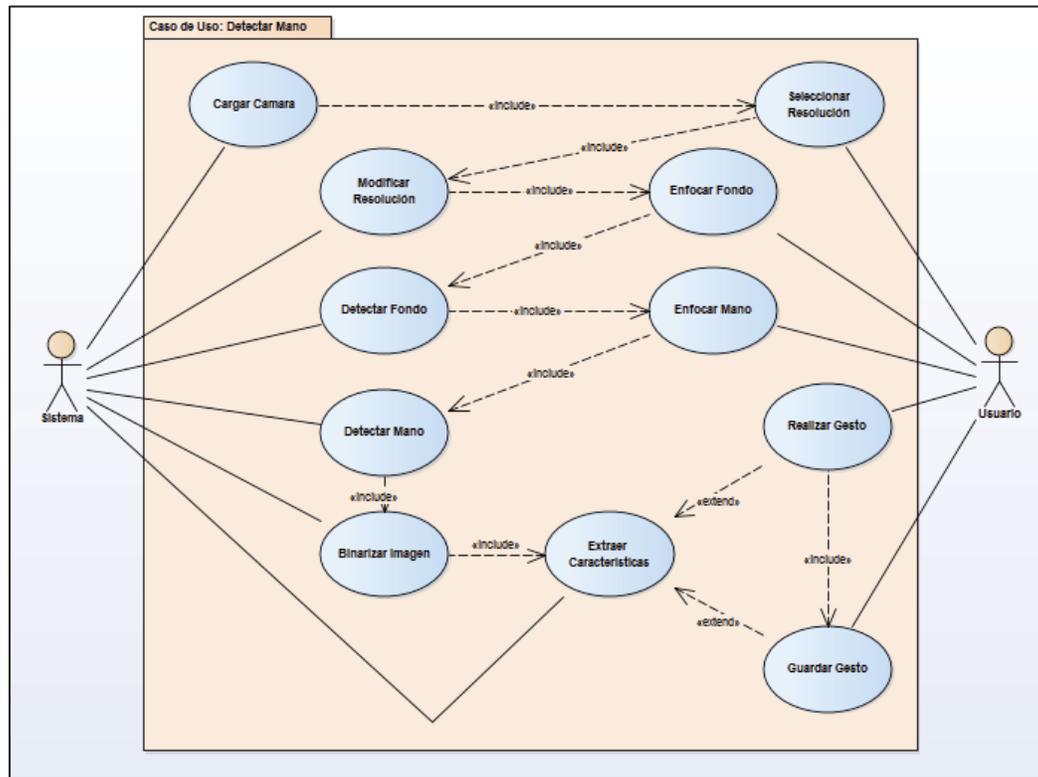


Figura 4.2 Diagrama de Caso de Uso Detectar Mano.

#### 4.2.3 Caso de Uso Guardar Gesto

Tabla 4.2 Especificación de Caso de Uso: Guardar Gesto.

ESPECIFICACION DE CASO DE USO: GUARDAR GESTO.	
<b>Caso de Uso</b>	Guardar Gesto
<b>Actores</b>	Sistema, Usuario
<b>Objetivo</b>	Guardar gesto indicado por el usuario para su posterior reconocimiento.
<b>Pre Condición</b>	<ul style="list-style-type: none"> <li>- El usuario debe ingresar a la aplicación.</li> <li>- El usuario debe enfocar con la cámara la mano del usuario.</li> </ul>
<b>Alcance</b>	Desde que el usuario ingresa a la aplicación realizando un seleccionar guardar gesto, entrenando o no, para su reconocimiento.
CURSO NORMAL DE LOS EVENTOS	
	<ol style="list-style-type: none"> <li>1. El usuario ingresa a la aplicación.</li> <li>2. El sistema enfoca la mano.</li> <li>3. El sistema guarda la imagen detectada de la mano.</li> </ol>

