

Escuela Técnica Superior de Ingeniería de Sistemas Informáticos

Procesamiento automático de ilustraciones:

Clasificación multi-etiqueta de cómics con deep learning



Doble Grado en Ingeniería del Software
y
Tecnologías para la Sociedad de la Información

Autor: Guillermo Iglesias Hernández

Director: Edgar Talavera Muñoz

Junio 2020

Agradecimientos

A Edgar, por ser un ejemplo a seguir, por confiar en mí desde el principio y
por guiarme a nivel académico y personal.

A mis padres y mis abuelos, sin su esfuerzo jamás podría haber llegado a
ningún sitio, espero no decepcionarlos nunca.³

A todos los compañeros con los que he compartido carrera, en especial a Diego
y Raúl por todos los momentos juntos y a Guille, Gonzalo y Yoel por
acompañarme, apoyarme y enseñarme todo lo que sé.

A Ana por ser mi gran apoyo y aceptarme como soy.

A mis amigos y amigas, todo aquel que haya pasado cualquier momento en el
“parterre” conmigo. Vosotros sois la verdadera parte fundamental de mi vida y
sin vosotros no soy nada.

Este trabajo no me pertenece, todos vosotros habéis hecho posible de una
manera u otra que esto salga adelante. Gracias.

Resumen / Abstract

Resumen

El presente trabajo de fin de grado (TFG) se enmarca en el campo de la *inteligencia artificial* y cómo se relaciona dicho campo con el *datascience*. Se presenta un proyecto dividido en dos grandes bloques con un único objetivo, la creación de sistemas inteligentes capaces de realizar una clasificación de varios elementos presentes en una ilustración de tipo cómic japonés. Con el desarrollo del proyecto se pretende constatar la posibilidad de procesar ilustraciones gracias al uso de *deep learning* para que en un futuro se pueda realizar un segundo trabajo de fin de grado en el que se desarrolle una inteligencia artificial capaz de generar ilustraciones a partir de una definición introducida por un usuario humano.

Se realiza la obtención, estudio, documentación y preprocesado de un *dataset* para el entrenamiento de inteligencias artificiales. Una vez obtenido el conjunto de datos se hace uso del *deep learning* para conseguir el resultado buscado. Para ello se define previamente el tipo de salida a buscar y se realizan diversas aproximaciones hasta encontrar una arquitectura final con la que se obtienen los resultados buscados. A medida que se desarrolla la inteligencia artificial el conjunto de datos es modificado para maximizar la eficiencia del sistema, primero unificando los canales de color en un único canal en blanco y negro y posteriormente reduciendo la dimensionalidad de las imágenes.

Una vez obtenido un modelo adecuado se pasan a realizar entrenos en la nube haciendo uso de una tarjeta gráfica potente que maximice la velocidad de los entrenos, los resultados de dichos entrenos son estudiados y comparados entre sí.

Abstract

The present end-of-degree work (TFG) is framed in the field of *artificial intelligence* and how it is related to *datascience*. The project is divided into two large blocks with a single objective, the creation of intelligent systems capable of classifying various elements present in a Japanese comic book type illustration. With the development of the project it is pretended to verify the viability of processing illustrations thanks to the use of *deep learning* so that in the future a second end-of-degree work can be carried out in which an artificial intelligence that is capable of generating illustrations from a definition introduced by a human user can be developed.

The obtaining, study, documentation and pre-processing of a *dataset* for the training of artificial intelligence is carried out. Once the set of data has been obtained deep learning is used to achieve the desired result. To do this, the type of output to be sought is defined beforehand and various approaches are made until a final solution is found with which the desired results are obtained. As artificial intelligence develops, the data set is modified to maximize the efficiency of the system, first unifying the color channels into a single black and white channel and then reducing the dimensionality of the images.

Once a suitable model is obtained, training sessions are carried out in the cloud using a powerful graphic card that maximizes the speed of the sessions. The results of these sessions are studied and compared with each other.

Índice de contenidos

1. Introducción	15
1.1. Trasfondo, ¿cómo hemos llegado aquí?	16
1.2. Motivaciones	20
1.3. Objetivos	21
1.3.1. Objetivo final	21
1.3.2. Objetivos principales	21
1.3.3. Objetivos secundarios	22
2. Estado del arte	23
2.1. Estructura del capítulo	24
2.2. Historia de la inteligencia artificial	24
2.2.1. 300 a.C.-1936 d.C	24
2.2.2. 1936-1960	25
2.2.3. 1960-1970	26
2.2.4. 1970-1980	27
2.2.5. 1980-1987	27
2.2.6. 1987-2011	28
2.3. Recientes hitos de la inteligencia artificial	29
2.4. Redes neuronales convolucionales	30
2.4.1. Base biológica del funcionamiento de las redes neuronales	30
2.4.2. Funcionamiento de las redes convolucionales	30
3. Metodología	33
3.1. Estructura del capítulo	34
3.2. Datamining del <i>dataset</i>	34
3.2.1. <i>Scraping</i> de <i>faneo.es</i>	34
3.2.2. Google Quick, Draw! dataset	36
3.2.3. Tagged Anime Illustrations	36
3.3. Motivos finales de la elección del <i>dataset</i>	38

3.4.	Estudio del dataset <i>Tagged Anime Illustrations</i>	38
3.4.1.	Origen del <i>dataset</i>	39
3.4.2.	Estudio del tratamiento realizado a las ilustraciones	39
3.4.3.	Imágenes poco útiles	41
3.4.4.	Estudio de los metadatos	43
3.4.5.	Ventajas y desventajas	49
3.4.6.	Preprocesado de los meta-datos.	50
3.4.7.	Resumen de la limpieza de meta-datos	55
3.5.	Limpieza de imágenes del <i>dataset</i>	57
3.6.	Estudio de <i>tags</i>	60
3.6.1.	Estructura del objeto <i>tags</i>	60
3.6.2.	Recuento de <i>tags</i>	61
3.6.3.	Descripción de <i>tags</i>	61
3.7.	Descripción de las redes neuronales clasificadoras	71
3.7.1.	Objetivo de las redes neuronales	71
3.7.2.	Salida de la red	71
3.8.	Procesado de las etiquetas para el entrenamiento	76
3.9.	Ahorro de recursos en el entrenamiento	76
3.10.	Entrenamiento con una red de neuronas densa	77
3.10.1.	Estructura de la red	78
3.10.2.	Estructura del entrenamiento	78
3.10.3.	Resultados del entrenamiento	78
3.11.	Transformación de las imágenes a blanco y negro	79
3.12.	Entrenamiento con una red de neuronas convolucional	80
3.12.1.	Salida de la red	80
3.12.2.	Estructura de la red	81
3.12.3.	Resultados del entrenamiento	82
3.13.	Cambios en el modelo de entrenamiento	83
3.13.1.	Reducción de dimensionalidad de las imágenes	83
3.13.2.	Desequilibrio de carga de imágenes en los <i>batches</i>	84
3.13.3.	Carga de imágenes basada en <i>epoch</i>	85
3.13.4.	Entrenamiento de nuestro modelo	88

3.13.5. Métricas del entrenamiento	88
3.14. Entrenamiento tras los cambios realizados	92
3.14.1. Resultados del entrenamiento	92
3.15. Búsqueda de entornos de entrenamiento remotos	93
3.15.1. <i>Google Colaboratory</i>	93
4. Resultados	95
4.1. Estructura del capítulo	96
4.2. Objetivos de los diferentes entrenamientos	96
4.3. Estructura general de las redes	97
4.4. Entrenamiento con 3 clases	97
4.4.1. Etiquetas a clasificar	97
4.4.2. Resultados del entrenamiento	100
4.4.3. Evolución de las predicciones	101
4.4.4. Predicciones sobre imágenes	103
4.5. Entrenamiento con 5 clases	106
4.5.1. Etiquetas a clasificar	106
4.5.2. Resultados del entrenamiento	109
4.5.3. Evolución de las predicciones	110
4.5.4. Predicciones sobre imágenes	111
4.6. Entrenamiento con 7 clases	113
4.6.1. Etiquetas a clasificar	113
4.6.2. Resultados del entrenamiento	116
4.6.3. Evolución de las predicciones	118
4.6.4. Predicciones sobre imágenes	120
5. Impacto social y medioambiental	121
5.1. Impacto del proyecto	122
6. Líneas de investigación	123
6.1. Posibles líneas de investigación	124
6.2. Línea de investigación principal	124
6.3. Líneas de investigación secundarias	124

