

Máster Universitario en Ingeniería Informática  
Curso 2018-2019

*Trabajo Fin de Máster*

# Reconocimiento de señas de la lengua de señas panameña mediante aprendizaje profundo

---

José Herazo Bravo

Tutor

Agapito Ledezma Espino

Leganés (Madrid), 2019

Escuela Politécnica Superior



Esta obra se encuentra sujeta a la licencia *Creative Commons*  
**Reconocimiento – No Comercial – Sin Obra Derivada**



# Agradecimientos

A mi tutor Agapito, por sus consejos, guía y confianza en el desarrollo de este trabajo.

Agradecimientos especiales a quienes participaron en la creación del dataset: Ana, Alex y Andrés.

A mi novia Ana, por darme su apoyo y contribuir con ideas. Y Andrés, por permitirme usar su ordenador.

A la Secretaría Nacional de Ciencia, Tecnología e Innovación de Panamá por darme la oportunidad de estudiar esta carrera.

Y especial mención a mi madre, mi padre y mi hermano, por el apoyo y cariño que recibo de ustedes.

# Resumen

El aumento de la capacidad de procesamiento, los avances en técnicas de aprendizaje automático y la cantidad de datos disponibles, han permitido mejorar el rendimiento de sistemas de reconocimiento de imágenes, permitiendo desarrollar proyectos en áreas como domótica, biología, genética, realidad aumentada, seguridad, y el reconocimiento de lenguas de señas.

Se han desarrollado muchos trabajos que intentan abordar el problema de interpretación de lenguas de señas, sin embargo, a pesar de los esfuerzos, aún no se desarrolla algún sistema que aborde este problema en la lengua de señas panameña.

En este trabajo, se diseña un sistema de traducción de lengua de señas en primera persona utilizando visión artificial sentando una solución conceptual de la cual se desarrolla un clasificador para la traducción de gestos estáticos dactilológicos pertenecientes al lenguaje de señas panameño mediante aprendizaje profundo.

Se entrenan dos clasificadores utilizando redes neuronales convolucionales como algoritmo de clasificación, y se crea y depura un conjunto de datos de entrenamiento que cuenta con alrededor de 55000 imágenes y la participación de 3 usuarios.

Finalmente, los clasificadores son evaluados utilizando el conjunto de datos de pruebas que cuenta con la participación de dos usuarios del cual uno es un nuevo sujeto para los clasificadores.

# Abstract

The increase of the processing capacity, the advances in automatic learning techniques and the amount of data available, have allowed to improve the performance of image recognition systems, allowing the development of projects in areas such as domotics, biology, genetics, augmented reality, security, and the recognition of sign languages.

Many works have been developed to address the problem of translating sign languages, however, despite the efforts, a system that addresses this problem for the Panamanian sign language has not been developed yet.

In this work, a first-person vision translation system for sign language using is designed, setting up a conceptual solution from which a classifier is developed for the translation of static gestures belonging to Panamanian sign language through deep learning techniques.

Two classifiers are trained using convolutional neural networks as a classification algorithm and a set of training data is created and refined with around 55000 images and the participation of 3 users.

Finally, the classifiers are evaluated using a test dataset that has the participation of two users of which one is a new subject for the classifiers.

# ÍNDICE DE CONTENIDO

---

Capítulo 1. Introducción.....	1
1.1 Motivación .....	1
1.2 Problemática .....	1
1.3 Objetivos .....	2
1.4 Alcance y limitaciones .....	2
1.5 Organización del documento .....	3
1.6 Lista de acrónimos .....	3
Capítulo 2. Estado del arte .....	5
2.1 Gestos manuales .....	5
2.2 La lengua de señas .....	6
2.2.1 Componentes de una lengua de señas.....	6
2.2.2 Diccionario dactilológico de la lengua de señas panameña .....	8
2.3 Situación de personas con discapacidad auditiva en Panamá.....	9
2.3.1 Población .....	9
2.3.2 Situación económica y social .....	10
2.4 Reconocimiento de gestos .....	11
2.4.1 Tecnologías de detección.....	11
2.4.2 Métodos de reconocimiento .....	13
2.5 Redes neuronales convolucionales .....	14
2.5.1 Historia .....	14
2.5.2 Extracción de características .....	15
2.5.3 Optimización .....	19
2.5.4 Regularización .....	20
2.5.5 Funciones de activación.....	21
2.5.6 Rendimiento .....	21
2.5.7 Desventajas .....	22
2.5.8 Metodología de desarrollo .....	22
2.5.9 Conclusiones.....	23
Capítulo 3. Análisis y diseño.....	25
3.1 Elección de diseño .....	25
3.1.1 Sobre el uso de cámaras.....	25

3.1.2	Sobre el algoritmo de clasificación .....	26
3.2	Diseño conceptual general de la solución .....	27
3.3	Diseño general del sistema desarrollado .....	29
3.4	Requisitos del sistema .....	30
Capítulo 4.	Creación del <i>dataset</i> .....	34
4.1	Recolección de datos .....	34
4.1.1	Usuarios y entornos .....	34
4.1.2	Procesos de recolección de datos .....	34
4.2	Proceso de aumento de datos .....	36
4.2.1	Agregación de fondo .....	36
4.2.2	Otras transformaciones .....	37
4.3	Detalles adicionales del <i>dataset</i> .....	38
4.4	Herramientas utilizadas .....	39
4.4.1	Cámaras .....	39
4.4.2	Herramientas software .....	42
4.4.3	Ordenador de desarrollo .....	42
4.5	Algoritmos desarrollados .....	42
4.5.1	Extracción de imágenes de vídeos .....	42
4.5.2	Cambio de fondos .....	43
Capítulo 5.	Creación del clasificador .....	44
5.1	Algoritmo de clasificación .....	44
5.1.1	Proceso de entrenamiento .....	44
5.1.2	Arquitectura .....	45
5.2	Herramientas utilizadas .....	47
5.2.1	Herramientas software .....	47
5.2.2	Hardware de aprendizaje profundo .....	47
Capítulo 6.	Pruebas y resultados .....	49
6.1	Descripción de las pruebas .....	49
6.2	Resultados de pruebas .....	50
6.2.1	Prueba P1.S .....	50
6.2.2	Prueba P1.I .....	51
6.2.3	Prueba P2.S .....	52
6.2.4	Prueba P2.I .....	53

6.3	Conclusiones y consideraciones .....	54
Capítulo 7.	Gestión del proyecto .....	57
7.1	Metodología de desarrollo .....	57
7.2	Planificación seguida por el estudiante.....	58
7.3	Planificación con equipo de trabajo .....	58
7.4	Presupuesto .....	60
7.4.1	Recursos humanos .....	60
7.4.2	Materiales y costes adicionales.....	62
7.4.3	Costes totales .....	62
7.5	Aspectos legales.....	63
7.6	Gestión de riesgos.....	63
7.6.1	Análisis de riesgos .....	64
7.6.2	Planes de acción.....	65
Capítulo 8.	Conclusiones.....	66
8.1	Conclusiones.....	66
8.2	Trabajos futuros .....	67
Referencias	.....	69



# ÍNDICE DE ILUSTRACIONES

---

Ilustración 1. Partes del cuerpo implicadas en los gestos (basado en: [3]) .....	5
Ilustración 2. Representación de la seña para la palabra week según el modelo <i>Movement-Hold</i> (basado en: [13] y [14]) .....	7
Ilustración 3. Alfabeto manual en lengua de señas panameña (fuente: [20]).....	8
Ilustración 4. Personas con discapacidad por cada mil habitantes de cada sexo según tipo (fuente: [21]).....	9
Ilustración 5. Personas con discapacidad por grupos de edad y tipo por cada mil habitantes. (fuente: [21]).....	10
Ilustración 6. Guante de datos de <i>CyberGlove III</i> (fuente: [24]).....	12
Ilustración 7. Guante de color utilizado por Luigi et al. (fuente: [31]) .....	13
Ilustración 8. Capa común de extracción de características conformada por una capa de convolución seguida de <i>pooling</i> .....	15
Ilustración 9. Arquitectura típica de una <i>CNN</i> basado en [45]......	16
Ilustración 10. Diferentes tipos de datos de entrada para una <i>CNN</i> y su profundidad ...	17
Ilustración 11. Operación de Convolución de una <i>CNN</i> .....	17
Ilustración 12. Operación <i>Max Pooling</i> 2x2 en una matriz de 4x4 .....	18
Ilustración 13. Red neuronal estándar y red neuronal después de aplicar <i>dropout</i> (fuente [55]). .....	20
Ilustración 14. Señas para letras K e Y observadas desde distintas perspectivas.....	25
Ilustración 15. Diagrama de Componentes del prototipo de solución propuesta .....	28
Ilustración 16. Diagrama de componentes sistema desarrollado.....	29
Ilustración 17. Etapas del desarrollo del sistema (ciclo de vida).....	30
Ilustración 18. Seña correspondiente a la letra H sobre un fondo verde, salón y un parque realizada por tres usuarios distintos. ....	34
Ilustración 19. Seña correspondiente a la letra O recolectada en un parque y un salón con variaciones en el fondo y proyección de sombras. ....	35
Ilustración 20. Seña de letra B generada por un usuario sobre fondo uniforme con dos proyecciones de luz distintas. ....	36
Ilustración 21. Imágenes con fondos nuevos.....	37
Ilustración 22. Variaciones en rotación, intensidad de luz y escalado en una imagen... ..	37
Ilustración 23. Montura y cámara superior.....	39
Ilustración 24. Usuario con cámara superior montada .....	40

Ilustración 25. Montura y cámara inferior.....	40
Ilustración 26. Usuario con cámara inferior montada .....	41
Ilustración 27. Campo de observación de ambas perspectivas de cámara para la seña correspondiente a la letra Q.....	41
Ilustración 28. Diagrama de flujo algoritmo de extracción de imágenes .....	43
Ilustración 29. Algoritmo para cambio de fondo de imágenes.....	43
Ilustración 30. Uso de <i>dropout</i> .....	45
Ilustración 31. Arquitectura <i>CNN</i> modelo superior.....	46
Ilustración 32. Arquitectura <i>CNN</i> modelo inferior.....	46
Ilustración 33. Señas para las letras D y O de los datasets de pruebas P1.I y P2.I .....	50
Ilustración 34. Matriz de confusión del modelo de cámara superior para la prueba P1.S (105 instancias).....	50
Ilustración 35. Medidas de precisión y <i>recall</i> del modelo de cámara superior para la prueba P1.S (105 instancias) .....	50
Ilustración 36. Matriz de confusión del modelo de cámara inferior para la prueba P1.I (225 instancias).....	51
Ilustración 37. Medidas de precisión y <i>recall</i> del modelo de cámara inferior para la prueba P1.I (255 instancias) .....	52
Ilustración 38. Matriz de confusión del modelo de cámara superior para la prueba P2.S (105 instancias).....	52
Ilustración 39. Medidas de precisión y <i>recall</i> del modelo de cámara superior para la prueba P2.I (105 instancias) .....	53
Ilustración 40. Matriz de confusión del modelo de cámara inferior para la prueba P2.I (255 instancias).....	53
Ilustración 41. Medidas de precisión y <i>recall</i> del modelo de cámara inferior para la prueba P2.I (255 instancias) .....	54
Ilustración 42. Señas para letras E e I ejecutadas por el usuario de la prueba P2.I.....	55
Ilustración 43. Señas para las letras D y X de ejecutadas por los usuarios de las pruebas P1.I y P2.I.....	55
Ilustración 44. Señas para las letras M, N y S ejecutadas por los usuarios de las pruebas P1.S y P2.S .....	56
Ilustración 45. Tabla de calificación de nivel de riesgos según probabilidad e impacto	64

# ÍNDICE DE TABLAS

---

Tabla 1. Niveles de prioridad y necesidad de los requisitos.....	31
Tabla 2. Lista y descripción de requisitos .....	31
Tabla 3. Tamaño del <i>dataset</i> .....	38
Tabla 4. Selección de datos para entrenamiento, validación y pruebas .....	39
Tabla 5. Especificaciones Técnicas Ordenador de Desarrollo .....	42
Tabla 6. Especificaciones técnicas máquina virtual .....	47
Tabla 7. Especificaciones técnicas ordenador personal.....	47
Tabla 8. Especificaciones técnicas del ordenador de altas prestaciones .....	48
Tabla 9. Pruebas realizadas .....	49
Tabla 10. Cronograma de trabajo seguido por el estudiante .....	58
Tabla 11. Cronograma de trabajo .....	58
Tabla 12. Duración y coste de actividades para el Director de Proyecto .....	60
Tabla 13. Duración y coste de actividades para el Experto en <i>Machine Learning</i> .....	61
Tabla 14. Duración y coste de actividades para el Desarrollador. ....	61
Tabla 15. Duración y coste de actividades para el Experto en lengua de señas.....	62
Tabla 16. Costes de materiales .....	62
Tabla 17. Desglose de costes del proyecto .....	62
Tabla 18. Riesgos identificados y su calificación.....	64
Tabla 19. Planes de acción .....	65

# Capítulo 1. INTRODUCCIÓN

---

Este capítulo aborda el tema introductorio del trabajo de fin de máster. Se expone la motivación del trabajo, la definición de la problemática, los objetivos del proyecto, su alcance y limitaciones, y se presenta la organización del documento.

## 1.1 MOTIVACIÓN

El aumento de la capacidad de procesamiento, los avances en técnicas de aprendizaje automático y la cantidad de datos disponibles, han permitido mejorar el rendimiento de sistemas de reconocimiento de imágenes, permitiendo desarrollar proyectos en áreas como domótica, biología, genética, realidad aumentada, seguridad, entre otros.

Se han desarrollado muchos trabajos que intentan abordar el problema de interpretación de lenguas de señas, llegándose a proponer sistemas de reconocimiento de señas para distintas lenguas, entre ellas: la Lengua de Señas India, Lengua de Señas Japonesa, Lengua de Señas Mexicana, etc. Sin embargo, a pesar de los esfuerzos, aún no se desarrolla algún sistema que aborde este problema en la Lengua de Señas Panameña, con lo cual el sistema propuesto es de los primeros en su tipo en Panamá.

Cooper *et al.* en [1] hace una revisión del estado del arte del reconocimiento de lengua de señas concluyendo que es un tema que aún se encuentra en su infancia y falta mucho por desarrollarse, razón por la cual, cualquier aportación es de vital importancia.

En cuanto al desarrollo de un sistema de reconocimiento de señas en primera persona, la mayoría de los sistemas desarrollados basados en visión artificial implican el uso de una cámara sostenida por una tercera persona, enfoque que puede ser inapropiado en algunos casos, ya que la persona con discapacidad no siempre va a encontrarse personas con las que quiera comunicarse que cuenten con un sistema de traducción. Permitirles a las personas con discapacidad auditiva contar con un sistema que pueda traducir sus señas y les aporte autonomía puede ser considerado de mayor utilidad.

Parte del sistema desarrollado en este trabajo de fin de máster sirve como fundamento para el desarrollo de sistemas de traducción de lenguas señas, en especial la lengua de señas panameña, con lo cual podría beneficiarse la población de personas con discapacidad auditivas en Panamá, que para el año 2010, se había cuantificado en un total de 15 mil personas [2].

## 1.2 PROBLEMÁTICA

Las personas con discapacidad auditiva necesitan comunicarse con otras personas a través de una lengua de señas, lengua que las personas sin discapacidad podrían ignorar, llevando a que las personas con problemas de audición tiendan a tener problemas para comunicarse en entornos sociales

Un sistema de reconocimiento de señas puede desarrollarse utilizando distintos enfoques, como puede ser, utilizando guantes con sensores para recolectar datos precisos de la configuración de la mano, o utilizando cámaras para posteriormente procesar las imágenes y generar modelos para traducir señas. El segundo enfoque ha sido muy popular desde el desarrollo de nuevas técnicas de aprendizaje automático y es el enfoque que seguirá el sistema propuesto.

El problema tratado en este trabajo es desarrollar un sistema que permita la traducción de la lengua de señas panameña y que le aporte autonomía a la persona con discapacidad utilizando visión artificial.

### 1.3 OBJETIVOS

El objetivo general de este trabajo es aplicar técnicas de aprendizaje profundo para el reconocimiento de señas de la lengua de señas panameña.

El desarrollo de este proyecto conlleva la definición de los siguientes objetivos específicos:

- Identificar y analizar las distintas soluciones en el estado del arte sobre reconocimiento de gestos.
- Proponer una base conceptual de un sistema de reconocimiento de lengua de señas en primera persona utilizando visión artificial
- Construir un conjunto de datos con imágenes de señas capturadas en primera persona.
- Entrenar y evaluar clasificadores para el sistema propuesto.

### 1.4 ALCANCE Y LIMITACIONES

El desarrollo e implementación de un trabajo como el propuesto puede llegar a ser muy costosa en cuanto a recursos y tiempo. Por tal razón, el trabajo no incluye la implementación y puesta en marcha de todo el sistema propuesto, limitándose únicamente en el desarrollo y evaluación de los clasificadores.

En cuanto a interpretación de señas, interpretar todo el abanico de señas de la lengua de señas panameña es muy complejo debido a la cantidad de gestos que abarca. Además, las señas con las que nos podemos encontrar pueden ser estáticas o dinámicas, dependiendo de la inclusión de un movimiento en la ejecución de la seña o no; y pueden incluir una o ambas manos. El trabajo se enfocará únicamente en traducir señas estáticas del alfabeto manual de la lengua de señas panameña, limitándose a unas 24 señas que representan letras del abecedario.

En cuanto al conjunto de datos, el proyecto se limita a la contribución de cuatro usuarios en la recolección del *dataset* dejando abierta la posibilidad de extender el tamaño del *dataset* en futuros trabajos.

En general, el trabajo no se extiende más allá de proponer un modelo general de un sistema de reconocimiento de señas en primera persona utilizando visión artificial y, desarrollar y evaluar un clasificador para el modelo propuesto.

## 1.5 ORGANIZACIÓN DEL DOCUMENTO

En este apartado se describe la organización del resto del documento.

En el Capítulo 2, Estado del arte, se presenta una revisión de la bibliografía de los temas abordados en el trabajo, como son: modelos lingüísticos para las lenguas de señas, trabajos que proponen sistemas de reconocimiento de gestos con distintos enfoques, el estado de las redes neuronales convolucionales y se analiza la situación actual de las personas con discapacidad auditiva en Panamá desde una perspectiva demográfica y social, y su lengua de señas.

En el Capítulo 3, Análisis y diseño, se analiza y propone un sistema de reconocimiento de lengua de señas basado en visión en primera persona y se detallan sus componentes. Se menciona las razones de ciertos aspectos de diseño y se detallan los requisitos del sistema desarrollado en este trabajo.

El Capítulo 4, Creación del *dataset*, presenta los pasos realizados para recolectar y depurar un conjunto de datos, el número de usuarios involucrados, la selección de datos para entrenamiento, validación y pruebas, y las herramientas utilizadas.

En el Capítulo 5, Creación del clasificador, se describe la creación de las redes neuronales convolucionales. Su proceso de entrenamiento, arquitectura utilizada, y herramientas utilizadas en el desarrollo.

El Capítulo 6, Pruebas y resultados, presentan los resultados de las pruebas de los clasificadores y su análisis. También se evalúan otros aspectos de diseño del sistema y detalles del rendimiento del clasificador.

En el Capítulo 7, Gestión del proyecto, se presenta la metodología utilizada en el desarrollo del sistema, el presupuesto, planificación del trabajo y gestión de riesgos.

Finalmente, el Capítulo 8, Conclusiones, expone las conclusiones del trabajo, las competencias alcanzadas, las posibles mejoras del sistema y trabajos futuros.

## 1.6 LISTA DE ACRÓNIMOS

A continuación, se presenta un listado de los acrónimos que se utilizan en la redacción del documento.

- 2D (Dos Dimensiones)
- 3D (Tres Dimensiones)
- *3dCNN* (*3D Convolutional Neural Network* - Redes Neuronales Convolucionales 3d)
- *AdaGrad* (*Adaptive Gradient* - Gradiente Adaptativa)

- *ADAM* (derivado de *Adaptative Moment Estimation* – Estimación Adaptativa de Momento)
- *CNN* (*Convolutional Neural Network* - Redes Neuronales Convolucionales)
- *CPU* (*Central Processing Unit* - Unidad Central de Procesamiento)
- *ELU* (*Exponential Lineal Unit* - Unidad Lineal Exponencial)
- *FPS* (*Frames per second* - Imágenes por Segundo)
- *GPU* (*Graphic Processing Unit* - Unidad de Procesamiento Gráfico)
- *HMM* (*Hidden Marcov Models* - Modelos Ocultos de Marcov)
- *HPC* (*High Performance Computer* - Ordenador de Altas Prestaciones)
- *kNN* (*k-Nearest Neighbor* - k-Vecinos más Cercanos)
- *LReLU* (*Leaky Rectified Lineal Unit*)
- *ML* (*Machine Learning* - Aprendizaje Automático)
- *PReLU* (*Parametric Rectified Lineal Unit* - Unidad Lineal Rectificada Paramétrica)
- *ReLU* (*Rectified Lineal Unit* - Unidad Lineal Rectificada)
- *RMSProp* (*Root Mean Square Propagation*)
- *RNN* (*Recurrent Neural Network* - Red Neuronal Recurrente)
- TFM (Trabajo de Fin de Máster)
- *SELU* (*Scaled Exponential Lineal Unit* – Unidad Lineal Exponencial Escalada)
- *SGD* (*Sthocastic Gradient Descent* - Descenso de Gradiente Estocástico)
- *SVM* (*Support Vector Machine* - Máquinas de Vectores de Soporte)

## Capítulo 2. ESTADO DEL ARTE

Este apartado corresponde a la revisión bibliográfica de temas relacionados al tema tratado. Algunos de estos temas forman parte fundamental para comprender el problema.

Se empieza por mencionar algunas estadísticas sobre la implicación de las manos en los gestos como manera de concienciación de la importancia del reconocimiento de gestos manuales. Luego se trata el modelado de las lenguas de señas desde un punto de vista lingüístico. Continúa con una breve introducción a la Lengua de Señas Panameña y su situación económico-social que sirve para reconocer el impacto del sistema propuesto. Se profundiza en los sistemas de reconocimiento de gestos, sus etapas de desarrollo, y tecnologías utilizadas. Finalmente, se repasa el estado del arte de las redes neuronales convolucionales reconociendo sus aportaciones en los sistemas de reconocimiento de gestos.

### 2.1 GESTOS MANUALES

Antes de abordar el tema de la lengua de señas, es necesario estudiar la importancia del uso de las manos en los gestos. Karam en su tesis doctoral [3], hizo una revisión literaria sobre las partes del cuerpo involucrada en el reconocimiento de gestos en el mundo de la interacción humano-computador, y muestra que las manos están involucradas en más del 60% de los gestos. Esto es razonable debido a que naturalmente para los humanos las manos son usadas como medios de comunicación.

Las manos son utilizadas en combinación con otras partes del cuerpo. La Ilustración 1 resume los hallazgos realizados por Karam, lo cual apunta a que la interpretación de gestos manuales es más atractiva en las aplicaciones desarrolladas que contemplen reconocimiento de gestos en la interacción.

