



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería
Informática**

**Visión artificial y aprendizaje
profundo para la estimación
de medidas en radiografías
dentales**



Presentado por Álgvar San Martín Liendo
en Universidad de Burgos
el 20 de septiembre de 2023

Tutores: Dr. César Ignacio García Osorio
José Miguel Ramírez Sanz



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



El Dr. César Ignacio García Osorio y José Miguel Ramírez Sanz, profesores del departamento de Ingeniería Informática, área de Lenguajes y Sistemas Informáticos

Exponen:

Que el alumno D. Álar San Martín Liendo, con DNI 22761906W, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado “Visión artificial y aprendizaje profundo para la estimación de medidas en radiografías dentales”.

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

En Burgos, 20 de septiembre de 2023

Visto bueno de los tutores:

Dr. César Ignacio García Osorio

José Miguel Ramírez Sanz

Resumen

La creciente accesibilidad a métodos de *Machine Learning* para el público general, entendido como estudiantes y profesionales sin una gran capacidad económica, posibilita que se puedan entrenar modelos sofisticados a un coste reducido. Sin embargo, el acceso a los datos que son necesarios para su entrenamiento sigue siendo un gran impedimento a la hora de conseguir resultados satisfactorios.

Este proyecto consiste en el desarrollo de un estudio y la base de una aplicación que permita facilitar el trabajo de los odontólogos mediante el uso de técnicas de visión artificial aplicadas al análisis de radiografías y la posterior estimación de la longitud del canal reticular de un diente sobre el que se pretende realizar una operación de *endodoncia*.

El trabajo actual parte de uno anterior desarrollado por un estudiante de la misma universidad que considera que el resultado no ha sido satisfactorio y las mediciones no son correctas con los datos con los que contaba [16]. Por lo tanto, se ha decidido revisarlo e intentar mejorarlo y ampliarlo ante la implicación del nuevo y más amplio conjunto de imágenes suministradas.

Descriptores

Machine Learning: también conocido como aprendizaje automático, es una rama de la inteligencia artificial que tiene por objetivo desarrollar programas que aprenden mediante un entrenamiento en el que se implica un conjunto de datos inicial.

Endodoncia: Operación que consiste en la extracción de la pulpa dentar, contenida en el canal reticular, donde se encuentra el nervio del diente y una serie de tejidos vivos, para ser sustituido por material inerte.

Abstract

Growing accessibility to machine learning methods for the general public, including students and professionals with limited financial means, makes it possible to train sophisticated models at a reduced cost. However, access to the data needed for their training remains a major obstacle to achieving satisfactory results.

This project consists of the development of a study and the basis of an application that will make it easier for dentists to work by using artificial intelligence techniques applied to the analysis of radiographs and the subsequent estimation of the length of the root canal of a tooth on which a root canal operation is to be performed.

The current work is based on previous work developed by a student at the same university who considers that the results have not been satisfactory and the measurements are not correct with the data available to them [16]. Therefore, it has been decided to review it and try to improve it and expand it in the face of the involvement of the new and wider set of images provided.

Keywords

Machine Learning: A branch of artificial intelligence that deals with the development of algorithms that can learn from data without being explicitly programmed.

Endodontics: A dental procedure that involves the removal of the pulp from a tooth. The pulp is the soft tissue inside the tooth that contains the nerves and blood vessels.

Índice general

Índice general	iii
Índice de figuras	v
Índice de tablas	vi
1. Introducción	1
1.1. Estructura	2
2. Objetivos del proyecto	5
2.1. Trabajo anterior	5
2.2. Objetivos	6
3. Conceptos teóricos	7
3.1. Endodoncia	7
3.2. Machine Learning	8
3.3. Visión Artificial	9
3.4. Aprendizaje Supervisado	9
3.5. Regresión	10
3.6. Redes Neuronales	10
3.7. Funciones de activación	11
3.8. Testeo	12
3.9. Función de pérdida	13
3.10. Redes Neuronales Residuales (ResNets)	14
4. Técnicas y herramientas	15
4.1. Técnicas	15

4.2. Herramientas	16
5. Aspectos relevantes del desarrollo del proyecto	21
5.1. Datos implicados	21
5.2. Segmentación	25
5.3. Regresión mediante CNN	27
6. Trabajos relacionados	39
6.1. Automatic wound detection and size estimation using deep learning algorithms	39
6.2. Deep Learning Measurement of Leg Length Discrepancy in Children Based on Radiographs	40
6.3. Use of the deep learning approach to measure alveolarbone level	40
6.4. Application of Deep Learning in Dentistry and Implantology	40
7. Conclusiones y Líneas de trabajo futuras	43
Bibliografía	45

Índice de figuras

3.1. Proceso de medición[33]	8
3.2. ANN simple y deep learning (figuras de elaboración propia)	11
5.1. Diente extraído (a) y diente en la mandíbula (b)	22
5.2. (a) Diente extraído; (b) diente en la mandíbula	23
5.3. Histograma de longitudes	24
5.4. Precisión de la clasificación (a) perdida total (b) a lo largo de un entrenamiento de 1 hora 30 minutos	26
5.5. Anotación incompleta (a) y anotación completa (b)	27
5.6. Segmentación de YOLO	28
5.7. Visualización del clasificador de media	30
5.8. Visualización del clasificador de regresión lineal	32
5.9. Tipos de recortes en los 3 distintos conjuntos de datos	33
5.10. Mascara del diente y la raíz	34
5.11. Fase de augmentation	34
5.12. MSE - Error medio cuadrático	35
5.13. Diagrama del proceso de segmentación y medición.	37

Índice de tablas

5.1. Datos con un clasificador de medias	29
5.2. Datos con modelo de regresión lineal R^2 medio = 0.63	31
5.3. Resultados incorrectos con early stopping	35
5.4. Resultados correctos, validación cruzada, 9 folds	35
5.5. Resultados con la máscara del diente y la raíz	36
5.6. Resumen de resultados	38

1. Introducción

La combinación de visión artificial y aprendizaje automático permite desarrollar herramientas que facilitan el trabajo de profesionales en situaciones que demandan una gran atención al detalle. Estas herramientas son especialmente útiles en casos en los que el ojo humano puede pasar por alto detalles imperceptibles, pero que pueden ser detectados por un sistema de visión artificial. El proceso de desarrollo de este tipo de software puede variar en complejidad, dependiendo de la calidad y naturaleza de los datos utilizados.

En el campo de la odontología, un número importante de las intervenciones dentales requieren el análisis de radiografías y la realización de estimaciones precisas. En caso de cometer errores en estas estimaciones, se podría poner en peligro la salud del paciente. Por ejemplo, en el caso de una endodoncia, durante el proceso de medición del canal radicular, si el dentista introduce el instrumento de medición fuera del diente y dentro de la mandíbula, podría afectar a un nervio, lo que resultaría extremadamente doloroso para el paciente.

Una herramienta que a partir de radiografías bidimensionales permita hacer una estimación inicial con un error menor de 1 mm no va a eliminar la necesidad de realizar una medición física sobre el diente del paciente, pero puede prevenir errores en los que el odontólogo haga una mala medición y provoque malestar en el paciente.

Los conceptos centrales del proyecto son *endodoncia*, *visión artificial* y *machine learning*.

1.1. Estructura

El trabajo está compuesto por 2 partes, la memoria y los anexos.

Memoria

La memoria está compuesta de los siguientes apartados:

1. **Introducción:** En esta sección se proporciona un contexto del problema al que se enfrentará el proyecto, así como una descripción de los documentos adjuntos y la estructura de cada uno de ellos.
2. **Objetivos del Proyecto:** Aquí se describen los objetivos generales y técnicos que se persiguen con el proyecto.
3. **Conceptos Teóricos:** Esta sección ofrece una explicación de los diferentes conceptos teóricos necesarios para comprender todo el proyecto.
4. **Técnicas y Herramientas:** Se presenta una breve descripción de todas las herramientas utilizadas y la metodología seguida a lo largo del proyecto.
5. **Aspectos Relevantes del Desarrollo del Proyecto:** En esta parte se exponen los aspectos más importantes surgidos durante el desarrollo del proyecto.
6. **Trabajos Relacionados:** Aquí se seleccionan algunos trabajos que están relacionados con la endodoncia y el aprendizaje profundo (*deep learning*).
7. **Conclusiones y Líneas de Trabajo Futuras:** Esta sección contiene las conclusiones finales, así como posibles mejoras o actualiza

Anexos

Además de la memoria se adjunta un documento de anexos con la información relativa al desarrollo práctico del trabajo, su planificación y sus componentes.

- A **Plan de proyecto:** En el primer anexo se encuentra la planificación inicial del proyecto, cuál ha sido la metodología de su desarrollo y si es viable en términos legales y económicos.

- B **Especificación de requisitos:** En este anexo se incluye una lista con los requisitos que debe cumplir el trabajo, así como el uso que los distintos usuarios va a hacer de este.
- C **Especificación de diseño:** Contiene los diseños de datos y estructuras de directorios que han sido utilizadas a lo largo del proyecto. También incluye diagramas de los procesos más importantes implicados.
- D **Documentación técnica de programación:** Este apartado está enfocado a los desarrolladores que vayan a trabajar sobre este proyecto, contiene todas las estructuras y herramientas necesarias para continuar el desarrollo del mismo modo que se planeó originalmente.
- E **Documentación de usuario:** Es el documento que se entregaría a los usuarios que vayan a usar el producto final. Se explica el funcionamiento del programa desde su perspectiva.

2. Objetivos del proyecto

En este apartado se explican los objetivos generales y técnicos que se buscan cumplir durante el desarrollo del proyecto. Además, se explica el contexto y origen del trabajo como continuación de uno previo.

2.1. Trabajo anterior

El trabajo anterior, que parte de la propuesta de un odontólogo, tenía como objetivo explorar la aplicación del *deep learning* en el campo de la odontología, desarrollar una aplicación web y ayudar en el proceso de obtención de medidas para la endodoncia con un error menor a 0.5 mm.

El apartado de segmentación del trabajo se desarrolló con éxito, sin embargo, en el apartado de medición no se contaba con la suficiente información, en forma de radiografías, como para que su resultado final se pudiese considerar satisfactorio. Además, cabe destacar que no se han publicado los resultados concretos, se menciona en la memoria el uso de *Intersection Over Union* como métrica para evaluar la segmentación del diente y la raíz, se muestra la precisión de la segmentación, pero no se dan detalles de cómo se han validado esos datos, la precisión de la medición sobre la segmentación no se muestra [16].