7. Conclusiones	125
7.1. Conclusiones generales	126
Referencias	126

Índice de figuras

1.	Reconocimiento de elementos de una carretera a través de <i>deep learning</i>	17
2.	Ejemplo de la salida buscada con la realización del presente TFG.	18
3.	Planos de funcionamiento del reloj auto-controlado inventado por Ctesibio.	25
4.	Ejemplo del funcionamiento de una red convolucional.	31
5.	Página de cómic con viñetas de tamaños y forma irregulares.	35
6.	Ejemplo de varios dibujos del dataset <i>Quick, Draw!</i>	36
7.	Bandas horizontales generadas al transformar la imagen original.	40
8.	Recuadro negro generado al transformar la imagen original.	40
9.	Imagen de cómic del dataset.	41
10.	Imagen de un título del dataset.	42
11.	Imagen que tras las transformaciones ha resultado ininteligible.	43
12.	Nota en una imagen subida a la página <i>danbooru.donmai.us</i>	44
13.	Relación de parentesco entre varias imágenes.	45
14.	Imágenes del pool Reaction Faces.	46
15.	Diagrama de flujo de los diferentes estados por los que puede pasar una imagen en la web.	47
16.	Proceso de limpieza de meta-datos.	55
17.	Reducción del tamaño de los archivos a medida que se producen las transformaciones en los mismos.	56
18.	Imagen de cómic útil para el entreno.	57
19.	Imagen de título útil para el entreno.	58
20.	Imagen de proporciones descompensadas útil para el entreno.	59
21.	Resultado de la contabilización del número de <i>tags</i> con muchas imágenes.	61
22.	Agrupaciones de las etiquetas del <i>dataset</i>	70
23.	Ejemplo de la salida buscada de las redes neuronales, con una única clasificación o con varias clasificaciones.	72
24.	Imagen de una chica con un lazo, en la que el lazo es el elemento principal de la ilustración.	74

25.	Ejemplo de combinación de redes para elegir el número de elementos presentes en una imagen.	75
26.	Esquema de la estructura de la red de neuronas densa.	78
27.	Imagen de una chica con el pelo azul antes y después de ser transformada a blanco y negro. Se puede observar que el color de su pelo no puede ser extraído de la imagen en blanco y negro.	79
28.	Representación visual de la red	81
29.	Resumen de la red proporcionado por la función <i>summary</i> librería <i>Keras</i>	82
30.	Imagen antes y después de la reducción de dimensionalidad.	84
31.	Contenido de los <i>arrays</i> donde se almacenan la información de las rutas de las imágenes y sus respectivas etiquetas.	87
32.	Gráfico de métricas de <i>accuracy</i> y <i>loss</i> para los <i>batches</i> a lo largo de un entreno.	89
33.	Gráfico de métricas de las medias de <i>accuracy</i> y <i>loss</i> para los <i>epoch</i> a lo largo de un entreno.	90
34.	Gráfico de métricas del <i>accuracy</i> a lo largo de las predicciones de un entreno.	91
35.	Matriz de confusión para la predicción de las etiquetas de las etiquetas sombrero, pelo corto y pelo largo.	92
36.	Arquitectura de las diferentes redes que usaremos para nuestros entrenos.	97
37.	Imágenes con gorros de formas muy dispares.	98
38.	Diagrama de conjuntos con las imágenes posibles entre las intersecciones de las etiquetas 1girl y short_hair, 1girl y long_hair y hat.	99
39.	Evolución de las medias de las métricas medidas para la clasificación de 3 clases.	100
40.	Matriz de confusión para el <i>batch</i> 150 del <i>epoch</i> 0.	101
41.	Matriz de confusión para el <i>batch</i> 0 del <i>epoch</i> 20.	102
42.	Matriz de confusión para el <i>batch</i> 0 del <i>epoch</i> 225.	102
43.	Resultados de la predicción de una imagen con el pelo de longitud media.	103
44.	Resultados de la predicción de una imagen con un yelmo como sombrero.	104
45.	Resultados de la predicción de una imagen con varios personajes con el pelo corto.	105

46.	Imágenes con un vestido abierto en forma de falda.	107
47.	Diagrama de conjuntos con las imágenes posibles entre las intersecciones de las etiquetas dress y skirt.	108
48.	Evolución de las medias de las métricas medidas para la clasificación de 5 clases.	109
49.	Matriz de confusión para el <i>batch</i> 0 del <i>epoch</i> 1.	110
50.	Matriz de confusión para el <i>batch</i> 100 del <i>epoch</i> 225.	110
51.	Resultados de la predicción de una imagen con una chica de falda y camiseta difícilmente diferenciables.	111
52.	Imagen de la predicción antes y después de bajar su resolución. .	112
53.	Resultados de la predicción de una imagen con una chica de falda.	112
54.	Imágenes con la etiqueta breasts en la que el pecho se presenta de maneras muy diferentes.	114
55.	Imagen con la etiqueta blush en la que la presencia de rubor es muy difícil de ver.	115
56.	Diagrama de conjuntos con las imágenes posibles entre las intersecciones de las etiquetas breasts y blush.	116
57.	Evolución de las medias de las métricas medidas para la clasificación de 7 clases.	117
58.	Matriz de confusión para el <i>batch</i> 200 del <i>epoch</i> 35.	118
59.	Matriz de confusión para el <i>batch</i> 150 del <i>epoch</i> 710.	119
60.	Resultados de la predicción de una imagen de una chica ruborizada, pelo largo y falda.	120

Índice de tablas

1.	Resumen de las ventajas e inconvenientes de las posibles opciones para obtener un <i>dataset</i>	38
2.	Descripción de los campos de un objeto json de una ilustración. .	49
3.	Motivos de la eliminación de los campos de los archivos de meta-datos.	52
4.	Decisiones tomadas sobre los campos de meta-datos tras el estudio realizado.	54
5.	Resultados de la contabilización de imágenes de proporciones desiguales.	59
6.	Descripción de los motivos por los que se deciden eliminar ciertas etiquetas de nuestra lista.	64
7.	Descripción de los motivos por los que se deciden agrupar ciertas etiquetas de nuestra lista.	64
8.	Descripción de los motivos por los que se deciden eliminar ciertas etiquetas de nuestra lista.	75

Introducción

1.1. Trasfondo, ¿cómo hemos llegado aquí?

Durante los últimos años el campo de la inteligencia artificial se ha erguido como uno de los desafíos más importantes a los que se enfrenta la humanidad. La creencia generalizada es que los avances en este campo harán que en décadas los humanos sean plenamente remplazados por máquinas conscientes de su existencia. Esta creencia en la singularidad¹ envuelve a todo lo relacionado con los avances en inteligencia artificial.

Durante la última década se han producido importantes avances en esta materia, sin embargo la realidad está muy alejada de la singularidad. Los últimos descubrimientos han permitido realizar tareas que antes se consideraban imposibles de realizar por una máquina. Todos estos proyectos están enfocados a que, gracias a la computación, se realicen tareas muy concretas y por lo tanto quedan muy lejos de la replicación completa de un ser humano.

La realidad es que actualmente no se plantean las aplicaciones de la inteligencia artificial como una posibilidad de reemplazar a un ser humano sino más bien como procesos para automatizar tareas concretas por parte de un ordenador. Este tipo de tareas pueden llegar a ser muy complejas e incluso en algunos casos las máquinas pueden llegar a obtener mejores resultados que un humano.

Con las herramientas actuales se pueden crear inteligencias artificiales capaces de adaptarse a diferentes funcionalidades basándose todas ellas en los mismos conceptos matemáticos e informáticos. Desde la diferenciación de imágenes de tomates maduros hasta los procesadores de lenguaje natural las inteligencias artificiales se basan en la misma base teórica y ahí es donde se encuentra la fortaleza de la inteligencia artificial actual, en la capacidad de adaptación de los algoritmos para poder resolver tareas diversas.

El modelo más utilizado en la actualidad son las redes neuronales artificiales a través del conocido como *Deep Learning*², este se basa en los principios fundamentales biológicos por los cuales funcionan las neuronas en una persona, gracias al estudio de cómo los impulsos eléctricos se propagan a través del cerebro humano se consiguen replicar ciertos aspectos del funcionamiento de la mente humana en una máquina. Este tipo de modelo computacional es el más utilizado actualmente pues propone soluciones para una gran diversidad de usos consiguiendo grandes resultados, la flexibilidad y relativa sencillez de las redes neuronales hacen que actualmente sea el modelo más utilizado.

¹La singularidad tecnológica es un término propuesto por Nicolas de Condorcet en el siglo XVIII para definir la posibilidad de que las máquinas se auto-mejoren recursivamente hasta llegar a un punto incontrolable para la inteligencia humana en el que los robots superen al ser humano.

²El *Deep Learning* es la disciplina de la inteligencia artificial que crea sistemas que son capaces de aprender automáticamente. Las redes de neuronas artificiales son un ejemplo de *deep learning*.

