



UNIVERSIDAD DE GRANADA

Facultad de Ciencias

GRADO EN FÍSICAS

TRABAJO FIN DE GRADO:

**SEGMENTACIÓN DE
IMÁGENES DENTALES Y
CARACTERIZACIÓN DE
LAS PROPIEDADES
ÓPTICAS DE LOS
INCISIVOS SUPERIORES**

Presentado por:
Andrés José Prado Vicente

Año académico 2022/2023

Abstract

The image segmentation neural networks allow the recognition and extraction of different objects from complete images. The objective of this work will be to train ours to extract the upper incisors of a complete set of teeth to carry out a study on them. This study will consist of recognizing the areas with different shades of white in the dental tissue. Its final would be could be to identify the exact materials with which to perform a dental restoration in case of loss or breakage of the tooth. The operation and precision of the neural network are correct, but the final results are far from reality. The main reason is the lighting of the images we will work with.

Resumen

Las redes neuronales de segmentación de imágenes permiten reconocer y extraer diferentes objetos de imágenes completas. El objetivo de este trabajo será entrenar a la nuestra para que extraiga los incisivos superiores de una dentadura completa con el fin de realizar un estudio sobre ellos. Este estudio consistirá en reconocer las zonas con distintos tonos de blanco en el tejido dental. Su uso final consistiría en identificar los materiales exactos con los que realizar una restauración dental en caso de pérdida o rotura del diente.

El funcionamiento y precisión de la red neural es correcto, pero los resultados finales distan de la realidad. El motivo principal es la iluminación de las imágenes con las que trabajaremos.

Índice

1	Introducción	3
2	Objetivos y metodología	6
2.1	Objetivos	6
2.2	Metodología	6
2.2.1	Preparación de las imágenes de trabajo	6
2.2.2	Red neuronal U-Net	7
2.2.3	Aumento de datos	7
2.2.4	Red neuronal pre-entrenada	8
2.2.5	Medida de precisión con k-fold CV y LOO	8
2.2.6	Espacio de color CIELAB	9
2.2.7	Métrica IoU	10
2.2.8	K-Means Clustering	10
2.2.9	Entornos de trabajo y librerías utilizadas	11
3	Resultados	12
3.1	Comparación de modelos	12
3.2	Medida de precisión del modelo pre-entrenado	14
3.3	Clustering de nuestro espacio de color dental	14
3.3.1	Estudio del color de los dientes	17
4	Conclusiones	18

1 Introducción

Durante decenas de millones de años, el cerebro de los seres humanos y de muchos otros animales ha evolucionado hasta ser capaz de cosas increíbles, entre ellas, la distinción de objetos. Esto ha sido un factor clave para la supervivencia de las especies. Ser capaz de diferenciar a un depredador de algo inofensivo o las setas comestibles de las venenosas, ofrece mucha ventaja a la hora de mantenerse con vida. ¿Pero cómo somos capaces de hacer estas diferenciaciones? ¿Cómo aprendemos? La respuesta está en las células que forman nuestro cerebro (las neuronas) y como interactúan entre sí.

Cuando una de ellas recibe un impulso, por ejemplo, a través del ojo, puede activarse o no, dependiendo del impulso que reciba. En caso de activarse, manda a su vez un impulso a algunas de las neuronas vecinas y estas hacen lo mismo una detrás de otra, todas interconectadas.

Hace unos 70 años se comenzó a intentar emular este proceso de manera artificial, primero con circuitos eléctricos y después con estructuras computacionales. A día de hoy, gracias al desarrollo de técnicas y tecnologías de los últimos años, las redes neuronales son la familia de algoritmos de *machine learning* más utilizadas.

Algunos de sus usos son:

- Reconocimiento de caracteres, imágenes y voz.
- Segmentación de objetos.
- Predicción bursátil.
- Traducción de idiomas.
- Conducción autónoma.
- Pronóstico de enfermedades.
- Clasificación de objetos.

Nosotros nos centraremos en la segmentación de objetos, pues es lo que trabajamos a lo largo del proyecto.

Al igual que nuestras neuronas, las redes neuronales se pueden entender como sucesivas capas de neuronas conectadas entre sí. La primera capa es la que recibe la información (input), así como las primeras neuronas que reciben la información que entra por los ojos cuando observamos algo. Cada neurona de esta capa puede interactuar con cada una de las neuronas de la siguiente capa, pero no todas lo hacen de la misma manera. Se puede representar como funciona una red neuronal con tan solo círculos y flechas (ver Figura 1).

Como podemos ver, hay conexiones (flechas) de cada n -neurona de la capa de entrada a las m -neuronas de la capa oculta (capa con la que no interactuaremos). En cada conexión entre neuronas hay dos variables, el peso y el sesgo. Esto hace que cada conexión pueda ser única.

El input a cada neurona de la capa de entrada será un número, x_i . Este número pasa a través de la neurona y es enviado a la siguiente, multiplicándolo por el valor que corresponda a su conexión (el peso, $\omega_{i,j}$). Cuando el producto de cada conexión llega a

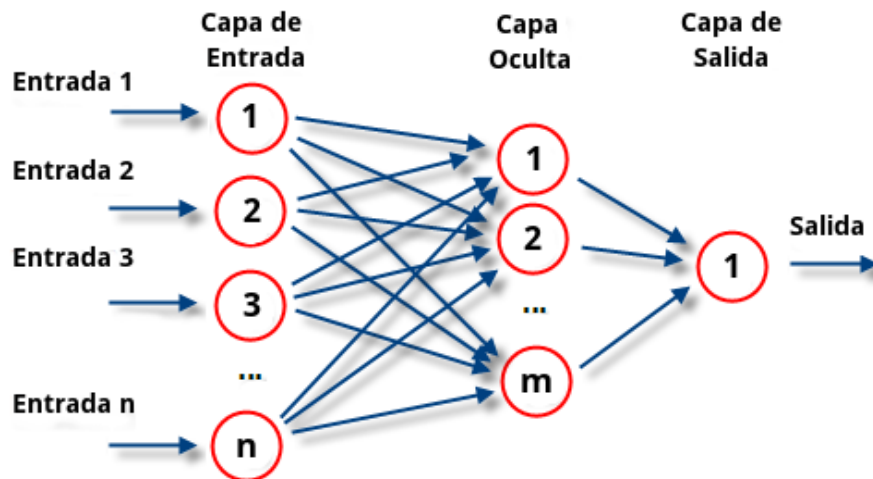


Figura 1: Ilustración esquemática del funcionamiento de una red neuronal simple. Imagen sacada de [1].

