UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO

DOCTORADO EN CIENCIAS DE LA COMPUTACIÓN

Desarrollo de algoritmos para Traductor automático de lenguaje de señas Mexicanas (LSM)

Tesis que presenta

Josué ESPEJEL CABRERA

Para obtener el Grado de

Doctor en Ciencias de la Computación

Asesor de Tesis:

Dr. Jair CERVANTES

Texcoco, Estado de México. Septiembre 2021

AUTONOMOUS UNIVERSITY OF MEXICO STATE

COMPUTER SCIENCE DEPARTMENT

Develop of algorithms for automatic Mexican sign language translation(MSL)

Submitted by

Josué ESPEJEL CABRERA

As a fulfillment of the requirement for the degree of

PhD Degree

Thesis Advisor

Ph.D. Jair Cervantes

Texcoco City, Mexico state. September 2021

Resumen

En años recientes, el desarrollo de algoritmos que asistan en la comunicación con gente sorda es un reto importante. El desarrollo de sistemas automatizados para traducir el lenguaje de señas es un tema de investigación vigente. Sin embargo, esto involucra varios procesos que van desde la captura del video, pre-procesamiento hasta la identificación y clasificación de las señas. El desarrollo de sistemas capaces de extraer características discriminativas para acentuar la capacidad de generalización de un clasificador es aun un problema muy complejo. El significado de una seña es a combinación del movimiento de la mano, forma de la mano y el punto de contacto de la mano con el cuerpo. Este trabajo presenta un método para detectar y traducir señas de las manos. Primero, se obtienen 15 cuadros por palabra, obteniendo 3 regiones de interés (manos y cara) de las cuales se obtienen características geométricas. Finalmente, se emplean diferentes clasificadores y se muestran los resultados experimentales.

Capítulo 1

Introducción

El lenguaje de señas es igual de complejo que cualquier lenguaje hablado, tiene sus propias reglas gramaticales y modismos. Poder generar un clasificador y posterior traductor tomando en cuenta las características anteriores propias del lenguaje de señas presenta un reto que se ha intentado resolver desde hace varios años. Algunas investigaciones han presentado resultados utilizando imágenes estáticas para identificar solo las letras del abecedario [83], hasta utilizar movimiento complejo de una o ambas manos [79][92], e incluso agregando partes del cuerpo o la detección del rostro para describir una o un conjunto de señas [61]

El uso de la visión artificial en la tecnología actual tiene diversas aplicaciones, usos como cámaras de seguridad, aplicaciones de fotos con reconocimiento de rostros, más recientemente los populares filtros para fotos usados en redes sociales, y actualmente se pueden observar algunos vehículos autónomos, y en mayor medida vehículos que se estacionan solos simplemente apretando un botón, aún falta tiempo, pero no tanto como el que se pensaba hace pocos años en que se pueda ver la totalidad de carros autónomos y posteriormente otro tipo de vehículos que mediante la visión artificial se

Seña estática (Letra A)

Ideograma de palabra

FIGURA 1.1: Diferencia entre seña estática (dactilología) y palabra (Ideograma)

dirijan de forma autónoma. Se puede hacer un comentario acerca de la tecnología militar, de la cual se han obtenido beneficios para la población, como el internet, pero en el caso de la inteligencia artificial se habla de dotar a misiles con esta tecnología. Se pensaría que se realizan investigaciones en el área de la inteligencia artificial para beneficiar a toda la población, pero lamentablemente también existen ejemplos como el anterior donde se observa la intención de implementar estos avances para perjudicarnos gravemente.

En el caso del reconocimiento de personas y rostros, particularmente en el de las manos y su movimiento, resulta una tarea complicada ya que el color de la piel, el color de la ropa, la utilización de playeras de manga corta o larga y la utilización de accesorios como pulseras relojes u otros pueden dificultar en gran medida la detección de las manos y sus movimientos, ya que durante la formación de la seña se toman en cuenta características como posición de los dedos, la ubicación de la palma de la mano y la dirección en la que se dirigen para poder determinar la seña que se está generando. Todas estas características presentan un gran reto que sigue siendo motivo de investigación, como se puede ver en la Figura 1.2

FIGURA 1.2: Traductores de conferencias de gobierno

La razón por la cual se continúa con la investigación es que, aunque se han obtenido muy buenos resultados en la clasificación de un conjunto de palabras seleccionadas del LSM, los tiempos de pre-procesamiento de las imágenes, la segmentación y la clasificación de estas es muy alto y aún debe ser reducido.

1.1. Planteamiento del problema

Existen discapacidades como la falta del habla que no son tan estudiadas, tal vez debido al porcentaje de la población que padece este problema, en México según el Instituto nacional de estadística y geografía,del total de la población con alguna discapacidad, el 8.3 % sufre problemas del habla o para escuchar, se suma a esto la falta de

infraestructura para la enseñanza de este grupo poblacional. Específicamente en esta investigación se observa la falta de algún traductor que a partir de una seña de LSM (Lengua de señas Mexicanas), el cual traduzca al lenguaje escrito el significado de esta. Actualmente existen algunas aplicaciones que se basan en animaciones o videos de personas, al escribir una palabra específica estas aplicaciones muestran la animación y/o video donde se muestra la seña que interpreta la palabra escrita.

Aplicaciones como las descritas anteriormente son muy útiles, sin embargo, se necesita un proceso que pueda hacer el procedimiento inverso, que a partir de una imagen y/o video muestre de manera escrita su significado. La gran diferencia es que lo que tenemos ahora en las aplicaciones existentes es un simple proceso de relacionar palabras introducidas para relacionarlas con algún diccionario de imágenes o de videos.

Una aplicación que pueda recibir como entrada una imagen y/o video y procesarlo para poder mostrar el significado de la seña conlleva el procesamiento de la imagen capturada, la problemática del entorno, condiciones de iluminación, distancia del objeto a la cámara, el color de la piel de la persona junto con el color de la ropa que porta, los posibles accesorios que porte en las manos como pulseras, reloj y anillos. Estos extras en las imágenes recibidas aumentan la dificultad de procesamiento porque se tiene que distinguir entre todos estos aspectos para determinar los elementos que pertenecen a la seña y que elemento no durante el procesamiento de la imagen.

Con esta investigación se plantea realizar una traducción de señas basado en secuencias imágenes tomando en cuenta todas las características antes mencionadas. Existen investigaciones que utilizan accesorios extra como guantes con sensores o utilizan marcadores corporales, pero con los accesorios se debe tener un cuidado extra en los sensores o dispositivos para la obtención de datos debido a lo delicado de los accesorios usados. Este cuidado de los accesorios es el que se pretende evitar utilizando solo una cámara de video para la obtención de imágenes.

Herramientas como el clasificador que se pretende desarrollar servirá ampliamente a la población en general, porqué no solo es ayudar a las personas sordomudas, por ejemplo, en el ámbito de la enseñanza, es ayudar al resto de la población a poder interactuar de manera sencilla y cotidiana con estas grandes personas que al comunicarse de manera diferente a la manera común en la población, ya que tienen que aprender en un sistema bilingüe, en el cual aprenden español y LSM. Con este tipo de herramientas no solo los familiares o personas más allegadas a la población sordomuda podrán interactuar con ellos, si no también cualquier persona en un futuro no muy lejano estando tal vez en una plaza comercial, en algún trabajo, o en lugares tan concurridos como

restaurantes o cafés, poder entender en cierta medida las señas que en este tiempo casi nadie tiene noción de su complejidad, sus reglas y por supuesto su significado es de gran importancia en una civilización que quiere ser cada vez más inclusiva.

1.2. Objetivos

Objetivo general Generar un clasificador y traductor de lenguaje de señas Mexicanas (LSM)

Para lograr este objetivo general se ha de cumplir, entre otras, con las tareas siguientes:

Objetivos Específicos

1. Obtener un conjunto de datos para la clasificación de señas con las características y/o condiciones necesarias para la investigación
2. Seleccionar las características idóneas del conjunto de datos obtenido para realizar una buena clasificación del mismo
3. Crear un algoritmo para la clasificación de las señas del LSM

1.3. Hipótesis

La implementación y/o de algoritmos de detección de objetos, el seguimiento de sus trayectorias permitirá implementar un clasificador ágil y eficiente de las señas pertenecientes al LSM

1.4. Estado del arte

En el reconocimiento de señas se pueden distinguir en las investigaciones dos ramas definidas:

- Letras/señas estáticas
- Palabras con movimiento de las manos

Así como en el español, donde las letras y los números son fácilmente separables en diferentes segmentos, en el LSM se muestra el mismo patrón para los números y palabras, sin embargo, al comparar las palabras en el idioma español que es la sucesión de letras, en el LSM no se presenta de la misma forma, en la formación de la palabra se tiene que tomar en cuenta lo siguiente:

- Movimiento de una o ambas manos
- Posición de la mano con respecto al cuerpo o contacto con alguna parte del mismo
- Movimiento de las manos hacia adelante o hacia atrás para denotar el tiempo presente, pasado o futuro de la palabra según sea el caso o hacia quién va dirigida.

Investigaciones realizadas en el lenguaje de señas han estado realizando en diferentes países para poder traducir todos los lenguajes de señas existentes. Como se comentó anteriormente, existen las señas estáticas y números [88][36][67], palabras y frases [14][17][56][52][29][1][65].

Además las investigaciones del lenguaje de señas se ayudan de diferentes métodos. En general los métodos en 3 diferentes áreas. El modelado 3D es también ampliamente usado en el lenguaje de señas, aqui se puede modelar, no solo una mano virtual 3D, si no también una representación virtuál de una persona, hoy en día el modelado 3D de la mano muestra todas las articulaciones y se puede distinguir el dorso y la palma de la mano, esta representación puede mostrar el movimiento de las manos, cabeza y pies. Se pueden usar sensores infrarrojos, marcadores corporales o múltiples cámaras para generar el modelo representativo.

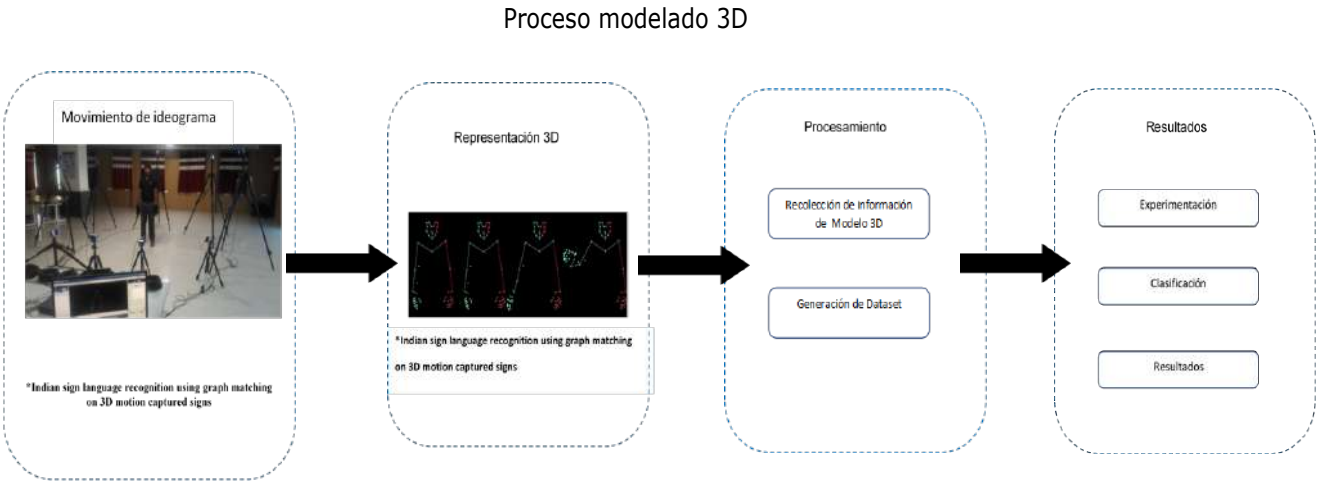

FIGURA 1.3: Representación del proceso modelado 3D

Existen investigaciones que se basan en el sensor de movimiento y representación 3D Kinect [79], así usando los sensores de movimiento se pueden reconocer las manos, su posición con respecto al cuerpo y el movimiento que realizan para la generación de la seña, se usan 223 señas y Hidden Markov models(HMM), obteniendo en el reconocimiento de la forma de la mano 28.7 %, posición de la mano78.3 %, movimiento de la mano 60 %, reconocimiento de la seña 33.8 %.

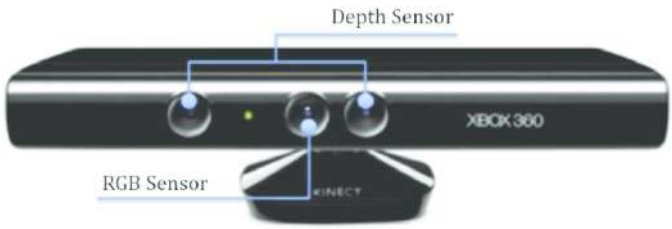

FIGURA 1.4: Sensor Kinect usado para modelar algunos objetos 3D(researchgate.net)

En [61] se divide el estudio de las señas usando Kinect en 5 partes, la postura de la persona, gestos, combinación de postura y gestos, comparación entre el método propuesto y métodos usados en otras investigaciones, y el tiempo de procesamiento. Obteniendo 42 % en la postura, 92 % en el gesto de la mano y 61 % en la combinación de la postura de la persona y gesto de las manos. En Zuzanna[69] la investigación se orienta hacia el LSE (Lenguaje de Señas Españolas) utilizando 91 señas de este lenguaje y usando HMM(Hidden Markov Models) para su reconocimiento y clasificación, Con la ayuda de un sensor de Leap Motion se obtiene información tridimensional de las manos, en el estudio se obtiene un resultado de 87.4 % para el modelo propuesto de clasificación de señas.

En Kumar [51] se propone un modelo de CNN para reconocer gestos del lenguaje de señas en 3D utilizando mapas de desplazamiento angular(JADMs), el cual codifica el color de la seña para entrenar la CNN, para la adquisición de la información 3D utilizan 57 marcadores corporales para detectar 200 señas, obteniendo una precisión de 92.71 %.

En [33] el sensor Kinect se utiliza para formar el esqueleto de la persona, obteniendo posición de las manos y los ángulos que forman con respecto al cuerpo, utilizando al igual que el anterior HMM se obtuvo un 89.25 % en la precisión de clasificación.

Una investigación realizada por Eepuri [50] utiliza marcadores corporales para obtener el movimiento de las manos y la cabeza en un ambiente 3D y posteriormente [51] proponen una combinación método basado en Leap Motion Sensor y Kinect para

detectar las manos y gestos de rostro que denotan el significado de las señas, utilizando HMM y Redes Neuronales clasifican 51 señas obteniendo 91.67 % de precisión. En [71]se traducen las señas utilizando para ello un sensor Leap Motion y Intel RealSense, con ellos se modela el esqueleto de las manos las articulaciones en 3D, así se obtienen los movimientos de cada uno de los dedos y las posibles flexiones realizadas, obteniendo una precisión de clasificación de 95 %.

También utilizando Kinect en Sun [85] se detectan 73 señas de palabras generando el esqueleto de la persona, y utilizando SVM se obtiene una precisión de clasificación de 86 %. En [4] se propone una metodología basada en el uso de sensor kinect para obtener 20 videos que representan palabras del lenguaje de señas Arabicas, obteniendo de cada palabra un video y 30 frames por video y también una secuencia de "deep images", las deep images muestran la región de interés y el fondo de la imagen claramente separadas y definidas. De cada una de las imagenes RGB y deep images se obtienen características invariantes, posteriormente se experimenta con 3 distintos conjuntos de datos, uno con características de imágenes RGB, otra con características de deep images y la última es la combinación de ambas características, obteniendo el mejor desempeño con el conjunto que tiene la combinación de características con 99.8 % y el desempeño más bajo con características de deep images con 97 %.

Li [55] propone un método para clasificar el alfabeto del ASL, sin embargo discriminan la letra *j* y *z*, ya que contienen movimiento, utilizando PCA y Redes neuronales utilizando auto-codificadores escasos, los cuales tienen mayor número de capas ocultas que la capa de entrada, al imponer una restricción de escasez a las unidades ocultas para que solo haya pocas capas ocultas estén activas. El conjunto de datos incluye imágenes RGB y deep images de las 24 letras del alfabeto obteniendo 99.10 % de presición utilizando el conjunto de datos combinado de ambas imágenes. En [59] implementan un sensor leap motion para clasificar el alfabeto del lenguaje Indio, la metodología consiste en grabar las coordenadas del movimiento de la mano durante la formación de cada seña, posteriormente calculan la distancia euclidiana, la medida de las distancias son calculadas comparando todas las señas obtenidas, una vez que obtienen las distancias utilizan la similaridad de coseno(cosine similarity), el cosine similarity comapra dos vectores de atributos, obteniendo una precisión de clasificación de 88.39 %. Una investigación similar se muestra en [64] donde usan 2 sensores leap motion para obtener las señas de 28 letras del alfabeto del lenguaje Árabe, ellos colocaron los dos sensores juntos durante la formación de las señas, seleccionando 12 de 23 características obtenidas con el sensor leap motion, incluyendo: longitud de los dedos, ancho de los dedos,

promedio de la posición de la punta de los dedos con respecto a los ejes x, y y z, el radio de la esfera de la mano, la posición de la palma de la mano con respecto al eje (x, y, z), movimiento de la mano, enrollamiento y estiramiento de la mano, obteniendo una precisión de clasificacide 97.7 % cuando comparan los dos sensores. En Chong [15] se implementa un sensor leap motion para clasificar 26 letras del alfabeto de señas Americano, aquí las características basicas que se consideraron incluyen el radio de la esfera de la palma, la posición de la palma de la mano, la posición de la punta de los dedos, en esta investigación agrupan las características en 5, la desviación estándar de la posición de la palma, el radio de la curvatura de la palma, distancia entre el centro de la palma con cada una de las puntas de los dedos, el ángulo entre dos puntas de dedos adyacentes además de su distancia tomando la punta de los mismos dedos adyacentes, finalmente combinaron grupos de características obteniendo 93.81 % de precisión de clasificación.

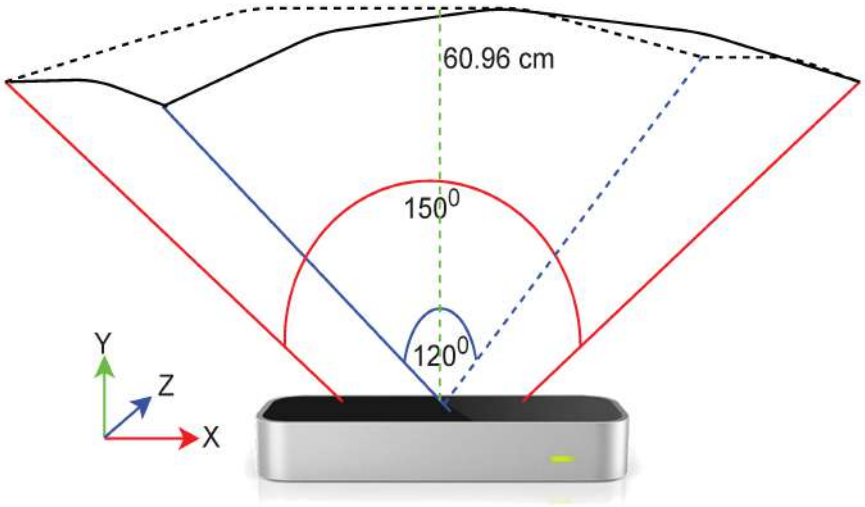

FIGURA 1.5: Sensor Leapmotion (doi.org/10.3390/electronics9121986)

Una de las metodologías generales en áreas de estudio es basado en el uso de sensores [50][29][**Abhishek:16**][1]. Los sensores que se usan en lenguaje de señas son colocados principalmente en guantes, sensores como acelerómetros, sensores de flexión, switches son de los mas usados al desarrollar métodos para la traducción del lenguaje de señas, ya que con la implementación de estos se puede obtener información acerca de la flexión o estiramiento de los dedos, la traslación o rotación de las manos, el contacto de las manos con alguna parte del cuerpo.

En [5] utilizan un guante con sensores y acelerómetro, de esta manera se reconoce el movimiento de cada dedo, su punto de flexión, rotación y movimiento de la mano

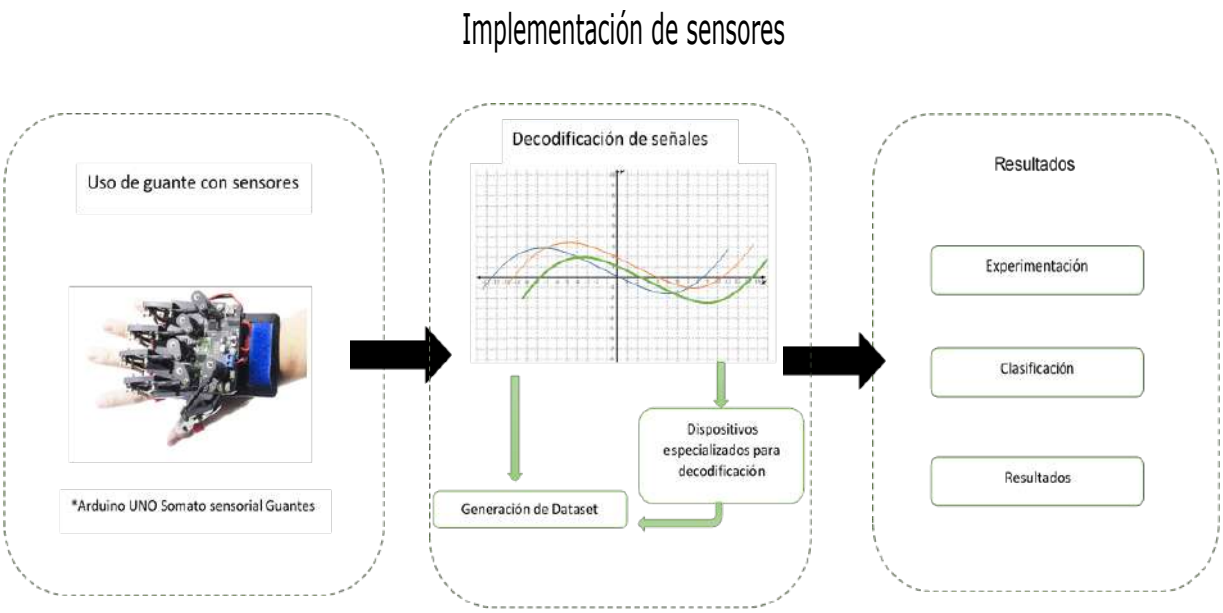

FIGURA 1.6: Representación de implementación de sensores

durante la formación de la seña, formando un vector 11×1 al obtener las distintas señales de los componentes del guante, obteniendo una precisión de clasificación de 89.26 %. También en [63] a partir de un guante incorporado con 5 sensores se detectan las flexiones de cada dedo en la configuración de la seña, utilizando un total de 50 señas y obteniendo una precisión de clasificación de 95 %. En [5] se implementa un método usando guantes con sensores, acelerómetros y sensores de presión y contacto, para de esta manera capturar todos los aspectos de la formación de la seña obteniendo 99.3 % de precisión. En [30] se utiliza un sensor de contracción muscular llamado MioArmband donde se detectan las contracciones musculares realizadas al generar las señas. En [2] se utiliza también el sensor MioArmband para detectar las letras del alfabeto, midiendo la contracción y relajación de los músculos del brazo obteniendo una precisión de lcasificación del 97 % en pruebas, pero reportando resultados muy bajos al intentar implementarlo en evaluaciones de tiempo real.

Otra área de investigacigeneral es la basada en imágenes, utilizando sensores generan pulsos eléctricos dependiendo de la rotación, traslación de las manos y flexión de los dedos, los cuales necesitan ser interpretados por algún aparato y/o aplicación, con las imágenes no se necesitan dispositivos extra para obtener e interpretar señales o pulsos, en general se puede usar una cámara simple o la cámara de un smartphone para obtener las imágenes y utilizar el procesamiento digital de imágenes (PDI) para poder obtener la información necesaria de la región de interés.

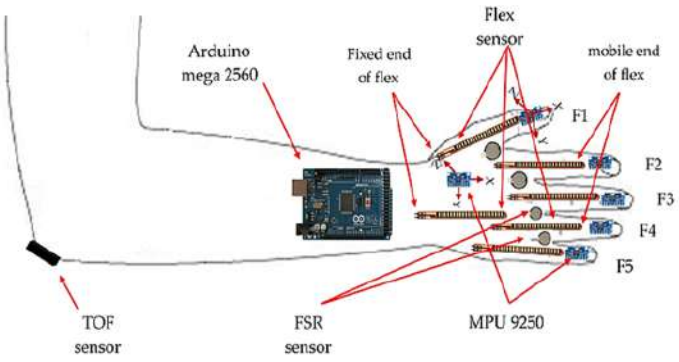

FIGURA 1.7: Esquena de guante con sensores (https://doi.org/10.1016/j.measurement.2020.108431)

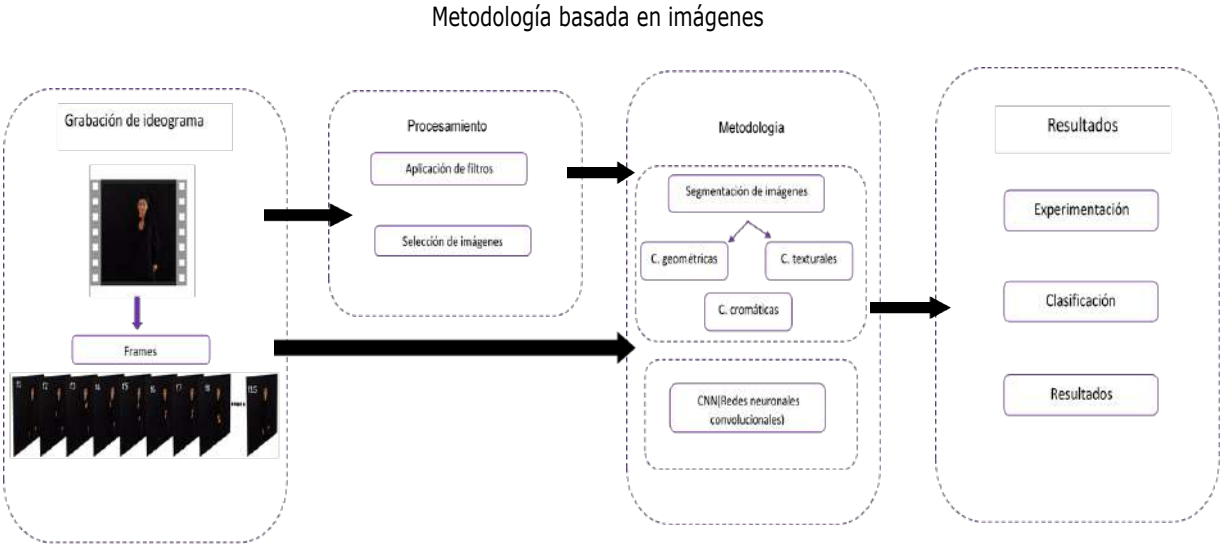

FIGURA 1.8: Modelo General basado en imágenes

Una investigación de Lenguaje de Señas en donde se emplean números [89], en donde se propone un sistema de clasificación donde se combinan Histograma y Wavelets en el reconocimiento de números 0-9, obteniendo resultados de 82.92 % y 98.17 % respectivamente.