Uno de las aplicaciones más interesantes de las redes neuronales es la visión artificial³. En este campo las redes neuronales permiten obtener unos resultados mucho mejores que con la utilización de algoritmos tradicionales. Gracias a ello se han producido avances muy importantes durante los últimos años permitiendo el desarrollo de campos tan importantes como la conducción autónoma (ver figura 1).



Figura 1: Reconocimiento de elementos de una carretera a través de *deep learning*.

Uno de los principales problemas derivados de la utilización de redes neuronales es el entrenamiento de las mismas, pues para conseguir que unas redes de neuronas comprendan y consigan solucionar un problema concreto primero deben enfrentarse a él numerosas veces y aprender cómo funciona. Este proceso se denomina *entrenamiento* de las redes neuronales y para realizarlo hacen falta una gran cantidad de datos del problema original para que las redes de neuronas aprendan de esos datos y puedan en última instancia enfrentarse a situaciones nunca antes vistas.

Para construir estos conjuntos de datos o *datasets* hace falta recopilar grandes volúmenes de información de manera automática. Gracias a la aparición de nuevos modelos de bases de datos no relacionales durante los últimos años se han podido recopilar datos de maneras que antes resultaban muy complejas. El desarrollo de estos nuevos métodos de recopilación de información ha supuesto el auge del *Big Data*⁴ que, junto a los nuevos descubrimientos de los últimos años en la inteligencia artificial, ha supuesto una retroalimentación entre ambos campos impulsando su desarrollo conjunto.

³La visión artificial o visión computacional es la rama de la informática ocupada de extraer y procesar información de imágenes o vídeos.

⁴El *Big Data* es la rama de la informática dedicada a la captura, gestión, procesamiento y análisis de grandes volúmenes de datos.

La ciencia de datos o *datascience* es la disciplina que se encarga de extraer y tratar información de grandes volúmenes de datos. Dentro de este campo encontramos a la *minería de datos* como la rama que se encarga de recoger los datos más significativos de una fuente con el fin de aportar el máximo valor para cuando estos sean tratados. Otra parte esencial del *datascience* es el preprocesamiento de los datos por el cual se normaliza y prepara la información recogida para que esta pueda ser tratada de una manera automática adecuadamente.

Respecto al campo de la inteligencia artificial, el *datascience* se encarga de extraer y preparar los conjuntos de datos o *datasets* para que posteriormente sirvan para crear las redes neuronales con las cuales funcione la inteligencia artificial.

Con el desarrollo del presente proyecto se pretende la obtención de redes neuronales capaces de escribir y dibujar cómics. Para conseguir estas redes primero se deben realizar como trabajos previos la obtención de un *dataset* que sirva para el entrenamiento de la inteligencia artificial, este *dataset* servirá como entrada de entrenamiento de las redes para que estas aprendan las características principales de un cómic y en un futuro las redes sean capaces de crear nuevas obras. Por otra parte la creación de las redes neuronales generadoras desde cero supone un trabajo demasiado ambicioso como para realizarlo sin un estudio previo. Debido a esto primero se propone la realización de redes de neuronas capaces de clasificar cómics atendiendo a los elementos presentes en sus ilustraciones. Una vez realizado este estudio previo se pueden crear y entrenar las redes generadoras de cómics con mayor conocimiento y probabilidad de éxito. En la figura 2 podemos ver un ejemplo del objetivo buscado con el desarrollo del presente trabajo.



Una chica con el pelo largo y de color verde, en el pelo tiene una horquilla y un lazo en su ropa

Figura 2: Ejemplo de la salida buscada con la realización del presente TFG.

Como el objetivo principal del proyecto es demasiado amplio como para ser recogido en un único trabajo, se ha tomado la decisión de dividir la tarea en dos partes. El presente Trabajo de Fin de Grado (TFG) supone por una parte la obtención, preprocesado y estudio de las características del *dataset* de entrenamiento, y por otra parte la implementación de las redes neuronales necesarias para realizar una clasificación supervisada de las ilustraciones. Parte del objetivo al realizar este trabajo es realizar un estudio de la viabilidad del *dataset* para implementar otro tipo de redes neuronales conocidas como *Generative Adversarial Neural Networks* (GAN) cuya particularidad es la de ser capaces de generar nuevas imágenes acordes al estilo con las que han sido entrenadas. Finalmente y con el objetivo de aprovechar la necesidad de entregar dos TFGs para completar el doble grado se va a realizar un proyecto más inmersivo y completo. Por ello, se pretende que el desarrollo culmine con la implementación de redes neuronales generadoras, cuyo esfuerzo no podría ser abordado en un solo TFG, aprovechando el trabajo y estudio previo necesario que se ha realizado en el presente proyecto.

Además, para su realización se ha tenido que realizar una labor de investigación previa muy importante, puesto que los conocimientos de *deep learning* adquiridos durante el grado eran limitados. Esto es debido a que para poder realizar una inteligencia artificial hay que conocer en profundidad los fundamentos teóricos en los que se basa y para ello se ha dedicado una gran parte del esfuerzo y tiempo en la obtención de estos conocimientos para que a la hora de desarrollar el trabajo este se realice de manera correcta.

1.2. Motivaciones

La principal causa de la realización del proyecto es la inmersión por completo en un proyecto relacionado con la inteligencia artificial. Puesto que en el grado la carga lectiva de este campo no es muy elevada se vio la posibilidad de cubrir esta falta a través del desarrollo del presente trabajo. Según he ido descubriendo las metodologías relacionadas en el campo he ido despertando una curiosidad por todo lo que envuelve el campo, además de encontrar un marco en el cual poder desarrollar mis inquietudes por la investigación.

La realización de un proyecto completo es uno de los grandes motivos por los que se escogió este trabajo. No suponía ninguna motivación formar parte de un proyecto en el que mi trabajo tuviera un impacto pequeño.

Con todo esto, se encontró un campo relacionado con las matemáticas que pudiera proporcionarme un marco en el que seguir formándome y que me permitiera desarrollar un proyecto completamente desde cero, además de estar en un ámbito relacionado con la investigación.

Dentro de las grandes posibilidades que la visión computacional ofrece, el aspecto relacionado con el arte es uno de los que se consideraron más interesantes debido a los grandes avances llevados a cabo durante los últimos años. La creación de una inteligencia artificial capaz de generar dibujos supone grandes implicaciones respecto a la concepción del ser humano de la inteligencia artificial. Este objetivo se pretende alcanzar a través de la realización del segundo TFG mencionado anteriormente.

1.3. Objetivos

Debido a la complejidad del proyecto se ha decidido dividir los objetivos del mismo en tres apartados para una mayor comprensión. De esta manera se define el *objetivo final* que remarca la finalidad resultante de combinar ambos TFGs y por otro lado, los *objetivos principales y secundarios* del presente trabajo.

1.3.1. Objetivo final

El objetivo final del desarrollo del proyecto es la creación de redes neuronales generadoras de cómics sintéticos. Esto define el fin último de los esfuerzos desarrollados en este trabajo, pues como se ha indicado anteriormente, para la creación de redes de neuronas generadoras primero debemos obtener un *dataset* y realizar un estudio para comprobar su viabilidad mediante la creación de redes clasificadora de ilustraciones.

La generación de imágenes de cómic es el motivo principal por el que se desarrollan ambos trabajos, sin embargo con el desarrollo de este trabajo no se pretende obtener esas redes sino que se pretende realizar un estudio de la viabilidad de crearlas para, en un futuro segundo TFG culminar el trabajo de su desarrollo.

1.3.2. Objetivos principales

Los objetivos principales del desarrollo de este trabajo son:

- **Obtención y estudio de un *dataset* de ilustraciones pertenecientes a un cómic para usarlo como entrada para el entrenamiento de redes neuronales:** Para poder desarrollar redes neuronales necesitamos conjuntos de datos para su entrenamiento, por eso debemos encontrar un *dataset* suficientemente amplio como para desarrollar nuestro objetivo. Además de su obtención debemos realizar un estudio profundo de su estructura y contenido con el fin de tener el mayor control posible de cómo se realizará el entrenamiento.
- **Preprocesado del *dataset* para el entrenamiento:** Una vez obtenido y estudiado el *dataset* debemos tratar su contenido para que pueda ser procesado automáticamente en el entrenamiento. Con este proceso además conseguiremos mejorar la eficiencia del entrenamiento pues adaptaremos el conjunto de datos para nuestra necesidad en concreto.
- **Realización de redes de neuronas capaces de identificar elementos compositivos de diversas ilustraciones:** Con estas redes se conseguirán clasificar ilustraciones atendiendo a sus elementos compositivos y no a sus meta-datos, es decir se hará una clasificación por su contenido y no por su forma.