una de las neuronas de la capa oculta, se suman y se añade un valor específico para cada una de estas neuronas (el sesgo, b_j). Por ejemplo, para la k -neurona de la capa oculta, tenemos el siguiente valor:

$$f_k = \sum_i^n x_i \omega_{i,k} + b_k \quad (1.1)$$

A continuación, se repite el mismo proceso y llegamos al valor de la neurona de la capa de salida, el output de la red.

$$output = \sum_j^m f_j \omega_j + b \quad (1.2)$$

El caso anterior es una red regular y con una sola neurona en la capa de salida. Un modelo así podría ayudarnos, por ejemplo, a predecir el precio de una casa en función de las variables de entrada (superficie, número de habitaciones y baños, ubicación...), pero los modelos pueden volverse extremadamente complejos, con muchísimas neuronas y capas ocultas.

Una inmensa cantidad de problemas se pueden resolver con estos algoritmos, pero para ello es necesario conocer el valor específico de estos pesos y sesgos. Estos valores son completamente aleatorios al inicializar la red, ajustarlos correctamente es el problema al que la red se enfrenta de manera autónoma.

Para ello, las redes neuronales se pueden entrenar hasta calibrar sus pesos y sesgos tanto como la propia red permita para cada problema planteado (no todas las redes neuronales pueden resolver el mismo problema de manera eficiente). Para entrenarla, necesitaremos parejas de *inputs* y *outputs*, es decir, ejemplos para los que ya conocemos la solución. De esta manera, a través de sucesivas iteraciones (llamadas épocas), la red irá ajustando de manera no supervisada sus pesos y sesgos hasta conseguir resultados similares a los de los ejemplos.

Para algunos usos, como la segmentación de objetos, este tipo de redes neuronales es insuficiente y necesitamos recurrir a las redes neuronales **convolucionales**. Las redes

neuronales convolucionales siguen un proceso interno similar, pero con alguna diferencia.

En las redes neuronales regulares, somos nosotros quienes le decimos a la red que características debe tener en cuenta a la hora de hacer la predicción. Sin embargo, las convolucionales sólo necesitan el input (en nuestro caso una imagen) y son ellas mismas las que se encargan de extraer que características del input son relevantes y pasarle esa información a las neuronas de la capa oculta.

Así, la predicción de la red no depende de la posición u orientación de los objetos en la imagen, sino de las características que la componen. Por ejemplo, una red convolucional entrenada para identificar números, será capaz de diferenciarlos sin importar quien los escriba, pues habrá aprendido que las características importantes son el número de círculos y rectas que pueden diferenciarse en la imagen.

La estructura de una red convolucional puede esquematizarse de la siguiente manera:

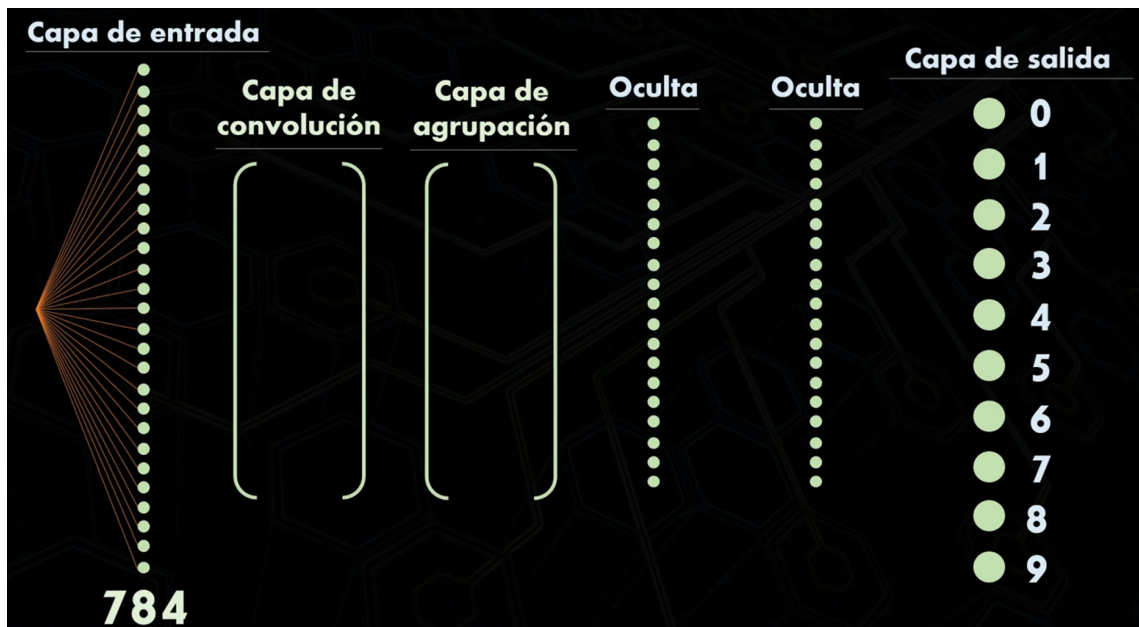


Figura 2: Esquema de una red neuronal convolucional. Foto tomada de [2].

2 Objetivos y metodología

2.1 Objetivos

Los objetivos de este estudio son:

- Analizar los datos disponibles, realizar tareas de visualización de datos que permitan comprender y abordar correctamente el desafío.
- Diseñar un algoritmo de segmentación de imágenes para identificar el tejido dental de incisivos en imágenes RGB haciendo uso de la librería TensorFlow.
- Diseñar un modelo de caracterización del mapa cromático dental haciendo uso del método K-Means.