En [44] se utilizan grabaciones de videos de señas del ASL(American Sign Languaje) donde se ubican en los frames de las señas la posición de los dedos en dos trayectorias dimensionales conectando la secuencia de frames de la seña y obteniendo así el movimiento que forma la seña. Usando 1804 clips de video del lenguaje obteniendo un 67.5 % de promedio de clasificación. El uso de los videos de señas en Revanthi [75] en comparación usan 3 frames por segundo para la obtención de características de la seña formada a partir de un conjunto de 26 señas del ISL(Indian Sign Languaje) y una vez

clasificado cada video y obtenido su significado se implementa una aplicación de voz que a partir del texto representativo de la seña se genera un audio equivalente a la seña generada.

Investigaciones basados en letras como Haifeng [81], se toma en cuenta el color de la piel inicialmente para la segmentación de la imagen RGB posteriormente se obtienen características geométricas de la mano y momentos de Hu, obteniendo una eficiencia del 85.73 %, al igual que en [86] donde proponen un clasificador obteniendo características geométricas obteniendo una precisión de clasificación del 90.19 %. En [72] y se propone el uso de PCA(Ánalisis de Componentes Principales), tomando los primeros 20 eigenvectores obteniendo un 78.49 % de eficiencia de clasificación. Usando también PCA e Histogramas en Ramesh [45] se clasifican 75 palabras del ISL obteniendo 94 % de precisión de clasificación.

En [14] se propone un método de clasificación basado en CNN(Redes Neuronales Convolucionales), donde dividen el proceso obteniendo primero las características espaciales de las imágenes de señas, posteriormente utilizan un BLSTM(Bidirectional Long Short Term Memory Layers) donde se modelan las características obtenidas inicialmente, finalmente con un CTC(Connectionist Temporal Classification) permite a las redes neuronales ser entrenados con los videos de las señas. obteniendo un 93.9 % de precisión de clasificación. Otra investigación utilizando CNN y SIFT(Scale invariant feature transformation) en [24] clasifican 50 señas del lenguaje de señas Indio, donde se utiliza la librería keras de python para desarrollar las redes neuronales convolucionales obteniendo una precisión de 92.8 %. Otra investigación donde se utiliza el color de la piel para detectar la región de las manos es en Raheja [73], donde utilizan filtros de piel para localizar la región de las manos a partir de videos de 80 señas, la secuencia de imágenes se convierten al espacio HSV ya que es menos sensitivo al cambio de brillo, y utilizan un clasificador SVM obteniendo una precisión de clasificación de 97.5 %. Una invetigación donde se utiliza tracking para el seguimiento de las manos fue en [47], donde se utiliza un algoritmo de flujo óptico llamado horn schunck optical flow algorithm, controlando la luminosidad del lugar donde se realizaron los videos y el color de las prendas que utilizaban las personas así como el fondo de la imagen para obtener grabaciones de 58 señas del lenguaje Indio utilizan una red neuronal backpropagation para clasificar obteniendo 82.58 % de precisión. En Kolles [49] se desarrolla una combinación de una CNN con HMM(Hidden Markov Models) utilizando conjuntos de datos de repositorios ampliamente utilizados [77][78][20], calculando la eficiencia de su método utilizando la estimación del error por palabra (WER) obteniendo un error mínimo

de 30 % en [77],38.3 % en [78] y 7.4 % en [20], comparandolos con metodología del estado del arte reportado en esta investigación. Otra investigación basada en CNN se observa en Liu [57], donde utiliza el modelo residual network layout basado en bloques de construcción residuales y clasifica diferentes conjuntos de datos empleados en otras investigaciones [18][19][16] para poder realizar una comparación con su modelo propuesto obteniendo una precisión de 73.5 %

La investigación realizada en [74] propone un método de clasificación de una frase, la cual se conforma de 18 palabras, aquí se mezclan letras y palabras usando una ANN (Red Neuronal Artificial) obteniendo un 96.9 % de precisión de clasificación. En [92] se usa un guante con 7 colores diferentes para ubicar los 5 dedos, la palma de la mano y el dorso como regiones de interés en las imágenes, se usan 14 señas en la investigación obteniendo un resultado de 96.7 % de precisión de clasificación usando Análisis de Componentes Principales.

Una investigación acerca del LSM (Lengua de Señas Mexicanas) realizada por Solís [83] usa una cámara para obtener imágenes de 21 de las 27 señas correspondientes al abecedario del LSM, para la obtención de las imágenes se incorporan 4 reflectores para eliminar cualquier tipo de sombra y con un perceptrón multicapa obteniendo un resultado de 93 % en la clasificación de las señas.

Dispositivos actuales como los smartphones son ampliamente usados por todas las personas alrededor del mundo, a pesar de que los primeros teléfonos celulares tenían características muy limitadas, los actuales tienen caracteriísticas parecidas a las computadoras comunes, por esta raźon investigaciones como la realizada por Rao [74] implementa un método basado en la captura de selfies, utiliza operador de bordes de sobel mejorado con morfología y umbral adaptativo obteniendo una segmentación casi perfecta de las manos y rostro, realizando una pequeña compensación del movimiento de las imágenes al ser tomadas. Para calcular el desempeño se utilizó el parámetro Word matching score (WMS) con un promedio de 90 %.

En el estado del arte descrito anteriormente podemos diferenciar los 3 grupos de técnicas generales en las investigaciones realizadas mencionados al inicio. Las razones por las que son igualmente utilizadas estas técnicas y no solo un conjunto en específico es que tienen ventajas unas sobre de otras, pero también tienen ciertas desventajas en comparación con otro grupo de técnicas. Se tiene que tomar en cuenta las ventajas y desventajas de la técnica que se utilizará durante la investigación, pero también la infraestructura con la que ya se cuenta y la experiencia utilizandola. En la Tabla 1.1 se agrupan los 3 principales grupos de técnicas usadas, y se muestran algunas de sus

principales ventajas y desventajas en su implementación.

Todo lo mostrado anteriormente muestra el amplio esfuerzo realizado por diversos investigadores alrededor del mundo han sido arduos al tratar de resolver el problema del reconocimiento del lenguaje de señas, usando para este fin, métodos muy diversos, tan diversos como son los que corresponden únicamente al campo de visión artificial, realizando el tratamiento de la imagen a partir de sus características cromáticas ,geométricas y texturales así a partir de las mismas poder clasificar con una precisión aceptable las diferentes señas, también hay quienes se valen de herramientas de otros campos, herramientas como los sensores corporales y los guantes con diferentes colores en ellos o con la adición de sensores de presión o movimiento, así complementando los métodos de tratamiento de imagen para poder determinar y traducir las diferentes señas. Todo lo mencionado anteriormente hace resaltar la gran necesidad que existe para desarrollar sistemas que puedan traducir los diferentes lenguajes de señas alrededor del mundo, el esfuerzo que se lleva a cabo para poder conseguirlo, y que todo esto sea la base de futuros sistemas especializados, rápidos y sencillos que traduzcan conversaciones completas en base a señas, de esta manera facilitar la interacción entre todos los sectores de población y finalmente eliminar la gran barrera con que cuentan las personas sordomudas hoy en día.

Además se puede aumentar el rango de uso de la detección de señas y/o gestos con las manos, la interacción hombre-máquina basadas en visión artificial, la implementación de los nuevos vehículos autónomos [9][34] los cuales no requieren interacción humana para su funcionamiento, uso de robots móviles [32][58], robots para ayuda [23][37], robots especializados en cirugías [82][57]. Los intentos para desarrollar y perfeccionar robots cada vez mas complejos traen mayores dificultades, y así la manera en que serán controlados tiene mucha importancia, algunos intentos se realizan mediante el uso de sensores [66][70], con guantes [7][41], como se mostro anteriormente estos dos métodos tienen ciertas desventajas en su uso. Usando movimientos corporales de manera natural para el control de robots se simplificaría el control y uso de robots o dispositivos autónomos. En Stantic [84] implementan 9 diferentes gestos manuales con sensores, acelerómetros y giroscopios para controlar un robot móvil, obteniendo dos diferentes conjuntos de datos, el primero consta de la combinación de la señal del giroscopio y acelerómetro, ya que estos describen el movimiento de las manos con precisión pero solo en pruebas *off-line*, el segundo conjunto es la combinación de las señales, se usa como clasificador arboles de decisión, Random Forest, Redes neuronales y Linear discriminant analysis para comparación de resultados, se utilizaron 20 voluntarios para

Tabla comparativa de las principales metodologías			
Técnicas	Sensores	PDI/Video	Modelado 3D
Ventajas	Alta precisión durante la formación de la seña. No existe perdida de información durante el movimiento de la mano. Fácil adquisición de información de todo movimiento o flexión de cada dedo.	El procesamiento de Imágenes es ampliamente usado debido a la baja complejidad de algunos métodos. No se requieren de dispositivos muy complejos o cámaras muy costosas en una ubicación compleja. Las imágenes pueden ser tomadas casi en cualquier lugar deseado.	El modelado 3D es invariante a las variaciones de luminosidad, vestimenta o cualquier accesorio que se porte. Con el modelado 3D de la mano se puede distinguir entre el dorso y la palma de la mano, además diferenciar cada dedo. Se puede modelar la traslación completa de la mano a través del cuerpo y mostrarse en diferentes ángulos.
Desventajas	Se necesitan de instrumentos especializados para la interpretación de algunas señales El costo de algunos de los sensores puede ser muy elevado y delicados, además de los cuidados y mantenimiento requeridos. El cambio o adición de sensores puede provocar la modificación del código usado y cambio de instrumentos.	El procesamiento de imagen es sensible al cambio de iluminación, emborronamiento y problemas de enfocado. Accesorios como cadenas, lentes o brazaletes pueden introducir ruido a la imagen El color del área de interés cuando es similar al color de fondo de la imagen dificulta mucho la segmentación	Para crear un entorno 3D es necesario un conjunto de cámaras especiales o sensores corporales para realizarlo. Ya sea en áreas abiertas o lugares cerrados es necesario la colocación de las cámaras especiales o lectores de sensores, esto dificulta modificaciones o cambios de lugar Puede ser muy costoso adquirir el equipo necesario como cámaras, sensores, tripiés o software para desarrollar experimentos con modelado 3D
References	[91][52][54]	[3][90][48]	[25][87][76][46]

CUADRO 1.1: Ventajas y Desventajas de las técnicas basadas en PDI, sensores, y modelado 3D

realizar la experimentación, concluyendo que las señas 1 y 2 obtienen el mejor resultado con 99 % y 99.5$ y 90.95 % de precisión general. En Farulla [27] se propone un nuevo sistema de telerehabilitación maestro-esclavo, donde se combinan video-based pose estimation(VPE) y tracking de manos, un exoesqueleto y un sensor que detecta la presión de las puntas de los dedos durante los ejercicios de rehabilitación, aquí la mano del operador que apoya en la rehabilitación es detectada y etiquetada a través de la estimación de la pose de la mano, esta pose es enviada al exoesqueleto colocada en la mano del paciente para realizar ejercicios de apriete con los dedos, el conjunto de datos consiste en las etiquetas generadas de la pose de la mano, la velocidad del movimiento y el apriete de la mano del paciente, se estima el error de la raíz de la media cuadrática entre el movimiento de la mano y el grado de movimiento del exoesqueleto. El máximo error obtenido fue de 9 y el menor de 0.5 con diferentes velocidades de movimientos. Una investigación donde se utiliza un sensor leap motion es en [6], con el sensor se genera un esqueleto tridimensional de la mano, con el SDK del dispositivo se recogen las señales de los movimientos de la mano y se envían a una mano artificial, obteniendo un buen control con el reconocimiento de las gestos de la mano. En [11] se propone una metodología para reconocer 8 diferentes gestos de la mano a partir de imágenes basado en el color de la piel, primero se utiliza un algoritmo de tracking facial para el reconocimiento facial, y a partir del reconocimiento obtener el color de piel del individuo para inicial el modelo de color que describa el color de piel durante las condiciones de iluminación presentes. Después se obtienen las regiones de las manos y el rostro y seleccionan el objeto que mas se parece al objeto anterior durante la secuencia de imágenes para evitar seleccionar objetos diferentes a las manos y rostro, se obtienen distancias escaladas de las manos y rostro, se utilizan HMM e Histogramas para clasificación obteniendo una precisión de 0.9 con HMM y 0.9 con histogramas. Una investigación basada en el control de prótesis para rehabilitación a través de movimientos de las manos [13], donde se usan 8 movimientos manuales y se implementa el dispositivo MIO-armband para obtener las contracciones musculares de la mano durante los movimientos realizados, obteniendo dos diferentes movimientos o gestos de cada uno de los participantes durante cada día de la experimentación, para disminuir el ruido en las señales se utiliza un filtro de señales, después se calcula el short-time average energy(STAE), desviación estandar(STD),mean power frequency(MPF), median frequency(MF) y mean square error(MSE) de las señales electromiográficas que se usan como valores de umbral para extraer los segmentos útiles de las señales sEMG, obteniendo un vector de 40 características de cada movimiento, y se reduce el conjunto

de datos implementando PCA, se utilizan cuatro diferentes clasificadores para realizar comparación de resultados, SVM, Random forest, Bayes, LDA. Obteniendo 92.25 % con SVM y el menor valor obtenido fue de 86.83 % con Bayes. En [39] se propone un sistema de reconocimiento de señas en tiempo real del alfabeto de señas Americano, aquí se obtienen imágenes de las señas realizadas, antes de realizar la umbralización de las imágenes se les aplica un filtro de mediana para eliminar el ruido, para la binarización de las imágenes se utiliza el método de Otsu, posteriormente se obtiene la región de la muñeca de la mano hasta los dedos, todo lo demás es desestimado, los dedos son rotados hacia arriba, al menos se buscan 15 pixeles consecutivos en el fondo de la imagen, si no se encuentran, la imagen se rota 90 grados y se revisa de nuevo el vector de pixeles, y así consecutivamente hasta que se encuentre la muñeca de la mano, así se obtienen 5 características, las puntas de los dedos, excentricidad, alargamiento, el pixel de segmentación y rotación también es usado como característica, se implementa una red neuronal backpropagation, se realiza primero una clasificación con cada una de las características por separado y una clasificación con todas las características en conjunto, de manera separada se obtuvo una precisión de 99.02 % y con la combinación de todas las características 99.6 %. La investigación realizada en [40] propone un método de control basado en el movimiento de las manos, se seleccionan cuatro movimientos de mano diferentes para realizar la investigación para el fácil control de un brazo robótico(UR5), los movimientos son cabeceo hacia abajo, cabeceo hacia arriba, giro a la izquierda y giro a la derecha, de la cabeza se obtienen los movimientos de pitch y giro, se tienen nueve ejes de inercia capturados por un sensor FSM-9 de los movimientos de la cabeza, el cual tiene 3 acelerómetros, tres giroscopios, y tres magnetómetros, para la experimentación se utiliza a 10 personas a las cuales se les explica el uso del dispositivo y posteriormente se ejecuta un ejercicio donde se intenta mover un cubo con el brazo robótico a través del movimiento de la cabeza. Durante la experimentación ninguno de los movimientos fue mal clasificada, se encontraron dos casos, el movimiento fue clasificado correctamente, o el movimiento no fue reconocido, obteniendo una precisión de 68.49 % de los movimientos completados.

Los gestos manuales no solo son parte de un idioma ajeno a la población en general, los gestos manuales están presentes en la vida diaria, los saludos, ademanes de enojo, dar respuestas rápidas a preguntar sin la necesidad de decirlo con palabras. Poder adaptar elementos sencillos usados en la vida diaria a la tecnología es lo que la vuelve cada día más intuitiva y sencilla de usar. El área de la robótica es una de las áreas de la tecnología con mayor auge, se divide en diversas ramas, como robots móviles[32][58],

robots de asistencia[23][37], especializados en el área de cirugías [82][57]. Estos ejemplos muestran la tendencia del uso de robots diversas áreas significativas en nuestras vidas. Debido a la importancia que cobrarán los robots en el corto tiempo, los intentos para desarrollar y perfeccionarlos acarrea grandes dificultades, la manera en que los robots son controlados es uno de los aspectos principales que se debe de tomar en cuenta, algunos son controlados mediante sensores[66][70], controlados con guantes[7][41], la desventaja general de estos ejemplos de control es el costo de los sensores usados para tal fin y los accesorios extra usados para decodificar las señales obtenidas de estos sensores. El poder de la visión computacional puede eliminar el uso de elementos extra para el control de un robot, entonces los gestos manuales pueden ser utilizados para poder controlar un robot, eliminando el costo de sensores de control y el costo de su mantenimiento.

Los algoritmos basados en visión computacional como el tracking nos permite seguir puntos de interés en una secuencia y además considerar los movimientos como órdenes específicas, entonces se puede considerar que se puede interactuar de una manera natural con las máquinas. Usando este control natural en Stantic [84] combina 9 diferentes gestos manuales con sensores como acelerómetros y giroscopios para controlar un robot móvil, obteniendo dos conjuntos de datos de características, el primero se obtiene de la combinación del giroscopio y el acelerómetro, estas características describen los movimientos manuales de manera precisa pero solo en experimentos *off-line*, el segundo conjunto de datos se deriva del primero para implementarlo en tiempo real, para la clasificación se utilizaron arboles de decisión, random forest, redes neuronales y linear discriminant analysis para experimentar con los conjuntos de datos, se utilizaron 20 voluntarios para utilizar los sensores de control del robot, siendo la seña 1 y 2 las que obtuvieron mejores resultados ($99\,\%$ y $99{,}5\,\%$) y 90.95 % de precisión general. Farulla [27] propone un nuevo sistema de telerehabilitación combinando estimación de pose de la mano por tracking (Video Pose Estimation), un exoesqueleto y un sensor que detecta la fuerza de presión ejercida por los dedos durante los ejercicios de rehabilitación, se realiza la pose de la mano y esta se envía al exoesqueleto del paciente para que realice el movimiento y la presión del sensor, el conjunto de datos consiste de las etiquetas generadas de la mano segmentada, la velocidad del movimiento y la presión ejercida, en esta investigación se calcula la raíz del error cuadrático medio entre el movimiento de la mano y el grado de movimiento del exoesqueleto. El error máximo obtenido fue 9 y el error mínimo de 0.5 a diferentes velocidades. Una investigación que mejora el control de una mano robótica usando una cámara es [6], donde se usa

un sensor leap motion que reproduce el esqueleto de una mano, usando el SDK del sensor se colectan las señales de los movimientos de la mano, los cuales son enviados a la mano artificial, obteniendo un buen control con el reconocimiento de señas. En [13], se propone un sistema de rehabilitación con prótesis controladas por señas, se utilizan 8 señas y sensor MIO Armband, este último obtiene las contracciones musculares del brazo durante la formación de la seña, grabando 2 diferentes señas por persona cada día, se utilizó un filtro de señales para reducir el ruido en la señal obtenida, entonces se calcula el short-time average energy(STAE), desviación estandar(STD), mean power frequency(MPF), frecuencia media(MF) y el error cuadrático medio(MSE) de las señales electromiogrficas tomadas como valor de umbral para extraer los segmentos sEMG, obteniendo un vector de 40 elementos de cada movimiento, se utilizó PCA para reducir el conjunto de datos, para evaluar el conjunto de datos resultante se utilizaron cuatro clasificadores diferentes (SVM, Random forest, Bayes, LDA) obteniendo 92.25 % con SVM y el menor de 86.83 % con bayes. En [11] se propone una metodología para reconocer 8 diferentes gestos manuales obtenidos de imágenes, basado en el color de la piel, primero se utiliza reconocimiento facial para obtener la región de la cara, de esta región se obtiene el color de la piel del individuo para iniciar el modelo de color que describe el color de la piel bajo las condiciones de iluminación en el momento. De esta manera obtienen la región de la cara y las manos y seleccionan el objeto que más se parece a estos durante la secuencia de imágenes para evitar seleccionar objetos diferentes a los de interés. Se obtienen las distancias escaladas entre la cara y las manos, usan HMM y Histograma para la experimentación, obteniendo un rate de0.9 con HMM y 0.9 usando Histograma. En [39] se propone un método de reconocimiento de señas en tiempo real correspondientes al alfabeto del lenguaje de señas Americanas, obteniendo imágenes de las señas del alfabeto, para el pre-procesamiento se utilizó un filtro de mediana para remover el ruido en la imagen, posteriormente se utilizó el método de Otsu para segmentar la imagen, de la imagen segmentada se seleccionó la región que comprende de la muñeca a los dedos, eliminando el resto de la sección segmentada, los dedos son rotados hacia arriba en la imagen, al menos 15 pixeles consecutivos son buscados desde el fondo de la imagen, si no se ubican la imagen se rota 90 grados y se revisa de nuevo hasta que se ubique la muñeca, obteniendo las puntas de los dedos, la excentricidad, alargamiento, los pixeles segmentados y la rotación son tomados como características. Se implemento una red neuronal back propagation, inicialmente se utilizó cada característica por separado para la clasificación y todas en un solo conjunto, de manera individual el desempeño obtenido fue de los pixeles segmentados con 99.02 % y de 99.6 %

con las características combinadas. La investigación realizada en [40] propone un método de control basado en la cabeza, se seleccionan 4 diferentes movimientos de cabeza en esta investigación para poder cambiar de manera sencilla entre los grupos de control de un brazo robótico (UR5), los movimientos particulares son: cabeceo hacia abajo, cabeceo hacia arriba, giro a la izquierda y giro a la derecha, de la cabeza se obtienen el movimiento de cabeceo y su giro. Se capturan 9 ejes de inercia de los movimientos de la cabeza usando un sensor FSM-9, el cual cuenta con 3 acelerómetros, 3 magnetómetros y 3 giroscopios. Los movimientos de la cabeza son mapeados hacia los movimientos del robot realizados por 10 individuos, quienes intentan mover un cubo con la mano robótica. En todos los experimentos realizados no se obtuvieron falsas clasificaciones, solo se obtuvieron dos casos, movimientos clasificados correctamente y movimientos no reconocidos, obteniendo 69.49 % de clasificación. Un método para reconocer señas humanas, propuesta por Jiang[42], implementa un sistema novedoso de reconocimiento de señas(HGR), el cual es clasificador de componentes de movimiento y clasificador de ubicación de componentes. Estas dos capas se complementan entre ellas, cuando la capa de movimiento se logra, el vocabulario original se divide en listas de candidato posible y candidato no posible. Los candidatos posibles se toman como entrada de la segunda capa, la salida de la segunda capa es el resultado del reconocimiento de la seña. Aquí se implementan un conjunto de Histogramas de "Movimiento de energía" para representar la secuencia de movimientos y poder implementar una reconstrucción de la aproximación del reconocimiento HGR, en la segunda capa se implementan métodos SIFT+BOW con una gran cantidad de pequeñas secuencias de videos. Los resultados fueron probados mediante la distancia *Levenshtein*, obteniendo en ambas capas una distancia de 24.02 %. En [43] se utiliza un sensor de muñeca que censa la electromiografía de superficie (sEMG) y la unidad de medición inercial (IMU) que puede estimar señas manuales, de los sensores se obtienen el características de dominio de tiempo, los cuales tienen bajo costo computacional y alta efectividad, las características de dominio seleccionadas son: valor medio absoluto, zero crossings, slope sign changes, waveform length de las señales IMU. Los valores medios absolutos contienen información de la amplitud y la fuerza de la señal. Se utiliza el clasificador LDA para identificar las señas, el conjunto de datos se divide en dos, primero las señas aéreas y las señas de superficie, obteniendo 92.6 % para el primer subconjunto y 88.8 % para el segundo. En [53] se propone un entorno de trabajo virtual para obreros, eliminando los riesgos de trabajo para las personas, creando para esto un robot humanoide, el cual puede imitar los movimientos realizados por el trabajador, para obtener los movimientos se implementa un

sensor Kinect que modela los movimientos en 3D, a partir de un tracking de manos se calculan sus coordenadas en diferentes sistemas de coordenadas, entonces se combinan los modelos completos en 3D de las manos y cuerpo y se crea un escenario 2D donde interacciona el humanoide con el entorno de trabajo.

Capítulo 2

Preliminar

En este Capítulo se muestran los fundamentos de los métodos empleados para resolver el problema de la identificación de señas.

2.1. Conceptos básicos sobre PDI

La tecnología abarca grandes áreas de nuestra vida, de entre ellas aquella que tiene que ver con las imágenes tiene un gran uso y relevancia, ya que todo mundo toma fotografías para recordar acontecimientos o graba videos de actos impresionantes, transmisión de programas informativos, culturales y mucho más, detrás de esto existe toda una ciencia que se encarga del estudio de las imágenes digitales, que se podrían dividir en dos áreas generales de interés, la mejora de las imágenes para su interpretación y el procesamiento de imágenes para su almacenamiento. A estas imágenes se les puede hacer modificaciones u operaciones por medio de computadoras o dispositivos inteligentes, a esto se le llama Procesamiento Digital de Imágenes(PDI).

2.1.1. Imagen

La materia prima del procesamiento de imágenes y la visión son las imágenes, las cuales se consideran como una representación del mundo fśico que tiene información importante, la cual es captada mediante un proceso de muestreo que se realiza generalmente por medios electrónicos. Una imagen se puede representar como una función $I(x, y)$, donde en cada conjunto de coordenadas x e y se tiene una intensidad de gris si se habla de una imagen a escala de grises o de un valor RGB si es una imagen a color, como se ejemplifica en la Figura 2.1. El modelo más común de representación de una imagen es por medio de una matriz, tal que:

FIGURA 2.1: a) Imagen RGB b) Imagen escala de grises

$$I(x,y) = \begin{bmatrix} I(1,1) & I(1,2) & \dots & I(1,n) \\ I(2,1) & I(2,2) & \dots & I(2,n) \\ \vdots & \vdots & \ddots & \vdots \\ I(m,1) & I(m,2) & \dots & I(m,n) \end{bmatrix} \tag{2.1}$$

Una imagen en escala de grises es una matriz cuyos valores han sido escalados para representar un determinado número de intervalos, los intervalos pueden ser en un rango de 256 intensidades, es decir intensidades en el rango $[0, 255]$, o en un rango de solo 2 intensidades $[0, 1]$ como se puede observar en la Figura 2.1 b),

2.1.2. Pre-procesamiento

Existe una enorme cantidad de operaciones para modificar la imagen, mejorarla o reducir el ruido en ésta. Las técnicas utilizadas para tal fín se conocen como técnicas de pre-procesamiento.