- **Estudio de viabilidad de la creación de redes generadoras:** El desarrollo de las redes clasificadoras servirá para confirmar la posibilidad de redes generadoras, pues si conseguimos que una inteligencia artificial sea capaz de diferenciar elementos de ilustraciones estaremos prácticamente seguros de que se puede realizar el proceso inverso, obteniendo ilustraciones completas a partir de los elementos que queremos en ellas. Decidimos que primero creamos las redes clasificadoras ya que son más sencillas que las generadoras.

1.3.3. Objetivos secundarios

Debido a la metodología que se pretende llevar a cabo, el aprendizaje de nuevas herramientas que no se vieron durante el desarrollo del grado son parte de los objetivos secundarios:

- **Lenguaje de programación *python*:** El lenguaje de programación *python* es uno de los más extendidos, y de las que mejores herramientas para la inteligencia artificial ofrece. El conocimiento de este lenguaje se considera imprescindible para la realización del proyecto.
- **Librerías de *python* para el desarrollo de inteligencia artificial:** Conocer y estudiar en profundidad las posibles funcionalidades de las librerías de *python*, en especial con *Tensorflow* y *Keras*. Estas librerías nos proporcionan las herramientas necesarias para poder crear las redes neuronales necesarias para nuestro proyecto.
- **Herramientas de procesamiento remoto:** El uso de herramientas de procesamiento de datos de forma remota, en concreto el uso de *Google Colab* para la investigación en inteligencia artificial.

De igual modo hay ciertas materias que se vieron durante el desarrollo de la carrera que han tenido que ser profundizadas para desarrollar el trabajo:

- **Formato de datos *JSON*:** Para la realización del trabajo se deberá dar un formato válido a los archivos de meta-datos⁵ de las imágenes. De igual manera dichos datos deben ser leídos y transformados para la preparación del *dataset*. Todo este proceso constituye una fuente de aprendizaje y profundización en el formato *JSON* y en concreto el tratamiento en *python* del mismo.

⁵Los meta-datos son los datos que describen el contenido de un archivo, es decir, los datos de la propia información de los datos.

Estado del arte

2.1. Estructura del capítulo

En esta sección repasaremos el marco teórico y cronológico en el que se basa el trabajo, repasando los fundamentos más importantes que hacen posible la realización del mismo y que plasman el estudio previo realizado para adquirir los conocimientos necesarios en el campo de *deep learning*.

2.2. Historia de la inteligencia artificial

Entendemos como inteligencia artificial a la posibilidad de automatizar tareas de la misma manera que un humano las realizaría, en este sentido podemos diferenciar varias etapas en la historia de la humanidad en cuanto al campo de la inteligencia artificial se refiere.

2.2.1. 300 a.C.-1936 d.C

Durante esta época podemos entender que la inteligencia artificial está presente en la lógica formal que se ha ido desarrollando a lo largo de la historia. Esta lógica plantea ciertas normas que permiten llevar a cabo la toma de decisiones siguiendo un conjunto de reglas previamente establecido.

Los hitos más relevantes de este periodo son:

- (*385 a.C-322 a.C*) - El filósofo griego Aristóteles es considerado como el padre de la lógica formal. Su trabajo relacionado con este campo se recoge en el conjunto de obras titulado *Órganon*[8], estas obras se desarrollaron a lo largo de la vida de Aristóteles y por tanto no se le puede dar una fecha exacta de creación más allá de el periodo en que su autor vivió. Se puede considerar la lógica aristotélica como la primera aproximación del ser humano a describir de una manera esquematizada el funcionamiento de la mente humana.
- (*siglo III a.C*) - Más adelante el inventor Ctesibio de Alejandría ideó un reloj capaz de reiniciar su ciclo cada año gracias a un conjunto de engranajes, un sifón y agua (ver figura 3). Esta invención se considera la primera máquina autorregulada ideada por el ser humano, y una de las primeras aproximaciones a la inteligencia artificial.

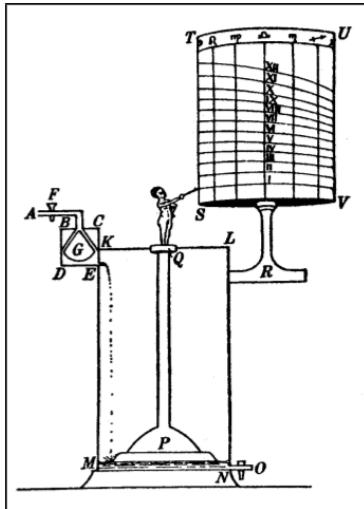


Figura 3: Planos de funcionamiento del reloj auto-controlado inventado por Ctesibio.

- (1315 d.C) - El filósofo y teólogo Ramón Llul ideó una máquina llamada *Ars Magna* capaz de automatizar razonamientos obteniendo como salida la veracidad o falsedad de ciertas proposiciones. El funcionamiento de este artefacto se recoge en el libro llamado *Ars Magna* y supone un importante avance en la concepción del ser humano pues abría la posibilidad de que una máquina siguiese un razonamiento en concreto.

2.2.2. 1936-1960

Este periodo constituye el inicio de la inteligencia artificial tal y como la conocemos hoy en día. Durante esta época se comienzan a hacer los primeros estudios para trasladar el pensamiento humano a una máquina.

Con los avances surgidos a comienzos del siglo XX en el campo de la biología y psicología se adquiere un mayor conocimiento de los mecanismos humanos de pensamiento. Por otra parte se producen las primeras aproximaciones a la computación que comienzan a dar luz sobre la posibilidad de automatizar ciertas tareas. También en esta época se crean los primeros modelos teóricos y definiciones de la inteligencia artificial.

Los acontecimientos más relevantes de este tiempo son:

- En 1936 Alan Turing diseña la *Máquina Universal* sentando las bases de la teoría de la computación.

- Warren McCulloch y Walter Pitts proponen en el año 1943 un modelo matemático que explica el funcionamiento de una neurona. Dicho modelo es la base del funcionamiento de las redes neuronales modernas.
- Alan Turing publica el artículo titulado *Computing Machinery and Intelligence*[1] en la revista *Mind* en el año 1950 en el que se considera profundamente la posibilidad de que una máquina pueda tomar decisiones como un humano lo haría.
- En el año 1951 Marvin Lee Minsky diseña la *Stochastic Neural Analog Reinforcement Calculator* (SNARC). Esta máquina es la primera en utilizar varios de los conceptos actuales de la inteligencia artificial como el aprendizaje por refuerzo.
- En 1956 se produce la conferencia de Dartmouth por la cual se bautiza a la inteligencia artificial.
- El desarrollo del perceptrón tal y como lo conocemos hoy en día se produce en el año 1957 por parte de Frank Rosenblatt basándose en los modelos propuestos por McMulloch y Pitts años antes. En este año se implementa por primera vez el funcionamiento de un perceptrón en un ordenador.

2.2.3. 1960-1970

Esta época constituye la consolidación de la idea de que un ordenador podría ser capaz de imitar la inteligencia humana, durante este periodo el mayor cambio que se produce respecto al anterior es que comienzan a surgir los primeros programas reales enfocados a la inteligencia artificial que pasa de ser un campo teórico a comenzar a tener aplicaciones prácticas reales. También se produce una apuesta por la inteligencia artificial, se comienza a invertir dinero en el campo impulsando el desarrollo y los avances de esta época.

Los avances más importantes de estos años son:

- El *General Problem Solver* (GPS) es un software desarrollado por Alan Newell y Herbert Simon en el año 1957 cuyo objetivo es la resolución general de problemas. El principal hito de este programa es que intenta imitar a la inteligencia humana consiguiendo adaptarse a diferentes situaciones, esto es, por primera vez se intenta que una máquina tenga el enfoque a la hora de resolver problemas que tendría un humano. Con el programa se consiguen realizar tareas como jugar al ajedrez o resolver tareas como las torres de Hanoi.
- En el año 1963 ARPA proporciona dos millones de dólares al MIT para la investigación en inteligencia artificial. Esto supone la apuesta por grandes compañías en la investigación en este campo.

- Daniel G Bobrow crea el software STUDENT en el año 1964. Este programa es capaz de resolver problemas de álgebra.
- En 1966 se funda el laboratorio de inteligencia artificial en Edinburgo por Donald Michie.
- En 1968 Quillian crea el esquema de representación de red semántica. Este esquema con forma de grafo proporciona relaciones entre diferentes elementos, lo que permite la clasificación de elementos atendiendo a sus características.

2.2.4. 1970-1980

Esta época supone el llamado primer invierno de la inteligencia artificial. Las altas expectativas puestas durante los años anteriores sobre la inteligencia artificial unidas a los pocos avances conseguidos hacen que se pierda la mayor parte de inversión e interés en el campo.