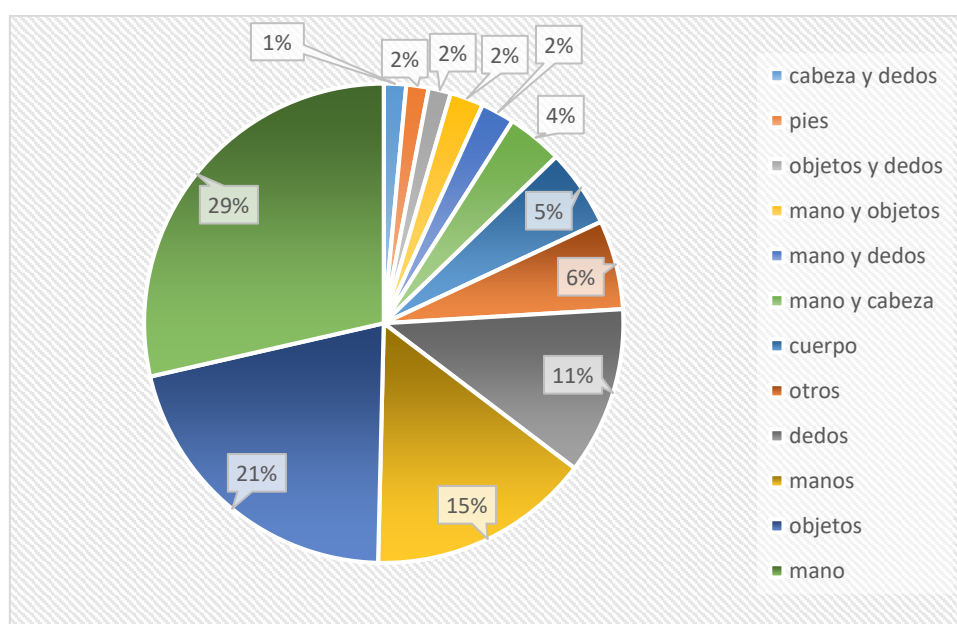


Ilustración 1. Partes del cuerpo implicadas en los gestos (basado en: [3])



El reconocimiento de gestos manuales y su posterior interpretación es el enfoque seguido por el sistema desarrollado. Similar a otras aplicaciones que intentan crear una interfaz adicional para el control del ordenador a través del reconocimiento de gestos como es el caso de [4] y [5], y demás aplicaciones que abordan la traducción de lenguas de señas como: Patel *et al.* [6] en el reconocimiento de lengua de señas indio, Nagasue *et al.* [7] desarrolla un sistema de traducción de señas en primera persona para la lengua de señas japonesa, o Aryanie *et al.* [8] con su sistema de traducción del alfabeto de la lengua de señas americana.

Estos gestos pueden clasificarse en base a relaciones temporales en: gestos estáticos y dinámicos [9]. Los gestos estáticos, son aquellos en los cuales no existe cambio alguno durante la ejecución del gesto, no ocurre movimiento. En cambio, los gestos dinámicos son aquellos en los cuales la posición y configuración de la mano cambia continuamente. Parte principal de las lenguas de señas son los gestos manuales, entre los cuales suele haber gestos estáticos y dinámicos, dependiendo de la implicación de movimiento en la ejecución del gesto.

## 2.2 LA LENGUA DE SEÑAS

La Real Academia de la Lengua define lengua como un “*sistema lingüístico considerado en su estructura*” o “*sistema de comunicación verbal propio de una comunidad humana y que cuenta generalmente con escritura*” a diferencia de lenguaje que hace referencia a la “*facultad del ser humano de expresarse y comunicarse con los demás a través del sonido articulado o de otros sistemas de signos*” [10]. Estas diferencias sugieren que se deba utilizar el vocablo lengua para referirse al sistema de comunicación utilizado por grupos de personas con discapacidad auditiva.

Con respecto al uso de la palabra señas o signos para referirse a los gestos expresados por las manos, en España es utilizada la expresión lengua de signos a diferencia de los países hispanoamericanos donde es popular referirse como lengua de señas.

Llevarle lengua de señas o lengua de signos indistintamente al sistema de comunicación utilizado por las personas con discapacidad auditiva en Panamá es aceptable, sin embargo, lengua de señas es el nombre oficial en Panamá.

Se considera dentro del documento la locución aceptada por la Organización de las Naciones Unidas (ONU por sus siglas en inglés) en la Convención sobre los Derechos de las Personas con Discapacidad [11], artículo 2:

“Por “lenguaje” se entenderá tanto el lenguaje oral como la lengua de señas y otras formas de comunicación no verbal.”

### 2.2.1 Componentes de una lengua de señas

La lengua de señas puede llegar a estar compuesta por la configuración de las manos, los gestos faciales, la postura corporal, y movimientos; y se puede abordar con diferentes modelos, como los propuestos por Stokoe y, Liddle y Johnson en [12].

Uno de los trabajos más citados sobre la estructura de la lengua de señas es el realizado por William Stokoe. En 1960, Stokoe propone que una seña tiene tres parámetros que se combinan simultáneamente. Estos tres parámetros son: la configuración de la mano, la ubicación y el movimiento. En sus trabajos, la orientación de las manos y los gestos no manuales se analizaron indirectamente [13].

Años después, Liddle y Johnson, en sus estudios sobre la lengua de señas americana, definen un modelo llamado *Movement-hold system* (Sistema de Movimiento-Retención), en donde proponen que una seña es una composición de segmentos fonológicos, similar a la composición de palabras en una lengua hablada. Estas señas son combinaciones secuenciales de retenciones y movimientos, cada uno con sus propias características [12].

Dentro de las características que puede tener una seña en cada segmento identifican cuatro: configuración de las manos, orientación, ubicación y gestos no manuales. Por ejemplo, la palabra *week* en inglés empieza con diferentes configuraciones para ambas manos y termina con una configuración diferente para la mano derecha (Ilustración 2).

Palabra: week			
	Segmentos		
MANO DERECHA	retención	movimiento	retención
configuración	forma 1		forma 1
ubicación	base de mano izquierda		punta de mano izquierda
orientación	palma abajo		palma abajo
gesto no manual	-		
MANO IZQUIERDA			
configuración	forma B		-
ubicación	al frente del torso		-
orientación	palma arriba		-
gesto no manual	-		-






Ilustración 2. Representación de la seña para la palabra *week* según el modelo *Movement-Hold* (basado en: [13] y [14])

También es posible clasificar los gestos en dinámicos y estáticos. Los gestos dinámicos son aquellos que llevan consigo algún movimiento, mientras que los estáticos no poseen movimiento alguno. Y en unimanuales o bimanuales, que hacen referencia al uso de una o ambas manos.

Además de estos componentes, toda lengua de señas cuenta con un diccionario dactilológico o alfabeto manual, que es la representación de las letras del alfabeto escrito en señas manuales. Este recurso es imprescindible en la educación en las escuelas de personas con discapacidad auditiva y puede ser usado para expresar nombres personales y palabras de una lengua hablada que no estén definidas dentro de una lengua de señas

[15] [16]. A este tipo de recursos se les conoce como *fingerspelling* en la lengua inglesa, problema de mucho interés en la comunidad científica, abordados en trabajos como el caso de [7], [17], [6], [18], [8], [19], similares al sistema propuesto en otras lenguas de señas.

Los modelos revisados hasta ahora pueden servir para modelar la lengua de señas panameña y proponer un sistema que pueda traducir gran parte de la lengua. Sin embargo, dado que el sistema que se desarrolla en este trabajo intenta traducir parte del diccionario dactilológico panameño, algunos de estos componentes no son necesarios, por lo que se dedica especial atención a la configuración de las manos.

### 2.2.2 Diccionario dactilológico de la lengua de señas panameña

La lengua de señas panameña, al igual que otras lenguas de señas, es tan compleja como una lengua hablada y es de vital importancia la interpretación de gestos faciales.

La lengua de señas panameña consta de su propio alfabeto manual, que difiere de otras lenguas de señas, y tiene sus propias señas para determinados conceptos. Por ejemplo: el mes de noviembre se define con un gesto que hace alusión a una persona tocando un tambor, lo cual es razonable en el país ya que dicho mes se caracteriza por tener varios días de desfiles donde predominan los tambores.

En el alfabeto manual de la lengua de señas panameña, cada letra está definida por una seña que puede representarse con una mano. Algunas letras como la eñe, la jota, la zeta y los dígrafos doble ele y doble erre implican movimiento, las demás letras del alfabeto se pueden representar con gestos estáticos. La Ilustración 3 muestra las configuraciones de las manos para cada una de las letras del alfabeto en lengua de señas panameña.

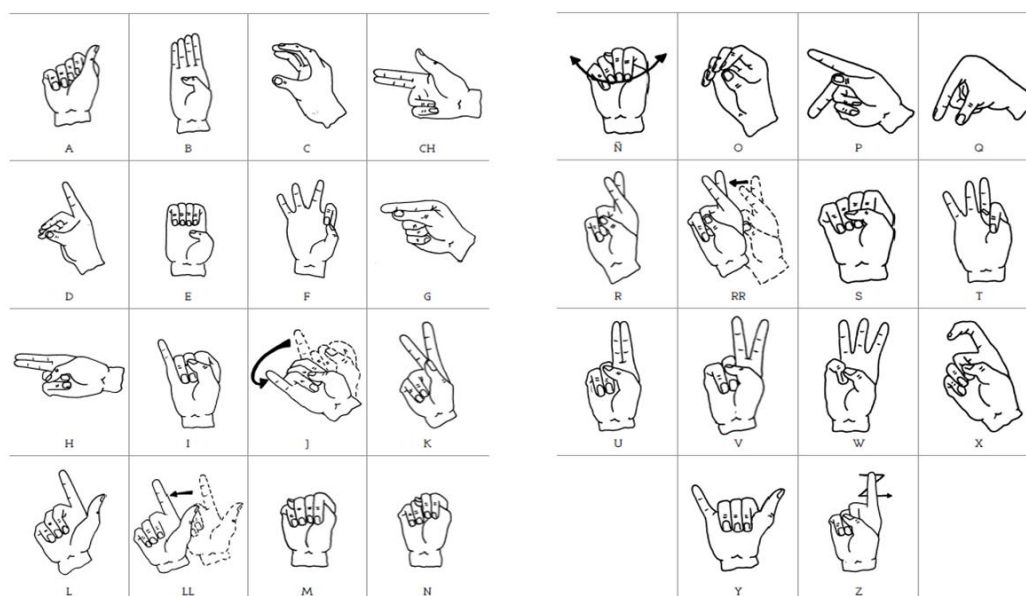


Ilustración 3. Alfabeto manual en lengua de señas panameña (fuente: [20])

En la revisión bibliográfica no se han encontrado proyectos que aborden el problema de la interpretación del diccionario dactilológico de la lengua de señas panameña, con lo cual este proyecto sería el primero que aborde esta lengua de señas.

## 2.3 SITUACIÓN DE PERSONAS CON DISCAPACIDAD AUDITIVA EN PANAMÁ

El Ministerio de Economía y Finanzas de Panamá publicó en el año 2010 un informe con estadísticas de personas con discapacidad en el país que nos sirve para conocer el impacto social de este trabajo. A pesar de haber pasado varios años, la última publicación encontrada con datos estadísticos sobre personas con discapacidad en el país data del año 2010.

En el informe una persona con sordera está definida como una persona que no puede oír ni hablar, aún con la ayuda de audífonos y se comunica por medio de señas, pero su inteligencia es normal. Formando parte del grupo que se podría beneficiar a futuro del desarrollo del sistema propuesto una vez implementado.

### 2.3.1 Población

En cuanto a la población de personas sordas, se menciona que para el año 2010, un 2,9% de la población padecía un tipo de discapacidad, del cual un 15,6% presentaban sordera, correspondiendo a un total de 15191 personas [2]. La siguiente ilustración muestra la distribución de las personas con discapacidad al momento, de la que podemos observar que la sordera es la cuarta mayor discapacidad presentada en el país después de la deficiencia física, ceguera y discapacidad mental.

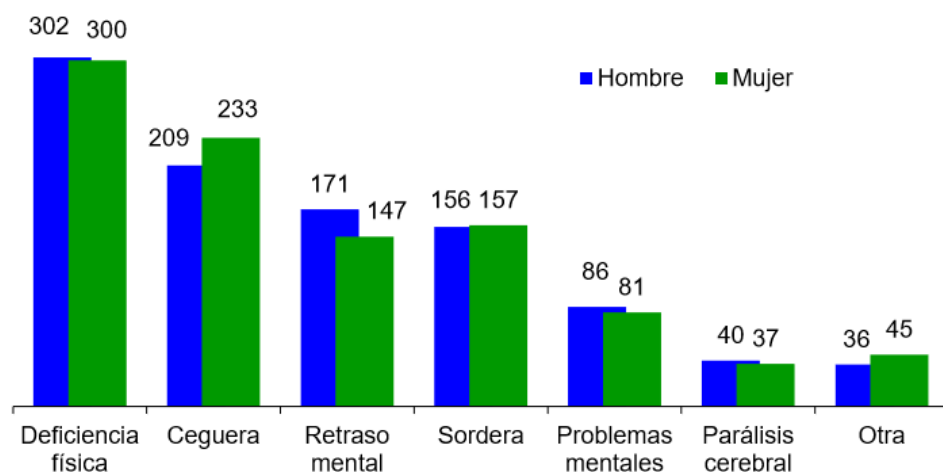


Ilustración 4. Personas con discapacidad por cada mil habitantes de cada sexo según tipo (fuente: [21])

Además, las personas con discapacidad se distribuyen según edad siguiendo una distribución exponencial (Ilustración 5). Siendo las personas más afectadas con sordera aquellas que llegan a edades avanzadas. Sin embargo, un grupo de los afectos no supera los 20 años, edad en la que el número de actividades que pueden realizar es mayor y pueden verse condicionadas, por ejemplo: acceder a educación.

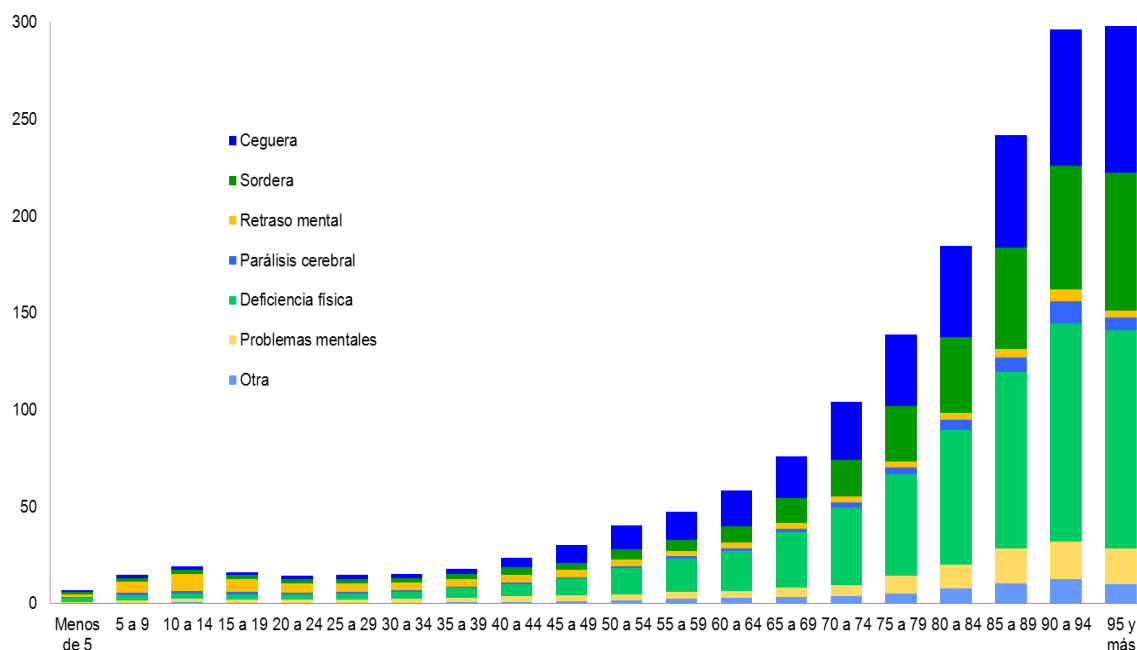


Ilustración 5. Personas con discapacidad por grupos de edad y tipo por cada mil habitantes. (fuente: [21])

### 2.3.2 Situación económica y social

Acerca de la población de personas con discapacidad económicamente activa, que hace referencia a las personas de más de 15 años que aportan mano de obra disponible para producir bienes y servicios económicos, sólo un 26,0% del total de personas con discapacidad se mantienen ocupadas, donde el resto podrían estar desocupadas por condiciones de salud, ser estudiantes o estar jubilados. Y de cada 100 personas económicamente activas, 92 se encontraban ocupadas (población que padecía ceguera, deficiencia física o sordera) [21].

Con respecto al tema de nivel de escolaridad, en Panamá el nivel de escolaridad se puede dividir sencillamente, en tres niveles: educación básica general, que tiene una duración de once años y se subdivide en dos años de preescolar, seis años de primaria y tres años de premedia; educación media, con una duración de tres años; y educación superior que corresponde a estudios universitarios [22]. El informe del Ministerio de Economía y Finanzas muestra que, a pesar de que el nivel de escolaridad tiene estrecha relación con la inserción en el mercado en la población, para las personas con alguna discapacidad estar escolarizado representa un problema. La mayoría de la población ocupada panameña con discapacidad apenas tenía una formación básica, en contraste con la población total con un 24,3% de personas con educación secundaria.

A pesar de mostrarse que parte de las personas con sordera se encuentran económicamente activas o estudiando, no se citan los problemas que estos padecen en sus ocupaciones, que pueden estar ligados a su capacidad para comunicarse. Un sistema de traducción de lengua de señas podría facilitar la comunicación a las personas con sorderas en el país, acceder a educación y así permitirles incluirse en la sociedad con menos dificultades y favorecer una inserción laboral satisfactoria.

## **2.4 RECONOCIMIENTO DE GESTOS**

Una arquitectura básica de este tipo de sistemas consta de tres etapas que son: detección, seguimiento y reconocimiento. El objetivo de la primera etapa es capturar la mano y separarla de los demás objetos. La segunda etapa implica seguir el movimiento que hace la mano en el entorno. Y la tercera etapa es la interpretación semántica del gesto [23].

La etapa de seguimiento es útil para el reconocimiento de gestos dinámicos, como puede ser apuntar un objeto con el dedo índice y moverlo de lado a lado junto con la mano. Sin embargo, como en este trabajo se desarrolla un clasificador de señas estáticas, se enfoca mayormente en las tecnologías de detección y reconocimiento, más que en las tecnologías de seguimiento.

### **2.4.1 Tecnologías de detección**

Existen diferentes tecnologías que pueden ser utilizadas para capturar la información interpretada por una persona con discapacidad auditiva. Dependiendo de la entrada de captura de información el sistema puede recolectar más o menos datos, haciendo cambiar el enfoque para abordar el problema.

En general, las tecnologías de captura de información se pueden clasificar en dos grandes grupos que son: sistemas basados en contacto; y sistemas basados en visión [23]. Los sistemas basados en contacto incluyen guantes y sensores que porta la persona con discapacidad, mientras que los sistemas basados en visión basan su funcionamiento en cámaras.

#### ***2.4.1.1 Sistemas basados en contacto***

Los sistemas basados en contacto usualmente son guantes que intentan recolectar información de la posición de distintas articulaciones de los dedos y manos con el fin de determinar su configuración. También se les conoce como guantes de datos. Una gran ventaja de esto es no necesitar una etapa de procesamiento de datos para obtener descriptores, como puede ser el caso de una imagen obtenida de una cámara [16].

De este tipo de soluciones existen productos comerciales. Por ejemplo, los desarrollados por *CyberGlove Systems* [24], *AnthroTronicx* [25] y *Fifth Dimension Technologies* [26]. La Ilustración 6 muestra la *Cyber Glove III*, producto comercial de *CyberGlove*.



*Ilustración 6. Guante de datos de CyberGlove III (fuente: [24])*

Algunos investigadores han abordado el problema de la interpretación de lenguaje de señas utilizando guantes con sensores, como es el caso de Jani *et al.* en [27] que diseñaron un guante sensorial y Samraj *et al* en [28] que prefirieron el uso de una solución comercial (5DT Data Glove Ultra de Fifth Dimention Technologies). Usualmente estos tipos de dispositivos son costosos e implican que el usuario deba tenerlos puestos.

#### **2.4.1.2 Sistemas basados en visión**

En el uso de sistemas de visión se pueden emplear cámaras infrarrojas, combinaciones de distintas cámaras creando una aproximación 3D del espacio, el sensor *Kinet* de Microsoft o cámaras convencionales.

El uso de este tipo de tecnologías implica un posterior preprocesamiento de la imagen que puede involucrar la detección de color, detección de manos y rostro, recorte de áreas de interés, reducción de ruido, etc.

El empleo de una sola cámara para el reconocimiento de configuraciones de las manos acota el espacio a imágenes 2D, lo que complica estimar profundidades. Para evitar este problema, algunos autores han diseñado combinaciones de cámaras en distintas posiciones con el fin de obtener información adicional como es el caso de Starner *et al.* en [29] que utilizan dos cámaras y Feris *et al.* en [30] que utilizan fuentes de luz para modelar profundidades.

También es posible añadir al guante etiquetas de colores en las manos para facilitar la detección, tal como lo realizaron de Luigi *et al.* en [31] usando guantes con colores específicos en los dedos para facilitar el reconocimiento. El guante de color utilizado se puede apreciar en la Ilustración 7.



Ilustración 7. Guante de color utilizado por Luigi *et al.* (fuente: [31])

Este tipo de soluciones permiten que el usuario goce de cierta libertad a diferencia de los guantes de datos que significan un sistema adicional en las manos y pueden llegar a restringir sus movimientos.

#### 2.4.1.3 Otros enfoques

A parte de los sistemas convencionales que utilizan cámaras o guantes con sensores, también se han desarrollado intentos más novedosos utilizando otras tecnologías como es el caso de Ekiz *et al* en [32] que utilizan *smartwatches* para recolectar datos para posteriormente traducir las señas y Ma *et al* [33] que utilizan señales *WiFi* y redes neuronales convolucionales para traducir 276 gestos.

Estos distintos enfoques podrían combinarse para conseguir sistemas más robustos y precisos.

#### 2.4.2 Métodos de reconocimiento

El problema de reconocimiento de señas puede subdividirse en dos tipos: problemas de reconocimiento de señas simples, donde el objetivo es identificar letras o palabras en específico sin componer oraciones; o reconocimiento continuo, donde se intenta modelar la gramática de la lengua y generar frases.

En la actualidad los sistemas de clasificación de gestos simples usualmente utilizan algoritmos de aprendizaje automático. Algunos de ellos pueden ser máquinas de vectores de soporte, métodos no paramétricos como  $k$  vecinos más cercanos ( $kNN$  por sus siglas en inglés), redes neuronales, algoritmos de aprendizaje profundo, etc.

En el ámbito de reconocimiento de señas simples, Rao *et al.* en [34] implementaron un sistema para reconocer señas en modo *selfie* utilizando redes neuronales convolucionales. Por el contrario, Nagasue *et al.* [7] prefiriendo preprocesar la imagen generando una representación binaria de la mano para posteriormente clasificarla utilizando el algoritmo de  $kNN$ .

Trabajos más antiguos han intentado clasificar gestos utilizando arboles de decisión como es el caso de Kadous [35] en el reconocimiento de gestos de la lengua de señas australiana.



Algunas señas pueden implicar gestos con movimiento, por ejemplo: una palabra representada por el gesto de abrir y cerrar el puño. En este tipo de problemas es posible utilizar redes neuronales convolucionales 3D (*3dCNN* por sus siglas en inglés) como es el caso de los trabajos de Soodtoetong y Gedkhawen [36] y Huang *et al.* en [37] en la clasificación de gestos dinámicos.

Otra alternativa para reconocer secuencias son las redes neuronales recurrentes (*RNN*), que han demostrado funcionar bien en problemas con características temporales. Lai y Yanushkevich en [38] donde utilizan *RNN* en combinación con redes neuronales convolucionales para reconocer secuencias de gesto.