Operadores puntuales

Los operadores básicos en procesamiento de imágenes son las operaciones puntuales en donde el valor de cada pixel es remplazado por un nuevo valor, este nuevo valor es obtenido a partir del valor del pixel antiguo. Si queremos aumentar el brillo incrementando el contraste, simplemente se multiplican todos los pixeles por un escalar. De manera inversa para reducir el contraste dividimos todos los valores por un escalar. Si todo el brillo es controlado por un nivel l y el rango es controlado por una ganancia k, los puntos en una nueva imagen N, puede ser relacionado con el brillo de la primera imagen O, por:

$$N_{x,y} = k * O_{x,y} + l \qquad \forall x, x \in 1, \dots, N \tag{2.2}$$

Este operador puntual remplaza el brillo de acuerdo con la relación lineal de brillo. En los controles de ganancia de contraste o rango si la ganancia es mayor a la unidad, el rango de salida será incrementado.

Operaciones Grupales

Los operadores grupales calculan el valor del nuevo pixel de los pixeles vecinos a este usando un proceso de agrupamiento. La operación grupal usualmente es expresada como convolución, donde la máscara es un grupo de coeficientes de peso. Esta máscara es usualmente cuadrada[60]. El tamaño es normalmente usado para describir el diseño que se usa; un diseño de 3×3 se refiere a 3 pixeles de alto y 3 de ancho. El cálculo del valor del nuevo pixel se obtiene de la colocación de la máscara en el punto de interés. Los valores de los pixeles son multiplicados por los coeficientes de peso correspondientes y adicionados a una suma total. La suma evalúa el nuevo valor del pixel central, y este se convierte en el pixel de la imagen de salida. Si la posición de la máscara aún no ha alcanzado el final de la línea el diseño se mueve horizontalmente al siguiente pixel y se realiza el proceso nuevamente.

En la Figura2.2 se muestra el proceso de convolución, en donde la imagen es calculada de la imagen original.

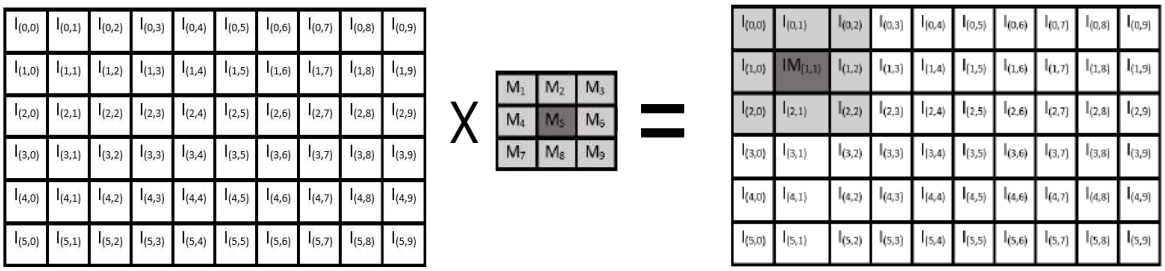

FIGURA 2.2: Máscara de Convolución

A partir del cálculo obtenido por la máscara de convolución del pixel central de la imagen original obtenemos la tonalidad del pixel en la nueva imagen. Ya que la máscara no se puede extender más allá de la imagen, la nueva imagen es más pequeña que la original ya que el valor nuevo no puede ser calculado por los puntos del borde de la nueva imagen, aunque existen algunas técnicas para eliminar esta desventaja. Cuando la máscara alcanza el final de la línea, esta se reposiciona al inicio de la siguiente línea. Para un agrupamiento de 3×3 se aplican 9 coeficientes de peso w, los cuales son

aplicados a los puntos de la imagen original para calcular el punto de la nueva imagen. La posición del nuevo punto es sombreado en la máscara [89].

Para calcular el valor de la nueva imagen N, en el punto de coordenadas x, y, la máscara en la Figura 2.2 realiza la operación en la imagen original O de la siguiente manera:

$$\begin{aligned} Ny = {} & w_0 * O_{x-1,y-1} + w_1 * O_{x,y-1} + w_2 * O_{x+1,y-1} + \\ & w_3 * O_{x-1,y} + w_4 * O_{x,y} + w_5 * O_{x+1,y} + \\ & w_6 * O_{x-1,y+1} + w_7 * O_{x,y+1} + w_8 * O_{x+1,y+1} \end{aligned} \tag{2.3}$$

La razón por la que la máscara no se coloca en el borde es porque al momento de ubicar la máscara en un punto del borde secciones de ella quedan fuera de la imagen y por lo tanto no obtienen información para generar el nuevo pixel. El ancho del borde es igual a la mitad del tamaño de la máscara. Para calcular valores de los pixeles del borde se tienen tres opciones:

- Colocar el borde como negro
- Asumir que la imagen se replica al infinito a lo largo de ambas dimensiones y calcular valores nuevos por cambios cíclicos del borde lejano.
- Calcular el valor del pixel de una pequeña área

Modificación de brillo

Existen situaciones en las que se necesita aumentar o disminuir el brillo de la imagen para poder seguir con el procesamiento de la misma. Un método simple para modificar el brillo de la imagen es agregar una constante a cada pixel de la imagen $f[m, n]$ obteniendo una imagen nueva $g[m, n]$ de la siguiente manera:

$$g[m, n] = f[m, n] + k \tag{2.4}$$

$$g[m,n] = f[m,n] + k \tag{2.5}$$

donde Eq2.4 se utiliza para aumentar el brillo de la imagen, y la Eq. 2.5 se utiliza para disminuir el brillo de la imagen, como se muestra en la Figura 2.3.

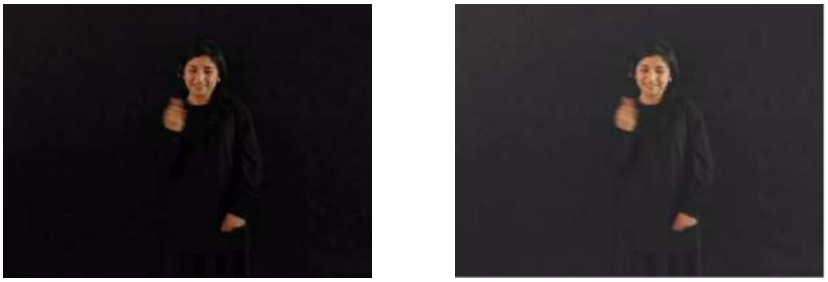

FIGURA 2.3: a) Imagen original b) Imagen modificada

.

Contraste

El ajuste de contraste se realiza al escalar todos los pixeles de la imagen por una constante k,

$$g[m,n] = f[m,n] * k \tag{2.6}$$

Filtros

Las imágenes no siempre tienen las condiciones ideales para poder obtener la información necesaria en nuestra investigación. Existen filtros pasa bajos y filtros pasa altos. Los filtros pasa bajos ayudan a eliminar el ruido que existe en la imagen, por otro lado los filtros pasa altos son usados para detectar bordes.

El filtro de la media se utiliza para reducir la cantidad de variaciones de intensidad en las imágenes, utilizando una máscara de `n` × `n`, con el cual se calcula el valor de la media de los pixeles de un vecindario, tomando en cuenta el pixel central y el tamaño de la mascara.

El filtro gaussiano se utiliza para suavizar imágenes utilizando máscaras de `m` × `n`, la función gaussiana se define como:

$$f(x) = \frac{1}{(2\pi\sigma^2)^{1/2}} \cdot (exp)(-\frac{x^2}{2\sigma^2})F(\varpi) = (exp)(-\frac{1}{2}\sigma^2\varpi^2) \tag{2.7}$$

Así el operador convolucional requerido para eliminar el ruido de la imagen debe tener una distribución gaussiana. Las máscaras más usadas son las siguientes **??**:

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \qquad \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

FIGURA 2.4: Mascaras de filtro promedio y Gaussiano

Detector de bordes

Las imágenes están compuestas por diferentes formas, algunas de estas formas son de interés, otras no, para poder resaltar objetos dentro de la imagen se tienen métodos para poder resaltar los bordes de los objetos que la conforman. Estos métodos de manera general estiman la intensidad de los gradientes de intensidad con la ayuda de máscaras de convolución.

La magnitud del gradiente puede ser calculado vectorialmente usando la transformación **??**:

$$g = (g_x^2 + g_y^2)^{1/2} \tag{2.8}$$

con **??** de la siguiente forma:

$$g = |g_x| + |g_y|$$

g=max($|g_x| + |g_y|$)(2.9)

El filtro de Canny es uno de los más usados para la detección de bordes, usando una máscara de convolución de 2×2 para calcular el gradiente de la imagen I(x,y) **??**:

$$P_x[i,j] = (I[i+1,j] - I[i,j] + I[i+1,j+1] - I[i,j+1])/2$$

$P_y[i,j] = (I[i,j+1] - I[i,j] + I[i+1,j+1] - I[i+1,j])/2$

(2.10)

la magnitud se calcula de la siguiente manera:

$$M(i,j) = \sqrt{f_x(i,j)^2 + f_y(i,j)^2} \tag{2.11}$$

El filtro de Prewit se utiliza para detectar bordes de manera vertical y horizontal en las imágenes, se pueden calcular ocho diferentes orientaciones aunque no de manera precisa, la máscara de convolución usada es de (3×3)**??** para las ocho direcciones. A continuación, se muestran las máscaras del filtro de Prewit**??**

$$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

El filtro de Sobel, al igual que en Prewit usa una máscara de convolución de (3×3) para detectar los bordes en las imágenes como se muestra a continuación:

$$\begin{bmatrix} -1 & 0 \\ -2 & 0 \\ -1 & 0 \end{bmatrix}$$

2.2. Segmentación

En esta sección se abordarán algunos métodos de segmentación de imágenes, estos métodos de procesamiento de las imágenes ayudaran en la localización de las areas de la imagen que forman las diferentes señas, estas son las manos y rostro, la utilización de estos métodos proporcionara las regiones de interés para esta investigación.

2.2.1. Método de Otsu

El método de Otsu es un método basado en el valor del umbral de la imágen, es igual al promedio de los niveles medios de dos clases divididas por este umbral. Suponiendo

que los pixeles dados de una imagen son representados en una escala L de grises donde $L\{1, 2, 3, .., L\}$, n_i denota el numero de pixeles en el nivel i en escala de grises, y N es el total de pixeles de la imagen $N = n_1 + n_2 + ... + n_L$. Dividiendo la imagen en dos clases C_0 y C_1 por el umbral T.

Donde la probabilidad está dada por:

$$p_i = \frac{n_i}{N}, p_i \geq 0, \sum_{i=1}^{L} pi = 1 \tag{2.12}$$

$$p0(T) = \sum_{t=1}^{T} p_i \tag{2.13}$$

$$p1(T) = \sum_{i=T+1}^{L} p_i \tag{2.14}$$

$$\tag{2.15}$$

Las clases están dadas por:

$$\mu_0(T) = \sum_{i=1}^{T} i \Pr(i \mid C_0) = \frac{1}{P_0(T)} \sum_{i=1}^{T} ip_i \tag{2.16}$$

$$\mu_1(T) = \sum_{i=T+1}^{L} i \Pr(i \mid C_1) = \frac{1}{P_1(T)} \sum_{i=T+1}^{L} ip_i \tag{2.17}$$

Varianza:

$$\sigma_0^2(T) = \sum_{i=1}^{T} [i - \mu_0(T)]^2 \Pr(i \mid C_0) = \frac{1}{P_0(T)} \sum_{i=1}^{T} [i - m_0(T)]^2 P_i \tag{2.18}$$

$$\sigma_1^2(T) = \sum_{i=T+1}^{L} [i - \mu_1(T)]^2 \Pr(i \mid C_1) = \frac{1}{P_1(T)} \sum_{i=T+1}^{T} [i - m_1(T)]^2 P_i \tag{2.19}$$

Función objetivo:

$$\sigma_w^2(T) = P_0(t)\sigma_0^2(T) + P_1(T)\sigma_1^2(T) \tag{2.20}$$

Umbral Óptimo:

$$\sigma_w^2(T^*) = \min_{1 \le T < L} \sigma_w^2(T) \tag{2.21}$$

$$T^* = \arg \min_{1 \le T < L} \sigma_w^2(T) \tag{2.22}$$

2.2.2. Método de segmentación basado en PCA

El método de PCA por sus siglas en inglés o análisis de componentes principales es una técnica muy útil en la clasificación de conjuntos de datos. Ya que por medio de este método se reduce nuestro conjunto de datos original a un nuevo conjunto de datos encontrando un nuevo conjunto de variables muy representativas para la clasificación de los datos. Los componentes principales son una serie de mínimos cuadrados que se ajustan en la muestra y son ortogonales al conjunto anterior.

Teniendo un conjunto de datos n de un vector de p variables

$$X = (x_1, x_2, ..., x_p) \tag{2.23}$$

El primer componente principal del conjunto de datos se obtiene:

$$z_1 = a_1^T X = \sum_{i=1}^{p} a_{i1} x_i \tag{2.24}$$

donde el vector a_1 se define:

$$a_1 = (a_{11}, a_{21}, ..., a_{p1}) \tag{2.25}$$

es elegido:

$$var[z_1] \text{ es máximo} \tag{2.26}$$

Entonces el *k-ésimo* componente principal se obtiene:

$$z_k = a_k^T X \tag{2.27}$$

con

$$k = 1, ..., p$$

y

$$a_k = (a_{1k}, a_{2k}, \ldots, a_{pk}) \tag{2.28}$$

$$var[z_k] \text{ es máximo} \tag{2.29}$$

sujeto a

$$cov[z_k, z_l] = 0 \tag{2.30}$$

con

$$k > l \geq 1 \tag{2.31}$$

$$a_k^T a_k = 1 \tag{2.32}$$

2.2.3. Método de Frontera óptima

Este operador selecciona pixeles con un valor particular, o que se encuentran dentro de un rango específico. Este operador puede ser usado para localizar objetos dentro de una imagen si se conoce su nivel o rango de brillo[60]. Esto implica que también se conoce el brillo del objeto. Existen dos maneras principales: umbralización uniforme y adaptativa. En el primero los pixeles por arriba de un nivel determinado se colocan como blancos. Los que se encuentran debajo del nivel se cambian a negros como se muestra en la Figura2.5

FIGURA 2.5: a) Imagen Original, b) Imagen segmentada

El método uniforme claramente requiere del conocimiento del nivel de grises de la imagen, o que los rasgos que se buscan no sean seleccionados en el proceso. Si el nivel no es conocido, se puede usar la ecualización de histograma o normalización de intensidad. Este es por supuesto un problema. Estos problemas pueden ser resueltos por simples aproximaciones como el operador de umbral.

Existen técnicas más avanzadas conocidas como el umbral optimizado. Estas técnicas buscan seleccionar un valor del borde que separe al objeto del fondo. Esto sugiere que el objeto tiene un rango de intensidad diferente al fondo, para que se pueda elegir un borde adecuado.

La base es utilizar el histograma normalizado de la imagen, donde el número de puntos de cada nivel es dividido por el total de puntos en la imagen. Como tal esto representa una distribución de probabilidad de los niveles de intensidad

$$p(l) = \frac{N(l)}{N^2} to$$

Esto puede ser usado para computar el cero y los momentos acumulativos de primer orden el histograma normalizado elevado al $k - ésimo$ nivel como:

$$w(k) = \sum_{l=1}^{k} p(l) \tag{2.35}$$

y

$$\mu(k) = \sum_{l=1}^{k} l * p(l) \tag{2.36}$$

El nivel total significativo de la imagen está dado por:

$$\mu T = \sum_{l=1}^{N_{\text{máx}}} l * p(l) \tag{2.37}$$

La discrepancia de la separabilidad de la clase está dada por

$$\sigma_B^2(k) = \frac{(\mu T * w(k) - \mu(k))^2}{w(k)(l - w(k))} \qquad \forall K \quad \in l, N_{\text{máx}} \tag{2.38}$$

La umbralización óptima es el nivel en el que la discrepancia de separabilidad de la clase se encuentra en su máximo, llamado el borde optimo T_{opt}

$$\sigma_B^2(T_{opt}) = \max_{l \le k \le N_{\text{máx}}} (\sigma_B^2(k)) \tag{2.39}$$

Estas técnicas son usadas con frecuencia en el reconocimiento de patrones estáticos: el umbral del objeto es clasificado de acuerdo a sus propiedades estáticas.

2.2.4. Método de Frontera adaptativa

De manera general el cálculo del umbral o frontera en las imágenes se calcula de la como 1 a todos los pixeles que sean iguales o mayores que el valor de la frontera, de lo contrario se hacen 0.

Donde la imagen b es la binarización resultante sobre la imagen I. En el frontera adaptativa, el umbral o frontera es calculado para cada pixel, basado en estadísticas de pequeños grupos de pixeles como su rango y varianza, o los parámetros de los pixeles vecinos. Esto puede ser realizado de diferentes maneras como la sustracción del fondo, el modelo de flujo de agua, la desviación y media estándar entre otras[**Romen**].

2.3. Extracción de características

En esta sección se abordarán los métodos de extracción de características de las imágenes empleadas, siendo este un aspecto de gran relevancia para obtener las características más relevantes y altamente discriminativas para una buena clasificación.

2.3.1. Características geométricas

Las características que se emplearon describen las propiedades básicas de la región a reconocer, estas son; área de la región, redondez de a mano, longitud del borde de la mano, elongación de la mano definida por la longitud y ancho de la mano, las coordenadas x e y del centro de gravedad, densidad, definida por la longitud de los bordes de la mano y el área de esta.

1. Densidad.
$$Densidad = \frac{(longitud\_región\_mano)^2}{A} \tag{2.40}$$

2. Distancia centroide. Distancia de los bordes de la mano al centroide (x_c, y_c)
$$r(t) = [(x(t) - x_c)^2 + (y(t) - y_c^2)^{1/2}] \tag{2.41}$$
donde:
$$x_c = \frac{1}{L}\sum_{t=0}^{L-1} x(t), y_c = \frac{1}{L}\sum_{t=0}^{L-1} y(t) \tag{2.42}$$
donde x_c es el centroide i, y_c es el centroide j, (x_c, y_c) es el centroide del objeto.

3. Perímetro

   Es el número de pixeles en el contorno de la región
$$P = \left(\sum_{t=1}^{N} \|x_{i+1} - x_i\|\right) + \|x_n - x_i\| \tag{2.43}$$
   Donde P es el perímetro de la región

4. Área

   El área se obtiene con el numero de pixeles de la región

5. Altura

   La altura de una región se define como:

$$h = x_{\text{máx}} - x_{\text{mín}} + 1 \tag{2.44}$$

6. Ancho

   El ancho esta definido por:

$$w = y_{\text{máx}} - y_{\text{mín}} + 1 \tag{2.45}$$

7. Redondez

   Esta característica indica la calidad de redondo de una región, se define como:

$$R = \frac{4 * A * \pi}{L^2} \tag{2.46}$$

Elipse Las características geométricas de un objeto pueden definir la forma de la región de interés de la imagen. La elipsidad determina la similitud del contorno de la region de interés con la forma geométrica en (x_i, y_i) para $i = 1, \ldots, L$

$$ax^2 + bxy + cy^2 + dx + ey + f = 0 \tag{2.47}$$

se obtiene

$$\|Xa\| \rightarrow min \tag{2.48}$$

donde X representa una matriz con L filas, la fila i es x_i, entonces se restringe a $\|a\| = 1$, y a es la última columna de V, donde $X = USV^{T]}$, S es una matriz diagonal de la misma dimensión de X con elementos no negativos de manera decreciente. U y V son matrices singulares.

La característica de Elipse se obtiene de la siguiente manera:

$$\left(\frac{x - x_0}{a_e}\right)^2 + \left(\frac{y - y_0}{b_e}\right)^2 = 1 \tag{2.49}$$

donde:

$a_e = \frac{1}{\sqrt{sa_p}}$ representa el eje mayor

$b_e = \frac{1}{\sqrt{sb_p}}$ representa el eje menor

$s = \frac{1}{v-f}$ representa el factor de escalamiento

$$v = t^T T t$$

$$T = \begin{bmatrix} a & b/2 \\ b/2 & c \end{bmatrix}$$

$t = \frac{1}{2} T^{-1} \begin{bmatrix} d \\ e \end{bmatrix}$ representa la translación

$$a_p = a\cos^2(\alpha) + b\cos(\alpha)\sin(\alpha) + c\cos^2(\alpha)$$

$$\alpha = \frac{1}{2}\tan^{-1}\left(\frac{b}{a-c}\right)$$

$$x_0 = t_1, y_0 = t_2$$

Los ejes de elipse se definen como a_e y b_e, el centro (x_0, y_0), la orientación α. La excentricidad se define como:

$$e_x = \frac{min(a_e, b_e)}{max(a_e, b_e)} \tag{2.50}$$

Momentos

Los momentos son muy empleados en reconocimiento de imágenes, estos permiten reconocer imágenes independientemente de su rotación, traslación o inversión.

Los momentos de orden $(p+q)$ son definidos como:

$$m_{pq} = \sum_x \sum_y x^p y^q \rho(x, y). \tag{2.51}$$

donde $\rho(x, y)$ es definida por la región segmentada. Los momentos de orden pequeño describen la forma de la región. Por ejemplo m_{00} describe el area de la región segmentada, mientras que m_{01} y m_{10} definen las coordenadas x e y del centro de gravedad. Sin embargo, los momentos m_{02}, m_{03}, m_{11}, m_{12}, m_{20}, m_{21} y m_{30} son invariantes a traslación, rotación e inversión. Los momentos centrales son invariantes a desplazamiento y pueden ser calculados mediante:

$$\mu_{pq} = \sum_{i,j \in R} (i - \bar{i})^p (j - \bar{j})^q \tag{2.52}$$

donde p, q pertenecen a la región segmentada y el centro de gravedad de la región es definido por:

$$\bar{i} = \frac{m_{10}}{m_{00}}, \bar{j} = \frac{m_{01}}{m_{00}} \tag{2.53}$$

Los momentos de Hu pueden ser obtenidos mediante:

$$\phi_1 = \eta_{20} + \eta_{02} \tag{2.54}$$

$$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \tag{2.55}$$

$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \tag{2.56}$$

$$\phi_4 = (\eta_{30} - 3\eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \tag{2.57}$$

$$\begin{aligned} \phi_5 = \quad & (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + \\ & (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \end{aligned} \tag{2.58}$$

$$\begin{aligned} \phi_6 = \quad & (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + \\ & 4(\eta_{11}(\eta_{30} + \eta 12)(\eta_{21} + \eta_{03}) \end{aligned} \tag{2.59}$$

$$\begin{aligned}\phi_7 = \quad & (3\eta_{21}-\eta_{03})(\eta_{30}+\eta_{12})[(\eta_{30}+\eta_{12})^2-3(\eta_{21}+\eta_{03})^2]- \\ & (\eta_{30}-3\eta_{12})(\eta_{21}+\eta_{03})[3(\eta_{30}+\eta_{12})^2-(\eta_{21}+\eta_{03})^2]\end{aligned} \tag{2.60}$$

donde:

$$\eta_{pq} = \frac{\mu_{rs}}{\mu 00^t}, t = \frac{p+q}{2}+1$$

$$to$$

(2.62)

Descriptores de Fourier

Los descriptores de Fourier son usados para determinar la forma de la región de interés de la imagen. Dado $x(k)$ y $y(k)$ como coordenadas del *k-ésimo* pixel de la frontera de la region de la imagen de N pixeles. El *k-ésimo* pixel al rededor del contorno tiene una posición (x_k, y_k), entonces se puede describir el contorno como dos ecuaciones paramétricas:

$$x(k) = x_k y(k) = y_k \tag{2.63}$$

se consideran las coordenadas del punto (x, y) en un plano complejo, y puede ser formado como $z(k) = x(k) + Iy(k)$ y el Descriptor de Fourier(FD) de la región de interés se define como el DFT de $z(k)$

$$a(u) = \frac{1}{N}\sum_{k=0}^{K-1} s(k)e^{-j2\pi uk/K}, u = 0, 1, 2, \ldots, K-1 \tag{2.64}$$

Otras características empleadas son descriptores de elipse, convexidad de región, momentos de Flusser y orientación, en total 57 características fueron extraídas de cada imagen.

$$b(x, y) = \left\{ \begin{array}{l} 1 : si\, I(x, y) \geq T(x, y) \\ 0 : De\ lo\ contrario \end{array} \right\} \tag{2.65}$$

Momentos de Gupta

Los momentos de Gupta son invariantes a la traslación y el escalamiento [35]. Obteniendo los N pixeles del contorno descritos por un conjunto de datos ordenado $x(i), y(i), i = 1, 2, \ldots, N$. La distancia euclidiana $z(i), i = 1, 2, \ldots, n$ de un vector conectando el centroide $(\bar{x}, \bar{y})$ y el conjunto ordenado de los pixeles del contorno forman un valor singular uno dimensional de la representacion funcional del contorno. Considerándose solo los contornos cerrados, la representación secuencial resultante es circular, de la siguiente manera:

$$z(N + i) = z(i) i = 1, 2, \ldots, N \tag{2.66}$$

Dado una representación secuencial de un contorno de N-puntos $z(i), i = 1, 2, \ldots, N$ de una forma binaria $Z(x, y)$ el r-ésimo momento puede ser estimado como:

$$m_r = \frac{1}{N} \sum_{i=1}^{N} [z(i)]^r \tag{2.67}$$

y el r-ésimo momento normalizado de la secuencia del contorno se define como:

$$\bar{m}_r = \frac{m_r}{(M_2)^2} = \frac{\frac{1}{N} \sum_{i=1}^{N} [z(i)]^r}{\left[\frac{1}{N} \sum_{i=1}^{N} [z(i) - m_i]^2\right]^{r/2}} \tag{2.68}$$

donde $\bar{m}_r$ y $\bar{M}_r$ son invariantes a la traslación, rotación y escalamiento notando que las coordenadas de la región transformada $H(u, v)$ esta relacionada a $G(x, y)$ por

$$H(u, v) = AG(x, y) + B \tag{2.69}$$

las coordenadas de las variables transformadas estan dadas por:

$$\begin{bmatrix} u_i \\ v_i \end{bmatrix} = \begin{bmatrix} \alpha \cos\theta & \sin\theta \\ -\sin\theta & \alpha \cos\theta \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} + \begin{bmatrix} \beta \\ \gamma \end{bmatrix} \tag{2.70}$$

donde β y γ son las variables de traslación, α es el factor de escala, θ es el angulo de

reotación. El contorno original de la región de interés siendo G y la región transformada siendo H. Esto es:

$$G = [g(1), g(2), \ldots, g(N)] \tag{2.71}$$

$$H = [h(1), h(2), \ldots, h(M)] \tag{2.72}$$

donde:

$$\text{g(l)}=\left[(x_l-\bar{x})^2+(y_l-\bar{y})^2\right]^{1/2} h(k)=\left[(u_l-\bar{u})^2+(v_k-\bar{v})^2\right]^{1/2} \bar{x}=\frac{m_{10}}{m_{00}}, \bar{y}=\frac{m_{01}}{m_{00}} \tag{2.73}$$

donde:

$$\text{m}_{pq} = \sum_p \sum_q x_p y_q G(x, y) \tag{2.74}$$

dado el m_r^H representa la r-ésima secuencia del momento del contorno de $H(u, v)$. de las Ecuaciones 2.68, los momentos del contorno de la región $H(u, v)$ son:

$$\bar{m}_r^H = \frac{\frac{1}{M}\sum_{k=1}^{M}[h(k)]^r}{\left[\frac{1}{M}\sum_{k=1}^{M}\left[h(k)-m_i^H\right]^2\right]^{r/2}} \tag{2.75}$$