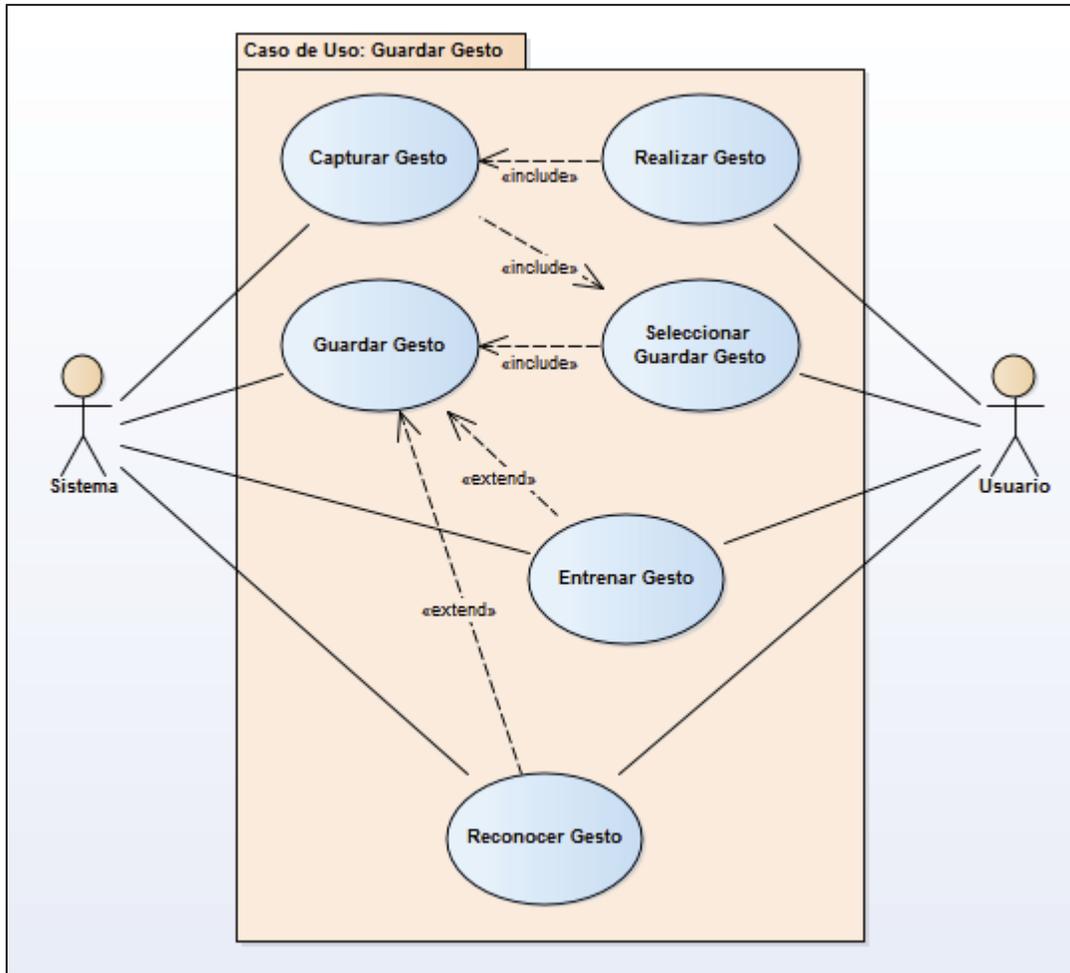


Figura 4.3 Diagrama de Caso de Uso Guardar Gesto.

#### 4.2.4 Caso de Uso Entrenar Gesto

Tabla 4.3 Especificación de Caso de Uso: Entrenar Gesto

ESPECIFICACION DE CASO DE USO: ENTRENAR GESTO	
<b>Caso de Uso</b>	Entrenar Gesto
<b>Actores</b>	Sistema, Usuario.
<b>Objetivo</b>	Entrenar cada tipo de seña para un mejor reconocimiento.
<b>Pre Condición</b>	<ul style="list-style-type: none"> <li>- El usuario debe ingresar a la aplicación.</li> <li>- El usuario debe enfocar con la cámara la mano.</li> <li>- El usuario realiza gesto.</li> </ul>
<b>Alcance</b>	Desde que el usuario ingresa a la aplicación configurando la resolución de la cámara de OpenCV.
<b>CURSO NORMAL DE LOS EVENTOS</b>	
<ol style="list-style-type: none"> <li>1. El usuario ingresa a la aplicación.</li> <li>2. El sistema enfoca la mano.</li> <li>3. El sistema detecta los puntos.</li> <li>4. El sistema guarda coordenadas de entrenamiento.</li> </ol>	

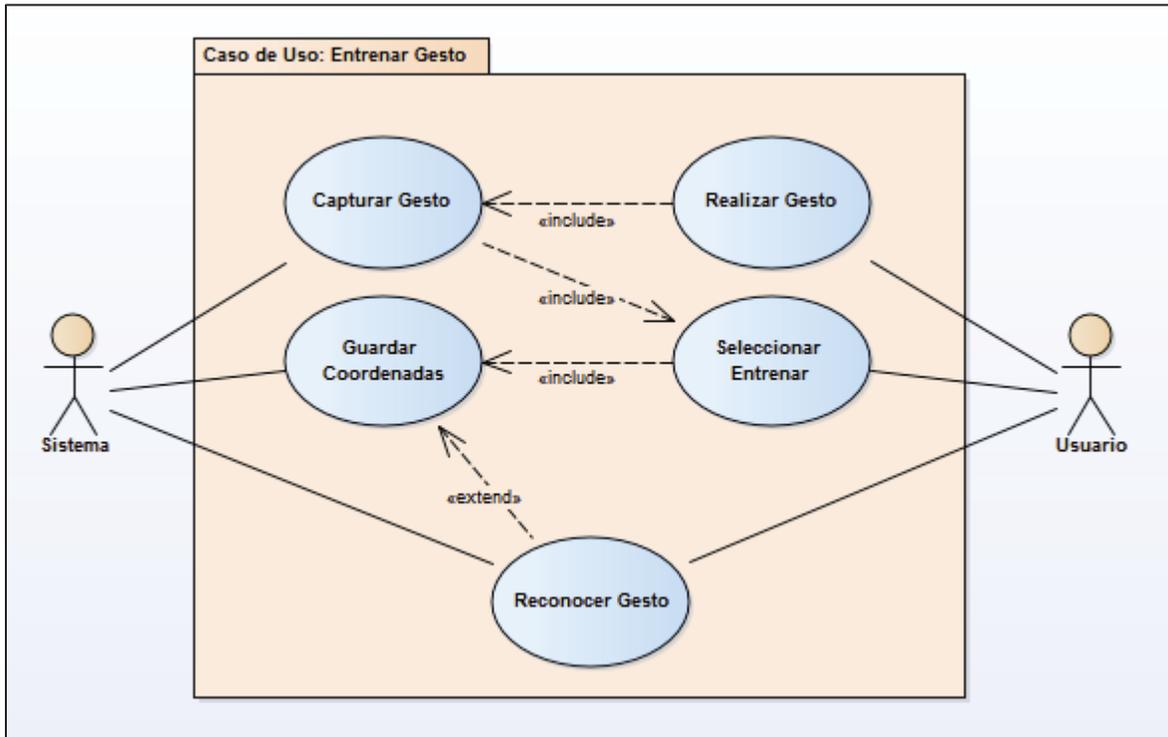


Figura 4.4 Diagrama de Caso de Uso Entrenar Gesto.

## 4.2.5 Caso de Uso Reconocer Gesto

Tabla 4.4 Especificación de Caso de Uso: Reconocer Gesto

ESPECIFICACION DE CASO DE USO: RECONOCER GESTO	
<b>Caso de Uso</b>	Reconocer Gesto
<b>Actores</b>	Sistema, Usuario.
<b>Objetivo</b>	Que el sistema sea capaz de reconocer los gestos indicados por el usuario, luego de haber realizado todo el procesamiento de imagen.
<b>Pre Condición</b>	<ul style="list-style-type: none"> <li>- El usuario debe ingresar a la aplicación.</li> <li>- El usuario debe enfocar con la cámara la mano.</li> <li>- El usuario debe mostrar la mano.</li> <li>- El usuario realiza gesto.</li> </ul>
<b>Alcance</b>	Desde que el usuario ingresa a la aplicación configurando o no el preprocesamiento de la imagen hasta que el sistema enfoque la mano detectando separaciones y calcule la cantidad de dedos enfocados.
<b>CURSO NORMAL DE LOS EVENTOS</b>	
<ol style="list-style-type: none"> <li>1. El usuario ingresa a la aplicación.</li> <li>2. El sistema enfoca la mano.</li> <li>3. El sistema detecta los puntos.</li> <li>4. Reconoce el gesto indicado por el usuario.</li> </ol>	