Los motivos por los que no se producen grandes avances durante estos años son principalmente la falta de conocimiento y de capacidades de los ordenadores. Estos factores son los causantes de que nazcan problemas imposibles de solucionar por el tiempo que llevaría lo cual provoca un desánimo en los expertos que comienzan a cuestionar la viabilidad de avances en el campo.

En este sentido durante este periodo se producen el siguiente acontecimiento:

- En el año 1973 el matemático inglés James Lighthill publica un informe del estado de la inteligencia artificial. En él se critican las investigaciones en áreas como la robótica o el procesamiento del lenguaje natural. Este informe supone la base por la cual el gobierno de Reino Unido terminase la financiación al campo de la inteligencia artificial excepto en dos universidades. Esto supone el punto de inflexión por el que comienza el invierno de la inteligencia artificial.

2.2.5. 1980-1987

Los sistemas expertos nacen durante este periodo. Con su nacimiento se produce un resurgimiento de la inteligencia artificial por los buenos resultados obtenidos. Con ello se vuelven a producir grandes inversiones en la investigación. Durante este periodo se abandona la arquitectura Von Neumann debido al auge del uso de *Prolog*⁶ como lenguaje máquina.

⁶El lenguaje de programación *Prolog* surgió a principio de los años 80 y se basa en escribir en las sentencias primero los consecuentes y luego los antecedentes al contrario que los lenguajes habituales.

El acontecimiento más relevantes de esta época son:

- En el año 1982 el gobierno japonés crea el proyecto denominado *Quinta generación de computadores* dotándolo de 850 millones de dólares de inversión. El principal objetivo de este proyecto era crear una nueva clase de ordenadores basados en la inteligencia artificial, tanto en software como en hardware.

2.2.6. 1987-2011

Se produce un segundo invierno de la inteligencia artificial. Las causas que lo provocan son principalmente el fracaso ante las grandes expectativas puestas en los sistemas expertos, que eran muy caros de mantener y podían llegar a obtener ante ciertas situaciones grandes fallos. La financiación de algunos gobiernos comienza a desaparecer provocando nuevamente un estancamiento de la investigación. Pese a ciertos esfuerzos por avanzar en las capacidades de la inteligencia artificial, estos hechos están aislados y casi siempre son insuficientes. Uno de los nichos donde se sigue desarrollando la inteligencia artificial es en la teoría de juegos llegando a conseguir grandes hazañas.

Los hitos que marcaron esta época son:

- El proyecto Quinta generación de computadores termina en el año 1993 cesando la financiación en este campo.
- Durante este periodo surge también la *paradoja de Moravec*. Tras los últimos avances llevados a cabo en el campo Hans Moravec, Rodney Brooks, Marvin Minsky afirman que las máquinas pueden tener resultados similares a los humanos en pruebas de inteligencia, sin embargo todas las pequeñas funciones del ser humano son muy difíciles de imitar por un robot.⁷
- En el año 1993 el supercomputador *Deep Blue* consigue derrotar a una partida de ajedrez al vigente campeón Gari Kaspárov. Pese a parecer un acontecimiento anecdótico, este hecho fue muy mediático y mostró los avances de la inteligencia artificial al gran público de la época.

⁷ «En general, no somos conscientes de nuestras mejores habilidades [...] somos más conscientes de los pequeños procesos que nos cuestan que de los complejos que se realizan de forma fluida» [4]

2.3. Recientes hitos de la inteligencia artificial en el área de la visión artificial

Desde el comienzo de la década de 2010 la inteligencia artificial ha sufrido una nueva época de esplendor, las inversiones en investigación, los avances y el interés por el público general hacen que se considere hoy en día como una de las materias más importantes de la ciencia y uno de los retos más importantes a los que se enfrenta la humanidad.

Se considera que actualmente la inteligencia artificial se encuentra en un buen momento. Hoy en día los mayores esfuerzos de la inteligencia artificial se centran en mejorar los sistemas actuales, buscando mejoras en los mismos o abriendo vías de investigación en nuevas arquitecturas que permitan mejorar resultados en problemas actuales.

Durante los últimos años los hecho más importantes sucedidos que conciernen a el presente trabajo son:

- En el año 2011 se publica el artículo *Flexible, High Performance Convolutional Neural Networks for Image Classification* por Dan Ciresan[5] en el que se presenta una implementación basada en GPU de las redes convolucionales ya conocidas anteriormente⁸. Gracias a esta implementación se consigue un aumento en la velocidad de los entrenamientos gracias a la paralelización de las operaciones producidas en una tarjeta gráfica, lo que permitió desarrollar de manera más rápida redes neuronales convolucionales, aumentando la velocidad a la que se desarrollaban los proyectos y, en última instancia, una innovación que impulsó el campo de la visión artificial por inteligencia artificial.
- En el año 2014 se publica el artículo *Generative Adversarial Networks* por Ian Goodfellow[6] en el que se propone una nueva arquitectura para el entrenamiento de redes neuronales. Las redes neuronales adversarias suponen un nuevo modelo en el que una inteligencia artificial es capaz de aprender ciertos aspectos a través de la competición con otra inteligencia artificial. Dicho entrenamiento se hace en paralelo consiguiendo resultados que antes eran imposibles. Con este nuevo esquema se dan lugar a nuevos proyectos y, en lo que nos concierne en este trabajo, a inteligencias artificiales capaces de crear artefactos como si un humano fuese el que lo realizase.
- En el año 2018 la compañía NVIDIA realiza un proyecto por el cual consigue la creación de imágenes de caras a través de inteligencia artificial[7]. Este proyecto supone un ejemplo a seguir para la realización del presente proyecto pues supone el uso de redes GAN para la creación de imágenes, en nuestro caso la creación sería de dibujos en vez de caras.

⁸En el año 1980 Kunihiko Fukushima presenta el modelo de redes de neuronas convolucionales[3]

2.4. Redes neuronales convolucionales

Las redes neuronales convolucionales son una implementación de redes neuronales artificiales que se basan en las operaciones matriciales para realizar sus cálculos. Debido a que las operaciones se realizan en forma de matriz, la manera de aprender de la red se asemeja al funcionamiento de las neuronas de la corteza visual.

2.4.1. Base biológica del funcionamiento de las redes neuronales

El modelo biológico en el que se basan este tipo de redes es el de la parte del cerebro conocida como *corteza visual primaria*. En esa parte del cerebro se mapea la visión humana y las neuronas presentes en ella reaccionan a estímulos de la visión, propagando la información a través de impulsos eléctricos de mayor o menor intensidad dependiendo de la potencia del estímulo.

Cuando las imágenes aparecen en la visión de la persona, las neuronas de las áreas visuales más bajas se encargan del reconocimiento de formas más sencillas reaccionando a formas simples como líneas y estas neuronas disparan estímulos a otras neuronas que son capaces de reconocer formas más complejas usando la información de las primeras[2]. De esta forma es cómo se produce el reconocimiento de imágenes a *bajo* y *alto* nivel, dependiendo de la profundidad de las neuronas que activen.

2.4.2. Funcionamiento de las redes convolucionales

Las redes convolucionales constan de múltiples capas en las que cada una de ellas cuenta con diversos filtros por los cuales pasa la información a procesar, por ejemplo una imagen. Después de que se apliquen los filtros de una capa se aplica una función no lineal de la misma forma que en un perceptrón tradicional. A medida que se suceden los filtros de las capas se van extrayendo características de más alto nivel de la información que se procesa, llegando a conseguir obtener la información buscada.

La arquitectura de las redes convolucionales se basa esencialmente en la coexistencia de tres tipos de capas:

Capas convolucionales: Este tipo de capas se corresponden con las neuronas sencillas de una perceptrón multicapa. Las operaciones efectuadas en estas capas se denominan convoluciones, en ellas se aplican una cantidad de filtros a una información gracias a los cuales se consigue extraer ciertas características de dicha información. A medida que las convoluciones se van sucediendo en las redes se van reduciendo el número de filtros y aumentando el tamaño de los mismos consiguiendo así la obtención de características de más alto nivel de la información procesada.

Capas de reducción de muestreo: Para poder procesar grandes volúmenes de información las redes convolucionales cuentan con capas dedicadas a la reducción de la dimensionalidad de la información que procesan. Estas capas se dedican a agrupar información, la salida debe representar la información de las neuronas que se agrupan para intentar reducir al máximo la pérdida de información.

Con el uso de este tipo de capas se consigue reducir el tamaño de la información a procesar, gracias a esto se aumenta la eficiencia del entrenamiento de las redes y se avanza hacia la obtención de características de alto nivel, pues para obtener la salida final de la red se necesitan pocas neuronas que generen la salida deseada.

Capa de aplanamiento: La capa de aplanamiento o *flatten* tiene como única función transformar las matrices de las convoluciones en un vector que sirva de entrada para un perceptrón con el cual se consiga la salida de la red. En esta capa se realizan operaciones que sencillamente se asigna a cada elemento de la matriz de origen un elemento del vector destino, consiguiendo así una única dimensión en la que se guardan todos los elementos de la matriz.