Momentos de Flusser

Los momentos invariantes con respecto a la rotación, traslación y escalamiento, al igual que Hu, los Momentos de Flusser [28] son ampliamente usados en procesamiento digital de imágenes (PDI). Los momentos complejos c_{pq} de orden $(p + q)$ de una función de imagen integrable $f(x, y)$ es definida como:

$$c_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}(x + iy)^p(x - iy)^q f(x, y)dxdy \tag{2.76}$$

donde i denota un número imaginario. Cada momento complejo puede ser expresado en terminos de momentos geométricos m_{pq} como

$$c_{pq} = \sum_{k=0}^{p} \sum_{j=0}^{q} (\frac{p}{q})(\frac{q}{j})(-1)^{q-j} \cdot i^{p+q-k-j} \cdot m^{k+j,p+q-k-j} \tag{2.77}$$

donde los momentos geométricos son definidos como:

$$m_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q \int (x, y) dx dy \tag{2.78}$$

y vice versa

$$m_{pq} = \frac{1}{2^{p+q} i^q} \sum_{k=0}^{p} \sum_{j=0}^{q} (\frac{p}{k})(\frac{q}{j})(-1)^{q-j} \cdot c^{k+j,p+q-k-j} \tag{2.79}$$

esto sigue de la definición de que solo los índices $p \geq q$ son significativos cuando se trata con momentos complejos porque $c_{pq} = c^*_{pq}$ (el asterisco denota conjugaciones complejas). En coordenadas polares,Eq 2.76 se convierte en la forma

$$c_{pq} = \int_0^{\infty} \int_0^{2\pi} r^{p+q+1} e^{i(p-q)\theta} f(r, \theta) dr d\theta \tag{2.80}$$

La Eq 2.80 implica invarianza a la rotación de la magnitud del momento $\left|c_{pq}\right|$ mientras la fase es cambiada por $(p-q)\alpha$, donde α es el ángulo de rotación. Mas precisamente, sustenta el momento de la imagen rotada

$$c'_{pq} = e^{-i(p-q)\alpha} \cdot c_{pq} \tag{2.81}$$

En ocasiones es necesario obtener características invariantes a la traslación, rotación, escalamiento y transformaciones relacionadas. Los momentos de Flusser obtienen características derivadas del momento central de segundo y tercer orden que son invariantes a transformaciones afines [28], estos pueden ser obtenidos de las siguientes ecuaciones:

$$
\begin{aligned}
I_1 &= \frac{\mu_{20}\mu_{02} - \mu_{11}^2}{\mu_{00}^4} \\
I_2 &= \frac{\mu_{30}^2\mu_{03}^2 - 6\mu_{30}\mu_{21}\mu_{12}\mu_{03} + 4\mu_{30}\mu_{12}^3 + 4\mu_{21}^3\mu_{03} - 3\mu_{21}^2\mu_{12}^2}{\mu_{00}^{10}} \\
I_3 &= \frac{\mu_{20}(\mu_{21}\mu_{03} - \mu_{12}^2) - \mu_{11}(\mu_{30}\mu_{03} - \mu_{21}\mu_{12}) + \mu_{02}(\mu_{30}\mu_{12} - \mu_{21}^2)}{\mu_{00}^7} \\
I_4 &= (\mu_{20}^3\mu_{03}^2 - 6\mu_{20}^2\mu_{11}\mu_{12}\mu_{03} - 6\mu_{20}^2\mu_{02}\mu_{21}\mu_{03} + 9\mu_{20}^2\mu_{02}\mu_{12}^2 + \\
&\quad 12\mu_{20}\mu_{11}^2\mu_{21}\mu_{03} + 6\mu_{20}\mu_{11}\mu_{02}\mu_{30}\mu_{03} - 18\mu_{20}\mu_{11}\mu_{02}\mu_{21}\mu_{12} - \\
&\quad 8\mu_{11}^3\mu_{30}\mu_{03} - 6\mu_{20}\mu_{02}^2\mu_{30}\mu_{12} + 9\mu_{20}\mu_{02}^2\mu_{21} + 12\mu_{11}^2\mu_{02}\mu_{30}\mu_{12} - \\
&\quad 6\mu_{11}\mu_{02}^2\mu_{30}\mu_{21} + \mu_{02}^3\mu_{30}^2)/(\mu_{00}^{11})
\end{aligned}
\tag{2.82}
$$

2.3.2. Características texturales

La superficie de las imágenes nos brinda información del área de estudio dentro de la imagen, que tan fina se encuentra la superficie, su propiedad áspera o lisa, que tan ondulado se encuentra, la irregularidad o alineado dentro de la región de interés de la imagen, de esta manera se podrían diferenciar objetos específicos dentro de la imagen, como podrían ser rocas, madera, zonas lisas, etc. El cálculo de la textura se obtiene de la dependencia espacial de las tonalidades de gris dentro de una matriz de frecuencias P_{ij} con que dos celdas de resolución vecinas, separadas por una distancia d, uno con una tonalidad de gris i y el otro con tonalidad gris j. La frecuencia de las matrices de dependencia espacial de escala de grises es la función de la relación angular entre las celdas vecinas de lomo se muestra a continuación**??**:

$$
\mathrm{P(i,j,0\circ)} = \{((k,l),(m,n)) \in (L_y \times L_x) \times (L_y \times L_x) \,|\, k - m = 0, |l - m| = d, \\
\mathrm{I(k,l)=i,\ I(m,n)=j}
\tag{2.83}
$$

$$\text{P(i,j,45}\circ) = \{((k,l),(m,n)) \in (L_y \times L_x) \times (L_y \times L_x) \,|\, (k-m=d, l-n=-d), \text{ or (k-m=-d,l-n=d)}, \text{I(k,l)=i, I(m,n)=j}\} \tag{2.84}$$

$$\text{P(i,j,90}\circ) = \{((k,l),(m,n)) \in (L_y \times L_x) \times (L_y \times L_x) \,\big|\, |k-m|=d, l-n=-d), \text{l-n=0,I(k,l)=i,I(m,n)=j}\} \tag{2.85}$$

$$\text{P(i,j,135}\circ) = \{((k,l),(m,n)) \in (L_y \times L_x) \times (L_y \times L_x) \,|\, (k-m=d, l-n=-d), \text{ or (k-m=-d,l-n=d)}, \text{I(k,l)=i,I(m,n)=j}\} \tag{2.86}$$

donde denota el número de elementos en el conjunto, las ecuaciones se pueden definir de manera general como:

$$p((k,l),(m,n) = \max\ \{|k-m|, |l-n|\} \tag{2.87}$$

2.3.3. Características cromáticas

2.4. Técnicas de clasificación

En el área de Inteligencia Artificial existen diversos métodos de clasificación, la selección de un buen método de clasificación es crucial para la obtención de buenos resultados en la experimentación, aqui se abordarán los métodos usados en las experimentaciones.

2.4.1. SVM

Las máquinas de vectores de soporte (SVM) es uno de los métodos de clasificación más usados para el modelado y clasificación de información, recientemente clasificado junto con los *métodos kernel*. Las ventajas de las SVM una excelente capacidad de generalización, que concierne en la capacidad de clasificar correctamente ejemplos que no están dentro de los rasgos de espacio usados en el entrenamiento[60]. Este método de clasificación es ampliamente usado en la bioinformática (entre otras disciplinas) debido a su exactitud, habilidad para tratar con cantidades muy grandes de información, como expresiones genéticas, y la flexibilidad en la modelación de diversas fuentes de información.

Cuando se entrena una SVM se necesitan realizar algunas decisiones: como procesar la información, que método de *kernel* usar y finalmente preparar los parámetros de la SVM 2.6.

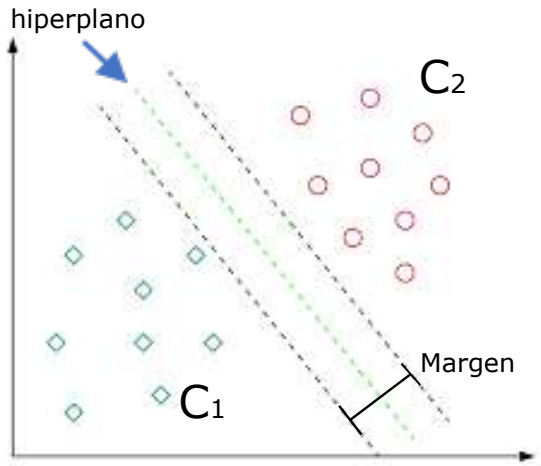

FIGURA 2.6: Hiperplano de separación (Towards Data Science)

Clasificadores Lineales

La información para un problema de aprendizaje de dos clases consiste de objetos etiquetados/clasificados con una de las dos etiquetas correspondientes a las clases que se tienen, por conveniencia se asume que las etiquetas $+1$ son ejemplos positivos y -1 son ejemplos negativos[10].

Teniendo el vector x con componentes x_i. Estos componentes denotan los $i-ésimos$ vectores de un conjunto de datos $\{(x_i, y_i)\}_i^n = 1$, donde y_i es la etiqueta asociada a x_i.

Los objetos x_i son llamados modelos o ejemplos. Se asume que los ejemplos pertenecen a un conjunto X. Inicialmente se toman los ejemplos como vectores, pero una vez introducidos en el modelo, esta afirmación podría ser diferente pudiendo ser entonces conjuntos u objetos discretos.

El concepto principal para definir un clasificador es el producto punto (*dot product*) entre dos vectores, también conocido como producto escalar o producto interno, definido como $w^T x = \sum_i w_i x_i$. Un clasificador lineal está basada en la función

$$f(x) = w^T x + b \tag{2.88}$$

El vector w es conocido como el vector de peso, y b es la tendencia. Considerando el caso $b = 0$, el conjunto de puntos de x tal que $w^T x = 0$ son todos los puntos que son perpendiculares a w y pasan por el origen en el plano de dos dimensiones, un plano de tres dimensiones, y en un hiperplano. La tendencia b traslada el hiperplano fuera del origen. El hiperplano

$$\{x : f(x) = w^T x + b = 0\} \tag{2.89}$$

Divide el espacio en dos partes, el signo de la función discriminativa f(x) denota el lado del hiperplano en donde se encuentra el punto. El límite entre las regiones positiva y negativa del plano son llamadas el límite de decisión del clasificador, como se muestra en la Figura 2.6. El límite de decisión definido por el plano es lineal porque los ejemplos son lineales.

Clasificadores no lineales

En diversas aplicaciones los clasificadores no lineales proveen una mejor exactitud. Aun así los clasificadores lineales tienen ventajas, una de ellas es que cuentan con algoritmos de entrenamiento simples. Podemos fácilmente observar el comportamiento de ambos tipos de clasificadores mapeando la información de un conjunto X usando la función no lineal $\phi : X \longrightarrow F$. En el espacio F la función discriminante es:

$$f(x) = w^T \phi(x) + b \tag{2.90}$$

Considerando el caso de un espacio bidimensional

$$\phi(x) = (x_i^2, \sqrt{2} x_1 x_2, x_2^2)^T \tag{2.91}$$

Representando un vector en términos de monomios de segundo grado. En este caso

$$w^T\phi(x) = w_1x_1^2 + \sqrt{2}w_2x_1x_2 + w_3x_2^2 \tag{2.92}$$

Resultando en un límite de desición para el clasificador, $f(x) = w^Tx + b = 0$, la cual es una sección cónica.

La aproximación al introducir rasgos no lineales no es fácil de escalar debido al número de rasgos de entrada. Cuando se mapea el ejemplo anterior la dimensionalidad de las características del espacio F es cuadrática en comparación de la dimensionalidad del espacio original. Esta complejidad cuadrática es factible con un espacio dimensional de poca información; pero cuando se trata con la información de expresiones genéticas, la cual puede contener miles de dimensiones, el complejo cuadrático en el número de dimensiones no es aceptable. Los métodos *kernel* resuelven este caso evitando el paso del mapeo de información de grandes dimensiones caracterí sticas-espaciales.

Supongamos que el peso del vector puede ser expresado como una combinación de ejemplos lineales de entrenamiento.

$$w = \sum_{i=1}^{n} \alpha_i x_i \tag{2.93}$$

entonces:

$$f(x) = \sum_{i=1}^{n} \alpha_i^T x + b \tag{2.94}$$

En el espacio de características F se forma la siguiente expresión:

$$f(x) = \sum_{i=1}^{n} \alpha_i \phi(x_i)^T \phi(x) + b \tag{2.95}$$

La representación en términos de α_i es conocida como la representación del límite de decisión. Como se indicó anteriormente el espacio F puede ser de gran dimensión haciendo este truco poco práctico a no ser que la función de *kernel* $k(x, x\prime)$ se defina como:

$$k(x, x\prime) = \phi(x)^T\phi(x\prime) \tag{2.96}$$

Pudiendo procesarlo eficientemente. En términos de la función de *kernel* la función de discriminación es:

$$f(x) = \sum_{i=1}^{n} \alpha_i (x.x_i) + b \tag{2.97}$$

Margen geométrico

Para un hiperplano dado entendemos por x_+ o x_- el punto más cercano entre los ejemplos negativos o positivos. La norma de un vector w se denota por $\|w\|$ es su longitud y está dada por $\sqrt{w^T w}$. Un vector unitario $\widehat{w}$ en dirección de w está dado por $w/\|w\|$ teniendo $\|w\| = 1$. De consideraciones geométricas simple consideramos un hiperplano f con respecto a un conjunto de datos D de la forma:

$$m_D(f) = \frac{1}{2}\widehat{w}^T(x_+ - x_-) \tag{2.98}$$

donde $\widehat{w}$ es un vector unitario en dirección de w, asumiendo que x_+ y x_- son equidistantes del límite de decisión.

$$f(x) = w^T x_+ + b = a \tag{2.99}$$

$$f(x) = w^T x_- + b = -a \tag{2.100}$$

para una constante $a > 0$.

Note que, multiplicando nuestros puntos por una constante, nuestro margen se incrementara en la misma cantidad, mientras que en realidad nuestro margen no cambio, solo se cambiaron las unidades con la cual lo limitamos. Para hacer el margen geométrico significativo ajustamos el valor de la función de decisión a los puntos más cercanos del hiperplano, y establecemos $a = 1$ Adicionando las dos ecuaciones y dividiéndolas por $\|w\|$ obtenemos:

$$m_D(f) = \frac{1}{2}\widehat{w}^T(x_+ - x_-) = \frac{1}{\|w\|} \tag{2.101}$$

Clasificadores de margen máximo

Los clasificadores de margen máximo es una función discriminante que maximiza el margen geométrico $\frac{1}{\|w\|}$ que es equivalente a minimizar $\|w\|^2$.Tomando lo anterior

$$\begin{aligned} min \qquad & \frac{1}{2}\|w\|^2 \\ sujeto\ a \quad : \quad & y_i(w^T x_i + b) \geq 1 \qquad i = 1, \ldots, n \end{aligned} \tag{2.102}$$

Las coacciones en esta formulación aseguran que el máximo clasificador clasificara cada ejemplo correctamente, lo cual es posible asumiendo que la información es linealmente separable. En la práctica en ocasiones la información no es linealmente separable; y aunque esta se pudiera separar, el margen más adecuado puede ser alcanzado permitiendo que el clasificador no clasifique algunos puntos. Para permitir errores reemplazamos las coacciones desiguales de la ecuación anterior

$$y_i(w^T x_i + b) \geq 1 - \xi_i \quad i = 1, \ldots, n \tag{2.103}$$

donde $\xi_i \geq 0$ son variables de poca utilidad que permiten a un ejemplo estar dentro del margen de $0 \leq \xi_i \leq 1$(también llamado margen de error) o no ser clasificados($\xi_i > 1$).Por lo tanto la variable no es clasificada cuando su valor es mayor a $1,\sum_i \xi_i$ es el conjunto de ejemplos que no son clasificados.

El objetivo de maximizar el margen minimizando $\frac{1}{2}\|\|^2$ es demostrado en el la ecuación $C\sum_i \xi_i$ para penalizar el error de margen y los elementos no clasificados.

La constante C coloca la importancia relativa de maximizar el margen y minimizar la cantidad de ejemplos de poca utilidad. Esta formulación es conocida como margen blando SVM (*soft-margin*). Utilizado el método de multiplicadores de Lagrange obtenemos la formulación dual

La formulación dual guía una expansión del vector de peso en términos de los ejemplos de entrada:

$$w = \sum_{i=1}^{n} y_i \alpha_i x_i \tag{2.104}$$

Los ejemplos x_i para cada $\alpha_i > 0$ son los puntos que no se encuentran en el margen cuando es usado el margen blando. Estos son los llamados vectores de soporte. La expansión en términos de los vectores de soporte es algunas veces escasa, y el nivel de escases es mayor en el porcentaje de error del clasificador. El problema de la formulación de la optimización de la SVM depende de la información solo a través de los

productos punto. El producto punto se puede por lo tanto reemplazar por una función *kernel* no lineal, así realizando un amplio margen de separación en el espacio de características. El problema de la optimización de las SVM era resuelto tradicionalmente con la formulación dual, pero recientemente se ha demostrado que la ecuación fundamental (Ec (3)), puede encaminar a eficientes métodos de aprendizaje basados en $kernel$.

2.4.2. Clasificación bayesiana

El método de clasificación bayesiana es un método que presenta buenos resultados al igual que métodos más utilizados como árboles de decisión y redes neuronales implementándose en cierto tipo de conjuntos de datos. También proveen una perspectiva muy útil de como entender otros algoritmos de clasificación que no son tan explícitos en el manejo de probabilística[62].

En la clasificación bayesiana se toma en cuenta lo siguiente:

- Cada dato nuevo de entrenamiento puede aumentar o disminuir la probabilidad de que la hipótesis es correcta
- El conocimiento *apriori* puede ser combinado con el conjunto de datos para determinar la probabilidad de la hipótesis
- Con la combinación de predicciones de múltiples hipótesis se puede realizar una clasificación de un nuevo dato observado
- Aun en casos de datos intratables en comparación de otros métodos el clasificador bayesiano puede proporcionar un estándar para realizar una decisión de clasificación optima.

La probabilidad inicial de que la hipótesis h se acepte se denota como $P(h)$, también llamada probabilidad apriori de h. Si se tiene un conocimiento previo tendrá mayor valor, de lo contrario se le dará el mismo valor a $P(h)$ de que ocurra o no ocurra el evento de hipótesis h.

El teorema de bayes se formula de la siguiente manera:

$$P(h \mid D) = \frac{P(D \mid h)P(h)}{P(D)} \tag{2.105}$$

donde $P(D)$ es la probabilidad *aposteriori* ed que se cumpla la hipótesis.

Máximo *aposteriori(MAP)*

$$h_{MAP} = \arg\max_{h \in H} P(h \mid D) \tag{2.106}$$

$$= \arg\max_{h \in H} \frac{P(D \mid h)P(h)}{P(D)} \tag{2.107}$$

$$= \arg\max_{h \in H} P(D \mid h)P(h) \tag{2.108}$$

Para atributos independientes entre sí, se realiza:

$$P(a_1, a_2, \ldots, a_n \mid v_j) =_i P(a_i \mid v_j) \tag{2.109}$$

$$vnb = \arg\max_{v_j \in V} P(v_j)_i P(a_i \mid v_j) \tag{2.110}$$

2.4.3. Redes neuronales

En inteligencia artificial las redes neuronales son ampliamente usadas, ya que tienen amplio poder de generalización para resolver problemas de clasificación de conjunto de datos diversos. Las redes neuronales son un conjunto interconectado de neuronas artificiales en capas, siendo cada uno de estas una primitiva elemental computacional [12].

Las neuronas se pueden clasificar en 3 tipos, de entrada, ocultas y de salida. Las redes neuronales pueden ser supervisadas, es decir, se intenta aproximar la función que describe de mejor el conjunto de datos empleado. Para un conjunto de datos $T = (X_k, D_k)_{k=1}^{Q}$ obtenido del espacio de patrones donde el vector $X_k \in R^n$ relacionado al vector de salida $D_k \in R^p$. El objetivo de un algoritmo de aprendizaje supervisado es describir el comportamiento de una función desconocida $f : R^n \rightarrow R^p$.

La neurona básica puede ser representada de la siguiente manera, si se tienen n entradas con n pesos asociados en las entradas, entonces la i-ésima unidad calcula alguna función f de la sumatoria de entradas y sus pesos dado por:

$$y_i = \sum_{j=1}^{n} w_{ij} y_j \tag{2.111}$$

donde w_{ij} se refiere a los pesos desde la j-ésima a la i-ésima unidad. La función f es la función de activación de las unidades. En el caso mas simple f es la función identidad donde la unidad de salida es idéntica a la unidad de entrada, a esto se le llama unidad

lineal. Una variante compara el cálculo de la suma en relación a un valor de la neurona llamado umbral. El proceso de umbral se realiza por comparación, si la sumatoria es mayor que el umbral entonces la salida de la neurona sera 1, de lo contrario si la salida es menor al umbral, la salida será 0. Por lo tanto, la salida y_i de la i-ésima neurona se puede describir como:

$$y_i = f(\sum_{j=1}^{n} w_{ij} y_i + b) \tag{2.112}$$

donde b es el bias o compensación de la neurona y f es una función de paso para el umbral:

$$f(x) = 1x > 0 \tag{2.113}$$

$$f(x) = 0x <= 0 \tag{2.114}$$

Perceptrón multicapa

Comúnmente una neurona con diversas entradas podría no ser suficiente, normalmente se necesitan de más trabajando de forma paralela, a esto se le puede llamar capa. Una capa de S neuronas tiene R entradas conectadas a cada una de las neuronas de entrada y se tiene una matriz de pesos tiene S filas.

La capa incluye la matriz de pesos, las sumatorias, el vector de bias. Cada elemento del vector de entrada p es conectado a cada neurona a través de la matriz de pesos W. Cada neurona tiene un bias b, una sumatoria, una función de transferencia f y una salida a_i. Tomados de manera conjunta como las salidas del vector a. Es común obtener un número de entradas a la capa diferente al número de neuronas ($R \neq S$).

Si se considera una red neuronal con muchas capas, cada una de ellas tiene su propia matriz de pesos W, su propio vector de bias b, un vector de entradas n y vector de salida a. Así tenemos diferentes matrices y vectores por cada capa, obteniendo:$a^1 = f^1(W^1p+b^1)$,$a^2 = f^2(W^2p+b^2)$ y así sucesivamente, obteniendo $a^n = f^n(W^np+b^n)$

Árbol de decisión

El árbol de decisión es un clasificador muy usado en minería de datos, debido a su sencillez, es una buena herramienta para procesos de elección entre diversos cursos de acción [22].

El árbol de decisión es un árbol donde los nodos internos son asociados con una decisión, y los nodos hoja son generalmente asociados con la etiqueta de salida. Un nodo intermedio prueba una o más valores atributo llevando a 2 o más conexiones o ramas. Cada rama o conexión está asociada con el posible valor de decisión. Para la versión más simple, cada nodo recibe un estatuto de decisión que sera tomado en la comparación, entonces el nodo tendrá dos salidas, una será si o "verdadero", y la segunda no o "falso". El nodo es asociado a una regla o prueba basada en los valores ocaracterísticas del conjunto de datos. Estas pruebas se pueden categorizar de la siguiente manera: enumerate

Axis-parallel test : En este caso la prueba se realiza de la forma $x > a_0$, donde x es la característica y a_0 es el valor de umbral

Prueba basada en combinación lineal de características: En esta la prueba se realiza de la forma:

$$\sum_{i=1}^{d} a_i x_i > a_0 \tag{2.115}$$

donde x_i es la i-ésima característica, y a_i es el peso asociado a el. Esta prueba envuelve una combinación lineal de los valores de características y el hiper-plano correspondiente no paralelo en ningún eje.

Prueba basada en combinación de características no lineales: Esta es la forma mas general de prueba posible, de la forma:

$$f(x) > 0 \tag{2.116}$$

donde $f(x)$ es cualquier función no lineal de los componentes de x

2.5. Técnicas de validación cruzada

En las investigaciones basadas en el Lenguaje Máquina, entre otras, se hace de vital importancia la comparación de resultados obtenidos a partir de diferentes clasificadores, por esta razón algunos investigadores se hán abocado al estudio de diferentes algoritmos y métodos comparativos para multiclasificadores. En[21] se realiza una investigación de clasificación de datos imbalanceados con técnicas IPADE-ID. En [31] estudian la multiclasificación de lenguaje maquina basado en algoritmos genéticos, utilizando para ello los siguientes algoritmos de comparación de multiclasificación: Pittsburgh Genetics interval Rule Learning Algorithm(Pitts-GIRLA), XCS, Generic Algorithm based Classifier System (GASSIST-ADI), Hierarchical Decision Rules (HIDER). Para la comparación de resultados de diferentes clasificadores se proponen los siguientes métodos para comparar los resultados obtenidos:

- Porcentaje de clasificación.
- Medidas Cohen´s kappa

El primer método es el más común, en el cual se toman los conjuntos clasificados correctamente del total de conjuntos.

El método Cohen´s kappa se utiliza para comparar el desempeño de los clasificadores tomando en cuenta el exito aleatorio del clasificador como estándar. Este método se basa en la matriz de confusión del clasificador.

