



UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA
Escuela de Ingeniería Informática



Estudio de aplicación de técnicas de aprendizaje profundo para la estimación de presencia de microplásticos en zonas costeras

María Cristina Benlliure Jiménez

Supervisado por:
Modesto F. Castrillón Santana
Adrián Peñate Sánchez

Marzo 2020

Resumen

La presencia de microplásticos en el mar es un problema que ha adquirido relevancia en los últimos tiempos. Para analizar su evolución se llevan los residuos plásticos de zonas costeras a laboratorios para su conteo exhaustivo. Debido a la complejidad de recopilación de muestras y tratamiento de datos es fundamental tanto la eficiencia como la estandarización durante el proceso.

El objetivo de este estudio es determinar la integración de herramientas automáticas, permitiendo acelerar su observación. Para ello se utilizaron imágenes capturadas directamente en la playa.

En el documento se comparan los resultados obtenidos tanto en la detección como en la estimación de plásticos, entre distintas arquitecturas de redes neuronales artificiales.

Abstract

Recently, the presence of micro-plastics at sea is an issue that has gained prominence. To analyze its evolution, plastic waste from seaside is taken to laboratories for its exhaustive counting. Due to the complexity of sample collection and data processing, efficiency and standardization are essential during the process.

The objective of this study is to determine the integration of automatic tools to accelerate its analysis. For this, images captured directly on the beach were used.

This document compares the results obtained in the detection of plastics as well as estimation of plastics, between different architectures of artificial neural networks.

Índice general

1. Introducción	1
1.1. La problemática de los microplásticos	2
1.2. Procedimiento actual	4
1.3. Objetivos del estudio	8
2. Metodología	10
2.1. Aprendizaje automático	11
2.2. Tecnologías usadas	16
2.2.1. Herramientas para la implementación de la aplicación de anotación .	16
2.2.2. Librerías usadas para la implementación del clasificador de microplásticos	18
3. Descripción y tratamiento de datos	19
3.1. Descripción de los datos	19
3.1.1. Conjunto de datos A	19
3.1.2. Conjunto de datos B	25
3.1.3. Conjunto de datos C	29
3.2. Aplicación de anotación	30
3.2.1. Interfaz gráfica de la aplicación	30
3.2.2. Estructura del código de la aplicación	36
4. Experimentos	37
4.1. Métrica para la evaluación de resultados	37
4.2. Determinación del tamaño de segmento	40
4.2.1. Celdas de 64 píxeles	40
4.2.2. Celdas de 16 píxeles	50
4.2.3. Celdas de 32 píxeles	55
4.3. Determinación de la arquitectura óptima	61
4.4. Entrenamiento con un mayor conjunto de datos	62
4.5. Estimación de la cantidad de gramos	67
5. Conclusiones	74

Índice de figuras

1.1. Cangrejo ermitaño en Okinawa, Japón	2
1.2. Zona delimitada de una playa	4
1.3. Arena recogida en una malla	5
1.4. Bolsa con la muestra siendo limpiada en agua de mar	5
1.5. Vaso de precipitados con la muestra obtenida en la playa	6
1.6. Separación y pesado de la muestra obtenida en la playa	7
1.7. Escáner que se usa para obtener las imágenes en alta resolución de los microplásticos	8
2.1. Estructura de una neurona humana	12
2.2. Estructura de una neurona	13
2.3. Estructura de una red neuronal	14
2.4. Ejemplo de una convolución	15
2.5. Captura de pantalla de NetBeans para la creación de GUI	17
3.1. Playa de la Cicer con microplásticos	20
3.2. Peso de los microplásticos por imagen, del conjunto de datos A	21
3.3. Peso de los microplásticos por imagen separados por tamaño, del conjunto de datos A	22
3.4. Partículas entre 1-5mm por imagen, del conjunto de datos A	23
3.5. Partículas entre 5-25mm por imagen, del conjunto de datos A	24
3.6. Playa de la Lambra con microplásticos	25
3.7. Cantidad de microplásticos en la playa Famara, con eje temporal	26
3.8. Cantidad de microplásticos en la playa Lambra, con eje temporal	27
3.9. Peso de los microplásticos por imagen, del conjunto de datos B	28
3.10. Playa de Cuervitos, arena delimitada con un marco de plástico	29
3.11. La aplicación escritorio con ventanas abiertas	30
3.12. Menú en la aplicación	31
3.13. Ventana emergente para la modificación del tamaño de segmento	31
3.14. Ventana emergente para la selección de la imagen a mapear	32
3.15. Interfaz del <i>JInternalFrame</i>	33
3.16. Imagen sin y con rejilla de 16x16 píxeles	34
3.17. Ventana de error cuando no existe un CSV	34
3.18. Ventana para seleccionar el CSV a abrir	35
3.19. Ventana con mensaje de confirmación para cerrar la aplicación	35

4.1. Matriz de confusión para un modelo binario	37
4.2. Matriz de confusión para las clases arena y plástico	38
4.3. Formulas para el análisis de resultados	39
4.4. Número de segmentos que contienen microplásticos por imagen, 64 píxeles	41
4.5. Total de segmentos que hay de arena y plástico, 64 píxeles	42
4.6. Transformaciones a realizar	42
4.7. Total de segmentos que hay de arena y plástico con datos aumentados, 64 píxeles	43
4.8. Exactitud de predicción del modelo durante el entrenamiento y test	45
4.9. Función de pérdida del entrenamiento y test	45
4.10. Matriz de confusión para segmentos de 64 píxeles	46
4.11. Estructura de la red neuronal usada	47
4.12. Fotografía perteneciente al conjunto de datos B de la playa Famara	48
4.13. Resultados de predicción para una fotografía perteneciente al conjunto de datos B de la playa Famara, 64 píxeles	49
4.14. Segmentos con plástico por imagen. Para 16 píxeles y 64 píxeles.	50
4.15. Cantidad de segmentos con plástico por imagen para 16 y 64 píxeles con valores normalizados	50
4.16. Total de segmentos de arena y plástico, 16 píxeles	51
4.17. Total de segmentos que hay de arena y plástico con datos aumentados, 16 píxeles	52
4.18. Matriz de confusión para segmentos de 16 píxeles	53
4.19. Microplástico fragmentado durante la segmentación	53
4.20. Resultados de predicción para una fotografía perteneciente al conjunto de datos B de la playa Famara, 16 píxeles	54
4.21. Matriz de confusión para segmentos de 32 píxeles	55
4.22. Resultados de predicción para una fotografía perteneciente al conjunto de datos B de la playa Famara, 32 píxeles	57
4.23. Fotografía 2 perteneciente al conjunto de datos B de la playa Famara	58
4.24. Resultados de predicción para otra fotografía perteneciente al conjunto de datos B de la playa Famara, 32 píxeles	59
4.25. Predicciones no tan buenas para otras fotos del conjunto de datos B	60
4.26. Tabla con los resultados de las distintas pruebas	61
4.27. Conteo de la cantidad de segmentos por clase	63
4.28. Matriz de confusión para imágenes de 1024	64
4.29. Matriz de confusión para imágenes de 2048	64
4.30. Resultados obtenidos para imágenes de 1024 y 2048 respectivamente	65
4.31. Otros resultados obtenidos para imágenes de 1024 y 2048 respectivamente	66
4.32. Matriz de correlación entre los gramos del conjunto de datos A y el número de segmentos con plástico	67
4.33. Regresión lineal para los datos del conjunto de datos A y el número de seg- mentos con plástico de 32 píxeles mapeados manualmente	68
4.34. Número de segmentos que contienen microplásticos por imagen, para un ta- maño de segmento de 32 píxeles	69
4.35. Imagen predicha del conjunto de datos B	70
4.36. Textura de arena de la Figura 4.35	70
4.37. Imagen predicha del conjunto de datos B	71

4.38. Matriz de correlación entre los gramos del excel y los estimados	71
4.39. Tabla con los gramos originales, número de segmentos predichos y gramos estimados para todas las imágenes del conjunto de datos B	72
4.40. Estimación de los gramos frente a los gramos originales para las imágenes del conjunto de datos B	73

Capítulo 1

Introducción

La acumulación de cantidades de plástico en los ecosistemas es uno de los problemas medioambientales a los que nos enfrentamos en la actualidad. Por sus características físicas y químicas, el plástico es un material muy duradero y difícil de degradar por los microorganismos que se encuentran en los ecosistemas. Científicos de la Universidad de Hawaii han publicado un artículo que demuestra que el plástico libera una variedad de sustancias químicas durante su degradación, algunas de las cuales tienen un impacto negativo en los organismos y ecosistemas, y otras generan gases de efecto invernadero, tales como metano y etileno [16].

Los cúmulos de plástico, además de la generación de sustancias químicas, suponen un peligro para la biodiversidad y la fauna, no es difícil encontrar fotos de animales atrapados o restos de animales con plástico en sus intestinos. Si una astilla de plástico le perfora el intestino, el animal puede morir con rapidez. Pero normalmente la ingestión de plástico se convierte en una desnutrición.

La motivación inicial para realizar este estudio procede en que analizar y ver la evolución de la contaminación de plásticos en las playas es un proceso largo y complejo, desde la recopilación de muestras y hasta el tratamiento de datos. Realizar una estimación previa de la cantidad de plásticos que hay en la zona costera agilizaría el proceso que realizan los biólogos permitiendo un análisis más rápido de la evolución de la contaminación en el litoral marino. Conociendo así zonas de pesca o piscifactorías pueden estar contaminadas y con probabilidad de que los organismos de la zona sufran de bioacumulación, en toxicología esto es el proceso de acumulación de sustancias químicas en

organismos vivos. El consumo de peces contaminados con ingredientes químicos del plástico o algunas sustancias químicas tóxicas que absorbe el plástico podría suponer un riesgo para la salud.



Figura 1.1: Cangrejo ermitaño en Okinawa, Japón. Foto: Shawn Miller, National Geographic [13]

1.1. La problemática de los microplásticos

Los microplásticos son piezas muy pequeñas de plástico que contaminan el medioambiente. Aunque no hay un acuerdo sobre a partir de qué tamaño puede considerarse microplásticos, la Administración Nacional Oceánica y Atmosférica (NOAA) utiliza el parámetro de menos de cinco milímetros de diámetro para clasificarlos [14].

Estas pequeñas partículas de plástico provienen de diversas fuentes entre ellas los cosméticos, ropa, artículos de pesca, desechos plásticos de uso cotidiano y procesos industriales. Dependiendo de su procedencia, se pueden dividir en dos categorías:

- **Los microplásticos primarios:** Son fabricados específicamente para ser utilizados en productos. Se utilizan generalmente como limpiadores faciales y cosméticos. En algunos casos, han sido utilizados en la medicina como vector farmacológico.
- **Los microplásticos secundarios:** Se derivan del proceso de deterioro de desechos plásticos más grandes como bolsas, botellas o redes de pesca. Con el tiempo, las propiedades físicas, biológicas, incluyendo la fotodegradación causada por la exposición al sol, puede reducir la integridad estructural de los desechos plásticos a un tamaño que es finalmente indetectable a simple vista. Este proceso de descomposición de material plástico grande en pedazos mucho más pequeños se conoce como fragmentación.

Se estima que la mitad del plástico que utilizamos cotidianamente son artículos de un solo uso, como envases o pajitas. El plástico de un solo uso tiene una vida útil promedio de 12 a 15 minutos; sin embargo, puede tardar entre 400 y 1.000 años en desintegrarse.

Debido a que los plásticos no se biodegradan y solo se fragmentan en partes más pequeñas, estas terminan siendo absorbidos o ingeridas por muchos seres vivos al no poder diferenciar entre si es su alimento marino, alojándose en sus organismos y pudiendo producir una muerte por intoxicación. Se han visto casos entre aves marinas, peces, escualos, mamíferos acuáticos e incluso una ballena varada que murió a causa de la ingesta de plástico.

En cuanto a los seres vivos más sensibles, se puede decir con certeza que son el crustáceo y el molusco. El primero por alimentarse del plancton del fondo marino donde suelen acabar estos microplásticos y el segundo porque los moluscos son los filtradores por excelencia por lo que retienen toda la contaminación existente en el mar. Además también existen otras especies de fauna marina que se están viendo afectadas por esta contaminación, como por ejemplo, las ballenas y los tiburones, que consumen grandes cantidades de pescado con restos de plástico.

El ciclo completo y movimiento de los microplásticos en el medioambiente aún no se conoce por completo y las consecuencias de estas partículas, que terminan alojadas en el cuerpo humano, todavía siguen siendo investigadas, pero, se han encontrado partículas en alimentos provenientes del mar, como la

sal, el agua embotellada y agua del grifo. La palabra microplástico fue elegida palabra del año 2018 por la fundación *Fundéu BBVA* [6].

1.2. Procedimiento actual

Hay dos maneras de recoger las muestras para contabilizar la cantidad de plásticos en el mar, o bien usando una red de copo en un barco para recolectar los plásticos que flotan en el mar, o bien en las costas. Hay que mencionar que en la primera forma de recogida se recolectarán plásticos de densidades pequeñas que floten, por lo que encontraremos más restos de poliestireno que en las costas, donde el viento se las lleva.



Figura 1.2: Zona delimitada de una playa

El proceso de recogida de datos en las costas descrito está realizado por el grupo EOMAR [12], Grupo de Ecofisiología de Organismos Marinos. Este

proceso comienza delimitando una zona de 50x50 centímetros por medio de un marco, evitando grandes superficies de piedras (ver Figura 1.2).

Una vez que tienen la superficie delimitada se procede a recolectar los primeros centímetros de profundidad de la zona con una espátula metálica hasta alcanzar un volumen de un litro. Luego se procede a pesar la muestra y a ponerla en una bolsa de malla de un milímetro (ver Figura 1.3).



Figura 1.3: Arena recogida en una malla

Las bolsas con las muestras se limpian con agua de mar para eliminar la arena y mantener solo los microplásticos y el material orgánico, para posteriormente llevarlas al laboratorio (ver Figura 1.4).



Figura 1.4: Bolsa con la muestra siendo limpiada en agua de mar

Una vez en el laboratorio se deja secar la muestra y se vierte el contenido de la bolsa en un vaso de precipitados, para luego añadir etanol al 96 % hasta alcanzar un volumen de 100 ml. Este paso permite separar los microplásticos y el material biológico, dejando en el fondo del vaso de precipitados los fragmentos de plástico y flotando el material orgánico (ver Figura 1.5).

Los microplásticos que quedan en el fondo se filtran con una red de malla de $50\ \mu\text{m}$ y se secan durante 24 horas a 60°C . Antes de ponderar la cantidad de partículas, se separan por tamaño con una malla de 5mm, separando las partículas entre 1 y 5 mm con las de entre 5 y 25mm (ver Figura 1.6).

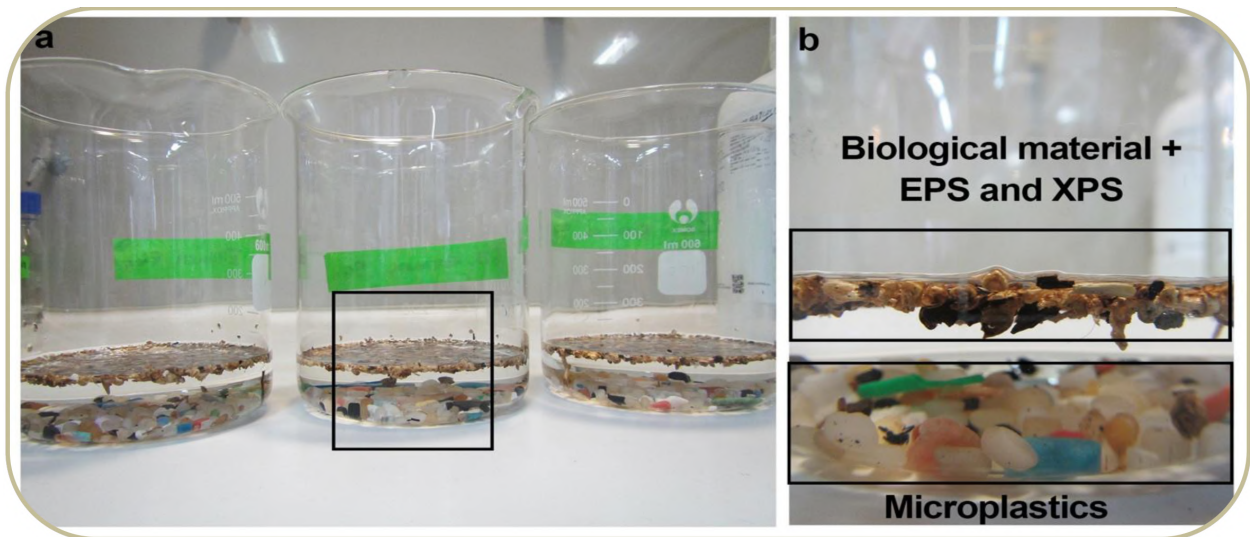


Figura 1.5: Vaso de precipitados con la muestra obtenida en la playa



Figura 1.6: Separación y pesado de la muestra obtenida en la playa

Una vez separadas se pesan y se cuentan el número de tipos de partículas que hay en la muestra. Para digitalizar las muestras obtenidas se usa un escáner donde se ponen los microplásticos con separaciones para evitar que se superpongan (Figura 1.7). Con el digitalizado de las partículas se pueden aplicar técnicas automatización que faciliten el trabajo de los biólogos. Como el sistema SMACC (*System for Microplastics Automatic Counting and Classification*) [1]. Un programa en Java que proporciona a los investigadores una herramienta para acelerar los procedimientos de cuantificación y clasificación de microplásticos, usando internamente inteligencia artificial, reduciendo a menos de la mitad el tiempo del proceso de conteo, respecto al procedimiento estándar.



Figura 1.7: Escáner que se usa para obtener las imágenes en alta resolución de los microplásticos

1.3. Objetivos del estudio

El objetivo de este estudio es agilizar el tiempo en el que los biólogos tardan en realizar una primera estimación de la cantidad de gramos que hay en las playas. Pudiendo saber la cantidad de plásticos que traen las corrientes marinas y la cantidad de plásticos que hay en el litoral perjudicando las distintas especies que la habitan.

Si se pudiera realizar una primera estimación de la cantidad de microplásticos que hay en la arena *in situ* se podría ver la evolución de playas con franjas temporales más reducidas, permitiendo monitorizar un mayor número de zonas costeras.

Así pues, este trabajo de fin de grado se centra en, la detección y estimación

de microplásticos, por medio de imágenes tomadas directamente en las playas. Para ello hay que interpretar los datos recogidos por el grupo EOMAR [12] para estudiar su utilización en el entrenamiento de redes neuronales, con el fin de definir una estrategia.

Una vez decidido cómo abordar el problema habrá que estructurar y etiquetar manualmente las imágenes, utilizando un programa auxiliar, para adaptarlas a la estrategia tomada determinando la máscara delimitadora de las zonas con microplásticos en la arena.

Además, habrá que realizar diversas pruebas a la espera de obtener unos resultados más afines a la realidad. Tras distintas pruebas habrá que contrastar y evaluar los distintos resultados obtenidos, tanto para las predicciones de la zona delimitada para plástico como la estimación de los gramos.

Capítulo 2

Metodología

Durante el proyecto se usó una metodología de desarrollo ágil, estas se basan en un desarrollo iterativo e incremental, donde los requisitos y soluciones evolucionan con el tiempo según la necesidad del proyecto. Para ello se usó como inspiración el método SCRUM, este se basa en una manera de trabajar en equipo donde se aplican de manera regular un conjunto de buenas prácticas para obtener una buena productividad en un proyecto. En SCRUM se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto.

Así pues, en el estudio se realizaron reuniones semanales grupales. Ahí se reúnen, tanto los tutores, como el grupo de alumnos que son tutorizados. En las reuniones los alumnos enseñábamos los progresos del proyecto respecto a la semana anterior y se fijaban objetivos a realizar.

Para la detección de las partículas de plástico en las distintas fotos disponibles se decidió a utilizar aprendizaje automático por medio de redes neuronales supervisadas (Sección 2.1).

Este proyecto se abordó con una metodología de “divide y vencerás”, con esto nos referimos a una estrategia de separar el problema inicial en problemas más pequeños. En mi caso, dividir las imágenes a procesar por la red neuronal en fragmentos más pequeños, dejando como resultado una matriz que indica si un segmento tiene plástico o no, permitiendo aplicar la matriz resultante como una máscara a la imagen y dejando únicamente los fragmentos con plástico visibles. A partir de ese resultado se intentó estimar la cantidad de gramos de plástico que puede haber en la imagen (Capítulo 4).

2.1. Aprendizaje automático

El **aprendizaje automático** o *machine learning* es una disciplina científica del ámbito de la inteligencia artificial que crea sistemas que aprenden automáticamente. Consiste en la capacidad del sistema para identificar una gran serie de patrones complejos determinados por una gran cantidad de parámetros. Hay tres tipos de aprendizaje automático:

- **Aprendizaje supervisado:** El objetivo del aprendizaje supervisado es el de crear una función capaz de predecir el valor correspondiente a cualquier objeto de entrada válida después de haber visto una serie de ejemplos, los datos de entrenamiento. Genera una función que transforma los datos de entrada en los resultados deseados. Es decir, se produce cuando se entrena al sistema con datos etiquetados. Otro término para el aprendizaje supervisado es la clasificación. Los clasificadores más utilizados son algunos tales como las redes neuronales o el algoritmo de los K vecinos más cercanos.
- **Aprendizaje no supervisado:** El sistema no conoce que son los datos y se dedica a buscar similitudes para agruparlos. Este tipo de aprendizaje usa algoritmos de clusterización o agrupamiento.
- **Aprendizaje semisupervisado:** El sistema usa datos de entrenamiento tanto etiquetados como no etiquetados: normalmente una pequeña cantidad de datos etiquetados junto a una gran cantidad de datos no etiquetados.

Dentro del aprendizaje supervisado encontramos las **redes neuronales artificiales**, estas están basadas en el funcionamiento de las redes de neuronas biológicas (ver Figura 2.1). Una neurona es una célula componente principal del sistema nervioso, cuya función principal es recibir, procesar y transmitir información a través de señales químicas y eléctricas. Estas están compuestas principalmente por el núcleo, el axón y dendritas. Las dendritas se encargan de captar los impulsos nerviosos que emiten otras neuronas. Estos impulsos, se procesan en el soma y se transmiten a través del axón que emite un impulso nervioso hacia las neuronas contiguas.

Ahora bien, las redes neuronales artificiales se basan en una idea sencilla: dados unos parámetros se busca predecir un cierto resultado. Por ejemplo,

sabiendo los píxeles de una imagen habrá una forma de saber qué número hay escrito. Todo esto usando matemáticas y estadística.

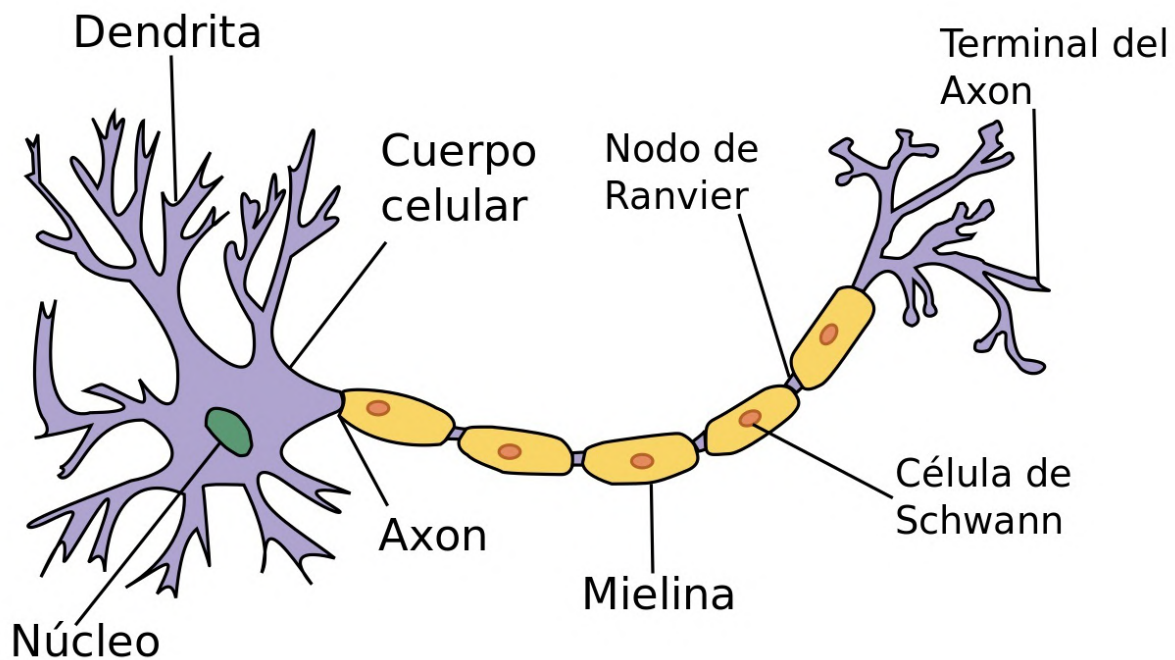


Figura 2.1: Estructura de una neurona humana [17]

Las redes neuronales están formadas por capas de neuronas. Una neurona es una función matemática, internamente usa todos los datos de entrada para realizar una suma ponderada, la ponderación viene dada por un peso que se le asigna a cada uno de los valores de entrada (X_1 y X_2) de la neurona. Es decir, cada conexión que llega a la neurona tendrá asociado un valor (W_1 y W_2) que servirá para definir con qué intensidad afecta cada variable de entrada. Cada neurona tiene también un valor asignado llamado sesgo o *bias* (b), que se encuentra dentro de la fórmula asignado al 1, este parámetro realiza la misma función que el término independiente de una regresión lineal. Se puede ver la estructura de una neurona en la Figura 2.2. Los pesos son los parámetros de nuestro modelo que se ajustan durante el entrenamiento para que la red aprenda.

Dentro de una neurona antes de obtener un dato de salida hay que pasar la suma ponderada por una función de activación que permitirá transformar la información de salida.

Hay varias funciones de activación algunas de ellas son:

- **La función escalón:** Para un valor de entrada mayor que un umbral el resultado será 1, sino será 0.
- **La función sigmoide:** Para un valor de entrada dado, hace que los valores muy pequeños se saturan en 0 y los muy grandes se saturan en 1. Sirve también para representar probabilidades que vienen en el rango de 0 a 1.
- **La función tangente hiperbólica (tanh):** Es similar a la sigmoide, pero su rango varía entre -1 y 1.
- **La función unidad rectificada lineal (relu):** Se comporta como una función lineal cuando el valor de entrada es positivo y una constante a 0 cuando el valor de entrada es negativo.

