UNIVERSIDAD DE LAS FUERZAS ARMADAS

INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y TELECOMUNICACIONES

CARRERA DE INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL

TRABAJO DE TITULACIÓN, PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERA EN ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL

TEMA: "RECONOCIMIENTO DE LENGUA DE SEÑAS ECUATORIANO MEDIANTE SVM USANDO CARACTERÍSTICAS DE PROFUNDIDAD Y COLOR"

AUTORA: ACURIO NOROÑA, LISSETTE ESTEFANÍA

DIRECTOR: MSc. LARCO BRAVO, JULIO CESAR

SANGOLQUÍ

2019

DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y TELECOMUNICACIONES

CARRERA DE INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL

AUTORÍA DE RESPONSABILIDAD

Yo, LISSETTE ESTEFANÍA ACURIO NOROÑA, declaro que el contenido, ideas y criterios del trabajo de titulación: "RECONOCIMIENTO DE LENGUA DE SEÑAS ECUATORIANO MEDIANTE SVM USANDO CARACTERÍSTICAS DE PROFUNDIDAD Y COLOR" es de mi autoría y responsabilidad, cumpliendo con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de la Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Consecuentemente, el contenido de la investigación mencionada es veraz.

Sangolquí, junio del 2019

Lissette Estefanía Acurio Noroña
C.I.: 1804351060

DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y TELECOMUNICACIONES

CARRERA DE INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL

AUTORIZACIÓN

Yo, LISSETTE ESTEFANÍIA ACURIO NOROÑA, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: "RECONOCIMIENTO DE LENGUA DE SEÑAS ECUATORIANO MEDIANTE SVM USANDO CARACTERÍSTICAS DE PROFUNDIDAD Y COLOR" en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi responsabilidad.

Sangolquí, junio del 2018

Lissette Estefanía Acurio Noroña
C.I.: 1804351060

DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y TELECOMUNICACIONES

CARRERA DE INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL

CERTIFICACIÓN

Certifico que el trabajo de titulación: "RECONOCIMIENTO DE LENGUA DE SEÑAS ECUATORIANO MEDIANTE SVM USANDO CARACTERÍSTICAS DE PROFUNDIDAD Y COLOR" fue realizado por la señorita LISSETTE ESTEFANÍA ACURIO NOROÑA, el mismo que ha sido revisado en su totalidad, analizando por la herramienta de similitud de contenido; por lo tanto cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de la Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Sangolquí, junio del 2019

Ing. Julio Cesar Larco Bravo, M.Sc
C.I.: 1710638808

DEDICATORIA

"*Intelligence without ambition is a bird without wings*"

Salvador Dalí

Dedico este trabajo a mis seres amados, a todas las personas que me alentaron, que me dieron valentía y fortaleza, y me desean lo mejor. Dedico este esfuerzo a mi familia, que no dejaron de estar a mi lado, preocupándose por mi bienestar y me dieron la mano para conseguir esta meta.

Dedico este logro a mi compañero de vida, Luis, que compartió días, noches y madrugadas de arduo trabajo, que sea un logro más para nosotros. Lo dedico también a mi Camila, que en sus risas, juegos, llantos y alegrías encontré mi mayor motivación para no desfallecer, lo dedico a ti mi pequeña, que te sientas orgullosa de donde provienes.

Lo dedico a mí, porque se necesita coraje, actitud, paciencia, carácter y decisión para ser una excelente profesional, una madre imperfecta, una compañera devota, una hija agradecida y una mejor persona en la vida.

Lissette Acurio

AGRADECIMIENTO

Mi gratitud a mi familia, por ser el apoyo incondicional en mi vida, gracias por sus palabras de aliento y sus consejos de vida. Gracias por amarme tanto, gracias por todos los sacrificios y gracias por siempre creer en mí.

Gracias al amor de mi pareja, su compañía en este proceso y su ayuda, que a pesar de las dificultades ha prevalecido nuestros sentimientos, gracias por todo lo que me has dado de ti.

Gracias a todas las personas que formaron, que forman y formarán parte de mi vida, porque de ellos he de aprender.

Y gracias a ti amigo/a, por darte el tiempo de leer este trabajo.

Lissette Acurio

INDICE DE CONTENIDO

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

RESUMEN

En el presente trabajo de titulación se propone un algoritmo de reconocimiento de señas del lenguaje dactilológicas ecuatoriano (LSEC) de una base de datos realizada previamente en el Instituto Nacional de Audición y Lenguaje, se aplicará un modelo basado en histogramas de gradientes orientados (HOG) para la extracción de características de los datos, y como entrenador se empleará al clasificador de máquinas de vectores de soporte (SVM). El modelo de reconocimiento se implementa utilizando los elementos de videos RGB y de profundidad, estos pasan por una etapa de preprocesamiento que consta en la alineación de los elementos y segmentación de la imagen, de esta manera, se extraerá la figura de la persona ejecutor de señas. Además, se lleva a cabo una reducción de dimensionalidad del vector HOG, correspondiente a la extracción de características de los datos, por medio de un modelo estadístico de análisis de componentes principales (PCA).

PALABRAS CLAVE:

- **LENGUA DE SEÑAS ECUATORIANO**
- **HOG**
- **SVM**
- **PCA**

ABSTRACT

The aim of the present study is to proposed an Ecuadorian sign language (LSEC) recognition of a database, we proposed a model based on the histogram of oriented gradients (HOG) for the extraction of features so that is used to train a support vector machine (SVM) classifier. For the recognition, the model is used RGB and depth data elements, there is implemented an alignment and segmentation process in order to remove background and leaving only the signer.In addition, it is applied a dimensionality reduction in the features HOG using a statistic model such as the principal component analysis (PCA).

KEY WORDS

- **ECUADORIAN SIGN LANGUAGE**
- **HOG**
- **SVM**
- **PCA**

CAPÍTULO I

1. INTRODUCCIÓN

1.1. Planteamiento del problema

Particularmente, la comunicación a partir de gestos humanos es un problema que requiere ser resuelto, el uso de recursos tecnológicos imparte nuevas soluciones innovadoras que ayudan a solventar la complejidad de resolución. Esta temática es relativamente nueva en el ambiente científico y es aplicada en diferentes áreas de estudio por su diversidad de aplicaciones. Sin embargo, la traducción del léxico humano varía según su ubicación a nivel mundial, por lo que se torna en un problema particular.

1.2. Antecedentes

La interpretación de lenguaje de signos como medio de comunicación para personas que presentan una discapacidad auditiva es un tema de inclusión social. En la actualidad, la tecnología ayuda a mejorar la calidad de vida de estas personas con el uso de aplicaciones de reconocimiento de señales o sistemas de traducción de lengua de señas (Leal Narváez, Leal Narváez, Henríquez Miranda, Pichón Pacheco, & Romero Martínez, 2016).

El reconocimiento de lengua de señas es un problema desafiante a nivel tecnológico, una posible solución se trata en utilizar un método de reconocimiento de imágenes que se apoya en el uso de sensores que recolecten la información y la aplicación de un algoritmo de reconocimiento e interpretación de lenguaje de señas. Dado que este no es un lenguaje universal ya que depende de su país, en su uso se distingue diferencias entre el significado de una seña con otra. En el Ecuador el Consejo Nacional de Igualdad de Discapacidades (CONADIS) cuenta con un diccionario de lengua de señas oficial ecuatoriano “Gabriel Román” que cuenta con alrededor de

5000 palabras las que incluyen gráficos y videos explicativos en formato web con el fin de observar la articulación de señas realizadas por profesionales (Consejo Nacional para la Igualdad de Discapacidades [CONADIS], s.f.).

El uso de herramientas tecnológicas ha sido clave para solventar esta problemática; la aplicación varía según la complejidad de la detección y la capacidad de la herramienta. Actualmente, el uso de sensores de movimiento facilita la recolección de información. Un ejemplo de un equipo que cuenta con estos recursos es Microsoft Kinect Sensor.

A partir del sensor Kinect de Microsoft se extrae una nube de puntos de información como profundidad, color y *skeleton frames* (Jana, 2012) y mediante diferentes técnicas y el uso de herramientas se puede realizar el reconocimiento de gestos estáticos o dinámicos.

Dentro de las técnicas prominentes que se desarrollaron para el reconocimiento de señas incluye el uso de elementos adicionales, tal es el caso del trabajo (Usachokcharoen, Washizawa, & Pasupa, 2015), para la distinción de las articulaciones se usó guantes combinados con diferentes colores, complementado con las características de profundidad, movimiento y color de los *frames* de video al momento de realizar el gesto. Sin embargo, los guantes presentan una limitante en cuanto a su tamaño lo cual implica la necesidad de que sean particulares para cada persona.

En el proyecto realizado por los autores Savur y Ferat se muestra otra técnica bajo el estudio de las señales EMG (Electromiografía) de superficie mediante el uso de electrodos para la adquisición de información fueron ubicados en el antebrazo derecho, siendo el método de extracción de características tales como dominio del tiempo, dominio de la frecuencia, densidad espectral de potencia y potencia promedio, para después ser aplicadas en un PCA (*Principal Component Analysis*). Como clasificador de datos hacen uso de SVM (*Support Vector Machine*)

y un algoritmo de aprendizaje, fueron utilizados para el trabajo de reconocimiento. Sin embargo, el proyecto fue realizado para señas estáticas del alfabeto estadounidense (Savur & Ferat, 2016).

Mediante el uso de distintas herramientas para el reconocimiento de lengua de señas se han obtenido resultados favorables, actualmente existe un impulso a las aplicaciones de traducción en tiempo real de señas dinámicas.

Acerca de los métodos para procesamiento de los datos utilizados están: el método HOG (*Histrogram of Oriented Gradients*), algoritmos en SVM (Yuqian & Wenhui, 2016), modelos estadísticos como NN (*Neuronal Network*), multi-dimensional HMM (*Hidden Markov Models*) y DTW (*Dynamic Time Warping*), entre otros más (Yanmei, Bing, Yen-Lun, Guoyuan, & Xinyu, 2015).

La implementación de un algoritmo de reconocimiento en tiempo real propuesto a partir de una red neuronal MLP (*Multi-Layer Perceptron*) con BP (*Back Propagation*) en conjunto con el sensor de Microsoft Kinect para obtener datos de profundidad, es otro de los proyectos realizados teniendo como resultado el reconocimiento del 96.5% de 26 letras del lenguaje de señas americano (Naglot & Kulkarni, 2016).

En otro trabajo se usó *Hidden Markov Model* (HMM) para la clasificación de 16 palabras de lenguaje de señas árabe con la extracción de once características de cada *frame* que consiste en la probabilidad de transición de estados obteniendo como resultado un reconocimiento promedio de 64.61% (Sarhan, EI-Sonbaty, & Youssef, 2015).

El proyecto realizado en el año 2016 para la lengua de señas chino presenta una nueva propuesta de extracción de características espacio-temporal describiendo cada *frame* mediante HOG y PCA, y siendo optimizado por la determinación de variaciones en la forma de la mano, al fusionar la probabilidad de trayectoria y forma de la mano como resultado se demuestra que el

método de estados ocultos adaptativos implementado es mejor que la línea base de métodos (Zhang, Zhou, Xie, Pu, & Li, 2016).

El reconocimiento de la trayectoria continua de gestos de manos es capturado por Kinect Sensor y procesada en OpenCV[1] siendo la información enviada a un controlador Arduino para ejecutar la acción de movimiento en un brazo robótico. Se considera características como la trayectoria de la mano, diferencias en tono de color de piel, tamaño y movimiento, se aplica como método de reconocimiento el modelo discreto HMM (Pal & Kakade, 2016).

En el año 2017, el trabajo realizado por los autores Gibran García, Felipe Trujillo y Santiago Caballero exponen el uso del guante multicolor en conjunto con Microsft Kinect sensor v1 para la adquisición de datos. Utiliza un algoritmo DTW para la interpretación de gestos y a partir de pruebas en tiempo real alcanza una precisión media de reconocimiento de un 98.57%. A pesar de ser DTW rápido y preciso, la limitación de este método es la complejidad del tiempo del algoritmo que afecta al rendimiento reduciendo el tamaño de la base de datos (García Bautista, Trujillo Romero, & Caballero Morales, 2017).

Un trabajo de investigación en India propone el uso de dos algoritmos de reconocimiento basado en reglas y DTW el cual consiste en la secuencia de posturas conectadas con el movimiento durante un periodo de tiempo y la variación de distancias entre dos diferentes secuencias de movimiento de señas, cabe indicar que debido al tamaño del vocabulario de señas que utiliza el método de reconocimiento alcanza un promedio de 96.25% para 40 señas (Ghotkar & Kharate, 2015).

Dado que los modelos de reconocimiento NN y HMM requieren una fase de entrenamiento más compleja se ve aligerada en su carga computacional, caso contrario, el modelo DTW

[1] Ver información en: https://opencv.org/

presenta una alta demanda computacional frente a un amplio vocabulario a pesar de presentar una mayor precisión (Xu, 2016).

Actualmente, el modelo de reconocimiento usando el clasificador *Support Vector Machine* ofrece mayor robustez en la detección para el análisis de rasgos en imágenes de cáncer en la piel durante el presente año obteniendo resultados favorables (Dai, 2018).

Si se complementa con la aplicación del descriptor HOG, el reconocimiento de patrones supone una mejor interacción frente a una gran cantidad de datos de entrenamiento. Es importante decir que el cálculo de los descriptores de HOG puede tener un coste en tiempo computacional grande pues requiere el cálculo de HOG por cada celda (Surhone, Tennoe, & Henssonow, 2010) (Miller, Vandome, & McBrewster, 2013). Por lo cual la segmentación de la imagen es necesaria para reducir la dimensión del vector característico. Este hecho se ve reflejado en los trabajos que se hablará a continuación.

En el año 2016, los autores Jun He, Zhandog Liu y Jihai Zhang presentan el reconocimiento de lengua de señas chino haciendo uso de las características extraídas de la trayectoria de la forma de la mano mediante HOG con SVM y validación HMM (VHMM) para considerar la relación entre manos y otras partes del cuerpo y la distancia relativa entre ellas RDF (*Relative Distance Feature*) Se pone a prueba con la información recolectada en dos bases de datos diferentes obteniendo resultados favorables (He, Liu, & Zhang, 2016).

Un proyecto similar (Hamed, Belal, & Mahar, 2016) se desarrolló en el mismo año utilizando los algoritmos HOG y SVM junto con las librerías de software de Kinect para el reconocimiento de posición, forma y trayectoria de mano. Cuenta con una base de datos de 72 palabras. El impacto de reconocimiento del experimento es de 89.6%, indican la lectura que la introducción de luz puede afectar las imágenes (Yuqian & Wenhui, 2016).

En busca de la mejora de la precisión de reconocimiento, el estudio realizado por Ayman Hamed, Nahla A. y Khaled M. Mahar muestra que utilizando RGB-ratio, la extracción de HOG de la imagen y luego aplicar PCA en un clasificador de máquina el sistema reconoce alfabetos árabes con una precisión del 99.2% (Hamed, Belal, & Mahar, 2016).

La fusión de los datos recogidos por Kinect de los sensores de profundidad y RGB en el trabajo para el reconocimiento de lengua de señas árabe obtuvo un mejor resultado general de 99.8% dado que mejora la precisión. Las imágenes recogidas son segmentadas mediante algoritmos diferentes para RGB y profundidad, para imágenes RGB se usó el modelo gaussiano para el color de piel de manos y rostro, después es convertida a escala de grises. La extracción de invarianzas fue necesario para la aplicación de análisis Fisher LDA (*Linear Discriminant Analysis*) (Aliyu, Mohandes, Deriche, & Badran, 2016).

Existe un trabajo previo de la Universidad de las Fuerzas Armadas ESPE donde se realizó una base de datos actual que cuenta con palabras de uso común del lenguaje dactilológico ecuatoriano donde se diferencian diferentes grupos nombrados como: “Adjetivos”, “Alimentos”, “Colores”, “Saludos” y “Juguetes & Cosas”, a más de las letras estáticas y dinámicas como son “ll”, “rr”, “ch”, “j” y “z” realizadas por personas profesionales entre hombres y mujeres de diferentes edades y contextura física (Hu & Teran, 2017).

La base consiste de 10 palabras diferentes para cada uno de los grupos de palabras siendo un total de 50 palabras con un número de muestras de 420. La información almacenada contiene datos de videos de profundidad, videos RGB y datos de *skeleton tracking* de *frames* importantes al momento de emplear la aplicación *Classification Learner* de MATLAB (Hu & Teran, 2017).

Alternativamente, un trabajo realizado en el año 2016, a través de un guante inteligente y la incorporación de un sistema de lenguaje basado en código Morse ofrece una opción de

comunicación en tiempo real para personas con discapacidad auditiva y trastornos del habla (Aguiar, y otros, 2016). A pesar de la nueva propuesta esta debe ganar mercado a diferencia del lenguaje de señas que es altamente aceptado por las instituciones.

1.3. Justificación e Importancia

La problemática social que enfrenta el grupo de personas con discapacidad auditiva en el Ecuador es no tener la facilidad de comunicarse libremente en su entorno social, por lo cual, es necesario un medio de comunicación como es el lenguaje de señas. Sin embargo, el conocimiento de este lenguaje no es general a nivel poblacional, por tanto, dificulta el entendimiento con este grupo de personas.

El método más común en la actualidad para realizar el reconocimiento de señas es mediante SVM, dado los resultados de precisión y rendimiento a una base de datos extensa y ampliable como se muestra en (Hamed, Belal, & Mahar, 2016) y (Aliyu, Mohandes, Deriche, & Badran, 2016) además en nuevos proyectos innovadores como se menciona en (Dai, 2018).

En el presente trabajo, para obtener la mejor precisión de reconocimiento se utilizará características tanto de profundidad como RGB que se obtendrán a partir de la información de la base de datos (Hu & Teran, 2017). Además, la realización de este trabajo ayudará a proyectos en el Instituto Nacional de Audición y Lenguaje (INAL) que fomenten sistemas de apoyo para la compresión de lenguaje de señas en base a palabras comunes de adjetivos, alimentos, colores, saludos, juguetes y cosas, que puedan ayudar al entendimiento para personas que desconocen el lenguaje.

1.4. Alcance del Proyecto

El presente proyecto tiene por objetivo realizar el reconocimiento de lenguaje de señas dinámicas y estáticas a partir de información de los datos obtenidos por Kinect Microsoft V2 de una base de datos pública de uso libre de lenguaje dactilológico ecuatoriano la cual contiene diferentes palabras de uso común como adjetivos, alimentos, colores, saludos, juguetes y cosas además del alfabeto ecuatoriano (Hu & Teran, 2017).

Como método de reconocimiento se pretende usar SVM para la clasificación de características tanto de profundidad como RGB ya que la combinación de esta información ayudará a mejorar la precisión de reconocimiento de palabras como lo demuestra los trabajos previos publicados (Dai, 2018), (Dalal & Triggs, 2005), (Feng & Yuan , 2013), (Hamed, Belal, & Mahar, 2016) and (Houssein Ahmed, Kpalma, & Osman Guedi, 2017).

Finalmente, se busca ayudar a proyectos de interpretación de lengua de señas del Instituto Nacional de Audición y Lenguaje (INAL) aportando de esta manera en un mejor estilo de vida para personas con discapacidad auditiva.

1.5. Objetivos

1.5.1. Objetivo General

- Realizar el reconocimiento de señas del lenguaje dactilológico ecuatoriano usando información de una base de datos existente por medio del modelo de SVM.

1.5.2. Objetivos Específicos

- Contrastar la información recogida sobre trabajos previamente realizados.

- Extraer características de profundidad y RGB a partir de la información almacenada en la base de datos
- Establecer algoritmos de clasificación SVM de acuerdo a la precisión de resultados.
- Analizar el desempeño del algoritmo de reconocimiento.
- Generar un documento con las evidencias del desarrollo del proyecto.

1.6. Metodología

El presente documento se encuentra distribuido de la siguiente manera: el primer capítulo detalla la problemática del tema, antecedentes generales, justificación e importancia, alcance del proyecto y objetivos generales y específicos. El capítulo dos describe conceptos teóricos relevantes para el proyecto tales como: el procesamiento digital de imágenes, algoritmo de reconocimiento SVM y HOG. El capítulo tres describe el desarrollo del proyecto, define aspectos generales de la lengua de señas ecuatoriano (LSEC), hace una descripción general de la base de datos a utilizar, la aplicación del algoritmo de reconocimiento, la clasificación y verificación de los datos procesados. Posteriormente, el capítulo cuatro detalla las pruebas y análisis de resultados del entrenamiento de los datos, Finalmente, el capítulo cinco se refiere a las conclusiones, recomendaciones en cuanto a la elaboración del proyecto y presenta propuestas para trabajos futuros.

CAPÍTULO II

2. MARCO TEÓRICO

En este capítulo se hablará brevemente sobre aspectos conceptuales y definiciones necesarias para la comprensión adecuada del trabajo. Inicialmente se contempla una introducción sobre el procesamiento digital de imágenes como su formación, representación, extracción de características y segmentación de la imagen; seguidamente se habla sobre generalidades del reconocimiento de patrones basado en *Support Vector Machine* como clasificador y el descriptor HOG, y por último una descripción del modelo estadístico PCA (*Principal Component Analysis*).