Capas de clasificación: Una vez producida la capa de aplanamiento de la información se clasifican las características extraídas durante las convoluciones haciendo uso de neuronas tal y como sucedería en un perceptrón multicapa. La misión principal de estas capas es obtener una salida clasificando las características extraídas gracias a las convoluciones.

En la figura 4 podemos ver un ejemplo de la estructura de una red de neuronas convolucional.

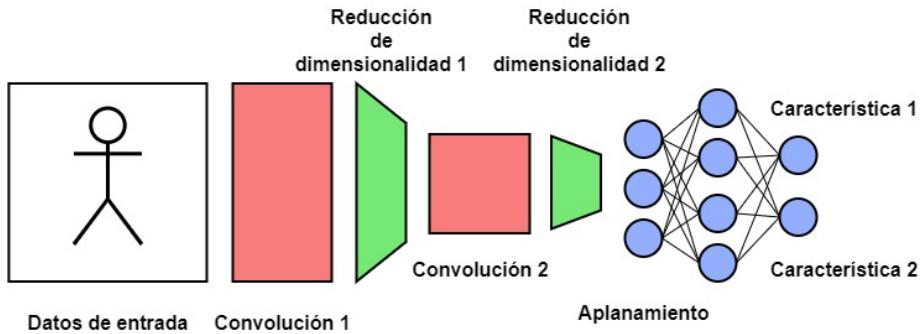


Figura 4: Ejemplo del funcionamiento de una red convolucional.

Metodología

3.1. Estructura del capítulo

Durante esta sección se repasará todo el trabajo realizado, dividiendo el contenido en dos grandes bloques, uno basado en la obtención, estudio y preprocesado del *dataset*, y el segundo bloque con todo el trabajo realizado de inteligencia artificial para la implementación de las redes neuronales.

3.2. Datamining del *dataset*

Para la creación de un algoritmo de inteligencia artificial se debe encontrar un conjunto de datos con el que podamos realizar el entrenamiento de las redes neuronales. Para la búsqueda de este *dataset* se debe realizar una búsqueda exhaustiva de posibles conjuntos de imágenes clasificadas con la suficiente calidad y la estructura necesaria para poder ser tratadas posteriormente. Además dicho conjunto ha de ser suficientemente grande como para poder realizar los entrenamientos de las redes evitando posibles resultados de *overfitting* que se produzcan en la red.

3.2.1. Scraping de *faneo.es*

Una de las posibilidades que se plantean es obtener un gran conjunto de imágenes a través de la descarga masiva de imágenes de cómic de la página *faneo.es*⁹. Con ello se obtendría un gran conjunto de datos con el que poder trabajar más adelante. Estos datos además podrían ser clasificados por estilo de cómic o autor abriendo la posibilidad de poder crear una inteligencia artificial capaz de distinguir entre estilos o autores.

Las imágenes de la web tienen formato de página de cómic, debido a esto se considera que la inteligencia artificial podría encontrar problemas a la hora de identificar una imagen. Es decir, debido a la naturaleza del formato, una imagen está formada a su vez de varias ilustraciones insertadas en cada una de las viñetas. Esto podría provocar que la inteligencia artificial tuviese problema a la hora de identificar cada dibujo y sus características.

⁹*faneo.es* es una página web donde diferentes autores pueden subir cómics que hayan creado para que puedan ser leídos por los usuarios de la página

Una posibilidad para solucionar este problema es dividir la imagen en cada una de las viñetas que la componen. Sin embargo esta tarea no es trivial puesto que la composición de cada página es distinta y por lo tanto no se puede aplicar la misma separación en viñetas para las diferentes imágenes. Existe la posibilidad de usar una inteligencia artificial para dividir estas imágenes en viñetas y con ello conseguir ilustraciones individuales sin embargo se desestima esta opción debido al gran coste que supondría en tiempo y que, incluso obteniendo buenos resultados, las imágenes individuales que se obtendrían como salida tendrían un formato no uniforme debido a que cada viñeta es de un tamaño, forma y proporciones diferente. En la figura 5 podemos observar un ejemplo de una página de cómic donde la segmentación de sus viñetas supondría las complicaciones anteriormente citadas.



Figura 5: Página de cómic con viñetas de tamaños y forma irregulares.

Otro problema es que tras la utilización de esta técnica no se obtendría un *dataset* clasificado por el contenido de sus imágenes, sino por la meta-information de ellas lo cual no es el objetivo buscado inicialmente.

Debido a todo esto se desestima la posibilidad de utilizar *scraping* para obtener el *dataset* buscado.

3.2.2. Google Quick, Draw! dataset

El juego *Quick Draw!* desarrollado por Google consiste en dibujar imágenes de las palabras que el juego propone. Mientras que el jugador dibuja las imágenes una inteligencia artificial intenta adivinar lo que la persona dibuja hasta que lo resuelve.

Las imágenes generadas por los usuarios de este juego están publicadas en un gran *dataset* publicado en internet de libre uso. Estos dibujos están clasificados por el elemento a dibujar lo que nos proporcionaría una gran cantidad de imágenes clasificadas por su contenido. Además todas las imágenes están en un mismo formato lo que facilita la labor de integración en nuestro sistema.

Sin embargo, las imágenes representan dibujos muy sencillos que en ningún caso son ilustraciones completas. Además la calidad artística de las imágenes es muy escasa, todos los dibujos son, previsiblemente, dibujados por usuarios comunes sin conocimientos artísticos y usando como instrumentos de dibujo un ratón o la pantalla táctil del móvil, lo que disminuye la calidad del *dataset*. En la figura 6 podemos observar varias ilustraciones de este *dataset*.

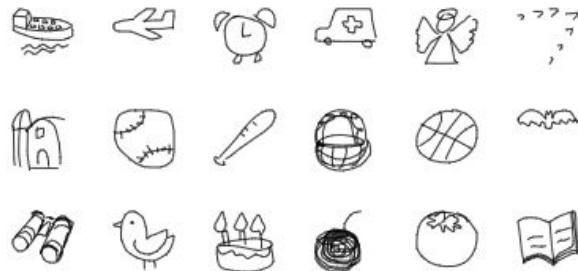


Figura 6: Ejemplo de varios dibujos del dataset *Quick, Draw!*

EL uso de este conjunto de datos se considera inapropiado, las imágenes están demasiado lejos del objetivo que se pretende conseguir con la realización del trabajo además de la baja calidad de estas.

3.2.3. Tagged Anime Illustrations

Siguiendo con la investigación de posibles conjuntos de datos para el entrenamiento se llega al *dataset* publicado en la web [Kaggle.com](https://www.kaggle.com/c/draw-a-manga-style-anime-illustration) denominado *Tagged Anime Illustrations*. Dicho *dataset* consta de aproximadamente 300.000 imágenes de ilustraciones artísticas de estilo manga, además cuenta con un subconjunto de imágenes de caras de personajes.

Las ilustraciones están realizadas por público general y la mayoría de ellas son *fan art*¹⁰. Estas imágenes vienen acompañadas de archivos de meta-datos en un formato pseudo-json. Por otra parte se encuentra en el *dataset* un subconjunto de imágenes de caras que vienen clasificadas dependiendo al personaje al que le pertenezcan.

Tras un estudio de la composición y estructura del *dataset* se decide escogerlo puesto que proporciona grandes ventajas respecto a las opciones anteriores. Posteriormente haremos un estudio más profundo del origen de las ilustraciones que dará sentido a muchos de los motivos por los que se escogerá finalmente esta fuente de datos.

Se desestima el uso del subconjunto de imágenes de caras debido a que, por una parte, una ilustración de una cara no es una ilustración completa, pues le faltan elementos compositivos como la ropa, la postura o el entorno. Con el desarrollo de este proyecto se quiere buscar recrear a través de inteligencia artificial una ilustración completa con todos los elementos que la componen. Por otra parte el número de imágenes por cada uno de los personajes es escaso, para la realización del entrenamiento buscamos aproximadamente 10.000 imágenes por cada una de las clasificaciones que deseamos obtener, sin embargo aquí encontramos alrededor de 100 imágenes por cada personaje, lo cual es insuficiente. Por estos motivos, además de porque el *dataset* de *Tagged Anime Illustrations* es idóneo para nuestra situación, decidimos no usar el subconjunto de imágenes de caras.

¹⁰Las ilustraciones *fan art* son dibujos realizadas por una persona que no tiene los derechos del contenido que está dibujando, por lo general fans de aquello que se dibuja.