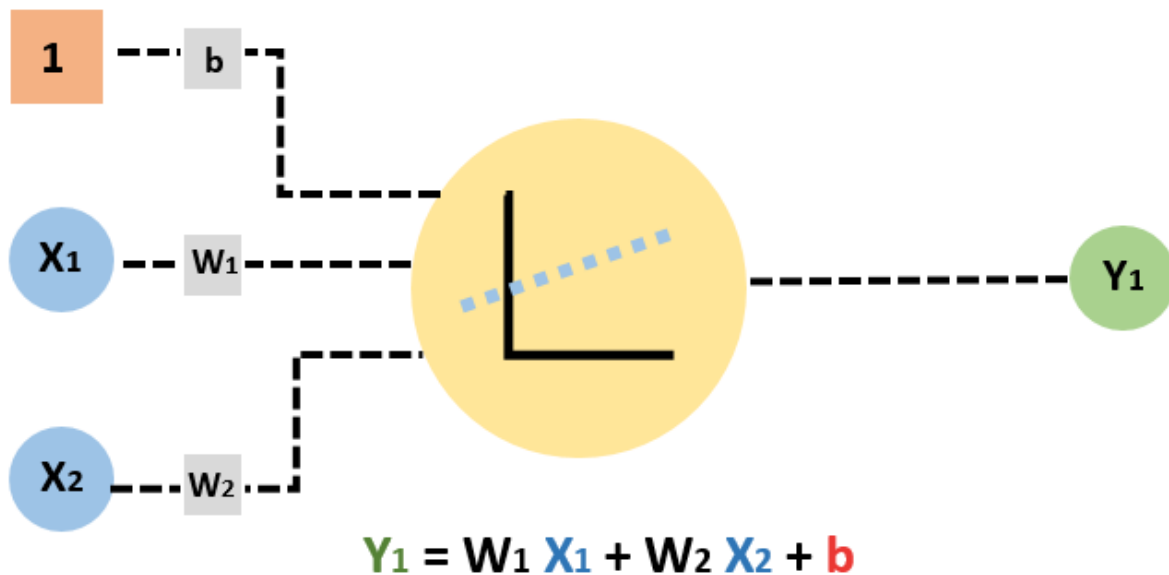


Figura 2.2: Estructura de una neurona (adaptada de [2])

La unión de muchas neuronas es lo que forma una red neuronal. Cuando están estructuradas en capas se denomina red multicapa, a la primera se le llama capa de entrada, a la última capa de salida y a las de en medio capas ocultas (Figura 2.3).

Para la clasificación de patrones se suelen usar capas de redes convolucionales, su principal ventaja es que cada parte de la red se le entrena para realizar una

tarea, por lo que el entrenamiento es más rápido. Es decir, las primeras capas pueden detectar líneas, curvas y se van especializando hasta llegar a capas más profundas que reconocen formas complejas como un rostro o la silueta de un animal.

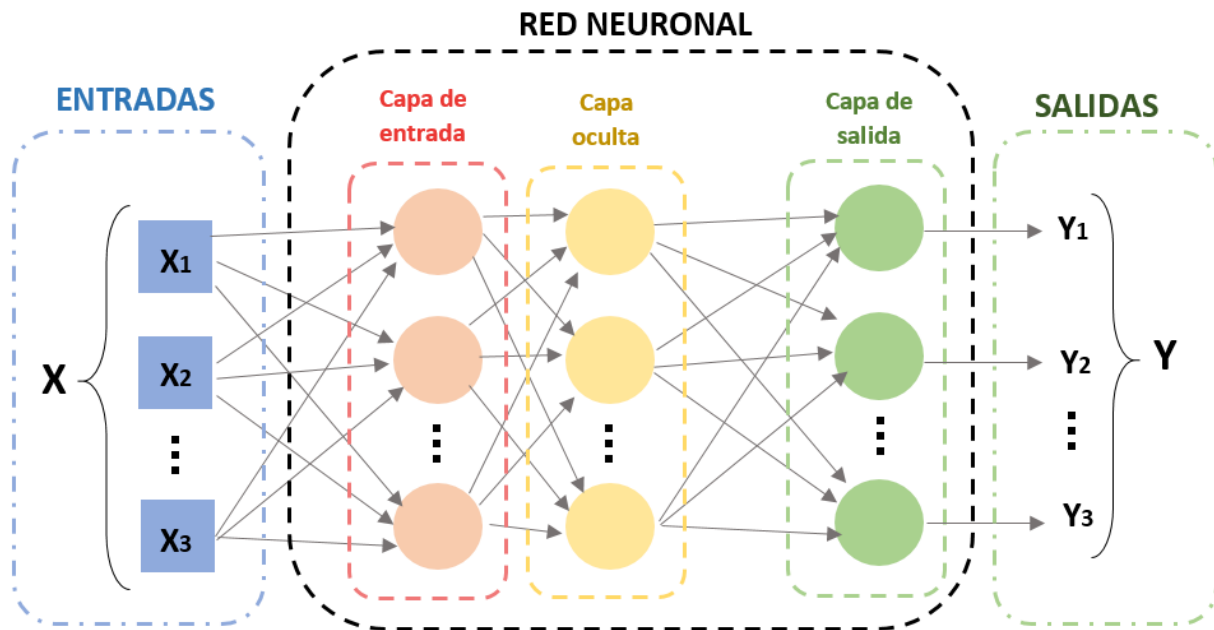


Figura 2.3: Estructura de una red neuronal (adaptada de [4])

Una red neuronal convolucional es una red multicapa que consta de capas convolucionales y capas de reducción alternadas, y finalmente tiene capas formadas por neuronas.

Una convolución consiste en tomar grupos de píxeles adyacentes de la imagen de entrada, e ir realizando el producto escalar contra una pequeña matriz que se llama kernel. El kernel de tamaño 3 por 3 píxeles, por ejemplo, recorre todas las matrices de entrada (de izquierda-derecha, de arriba-abajo) y genera una nueva matriz de salida. Figura 2.4.

En la reducción se disminuye la cantidad de parámetros al quedarse con las características más comunes. La forma de reducir parámetros se realiza mediante la extracción de estadísticas como el promedio o el máximo de una región fija del mapa de características, al reducir características el método pierde precisión aunque mejora su compatibilidad.

Una de las tácticas que se usan para evitar que la red realice *overfitting* es poner capas de *dropout*. *Dropout* es un método que desactiva un número de

conexiones neuronales de una red neuronal de forma aleatoria. En cada iteración de la red neuronal *dropout* desactiva diferentes conexiones, las conexiones desactivadas no se toman en cuenta para el *backpropagation* lo que obliga a las neuronas cercanas a no depender tanto de las neuronas desactivadas.

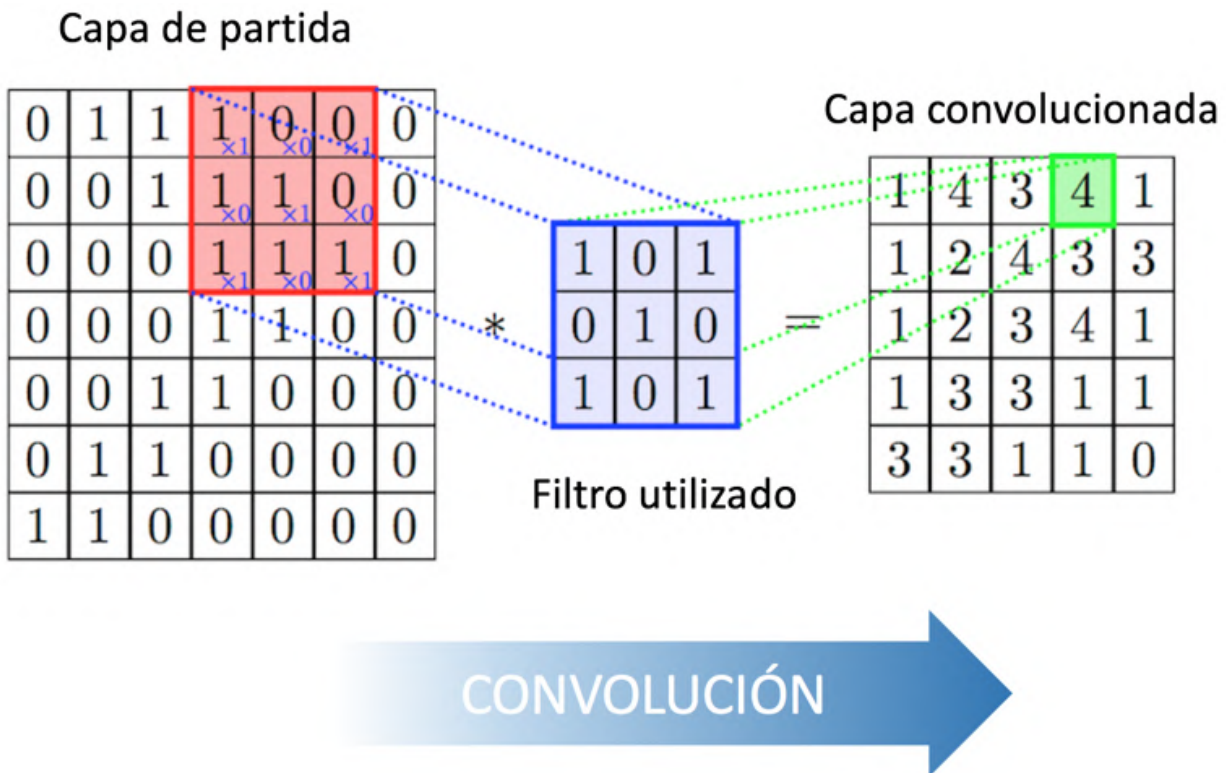


Figura 2.4: Ejemplo de una convolución [5]

El *backpropagation* es un algoritmo para el aprendizaje supervisado de redes neuronales artificiales que utiliza el descenso según el gradiente. Las salidas de error se propagan hacia atrás, partiendo de la capa de salida, hacia todas las neuronas de la capa oculta que contribuyen directamente a la salida, todo este proceso se realiza de forma eficiente.

Para poder entrenar la red neuronal con un aprendizaje supervisado necesitamos indicarle a la red el resultado para que encuentre un patrón. Así pues se ha decidido segmentar y mapear manualmente las imágenes, indicando para segmento si contiene plástico o no.

2.2. Tecnologías usadas

En el desarrollo del trabajo de fin de grado se utilizaron principalmente como lenguajes de programación **Python 3** y **Java**. Java únicamente se usó para la creación de un programa auxiliar para el etiquetado de las imágenes (Sección 3.2). Mientras que Python se usó para toda la parte de inteligencia artificial y redes neuronales. El código en Python se ejecutó usando los cuadernos **Jupyter** (Jupyter Notebook), estos cuadernos combinan el procesamiento de texto con código ejecutable en distintos lenguajes, en nuestro caso Python, permitiendo una fácil reproducción de los distintos resultados obtenidos. Jupyter se ejecuta en local como una aplicación cliente-servidor.

Como herramienta de instalación para de Jupyter y las distintas librerías que se utilizaron, se usó **Anaconda**. Anaconda es una distribución de Python que funciona como un gestor de entorno y un gestor de paquetes, permitiendo instalar los distintos paquetes tanto en la propia consola de Anaconda (Anaconda prompt) o en la interfaz gráfica que se dispone si se instala Anaconda Navigator. También permite la creación de entornos virtuales para instalación de paquetes en distintas versiones, es decir, cada entorno puede tener librerías de Python de forma que no intervengan en los otros entornos.

Por otro lado durante el entrenamiento de las distintas arquitecturas de redes neuronales al tardar demasiado en local se realizó en los cuadernos de **Google Colaboratory** utilizando la capacidad de cómputo de las GPUs de Google. Colaboratory es un entorno gratuito de Jupyter Notebook que no requiere configuración y que se ejecuta completamente en la nube [7].

2.2.1. Herramientas para la implementación de la aplicación de anotación

Para la realización del programa con interfaz gráfica se decidió usar **Java Swing**. Java Swing es una herramienta de interfaz gráfica de usuario, en inglés Graphical User Interface (GUI), que incluye un amplio conjunto de *widgets* y es independiente de la plataforma.

La biblioteca Swing está construida sobre el conjunto de herramientas de *widgets* abstractos de Java (AWT), un kit de herramientas GUI más antiguo que depende de la plataforma.

Como entorno de desarrollo integrado, en inglés *Integrated Development Environment* (IDE), usó **NetBeans**. Ya que, este IDE permite la creación de interfaces gráficas por medio de una interfaz gráfica, agilizando el proceso de creación de la aplicación.

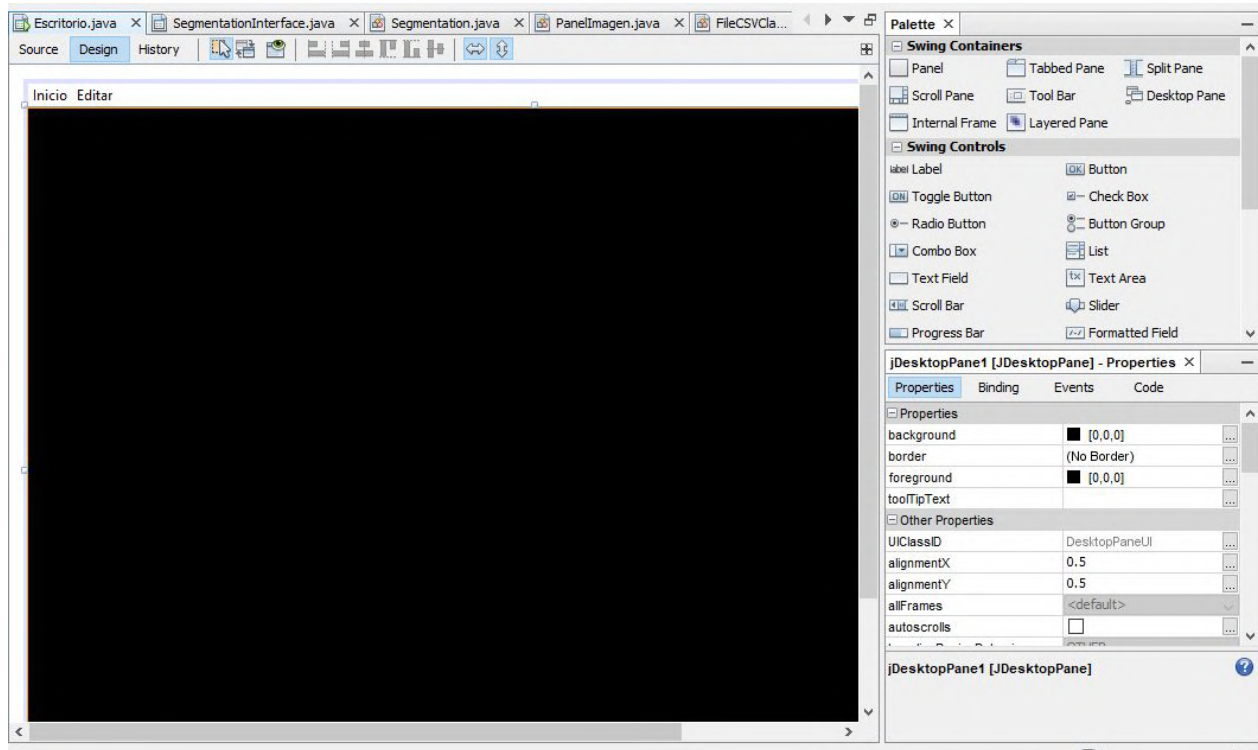


Figura 2.5: Captura de pantalla de NetBeans para la creación de GUI

Usando la GUI de NetBeans podemos ajustar los contenedores visualmente, modificar las variables por medio del menú lateral, añadir elementos a los contenedores y ver el código autogenerado en la pestaña superior de *Source*, como se puede ver en la Figura 2.5.

El código autogenerado no es editable desde *Source*, pero siempre se puede añadir métodos extras y variables extras. En cuanto a los eventos de los elementos se tiene que seleccionar el objeto al que se va a añadir el evento y o bien añadirlo desde click derecho o desde la pestaña *Events* del menú.

2.2.2. Librerías usadas para la implementación del clasificador de microplásticos

En cuanto a los paquetes usados para la implementación del clasificador se usaron algunos de uso común como **NumPy** de Python y otras más específicas, tanto para inteligencia artificial, como para el tratamiento de datos e imágenes. Respecto a la aplicación de los conocimientos de redes neuronales se usó **Keras**. Keras es una capa de software que trabaja sobre **Tensorflow**, ofrece un conjunto de abstracciones más intuitivas y de alto nivel haciendo más sencillo el desarrollo de modelos de aprendizaje profundo o *Deep learning*. Tensorflow es una biblioteca de código abierto para aprendizaje automático desarrollado por Google. En 2017, el equipo de TensorFlow de Google decidió ofrecer soporte a Keras en la biblioteca de core de TensorFlow.

Para el tratamiento de las imágenes, ya fuera recortar, redimensionar o pegar, se usó *Python Imaging Library* (PIL), conocida actualmente como **Pillow**. Pillow es una biblioteca de código abierto para Python y ofrece varios procedimientos estándar para la manipulación de imágenes, entre ellos: manipulación por píxel, manejo de transparencia, aplicación de filtros a las imágenes, aplicación de mejoras para la imagen o agregar texto.

La visualización de los resultados se realizó con una biblioteca para la generación de gráficos a partir de datos, **Matplotlib** de Python. Matplotlib está basada en Matlab, pero su principal ventaja respecto Matlab es que es completamente gratuita, además de que no necesitamos toda la cantidad de *toolboxes* de Matlab.

En cuanto al tratamiento de los excels de los datos se usó la biblioteca **Pandas**. Está escrita como extensión de NumPy para manipulación y análisis de datos, en particular, ofrece estructuras de datos y operaciones para manipularlas. Además, ofrece la posibilidad de graficar esas estructuras de datos usando internamente Matplotlib.

Capítulo 3

Descripción y tratamiento de datos

3.1. Descripción de los datos

Para el estudio de la detección de microplásticos se usaron imágenes del grupo EOMAR [12], tomadas directamente en la playa delimitadas por un marco cuadrado de 50×50 centímetros.

Antes de usar las imágenes se procedió a recortarlas con Photoshop para que el marco de plástico quedara fuera del análisis, ya que al ser plástico el detector automático sobreestimaría la cantidad de plástico de la imagen y no obtendríamos resultados aproximados a la realidad. Además, los datos que tenemos de los archivos Excel solo tienen en cuenta las partículas que se encuentran dentro de la zona delimitada.

Una vez que todas las imágenes se encuentran recortadas en proporción 1:1, se realizó un pequeño *script* que se encarga de redimensionar las imágenes en función del tamaño que se requiera. Todo el conjunto de datos que tenemos se pueden clasificar en tres conjuntos de datos con distintas características.

3.1.1. Conjunto de datos A

Nueve imágenes, las cuales salen delimitadas con un marco de plástico. Las localizaciones de las imágenes corresponden tres a la playa de Bocabarranco al este de la isla de Gran Canaria, otras tres a la playa Cicer situada en la Playa de Las Canteras al norte de Gran Canaria y las últimas tres imágenes corresponden a la Playa de los Cuervitos. Hay que mencionar que la playa

de la Cicer es la única de las tres donde diariamente pasan vehículos con maquinaria que realizan el trabajo de limpieza y cribado de la arena.



Figura 3.1: Playa de la Cicer con microplásticos

Además de las imágenes tenemos un archivo Excel donde están los resultados por imagen del recuento de partículas y el peso en mg por tamaño de partícula en ambos casos. Para entender mejor la información que tenemos en el conjunto de datos A se han realizado una serie de gráficas. Primero vamos a visualizar que imagen o imágenes tienen mayor cantidad de miligramos de plástico.

En la Figura 3.2 podemos ver que aquellas que más plástico tienen son las fotos de la ubicación de la Cicer seguidas de Bocabarranco, ya que aunque esta localización tenga dos imágenes con pocos gramos, los gramos que tienen son de partículas grandes de más de 5mm.

En la Figura 3.3 podemos ver mejor la diferencia entre las partículas grandes

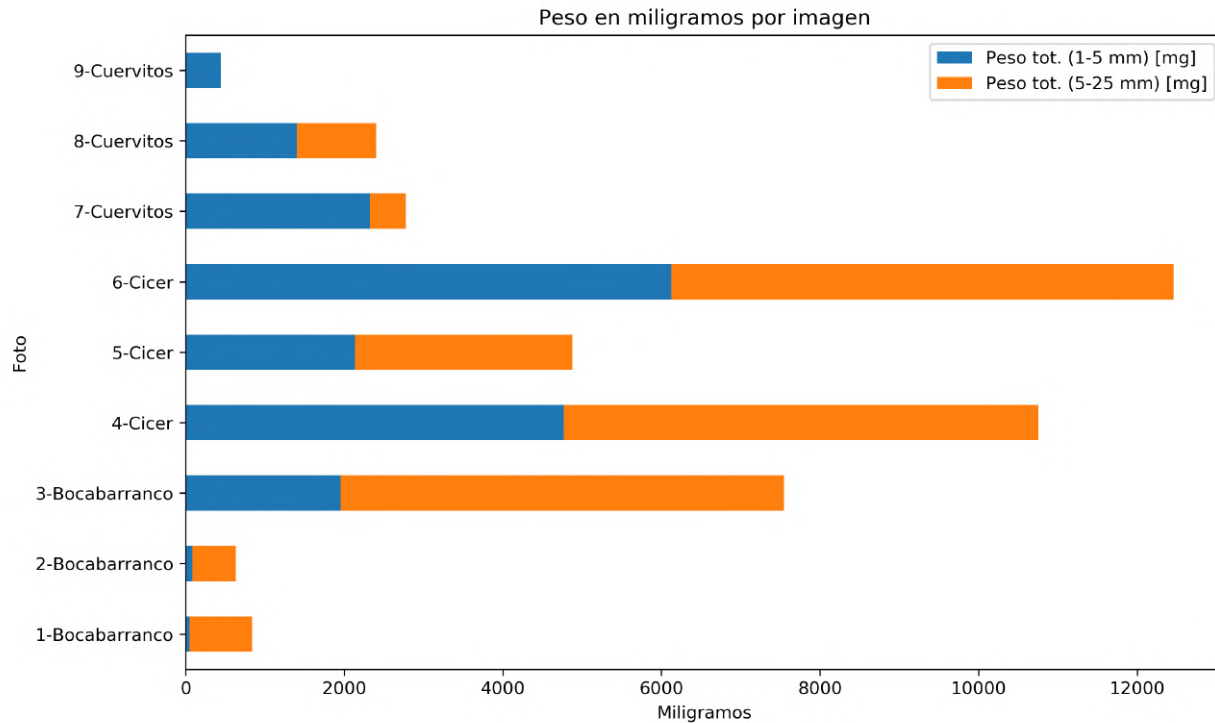


Figura 3.2: Peso de los microplásticos por imagen, del conjunto de datos A

y pequeñas entre localizaciones, donde en Bocabarranco y Cicer siempre hay más miligramos de plástico en partículas entre 5 y 25 milímetros. Mientras que por otro lado en Cuervitos hay más microplásticos de tamaño inferiores a 5 milímetros, incluso en la tercera muestra tomada no hay ni un solo segmento mayor a 5 milímetros, por lo que podríamos considerar que es la playa más limpia entre las tres.

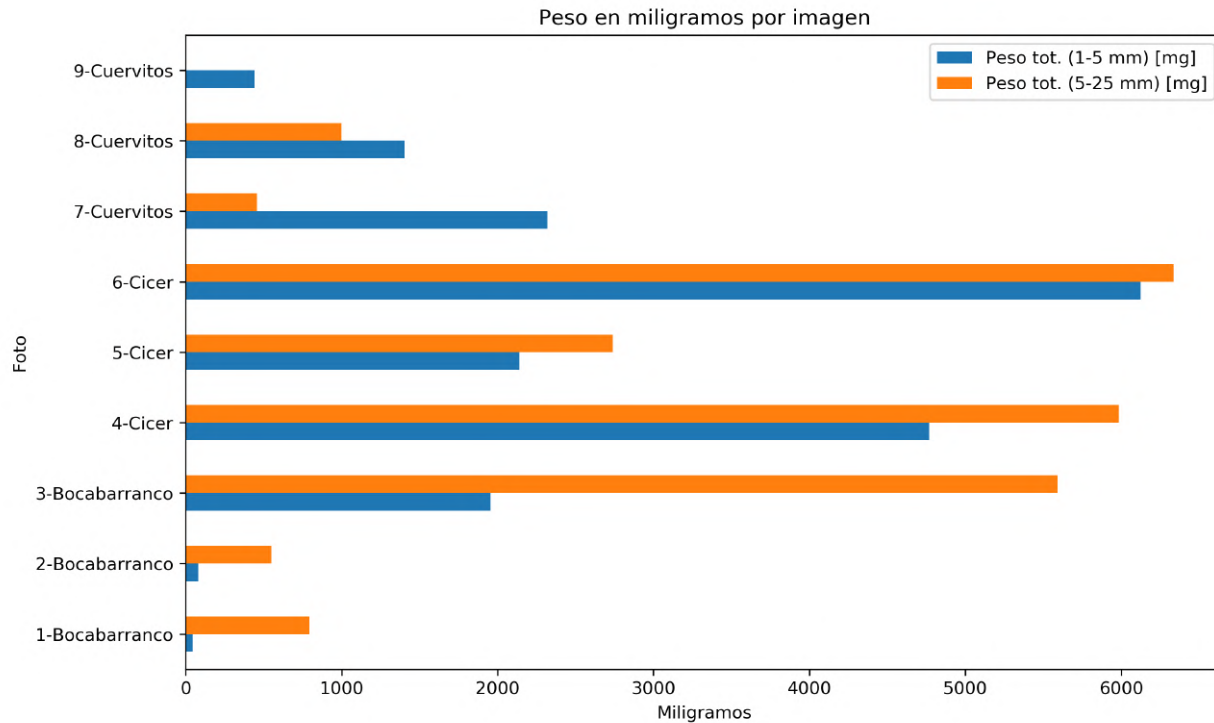


Figura 3.3: Peso de los microplásticos por imagen separados por tamaño, del conjunto de datos A

Para hacer un análisis más exhaustivo sobre qué tipo de basura se acumula en las playas se graficaron los datos donde se indican a qué tipo pertenece cada partícula que llegó al laboratorio. Primero analizaremos los microplásticos entre 1 y 5 milímetros. Como se puede ver en la gráfica de la Figura 3.4 los microplásticos más abundantes son los fragmentos, es decir microplásticos secundarios que proceden de desechos plásticos más grandes que se han deteriorado. El segundo tipo más común son los EPS/XPS/Foams o Poliestireno expandido, Poliestireno Extruido y Foams, estos plásticos se usan para producir envases, tablas de surf y protectores para el transporte de materiales, además de usarse en aislamientos y en construcción.