En caso del problema de clasificación continua, una manera común de abordarlo es usando Modelos Ocultos de Markov (*HMM* por sus siglas en inglés) debido su capacidad de modelar características temporales. Sin embargo, este tipo de problemas implica reconocer palabras antes de modelar frases. Starner *et al.* [29] diseñaron un sistema que reconoce exitosamente frases de un léxico de 40 palabras de la lengua de señas americana utilizando *HMM*, y Liu en [39] utilizaron *kNN* para reconocer palabras y posteriormente *HMM* para componer secuencias.

## 2.5 REDES NEURONALES CONVOLUCIONALES

Goodfellow *et al.* en [40] define redes neuronales convolucionales (*CNN* por sus siglas en inglés) como un tipo de red neuronal que utiliza convoluciones en vez de multiplicación de matrices en al menos una de sus capas. En otras palabras, las *CNN* son un tipo de red neuronal que implementa en sus neuronas de entrada una operación matemática llamada convolución que facilita el proceso de detección de características en los datos. Este proceso sustituye el procedimiento de ingeniería de características reduciendo el tiempo de preprocesado de datos.

### 2.5.1 Historia

Las *CNN* inspiradas en los trabajos realizados por los neurocientíficos Hubel y Wiesel [41] en el estudio de la corteza visual de un gato, donde identificaron que ciertas neuronas de la corteza visual primaria reaccionan a patrones simples en imágenes, como líneas verticales u oblicuas, y están ordenadas jerárquicamente permitiendo detectar objetos. Este experimento sentó la base biológica de las redes neuronales convolucionales.

Por tal razón, las *CNN* funcionan de manera apropiada en problemas de reconocimiento de imágenes, que consta de capas de neuronas interconectadas dedicadas a reconocer patrones en imágenes simulando el comportamiento de la corteza visual.

Una de las primeras aplicaciones de las redes neuronales convolucionales fue propuesta por LeCun en [42] donde implementó un sistema capaz de reconocer dígitos escritos a mano utilizando convoluciones para extraer características y una red neuronal sencilla de dos capas.

Hitos importantes en el desarrollo de las redes neuronales convolucionales se dan en el año 2012 cuando Krizhevsky *et al.* [43] implementan la primera red neuronal

convolucional en ganar en *ImageNet Challenge: Classification Task*<sup>1</sup> y en el año 2015 donde He *et al.* [44] desarrollan una arquitectura de red neuronal convolucional llamada *ResNet* que logra vencer al humano en la clasificación de imágenes.

Desde entonces las redes neuronales convolucionales han continuado desarrollándose<sup>2</sup> y mejorando gracias a la aportación de investigadores llegando a ser lo que se conocen hoy en día.

### 2.5.2 Extracción de características

El proceso de extracción de características de las *CNN* es el proceso más importante de la tecnología. Este segmento de las redes convolucionales consiste en una capa de entrada donde se pasa la imagen o el conjunto de imágenes a la red, seguidas de capas sucesivas de convolución y *pooling*, que generan como salida un vector de características.

Una capa típica de extracción de características en las *CNN* consiste en tres etapas. La primera de ella consta de varias ventanas de convolución que se ejecutan en paralelo produciendo un conjunto de activaciones. En la segunda etapa, cada una de estas activaciones pasa por una función de para evitar linealidades, por ejemplo: *ReLU*. Esta etapa también es conocida como la etapa de detección. La tercera etapa puede ser el uso de una capa de *pooling* para modificar la entrada de la siguiente capa [40]. En Ilustración 8 se puede apreciar este proceso.

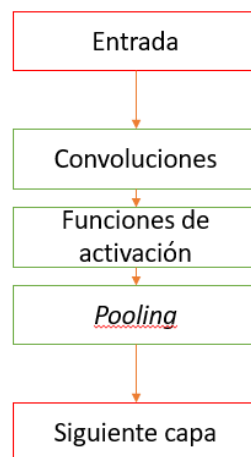


Ilustración 8. Capa común de extracción de características conformada por una capa de convolución seguida de *pooling*

Estas capas típicas de extracción de características se suelen apilar entre sí de forma secuencial dando como resultado una arquitectura formada por un conjunto de capas cuya entrada es una imagen y la salida un vector de características que posteriormente se conecta a una red neuronal para el proceso de clasificación. En la Ilustración 9 se puede

---

<sup>1</sup> El *ImageNet Challenge: Classification Task* es una competición que suele ser utilizada como punto de referencia en el rendimiento de clasificadores de imágenes. (Enlace: <http://image-net.org/> visitado por última vez el 22 de mayo de 2019)

<sup>2</sup> Los resultados de la búsqueda de las palabras “redes neuronales convolucionales” en *Scholar Google* pasaron de ser 15000 en el 2015 a ser 47600 en el año 2018

apreciar una arquitectura típica de *CNN* con 4 capas de convoluciones (líneas discontinuas negras) seguidas de capas de *pooling* que reducen el alto y ancho de la imagen.

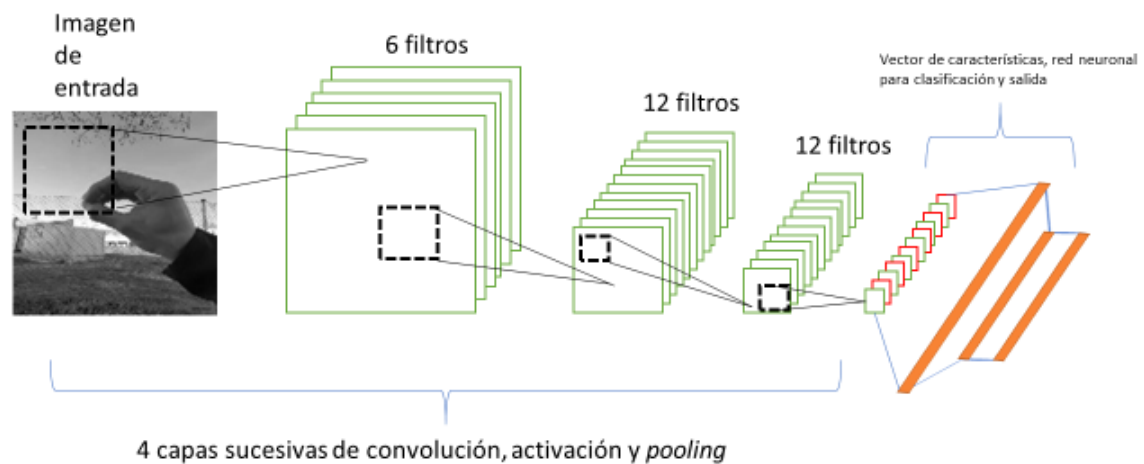


Ilustración 9. Arquitectura típica de una *CNN* basado en [45].

Cada proceso de convolución implica deslizar una ventana (líneas discontinuas negras) sobre toda la imagen correspondiente a un filtro, que es un mapa bidimensional correspondiente a una característica que se quiere identificar, pueden representar contrastes, manchas o líneas.

Entre más filtros se apliquen en una capa de extracción de características más profunda se hace la siguiente imagen. Por ejemplo, en la Ilustración 9, la primera capa de convolución se aplica sobre la imagen de entrada con la idea de identificar 6 posibles filtros, haciendo que la imagen siguiente tenga una profundidad de 6.

### 2.5.2.1 Datos de entrada

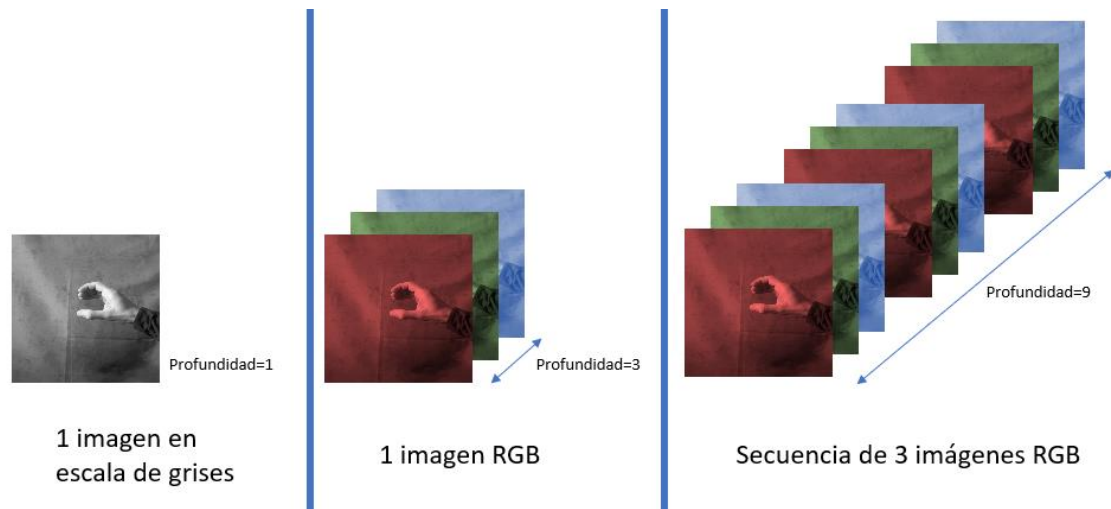
Usualmente los datos de entrada de una red neuronal convolucional son imágenes con una determinada profundidad, comúnmente imágenes con profundidad 3 debido a los tres canales *red*, *green* y *blue*. También es posible que la entrada sea una imagen de profundidad 1 si es el caso de una imagen en escala de grises o binaria.

Las redes neuronales convolucionales también pueden utilizarse para detectar características en datos volumétricos, como pueden ser datos de tomografías computarizadas [46]. En este caso se estaría hablando de redes neuronales convolucionales 3D y su única diferencia es que de entrada recibe matrices apiladas.

Debido a esta característica de las redes convolucionales, es posible apilar imágenes con la intención de clasificar gestos con características temporales, por ejemplo: 10 *frames* de un vídeo de una persona pateando un balón.

En la Ilustración 10 se pueden apreciar varios ejemplos de posibles datos de entrada para una *CNN*: la imagen de la izquierda corresponde a una imagen en escala de grises; la imagen del medio es una imagen captada por una cámara *RGB* que corresponde a tres matrices, cada una representando una imagen captada por cada canal de la cámara; en la

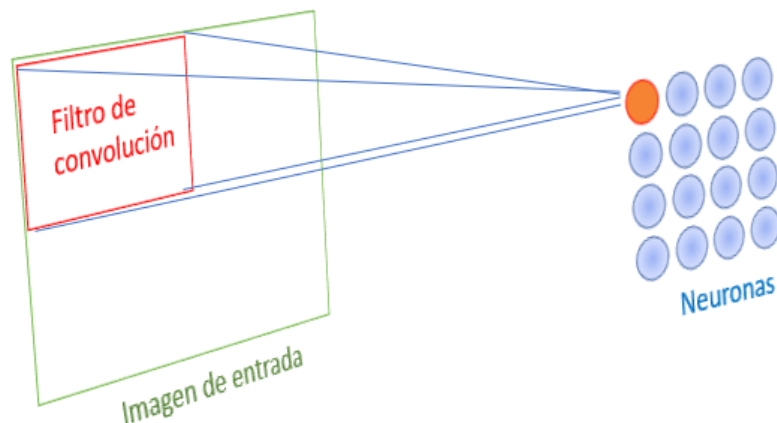
tercera imagen se puede apreciar tres imágenes *RGB* apiladas que podrían representar una secuencia de movimiento.



*Ilustración 10. Diferentes tipos de datos de entrada para una CNN y su profundidad*

### 2.5.2.2 Convoluciones

Las convoluciones son operaciones básicas de las *CNN*. En las operaciones de convolución, un filtro o *kernel*, se desliza sobre una imagen un mapa bidimensional con las secciones de la imagen en las que aparece una característica [40]. Este proceso se repite varias veces con diferentes filtros creando varios de estos mapas bidimensionales. En la Ilustración 11 se puede apreciar la aplicación de una convolución sobre una imagen de entrada. Este *kernel* posteriormente se desplaza hacia la derecha de la imagen de entrada conectando con otra neurona que se activará dependiendo de la intensidad con que se encuentra una característica.



*Ilustración 11. Operación de Convolución de una CNN*

Cada uno de estos mapas bidimensionales se apilan creando una nueva matriz tridimensional con los datos de todas las características encontradas, que posteriormente pueden ser procesados por otra capa de convolución.

La salida de una convolución es la entrada de una neurona que puede activarse o no dependiendo de la intensidad con la que aparece una característica en una porción de la imagen.

Las características que extraen estas capas de convoluciones pueden ser pequeños cambios de contraste, manchas, cambios de colores, etc. Y dependiendo del tamaño del filtro, se pueden extraer características más grandes o pequeñas. El tamaño de cada filtro de convolución puede ser visto como un hiperparámetro de una *CNN*.

### 2.5.2.3 Pooling

La idea bajo el uso de las capas de *pooling* es hacer la imagen invariante a pequeñas traslaciones en la entrada. Si la imagen de entrada se mueve un poco, después de una capa de *pooling* la salida puede mantenerse igual.

Una función de *pooling* reemplaza la salida de la red en un lugar determinado por un resumen estadístico de las salidas vecinas [40]. Por ejemplo: *mean pooling* reemplaza la salida siguiente con la media de las salidas de una vecindad rectangular.

Este proceso reduce el tamaño de la siguiente imagen reduciendo el coste computacional para procesar la siguiente capa.

Hay distintas maneras de operar en las capas de *pooling*. Una de estas formas es usar *max pooling* que reduce el valor de una región al valor máximo encontrado y otra manera es haciendo *stochastic pooling* [47], en donde se selecciona aleatoriamente un valor dentro de un grupo. En la Ilustración 12 se puede apreciar el proceso de *max pooling* 2x2 (2x2 haciendo referencia al tamaño del grupo que se va a reducir) en donde los cuatro valores de la esquina superior izquierda se reducen a su mayor local: 0.9; lo mismo que sucede con los grupos contiguos.

0.3	0.9	0.4	0.3
0.5	0.9	0.5	0.8
0.2	0.8	0.0	0.1
0.3	0.5	0.7	0.5

0.9	0.8
0.8	0.7

Ilustración 12. Operación Max Pooling 2x2 en una matriz de 4x4

El tamaño del *pooling* afecta el tamaño de la imagen reducida. Un *pooling* 2x2 reduce la imagen a la mitad de su tamaño mientras que *pooling* más grandes reducen aún más el tamaño de la imagen.

En el ámbito del reconocimiento de lengua de señas, Rao *et al.* [34] concluyeron que entre las técnicas *max pooling*, *mean pooling* y *stochastic pooling*, la última les dio mejor

resultado, haciendo que la red neuronal converja más rápido y mejorando la capacidad de generalización del clasificador.

### 2.5.3 Optimización

La salida de la capa de extracción de características conformada por convoluciones, es un vector de características que conecta con una red neuronal, también conocida como capas densas, cuyo objetivo es aprender a clasificar entre clases. El objetivo del proceso de optimización es encontrar el conjunto de valores para cada peso de cada neurona de la red neuronal que minimice una función de coste.

#### 2.5.3.1 Funciones de coste

La optimización se trata de minimizar una función de coste o *loss function*. Las funciones de coste indican qué tan malos son los errores que comete el clasificador, y existen distintas funciones de coste comunes, por ejemplo: error cuadrático y regresión logística. Estas funciones también son conocidas como criterio, función objetivo o función error [40].

Dependiendo de la tarea que se esté realizando, se tendrá que usar una u otra función de coste. Para problemas de regresión, funciones como el error absoluto medio o error cuadrado medio son más apropiadas; mientras que, para problemas de clasificación, *hinge loss* o *squared hinge loss* [48].

Una función de coste ampliamente utilizada en problemas de clasificación con aprendizaje profundo es entropía cruzada o *cross entropy* [49].

También es posible definir funciones de coste específicas según la naturaleza del problema. Por ejemplo, Deng *et al.* en [50] proponen una función de coste llamada *ArcFace* específicamente para problemas de clasificación de rostros.

#### 2.5.3.2 Algoritmos de optimización

El problema de optimización puede ser visto como encontrar la mejor combinación de valores que minimiza la función de coste [51]. Este problema de optimización puede resolverse con algoritmos de búsqueda. Estos algoritmos de búsqueda pueden dividirse en algoritmos no adaptativos o algoritmos adaptativos.

Un ejemplo de algoritmo no adaptativo es *SGD* (descenso de gradiente estocástico). Este es un algoritmo iterativo que en cada paso calcula la dirección en la que debe ajustarse los pesos para minimizar la función de coste basado en el cálculo de gradientes. El tamaño del paso que se debe dar una vez encontrada una dirección que minimiza la función de coste es llamada tasa de aprendizaje o *learning rate* [52].

Un *learning rate* muy alto tendrá dificultades para optimizar el problema y un *learning rate* muy bajo puede estancarse en un mínimo local. En los métodos no adaptativos el valor del *learning rate* se mantiene fijo a medida que el algoritmo se aproxima a un mínimo. Por el contrario, en los métodos adaptativos, se intenta ir ajustando el valor del *learning rate* a medida se avanza en el problema.

Uno algoritmo adaptativo es *AdaGrad* (gradiente adaptativa) que permite encontrar un valor para el *learning rate* apropiado con el paso de las iteraciones [53]. También existen alternativas como *ADAM*, *AdaDelta* y *RMSProp*.

A. Wilson *et al.* en [54] realizaron una investigación sobre el rendimiento de estos algoritmos y concluyeron que, a pesar de la gran popularidad de algoritmos como *ADAM*, *SGD* puede llegar a mejores tasas de acierto que los demás algoritmos adaptativos, por lo que sugieren el uso de *SGD* sobre los demás métodos.

#### 2.5.4 Regularización

Los métodos de regularización evitan que una red neuronal se sobreajuste al conjunto de datos de entrenamiento. Algunos de estos métodos son regularización L1, regularización L2 y *dropout*, propuesto por Sivastra *et al.* en [55].

L1 y L2 evitan que los pesos de la red neuronal se incrementen demasiado en una dirección [51]. Por el contrario, *dropout* es un mecanismo que funciona omitiendo algunas neuronas aleatoriamente según cierta probabilidad durante el proceso de entrenamiento. En Ilustración 13 izquierda, se aprecia una red neuronal sin pasar por el proceso de *dropout* y en la Ilustración 13 derecha, se puede observar la misma red neuronal después de sufrir un proceso de *dropout* del 50% de las neuronas en cada capa.

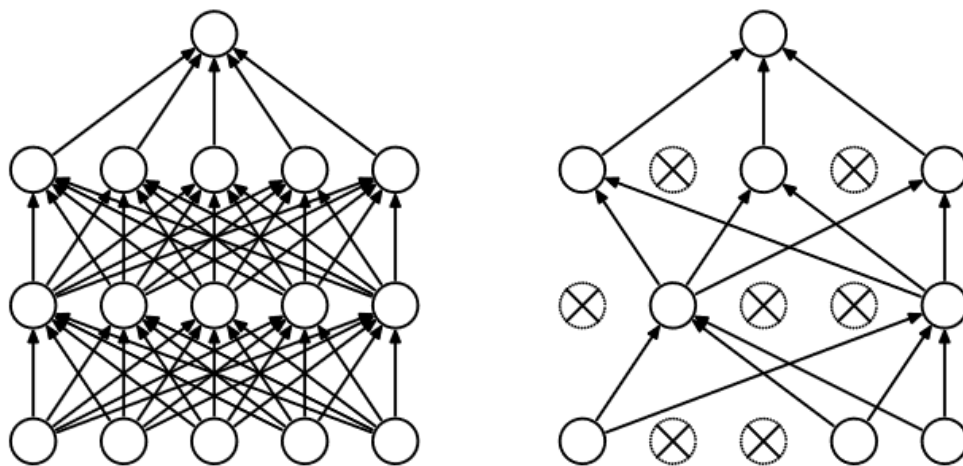


Ilustración 13. Red neuronal estándar y red neuronal después de aplicar dropout (fuente [55]).

Otro método de regularización es detención temprana o *early stopping*. Con *early stopping* logramos detener el proceso de entrenamiento de la red neuronal justo en el mejor modelo encontrado de acuerdo con las métricas que hayamos definido. Por ejemplo: detener el proceso de entrenamiento cuando la tasa de acierto en validación disminuya. *Early stopping* es un método fácil de implementar para evitar sobreajuste [56].

*Stochastic pooling* también puede ser considerado un método de regularización como lo proponen Zeiler y Fergus en [47] para mejorar la capacidad de generalización del clasificador.

Autores como Hernández-García y König en [57] han analizado los beneficios de utilizar técnicas de aumento de datos para mejorar la capacidad de generalización del modelo concluyendo que el uso de técnicas de aumento de datos por sí solas pueden alcanzar el

mismo rendimiento que otras técnicas de regularización como *dropout*. El proceso de aumento de datos se basa en añadir ligeras variaciones en las imágenes de entrada permitiendo la inclusión de nuevos datos.

Estas técnicas pueden usarse en conjunto para mejorar el rendimiento del modelo. Por ejemplo: Krizhevsky *et al.* en [43] han utilizado *dropout* en combinación con técnicas de aumento de datos para evitar sobreajuste en su *CNN*.

### 2.5.5 Funciones de activación

Es posible definir una variedad de funciones de activación distintas en cada neurona de la red neuronal. Las funciones de activación también son conocidas como funciones de transferencia [58]. Estas funciones permiten activar o no una neurona dependiendo de la intensidad del estímulo recibido.

Entre las funciones de activación más populares están la función sigmoideal, la función lineal; la función *softmax*; y funciones no lineales como *ReLU* (unidad lineal rectificadora) y sus variantes, y *ELU* (unidad lineal exponencial).

La decisión de utilizar una u otra depende en parte de la naturaleza del problema, por ejemplo: la función sigmoideal es apropiada para problemas de clasificación binaria; mientras que, la función lineal es útil como capa de salida en problemas de regresión; por otro lado, la función *softmax*, es la adecuada como capa de salida en problemas de clasificación multi-clase [51].

Actualmente, la mayoría de los problemas de aprendizaje profundo se abordan con la función de activación *ReLU* y la función *softmax* en la capa de salida [58]. Arquitecturas como *AlexNet* [43] y *ResNet* [44] utilizan estas funciones.

A pesar de la popularidad de la función *ReLU*, existen otras funciones no lineales como *LReLU* (*Leaky ReLU*), *PReLU* (*Parametric ReLU*), *SELU* (*Scaled ELU*)

Pedamonti en [59], ha realizado una comparación del rendimiento de diferentes funciones de activación no lineales en un problema de clasificación de imágenes, concluyendo que la función *ELU* puede llegar a tener mejor rendimiento que las funciones *ReLU*, *LReLU* y *PReLU*, además de que el tiempo de aprendizaje suele ser mejor que en las demás funciones de activación.

### 2.5.6 Rendimiento

El rendimiento de las redes neuronales convolucionales está muy ligadas a una selección del tamaño del *dataset* correcta. Sin embargo, como lo considera Kelly en [60], es una decisión que depende en mucho del problema en cuestión y que, para problemas de clasificación como el *ImageNet Challenge*, se necesitarían alrededor de 1000 imágenes por clase. Temas como las variaciones entre las imágenes también es un aspecto importante a tomar en cuenta, considera Al-zuhari en [60].

Goodfellow en [40] sugiere que una manera de decidir cuántos datos adicionales recolectar es empezar a entrenar el modelo con un conjunto de datos pequeño e ir



doblando su tamaño en sucesivos entrenamientos. Esta relación nos permite identificar el nivel de mejora de la *CNN* respecto al número de ejemplos en el *dataset*.