Por lo tanto, se asume el relevo de mejorar y documentar los procesos anteriores con el fin de obtener un modelo más preciso, además de conocer sus limitaciones. En segundo lugar, se tienen en cuenta las líneas de trabajo futuras planteadas que se incluyen en los objetivos de este nuevo proyecto.

2.2. Objetivos

Objetivos Generales

Los objetivos generales son aquellas metas que intentamos alcanzar como resultado de la finalización satisfactoria del trabajo.

1. Investigar la aplicación de técnicas de *machine learning* convencional y de *deep learning*, tanto para la segmentación de las radiografías como para la estimación de la medida de la raíz.
2. Elaborar un resumen con los resultados de cada modelo de tal forma que estos pueden ser evaluados de la forma más objetiva posible.
3. Minimizar el error obtenido en la medición a menos de 1.5 mm que es lo que se consideraría menor al error humano.
4. Crear una aplicación interactiva y sencilla para poder ser usada por un usuario sin conocimientos técnicos.

Objetivos Técnicos

Los objetivos técnicos son los que se tienen que completar para considerar que el proyecto realizado funciona correctamente y, por tanto, se puede dar por concluido.

1. Crear un *pipeline* y herramientas para el procesamiento de los datos nuevos que se van a utilizar en el proyecto y para los futuros que podrían ayudar a la mejora de este.
2. Mejorar el modelo de segmentación de tal forma que permita la clasificación de varias instancias de diente y raíz para una misma radiografía.
3. Aplicar técnicas de *machine learning*, no solo para la segmentación, sino también para la medición de la raíz en forma de modelos de regresión.
4. Aplicar técnicas de diseño escalable y mantenible para la elaboración de una aplicación web en la que se transfieran los resultados de las investigaciones para darles un uso práctico.

3. Conceptos teóricos

3.1. Endodoncia

Una *endodoncia* [39] es el vaciado del canal reticular, retirando la pulpa dental (vasos sanguíneos y nervios) para posteriormente ser rellenado con material inerte. Este tratamiento se realiza en casos extremos en los que por una infección grave de caries o por otros motivos menos comunes, se produce una inflamación de la pulpa que resulta en dolor y molestia para el paciente, pero todavía no se ha producido un daño irreversible y no es necesario retirar el diente al completo.

En este proyecto nos centraremos en los pasos preliminares de la operación, cuando se le hace una radiografía al paciente y el odontólogo decide que se va a proceder a hacer la endodoncia. En este punto se estimaría la profundidad del canal reticular a partir de la radiografía, ahora con el software obtenido del proyecto, después se introduciría el dispositivo de medición y finalmente se obtendría la medida exacta.

En resumen, nuestro software toma mediciones a partir de la 1ª radiografía, cuando todavía no se ha introducido el aparato de medición, y se dejaría al profesional realizar el ajuste de la segunda, con el instrumento dentro del diente.

Nos encontramos con la particularidad de trabajar sobre *radiografías* que difieren de las imágenes convencionales en que el sensor no recibe luz, y, por tanto, el reflejo del entorno, sino los rayos X al atravesar un cuerpo y su densidad. Resulta relevante ya que un objeto que solapa a otro puede no mostrarse si la densidad del primero es mayor.

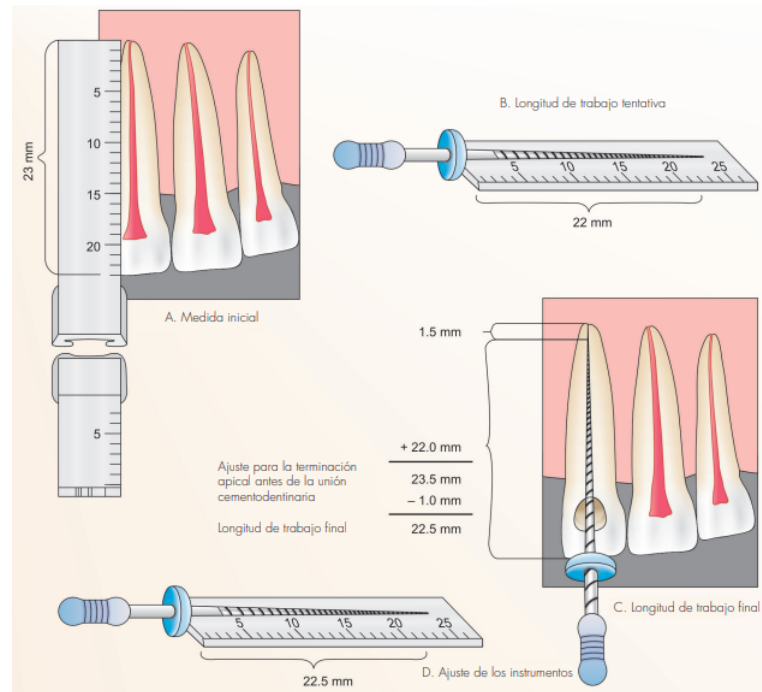


Figura 3.1: Proceso de medición[33]

3.2. Machine Learning

El *machine learning* o aprendizaje automático es una rama de la Inteligencia Artificial que consiste en que un programa informático aprenda con la experiencia, con los datos históricos o reconociendo patrones, en vez de programar su comportamiento únicamente con el conocimiento del programador. Para ello se desarrollan entrenamientos en los que se utilizan conjuntos de datos relacionados con el propósito del programa para que este encuentre patrones o relaciones en ellos.

Esta tecnología se utiliza en casos en los que desarrollar un algoritmo que resuelva el problema mediante programación tradicional es demasiado costoso porque los conjuntos de datos tienen demasiadas dimensiones o simplemente no se sabe como hacerlo. Este es el caso de la segmentación de las radiografías, donde identificar que es un diente o una raíz sería demasiado complicado para programarlo desde cero, pero es una tarea mucho más simple gracias al *machine learning*.

La principal característica del aprendizaje automático (*machine learning*) es que requiere una cantidad significativa de datos, con conjuntos lo suficientemente variados que representen a la población sobre la cual se desea

trabajar. En el caso de los dientes, idealmente se necesitarían radiografías de dientes en diferentes posiciones de la mandíbula, de diversos tamaños, de distintas edades, etc. Esto aseguraría que cuando se aplique el algoritmo a nuevas radiografías, estas tengan similitudes aprendidas con alguna de las imágenes del conjunto inicial [21, 41].

3.3. Visión Artificial

La *visión artificial* es una disciplina científica que se enfoca en el desarrollo de algoritmos y sistemas capaces de procesar, analizar e interpretar información visual del mundo real. Su objetivo principal es emular la capacidad del sistema visual humano para comprender y extraer conocimiento de las imágenes o secuencias de imágenes.

La visión artificial se basa en el procesamiento de imágenes digitales, utilizando técnicas y algoritmos avanzados para detectar patrones, reconocer objetos, segmentar regiones de interés y realizar diversas tareas de análisis visual. Estos sistemas son ampliamente utilizados en aplicaciones como reconocimiento facial, detección de objetos en tiempo real, seguimiento de objetos en vídeos, análisis médico y muchas otras áreas donde el procesamiento y comprensión de imágenes es fundamental.

En nuestro caso, aplicamos la visión artificial para la detección de objetos (dientes y raíces) mediante segmentación en radiografías [20, 40].

3.4. Aprendizaje Supervisado

El *Aprendizaje Supervisado*, subcategoría del *Machine Learning*, es un tipo de aprendizaje para modelos de *Machine Learning* en los que los datos que se utilizan han sido previamente etiquetados, de forma correcta, para obtener la salida deseada.

Este aprendizaje puede dividirse en dos tipos: la *clasificación*, donde se busca asignar una etiqueta a una entidad desconocida, y la *regresión*, donde se busca estudiar la dependencia de las variables y finalmente obtener una predicción en forma de valor continuo [42].

Existe una gran variedad de algoritmos que se pueden utilizar para el aprendizaje supervisado como *redes neuronales*, *regresión lineal*, *máquinas de vectores de soporte*... pero en este trabajo solo vamos a contemplar las *redes neuronales* y la *regresión lineal* dado que las primeras son lo suficientemente diversas como para ser flexibles en el desarrollo y la *regresión lineal* sirve como

referencia al ser el modelo estadístico más simple con el que se consiguen resultados [18].

3.5. Regresión

La *regresión* es un método estadístico en el que modelamos la relación entre variables independientes o características que son la entrada y variables dependientes o etiquetas que son la salida o predicción del modelo [42].

La regresión, además de ser útil para predecir valores continuos, nos proporciona información sobre la relación entre las características y la predicción. Esto resulta útil cuando conocemos las variables independientes de entrada, como ocurre en las técnicas de aprendizaje automático tradicionales y en las redes neuronales simples. En estos casos, el desarrollador es responsable de seleccionar dichas variables. Sin embargo, en las redes neuronales más complejas, como las redes neuronales convoluciones, existen capas dedicadas a la extracción de características y es más complicado encontrar las correlaciones de las características extraídas por la red [32].

3.6. Redes Neuronales

Las *redes neuronales* o redes neuronales artificiales (ANN) son un tipo de modelo de *machine learning* que basan su funcionamiento en el de las neuronas naturales y como se comunican entre ellas. Entendemos cada neurona como un modelo de regresión, con n entradas de neuronas anteriores y una salida transmitida a una o varias neuronas inferiores. Estos modelos de regresión contienen una función que se conoce como *función de activación* que deciden la salida de su neurona [19].

Las ANN están estructuradas en capas de nodos, que representan a las neuronas, y estas se comunican con las neuronas de otras capas inferiores, puede ser con la capa inmediatamente inferior o más profundas como en el caso de las redes neuronales residuales como *ResNet* [15].

Estos algoritmos tienen la particularidad de necesitar un “entrenamiento” para funcionar correctamente, para el que se necesita recopilar una cantidad importante de datos, estos datos se introducen en las ANN durante el proceso de entrenamiento.

Dentro de las redes neuronales encontramos la categoría de *deep learning* que se utiliza para referirse a las redes que tienen más de tres capas, capa de entrada, n capas ocultas y capa salida [19].