También podemos ver que en algunas imágenes como tercer lugar se encuentran los Pellets, estos pequeños trozos de plástico termomoldeable utilizados en los procesos de formación del plástico en moldes de extrusión o inyección, son un componente vital de casi todo proceso de fabricación del plástico.

Entre los fragmentos entre 5 y 25 milímetros encontramos también que los microplásticos predominantes son los fragmentos procedentes de plásticos más

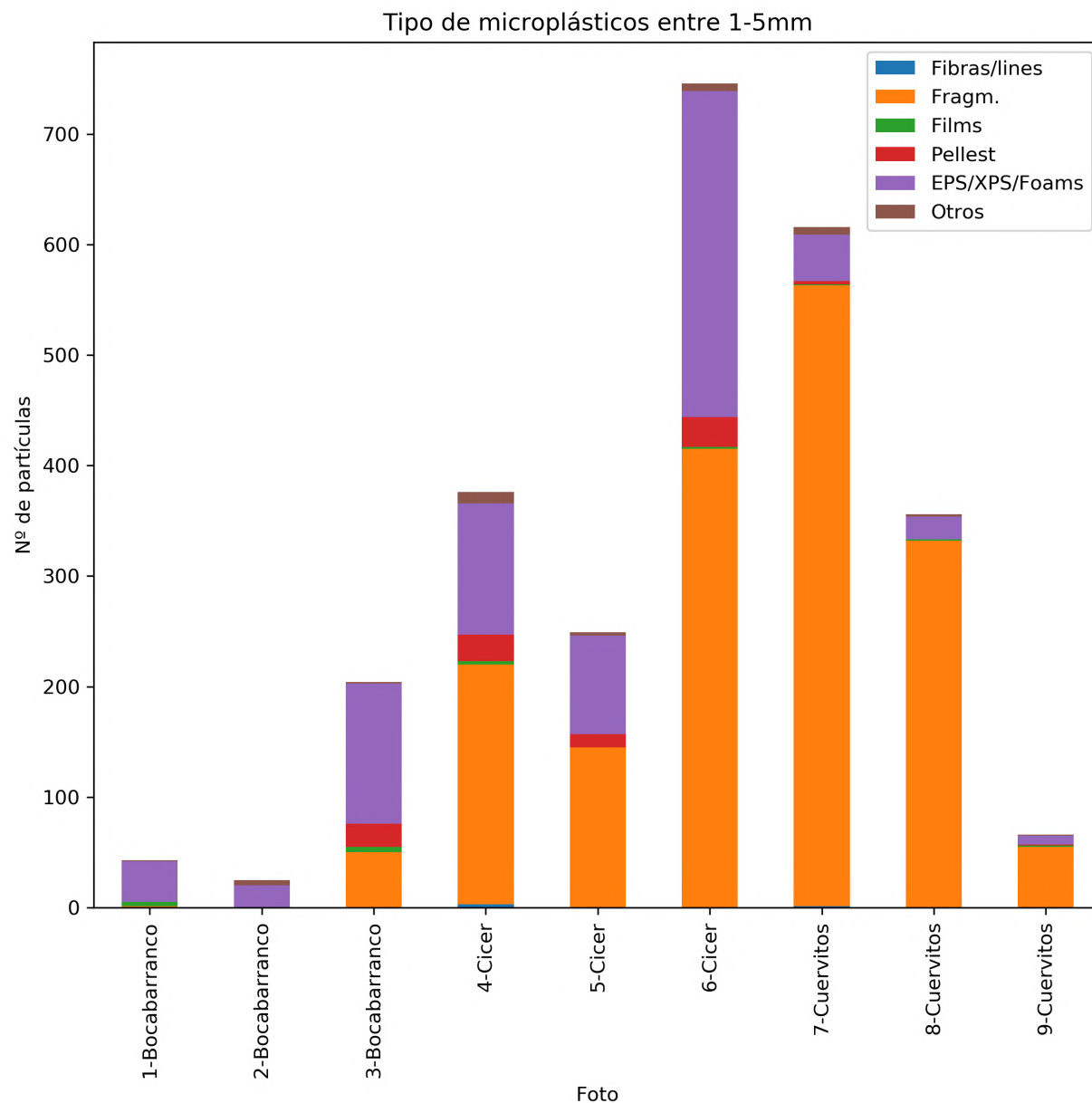


Figura 3.4: Partículas entre 1-5mm por imagen, del conjunto de datos A

grandes como se puede ver en la Figura 3.5. Otro dato curioso a destacar es que en comparación a las otras dos playas en la de Bocabarranco hay un número importante de colillas recogidas. Esto cabe mencionarse ya que es probable que esas colillas vengan de la propia playa y no de las corrientes marinas como puede ser el caso los fragmentos o el poliestireno.

Las colillas aunque tarden menos tiempo en degradarse contaminan litros de agua debido a las sustancias que liberan al contacto con la misma. En el

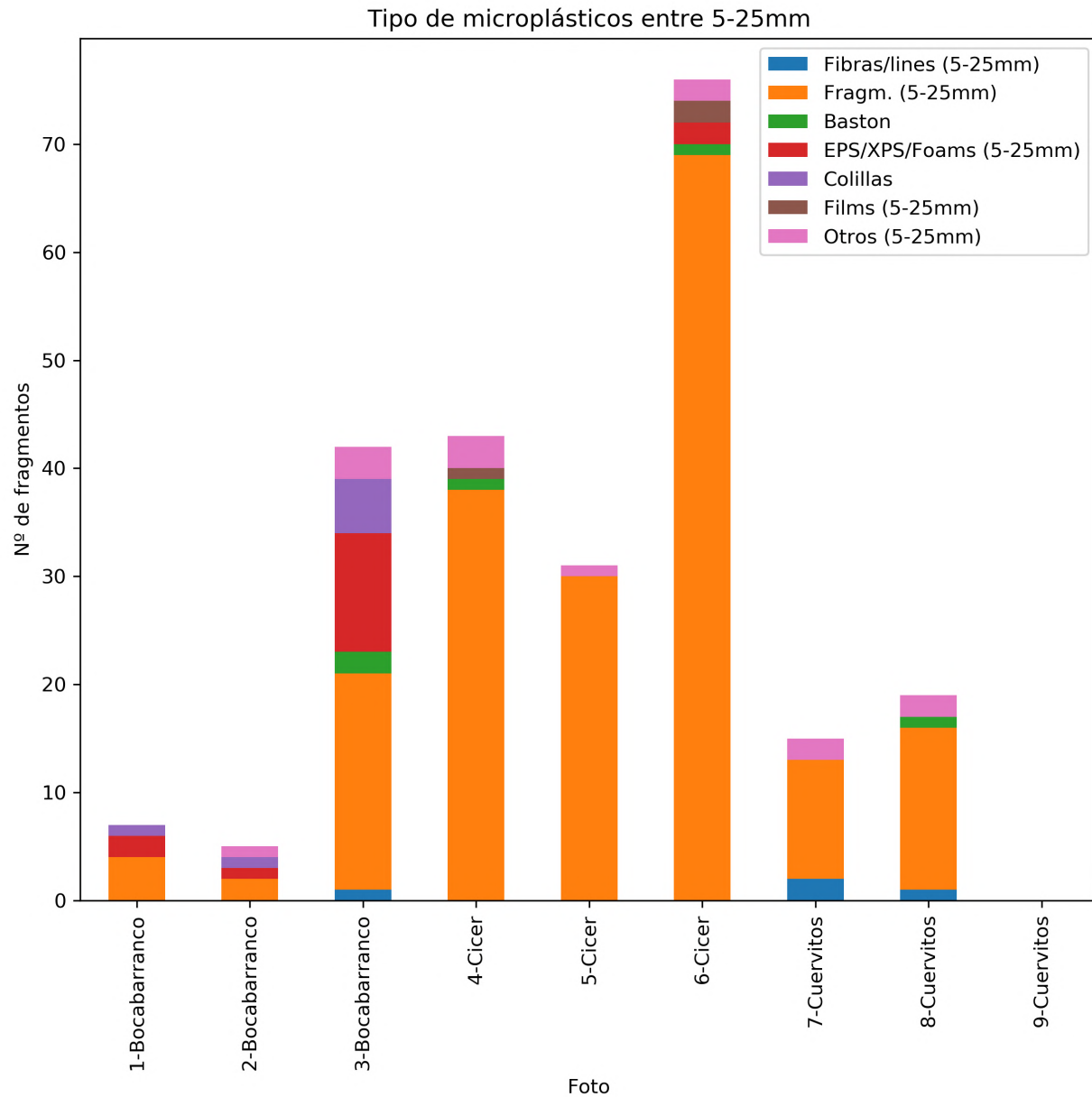


Figura 3.5: Partículas entre 5-25mm por imagen, del conjunto de datos A

filtro de los cigarros donde, según un informe del Comité Nacional para la Prevención del Tabaquismo [18], se acumula una gran parte de los componentes nocivos del tabaco, tales como, acetona, amoniaco o naftalina, compuestos presentes en algunos productos de limpieza del hogar, pero también de arsénico, polonio, metanol, disolventes industriales o elementos radiactivos.

3.1.2. Conjunto de datos B

Cincuenta y dos imágenes donde la delimitación de la zona está con un metro y dibujada en la arena. Las fotos corresponden a dos ubicaciones distintas la playa de Famara en la isla de Lanzarote y la playa de Lambra en La Graciosa.



Figura 3.6: Playa de la Lambra con microplásticos

Además hay un archivo Excel donde solo se tiene el peso en mg/m^2 en función del tamaño de la partícula. Se denomina una partícula micro cuando su tamaño se encuentre entre 1 y 5 milímetros, mientras que se denomina meso cuando su tamaño se encuentre entre 5 y 25 milímetros.

Para ver la evolución de la cantidad de microplásticos que se han encontrado los biólogos en las dos localizaciones, hemos separado los datos de Lambra y Famara para graficarlos individualmente ordenados por tiempo con las muestras que tenemos. Figura 3.7 para la playa de Famara y la Figura 3.8 para la playa de Lambra.

Como para una misma fecha y ubicación hay tres fotos se ha realizado la media de las tres para obtener un dato representativo de ese mismo día. Hay que tener en cuenta que la toma de las muestras no está tomada de forma periódica, por lo que entre las distintas muestras puede haber separaciones desde una semana hasta 2 meses.

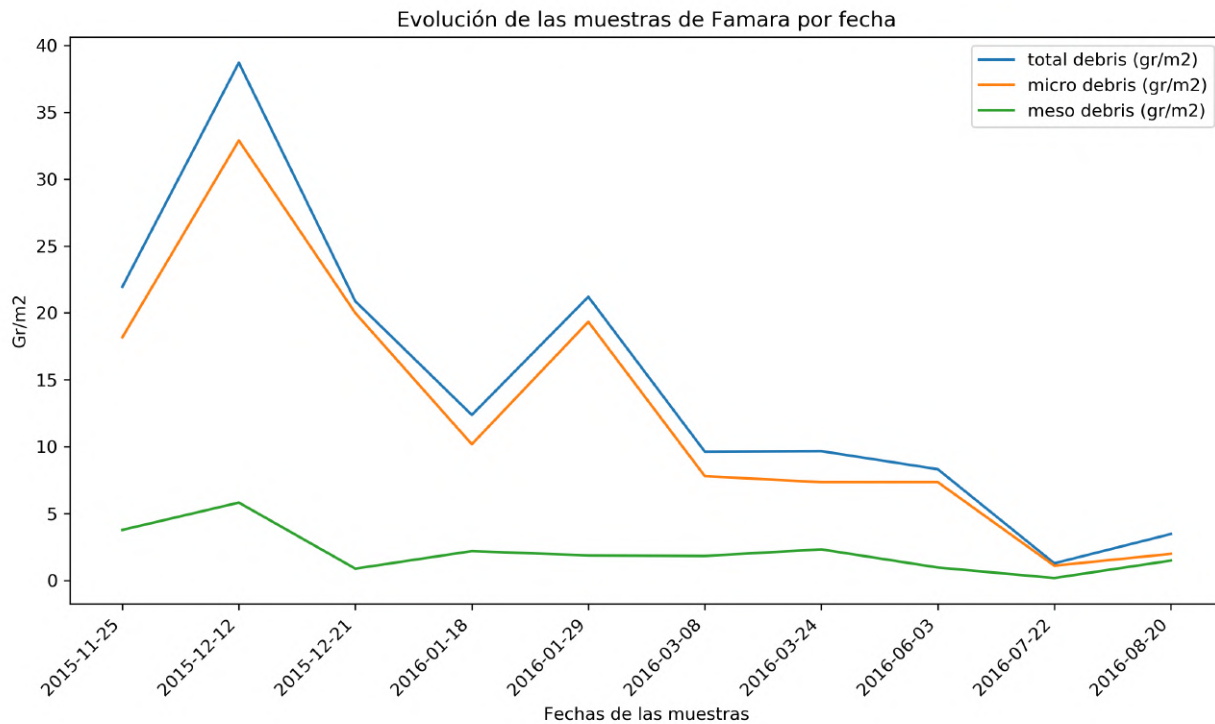


Figura 3.7: Cantidad de microplásticos en la playa Famara, con eje temporal

Además tenemos también una gráfica en la Figura 3.9 donde podemos observar para cada imagen si hay más cantidad de microplásticos con un tamaño de entre 1 y 5 milímetros, o bien entre 5 y 25 milímetros.

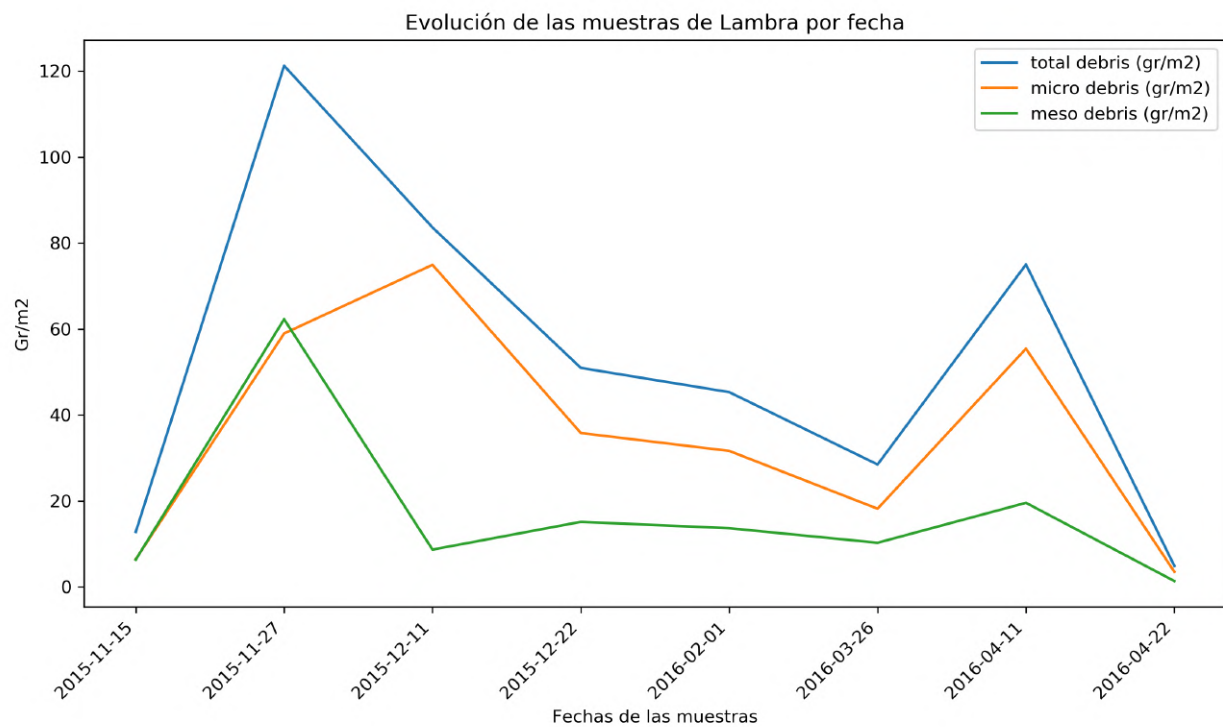


Figura 3.8: Cantidad de microplásticos en la playa Lambra, con eje temporal

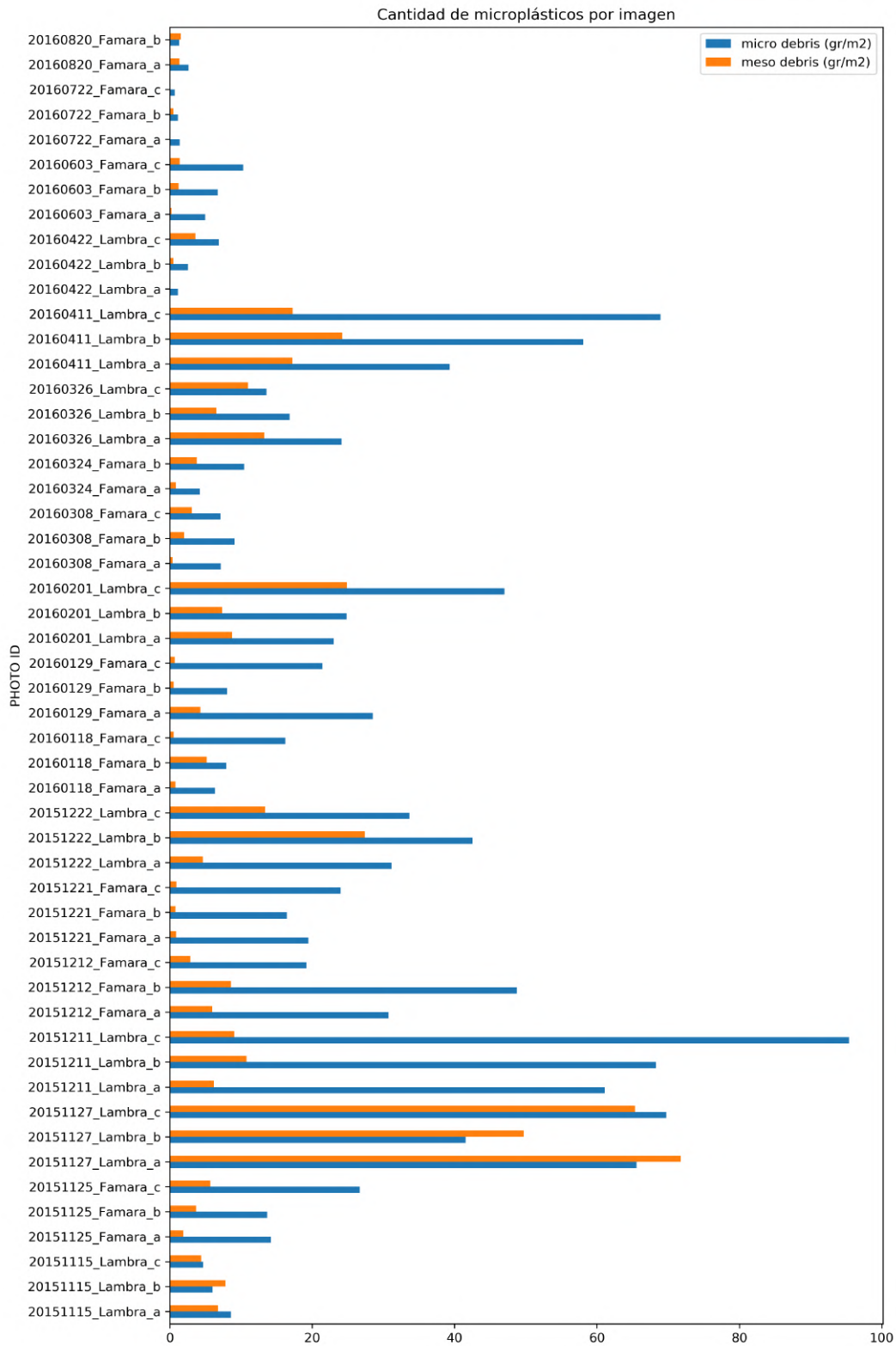


Figura 3.9: Peso de los microplásticos por imagen, del conjunto de datos B

3.1.3. Conjunto de datos C

Dieciocho imágenes delimitadas con un marco de plástico, donde no tenemos ningún Excel para saber el total de plásticos que hay en cada imagen. Las imágenes son tanto de La Laja, Cuervitos, Bocabarranco, Águila, Veneguera y de la Cicer, todas pertenecientes a la isla de Gran Canaria.



Figura 3.10: Playa de Cuervitos, arena delimitada con un marco de plástico

3.2. Aplicación de anotación

Para poder entrenar la red neuronal es necesario tener etiquetados los datos de entrenamiento, por eso se realizó un programa externo para poder mapear los segmentos de forma manual. El programa, realizado en Java, guarda los resultados del mapeo en un archivo CSV que posteriormente se leerá en Python para el entrenamiento de la red.

3.2.1. Interfaz gráfica de la aplicación

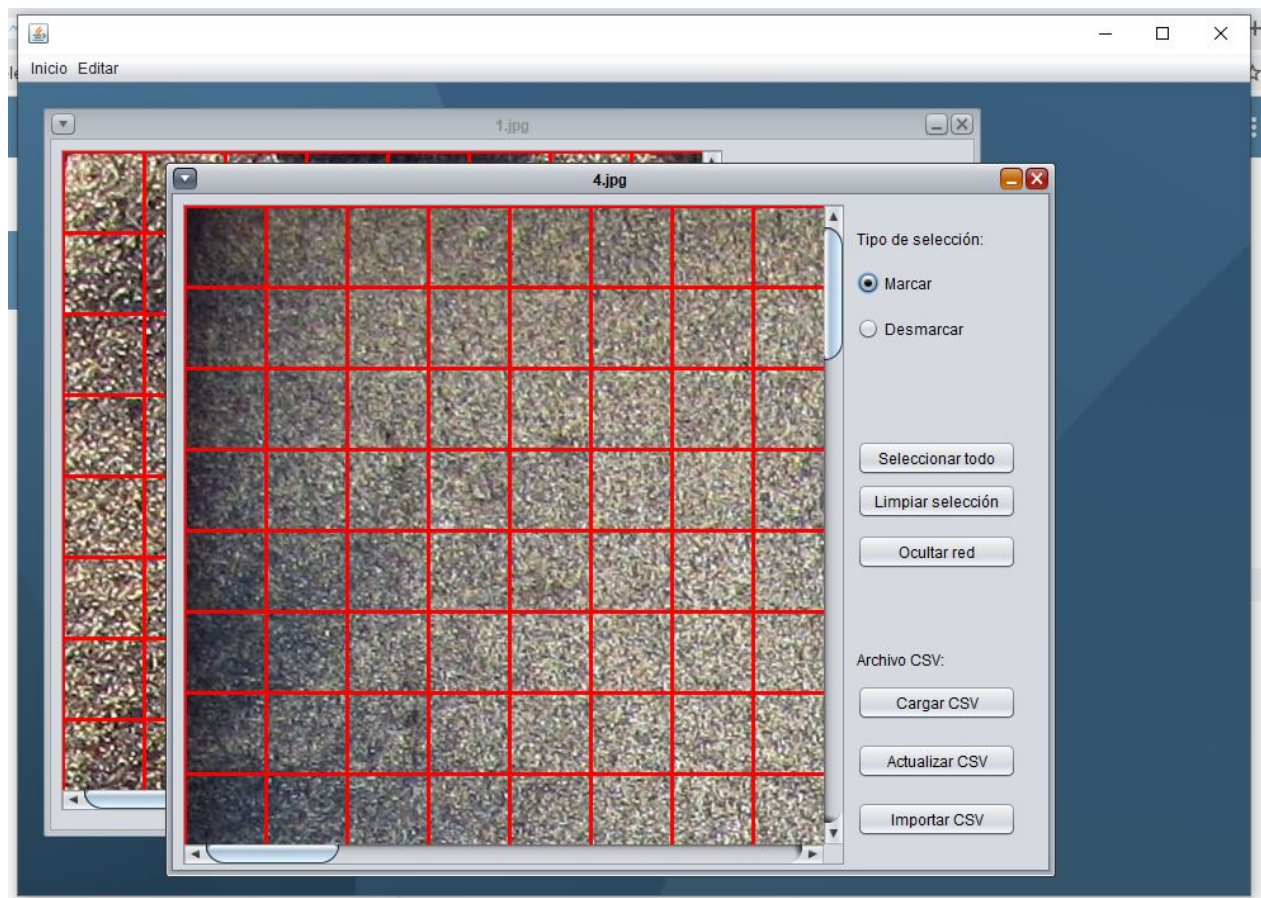


Figura 3.11: La aplicación escritorio con ventanas abiertas

La aplicación está basada en un panel *JDesktopPane*, el cual funciona como un escritorio, donde tiene ventanas internas, *JInternalFrame*, que se pueden minimizar, mover y cerrar sin que se cierre la aplicación.

Las ventanas internas corresponden a cada imagen que se abra, donde se visualizarán y se podrán marcar los segmentos que contengan plástico (Figura

3.11). Además como podemos ver en la Figura 3.12 la aplicación consta de un menú superior donde encontramos las opciones de abrir y seleccionar el tamaño de píxeles de la cuadrícula de segmentación.