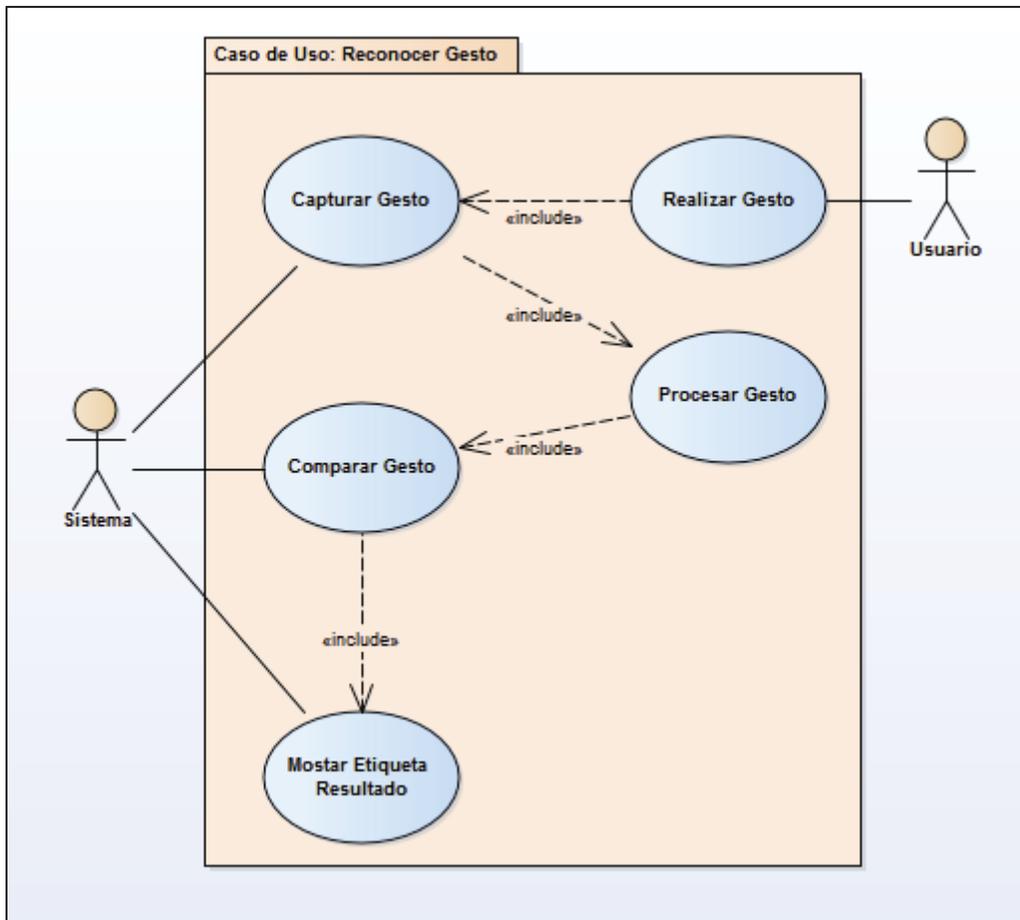


Figura 4.5 Diagrama de Caso de Uso Reconocer Gesto.

## 5.Desarrollo de Sistema

Para lograr un óptimo reconocimiento de señas, se realizaron diferentes técnicas. Los algoritmos posteriormente utilizados corresponden a la librería OpenCV, tanto los algoritmos de pre procesamiento, detección, reconocimiento y entrenamiento.

### 5.1 Arquitectura para el Reconocimiento en Android

Se demuestra en el siguiente diagrama la arquitectura utilizada para implementar el reconocimiento en el dispositivo móvil Android con SVM.