3.3. Motivos finales de la elección del dataset

En la tabla 1 se muestra un resumen de los motivos finales por los que se escogió como *dataset* para utilizar en el proyecto *Tagged Anime Illustrations*:

Dataset	Imágenes de manga	Calidad artística
Scraping de faneo.es	Sí, pero sería necesario filtrar qué ilustraciones descargar	Muy alta
Quick, Draw!	No	Muy baja
Tagged Anime Illustrations	Muy alta	Alta
Dataset	Meta-information de los elementos de la imagen	Meta-datos estructurados
Scraping de faneo.es	No, para conseguirla sería necesario procesar las imágenes	No
Quick, Draw!	Sí, pero muy poca	Sí
Tagged Anime Illustrations	Sí	Sí, pero ha de ser preprocesada mínimamente para conseguir una estructura correcta

Tabla 1: Resumen de las ventajas e inconvenientes de las posibles opciones para obtener un *dataset*.

3.4. Estudio del dataset *Tagged Anime Illustrations*

El *dataset* almacenado en la web *Kaggle.com* no es el *dataset* original del que provienen las imágenes. La fuente primaria de las imágenes y sus meta-datos es la página web *danbooru.donmai.us* que es una web a la que los diferentes usuarios pueden subir ilustraciones y añadirle diferentes etiquetas para facilitar su búsqueda a otros usuarios.

Las ilustraciones tienen todas formato .jpg con dimensiones de 512 píxeles x 512 píxeles. Dichas imágenes vienen clasificadas con diferentes etiquetas que identifican los diferentes elementos presentes en los dibujos. Además de estos datos que son los más interesantes se almacena mucha más información que en un futuro se estudiará para ver si se puede obtener información a partir de ella.

3.4.1. Origen del dataset

Las imágenes de *Tagged Anime Illustrations* fueron obtenidas del dataset denominado *Danbooru2017*¹¹. En dicha página se puede acceder a una breve documentación que explica que los datos almacenados en *Kaggle.com* son un subconjunto de los datasets *Danbooru2017* y *Nagadomi's moeimouto face*. Dichas imágenes están calificadas todas como no sexuales (SFW¹²).

El dataset contiene 3 carpetas:

- ***danbooru-images***: Se encuentran almacenadas las diferentes ilustraciones ordenadas por carpetas donde el nombre de la misma hace referencia a los últimos cuatro dígitos del nombre de las imágenes almacenadas en ella. De esta forma se tienen 151 carpetas que almacenan imágenes con id's cuyos últimos cuatro dígitos van desde el 0000 hasta el 0150.
- ***dangooru-metadata***: Se encuentran los archivos de texto (con extensión .txt) correspondientes a los meta-datos del dataset original. Dichos metadatos tienen formato pseudo-json, más adelante serán tratados para poder operar con ellos correctamente.
- ***moeimouto-faces***: Esta carpeta contiene ilustraciones de caras ordenadas en carpetas correspondientes al personaje al que pertenece dicha cara. Como ya hemos indicado en la sección 3.2.3 no haremos uso de esta carpeta.

3.4.2. Estudio del tratamiento realizado a las ilustraciones

Las ilustraciones originalmente guardadas en la página *danbooru.donmai.us* son tratadas para ser almacenadas en imágenes de 512 píxeles x 512 píxeles.

Para producir esta transformación las imágenes son contraídas hasta que una de sus dos dimensiones ocupe menos de 512 píxeles, una vez son reducidas, se rellenan los píxeles restantes con píxeles en color negro.

En la figura 7 se puede observar cómo la imagen original (izquierda) es tratada de manera que al reducir su tamaño se generan dos bandas negras en las partes superior e inferior debido a que la imagen original tiene mayor tamaño de ancho que de alto. La imagen generada (derecha) pasa a formar parte del dataset.

¹¹ Publicado en la página web www.gwern.net/Danbooru2019

¹² *Suitable for Work*, es una expresión usada en el argot de internet para indicar que cierto contenido tiene carácter sexual



Figura 7: Bandas horizontales generadas al transformar la imagen original.

Por otra parte, puede darse el caso en que una imagen originalmente no ocupe más de 512 píxeles en ninguna de sus dos dimensiones, en este caso la imagen no se extiende, sino que se rellenan los huecos sin imagen con píxeles negros.

En la figura 8 podemos observar un ejemplo de la transformación que sufren las imágenes en las que ninguna de sus dos dimensiones originales supera los 512 píxeles.



Figura 8: Recuadro negro generado al transformar la imagen original.

Estas transformaciones hacen que la mayor parte de las imágenes tengan bandas o recuadros negros alrededor de la propia imagen. En este punto se toma en cuenta los problemas que puedan surgir a partir de la presencia de recuadros y bandas negras. A la hora de clasificar las imágenes se considera que no es importante puesto que, cuando a través de la red de neuronas se analice la imagen, se espera que la inteligencia artificial sea capaz de identificar estos bordes negros como una parte de la ilustración sin valor alguno.

3.4.3. Imágenes poco útiles

A la hora de hacer un primer estudio de las imágenes que componen nuestro conjunto de datos se observan algunas ilustraciones que no podemos utilizar para nuestro cometido. Este tipo de imágenes son:

- **Páginas de cómic:** Corresponden a imágenes de páginas completas de cómic, estas ilustraciones no son del todo idóneas para nuestro entrenamiento ya que están compuestas por varias ilustraciones. En la figura 9 podemos observar una imagen del *dataset* que es un cómic compuesto por varias subilustraciones separadas por viñetas.



Figura 9: Imagen de cómic del dataset.

- **Títulos:** Están compuestas por letras con un fondo, desde el punto de vista artístico no tienen valor como ilustración y por tanto no son útiles para nuestro objetivo. En la figura 10 se puede observar una imagen del *dataset* correspondiente a un título sin ningún valor artístico.



Figura 10: Imagen de un título del dataset.

- **Imágenes inservibles:** Debido a las transformaciones que ya se han indicado en la sección 3.4.2 hay ciertas imágenes que han sufrido una transformación que ha resultado en que sean inservibles. Por ejemplo hay imágenes cuyas proporciones hacen que el tamaño de los cuadros negros sea mucho mayor que la propia imagen resultando en algunos casos que la ilustración original no sea visible, por ejemplo se puede observar que en la figura 11 es prácticamente imposible diferenciar la ilustración original.

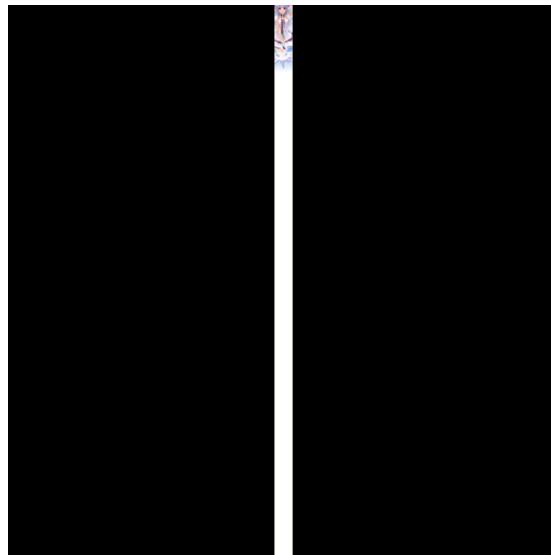


Figura 11: Imagen que tras las transformaciones ha resultado ininteligible.

3.4.4. Estudio de los metadatos

En los distintos archivos de meta-datos se puede encontrar la información relativa a cada una de las imágenes. La mayoría de estos campos son fáciles de identificar sin embargo hay cierta información de las imágenes que se deben estudiar:

- **Notas:** El sistema de notas de la página *danbooru.donmai.us* permite a los usuarios seleccionar ciertas partes de una imagen para realizar comentarios, por ejemplo traducciones. La figura 12 es un ejemplo del uso de notas para traducir un texto en japonés.

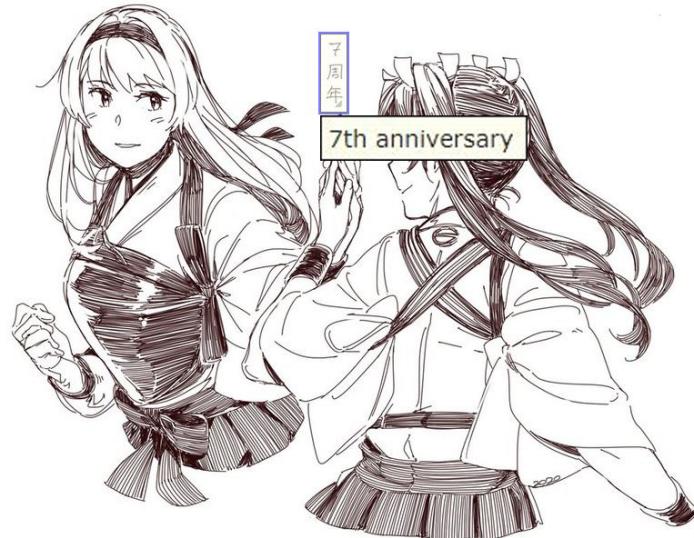


Figura 12: Nota en una imagen subida a la página *danbooru.donmai.us*.