2.2 Metodología

2.2.1 Preparación de las imágenes de trabajo

Para llevar a cabo este estudio se hace uso de un conjunto de imágenes procedente de la Universidad de Finlandia [3]. En el dataset encontramos imágenes de tipo *tif*. Las imágenes *.tif* son conjuntos de imágenes anidadas en un solo archivo. En este dataset hay dos conjuntos de estas imágenes:

- **Imágenes hiperespectrales:** Contienen la fotografía en RGB y en 204 diferentes longitudes de onda (blanco y negro).
- **Máscaras:** Distintas máscaras (imágenes binarias) que separan detalles de las imágenes hiperespectrales, como el tejido dental y los reflejos de luz en la imagen.

Para llevar a cabo el entrenamiento necesitaremos nuestras parejas de *inputs* y *outputs*. Nuestros *inputs* serán las imágenes RGB y nuestros *outputs* los incisivos superiores de la máscara dental.

Por otro lado, haremos uso de la máscara de reflejos para generar un mapa cromático con el que realizaremos un estudio de las imágenes obtenidas con la red neuronal.

Aquí podemos ver un ejemplo:

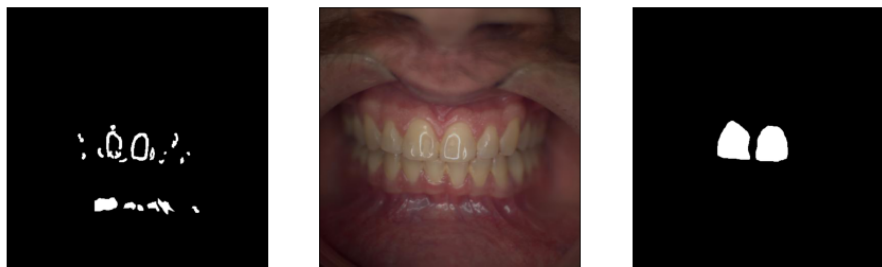


Figura 3: De izquierda a derecha: Máscara de reflejos. Imagen RGB. Incisivos recortados de la máscara dental.

2.2.2 Red neuronal U-Net

El modelo de red neuronal que usaremos para la segmentación será U-Net.

Este modelo queda perfectamente definido en [4] de la siguiente manera:

«U-NET es un modelo de red neuronal dedicado a tareas de visión artificial (Computer Vision) y más concretamente a problemas de segmentación semántica. Se trata de un modelo de red neuronal totalmente convolucional. Este modelo fue desarrollado originalmente por Olaf Ronneberger, Phillip Fischer y Thomas Brox en 2015 para la segmentación de imágenes médicas.

La arquitectura de U-NET consta de dos “vías”. La primera es la de la contracción, también llamada codificador. Se utiliza para captar el contexto de una imagen.

En realidad, se trata de un conjunto de capas de convolución y de capas de “max pooling” que permiten crear un mapa de características de una imagen y reducir su tamaño para disminuir el número de parámetros de la red.

La segunda vía es la de la expansión simétrica, también llamada decodificador.»

El trabajo de las capas de “max pooling” es reducir el tamaño de la imagen perservando sus características más significativas. De esta manera se reducen el número de píxeles, y por tanto, el de número de neuronas necesarias en las sucesivas capas manteniendo un buen resultado.

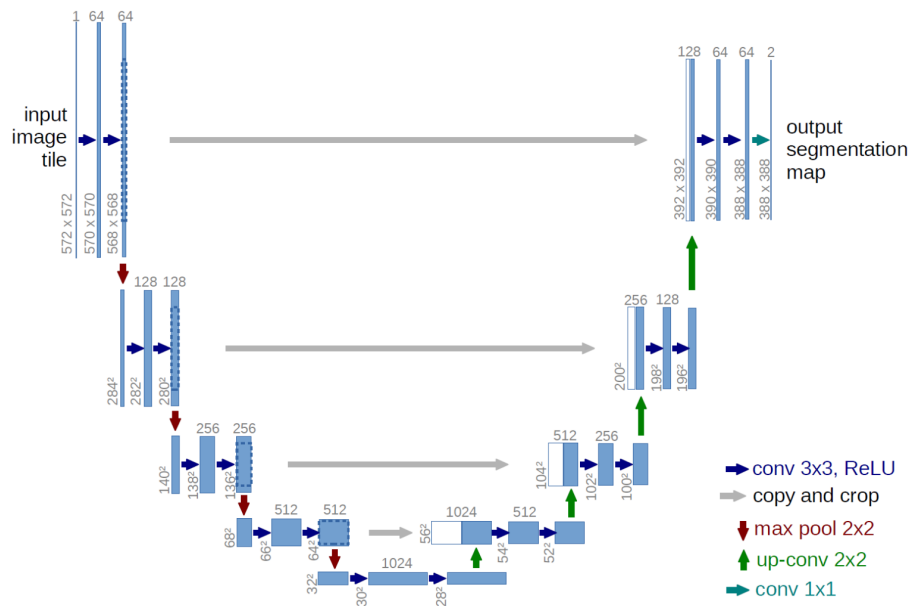


Figura 4: Estructura de la red neuronal U-Net. Imagen sacada de [4].

2.2.3 Aumento de datos

Uno de los problemas más comunes cuando se trabaja con conjuntos de imágenes pequeños (pocas decenas de imágenes, como es nuestro caso), es que la red neuronal “memoriza” estas pocas imágenes y ajusta sus pesos y sesgos de manera extremadamente precisa. Esto produce resultados muy pobres cuando se pasan imágenes externas al conjunto de entrenamiento por la red neuronal (a menos que sean muy parecidas a

alguna de ellas). Por norma general, este tipo de entrenamientos producen redes neuronales bastante inútiles, incapaces de predecir nada con precisión.

Este problema se soluciona mediante un aumento artificial en el número de imágenes de entrenamiento. El aumento de datos es una técnica muy utilizada en *machine learning* que consiste en realizar pequeñas transformaciones a los datos de entrenamiento, obteniendo imágenes nuevas con las que poder entrenar en cada una de las épocas. De esta manera, el modelo no ajusta los pesos y sesgos para únicamente un pequeño grupo de datos de entrenamiento, sino que se prepara para una mayor variedad de posibles *inputs*.

TensorFlow cuenta con una función que nos ayuda en esta tarea, *ImageDataGenerator*, la cual nos permite elegir el rango de giro, desplazamiento, zoom, etc, al que sometemos de manera aleatoria y conjunta a cada una de las parejas de entrenamiento.