La diferencia entre estos dos métodos es el marcador de clasificaciones correctas, el primer método cuantifica todos los datos clasificados positivos de todas las clases, en cambio el método kappa numera las clasificaciones positivas por cada clase, de esta manera la cuantificación es menos sensible al factor aleatorio de las diferentes simulaciones generadas por los clasificadores en cada clase.

Para poder medir y comparar los resultados obtenidos de diversos clasificadores primero es necesario distinguir que existen dos tipos de análisis: el análisis de un conjunto de datos y el análisis de multiconjuntos de datos. El análisis de un conjunto de datos se da cuando se obtienen resultados de dos o más algoritmos de clasificación utilizando un solo conjunto de datos, el otro tipo de análisis es aquel en el que se tienen dos o más conjuntos de datos y son implementados dos o mas algoritmos para cada uno de manera simultánea. También cabe aclarar que existen los análisis paramétricos y los no paramétricos, la diferencia es el nivel de mediciones representadas por el conjunto de datos, de esta manera el análisis paramétrico tiene como característica que los valores del conjunto pertenecen a un rango.

Para saber que en efecto nuestro conjunto de datos es paramétrico se tienen que revisar las siguientes condiciones:

- Independencia: Se dice que dos eventos son independientes cuando un evento ocurre y este no modifica la probabilidad de que el otro evento ocurra.
- Normalidad: Una observación es normal cuando su comportamiento sigue una forma de distribución gaussiana, con un valor μ y una varianza σ^2.

Para revisar estas condiciones se utilizan los métodos *Saphiro-Wilk*(SW) el cual analiza el nivel de simetría y forma de la curva para poder calcular la diferencia que existe con respecto a la distribución gaussiana obteniendo un valor p de la suma de los cuadrados de las discrepancias; y *D 'Agostino Pearson*(DP) que calcula la oblicuidad y forma de la curva para cuantificar que tan lejos está la distribución de los datos a la distribución Gaussiana en términos de asimetría y forma calculando p a partir de estas discrepancias.

Se aplican los métodos DP y SW al conjunto de datos considerando un nivel de significancia α=0.5, al aplicarlos se obtiene como resultado que el conjunto de datos no satisface las características necesarias. Los investigadores están familiarizados con las pruebas de comparación de datos paramétricos y no paramétricos, usando pruebas como *t*-test y Wilcoxon, la aplicación de este tipo de pruebas es correcto cuando se quiere saber las diferencias entre los dos métodos, sin embargo, no se deben usar cuando se quieren comparar más de dos métodos. En la comparación de dos métodos existe un error asociado que incrementa de acuerdo al número de comparaciones realizadas. A este error se le conoce como the *family-wise error rate*(FWER), definido como la probabilidad de tener al menos un error en la familia de hipótesis. Algunos autores utilizan la corrección Bonferroni para poder aplicar la prueba *t*-test.

La prueba Wilcoxon para datos no paramétricos nos permite detectar diferencias significantes entre el comportamiento de dos clasificadores. Siendo la diferencia d_i de los resultados en la *i-ésima* salida de los dos clasificadores de N_{ds} conjuntos de datos. Las diferencias son ranqueadas(*rank*) de acuerdo a sus valores absolutos, en caso de un empate entre ambos se realiza un promedio de sus resultados. Teniendo a R^+ como la suma de resultados del primer algoritmo y a R^- como la suma de resultados del siguiente algoritmo entonces:

$$R^+ = _{di>0}\, rank8di) + 1/2_{di=0} rank(di)$$

$$R^- = _{di>0}\, rank8di) + 1/2_{di=0} rank(di)$$

Siendo T la mas pequeña de las sumas entre R^+ y R^-

$$T = \text{mín}(R^+, R^-)$$

Si T es menor o igual al valor de la distribucion Wilcoxon la hipótesis nula es rechazada. La prueba de Wilcoxon es más sensible que *t*-test, de esta manera esta prueba es mas segura ya que esta no asume una distribución normal en nuestro conjunto de datos. Cuando los supuestos del *t*-test son conocidos, la prueba Wilcoxon es menos poderosa que *t*-test, sin embargo, cuando los supuestos son violados, la prueba Wilcoxon puede ser más poderosa que *t*-test.

Si se quiere realizar una comparación de más de dos algoritmos aplicados al conjunto de datos, como por ejemplo comparar el desempeño de un algoritmo de clasificación propuesto y comparar su desempeño con los preexistentes existen pruebas como la prueba *Friedman[8]* y *post-hocs.*

El test Friedman bajo la hipótesis nula inicia dando por hecho que todos los algoritmos son equivalentes, para rechazar tal hipótesis se debe presentar una diferencia entre los algoritmos en cuestión. Enumera de mayor a menor los algoritmos de acuerdo a su resultado, es decir los ranquea dándole al mejor algoritmo la posición 1, al segundo la 2, y así hasta colocar todos los algoritmos usados, en caso de empates se realiza un promedio de su ranking, y posteriormente se aplica lo siguiente:

$$Rj = \frac{1}{Nds} \sum_i r_i^j \tag{2.117}$$

$$X_F^2 = \frac{12 N_{ds}}{k(k+1)} \left[\sum jR_j^2 - \frac{k(k+1)^2}{4} \right] \tag{2.118}$$

Bajo la hipótesis nula que dice que todos los algoritmos son equivalentes entonces sus posiciones o ranks (R_j) deben ser iguales y su distribución de acuerdo a X_F^2 con $k-1$ grados de libertad.

El test de *Iman and Davenport* se deriva del test de Friedman Una vez concluido aceptando o rechazando la hipótesis nula se implementa el test *post-hoc* para determinar si el algoritmo propuesto tiene un desempeño mejor al resto de los algoritmos con que se compara.

Uno de los *post-hoc* tests más simples es el *Bonferroni-Dunn* test, que es usado cuando se quiere compara un algoritmo propuesto sobre los algoritmos de control, con este método nosotros sabemos si la diferencia entre los algoritmos es significativamente diferente si el promedio correspondiente es al menos tan bueno como su diferencia critica(CD):

$$CD = q_x \sqrt{\frac{k(k+1)}{6N}} \tag{2.119}$$

El test de *Holm* es un procedimiento de comparación múltiple utilizando la siguiente ecuación:

$$z = (Ri - Rj)/\sqrt{\frac{k(k+1)}{6N_{ds}}} \tag{2.120}$$

donde el valor z es usado para encontrar la probabilidad en la tabla de distribución normal propuesta por el método.

El test de Holm va un paso mas adelante ya que prueba la hipótesis secuencialmente ordenadas por su significancia p. el test compara cada p_i con $\alpha/(k-i)$ iniciando con el valor p mas significativo. si p1 es menor que $\alpha/(k-i)$ la hipótesis es rechazada y procedemos a comparar p_2, y así sucesivamente.

El método de *Hochberg* trabaja de manera opuesta al método anterior ya que inicia comparando la última p.

Todos estos métodos son usados para comparar los diferentes algoritmos propuestos en un estudio, y de esta manera comprobar la hipótesis nula, estas hipótesis son probadas en pares, de esta manera se puede entender que no todas las combinaciones de hipótesis falsas y verdaderas son posibles[26]. Si se tienen tres hipótesis c_1,c_2,c_3 y al menos una de ellas es falsa al menos otra debe ser falsa, teniendo en cuenta esto *Shaffer* propuso dos procedimientos para ajustar el valor de α:

- Shaffer static procedure:
- shaffer dynamic procedure:

Bergmann y Hommel propusieron también un método elemental para encontrar todas las hipótesis que no pueden ser rechazadas, por ejemplo, de un conjunto de hipótesis($h1, h2, h3$) se realizan todas las comparaciones en pares posibles, de cada par de hipótesis se agrupan las que pueden ser posibles al mismo tiempo, a estas se les

conoce como conjuntos exhaustivos y se toma como el conjunto E, de esta manera el algoritmo Bergmann-Hommel reduce el tiempo de búsqueda ya que todos los conjuntos de comparaciones que no pretenezcan al conjunto A son rechazadas.

$$A = \cup\{I : Iexhastuve, \min\{Pi : i \in I\} > \alpha/|I|\} \tag{2.121}$$

De tal manera en toda investigación en donde se propone un nuevo algoritmo de clasificación comparándolo con otros preestablecidos de control es necesario realizar una comparación de ellos implementando técnicas de prueba de hipótesis para averiguar si existen diferencias significantes entre el método propuesto y los métodos de control, iniciando por determinar si el conjunto de datos es paramétrico o no-paramétrico, dependiendo si el experimento realizado se implementa un solo conjunto de datos o diversos conjuntos de datos para diversos algoritmos de clasificación, la posterior aplicación de un procedimiento post-hoc y poder comprobar la hipótesis nula o rechazarla.

2.5.1. Interpolación de Datos

En la vida real los métodos para construcción de curvas algunas veces inician con puntos y vectores, a continuación se desarrollaran algunas propiedades de estos elementos matemáticos[80].

Los puntos y los vectores son entes matemáticos diferentes. Un punto no tiene dimensiones; el solo representa un lugar en el espacio. Un vector, por otro lado, solo tienen atributos de dirección y magnitud. La gente tiende a confundir puntos y vectores porque es natural confundir un punto p con el vector v que va desde el origen hasta el punto p

Curvas

En la práctica, las curvas son especificadas por el usuario en términos de puntos y son construidas en un proceso interactivo. Inicialmente se introducen las coordenadas de puntos y se proyectan para generar una imagen[80].

Una columna x de $\Re^d$ cuyas coordenadas dependen de un parámetro t recorre una curva paramétrica[68]:

$$x(t) = \begin{bmatrix} {}_1(t) \\ . \\ . \\ . \\ x_d(t) \end{bmatrix} \tag{2.122}$$

Usualmente pensamos en $x(t)$ como una curva de puntos, en particular si las funciones coordenadas $x_i(t)$ son polinomios de grado menor o igual que n entonces $x(t)$ es una curva polinómica de grado n en t. El gráfico de una función $x(t)$ es una forma espiral

$$x(t) = \begin{bmatrix} t \\ x(t) \end{bmatrix} \tag{2.123}$$

Las curvas descritas por los gráficos de funciones se denominan curvas funcionales. una columna x que depende de dos parámetros s y t, describe una superficie paramétrica.

$$x(s,t) = \begin{bmatrix} {}_1(s,t) \\ . \\ . \\ . \\ x_d(s,t) \end{bmatrix} \tag{2.124}$$

La superficie se denomina polinómica de grado total n si los x_i si los polinomios de grado total, menor o igual que n en s y t y por lo menos uno de los x_i tiene grado total n.

Dados cuatro puntos (bidimensional o tridimensional) $P1, P2, P4, P4$, se busca una curva que pase a través de estos puntos y tiene la forma:

$$P(t) = at^3 + bt^2 + ct + d = (t^3, t^2, t, 1)(a, b, c, d)^T = T(t)A \text{for} 0 \leq t \leq 1 \tag{2.125}$$

donde cada uno de los cuatro coeficientes a, b, c, d es un par, $T(t)$ es el vector fila $(t^3, t^2, t, 1)$ y A es el vector columna $(a, b, c, d)^T$, y las incógnitas son a, b, c, d.

Ya que los cuatro puntos pueden localizarse en cualquier parte, se puede hacer solo una supocisión que P_1 y P_4 son los puntos extremos de la curva y P_2 y P_3 son puntos

intermedios de la curva, entonces se pueden usar los valores equidistantes del parámetro t. Entonces tenemos las siguientes ecuaciones

$$\text{P(0)=P}_1, P(1/3) = P_2, P(2/3) = P_3 \text{ y } P(1) = P_4 \tag{2.126}$$

, esto es:

$$a(0)^3 + b(0)^2 + c(0) + d = P_1 \tag{2.127}$$
$$a(1/3)^3 + b(1/3)^2 + c(1/3) + d = P_2 \tag{2.128}$$
$$a(2/3)^3 + b(2/3)^2 + c(2/3) + d = P_3 \tag{2.129}$$
$$a(1)^3 + b(1)^2 + c(1) + d = P_4 \tag{2.130}$$

y la solución de la ecuación es:

$$a = -(9/2)P_1 + (27/2)P_2 - (27/2)P_3 + (9/2)P_4 \tag{2.131}$$
$$b = 9P_1 - (45/2)P_2 + 18P_3 - (9/2)P_4 \tag{2.132}$$
$$c = -(11/2)P_1 + 9P_2 - (9/2)P_3 + P_4 \tag{2.133}$$

substituyendo en la ecuación2.125

$$P(t) = (-4{,}5t^3 + 9t^2 - 5{,}5t + 1)P_1 + (13{,}5t^3 - 22{,}5t^2 + 9t)P_2 + \tag{2.134}$$
$$(-13{,}5t^3 + 18t^2 - 4{,}5t)P_3 + (4{,}5t^3 - 4{,}5t^2 + t)P_4 \tag{2.135}$$

Como los cuatro puntos son arbitrarios, las cuatro ecuaciones pueden ser escritas como la ecuación simple $at^3 + bt^2 + ct + d = 1$ para cualquier t. Esta solución puede ser entonces $a = b = c = 0$ y $d = 1$

Entonces se concluye que cuando los cuatro puntos P_i son 1 a debe ser cero

Curvas de Bézier

La unicidad del polinomio simétrico y su relación con la presentación de Bezier está dada por el siguiente teorema, el cual se generaliza de la siguientes maneras**??**:

Para cada curva polinómica $b(u)$ de grado $\leq n$ existe un único polinomio simétrico de n variables $b[u_1, ..., u_n]$ el cual es afín a su diagonal satisface $b[u, ..., u] = b(u)$

$$b_i = b[a^{n-i}, .., ab^i, .., b], i = 0, ..., n \tag{2.136}$$

son los puntos de Bezier de $b(u)$ sobre $[a, b]$

El cálculo de los puntos de Bezier sobre $[a, c]$ y $[c, b]$ se denomina subdivisión. Al subdividir respectivamente una curva polinómica $b(u)$ se genera una partición $[a_0, a_1], [a_1, a_2], ..., [a_{k-1}, a_k]$ de su dominio. La unión de los polígonos de Bezier sobre los subintervalos se denomina el polígono de Bezier compuesto de b sobre $[a_0, a_1, ..., a_k]$. En general el polígono compuesto consta de $kn + 1$ vértices.

Capítulo 3

Datos

En este Capítulo se desarrolla la adquisición de información.

3.1. Imágenes de señas estáticas (Dactilología)

Como se mencionó anteriormente las investigaciones se pueden dividir en dos grupos, imágenes estáticas, las cuales comprenden el abecedario, que en el caso del LSM consta de 29 letras, aunque en general las letras se toman como imágenes donde no existe un movimiento de las manos, existen letras que sí presentan movimiento para su concepción:

- j
- k
- ñ
- q
- x
- z

de las cuales la letra *j* y *z* presentan un movimiento amplio el cual en investigaciones del abecedario del LSM se han excluido debido a su formación con el amplio movimiento que necesita. El resto de las letras solo tienen un ligero movimiento hacia arriba y abajo o hacia los lados para denotarlo, por lo tanto se toman como la misma forma de la mano simplemente con una pequeña variación de lugar, el cual se puede compensar con los momentos invariantes a translación o distancia. Por esta razón es que en la literatura se encuentran investigaciones del abecedario del LSM donde se reconocen 27 señas aunque el abecedario completo consta de 29, como se muestra en la

Figura3.1.

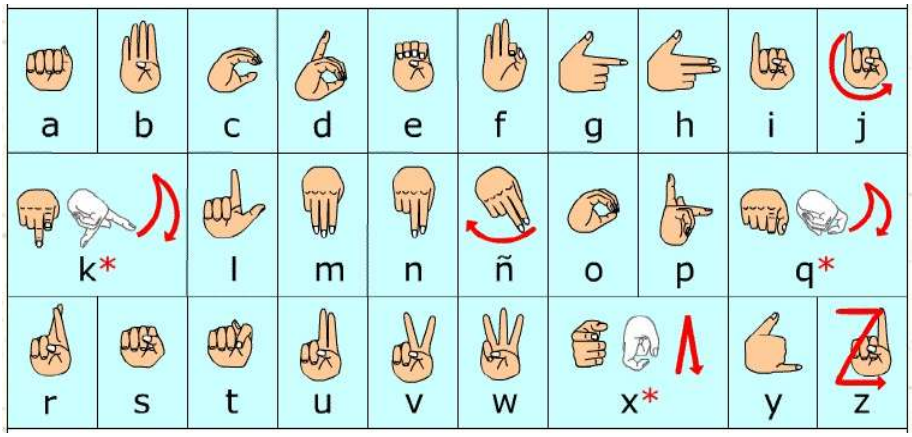

FIGURA 3.1: Abecedario del LSM (www.guiadisc.com)

Para esta investigación se tomaron imágenes de las 29 letras del abecedario, se solicitó la ayuda por escrito a la institución IPPLIAP(Instituto Pedaggíco para Problemas del Lenguaje, IAP) [38], donde se imparten clases de nivel básico a personas de problemas auditivos o del habla, también se solicitó por escrito la ayuda del Centro Educativo para el Sordo(CES). Para la adquisición de las imágenes se utilizó una cámara sony, un tripié y un fondo color gris, las condiciones de iluminación de las imágenes fueron variables, ya que las imágenes fueron tomadas tanto al aire libre como dentro de aulas.

Los rangos de edad de las personas que participaron para la grabación de las señas van desde niños de 7 años, hasta adultos de 25 años, en su mayoría siendo alumnos de las escuelas a donde se solicitó el apoyo, participando también profesores y personal de las escuelas.

Como se a mencionado anteriormente, la vestimenta tiene mucho peso para la detección y traducción de las señas, asi como los cambios de iluminación. Durante la grabación de las señas del abecedario las condiciones de iluminación no pudieron ser controladas, debido a los cambios de locaciones, por lo tanto, en las imágenes se tiene diferentes intensidades de luminosidad.

La vestimenta de las personas también es relevante para la detección de las señas, los participantes tenían vestimenta variada, portaban chamarras, playeras y camisas de manera general, sin embargo, para este conjunto específico se realizó un recorte de

las imágenes, ya que al ser señas estáticas solo importa la zona de la mano, ya que su configuración es lo que denota el significado.

También tomaron imágenes de los números, de la siguiente manera, se solicitó a los participantes que realizaran las seãs de los números del 0 al 9, y posteriormente las señas de las decenas del 10 al 100 3.2. Para la adquisición de las imágenes del abecedario se utilizó un fondo de color gris, las personas que participaron portaban uniforme escolar, no se cuidó la iluminación del lugar, ya que el fondo se colocó en una pared en el patio de la escuela, por lo tanto, se observan variaciones de iluminación en las imágenes.

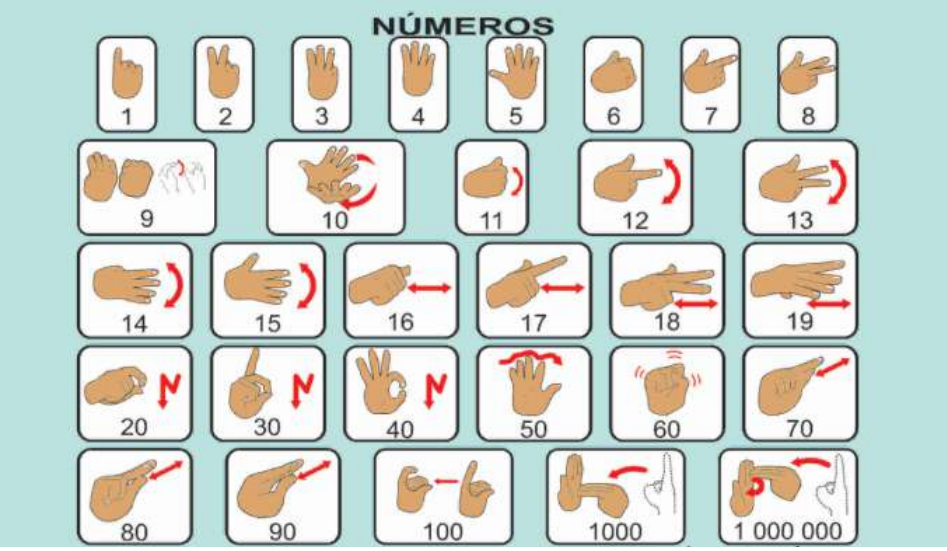

FIGURA 3.2: Números del LSM (Sumas y restas con LSM)

3.2. Palabras del LSM (IDEOGRAMAS)

Para la selección de palabras del LSM se seleccionaron junto con personal de centro de enceñanza CES las palabras que se utilizan más comúnmente y que se enseñan en etapas básicas de la educación a sordomudos. En total se seleccionaron 249 palabras del LSM como se muestra en la tabla3.1, este listado se definió con la ayuda del director del centro CES, ya que en su experiencia, estas palabras son algunas de las palabras básicas que las personas sordomudas aprenden en sus estudios. Se intento abarcar diversos tópicos para que fuera un listado suficientemente variado para poder servir como una base para un futuro diccionario del LSM.

Palabras seleccionadas de LSM	
Saludos	Buenos días, Buenas tardes, Buenas noches,Grácias, Por favor, Nos vemos,
Tiempo	Dá, Hora, Semana, Minutos, Segundos
Días de la semana	Lunes, Martes, Miércoles, Jueves, Viernes, Sábado, Domingo
Meses	Enero, Febrero, Marzo, Abril, Mayo, Junio, Julio, Agosto, Septiembre, Octubre, Noviembre, Diciembre
Elementos de escuela	Calificación, Clase, Libreta, Escuela, Lápiz, Lectura, Lenguaje de seãs, Examen, Escritura , Sacapuntas, Regla, Colores
Familiares	Padre, Madre, Hermano, Hermana, Primo,Prima, Tío, Tía, Abuelo, Abuela, Esposa, Esposo, Familia, Hijo, Hija, Hombre, Mujer, Novia, Novio, Sobrina, Sobrino
Elementos de la casa	Cuarto, Baño, Cocina, cama, Casa, Closet, Cortinas, Cuna, Techo, Piso, Escaleras, Escoba, Lampara, Mesa, Pared, Sala, Vidrio, Ventana, Puerta
Adjetivos	Adulto, Joven, Ninõ, Bebe, Feo, Bonito, Malo, Sordo, Tonto, Fuerte, Gordo, Alto, Chico, Bien, Ciego, Débil, Delgado
Comidas	Cuchara, Cuchillo, Plato, Vaso, Lunch, Cena, Desayuno, Tenedor, Servilleta, Sal, Quiero más, No me gustó
Ropa	Aretes, Blusa, Botas, Playera, Collar, Guantes, Pantalón, Pijama, Shorts, Traje, Vestido, Zapatos, Falda, Ropa Interior
Partes del cuerpo	Barba, Bigote, Brazo, Boca, Cabello, Cabeza, Rostro, Pies, Dientes, Ojos, Cejas, Nariz, Orejas, Mejilla, Uña
Vehículos	Avión, Bote, Bicicleta, Camioneta, Helicóptero, Carro, Motocicleta, Taxi, Tractor, Trén, Metro, Combi, Van
Lugares de interés	Aeropuerto, Librería, Jardín, Cine, Circo, Edificio, Hospital, Hotel, Mercado, Museo, Restaurant, Escuela, Centro comercial, Cafetería
Pronombres	Yo, Tu, El, Ella, Ellos, Ellas, Nosotros, Nosotras, Ustedes, Ninguno, Alguno,
Verbos	Abrazar, Amar, Arreglar, Asustar, Ayudar, Buscar, Callar, Cerrar, Creer, Comer, Detener, Dormir, Golpear, Guardar, Jugar, Levantar, Llorar, Mentir, Oír, Olvidar, Hacer, Reír, Tirar, Acomodar, Limpiar
Oficios	Actor, Bombero, Doctor, Maestro, Camarero, Policía, Presidente, Secretaria, Carpintero, Mecánico, Zapatero, Estilista, Costurera
Estados del país	Aguascalientes, Baja California Nte. Baja California Sur, Campeche, Chihuahua, Chiapas, Coahuila, Colima, Durango, Guanajuato, Guerrero, Hidalgo, Jalisco,Ciudad de México, Estado de México, Michoacán, Morelos, Nayarit, Nuevo León, Oaxaca, Puebla, Querétaro, Quintana Roo, San Luis Potosí, Sinaloa, Sonora, Tabasco, Tamaulipas, Tlaxcala, Veracruz, Yucatán, Zacatecas

CUADRO 3.1: Lista de palabras

La diferencia entre las letras y las palabras en el LSM es muy grande, las letras son, con algunas excepciones, formas estáticas de una sola mano formando una letra, en algunas de las letras existe una pequeña oscilación de la mano al generar la seña, teniendo ahí un movimiento, aunque pequeño, primordial para su significado.

Para las palabras del LSM o ideogramas se necesitan más aspectos que solo la forma de la mano para obtener su significado. Una palabra del LSM está conformada por la forma de la mano, la posición de las manos con respecto al cuerpo, siendo la zona desde la cintura hacia la cara el área principal donde se generan los movimientos de una o ambas manos, siendo raras las señas que salen de esta área, también se tiene que tomar en cuenta los puntos de contacto de las manos con el cuerpo, si los movimientos se realizan hacia adelante o hacia atrás.

- La forma de la mano
- La dirección de una o ambas manos
- El punto de contacto de las manos con alguna parte del cuerpo o la cabeza

Por lo tanto, teniendo en cuenta la problemática encontrada en investigaciones similares, se evitó el uso de accesorios como joyería, relojes, pulseras, etc. También para evitar problemas de segmentación de las áreas de interés se utilizó ropa contrastante, para así evitar problemas de segmentación con ropa que pudiera ser tomada como parte de las áreas de interés. En esta ocación se tomaron secuencias de videos de las señas seleccionadas dentro de un lugar cerrado. Sin embargo, lo que no se tomó en cuenta fueron los cambios de iluminación, debido a la idea de que al ser un lugar cerrado no habría problemas de iluminación.

Ya que las palabras del LSM constan de movimientos de una o ambas manos, se tomaron videos cortos de cada una de las 249 palabras, el tiempo de la formación de las señas varía entre 1 y 2 segundos.

Para esta colección de videos se solicitó el apoyo solamente de estudiantes de la escuela CES, siendo en total 11 estudiantes mayores a 13 años, ya que alumnos de menor edad podrían variar mucho en el movimiento de las señas listadas, y por lo tanto, futuros problemas durante la experimentación.

Capítulo 4

Identificación de LSM mediante extractores de características

En este Capítulo se desarrolla la metodología de clasificación de señas mediante segmentación de imágenes por su color y obtención de características de las regiones de interés.