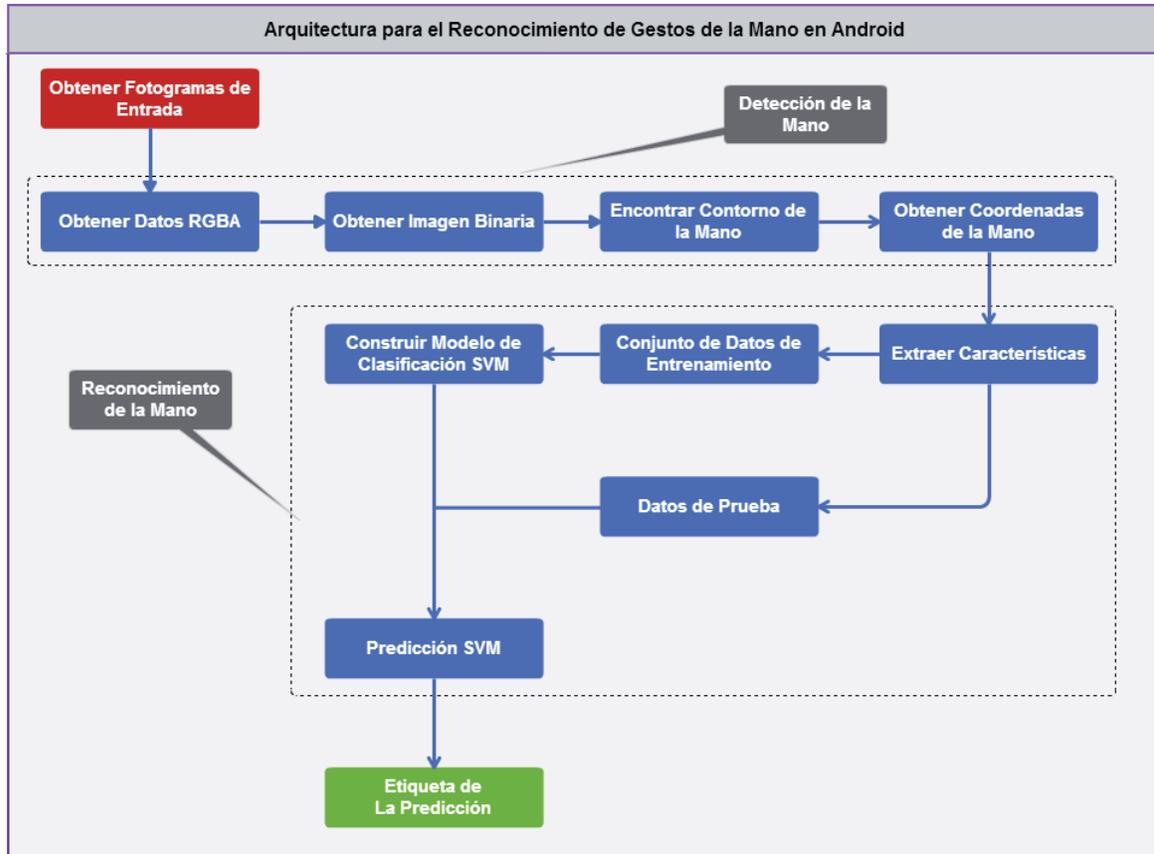


Figura 5.1 Arquitectura para el Reconocimiento de la mano en Android.

## 5.2 Detección de la Mano

### 5.2.1 Detección del Fondo y Mano por Separados

El principal problema del reconocimiento con cámara digital se presenta en el fondo irregular de la imagen. Una solución al problema es detectar el fondo antes de enfocar la mano, así al detectar la mano, se calcula la diferencia entre el fondo y el fondo con la mano, dando como resultado la mano sin fondo.

Este método en teoría es eficiente, pero hay factores que influyen y pueden afectar el resultado; entre ellos están la luz, el movimiento de la cámara y el movimiento de la mano.

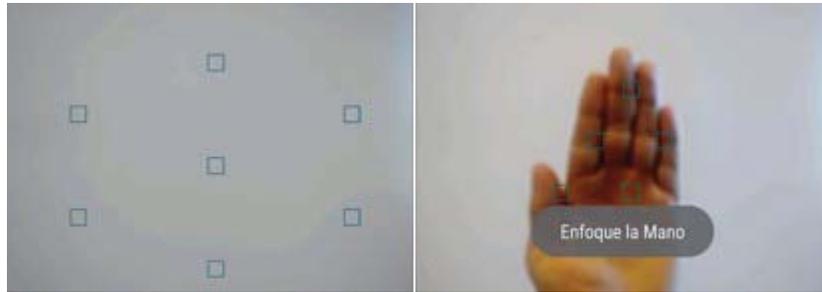


Figura 5.2 Detección de Fondo y Detección de la Mano.

### 5.2.2 Imagen Binaria

La imagen binaria es una ayuda para el usuario más que para el sistema, ya que muestra la diferencia que existe entre el fondo y la mano.



Figura 5.3 Imagen Binarizada

### 5.2.3 Detección de Contornos y Estructura de la Mano

La detección de cada elemento que compone la mano (palma, dedos, contorno) es esencial para posteriormente guardar y comparar coordenadas. Además, permite que el usuario visualice el comportamiento del detector de la mano.

### 5.2.4 Enumeración de los dedos de la mano

La enumeración de la mano se basa en el punto anterior, enumera los dedos de 0 a 4. Es la principal base para el entrenamiento de coordenadas de los dedos.

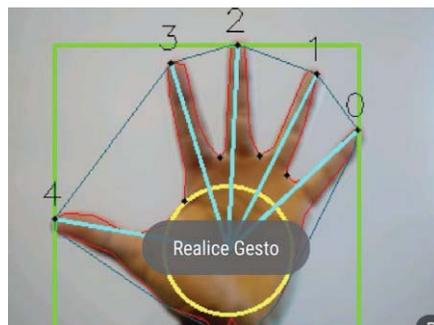


Figura 5.4 Mano detectada con extracción de características, enumerada de 0 a 4.

### 5.3 Entrenamiento de Gestos

El proceso de entrenamiento es bastante sencillo desde el punto de vista del usuario, sólo debe realizar el gesto, seleccionar “Guardar Gesto” y luego seleccionar “Entrenar”. Sin embargo, el proceso algorítmico es un poco complejo. Al seleccionar “Guardar Gesto”, el sistema guarda la imagen en la carpeta “Lense”. Al seleccionar “Entrenar”, el sistema guarda las coordenadas de la mano en el archivo “entrenamiento.txt” dentro de la carpeta “Lense”.



Figura 5.5 Entrenamiento Letra B.

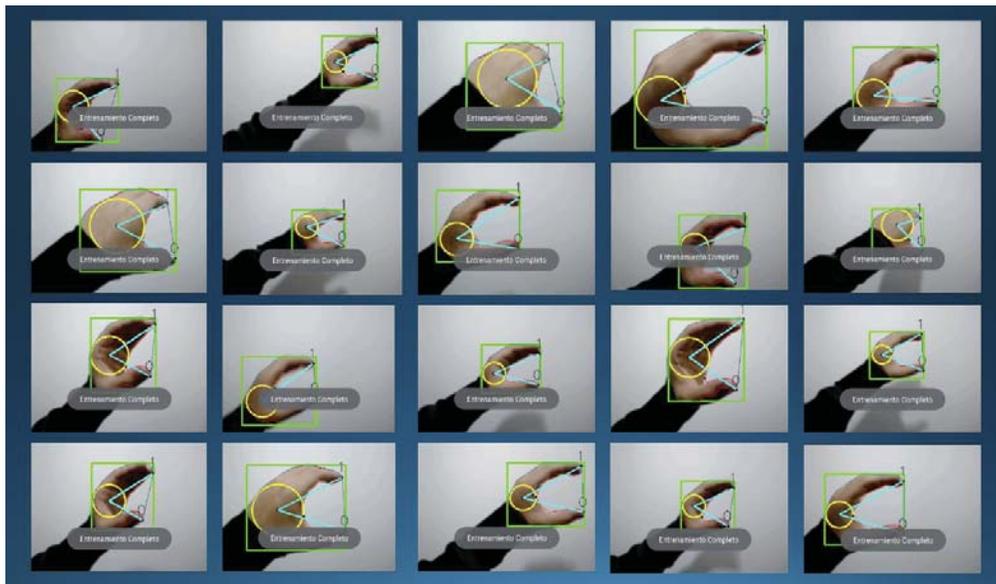


Figura 5.6 Entrenamiento Letra C.