2.1. Procesamiento digital de imágenes

El procesamiento digital de imágenes refiere al procesamiento de la imagen digital por medio de un dispositivo capaz de representar la imagen. Dado que una imagen se compone de varios elementos, cada uno de estos tiene su valor de estudio (Gonzales & Woods).

Una imagen digital es una matriz bidimensional formada por varios elementos de información, el elemento en su menor expresión es el pixel, que es la menor unidad homogénea, y el conjunto de pixeles forman la imagen. Las propiedades del pixel es su valor de intensidad y locación (x, y), como se aprecia en la **(**Figura 1), el número de columnas y filas determinan el tamaño de la imagen $(m \times n)$ (Moulick & Ghosh, 2013).

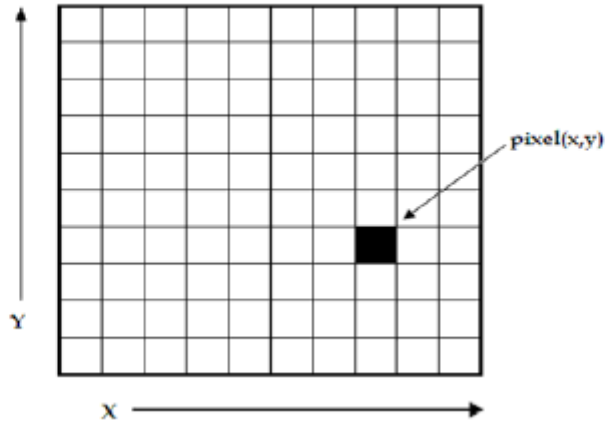

Figura 1. Imagen digital, representación de un pixel caracterizado por su valor y coordenada (x, y)
Fuente: (Moulick & Ghosh, 2013)

El valor de un pixel está dado por la representación de tres canales RGB (Red, Green, Blue) que en español viene siendo (Rojo, Verde, Azul). Estos pueden tomar valores de 0 a 255, donde, 0 representa la ausencia del color y 255 el máximo valor de ese color en un punto. Se puede apreciar en la siguiente Figura 2, lo antes mencionado, además de la variación de la imagen según cada canal siendo la conformación de estos la intensidad de la imagen.

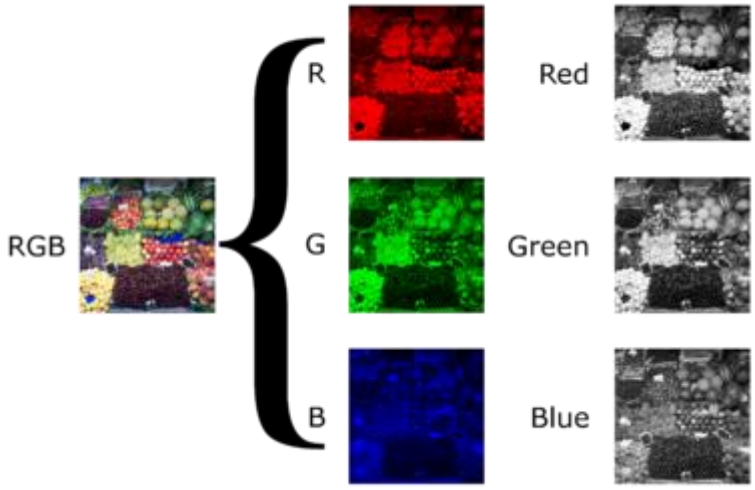

Figura 2. Composición de canales RGB a escala de grises
Fuente: (Dilmen, 2012)

La intensidad de un pixel se relaciona con la cantidad de luz emitida, se puede definir como profundidad del pixel o también conocida como resolución de niveles de una imagen en escala de grises (López, 2014). El valor de intensidad está caracterizado generalmente por la suma de los tres canales, se puede normalizar utilizando el máximo de los tres canales según la aplicación como se ve en la ecuación (1):

$$(R + G + B) \cdot \frac{255}{\max\limits_{i \epsilon I}(R_i + G_i + B_i)} \tag{1}$$

Donde:

R, corresponde al canal del rojo

G, corresponde al canal del verde

B, corresponde al canal del azul

I, corresponde a la imagen

max, es el máximo que varía entre el rango 0 y 255

Usualmente el valor de menor intensidad corresponde al negro y la mayor corresponde al blanco (Vanrell, 2015). Para entender la formación de un pixel y la razón del valor de cada canal se explica en la siguiente sección.

2.1.1. Formación de la imagen

La composición del color de un pixel viene dada por la luz emitida en el ambiente, la sensibilidad de la cámara y el espacio o superficie donde se refleja la luz que es detectada por la cámara. La luz es reflejada sobre el objeto como una onda, estas varían entre largas y cortas, dentro del rango de la luz visible las ondas más largas corresponden a la gama de la luz roja y las cortas son de la luz azul (Pliego).

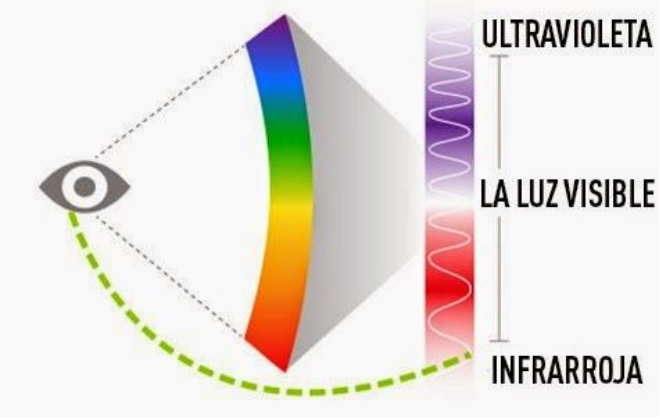

Figura 3. Rango del espectro de la luz visible para el ojo humano
Fuente: (Biologico, 2014)

Del tipo de material de la superficie dependerá de si las longitudes de onda son absorbidas o reflejadas. La luz reflejada es el resultado del producto de la luz emitida $I(\lambda)$ por el porcentaje de la luz reflejada $S(\lambda)$ como en la ecuación (2) a continuación, donde λ, es la longitud de onda: (Vanrell, 2015)

$$I(\lambda) \cdot S(\lambda) \tag{2}$$

Gráficamente, este producto se representará como la Figura 4.

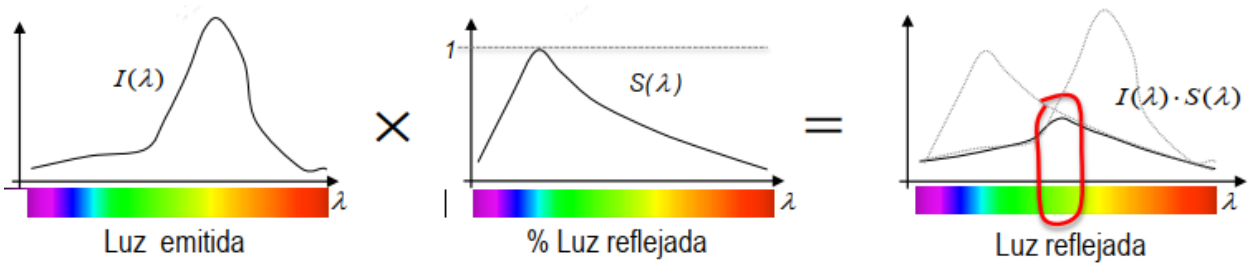

Figura 4. Representación de la función de la luz reflejada siendo el producto de la función *I* por la función *S*
Fuente: (Vanrell, 2015)

En cuanto a la sensibilidad de la cámara, la captación de la imagen se genera al haber una incidencia de luz en cada uno de los sensores según la longitud de onda que detecten, la señal receptada genera un impulso eléctrico cuya intensidad es equivalente a la intensidad de la luz incidente. Entonces la captación de la luz nos indicará un umbral superior e inferior, donde el límite superior se lo denomina saturación y el rango se lo expresa como profundidad de la imagen (Grau, 2003).

2.1.2. Representación de una imagen digital

Brevemente, veremos sobre diferentes tipos de imágenes basadas en el tipo de sensor de la cámara, en nuestro caso, para la recolección de grabaciones que conforman la base de datos de estudio se utilizó el dispositivo Kinect v2 de Xbox One por su capacidad de resolución. (Hu & Teran, 2017).

2.1.2.1. Imagen RGB

Una imagen a color o imagen RGB está conformada por tres funciones que se asocian a las longitudes de onda de los colores rojo (R), verde (G) y azul (B). Cada pixel consta de tres

valores, por lo tanto, corresponde a un byte por color (García, 2016/2017), se puede observar en la siguiente Figura 5.

Figura 5. Imagen RGB
Fuente: (García, 2016/2017)

2.1.2.2. Imagen escala de grises

Esta imagen se forma a partir de las intensidades de los componentes de RGB, el valor del pixel en un punto será de 1 byte, este tendrá 256 niveles de gris, en donde el menor valor 0 corresponde al negro y el mayor valor 255 corresponde al blanco (García, 2016/2017), se aprecia la descripción en la Figura 6.

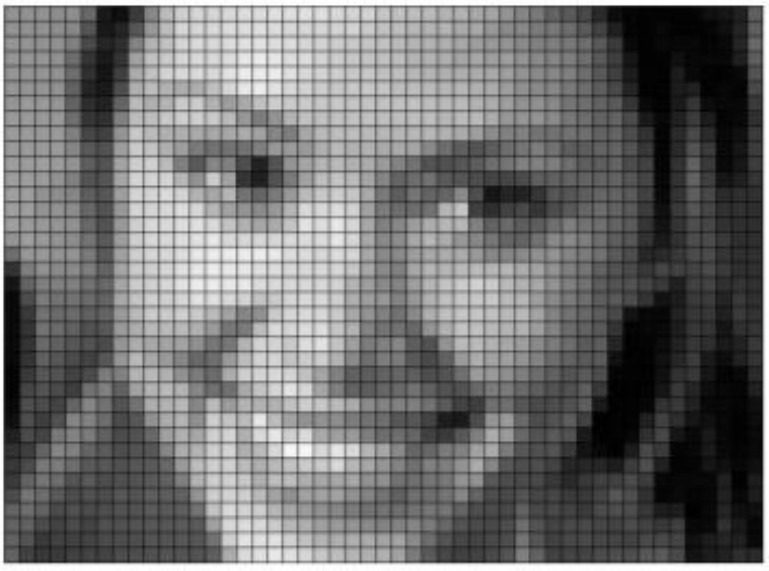

Figura 6. Imagen en escala de grises
Fuente: (García, 2016/2017)

2.1.2.3. Imagen binarizada

En una imagen binaria los pixeles son normalizados a 0 para el negro y 1 para el blanco, siendo los equivalentes al 0 y 255, respectivamente (López, 2014). El pixel entonces tendrá el valor de 1 bit, la Figura 7 muestra la normalización de la Figura 6.

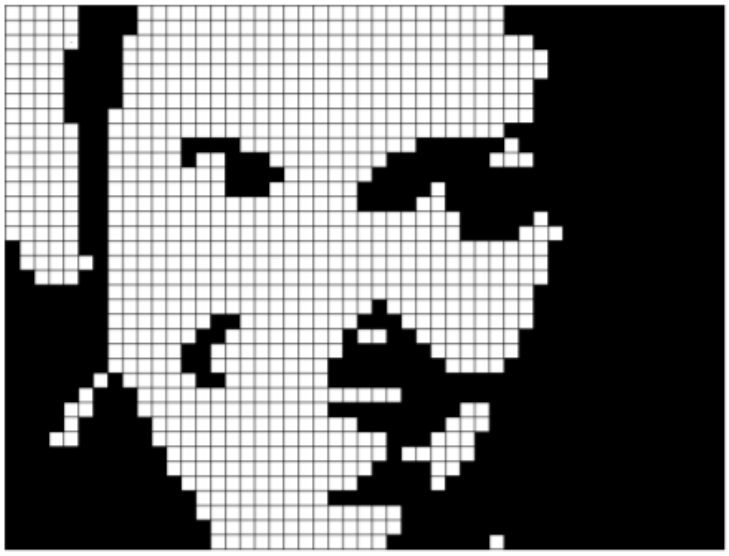

Figura 7. Imagen binarizada
Fuente: (García, 2016/2017)

2.1.2.4. Imagen RGB-D

Una imagen RGB-D es la combinación de la imagen RGB a la cual se le añade un canal que contiene información de la profundidad del objeto respecto al sensor de la cámara. El valor del pixel corresponde a la distancia en el plano (Wu, 2015). Un ejemplo de imagen RGB-D se puede apreciar en la Figura 8.

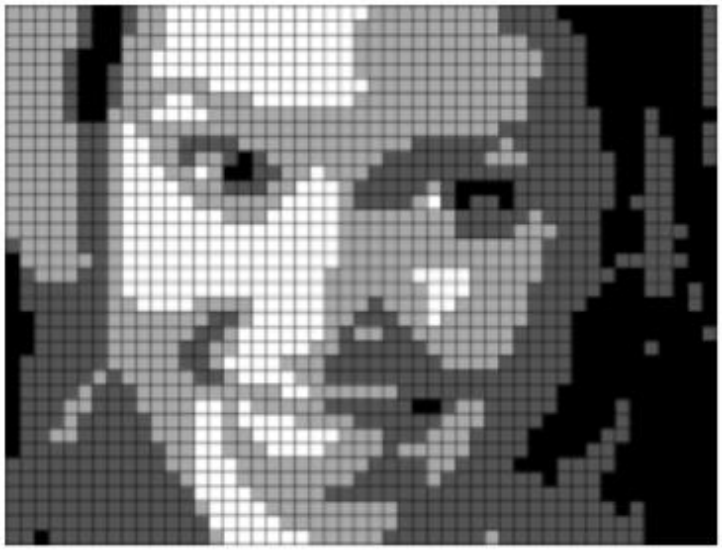

Figura 8. Imagen RGB-D
Fuente: (García, 2016/2017)

2.1.3. Extracción de características

Para la realización del proyecto es necesario partir de descriptores que ayuden a la extracción de características de una imagen para la detección de manos y rostro de un conjunto de *frames* provenientes de una variedad de señas de palabras.

2.1.3.1. Operaciones lógicas y aritméticas

La operación lógica que se realizará es AND utilizando dos imágenes procesadas individualmente, de tal forma que se obtenga una imagen nueva en la cual se ha eliminado el fondo de la imagen, quedando únicamente la persona. Las operaciones lógicas son útiles en la detección de características y análisis de formas, además de definición de máscaras. Esta operación ayudará a eliminar ciertas áreas de la imagen con el fin de definir a la persona y posteriormente, únicamente, manos y rostros, se aprecia en la Figura 9.

Figura 9. Imagen AND

En cambio, la operación aritmética suma y resta se utiliza generalmente para eliminar ruido, de igual manera la operación multiplicación y división ayuda a efectuar una correlación entre imágenes (López, 2014).

2.1.3.2. Segmentación de imágenes

Dado una imagen se busca diferenciar distintas regiones según ciertas características de interés, como método de detección se recurre a la segmentación que es la subdivisión de una imagen según la parte a identificar. La segmentación puede ser a partir de regiones, de identificación de bordes, líneas o curvas, entre otros (Palomino & Concha, 2009). Se puede entender también como la clasificación de puntos (pixeles) en una imagen con características homogéneas que permiten la discriminación entre regiones.

Algunas de las técnicas que se nombra en el trabajo de (Palomino & Concha) como algoritmos de segmentación se basan en las propiedades de los valores de los pixeles en escala de grises como son la discontinuidad y similaridad.

La discontinuidad consiste en la división de la imagen basándose en cambios bruscos del nivel del gris, ayuda a la detección de puntos aislados, detección de líneas y detección de bordes.

En cambio, la similaridad permite construir regiones según la regularidad de los valores en la escala de grises. Sus principales métodos son: umbralización, crecimiento de región y división y fusión de regiones (Palomino & Concha, 2009).

Al realizar la operación de segmentación se realiza una discrimación de pertenencia del pixel dentro de un conjunto determinado, en este caso, se hará en base a las propiedades del pixel según su entorno. Un ejemplo gráfico de este método se aprecia en la Figura 10, donde que, se observa la imagen de la izquierda en su formato original, al aplicar segmentación se obtiene una nueva imagen, siendo la parte de interés de blanco y la negra son los pixeles que se descartan del conjunto de interés.

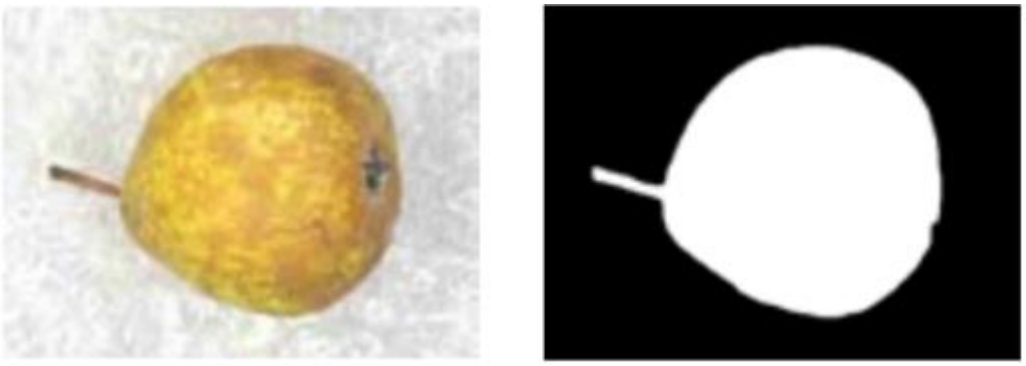

Figura 10. Segmentación de imagen
Fuente: (Palomino & Concha, 2009)

La forma en que se aplica este método en este trabajo es para diferenciar el fondo de la persona, específicamente de las manos y rostro, a estos los definiremos como objetos.

2.2. Histogramas de Gradientes Orientados HOG

En base al artículo realizado por Dalal y Triggs (2005) sobre la detección de peatones se demuestran experimentalmente que los descriptores de Histogramas de Gradientes Orientados o en inglés *Histogram of Oriented Gradients* (HOG) superan a otros descriptores existentes basada en SVM lineal, y a partir de este, se realizaron métodos posteriores para detectar cualquier otro tipo de objetos (Dalal & Triggs, 2005).

2.2.1. Descriptores Gradiente

El descriptor HOG permite representar la información de una imagen expresada en un vector, esta información ayuda a la detección de objetos y reconocimiento de imágenes. HOG se basa en la distribución de las direcciones de los gradientes, la magnitud de los gradientes alrededor de bordes y esquinas, se utilizan como características; esto se debe que regiones de cambios bruscos de intensidad indica la formación del objeto (Mallick, 2016).

Se utiliza la información del gradiente de la imagen a partir de la combinación de histogramas locales que se calcula en celdas que se distribuyen de forma uniforme, en la Figura 11 se detalla dicha descripción, dado la imagen original (Figura 11.a), al aplicar el descriptor HOG proporciona información de las orientaciones de los contornos en cada posición (Figura 11.b).

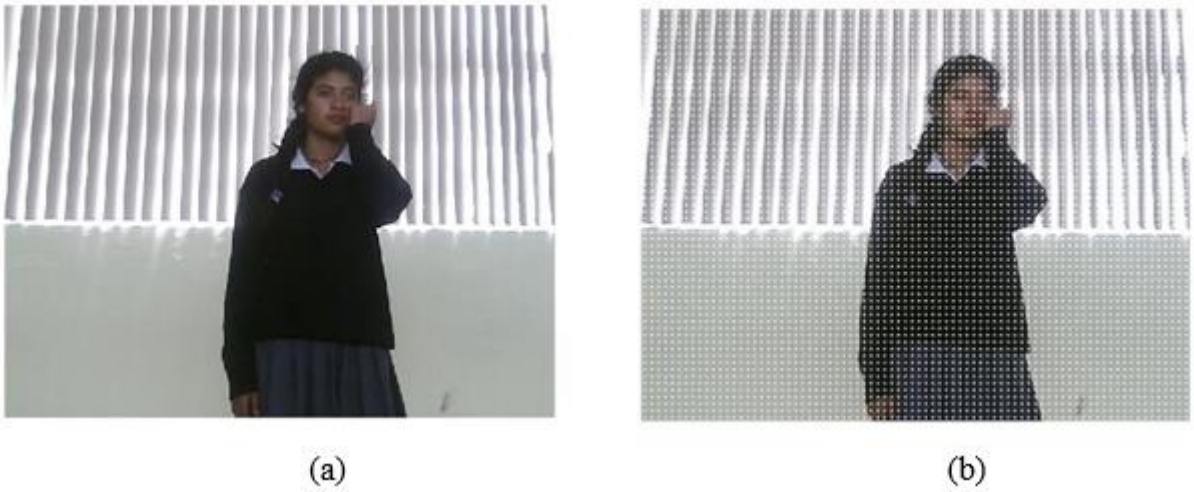

Figura 11. Ejemplo de descriptor HOG. (a) Imagen Original y (b) Imagen aplicada descriptor HOG

De esta manera, la aplicación del histograma de gradiente orientado ayudará a distinguir la forma de los objetos presentes. En los siguientes puntos se detalla ampliamente sobre este descriptor.