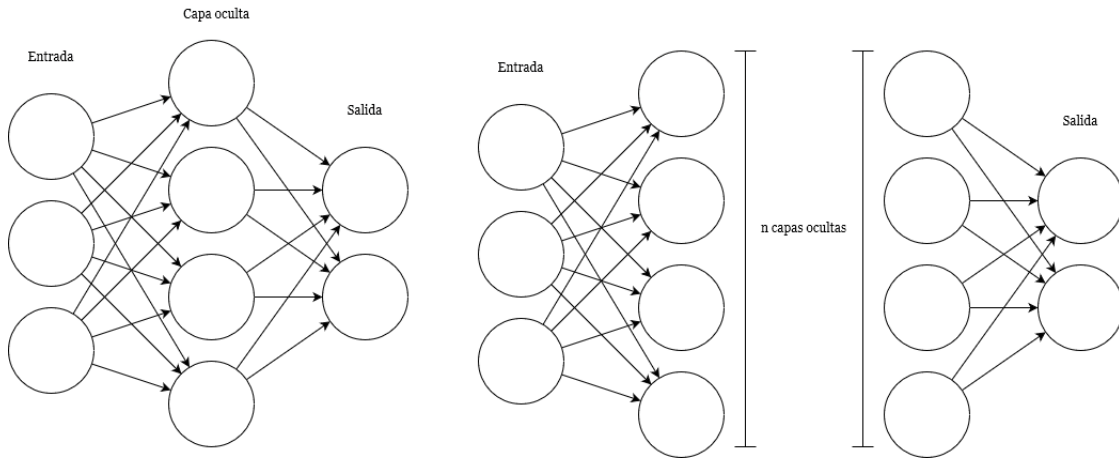


Figura 3.2: ANN simple y deep learning (figuras de elaboración propia)

3.7. Funciones de activación

Como se ha explicado en el punto de redes neuronales 3.6, cada neurona contiene una función que determina su salida con base en la suma de las entradas por sus pesos más una constante *bias*.

$$\text{entrada función} = \sum (\text{entrada} * \text{peso}) + \text{bias}$$

Existen diferentes tipos de funciones según su propósito, las más comunes son [2, 25]:

1. **Step**: representa un escalón y su salida es binaria, se activa si la entrada ha alcanzado un valor conocido como *threshold* o umbral.

$$f(x) = \begin{cases} 1 & \text{si } x \geq \text{umbral} \\ 0 & \text{si } x < \text{umbral} \end{cases}$$

2. **Linear**: es una función lineal normal, se aplica una transformación lineal a la variable de entrada y se le suma un bias.

$$f(x) = xA^T + \text{bias}$$

3. **ReLU**: representa la función lineal rectificadora, es decir, que la función será como una típica lineal, pero con un umbral mínimo que garantiza que la salida no será negativa.

$$f(x) = \max(0, x)$$

4. **Sigmoide:** el equivalente a la regresión logística, se suele utilizar para calcular la probabilidad de que la entrada de la red se corresponda con una clase.

$$f(x) = \frac{1}{1 + e^{-x}}$$

5. **Softmax:** aplica la función *softmax* a un tensor de entrada, es decir, su entrada no es un único valor como en las funciones anteriores, sino que la operación se hace sobre todas las entradas de la neurona y devuelve un tensor de salida donde todos sus elementos suman 1. Es útil para tareas de clasificación multi-clase.

$$Softmax(x) = \frac{\exp(x_i)}{\sum_j \exp(x_j)}$$

3.8. Testeo

El *Testeo* de un modelo es el proceso por el cual nos aseguramos que el rendimiento de un modelo va a ser aceptable en un entorno real. En el contexto del *machine learning*, no se puede realizar utilizando pruebas de caja blanca, ya que no es posible modelar los diversos caminos que se presentan en los algoritmos convencionales. Sin embargo, es importante determinar la precisión de nuestras salidas ante un conjunto de entradas utilizando pruebas de caja negra. Además, cuando encontramos un error en el modelo, la solución no es sencilla de encontrar, no podemos modificar directamente el modelo si no es a través de un proceso de entrenamiento nuevo.

Lo primero que necesitamos testear es la calidad de los datos, que no hay datos nulos, que los valores están entre los rangos correctos o que contamos con una distribución de datos igualada. Durante el entrenamiento encontramos la modificación de hiperparámetros para buscar un buen resultado y la validación cruzada para comprobar el rendimiento en conjuntos de entrenamiento y test distintos. Para el despliegue del modelo, como precaución final debemos asegurarnos que no vamos a encontrarnos datos radicalmente distintos a los usados para el entrenamiento, como trabajamos con radiografías esta no es una preocupación siempre que el modelo del aparato de rayos-X sea el mismo [17].

3.9. Función de pérdida

En aprendizaje automático, una función de pérdida es una función que mide la discrepancia entre la predicción de un modelo y la salida deseada. La función de pérdida se utiliza para entrenar un modelo, ya que el modelo se actualiza para minimizar el valor de salida de esta función[10].

Existen diferentes tipos de funciones de pérdida, cada una de las cuales tiene sus propias ventajas y desventajas. Las funciones de pérdida más comunes en el ámbito de la regresión:

- **Función de pérdida cuadrática media (MSE):** La función de pérdida MSE es la función de pérdida más utilizada. Es adecuada para problemas de regresión, ya que penaliza los errores de manera proporcional a su distancia al valor real.
- **Función de pérdida del valor absoluto (MAE):** La función de pérdida MAE es otra función de pérdida popular. Es adecuada para problemas de regresión, ya que penaliza los errores de manera proporcional a su magnitud.
- **Función de pérdida de Huber:** La función de pérdida de Huber es una combinación de la función de pérdida MSE y la función de pérdida MAE. Es más robusta a los *outliers* que la función de pérdida MSE.

La elección de la función de pérdida adecuada depende del problema específico que se esté abordando. En general, la función de pérdida MSE es una buena opción para problemas de regresión que no están afectados por *outliers*. La función de pérdida MAE es una buena opción para problemas de regresión que están afectados por *outliers*. La función de pérdida de Huber es una buena opción para problemas de regresión que pueden estar afectados por *outliers* [4].

En el contexto del trabajo actual, la función con mejores resultados ha sido la de Huber, usar el MSE o el MAE provoca sobreajuste en entrenamientos largos o con un *learning rate* muy bajo (< 0.001).

3.10. Redes Neuronales Residuales (ResNets)

Las redes neuronales residuales (ResNets) son un tipo de red neuronal artificial que utiliza conexiones de salto para conectar capas a través de múltiples niveles de profundidad. Estas conexiones permiten que la información fluya a través de la red de una manera más eficiente, lo que puede ayudar a evitar el problema del desvanecimiento de los gradientes.

Las ResNets fueron introducidas por He et al. en 2015. En su trabajo, demostraron que las ResNets podían entrenarse con una profundidad de hasta 152 capas, que era mucho mayor que la profundidad de las redes neuronales artificiales que se habían entrenado anteriormente.

Las ResNets se han utilizado con éxito en una variedad de tareas de aprendizaje automático, incluyendo clasificación de imágenes, reconocimiento de voz y traducción automática [26].

En cuanto a su aplicación en este proyecto, las ResNets han permitido entrenar redes complejas con una cantidad relativamente pequeña de datos sin producir sobreajuste.

4. Técnicas y herramientas

Esta sección tiene como objetivo presentar algunas de las técnicas y herramientas relevantes en el desarrollo del proyecto.

4.1. Técnicas

Las técnicas más importantes del proyecto las podemos separar en orientadas a la organización de este y orientadas al desarrollo [14].

Etapas del proyecto

1. **Análisis del problema:** Se ponen en claro los objetivos y se intenta hacer una selección inicial de herramientas y técnicas que van a ser útiles en la resolución. En este paso se decide dividir el problema en 2 partes, uno de segmentación y otra de regresión y que se va a usar aprendizaje supervisado.
2. **Obtención de datos:** Consiste en la búsqueda información para elaborar uno o varios conjuntos de datos para el entrenamiento de los modelos. Es el paso más complejo de realizar dada la naturaleza de los datos, existen conjuntos útiles disponibles al público para la segmentación, pero no para la regresión, pero se cuenta con el provisto por el odontólogo.
3. **Exploración de los datos:** Una vez obtenidos los datos para el proyecto se procede a analizarlos en busca de relaciones e inconsistencias, para ello se hacen visualizaciones y análisis en busca de datos nulos u *outliers* que puedan ser señal de errores en el conjunto.

4. **Preparación de los datos:** Se desarrollan las transformaciones que se van a aplicar a los datos antes de ser usados en el entrenamiento, se eliminan características que tienen poca relación con el objetivo o que no pueden ser limpiadas. Este punto es crítico cuando se trabaja con características en forma de columnas, pero cuando se trabaja con imágenes no hay tantas opciones a la hora de adaptar los datos, a excepción del tratamiento de imágenes, que es muy limitado en el ámbito de las radiografías.
5. **Selección del modelo y entrenamiento:** Una vez se tienen listos los datos se elige un modelo a entrenar, se parte el conjunto en 2 o 3 subconjuntos y se comprueban los resultados del entrenamiento hasta encontrar un modelo óptimo. Cuando se encuentra un grupo de modelos aceptable se procede a hacer un ajuste fino de los hiperparámetros para minimizar el error.
6. **Despliegue:** El resultado del entrenamiento en el modelo óptimo y los *pipelines* de transformación de datos se guardan para poder ser utilizado en la aplicación real. El despliegue se debe monitorizar en busca de cambios en los datos de entrada, si el entorno cambia, también lo hacen los datos y esto puede afectar a nuestros resultados. En estos casos debemos tener en cuenta que la máquina de rayos-x puede cambiar.

4.2. Herramientas

Se resumen las herramientas más importantes implicadas en el desarrollo.

Git

Git es un sistema de control de versiones distribuido ampliamente utilizado en el desarrollo de software. Permite realizar un seguimiento de los cambios en archivos y carpetas a lo largo del tiempo, lo que facilita la colaboración en proyectos entre múltiples desarrolladores, a pesar de que en este caso hay un solo desarrollador, por lo que su valor reside en el control de versiones.

Con *Git*, los cambios se registran en forma de *commits*, que representan instantáneas del estado de los archivos en un momento dado [13].

Python

Python es un lenguaje de programación de alto nivel que se ha utilizado a lo largo de todo el proyecto. *Python* es ideal para el *machine learning* debido a la cantidad de módulos desarrollados con este propósito, además de toda la documentación disponible y la sencillez de su uso. La gran mayoría de los trabajos encontrados similares a este estaban desarrollados en este lenguaje, existiendo una minoría realizados mediante *matlab* [36].

Anaconda

Anaconda es una plataforma dedicada a simplificar la gestión de paquetes y entornos para *Python* y *R* orientada al análisis de datos y al desarrollo. Sus características principales son su gestor de paquetes *Conda*, similar a *Pip*, que permite además crear entornos virtuales. En el proyecto se ha utilizado la versión *miniconda* que incluye el mínimo número de paquetes para funcionar [3].