Al desarrollar una *CNN*, si esta se diseña adecuadamente y se cuenta con un *dataset* apropiado, es posible esperar tasas de acierto muy altas, como es el caso de [44] que logró superar la tasa de error del humano en un *dataset* en el *ImageNet Challenge: Classification Task* [61].

En cuanto a tiempo de entrenamiento, es posible reducirlo con una eficiente paralelización del trabajo en ordenadores de altas prestaciones (o *HPC* por sus siglas en inglés) debido a la naturaleza paralela de las *CNN*, cualidad que les permite ser representadas como operaciones matemáticas entre matrices. Librerías para el desarrollo de *CNN* como *Keras* [62], utilizan *CUDA* [63] para implementar el proceso de entrenamiento en *GPU*.

### 2.5.7 Desventajas

Como cualquier clasificador de imágenes, las redes neuronales convolucionales se enfrentan a algunos problemas que hay que tomar en cuenta antes de desarrollar sistemas de reconocimiento de imágenes.

Uno de estos problemas es la oclusión de objetos. En el caso del sistema propuesto, una mano puede ocluir la otra y ocasionar errores. Las variaciones de punto de vista o ángulos, es un problema usual en estos sistemas, sin embargo, se resuelven en este trabajo manteniendo una posición fija para las cámaras.

Cambios en las condiciones de iluminación afectan la precisión del modelo. Imágenes en contraluz o con baja iluminación no serán bien clasificadas.

Las deformaciones en los objetos tienden a confundir a los clasificadores y en el caso del sistema propuesto, las manos pueden tratarse como objetos deformables. Está muy relacionado a la configuración de las manos donde cambios extremos en la ejecución de una señal generaría confusiones en el sistema.

Los fondos desordenados y confusos también empeoran el rendimiento de los clasificadores. En caso del sistema propuesto, los fondos son muy variantes y podrían afectar el resultado del modelo. Por ejemplo: si de fondo se encontrase otra mano a parte de la mano del usuario.

Un último inconveniente que se puede encontrar son las variaciones de escala. La mano más cerca o más lejos cambia de tamaño. Sin embargo, en el trabajo realizado, es un problema que puede ser fácilmente tratado ya que la mano no puede desprenderse del cuerpo en un sistema de visión en primera persona.

### 2.5.8 Metodología de desarrollo

Goodfellow [40], propone una metodología práctica para el desarrollo de proyectos de aprendizaje profundo.

La metodología se puede resumir en tres pasos que son:

1. Determinar los objetivos en cuanto a métricas de rendimiento.

2. Seleccionar un modelo de trabajo incluyendo la estimación del rendimiento alcanzable.
3. Configurar el sistema y hacer cambios graduales basado en los hallazgos en los entrenamientos.

El experto en aprendizaje automático tendrá que realizar varias tareas que pueden ser recolectar más datos, decidir una arquitectura, definir una medida de rendimiento, etc.

En cuanto a determinar el objetivo del sistema, un ejemplo de ello puede ser proponerse alcanzar tasas de acierto del 95%. La selección de estos criterios puede basarse en referencia a otros trabajos similares realizados.

El segundo paso implica analizar el problema y verificar si puede resolverse con técnicas de *deep learning*. Si el problema es apropiado para redes neuronales convolucionales, seleccionar *ReLU* como función de activación y gradiente descendiente estocástica como algoritmo de optimización puede ser un buen comienzo.

Otro aspecto para verificar en el segundo paso es la inclusión de técnicas de regularización. A no ser que el conjunto de datos contenga millones de ejemplos o más, es necesario empezar el proceso de entrenamiento utilizando alguna técnica de regularización, por ejemplo: *batch normalization* o *dropout*.

El siguiente paso implica entrenar el modelo y verificarlo con el *dataset* de pruebas. Si el rendimiento es aceptable el trabajo está terminado, de otro modo es necesario realizar varias ejecuciones de entrenamiento hasta alcanzar el objetivo variando los hiperparámetros de la red neuronal convolucional. Entre los cambios que se pueden realizar están: cambiar la intensidad de regularización, añadir más capas de convoluciones, variar los algoritmos de optimización, entre otros.

Si es complicado alcanzar el objetivo deseado, la manera más efectiva de mejorar el rendimiento del modelo es aumentar el tamaño del conjunto de datos, de no ser así, la única forma de mejorarlo es ajustar el algoritmo.

### 2.5.9 Conclusiones

El análisis de distintos trabajos en el estado del arte permite concluir que es posible la utilización de guantes de colores para facilitar el proceso de segmentación de la mano, sin embargo, su uso puede descartarse si se utilizan *CNN* como algoritmo de clasificación.

Las cámaras infrarrojas o estereoscópicas serían buena opción debido a su capacidad de estimar profundidades, aunque las cámaras *RGB* son una solución aceptable cuando las imágenes son tratadas con una *CNN*.

Utilizar *CNN* es una alternativa apropiada para procesar imágenes que requiere de poco preprocesado y es posible aspirar a tasas de acierto bastante altas a diferencia de otros algoritmos de clasificación.

Otras consideraciones que surgen del análisis del estado del arte con respecto al diseño de una *CNN* son las siguientes:

- Reutilizar una arquitectura como la propuesta por Rao *et al.* en [34] sirve de primera aproximación en la definición de arquitectura de la *CNN*, que posteriormente se puede ajustar hasta alcanzar el rendimiento deseado.
- El análisis de la literatura sugiere que utilizar técnicas de aumento de datos y *dropout* es una alternativa para mejorar la capacidad de generalización de los clasificadores.
- Y una opción que puede mejorar la capacidad de generalización del clasificador es *stochastic pooling*, sin embargo, el uso de *max pooling* ha sido utilizado mayormente.
- Como función de activación para la *CNN* es posible utilizar *ReLU* como primera opción e ir probando otras funciones posteriormente.
- La función de coste apropiada para el problema es *categorical cross entropy* y función de salida *softmax*.
- Se podría utilizar *ADAM* como algoritmo de optimización, sin embargo, el estado del arte sugiere que el empleo de *SGD* puede llevar a mejores resultados.

## Capítulo 3. ANÁLISIS Y DISEÑO

En este capítulo se presentan las decisiones de diseño del sistema con respecto al uso de cámaras y selección del algoritmo de clasificación. Se presenta y detalla el diagrama de componentes de la solución conceptual que resuelve la problemática en cuestión. Y se presenta un diagrama de componentes y procesos realizado en la elaboración de este trabajo.

### 3.1 ELECCIÓN DE DISEÑO

En este apartado se revisan algunos aspectos que influyeron en la decisión de diseño, como el uso de dos cámaras para resolver ambigüedades, tamaño de imagen de entrada y salida del modelo.

#### 3.1.1 Sobre el uso de cámaras

Se seleccionan 24 señas del diccionario dactilológico panameño. De este conjunto hay ciertas señas que vistas desde una perspectiva lucen iguales y necesitan otra perspectiva para apreciarse diferencias. Por ejemplo, las señas para las letras K e Y son prácticamente iguales con una pequeña diferencia en la posición del dedo pulgar y lucen muy similares si son observadas desde una cámara colocada en el pecho de un usuario, pero tienen diferencias significativas si son observadas desde arriba (Ilustración 14). Por esta razón, se generan dos modelos para dos cámaras que portará el usuario, cada una con una perspectiva distinta.



Ilustración 14. Señas para letras K e Y observadas desde distintas perspectivas

Se identificaron dos ubicaciones para las cámaras, una en el pecho y otra en la cabeza del usuario. El resultado es un sistema portable que cuenta con dos cámaras: una cámara inferior, ubicada en el pecho del usuario; y otra cámara superior, montada en la cabeza.

Para seleccionar las imágenes que se tratarían con un modelo u otro, se revisaron todas las señas captadas desde la cámara ubicada en el pecho, se separaron las que no tienen diferencias significativas desde esta perspectiva y se asignaron al modelo para la cámara superior. Con esto, se logra que las señas que son difíciles de diferenciar desde una perspectiva sean interpretadas desde otra, y viceversa.

### **3.1.2 Sobre el algoritmo de clasificación**

Se revisa la decisión de utilizar redes neuronales convolucionales frente a otros clasificadores, y se analizan algunos aspectos de diseño del clasificador que se tienen que establecer antes de trabajar en la arquitectura de la red neuronal convolucional y se mantiene fijos a lo largo del proceso de entrenamiento. Uno de estos aspectos es el tamaño de imagen de entrada, que afecta el rendimiento del modelo y la función de salida del clasificador.

#### **3.1.2.1 CNN sobre otros algoritmos de clasificación**

Se ha decidido utilizar las redes neuronales convolucionales para los modelos porque tienen una serie de ventajas convenientes sobre otros algoritmos de clasificación:

- Es la tecnología que mejores resultados ha dado en las últimas investigaciones en visión artificial.
- Se reduce el tiempo invertido en ingeniería de característica ya que este proceso es llevado a cabo por las capas de convoluciones.
- Se hace menos necesario el preprocesado de datos ya que se puede trabajar con la imagen cruda.
- Es posible aspirar a tasas de acierto más altas que otros algoritmos de clasificación.

Con estas ventajas, las redes neuronales convolucionales se convierten en la mejor opción para abordar el problema.

#### **3.1.2.2 Tamaño de imagen**

Un aspecto relevante antes de desarrollar un clasificador para imágenes es decidir un tamaño de imagen adecuado. Si el tamaño de la imagen es muy grande, implicaría tiempos de respuesta más lentos y un proceso de aprendizaje más costoso, con imágenes muy pequeñas es posible perder información relevante.

Las investigaciones actuales tienden a utilizar imágenes que no sobrepasan los 256x256 píxeles (la arquitectura *ResNet* utiliza imágenes de 224x224 píxeles [44] siendo el tamaño más grande encontrado en la revisión bibliográfica). Con imágenes de este tamaño es posible distinguir a simple vista cada una de las clases para los modelos, sin embargo, se ha decidido que un tamaño apropiado para las imágenes es 125x125 píxeles, tamaño que

aún permite diferenciar las clases entre sí y se mantiene dentro de lo normalmente utilizado en investigaciones recientes.

### **3.1.2.3 Función de salida**

Como el sistema propuesto es una combinación de dos clasificadores multiclases, uno para 17 clases y otro para unas 7 clases, es conveniente crear modelos que generen como salida la probabilidad con que la que se predice una clase, y la función de activación apropiada para trabajar con este tipo de clasificadores es *softmax*. Función que se utiliza como salida de ambos modelos.

### **3.1.2.4 Primera aproximación de arquitectura para la CNN**

Para la selección de una arquitectura inicial para la *CNN*, se ha empezado por combinar cuatro capas de convolución y dos de *pooling*, dos de convolución seguida de una de *pooling* basado en los trabajos de [34] como una aproximación inicial.

La técnica de *pooling* decidida ha sido *max pooling* debido a su popularidad en investigaciones recientes. Esta técnica de *pooling* se ha mantenido fija durante todo el proceso de ajuste del modelo.

El estado del arte recomienda utilizar *ReLU* como función de activación y por tal razón ha sido la elección inicial para abordar el problema.

Como técnica de regularización, se ha utilizado *dropout* en la capa de entrada de las capas densas y entre la capa oculta. Además, se ha combinado este método con técnicas de aumento de datos según lo recomendado en el estado del arte. La intensidad en la que se ha utilizado *dropout* se ha ajustado según se iba evaluando modelos.

Esta elección de arquitectura ha servido como un esfuerzo inicial en la configuración de las *CNN*.

## **3.2 DISEÑO CONCEPTUAL GENERAL DE LA SOLUCIÓN**

Se ha realizado un análisis de los posibles componentes de un sistema que solucione la problemática abordada con el fin de proponer un sistema y desarrollar un clasificador para el prototipo.

El siguiente diagrama de componentes corresponde al diseño de un prototipo que tiene la intención de resolver la comunicación de personas con discapacidad auditiva utilizando cámaras en primera persona (Ilustración 15).

En este trabajo de fin de máster se desarrolla el subsistema de Visión Artificial y los componentes Clasificador Cámara Superior y Clasificador Cámara Inferior (componentes en recuadros rojos de la Ilustración 15).

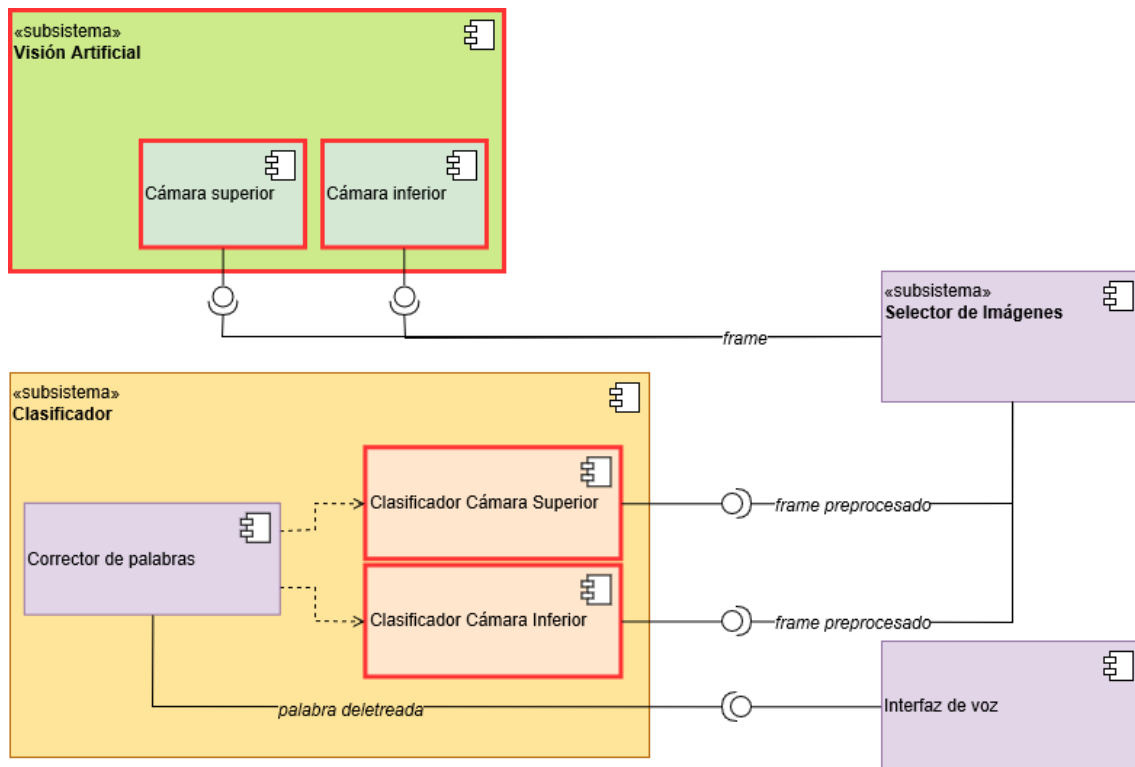


Ilustración 15. Diagrama de Componentes del prototipo de solución propuesta

El subsistema de Visión Artificial está compuesto dos cámaras y su objetivo es capturar imágenes en tiempo real. El desarrollo de este subsistema es parte del trabajo realizado en este trabajo de fin de máster en la elaboración de un *dataset* para los clasificadores.

El subsistema Selector de imagen tendría varias tareas a realizar. La primera sería detectar si el usuario está ejecutando una señal o no. La segunda tarea sería identificar si se ha ejecutado una o varias señales y separarlas. La tercera tarea: seleccionar las imágenes y pasarlas al clasificador.

Este subsistema podría estar compuesto por un clasificador binario que indique en si se está ejecutando una señal o no, un algoritmo que identifique en qué instante de tiempo se ha ejecutado una señal y un algoritmo para ajustar la imagen a la entrada del subsistema Clasificador.

El subsistema Clasificador tiene por objetivo interpretar una sucesión de letras ejecutadas e identificar la palabra que quiere comunicar el usuario. Este subsistema está compuesto por dos clasificadores, uno para cada cámara, que tendrán la tarea de identificar una cantidad de letras; y un corrector de palabras, cuyo objetivo sería generar la palabra que quiere comunicar el usuario.

La interfaz de voz funcionaría como salida del sistema y permitiría reproducir la palabra identificada en un altavoz.

### 3.3 DISEÑO GENERAL DEL SISTEMA DESARROLLADO

En este trabajo se desarrollan determinados componentes de la solución conceptual propuesta (componentes en rojo Ilustración 15). Desarrollar estos componentes implica generar elementos adicionales como una base de datos de imágenes o *dataset*, un algoritmo de clasificación y algoritmos auxiliares para crear el *dataset*. El sistema desarrollado en este trabajo está dividido en componentes que se pueden apreciar en la Ilustración 16 (los elementos en recuadros rojos corresponden a los mismos componentes en la Ilustración 15).

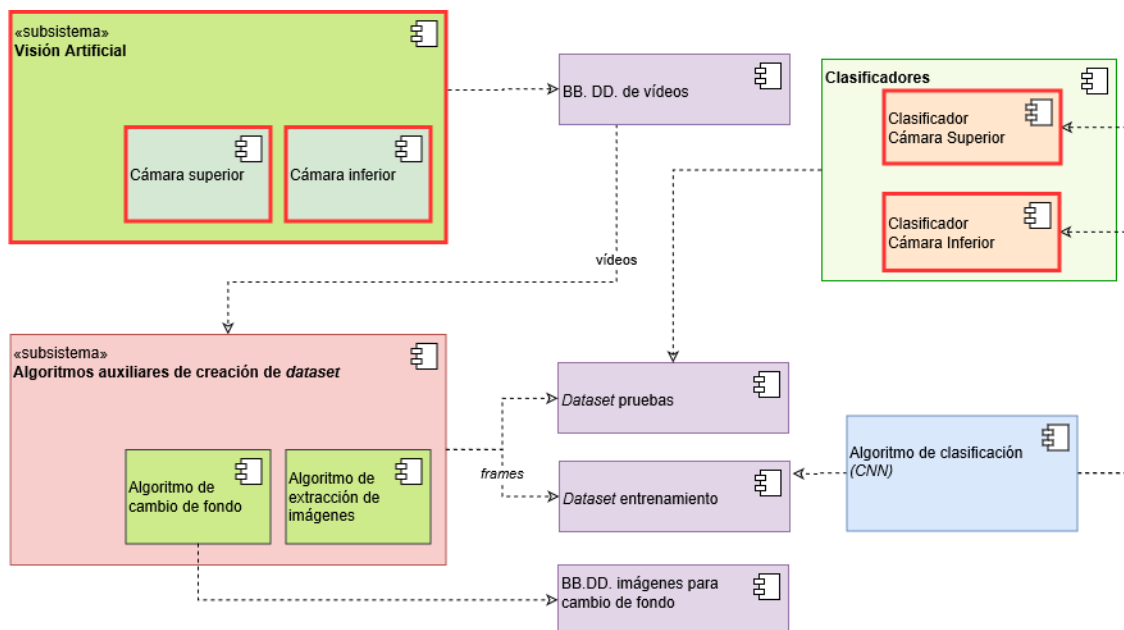


Ilustración 16. Diagrama de componentes sistema desarrollado

El subsistema de Visión Artificial está compuesto por dos cámaras, como en la solución conceptual propuesta. Se desarrollan dos algoritmos para automatizar el proceso de extracción de imágenes de los vídeos, ambos subcomponentes en verde del subsistema Algoritmos auxiliares de creación de *dataset*. Los componentes en color violeta de la ilustración corresponden a las bases de datos generadas para el desarrollo del sistema que contienen: vídeos, imágenes para cambio de fondo, imágenes de entrenamiento y pruebas. El Algoritmo de clasificación (componente de color celeste) es utilizado para generar los clasificadores (componentes en anaranjado) del sistema desarrollado.

El proceso de interacción entre los componentes se puede apreciar mejor en la Ilustración 17 en el que se visualiza las etapas seguidas en el desarrollo del sistema.



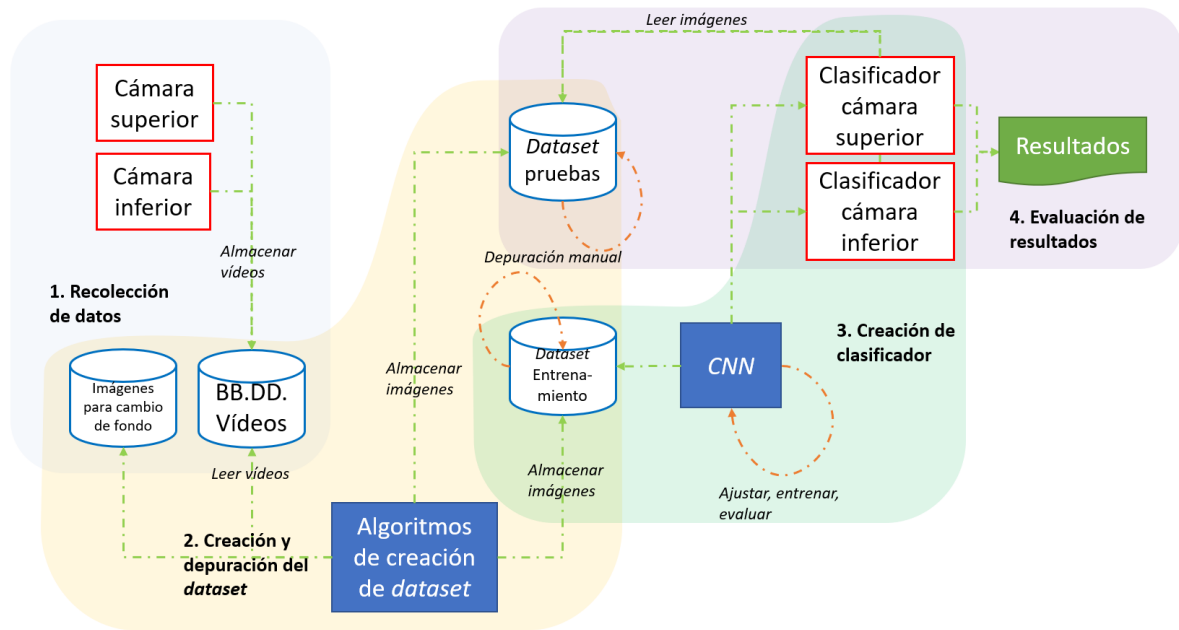


Ilustración 17. Etapas del desarrollo del sistema (ciclo de vida)

En la primera etapa (1. Recolección de datos), se utilizan las cámaras para recolectar videos de las señas ejecutadas por los usuarios y almacenarlos en una base de datos, además se recolectan las imágenes que formaran parte del proceso de cambio de fondo como aumento de datos. La segunda etapa (2. Creación y depuración del *dataset*) es en donde se utilizan los algoritmos desarrollados para la creación del *dataset*. La creación inicia con la extracción de imágenes con los algoritmos auxiliares y termina con un proceso manual de depuración sobre los *datasets* de pruebas y entrenamiento, en donde se eliminan imágenes con manchas de contraluz, borrosas e inútiles.

La siguiente etapa (3. Creación del clasificador) es un proceso iterativo de ajuste de hiperparámetros, entrenamiento y evaluación cuyo objetivo es encontrar la mejor arquitectura de *CNN* para el conjunto de datos. Esta etapa implica la interacción del *dataset* de entrenamiento con la *CNN* para generar los clasificadores.

Finalmente, una vez concluida la etapa 3, se almacenan los clasificadores y se evalúan con el *dataset* de pruebas (4. Evaluación de resultados).

El proceso descrito en la Ilustración 17 también hace referencia al ciclo de vida del sistema en este trabajo, que inicia con una etapa de recolección de datos y finaliza con la evaluación de los clasificadores generados.