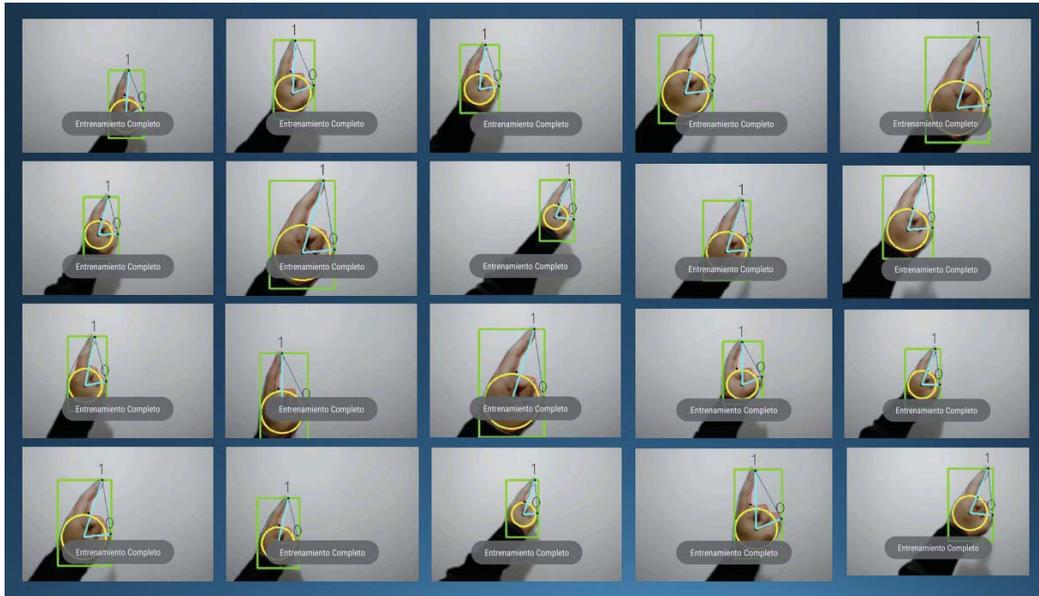


Figura 5.7 Entrenamiento Letra D.

## 5.4 Reconocimiento de Gestos

El reconocimiento se basa principalmente en las coordenadas, pero utiliza las imágenes para comparar colores. Los datos del archivo “entrenamiento.txt” son separados en distintas clases por el algoritmo SVM de OpenCV.

## 5.5 Implementación

### 5.5.1 Android

Android es el sistema operativo utilizado en más de mil millones de teléfonos inteligentes y tablets. Estos dispositivos nos endulzan la vida, por eso decidimos poner a cada versión de Android el nombre de un dulce. Cada versión de Android hace que algo nuevo sea posible, como obtener direcciones o cortar en rodajas una fruta virtual.

#### 5.5.1.1 Sistema Android

Android 6.0 Marshmallow, Android llega a las pantallas grandes y pequeñas – desde teléfonos y tablets a relojes, televisores y automóviles. Marshmallow tiene un estilo visual atrevido y una respuesta táctil fluida de diseño de materiales.

- **Diseño de Materiales:** Consigue un aspecto totalmente nuevo y la sensación de que hace que sea más fácil de navegar el dispositivo. Basado en sombras y movimiento, diseño de material une principios clásicos de buen diseño con la invocación de lo que es posible a través de la tecnología.

#### 5.5.2 OpenCV para Android

OpenCV (Open Source Computer Vision Library: <http://opencv.org>) es una biblioteca BSD-licencia de código abierto que incluye varios cientos de algoritmos de visión por

computador. El documento describe el llamado OpenCV API 2.x, que es esencialmente un C++ API, como opuesta a la API 1.x OpenCV C-basado.

OpenCV tiene una estructura modular, lo que significa que el paquete incluye varias bibliotecas compartidas o estáticas. Los siguientes módulos están disponibles:

- Core funcionalidad - un módulo compacto que define las estructuras de datos básicos, incluyendo la densa Mat matriz multi-dimensional y funciones básicas utilizadas por todos los demás módulos.
- El procesamiento de imágenes - un módulo de procesamiento de imagen que incluye la imagen no lineal y filtrado, transformaciones geométricas imagen (redimensionar, afín y la perspectiva de deformación, genérica Reasignación basada en tablas), conversión de espacio de color, histogramas, y así sucesivamente.
- vídeo - un módulo de análisis de vídeo que incluye la estimación de movimiento, sustracción de fondo, y los algoritmos de seguimiento de objetos.
- calib3d - algoritmos básicos de múltiples vistas geometría, la calibración individual y equipo de música de cámara, objeto plantean estimación, algoritmos de correspondencia estéreo y elementos de reconstrucción 3D.
- features2d - detectores de rasgos sobresalientes, descriptores y comparadores descriptores.
- objdetect - detección de objetos e instancias de las clases predefinidas (por ejemplo, caras, ojos, las tazas, las personas, vehículos, etc.).
- highgui - una interfaz fácil de usar para capacidades simples de interfaz de usuario.
- videoio - una interfaz fácil de usar para la captura de vídeo y codecs de vídeo.
- gpu - algoritmos acelerados por GPU de diferentes módulos OpenCV.
- ... Algunos otros módulos auxiliares, tales como Flann y Google envoltorios de prueba, enlaces Python y otros.

### 5.5.2.1 OpenCV Manager

Biblioteca OpenCV es utilizado por otras aplicaciones para la mejora de imagen, composición panorámica, detección de objetos, reconocimiento, etc. OpenCV Manager proporciona la mejor versión de la OpenCV para su hardware. También recibe las últimas actualizaciones de estabilidad y rendimiento para la biblioteca.