Para aplicar aumento de datos en a los datos de entrenamiento es necesario separarlo en dos grupos a los que llamaremos: *training*, son los que la red usa para entrenar; y *validation*: con los que la red utiliza para testearse a sí misma y reajustar los pesos y sesgos. Esto también ocurre en los modelos sin aumento de datos, pero no es necesario separarlos con anterioridad.

2.2.4 Red neuronal pre-entrenada

Como hemos visto en 1 Introducción, el objetivo final del entrenamiento de las redes neuronales es ajustar sus pesos y sesgos de manera precisa. Una red preentrenada puede disminuir el número de épocas necesarias para conseguir un resultado óptimo, pues nos facilita una red con unos valores predefinidos por un entrenamiento previo con una gran base de datos.

El modelo U-Net pre-entrenado utilizado en este estudio proviene de la librería “segmentation_models” [5].

En el apartado 3.1 se lleva a cabo una comparación entre un modelo U-Net pre-entrenado con aumento de datos, un modelo U-Net nuevo (sin entrenamiento previo) y un modelo U-Net nuevo con aumento de datos.

2.2.5 Medida de precisión con k-fold CV y LOO

k-fold Cross-Validation es un método estadístico muy utilizado para estimar la calidad de los resultados de una red neuronal. Es un método simple y fácil de implementar que da estimaciones bastante precisas. La idea consiste en separar (de manera aleatoria) los datos de entrenamiento en k -grupos. Se utiliza un bucle **for** para realizar k entrenamientos del modelo. En cada uno de ellos, se entrena el modelo con $(k - 1)$ grupos, dejando uno de ellos fuera para testear la calidad de predicción del modelo. El grupo con el que se testea en cada iteración va rotando, de manera que cada grupo sirve como test una sola vez.

El método *Leave One Out* (LOO) sigue una idea similar al explicado anteriormente, pero en vez de dividir la muestra en grupos del mismo tamaño, lo hace separando una pareja de datos (imagen y máscara) del resto. De esta manera, mediante un bucle **for** con tantas iteraciones como número de parejas de entrenamiento, se consigue que todas ellas sean imágenes de test de manera individual una única vez.

Este método es muy utilizado cuando el número de datos de entrenamiento es bajo.

2.2.6 Espacio de color CIELAB

El espacio CIELAB, también conocido como $L^*a^*b^*$, es un espacio tridimensional donde se representan los colores de manera objetiva, así como en el espacio RGB al que tan acostumbrados estamos (se trata de un cambio de coordenadas).

“El espacio de color $L^*a^*b^*$ fue modelado en base a una teoría de color oponente que establece que dos colores no pueden ser rojo y verde (ni amarillo y azul) al mismo tiempo.” [6]

Las coordenadas en el espacio son las siguientes:

- L^* = Luminosidad.
- a^* = Coordenadas rojo/verde (+a rojo, -a verde).
- b^* = Coordenadas amarillo/azul (+b amarillo, -b azul).

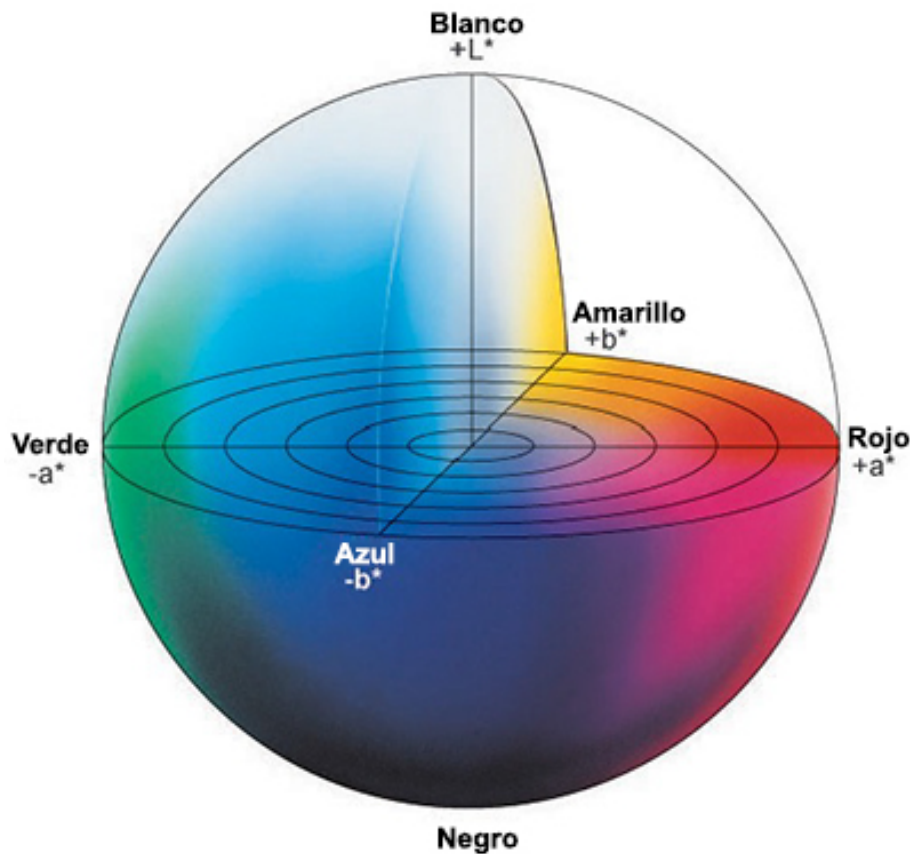


Figura 5: Representación tridimensional del espacio CIELAB. Imagen sacada de [6].

Mediante la intersección de ambas máscaras vistas en la Figura 3 obtendremos una máscara de los brillos de los dientes, la cual utilizaremos para extraer los tonos de blanco del tejido dental (Figure 6). Más adelante, representaremos estos tonos en el espacio CIELAB.



Figura 6: De izquierda a derecha: Intersección de las máscaras. Imagen en RGB. Extracción de dientes incisivos sin reflejos.

2.2.7 Métrica IoU

La métrica *Intersection over Union* es una medida utilizada para conocer la precisión de las predicciones de los modelos de segmentación.

La métrica consiste en comparar la Intersección de la máscara de salida de la red neuronal y la máscara esperada con la unión de ambas. En caso de ser exactamente iguales, el valor de IoU es 1.