### 3.4 REQUISITOS DEL SISTEMA

La siguiente recopilación de requisitos hace referencia a los requisitos necesarios para el desarrollo del trabajo de fin de máster, que contempla el desarrollo del subsistema de Visión Artificial, los *datasets*, y los componentes Clasificador Cámara Superior y Clasificador Cámara Inferior descritos en el apartado anterior.

Los requisitos del proyecto se dividen en requisitos de *dataset*, requisitos del clasificador, requisitos del componente de visión, requisitos de la base de datos de vídeos y requisitos de los algoritmos. Se clasifican según los siguientes niveles de prioridad y necesidad (Tabla 1).

Tabla 1. Niveles de prioridad y necesidad de los requisitos

Niveles de prioridad		
baja	media	alta
Niveles de necesidad		
opcional	deseable	esencial

La siguiente tabla (Tabla 2) contiene la descripción de los requisitos del proyecto y su clasificación según prioridad y necesidad.

Tabla 2. Lista y descripción de requisitos

Identificador	Descripción	Prioridad	Necesidad
	<b>Datasets (BB.DD. de imágenes)</b>		
RQ.1	Las imágenes deben tener una resolución de 125x125 píxeles	media	deseable
RQ.2	Las imágenes deben estar separadas en carpetas por clases	media	esencial
RQ.3	Las imágenes deben estar en formato .JPG	media	deseable
RQ.4	<i>Dataset</i> de entrenamiento debe tener al menos 1000 imágenes por clase	alta	esencial
RQ.5	<i>Dataset</i> de entrenamiento debe contener imágenes con condiciones de luz distintas	alta	esencial
RQ.6	<i>Dataset</i> de entrenamiento debe contener imágenes con escenarios de fondo distintos	alta	esencial
RQ.7	<i>Dataset</i> de entrenamiento debe contar con la participación de al menos 3 usuarios	alta	esencial
RQ.8	<i>Dataset</i> de Pruebas 1 debe ser recolectado por un usuario que haya participado en el <i>dataset</i> de entrenamiento	media	deseable
RQ.9	<i>Dataset</i> de Pruebas 1 debe contener al menos 15 imágenes por clase	alta	deseable
RQ.10	<i>Dataset</i> de Pruebas 2 debe ser recolectado por un usuario que no haya participado en el <i>dataset</i> de entrenamiento	alta	esencial
RQ.11	<i>Dataset</i> de Pruebas 2 debe reflejar el uso del sistema en un entorno real	alta	esencial
RQ.12	<i>Dataset</i> de Pruebas 2 debe contener al menos 15 imágenes por clase	alta	deseable

	<b>Clasificador</b>		
RQ.13	Se debe emplear el uso de <i>CNN</i> como clasificador	media	deseable
RQ.14	La tasa de acierto media del clasificador debe superar el 75% en la Prueba 1 y Prueba 2	baja	opcional
RQ.15	El clasificador de cámara superior debe identificar las letras A, K, M, N, S, V, Y	media	opcional
RQ.16	El clasificador de cámara inferior debe identificar las letras B, C, D, E, F, G, H, I, L, O, P, Q, R, T, U, W, X	media	opcional
	<b>Componente de visión</b>		
RQ.17	La cámara superior debe colocarse en la cabeza del usuario	alta	esencial
RQ.18	La cámara inferior debe estar colocada en el pecho del usuario	alta	esencial
	<b>BB.DD. de vídeos</b>		
RQ.19	Los vídeos deben ser almacenados en resolución 640x480 píxeles	media	deseable
RQ.20	El formato de vídeos debe ser <i>.MP4</i>	media	deseable
RQ.21	La velocidad de vídeo debe ser 60 <i>fps</i>	media	opcional
	<b>Algoritmo cambio de fondo</b>		
RQ.22	El programa debe permitir elegir la carpeta con imágenes para el fondo	alta	esencial
RQ.23	El programa debe cambiar el fondo de cada <i>frame</i> del vídeo por otro dentro de la carpeta de imágenes de fondo	alta	deseable
RQ.24	El programa debe almacenar todas las imágenes resultantes en una carpeta	alta	opcional
RQ.25	Las imágenes resultantes deben tener una resolución de 125x125 píxeles	alta	esencial
RQ.26	El formato de las imágenes resultantes debe ser <i>.JPG</i>	media	deseable
RQ.27	El programa debe visualizar una barra de progreso	baja	opcional

RQ.28	El programa debe permitir configurar la velocidad del vídeo	baja	opcional
	<i>Algoritmo de extracción de imágenes</i>		
RQ.29	El programa debe permitir elegir la carpeta con imágenes para el fondo	alta	esencial
RQ.30	El programa debe almacenar todas las imágenes en una carpeta	alta	opcional
RQ.31	Las imágenes resultantes deben tener una resolución de 125x125 píxeles	alta	esencial
RQ.32	El formato de las imágenes resultantes debe ser <i>.JPG</i>	media	deseable
RQ.33	El programa debe visualizar una barra de progreso	baja	opcional
RQ.34	El programa debe permitir configurar la velocidad del vídeo	baja	opcional

## Capítulo 4. CREACIÓN DEL *DATASET*

En este capítulo se aborda el proceso seguido para la recolección de datos, los escenarios utilizados, las condiciones de iluminación, el número de usuarios que participaron y características de las imágenes. También se mencionan las técnicas utilizadas para aumentar el tamaño del *dataset*, la selección de los datos de entrenamiento y validación, y los *datasets* utilizados para pruebas.

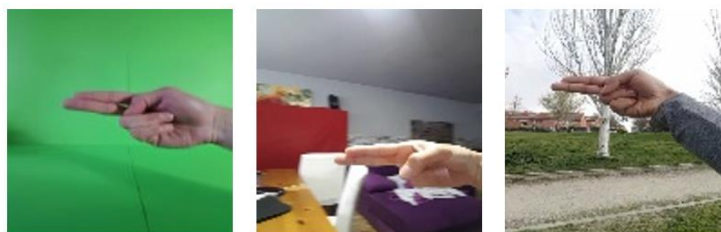
### 4.1 RECOLECCIÓN DE DATOS

Los datos se han recopilado como vídeos a resolución de 640x480 píxeles mientras los usuarios realizaban cada seña. Cada uno de los usuarios generó varios vídeos de los cuales posteriormente se les extrajo los *frames* escalados a una resolución de 125x125 píxeles con la ayuda de programas desarrollados para este fin. Finalmente, estos *frames* se han separados en categoría, una por cada letra.

#### 4.1.1 Usuarios y entornos

Para el proceso de recolección de datos se entrenaron cuatro usuarios. Tres de ellos generaron el *dataset* que se utilizó para entrenamiento y validación, y los datos del cuarto usuario se utilizaron para pruebas sobre el modelo.

Se seleccionaron tres entornos distintos para la recolección de datos. Dos de estos entornos son el interior de una casa y un parque, que nos permitió recolectar imágenes en diferentes condiciones de iluminación y de paisaje de fondo. Y el tercer entorno consta de un fondo de color uniforme, que posteriormente se modificó artificialmente para generar fondos distintos con el objetivo de complicar el proceso de aprendizaje de las redes convolucionales. En la siguiente ilustración se puede apreciar ejemplos de estos tres entornos.



*Ilustración 18. Seña correspondiente a la letra H sobre un fondo verde, salón y un parque realizada por tres usuarios distintos.*

#### 4.1.2 Procesos de recolección de datos

La recolección de datos se dividió en tres partes. Una parte corresponde a los datos recolectados en el parque y en interiores por tres usuarios. Una segunda corresponde a los datos recolectados sobre fondo uniforme por dos usuarios. Y una última etapa son los datos recolectados por un cuarto usuario utilizados para pruebas.

#### ***4.1.2.1 Datos recolectados en el parque e interiores***

Para recolectar imágenes en el parque y en interiores se les pidió a los usuarios que se pusieran ambas cámaras, que realizaran cada una de las señas por 10 segundos y que caminaran o giraran alrededor suyo. El proceso de moverse en el entorno permitía conseguir imágenes lo suficientemente distintas, variando la proyección de sombras y el fondo de la imagen. La siguiente ilustración muestra una secuencia de cinco imágenes generadas por un usuario en el parque.

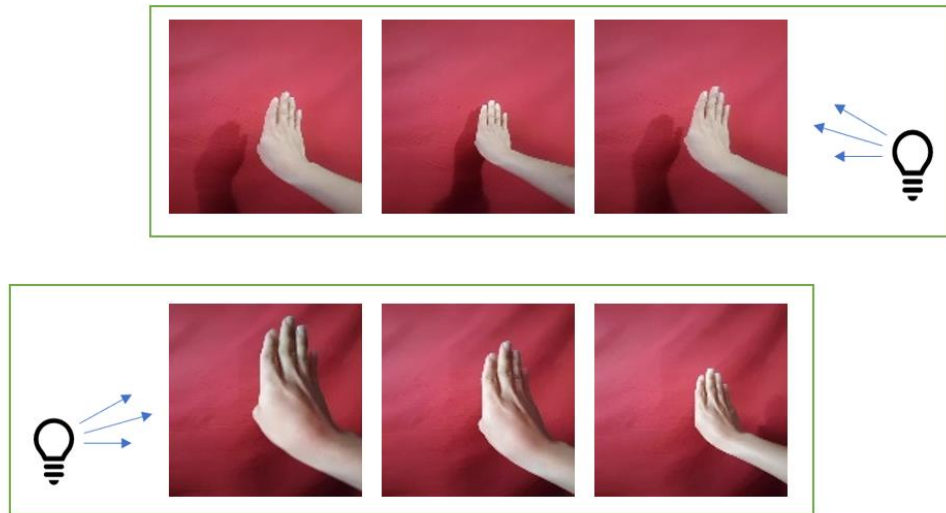


*Ilustración 19. Seña correspondiente a la letra O recolectada en un parque y un salón con variaciones en el fondo y proyección de sombras.*

#### ***4.1.2.2 Datos recolectados con fondo uniforme***

Con respecto a la recolección de datos sobre fondo uniforme, se les pidió a los usuarios que se colocaran frente a una pantalla con fondo liso, que realizaran cada una de las señas por 10 segundos mientras realizaban ligeros movimientos. Este proceso se repitió dos veces por dos usuarios, cambiando la proyección de luz sobre la mano.

En la primera etapa de recolección se proyectó una luz blanca desde la derecha del usuario en dirección izquierda, y en la segunda etapa se proyectó en el sentido inverso. Este proceso nos permitió obtener imágenes con un perfil distinto, como se muestra en la Ilustración 20, que posteriormente sufrieron una modificación en su fondo.



*Ilustración 20. Seña de letra B generada por un usuario sobre fondo uniforme con dos proyecciones de luz distintas.*

#### **4.1.2.3 Datos recolectados para pruebas**

Se prepararon dos conjuntos de datos para pruebas. Uno de estos se recolectó con uno de los usuarios que aportaron en la creación del *dataset* de entrenamiento, el otro conjunto de datos se recolectó con ayuda de un cuarto usuario, con el objetivo de evaluar la capacidad de generalización del modelo.

## **4.2 PROCESO DE AUMENTO DE DATOS**

Se realizaron dos tipos de transformaciones sobre los datos con el objetivo de aumentar el tamaño del *dataset*. Una de estas transformaciones es la agregación de fondos nuevos, y el otro grupo de transformaciones corresponden a cambios en la iluminación, rotaciones y acercamientos.

### **4.2.1 Agregación de fondo**

Este proceso consiste en añadirle un fondo nuevo a una imagen sin alterar el contorno de la mano del usuario. Estas transformaciones se aplicaron únicamente a los datos que fueron recolectados con fondo uniforme y es prácticamente el mismo proceso que se realiza para cambiar los ambientes en las películas.

Para la agregación de fondos se creó un programa que automatiza este proceso. Básicamente, el programa toma una imagen con fondo uniforme, se convierte en binaria la imagen separando la mano del resto, realiza operaciones con esta imagen binaria y una imagen que formará parte del fondo, y el resultado es una imagen con un fondo distinto. En la siguiente ilustración se puede apreciar el resultado de este proceso.





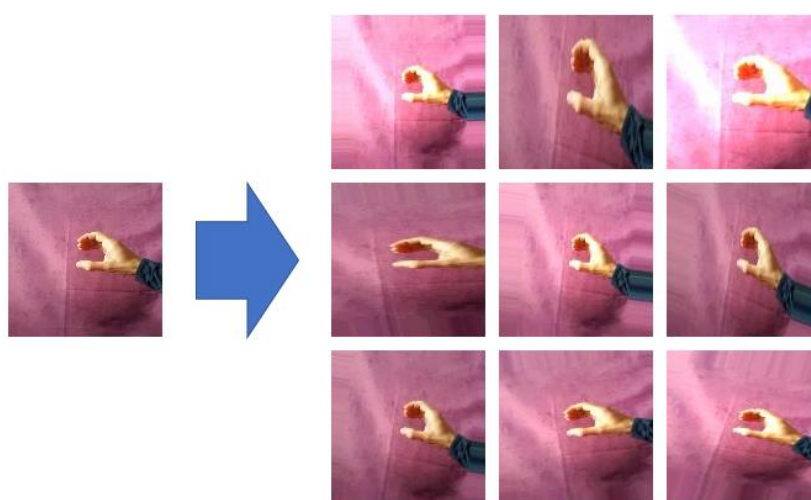
*Ilustración 21. Imágenes con fondos nuevos.*

El conjunto de imágenes que se utilizó como fondos nuevos son un grupo de 500 fotos. Entre estas fotos se seleccionaron imágenes en donde pueden apreciarse personas, monumentos, montañas, carreteras, árboles, parques con nieve, bicicletas, y demás., intentando recopilar imágenes distintas que pudieran simular fondos nuevos.

#### 4.2.2 Otras transformaciones

Además de cambiar el fondo de las imágenes artificialmente, también se realizaron otras transformaciones sobre las imágenes durante el proceso de entrenamiento. Estas transformaciones fueron ligeras rotaciones, cambios en la iluminación, y escalado.

Se eligieron transformaciones que preservaran la integridad de las etiquetas controlando la medida de ángulos a rotar, la variación de intensidad de luz y el tamaño del *zoom* añadido. El resultado es una imagen con pequeñas variaciones seleccionadas aleatoriamente en su rotación, intensidad de luz y escalado (Ilustración 22).



*Ilustración 22. Variaciones en rotación, intensidad de luz y escalado en una imagen.*



### 4.3 DETALLES ADICIONALES DEL DATASET

Se dedica este apartado a revisar detalles sobre el tamaño del *dataset*, la selección de datos para entrenamiento, validación y pruebas, y las herramientas utilizadas.

La siguiente tabla resume algunos detalles sobre el *dataset* de entrenamiento y validación en cuanto a su tamaño para cada modelo.

Tabla 3. Tamaño del dataset

Modelo	Cámara Inferior	Cámara Superior
Cantidad de clases	17	7
Clases	B, C, D, E, F, G, H, I, L, O, P, Q, R, T, U, W, X	A, K, M, N, S, V, Y
<b>Imágenes por clase</b>	<b>2300*</b>	<b>2300*</b>
▪ Con fondo aumentado	900*	900*
▪ En parque	700*	700*
▪ En salón	700*	700*
<b>Total de imágenes</b>	<b>39197</b>	<b>16209</b>

*\*aproximadamente por motivos de depuración*

El número de imágenes por clase es aproximado ya que las imágenes han pasado por un proceso de depuración manual en donde se han eliminado imágenes con manchas de contraluz, imágenes borrosos e imágenes sin una representación clara de la señal ejecutada.

Con respecto a los datos utilizados para pruebas, se crea un *dataset* con 15 imágenes por letra para cada una de las pruebas y para cada uno de los modelos. Para el modelo de cámara superior se obtuvo unas 105 imágenes para cada prueba y para el modelo de la cámara inferior un total de 255 imágenes por prueba.

En cuanto a la división de los datos, la siguiente tabla resume la elección de separación en datos de entrenamiento, validación y pruebas por usuarios. Debido a la cantidad de imágenes recolectadas, se ha optado por utilizar la mayor cantidad de datos posibles para entrenamiento, por tal razón sólo un grupo fue seleccionado para validación.

Tabla 4. Selección de datos para entrenamiento, validación y pruebas

Usuarios	Entornos				
	Parque	Interiores	Fondo aumentado	Pruebas P1	Pruebas P2
#1	Entrenamiento	Entrenamiento	Entrenamiento	P1.S-P1.I	-
#2	Entrenamiento	Validación	Entrenamiento	-	-
#3	Entrenamiento	Entrenamiento	-	-	-
#4	-	-	-	-	P2.S-P2.I

## 4.4 HERRAMIENTAS UTILIZADAS

### 4.4.1 Cámaras

Para el componente de visión se ha utilizado una cámara de acción gran angular y la cámara de un teléfono inteligente. Ambas cámaras sujetas al usuario.

#### 4.4.1.1 Cámara superior

Para la cámara superior se ha utilizado una cámara de acción *Xiaomi Yi Cam* con un sistema de soporte comúnmente utilizado en los deportes extremos. La cámara y su sistema de montura se pueden apreciar en la Ilustración 23.



Ilustración 23. Montura y cámara superior

La cámara debe ser colocada en la frente del usuario enfocando ligeramente inclinada el suelo, como se encuentra en la Ilustración 24.



*Ilustración 24. Usuario con cámara superior montada*

#### **4.4.1.2 Cámara inferior**

Con respecto a la cámara inferior, se ha utilizado la cámara de un teléfono inteligente: un *Samsung Galaxy S7* montado con un arnés como se puede apreciar en la Ilustración 25.



*Ilustración 25. Montura y cámara inferior*

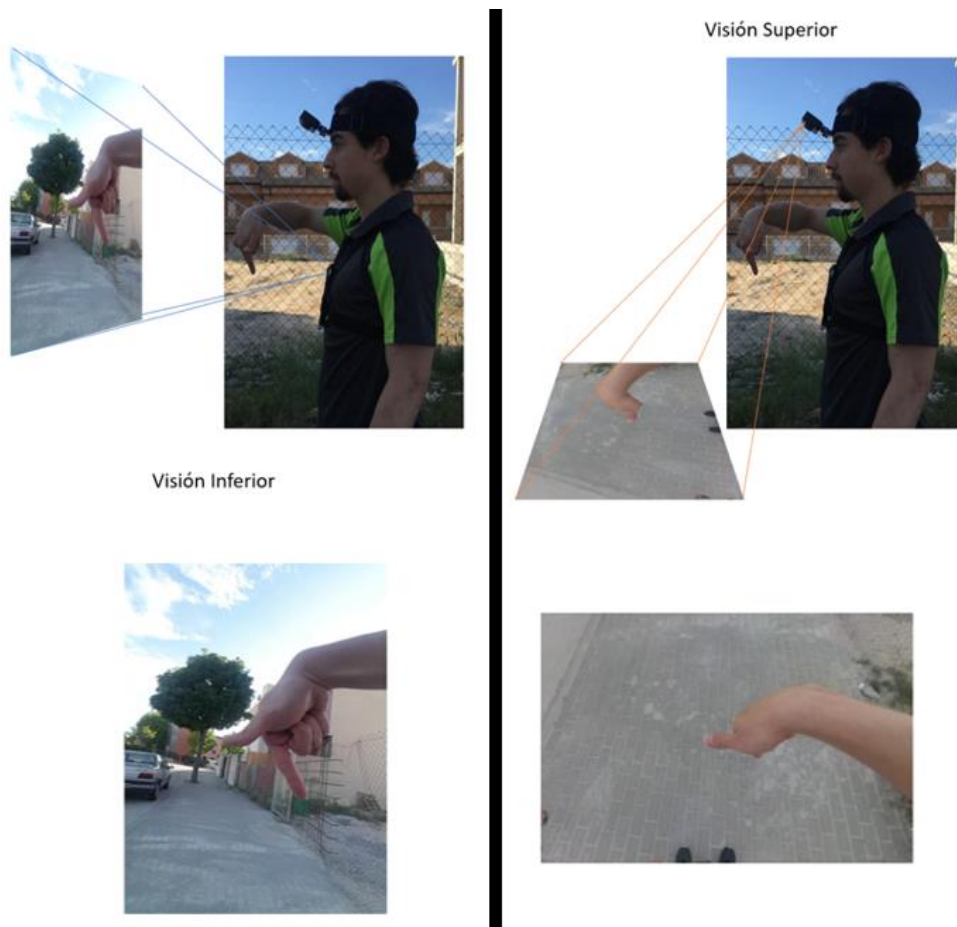
Se coloca a través de una montura de pectorales diseñada para cámaras deportivas pero que presenta un uso conveniente y oportuno para el proyecto. En la Ilustración 26 se puede apreciar su uso.



*Ilustración 26. Usuario con cámara inferior montada*

#### **4.4.1.3 Campo de observación**

El uso de una cámara superior y una inferior nos permite tener dos perspectivas distintas para una misma seña. De este modo se pueden generar dos clasificadores, uno para cada cámara. Con el uso de dos cámaras se podrá diferenciar las señas que son muy similares desde una perspectiva separándolas en dos clasificadores. En la Ilustración 27 se puede apreciar el uso de las dos cámaras y el campo de observación que nos permiten.



*Ilustración 27. Campo de observación de ambas perspectivas de cámara para la seña correspondiente a la letra Q.*

#### 4.4.2 Herramientas software

Para desarrollar los programas que automatizaron parte de la creación del *dataset*, se ha utilizado *C++* como lenguaje de programación y *OpenCV* como librería de procesamiento de imágenes, que han permitido realizar todas las operaciones necesarias sobre las imágenes.

Como entorno de desarrollo se utilizó *Microsoft Visual Studio 2017*.

#### 4.4.3 Ordenador de desarrollo

Las especificaciones sobre el ordenador utilizado para desarrollo están descritas en la Tabla 5.

Tabla 5. Especificaciones Técnicas Ordenador de Desarrollo

ESPECIFICACIONES: Ordenador Desarrollo	
<b>Modelo CPU</b>	Intel(R) Core (TM) i7-8550U CPU @ 1,80 GHz
<b>Memoria</b>	8,0 GiB
<b>Sistema Operativo</b>	Windows 10 Home

### 4.5 ALGORITMOS DESARROLLADOS

Este apartado presenta el diagrama de flujo de los algoritmos desarrollados para la creación del *dataset*.

#### 4.5.1 Extracción de imágenes de vídeos

La Ilustración 28 muestra el diagrama de flujo del *software* diseñado para extraer imágenes de los vídeos recolectados. Para los vídeos recolectados con el *Samsung Galaxy S7* ha sido necesario rotar las imágenes.

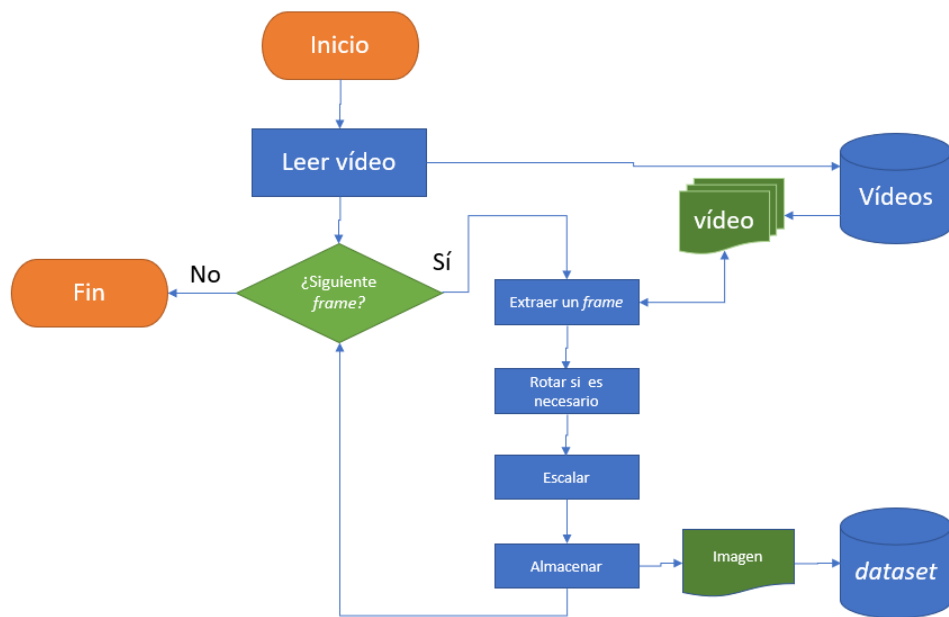


Ilustración 28. Diagrama de flujo algoritmo de extracción de imágenes

#### 4.5.2 Cambio de fondos

El algoritmo diseñado para añadir un fondo nuevo a las imágenes recolectadas con fondo uniforme se presenta en la Ilustración 29.