2.2.2. Cálculo del Gradiente

La formación del vector gradiente es dado por la diferencia de intensidad tanto en horizontal, dx, como en vertical, dy, para cada pixel de la imagen, se siguen las siguientes ecuaciones:

$$dx = I(x+1, y) - I(x-1, y) \tag{3}$$

$$dy = I(x, y+1) - I(x, y-1) \tag{4}$$

Dado una imagen I (Figura 12), se requiere el valor del gradiente en un punto (x, y) para obtener el valor local de dirección de cambio de intensidad máximo y magnitud del cambio de dirección de máxima variación.

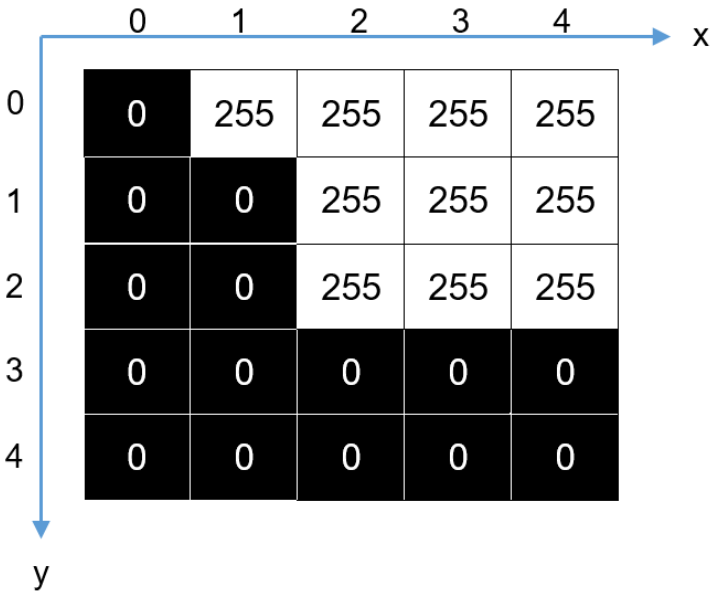

Figura 12. Imagen I

Al aplicar las ecuaciones (3) y (4) en un pixel (1, 1) el valor de la diferencia de intensidad en la dirección horizontal de *x* y de la diferencia de intensidad vertical de *y* se representa de la siguiente manera:

$$dx = 255 - 0 = 255 \tag{5}$$

$$dy = 255 - 0 = 255 \tag{6}$$

A partir de estos datos se define un vector con una orientación y magnitud que sigue las siguientes ecuaciones, respectivamente:

$$\theta(x, y) = \arctan\frac{dy}{dx} \tag{7}$$

$$g(x, y) = \sqrt{dx^2 + dy^2} \tag{8}$$

Donde:

θ, es la orientación del gradiente

g, es la magnitud del gradiente

Se puede apreciar en la Figura 13 la aplicación del descriptor gradiente, la primera es la imagen original en escala de grises a la cual se le ha aplicado el cálculo de gradiente en *x* y en *y*, combinando estas imágenes da como resultado el gradiente global para cada pixel, el efecto que se logra es la formación de la silueta, en este caso de la persona.

Figura 13. Imagen de una persona en escala de grises aplicada el gradiente en dirección horizontal y vertical y la combinación de ambos
Fuente: (Valveny, HOG - Cálculo del gradiente, 2015)

La representación a nivel de pixel es difícil de manejar con un clasificador ya que requiere como entrada una representación global de toda la imagen (Valveny, HOG - Cálculo del gradiente, 2015).

2.2.3. Cálculo del Histograma de Gradiente

Se vio la representación del gradiente de forma local para cada uno de los pixeles, se requiere expresar la información de la imagen en un vector característico global, para esto se sigue dos pasos, el primer paso es la división de la imagen en celdas y el segundo es el cálculo de histogramas para todas las celdas para la formación de la representación de características global. Cada paso se explica a detalle a continuación (Valveny, HOG - Cálculo de los histogramas, 2015).

Para el cálculo del histograma de orientaciones en una celda, el primer parámetro que se fijará es el tamaño de la celda y la división del rango de orientaciones en un número de intervalos fijo. Dado que se utilizará el comando *extractHOGFeatures* de MATLAB por defecto el tamaño de celda es de 8x8 pixeles, el número de intervalo para el rango de orientación está predeterminado de 9 espacios uniformes de 0 a 180º, esta opción dos gradientes con la misma dirección, pero signos contrarios se consideran equivalentes se asignan en el mismo intervalo (The MathWorks, Inc., 2013).

Calculado el descriptor de pixel se asigna el valor de la magnitud del gradiente en función de su orientación, la acumulación de todos gradientes asignados en un mismo intervalo es el valor que se obtendrá en el histograma, sigue la siguiente ecuación:

$$h(k) = \sum_{(x,y)\in C} \omega_k(x,y)g(x,y) \tag{9}$$

Donde:

$h(k)$, es la expresión para el valor del histograma en cierta posición k

ω_k, es el factor que determina la asociación del gradiente en el intervalo k

C, es la celda que contiene un conjunto de pixeles

Se expresa entonces dentro de una celda como k a la posición de cada uno de los nueve intervalos según el rango de orientación del gradiente, la suma de las magnitudes $g(x,y)$ para todos los pixeles y todos los gradientes de la celda ponderado por un factor que determina la asociación del gradiente según el intervalo.

El factor de asignación se definirá de la siguiente forma:

$$\omega_k(x,y) = \begin{cases} 1 & si\ (k-1)\delta\theta \le \theta(x,y) < k\delta\theta \\ 0 & en\ caso\ contrario \end{cases} \tag{10}$$

Donde:

$\delta\theta$, es el rango de cada uno de los intervalos de orientación del histograma

Para todos los gradientes cuya orientación este dentro de los límites definidos por el intervalo valdrán uno, caso contrario su valor será cero.

Este método es bastante simple, pero en el caso de que existan orientaciones similares esto puede implicar variaciones significativas al final, por lo que, se asigna a cada gradiente a los dos intervalos cercanos con un peso proporcional a la distancia de la orientación al centro del intervalo. Se ajusta a la siguiente expresión:

$$\omega_k(x,y) = max\left(0, 1 - \frac{\theta(x,y) - \theta_k}{\delta\theta}\right) \tag{11}$$

Donde:

θ_k, es la orientación del centro del intervalo a un determinado intervalo k

En la Figura 14 se muestra de manera didáctica la distancia para el cálculo de asociación de un determinado gradiente en un intervalo dado, siendo las flechas rojas gradientes con orientaciones muy similares que han quedado asignados a intervalos diferentes.

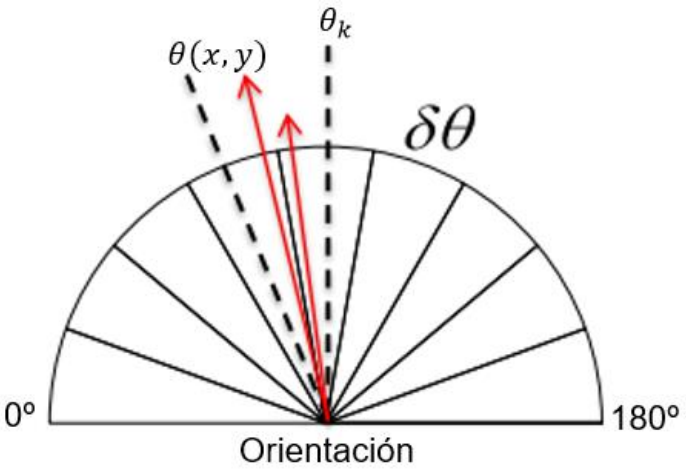

Figura 14. Cálculo de factor de ponderación para

cada una de las posiciones del histograma

El valor de la distancia $\theta(x,y) - \theta_k$ puede tomar los siguientes valores para el factor de ponderación:

$$\begin{aligned} \theta(x,y) - \theta_k \approx 0 &\quad \rightarrow \quad \omega \approx 1 \\ \theta(x,y) - \theta_k \approx \delta\theta &\quad \rightarrow \quad \omega \approx 0 \\ \theta(x,y) - \theta_k > \delta\theta &\quad \rightarrow \quad \omega = 0 \end{aligned} \tag{12}$$

Este proceso se repite para todas las celdas que se dividen la imagen, ver Figura 15.

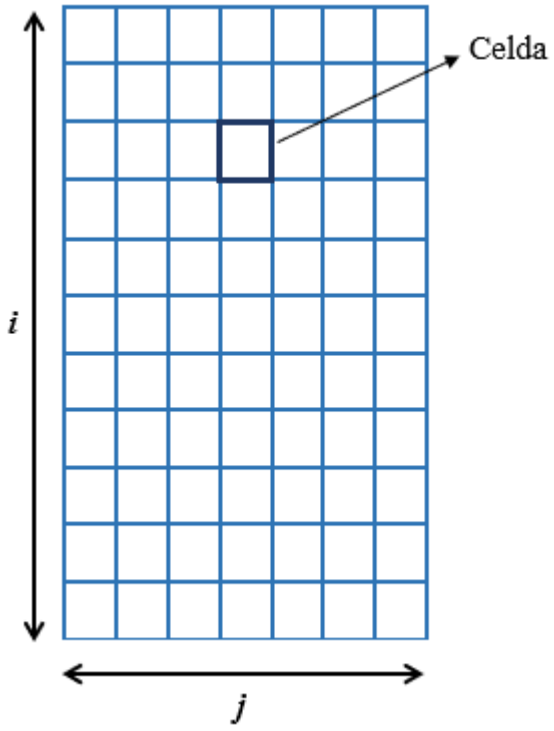

Figura 15. Disposición de celdas en imagen

Entonces a cada celda corresponde un histograma, lo que corresponde es hacer la integración espacial de todos los histogramas usando la ecuación

$$h_{ij}(k) = \sum_{(x,y)\in C_{ij}} \omega_k(x,y)\, g(x,y) \tag{13}$$

Donde:

$h_{ij}(k)$, es la expresión para el cálculo del histograma para cada una de las celdas en un determinado intervalo k

C_{ij}, es la celda en su correspondiente posición (i, j)

Dado que se puede presentar el mismo problema que en la asignación de orientaciones se incurre al mismo método de interpolación para evitar variaciones significativas en el resultado final, por ejemplo, en la Figura 16 se encuentran dos pixeles muy cercanos que han sido asignados en dos celdas diferentes, los pixeles son representados por puntos negros.

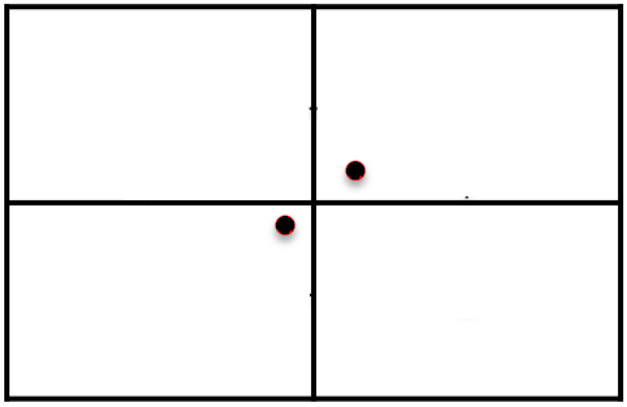

Figura 16. Pixeles en celdas cercanas

La solución es asignar a cada pixel, representados por puntos azules, a las cuatro celdas más cercanas con un peso proporcional a la distancia del pixel al centro de cada celda (Figura 17). Se realiza el cálculo de distancia en dirección *x* y dirección *y*.

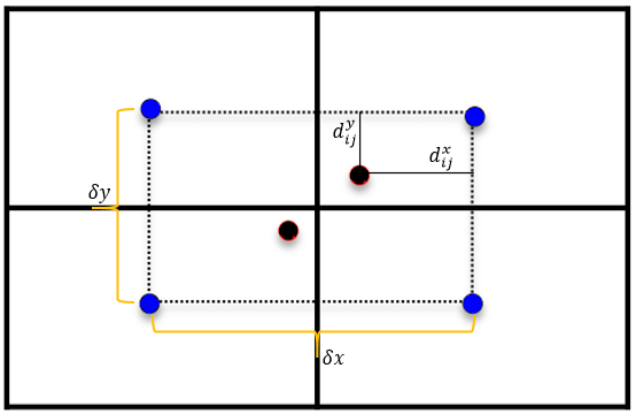

Figura 17. Distancia del pixel al centro de cada celda

Similar al peso de distancias en la orientación del gradiente se calcula para el factor de asignación de cada uno de los pixeles de las celdas normalizadas por la distancia entre los centros de las celdas según la dirección horizontal y vertical.

$$\omega_{ij}^{x}(x,y) = max\left(0, 1 - \frac{d_{ij}^{x}}{\delta x}\right) \tag{14}$$

$$\omega_{ij}^{y}(x,y) = max\left(0, 1 - \frac{d_{ij}^{y}}{\delta y}\right) \tag{15}$$

Donde:

$\omega_{ij}^{x}(x,y)$, es la expresión del valor de asignación de un pixel en una celda en el eje x

$\omega_{ij}^{y}(x,y)$, es la expresión del valor de asignación de un pixel en una celda en el eje y

d_{ij}^{x}, es la distancia desde el pixel hacia el centro de la celda en sentido horizontal

d_{ij}^{y}, es la distancia desde el pixel hacia el centro de la celda en sentido vertical

δx, es la distancia entre los centros de dos celdas en sentido horizontal

δy, es la distancia entre los centros de dos celdas en sentido vertical

Aplicando las ecuaciones (14) y (15) en la ecuación (13) se obtiene la ponderación final y el valor concreto del histograma en cada celda, más no el descriptor HOG final (Valveny, HOG - Cálculo de los histogramas, 2015).

$$h_{ij}(k) = \sum_{(x,y)\in C_{ij}} \omega_{ij}^{x}(x,y)\omega_{ij}^{y}(x,y)\omega_{k}(x,y)g(x,y) \tag{16}$$

Se ha hablado de que este proceso se realiza en una imagen a escala de grises, en el caso que sea una imagen a color, el color se prioriza según su dominante para cada pixel, se realiza el cálculo del gradiente en los tres canales por separado y se toma el gradiente con mayor magnitud (Valveny, HOG - Cálculo del gradiente, 2015).

2.2.4. Descriptor HOG

Después de haber realizado el cálculo de histograma se introduce la normalización y agrupación de histogramas en forma de bloques para obtener el vector característico final, esto con el objetivo de conseguir la máxima invarianza posible dado que la entrada de una imagen no se vea afectada por una diferente iluminación, posición, escala, aspecto, entre otros.

Dicha normalización no debe ser únicamente uniforme a lo largo de la imagen, sino que es necesario una normalización local según la región de la imagen, por lo que, el bloque ayudará a la agrupación de celdas vecinas (Valveny, HOG - Cálculo del descriptor, 2015).

El número de celdas por defecto del comando *extractHOGFeatures* es de dos por dos para formar el bloque, para garantizar la normalización adecuada se realiza una superposición de bloques, se colocan con la separación de una celda entre ellos tanto en horizontal como en vertical, la Figura 18 muestra una imagen con seis celdas y señala el bloque (The MathWorks, Inc., 2013).

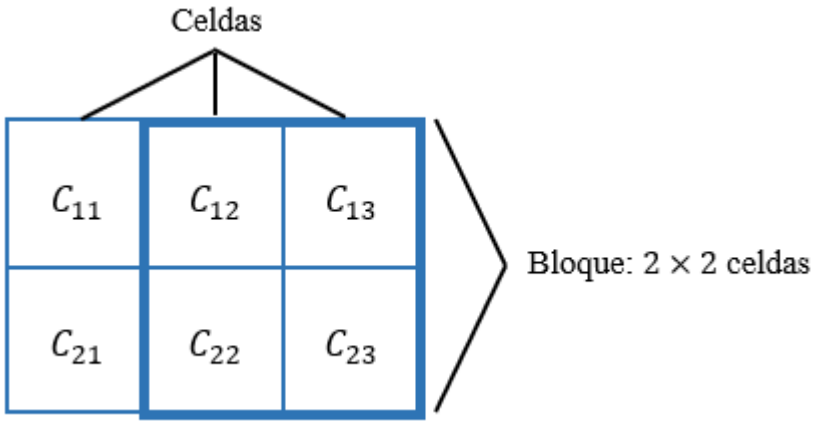

Figura 18. Representación de bloques 2x2 celdas

La normalización se obtiene dividiendo cada uno de los componentes del vector resultante de la concatenación de todas las celdas por su norma *L2*, también conocida como norma euclídea. La norma de un vector se expresa de la siguiente forma, siendo la ecuación (17) un único vector por bloque concatenando los histogramas de todas las celdas y la ecuación (18) su normalización.

$$v = (x_1, \dots, x_n) \tag{17}$$

$$v' = \frac{v}{\sqrt{\|v\|_2^2 + \varepsilon}} \tag{18}$$

Donde:

v, es la concatenación de los valores cada histograma que conforma un bloque

x_1, es el valor del histograma inicial

x_n, es el valor del histograma final

v', es la expresión para la normalización de cada uno de los histogramas a nivel de bloque

$\|v\|_2$, es la norma *L2* del vector v

ε, es un valor muy pequeño para evitar división por cero en el caso de que la intensidad sea constante a lo largo del bloque

La noma de un vector se expresa como la raíz cuadrada de la suma de todos los componentes del vector, como se ve en la siguiente ecuación:

$$\|v\|_2 = \sqrt{\sum x_i^2} \tag{19}$$

Donde:

x_i, son cada histograma correspondiente a un bloque

Cada normalización será diferente ya que depende de la región de celdas, en cuanto a la redundancia del solapamiento se conseguirá una representación más robusta. Todo dependerá de los parámetros mencionados a lo largo de la sección, estos parámetros habitualmente funcionan bien en muchas de las aplicaciones, pero se pueden configurar según la necesidad. Una manera de saber la dimensión final del descriptor HOG, es decir, cuantos componentes tiene el vector se calcula a partir de la siguiente ecuación (21), con la ecuación (20) se calcula el número de bloques en la imagen en sentido horizontal y horizontal:

$$n°bloques = n°celdas - n°celdas/bloque + 1 \tag{20}$$

$$n = n°bloques \times n°celdas/bloque \times n°intervalos\_histograma \tag{21}$$

Donde:

$n°bloques$, número de bloques posibles en una imagen

$n°celdas$, número de celdas a lo largo del eje

$n°intervalos\_histograma$, número de intervalos de los histogramas en cada una de las celdas

$n°celdas/bloque$, es el número de celdas por bloque

El descriptor HOG es entonces una parte esencial para la detección de objetos, el próximo paso se enfoca en la clasificación del detector (Valveny, HOG - Cálculo del descriptor, 2015).

2.3. Algoritmo de reconocimiento SVM

La combinación del descriptor HOG y el clasificador SVM junto con pasos complementarios permite la construcción de un detector de objetos que será utilizado como algoritmo de reconocimiento de señas.

2.3.1. SVM para clasificación

Es un método que se lo conoce inicialmente por ser un clasificador binario lineal, es decir, existe una única frontera que hace distinción a dos categorías, que según el caso será representada por una línea o un hiperplano.

Actualmente, existe la posibilidad de hacer distinción multiclase o a su vez que exista el caso que no se puede separar linealmente los conjuntos, para aquello la explicación parte de su formulación inicial en su parte matemática donde se realiza la adaptación correspondiente, pero dado que este no es nuestro tema de estudio solo se detallará matemáticamente la formulación básica a continuación (Universitat Autònoma de Barcelona, 2019).

2.3.2. SVM Lineal

Las características principales de las Máquinas de Vectores de Soporte se basan en determinar el margen máximo entre las dos clases y para encontrar esta solución se ayudan a partir de vectores de soporte. En la Figura 19. Explicación SVM sobre conceptos básicos se encuentra dos conjuntos de datos definidos como clase A y clase B que corresponden a la información que será clasificada, SVM al ser un modelo discriminativo se determina un hiperplano solución que divide el espacio de características en dos regiones disjuntas, esta solución se basa en la optimización de la distancia máxima entre los hiperplanos de los vectores de soporte de la clase A y clase B, en la figura se aprecia como la región intermedia entre las líneas entrecortadas. Los vectores de soporte

son aquellas muestras particulares que ayudan a cumplir con la condición de margen máximo, gráficamente se encuentran diferenciadas como se aprecia.