Jupyter lab

Jupyter lab es un entorno de desarrollo integrado (IDE) orientado a Python, Julia y R que permite un desarrollo interactivo mediante la implementación de “celdas” que comparten el mismo contexto, pero que tienen una salida individual, de esta forma podemos comprobar salidas, dibujar gráficos o añadir notas en formato *markdown* entre los bloques de código. Jupyter lab es el entorno segunda generación que sucede a *Jupyter Notebook*, añadiendo utilidades como la gestión de pestañas, visualización del sistema de archivos o una interfaz para el gestor de paquetes *Conda* [22].

Librerías

Una de las principales ventajas de usar Python es su gran repertorio de librerías, a las que se puede acceder bien por medio de su instalador de paquetes por defecto, *Pip* o por medio de *Conda*.

- **Scikit-Learn:** es la biblioteca de referencia a la hora de llevar a cabo proyectos de análisis de datos y *machine learning* debido a su interfaz fácil de usar y la extensa documentación elaborada. contiene desde funciones para el tratamiento de datos, hasta un conjunto de los algoritmos de machine learning más comunes como modelos de regresión, árboles de decisión o máquinas de soporte vectorial, lo que

la hace indispensable en cualquier proyecto de aprendizaje automático [6, 14].

- **Pandas:** es una biblioteca orientada al análisis de datos que proporciona estructuras y herramientas para facilitar el manejo de datos. El punto más importante de *Pandas* es el concepto de *DataFrame* que representa una tabla de datos con columnas identificadas por un nombre donde se pueden almacenar datos de varios tipos, también introducidos por *Pandas*, para extender los básicos de *Python*. También introduce una interfaz para la lectura y escritura de estos datos desde fuentes externas, facilitando el acceso a datos desde ficheros *CSV*, *Excel* o incluso desde bases de datos *SQL* [31].
- **Pytorch:** en el campo de las redes neuronales nos encontramos con 3 librerías clave: Keras, Pytorch y TensorFlow, siendo la última la base sobre la que se han desarrollado las 2 primeras. Pytorch y Keras tienen el mismo propósito, facilitar una interfaz para la contracción de redes neuronales profundas, sin embargo, tienen dos enfoques distintos, Keras se enfoca en el alto nivel y el prototipado, mientras que Pytorch permite hacer operaciones de más bajo nivel y ofrece un mejor rendimiento con grandes volúmenes de datos [35, 8].
- **OpenCV:** es una biblioteca dedicada a la visión artificial de código abierto extremadamente popular en el tratamiento de imágenes y vídeo. Es compatible con la mayoría de librerías mencionadas previamente, además cuenta con una gran comunidad, lo que hace sencillo el encontrar recursos para el proyecto o encontrar soluciones ante problemas derivados de su uso [5].
- **PIL, Pillow:** Pillow, un *fork* de la librería anteriormente conocida como *PIL*, es una librería de visión artificial, muy similar a *OpenCV* que implementa funcionalidades para el trabajo con imágenes, es necesaria por compatibilidad con algunas transformaciones de imágenes en la etapa de *data augmentation* [9].

Detectron2

Detectron2 es una plataforma de código abierto impulsada por Facebook orientada a la segmentación y detección multi-clase de objetos en imágenes y vídeo. Usa principalmente redes *Mask R-CNN* y *Fast R-CNN*.

Detectron2 cuenta con su propio *model zoo*¹ donde encontramos modelos preentrenados a partir de los que podemos aprovechar técnicas de *transfer learning*, esto nos permite obtener más precisión con menos datos [27] un punto importante teniendo en cuenta que contamos con un número reduciendo de imágenes [37].

Tmux

Tmux [38] es lo que se conoce como un *multiplexador de terminales*, permite correr varios procesos en una sola terminal y lo más importante, permite mantener procesos activos aunque cerremos la terminal. Es especialmente útil en el desarrollo de entrenamientos largos, donde podemos cerrar la terminal base y esperar a que termine. En horas o días podemos volver a abrir la terminal y comprobar que el proceso ha finalizado.

Label Studio

Label Studio [1] es una herramienta *Open Source* para el etiquetado de información enfocada al aprendizaje automático. Está desarrollada en Python y su instalación es muy sencilla gracias a que se puede hacer mediante *Pip* en un servidor *Ubuntu*. Se puede acceder a la aplicación mediante cualquier navegador, lo que la hace flexible en el caso de que varias personas trabajen sobre el mismo proyecto.

Vue.js

Vue.js es un *Framework* basado en *JavaScript* enfocado a simplificar el desarrollo de interfaces de usuario reactivas, es decir, que los datos mostrados en pantalla cambien de forma asíncrona sin necesidad de recargar la página. El uso de *Vue.js* ha permitido desarrollar en muy poco tiempo una página web atractiva y ágil, ahorrando decenas de horas en comparación con un desarrollo básico con HTML, JavaScript y CSS.

FastApi

FastApi es un *framework* orientado al lado del servidor dirigido a desarrollar API con Python. Es sencillo de usar si se tienen conocimientos de *Flask* y tiene como ventaja la generación de documentación automática.

¹https://github.com/facebookresearch/detectron2/blob/main/MODEL_ZOO.md

SQAlchemy

SQAlchemy es un ORM de Python que permite interpretar las tablas de una base de datos relacional como si fuesen clases de Python. Entre las ventajas de su uso se encuentra la capa de seguridad que añade, elimina la necesidad de escribir consultas a mano y permite actualizar la estructura de la base de datos modificando las clases Python. El uso de este ORM ha supuesto una gran flexibilidad a la hora de interactuar con los datos, en especial gracias a que permite configurar herencias y se ha podido cambiar su estructura durante el desarrollo en cuestión de minutos.

ONNX

Open Neural Network Exchange (ONNX) es un formato de intercambio de modelos de aprendizaje automático. Es un formato abierto y estandarizado que permite a los desarrolladores de aprendizaje automático compartir modelos entre diferentes frameworks y plataformas. Este formato permite el entrenamiento, exportación e implementación de modelos independientemente del framework o la biblioteca con la que se creasen originalmente. Esta herramienta ha permitido generar una librería de modelos en la aplicación web que son compatibles sin que haga falta importar sus librerías específicas o mantener distintas extensiones de archivo.

5. Aspectos relevantes del desarrollo del proyecto

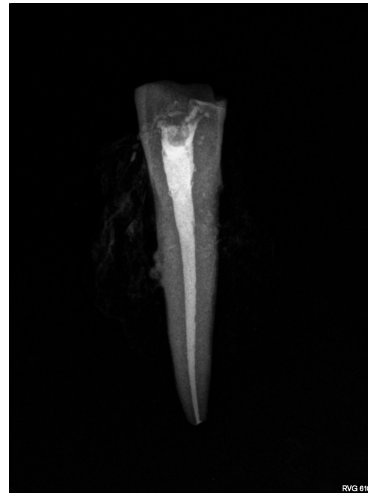
Este capítulo presenta los aspectos más relevantes del desarrollo del proyecto, así como un resumen de la parte teórica. Se destacan los puntos clave y las particularidades de los entrenamientos de los dos problemas planteados. Además, se realizará una comparación entre estos procedimientos y los resultados obtenidos en relación con el trabajo anterior.

5.1. Datos implicados

Anteriormente, contábamos con 20 imágenes, 10 de radiografías reales, de una máquina no especificada y de baja resolución (872×1148 píxeles), y 10 de dientes extraídos y aislados y 10 de dientes que todavía se encuentran en la mandíbula del paciente como se muestra en la Figura 5.1 [16].

Actualmente, hemos recibido 2 conjuntos de imágenes nuevos que vamos a añadir al proceso de entrenamiento, un conjunto de 125 radiografías de dientes extraídos similares al caso (a) de la Figura 5.1 y 54 radiografías reales de pacientes de alta resolución (1168×1562 píxeles), como se muestra en la Figura 5.2, sin duda el aumento de datos es considerable, pero seguimos por debajo de una cantidad ideal, con al menos 250 imágenes [37], que es una cantidad aceptable, pero pequeña en comparación con sets conocidos como *CIFAR-10* que tiene 60000 imágenes [28].

El segundo problema lo encontramos en una primera exploración de los datos, es que las anotaciones de la segmentación se encuentran en archivos formato *JSON* dentro de carpetas junto a cada imagen, es decir, la estructura de este conjunto es una carpeta que contiene sub-carpetas, cada una con



(a) Diente extraído

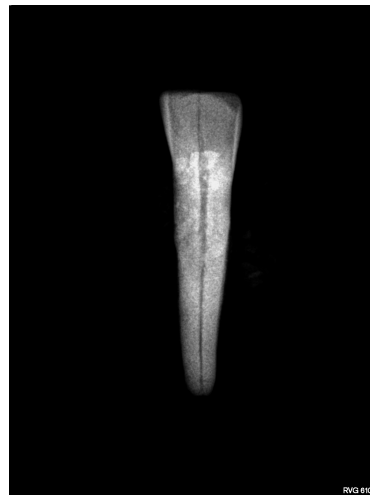


(b) Dientes en un contexto real

Figura 5.1: Diente extraído (a) y diente en la mandíbula (b)

una o dos imágenes y un archivo con la segmentación. La medición de estos dientes viene en una tabla embebida en un archivo *PDF* adjunto, que no es un formato sencillo con el que trabajar.

El primer paso del proyecto ha sido adaptar la estructura de carpetas y ficheros a una que pueda ser consumida por nuestros algoritmos. La estructura idónea para la mayoría de proyectos consiste en separar las



(a) Diente extraído



(b) Dientes en un contexto real

Figura 5.2: (a) Diente extraído; (b) diente en la mandíbula

imágenes en una carpeta aparte y agrupar las mediciones en un archivo estructurado, fácil de leer, que puede estar en formato CSV, JSON, YAML... o cualquier otro con el que estemos familiarizados. En nuestro caso se ha usado *JSON*, estructurado conforme al estándar del *dataset COCO* [30].