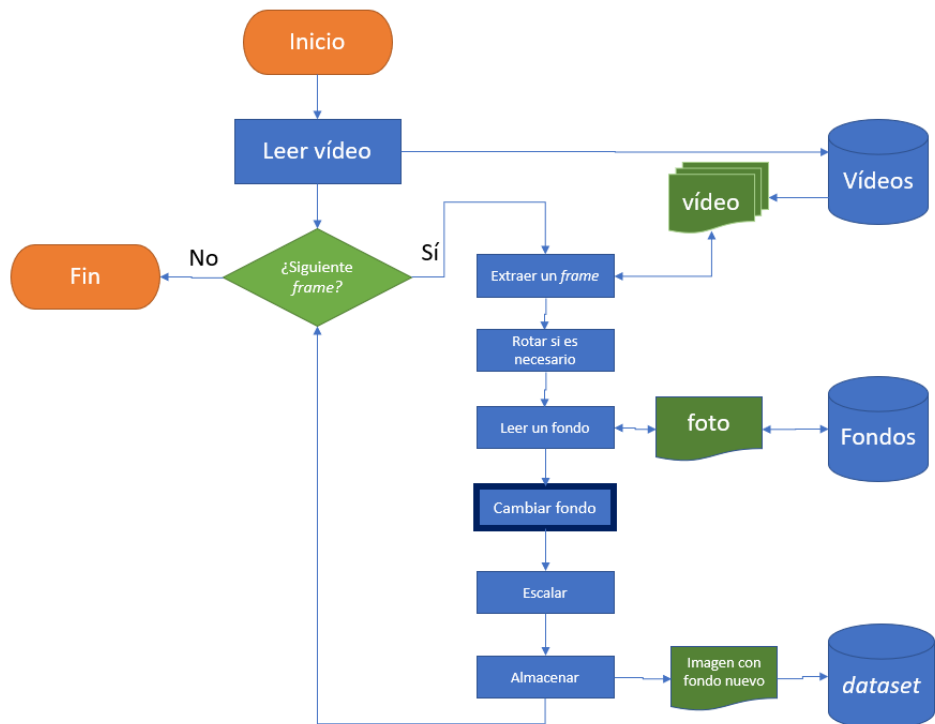


Ilustración 29. Algoritmo para cambio de fondo de imágenes

# Capítulo 5. CREACIÓN DEL CLASIFICADOR

---

En este capítulo se explica la implementación de los clasificadores. Se menciona el proceso realizado para configurar los clasificadores y se detalla la arquitectura final alcanzada. También se incluye un apartado con las herramientas utilizadas.

## 5.1 ALGORITMO DE CLASIFICACIÓN

Como algoritmo de clasificación se ha utilizado una *CNN*. Esta sección explica el proceso de entrenamiento y configuración de la *CNN* y la arquitectura obtenida.

### 5.1.1 Proceso de entrenamiento

Durante el proceso de ajuste de una *CNN* se realizaron varios entrenamientos hasta encontrar la configuración adecuada. Para iniciar este proceso ha sido necesario elegir una métrica de rendimiento utilizada y un enfoque para configurar los hiperparámetros.

#### 5.1.1.1 Métricas de rendimiento

Se ha utilizado como métrica de rendimiento la precisión del modelo y el coste. La precisión es el total de imágenes clasificadas correctamente entre el total de imágenes clasificadas; y el coste puede ser expresado como qué tanto penalizamos los errores que comete el modelo.

De las diferentes funciones de coste que existen la más utilizada en este tipo de problemas es *categorical cross-entropy*. Por tal razón ha sido la elección como función de coste.

Con respecto al objetivo de rendimiento, como en este trabajo se desarrolla una primera aproximación en el desarrollo del sistema completo, no se ha planteado alcanzar una precisión en específico, por ejemplo: 80%. Sin embargo, se aspira a una precisión que al menos supere el 50%. Hay señas que son muy similares y se consideran confusiones entre ellas como un comportamiento aceptable del modelo.

El objetivo general en cuanto a rendimiento ha sido tratar de reducir lo más posible la función de coste y tratar de hacer llegar lo más alto posible la precisión del modelo en el conjunto de validación, y con esto poder concluir sobre la viabilidad de desarrollar un proyecto de este tipo.

#### 5.1.1.2 Enfoque de ajuste

Dentro de una red neuronal convolucional se pueden ajustar muchos hiperparámetros que pueden ser: el tamaño de las ventanas de convolución, la cantidad de capas de convolución, el uso de técnicas de regulación, tasa de aprendizaje, optimizador utilizado, funciones de activación, entre otras.

El enfoque ha sido ir variando estos hiperparámetros de la red neuronal convolucional hasta hacerla llegar a resultados aceptables. Entre los ajustes más realizados en el proceso de entrenamiento están: optimizador utilizado, tasa de aprendizaje, intensidad de *dropout*, número de capas de convolución, capas de *pooling*; y en algunos casos: cambios en la función de activación y número de capas ocultas.

### 5.1.1.3 Regularización

Como regularización se han utilizado las técnicas de aumento de datos mencionadas en el apartado 4.2 (Proceso de aumento de datos) y *dropout* en la capa densa de la *CNN* como se muestra en Ilustración 30.



Ilustración 30. Uso de dropout

## 5.1.2 Arquitectura

Las siguientes son las arquitecturas finales para ambos modelos. Ambos modelos son iguales con diferencia en la capa de salida.

### 5.1.2.1 Modelo cámara superior

En Ilustración 31 se puede apreciar la arquitectura de la *CNN* para el modelo superior. Se utilizaron 7 capas de convolución combinadas con dos capas de *pooling* en la capa de extracción de características. La capa densa cuenta con una entrada de 64 neuronas, seguida de una capa oculta de otras 64 neuronas y; 7 neuronas de salida, una por cada letra.



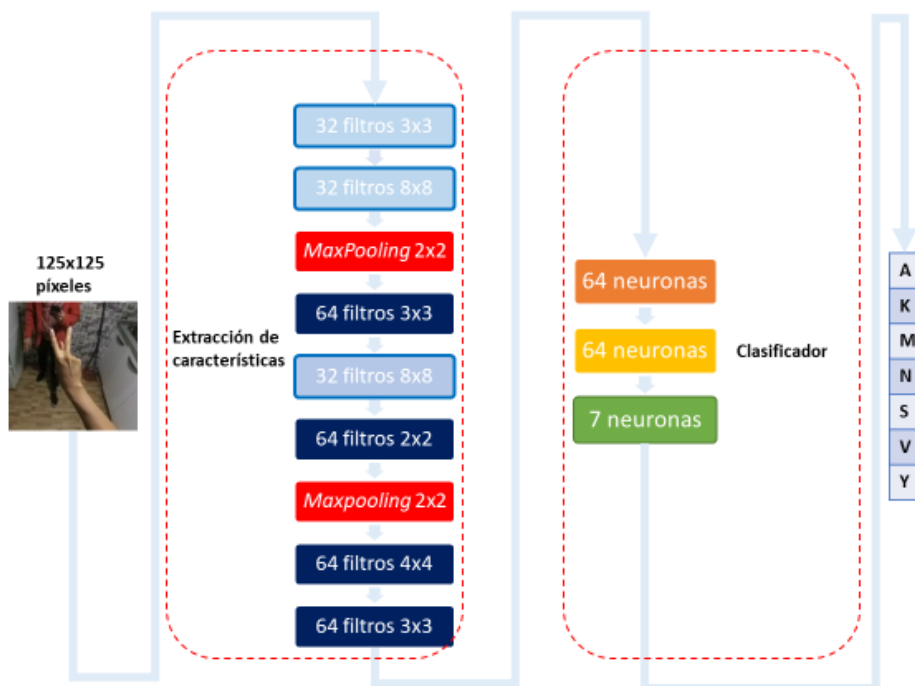


Ilustración 31. Arquitectura CNN modelo superior

#### 5.1.2.2 Modelo cámara inferior

La CNN para el modelo inferior es igual que la CNN para el modelo superior. La única diferencia es la cantidad de neuronas de salida (Ilustración 32).

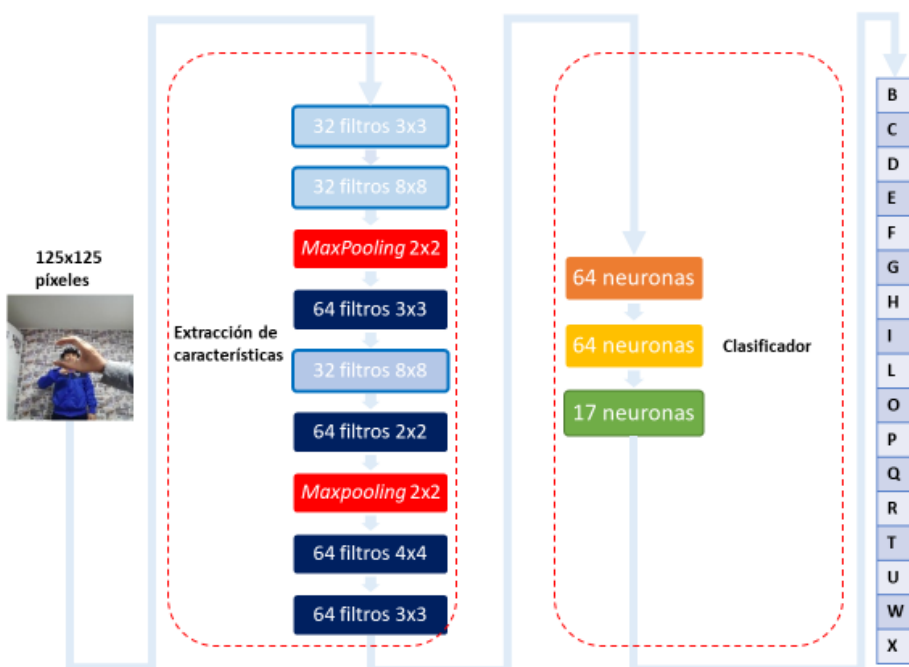


Ilustración 32. Arquitectura CNN modelo inferior

## 5.2 HERRAMIENTAS UTILIZADAS

### 5.2.1 Herramientas software

Para implementar la red neuronal se ha utilizado el lenguaje de programación *Python* y la librería *Keras* para el desarrollo de la arquitectura de la red neuronal. Y se ha realizado todo el desarrollo sobre *Jupyter Notebook* ya que nos permite trabajar en distintos servidores desde un entorno web.

### 5.2.2 Hardware de aprendizaje profundo

Para entrenar la red neuronal se han ejecutado entrenamientos en una máquina virtual en *Google Cloud Engine*, un ordenador portátil y un servidor de altas prestaciones.

A pesar de que la máquina virtual en *Google Cloud Engine* es la más lenta, con estos ordenadores ha sido posible ejecutar tres configuraciones de la red neuronal en paralelo, lo cual ha permitido un acercamiento al modelo deseado con mayor rapidez.

#### 5.2.2.1 Google Cloud Engine

Se ha utilizado un ordenador básico de *Google Cloud Engine* cuyas especificaciones se encuentran en la Tabla 6.

Tabla 6. Especificaciones técnicas máquina virtual

ESPECIFICACIONES: Máquina Virtual	
Número de CPU	6
Modelo CPU	Intel(R) Xeon(R) CPU @ 2.00GHz
Memoria	22,5 GiB
Sistema Operativo	Debian GNU/Linux 9 (stretch)

#### 5.2.2.2 Ordenador personal

El ordenador personal utilizado tiene las siguientes especificaciones técnicas.

Tabla 7. Especificaciones técnicas ordenador personal

ESPECIFICACIONES: Ordenador personal	
Número de CPU	1
Modelo CPU	Intel(R) Core (TM) i7-6500U CPU @ 2.50GHz
Memoria	8,0 GiB
Sistema Operativo	Windows 10 Education
GPU	
Modelo	NVIDIA GeForce GTX 950M

<b>Número de Núcleos</b>	640
<b>Memoria</b>	4,00 GiB

### 5.2.2.3 Servidor de altas prestaciones

Las especificaciones técnicas correspondientes al servidor proporcionado por el tutor se encuentran resumidas en la siguiente tabla.

*Tabla 8. Especificaciones técnicas del ordenador de altas prestaciones*

<b>ESPECIFICACIONES: Servidor de Altas Prestaciones</b>	
<b>Número de CPU</b>	20
<b>Modelo CPU</b>	Intel(R) Core (TM) i9-7900X CPU @ 3.30GHz
<b>Memoria</b>	31 GiB
<b>Sistema Operativo</b>	Ubuntu 16.04.6 LTS
<b>GPU1</b>	
<b>Modelo</b>	NVIDIA TITAN V
<b>Número de Núcleos</b>	5120
<b>Memoria</b>	12 GiB
<b>GPU2</b>	
<b>Modelo</b>	NVIDIA GeForce GTX 1080
<b>Número de Núcleos</b>	3584
<b>Memoria</b>	11 GiB

## Capítulo 6. PRUEBAS Y RESULTADOS

Este capítulo incluye las pruebas realizadas sobre los clasificadores, su descripción y objetivos, los resultados de las pruebas y su análisis.

### 6.1 DESCRIPCIÓN DE LAS PRUEBAS

Se han realizado dos pruebas para cada clasificador. Dos pruebas para el clasificador de cámara superior y dos pruebas para el clasificador de cámara inferior. La Tabla 9 resume las pruebas realizadas con sus objetivos.

Tabla 9. Pruebas realizadas

Identificador	Descripción	Objetivo
P1.S	Imágenes de usuario familiar para el clasificador de cámara superior	Medir y analizar el rendimiento del clasificador con un usuario que participó en la generación de datos de entrenamiento
P1.I	Imágenes de usuario familiar para el clasificador de cámara inferior	
P2.S	Imágenes de usuario nuevo en escenario real para el clasificador de cámara superior	Medir y analizar el rendimiento del clasificador con un usuario que participó en la generación de datos de entrenamiento
P2.I	Imágenes de usuario en escenario real para el clasificador de cámara inferior	

El *dataset* de pruebas P1 contiene imágenes recolectadas por un usuario que participó en el *dataset* utilizado para entrenamiento, pero en un escenario nuevo (imágenes con un fondo distinto) para los clasificadores. Se podría decir que la mano de este sujeto es familiar para los modelos.

El *dataset* de pruebas P2 está compuesto por imágenes recolectadas por un usuario nuevo en un escenario que refleja un uso real del sistema: con otra persona en frente suponiendo una conversación. Se recolectaron estas imágenes con un sujeto frente al usuario mientras este ejecutaba las señas, ilustrando el caso en el que el usuario quiera comunicarse con otra persona. En la Ilustración 33 se puede apreciar extractos de estos *datasets*.

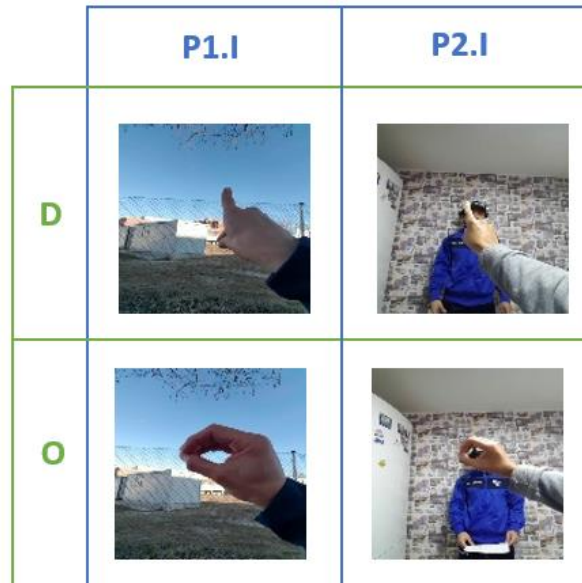


Ilustración 33. Señas para las letras D y O de los datasets de pruebas P1.I y P2.I

## 6.2 RESULTADOS DE PRUEBAS

### 6.2.1 Prueba P1.S

La matriz de confusión y las medidas de precisión y *recall* para cada clase, resultado de la prueba P1.S se pueden apreciar en la Ilustración 34 e Ilustración 35.

		PREDICCIÓN						
		A	K	M	N	S	V	Y
VALOR REAL	A	1	0	0	0	0	0	0
	K	0	0	0	0	0	0,1	0,9
	M	0,1	0	0,9	0	0	0	0
	N	0	0	0,7	0,3	0	0	0
	S	0	0	0,9	0,1	0	0	0
	V	0	0	0	0	0	0,7	0,3
	Y	0	0	0	0	0	0	1

Ilustración 34. Matriz de confusión del modelo de cámara superior para la prueba P1.S (105 instancias)

	Precisión	Recall
A	0,94	1,00
K	0,00	0,00
M	0,37	0,93
N	0,67	0,27
S	0,00	0,00
V	0,91	0,67
Y	0,44	1,00

Ilustración 35. Medidas de precisión y recall del modelo de cámara superior para la prueba P1.S (105 instancias)

De este modelo, la precisión media es 0,48. La seña que mejor clasifica es la A, con un *recall* de 1,00, lo que quiere decir que todas las imágenes con señas de letra A fueron clasificadas como letra A. Este mismo comportamiento es observado con la seña Y, sin embargo, el sistema suele clasificar todas las señas K como pertenecientes a la clase Y. Estas dos señas, K e Y, son ligeramente diferentes, con lo cual es posible mejorar el modelo.

Las señas M, N y S son sumamente similares. En este caso, la precisión media entre estas tres clases es de 0,34, lo que permite concluir que el algoritmo suele adivinar la clasificación de imágenes con cualquiera de estas clases, equivocándose dos de cada tres intentos. Sin embargo, la clasificación entre estas tres señas es la más complicada, siendo la única diferencia entre las tres la posición del dedo pulgar (Ilustración 44, página 56).

En general, el modelo suele confundir las letras A con la Y, y las letras M, N y S entre sí. La letra V es tiene una precisión aceptable, pero con confusiones con la letra Y.

## 6.2.2 Prueba P1.I

La matriz de confusión y las medidas de precisión y *recall* para cada clase resultado de la prueba P1.I se pueden apreciar en la Ilustración 36 e Ilustración 37.

		PREDICCIÓN																	
		B	C	D	E	F	G	H	I	L	O	P	Q	R	T	U	W	X	
VALOR REAL	B	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	C	0	0,9	0	0	0	0,1	0	0	0	0	0	0	0	0	0	0	0	
	D	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	E	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
	F	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	
	G	0	0	0	0	0	0,8	0	0,1	0,1	0	0	0	0	0,1	0	0	0	
	H	0	0	0	0	0	0,5	0,5	0	0	0	0	0	0	0	0	0	0	
	I	0	0	0	0	0	0,1	0,1	0,7	0	0	0	0	0	0,1	0	0	0	
	L	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	
	O	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	
REAL	P	0	0	0	0	0	0,1	0	0	0	0	0,9	0	0	0	0	0	0	
	Q	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	
	R	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	
	T	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	
	U	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	
	W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	
	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	

Ilustración 36. Matriz de confusión del modelo de cámara inferior para la prueba P1.I (225 instancias)

	Precisión	Recall
B	1,00	1,00
C	1,00	0,87
D	1,00	1,00
E	1,00	1,00
F	1,00	1,00
G	0,50	0,80
H	0,89	0,53
I	0,92	0,73
L	0,94	1,00
O	1,00	1,00
P	1,00	0,93
Q	1,00	1,00
R	1,00	1,00
T	0,88	1,00
U	1,00	1,00
W	1,00	1,00
X	1,00	1,00

*Ilustración 37. Medidas de precisión y recall del modelo de cámara inferior para la prueba P1.I (255 instancias)*

El rendimiento del clasificador creado para la cámara inferior es bastante aceptable. La mayoría de las señas son clasificadas sin muchos inconvenientes, siendo la precisión media de 0,95.

La mayoría de los errores cometidos por este clasificador están relacionados a confusión entre las letras G y H. Cuando el algoritmo no está seguro de si la seña de la imagen es H o no, suele interpretarla como G. Sin embargo, para un sistema de traducción, un algoritmo que rectifique este tipo de problemas sería una solución.

### 6.2.3 Prueba P2.S

La matriz de confusión y las medidas de precisión y *recall* para cada clase resultado de la prueba P2.S se pueden apreciar en la Ilustración 38 e Ilustración 39.

		PREDICCIÓN						
		A	K	M	N	S	V	Y
VALOR	A	1	0	0	0	0	0	0
	K	0,1	0,7	0	0,1	0	0	0
	M	0	0	1	0	0	0	0
	N	0,1	0	0,3	0,5	0	0	0,1
	S	0,1	0	0,4	0,5	0	0	0,1
	V	0	0,7	0	0	0	0,3	0
	Y	0	0,1	0,1	0,8	0	0	0

*Ilustración 38. Matriz de confusión del modelo de cámara superior para la prueba P2.S (105 instancias)*

	Precisión	Recall
A	0,79	1,00
K	0,48	0,73
M	0,54	1,00
N	0,28	0,53
S	0,00	0,00
V	1,00	0,27
Y	0,00	0,00

Ilustración 39. Medidas de precisión y recall del modelo de cámara superior para la prueba P2.I (105 instancias)

El rendimiento del clasificador de cámara superior para la prueba P2.S, que implica la inclusión de un usuario nuevo, se asemeja a los resultados obtenidos en la prueba P1.S. La precisión media de este clasificador en esta prueba es del 0,44.

Al igual que en la prueba P1.S, la señal para la letra A es la que mejor clasifica el modelo, y las letras M, N y S se confunden entre sí. Sin embargo, a diferencia de la prueba P1.S, la letra Y tiene una tasa de acierto nula y la mayoría las clasifica como N.

A pesar de que se podría esperar un comportamiento similar a la prueba P1.S, probablemente las diferencias entre estos resultados y los anteriores se deba a una ejecución distinta de cada señal por parte del usuario.

#### 6.2.4 Prueba P2.I

La matriz de confusión y las medidas de precisión y *recall* para cada clase resultado de la prueba P2.I se pueden apreciar en la Ilustración 40 e Ilustración 41.