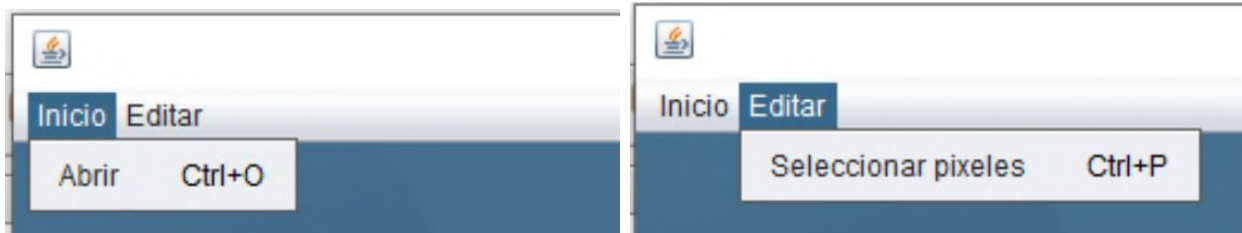


Figura 3.12: Menú en la aplicación

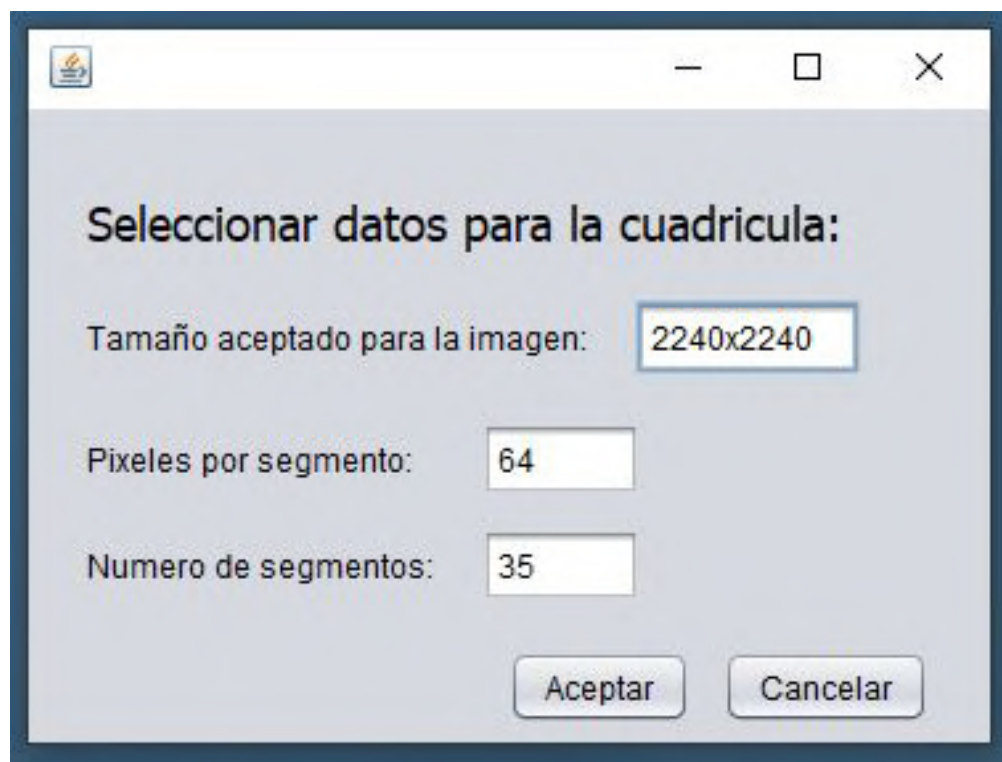


Figura 3.13: Ventana emergente para la modificación del tamaño de segmento

En la opción de selección de píxeles podemos observar en la Figura 3.13 que se indica el tamaño de la imagen 2240 (también se acepta el tamaño de 2048 píxeles), el tamaño en píxeles del segmento y el número de segmentos que tiene que tener la imagen. El usuario puede modificar o bien los píxeles o bien el número de segmentos, el otro campo se modificará automáticamente. Si modificamos el tamaño a 2048 se modificarán también los otros dos campos de forma automática. Hay que tener en cuenta que la aplicación solo permite

segmentos de tamaño 16,32 ó 64 píxeles, por lo que está limitada a esas opciones.

En cuanto a la opción de abrir imagen se despliega un *JDialog* (Figura 3.14) que si se selecciona cualquier otro archivo que no sea una imagen da mensaje de error y no abre el panel interno.

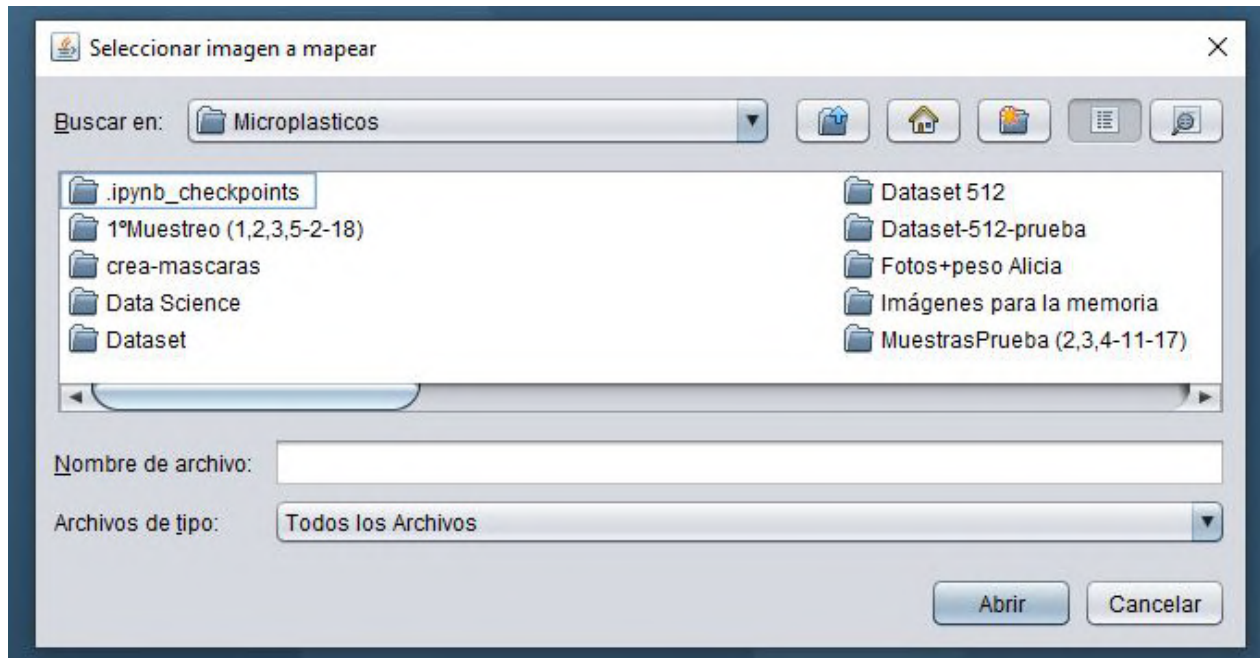


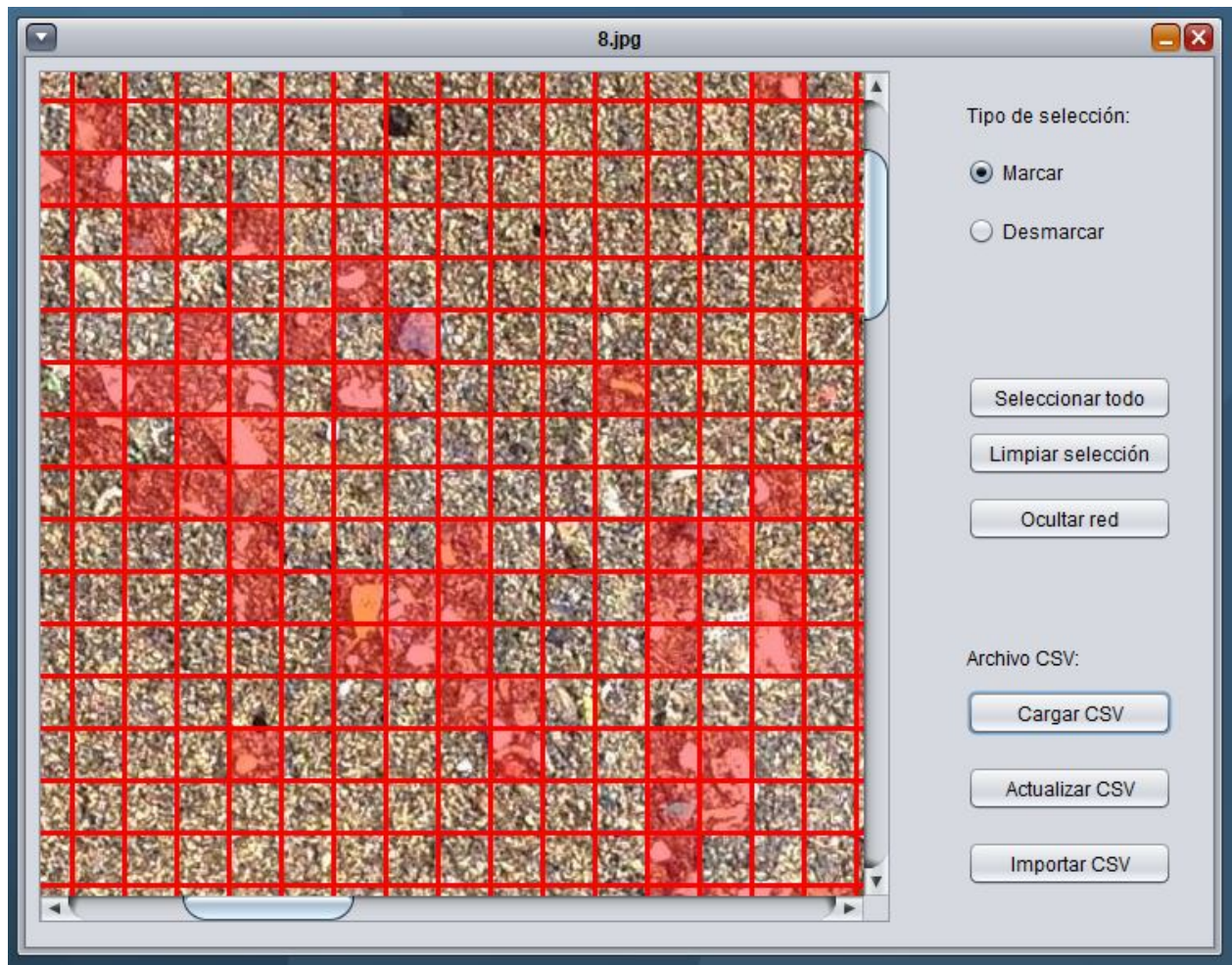
Figura 3.14: Ventana emergente para la selección de la imagen a mapear

Si al abrir la imagen no hay ningún error se mostrará un *JInternalFrame* donde se visualiza la imagen con una rejilla dentro de un *JScrollPane*, además encontramos las opciones para poder mapear la imagen. Figura 3.15.

A la derecha del panel de la imagen encontramos un grupo de radio *buttons* y una serie de botones. Si el *radio button* está con la opción de “Marcar” al hacer click en una casilla de la imagen se marcará en rojo, por otro lado si está con la de “Desmarcar” se quitará el color si está marcado.

Por otro lado el botón de “Seleccionar todo” deja todas las casillas marcadas y el “Limpiar selección” deja todas desmarcadas. La selección de casillas se guarda internamente con una matriz del tamaño del número de segmentos que tenga la imagen. También tiene selección múltiple con el click derecho y arrastrando el cursor.

También tenemos el botón de “Ocultar red” ya que para tamaños de segmen-

Figura 3.15: Interfaz del *JInternalFrame*

tos muy pequeños la red no permite ver con claridad lo que hay en la imagen y dificulta la marcación. Figura 3.16.

En cuanto al guardado de la marcación se guarda en un archivo CSV cuando se pulsa en el botón “Actualizar CSV”, el archivo se guarda en una carpeta llamada CSV que se encuentra dentro del proyecto. El nombre del archivo se autogenera en función del nombre de la imagen y del tamaño de segmento seleccionado, ejemplo: “*DatosImg8Seg32.csv*”.

Cada vez que se abre una imagen si queremos cargar los datos del CSV para visualizarlos tenemos que pulsar en “Cargar CSV”. Si no encuentra el CSV con el nombre autogenerado que en la ubicación por defecto (Figura 3.17) habrá que proceder a abrirlo con la opción de “Importar CSV” donde podemos indicarle que CSV queremos abrir. Cuando pulsemos en el botón de “Importar

CSV” se nos abrirá un *JDialog*, Figura 3.18, para que seleccionemos el archivo. Luego de seleccionar que CSV queremos importar tenemos que seleccionar que tamaño de píxel tiene la rejilla con la que se guardaron los datos de mapeado, hay que elegir entre 16, 32 y 64 píxeles.



Figura 3.16: Imagen sin y con rejilla de 16x16 píxeles



Figura 3.17: Ventana de error cuando no existe un CSV

También podemos usar el Importar cuando tenemos las anotaciones de segmentos grandes y queremos trabajar con segmentos más pequeños, así cargamos las anotaciones y solo tenemos que retocar, facilitando así el trabajo.

Al salir de la aplicación saldrá un *JDialog* que nos preguntará si deseamos cerrar la aplicación Figura 3.19.

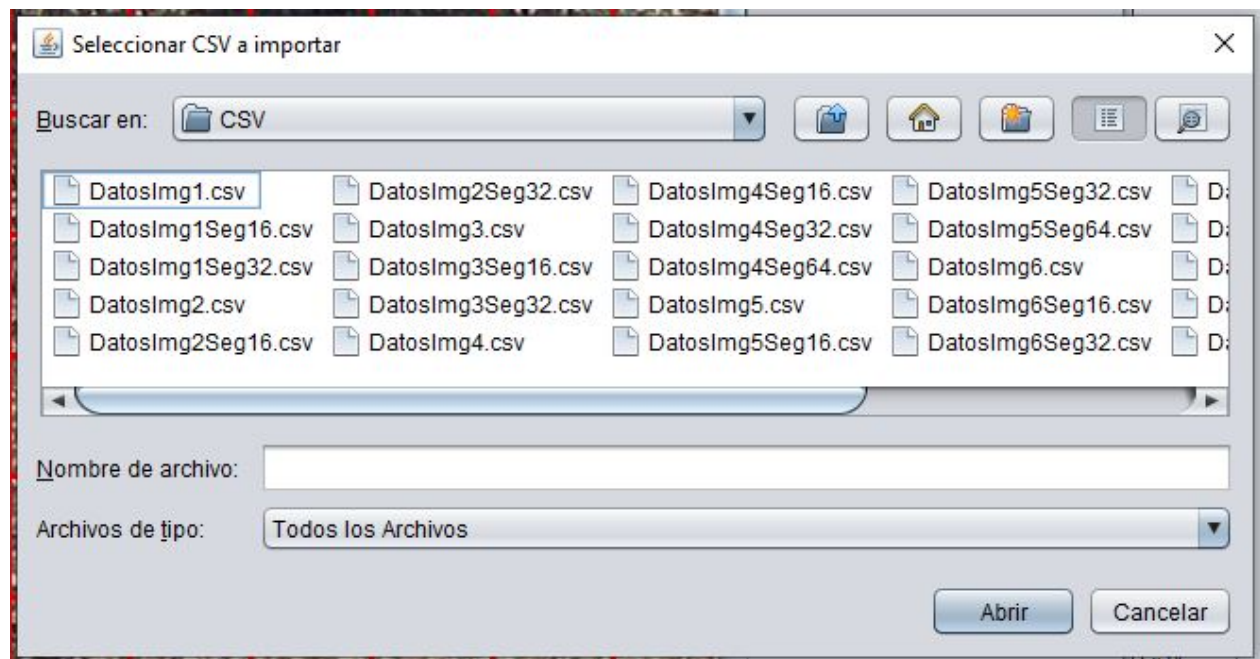


Figura 3.18: Ventana para seleccionar el CSV a abrir

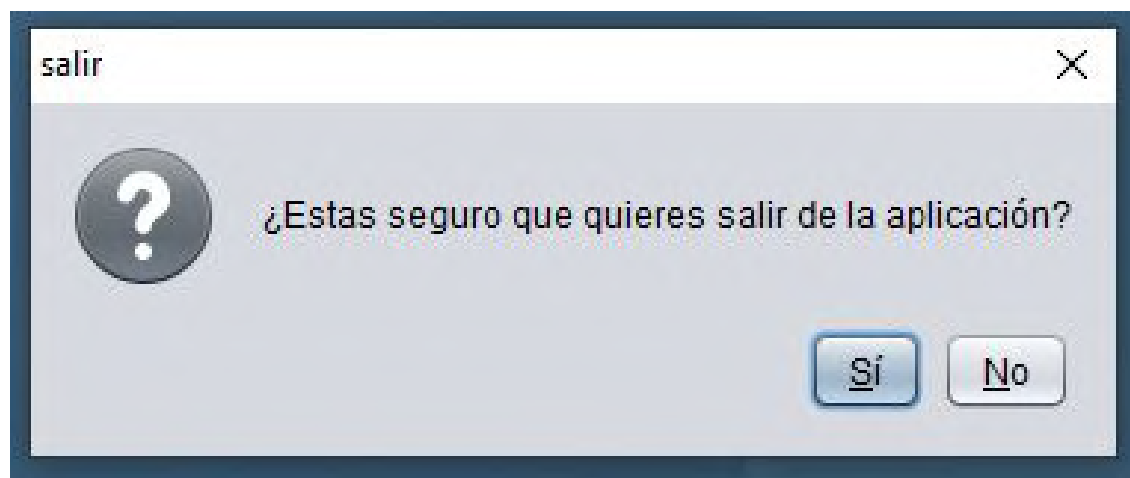


Figura 3.19: Ventana con mensaje de confirmación para cerrar la aplicación

3.2.2. Estructura del código de la aplicación

- **Escritorio:** Clase de Java donde se encuentra el *main* del programa. Es donde se crea e inicializa el *JDesktopPane*, se encuentran los eventos del menú para abrir imagen y modificar los segmentos y se generan las ventanas internas.
- **SelectDataToSegment:** Se genera una ventana donde se cambia el tamaño de segmento y el número de segmentos. Esos valores se almacenan y si escritorio los necesita usa los métodos *get*.
- **SegmentationInterface:** Clase de los *JInternalFrame* de la aplicación. Esta ventana interna consta de un *JScrollPane* donde se pinta el *PanelImagen* y en función de que evento se active llamará a otras clases para modificar los valores del objeto. El método más importante que contiene esta clase es el que se encarga de calcular la posición del cursor en función de la ventana y el *scroll* para poder marcar su posición correspondiente a la máscara.
- **Segmentation:** Clase donde se crea y se almacena la matriz de la máscara, cualquier modificación de la matriz se realiza llamando a los métodos de la clase *Segmentation*. El tamaño de la matriz se especifica en el constructor.
- **Imagen:** Pequeña clase donde tenemos cargada la imagen con *BufferedImage* y almacenamos su tamaño.
- **PanelImagen:** Clase *JPanel* donde se refresca la imagen constantemente añadiendo la rejilla si el botón “Ocultar red” no se encuentra pulsado además de los segmentos marcados con la máscara.
- **FileCSVClass:** Esta clase se encarga de la lectura y guardado del archivo CSV. Para la lectura de CSV obtiene la información y la pasa a un objeto de tipo *Segmentation*. En cuanto a la escritura lee el objeto *Segmentation* y va insertando tanto el nombre correcto del segmento como 0 si es arena y 1 si es plástico en un archivo CSV.

Capítulo 4

Experimentos

En este capítulo se describirán las distintas pruebas y procedimientos que se realizaron, además de los resultados obtenidos.

4.1. Métrica para la evaluación de resultados

Para analizar los resultados de aprendizaje de las distintas arquitecturas de redes neuronales usaremos la matriz de confusión [9]. Esta herramienta permite visualizar el desempeño de un algoritmo de clasificación, ya que, mostrará cómo se está clasificando dicho algoritmo a partir de un conteo de aciertos y errores para cada una de las clases.

Valor real	Positivos	Verdaderos Positivos (VP)	Falsos Negativos (FN)
	Negativos	Falsos Positivos (FP)	Verdaderos Negativos (VN)
		Predicción Positivos	Predicción Negativos

Figura 4.1: Matriz de confusión para un modelo binario

Cada columna de la matriz representa el número de predicciones de cada clase, mientras que cada fila representa cada una de las clases reales. Uno de los beneficios de las matrices de confusión es que permite ver si el sistema está confundiendo las diferentes clases del modelo.

Por ejemplo, en un modelo binario donde consideramos que un resultado puede ser positivo o negativo obtendríamos una matriz de confusión como la de la Figura 4.1. Donde VP es la cantidad de positivos que fueron clasificados correctamente como positivos por el modelo, VN es la cantidad de negativos que fueron clasificados correctamente como negativos por el modelo, FN es la cantidad de positivos que fueron clasificados incorrectamente como negativos (Error tipo 2, Falsos Negativos) y FP es la cantidad de negativos que fueron clasificados incorrectamente como positivos (Error tipo 1, Falsos positivos).

Para nuestro problema de clasificar arena y plástico nos encontramos que obtendremos una matriz de confusión como en la estructura de la Figura 4.2. Donde en “Aciertos arena” estará el porcentaje de muestras que están etiquetadas como arena y la red a predicho que son arena. En “Fallos arena” se encuentra el porcentaje de muestras que están etiquetadas como arena y la red predijo como como plástico. En “Fallos plástico” se encuentran el porcentaje de muestras que tienen partículas de plástico y la red ha clasificado como arena. Y en “Aciertos plástico” se encuentran el porcentaje de muestras que están etiquetadas como plástico y la red ha clasificado como plástico.

Valor real	Arena	Aciertos Arena	Fallos Arena
	Plástico	Fallos Plástico	Aciertos Plástico
		Predicción Arena	Predicción Plástico

Figura 4.2: Matriz de confusión para las clases arena y plástico

Los resultados obtenidos en esta matriz se utilizan para calcular distintas métricas (ver Figura 4.3). La **exactitud** (o *accuracy* en inglés) de un modelo se usa para referirse a como de cerca está el modelo de la medición real, es básicamente, el número total de predicciones correctas dividido por el número total de predicciones. En cuanto a la **precisión** identifica la frecuencia con la que un modelo predijo correctamente esa clase. Otra métrica que usamos es el **recall** que para una clase dada expresa como de bien puede el modelo detectar a esa clase. Si normalizamos los valores de la matriz de confusión en los aciertos de cada clase obtendríamos el cálculo del *recall*. A partir de esto se pueden dar cuatro casos [9].

- **Alta precisión y alto *recall***: el modelo maneja perfectamente esa clase.
- **Alta precisión y bajo *recall***: el modelo no detecta la clase muy bien, pero cuando lo hace es altamente confiable.
- **Baja precisión y alto *recall***: La clase detecta bien la clase pero también incluye muestras de otras clases.
- **Baja precisión y bajo *recall***: El modelo no logra clasificar la clase correctamente.

$$\begin{aligned}
 \text{Exactitud o Accuracy} &= \frac{\text{Orange} + \text{Blue}}{\text{Orange} + \text{Blue} + \text{Green} + \text{Red}} \\
 \text{Precisión arena} &= \frac{\text{Orange}}{\text{Orange} + \text{Green}} & \text{Precisión plástico} &= \frac{\text{Blue}}{\text{Blue} + \text{Red}} \\
 \text{Recall arena} &= \frac{\text{Orange}}{\text{Orange} + \text{Red}} & \text{Recall plástico} &= \frac{\text{Blue}}{\text{Blue} + \text{Green}}
 \end{aligned}$$

Figura 4.3: Formulas para el análisis de resultados

4.2. Determinación del tamaño de segmento

Una de las pruebas imprescindibles a realizar es saber con qué tamaño de celda obtenemos mejores resultados. Hay que tener en cuenta que al recortar la imagen esos fragmentos tendrán información aislada, es decir, según nuestra arquitectura la red no tendrá en cuenta las celdas adyacentes, por lo que, si estos son muy pequeños la red no será capaz de identificar que hay en la imagen, mientras que si son muy grandes no obtendríamos una aproximación cercana a la zona delimitada con microplásticos. Para ello realizaron las pruebas con segmentos de tamaño 64, 16 y 32 píxeles, en orden cronológico, entrenados con la misma arquitectura.

4.2.1. Celdas de 64 píxeles

El objetivo de esta primera prueba es detectar las partículas de plástico usando redes neuronales con aprendizaje supervisado, es decir, durante el entrenamiento la red ya sabe el resultado que tiene que aprender.

Para la detección de los segmentos de plástico se trabajaron con las 9 imágenes del conjunto de datos A, todas las imágenes se recortaron por el marco delimitador en proporción 1:1, se redimensionaron a tamaño de 2240 píxeles y se segmentaron en cuadrados de 64 píxeles. Para la segmentación de la imagen se usó la librería *Python Imaging Library* (PIL), conocida actualmente como Pillow. Cada segmento fue nombrado de la siguiente manera: “imgD+(nombre de la imagen)+R(Row)+(número de la fila)+C(Colum)+(número de la columna)”, por ejemplo, “imgID1-R0C0.jpg”.

Como se trabajó con redes supervisadas es un requisito mapear manualmente los segmentos que tienen plástico a los que son arena, para ello se usó la aplicación de Java descrita en el apartado anterior. Donde se obtuvo un archivo CSV el cual para cada segmento se indica un 0 si es arena o un 1 si es plástico.

En la gráfica 4.4 podemos observar un balance de la cantidad de fragmentos que están mapeados como plástico por imagen. Como se puede ver en el gráfico las imágenes con mayor densidad de plástico son la 3 y la 6, seguida de la 4.

Para el entrenamiento de la red, vamos a pasarle los segmentos individualmente por lo que no resulta práctico mantener los fragmentos de cada imagen

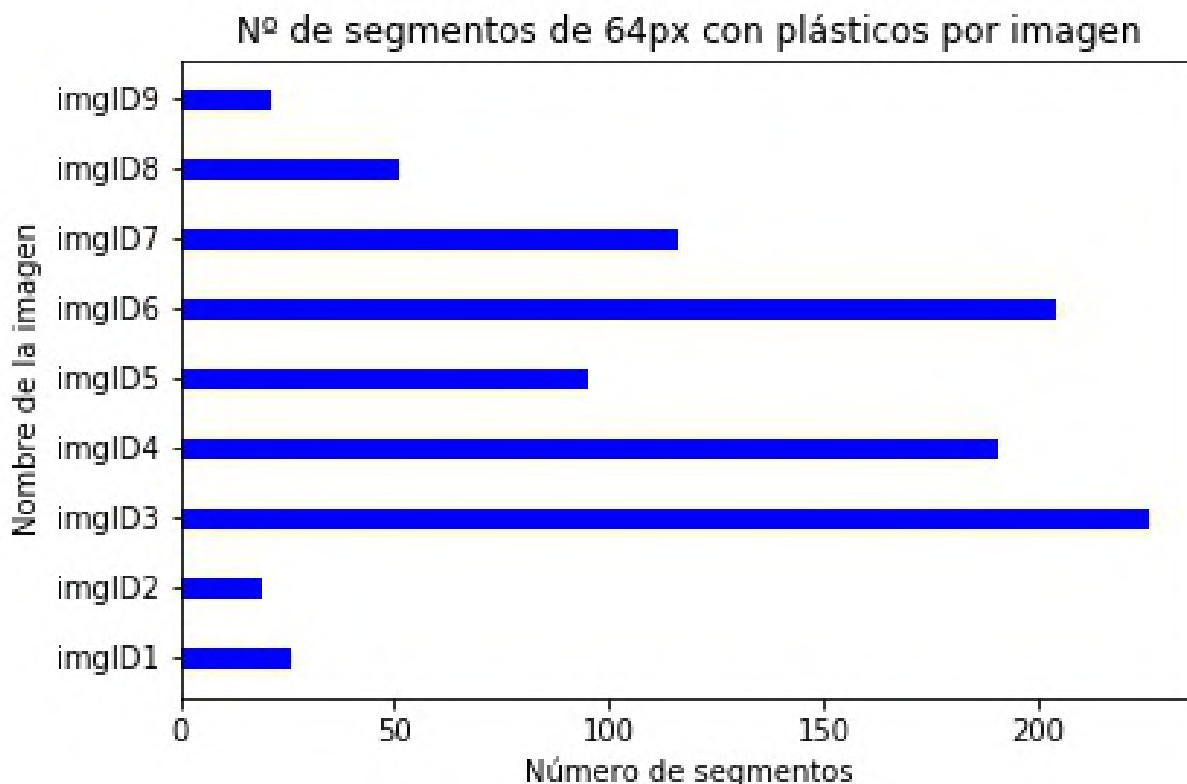


Figura 4.4: Número de segmentos que contienen microplásticos por imagen, 64 píxeles

separados durante el proceso. Así que se procedió a juntarlos todos y ver la proporción de plástico y arena que se tenía en total.(Figura 4.5).