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

Figura 7: Métrica IoU (Intersection over Union).

2.2.8 K-Means Clustering

Los algoritmos de *clustering* son algoritmos no supervisados que tienen como objetivo encontrar los centroides de un conjunto de coordenadas, en nuestro caso, los puntos que representan los tonos de blanco en el espacio CIELAB.

k-means es uno de los métodos más utilizados para llevar a cabo este tipo de estudios. Es capaz de separar las coordenadas introducidas en k -clústeres centradas en k -centros. Su objetivo es minimizar la suma de las distancias al cuadrado de todos los puntos y el

centro cada clúster.

$$J = \sum_{j=1}^k \sum_{i=1}^n ||x_i - c_j||^2 \quad (2.1)$$

donde k es el número de clústeres, n es el número de elementos, x_i son las coordenadas de cada elemento y c_j son las coordenadas de cada uno de los centroides. El proceso comienza como las redes neuronales, con valores aleatorios. A partir de sucesivas iteraciones los valores de los centros se van ajustando hasta que los mismos elementos quedan dentro de los mismos clúster durante dos iteraciones (o se llega a un valor máximo de iteraciones preestablecido).

Nosotros representaremos los píxeles de todo el tejido dental **sin reflejos** en el espacio L^*a^*b y utilizaremos K-means para calcular los centroides, es decir, los tonos de color más representativos de nuestro espacio de color dental.

2.2.9 Entornos de trabajo y librerías utilizadas

Para la realización de este trabajo se ha programado en dos entornos distintos:

- **Google Colab.** Visualización, análisis y tratamiento de imágenes. Extracción de las imágenes RGB y máscaras dentales de los archivos *.tif*.
- **Jupyter Notebook a través de Miniconda.** Creación de un entorno con las librerías de *python* necesarias. Entrenamiento de los modelos, extracción de píxeles con tejido dental y *clustering*.

Las librerías utilizadas a lo largo del código pueden consultarse en la Figura 8.

```
import os
import numpy as np
from libtiff import TIFF # Lectura de imágenes .tif
from skimage.io import imread, imshow # Para Leer y mostrar imágenes
from skimage.transform import resize # Para modificar el tamaño de las imágenes de entrenamiento
from skimage import color # Para pasar del espacio RGB al L*a*b
import matplotlib.pyplot as plt # Mostrar gráficos
import datetime # Para la creación de número aleatorios junto con la librería random
import cv2
import tensorflow as tf # Para definir la red neuronal sin entrenamiento previo
import random
import segmentation_models as sm # De aquí sacaremos una red neuronal pre-entrenada
from PIL import Image
from mpl_toolkits.mplot3d import Axes3D # Visualización del espacio L*a*b
from matplotlib import pyplot
from tensorflow.keras.preprocessing.image import ImageDataGenerator # Aumento de datos
```

Figura 8: Librerías utilizadas.

El código en cuestión y las imágenes utilizadas puede encontrarse en Github [7].

3 Resultados

Como hemos visto en alguna de las figuras anteriores, no trabajaremos con todos los incisivos, sino que no limitaremos al estudio de los incisivos centrales superiores. Esto se debe al poco número de imágenes de entrenamiento a nuestra disposición (un total de 25 imágenes útiles). Incluir más dientes haría que los resultados fuesen mucho menos precisos.

De esta manera tendremos mejores segmentaciones con las que trabajar mientras que mantendremos la dificultad del problema, pues el único cambio es recortar más o menos dientes de las máscaras dentales que usaremos para los entrenamientos.

3.1 Comparación de modelos

Con el fin de elegir con que modelo relizar la segmentación de los incisivos, decidimos comparar el entrenamiento de 3 modelos distintos. Por un lado, un modelo de la red neuronal U-Net sin pre entreno; por otro, uno idéntico añadiendo aumento de datos (AD); y por último, uno pre entrenado añadiendo también aumento de datos.

En la Figura 9 podemos observar el valor de la métrica IoU para cada uno de los modelos a medida que avanzan las épocas. Se muestra tanto el valor de IoU para el conjunto de *training* como para el conjunto de validación.

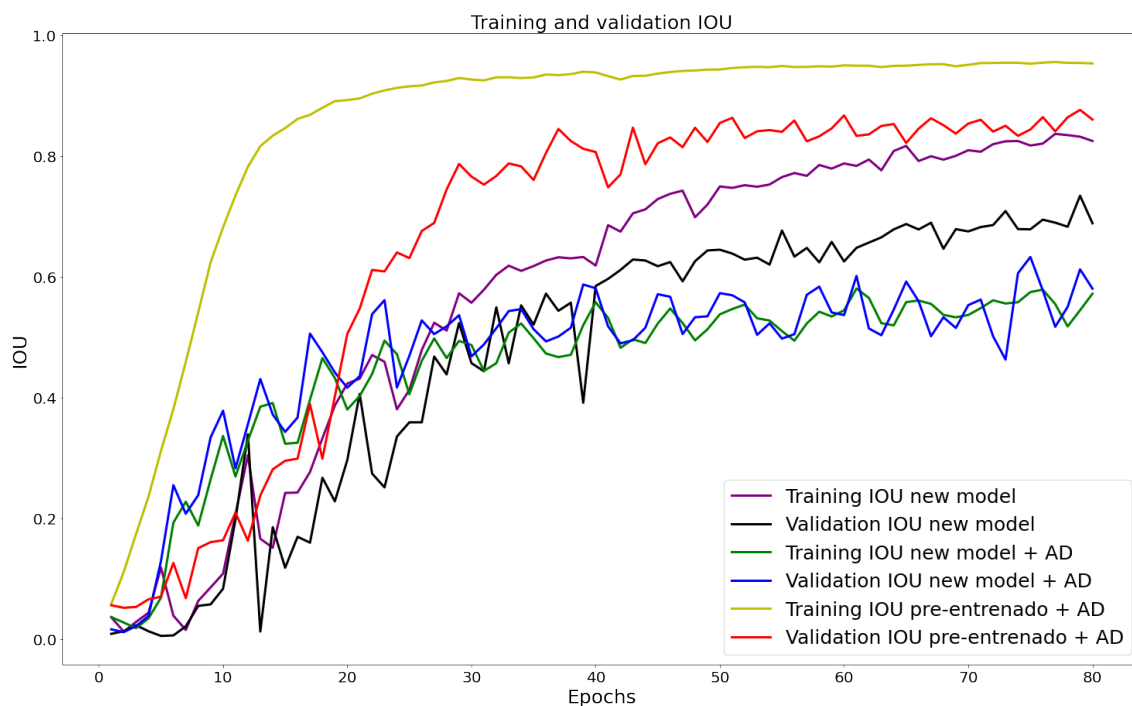


Figura 9: Comparación de IoU entre los diferentes modelos.