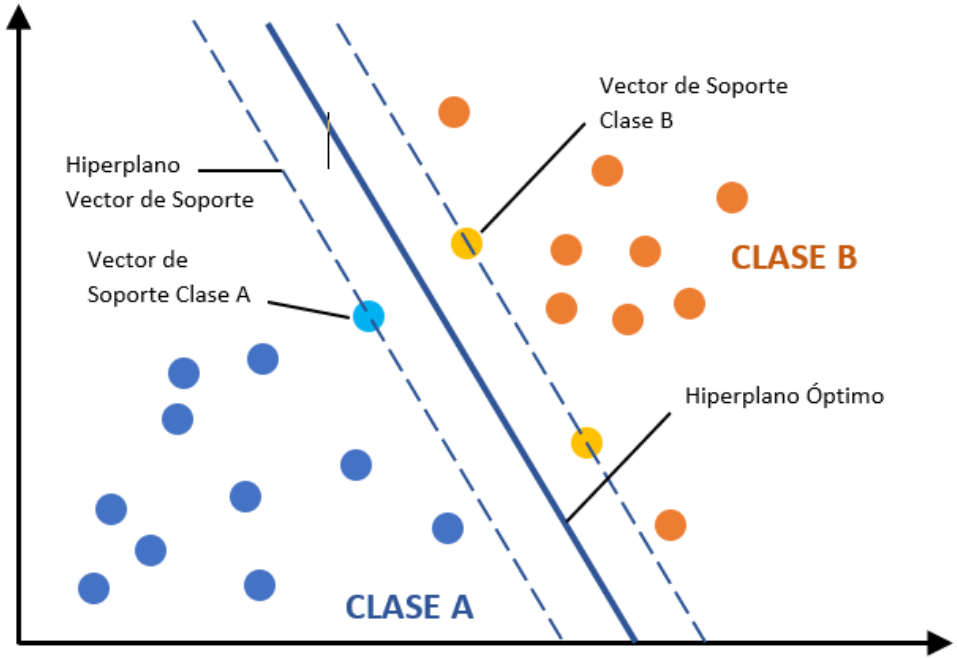

Figura 19. Explicación SVM sobre conceptos básicos

Matemáticamente, el hiperplano solución se lo expresa a partir del vector,

$$w^T x_i + b = 0 \tag{22}$$

Donde:

w, es el vector ortogonal al hiperplano solución

T, es la terminología que indica el producto escalar entre dos vectores

x_i, son los puntos sobre el hiperplano que satisfacen la expresión

b, es el coeficiente de intersección

Para encontrar esta solución se obtiene como el hiperplano medio de los hiperplanos de los vectores de soporte que se denominan como H^+ y H^-, en la Figura 20 se aprecia también las distancias entre el hiperplano solución y los hiperplanos H^+ y H^-, estas son d^+ y d^-.

$$H^+ \rightarrow w^T x_i + b = +1 \quad (23)$$

$$H^- \rightarrow w^T x_i + b = -1 \quad (24)$$

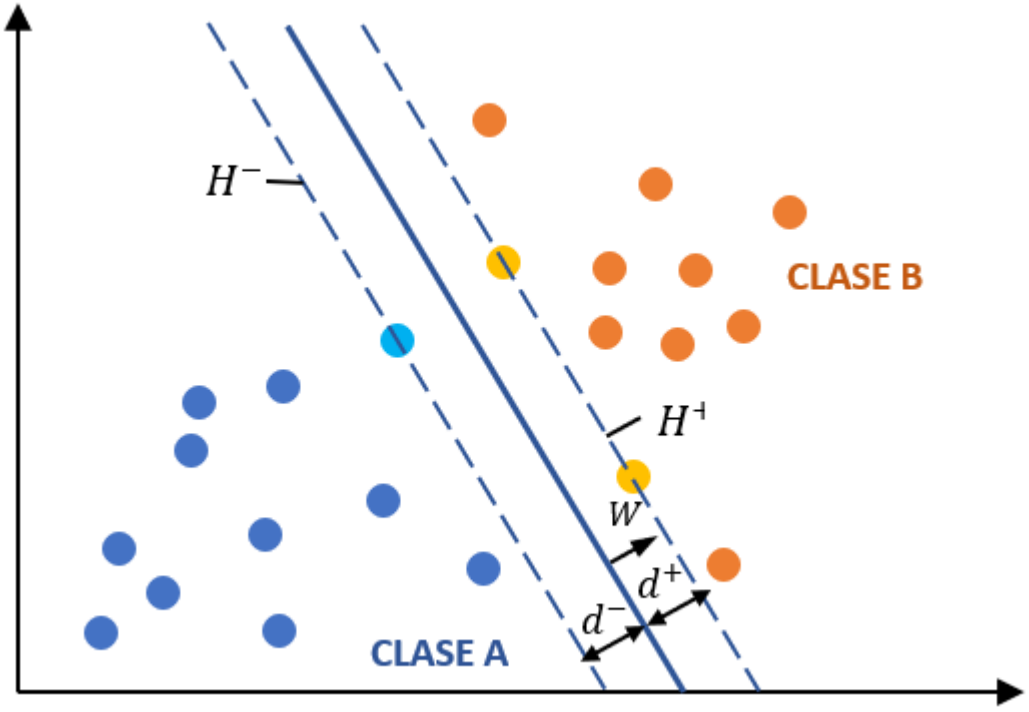

Figura 20. Representación de hiperplanos de vectores de soporte y el hiperplano de solución

De esta forma, al aplicar los parámetros del hiperplano w y b a las muestras de la clase B cumplen con la condición,

$$w^T x_i + b \geq +1 \quad (25)$$

En cambio, la condición que se cumple para las muestras de la clase A es la siguiente,

$$w x_i + b \leq -1 \quad (26)$$

Estas dos condiciones se pueden reducir a una sola expresión introduciendo una variable y_j que corresponderá a valor positivo o negativo respecto a la región, y de esta forma queda expresada la condición de clasificación,

$$y_j(w^T x_i + b) \geq 1 \tag{27}$$

Para el cálculo de las distancias d^+ y d^- es a partir de los dos hiperplanos H,

$$d^+ = d^- = \frac{|wx + b|}{\|w\|} = \frac{1}{\|w\|} \tag{28}$$

Donde $\|w\|$ corresponde a la norma euclídea del vector W, desarrollando matemáticamente el $margen$ va a depender únicamente del parámetro w,

$$margen = d^+ + d^- = 2\frac{1}{\|w\|} \tag{29}$$

La solución del SVM consiste en obtener los valores w y b que corresponde a la solución del hiperplano, se resuelve entonces como problema de optimización cuadrática donde la función Φ que depende únicamente de w,

$$\Phi(w) = \frac{1}{2}w^T w \tag{30}$$

La cual está sujeta a la condición de clasificación (27) se transforma entonces en un problema de optimización cuadrática, se plantea entonces una función auxiliar Lagrangiano, (31), que se construye con la suma de las funciones a optimizar,(32), más las restricciones, (33), que está sujeto el problema, estas restricciones son multiplicadas por factores α siendo estos los multiplicadores de Lagrange, se expresa entonces de la siguiente manera:

$$L(x, \alpha) = f(x) + \sum_i \alpha g_i(x) \quad \forall \alpha_i \geq 0 \tag{31}$$

$$f(x) \rightarrow \Phi(w) \tag{32}$$

$$g_i(x) \rightarrow y_j(w^T x_i + b) \geq 1 \tag{33}$$

Donde:

α, son los multiplicadores de Lagrange

$f(x)$, es la función a optimizar

$g_i(x)$, la función de condición

La solución del problema se obtiene con la minimización del lagrangiano con respecto a w y b como se muestra a continuación,

$$L(w, b, \alpha_i) = \frac{1}{2} w^T w - \sum_i \alpha_i [y_j(w^T x_i + b) - 1] \tag{34}$$

$$\frac{\partial L}{\partial w} = w - \sum_i \alpha_i y_i x_i = 0 \quad \Rightarrow \quad w = \sum_i \alpha_i y_i x_i \tag{35}$$

$$\frac{\partial L}{\partial b} = -\sum_i \alpha_i y_i = 0 \quad \Rightarrow \quad \sum_i \alpha_i y_i = 0 \quad \alpha_i \geq 0 \tag{36}$$

$$L(w, b, \alpha_i) = \frac{1}{2}\left(\sum_i \alpha_i y_i x_i\right)^T \left(\sum_j \alpha_j y_j x_j\right) - \sum_i \alpha_i y_i \left(\sum_j \alpha_j y_j x_j\right)^T x_i + \sum_i \alpha_i \tag{37}$$

$$L(w, b, \alpha_i) = -\frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j x_i x_j + \sum_i \alpha_i \tag{38}$$

De esta manera se plantea una nueva función (39) cuya maximización depende de la condición obtenida en (36), lo cual resulta en la solución del *Support Vector Machine* y es un problema de optimización denominado SVM Dual,

$$\max_{\alpha} \Theta(\alpha) = -\frac{1}{2}\sum_{i}\sum_{j}\alpha_i\alpha_j y_i y_j x_i x_j + \sum_{i}\alpha_i \tag{39}$$

De esta resolución se rescata ciertas propiedades que resultan favorables para adaptaciones de SVM en aquellos casos que la combinación no sea linealmente separable por lo cual se usa el concepto de los kernels.

Se observa en (35) viene dado por la combinación lineal de las muestras de entrenamiento asociados por multiplicadores de Lagrange, por tanto, los vectores de soporte son las muestras constituidas por los multiplicadores mayores de cero (Universidad Autónoma de Barcelona, 2018).

2.4. Modelo estadístico PCA

Con la finalidad de mejorar la detección y reducir la complejidad computacional la aplicación del algoritmo PCA se adopta en este trabajo, el cual trabaja como un discriminador del vector de características producto del algoritmo HOG. PCA es un modelo de análisis estadístico multivariado que selecciona las variables más importantes descartando redundancias en la información original, su propiedad más destacable es la reducción de dimensiones de las características resultado en una expresión más definida.

Dado un número de muestras N en un tamaño de imagen $M = m \times n$, y los conjuntos de muestras de entrenamiento es $X = \{x_1, x_2, \dots, x_N\}$, entonces la media de la muestra de entrenamiento es,

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{40}$$

Siendo que existe diferencias entre cada imagen de entrenamiento y la media será (41) , el interés es encontrar variaciones en los componentes principales, estos son los valores propios que son los valores aplicados a cada componente (The MathWorks, Inc, s.f.), tal es, la matriz de covarianza de las muestras de entrenamiento,

$$A = [x_1 - \bar{x}, x_2 - \bar{x}, \dots, x_N - \bar{x}] \tag{41}$$

$$S = \sum_{i=1}^{N} (x_i - \bar{x})(x_i - \bar{x})^T = A\,A^T \tag{42}$$

Donde:

A, es una matriz que contiene la diferencia entre cada muestra de entrenamiento con su valor medio

Como la matriz de covarianza es una matriz simétrica real, sus valores propios son números reales que se ordenan de mayor a menor, siendo el valor máximo la principal dirección (Jiang & Xiong, 2012).

2.5. Matriz de Confusión

Por último, se necesita una herramienta para visualizar el rendimiento del clasificador, se supondrá dos categorías “palabra correcta” y la “palabra incorrecta”, lo cual será una matriz 2x2 que representará el comportamiento de los resultados clasificados sobre estas categorías. Se aprecia en la Figura 21 dos etiquetas, las instancias reales versus las predicciones de la clasificación, lo que deja a cuatro posibles resultados.

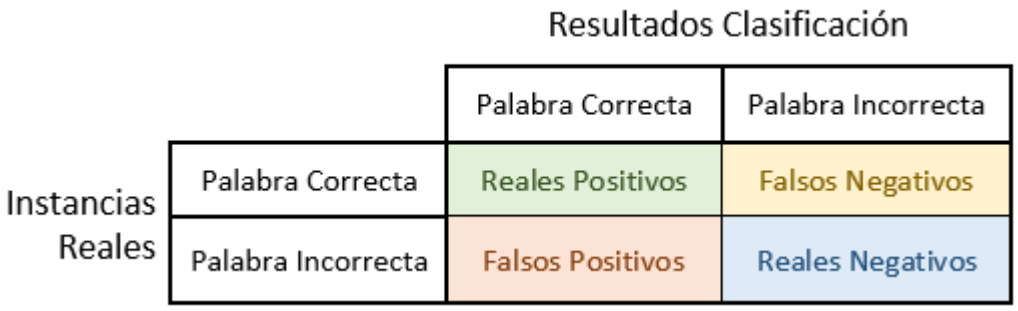

Figura 21. Matriz de Confusión conceptos básicos

El primer caso, reales positivos representa los candidatos que son correctamente clasificados, el segundo caso, falsos positivos representa los candidatos que han sido incorrectamente clasificados, el tercer caso, los falsos negativos representan los candidatos que son "palabras correctas" pero han sido incorrectamente clasificados, y el último caso, reales negativos representan los candidatos que han sido clasificados correctamente como "palabra incorrecta".

La importancia de este cuadro radica en el análisis que se obtendrá, estas serán la exactitud, la precisión, la sensibilidad y la especificidad.

$exactitud$ se refiere a la proximidad entre el resultado y la clasificación exacta,

$$exactitud = \frac{Reales\ Positivos + Reales\ Negativos}{Predicciones\ Totales} \tag{43}$$

Mientas que la $precisión$ se refiere a la calidad de respuesta del clasificador,

$$precisión = \frac{Reales\ Positivos}{Reales\ Positivos + Falsos\ Positivos} \tag{44}$$

La $sensibilidad$ se refiere a la medida de la eficiencia en la clasificación de todos los elementos de una categoría,

$$sensibilidad = \frac{Número\ Reales\ Positivos}{Número\ Reales\ Positivos + Numero\ Falsos\ Negativos} \tag{45}$$

Y la $especificiddad$ evalúa la eficiencia en la clasificación de todos los elementos que no pertenecen a la categoría,

$$especificiddad = \frac{Número\ Reales\ Negativos}{Número\ Reales\ Negativos + Numero\ Falsos\ Positivos} \quad (46)$$

Cabe mencionar que a la medida de sensibilidad se la relaciona como la tasa de reales positivos o por sus siglas en inglés TPR, en cambio, a la tasa de falsos positivos se la expresa como $(1 - especifidad)$ siendo su abreviatura en inglés FPR (Universidad Autónoma de Barcelona, 2018).

CAPITULO III

3. CLASIFICACIÓN DATOS CON SVM

En este capítulo se describe las etapas de desarrollo para el reconocimiento y clasificación de las palabras de la base de datos. Se tomó como modelo base la metodología del sistema de reconocimiento de (Hamed, Belal, & Mahar, 2016), cabe aclarar que la base de datos con la que se cuenta es diferente, por lo que se ajustó la técnica de reconocimiento con el fin de mejorar resultados en el proyecto y se ejecuta el procedimiento de la Figura 22.

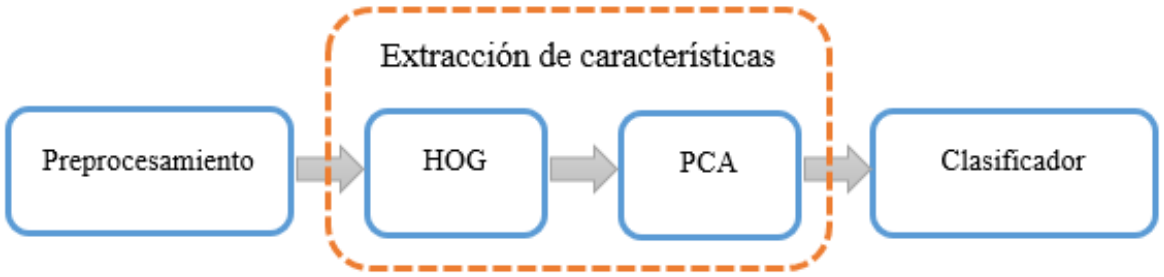

Figura 22. Diagrama del sistema para clasificación de muestras

3.1. Lengua de Señas Ecuatoriano (LSEC)

El lenguaje de comunicación para las personas con discapacidad auditiva se presenta como la lengua de señas que se distingue según la ubicación geográfica, existen por ejemplo lengua de señas americana que corresponden a Norteamerica, así como también la lengua de señas arábica presente en Arabia, de igual manera, en el Ecuador corresponde una lengua de señas oficial ecuatoriana (Voskresensky & Ivanova, 1999).

En general, la lengua de señas se trata de un conjunto de posturas de los miembros corporales superiores, y movimientos de las manos y/o dedos para la formación de una palabra o frase, la lengua de señas no presenta un cuerpo gramatical, y regionalmente se atribuyen variantes en ciertas expresiones (Consejo Nacional para la Igualdad de Discapacidades [CONADIS], s.f.).

3.2. Palabras y letras de la base de datos

Para los elementos de la base de datos se hace referencia al trabajo realizado por (Hu & Teran, 2017) el cual cuenta con 5 categorías cada una cuenta con 10 tipos diferentes de palabras/frases, y además el abecedario (A-Z) dactilológico español ecuatoriano, se explica a continuación las características que cuentan los elementos de la base de datos.

3.2.1. Características de las palabras

La base de datos cuenta con un total de 50 palabras/frases con diferentes repeticiones realizadas por 25 personas entre estos docentes y estudiantes del Instituto Nacional de Audición y Lenguaje (INAL), en la Tabla 1 se presenta las palabras correspondientes a la base de datos.

Tabla 1
Palabras y Frases de la Base de Datos

Adjetivos	Alimentos	Colores	Juguetes&Cosas	Saludos
Alto	Huevo	Amarillo	Avión	Buenos días
Bajo	Leche	Azul	Barco	Buenas noches
Bonito	Limón	Blanco	Carro	Buenas tardes
Feo	Manzana	Café	Casa	Chao
Grande	Naranja	Gris	Mesa	Como estas
Limpio	Pan	Morado	Muñeco	Disculpa
Nuevo	Papaya	Negro	Pelota	Gracias
Pequeño	Pera	Rojo	Reloj	Hola
Sucio	Plátano	Rosado	Silla	Mucho gusto
Viejo	Sandia	Verde	Tren	Por favor

Se destaca que existe diversidad en cuanto a los sujetos que realizan las repeticiones, se diferencian por edad, género y color de piel, en edad se establece un rango de entre los 14 y 45

años de edad, en género son 11 hombres y 14 mujeres, y en cuanto a color de piel se denotan variedad, este último factor se tomará en cuenta más adelante.

Para la formación de la palabra o frase involucra el movimiento de brazos, manos y dedos, en general, el movimiento predomina más con el uso del miembro izquierdo o la combinación de ambos miembros.

El número de muestras realizadas en el trabajo de (Hu & Teran, 2017) para las palaras son un total de 420 las cuales se distribuyen en 6 repeticiones con diferentes sujetos para cada palabra, salvo en la categoría Saludos donde las repeticiones son 18 para cada palabra con 10 sujetos diferentes.

3.2.2. Características de las letras

La siguiente parte de la base de datos cuenta con las muestras del abecedario del lenguaje dactilológico español ecuatoriano que se aprecia en la Tabla 2.

Tabla 2
Abecedario Dactilológico Español Ecuatoriano

Letras					
A	E	J	N	R	V
B	F	K	Ñ	RR	W
C	G	L	O	S	X
CH	H	LL	P	T	Y
D	I	M	Q	U	Z

Este abecedario difiere del español tradicional en letras adicionales como la “CH”, “LL” y la “RR”, estas dos últimas, sin embargo, se forman con la misma postura de las letras individuales “L” y “R” respectivamente más un movimiento adicional para su distinción.

Se considera dos grupos de letras, ya sean en su escritura mayúsculas o minúsculas, el primer grupo carece de movimiento y se las denominará letras estáticas, el segundo grupo como se mencionó en el caso de la “LL” y “RR” presentan un movimiento de la mano y/o dedos como la “J”, “Ñ” y “Z”.

El número de muestras corresponde a 4 repeticiones por letra dando un total de 120 muestras, en cuanto a los sujetos que realizaron las señas, fueron realizados por 4 personas diferentes que se dividen en 2 mujeres y 2 hombres.

El presente trabajo enfatiza la aplicación del algoritmo en las muestras que se las define por el movimiento dinámico de los miembros superiores, más no se concentrará en las señas estáticas por la gran cantidad de trabajos que ya abarcan esta la problemática con señas estáticas como es el trabajo de (Feng & Yuan , 2013), (Carneiro, y otros, 2016), (Dong, Leu, & Yin, 2015) y otros.

3.3. Información general de la base de datos

Se describe entonces aspectos generales de los elementos de la base de datos, inicialmente se cuenta con que todas las muestras se encuentran en formato .mat, lo que facilita la carga de datos en lenguaje MATLAB.

Existen tres tipos de elementos que son videos en escala de color o RGB, videos de profundidad o Depth y el *skeleton tracking* mismos que fueron recolectados por medio de un módulo físico Kinect v2. Se debe anotar que tanto el video RGB como el de profundidad trabajan con imágenes de 8 bits y 16 bits respectivamente, sin embargo, para reducir requisitos de memoria se almacena las imágenes utilizando el tipo de clase numérico 4-D *uint8* y 4-D *uint16* (The MathWorks, s.f.).

Usando esta base de datos se implementará el algoritmo de reconocimiento haciendo uso de los *frames* característicos de los videos RGB y de profundidad correspondientes a la muestra. Después de un análisis de los datos se observó entonces que existe diferencia en la cantidad de *frames* de los videos de palabras con respecto a los videos del abecedario, un video correspondiente a una palabra cuenta con 100 *frames*, en cambio, el video de una letra está formado de 40 *frames,* la razón radica en el tiempo de ejecución de la seña.