Como podemos observar la escasez de segmentos con plástico en comparación con los de sin plástico es notable, tenemos únicamente 949 segmentos con plástico y unos 10076 de arena, así pues, el siguiente paso es balancear las clases.

Entrenar una red con las clases tan desbalanceadas como se encuentran actualmente afectaría a los algoritmos en su proceso de generalización de la información. No podemos pretender que logre diferenciar una clase de otra pasándole 10/11 partes de las fotos de una sola clase. Lo más probable que la red se limite a responder siempre «arena » puesto que así tuvo un acierto del 92 % en su fase de entrenamiento.

Para balancear las clases se hará uso de técnicas de *Data Augmentation*, usaremos diversas rotaciones con los segmentos de plástico septuplicando los datos de esa clase.

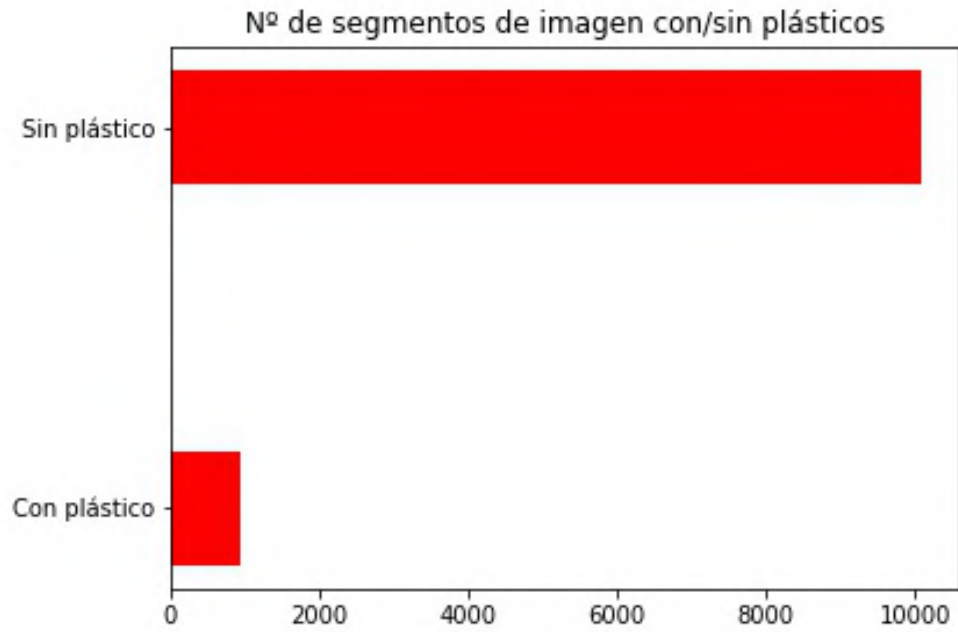


Figura 4.5: Total de segmentos que hay de arena y plástico, 64 píxeles

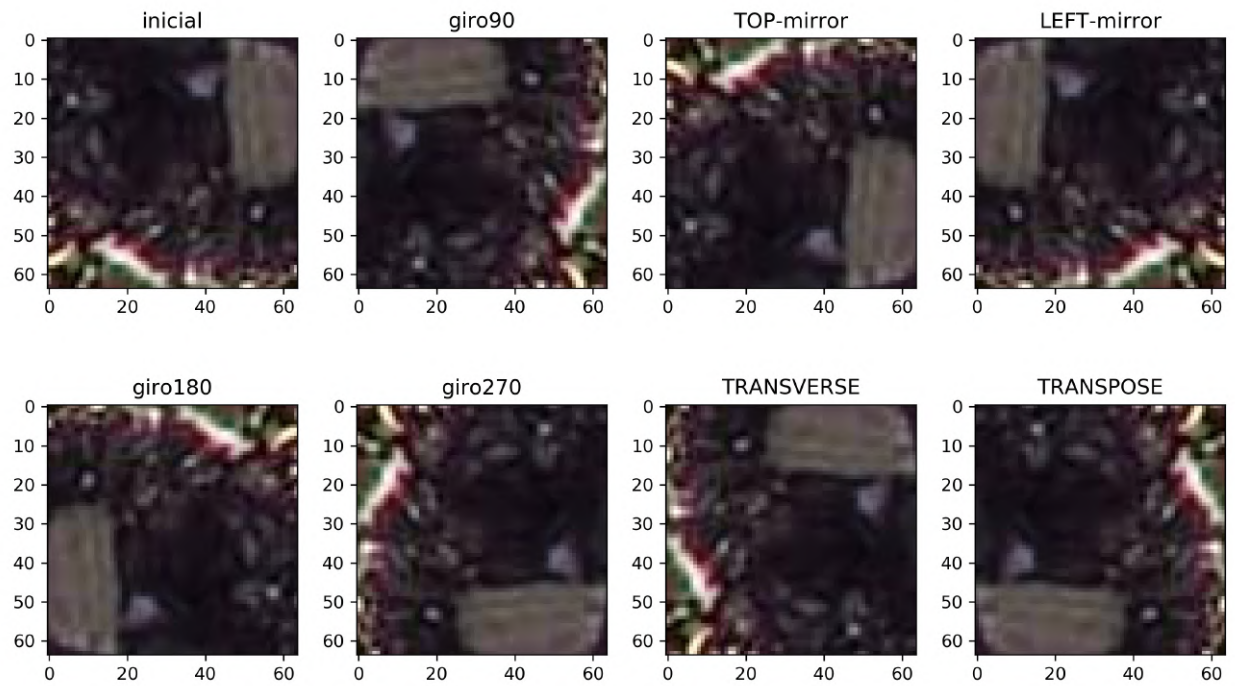


Figura 4.6: Transformaciones a realizar

Para cada segmento se realizaron las siguientes rotaciones visibles en la Figura 4.6:

- Giro de 90 grados
- Giro de 180 grados
- Giro de 270 grados
- Espejo de la imagen inicial (LEFT-mirror)
- Giro de 90 grados en espejo (TRANSPOSE)
- Giro de 180 grados en espejo (TOP-mirror)
- Giro de 270 grados en espejo (TRANSVERSE)

Una vez realizado el Data Augmentation volvemos contar todos los segmentos para visualizar la proporción de las clases, Figura 4.7.

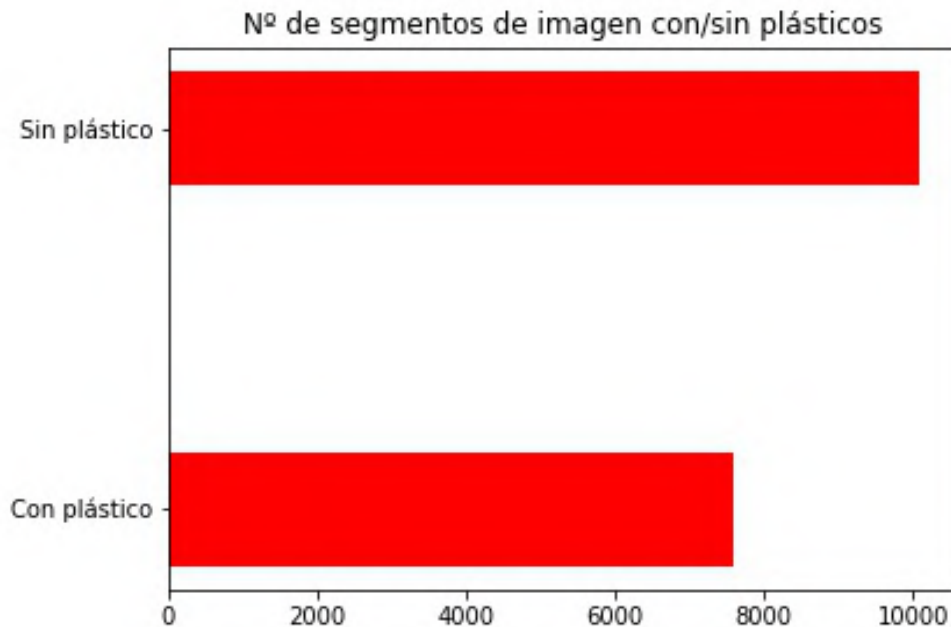


Figura 4.7: Total de segmentos que hay de arena y plástico con datos aumentados, 64 píxeles

La cantidad de segmentos con plástico ha aumentado a 7592. Como todavía sigue siendo notable la diferencia de clases se seleccionaran unos 7592 segmentos sin plástico de forma aleatoria para igualar ambas clases.

Una vez ya igualadas las clases se procede a dividir los datos en un conjunto de entrenamiento y uno de test. Para el conjunto de entrenamiento o *train*

se selecciona un 70 % de las imágenes manteniendo la misma proporción de ambas clases y para el de test un 30 % manteniendo también la misma proporción de clases. Una vez separados hay que desordenar estos conjuntos ya que si a la red le pasamos primero todos los segmentos de arena no generalice bien para los de plástico después.

El último paso antes de pasarle los datos a la red es cargar y normalizar los valores de los píxeles. Los colores de los píxeles tienen valores que van de 0 a 255, haremos una transformación de cada píxel y quedándonos con un valor entre 0 y 1.

La clasificación binaria que estamos realizando con una arquitectura de red neuronal sencilla basta. Así que la red estará estructurada por 2 capas convolucionales con *maxpooling* y 3 capas densas, donde la última capa densa es una neurona con una función de activación sigmoide. Como optimizador se usó Adam. En la Figura 4.11 se puede visualizar la arquitectura de forma más detallada

Para el entrenamiento seleccionamos 10 épocas o *epoch* y observamos los resultados obtenidos. En la gráfica de la Figura 4.8 podemos observar el número total de predicciones correctas dividido por el número total de predicciones. Como observamos a partir de la época 4 la exactitud de la predicción o *accuracy* de test empieza a bajar, esto se debe a que se está produciendo *overfitting*, o dicho de otra manera, la red está memorizando el conjunto de entrenamiento y fallando en la generalización de la información.

Por otro lado tenemos también la gráfica de la Figura 4.9 donde se muestra el valor de la función pérdida del entrenamiento, también se observa que a partir de la época 4 se pierden menos valores del conjunto de entrenamiento pero pierde más en el de test, por lo tanto está fallando en las predicciones.

Para evitar que guardemos el modelo con los pesos de tal forma que tenga *overfitting* vamos guardando el modelo después de cada época solo si la función de pérdida del conjunto de test ha bajado. Conservando así el mejor resultado posible del entrenamiento.

Una vez que tenemos el mejor resultado del entrenamiento usaremos una matriz de confusión visualizando los aciertos tiene de cada clase con las imágenes de test. La matriz de confusión para nuestro modelo es la correspondiente con la Figura 4.10. Donde tenemos un 94 % de acierto para las muestras de arena

y un 94 % para las muestras con plástico.

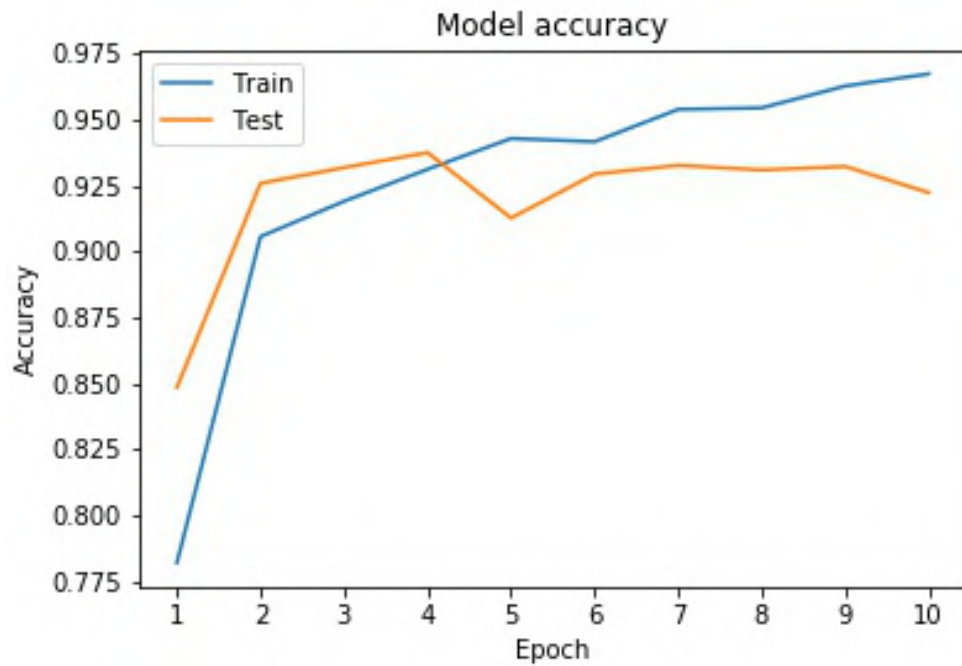


Figura 4.8: Exactitud de predicción del modelo durante el entrenamiento y test

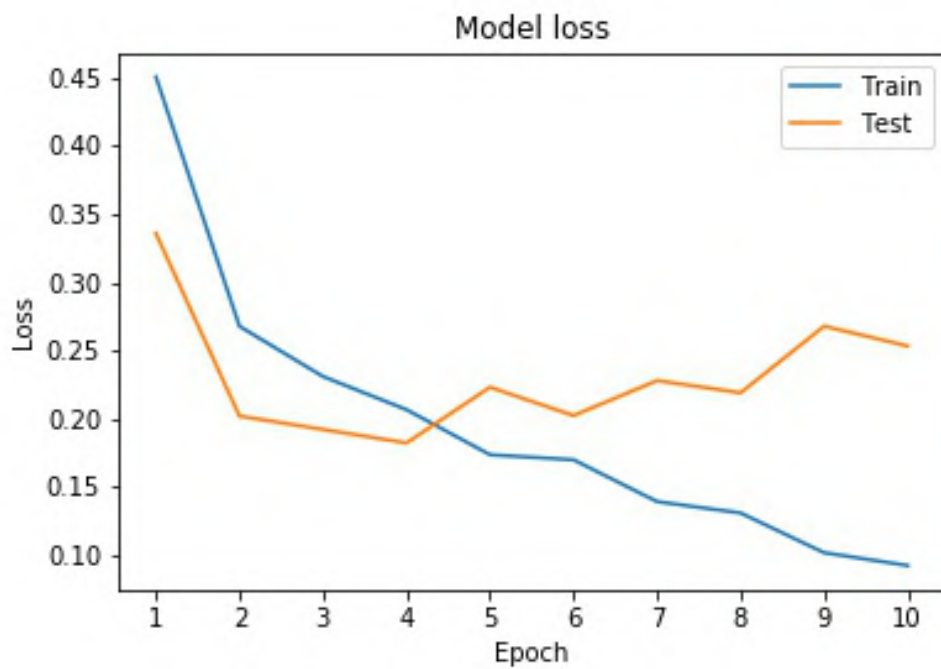


Figura 4.9: Función de pérdida del entrenamiento y test

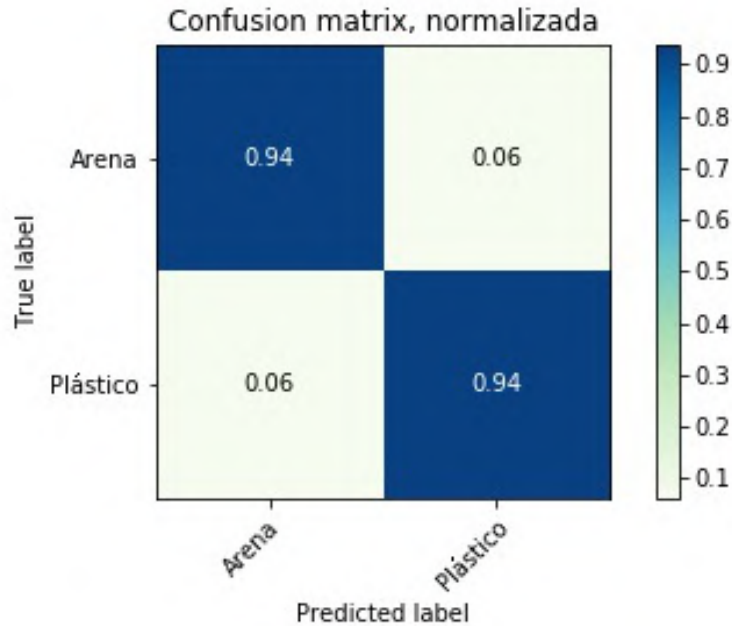


Figura 4.10: Matriz de confusión para segmentos de 64 píxeles

Para comprobar los resultados de la red le pasaremos todos los fragmentos de una imagen del conjunto de datos B para que prediga que segmentos considera que tienen partículas de plástico y cuales considera que es arena. Primero hay que realizar el mismo proceso que con las anteriores imágenes, recortarla en proporción 1:1, redimensionarla a 2240 píxeles y segmentarla en 64 píxeles.

Una vez le pasemos los segmentos a la red para que prediga visualizaremos la imagen pintando todos los fragmentos que considere que no tienen plástico de negro. Como podemos observar en la fotografía de la Figura 4.13 la separación de arena a plástico está bien, pero al ser partículas tan pequeñas sigue habiendo más arena que partículas, por lo que, si intentamos estimar la cantidad de gramos por medio de la cantidad de segmentos se sobrestimaría, ya que, para un mismo número de segmentos con mayor densidad de plásticos obtendríamos el mismo resultado que para una imagen con menor densidad. Así pues, vamos a realizar el mismo procedimiento para segmentos más pequeños.

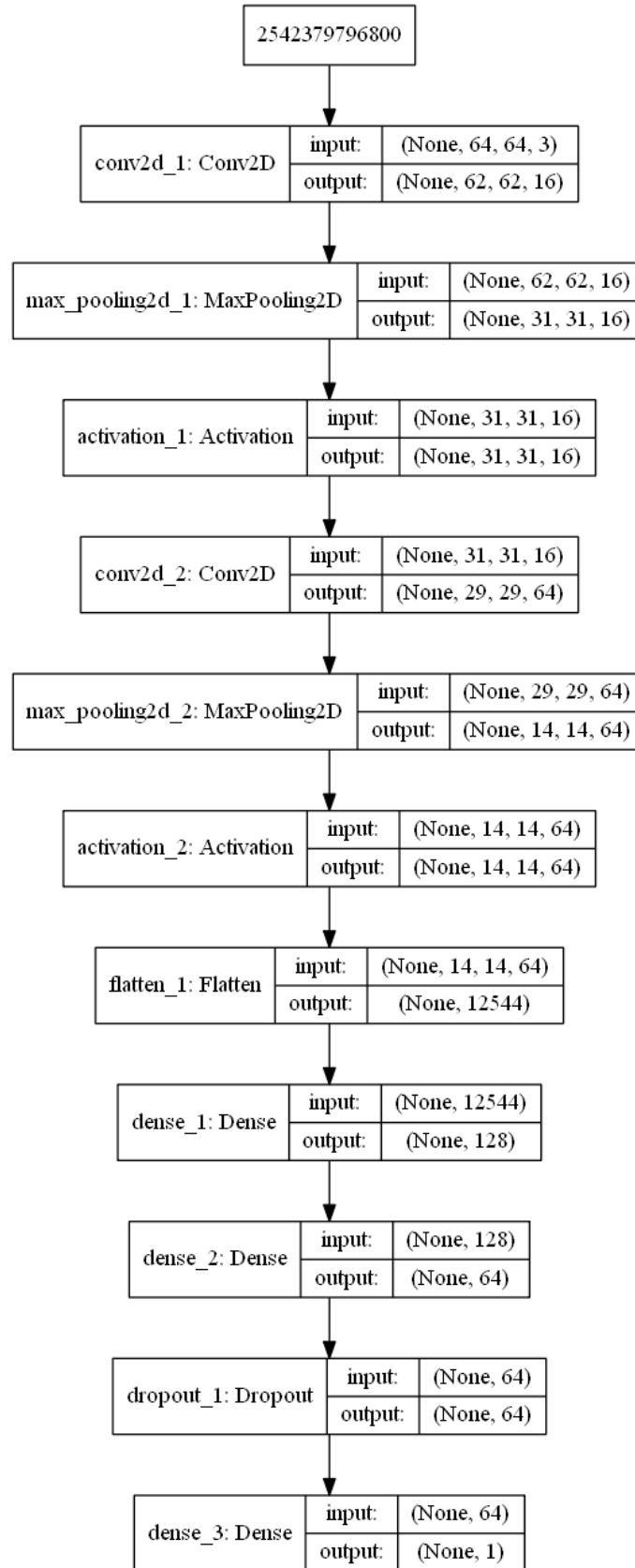


Figura 4.11: Estructura de la red neuronal usada



Figura 4.12: Fotografía perteneciente al conjunto de datos B de la playa Famara



Figura 4.13: Resultados de predicción para una fotografía perteneciente al conjunto de datos B de la playa Famara, 64 píxeles

4.2.2. Celdas de 16 píxeles

Al igual que en la segmentación de 64 por 64 píxeles, primero se mapearon las imágenes para obtener la máscara que nos indicará que segmento de la imagen es plástico. Una vez mapeadas visualizamos cuantos segmentos tienen de plástico por imagen.

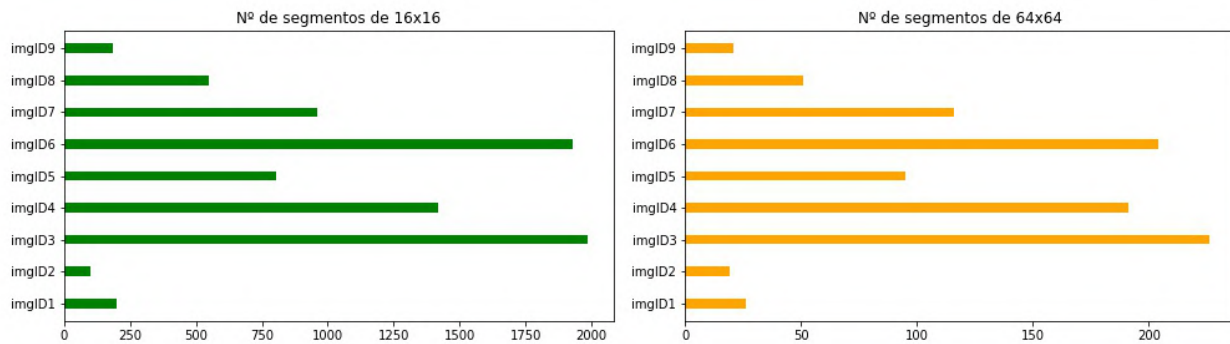


Figura 4.14: Segmentos con plástico por imagen. Para 16 píxeles y 64 píxeles.

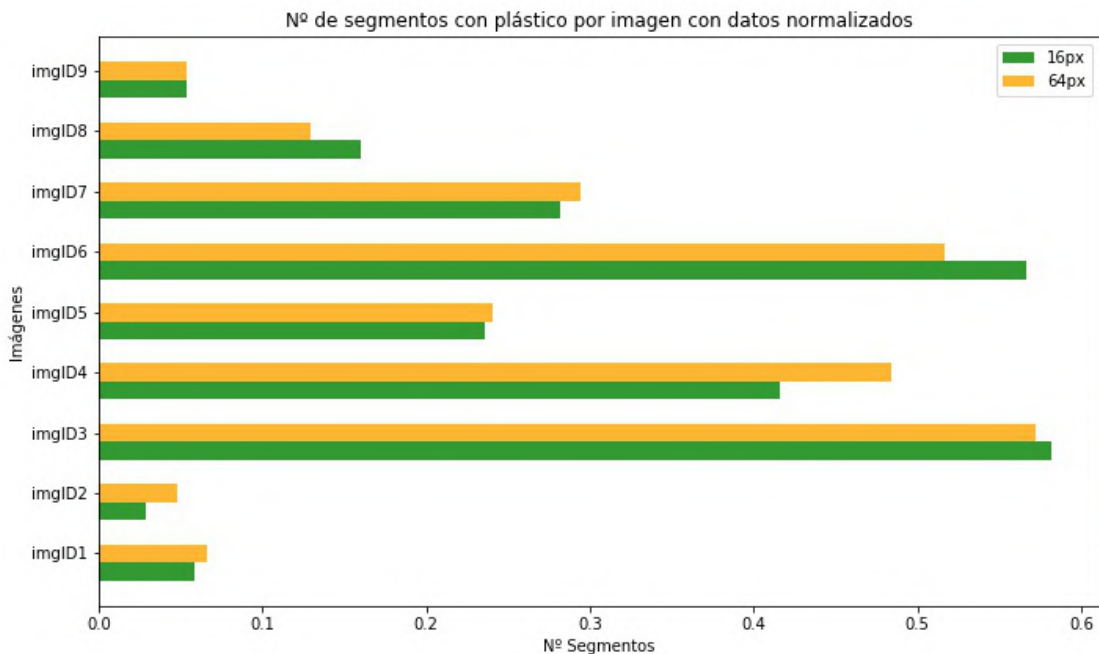


Figura 4.15: Cantidad de segmentos con plástico por imagen para 16 y 64 píxeles con valores normalizados

Como podemos observar en la gráfica de la Figura 4.14 la proporción de segmentos de 16 píxeles se conserva prácticamente igual para las distintas

imágenes respecto a la segmentación de 64 píxeles, la principal diferencia es que casi se octuplica el número de segmentos de forma global como se observa en el eje de coordenadas X de la figura. Para visualizarlo más claramente normalizamos ambos valores y los visualizamos en la misma gráfica Figura 4.15.