Además de las segmentaciones, debemos comprobar también las medidas de los dientes provistas por el dentista, una vez comprobado que no hay valores nulos y que estos se encuentran en el rango esperado, no hay dientes de menos de 10 mm o mayores de 30 mm, que aunque podrían aparecer en un contexto real, serían situaciones excepcionales. En el histograma 5.3

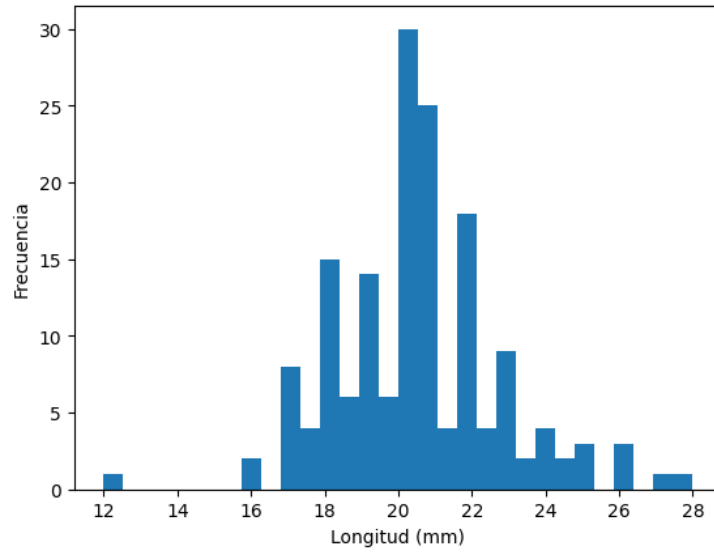


Figura 5.3: Histograma de longitudes

podemos comprobar que nos encontramos ante un conjunto de datos muy desbalanceado, con un pico muy notable entre los 20 mm y 21 mm y ninguna instancia entre 12 mm y 16 mm. Estos valores, 55 instancias de 162, se encuentran muy cercanos a la media, a tan solo 0.5 mm, por lo que si el modelo se ajusta a la media, tendremos 55 errores de 0.5 mm, lo que se debe tener en cuenta a la hora de evaluar los modelos. En trabajos donde se utilizan datos tabulares [43], se pueden imputar los datos que faltan, pero nosotros no podemos generar imágenes nuevas.

Aumento del número de datos del conjunto

La cantidad de datos sigue pareciendo escasa, en comparación con la usada en otros trabajos: [12] usa 1500 imágenes panorámicas de distintas resoluciones para la segmentación, con >95 % de precisión, pero también resalta la falta de detalle de este tipo de imágenes en comparación con las radiografías *intraorales*, que son precisamente las que usamos en este trabajo. También encontramos trabajos [23] que obtienen una puntuación *Dice* [45] del 91 %, precisión del 96 % usando únicamente 116 imágenes.

Se han explorado algunos de los repositorios de datos públicos más conocidos² en busca de nuevas imágenes para ampliar el conjunto inicial:

²Kaggle, Amazon AWS Datasets y UC Irvine Machine Learning Repository

- [Kaggle](#)
- [Amazon AWS Datasets](#)
- [UC Irvine Machine Learning Repository](#)

Sin embargo, es prácticamente imposible encontrar radiografías, principalmente debido a su naturaleza médica, un área donde los datos suelen ser estrictamente privados, siendo consultados únicamente por el médico y el paciente.

5.2. Segmentación

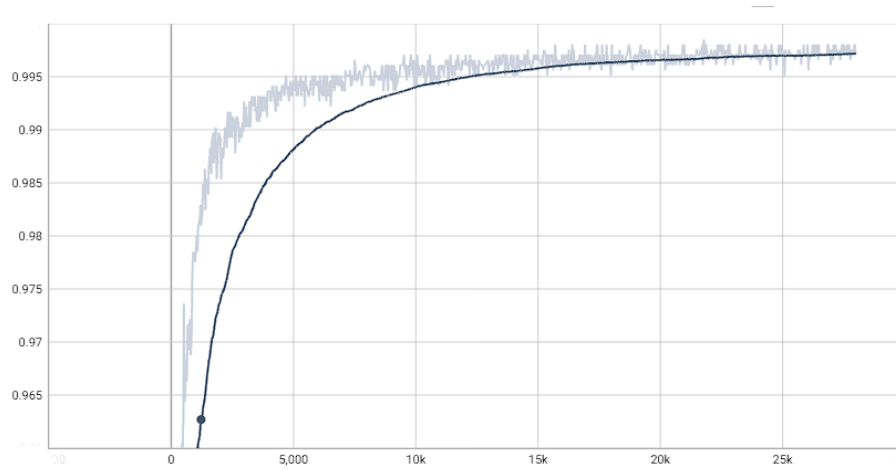
El problema de las anotaciones incompletas

Para resolver el problema de la segmentación, primero comprobamos que la anotación manual de las imágenes es correcta, el proceso se detallará en los anexos, pero lo primero que nos llama la atención es que, a pesar de contar con la segmentación correcta, cada radiografía cuenta solo con 2 instancias segmentadas, un diente y una raíz que se corresponden con el que se va a realizar la operación, como se muestra en la Figura 5.5. Esta característica supone un problema a la hora de la segmentación, ya que estamos introduciendo falsos negativos al modelo, que tiene como consecuencia que la decisión de los entrenamientos iniciales fuese cercano a 0 %.

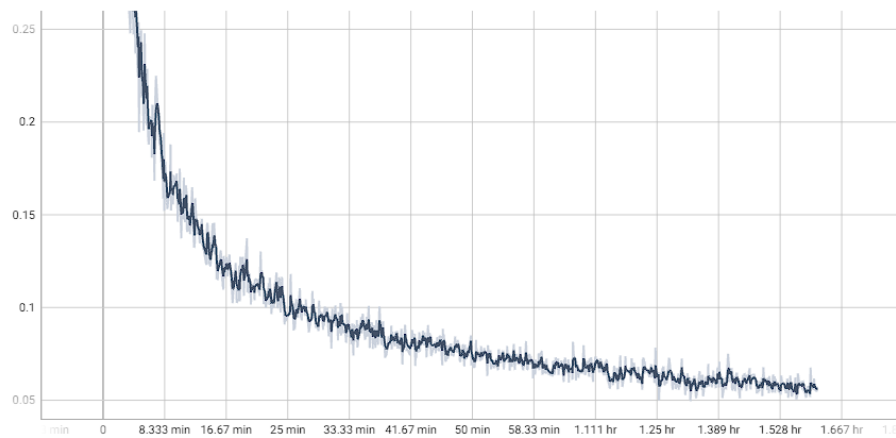
Para solventar este problema, se ha utilizado la herramienta *label-studio*, mencionada en el punto 4.2, para completar a mano las anotaciones sobre todos los dientes, de este modo eliminamos los falsos negativos y permitimos la segmentación de múltiples instancias en la misma imagen.

Resultados de la segmentación

A continuación, en la Figura 5.4 se recogen algunas de las métricas extraídas del entrenamiento de Detectron2. Estas métricas son extraídas del panel de control *TensorBoard*, que recoge los resultados del entrenamiento en tiempo real y lo representa en varios gráficos. Un detalle reseñable de la Figura 5.4 es que en el gráfico (b) se muestra la unión de todas las funciones de pérdida que se usan en las distintas tareas, por lo que se obtiene un valor relativo de la evolución del modelo.



(a)



(b)

Figura 5.4: Precisión de la clasificación (a) perdida total (b) a lo largo de un entrenamiento de 1 hora 30 minutos

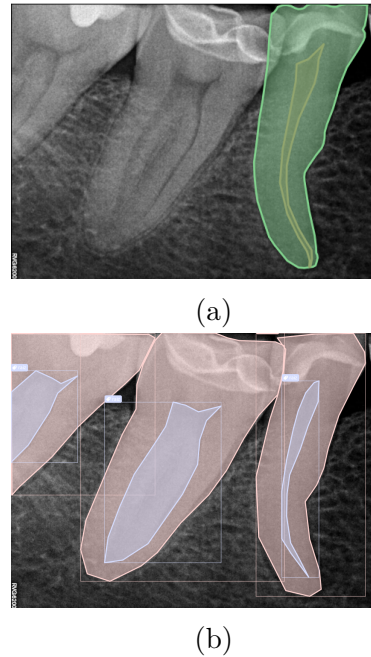


Figura 5.5: Anotación incompleta (a) y anotación completa (b)

Las diferencias entre la segmentación de YOLO y Detectron2

Después de haber desarrollado la mayoría del trabajo basándose en Detectron2 como modelo de segmentación, se decide comprobar el rendimiento de *YOLOv8* y ver si existe alguna diferencia reseñable. Lo primero que llama la atención una vez entrenado el modelo, al aplicarse sobre una imagen, es que la segmentación del diente no incluye la raíz, sino que esta aparece como una cavidad, como se muestra en la Figura 5.6. En consecuencia, los modelos de regresión que se apliquen más adelante en el trabajo no podrán ser los mismos que se apliquen con Detectron.

5.3. Regresión mediante CNN

En el trabajo anterior [16] solo se menciona que el error es mayor a 0.5 mm, pero no se especifica ningún valor concreto de los resultados. Se descarta la posibilidad de obtener resultados fiables mediante técnicas geométricas en las radiografías anteriores, ya que la longitud en píxeles no se corresponde al 100 % con la longitud real.

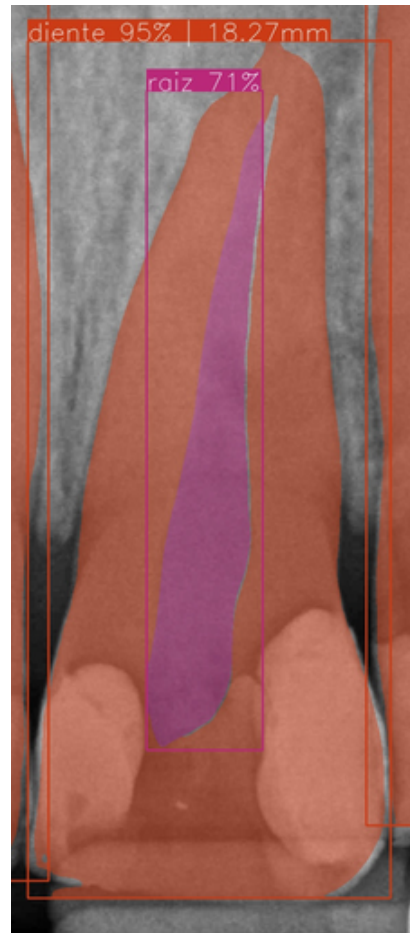


Figura 5.6: Segmentación de YOLO

Propuesta

Para intentar solventar la diferencia entre los tamaños de las imágenes y las medidas de los dientes, se plantea la hipótesis de que sería posible utilizar técnicas de *deep learning* enfocadas en la regresión que encuentren características relacionadas con la longitud del diente a partir de las cuales obtener mediciones más precisas.

