



UNIVERSIDAD DE BURGOS  
ESCUELA POLITÉCNICA SUPERIOR  
Grado en Ingeniería Informática



TFG del Grado en Ingeniería  
Informática

**Sign2Text - Transcripción de  
lenguaje de signos (a nivel de  
palabra) mediante DL**



Presentado por Iván Ruiz Gázquez  
en Universidad de Burgos — 29 de junio  
de 2022

Tutores: Dr. Daniel Urda Muñoz y Dr. Bruno  
Baruque Zanon





UNIVERSIDAD DE BURGOS  
ESCUELA POLITÉCNICA SUPERIOR  
Grado en Ingeniería Informática



D. nombre tutor, profesor del departamento de nombre departamento, área de nombre área.

Expone:

Que el alumno D. Iván Ruiz Gázquez, con DNI dni, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado título de TFG.

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 29 de junio de 2022

Vº. Bº. del Tutor:

Vº. Bº. del co-tutor:

D. nombre tutor

D. nombre co-tutor



## Resumen

En este primer apartado se hace una **breve** presentación del tema que se aborda en el proyecto.

## Descriptores

Palabras separadas por comas que identifiquen el contenido del proyecto Ej: servidor web, buscador de vuelos, android ...

## **Abstract**

A **brief** presentation of the topic addressed in the project.

## **Keywords**

keywords separated by commas.

---

# Índice general

---

---

## Índice de figuras

---



---

# Índice de tablas

---



---

# Introducción

---

Descripción del contenido del trabajo y del estructura de la memoria y del resto de materiales entregados.



---

# Objetivos del proyecto

---

La principal motivación del proyecto es:

**Mejorar la interacción entre personas que necesitan comunicación por signos y personas que no conocen el lenguaje de signos, intentando aumentar la calidad de vida de las primeras.**

## Objetivos teóricos

1. Creación de un modelo de DL<sup>1</sup> capaz de clasificar ASL<sup>2</sup> a nivel palabra.
2. Poner en producción un método para que los usuarios puedan probar el modelo de forma libre y transparente.
3. Generar una API de código abierto para que otros desarrolladores puedan crear plataformas de cliente, con el objeto de aumentar la audiencia y alcance del proyecto.
4. Liberar y contenedorizar el modelo para su libre distribución.
5. Creación de herramientas que permitan el tratamiento de datos y la transformación entre distintos formatos<sup>3</sup>.
6. Estudio del comportamiento, estructura e hiperparametrización de redes neuronales convolucionales.

---

<sup>1</sup>*Deep Learning*

<sup>2</sup>*American Sign Language*

<sup>3</sup>i.e.: extracción de fotogramas de un video, concatenación de imágenes

7. Aprender a fondo el uso de PyTorch[?], un *framework* para acelerar la creación y prototipado de redes neuronales.
8. Exportación del modelo entrenado a un formato estándar que facilite la compatibilidad con el mayor número de librerías en distintos lenguajes de programación.
9. Mantenimiento de una *codebase* que favorezca la integración continua (*linting*<sup>4</sup>, *formatting*<sup>5</sup> y *type-checking*<sup>6</sup>) de código, siguiendo así los estándares en contribuciones *Open Source*.

Estos objetivos representan en general la filosofía y los objetivos teóricos del proyecto. Si entramos más en detalle sobre los aspectos técnicos y las *features* que se esperan obtener al finalizar el proyecto, obtenemos los siguientes puntos:

### ***Features* esperadas**

- Debemos ser capaces de inferir resultados del modelo entrenado a tiempo real.
- Se desarrollará un modelo fácilmente escalable y adaptable a distintos dataset de cualquier tamaño.<sup>7</sup>
- Se realizará una fase de *data augmentation* sobre los datos iniciales.
- Se implementarán distintas redes neuronales para los distintos formatos de datos (imagen y video). Deben ser fácilmente refactorizables y mantenibles.
- Cada vez que se estudie el uso de un nuevo formato de dataset, se creará una nueva red, manteniendo la usabilidad de la anterior intacta.
- A lo largo de las pruebas y entrenamientos de la red, vamos a probar distintos *schedulers*, *optimizers* y *criteria*<sup>8</sup> buscando el que mejor se adecúe al formato de los datos y la estructura de la red.