La ejecución de una letra precisa menor tiempo, aproximadamente una persona tarda 2 segundos en realizar la seña en general. Sin embargo, el movimiento y gesto corporal para formular una palabra varía según la persona y la palabra por sí misma.

El último aspecto a considerar previo al proceso de reconocimiento es el entorno en el cual se realizaron las grabaciones, existen 3 escenarios como se puede ver en la Figura 23, se puede ver que la luminosidad no es constante y el enfoque de los sensores varían en algunos puntos.

Figura 23. Diferentes escenarios encontrados en las muestras

Estos factores serán corregidos como proceso previo al reconocimiento, el cual se explicará en el siguiente enumerado.

3.4. Preprocesamiento de información

En este apartado se detalla los ajustes realizados para la posterior extracción de características resultado de la combinación de los elementos RGB y *Depth.*

Se inicia con el redimensionamiento de tamaño de los *frames* de profundidad con respecto a los *frames* RGB para la superposición de ambas imágenes, continua con la segmentación de la imagen resultante, lo que se conoce como eliminación del *background* donde solo queda únicamente el cuerpo de la persona, y, por último, la selección de *frames* de interés evitando así una saturación en memoria de datos redundantes a la hora de la clasificación

3.4.1. Redimensionamiento de imágenes

Se parte detallando las dimensiones características de los *frames* RGB y los *frames Depth* en la Tabla 3.

Tabla 3
Detalle tamaño de frames

Dimensión	RGB	Depth
Fila	480	424
Columna	640	512

Se aprecia que la dimensión de los *frames* Depth es menor, esto se debe a las especificaciones de calibración dispuesta originalmente en los sensores del dispositivo Kinect v2.

El ajuste por lo cual se realiza a estos elementos para obtener la misma dimensión que los *frames* RGB y luego realizar el recorte del área de interés.

Se presenta una diferencia de tamaños, la información contenida no es la misma, en la Figura 24,**¡Error! No se encuentra el origen de la referencia.** el *frame* Depth presenta un mayor

enfoque del entorno de grabación, al contrario de los *frames* RGB que engloban mayormente al sujeto.

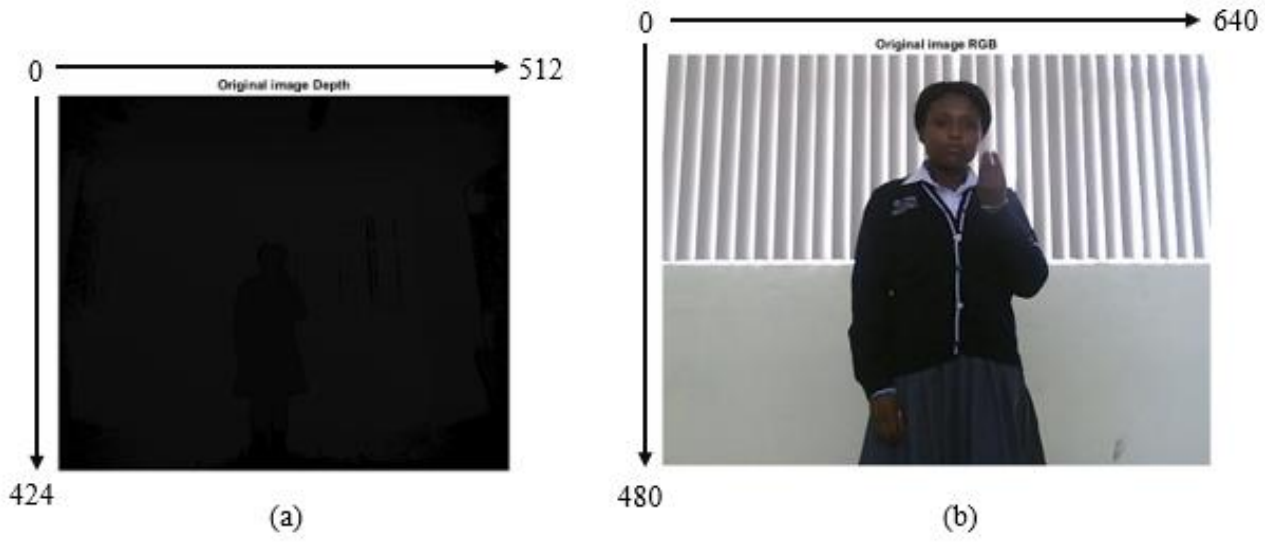

Figura 24. Diferencias frame Deph vs. frame RGB

Para la redimensión se tiene un *frame Depth* como la Figura 24.a al cual se le aplica un zoom de 2.4, este el valor que se obtuvo a partir de pruebas para tener la mayor similitud con el *frame* RGB; el siguiente código describe la adquisición del valor de las dimensiones de la imagen en ambos ejes, se define las medidas para el recorte de la imagen y la aplicación del factor zoom a la imagen, el código completo estará en el anexo donde se encontrarán los siguientes términos:

`InPicture`, corresponde a la imagen Depth.

`xSize`, es el número de filas de la imagen de entrada.

`ySize`, es el número de columnas de la imagen de entrada.

`ZoomFactor`, es el zoom de valor 2.4 que se aplica a la imagen.

`xCrop`, es la medida a ser recortada a lo largo del eje x.

`yCrop`, es la medida a ser recortada a lo largo del eje y.

`zoomPicture`, es la imagen resultante al aplicar zoom a la imagen de entrada.

Definido el acercamiento de la imagen se procede con el recorte del mismo bajo las siguientes consideraciones, se determinó que el enfoque del sensor de profundidad difiere del enfoque del sensor RGB para la alineación de los *frames* Depth y se especifica 5 posiciones para la alineación de los *frames*, este factor se genera posiblemente debido un desajuste en la calibración de los sensores del dispositivo Kinect v2.

Las 5 posiciones se definieron a partir del análisis y pruebas de todo la base de datos y se determinó que estas fueron suficientes para el ajuste, se presenta un ejemplo de los pasos realizados en la Figura 25.

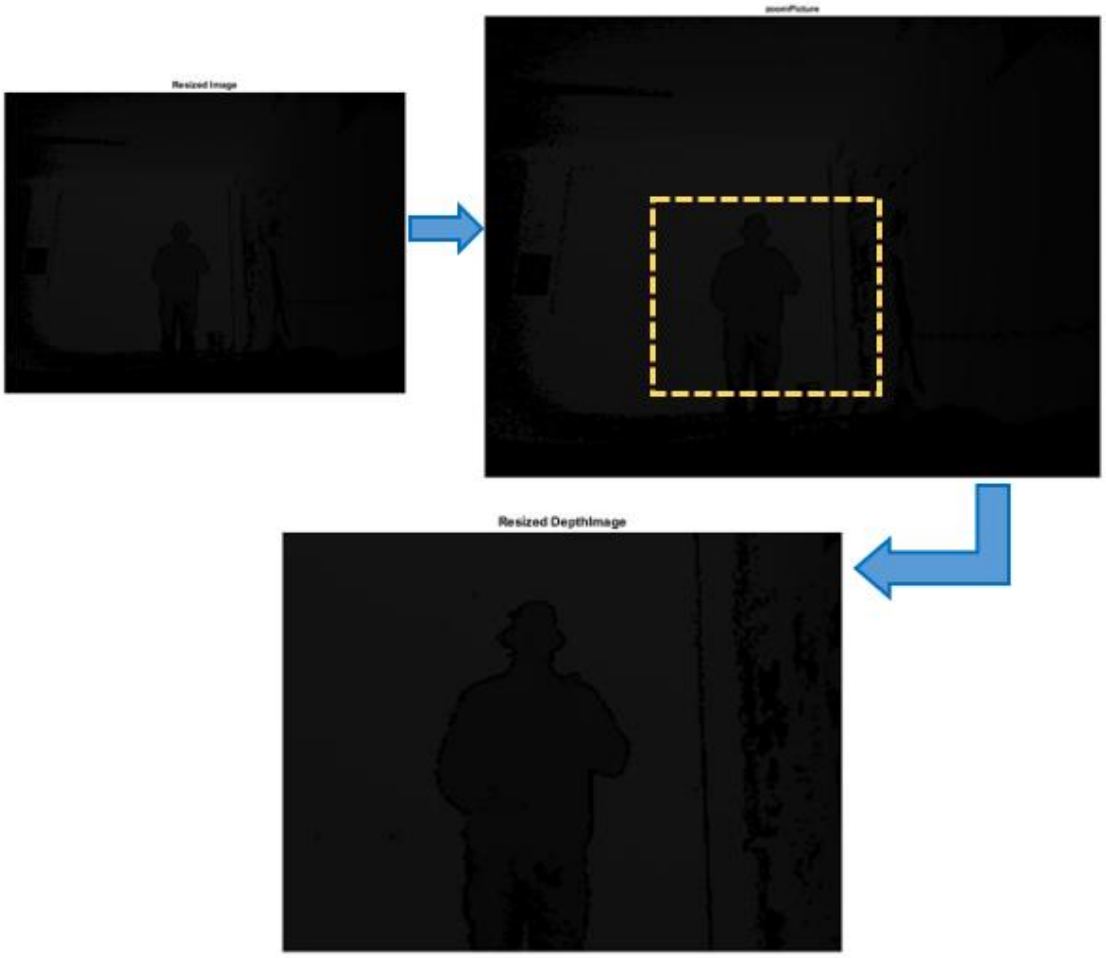

Figura 25. Ejemplo de redimensionamiento y alineación de imagen *depth*

En la Figura 25 se puede ver que se parte con la imagen inicial *Depth*, seguidamente se le aplica el factor zoom 2.4 y se obtiene una nueva imagen. En esta nueva imagen se define una ventana que se deslizará según la posición correspondiente, esta ventana se aprecia de color amarillo con línea entrecortada y para sus dimensiones se definió el ancho y el largo de la imagen a partir de las variables `xCrop` y `yCrop`.

Para determinar la ubicación de esta ventana se realizó varias pruebas para todas las grabaciones y se establece los puntos iniciales y finales en base a 5 posiciones, esta ventana a la final contendrá la imagen resultante que coincide con la imagen RGB. En definitiva, este

procedimiento es esencial para proseguir con la segmentación de los *frames* y conseguir una alineación lo más cercana.

El código a continuación realiza la alineación de las imágenes con la formación de la ventana a partir de la posición correspondiente a la imagen, código estará en el anexo.

Donde `OutPicture` es resultado de la imagen después de ser recortada, independiente de cada alineamiento que se tome la dimensión final de la imagen será $480x640$.

3.4.2. Segmentación de máscara

Una vez definido la imagen recortada se hace la transformación de la imagen de profundidad en una imagen en escala de grises al tomar un canal de color, y seguido se binariza la imagen para conseguir la segmentación de la persona, la imagen resultante se ve en la Figura 26, donde el área de interés es la parte color blanco.

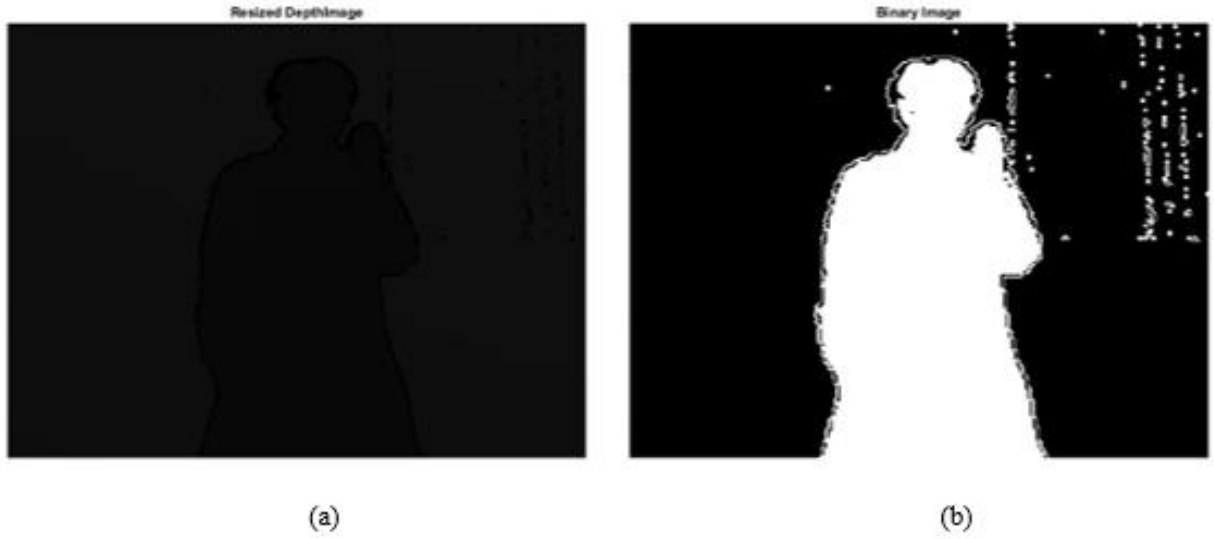

Figura 26. Imagen Depth Recortada e imagen segmentada

El código abajo convierte la imagen *depth* definida con la variable `grayImage` a escala de grises y después se binariza obteniendo solo los valores de los pixeles entre un rango de 1000 y 3500 que representan la intensidad de los pixeles que forman al sujeto.

Parte del proceso del proyecto (Hamed, Belal, & Mahar, 2016) es la segmentación de manos y rostro de la persona, sin embargo, se decidió omitir este paso debido a las diferentes tonalidades de color de piel, se puede ver en la Figura 27, este factor afectaba el resultado de reconocimiento ya que eliminaba parcial o completamente partes corporales.

Figura 27. Muestra de diferentes tonalidades de piel

Es importante definir un modelo efectivo que introduzca cualquier tipo de color de piel y al segmentar la imagen discrimine ropa y demás elementos que no son parte de la formulación de la seña.

3.4.3. Operación AND

A partir de la imagen segmentada se hace la combinación con la imagen a color para obtener finalmente la silueta de la persona, se aprecia el resultado en Figura 28 la imagen final.

Figura 28. *Imagen Depth&RGB*

Se debe considerar tres pasos para conseguir la combinación de las imágenes *depth* y RGB, en el código a continuación se explican en tres partes este procedimiento.

Primero, se hace la división de canales *red, green* y *blue* de la imagen `RGBImageResiz`.

Segundo, la operación lógica se realiza mediante el comando `immultiply`, dicho comando multiplica cada elemento de las matrices con su correspondiente elemento, en este caso sería cada canal de la imagen RGB con la imagen *Depth* binarizada.

Por último, la reconstrucción de la imagen resultante mediante la concatenación de los tres canales `red_c`, `green_c` y `blue_c`.

El código se presentará en el anexo y las acciones realizadas en las secciones 3.4.1, 3.4.2 y 3.4.3 se aplican a cada *frame* significativo o *frame* de interés.

3.4.4. *Frames* de Interés

Como se mencionó anteriormente cada video cuenta con un número de *frames* según el dato, en el caso de las palabras es de 100 *frames* y en el caso de las letras del abecedario es de 40 *frames,* si bien el conjunto de *frames* forman el signo se puede descartar un gran número de estos.

Durante las observaciones se pudo determinar un número de *frames* específicos que logran representar al seña, esta acción reduce la redundancia de *frames,* ya que muchos son similares y con esto, a su vez, se logra aminorar el tiempo de procesamiento y carga computacional.

Se determinó que para la selección de *frames* resulta mejor hacer la observación a partir de los videos a color, se rescata entonces 11 *frames* a lo largo del video, se ve en la Tabla 4 dichos frames.

Tabla 4
Frames de Interés de videos RGB

N° Frame	Frame Interés
1	10
2	16
3	22
4	28
5	34
6	40
7	46
8	52
9	58
10	64
11	83

Sin embargo, al momento de verificar los mismos en los videos de profundidad no resultan ser las siluetas de las imágenes correspondientes, este fenómeno se debe a una pérdida de paquetes

que se deriva en la comunicación de los sensores al hacer la captura de información y ser enviada hacia el ordenador.

Este no es la única pérdida de información que se observa, dentro de la imagen se aprecia huecos, que se nota a lo largo de la silueta de la persona en la Figura 28 como resultado del ruido, pero este factor no llega a afectar el reconocimiento y se trabaja sin dificultad.

Se realizaron entonces varias pruebas para determinar un posible patrón de *frames* específicos correspondientes al video de profundidad que se ajusten con los de color y lleguen a cumplir con la segmentación deseada, en la Tabla 5 se muestra los *frames* resultantes para cada video.

Tabla 5

Frames Específicos

***Frames* RGB**	***Frames* Depth**
10	10
16	16
22	22
28	28
34	34
40	48
46	52
52	58
58	72
64	76
83	83

A pesar de que este patrón se adapta a la mayoría de las palabras, la variación de *frames* depende en casi todos los casos del tiempo de ejecución de la seña, y su variabilidad es mayormente en el cuarto, quinto y sexto *frame*.

3.5. Algoritmo de reconocimiento

Una vez efectuado los ajustes con la imagen final se transforma a escala de grises para ejecutar el comando `extractHOGFeatures`, este comando extrae características del histograma de gradientes orientados de una imagen en escala de grises, como resultado se obtiene un vector característico con la información de forma local de regiones dentro de la imagen (The MathWorks, Inc., 2013). En la Figura 29 se ve representado los gradientes de la imagen.

Figura 29. Imagen en escala de grises y aplicada el descriptor HOG

La dimensión del vector característico HOG es de 1x167796, lo cual es el resultado del número de celdas multiplicado por el número de bloques que recorre toda la imagen.

Y si se recopila en un solo vector la información de los 11 *frames* específicos este da un valor excesivo para la etapa de clasificación.

Se vio una opción inicial para reducir el tamaño del vector el cual fue segmentar el área de análisis en manos y rostro, esto derivó en complicaciones con las diferentes tonalidades de piel, y a pesar de definir un área menor el vector característico final seguía siendo excesivo para el análisis del clasificador.

La solución entonces fue la aplicación de una técnica estadística PCA sin la necesidad de segmentar partes del cuerpo. El código abajo realiza la agrupación de los datos HOG `data` de cada *frame* en una matriz `Xn` correspondiente a una palabra, después se aplica el comando `pca` para cada palabra `Xn` para obtener los valores de los principales componentes en diferentes representaciones como su `score`,`latent` y `explained`. Por último, agrupar los valores `latent` en una matriz `Yn`.

Se muestra en el anexo, y se determina los términos importantes,

`Xn`, es la matriz de datos HOG de los *frames* de cada palabra expresados como `data`.

`score`, es una matriz conformada por las observaciones de cada componente.

`latent`, corresponde a los datos a ingresar al clasificador ya que son las varianzas de los componentes principales.

`explained`, son los valores de `latent` explicados en forma porcentual.

Se hará un ejemplo que describe el análisis del PCA, dado un conjunto de 5 palabras distintas donde cada palabra se caracteriza con 11 *frames* y cada *frame* se obtuvo el vector característico HOG previamente.

Se agrupa los vectores característicos `data` en una sola matriz que será de $11x167796$ de una palabra y se ejecuta el comando `pca`, se obtiene que, a partir del `score` calculado existe variabilidad de los datos, estos datos corresponden a información procedente de los 10

componentes principales que abreviadamente se denominarán PC, en sus siglas en inglés. Idealmente el modelo PCA realiza una reducción de dimensiones, esta reducción ayuda a explicar la información mayormente en sus 3 primeros componentes que suelen representar el 80% o más de la información, visualmente sería como la Figura 30.

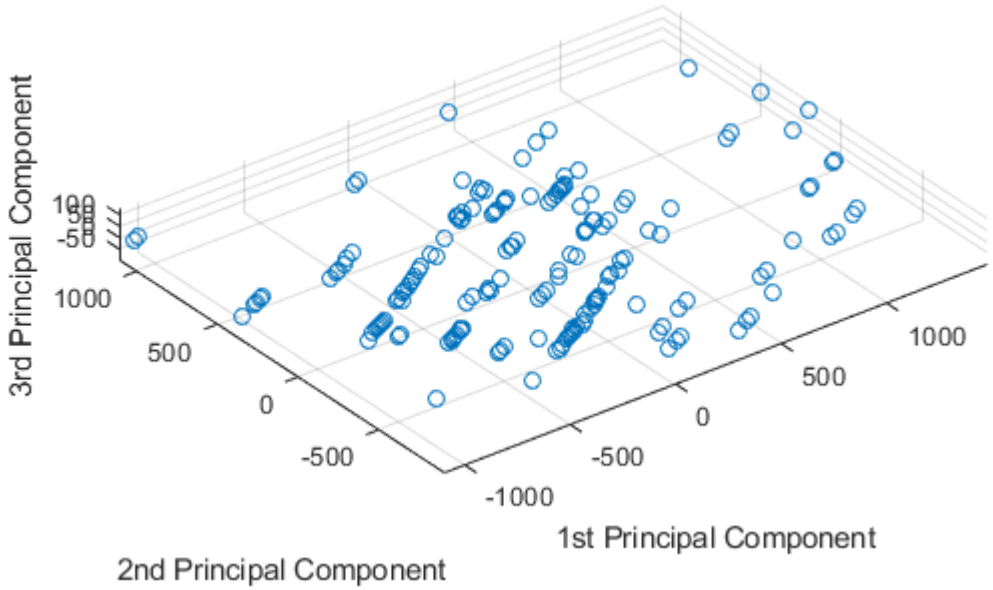

Figura 30. Representación PCA en el espacio de los 3PCs

Fuente: (The MathWorks, Inc, 2012)

En el caso de este trabajo, el resultado de los `score` correspondientes a los 10 componentes principales se representó gráficamente los 3 PCs más importantes y se obtuvo la Figura 31, se ve que no existe una concentración importante en alguno de los tres componentes, a diferencia de la Figura 30, lo que indica que se necesita una inspección del resto de los 7 componentes, para lo cual se hará uso de los valores en `explained`.