## 5.6 Visualización

En la sección de visualización se pueden observar capturas de pantalla de la aplicación desarrollada:

En la figura 5.8 se observa el logo de la aplicación, y su menú principal, pudiendo elegir entre las opciones de Traducir, Mostrar Abecedario, Reconocer Señas:

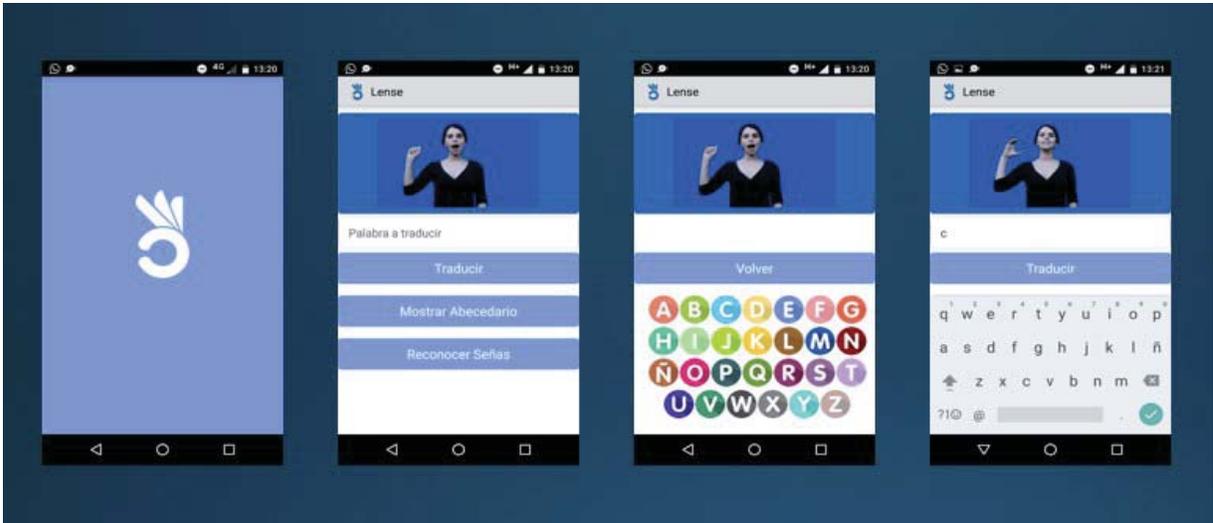


Figura 5.8 Inicio de la aplicación.

Al seleccionar la opción Reconocer Señas como se muestra en la En la figura 5.8, se ingresa a la interfaz principal de reconocimiento de señas:

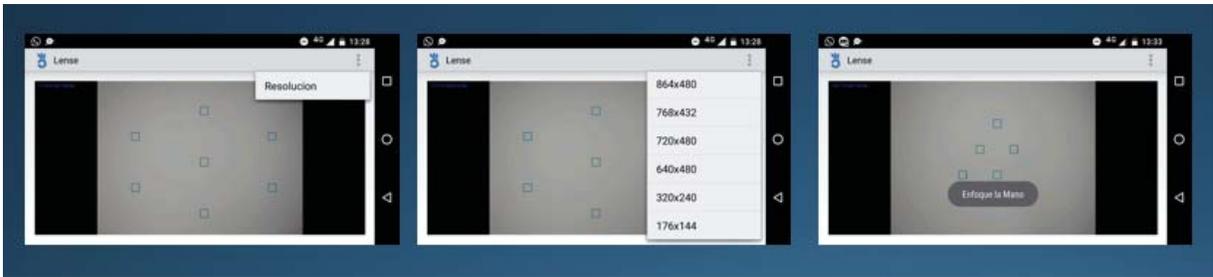


Figura 5.9 Interfaz de Reconocimiento.

El usuario debe elegir la resolución adecuada según las capacidades del dispositivo a utilizar,

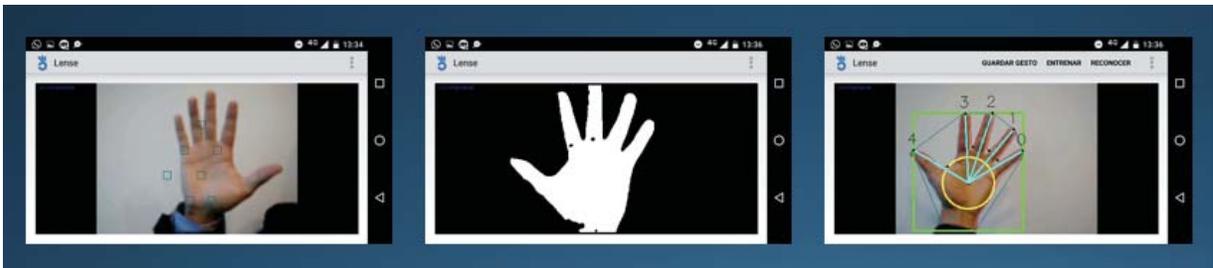


Figura 5.10 Captar Fondo, binarizar imagen captada y Obtener Características.

en la figura 5.10 el usuario es capaz de obtener la forma de la mano y posteriormente obtener características para ir al reconocimiento.

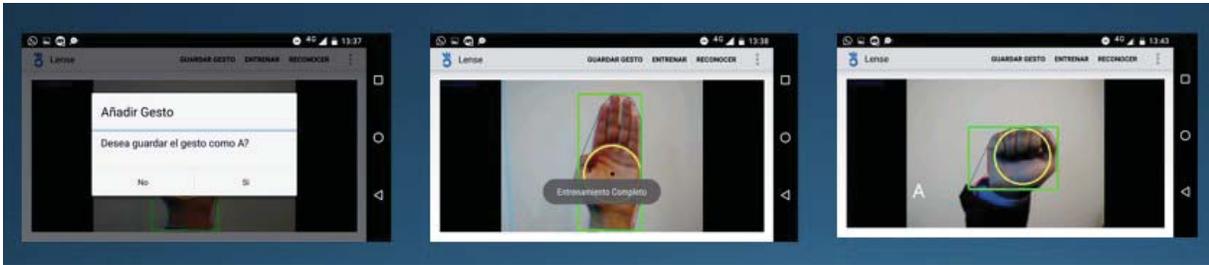


Figura 5.11 Reconocimiento de gesto almacenado y entrenado.

En la figura 5.11 el usuario decide añadir un gesto al sistema para luego ser entrenado y seguidamente reconocer el gesto con la etiqueta A.

## 6. Pruebas y Resultados

### 6.1 Capturas Realizadas

En los siguientes casos de estudio, se realizaron pruebas a las primeras 4 letras del alfabeto con resolución de imagen 320x240 de la cámara proporcionada por OpenCV.