Como se puede observar, los 3 modelos aprenden (aumenta su valor de IoU) a medida que el entrenamiento avanza. Esto indica que la red neuronal U-Net es una elección acertada para estudiar este problema, si no fuera el caso, los modelos se quedarían estancados en un valor muy bajo. Sin embargo, podemos ver lo rápido que el modelo pre entrenado es capaz de superar el 80% de precisión en los datos de training, en tan solo

20 épocas. Mientras que ninguno de los otros 2 se acercan hasta mucho después. Cabe también destacar que los 3 modelos comienzan con una precisión prácticamente nula. Cabía esperar este comportamiento por parte de los nuevos, debido al valor aleatorio de sus pesos y sesgos, pero verlo también en la pre entrenada nos indica que no fue exactamente diseñada para un problema parecido, aunque sí similar, pues vemos lo rápido que aprende.

Tras el entrenamiento, podemos utilizar el modelo de validación como conjunto de test y obtener una métrica final del error de la predicción de este modelo sobre este conjunto. En la Tabla 1 podemos ver este valor y el tiempo medio (aproximado) empleado por cada modelo en entrenar una época.

Modelo	IoU test	segundos/época
Nuevo	0.842	2.5
Nuevo + AD	0.789	11
Pre entrenado + AD	0.878	14

Tabla 1: Valores de IoU para los diferentes modelos.

Con el fin de terminar de comparar los modelos, podemos mostrar alguno de los resultados para las imágenes de testeo (Figura 10).

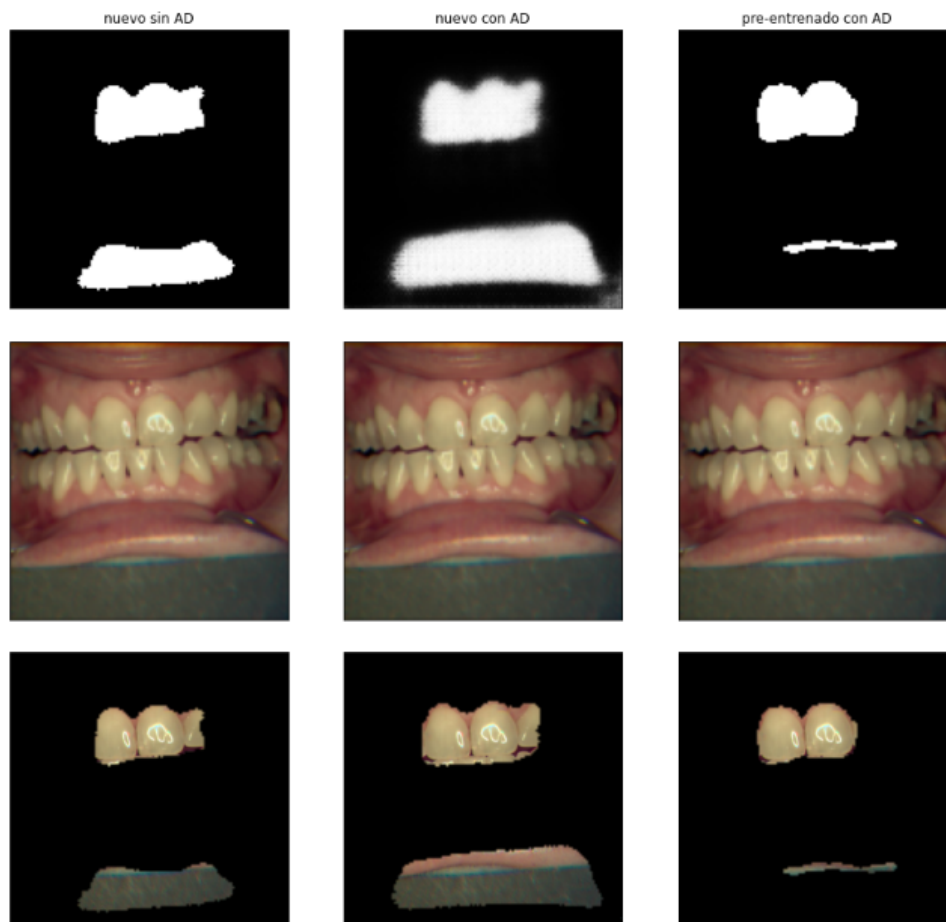


Figura 10: Comparación de resultados entre los diferentes modelos.

Vemos que es un resultado acorde con los valor de error que hemos visto en la tabla 1. Cabe resaltar que las predicciones que realizan los modelos no son binarias, sino que sus píxeles tienen valores que oscilan entre 0 y 1. Antes de aplicarlas sobre ninguna imagen RGB pasan por un *threshold* que convierte los píxeles con un valor > 0.5 en 1 y desecha los demás.

También cabe comentar que los objetos menores que la red identifica como dientes, como la línea inferior de la última foto, son eliminados en un tratamiento a posteriori como se verá más adelante.

Por lo visto hasta el momento, decidimos quedarnos con el modelo pre entrenado para continuar con el estudio, pues aunque los tiempos de entrenamiento son mayores, el resultado merece la pena.

3.2 Medida de precisión del modelo pre-entrenado

Para tener una medida más exacta de la precisión del modelo vamos a realizar un y un *LOO* con nuestras 25 parejas de entrenamiento. La media de las métricas de error IoU obtenidas tanto para los conjuntos de validación como los de test en estos dos métodos de estudio se muestran en la Tabla 2.