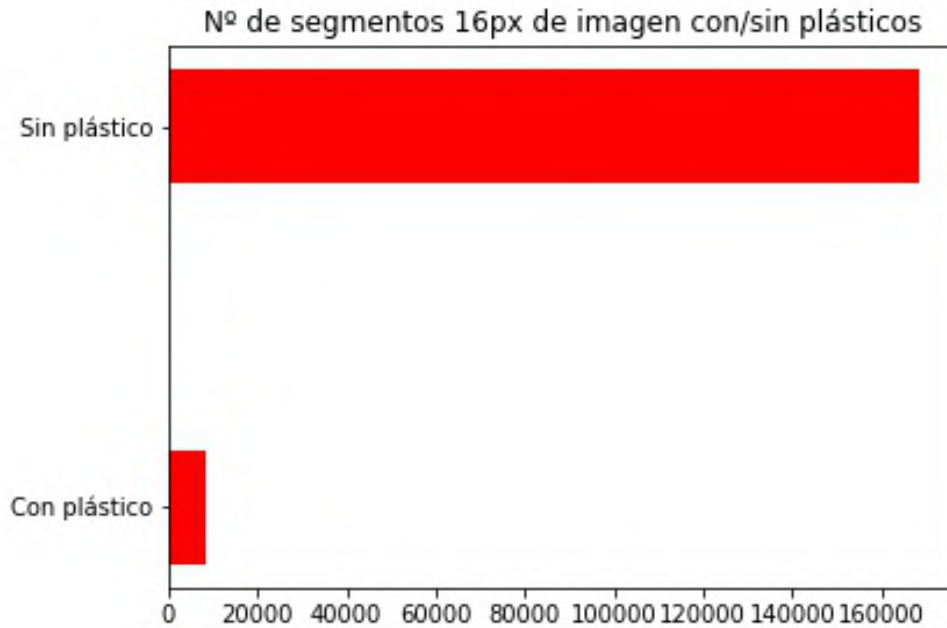


Figura 4.16: Total de segmentos de arena y plástico, 16 píxeles

En la Figura 4.15 podemos observar que en algunas imágenes tales como la 6 o la 8 la cantidad de segmentos parece que es proporcionalmente mayor para los segmentos de 16 píxeles en comparación que los de 64 píxeles, pero si observamos en la Figura 4.16 en comparación con la Figura 4.6, podemos ver que de forma global que la diferencia entre la cantidad de segmentos con plástico y arena ha crecido significativamente. Es decir, que al aumentar la precisión de forma global en todas las imágenes se ha reducido el porcentaje de plásticos total. Por lo que podemos concluir que en la Figura 4.15 se ha reducido en todas las imágenes la cantidad de segmentos de plástico de forma uniforme aunque parezca que en algunos casos han aumentado.

Este falso aumento en la Figura 4.16 para imágenes tales como la 6 o la 8 pareciendo que la cantidad proporcional de segmentos es mayor en la de 16 píxeles que 64 píxeles se debe a que la diferencia de la disminución es menor respecto a otras imágenes. El motivo es que debido a la densidad de plástico

de los segmentos de 64 píxeles ,ya que, en algunos casos no ocupaban ni una cuarta parte del segmento, mientras que en otras ocupaba su totalidad. Por lo que, para las imágenes con mayor densidad de plásticos por segmento la disminución de segmentos de 16 píxeles será proporcionalmente menor.

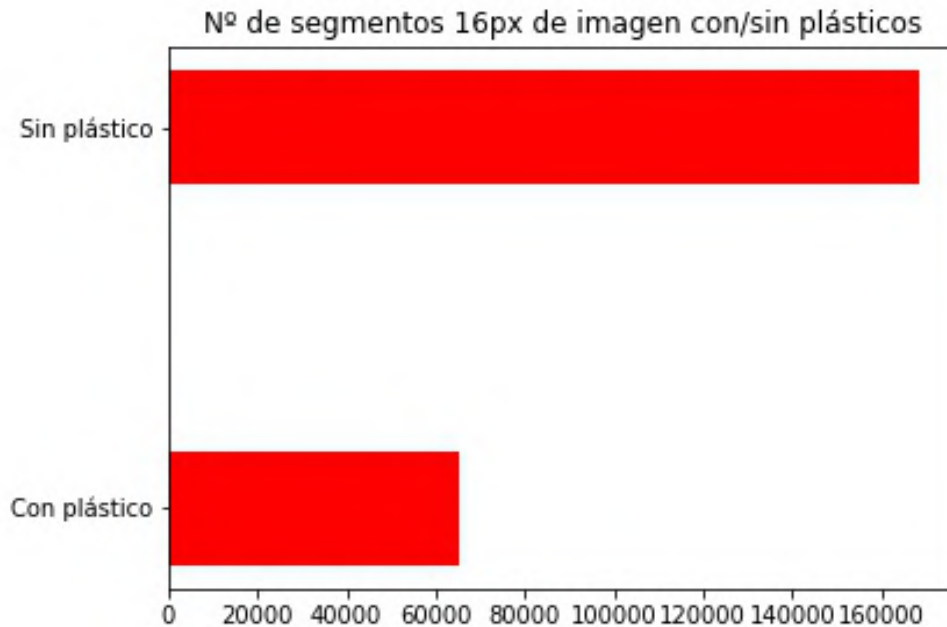


Figura 4.17: Total de segmentos que hay de arena y plástico con datos aumentados, 16 píxeles

Como la diferencias de clases es tan grande tenemos que hacer *data augmentation* con las mismas transformaciones que para 64 píxeles (Figura 4.6). Una vez aumentados los datos septuplicando los segmentos de plásticos hasta un total de 65000 podemos ver en la Figura 4.17 que la diferencia sigue siendo enorme, así que reduciremos el conjunto de arena para igualar las clases seleccionando de forma aleatoria los segmentos de arena.

Como arquitectura de red usaremos la misma estructura que en la primera prueba formada por 2 capas convolucionales con *maxpooling* y 3 capas densas, donde la última capa densa es una neurona con una función de activación sigmoide. Al igual que antes se usó como optimizador Adam.

De la misma manera durante el entrenamiento de la red después de cada época se comprobaba si la función de validación para el conjunto de validación había mejorado para guardar el modelo y quedarse con los mejores pesos para la red. Una vez finalizado el entrenamiento obtenemos la matriz de confusión para comprobar los resultados de la clasificación de la red para cada tipo de

los datos.

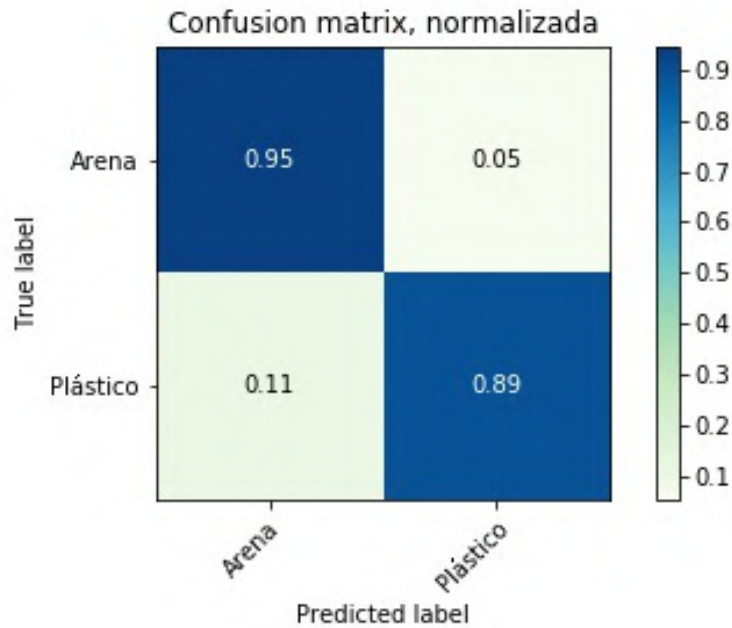


Figura 4.18: Matriz de confusión para segmentos de 16 píxeles

Como se puede comprobar en la Figura 4.2 de la sección anterior, para los segmentos de 64 píxeles los aciertos en la detección de plásticos eran de 0.94, y como se muestra en la Figura 4.18, para segmentos de 16 píxeles baja a un 0.89. Esto se debe a que como los segmentos son tan pequeños perdemos el factor entorno de la imagen donde los píxeles cercanos forman un papel importante a la hora de detectar las partículas. También hay que tener en cuenta que cuanto más segmentemos la imagen mayor son las posibilidades de que los microplásticos acaben en una intersección de la segmentación (ver Figura 4.19), por lo que, le resultará más difícil de identificar si es plástico.

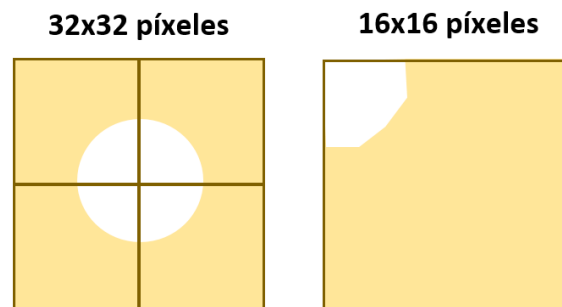


Figura 4.19: Microplástico fragmentado durante la segmentación

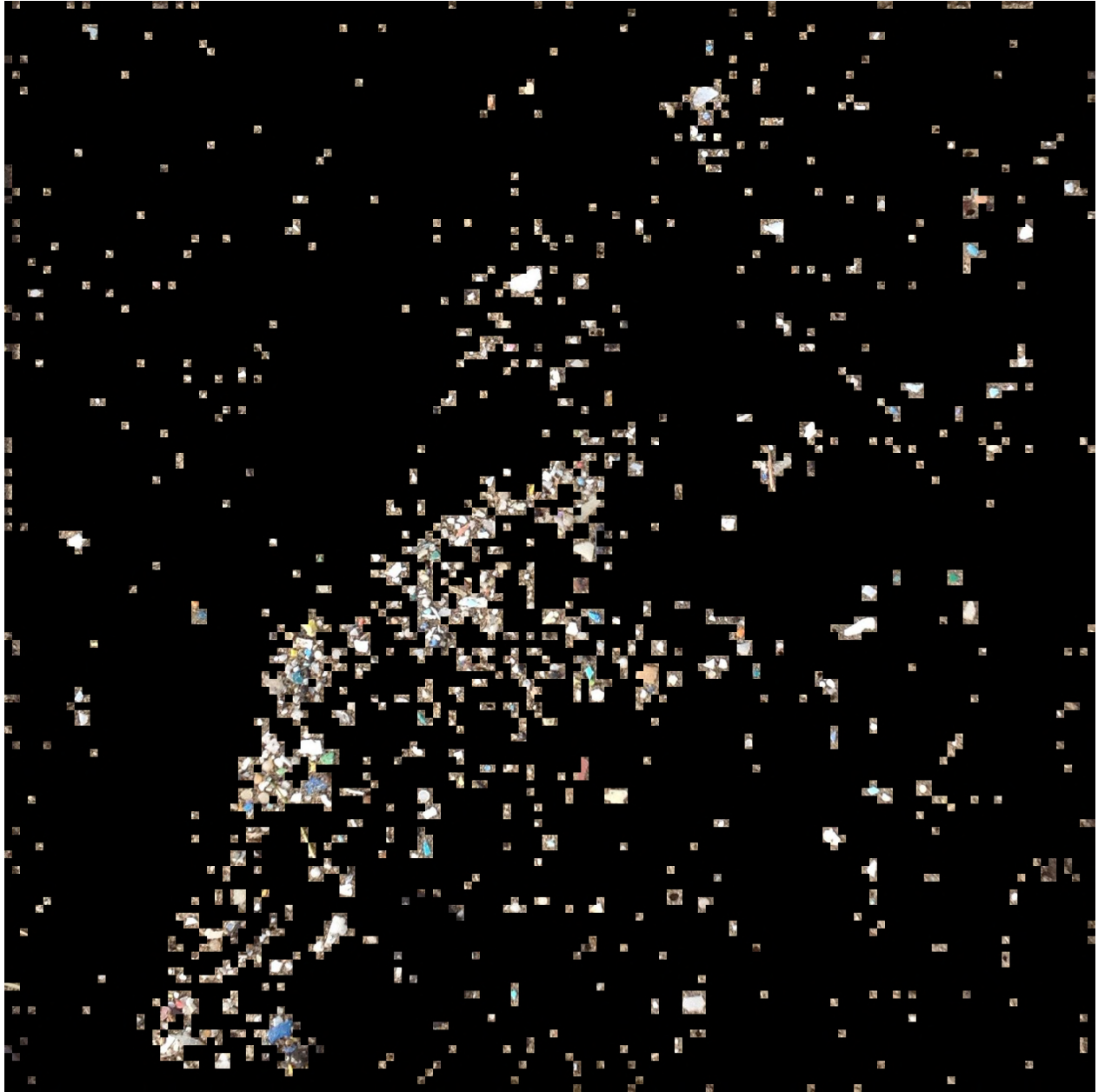


Figura 4.20: Resultados de predicción para una fotografía perteneciente al conjunto de datos B de la playa Famara, 16 píxeles

Podemos visualizar la predicción de la red para la misma imagen del conjunto de datos B (Figura 4.10, predicción para 64 píxeles en la Figura 4.12) que se mostró en la prueba de 64 píxeles, Figura 4.20.

Así pues, vamos a proceder a realizar el mismo procedimiento para segmentos de 32 píxeles, ya que, es un punto intermedio entre 64 píxeles donde teníamos mucha arena y entre 16 píxeles donde no se tienen en cuenta los píxeles más

cercanos.

4.2.3. Celdas de 32 píxeles

Para saber que segmentos son plástico se volvió a mapear las imágenes con la aplicación de Java usando los resultados mapeados para 64 píxeles por medio de la opción "Importar CSV" del programa.

Una vez obtenidos los CSV del mapeo de las imágenes, se realizó exactamente el mismo procedimiento de tratamiento de los datos y de datos aumentados. Para el entrenamiento se usó una proporción de 70 % de los datos balanceados como conjunto de entrenamiento y 30 % como conjunto de validación, con la misma arquitectura de 2 capas convolucionales con *maxpooling* y 3 capas densas.

Una vez entrenada la red guardando los pesos de los mejores resultados obtenemos la matriz de confusión para comparar los resultados.

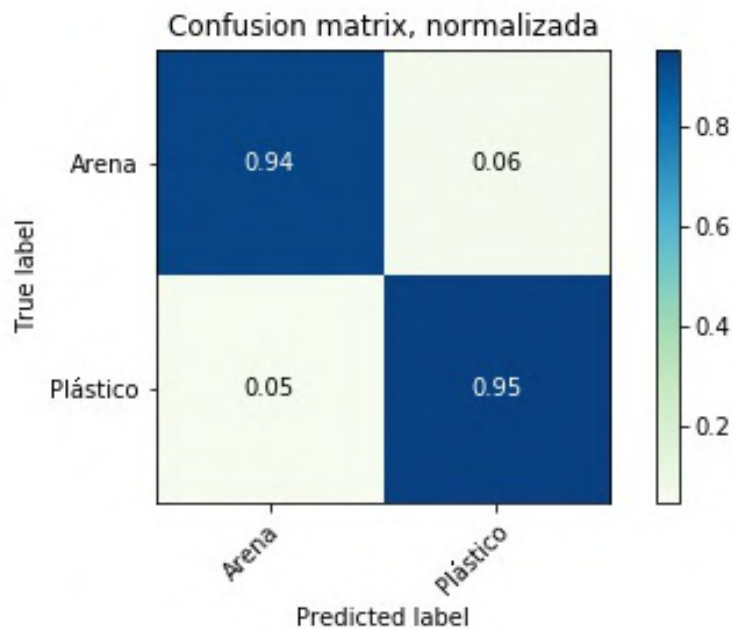


Figura 4.21: Matriz de confusión para segmentos de 32 píxeles

Como podemos observar en la Figura 4.21 el porcentaje de aciertos para plásticos del conjunto de test sube de 89 % a 95 % respecto a los resultados obtenidos para los segmentos de 16 píxeles (Figura 4.18). Esta subida se debe a que al tener segmentos con el doble de información la red es capaz de

detectar mejor los contornos y bordes. También hay que tener que cuenta que a mayor división de segmentos mayor es la probabilidad que de un fragmento de plástico caiga en una intersección de la segmentación de la imagen, lo que produce que a la red le resulte más complicado identificar que es ese objeto del cual solo se ven 3 píxeles.

En cuanto a la detección de arena baja un 1 % de 95 a 94, quedándose un porcentaje de acierto igual que para la segmentación de 64 píxeles (Figura 4.2). Respecto a la detección de plásticos para 64 píxeles sube un 1 %, es decir, para la segmentación de 32 píxeles nos encontramos frente a un 95 % y para 64 píxeles a un 94 %. La diferencia no es significativa, pero la máscara que obtenemos es más cercana a la realidad ya que tenemos el doble de precisión para la detección del contorno, aunque de ambas formas sea siendo cuadriculada.

Podemos ver la misma imagen del conjunto de datos B (Figura 4.10) predicha para la red entrenada con los segmentos de 32 píxeles, Figura 4.22. Seguido de los resultados para otra imagen del mismo conjunto de datos también de Famara, Figura 4.23 para la imagen original, Figura 4.24 para la imagen predicha.

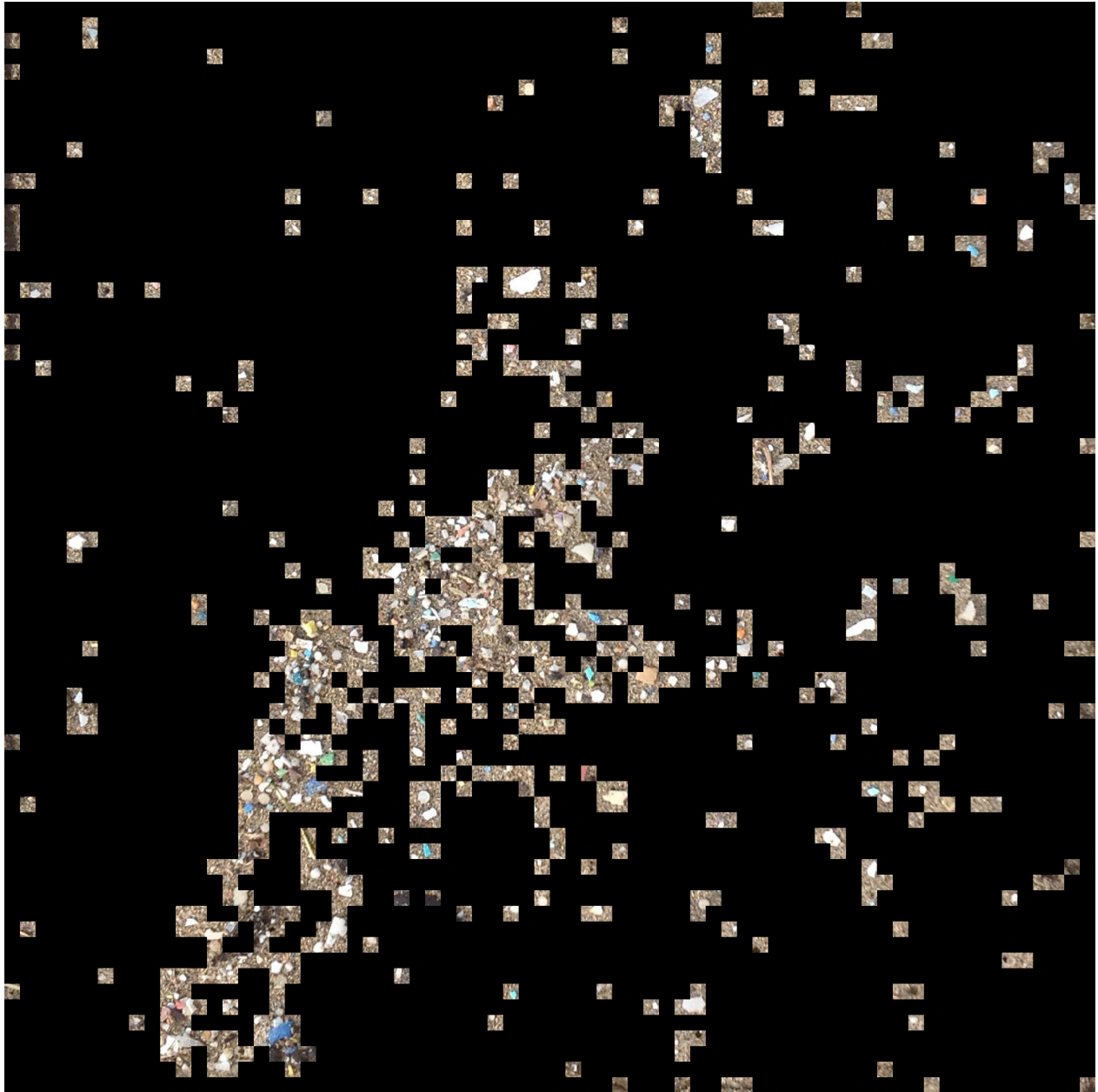


Figura 4.22: Resultados de predicción para una fotografía perteneciente al conjunto de datos B de la playa Famara, 32 píxeles



Figura 4.23: Fotografía 2 perteneciente al conjunto de datos B de la playa Famara

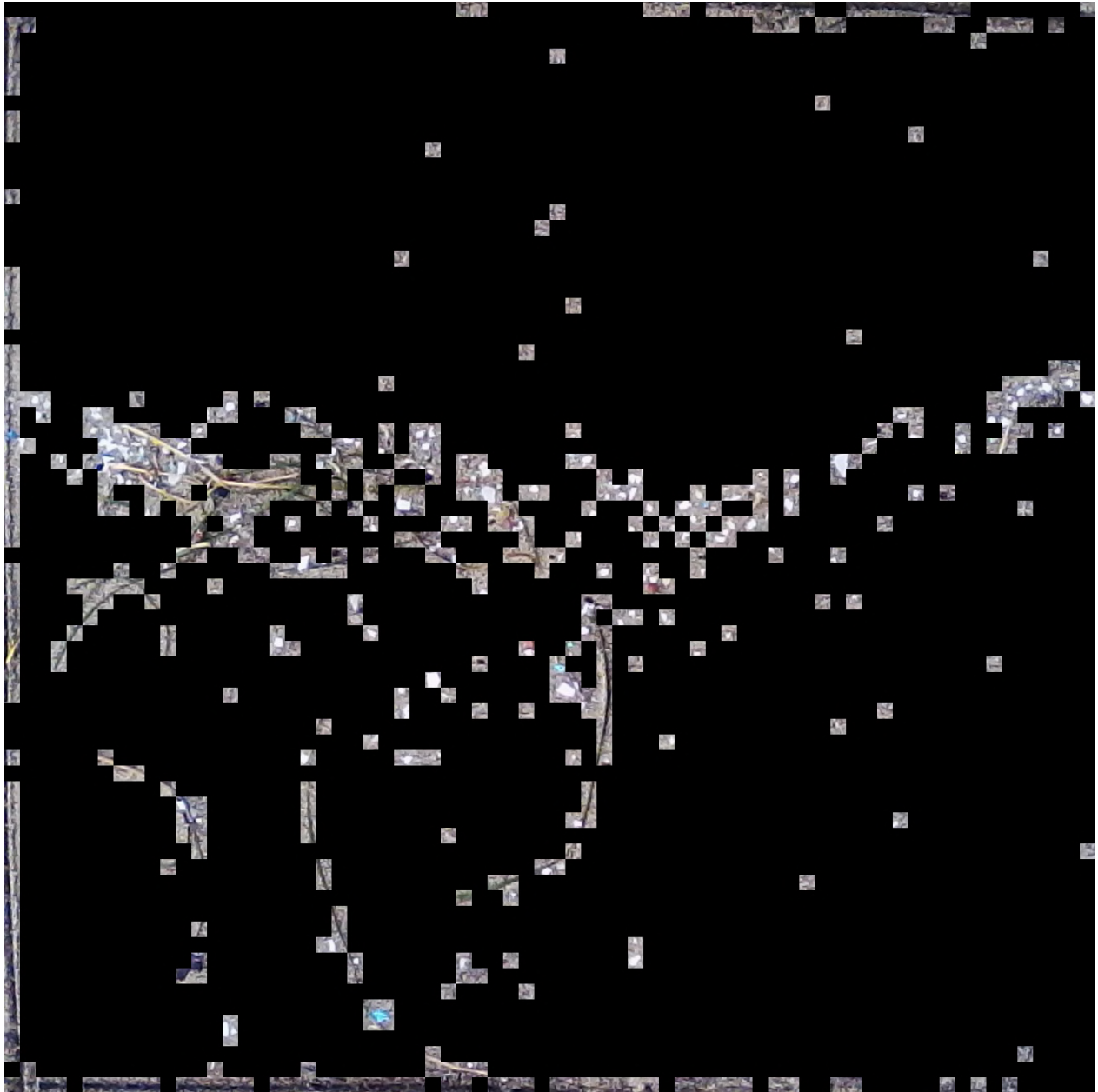


Figura 4.24: Resultados de predicción para otra fotografía perteneciente al conjunto de datos B de la playa Famara, 32 píxeles

Aparte de los resultados buenos también encontramos otras predicciones no tan favorables debido a la irregularidad del terreno, donde le resulta difícil identificar a la red los microplásticos (ver Figura 4.25).