		PREDICCIÓN																	
		B	C	D	E	F	G	H	I	L	O	P	Q	R	T	U	W	X	
V A L O R	B	0,7	0	0	0	0	0	0	0,1	0	0	0	0	0	0	0	0,1	0,1	
	C	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	D	0	0	0,7	0	0	0	0	0	0,3	0	0	0	0	0	0	0	0	
	E	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
	F	0	0	0	0	0,6	0	0	0	0	0,4	0	0	0	0	0	0	0	
	G	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	
	H	0	0,7	0	0	0	0,2	0,1	0	0	0	0	0	0	0	0	0	0	
	I	0	0	0	0,9	0	0	0	0,1	0	0	0	0	0	0	0	0	0	
	L	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	
	O	0	0,1	0	0	0	0	0	0	0	0,9	0	0	0	0	0	0	0	
R E A L	P	0	0,7	0	0	0	0,1	0	0	0	0,1	0	0	0	0	0	0	0,1	
	Q	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	
	R	0	0	0	0	0	0,2	0	0	0	0	0	0	0,8	0	0	0	0	
	T	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	
	U	0,3	0	0	0,1	0	0	0	0	0	0	0	0	0	0	0	0,6	0	
	W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	
	X	0	0	0,7	0	0	0	0	0	0,3	0	0	0	0	0	0	0	0	

Ilustración 40. Matriz de confusión del modelo de cámara inferior para la prueba P2.I (255 instancias)



	Precisión	Recall
B	0,69	0,73
C	0,39	1,00
D	0,50	0,73
E	0,52	1,00
F	1,00	0,65
G	0,65	1,00
H	1,00	0,67
I	0,67	0,65
L	0,65	1,00
O	0,67	0,93
P	0,00	0,00
Q	1,00	1,00
R	1,00	0,80
T	1,00	1,00
U	1,00	0,60
W	0,88	1,00
X	0,00	0,00

*Ilustración 41. Medidas de precisión y recall del modelo de cámara inferior para la prueba P2.I (255 instancias)*

A pesar de que en la prueba P1.I se obtuvo excelentes resultados, en el caso de un usuario nuevo para el modelo, como es el caso de esta prueba (P2.I), el sistema obtiene una tasa de acierto más baja, siendo de media 0,68. Esto nos lleva a considerar que el modelo desarrollado tiene problemas para generalizar.

En esta prueba, el modelo tiende a clasificar las señas H y P como letras C; y las seña X como D siendo los errores más importantes del modelo.

### 6.3 CONCLUSIONES Y CONSIDERACIONES


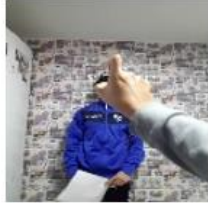


Existen diferencias entre los *dataset* de pruebas que hay que considerar al realizar conclusiones. El *dataset* de pruebas 1 contiene señas que aparecen ocupando mayor parte de la imagen que en el *dataset* de pruebas 2, en donde el usuario alejó más la mano de la cámara. Esta diferencia pudo afectar el rendimiento de los clasificadores dado que algunas señas son más grandes que otras.

Las pruebas P1 probablemente sean unas pruebas más sencillas para los clasificadores dado que las realiza un sujeto que ha aparecido en el *dataset* de entrenamiento, la condición de luz es más uniforme, con la aparición de menos contrastes. Por el contrario, en las pruebas P2 las condiciones de luz son más intensas resaltando con mayor intensidad los contornos y el fondo es menos uniforme que en las pruebas P1. Por ejemplo, una confusión habitual en la prueba P2.I estaba en clasificar la letra I como E a pesar de tener configuraciones de manos distintas, con lo que podemos concluir que no funciona apropiadamente si las imágenes no son completamente claras. En la Ilustración 42 se puede apreciar estas dos señas ejecutadas por el usuario en la prueba P2.I.








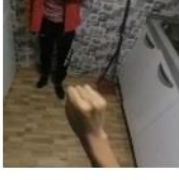
*Ilustración 42. Señas para letras E e I ejecutadas por el usuario de la prueba P2.I*

Otra consideración es la forma en la que el sujeto realiza la seña. El sujeto de la prueba 2 realiza algunas señas de manera distinta a lo visto con los 3 usuarios del *dataset* de pruebas. Por ejemplo: una confusión común en la prueba P2.I fue tender a clasificar la seña X como D, y guarda relación con la configuración de la mano ejecutada por el usuario. Problema que no sucedía con las mismas letras en la prueba P1.I. En la Ilustración 43 se puede apreciar la ejecución de la misma seña por ambos usuarios para las letras D y X.

	Usuario P1.I	Usuario P2.I
D		
X		

*Ilustración 43. Señas para las letras D y X de ejecutadas por los usuarios de las pruebas P1.I y P2.I*

Un error común en las pruebas P1.S y P2.S estuvo en clasificar las letras M, N y S, letras que poseen diferencias mínimas entre sí, siendo la única diferencia la posición del dedo pulgar (Ilustración 44). A raíz de esto, se puede considerar que este error es aceptable para el clasificador desarrollado.

	Usuario P1.I	Usuario P2.I
M		
N		
S		

*Ilustración 44. Señas para las letras M, N y S ejecutadas por los usuarios de las pruebas P1.S y P2.S*

Las condiciones de luz han sido distintas en cada prueba, lo que nos lleva a considerar que en escenarios con menos intensidad de luz y con una representación clara de la mano el sistema funciona mejor.

Finalmente, es probable que los clasificadores no hayan logrado aprender todo el espacio de características necesario para clasificar cualquier seña en cualquier entorno por cualquier usuario, con lo cual, el clasificador puede mejorarse con la inclusión de más usuarios y entornos en el *dataset*, tema que queda pendiente para trabajos futuros.

# Capítulo 7. GESTIÓN DEL PROYECTO

---

En este capítulo se presenta la metodología utilizada para gestionar el proyecto, la planificación del proyecto, el presupuesto y un plan de gestión de riesgos enfocada a una realización real del trabajo en la que se cuenta con un equipo de trabajo y financiación.

Adicionalmente, se incluye la planificación general llevada por el estudiante con sus horas trabajadas.

## 7.1 METODOLOGÍA DE DESARROLLO

Se ha decidido utilizar una adaptación de la metodología de desarrollo en cascada para la realización del proyecto. Metodología que consiste en dividir el proyecto en etapas de desarrollo que se realizan de forma secuencial. Sirve como un modelo de proceso útil en situaciones en las que los requisitos son fijos y el trabajo avanza en forma lineal hacia el final [64].

Una de las razones por las cuales se ha decidido utilizar esta metodología es su facilidad de gestión, suponiendo una planificación detallada al inicio del proyecto y luego un seguimiento, y porque la naturaleza de este proyecto supone una definición clara de los requisitos y objetivos del sistema antes de iniciarse el trabajo de desarrollo. Lo cual no supone cambios inesperados que deban contemplarse una vez avanzado el trabajo.

Esta metodología también supone ventajas sobre el proyecto dado que este puede dividirse en componentes que pueden desarrollarse y probarse de manera independiente. Esto permite que, de contar con un equipo de desarrollo grande, se puedan trabajar todos los componentes en paralelo.

Para la elaboración de este trabajo se siguen las siguientes etapas:

1. Análisis del problema
2. Diseño del sistema
3. Plan de Gestión del Proyecto
4. Desarrollo
5. Experimentación

En la etapa de análisis se estudia el problema a profundidad, el estado actual de las tecnologías y soluciones actuales, con el objetivo de obtener conocimiento suficiente para diseñar un prototipo.

Luego, en diseño del sistema se plantea a detalle la solución propuesta. Se especifican los componentes y las herramientas necesarias para su desarrollo y despliegue.

La siguiente etapa corresponde a la gestión del proyecto. Se establece la planificación, el presupuesto y se prepara un plan de riesgos con el fin de especificar un plan detallado a seguir en la elaboración de la solución.

Le sigue la etapa de desarrollo, que tiene la finalidad de construir el sistema y comprobar el correcto funcionamiento de cada componente.

Finalmente, el proyecto termina con la experimentación del sistema que pretende generar resultados y plantear a investigaciones futuras la viabilidad de un sistema de este tipo.

## 7.2 PLANIFICACIÓN SEGUIDA POR EL ESTUDIANTE

La siguiente tabla (Tabla 10) muestra la planificación general seguida por el estudiante para el desarrollo completo del trabajo de fin de máster.

Tabla 10. Cronograma de trabajo seguido por el estudiante

Actividades	Fecha inicio	Fecha fin	Duración (días)	Duración (horas)
PROYECTO TFM	10/01/2019	02/05/2019	76	440
ANÁLISIS DEL PROBLEMA	10/01/2019	06/02/2019	19,5	136
DISEÑO DEL SISTEMA	06/02/2019	22/02/2019	12	56
<b>COMPONENTE DE VISIÓN</b>	<b>11/02/2019</b>	<b>14/02/2019</b>	<b>3</b>	<b>12</b>
<b>COMPONENTE INTELIGENCIA ARTIFICIAL</b>	<b>14/02/2019</b>	<b>20/02/2019</b>	<b>4</b>	<b>16</b>
<b>DEFINICIÓN GENERAL DE REQUISITOS</b>	<b>20/02/2019</b>	<b>22/02/2019</b>	<b>2</b>	<b>16</b>
PLAN DE GESTIÓN	21/02/2019	07/03/2019	10	40
DESARROLLO	07/03/2019	25/04/2019	30	296
<b>COMPONENTE DE VISIÓN</b>	<b>08/03/2019</b>	<b>12/03/2019</b>	<b>2</b>	<b>8</b>
<b>CONJUNTO DE DATOS</b>	<b>12/03/2019</b>	<b>01/04/2019</b>	<b>14</b>	<b>60</b>
<b>CLASIFICADOR</b>	<b>01/04/2019</b>	<b>25/04/2019</b>	<b>13</b>	<b>72</b>
PRUEBAS Y EVALUACIÓN	25/04/2019	02/05/2019	5	20
REDACCIÓN DEL DOCUMENTO	02/05/2019	21/06/2019	50	40

## 7.3 PLANIFICACIÓN CON EQUIPO DE TRABAJO

La planificación detallada del proyecto se puede apreciar en la siguiente tabla (Tabla 11). Esta planificación ha sido realizada considerando la participación de cuatro colaboradores: un director del proyecto, un experto en *machine learning*, un desarrollador y un experto en lengua de señas panameña y la duración presentada contempla el trabajo de cuatro personas. Sin embargo, es la misma planificación seguida por el estudiante con diferencia en las horas trabajadas.

Tabla 11. Cronograma de trabajo

Actividades	Fecha inicio	Fecha fin	Duración (días)	Duración (horas)
PROYECTO	10/01/2019	02/05/2019	76	800
ANÁLISIS DEL PROBLEMA	10/01/2019	06/02/2019	23	272

A.1 Revisión soluciones actuales	10/01/2019	18/01/2019	7	56
A.2 Revisión modelos de aprendizaje automático	21/01/2019	23/01/2019	3	48
A.3 Estudio CNN	21/01/2019	30/01/2019	8	128
A.4 Estudio lengua de señas	30/01/2019	04/02/2019	3	24
A.5 Análisis situación actual en Panamá	04/02/2019	06/02/2019	2	16
<b>DISEÑO DEL SISTEMA</b>	<b>06/02/2019</b>	<b>22/02/2019</b>	<b>12</b>	<b>112</b>
A.6 Propuesta y diseño general	06/02/2019	11/02/2019	3	24
COMPONENTE DE VISIÓN	11/02/2019	14/02/2019	3	24
A.7 Diseño de componente	11/02/2019	13/02/2019	2	16
A.8 Definición de requisitos	13/02/2019	14/02/2019	1	8
COMPONENTE INTELIGENCIA ARTIFICIAL	14/02/2019	20/02/2019	4	32
A.9 Selección de modelo de aprendizaje automático	14/02/2019	15/02/2019	1	8
A.10 Especificaciones del conjunto de datos	15/02/2019	19/02/2019	2	16
A.11 Definición de requisitos	19/02/2019	20/02/2019	1	8
DEFINICIÓN GENERAL DE REQUISITOS	20/02/2019	22/02/2019	2	32
A.12 Especificación de requisitos	20/02/2019	21/02/2019	1	8
A.13 Resumen y análisis	21/02/2019	22/02/2019	1	24
<b>PLAN DE GESTIÓN</b>	<b>21/02/2019</b>	<b>07/03/2019</b>	<b>10</b>	<b>80</b>
A.14 Selección de metodología	21/02/2019	26/02/2019	3	24
A.15 Definición plan de desarrollo	26/02/2019	01/03/2019	3	24
A.16 Definición de presupuesto	01/03/2019	05/03/2019	2	16
A.17 Elaboración plan de gestión de riesgos	05/03/2019	07/03/2019	2	16
<b>DESARROLLO</b>	<b>07/03/2019</b>	<b>25/04/2019</b>	<b>30</b>	<b>296</b>
A.18 Preparación de ordenadores	07/03/2019	08/03/2019	1	16
COMPONENTE DE VISIÓN	08/03/2019	12/03/2019	2	16
A.19 Implementación de sistema de cámaras	08/03/2019	11/03/2019	1	8
A.20 Evaluación	11/03/2019	12/03/2019	1	8
CONJUNTO DE DATOS	12/03/2019	01/04/2019	14	120
A.21 Preparación de entornos	12/03/2019	13/03/2019	1	16
A.22 Desarrollo programas auxiliares	13/03/2019	20/03/2019	5	40
A.23 Recolección de datos	20/03/2019	27/03/2019	5	40
A.24 Depuración y verificación	27/03/2019	01/04/2019	3	24
CLASIFICADOR	01/04/2019	25/04/2019	13	144
A.25 Definición de arquitectura inicial	01/04/2019	02/04/2019	1	8
A.26 Creación del clasificador	02/04/2019	03/04/2019	1	8
A.27 Entrenamiento y ajuste	03/04/2019	24/04/2019	15	120
A.28 Análisis de rendimiento	24/04/2019	25/04/2019	1	8
<b>PRUEBAS Y EVALUACIÓN</b>	<b>25/04/2019</b>	<b>02/05/2019</b>	<b>5</b>	<b>40</b>

A.29 Recolección de datos	25/04/2019	30/04/2019	3	24
A.30 Realización de pruebas	30/04/2019	01/05/2019	1	8
A.31 Análisis de pruebas	01/05/2019	02/05/2019	1	8

## 7.4 PRESUPUESTO

El presupuesto se ha dividido principalmente en dos partes: recursos humanos y, materiales y gastos adicionales. El presupuesto Luego se ha realizado un desglose de costes totales considerando costes de gestión y riesgos.

### 7.4.1 Recursos humanos

La ejecución del proyecto necesita de la colaboración de personal capacitado. Se ha identificado cuatro roles para llevar a cabo el proyecto: director de proyecto, experto en *machine learning*, un desarrollador y un asesor experto en lengua de señas panameña.

El director tendrá la responsabilidad de dirigir el proyecto y debe ser un ingeniero informático, el experto en *machine learning* llevará a cabo el diseño, implementación y evaluación del clasificador; el desarrollador cooperará en el desarrollo de software auxiliar; y el experto en lengua de señas panameña brindará asesoría.

El salario<sup>3</sup> de cada colaborador es el siguiente:

- Director de proyecto: 29,74 €/hora
- Experto en *machine learning*: 27,52 €/hora
- Desarrollador: 20,87 €/hora
- Experto en lengua de señas: 29,74 €/hora

Las siguientes tablas (Tabla 12, Tabla 13, Tabla 14 y Tabla 15) detallan las actividades asignadas, su duración y coste total para cada colaborador. Semana Santa (del 14 al 21 de abril) y domingos no son laborables, y los sábados son laborables media jornada.

Tabla 12. Duración y coste de actividades para el Director de Proyecto

DIRECTOR DE PROYECTO		
Actividades	Duración (horas)	Coste (€)
A.1 Revisión soluciones actuales	56	1665,44
A.2 Revisión modelos de aprendizaje automático	24	713,76
A.3 Estudio CNN	64	1903,36
A.4 Estudio lengua de señas	24	713,76
A.5 Análisis situación actual en Panamá	16	475,84
A.6 Propuesta y diseño general	24	713,76
A.7 Diseño de componente	16	475,84

<sup>3</sup> De conformidad con la Resolución de 22 de febrero de 2018, de la Dirección General de Empleo, por la que se registra y publica el XVII Convenio colectivo estatal de empresas de consultoría y estudios de mercado y de la opinión pública

<i>A.8 Definición de requisitos</i>	8	237,92
<i>A.12 Especificación de requisitos</i>	8	237,92
<i>A.13 Resumen y análisis</i>	8	237,92
<i>A.14 Selección de metodología</i>	24	713,76
<i>A.15 Definición plan de desarrollo</i>	24	713,76
<i>A.16 Definición de presupuesto</i>	16	475,84
<i>A.17 Elaboración plan de gestión de riesgos</i>	16	475,84
<b>totales</b>	<b>328</b>	<b>9754,72</b>

Tabla 13. Duración y coste de actividades para el Experto en Machine Learning

EXPERTO EN MACHINE LEARNING		
Actividades	Duración (horas)	Coste (€)
<i>A.3 Estudio CNN</i>	64	1761,28
<i>A.9 Selección de modelo de aprendizaje automático</i>	8	220,16
<i>A.10 Especificaciones del conjunto de datos</i>	16	440,32
<i>A.11 Definición de requisitos</i>	8	220,16
<i>A.13 Resumen y análisis</i>	8	220,16
<i>A.18 Preparación de ordenadores</i>	8	220,16
<i>A.19 Implementación de sistema de cámaras</i>	8	220,16
<i>A.20 Evaluación</i>	8	220,16
<i>A.21 Preparación de entornos</i>	8	220,16
<i>A.23 Recolección de datos</i>	40	1100,8
<i>A.24 Depuración y verificación</i>	24	660,48
<i>A.25 Definición de arquitectura inicial</i>	8	220,16
<i>A.26 Creación del clasificador</i>	8	220,16
<i>A.27 Entrenamiento y ajuste</i>	120	3302,4
<i>A.28 Análisis de rendimiento</i>	8	220,16
<i>A.29 Recolección de datos</i>	24	660,48
<i>A.30 Realización de pruebas</i>	8	220,16
<i>A.31 Análisis de pruebas</i>	8	220,16
<b>totales</b>	<b>384</b>	<b>10567,68</b>

Tabla 14. Duración y coste de actividades para el Desarrollador.

DESARROLLADOR		
Actividades	Duración (horas)	Coste (€)
<i>A.13 Resumen y análisis</i>	8	166,96
<i>A.18 Preparación de ordenadores</i>	8	166,96
<i>A.21 Preparación de entornos</i>	8	166,96
<i>A.22 Desarrollo programas auxiliares</i>	40	834,8
<b>totales</b>	<b>64</b>	<b>1335,68</b>



Tabla 15. Duración y coste de actividades para el Experto en lengua de señas

EXPERTO EN LENGUA DE SEÑAS		
Actividades	Duración (horas)	Coste (€)
A.4 Estudio lengua de señas	24	713,76
A.5 Recolección de datos	24	713,76
A.6 Realización de pruebas	8	237,92
<b>totales</b>	<b>328</b>	<b>1665,44</b>

#### 7.4.2 Materiales y costes adicionales

Los materiales y demás que se necesitarán para la realización formal del proyecto se resumen en la Tabla 16.

Tabla 16. Costes de materiales

Material	unidades	Coste	Periodo de amortización	Tiempo de uso	Coste en proyecto
Cámaras	2	59,49 €/u	60 meses	4 meses	7,93 €
Ordenadores	3	549 €/u	60 meses	4 meses	109,80 €
Montaje para cámaras	1	17,49 €/u	60 meses	4 meses	1,17 €
Licencia <i>Microsoft Visual Studio</i>	1	45 €/mes	-	2 meses	90,00 €
Licencia <i>Microsoft Office</i>	3	8,80 €/mes	-	4 meses	79,20 €
Ordenador de altas prestaciones	1	1,17 €/hora	-	15 días	421,20 €
Alquiler de oficina	-	800 €/mes		4 meses	3.200,00 €
<b>total</b>					<b>3.909,30 €</b>

Entre los materiales se contempla un ordenador de altas prestaciones de alquiler en *Google Cloud Engine* con especificaciones necesarias para los trabajos de aprendizaje automático. Y tres ordenadores, uno para el director del proyecto, un ordenador para el desarrollar y un ordenador para el experto en *machine learning*.

#### 7.4.3 Costes totales

La Tabla 17 resume el desglose de coste total del proyecto propuesto incluida una reserva para riesgos y gestión.

Tabla 17. Desglose de costes del proyecto

Recurso	Coste
Recursos humanos	23.323,52 €
Materiales	3.909,30 €
<b>Coste sin gastos indirectos y reservas</b>	<b>27.232,82 €</b>

Gastos indirectos (5%)	1.361,64 €
Reservas para contingencia y gestión (20%)	5.446,56 €
<b>Coste sin IVA</b>	<b>34.041,02 €</b>
IVA (21%)	7.148,61 €
<b>Coste total</b>	<b>41.189,64 €</b>

## 7.5 ASPECTOS LEGALES

Para la creación del conjunto de datos se necesita del apoyo de personas externas al proyecto que contribuyan con imágenes en las que aparecerá su mano o incluso alguna otra parte del cuerpo. Este tipo de dato es considerado un dato de carácter personal según el REGLAMENTO (UE) 2016/679 DEL PARLAMENTO EUROPEO Y DEL CONSEJO de 27 de abril de 2016 relativo a la protección de las personas físicas en lo que respecta al tratamiento de datos personales y a la libre circulación de estos datos y por el que se deroga la Directiva 95/46/CE (Reglamento general de protección de datos) o RGPD. Que en el apartado 1, artículo 5, del RGPD se define dato personal como:

“toda información sobre una persona física identificada o identificable («el interesado»); se considerará persona física identificable toda persona cuya identidad pueda determinarse, directa o indirectamente, en particular mediante un identificador, como por ejemplo un nombre, un número de identificación, datos de localización, un identificador en línea o uno o varios elementos propios de la identidad física, fisiológica, genética, psíquica, económica, cultural o social de dicha persona.”

Los procesos para seguir de conformidad al RGPD son:

- Designar a un responsable y encargado del tratamiento
- Aplicar las medidas técnicas y organizativas necesarias para garantizar la seguridad de los datos
- Solicitar el consentimiento de tratamiento de datos al interesado y fines específicos
- Notificar sus derechos, y comunicar el uso de los datos y sus responsables, y el destinatario de los datos, de manera clara.

El proveedor de servicios en la nube, *Google Cloud*, cumple con lo especificado en el RGPD, tiene las certificaciones ISO 27001 (gestión de la seguridad de la información), ISO 27017 (seguridad en la nube) y ISO 27018 (privacidad en la nube), garantizando las medidas técnicas necesarias para la seguridad de los datos [65].

## 7.6 GESTIÓN DE RIESGOS

Se ha iniciado el proceso de gestión de riesgos con un análisis de los posibles riesgos calificándolos según nivel de amenaza, se define un umbral de tolerancia y planes de acción necesarios.

### 7.6.1 Análisis de riesgos

Se decide medir el nivel de amenaza de un riesgo de acuerdo con su probabilidad de ocurrencia e impacto utilizando escalas de 1 a 5. La siguiente tabla resume los niveles de amenaza para la probabilidad e impacto y su calificación.

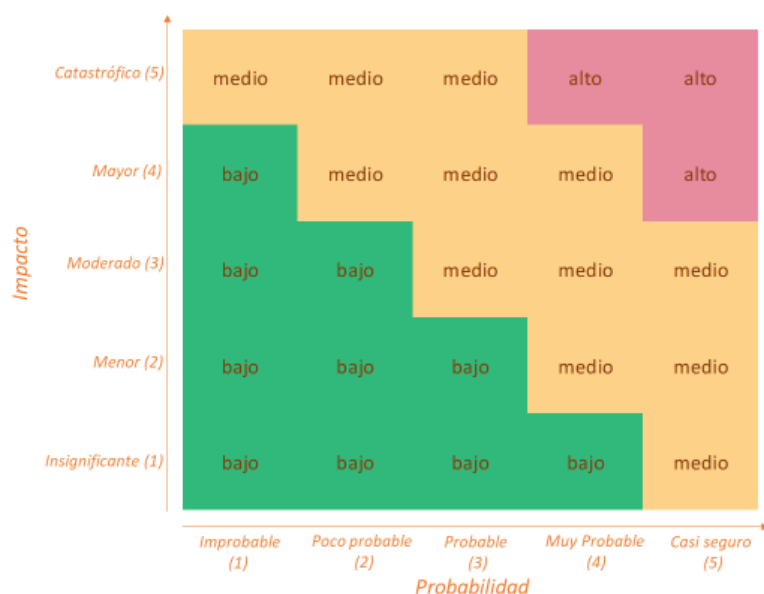


Ilustración 45. Tabla de calificación de nivel de riesgos según probabilidad e impacto

Basados en esta tabla, se han identificado los riesgos ligados al desarrollo del proyecto y su calificación. La Tabla 18 contiene una lista de todos los riesgos identificados por categorías.