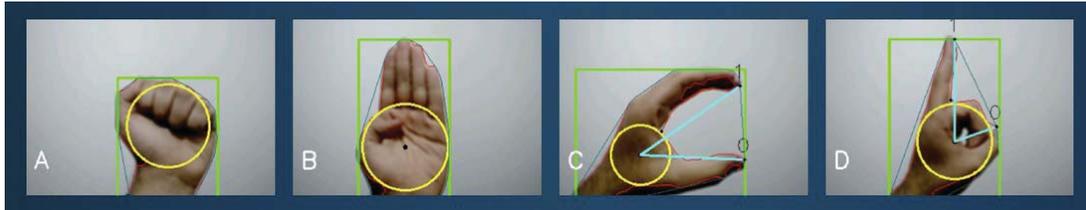


Figura 6.1 Letras del Alfabeto A - B - C - D.

También hubo casos en los que los resultados no fueron los esperados, por sobre todo en el ambiente no controlado en donde se provocó la gran parte de los falsos positivos, pero en ambiente semi-controlado bajo luz normal este también tendió a entregar falsos resultados como se demuestra en la figura 6.2.

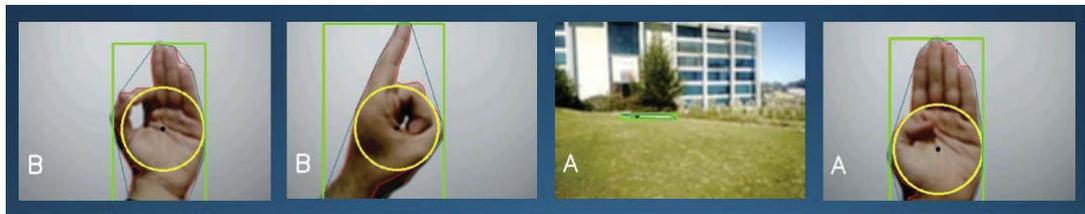


Figura 6.2 Falsos Positivos.

#### 6.1.1 Resultados Caso de Estudio N°1 Ambiente Controlado

A continuación se muestran los resultados en la tabla 6.1 obtenidos sobre un fondo blanco y textura plana, con alta luz natural y artificial, y sin sombra visible:

Tabla 6.1 Estudio N°1, 35 Pruebas en Ambiente Controlado.

LETRA	Nº PRUEBAS	ACIERTO	FALSOS POSITIVOS	FALSOS NEGATIVOS
A	35	100%	0%	0%
B	35	100%	0%	0%
C	35	100%	0%	0%
D	35	100%	0%	0%

#### 6.1.2 Resultados Caso de Estudio N°2 Ambiente no Controlado

En la siguiente tabla se muestran los datos obtenidos sobre un fondo de diversos colores, profundidades y texturas, bajo luz natural y sin luz artificial, y con mucha sombra visible:

Tabla 6.2 Estudio N°2, 35 Pruebas en Ambiente no Controlado.

LETRA	N° PRUEBAS	ACIERTO	FALSOS POSITIVOS	FALSOS NEGATIVOS
<b>A</b>	35	80%	20%	0%
<b>B</b>	35	76%	24%	0%
<b>C</b>	35	87%	13%	0%
<b>D</b>	35	77%	23%	0%

### 6.1.3 Resultados Caso de Estudio N°3 Ambiente bajo luz artificial

Los datos obtenidos representan los errores y aciertos obtenidos sobre un fondo blanco con luz artificial, la cual genera mucha sombra y variados tonos de color en el fondo:

Tabla 6.3 Estudio N°1, 35 Pruebas en Ambiente bajo luz artificial.

LETRA	N° PRUEBAS	ACIERTO	FALSOS POSITIVOS	FALSOS NEGATIVOS
<b>A</b>	35	90%	10%	0%
<b>B</b>	35	86%	14%	0%
<b>C</b>	35	97%	3%	0%
<b>D</b>	35	87%	13%	0%

Por ende podemos concluir que para obtener los mejores resultados, es necesario ubicar la mano en un lugar perfectamente iluminado logrando un 100% de reconocimiento efectivo sin dar paso a falsos positivos, además tras la realización de todas las pruebas es necesario por lo menos entrenar más de 30 veces el mismo gesto.

## 7. Conclusión

La lengua de señas es una lengua que todos deberíamos dominar; sin embargo, hay ciertas limitaciones que desincentivan el aprendizaje de esta; una de las limitaciones es que la lengua varía de zona en zona, por lo que la comunicación se verá acomplejada, otra limitación es que no es una lengua requerida ni necesaria para la comunicación de la mayoría de las personas.

En otras palabras, la mayor parte de la población mundial no tiene gran interés en aprender la lengua ya que no necesita de ella para comunicarse, y es la minoría de la población la que necesita de esta lengua, pero que de todas formas sigue siendo un porcentaje considerable.

Es difícil identificar a una persona sorda, ya que estas personas se sienten intimidadas por el resto de la población, evitando el contacto y con mayor razón aún, la comunicación. Lense® será una herramienta de incentivo al aprendizaje de esta lengua, una manera de sensibilizar al usuario.

Al momento de reconocer señas se logró identificar las distintas limitaciones, como por ejemplo la luz, ya que la cámara detecta frecuencias de luz, y entre más altas sean estas frecuencias, más fácil será detectar colores, diferencias, y por lo tanto extraer características. Otra de las limitaciones es la sombra que genera la luz, ya que la sombra genera variaciones de colores, distorsiona la realidad y por lo tanto confunde a la cámara.

Si bien se lograron los objetivos propuestos en el presente proyecto, aún falta mucho por investigar, descubrir y crear en el mundo de la detección de características y la inteligencia artificial, partiendo por la detección en un ambiente no controlado. Se han comprobado las deficiencias de la cámara digital, en comparación a otras tecnologías de detección como Kinect; la baja luz afecta radicalmente los resultados, más aún en un ambiente nocturno (sin luz natural).

Lense® continuará optimizando sus procesos y técnicas de reconocimiento, técnicas de entrenamiento y, sobretodo, las técnicas de detección en ambientes no controlados, es decir, ambientes con poca luz, movimiento y con altos casos de sombra, hasta lograr un reconocimiento independiente de factores que, hoy en día, afectan en su totalidad el mundo de las cámaras digitales; y así, lograr nuestro objetivo principal, promover la enseñanza de la lengua de señas y la inclusión.