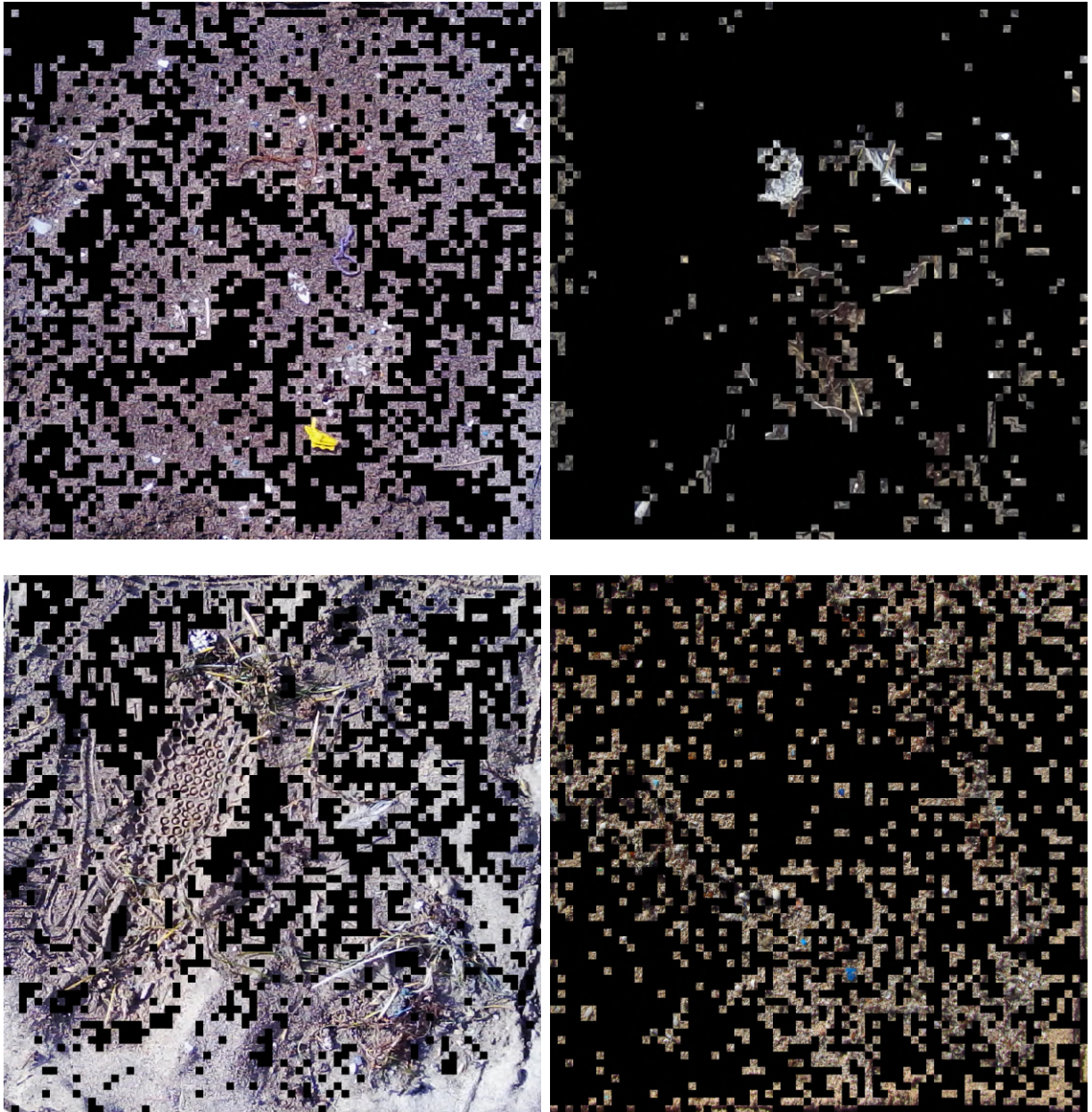


Figura 4.25: Predicciones no tan buenas para otras fotos del conjunto de datos B

4.3. Determinación de la arquitectura óptima

Para realizar las pruebas de la Sección 4.2 se usó únicamente una arquitectura pequeña con dos capas convolucionales (con *maxpooling* cada una) y tres capas densas. Como con una arquitectura pequeña hemos obtenido buenos resultados no se ha optado por probar con arquitecturas más grandes estilo VGG16 [10], ya que su tiempo de entrenamiento es elevado y nuestro conjunto de datos consta de segmentos pequeños que no tienen el nivel de detalle de los modelos pre-entrenados con los pesos del conjunto de datos de ImageNet [8]. Así pues, se procederá a variar el número de capas y de filtros y observar los resultados.

Todas estas pruebas se realizaron con un subconjunto del Conjunto de datos A y a un tamaño de 32 píxeles. Es decir, volvió a equilibrar las clases y separar entre conjunto de entrenamiento y validación, por lo que, no son exactamente los mismos datos de entrenamiento que en las pruebas de segmentos de la Sección 4.2.

Nombre de la prueba	Número de capas convolucionales	Número de MaxPooling	Número de capas densas	Aciertos para arena	Aciertos para plástico
Prueba A	2	2	3	0.94	0.96
Prueba B	2	2	4	0.93	0.96
Prueba C	3	3	3	0.93	0.97
Prueba D	1	1	2	0.95	0.90
Prueba E	3	3	2	0.95	0.96
Prueba F	2	2	2	0.96	0.94

Figura 4.26: Tabla con los resultados de las distintas pruebas

Como podemos observar en la Figura 4.26, la denominada Prueba A es la misma arquitectura que se usó en la Sección 4.2. A partir de ahí se procedió a ir variando el número de capas, a mayor número de capas densas aumentaba el tiempo de entrenamiento y bajaba el porcentaje de aciertos para la arena. Mientras que al aumentar el número de capas convolucionales mejora el porcentaje de aciertos para plástico, aunque si se intentaba aumentar el número de capas convolucionales a 6 el tiempo de aprendizaje se elevaba demasiado y con el conjunto de datos seleccionado ya se obtenían buenos resultados.

4.4. Entrenamiento con un mayor conjunto de datos

Para esta prueba se utilizaron tanto el conjunto de datos A como el B, en total unas 62 imágenes. Para ello se separaron manualmente 45 imágenes para el conjunto de entrenamiento y unas 17 para el conjunto de validación. Las imágenes de validación se seleccionaron de forma que hubiera mínimo una foto por tipo de arena distinto, para así ver los resultados de la red con distintas texturas. Todas las imágenes se mapearon con el programa auxiliar de Java (ver Sección 3.2), para ello se seleccionó un tamaño de segmento de 32 píxeles y un tamaño de imagen de 2048. Los CSV obtenidos con el mapeo sirven también para imágenes de 1024 por 1024 píxeles y segmentos de 16 píxeles, ya que se mantiene la misma proporción. Mapeando con un tamaño de imagen superior permite que sea más sencillo diferenciar a simple vista que es plástico.

Primero realizaremos la prueba para las imágenes de tamaño 1024 píxeles. Todas las imágenes se segmentaron en fragmentos de 16 por 16 píxeles, guardando cada segmento de forma ordenada, es decir, si una imagen pertenece al conjunto de entrenamiento los segmentos irán a la carpeta “Train”, y si además en ese segmento hay plástico irá a la carpeta “Plástico” que se encuentra dentro de “Train”, si es arena irá a la carpeta “Arena”, y el mismo caso para el conjunto de validación, cuyos segmentos irán a la carpeta “Validation”.

Una vez estructuradas las imágenes se aumentaron los datos de los segmentos del conjunto de entrenamiento, usando las mismas rotaciones y transformaciones que en la Sección 4.2, ver Figura 4.6. Al igual que en casos anteriores la diferencia entre las dos clases sigue siendo bastante grande, por lo que de forma aleatoria desplazaremos los segmentos sobrantes a otra carpeta, para que no interfieran en el entrenamiento de la red. También se desplazaron los segmentos sobrantes del conjunto de validación para que durante el entrenamiento la tasa de aciertos no se vea sesgada.

En cuanto a la arquitectura se utilizó la misma mostrada anteriormente (ver Figura 4.11). Una vez obtenidos los resultados visualizamos la matriz de confusión con los resultados predichos para el conjunto de validación. En la Figura 4.28 se puede ver que la tasa de aciertos para ambas clases se ha reducido drásticamente respecto a pruebas anteriores, esto se debe a que en el conjunto

de datos B encontramos mucha variedad de arena, donde se puede encontrar arena mezclada con algas, con conchas o con piedras pequeñas, de forma que a simple vista resulta difícil saber que es plástico y que no. Es decir, esta arquitectura se entrenó con un conjunto de datos mucho más complicado que las pruebas anteriores.

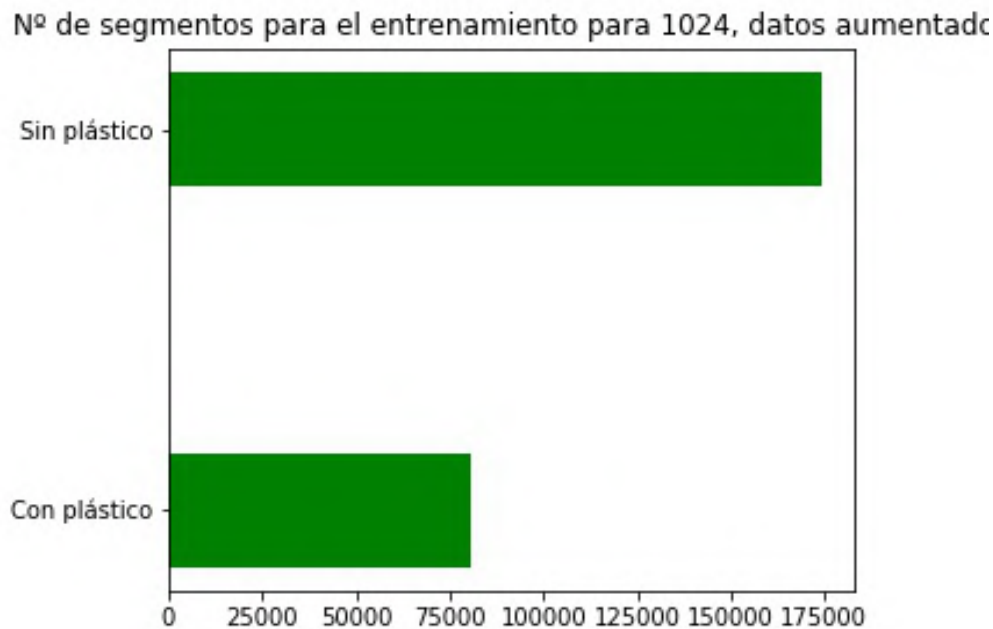


Figura 4.27: Conteo de la cantidad de segmentos por clase

Luego de obtener los resultados se realizó el mismo procedimiento para un tamaño de imagen de 2048 por 2048 píxeles y segmentos de 32 píxeles. Donde se obtuvieron los resultados de la Figura 4.29. En este caso la tasa de aciertos para arena ha aumentado considerablemente, y la de plástico ha disminuido un poco. Seguidamente observaremos en la Figura 4.30 y Figura 4.31, algunos resultados de predicciones para ambas arquitecturas, donde veremos una comparativa para las mismas imágenes.

Como podemos observar en las Figura 4.30 y 4.31, aquellos resultados predichos por la red entrenada con imágenes de tamaño 2048 y segmentos de 32 píxeles, dan mejores resultados respecto a los del entrenamiento de imágenes de 1024 y segmentos de 16 píxeles. Aunque esto no quiere decir que sean los resultados idóneos ya que en ha perdido precisión en la clasificación algunas imágenes respecto a pruebas anteriores (ver Figura 4.24), pero también ha ganado precisión en otras imágenes del conjunto de datos (ver Figura 4.25).

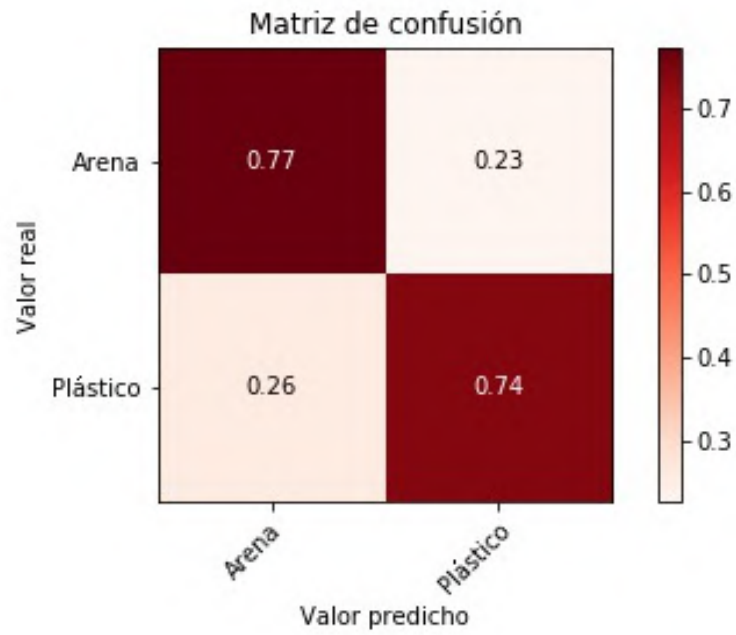


Figura 4.28: Matriz de confusión para imágenes de 1024

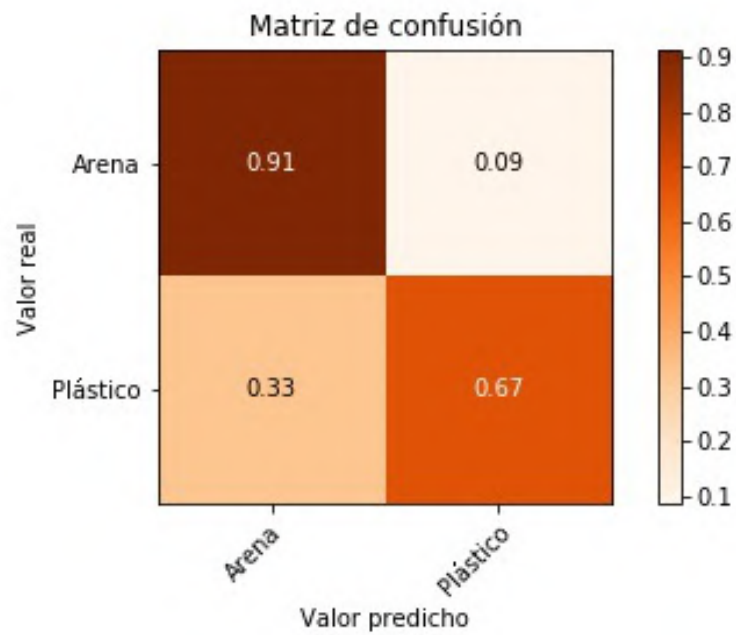


Figura 4.29: Matriz de confusión para imágenes de 2048

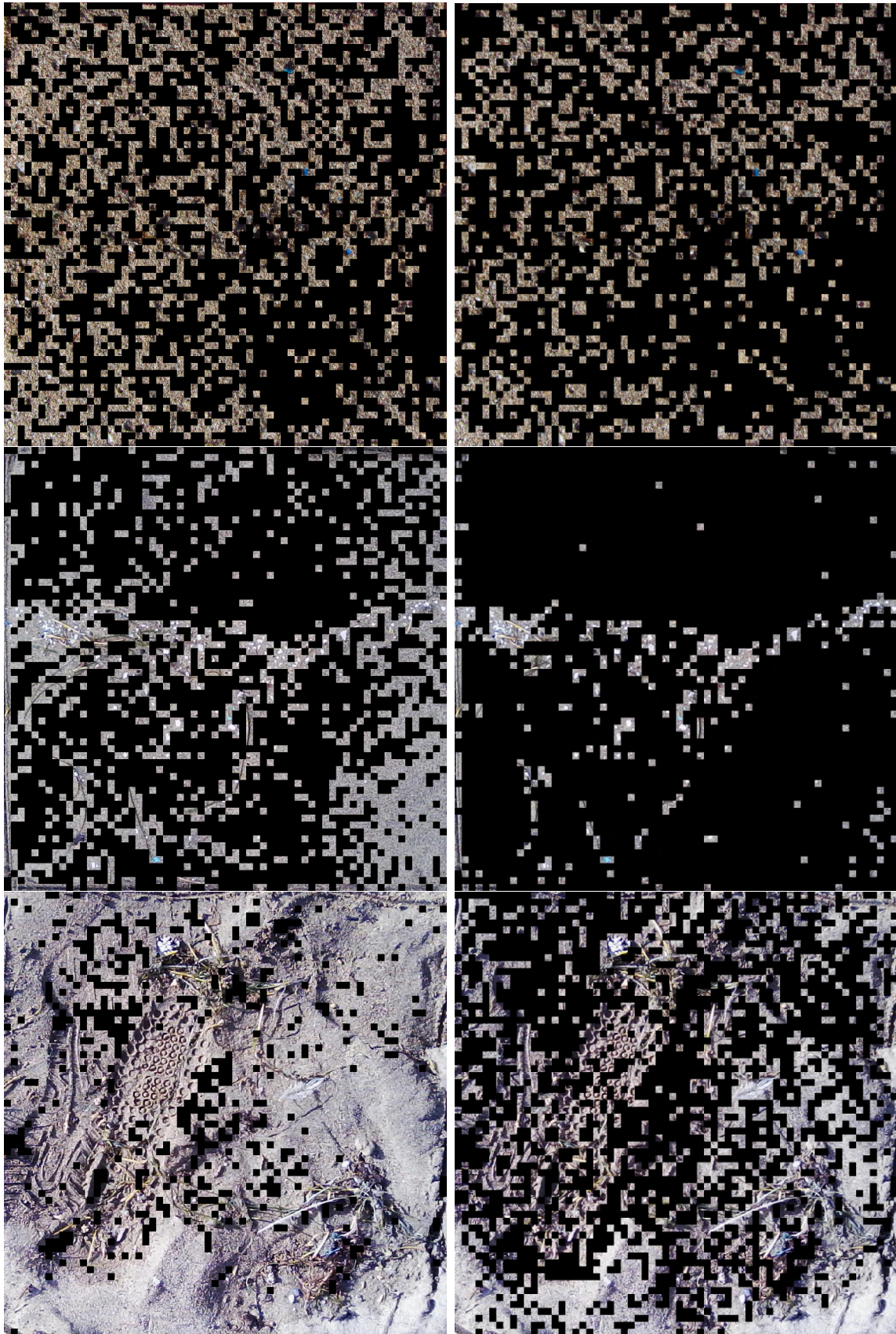


Figura 4.30: Resultados obtenidos para imágenes de 1024 y 2048 respectivamente

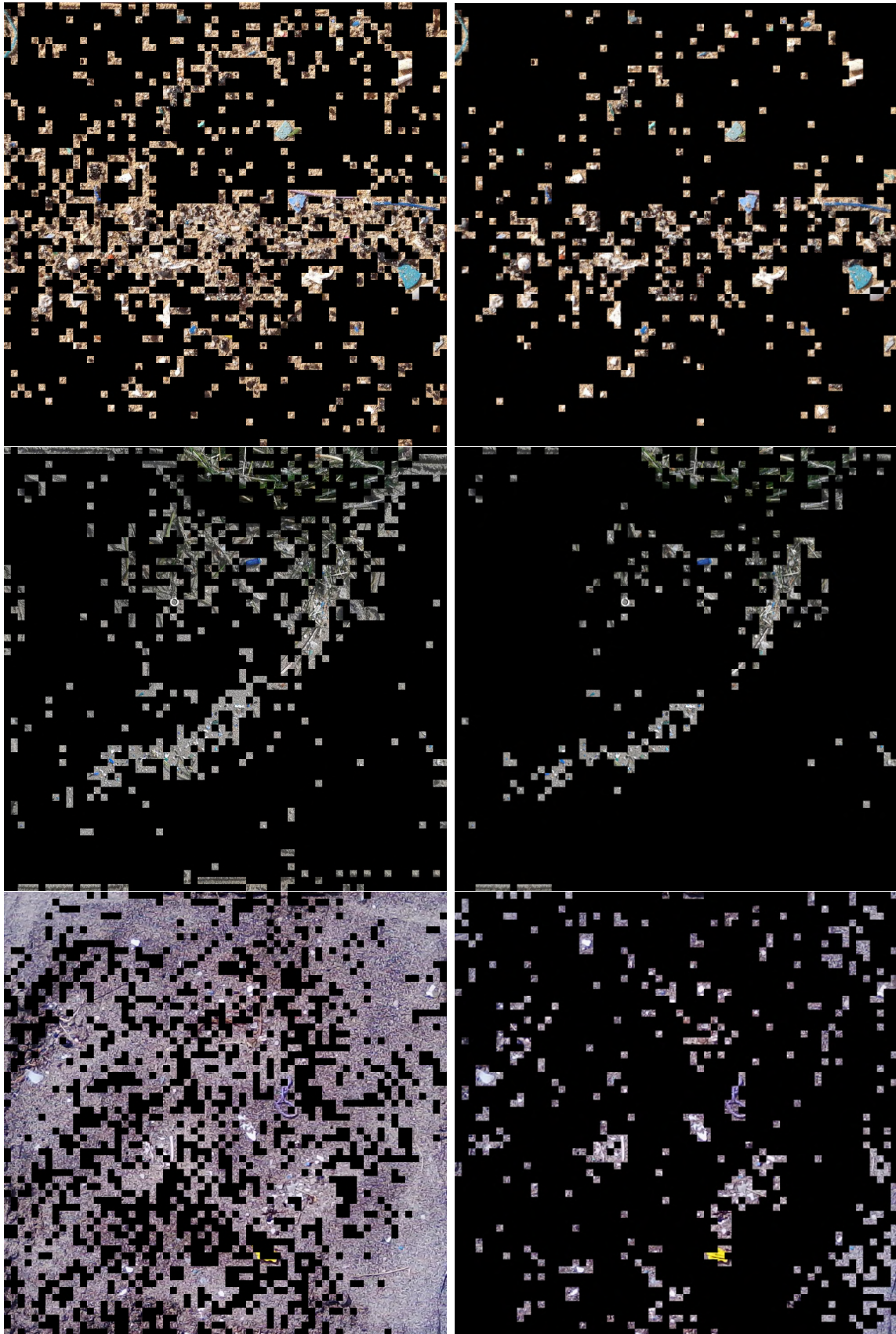


Figura 4.31: Otros resultados obtenidos para imágenes de 1024 y 2048 respectivamente

4.5. Estimación de la cantidad de gramos

Para poder realizar una estimación de la cantidad de gramos de plástico que hay por imagen se usaron los archivos excel de los conjuntos de datos A y B, donde se encuentran las mediciones realizadas en el laboratorio.

Como primera instancia vamos a ver la correlación entre los gramos del conjunto de datos A y el número de segmentos con plástico que se mapearon manualmente, Figura 4.32.

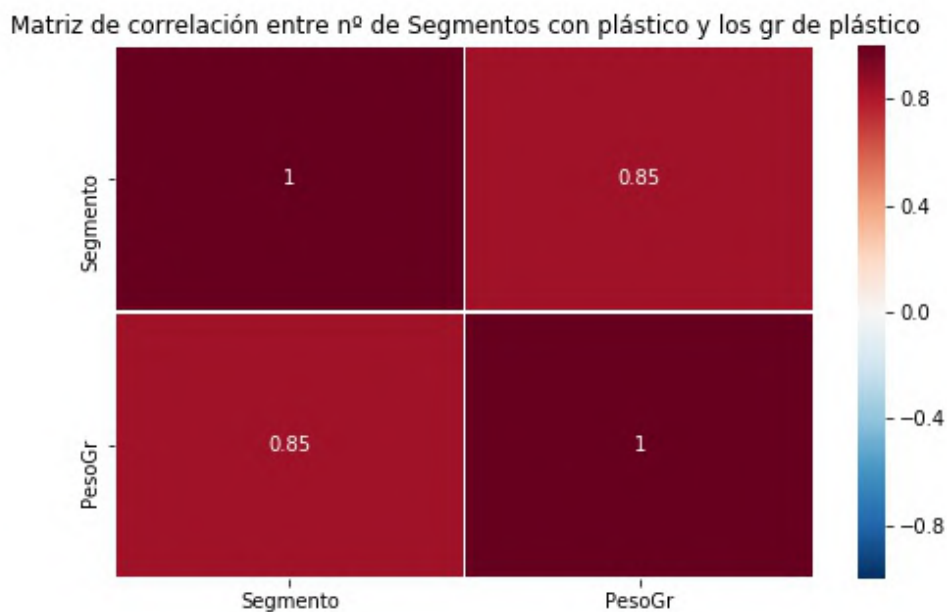


Figura 4.32: Matriz de correlación entre los gramos del conjunto de datos A y el número de segmentos con plástico

Como se puede ver en la Figura 4.32, la cantidad de gramos y el número de segmentos se encuentran muy correlacionados. Partiendo de esto podemos realizar un modelo de regresión lineal simple para una primera estimación. El modelo de regresión lineal simple tiene la siguiente expresión:

$$Y = \alpha + \beta X + \varepsilon$$

En donde α es la ordenada en el origen (el valor que toma Y cuando X vale

0), β es la pendiente de la recta (e indica cómo cambia Y al incrementar X en una unidad) y ϵ una variable a la que llamaremos error.

Una vez obtenido el modelo podemos ver el resultado de la regresión respecto a la aproximación de la realidad en la Figura 4.33.

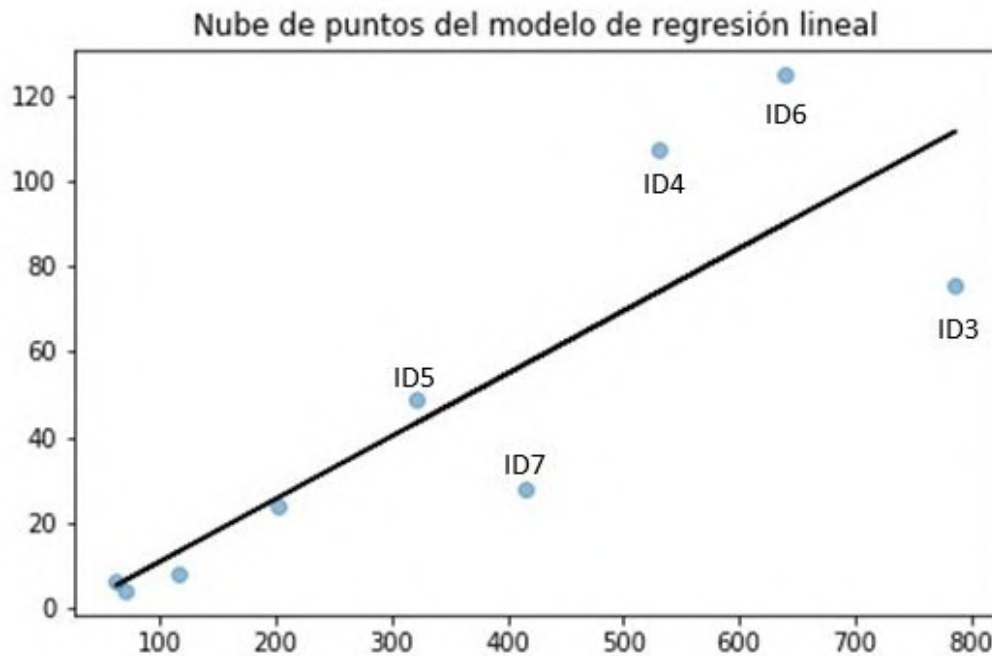


Figura 4.33: Regresión lineal para los datos del conjunto de datos A y el número de segmentos con plástico de 32 píxeles mapeados manualmente

En la Figura 4.33 podemos observar que hay puntos en donde el modelo se aproxima mucho a la realidad y otros en los que aleja bastante. Si tenemos en cuenta que los gramos de la cantidad de plástico que hay por imagen son: [8.403, 6.315, 75.466, 107.536, 48.773, 124.632, 27.758, 24.015, 4.417], empezando por la imagen 1. Junto con el número de plásticos por imagen en segmentos de 32x32, empezando por la imagen 1 son: [116, 62, 787, 530, 321, 640, 416, 202, 70]. Podemos razonar los resultados del modelo de la regresión lineal.

Si observamos en la Figura 4.34, podemos ver que las imágenes que más segmentos con plástico tienen son la 3 y la 6, con 787 y 640 respectivamente, pero mirando los gramos la imagen 3 tiene 75.466, mientras que la 6 tiene 124.632, es decir, entre las dos la imagen con menos segmentos tiene casi el doble de gramos de plástico que la otra. Por lo que, el modelo de regresión

lineal sobrestimaré los gramos de la imagen 3 y subestimaré los de la imagen 6, como se ve en la Figura 4.33. Eso se debe que para un mismo número de segmentos de plástico la densidad de partículas puede ser distinta.