4.1. Metodología

La primera metodología propuesta se basa en el procesamiento de imágenes a color como se muestra en la Figura 4.1, para este procedimiento se usa el conjunto de datos solamente de palabras del LSM. Como se mencionó anteriormente este conjunto se compone de videos de secuencias de movimientos de la persona que denotan la palabra del LSM.

4.1.1. Adquisición de imágenes

Para cada seña el tiempo promedio del video es de solo 2 segundos. la posición inicial de la persona es con las manos a los lados, de la misma manera queda en esta pocisión la persona al hacer la seña. De la secuencia grabada se obtuvieron en promedio 15 imágenes las cuales representan toda la secuencia de movimientos 4.2, en formato RGB 640×480

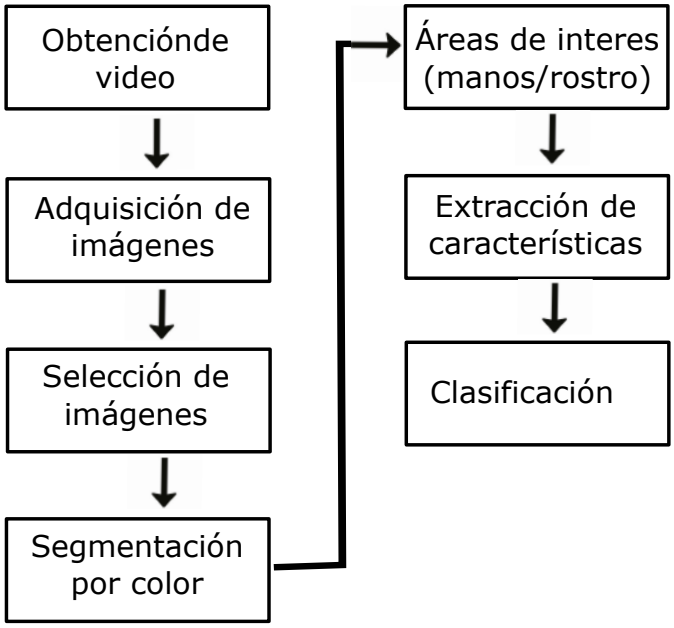

FIGURA 4.1: Metodología Propuesta

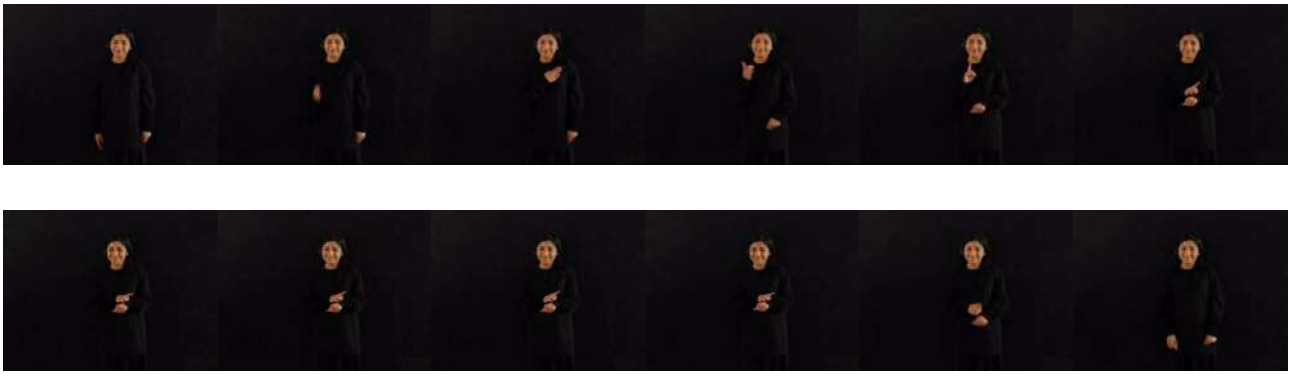

FIGURA 4.2: Secuencia completa de imágenes

Al tomar en cuenta la cantidad de imágenes se tiene un conjunto demasiado amplio que procesar, además el poder de cómputo necesario para poder procesar todo debería ser mayor para poder tener un proceso ágil, sin embargo al no tener la capacidad para tener un buen equipo para poder procesar la totalidad de imágenes de cada secuencia se optó por implementar un proceso de selección para disminuir la cantidad

de imágenes.

4.1.2. Discriminación de Imágenes

Tomando en cuenta que la secuencia de video siempre tendría las mismas características, las cuales son al inicio de la secuencia la persona se encuentra en reposo con los brazos a los costados, y al finalizar el movimiento la persona regresa a la postura de reposo con los brazos a los costados 4.3.

FIGURA 4.3: Posición de reposo

Entonces se da por echo que las imágenes iniciales y las finales siempre serán iguales para todos los casos. Por lo tanto, se pueden eliminar una cantidad de imágenes en el inicio de la secuencia de video, también se pueden eliminar imágenes de la parte final de la secuencia del video. Así la información mas representativa de la formación de la seña se encentra en la parte media de la secuencia del video.

Tomando el conjunto de imágenes I se selecciona la sección media m donde se da por hecho que es la parte mas significativa del video, entonces se realizan saltos en la

selección de imágenes i para obtener el conjunto final C de la siguiente manera:

$$i_1 = I_{n/2} \tag{4.1}$$
$$i_2 = I_{(n/2)-2} \tag{4.2}$$
$$i_3 = I_{(n/2)+2} \tag{4.3}$$
$$i_4 = I_{(n/2)+4} \tag{4.4}$$
$$C = i_1, i_2, i_3, i_4 \tag{4.5}$$
$$\tag{4.6}$$

El conjunto C siempre se conforma de 4 imágenes de la parte media de la secuencia del video. Al tener un promedio de 15 imágenes por seña, al realizar el proceso de seleción de C se descarta alrededor del 75 % de las imágenes totales obtenidas de las 249 palabras seleccionadas para la investigación.

La discriminación de ese porcentaje del conjunto de imágenes puede ser prejudicial, existe la posibilidad de que dentro de las imágenes descartadas exista información altamente discriminante durante la clasificación de las señas, pero con este método se pretenden dos cosas muy importantes:

1. Disminuir el conjunto de datos final

2. Disminuir el tiempo de procesamiento de las imágenes

La hipótesis hecha para poder descartar un número significativo de imágenes es que, solo se eliminaron imágenes similares y sin relevancia, lo que al final se traducirá en un conjunto de datos sin información repetida y sin características poco discriminantes para una buena clasificación.

En la Figura 4.4 se ejemplifica el proceso de selección de imágenes las cuales son las más representativas de la seña formada y las que aportan la información más discriminante para su clasificación.

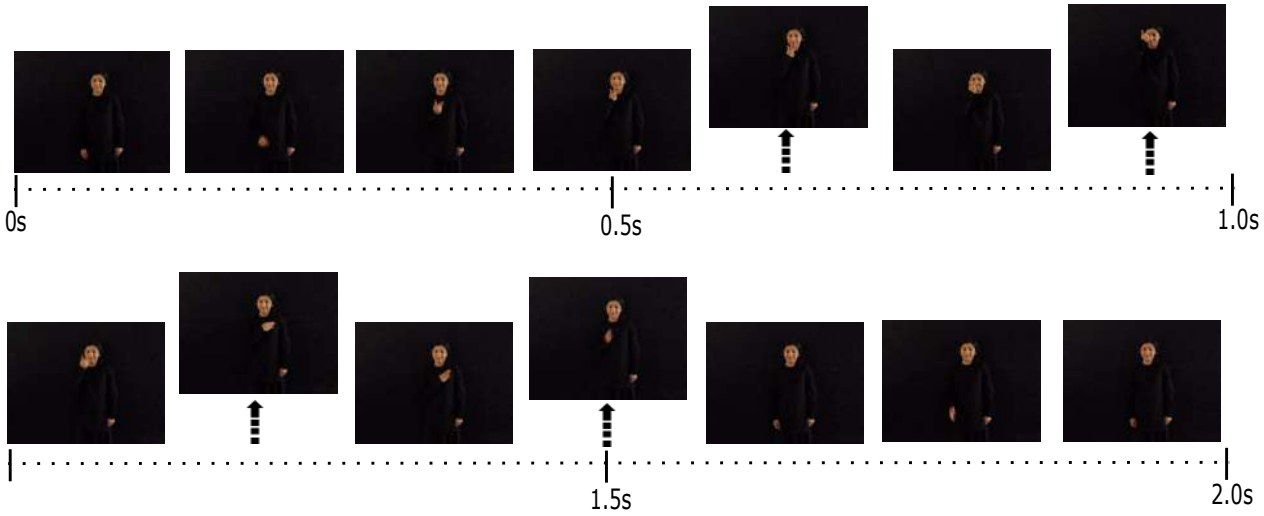

FIGURA 4.4: Selección de Imágenes

Segmentación por color

En el PDI (Procesamiento Digital de Imágenes) se utilizan en general dos tipos de imagen, imágenes a colores, e imágenes en escala de grises, siendo este último tipo el más utilizado, las imágenes en escala de grises se representa como una matriz cuadrada de $N \times M$ celdas, y los valores en una escala de 256 intensidades, desde el 0 que representa el color negro, hasta el 255 que representa el color blanco 4.5.

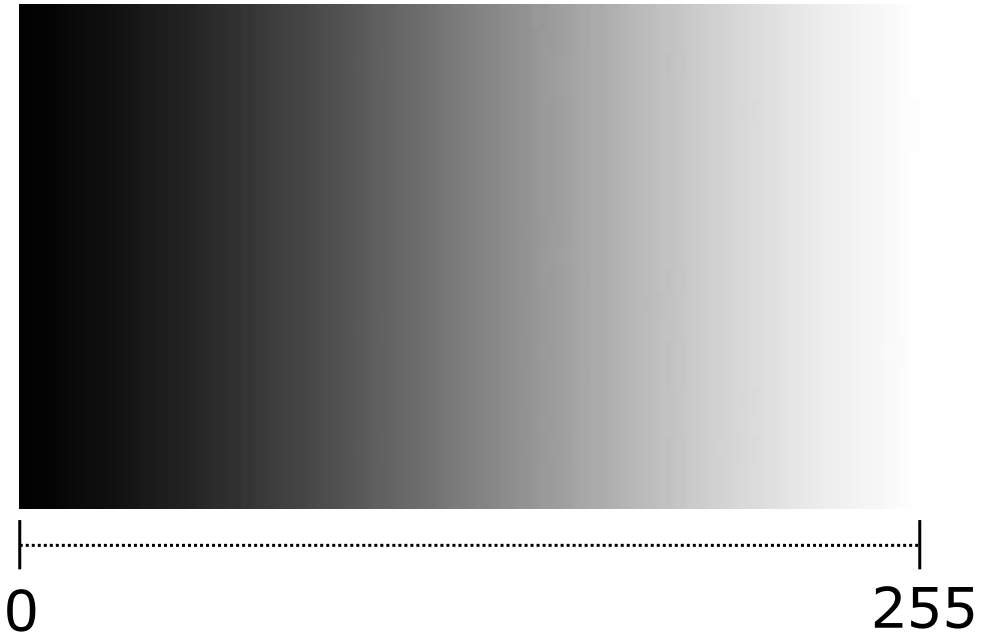

FIGURA 4.5: Escala de grises

Las imágenes a color en el campo RGB (Red, Green, Blue) se representan como matrices cúbicas donde se obtiene por separado la intensidad de rojo, verde y azul, la combinación de estos 3 colores primarios forman la gama completa de colores como la Figura4.7.

Otro espacio de colores es el HSV (Hue, Saturation, Value) **??**, es también usado en el PDI. El espacio HSV a comparación del espacio RGB es menos susceptible a la variación de la luminosidad de las imágenes. Otro aspecto del espacio HSV es que no se tiene que realizar un pre-procesamiento como la aplicación de un filtro Gaussiano o filtro promedio [73], a comparación del espacio RGB donde es muy común implementarlo para obtener buenos resultados de segmentación.

Así el conjunto de imágenes C se pasa al espacio HSV para seguir con el proceso de segmentación de las áreas de interés. El proceso de segmentación por color es el siguiente:

En la Figura **??** se puede observar la imagen original y un método ampliamente usado para la segmentación de imágenes, el método de Otsu generalmente obtiene muy buenos resultados para la segmentación de imágenes, sin embargo como se puede observar en la imagen original existe secciones con brillo excesivo y por lo tanto el método no tiene buenos resultados, al pasar la imagen a escala de grises, algunas secciones del

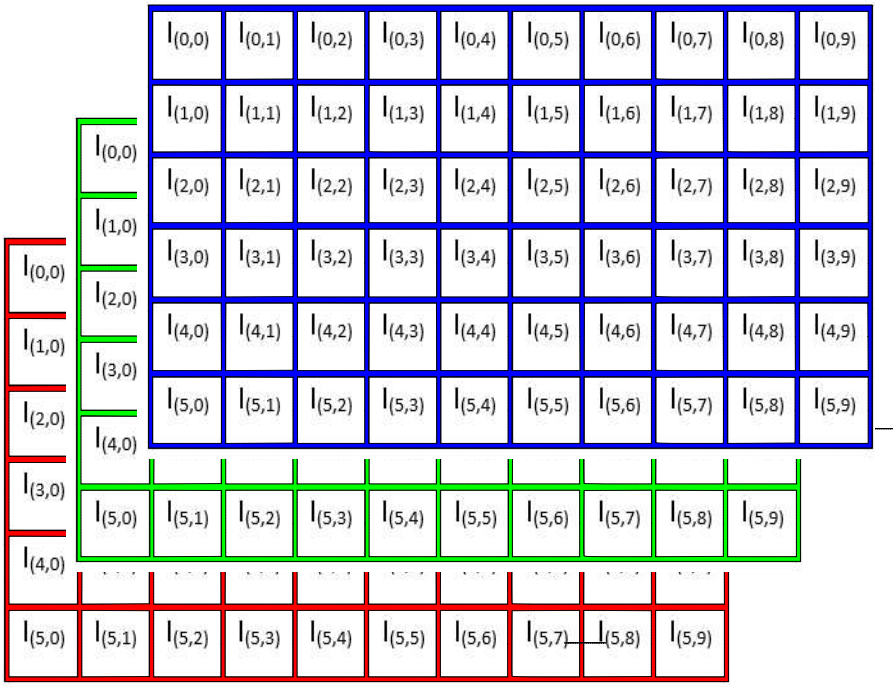

FIGURA 4.6: Representación de matrices de RGB

fondo de la imagen muestran una tonalidad similar a las regiones de interés, el resultado es que la imagen es prácticamente dividida en dos regiones en forma diagonal, resultados similares fueron obtenidos con otros métodos usados en el estado del arte para la segmentación de imágenes en escala de grises. Así el método propuesto se basa en el espacio de color, específicamente el espacio HSV, por sus propiedades, de las cuales para esta investigación la casi nula influencia de las variaciones de brillo en las imágenes, además en el espacio RGB el color es el resultado de la combinación de 3 pigmentos o colores primarios, en el espacio HSV el color es la región específica de la base del cono de la representación del espacio denotada por un ángulo α y el cambio de brillo u oscuridad se denota por la altura h del cono en el mismo ángulo α de la base, entonces si podemos encontrar el color específico denotado por α, la diferencia de brillo en las imágenes no tendrían influencia, por lo tanto, si encontramos el α podemos segmentar las regiones de interés descritas anteriormente y el fondo correcto de nuestra imagen.

Finalmente en la sección c de la Figura**??** se muestra la segmentación final de las áreas de interés, el algoritmo 1 describe el procedimiento implementado en esta metodología.

Una vez que se sabe las regiones de interés de nuestra imagen, estas regiones son

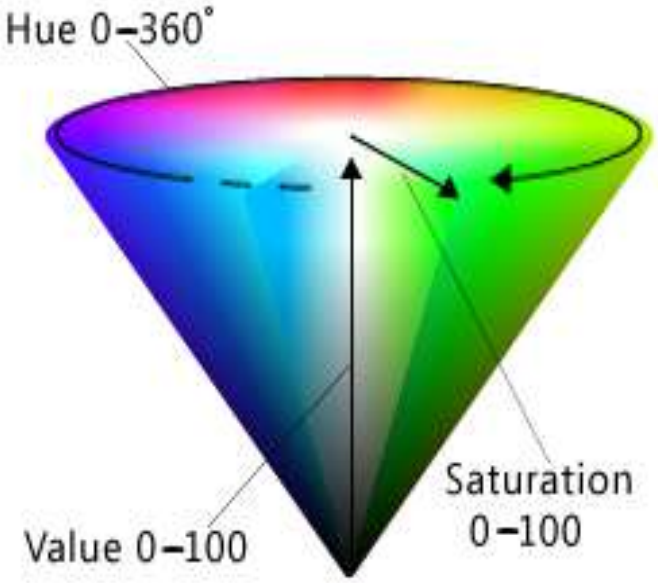

FIGURA 4.7: Representación de matrices de RGB (HTML Colors)

segmentadas usando el algoritmo mostrado arriba, se pueden posteriormente extraer las características de cada una de las tres regiones de interés, como se comentó anteriormente, algunas de las imágenes no tienen el total de regiones esperadas, para evitar problemas durante el proceso de extracción de características, se replican las características obtenidas hasta obtener el vector v de la dimensión esperada, en el Algoritmo 2 se muestra el procedimiento utilizado para obtener el vector v

Una vez que se obtienen las regiones de interés(manos y rostro) con la segmentación por color, la imagen fue binarizada y se obtuvieron las características geométricas de

FIGURA 4.8: a) Imagen original, b) Segmentación Otsu, c) Segmentación por color

Algorithm 1 Proceso de segmentación de color de la piel

Iniciar:
Se toma el matiz(H) del espacio (HSV) como un vector $\varphi[sin\theta, cos\theta]$
Se crea una red neuronal de 5×5 Matriz de neuronas

Se toman las imágenes HSV para entrenar la red en el siguiente proceso:
for *ángulo* $\theta \in \{0, 15, 30, ..., 360\}$ **do**
    end
    $\varphi_i = H\theta$
$Train\varphi_i \rightarrow$ NN

Con la red entrenada se realiza el siguiente proceso
for *p(x,y),* $\forall \{x = 1, ..., n, y = 1, ..., m\}$ ▷ *Para cada pixel en la imagen* **do**
    end
    $p_{x,y} \rightarrow NN = p_{x,y}(new)$ ▷ La neurona exitada representa el color que se asignara al pixel

$f[H, S, V] \rightarrow f[R, G, B]$ ▷ Imagen segmentada

Algorithm 2 Extracción de características

procedure CARACTERÍSTICAS $(r)_1, r_2, r_3 \in R$ ▷ Las manos y rostro
    for $f = 0 : n$ **do**
        end
        Para cada imagen en la secuencia
    for *r=1:3* **do**
        end
    a← *Humoments*(*r*)
    b← *FourierDescriptors*(*r*)
    c← *Ellipse*(*r*)
    d← *GuptaDescriptors*(*r*)
    e← *Flussermoments*(*r*)
return a+b+c+d+e ▷ Vector completo de características

cada área de interés, obteniendo un vector v con 3 conjuntos de características geométricas.

El conjunto de datos obtenido es de 57×2241, el clasificador propuesto es SVM y para comparar se utilizó Arboles de decisión, clasificador Bayesiano, Redes neuronales.

4.2. Resultados

Se considera para la comparación de resultados de cada clasificador la precisión general obtenida, el F-measure y el area bajo la curva(AUC). Para validar los resultados se utilizó validación cruzada k-fold cross validation con $k = 10$

La precisión general es el número de verdaderos positivos dividido entre la suma de los verdaderos positivos y los falsos positivos como se muestra a continuación:

$$precision = \frac{TP}{TP + FP} \tag{4.7}$$

El F-Measure es una medida de precisión obtenida de la siguiente manera:

$$F - Measure = 2 \cdot \frac{precision \cdot Recall}{precision + Recall} \tag{4.8}$$

donde el Recall se obtiene:

$$Recall = \frac{tn}{tn+fp}$$

El AUC(Area Under Curve) denota la capacidad del clasificador para distinguir y clasificar de manera correcta cada uno de los elementos del conjunto de datos.

Los resultados mostrados en la Tabla 4.1 muestran que los resultados obtenidos por el clasificador de Redes neuronales son los mas bajos en comparación con los demás clasificadores de comparación(Bayes y Arbol de decisión), se observa como la clase 230 es la que peor resultado tiene en mas de un clasificador. En la Tabla 4.2 se muestra una baja procesión de clasificación con SVM(clasificador propuesto) comparado con los otros métodos, la clase 198 tiene la menor precisión en SVM junto con Redes neuronales, también se observa que los clasificadores de comparación tienen mejor desempeño en las clases mostradas.

Finalmente en la Tabla 4.3 muestra el resultado general de cada uno de los clasificadores, su precisión, su sensitividad, su área bajo la curva(AUC) de la curva ROC.

Tabla comparativa												
Bayes			Árboles de decisión			SVM			Redes neuronales			Clase
Precisión	Measure	ROC	Precisión	Measure	ROC	Precisión	Measure	ROC	Precisión	Measure	ROC	
0.556	0.714	0.998	0.8	0.533	0.959	1	0.947	1	0.889	0.842	0.923	11
0.55	0.71	0.99	1	0.952	1	1	0.957	1	0.889	0.8	1	23
0.429	0.6	0.998	0.889	0.889	0.995	1	1	1	0.286	0.25	0.994	171
0.5	0.667	0.999	1	0.364	0.923	0.6	0.75	0.999	0.636	0.7	0.953	191
0.385	0.556	0.995	1	1	1	1	1	1	0.727	0.8	0.999	212
0.4	0.571	1	0.714	0.667	0.914	1	1	1	1	1	1	230

CUADRO 4.1: Tabla comparativa del bajo desempeño de los clasificadores de comparación

Tabla comparativa												
Bayes			Árboles de decisión			SVM			Redes neuronales			Clase
Precisión	Measure	ROC	Precisión	Measure	ROC	Precisión	Measure	ROC	Precisión	Measure	ROC	
1	1	1	0.909	0.952	1	0.625	0.769	0.999	0.529	0.667	0.999	126
1	1	1	0.9	0.947	1	0.667	0.333	0.998	0.5	0.471	0.998	127
0.5	0.667	0.999	1	0.364	0.923	0.6	0.75	0.999	0.636	0.7	0.953	191
1	1	1	0.833	0.667	0.986	0.6	0.76	0.999	0.571	0.5	0.999	195
1	1	1	0.857	0.75	0.98	0.643	0.783	0.999	0.583	0.667	0.998	197
1	0.667	0.996	0	0	0.979	0.5	0.429	0.999	0.5	0.5	0.997	198

CUADRO 4.2: Tabla comparativa del mal desempeño del clasificador propuesto y los clasificadores de comparación

El peor desempeño se observa en el clasificador arboles de decisión con 77.72 %, y el mejor desempeño se obtuvo con el clasificador propuesto de SVM con una precisión de 96.27 %.

En la Figura 4.9 se muestra visualmente el desempeño del clasificador SVM cuando la presición de las clases es menor a 100 %, como se puede observar las clases en la gráfica son relativamente pocas, ya que de las 249 clases existentes, solo 36 tienen una precisión baja, siendo la clase 198 la menor.

En la Figura 4.10 se muestra al igual que la figura anterior, las clases con desempeño bajo, para el clasificador bayesiano se pueden observar valores en 0, en las clases 138 y 211, y la clase 212 con la precisión de 38,5 %, se muestran 38 clases en esta gráfica con un desempeño poco uniforme.

La Figura 4.11 se muestran las clases con desempeño bajo obtenidas con Arboles

Resultados generales por clasificador			
Clasificador	Precisión	Sensitividad	AUC
Bayesiano	93,74 %	93,45 %	99,9 %
Arboles de decisión	77,22 %	77,72 %	95,17 %
SVM	96,27 %	95,9 %	99,9 %
Redes Neuronales	64,89 %	64,1 %	99,2 %

CUADRO 4.3: Resultados generales

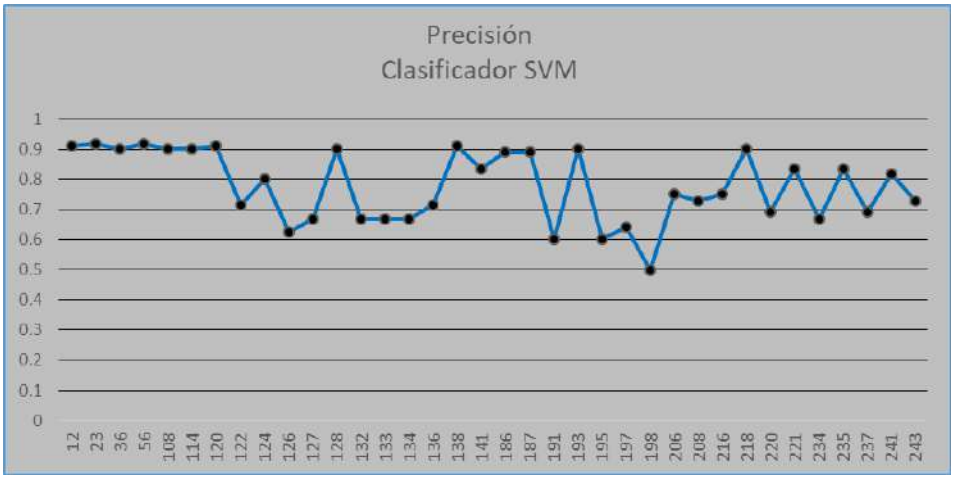

FIGURA 4.9: Baja precisión obtenida con SVM

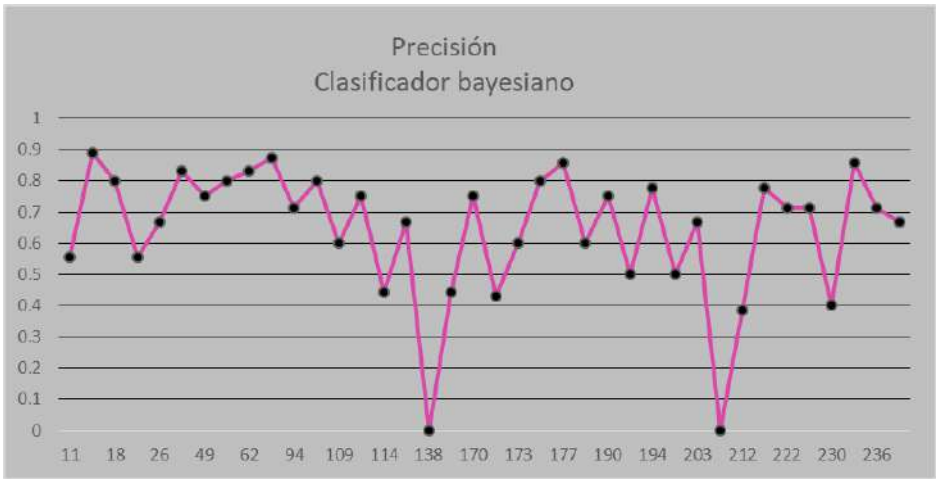

FIGURA 4.10: Baja precisión obtenida con Bayes

de decisión, a diferencia de las gráficas anteriores, esta gráfica cuenta con 91 clases con desempeño bajo, se observa que las clases 97, 198, 233, 235 y 236 tienen una precisión de 0, además de la clase 184 con una precisión de 13 % como las más bajas mostradas.