Tabla 18. Riesgos identificados y su calificación

Categorías	Id.	Descripción	Impacto	Probabilidad	Calificación
Organizativo	R.1	Retrasos	5	4	Alto
	R.2	Falta desarrollador	4	1	Bajo
	R.3	Falta experto en ML	4	1	Bajo
	R.4	Falta experto en lengua de señas panameña	5	4	Alto
Técnico	R.5	Averías en las cámaras	4	2	Medio
	R.6	Averías en el ordenador adicional de ML	4	2	Medio
	R.7	Averías en el ordenador del director de proyecto	2	2	Bajo
	R.8	Averías en el ordenador de desarrollo	4	2	Medio
	R.9	Pérdida de ficheros	4	2	Medio
Externo	R.10	Proveedor de cómputo deja de prestar servicios o incumple	5	1	Medio

### 7.6.2 Planes de acción

Según el nivel de amenaza del riesgo se ha decidido afrontarlo según tres estrategias: mitigar, afrontarlo para tratar de reducir su impacto cuando suceda; evitar, definir un plan para reducir su probabilidad de ocurrencia; y aceptar: no tomar acciones.

Los riesgos considerados medios o altos incluirán un plan de acción, mientras que los riesgos calificados como bajos se aceptarán y no incluirán un plan de acción. La Tabla 19 contiene los planes de acción para cada riesgo identificado con un nivel de amenaza medio o alto.

*Tabla 19. Planes de acción*

Id. Riesgo	Estrategia	Plan de acción
R.1	mitigar	Contratar recursos humanos adicionales
R.4	mitigar	Contratar experto en otra lengua de señas
R.5	mitigar	Contratar servicios de reparación
R.6	mitigar	Contratar servicios de reparación
R.8	mitigar	Contratar servicios de reparación
R.9	evitar	Hacer copias de seguridad
R.10	mitigar	Cambiar de proveedor

## Capítulo 8. CONCLUSIONES

---

Este capítulo concluye el trabajo de fin de máster. Se mencionan las conclusiones del trabajo, que incluye aspectos técnicos y de diseño; y se mencionan los trabajos futuros que podrían realizarse a partir del trabajo realizado.

### 8.1 CONCLUSIONES

El objetivo general de este trabajo ha sido aplicar técnicas de aprendizaje profundo para el reconocimiento de señas de la lengua de señas panameña, lo cual ha sido desarrollado y evaluado en este trabajo de fin de máster. Se ha diseñado un sistema nuevo, y propuesto una técnica de aumento de datos para este problema, con lo cual se hacen contribuciones innovadoras en el área.

Se identificaron y analizaron las distintas soluciones en el estado del arte sobre reconocimiento de gestos, que incluyen desde el uso de guantes con sensores hasta técnicas de visión artificial.

Se construyó un conjunto de datos específico con 2300 imágenes para 24 clases acumulando un total de alrededor de 55000 imágenes que fueron utilizadas para entrenar los clasificadores y se recolectaron unas 360 imágenes para su evaluación, con la participación de 4 usuarios y el apoyo de algoritmos que automatizaron parte del trabajo. El modelo resultado estuvo compuesto por dos *CNN*.

Entre otras observaciones, se puede considerar que entre más usuarios participen en el *dataset*, mejor representará todo el espacio de características con lo que pueden encontrarse en un entorno real, ya sea una mano más oscura, distintas formas de ejecutar las señas o variaciones en el tamaño de la mano.

En el proceso de recolección de datos para las pruebas se ha detectado que la posición de la cámara superior es variable y afecta el campo de visión del sistema. Esta ubicación restringe la posición que debe tener el usuario al usar el sistema, si ladea su cabeza o mira en otra dirección la seña ejecutada no sería captada por la cámara superior. Sin embargo, siempre y cuando el usuario se mantenga erguido el sistema no tendrá problemas al capturar la seña.

Otro aspecto importante es que la *CNN* funciona mejor con usuarios que ha visto en el *dataset*. Este es un aspecto relevante que se puede considerar en trabajos futuros.

Las *CNN* funcionan apropiadamente para abordar el problema propuesto, pero se tiene que invertir tiempo en configurarlas para el problema.

Finalmente, se considera oportuno mencionar que los resultados obtenidos han demostrado que es posible implementar el sistema con éxito.

## 8.2 TRABAJOS FUTUROS

La solución propuesta, a pesar de no alcanzar una tasa de acierto suficientemente alta, es posible mejorarla con la participación de más usuario en el conjunto de datos en otros escenarios, lo que puede impactar significativamente el rendimiento del clasificador, con lo cual aumentar el tamaño del *dataset* sería interesante. También aumentar el *dataset* de pruebas podría llevar a un análisis más provechoso.

Realizar un análisis más minucioso sobre el comportamiento del modelo con herramientas visuales como los mapas de activación de clases añadirían valor adicional al análisis del modelo.

El prototipo desarrollado cuenta con dos clasificadores de *CNN*, cada uno con una tarea de clasificación distinta. Estos clasificadores no fueron conectados entre sí, sino que fueron evaluados por separado. En próximos trabajos sería de utilidad unificar estos dos clasificadores.

De las dos *CNN* implementadas, una tenía la tarea de clasificar 17 señas y otra la de clasificar 7 señas distintas. Podrían equilibrarse las tareas de cada clasificador separando las señas más conflictivas en cada uno. Esto podría mejorar la tasa de acierto del clasificador.

En este trabajo se desarrolló una técnica específica de aumento de datos que consiste en agregar un fondo nuevo a una imagen. En su momento consideramos que este proceso nos permitiría ahorrar tiempo en la creación del *dataset* y nos permitía contar con imágenes nuevas. El impacto que tiene el uso de esta técnica en la capacidad de generalización del clasificador no fue evaluado en este proyecto, con lo cual, podría ser interesante hacer pruebas sobre esta técnica y verificar su factibilidad.

En este trabajo se han utilizado cámaras *RGB*, pero también existe la posibilidad de usar cámaras *RGB-d* o cámaras estereoscópicas. Este tipo de cámaras permite medir profundidades, con lo cual sería posible separar la mano del fondo y facilitar el proceso de clasificación. Otra alternativa para separar el fondo de la mano es utilizar otra *CNN* específica para segmentar objetos.

El prototipo contaba con el uso de una cámara gran angular y una cámara de un móvil que en conjunto conformaban el sistema de visión. Sin embargo, se observó que el uso de estas dos cámaras restringe en cierta medida el área de ejecución de señas del usuario, con lo cual, es posible implementar el uso de cámaras gran angular para aumentar el campo de visión, lo que añadiría mayor naturalidad en la comunicación.

Conectar los componentes desarrollados en este trabajo y realizar pruebas en tiempo real contribuirían a evaluar el tiempo de respuesta del modelo y otros aspectos como la usabilidad. En este tipo de pruebas también es posible evaluar casos de uso, como deletrear palabras, que consideramos funcionaría con precisión utilizando correctores ortográficos a pesar de la tasa de acierto alcanzada.

El alfabeto manual de la lengua de señas panameña cuenta con ciertas señas dinámicas que no fueron abordadas en este trabajo. Sería interesante abordar este problema con *3dCNN*.

Finalmente, podría expandirse la capacidad del sistema incluyendo otros gestos manuales de la lengua de señas panameña que representen frases cotidianas, lo que le añadiría más utilidad en una comunicación.

# REFERENCIAS

---

- [1] H. Cooper, B. Holt y R. Bowden, «Sign Language Recognition,» de *Visual Analysis of Humans*, London, Springer, 2011.
- [2] Contraloría General de la República de Panamá, «Sistema de Indicadores de Enfoque de Género en Panamá,» [En línea]. Available: <https://www.contraloria.gob.pa/inec/siegepa/indicador.asp?IDROW=090101>. [Último acceso: 18 febrero 2019].
- [3] M. Karam, «A framework for research and design of gesture-based human computer interactions (PhD Thesis),» Faculty of Engineering, Science and Mathematics School of Electronics and Computer Science. University of Southampton, 2006.
- [4] S. Thakur, R. Mehra y B. Prakash, «Vision based computer mouse control using hand gestures,» de *2015 International Conference on Soft Computing Techniques and Implementations (ICSCTI)*, Faridabad, India, 2015.
- [5] Y. Liu y P. Zhang, «Vision-Based Human-Computer System Using Hand Gestures,» de *2009 International Conference on Computational Intelligence and Security*, Beijing, China, 2009.
- [6] U. Patel y A. G. Ambekar, «Moment Based Sign Language Recognition for Indian Languages,» de *2017 International Conference on Computing, Communication, Control and Automation (ICCUBE)*, Pune, India, 2017.
- [7] A. Nagasue, J. K. Tan, H. Kim y S. Ishikawa, «Japanese finger-spelling recognition using a chest-mounted camera,» de *2012 Proceedings of SICE Annual Conference (SICE)*, Akita, 2012.
- [8] D. Aryanie y Y. Heryadi, «American sign language-based finger-spelling recognition using k-Nearest Neighbors classifier,» de *2015 3rd International Conference on Information and Communication Technology (ICoICT)*, Nusa Dua, Bali, 2015.
- [9] P. K. Pisharady y M. Saerbeck, «Recent methods and data bases in vision-based hand gesture recognition: A review,» *Computer Vision and Image Understanding*, vol. 141, pp. 152-165, 2015.
- [10] Real Academia Española, «Diccionario de la Lengua Española. Edición del Tricentenario,» 2018. [En línea]. Available: <https://dle.rae.es/?w=diccionario>. [Último acceso: 20 febrero 2019].
- [11] Organización de las Naciones Unidas, «Convención Internacional sobre los Derechos de las Personas con Discapacidad,» 3 mayo 2008. [En línea]. Available:



<https://www.un.org/disabilities/documents/convention/convoptprot-s.pdf>. [Último acceso: 3 junio 2017].

- [12] S. Liddell y J. Robert, «American Sign Language: The Phonological Base,» *Sign Language Studies*, vol. 64, pp. 195-277, 1989.
- [13] C. Valli y C. Lucas, *Linguistics of American Sign Language*, Washington D.C.: Gallaudet University Press, 2000.
- [14] J. Lapiak, «Hand Speak,» 2000. [En línea]. Available: <https://www.handspeak.com/>. [Último acceso: 2019 Febrero 19].
- [15] P. M. Palomares y J. V. Ciordia, «El Alfabeto Manual Adoptado por el Real Colegio de Sordo-mudos de Madrid, (1805-1814). Una laguna historiográfica resuelta,» *Revista Española de Pedagogía*, vol. 74, nº 263, pp. 149-166, 2016.
- [16] F. Ronchetti, L. Lanzarini y A. Rosete, «Reconocimiento de gestos dinámicos y su aplicación al lenguaje de señas (tesis doctoral),» Universidad de la Plata, Buenos Aires, Argentina, 2016.
- [17] E. A. López, «Reconocimiento Automático de lenguaje de signos: Lenguaje ASL,» Universitat de Barcelona, Barcelona, 2009.
- [18] M. A. Uddin y S. A. Chowdhury, «Hand sign language recognition for Bangla alphabet using Support Vector Machine,» de *2016 International Conference on Innovations in Science, Engineering and Technology (ICISSET)* , Dhaka, Bangladesh, 2016.
- [19] K. Silanon y N. Suvonvorn, «Fuzzy finger shapes and hand appearance features for Thai letter finger spelling,» de *2017 14th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, Phuket, Thailand, 2017.
- [20] Secretaría Nacional de Discapacidad (SENADIS), «Lengua de Señas Panameñas,» Panamá.
- [21] Ministerio de Economía y Finanzas, «Atlas Social de Panamá. Situación social de las personas con discapacidad en Panamá,» Panamá, 2010.
- [22] Contraloría General de la República, «Organización del Sistema Educativo Nacional,» [En línea]. Available: <https://www.contraloria.gob.pa/inec/Archivos/P1231Organizacion.pdf>. [Último acceso: 18 febrero 2019].
- [23] S. S. Rautaray y A. Agrawal, «Vision based hand gesture recognition for human computer interaction: a survey,» *Artificial Intelligence Review*, vol. 43, nº 1, pp. 1-54, 2015.

- [24] CyberGlove Systems Inc., [En línea]. Available: <http://www.cyberglovesystems.com/>. [Último acceso: 20 febrero 2019].
- [25] AnthroTronix, [En línea]. Available: <http://www.anthrotronix.com/>. [Último acceso: 20 febrero 2019].
- [26] Fifth Dimension Technologies, «5DT,» [En línea]. Available: <http://www.5dt.com/>. [Último acceso: 20 febrero 2019].
- [27] A. B. Jani, N. A. Kotak y A. K. Roy, «Sensor Based Hand Gesture Recognition System for English Alphabets Used in Sign Language of Deaf-Mute People,» de *2018 IEEE SENSORS*, New Delhi, India, 2018.
- [28] A. Samraj, N. Mehrdel y S. Sayeed, «Sign language communication and authentication using sensor fusion of hand glove and photometric signals,» de *2017 8th International Conference on Information Technology (ICIT)*, Amman, Jordan, 2017.
- [29] T. Starner, J. Weaver y A. Pentland, «Real-time American sign language recognition using desk and wearable computer based video,» *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, nº 12, pp. 1371-1375, 1998.
- [30] R. Feris, M. Turk, R. Raskar, K. Tan y G. Ohashi, «Exploiting Depth Discontinuities for Vision-Based Fingerspelling Recognition,» de *2004 Conference on Computer Vision and Pattern Recognition Workshop*, Washington, DC, USA, 2004.
- [31] L. Lamberti y F. Camastra, «Real-Time Hand Gesture Recognition Using a Color Glove,» de *In: Maino G., Foresti G.L. (eds) Image Analysis and Processing – ICIAP 2011. ICIAP 2011. Lecture Notes in Computer Science*, vol. 6978, Springer, Berlin, Heidelberg, 2011.
- [32] D. Ekiz, G. Ege Kaya, S. Buğur, S. Güler, B. Buz, B. Kosucu y B. Arnrich, «Sign sentence recognition with smart watches,» de *2017 25th Signal Processing and Communications Applications Conference (SIU)*, Antalya, Turkey, 2017.
- [33] Y. Ma, G. Zhou, S. Wang, H. Zhao y W. Jung, «SignFi: Sign Language Recognition Using WiFi,» *IMWUT*, vol. 2, pp. 23:1-23:21, 2018.
- [34] G. A. Rao, K. Syamala, P. V. V. Kishore y A. S. C. S. Sastry, «Deep convolutional neural networks for sign language recognition,» de *2018 Conference on Signal Processing And Communication Engineering Systems (SPACES)*, Vijayawada, India, 2018.
- [35] M. Kadous, «Machine recognition of Auslan signs using Power gloves towards large lexicon recognition of sign language,» Master's thesis, University of New South Wales, Computer Science and Engineering, 1996.

- [36] N. Soodtoetong y E. Gedkhaw, «The Efficiency of Sign Language Recognition using 3D Convolutional Neural Networks,» de *2018 15th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, Chiang Rai, Thailand, 2018.
- [37] J. Huang, W. Zhou, H. Li y W. Li, «Sign Language Recognition using 3D convolutional neural networks,» de *2015 IEEE International Conference on Multimedia and Expo (ICME)*, Turin, Italia, 2015.
- [38] K. Lai y S. N. Yanushkevich, «CNN+RNN Depth and Skeleton based Dynamic Hand Gesture Recognition,» de *2018 24th International Conference on Pattern Recognition (ICPR)*, Beijing, China, 2018.
- [39] S. Liu y Q. Xiaio, «A Signer-Independent Sign Language Recognition System Based on the Weighted KNN/HMM,» de *2015 7th International Conference on Intelligent Human-Machine Systems and Cybernetics*, Hangzhou, China, 2015.
- [40] I. Goodfellow, Y. Bengio y A. Courville, *Deep Learning*, Cambridge, Massachusetts: The MIT Press, 2017.
- [41] D. H. Hubel y T. N. Weisel, «Receptive fields of single neurones in the cat's striate cortex,» *The Journal of Physiology*, vol. 148, n° 3, pp. 574-591, 1959.
- [42] Y. LeCun y e. al., «Handwritten digit recognition: applications of neural network chips and automatic learning,» *IEEE Communications Magazine*, vol. 27, n° 11, pp. 41-46, 1989.
- [43] A. Krizhevsky, I. Sutskever y G. E. Hinton, «ImageNet Classification with Deep Convolutional Neural Networks,» *In Advances in neural information processing systems* 25, pp. 1097-1105, 2012.
- [44] K. He, X. Zhang, S. Ren y J. Sun, «Deep Residual Learning for Image Recognition,» Diciembre 2015. [En línea]. Available: <https://arxiv.org/abs/1512.03385>.
- [45] Y. LeCun, L. Bottou, Y. Bengio y P. Haffner, «Gradient-based learning applied to document recognition,» *Proceedings of the IEEE*, vol. 68, n° 11, pp. 2278 - 2324, 1998.
- [46] S. Hamidian, B. Sahiner, N. Petrick y A. Pezeshk, «3D Convolutional Neural Network for Automatic Detection of Lung Nodules in Chest CT.,» *Proceedings of SPIE--the International Society for Optical Engineering*, vol. 10134, n° 1013409, 2017.
- [47] M. Zeiler y R. Fergus, «Stochastic Pooling for Regularization of Deep Convolutional Neural Networks,» 2013. [En línea]. Available: <https://arxiv.org/abs/1301.3557>.

- [48] F. Nie, H. Zhanxuan y X. Li, «An investigation for loss functions widely used in machine learning,» *Communications in Information and Systems*, vol. 18, pp. 37-52, 2018.
- [49] Y. Srivastava, V. Murali y S. R. Dubey, «A Performance Comparison of Loss Functions for Deep Face Recognition,» 2019. [En línea]. Available: <https://arxiv.org/abs/1901.05903>.
- [50] J. Deng, J. Guo, N. Xue y S. Zafeiriou, «ArcFace: Additive Angular Margin Loss for Deep Face Recognition,» 2018. [En línea]. Available: <https://arxiv.org/abs/1801.07698>.
- [51] J. Patterson y A. Gibson, Deep Learning. A practitioner's approach, Sebastopol, CA: O'Reilly, 2017.
- [52] Stanford University, «CS231n: Convolutional Neural Networks for Visual Recognition,» [En línea]. Available: <http://cs231n.stanford.edu/>. [Último acceso: 28 mayo 2019].
- [53] J. Duchi, E. Hazan y Y. Singer, «Adaptive Subgradient Methods for Online Learning and Stochastic Optimization,» *Journal of Machine Learning Research*, vol. 12, pp. 2121-2159, 2011.
- [54] A. Wilson, R. Roelofs, M. Stern, N. Srebro y B. Recht, «The Marginal Value of Adaptive Gradient Methods in Machine Learning,» 23 mayo 2017. [En línea]. Available: <https://arxiv.org/abs/1705.08292>.
- [55] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever y R. Salakhutdinov, «Dropout: A Simple Way to Prevent Neural Networks from Overfitting,» *Journal of Machine Learning Research*, vol. 15, pp. 1929-1958, 2014.
- [56] J. Bengio, «Practical recommendations for gradient-based training of deep architectures,» 16 septiembre 2012. [En línea]. Available: <https://arxiv.org/abs/1206.5533>.
- [57] A. Hernández-García y P. König, «Data augmentation instead of explicit regularization,» 11 junio 2018. [En línea]. Available: <https://arxiv.org/abs/1806.03852>.
- [58] C. Nwankpa, W. Ijomah, A. Gachagan y S. Marshall, «Activation Functions: Comparison of trends in Practice and Research for Deep Learning,» 8 noviembre 2018. [En línea]. Available: <https://arxiv.org/abs/1811.03378>.
- [59] D. Pedamonti, «Comparison of non-linear activation functions for deep neural networks on MNIST classification task,» 8 abril 2018. [En línea]. Available: <https://arxiv.org/abs/1804.02763>.
- [60] Research Gate, «What is the minimum sample size required to train a deep learning model - CNN?,» 2 febrero 2016. [En línea]. Available:

[https://www.researchgate.net/post/What\\_is\\_the\\_minimum\\_sample\\_size\\_required\\_to\\_train\\_a\\_Deep\\_Learning\\_model-CNN](https://www.researchgate.net/post/What_is_the_minimum_sample_size_required_to_train_a_Deep_Learning_model-CNN). [Último acceso: 15 junio 2019].

- [61] K. He, X. Zhang, S. Ren y J. Sun, «Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification,» 6 febrero 2015. [En línea]. Available: <https://arxiv.org/abs/1502.01852>. [Último acceso: 17 mayo 2019].
- [62] F. Chollet, «Keras: The Python Deep Learning library,» 2018. [En línea]. Available: <https://keras.io/>. [Último acceso: 25 mayo 2019].
- [63] NVIDIA, «Procesamiento Paralelo CUDA,» [En línea]. Available: <https://www.nvidia.es/object/cuda-parallel-computing-es.html>. [Último acceso: 25 mayo 2019].
- [64] R. S. Pressman y J. M. Troya, Ingeniería del Software, 7ma ed., McGraw Hill, 1988.
- [65] Google LLC, «General Data Protection Regulation (GDPR),» mayo 2018. [En línea]. Available: [https://cloud.google.com/security/gdpr/resource-center/pdf/googlecloud\\_gdpr\\_whitepaper\\_618.pdf](https://cloud.google.com/security/gdpr/resource-center/pdf/googlecloud_gdpr_whitepaper_618.pdf). [Último acceso: 5 junio 2019].
- [66] S. C. Wong, A. Gatt, V. Stamatescu y M. D. McDonnell, «Understanding Data Augmentation for Classification: When to Warp?,» de *2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, Gold Coast, QLD, Australia, 2016.
- [67] G. Strezoski, D. Stojanovski, I. Dimitrovski y G. Madjarov, «Hand Gesture Recognition Using Deep Convolutional Neural Networks,» de *ICT Innovations 2016. Advances in Intelligent Systems and Computing*, 2016.
- [68] D. A. Forsyth, J. Pone, S. Mukherjee y A. K. Bhattacharjee, Computer Vision. A modern Approach, 2nd ed., Boston: Pearson, 2012.
- [69] J. Cervantes, F. Lamont, J. H. Santiago, J. Cabrera y A. Trueba, «Clasificación del Lenguaje de Señas Mexicano con SVM generando datos artificiales,» *rvin*, vol. 10, nº 1, pp. 328-341, 2013.
- [70] S. Thakur, R. Mehra y B. Prakash, «Vision based computer mouse control using hand gestures,» de *2015 International Conference on Soft Computing Techniques and Implementations (ICSCTI)*, Faridabad, 2015.
- [71] J. Janke, M. Castelli y A. Popovic, «Analysis of the proficiency of fully connected neural networks in the process of classifying digital images: Benchmark of different classification algorithms on high-level image features from convolutional layers,» *Expert Systems with Applications*, vol. 135, pp. 12-38, 2019.