## 8. Referencias Bibliográficas

- [1] Digital Fotored [en línea]: “Contraste” [Consulta: 15 Mayo. 2016] (<http://www.digitalfotored.com/imagendigital/contraste.htm>).
- [2] Pedro Peguero [en línea]: “Realce y Restauración de imagen” [Consulta: 25 Mayo. 2016] ([http://www.lpi.tel.uva.es/muitic/pim/docus/Realce\\_y\\_restauracion.pdf](http://www.lpi.tel.uva.es/muitic/pim/docus/Realce_y_restauracion.pdf)).
- [3] “Visión por Computador”, Gonzalo Pajares Martinsanz, Jesús Manuel de la Cruz García, 2001.
- [4] Isabel [en línea]: “Cómo funciona la lengua de señas” [Consulta: 04 Mayo. 2016] (<http://www.batanga.com/curiosidades/4910/como-funciona-la-lengua-de-senas>).
- [5] Cristian [en línea]: “Formatos de Imagen Digital” [Consulta: 04 Junio. 2016] ([http://www.revista.unam.mx/vol.6/num5/art50/may\\_art50.pdf](http://www.revista.unam.mx/vol.6/num5/art50/may_art50.pdf)).
- [6] J.J. Baez Rojas y M.A. Alonso Pérez [en línea]: “Espacio HSI” [Consulta: 23 Junio. 2016] (<http://www.scielo.org.mx/pdf/rmfe/v54n2/v54n2a11.pdf>).
- [7] B. Aldalur1 y M. Santamaría [en línea]: “Realce de imágenes: filtrado espacial” [Consulta: 09 Mayo. 2016] (<http://www.aet.org.es/revistas/revista17/AET17-04.pdf>).
- [8] “Reconocimiento gestual mediante técnicas avanzadas de visión por computador” (<ftp://tesis.bbt.ull.es/ccppytec/cp346.pdf>).
- [9] Guías Prácticas [en línea]: “Ruido en la imagen digital” [Consulta: 18 Junio. 2016] (<http://www.guiaspracticas.com/camaras-de-fotos/ruido-en-la-imagen-digital>).
- [10] Manos que hablan [en línea]: “Alfabeto Lengua de Señas Chilenas” [Consulta: 19 Junio. 2016] (<http://manosquehablan.com.ar/alfabetos/chile/>).
- [11] Universidad de Jaen [en línea]: “Detección de bordes en una imagen” [Consulta: 18 Abril. 2016] ([http://www4.ujaen.es/~satorres/practicas/practica3\\_vc.pdf](http://www4.ujaen.es/~satorres/practicas/practica3_vc.pdf)).
- [12] Escuela Superior de Ingenieros [en línea]: “Concepto de Textura” [Consulta: 25 Abril. 2016] (<http://bibing.us.es/proyectos/abreproy/11494/fichero/PROYECTO%252FCapitulo+3.pdf>).
- [13] [en línea]: “Binarizar Imágenes” ([http://www.scielo.org.co/scielo.php?script=sci\\_arttext&pid=S169233242012000200014](http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S169233242012000200014)).
- [14] [en línea]: “Segmentación de imágenes” (<http://alojamientos.us.es/gtocoma/pid/tema4.pdf>).
- [15] Aforgetnet [en línea]: “Descripción AForget.NET” [Consulta: 12 Noviembre. 2015] (<http://www.aforgetnet.com/framework/docs/>).
- [16] Wikipedia [en línea]: “Descripción LeapMotion” [Consulta: 12 Noviembre. 2015] ([https://en.wikipedia.org/wiki/Leap\\_Motion](https://en.wikipedia.org/wiki/Leap_Motion)).
- [17] OpenArch[en línea]: “Descripción Openarch” [Consulta: 12 Noviembre. 2015] (<http://www.openarch.cc/es>).
- [18] Wisee [en línea]: “Descripción Wisee” [Consulta: 12 Noviembre. 2015] (<http://wisee.cs.washington.edu/>).

- [19] Microsoft [en línea]: “Descripción Kinect” [Consulta: 12 Noviembre. 2015] (<https://dev.windows.com/en-us/kinect/tools>).
- [20] OpenCV [en línea]: “Descripción OpenCV” [Consulta: 12 Noviembre. 2015] (<http://opencv.org/>).
- [21] “Hand modeling, analysis and recognition”, Ying Wu and T S Huang, 2001.
- [22] “Gesture Recognition: A Survey, Systems, Man, and Cybernetics, Part C: Applications and Reviews”, S. Mitra and T. Acharya, 2007.
- [23] “Vision-based human motion analysis: An overview, Computer Vision and Image Understanding” Ronald Poppe, 2007.
- [24] “A comparative study of two state-of-the-art sequence processing techniques for hand gesture recognition, Computer Vision and Image Understanding”, Agnes Just and Sebastien Marcel, 2009.
- [25] “A Visual System for Hand Gesture Recognition in Human-Computer Interaction, Image Analysis”, Matti-Antero Okkonen, Vili Kellokumpu, Matti Pietikainen, and Janne Heikkila, 2007.
- [26] “Visual interpretation of hand gestures for Human-computer interaction”, V. I. Pavlovic, R. Sharma, and T. S. Huang, 1997.
- [27] “Gesture Recognition Technology”, Pallavi Halarnkar, Sahil Shah, Harsh Shah, HARDIK Shah, and Jay Shah 2012.
- [28] “A feature-based approach to continuous-gesture analysis”, Alan Daniel Wexelblat, 1994.
- [29] “A taxonomy of gestures in human computer interactions”, Maria Karam and M. C. Schraefel, 2005.
- [30] “A survey of advances in vision-based human motion capture and analysis, Computer Vision and Image Understanding”, Thomas B Moeslund, Adrian Hilton, and Volker Kruger, 2006.
- [31] “A real time hand gesture recognition system using motion history image”, Chen-Chiung Hsieh, Dung-Hua Liou, and D Lee, 2010.
- [32] “Características Smartphone Galaxy S6 Edge” (<http://www.samsung.com/es/consumer/mobile-devices/smartphones/galaxy-s/SM-G925FZDFPHE>)