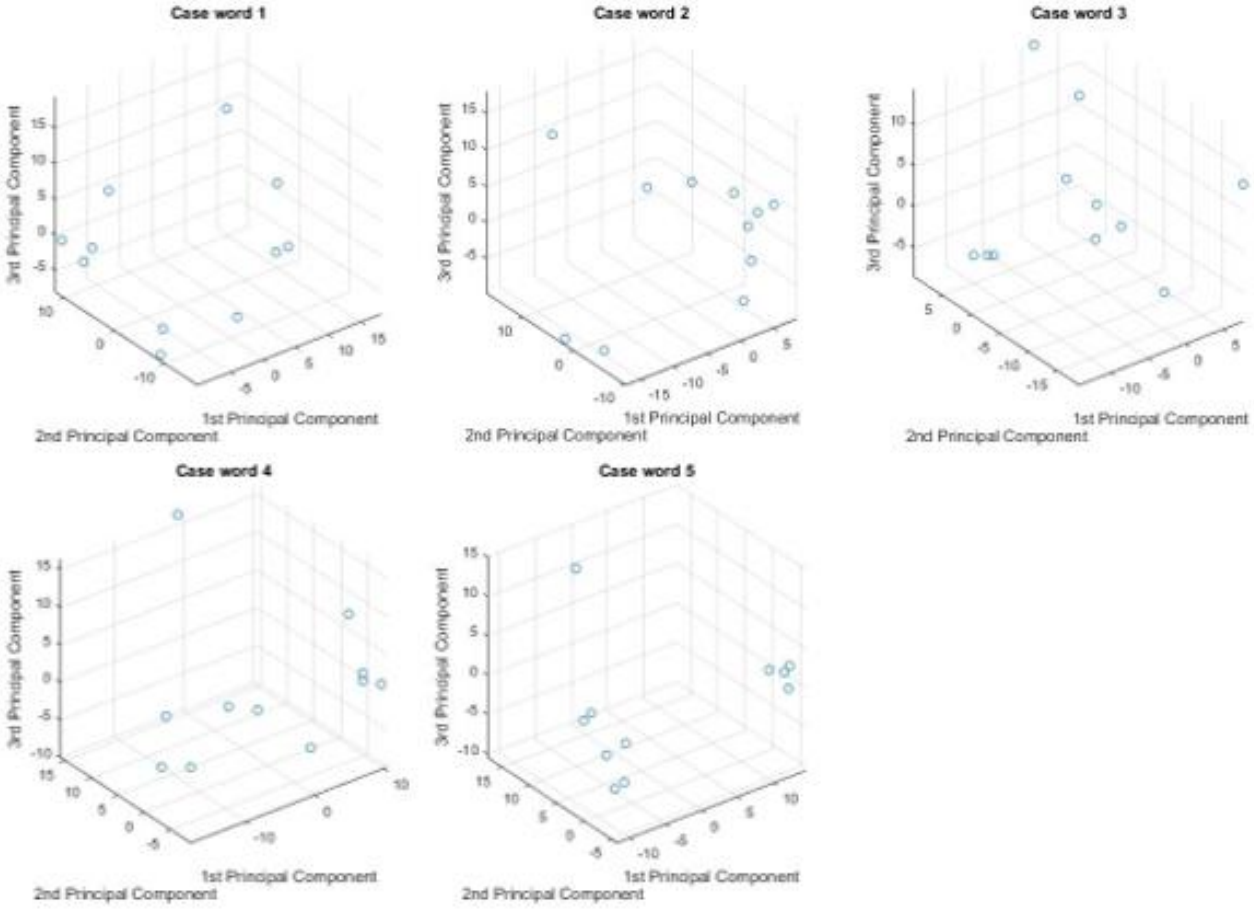

Figura 31. PCA-scores correspondientes a 5 palabras

Dado que la representación anterior de los 3 PCs no basta se hace el análisis entonces de los 10 componentes principales que en conjunto contribuyen a una mayor comprensión de los datos y su aporte en forma de porcentajes.

En la Figura 32 se define diferentes curvas para las diferentes palabras, la curva representa a cada componente y su correspondiente variación entre ellas con su equivalente en porcentaje. Se aprecia que la suma de los tres primeros PCs no representan en su totalidad la información.

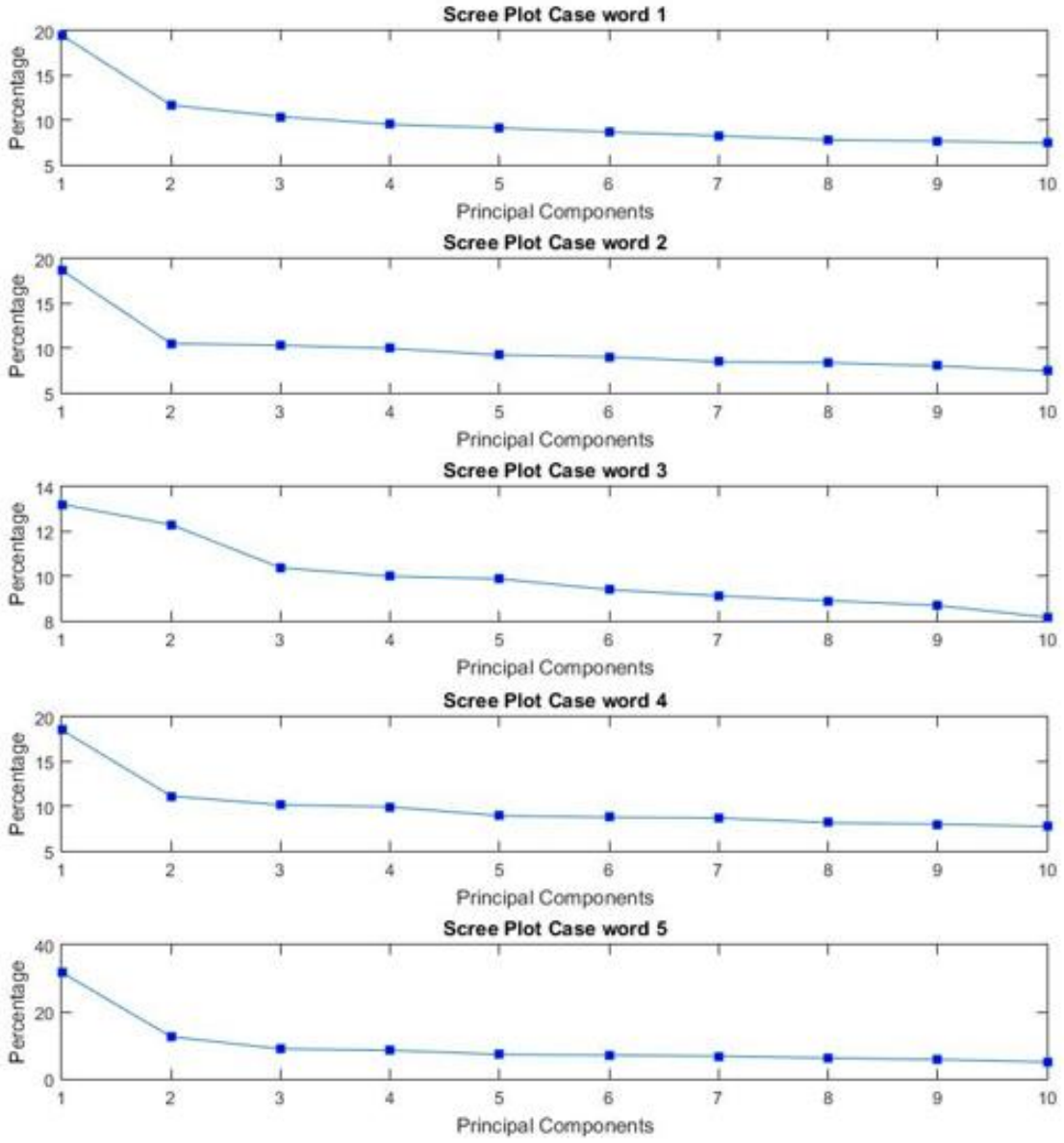

Figura 32. *Scree plot para diagnóstico de variación de cada componente*

Se observa que los PC1 captura un porcentaje menor al 50% de variación y a partir de estos los porcentajes de los siguientes componentes decrecen, lo que implica que para la clasificación se deberá usar los datos correspondientes a todos los componentes con la finalidad de evitar pérdidas de información no deseadas.

A pesar de no cumplir con la curva ideal, lo único que indica es que PCA no es la mejor manera de visualizar los datos ya que sus tres principales componentes no describen en su mayoría el comportamiento de los datos, sin embargo, se puede hacer uso de su todos los componentes para la clasificación a continuación.

3.6. Clasificación de Información MATLAB

Para esta última etapa del reconocimiento se hace uso de la aplicación *Classification Learner* de MATLAB, este cuenta con diferentes modelos de clasificador entre los cuales se halla el algoritmo clasificador SVM mismo que será seleccionado.

La interfaz se ve en la Figura 33, el algoritmo de SVM realiza un aprendizaje automático al cual se suministra la información deseada y en respuesta otorga el diagnóstico final de los datos por medio de la matriz de confusión y la generación del modelo de entrenamiento para la respuesta de predicciones de nuevos datos.

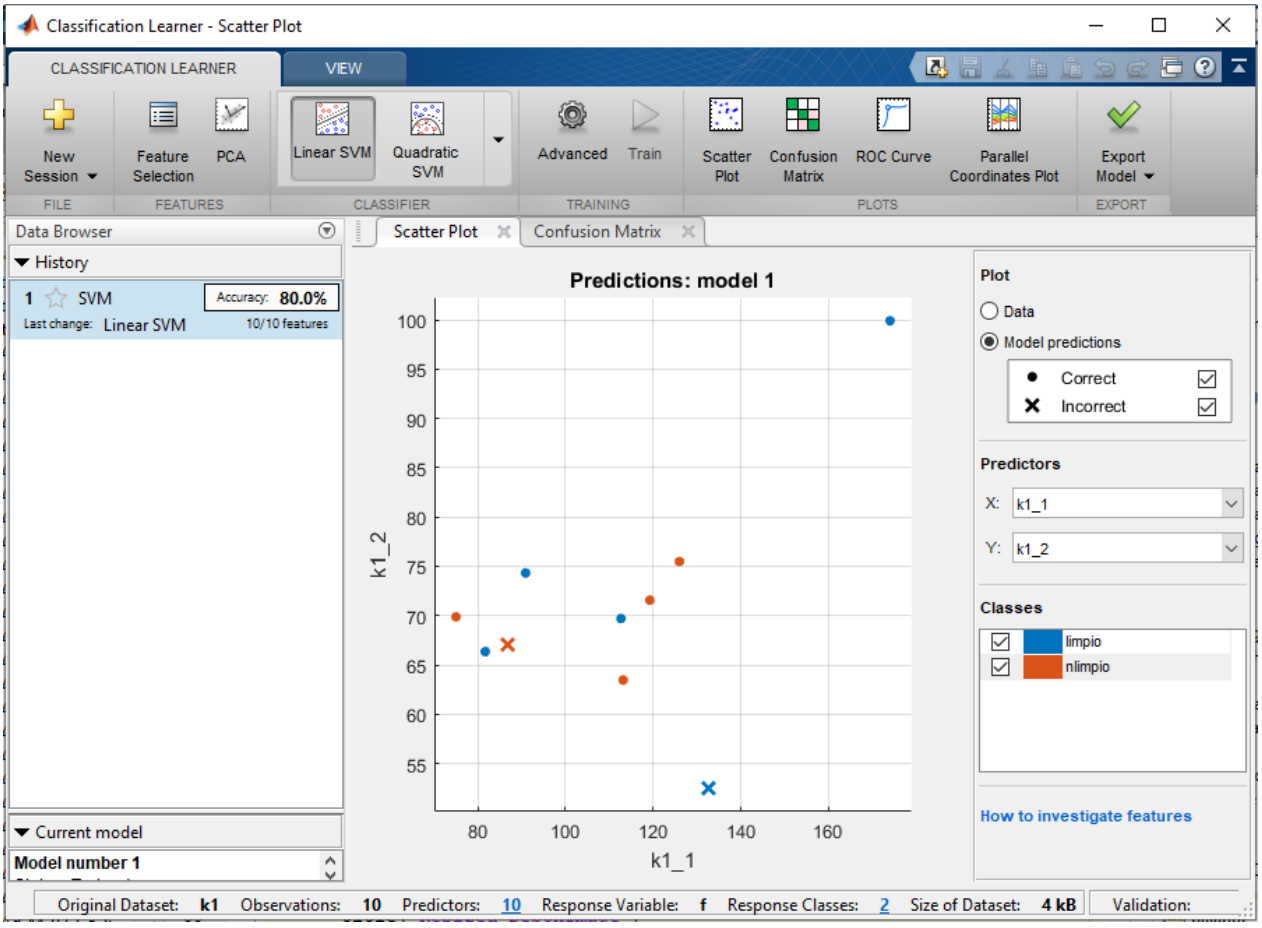

Figura 33. *Interfaz Classification Learner de MATLAB*

Se ingresa entonces una tabla de valores con las muestras positivas y negativas con carga balanceada, esto quiere decir, se va a tener como entrada de datos 5 palabras “limpio” y 5 palabras “no limpio”, ver Figura *34* , el contenido de la variable `k1` corresponde a los valores `latent` y `f` es la denominación general.

10x2 table

	1 k1										2 f
1	86.8186	67.0629	60.2968	57.6250	52.7644	49.8398	49.5690	45.7314	42.8903	40.4123	limpio
2	90.8945	74.3256	52.4599	47.4598	46.1340	40.1804	38.6345	37.4918	33.8771	31.4100	limpio
3	81.7015	66.3506	47.1047	44.2746	40.5053	37.9632	36.8799	32.5860	31.2365	30.4190	limpio
4	112.6623	69.7104	57.8410	47.2776	44.6433	37.5071	37.4548	36.0922	34.0839	31.2518	limpio
5	174.0084	99.9147	68.0301	63.2074	52.4651	45.0566	43.7192	38.8909	35.3191	31.5025	limpio
6	125.9618	75.4888	67.0753	61.4530	58.9538	55.9739	53.1952	50.3362	49.4325	47.9537	nlimpio
7	113.1489	63.4613	62.3201	60.2486	55.7087	54.4944	51.1757	50.5285	48.4027	44.9725	nlimpio
8	75.0453	69.8786	58.9111	56.7547	56.0896	53.3505	51.7522	50.5415	49.3310	46.2937	nlimpio
9	119.2115	71.5745	65.2572	63.7826	57.4441	56.2678	55.8636	52.4273	51.3393	49.8301	nlimpio
10	132.6264	52.4900	37.2907	35.6200	30.4816	29.4661	28.2918	25.9756	24.0136	21.0696	nlimpio

Figura 34. *Valores de entrada tipo table para la etapa de clasificación*

Se hace la elección del modelo de clasificación, en este caso será *Linear SVM*, el algoritmo calcula la precisión del modelo clasificador y como parte de la interfaz presenta la matriz de confusión para entender aciertos y fallos que se aprecia en la Figura 35.

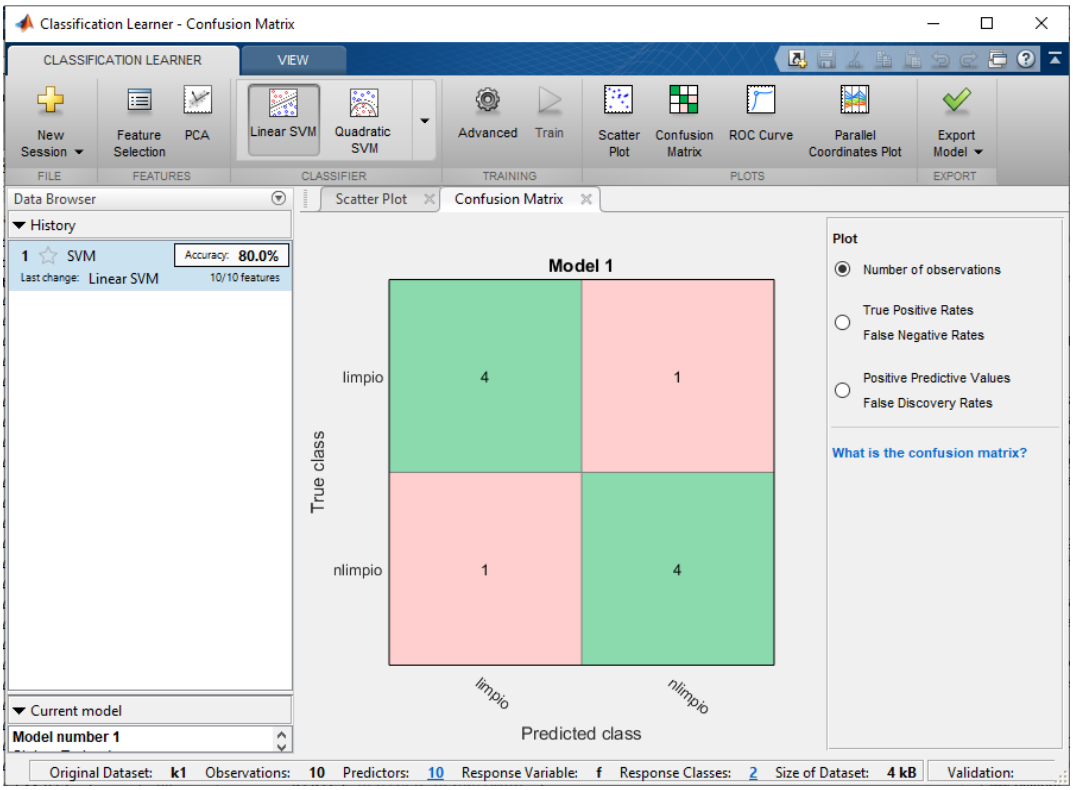

Figura 35. *Selección de modelo clasificador y obtención de matiz de confusión*

Logrado el entrenamiento se hace la exportación del modelo y se hace el reconocimiento con nuevos datos, mediante el siguiente código que se genera al exportar el modelo:

```
kt1=table(k11);
kt1.Properties.VariableNames = {'k1'};
yfit = trainedClassifierk1.predictFcn(kt1)
```

Donde `k11` corresponde al nuevo dato a predecir, este, por tanto, tiene que tener las mismas características que las muestras entrenadas por lo que sus valores corresponden al vector `latent`.

Al final, el resultado de la predicción es positiva bajo el modelo de entrenamiento seleccionado.

En resumen, en este capítulo se desarrolló la aplicación de un algoritmo de reconocimiento basado en la extracción de características HOG para los datos de color y profundidad de una base de datos proporcionada inicialmente, donde estos datos pasaron por una etapa previa de procesamiento donde se realizó un ajuste en dimensionalidad y alineación entre imágenes para obtener una segmentación de la persona, y finalmente, empleando el modelo estadístico PCA para reducir la dimensionalidad del vector característico HOG, ser clasificados en base un modelo SVM.

Se hace entonces el análisis de resultados para diferentes pruebas con distintas palabras seleccionadas de la base de datos en el siguiente capítulo.

CAPITULO IV

4. PRUEBAS Y RESULTADOS

En el capítulo anterior se describió el procedimiento del entrenamiento del sistema, en este capítulo se explica la ejecución de las pruebas realizadas y se hace el análisis de los resultados obtenidos con el fin de determinar cuál es el mejor predictor para el presente trabajo.

En la primera sección se hace una explicación de las muestras de entrenamiento que han sido divididas como candidatos positivos y candidatos negativos, los candidatos positivos se tratarán entonces del conjunto de elementos conocidos a ser entrenados mientras que los candidatos negativos son los elementos no conocidos. Y la segunda sección se describe los resultados obtenidos de cada prueba y el análisis correspondiente.

4.1. Muestras de entrenamiento

Se ha explicado en el inciso 3.2 acerca de las características de la base de datos y su contenido, para la etapa de clasificación se escogió 10 palabras/frases de la base de datos, estas serán las muestras de entrenamiento.

Hay que tomar en cuenta que cada palabra/frase cuenta con un número de repeticiones determinado según la categoría a la que pertenece ya que se hará el análisis respectivo de cada muestra.