Estas medidas nos dan la precisión media que cabría esperar si utilizamos el modelo con cualquiera de los 20 (ó 24) datos de prueba para predecir las máscaras de 5 (ó 1) imágenes externas al entrenamiento. Podemos observar que la precisión en el conjunto de validación es menor al 80% para el *5-fold CV*, menos de lo que obteníamos con el modelo sin entrenamiento previo. Esto se debe a hemos retirado un 20% del dataset para usarlo como test en cada una de las iteraciones. Este 20% es muy notable en la calidad de los resultados debido a tener tan pocos datos con los que trabajar.

Con el método *LOO*, sin embargo, el valor de IoU se asemeja bastante al que veíamos en la Figura 9. Era de esperar, pues la diferencia de tamaño entre los datasets de entrenamiento es sólo de una imagen, es decir, de un 4%.

Es destacable que con tan solo 4 imágenes más el error medio cometido por el modelo en los conjuntos de test se reduce un 30%, la precisión aumenta de 0.7 a 0.79.

	IoU validation	IoU test
5-fold VC	0.78 ± 0.09	0.70 ± 0.09
LOO	0.93 ± 0.06	0.79 ± 0.07

Tabla 2: Medidas de precisión IoU mediante los métodos *5-fold CV* y *LOO*.

3.3 Clustering de nuestro espacio de color dental

Tras aplicar las máscaras de las paletas y substraer las máscaras de los reflejos de todas las imágenes de entrenamiento, obtenemos un conjunto de imágenes con tan solo tejido dental. Se pueden ver dos ejemplos en la Figura 11.

Si juntamos todos los píxeles de tejido dental de todas las imágenes RGB y los pasamos al espacio L^*a^*b , podemos representarlo y obtener la Figura 12.

Sobre esta figura podemos realizar un clustering utilizando el método K-Means. Usaremos $k=16$, dado que es el número de tonos de blanco que establece la guía de colores

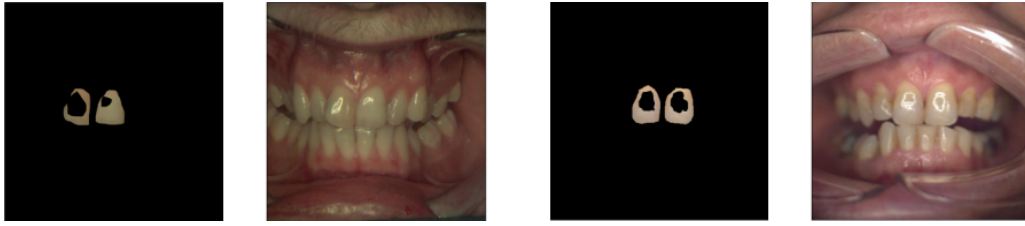


Figura 11: Tejido dental de dos imágenes diferentes.

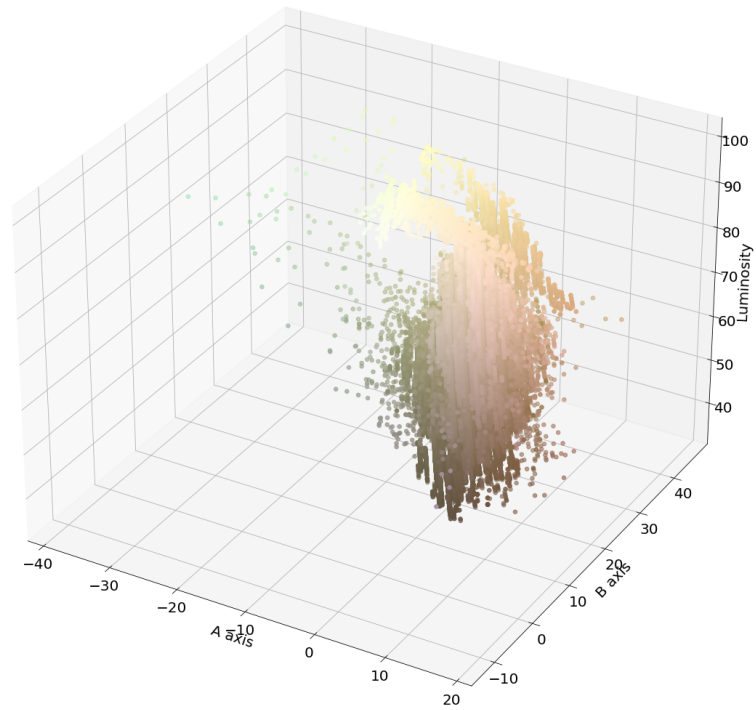


Figura 12: Representación de los píxeles con tejido dental en el espacio tridimensional L^*a^*b .

Classical Vita. De esta manera obtenemos los 16 centroides de nuestra nube de puntos y podemos representar la Figura 13.

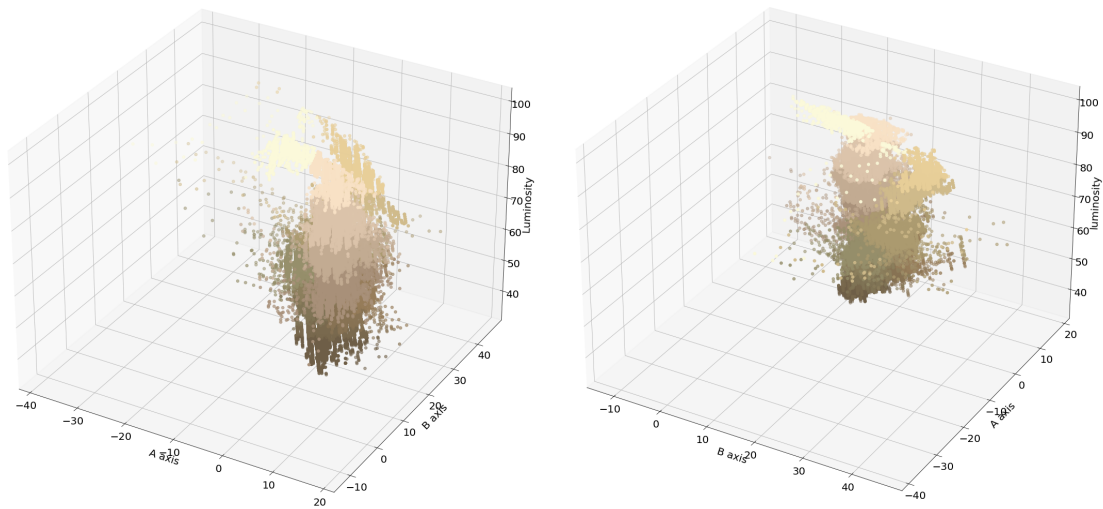


Figura 13: Representación de los 16 clústeres en nuestro espacio dental. Las gráficas son idénticas, salvo por una rotación de 90°.