La Figura 4.12 muestra el resultado obtenido con Redes neuronales, de manera general es el clasificador con el mayor número de clases dentro con 221, sin embargo no cuenta con resultados de 0, se muestra la clase 181 como la que menor precisión obtuvo con 20 %

La Figura 4.13 muestra la gráfica comparativa de la precisión y área bajo la curva(AUC) de los cuatro clasificadores usados en la experimentación, en el se observa que el desempeño del clasificador propuesto fue superior al desempeño mostrado por los clasificadores con que se comparó, la precisión general es de 96,27 % y AUC de 1.

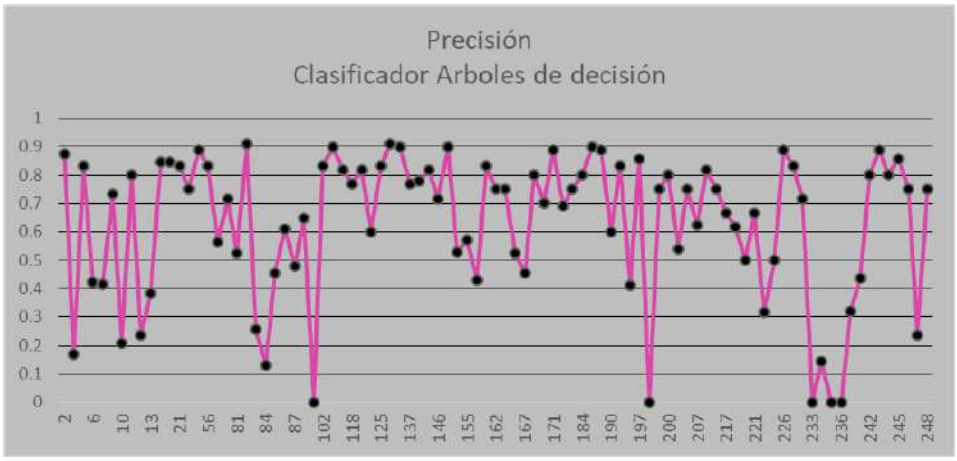

FIGURA 4.11: Baja precisión obtenida con Arbol de decisión

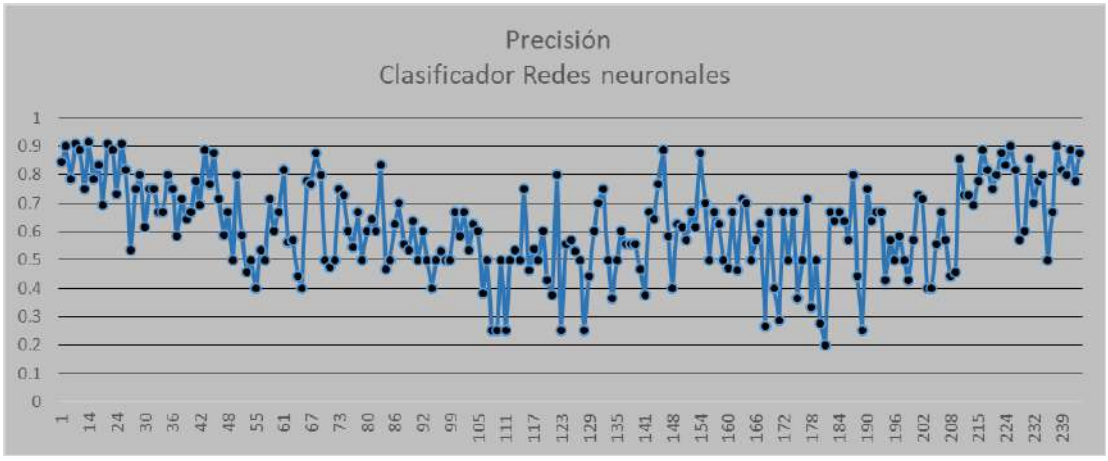

FIGURA 4.12: Baja precisión obtenida con Redes neuronales

Se muestra que el clasificador Bayesiano tiene buen desempeño con 93,74 % de precisión y valor de AUC de 0.999. Por el contrario, el clasificador con menor desempeño fueron Redes neuronales con 64,89 % y AUC de 0.992.

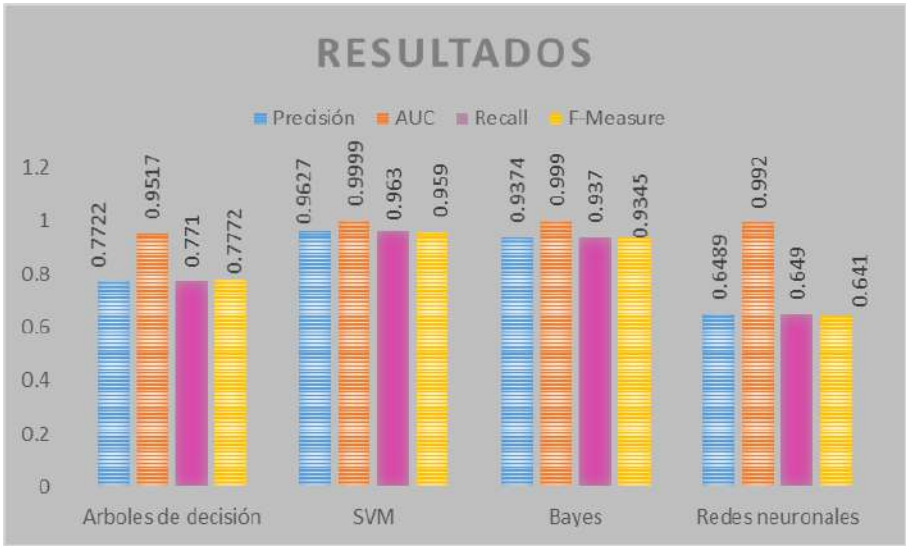

FIGURA 4.13: Resultados generales

Capítulo 5

Control de un robot mediante LSM

El control de dispositivos ha tenido una continua evolución a través del tiempo, usando palancas, cables, cuerdas, dispositivos infrarrojos, etc. La métodologia propone el control de un robot utilizando señas del lenguaje de seexicanas (LSM), para este fín se contemplan 12 seiferentes del alfabeto, su característica es la diferencia y facil distinción entre cada una de ellas con el resto y la sencillez de su formación como se muestra en la Figura5.2

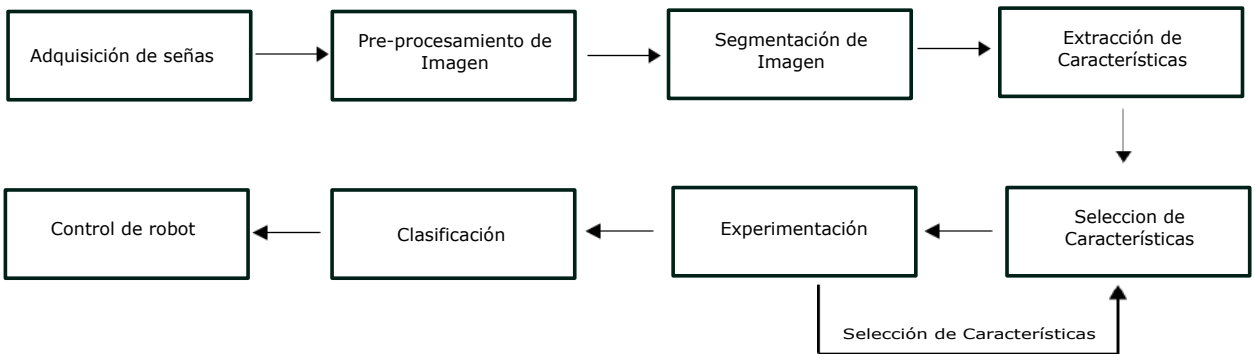

FIGURA 5.1: Metodología propuesta

5.0.1. Pre-procesamiento de imágenes

En el área de visión computacional se cuenta con diversos métodos para procesar la información más relevante de la imagen para obtener un resultado esperado, sin embargo, la mayoría de estos métodos son sensibles al ruido en la imagen, emborronamiento, variaciones en la luminosidad, entre otros. Para evitar los problemas generados por estas anomalías, se realiza un pre-procesamiento a las mismas, se aplican diversos

filtros, como filtro Gaussiano, mediana, etc. Tomando en cuenta lo anterior, al conjunto de datos se le aplica un filtro Gaussiano para eliminar ruido en la imagen y poder obtener buenos resultados en el proceso de segmentación.

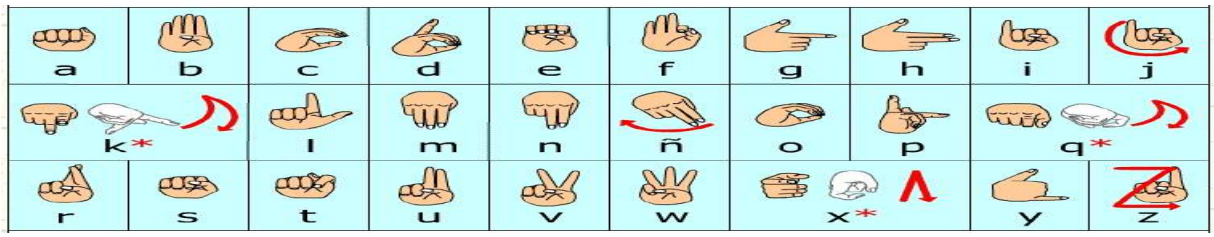

FIGURA 5.2: Abecedario del LSM (www.guiadisc.com)

5.0.2. Segmentación de la imagen

Las imágenes se dividen en dos secciones, el area o región de interés y el fondo o background, en la región de interés se encuentra las formas u objetos de interés para la investigación, para poder distinguir estas áreas de interés del resto de la imagen existen técnicas que segmentan la imagen obteniendo la región de interés. Para esta investigación se utilizó la segmentación por el método de Otsu, el cual es una técnica exhaustiva que tiene muy buenos resultados y es ampliamente utilizada en el estado del arte. Con el fondo de color gris de las imágenes la segmentación por el método de Otsu regresa el área de interés, que para esta investigación es la región de la mano de las personas cuando forman cada una de las seel alfabeto del LSM.

5.0.3. Extracción de características

En PDI se pueden dividir de manera general 3 grupos de características, las características geométricas, las cuales denotan la forma de la región de la imagen, las características cromaticas, las cuales se basan en la tonalidad, saturación y brillo de la región de interés y las características texturales, que denotan la forma que tiene la superficie a la vista, su rugosidad, suavidad, entre otras.

Las características seleccionadas para esta investigación son las características geométricas, para obtener información de la forma de cada una de las seeleccionadas, también se utilizan solo un tipo para minimizar el tiempo utilizado para la extracción de características.

Primero se obtienen las imágenes de los gestos manuales utilizando un fondo parcialmente controlado, es decir, solo se buscó un poco de contraste en el área de interés, sin controlar el resto del fondo ya que se trata de gestos estáticos.

Estos gestos/señas representan la posible orden específica de movimiento hecho al robot.

a	f	u
b	g	v
c	n	w
d	o	y

CUADRO 5.1: Señas del abecedario seleccionadas

5.1. Resultados

De los gestos manuales pertenecientes a las sílabas del Lenguaje de señas Mexicanas se obtuvieron imágenes representativas de 12 señas seleccionadas de las 29 silabas, cuya característica es la poca similitud, para poder eliminar en lo posible confusiones por parte de las personas y también problemas en la clasificación.

Se consideró la precisión general de cada clasificador propuesto para esta metodoloía, su F-measure, Recall y área bajo la curva(AUC). para poder validar los resultados se utilizó validación cruzada k-fold cross validation con $k = 10$.

La precisión general es el número de verdaderos positivos (TP) divididos por la suma de verdaderos positivos y falsos positivos (FP) de la siguiente manera:

$$precision = \frac{TP}{TP + FP} \tag{5.1}$$

Recall es obtenido por el número de verdaderos negativos (TN) divididos por la suma de verdaderos negativos (TN) y falsos negativos (FN) de la siguiente manera:

$$Recall = \frac{TN}{TN + FN} \tag{5.2}$$

El F-Measure es una medida de precisión obtenida a partir de:

$$F-Measure = 2 \cdot \frac{precision \cdot Recall}{precision + Recall} \tag{5.3}$$

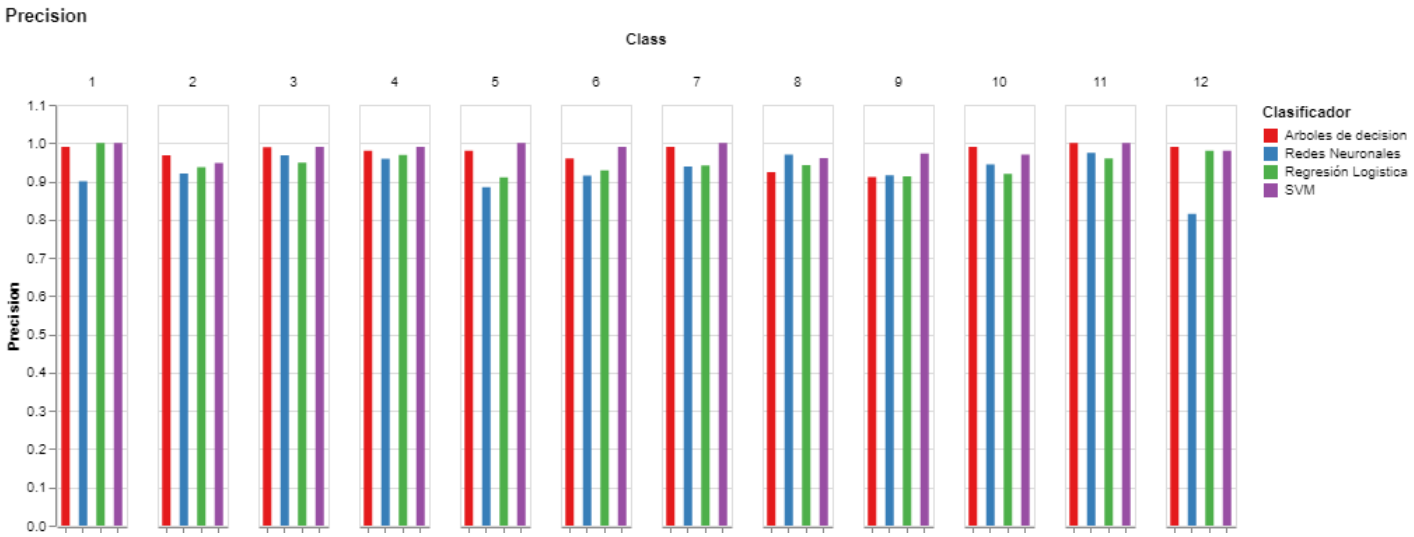

FIGURA 5.3: Precisir clase

El Área bajo la curva (AUC) denota la capacidad del clasificador para distinguir y clasificar correctamente los datos de entrada, esta métrica es una de las más importantes métricas de evalucación. El AUC se calcula de la siguiente manera:

$$AUC = \frac{\sum t_0 \in D^0 \sum t_1 \in D^1 1[f(t_0 < f(t_1))]}{|D^0| \cdot |D^1|} \tag{5.4}$$

donde:

$1[f(t_0) < f(t_1)]$ denota un indicador de funsión que regresa 1 si $f(t_0) < f(t_1)$ de lo contrario regresa 0.

D^0 es el conjunto de ejemplos negativos.

D^0 es el conjunto de ejemplos positivos.

El dataset resultante obtenido es de 45×1238 características correspondientes a las 12 señas seleccionadas del alfabeto del LSM. Se utilizaron 4 diferentes clasificadores para comparar los resultados obtenidos con el dataset.

Como se puede observar en la matriz de confusión de los clasificadores SVM Figura5.4, Redes Neuronales figura5.7, Árbol de decisión Figura5.5, Regresión Logística Figura5.6, todos tienen un comportamiento muy similar en la clasificación de los datos.

Las señas que mejor preisión de clasificación obtuvieron con el clasificador SVM son 1(99 %), 17(96.1 %), 6(94.9 %); 27(96.8 %), 26(95.9 %), 15(96.9 %) con Regresión Lineal; 26(98.7 %), 17(95 %), 6(94.8 %) con Redes Neuronales; 1(96 %), 26(94.2 %), 27(95.7 %) con Árboles de decisión.

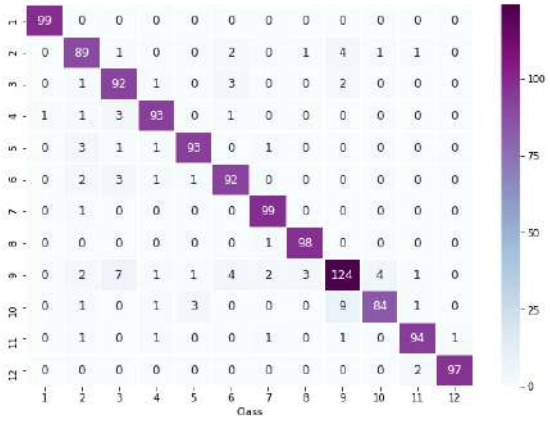

FIGURA 5.4: Matre confusi clasificador SVM

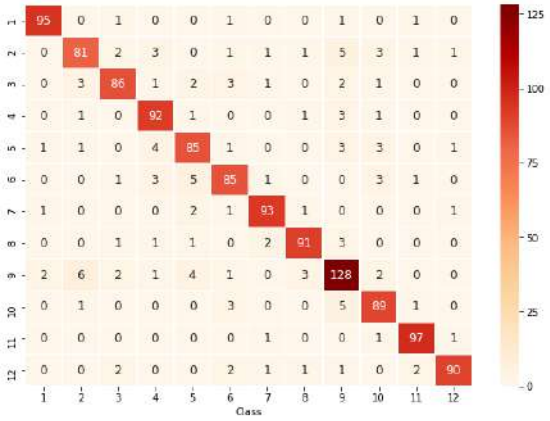

FIGURA 5.5: Matre confusi clasificador Arboles de decisi

La Figura muestra la comparación de la precisión general de todas las señas seleccionadas con cada uno de los clasificadores, como se comentó anteriormente, todos los clasificadores tienen un comportamiento similar, entonces es difícil seleccionar un clasificador que visualmente en la gráfica sobresalga de los demás, sin embargo SVM y

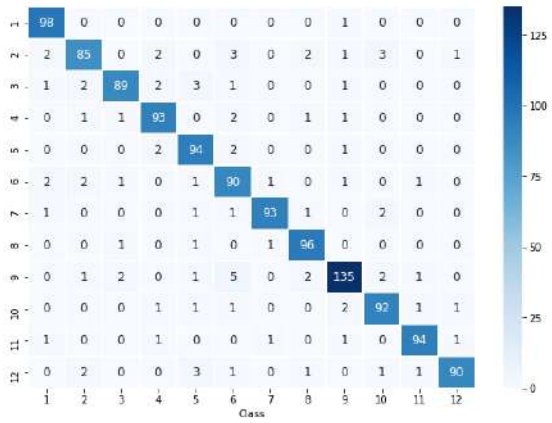

FIGURA 5.6: Matre confusi clasificador regresigica

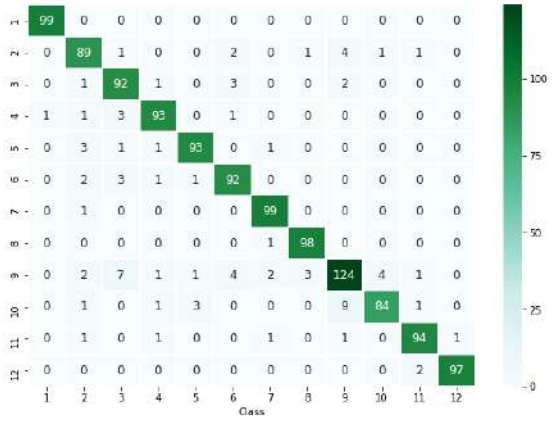

FIGURA 5.7: Matre confusi clasificador Redes neuronales

Redes Neuronales tienen mejor desempeño que el resto de los clasificadores por cada clase observando las gráficas obtenidas.

La Figura5.8 muestra una comparativa del desempeño general obtenido de los clasificadores, de manera general todos obtienen un buen desempeño, pero SVM tiene la mejor precisión general con 93,43 %, y del otro extremo Árboles de decisión con la preisión más baja de 90,06 %

Finalmente en la Figura5.9 muestra la comparación de los AUC de los cuatro clasificadores, en este gráfico se puede observar la variación de los desempeños obtenidos, el clasificador con mayor fluctuación se observa en el clasificador de Redes Neuronales, y el clasificador mas estable se puede observar en el gráfico perteneciente a SVM.

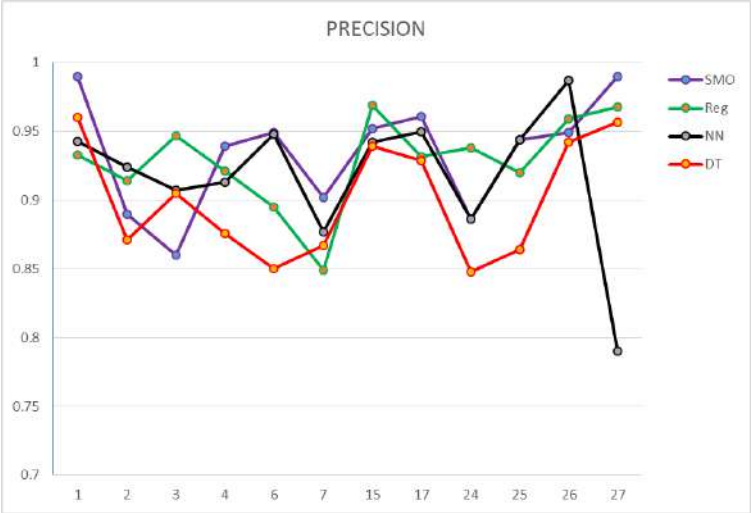

FIGURA 5.8: Precisir Clase

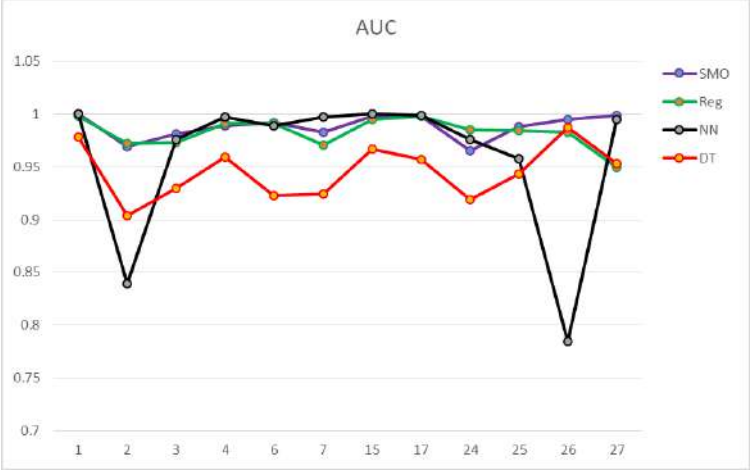

FIGURA 5.9: AUC

Capítulo 6

Conclusiones

En este capítulo se mustran inicialmente las conclusiones de cada una de las metodologías propuestas en la tesis, y posteriormente se darán las conclusiones generales obtenidas comparando cada una de las metodologías propuestas.

6.1. Metodología segmentación por color

En esta metodología se propuso un método de segmentación de las señas del Lenguaje de Señas Mexicanas basado en el color de la piel en el espacio HSV. Los resultados obtenidos muestran que las características geométricas seleccionadas beneficiaron en el resultado final, no solo porque la precisión es alta, pero también la variación de la iluminación en las imágenes durante la grabación de las señas no supuso un problema con el método propuesto, lo que si se suscitó con otras metodologías de segmentación usadas dentro del estado del arte. El espacio HSV se ve poco afectado por las variaciones de iluminación en las imágenes, aunque se controló el color del fondo de la imagen y el color de la vestimenta de las personas, el brillo no pudo ser controlado, aun así se obtuvo alta precisión de clasificación de las señas del LSM.

En la primera tabla de resultados de esta metodología se nota que la clase 181 tiene una precisión de clasificación menor de $20\,\%$ al usar redes neuronales como clasificador, siendo este un mal desempeño, posiblemente debido a la variación en el movimiento de las manos de las personas, algunos de estos movimientos siendo muy cercanos al rostro, y también siendo posible solamente el mal desempeño del clasificador, la clase 181 tiene una variación de movimientos en los brazos, algunas personas extienden sus brazos ampliamente y algunas los mantienen más pegados al cuerpo.

Aunque los resultados obtenidos son muy buenos, el tiempo necesario para segmentar cada una de las imágenes de la secuencia de video es alto. La alta cantidad de características requiere largo tiempo de procesamiento para ser obtenidas. El resultado

obtenido prueba que la metodología aplicada de selección de la secuencia de imágenes fue correcta debido a la precisión final obtenida, con esta metodología el tamaño del conjunto de datos se redujo casi a la mitad del tamaño si se hubiera considerado la secuencia completa de cada video clip.

6.2. Metodología Control de un robot mediante LSM

En esta metodología se propuso un método sencillo y generalizado para controlar un robot móvil por medio de señas manuales. Las señas seleccionadas para este propósito fueron las más discriminativas para evitar alguna similaridad y posible confusión. Las señas del alfabeto del LSM tienen la característica de que todas son estáticas y muy simples de realizar.

El método propuesto extrae las características geométricas de la forma de la mano. Algunas características pueden extraerse fácilmente de la imagen, de esta manera, el tiempo usado en este proceso se reduce en comparación que si se seleccionaran características más complejas. Los resultados obtenidos con los clasificadores utilizados en esta metodoloía propuesta muestran que en todos los casos las características seleccionadas fueron las mejores para la correcta clasificación de las señas, los resultados para cada clasificador está por encima del 90 %, obteniendo 98.35 % con SVM, 94.57 % con regresión lineal, 92.54 % con redes neuronales y 97.26 % con árboles de decisión.