---

<sup>4</sup>Voz ingl. Estudio de limpieza, orden, calidad y redundancia

<sup>5</sup>Formatear: Correcta estructura y legibilidad

<sup>6</sup>Comprobación de tipados de variables y funciones

<sup>7</sup>El formato de los datasets debe ser el mismo. Un modelo entrenado para clasificación de imágenes no podrá clasificar en formato video o audio.

<sup>8</sup>Funciones de cálculo de pérdida

- Se mantendrán unas estadísticas a tiempo real de la fase de *trainning* y *test* mediante el uso de **Tensorboard**, una herramienta analítica que permite mantener un *log* de imágenes generadas en la ejecución; así como gráficas de costes y *accuracies*.





---

# Conceptos teóricos

---

Este proyecto es un proyecto de *machine learning*. Dentro del *machine learning*, es de aprendizaje supervisado. Más concretamente intenta resolver un problema de clasificación mediante deep learning. Las redes neuronales aplicadas para la clasificación serán *ResNets* y *CNN* (*Convolutional Neural Networks*). Veamos en detalle todos estos terminos en los siguientes apartados.

## 3.1. Machine Learning

Según Tom Mitchell [?], un programa de ordenador aprende de una experiencia  $E$  con respecto a alguna tarea  $T$  si su medida de desempeño  $P$  (del inglés *performance*) mejora con  $E$ .

El uso del machine learning se origina en la búsqueda de soluciones a problemas que no podemos resolver programáticamente. Determinar las soluciones se centra en la recogida y análisis de datos con la finalidad de buscar relaciones entre ellos.

Para aclarar este concepto, veamos un ejemplo: Imaginemos que debemos estimar el precio de un coche usado pero no tenemos la fórmula exacta para valorar su estado. Lo que si sabemos es que el precio del coche aumenta y disminuye dependiendo de sus propiedades, como la marca, el kilometraje, o el desgaste. No sabemos tampoco cuales de las propiedades tiene más importancia. Sabemos seguro que cuantos más kilómetros tenga el coche, menor será su precio, pero no sabemos a que escala ocurre esto.

Para determinar la solución necesitamos estudiar el precio de coches en el mercado y anotar sus características. Estos datos son los que daremos a nuestro programa para que busque las relaciones entre propiedades e indique el precio óptimo de los coches.

Saliendo del ejemplo, el machine learning se divide en distintos tipos según la cantidad de datos (o supervisión) que otorgamos a nuestro programa.

## Unsupervised Learning

El *unsupervised learning* o aprendizaje no supervisado, tiene su principal característica, en que no necesita una muestra de datos para aprender.

[https://scholar.google.com/scholar?start=10q=unsupervised+learninghl=es&as\\_sdt=0,5](https://scholar.google.com/scholar?start=10q=unsupervised+learninghl=es&as_sdt=0,5)

<https://profs.info.uaic.ro/ciortuz/SLIDES/2017s/ml0.pdf>

<https://dl.matlabyar.com/mousa/Machine>

[https://books.google.es/books?hl=es&lr=id=2nQJEAAAQBAJoi=fndpg=PR7dq=mTqzBD2XTP8PaFW4EJ7SxJZUpZo8redir\\_esc=yv=onepageq=machine](https://books.google.es/books?hl=es&lr=id=2nQJEAAAQBAJoi=fndpg=PR7dq=mTqzBD2XTP8PaFW4EJ7SxJZUpZo8redir_esc=yv=onepageq=machine)

<https://beta.openai.com/playground>

## Semi-supervised Learning

## Supervised Learning

### Regression

### Classification

#### 1. Redes Neuronales

- *Perceptron*
- *Feedforward neural network*
- *Backpropagation*
- *Recurrent neural network*