Esta hipótesis se basa en que estamos trabajando sobre imágenes bidimensionales, pero el objeto que queremos medir es tridimensional, por lo que nos falta información de una dimensión. Por lo tanto, si el diente se inclina o curva sobre los ejes x e y , esto se va a reflejar en la radiografía, sin embargo, si la inclinación es sobre el eje z , del cual no tenemos información aparente, y el diente parecerá que está recto, provocando un error en la medición. Por estos motivos se teorizó sobre la posibilidad de usar *Redes*

Métrica	Etiqueta	Salida	Error (mm)	Error cuadrado
Instancias	162	162	162	162
Media	20.5	20.5	1.8	5.4
Desviación típica	2.3	0	1.5	9.4
Mínimo	12.0	20.5	0.0	0.0
25 %	19.0	20.5	0.5	0.2
50 %	22.0	20.5	1.5	2.2
75 %	22.0	20.5	2.5	6.3
Máximo	28.0	20.5	8.5	72.3

Tabla 5.1: Datos con un clasificador de medias

Neuronales Convolucionales que seleccionen *features* sobre las que hacer la regresión.

Baseline

Para comprobar que los resultados tienen sentido, y que realmente nuestro modelo no tiene *underfit*, calculamos la media de las medidas y analizamos los datos estadísticos obtenidos mostrados en la tabla 5.1.

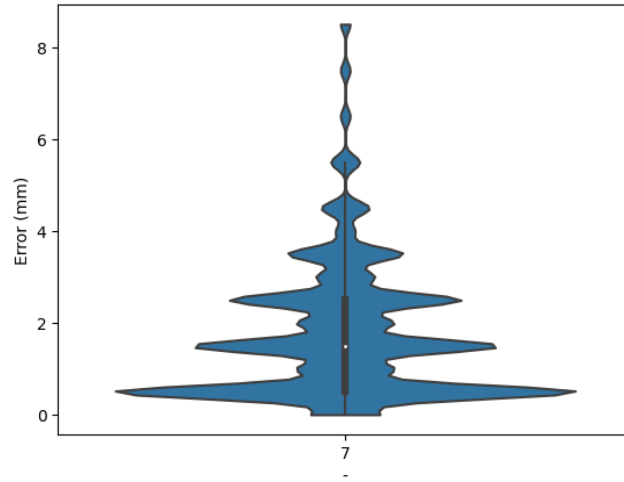
De estos datos podemos extraer las siguientes observaciones:

- Un clasificador que tenga un error medio > 1.8 mm tendrá un rendimiento peor que no usar ningún clasificador.
- Una desviación típica menor que 2.3 mm puede indicar que el modelo se está ajustando a la media.
- Al menos la mitad de los valores, que se encuentran entre los percentiles 25 % y 75 %, tienen una diferencia máxima de solo 1.5 mm, el dataset está desbalanceado.

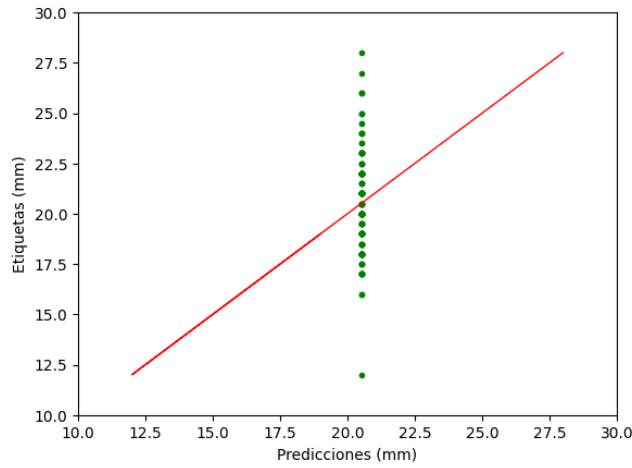
Si ahora lo representamos como un *violinplot* y *scatterplot*, para comprobar la distribución de los valores de error, mencionar que la correlación en este caso entre las predicciones y las etiquetas es 0 (tabla 5.7):

Modelo de regresión lineal

Para comprobar como de bueno sería un clasificador que se basase únicamente en 3 características fácilmente extraíbles, la *boundary box* de la



(a) Violinplot



(b) Scatterplot

Figura 5.7: Visualización del clasificador de media

segmentación, tanto anchura como altura y la posición del diente dentro de la mandíbula, hacemos un entrenamiento de un modelo de regresión lineal con validación cruzada, 6 folds, y analizamos los resultados:

Al contrario que en el trabajo anterior, en este caso podemos comprobar como sí que existe una relación lineal entre el tamaño de la imagen y las mediciones reales. Comprobado esto, pasamos a intentar mejorar los resultados mediante *redes neuronales convolucionales (CNN)*.

Métrica	Etiqueta	Salida	Error(mm)
Instancias	162.0	162.0	162.0
Media	20.5	20.5	1.1
Desviación típica	2.3	1.9	1.0
Mínimo	12.0	14.9	0.0
25 %	19.0	19.3	0.4
50 %	20.5	20.4	0.8
75 %	22.0	21.6	1.4
Máximo	28.0	26.1	6.3

Tabla 5.2: Datos con modelo de regresión lineal R^2 medio = 0.63

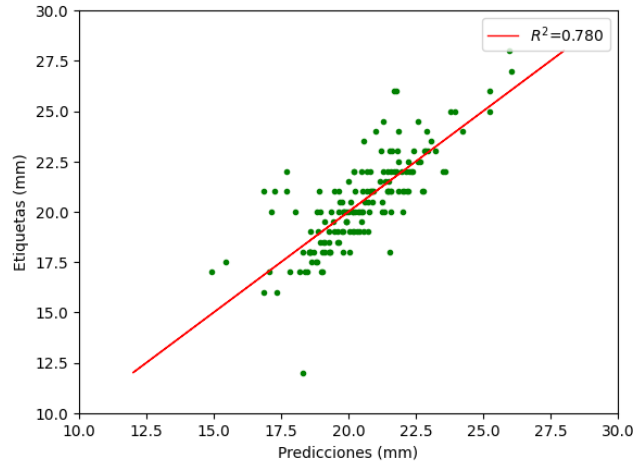
Preparación de los datos para las CNN

Antes de entrenar las redes neuronales, necesitamos estructurar los datos de tal forma que tengamos la certeza de estar midiendo únicamente un diente cada vez y que este mantiene las proporciones cuando se introduce en la red. Por lo tanto, tenemos que evitar cortar el diente o modificar su altura o anchura en píxeles, pero podemos rotarlo y voltearlo en el *data augmentation*.

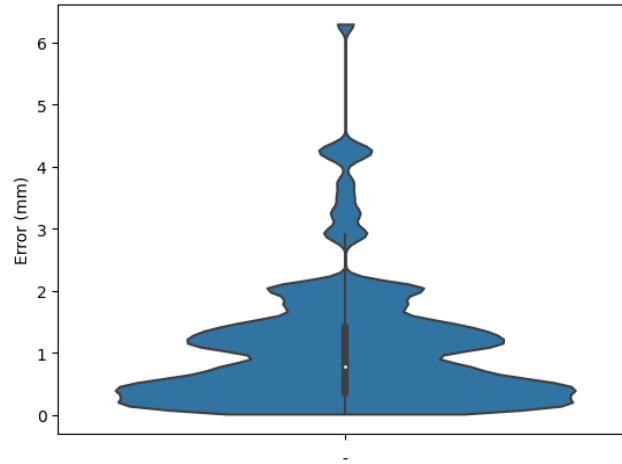
Siguiendo esta línea, planeamos la primera solución: recortar cada diente según su *boundary box* y generar un dataset con estos recortes. Este dataset se crea a partir de los recortes manuales del odontólogo para evaluar la precisión del modelo en una situación ideal, aunque posteriormente, los recortes serán provistos por el modelo de segmentación de forma automática.

más adelante, y buscando mejorar los resultados del dataset de los recortes de la *boundary box*, se generan 2 conjuntos más, uno recortando cada diente según la segmentación, hecha también a mano, y finalmente uno con las máscaras de cada diente, de tal forma que solo se puede apreciar el contorno, como se muestra en la Figura 5.9. Después de incluir YOLO como herramienta de segmentación, se genera un 4º conjunto con el contorno del diente y la raíz, mostrado en la Figura 5.10.

Una vez recortados los dientes, nos encontramos que estos tienen dimensiones distintas, y una *CNN* solo puede recibir imágenes de unas dimensiones concretas, pero si redimensionamos todas las imágenes al mismo tamaño, nos encontramos con que se pierden las proporciones y la equivalencia en píxeles, por lo que se decide crear un marco común, se inserta la imagen en el centro y el espacio sobrante se rellena, de esta forma se mantienen las dimensiones de los dientes y la imagen de entrada a la red es del mismo



(a) Scatterplot



(b) Violinplot

Figura 5.8: Visualización del clasificador de regresión lineal

tamaño siempre. En la Figura 5.11, junto que las modificaciones que se hacen a la imagen, podemos ver el resultado final de este proceso.

Para mejorar la precisión y evitar el *overfit* del modelo, se implementa un conjunto de transformaciones, únicamente para el conjunto de entrenamiento, que se aplican a cada imagen con cierta probabilidad en cada *epoch*, es decir, las rotaciones que se aprecian en la Figura 5.11 no son fijas, sino que se deciden cada vez que esa imagen pasa por el modelo. De este modo conseguimos una cantidad casi ilimitada de variaciones de las imágenes sin necesidad de almacenarlas en disco, lo que ocuparía demasiado

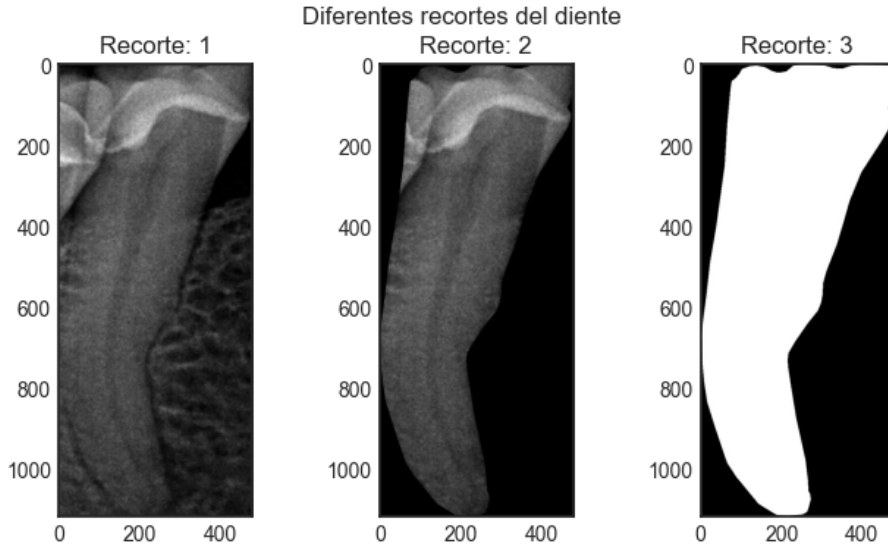


Figura 5.9: Tipos de recortes en los 3 distintos conjuntos de datos

espacio, además de este modo somos más flexibles a la hora de introducir transformaciones nuevas para comprobar su efecto.