Las señas pueden ser relacionadas a cualquier orden individual para controlar el robot, estas pueden ser seleccionadas a conveniencia del usuario. En el conjunto de datos, se incluyeron 12 diferentes señas del LSM, que pueden ser codificadas para ordenes individuales o combinación de estos, ordenes básicas como mover adelante, mover atrás, giro a la izquierda, giro a la derecha, acelerar, detenerse, o la combinación de estos.

El resultado de esta metodología es suficientemente bueno para intentar una ampliación más robusta del control del robot a partir de visión artificial, utilizando medios de comunicación como Bluetooth o WiFi, desde la computadora hacia el robot, y así controlarlo de manera natural usando gestos manuales, haciendo el control más sensitivo para los usuarios comunes. La simplicidad aquí puede contribuir a una manera sencilla de interactuar con los robots en un corto tiempo, de manera sencilla para cualquier persona.

Como trabajo futuro, se planea identificar el transmisor mas eficiente, y los parámetros óptimos para transmitir la orden al robot. Además, se contempla seleccionar individuos a los cuales se les explicara de manera rápida la metodología del control,

y se les pedirá probar el dispositivo, de esta manera obtener retroalimentación de su experiencia usando el control del robot con señas.

6.3. Conclusiones generales

Esta investigación propone metodologías para poder interpretar el lenguaje de señas Mexicanas (LSM), debido a la diferencia entre la dactilología (señas estáticas) que pertenecen principalmente al abecedario y a los ideogramas, que son movimientos manuales para formar palabras del LSM, crear un clasificador que pueda englobar todos los aspectos, tanto de ideogramas como la dactilología, supone poder superar las complicaciones inherentes tanto de la dactilología como de los ideogramas.

Los resultados obtenidos tanto para las palabras, como para las letras con los métodos propuestos muestran que las metodologías desarrolladas y las características seleccionadas son las más idóneas para la clasificación de las palabras y las letras del LSM. El proceso que se lleva a cabo tiene alto costo en tiempo de proceso, ya que las palabras al tener movimiento se tienen que procesar diversos frames de la secuencia, por lo tanto, el tiempo es más alto, a diferencia que si solo se procesaran imágenes del abecedario.

Como trabajo futuro se pretende incorporar ambas metodologías en un solo sistema lo suficientemente robusto para abarcar el procesamiento de ideogramas y dactilología del LSM.

Este trabajo también contribuye a una base de datos de 249 palabras del LSM, 29 letras del abecedario, y los números del LSM, en México existen trabajos sobre este tema, pero hasta ahora ninguno con una base de datos tan grande de palabras (ideogramas) del LSM. La finalidad de esta investigación siempre ha sido ayudar a las personas sordas a poder convivir de manera más sencilla con el resto de la población, con la evolución tecnológica que se vive todos los días, llegara el momento en que todas las personas puedan vivir una vida tan normal como la que vivimos la mayoría de nosotros, en que las discapacidades, no solo la discapacidad auditiva, dejen de ser una barrera para que podamos vivir una vida tranquila en igualdad de condiciones, y ya no exista la discriminación por la falta de conocimiento de las necesidades de las minorías.

Capítulo 7

Trabajo Futuro inmediato

En este capítulo especial se comenta el trabajo pendiente, el trabajo realizado y artículos.

Se cuenta con una participación en el artículo de conferencia WEA 2018“Automatic Calculation of body Mass Index Using Digital Image Processing”

Se cuenta con una participación en artículo de revista RISTI 2020 “Cálculo Automático de Masa Corporal Usando Visión Artificial” que es la continuación y ampliación del artículo de conferencia

El primer artículo: Mexican Sign Language Segmentation using Color Based Neuronal Networks to detect the individual skin color”. enviado a revista Expert Systems with Applications con factor de impacto de 5.7 fue aceptado y publicado en julio de 2021.

Se envio el segundo artículo con el título: Control de un robot movil mediante LSM y algoritmos de visión artificial. a la revista SENSORS.

Bibliografía

[1] Kalpattu S. Abhishek, Lee Chun Kai Qubeley y Derek Ho. «Glove-based hand gesture recognition sign language translator using capacitive touch sensor». En: *2016 IEEE International Conference on Electron Devices and Solid-State Circuits (EDSSC)*. IEEE, 2016. DOI: 10.1109/edssc.2016.7785276.

[2] Joao Gabriel Abreu y col. «Evaluating Sign Language Recognition Using the Myo Armband». En: *2016 XVIII Symposium on Virtual and Augmented Reality (SVR)*. IEEE, 2016. DOI: 10.1109/svr.2016.21.

[3] Washef Ahmed, Kunal Chanda y Soma Mitra. «Vision based Hand Gesture Recognition using Dynamic Time Warping for Indian Sign Language». En: *2016 International Conference on Information Science (ICIS)*. IEEE, 2016. DOI: 10.1109/infosci.2016.7845312.

[4] S. Aliyu y col. «Arabie sign language recognition using the Microsoft Kinect». En: *2016 13th International Multi-Conference on Systems, Signals & Devices (SSD)*. IEEE, 2016. DOI: 10.1109/ssd.2016.7473753.

[5] Aqsa Ali y col. «Hand Gesture Interpretation Using Sensing Glove Integrated With Machine Learning Algorithms». en. En: (2016). DOI: 10.5281/ZENODO.1127466.

[6] J. S. Artal-Sevil y J. L. Montanes. «Development of a robotic arm and implementation of a control strategy for gesture recognition through Leap Motion device». En: *2016 Technologies Applied to Electronics Teaching (TAEE)*. IEEE, 2016. DOI: 10.1109/taee.2016.7528373.

[7] Anand Asokan, Allan Joseph Pothen y Raj Krishnan Vijayaraj. «ARMatron — A wearable gesture recognition glove: For control of robotic devices in disaster management and human Rehabilitation». En: *2016 International Conference on Robotics and Automation for Humanitarian Applications (RAHA)*. IEEE, 2016. DOI: 10.1109/raha.2016.7931882.

[8] Rukshan Batuwita y Vasile Palade. «FSVM-CIL: Fuzzy Support Vector Machines for Class Imbalance Learning». En: *IEEE Transactions on Fuzzy Systems* 18.3 (2010), págs. 558-571. DOI: 10.1109/tfuzz.2010.2042721.

[9] David S. Bayard y col. «Vision-Based Navigation for the NASA Mars Helicopter». En: *AIAA Scitech 2019 Forum*. American Institute of Aeronautics y Astronautics, 2019. DOI: 10.2514/6.2019-1411.

[10] Asa Ben-Hur y Jason Weston. «A User's Guide to Support Vector Machines». En: *Methods in Molecular Biology*. Humana Press, 2009, págs. 223-239. DOI: 10.1007/978-1-60327-241-4_13.

[11] Florian A. Bertsch y Verena V. Hafner. «Real-time dynamic visual gesture recognition in human-robot interaction». En: *2009 9th IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2009. DOI: 10.1109/ichr.2009.5379541.

[12] Siddhartha Bhattacharyya y Ujjwal Maulik. *Soft Computing for Image and Multimedia Data Processing*. Springer Berlin Heidelberg, 2013. DOI: 10.1007/978-3-642-40255-5.

[13] Feifei Bian, Ruifeng Li y Peidong Liang. «SVM based simultaneous hand movements classification using sEMG signals». En: *2017 IEEE International Conference on Mechatronics and Automation (ICMA)*. IEEE, 2017. DOI: 10.1109/icma.2017.8015855.

[14] Necati Cihan Camgoz y col. «SubUNets: End-to-End Hand Shape and Continuous Sign Language Recognition». En: *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2017. DOI: 10.1109/iccv.2017.332.

[15] Teak-Wei Chong y Boon-Giin Lee. «American Sign Language Recognition Using Leap Motion Controller with Machine Learning Approach». En: *Sensors* 18.10 (2018), pág. 3554. DOI: 10.3390/s18103554.

[16] *Continuous Gesture Recognition (ICPR '16)*. URL: http://chalearnlap.cvc.uab.es/dataset/22/description/.

[17] Runpeng Cui, Hu Liu y Changshui Zhang. «Recurrent Convolutional Neural Networks for Continuous Sign Language Recognition by Staged Optimization». En: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017. DOI: 10.1109/cvpr.2017.175.

[18] *Data-Set NGT*. URL: https://www.ru.nl/corpusngtuk/.

[19] *Data-Set Repository Sing Languaje Corpora*. URL: http://www.plm.uw.edu.pl/en/node/327.

[20] *Database for Signer-Independent Continuous Sign Language Recognition*. URL: https://www.phonetik.uni-muenchen.de/forschung/Bas/SIGNUM/.

[21] Janez Demšar. *Statistical Comparisons of Classifiersover Multiple Data Sets*. Journal of Machine Learning Research 7 (2006) 1–30. 2006.

[22] V.S. Devi y M.N. Murty. *Pattern Recognition: An Introduction*. Universities Press, 2011. ISBN: 9788173717253. URL: https://books.google.com.mx/books?id=pvZDMwEACAAJ.

[23] Pei Di y col. «Fall Detection and Prevention Control Using Walking-Aid Cane Robot». En: *IEEE/ASME Transactions on Mechatronics* 21.2 (2016), págs. 625-637. DOI: 10.1109/tmech.2015.2477996.

[24] Abhishek Dudhal y col. «Hybrid SIFT Feature Extraction Approach for Indian Sign Language Recognition System Based on CNN». En: *Proceedings of the International Conference on ISMAC in Computational Vision and Bio-Engineering 2018 (ISMAC-CVB)*. Springer International Publishing, 2019, págs. 727-738. DOI: 10.1007/978-3-030-00665-5_72.

[25] A. Samir Elons, Magdy Abull-ela y M.F. Tolba. «A proposed PCNN features quality optimization technique for pose-invariant 3D Arabic sign language recognition». En: *Applied Soft Computing* 13.4 (2013), págs. 1646-1660. DOI: 10.1016/j.asoc.2012.11.036.

[26] S. García. F. Herrera. *An Extension on "Statistical Comparisons of Classifiers over Multiple DataSets "for all Pairwise Comparisons*. JournalofMachineLearningResearch9. 2008.

[27] Giuseppe Airò Farulla y col. «Vision-Based Pose Estimation for Robot-Mediated Hand Telerehabilitation». En: *Sensors* 16.2 (2016), pág. 208. DOI: 10.3390/s16020208.

[28] Jan Flusser y Tomás Suk. «Pattern recognition by affine moment invariants». En: *Pattern Recognition* 26.1 (1993), págs. 167-174. DOI: 10.1016/0031-3203(93)90098-h.

[29] Jakub Galka y col. «Inertial Motion Sensing Glove for Sign Language Gesture Acquisition and Recognition». En: *IEEE Sensors Journal* 16.16 (2016), págs. 6310-6316. DOI: 10.1109/jsen.2016.2583542.

[30] Julián García y col. «Prototyping and evaluating glove-based multimodal interfaces». En: *Journal on Multimodal User Interfaces* 2.1 (2008), págs. 43-52. DOI: 10.1007/s12193-008-0005-1.

[31] S. García y col. «A study of statistical techniques and performance measures for genetics-based machine learning: accuracy and interpretability». En: *Soft Computing* 13.10 (2008), págs. 959-977. DOI: 10.1007/s00500-008-0392-y.

[32] Luis Garrote y col. «3D point cloud downsampling for 2D indoor scene modelling in mobile robotics». En: *2017 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*. IEEE, 2017. DOI: 10.1109/icarsc.2017.7964080.

[33] Archana Ghotkar, Pujashree Vidap y Kshitish Deo. «Dynamic Hand Gesture Recognition using Hidden Markov Model by Microsoft Kinect Sensor». En: *International Journal of Computer Applications* 150.5 (2016), págs. 5-9. DOI: 10.5120/ijca2016911498.

[34] Shuai Guo, Qizhuo Diao y Fengfeng Xi. «Vision Based Navigation for Omnidirectional Mobile Industrial Robot». En: *Procedia Computer Science* 105 (2017), págs. 20-26. DOI: 10.1016/j.procs.2017.01.182.

[35] L. Gupta y M.D. Srinath. «Contour sequence moments for the classification of closed planar shapes». En: *Pattern Recognition* 20.3 (1987), págs. 267-272. DOI: 10.1016/0031-3203(87)90001-x.

[36] Ayman Hamed, Nahla A. Belal y Khaled M. Mahar. «Arabic Sign Language Alphabet Recognition Based on HOG-PCA Using Microsoft Kinect in Complex Backgrounds». En: *2016 IEEE 6th International Conference on Advanced Computing (IACC)*. IEEE, 2016. DOI: 10.1109/iacc.2016.90.

[37] Jian Huang y col. «Posture estimation and human support using wearable sensors and walking-aid robot». En: *Robotics and Autonomous Systems* 73 (2015), págs. 24-43. DOI: 10.1016/j.robot.2014.11.013.

[38] *Instituto de enseñanza sordomudos mexico*. URL: http://masmyt.org.mx/Instituciones/OSC/Instituto%20Pedag%C3%B3gico%20para%20Problemas%20del%20Lenguaje,%20IAP.

[39] Md. Mohiminul Islam, Sarah Siddiqua y Jawata Afnan. «Real time Hand Gesture Recognition using different algorithms based on American Sign Language». En: *2017 IEEE International Conference on Imaging, Vision & Pattern Recognition (icIVPR)*. IEEE, 2017. DOI: 10.1109/icivpr.2017.7890854.

[40] Anja Jackowski, Marion Gebhard y Axel Graser. «A novel head gesture based interface for hands-free control of a robot». En: *2016 IEEE International Symposium on Medical Measurements and Applications (MeMeA)*. IEEE, 2016. DOI: 10.1109/memea.2016.7533744.

[41] Li-Hu Jhang, Carlo Santiago y Chian-Song Chiu. «Multi-sensor based glove control of an industrial mobile robot arm». En: *2017 International Automatic Control Conference (CACS)*. IEEE, 2017. DOI: 10.1109/cacs.2017.8284267.

[42] Feng Jiang y col. «Spatial and temporal pyramid-based real-time gesture recognition». En: *Journal of Real-Time Image Processing* 13.3 (2016), págs. 599-611. DOI: 10.1007/s11554-016-0620-0.

[43] Shuo Jiang y col. «Feasibility of Wrist-Worn, Real-Time Hand, and Surface Gesture Recognition via sEMG and IMU Sensing». En: *IEEE Transactions on Industrial Informatics* 14.8 (2018), págs. 3376-3385. DOI: 10.1109/tii.2017.2779814.

[44] Hancheol Park Jung-Ho Kim Najoung Kim y Jong C. Park. *Enhanced Sign Language Transcription System via Hand Tracking and Pose Estimation*. Journal of Computing Science and Engineering, Vol. 10, No. 3. 2016.

[45] Ramesh M. Kagalkar y S.V Gumaste. *Gradient Based Key Frame Extraction for Continuous Indian Sign Language Gesture Recognition and Sentence Formation in Kannada Language: A Comparative Study of Classifiers*. IJCSE Vol. 4, Issue-9. 2016.

[46] P. V. V. Kishore y col. «Motionlets Matching With Adaptive Kernels for 3-D Indian Sign Language Recognition». En: *IEEE Sensors Journal* 18.8 (2018), págs. 3327-3337. DOI: 10.1109/jsen.2018.2810449.

[47] P.V.V. Kishore y col. «Optical Flow Hand Tracking and Active Contour Hand Shape Features for Continuous Sign Language Recognition with Artificial Neural Networks». En: *2016 IEEE 6th International Conference on Advanced Computing (IACC)*. IEEE, 2016. DOI: 10.1109/iacc.2016.71.

[48] Oscar Koller, Jens Forster y Hermann Ney. «Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers». En: *Computer Vision and Image Understanding* 141 (2015), págs. 108-125. DOI: 10.1016/j.cviu.2015.09.013.

[49] Oscar Koller y col. «Deep Sign: Enabling Robust Statistical Continuous Sign Language Recognition via Hybrid CNN-HMMs». En: *International Journal of Computer Vision* 126.12 (2018), págs. 1311-1325. DOI: 10.1007/s11263-018-1121-3.

[50] E. Kiran Kumar y col. «3D Motion Capture for Indian Sign Language Recognition (SLR)». En: *Smart Computing and Informatics*. Springer Singapore, 2017, págs. 21-29. DOI: 10.1007/978-981-10-5547-8_3.

[51] E. Kiran Kumar y col. «Training CNNs for 3-D Sign Language Recognition With Color Texture Coded Joint Angular Displacement Maps». En: *IEEE Signal Processing Letters* 25.5 (2018), págs. 645-649. DOI: 10.1109/lsp.2018.2817179.

[52] Pradeep Kumar y col. «A multimodal framework for sensor based sign language recognition». En: *Neurocomputing* 259 (2017), págs. 21-38. DOI: 10.1016/j.neucom.2016.08.132.

[53] Manoj Kurien y col. «Real-time simulation of construction workers using combined human body and hand tracking for robotic construction worker system». En: *Automation in Construction* 86 (2018), págs. 125-137. DOI: 10.1016/j.autcon.2017.11.005.

[54] Kengo Kuroki y col. «A remote conversation support system for deaf-mute persons based on bimanual gestures recognition using finger-worn devices». En: *2015 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*. IEEE, 2015. DOI: 10.1109/percomw.2015.7134101.

[55] Shao-Zi Li y col. «Feature learning based on SAE–PCA network for human gesture recognition in RGBD images». En: *Neurocomputing* 151 (2015), págs. 565-573. DOI: 10.1016/j.neucom.2014.06.086.

[56] Kian Ming Lim, Alan W.C. Tan y Shing Chiang Tan. «A feature covariance matrix with serial particle filter for isolated sign language recognition». En: *Expert Systems with Applications* 54 (2016), págs. 208-218. DOI: 10.1016/j.eswa.2016.01.047.

[57] Tao Liu, Wengang Zhou y Houqiang Li. «Sign language recognition with long short-term memory». En: *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2016. DOI: 10.1109/icip.2016.7532884.

[58] Ana Milena Lopez Lopez y Jorge Enrique Ardila Uribe. «Visual servo control law design using 2D vision approach, for a 3 DOF robotic system built with LEGO EV3 and a Raspberry Pi». En: *2016 XXI Symposium on Signal Processing, Images and Artificial Vision (STSIVA)*. IEEE, 2016. DOI: 10.1109/stsiva.2016.7743360.

[59] Rajesh B. Mapari y Govind Kharat. «Real time human pose recognition using leap motion sensor». En: *2015 IEEE International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN)*. IEEE, 2015. DOI: 10.1109/icrcicn.2015.7434258.

[60] Alberto Aguado Mark Nixon. *Feature Extraction & Image Processing*. Elsevier Second Edition. 2008.

[61] Marc Martínez-Camarena y col. «Reasoning about Body-Parts Relations for Sign Language Recognition». En: (21 de jul. de 2016). arXiv: http://arxiv.org/abs/1607.06356v1 [cs.CV].

[62] Tom M. Mitchell. *Machine Learning*. McGraw-Hill Education, 1997. ISBN: 9780070428072. URL: https://www.amazon.com/Machine-Learning-Tom-M-Mitchell/dp/0070428077?SubscriptionId=AKIAIOBINVZYXZQZ2U3A&tag=chimbori05-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=0070428077.

[63] Tri Listyorini. Mohammad Iqbal Endang Supriyati. *SIBI Blue: Developing Indonesian Sign Language Recognition System Based On The Mobile*. International Journal of Information Technology, Computer Science and Open Source Vol.1, No.1. 2017.

[64] M. Mohandes, S. Aliyu y M. Deriche. «Prototype Arabic Sign language recognition using multi-sensor data fusion of two leap motion controllers». En: *2015 IEEE 12th International Multi-Conference on Systems, Signals & Devices (SSD15)*. IEEE, 2015. DOI: 10.1109/ssd.2015.7348113.

[65] Deepali Naglot y Milind Kulkarni. «Real time sign language recognition using the leap motion controller». En: *2016 International Conference on Inventive Computation Technologies (ICICT)*. IEEE, 2016. DOI: 10.1109/inventive.2016.7830097.

[66] Susmit Nanda y col. «Real-time surface material identification using infrared sensor to control speed of an arduino based car like mobile robot». En: *Proceedings of the 2015 Third International Conference on Computer, Communication, Control and Information Technology (C3IT)*. IEEE, 2015. DOI: 10.1109/c3it.2015.7060171.

[67] Marlon Oliveira y col. «Irish Sign Language Recognition Using Principal Component Analysis and Convolutional Neural Networks». En: *2017 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*. IEEE, 2017. DOI: 10.1109/dicta.2017.8227451.

[68] Marco Paluszny, Hartmut Prautzsch y Wolfgang Boehm. «Métodos de Bézier y B-splines». En: (ene. de 2005). DOI: 10.5445/KSP/1000002481.

[69] Zuzanna Parcheta y Carlos-D. Martínez-Hinarejos. «Sign Language Gesture Recognition Using HMM». En: *Pattern Recognition and Image Analysis*. Springer International Publishing, 2017, págs. 419-426. DOI: 10.1007/978-3-319-58838-4_46.

[70] Y. Pititeeraphab y col. «Robot-arm control system using LEAP motion controller». En: *2016 International Conference on Biomedical Engineering (BME-HUST)*. IEEE, 2016. DOI: 10.1109/bme-hust.2016.7782091.

[71] Luis Quesada, Gustavo López y Luis Guerrero. «Improving Deaf People Accessibility and Communication Through Automatic Sign Language Recognition Using Novel Technologies». En: *Advances in Intelligent Systems and Computing*. Springer International Publishing, 2016, págs. 497-507. DOI: 10.1007/978-3-319-41962-6_44.

[72] A. Rama Mohan Reddy R. Madana Mohana. *Machine Learning and Data Mining Techniques for Sign Language Recognition and Retrieval System*. IJECRT- International Journal of Engineering Computational Research and Technology. 2016.

[73] J. L. Raheja, A. Mishra y A. Chaudhary. «Indian sign language recognition using SVM». En: *Pattern Recognition and Image Analysis* 26.2 (2016), págs. 434-441. DOI: 10.1134/s1054661816020164.

[74] G. Ananth Rao y P.V.V. Kishore. «Selfie video based continuous Indian sign language recognition system». En: *Ain Shams Engineering Journal* 9.4 (2018), págs. 1929-1939. DOI: 10.1016/j.asej.2016.10.013.

[75] P. Revathi. *Sign Language Recognition Using Principal Component Analysis*. International Journal of Advance Research in Computer Science and Management Studies Volume 4, Issue 6. 2016.

[76] Mliki H. Beji R.E. Hammami M. Romdhane N.B. *Combined 2d/3d traffic signs recognition and distance estimation*. IEEE Intelligent Vehicles Symposium (IV), pp. 355–360. IEEE. 2016.

[77] *RWTH-PHOENIX-Weather*. URL: https://www-i6.informatik.rwth-aachen.de/~forster/database-rwth-phoenix.php.

[78] *RWTH-PHOENIX-Weather 2014: Continuous Sign Language Recognition Dataset*. URL: https://www-i6.informatik.rwth-aachen.de/~koller/RWTH-PHOENIX/.

[79] Shinji Sako, Mika Hatano y Tadashi Kitamura. «Real-Time Japanese Sign Language Recognition Based on Three Phonological Elements of Sign». En: *HCI International 2016 – Posters' Extended Abstracts*. Springer International Publishing, 2016, págs. 130-136. DOI: 10.1007/978-3-319-40542-1_21.

[80] David Salomon. *Curves and Surfaces for Computer Graphics*. Springer New York, 8 de sep. de 2005. 480 págs. ISBN: 0387241965. URL: https://www.ebook.de/de/product/3195493/david_salomon_curves_and_surfaces_for_computer_graphics.html.

[81] Haifeng Sang y Hongjiao Wu. «A Sign Language Recognition System in Complex Background». En: *Biometric Recognition*. Springer International Publishing, 2016, págs. 453-461. DOI: 10.1007/978-3-319-46654-5_50.

[82] Alexey A. Shvets y col. «Automatic Instrument Segmentation in Robot-Assisted Surgery using Deep Learning». En: *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2018. DOI: 10.1109/icmla.2018.00100.

[83] J-Francisco Solís-V. y col. «Mexican sign language recognition using normalized moments and artificial neural networks». En: *Optics and Photonics for Information Processing VIII*. Ed. por Abdul A. S. Awwal y col. SPIE, 2014. DOI: 10.1117/12.2061077.

[84] Ivo Stančić, Josip Musić y Tamara Grujić. «Gesture recognition system for real-time mobile robot control based on inertial sensors and motion strings». En: *Engineering Applications of Artificial Intelligence* 66 (2017), págs. 33-48. DOI: 10.1016/j.engappai.2017.08.013.

[85] Chao Sun y col. «Latent support vector machine for sign language recognition with Kinect». En: *2013 IEEE International Conference on Image Processing*. IEEE, 2013. DOI: 10.1109/icip.2013.6738863.

[86] Naveen Rawat Tamanna. *A Review: Hand Recognition System and Its Procedure*. Imperial Journal of Interdisciplinary Research (IJIR)Vol-2, Issue-8, 2016.

[87] Min Tan y col. «Weakly Supervised Metric Learning for Traffic Sign Recognition in a LIDAR-Equipped Vehicle». En: *IEEE Transactions on Intelligent Transportation Systems* 17.5 (2016), págs. 1415-1427. DOI: 10.1109/tits.2015.2506182.

[88] Wenjin Tao, Ming C. Leu y Zhaozheng Yin. «American Sign Language alphabet recognition using Convolutional Neural Networks with multiview augmentation and inference fusion». En: *Engineering Applications of Artificial Intelligence* 76 (2018), págs. 202-213. DOI: 10.1016/j.engappai.2018.09.006.

[89] Asha Thalange y S.K. Dixit. «COHST and Wavelet Features Based Static ASL Numbers Recognition». En: *Procedia Computer Science* 92 (2016), págs. 455-460. DOI: 10.1016/j.procs.2016.07.367.

[90] Alaa Tharwat y col. «SIFT-Based Arabic Sign Language Recognition System». En: *Advances in Intelligent Systems and Computing*. Springer International Publishing, 2015, págs. 359-370. DOI: 10.1007/978-3-319-13572-4_30.

[91] Noor Tubaiz, Tamer Shanableh y Khaled Assaleh. «Glove-Based Continuous Arabic Sign Language Recognition in User-Dependent Mode». En: *IEEE Transactions on Human-Machine Systems* 45.4 (2015), págs. 526-533. DOI: 10.1109/thms.2015.2406692.

[92] Zhengchao Zhang y col. «Recognition of Chinese Sign Language Based on Dynamic Features Extracted by Fast Fourier Transform». En: *Lecture Notes in Computer Science*. Springer International Publishing, 2016, págs. 508-517. DOI: 10.1007/978-3-319-48896-7_50.