4.1.1. Descripción realización de pruebas

Dentro de las 50 palabras que forman parte de la base de datos se usó 2 palabras correspondiente a cada categoría, a estas muestras se las denominará como los candidatos positivos y se detallan en la Tabla 6.

Tabla 6
Palabras y Frases de la Base de Datos

Adjetivos	Alimentos	Colores	Juguetes&Cosas	Saludos
Grande	Huevo	Blanco	Mesa	Buenos días
Limpio	Leche	Negro	Tren	Gracias

Las palabras de las cuatro primeras categorías tienen 6 repeticiones por palabra, para el algoritmo de entrenamiento se utilizará 5 candidatos positivos y 5 candidatos negativos, la muestra restante se la aplica para la predicción y se irá alternando las muestras para obtener los resultados.

Para la última categoría las repeticiones por palabra son 18 de cada una, por lo que se usará 10 candidatos para el entrenamiento y los 8 restantes para la predicción, en la Figura 36 se muestra la selección de muestras.

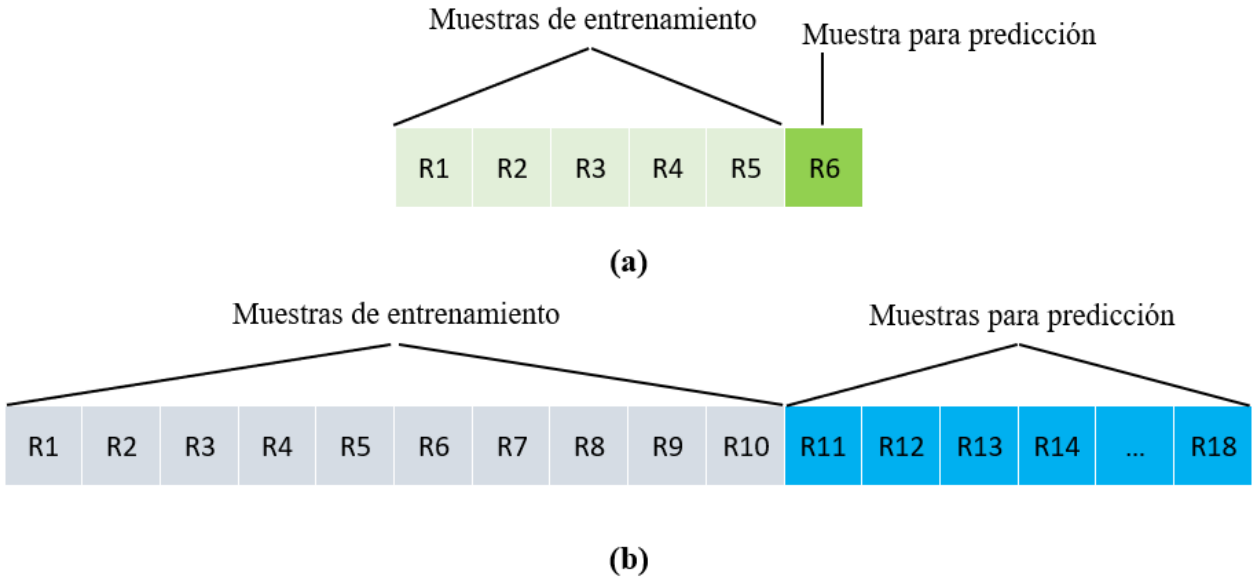

Figura 36. Muestras de entrenamiento y para predicción (a) para 6 repeticiones y (b) para 18 repeticiones

Para el conjunto de los candidatos negativos se hizo una elección aleatoria de todo el conjunto de palabras eliminando las palabras a ser predichas, este grupo será utilizado de manera constante para todo el entrenamiento.

A continuación, en la Figura 37 se presenta la forma en que se realizará las pruebas para las muestras con 6 repeticiones.

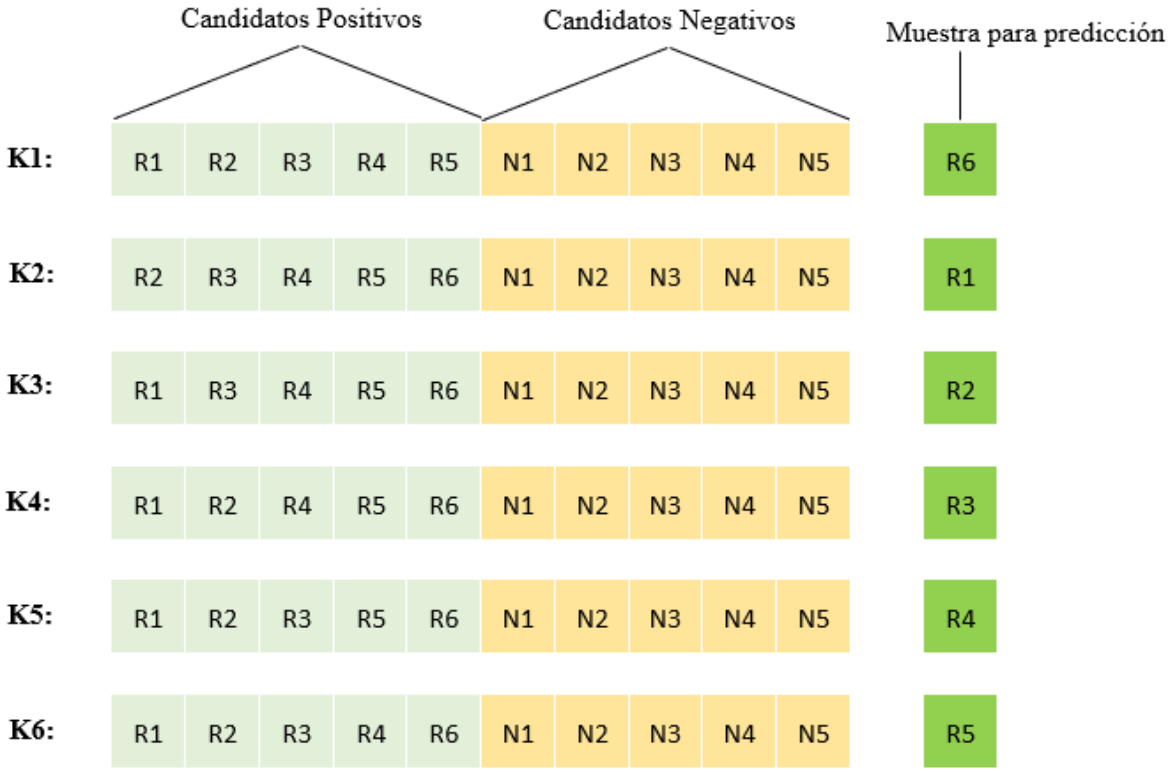

Figura 37. Método para las pruebas de las muestras de entrenamiento con 6 repeticiones

Para las pruebas de las muestras con 18 repeticiones se realiza como en la Figura 38

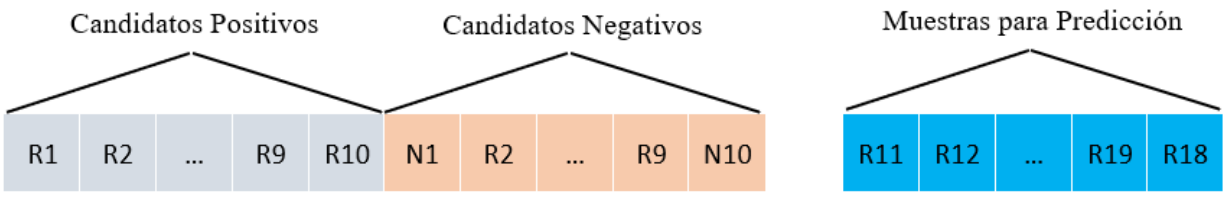

Figura 38. Método para las pruebas de las muestras de entrenamiento con 18 repeticiones

4.1.2. Observaciones base de datos

Existen palabras dentro de la base de datos que no serán tomadas en cuenta para el análisis, se presentan los siguientes inconvenientes:

- Realización de seña con el miembro contrario, es decir, de las señas que se realizan con un miembro en su mayoría han sido ejecutadas con el miembro izquierdo, pero existen pocos casos donde se encontró personas ejecutando la seña con el miembro derecho.
- Parte de mano o brazo fuera del margen de grabación, es decir, durante la grabación el enfoque del sensor no abarca manos o brazos durante la ejecución de la seña provocando que se recorte la imagen de la persona de su miembro.
- Destiempo de grabación o desincronización ejecución de seña con inicio de grabación, es decir, la grabación no inicia con la ejecución de la seña o finaliza antes de terminar de ejecutar la seña.
- Ejecución diferente de seña para cada persona, es decir, dentro de las repeticiones de una palabra/frase algunas palabras no corresponde con la ejecución de la seña ya que no son los mismos movimientos.

Detallado los candidatos que ingresan al clasificador y sus especificaciones respectivas se procede a hacer las pruebas.

4.2. Prueba 1

El desarrollo de resultados para la palabra “limpio” se obtuvo utilizando el modelo clasificador SVM con una validación cruzada de 10, se determina para los 6 entrenamientos,

denominados como K1, K2, K3, K4, K5 y K6, los porcentajes de precisión según el tipo de clasificador SVM, se muestra en la Tabla 7 los resultados.

Tabla 7
Prueba: limpio

Tipo SVM	K1	K2	K3	K4	K5	K6	Media (%)
Linear (%)	80	80	70	70	70	70	73.33
Quadratic (%)	80	90	90	90	90	80	86.67
Cubic (%)	70	90	70	70	70	60	71.67
Fine Gaussian (%)	40	60	40	40	50	20	41.67
Medium Gaussian (%)	70	80	70	70	70	80	73.33
Coarse Gaussian (%)	0	0	0	0	0	0	0

El mejor promedio del porcentaje de precisión del clasificador para la palabra “limpio” es de 86.67%, con un modelo clasificador tipo cuadrático.

A continuación, para la etapa de predicción se utiliza la repetición restante, los aciertos y fallos de cada entrenamiento utilizando el modelo clasificador cuadrático se ve en la Tabla 8.

Tabla 8
Predicción: limpio

Detalle	K1	K2	K3	K4	K5	K6
Predicción	Correcta	Incorrecta	Correcta	Correcta	Correcta	Correcta

El porcentaje de aciertos es de 83.33% y de fallos 16.67% para un modelo de entrenamiento SVM tipo cuadrático.

4.3. Prueba 2

El desarrollo de resultados para la palabra “grande” se obtuvo utilizando el modelo clasificador SVM con una validación cruzada de 10, se determina para los 6 entrenamientos,

denominados como K1, K2, K3, K4, K5 y K6, los porcentajes de precisión según el tipo de clasificador SVM, se muestra en la Tabla 9 los resultados.

Tabla 9
Prueba: grande

Tipo SVM	K1	K2	K3	K4	K5	K6	Media (%)
Linear (%)	60	80	60	40	60	50	58.33
Quadratic (%)	50	50	70	40	40	30	45.67
Cubic (%)	60	50	60	50	60	60	56.67
Fine Gaussian (%)	50	20	20	20	20	10	23.33
Medium Gaussian (%)	60	60	60	0	0	40	36.67
Coarse Gaussian (%)	0	0	0	0	0	0	0

El mejor promedio del porcentaje de precisión del clasificador para la palabra “grande” es de 58.33%, con un modelo clasificador tipo lineal.

A continuación, para la etapa de predicción se utiliza la repetición restante, los aciertos y fallos de cada entrenamiento utilizando el modelo clasificador lineal se ve en la Tabla 10.

Tabla 10
Predicción: grande

Detalle	K1	K2	K3	K4	K5	K6
Predicción	Incorrecta	Incorrecta	Correcta	Correcta	Correcta	Correcta

El porcentaje de aciertos es de 66.67% y de fallos 33.33% para un modelo de entrenamiento SVM tipo lineal.

4.4. Prueba 3

El desarrollo de resultados para la palabra “huevo” se obtuvo utilizando el modelo clasificador SVM con una validación cruzada de 10, se determina para los 6 entrenamientos, denominados

como K1, K2, K3, K4, K5 y K6, los porcentajes de precisión según el tipo de clasificador SVM, se muestra en la Tabla 11 los resultados.

Tabla 11
Prueba: huevo

Tipo SVM	K1	K2	K3	K4	K5	K6	Media (%)
Linear (%)	60	60	80	60	60	80	66.67
Quadratic (%)	60	50	70	50	50	70	58.33
Cubic (%)	80	80	70	70	80	70	75
Fine Gaussian (%)	70	50	60	70	80	60	65
Medium Gaussian (%)	60	60	80	60	70	80	68.33
Coarse Gaussian (%)	0	0	50	0	0	0	8.33

El mejor promedio del porcentaje de precisión del clasificador para la palabra “huevo” es de 75%, con un modelo clasificador tipo cúbico.

A continuación, para la etapa de predicción se utiliza la repetición restante, los aciertos y fallos de cada entrenamiento utilizando el modelo clasificador cúbico se ve en la Tabla 12.

Tabla 12
Predicción: huevo

Detalle	K1	K2	K3	K4	K5	K6
Predicción	Correcta	Correcta	Correcta	Correcta	Correcta	Correcta

El porcentaje de aciertos es de 100% y de fallos 0% para un modelo de entrenamiento SVM tipo cúbico.

4.5. Prueba 4

El desarrollo de resultados para la palabra “leche” se obtuvo utilizando el modelo clasificador SVM con una validación cruzada de 10, se determina para los 6 entrenamientos, denominados

como K1, K2, K3, K4, K5 y K6, los porcentajes de precisión según el tipo de clasificador SVM, se muestra en la Tabla 13 los resultados.

Tabla 13
Prueba: leche

Tipo SVM	K1	K2	K3	K4	K5	K6	Media (%)
Linear (%)	70	60	80	60	70	80	70
Quadratic (%)	20	20	50	40	30	50	35
Cubic (%)	50	50	60	60	50	50	53.33
Fine Gaussian (%)	20	20	40	40	40	40	33.33
Medium Gaussian (%)	60	60	80	60	60	80	66.67
Coarse Gaussian (%)	0	0	0	0	0	0	0

El mejor promedio del porcentaje de precisión del clasificador para la palabra “leche” es de 70%, con un modelo clasificador tipo lineal.

A continuación, para la etapa de predicción se utiliza la repetición restante, los aciertos y fallos de cada entrenamiento utilizando el modelo clasificador lineal se ve en la Tabla 14.

Tabla 14
Predicción: leche

Detalle	K1	K2	K3	K4	K5	K6
Predicción	Correcta	Correcta	Incorrecta	Correcta	Correcta	Incorrecta

El porcentaje de aciertos es de 66.67% y de fallos 33.33% para un modelo de entrenamiento SVM tipo lineal.

4.6. Prueba 5

El desarrollo de resultados para la palabra “blanco” se obtuvo utilizando el modelo clasificador SVM con una validación cruzada de 10, se determina para los 6 entrenamientos,

denominados como K1, K2, K3, K4, K5 y K6, los porcentajes de precisión según el tipo de clasificador SVM, se muestra en la Tabla 15 los resultados.

Tabla 15
Prueba: blanco

Tipo SVM	K1	K2	K3	K4	K5	K6	Media (%)
Linear (%)	70	70	70	70	70	70	70
Quadratic (%)	70	60	70	70	60	70	66.67
Cubic (%)	40	60	60	80	70	50	60
Fine Gaussian (%)	40	30	40	70	40	60	46.67
Medium Gaussian (%)	60	50	60	80	50	80	63.33
Coarse Gaussian (%)	0	0	0	0	0	0	0

El mejor promedio del porcentaje de precisión del clasificador para la palabra “blanco” es de 70%, con un modelo clasificador tipo lineal.

A continuación, para la etapa de predicción se utiliza la repetición restante, los aciertos y fallos de cada entrenamiento utilizando el modelo clasificador lineal se ve en la Tabla 16.

Tabla 16
Predicción: blanco

Detalle	K1	K2	K3	K4	K5	K6
Predicción	Correcta	Correcta	Correcta	Incorrecta	Correcta	Correcta

El porcentaje de aciertos es de 83.33% y de fallos 16.67% para un modelo de entrenamiento SVM tipo lineal.

4.7. Prueba 6

El desarrollo de resultados para la palabra “negro” se obtuvo utilizando el modelo clasificador SVM con una validación cruzada de 10, se determina para los 6 entrenamientos, denominados

como K1, K2, K3, K4, K5 y K6, los porcentajes de precisión según el tipo de clasificador SVM, se muestra en la Tabla 17 los resultados.

Tabla 17
Prueba: negro

Tipo SVM	K1	K2	K3	K4	K5	K6	Media (%)
Linear (%)	50	50	60	70	50	70	58.33
Quadratic (%)	60	50	50	70	50	60	56.67
Cubic (%)	60	60	60	80	60	60	63.33
Fine Gaussian (%)	40	20	20	50	40	40	35
Medium Gaussian (%)	60	70	70	80	70	70	70
Coarse Gaussian (%)	0	0	0	30	0	0	5

El mejor promedio del porcentaje de precisión del clasificador para la palabra “negro” es de 70%, con un modelo clasificador tipo *medium gaussian*.

A continuación, para la etapa de predicción se utiliza la repetición restante, los aciertos y fallos de cada entrenamiento utilizando el modelo clasificador *medium gaussian* se ve en la Tabla 18.

Tabla 18
Predicción: negro

Detalle	K1	K2	K3	K4	K5	K6
Predicción	Correcta	Correcta	Correcta	Incorrecta	Correcta	Correcta

El porcentaje de aciertos es de 83.33% y de fallos 16.67% para un modelo de entrenamiento SVM tipo medium gaussian.

4.8. Prueba 7

El desarrollo de resultados para la palabra “mesa” se obtuvo utilizando el modelo clasificador SVM con una validación cruzada de 10, se determina para los 6 entrenamientos, denominados

como K1, K2, K3, K4, K5 y K6, los porcentajes de precisión según el tipo de clasificador SVM, se muestra en la Tabla 19 los resultados.

Tabla 19
Prueba: mesa

Tipo SVM	K1	K2	K3	K4	K5	K6	Media (%)
Linear (%)	30	50	70	30	40	40	43.33
Quadratic (%)	40	30	50	40	50	40	41.67
Cubic (%)	50	40	50	40	50	50	46.67
Fine Gaussian (%)	30	20	20	30	50	50	33.33
Medium Gaussian (%)	0	0	0	30	50	40	20
Coarse Gaussian (%)	0	0	0	0	0	0	0

El mejor promedio del porcentaje de precisión del clasificador para la palabra “mesa” es de 46.67%, con un modelo clasificador tipo cúbico.

A continuación, para la etapa de predicción se utiliza la repetición restante, los aciertos y fallos de cada entrenamiento utilizando el modelo clasificador cúbico se ve en la Tabla 20.

Tabla 20
Predicción: mesa

Detalle	K1	K2	K3	K4	K5	K6
Predicción	Correcta	Correcta	Incorrecta	Incorrecta	Incorrecta	Incorrecta

El porcentaje de aciertos es de 33.33% y de fallos 66.67% para un modelo de entrenamiento SVM tipo cúbico.

4.9. Prueba 8

El desarrollo de resultados para la palabra “tren” se obtuvo utilizando el modelo clasificador SVM con una validación cruzada de 10, se determina para los 6 entrenamientos, denominados

como K1, K2, K3, K4, K5 y K6, los porcentajes de precisión según el tipo de clasificador SVM, se muestra en la Tabla 21 los resultados.

Tabla 21
Prueba: tren

Tipo SVM	K1	K2	K3	K4	K5	K6	Media (%)
Linear (%)	0	60	0	0	50	70	30
Quadratic (%)	60	50	50	60	70	70	60
Cubic (%)	70	70	70	70	80	70	71.67
Fine Gaussian (%)	20	40	20	50	50	50	38.33
Medium Gaussian (%)	40	60	50	60	70	70	58.33
Coarse Gaussian (%)	0	0	0	0	0	0	0

El mejor promedio del porcentaje de precisión del clasificador para la palabra “tren” es de 71.67%, con un modelo clasificador tipo cúbico.

A continuación, para la etapa de predicción se utiliza la repetición restante, los aciertos y fallos de cada entrenamiento utilizando el modelo clasificador cúbico se ve en la Tabla 22.

Tabla 22
Predicción: tren

Detalle	K1	K2	K3	K4	K5	K6
Predicción	Correcta	Correcta	Correcta	Correcta	Correcta	Incorrecta

El porcentaje de aciertos es de 83.33% y de fallos 16.67% para un modelo de entrenamiento SVM tipo cúbico.

4.10. Prueba 9

El desarrollo de resultados para la palabra “gracias” se obtuvo utilizando el modelo clasificador SVM con una validación cruzada de 20 para el entrenamiento denominado como K, se determina para los datos de 10 repeticiones de dicha palabra, en el entrenamiento se obtuvo un

porcentaje de precisión los porcentajes de precisión según el tipo de clasificador SVM, se muestra en la Tabla 23 los resultados.