Podemos comparar los tonos de color obtenidos mediante el clustering con la guía dental Classical Vita de manera cuantitativa (Figura 14). En la figura podemos observar una clara diferencia en los tonos de blanco. Esta discrepancia se debe principalmente a los siguiente motivos:



Figura 14: Comparación de la paleta de colores experimental con la guía dental Classical Vita.

- Diferente iluminación en las imágenes de muestra, como puede apreciarse comparando las imágenes RGB que aparecen en algunas de figuras mostradas anteriormente.
- Las imágenes, tras aplicar las máscaras, no sólo incluyen la parte frontal del diente, sino también sus bordes, donde la iluminación es menor y el color varía.

3.3.1 Estudio del color de los dientes

Ahora podemos hacer uso de la segmentación dental de nuestro modelo entrenado y realizar un estudio de su color dental gracias a la paleta de color que hemos obtenido del *clustering*. Para ello, sustituiremos con el centroide más cercano al color de cada uno de los píxeles de la imagen en el espacio tridimensional CIELAB y lo representaremos sobre los incisivos recortados por la red neuronal. La imagen resultante puede observarse en la Figura 15.

Como vemos el resultado dista bastante de lo que podríamos desear. Los tonos de blanco de la paleta de colores que hemos obtenido no son representativos de los colores reales de los dientes.



Figura 15: Procesado del color de un diente con los centroides obtenidos de manera experimental.

4 Conclusiones

Los objetivos principales de este trabajo eran construir una red neuronal capaz de extraer los incisivos superiores de una dentadura completa y crear una paleta de colores con la que poder clasificar las zonas con distintos tonos de blanco en el tejido dental.

En primer lugar, realizamos una comparación entre los diferentes entrenamientos que podíamos seguir para nuestra red neural U-Net. Determinamos que la mejor opción era optar por un modelo pre entrenado, con unos primeros pesos y sesgos ya calculados en un estudio ajeno al nuestro, pero con similitudes suficientes. Después de re entrenar este modelo haciendo uso de un aumento de datos, obtuvimos una red neuronal capaz de extraer los dientes con un acierto medio superior al 87 % (métrica IoU).

A continuación, se hizo uso de las máscaras dentales y de brillos para extraer el tejido dental sin los reflejos de las fotografías. De esta manera pudimos representar los colores de cada uno de los píxeles de estas imágenes en el espacio CIELAB.

Haciendo uso de un algoritmo de *clustering*, encontramos los centroides de esta distribución de píxeles y generamos nuestra propia guía dental Classical Vita. Vimos que nuestra guía tenía tonos mucho más oscuros que la guía convencional.

Estas diferencias se deben principalmente a que la iluminación de las imágenes no es homogénea y que parte de los píxeles tomados para construir nuestra guía pertenecían a zonas externas del diente, por lo que eran más oscuras, debido a la curvatura del mismo.

Para mejorar los resultados obtenidos sería necesaria una base de datos más amplia con la que poder entrenar nuestro modelo. De esta manera, no tendríamos que conformarnos sólo con los incisivos superiores, sino que podríamos intentar llegar hasta los caninos. Más allá de ellos, necesitaríamos fotografías desde otros ángulos donde se capturasen las muelas de frente.

Por otro lado, si esta base de datos tuviese una mejor iluminación, nuestra guía dental seguramente se asemejase más a la Classical Vita y podríamos confiar más en el resultado.

Conclusions

The main objectives of this thesis were to build a neural network capable of extracting the upper incisors of a complete set of teeth and to create a colour palette to classify areas with different shades of white in dental tissue.

First of all, we made a comparison between the different training paths that we could follow for our U-Net neural network. We determined that the best option was to opt for a pre-trained model, with initial weights and biases already calculated in a study other than ours, but with sufficient similarities. After retraining this model using data augmentation, we obtained a neural network capable of extracting teeth with a mean accuracy greater than 87% (IoU metric).

The dental masks were then used to extract the dental tissue without the reflections of the photographs. This way, we were able to represent the colours of each of the pixels of these images in CIELAB space.

Using a clustering algorithm, we found the centroids of this pixel distribution and generated our own Classical Vita dental guide. We saw that our guide had much darker tones than the conventional guide.

These differences are mainly because the illumination of the images is not homogeneous and that part of the pixels taken to build our guide belonged to external areas of the teeth, therefore they were darker, due to their curvature.

To improve the results obtained, we will need a larger database to train our model. This way, we could do the study not only for the incisors but for even the canines. Beyond them, we would need photographs from other angles where the molars were captured from the front.

Also, if this database had better lighting, our dental guide would surely be more similar to the Classical Vita and we could be more confident in the result.

Referencias

- [1] Jorge Calvo. Crear red neuronal desde las matemáticas. <https://jorgemcalvo.medium.com/crear-red-neuronal-desde-las-matem%C3%A1ticas-9b3b30afe372>.
- [2] Ringa Tech. Redes neuronales convolucionales - clasificación avanzada de imágenes con ia / ml (cnn). <https://www.youtube.com/watch?v=4sWhhQwHqug&t>.
- [3] Universidad de Finlandia. Oral and dental spectral image database. <https://sites.uef.fi/spectral/odsi-db/>.
- [4] U-net : todo lo que tienes que saber sobre la red neuronal de computer vision. <https://datascientest.com/es/u-net-lo-que-tienes-que-saber>.
- [5] Segmentation models. <https://segmentation-models.readthedocs.io/en/latest/tutorial.html>.
- [6] Entendiendo el espacio de color cie l*a*b*. <https://sensing.konicaminolta.us/mx/blog/entendiendo-el-espacio-de-color-cie-lab/>.
- [7] Andrés Prado. Teeth-characterization. <https://github.com/andresprado99/Teeth-Characterization>.