Selección del modelo

Para seleccionar el modelo más adecuado, utilizamos la técnica de validación cruzada [44]. Esta técnica nos proporciona una visión más realista del rendimiento y nos ayuda a evitar el sobreajuste. Además, para determinar el mejor modelo en cada entrenamiento o *fold*, aplicamos una técnica conocida como *early stopping* [11]. En esta técnica, guardamos el modelo en el epoch en el que obtiene la menor pérdida de validación. Utilizamos el Error Cuadrático Medio (MSE) como métrica de evaluación, ya que penaliza de manera exponencial los errores más grandes, en comparación con los errores más pequeños [34]. Es importante destacar que el MSE tiene la ventaja de ser una métrica intuitiva y medir la discrepancia entre las predicciones y los valores reales, sin embargo, también debemos tener en cuenta que el MSE puede provocar que la red ignore algunas características importantes y sesgarse hacia la media [34].

Antes de aplicar validación cruzada, se comete un error de procedimiento y se considera como válido el modelo obtenido del *early stopping* lo que llevo a dar por buenos resultados excepcionalmente precisos en los que el modelo se ajustaba al conjunto de validación y aunque no tenía acceso a sus datos

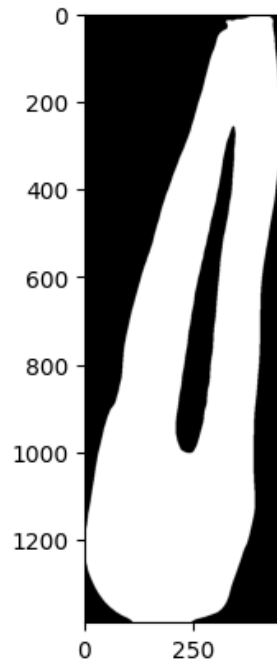


Figura 5.10: Mascara del diente y la raíz

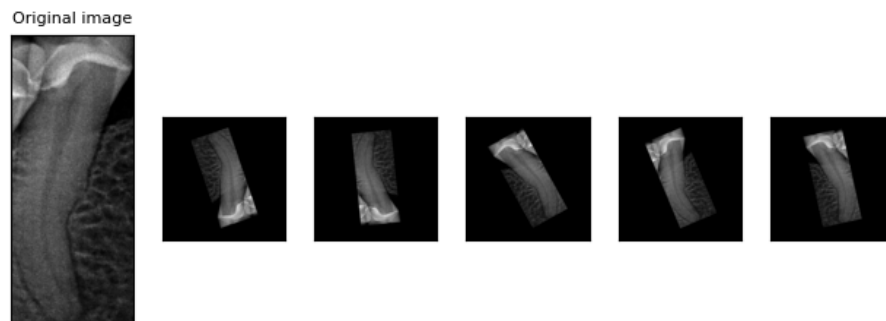


Figura 5.11: Fase de augmentation

de forma directa se estaba probando sobre ellos en cada *epoch*, se muestra en la Figura 5.3.

Una vez localizado este error, se realizan entrenamientos con distintos números de *epoch* y *learning rate* con el objetivo de comprobar el efecto de estos hiperparámetros en este problema.

$$MSE = \frac{1}{n} \sum_{i=1}^n (\hat{Y}_i - Y_i)^2$$

Figura 5.12: MSE - Error medio cuadrático

	Etiqueta	Salida	Error(mm)
Instancias	37.0	37.0	37.0
Media	20.0	20.0	0.5
Desviación típica	2.0	1.9	0.3
Mínimo	16.0	16.9	0.03
25 %	19.0	19.0	0.2
50 %	20.0	19.9	0.4
75 %	21.0	21.0	0.6
Máximo	27.0	27.4	1.1

Tabla 5.3: Resultados incorrectos con early stopping

	Etiqueta	Salida	Error (mm)	Error cuadrado
Instancias	162.0	162.0	162.0	162.0
Media	20.5	20.3	1.5	3.9
Desviación típica	2.3	1.4	1.2	7.0
Mínimo	12.0	16.2	0.0	0.0
25 %	19.0	19.4	0.6	0.4
50 %	20.5	20.0	1.1	1.3
75 %	22.0	21.1	2.0	4.0
Máximo	28.0	25.0	7.6	59.1

Tabla 5.4: Resultados correctos, validación cruzada, 9 folds

Resultados de la regresión por CNN

Después de haber realizado multitud de experimentos para buscar unos hiperparámetros que produzcan el mejor modelo posible, se obtienen los resultados plasmados en la Tabla 5.4. Estos resultados, obtenidos con el conjunto de datos elaborado con las máscaras de los dientes, como se muestra en la Figura 5.9, muestran un error demasiado alto, estando la mayoría por encima de 1 mm.

Cuando se revisa el proyecto y se experimenta con la segmentación de YOLO, se incorpora un recorte de máscara nuevo, donde a diferencia que en la Figura 5.9, se extrae la raíz para reflejar su contorno negativo dentro del diente. Aplicando el mismo entrenamiento que en el caso anterior se obtienen unos resultados ligeramente mejores, como se plasma en la Tabla 5.5.

	Etiqueta	Salida	Error(mm)	Error cuadrado
Instancias	162.0	162.0	162.0	162.0
Media	20.5	20.2	1.3	2.8
Desviación típica	2.2	1.7	1.1	4.9
Mínimo	16.0	16.7	0.1	0.0
25 %	19.0	18.8	0.4	0.2
50 %	20.5	19.9	1.0	1.0
75 %	22.0	21.5	1.9	3.45
Máximo	27.0	26.3	6.5	42.0

Tabla 5.5: Resultados con la máscara del diente y la raíz

Proceso completo

El resultado de unir todos los procesos mencionados a lo largo de esta sección resulta en una cadena de procesos que son invisibles al usuario, pero que generan un resultado visual y sencillo de comprender. Se muestra en detalle en la Figura 5.13

Comparativa de resultados

A continuación, en la Tabla 5.6, se resumen los resultados, en milímetros, de los modelos obtenidos a modo de comparativa, para facilitar la selección del modelo correcto. En negrita se resalta el mejor resultado de cada columna.

Sorprende comprobar que la regresión lineal ha rendido mejor que los métodos de *deep-learning*, lo que podría indicar, por una parte, que la cantidad de información es insuficiente para el entrenamiento de las redes neuronales o que el método usado es incorrecto, pero por el momento no se tienen datos suficientes para confirmar ninguna de las hipótesis.

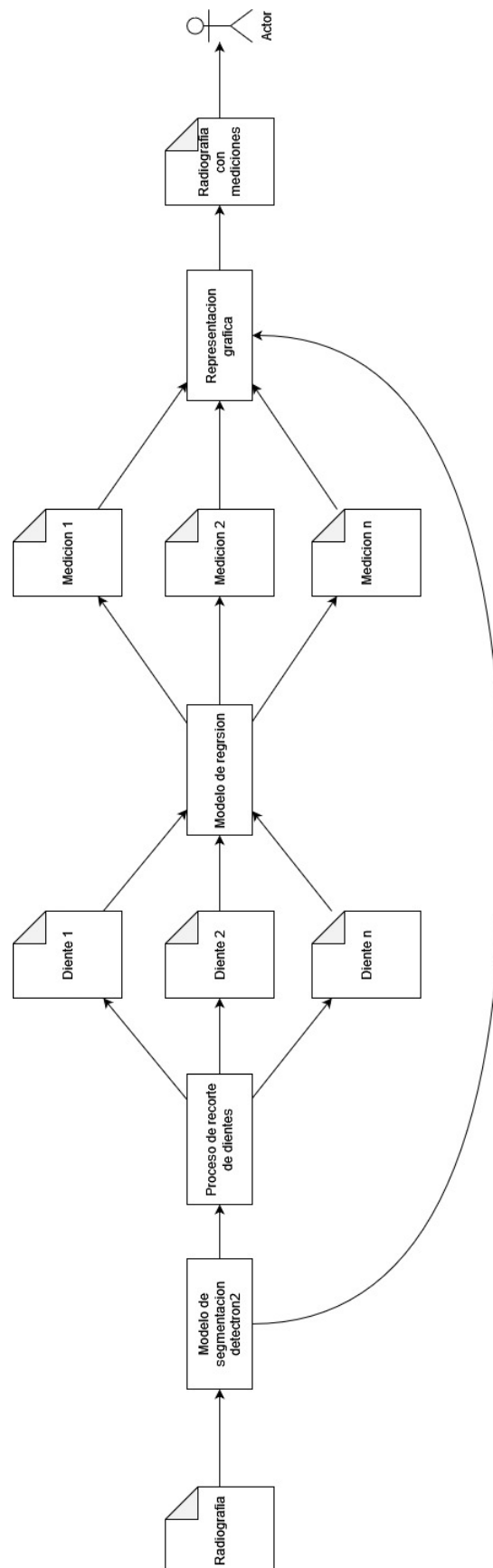


Figura 5.13: Diagrama del proceso de segmentación y medición.

Modelo	Error medio	Desviación típica	P 75	Máximo
Clasificador de medias	1.8	1.5	2.5	8.5
Regresión lineal	1.1	1.0	1.4	6.3
CNN con contorno de diente	1.5	1.2	2.0	7.6
CNN con contorno de diente y raíz	1.3	1.1	1.9	6.5

Tabla 5.6: Resumen de resultados

6. Trabajos relacionados

En este capítulo revisaremos brevemente los trabajos en los que nos hemos basado para llevar a cabo cada una de las etapas del trabajo.

6.1. Automatic wound detection and size estimation using deep learning algorithms

Carrion et al [7] exploran el uso de regresión por medio de *deep learning* para estimar el tamaño de las heridas con el objetivo de mejorar el tratamiento de estas. Los objetivos de este trabajo son los mismos que los nuestros, pero en distinto campo, facilitar tareas que otra forma serían complicadas o laboriosas.