- *Convolutional neural network*

- *Autoencoder*



## 3.2. Carga de datos

Extracción de frames de videos

## 3.3. Tratamiento de los datos

Normalización

*Data Augmentation*

Transformación de datos a tensores

*Batching*

*Subsampling*

Eliminación de datos innecesarios

## 3.4. Red Neuronal ResNet

## 3.5. Estructura de la red convolucional

Capas convolucionales

*Downsampling*

*MaxPooling*

*BatchNormalization*

*Flatten layer*

*Dense layers*

*Dropout*

*Lineal layers*

*Leaky ReLU*

*ReLU*

*Capas de salida*

Sigmoide

Softmax

## 3.6. Entrenamiento e hiperparametrización

*Pipeline* del proceo

*Optimizer*

*Scheduler*

### 3.9. Exportación del modelo

Formatos estándares

ONNX

### 3.10. Copresión del modelo

Consecuencias

Velocidad

*Accuracy*

Métodos de compresión

*Quantization*

*Prunning*

### 3.11. Estudio de escalabilidad del modelo

Vamos a comprobar hasta que punto la red neuronal es capaz de mantener la precisión de clasificación con el aumento de las etiquetas. Para todo lo anterior el número de etiquetas a clasificar han sido 8. Vamos a aumentar esto hasta 100 etiquetas.

En la siguiente tabla blabla

---

## Técnicas y herramientas

---

Esta parte de la memoria tiene como objetivo presentar las técnicas metodológicas y las herramientas de desarrollo que se han utilizado para llevar a cabo el proyecto. Si se han estudiado diferentes alternativas de metodologías, herramientas, bibliotecas se puede hacer un resumen de los aspectos más destacados de cada alternativa, incluyendo comparativas entre las distintas opciones y una justificación de las elecciones realizadas. No se pretende que este apartado se convierta en un capítulo de un libro dedicado a cada una de las alternativas, sino comentar los aspectos más destacados de cada opción, con un repaso somero a los fundamentos esenciales y referencias bibliográficas para que el lector pueda ampliar su conocimiento sobre el tema.





---

## Aspectos relevantes del desarrollo del proyecto

---

Este apartado pretende recoger los aspectos más interesantes del desarrollo del proyecto, comentados por los autores del mismo. Debe incluir desde la exposición del ciclo de vida utilizado, hasta los detalles de mayor relevancia de las fases de análisis, diseño e implementación. Se busca que no sea una mera operación de copiar y pegar diagramas y extractos del código fuente, sino que realmente se justifiquen los caminos de solución que se han tomado, especialmente aquellos que no sean triviales. Puede ser el lugar más adecuado para documentar los aspectos más interesantes del diseño y de la implementación, con un mayor hincapié en aspectos tales como el tipo de arquitectura elegido, los índices de las tablas de la base de datos, normalización y desnormalización, distribución en ficheros<sup>3</sup>, reglas de negocio dentro de las bases de datos (EDVHV GH GDWRV DFWLYDV), aspectos de desarrollo relacionados con el WWW... Este apartado, debe convertirse en el resumen de la experiencia práctica del proyecto, y por sí mismo justifica que la memoria se convierta en un documento útil, fuente de referencia para los autores, los tutores y futuros alumnos.



---

## **Trabajos relacionados**

---

Este apartado sería parecido a un estado del arte de una tesis o tesina. En un trabajo final grado no parece obligada su presencia, aunque se puede dejar a juicio del tutor el incluir un pequeño resumen comentado de los trabajos y proyectos ya realizados en el campo del proyecto en curso.



---

## **Conclusiones y Líneas de trabajo futuras**

---

Todo proyecto debe incluir las conclusiones que se derivan de su desarrollo. Éstas pueden ser de diferente índole, dependiendo de la tipología del proyecto, pero normalmente van a estar presentes un conjunto de conclusiones relacionadas con los resultados del proyecto y un conjunto de conclusiones técnicas. Además, resulta muy útil realizar un informe crítico indicando cómo se puede mejorar el proyecto, o cómo se puede continuar trabajando en la línea del proyecto realizado.