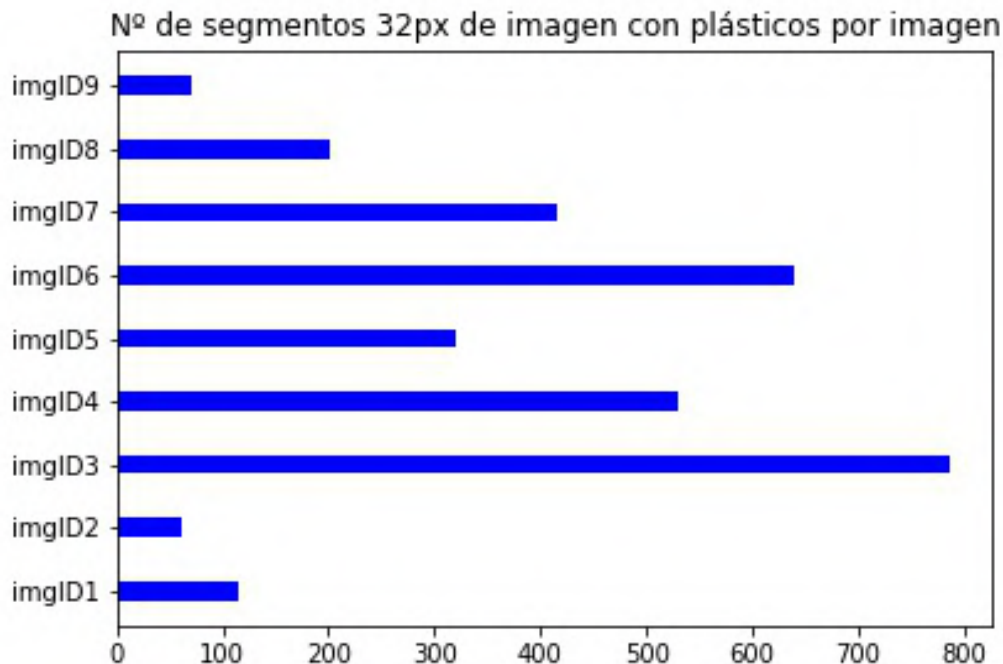


Figura 4.34: Número de segmentos que contienen microplásticos por imagen, para un tamaño de segmento de 32 píxeles

Ahora bien vamos a predecir el número de segmentos para todas las imágenes del conjunto de datos B usando la red neuronal entrenada con segmentos de 32 píxeles. Una vez obtenida la estimación de segmentos vamos a estimar los gramos con el modelo de regresión lineal.

Tanto en la tabla de la Figura 4.39 y la Figura 4.40 podemos ver que la estimación se encuentra muy sobrestimada. En primera instancia sobrestimada por el número de segmentos y luego sobrestimada por el modelo de regresión lineal.

Si visualizamos algunas de las imágenes donde la predicción del número de segmentos con plástico está excesivamente sobrestimado, podemos observar que muchos de los fallos provienen de texturas de arena distintas a las que se encuentran en el conjunto de entrenamiento. Tales como la imagen con su máscara predicha en la Figura 4.35. Donde la arena tiene una textura grumosa con algas enterradas (ver Figura 4.36), creando contornos y sombras

dificultando la detección de plástico y dando falsos positivos donde solo hay algas. Por lo tanto la red dará como resultado muchos más segmentos con microplásticos que los que hay en realidad.

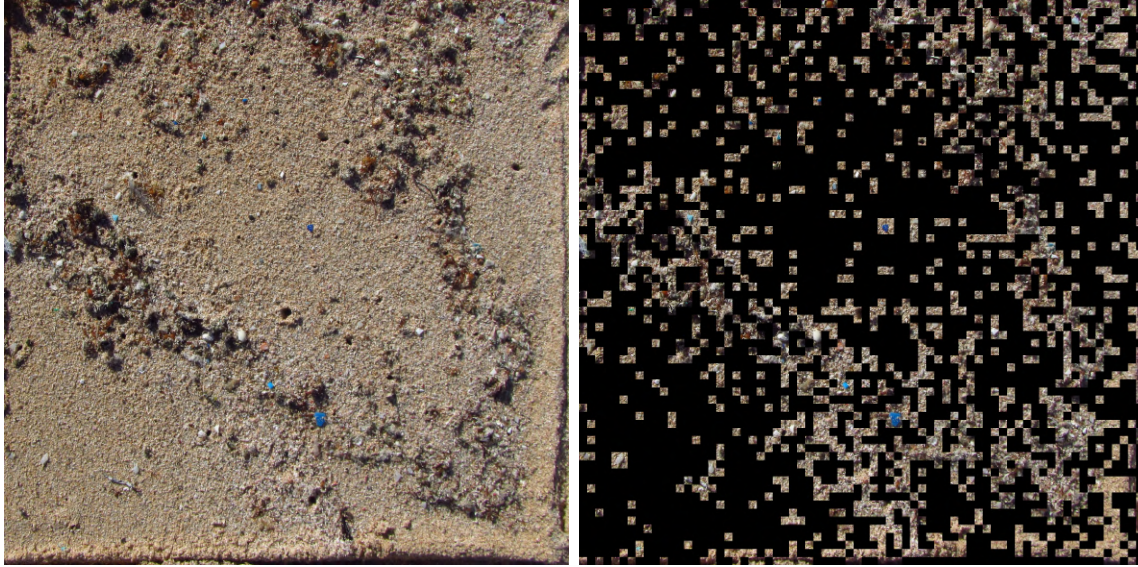


Figura 4.35: Imagen predicha del conjunto de datos B



Figura 4.36: Textura de arena de la Figura 4.35

Otro ejemplo podría ser la imagen con su máscara en la Figura 4.37. Donde la arena es mucho más blanca que las del conjunto de entrenamiento, además de contener muchas irregularidades en la superficie, ya sea por algas o por marcas en la arena.

También hay que tener en cuenta que las fotos del conjunto de datos A tienen una mayor calidad de imagen que las del conjunto de datos B, lo que produce que al redimensionar las imágenes más pequeñas se produzca ruido en la imagen, lo que luego se traduce en ruido durante la predicción de la red.

Sabiendo esto si hacemos una matriz de correlación entre los gramos de plástico reales y los gramos de plástico estimados por el modelo (ver Figura 4.38),

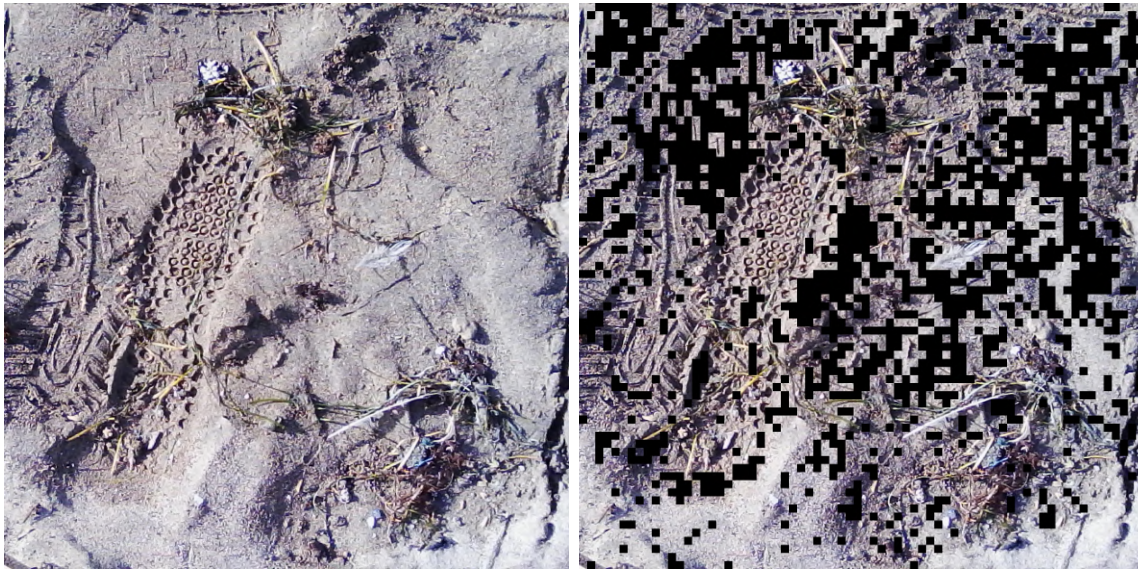


Figura 4.37: Imagen predicha del conjunto de datos B

podemos ver que la relación entre ambos datos es casi nula, por lo que el modelo de regresión lineal realizado para el conjunto de datos A, no es aplicable al conjunto de datos B y mucho menos aplicable con las predicciones sobrestimadas de la red neuronal.

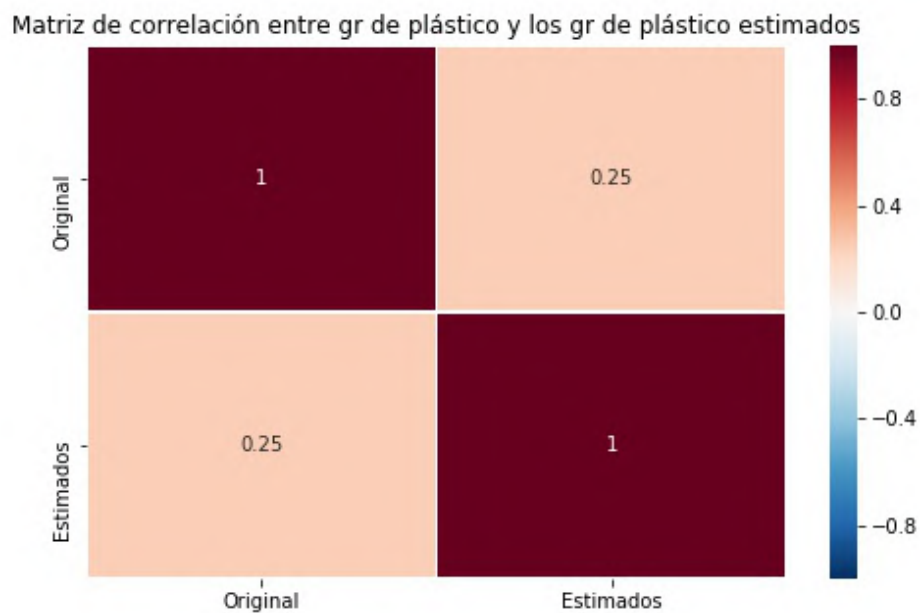


Figura 4.38: Matriz de correlación entre los gramos del excel y los estimados

	Imagen Nombre	Original gr	Predecido regr	Precedido red
0	20151115_Lambra_a.jpg	15.407	613.617	4217
1	20151115_Lambra_b.jpg	13.863	253.822	1759
2	20151115_Lambra_c.jpg	9.13	561.214	3859
3	20151125_Famara_a.jpg	16.091	48.748	358
4	20151125_Famara_b.jpg	17.366	38.794	290
5	20151125_Famara_c.jpg	32.404	91.051	647
6	20151127_Lambra_a.jpg	137.404	431.085	2970
7	20151127_Lambra_b.jpg	91.25	385.562	2659
8	20151127_Lambra_c.jpg	135.09	264.215	1830
9	20151211_Lambra_a.jpg	67.256	368.289	2541
10	20151211_Lambra_b.jpg	79.098	391.27	2698
11	20151211_Lambra_c.jpg	104.5	189.856	1322
12	20151212_Famara_a.jpg	36.704	185.025	1289
13	20151212_Famara_b.jpg	57.353	307.396	2125
14	20151212_Famara_c.jpg	22.066	124.718	877
15	20151221_Famara_a.jpg	20.388	122.083	859
16	20151221_Famara_b.jpg	17.291	67.045	483
17	20151221_Famara_c.jpg	24.938	104.957	742
18	20151222_Lambra_a.jpg	35.803	481.731	3316
19	20151222_Lambra_b.jpg	69.984	392.002	2703
20	20151222_Lambra_c.jpg	47.054	454.505	3130
21	20160118_Famara_a.jpg	7.171	184.44	1285
22	20160118_Famara_b.jpg	13.148	149.895	1049
23	20160118_Famara_c.jpg	16.794	117.984	831
24	20160129_Famara_a.jpg	32.84	27.523	213
25	20160129_Famara_b.jpg	8.624	38.941	291
26	20160129_Famara_c.jpg	22.151	64.996	469
27	20160201_Lambra_a.jpg	31.766	75.681	542
28	20160201_Lambra_b.jpg	32.258	72.315	519
29	20160201_Lambra_c.jpg	71.924	78.902	564
30	20160308_Famara_a.jpg	7.533	80.658	576
31	20160308_Famara_b.jpg	11.12	170.095	1187
32	20160308_Famara_c.jpg	10.22	343.698	2373
33	20160324_Famara_a.jpg	5.065	390.099	2690
34	20160324_Famara_b.jpg	14.268	378.828	2613
35	20160326_Lambra_a.jpg	37.437	101.737	720
36	20160326_Lambra_b.jpg	23.416	21.083	169
37	20160326_Lambra_c.jpg	24.593	29.719	228
38	20160411_Lambra_a.jpg	56.527	167.899	1172
39	20160411_Lambra_b.jpg	82.287	179.024	1248
40	20160411_Lambra_c.jpg	86.215	147.992	1036
41	20160422_Lambra_a.jpg	1.15	236.843	1643
42	20160422_Lambra_b.jpg	3.076	219.131	1522
43	20160422_Lambra_c.jpg	10.533	218.545	1518
44	20160603_Famara_a.jpg	5.213	71.144	511
45	20160603_Famara_b.jpg	7.982	39.087	292
46	20160603_Famara_c.jpg	11.744	117.838	830
47	20160722_Famara_a.jpg	1.469	86.221	614
48	20160722_Famara_b.jpg	1.665	89.002	633
49	20160722_Famara_c.jpg	0.705	112.568	794
50	20160820_Famara_a.jpg	4.021	505.591	3479
51	20160820_Famara_b.jpg	2.921	70.265	505

Figura 4.39: Tabla con los gramos originales, número de segmentos predichos y gramos estimados para todas las imágenes del conjunto de datos B

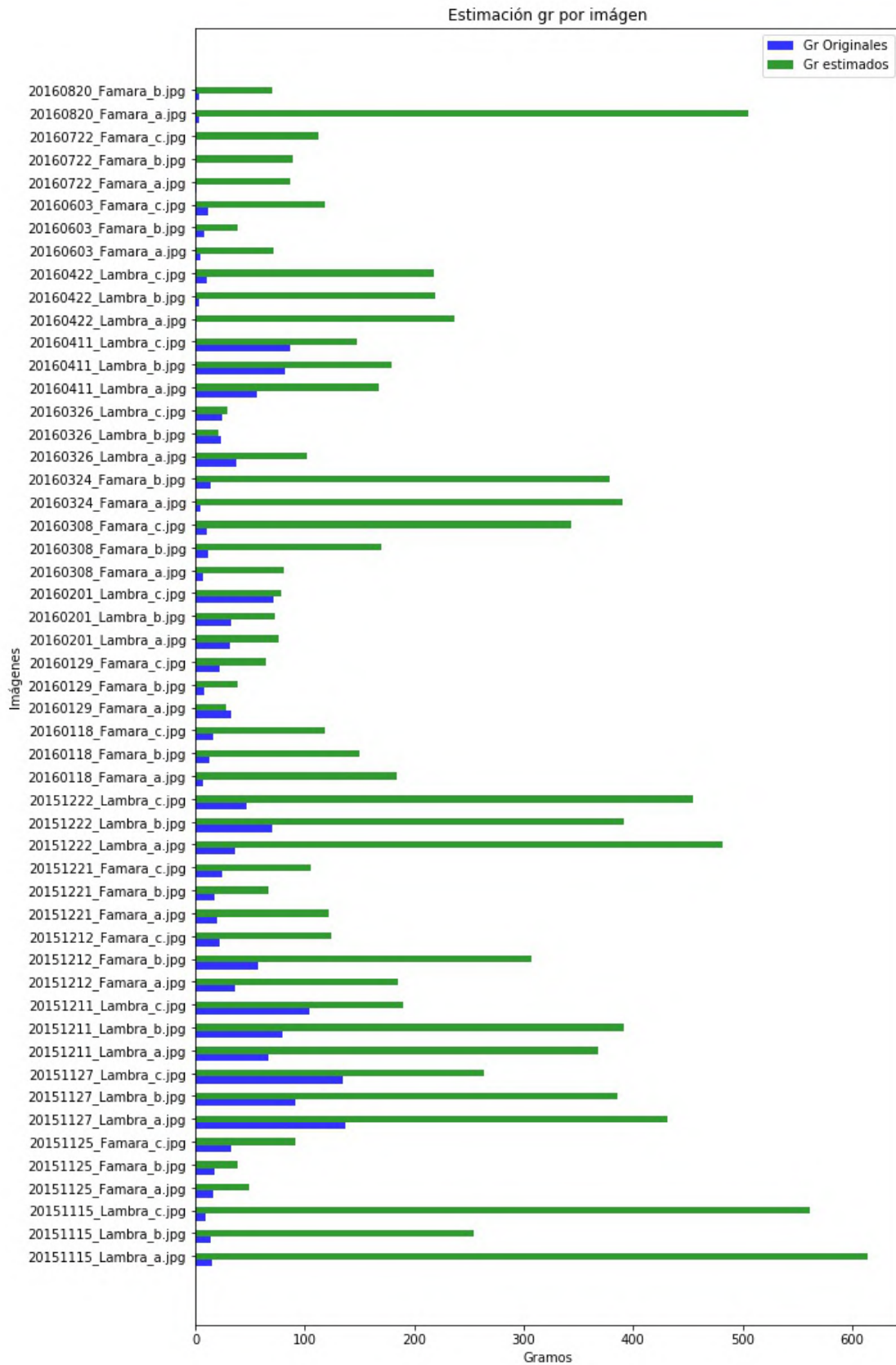


Figura 4.40: Estimación de los gramos frente a los gramos originales para las imágenes del conjunto de datos B

Capítulo 5

Conclusiones

Todo este proyecto ha supuesto un refuerzo de los conocimientos básicos que tenía y un aprendizaje de todo lo que conlleva trabajar con inteligencia artificial, específicamente redes neuronales. La preparación del conjunto de datos me parece una de las partes más importantes para que la red neuronal produzca buenos resultados. Si los datos no están balanceados o no tenemos un conjunto representativo del problema como datos de entrenamiento, obtendremos resultados sesgados o no representativos de la realidad.

Este trabajo de fin de grado ha abordado el problema de la detección y estimación de plástico, con la segmentación de las imágenes y el uso de aprendizaje profundo con un clasificador binario, para luego proceder con una regresión lineal para la estimación. Como se comprobó en los resultados del desarrollo del proyecto si conseguimos un mejor modelo de predicción, podremos obtener un mejor modelo de estimación de los gramos, ya que, si detecta muchos segmentos como plástico cuando solo son arena se sobreestimarán los resultados.

Si bien es cierto que hay muchos factores que afectan a la detección de plásticos pequeños usando únicamente imágenes, ya que, durante el mapeo manual resultaba difícil saber que era plástico y que no. En las fotos podemos encontrar fragmentos de conchas de tonos blancos que a simple vista podría pasar por plástico, pequeñas rocas, redes de pesca rotas enredadas con algas, colillas de tabaco y todo lo mencionado junto y enterrado en la arena, por lo que para automatizar de forma más efectiva el proceso de detección habría que tener un mapeo donde se supiera que es cada cosa con exactitud.

También hay que tener en cuenta que la forma de plantear la solución tiene limitaciones, ya que dependiendo de la segmentación caen microplásticos en los bordes de los segmentos, por lo que, a la red le resultará más difícil identificar que es esa zona de píxeles cortada.

Para obtener resultados más precisos usando el procedimiento de la segmentación de las imágenes se podría volver a fragmentar los segmentos de plástico para obtener una máscara más precisa. Es decir, pongamos que trabajamos con segmentos de 32 píxeles, una vez obtenido los resultados predichos se procedería a fragmentar los segmentos en, por ejemplo, 16 píxeles teniendo así una mayor precisión, pero sin sacrificar tanto la pérdida de información. Además, se podría añadir un tercer tipo a la hora de clasificar teniendo así: arena, plástico y otros, donde otros sería desde algas, piedras o todo aquello que no quede claro que es. El problema también se puede abordar usando arquitecturas más complejas de redes neuronales que realicen su clasificación píxel a píxel, tales como U-Net [15], YOLACT [3] o Mask R-CNN [11].

En este trabajo se ha cubierto la competencia propia del trajo de fin de grado TFG01, y la competencia específica CP04 de la mención de Computación del Grado de Ingeniería Informática. Está ha sido cubierta con la realización de tareas de investigación y estudio de redes neuronales.

TFG01: Ejercicio original a realizar individualmente y presentar y defender ante un tribunal universitario, consistente en un proyecto en el ámbito de las tecnologías específicas de la Ingeniería en Informática de naturaleza profesional en el que se sinteticen e integren las competencias adquiridas en las enseñanzas.

CP04: Capacidad para evaluar la complejidad computacional de un problema, conocer estrategias algorítmicas que puedan conducir a su resolución y recomendar, desarrollar e implementar aquella que garantice el mejor rendimiento de acuerdo con los requisitos establecidos.

Bibliografía

- [1] Javier Lorenzo Navarro, Modesto Castrillón Santana, Enrico Santesarti, Maria De Marsico, Ico Martínez, Eugenio Raymond, May Gómez y Alicia Herrera (2020). SMACC: A System for Microplastics Automatic Counting and Classification. [urlhttps://ieeexplore.ieee.org/document/8976153?source=authoralertpubid=ra-577dc9f623862778ct=1title=Sharepco=tbxnj-1.0](https://ieeexplore.ieee.org/document/8976153?source=authoralertpubid=ra-577dc9f623862778ct=1title=Sharepco=tbxnj-1.0). Accedido 12-03-2020.
- [2] Carlos Santana Vega (2018). Redes neuronales: La Neurona. Accedido 13-05-2019.
- [3] Daniel Bolya, Chong Zhou, Fanyi Xiao, Yong Jae Lee (2019). U-Net: Convolutional Networks for Biomedical Image Segmentation. [urlhttps://www.immersivelimit.com/tutorials/yolact-with-google-colab](https://www.immersivelimit.com/tutorials/yolact-with-google-colab). Accedido 10-02-2020.
- [4] Desconocido (2005). UNIDAD 4 REDES NEURONALES. [urlhttps://sites.google.com/site/mayinteligenciartificial/unidad-4-redes-neuronales](https://sites.google.com/site/mayinteligenciartificial/unidad-4-redes-neuronales). Accedido 28-02-2020.
- [5] Diego Calvo (2017). Red neuronal convolucional cnn. [urlhttp://www.diegocalvo.es/red-neuronal-convolucional/](http://www.diegocalvo.es/red-neuronal-convolucional/). Accedido 28-02-2020.
- [6] Fundéu BBVA (2018). Microplástico, palabra del año 2018 para la fundéu bbva. [urlhttps://www.fundeu.es/recomendacion/microplastico-palabra-del-ano-2018/](https://www.fundeu.es/recomendacion/microplastico-palabra-del-ano-2018/). Accedido 17-02-2020.
- [7] Google Colaboratory (2019). Te damos la bienvenida a colab. [urlhttps://colab.research.google.com/notebooks/welcome.ipynb?authuser=1hl=es](https://colab.research.google.com/notebooks/welcome.ipynb?authuser=1hl=es). Accedido 21-10-2019.
- [8] ImageNet (2010). ImageNet. [urlhttp://www.image-net.org/](http://www.image-net.org/). Accedido 20-03-2020.
- [9] Juan Ignacio Bagnato (2016). Clasificación con datos desbalanceados. [urlhttps://www.aprendemachinelearning.com/clasificacion-con-datos-desbalanceados/](https://www.aprendemachinelearning.com/clasificacion-con-datos-desbalanceados/). Accedido 17-02-2020.
- [10] K. Simonyan and A. Zisserman (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. [urlhttps://arxiv.org/abs/1409.1556](https://arxiv.org/abs/1409.1556). Accedido 20-03-2020.

- [11] Kaiming He, Georgia Gkioxari, Piotr Dollar, Ross Girshick (2018). Mask R-CNN. [urlhttps://arxiv.org/pdf/1703.06870.pdf](https://arxiv.org/pdf/1703.06870.pdf). Accedido 19-03-2020.
- [12] Martínez I., Herrera A., Lorenzo-Navarro J., Castrillón Santana M., Packard T.T., Gómez M. (2019). Optimization of large microplastic (1-5mm) sampling methodologies. [urlhttp://eomar.ulpgc.es/papers](http://eomar.ulpgc.es/papers). Accedido 18-03-2020.
- [13] Natasha Daily, National Geographic (2019). ¿Cómo afectan los residuos plásticos a los animales? [urlhttps://www.nationalgeographic.com.es/naturaleza/grandes-reportajes/como-afectan-residuos-plasticos-a-animales_12738/1](https://www.nationalgeographic.com.es/naturaleza/grandes-reportajes/como-afectan-residuos-plasticos-a-animales_12738/1). Accedido 06 – 03 – 2020.
- [14] National Oceanic and Atmospheric Administration, NOAA (2019). What are microplastics? [urlhttps://oceanservice.noaa.gov/facts/microplastics.html](https://oceanservice.noaa.gov/facts/microplastics.html). Accedido 15-02-2020.
- [15] Olaf Ronneberger, Philipp Fischer, Thomas Brox (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. [urlhttps://lmb.informatik.uni-freiburg.de/people/ronneber/u-net/](https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net/). Accedido 10-02-2020.
- [16] Sarah-Jeanne Royer, Sara Ferrón, Samuel T. Wilson and David M. Karl (2018). Production of methane and ethylene from plastic in the environment. [urlhttps://journals.plos.org/plosone/article?id=10.1371/journal.pone.0200574](https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0200574). Accedido 06-03-2020.
- [17] US Federal (public domain), redrawn by User:Dhp1080 in Illustrator (2007). Neurona. [urlhttps://eu.wikipedia.org/wiki/Fitxategi:Neurona.svg](https://eu.wikipedia.org/wiki/Fitxategi:Neurona.svg). Accedido 16-03-2020.
- [18] Vanesa Sanchez (2013). British American Tobacco inicia en España el primer proyecto de reciclado de cigarrillos. [urlhttps://www.cnpt.es/detalle-noticia.asp?id_noticia=111/](https://www.cnpt.es/detalle-noticia.asp?id_noticia=111/). Accedido 24-02-2020.