El *pipeline* que planean es similar al nuestro, primero se detectan las partes interesantes de la imagen, luego se recortan y por último se hace la regresión.

Su conjunto de datos se compone de 256 imágenes de heridas en ratones de un diámetro de aproximadamente 10 mm, sin embargo, estas imágenes tienen un objeto de referencia común, pero puede estar dañado o rotado, lo que se considera uno de los retos de este trabajo.

En la fase de segmentación, utilizando Yolov3 alcanzan una precisión medida con la métrica *intersección sobre la unión* del 96 %. En la fase de regresión buscan estimar el área en el que la herida se está cicatrizando y obtienen un RMSE de $4mm^2$, lo que los autores consideran satisfactorio

6.2. Deep Learning Measurement of Leg Length Discrepancy in Children Based on Radiographs

Zheng et al en este artículo [46], de forma similar al anterior, buscan estimar la diferencia de longitud de las piernas en niños para ayudar a diagnosticar desórdenes muscular-esqueléticos.

El procedimiento que desarrolla consiste en 2 etapas de segmentación, en una primera, se separan las 2 piernas y en la segunda se segmentan los huesos. Una vez obtenidos los huesos, se hacen mediciones individuales de cada uno y se comparan los valores de cada pierna y de cada hueso para obtener la precisión. De este proceso se ha obtenido la idea de utilizar las máscaras para las mediciones y no el recorte de la radiografía como tal.

Se marcan como objetivo obtener un error medio absoluto menor que 5 cm y obtienen 0.5 cm de media, por lo que se considera un éxito. Además, compara el tiempo del software con el que tardaría un humano, siendo mucho más rápido el programa.

6.3. Use of the deep learning approach to measure alveolarbone level

Lee et al [29] en este caso, al igual que nosotros, trabajan sobre radiografías de dientes para ayudar en la diagnosis de enfermedades dentales. Se busca estimar la perdida de hueso en la mandíbula debido a la inflamación gingival. En el proceso se utilizan 693 imágenes de 37 pacientes

Una peculiaridad de este artículo es que superpone máscaras de varias partes del diente después de segmentarlas en la radiografía original, pero luego realiza la medición de forma geométrica a partir de los puntos obtenidos en las intersecciones de las máscaras.

6.4. Application of Deep Learning in Dentistry and Implantology

Kang et al [24] hacen un sumario de una serie de trabajos importantes en el área de la odontología, en su mayoría enfocados a la segmentación y clasificación para la diagnosis. Este artículo se usó de referencia a la hora de

buscar conjuntos de datos con los que ayudar a entrenar el modelo, pero se encontraron escasas imágenes útiles.

7. Conclusiones y Líneas de trabajo futuras

Se comenzó este trabajo partiendo de la asunción que las nuevas imágenes, por su número y calidad unidas a la implementación de métodos de *deep learning*, iban a ser suficientes para lograr una precisión satisfactoria para los estándares del odontólogo, mejorando los resultados respecto al método anterior de medición que era más inflexible.

Durante la primera parte se solventaron los problemas de sobreajuste que se arrastraban del etiquetado incompleto del que se partía sin demasiados inconvenientes.

En la segunda parte, donde aplicamos la regresión, es donde se han encontrado la mayoría de problemas. Primero se dieron como buenos unos resultados con sobreajuste, al caer en el error de no crear un set separado de test, después nos dimos cuenta de que las redes neuronales convolucionales no estaban ayudando en todo lo que pensábamos a la hora de hacer las mediciones y muchas veces se alcanzaba un subajuste a la media porque se ignoraban las características extraídas. A raíz de todos estos problemas y basándonos en otros trabajos, se llega a la conclusión de utilizar únicamente las mascarás, ignorando todas las posibles características que pensábamos se podían obtener de la imagen.

En un futuro sería deseable mejorar la precisión de las mediciones para alcanzar el objetivo de bajar el error máximo de 1 mm, aunque esta parece una tarea realmente complicada.

Bibliografía

- [1] Open Source Data Labeling | Label Studio. <https://labelstud.io/>, 2023. Accedido el 2023-06-27.
- [2] Pytorch layers. <https://pytorch.org/docs/stable/nn.html>, 2023. Accedido el 2023-06-25.
- [3] Anaconda. About anaconda. <https://www.anaconda.com/about-us>, 2023. Accedido el 2023-06-26.
- [4] ML Blogger. 14 Loss functions you can use for Regression - ML Blogger - Medium. 1 2023.
- [5] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.
- [6] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. API design for machine learning software: experiences from the scikit-learn project, 2013.
- [7] Héctor Carrión, Mohammad Jafari, Michelle D. Bagood, Hsin-Ling Yang, Roslyn Rivkah Isseroff, and Marcella M. Gomez. Automatic wound detection and size estimation using deep learning algorithms. *PLOS Computational Biology*, 18(3):e1009852, 3 2022.
- [8] Francois Chollet et al. Keras, 2015.
- [9] Alex Clark. Pillow (pil fork) documentation, 2015.

- [10] colaboradores de Wikipedia. Función de pérdida. https://es.wikipedia.org/wiki/Funci3n_de_p3rdida, 9 2023.
- [11] Wikipedia contributors. Early stopping. https://en.wikipedia.org/wiki/Early_stopping, 12 2022.
- [12] Mrinal Kanti Dhar. S-R2F2U-Net: A single-stage model for teeth segmentation. <https://arxiv.org/abs/2204.02939>, 4 2022.
- [13] Git. About git. <https://git-scm.com/about>, 2023. Accedido el 2023-06-26.
- [14] Aurélien Géron. *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly, 2019.
- [15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.
- [16] Ismael Franco Hernando. https://github.com/ifh1001/TFG_X-Teeth/blob/main/doc/latex/memoria.pdf, 2022. Accedido el 2023-06-20.
- [17] Joel Hodgson. Towards Data Science the 5 stages of machine learning validation. <https://towardsdatascience.com/the-5-stages-of-machine-learning-validation-162193f8e5db>, 2022. Accedido el 2023-06-25.
- [18] IBM. Aprendizaje supervisado - ibm. <https://www.ibm.com/es-es/topics/supervised-learning>. Accedido el 2023-06-21.
- [19] IBM. Redes neuronales - ibm. <https://www.ibm.com/es-es/topics/neural-networks>. Accedido el 2023-06-21.
- [20] IBM. Visión artificial - ibm. <https://www.ibm.com/es-es/topics/computer-vision>. Accedido el 2023-06-21.
- [21] IBM. <https://www.ibm.com/mx-es/analytics/machine-learning>, 2022. Accedido el 2023-06-21.
- [22] Jupyter. Jupyter. <https://jupyter.org/>, 2023. Accedido el 2023-06-26.
- [23] Jytjyt. GitHub - jytjyt05/Teeth-Seg: Teeth segmentation in 2D X-ray images using Unet and Res-Unet. <https://github.com/jytjyt05/Teeth-Seg>. Accedido el 2023-06-26.

- [24] Dae-Young Kang, Pham Duong, and Jung-Chul Park. Application of deep learning in dentistry and implantology. *The Korean Academy of Oral and Maxillofacial Implantology*, 24:148–181, 09 2020.
- [25] Nikhil Ketkar. *Deep learning with python*. Apress, 2017.
- [26] Muhammad Rizwan Khan. Residual Networks (ResNets) - towards data science. 3 2022.
- [27] Kraja. Object Detection with Detectron2 - Kraja - Medium. 12 2021.
- [28] Alex Krizhevsky. Learning Multiple Layers of Features from Tiny Images. <https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf>, 1 2009.
- [29] Chun-Teh Lee, Tanjida Kabir, Jiman Nelson, Sally Sheng, Hsiu-Wan Meng, Thomas E. Van Dyke, Muhammad F. Walji, Xiaoqian Jiang, and Shayan Shams. Use of the deep learning approach to measure alveolar bone level. *Journal of Clinical Periodontology*, 49(3):260–269, 12 2021.
- [30] Tsung-Yi Lin. Microsoft COCO: Common Objects in Context. <https://arxiv.org/abs/1405.0312>, 5 2014.
- [31] Wes McKinney et al. Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference*, volume 445, pages 51–56. Austin, TX, 2010.
- [32] Mayank Mishra. Towards Data Science convolutional neural networks, explained. <https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939>, 2020. Accedido el 2023-06-22.
- [33] Librería Servicio Médico. <http://www.libreriaserviciomedico.com/files/9676>, 2020. Accedido el 2023-06-20.
- [34] Aryansh Omay. The Disadvantage of MSE Loss and How to Remove Them. 12 2021.
- [35] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances*

- in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.
- [36] Python. About python. <https://www.python.org/about/>, 2023. Accedido el 2023-06-26.
 - [37] Saleh Shahinfar, Paul D. Meek, and Gregory Falzon. “How many images do I need?” Understanding how sample size per class affects deep learning model performance metrics for balanced designs in autonomous wildlife monitoring. *Ecological Informatics*, 57:101085, 5 2020.
 - [38] Tmux. Tmux. <https://github.com/tmux/tmux/wiki>. Accedido el 2023-06-25.
 - [39] Wikipedia. Endodoncia — wikipedia, la enciclopedia libre. <https://es.wikipedia.org/w/index.php?title=Endodoncia&oldid=140847047>, 2022. Accedido el 2022-1-10.
 - [40] Wikipedia. Visión artificial - wikipedia, la enciclopedia libre. https://es.wikipedia.org/wiki/Visi3n_artificial, 2022. Accedido el 2023-06-21.
 - [41] Wikipedia. https://es.wikipedia.org/wiki/Aprendizaje_autom3tico, 2023. Accedido el 2023-06-21.
 - [42] Wikipedia. Regression analysis - wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/Regression_analysis, 2023.
 - [43] Yuzhe Yang. Delving into Deep Imbalanced Regression. <https://arxiv.org/abs/2102.09554>, 2 2021.
 - [44] Luke A. Yates, Zach Aandahl, Shane A. Richards, and Barry W. Brook. Cross validation for model selection: A review with examples from ecology. *Ecological Monographs*, 93(1), 1 2023.
 - [45] Yerramvarun. Understanding dice coefficient. *Kaggle*, 12 2020.
 - [46] Qiang Zheng, Sphoorti Shellikeri, Hao Huang, Misun Hwang, and Raymond W. Sze. Deep Learning Measurement of Leg Length Discrepancy in Children Based on Radiographs. *Radiology*, 296(1):152–158, 7 2020.