Tabla 23
Prueba: gracias

Tipo SVM	**K**
Linear (%)	75
Quadratic (%)	65
Cubic (%)	85
Fine Gaussian (%)	55
Medium Gaussian (%)	65
Coarse Gaussian (%)	25

El mejor porcentaje de precisión del clasificador para la palabra “gracias” es de 85%, con un modelo clasificador tipo cúbico.

A continuación, se visualiza el desempeño del algoritmo de aprendizaje con un modelo clasificador cúbico como se muestra en la Figura 39 por medio de la matriz de confusión.

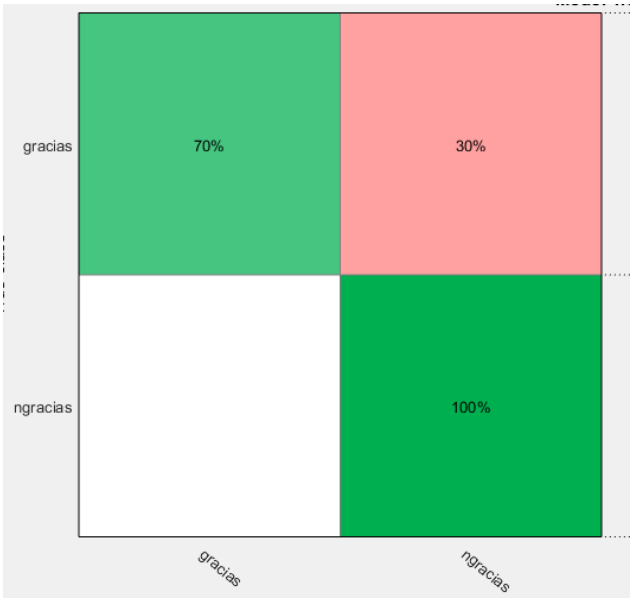

Figura 39. Matriz de Confusión palabra gracias

En cuanto a los resultados de las predicciones para las 8 repeticiones restantes se presentó un porcentaje se muestra en la Tabla 24 un total de aciertos del 75% y un porcentaje de fallas del 25%.

Tabla 24

Predicción: gracias

Detalle	**K**							
Predicción	Correcta	Correcta	Correcta	Correcta	Correcta	Correcta	Incorrecta	Incorrecta

4.11. Prueba 10

El desarrollo de resultados para la palabra “gracias” se obtuvo utilizando el modelo clasificador SVM con una validación cruzada de 20 para el entrenamiento denominado como K, se determina para los datos de 10 repeticiones de dicha palabra, en el entrenamiento se obtuvo un porcentaje de precisión los porcentajes de precisión según el tipo de clasificador SVM, se muestra en la Tabla 25 los resultados.

Tabla 25

Prueba: hola

Tipo SVM	**K**
Linear (%)	85
Quadratic (%)	75
Cubic (%)	80
Fine Gaussian (%)	85
Medium Gaussian (%)	65
Coarse Gaussian (%)	40

El mejor porcentaje de precisión del clasificador para la palabra “gracias” es de 85%, con un modelo clasificador tipo lineal y cúbico.

A continuación, se visualiza el desempeño del algoritmo de aprendizaje con un modelo clasificador lineal como se muestra en la Figura 40 por medio de la matriz de confusión.

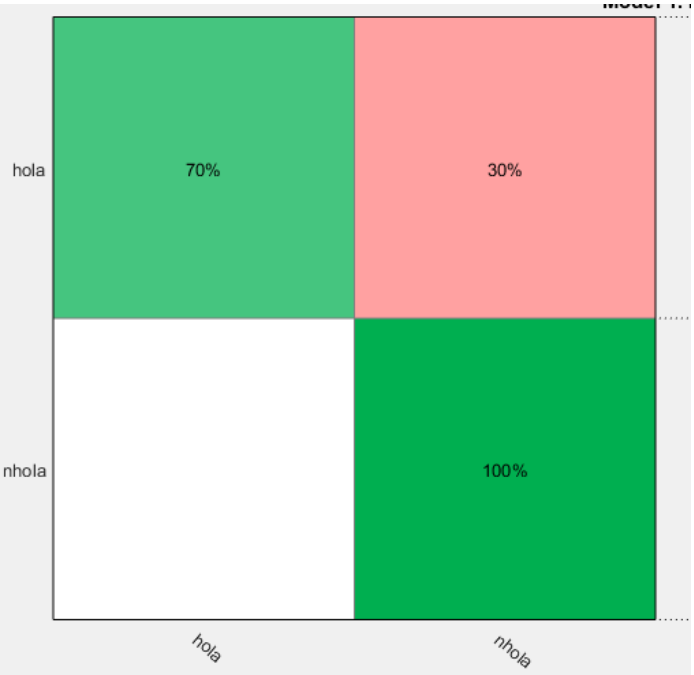

Figura 40. Matriz de Confusión palabra hola

En cuanto a los resultados de las predicciones para las 8 repeticiones restantes se presentó un porcentaje se muestra en la Tabla 26 un total de aciertos del 87.5% y un porcentaje de fallas del 12.5%.

Tabla 26

Predicción: hola

Detalle	**K**							
Predicción	Correcta	Incorrecta	Correcta	Correcta	Correcta	Correcta	Correcta	Correcta

Se concluye que para cada prueba realizada el modelo de clasificador SVM se ajusta particularmente a cada palabra, de estos modelos de los cuales sobresalen el SVM tipo lineal que se ajusta a la palabra “grande”, “leche”, “blanco” y “hola”, y el tipo cúbico que se ajustan a las palabras “huevo”, “mesa”, “tren” y “gracias”. En cuanto a las dos palabras restantes para la palabra “limpio” se tuvo un resultado favorable utilizando el SVM tipo cuadrático y para la palabra “negro” se aplicó favorablemente el SVM tipo *medium gaussian,* se aprecia los

porcentajes en la Tabla 27 a continuación. Se hará la abreviación de los modelos de SVM de la siguiente forma *Linear* (L), *Quadratic* (Q), *Cubic* (C) y *Medium Gaussian* (MG)

Tabla 27

Tipo de SVM y porcentaje de precisión de clasificador según la palabra

Detalles	**Palabras**									
	limpio	grande	huevo	leche	blanco	negro	mesa	tren	gracias	hola
Tipo SVM	Q	L	C	L	L	MG	C	C	C	L/C
% Precisión	86.67	75	75	70	70	70	46.67	71.67	75	75

Para la palabra “hola” se puede ver que cumple con la misma precisión del clasificador ya sea este lineal o cúbico.

Dado que el 50% de las palabras se adaptan mayormente utilizando el modelo clasificador SVM tipo cúbico, se designa entonces como el predictor cúbico el que mejor resultados va a dar como clasificador de las muestras durante el entrenamiento.

CAPITULO V

5. CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

Fue necesario un preprocesamiento de los datos que consisten en la redimensión de la imagen de profundidad y la alineación de la misma con la imagen RGB, esta problemática surge debido a la preparación inadecuado del escenario durante el uso del sensor del dispositivo Kinect v2 a la hora de las grabaciones de los elementos y a la calibración del dispositivo previo o durante la grabación.

Se aplicó el mismo factor de agrandamiento de imagen y, en cuanto a la alineación, se realizó pruebas para toda la base de datos.

Las pruebas de preprocesamiento dieron como resultado la existencia de 5 posiciones que puede corresponder según la persona y el escenario. Esta etapa es necesaria ya que de esta forma se adapta la información para realizar la operación AND y posterior segmentación.

Durante la etapa de segmentación se descartó la segmentación de manos y cabeza, contrario de lo que dictan en trabajos anteriormente referenciados, debido a que esta acción provocaría la pérdida parcial o completa de información ya que se presentan variedad en tonalidad de piel.

Se apreció que en ciertos casos se presentan condiciones en el entorno de la grabación que afectan la detección del individuo, como es la luminosidad y/o tiempo de ejecución acelerada de la seña causando borrosidad del miembro superior.

Se realizó un análisis de los 100 *frames* de los video RGB de cada palabra a fin de determinar la latencia de *frames* significativos que representen la seña sin perder su significado, se resumen en 11 *frames* significativos o de interés que cumplen con este propósito.

A pesar de lo anterior mencionado, durante el emparejamiento de los *frames* RGB con los *frames* de profundidad se encontró desigualdad entre ellos, para lo cual se realizó diferentes pruebas para la corrección de los *frames* de profundidad.

Realizada la extracción de características HOG a la imagen segmentada se propuso la reducción de dimensionalidad del vector mediante la aplicación del modelo estadístico PCA para facilitar el proceso de clasificación durante el tiempo de entrenamiento.

Se determinó que el algoritmo de reconocimiento implementado es fiable, se aprecia durante la realización de las pruebas, donde se ejecutó el modelo de clasificación SVM *Single-class* de MATLAB para el entrenamiento de diferentes palabras, que el mejor porcentaje de precisión es de 85% para un conjunto de 10 datos para entrenamiento y su tasa de predicción llega a ser de 87.5% de aciertos, y el mejor porcentaje de precisión de entrenamiento para 6 datos es de 90% y puede ser mejorado con el aumento de datos para el entrenamiento y su tasa de predicción llega a ser de 83.33% de aciertos, para ambos casos se opta por usar un clasificador tipo cúbico.

5.2. Recomendaciones

Se recomienda realizar mayor cantidad de grabaciones y expandir a nuevas categorías de palabras en conjunto con una persona especializada teniendo en cuenta que durante la grabación el dispositivo debe encontrarse adecuadamente calibrada y en un entorno apropiado.

Se recomienda que durante las grabaciones exista un control de luminosidad solar en medio del entorno ya que puede provocar desfase del nivel de profundidad.

5.3. Trabajos futuros

Como trabajos futuros se propone en base al algoritmo propuesto la integración de técnicas de reconocimiento de miembros corporales mediante en conjunto con la adaptación de equipos con mayores capacidades de procesamiento de la imagen, para la captura de datos durante la formación de oraciones y su interpretación.

Además se propone realizar un intérprete de señas o traductor mediante una aplicación o software que realice el reconocimiento continuo de señas ecuatorianas, y la posibilidad de crear una herramienta didáctica para docentes, alumnos y personas que presenten o no una discapacidad auditiva.

6. BIBLIOGRAFÍA

Aguiar, S., Erazo, A., Romero, S., Garcés, E., Atiencia, V., & Poveda Figueroa, J. (2016). Development of a Smart Glove as a Communication Tool for People with Hearing Impairment and Speech Disorders. *2016 IEEE Ecuador Technical Chapters Meeting (ETCM)*, 1-6. doi:10.1109/ETCM.2016.7750815

Aliyu, S., Mohandes, M., Deriche, M., & Badran, S. (2016). Arabic Sign Language Recognition Using the Microsoft Kinect. *2016 13th International Multi-Conference on Systems, Signals & Devices (SSD)*, 301-306. doi:10.1109/SSD.2016.7473753

Biologico, R. (04 de 12 de 2014). *El ojo humano sí puede ver la luz infrarroja*. Obtenido de BIO-HAZARD.COM: http://www.riesgobioquimico.com/2014/12/el-ojo-humano-si-puede-ver-la-luz.html

Carneiro, S. B., Santos, E. D., Barbosa, T. M., Ferreira, J. O., Alcalá, S. G., & Rocha, A. F. (2016). Static gestures recognition for Brazilian Sign Language with kinect sensor. *2016 IEEE SENSORS*, 1-3. doi:10.1109/ICSENS.2016.7808522

Consejo Nacional para la Igualdad de Discapacidades [CONADIS]. (s.f.). *Diccionario de Lengua de Señas Ecuatoriano "Gabriel Román"*. Obtenido de https://www.consejodiscapacidades.gob.ec/diccionario-de-lengua-de-senas-ecuatoriano-gabriel-roman/

Dai, H. (2018). Research on SVM improved algorithm for large data classification. *2018 IEEE 3rd International Conference on Big Data Analysis (ICBDA)*, 181-185. doi:10.1109/ICBDA.2018.8367673

Dalal, N., & Triggs, B. (2005). Histograms of Oriented Gradients for Human Detection. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), 1*, 886-893. doi:10.1109/CVPR.2005.177

Dilmen, N. (12 de 03 de 2012). *Demonstration of how RGB image split into its three RGB channels. Tricolor, trichromacy, 3x grayscale*. Obtenido de Wikimedia Commons: https://commons.wikimedia.org/wiki/File:Beyoglu_4671_tricolor.png

Dong, C., Leu, M. C., & Yin, Z. (2015). American Sign Language alphabet recognition using Microsoft Kinect. *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 44-52. doi:10.1109/CVPRW.2015.7301347

Feng , K.-p., & Yuan , F. (2013). Static Hand Gesture Recognition Based on HOG Characters and Support Vector Machines. *2013 2nd International Symposium on Instrumentation and Measurement, Sensor Network and Automation (IMSNA)*, 936-938. doi:10.1109/IMSNA.2013.6743432

García Bautista, G., Trujillo Romero, F., & Caballero Morales, S. O. (2017). Mexican Sign Language Recognition Using Kinect and Data Time Warping Algorithm. *2017 International Conference on Electronics, Communications and Computers (CONIELECOMP)*, 1-5. doi:10.1109/CONIELECOMP.2017.7891832

García, G. (2016/2017). *Procesamiento de Imágenes.* Curso, Universidad de Murcia, Informática y Sistemas, Murcia. Obtenido de http://dis.um.es/profesores/ginesgm/

Ghotkar, A. S., & Kharate, G. K. (2015). Dynamic Hand Gesture Recognition and Novel Sentence Interpretation Algorithm for Indian Sign Language Using Microsoft Kinect Sensor. *Jounal of Pattern ecognition Research 1*, 24-38. doi:10.13176/11.626.

Gonzales, R. C., & Woods, R. E. (s.f.). *Digital Image Processing.*

Grau, J. F. (2003). *Técnicas de análisis de imagen: Aplicaciones en Biología* (Vol. 65). Universitat de Valéncia.

Hamed, A., Belal, N. A., & Mahar, K. M. (2016). Arabic sign language alphabet recognition based on HOG-PCA using Microsoft Kinect in complex backgrounds. *2016 IEEE 6th International Conference on Advanced Computing (IACC)*, 451-458. doi:10.1109/IACC.2016.90

He, J., Liu, Z., & Zhang, J. (2016). Chinese Sing Language Recognition based on Trajectory and Hand Shape Features. *2016 Visual Communications and Image Processing (VCIP)*, 1-4. doi:10.1109/VCIP.2016.7805564

Houssein Ahmed, A., Kpalma, K., & Osman Guedi, A. (2017). Human Detection using HOG-SVM, Mixture of Gaussian and Background Contours Subtraction. *2017 13th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, 334-338. doi:10.1109/SITIS.2017.62

Hu, Z. P., & Teran, L. (2017). *Desarrollo de una base de datosdel lenguaje de signos español ecuatoriano a travésdel sensor Kinect V2 de XBOX ONE.* Tesis de pregrado, Universidad de las Fuerzas Armadas - ESPE, Quito.

Jana, A. (2012). *Kinect for Windows SDK Programming Guide.* Packt Publishing Ltd.

Jiang, J., & Xiong, H. (2012). Fast Pedestrian Detection Based on HOG-PCA and Gentle AdaBoost. *2012 International Conference on Computer Science and Service System*, 1819-1822. doi:10.1109/CSSS.2012.453

Leal Narváez, E., Leal Narváez, N., Henríquez Miranda, C., Pichón Pacheco, L., & Romero Martínez, S. (2016). Aplicación integrada a la tecnología Kinect para el reconocimiento e interpretación de la Lengua de Señas Colombianas. *Escenarios, 14*(2), 7 - 19. doi:http://dx.doi.org/10.15665/esc.v14i2.928

López, G. O. (2014). *Implementación de algoritmos de procesamiento de imágenes en FPGA.* Tesis de Grado, Mexico.

Mallick, S. (6 de 12 de 2016). *Histogram of Oriented Gradients*. Obtenido de Learn OpenCV: https://www.learnopencv.com/histogram-of-oriented-gradients/

Miller, F. P., Vandome, A. F., & McBrewster, J. (2013). *Histogram.* Alphascript Publishing.

Moulick, H. N., & Ghosh, M. (2013). IMAGE COMPRESSION USING K-MEANS CLUSTERING AND NUCLEAR MEDICINE IMAGE PROCESSING. *International Journal of Innovative Research in Computer and Communication Engineering, 1*(4).

Naglot, D., & Kulkarni, M. (2016). Real Time Sign Language Recognition using the Leap Motion Controller. *2016 International Conference on Inventive Computation Technologies (ICICT)*, 1-5. doi:10.1109/INVENTIVE.2016.7830097

Pal, D. H., & Kakade, S. M. (2016). Dynamic Hand Gesture Recognition Using Kinect Sensor. *2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC)*, 448-453. doi:10.1109/ICGTSPICC.2016.7955343

Palomino, N. L., & Concha, U. R. (2009). Técnicas de Segmentación en Procesamiento Digital de Imágenes. *Revista de Investigación de Sistemas e Informática, 6*(2), 9-16.

Pliego, A. (s.f.). *Una pantalla en la retina.* Obtenido de CIENCIORAMA: http://www.cienciorama.unam.mx/a/pdf/286_cienciorama.pdf

Sarhan, N. A., EI-Sonbaty, Y., & Youssef, S. M. (2015). HMM-Based Arabic Sign Language Recognition Using Kinect. *2015 Tenth International Conference on Digital Information Management (ICDIM)*, 169-174. doi:10.1109/ICDIM.2015.7381873

Savur, C., & Ferat, S. (2016). American Sign Language Recognition System by Using Surface EMG Signal. *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 002872-002877. doi:10.1109/SMC.2016.7844675

Surhone, L. M., Tennoe, M. T., & Henssonow, S. F. (2010). *Histogram of Oriented Gradients.* Betascript Publishing.

The MathWorks, I. (s.f.). *MathWorks*. Recuperado el 17 de 05 de 2019, de 8-Bit and 16-Bit Images: https://www.mathworks.com/help/matlab/creating_plots/working-with-8-bit-and-16-bit-images.html

The MathWorks, Inc. (2012). *Percent Variability Explained by Principal Components.* Recuperado el 04 de 06 de 2019, de Percent Variability Explained by Principal Components: https://www.mathworks.com/help/stats/pca.html

The MathWorks, Inc. (s.f.). *MathWorks*. Recuperado el 08 de 04 de 2019, de pca: https://www.mathworks.com/help/stats/pca.html#bth9ibe-latent

The MathWorks, Inc. (2013). *Mathworks*. Obtenido de extractHOGFeatures: https://la.mathworks.com/help/vision/ref/extracthogfeatures.html

Universidad Autónoma de Barcelona. (10 de 2018). *Coursera*. Recuperado el 11 de 11 de 2018, de Detección de Objetos: https://www.coursera.org/learn/deteccion-objetos/home/info

Universitat Autònoma de Barcelona. (2019). *COURSERA*. Recuperado el 03 de 05 de 2019, de Support Vector Machine (SVM): Conceptos básicos: https://www.coursera.org/lecture/clasificacion-imagenes/support-vector-machines-svm-conceptos-basicos-52lRD

Usachokcharoen, P., Washizawa, Y., & Pasupa, K. (2015). Sign Language Recognition with Microsoft Kinect's Depth and Colour Sensors. *2015 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, 186-190. doi:10.1109/ICSIPA.2015.7412187

Valveny, E. (2015). *HOG - Cálculo de los histogramas.* Material de Curso, Universitat Autònoma de Barcelona, Departamento de Ciencias de la Computación.

Valveny, E. (2015). *HOG - Cálculo del descriptor.* Material de Curso, Universitat Autònoma de Barcelona, Departamento de Ciencias de la Computación.

Valveny, E. (2015). *HOG - Cálculo del gradiente.* Material de Curso, Universitat Autònoma de Barcelona, Departamento de Ciencias de la Computación.

Vanrell, M. (2015). *Introducción a la detección de objetos: Formación de la Imagen.* Material de Curso, Universitat Autònoma de Barcelona, Departamento de Ciencias de la Computación.

Voskresensky , A., & Ivanova, M. (1999). A methodical and didactical complex "Sign Language". *DIALOG'99: Computer linguistics and applications*.

Wu, X. (2015). *What is the difference between depth and RGB-depth images?* Obtenido de https://www.researchgate.net/post/What_is_the_difference_between_depth_and_RGB-depth_images

Xu, J. (2016). Sign Language Translation Using Kinect And Dynamic Time Warping.

Yanmei, C., Bing, L., Yen-Lun, C., Guoyuan, L., & Xinyu, W. (2015). A Real-time Dynamic Hand Gesture Recognition System Using Kinect Sensor. *2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2026-2030. doi:10.1109/ROBIO.2015.7419071

Yuqian, C., & Wenhui, Z. (2016). Research and Implementation of Sign Language Recognition Method Based on Kinect. *2016 2nd IEEE International Conference on Computer and Communications (ICCC)*, 1947-1951. doi:10.1109/CompComm.2016.7925041

Zhang, J., Zhou, W., Xie, C., Pu, J., & Li, H. (2016). Chinese sign language recognition with adaptive HMM. *2016 IEEE International Conference on Multimedia and Expo (ICME)*, 1-6. doi:10.1109/ICME.2016.7552950