- **Herencia:**¹³ Hay ciertas imágenes relacionadas con otras; esta relación es de imagen-padre e imagen-hija. La imagen padre se elige como la “mejor” opción del grupo de imágenes con parentesco y las diferentes imágenes hijas que puede tener son modificaciones de ella. De esta forma se pueden encontrar variaciones sobre una misma ilustración como se puede ver en la figura 13 formada por tres imágenes con relación de parentesco.



Figura 13: Relación de parentesco entre varias imágenes.

¹³Información oficial de la página web: https://danbooru.donmai.us/wiki_pages/help:post_relationships

- **Pools:**¹⁴. Los *pools* son grupos de imágenes con características comunes (ver figura 14). Las diferentes imágenes que forman un grupo *pool* están ordenadas. Comúnmente se usan los grupos *pools* para agrupar diferentes imágenes correspondientes a un cómic o una serie de dibujos de un artista en concreto, sin embargo no es el único uso que se puede hacer del campo. En la figura 14 podemos observar varias imágenes pertenecientes al *pool Reaction Faces*.



Figura 14: Imágenes del pool Reaction Faces.

- **Aprobación:** La página *danbooru.donmai.us* tiene un sistema de regulación de las imágenes que se suben que se basa en el trabajo de la comunidad. Los usuarios moderadores de la web pueden aprobar o no las imágenes que se van a subir, si una imagen no es aprobada no se muestra en la web a no ser que se acceda a la pestaña de imágenes canceladas. Las imágenes subidas a la web pueden ser marcadas por los usuarios a través de *flags*, de esta manera cuando una imagen es *flaggeada* se envía a la cola de moderación para su posterior evaluación. Los motivos para eliminar una imagen son la violación de las normas de la página web¹⁵ o la baja calidad de la misma¹⁶.

¹⁴Información oficial de la página web: https://danbooru.donmai.us/wiki_pages/help:pools

¹⁵Normas de danbooru.donmai.us: https://danbooru.donmai.us/static/terms_of_service

¹⁶https://danbooru.donmai.us/wiki_pages/howto%3Aflog

- **Estado:** Las diferentes imágenes pasan por diferentes estados a la hora de ser publicadas en la página web. Cuando una imagen se sube pasa al estado pendiente de aprobación (*pending*) hasta que pasa los filtros de aprobación, si la imagen es aceptada obtiene el estatus *activa* (*active*) y si no se aprueba pasa a estado *borrada* (*deleted*). Una vez clasificada la imagen se bloquea su estado. En la figura 15 podemos observar los diferentes estados por los que puede pasar una ilustración desde que esta se sube a la página web.

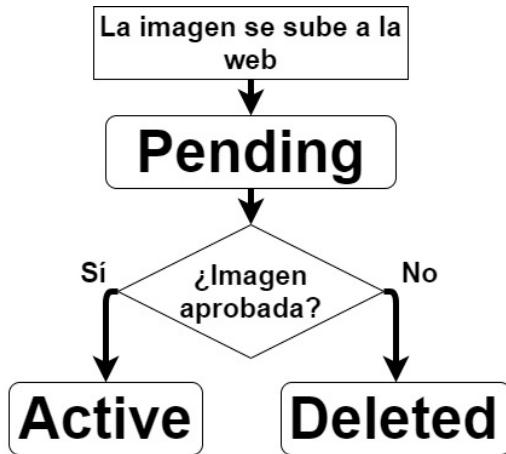


Figura 15: Diagrama de flujo de los diferentes estados por los que puede pasar una imagen en la web.

La tabla 2 describe la información de los diferentes campos de los archivos de meta-datos:

Nombre de tag	Descripción
id	Número que identifica el nombre de la imagen almacenada en el dataset.
created_at	Fecha correspondiente al momento en que la imagen fue subida a danbooru.donmai.us.
uploader_id	Número identificativo del perfil del usuario creador de la imagen.
score	Puntuación de la imagen.
source	Enlace a la fuente original de la imagen si no es danbooru.donmai.us.
md5	Hash md5 de la imagen.
last_commented_at	Fecha del último comentario en la publicación de la imagen.

rating	Clasificación de las imágenes atendiendo al contenido sexual de la misma. <ul style="list-style-type: none">■ s=safe. Imágenes sin ningún contenido sexual.■ q=questionable. Imágenes con contenido sexual no explícito.■ e=explicit. Imágenes con contenido sexual explícito.
image_width	Tamaño horizontal en píxeles de la imagen original antes de aplicarse la transformación descrita en la sección 3.4.2.
image_height	Tamaño vertical en píxeles de la imagen original antes de aplicarse la transformación descrita en la sección 3.4.2.
is_note_locked	Valor booleano correspondiente a si la imagen original tenía bloqueadas las notas.
file_ext	Extensión del formato de la imagen original subida.
last_noted_at	Fecha de la última nota de la imagen.
is_rating_locked	Valor booleano correspondiente a si la imagen original tenía bloqueada la clasificación.
has_children	Valor booleano correspondiente a si la imagen original tenía imágenes hijas relacionadas.
approver_id	Número identificativo del usuario o de los usuarios que han aprobado la publicación de la imagen.
file_size	Tamaño en kilobytes de la imagen original.
is_status_locked	Valor booleano correspondiente a si la imagen original tenía el estado bloqueado.
up_score	Número de votos positivos de la imagen original en la web.
down_score	Número de votos negativos de la imagen original en la web.
is_pending	Valor booleano correspondiente a si el estado de la imagen original era Pending cuando se añadió al dataset.
is_flagged	Valor booleano correspondiente a si la imagen original tenía alguna flag cuando se añadió al dataset.
is_deleted	Valor booleano correspondiente a si el estado de la imagen original era Deleted cuando se añadió al dataset.

updated_at	Fecha de la publicación de la imagen.
is_banned	En la página danbooru.donmai.us no encontramos información sobre este campo, discurremos que es un valor booleano correspondiente a si el artista de la imagen original estaba baneado de la página web. Más tarde pasaremos a estudiar más en profundidad dicho campo.
pixiv_id	Si la imagen original procede de la página pixiv.net ¹⁷ corresponde con su número identificativo.
tags	Etiquetas de la imagen. En este campo se guardan las diferentes características de la ilustración. Más adelante pasaremos a estudiar más en profundidad dicho campo en profundidad puesto que es la base fundamental de estos archivos.
pools	<i>Array</i> con el id de las <i>pools</i> a las que pertenece la imagen.
favs	Lista con los números identificativos de los usuarios que añadieron a su lista de favoritos la imagen original cuando se añadió al <i>dataset</i> .

Tabla 2: Descripción de los campos de un objeto json de una ilustración.

3.4.5. Ventajas y desventajas

Las principales ventajas por las que se escoge el *dataset* de *Danbooru2017* finalmente son:

- La calidad artística de las imágenes es idónea, pese a que los dibujos sean *fan art* la calidad de ellos es muy alta. Como ya se ha indicado, el sistema de aprobación de imágenes de la página web original previene de que se cuelguen imágenes de una baja calidad.
- Los meta-datos de las ilustraciones son muy completos y correctos debido a que los propios usuarios publicadores de las imágenes fueron los encargados de darles las etiquetas que se recogen en el *dataset*.

¹⁷pixiv.net es una página similar a danbooru.donmai.us donde los artistas pueden subir sus obras artísticas

- La dimensión del *dataset* es suficientemente grande como para el uso que se va a hacer de él. El hecho de que cuente con unas 300.000 imágenes ayuda al entrenamiento de la red neuronal que pretendemos crear, dificultando que se produzca *overfitting*¹⁸ y permitiéndonos la clasificación de muchos elementos de los meta-datos por separado, consiguiendo una clasificación más detallada.
- La estructura de los archivos de meta-datos al ser prácticamente *json* podrá ser modificada a nuestro antojo con el fin de realizar un uso intensivo de estos datos, tanto para darles una estructura final adecuada como para profundizar en la información que contiene.

Sin embargo, las desventajas de elegir este dataset son:

- Se encuentran ciertas imágenes que no son útiles para el uso que se hará de ellas. Este tipo de imágenes ya han sido definidas en la sección 3.4.3, más adelante se estudiará el peso de este tipo de imágenes en el conjunto del *dataset*.
- La estructura de los archivos de meta-datos ha de ser modificada para poder ser tratada correctamente. Este proceso no supone un gran problema puesto que la forma general es la de un archivo *json*.
- No todos campos de los meta-datos son útiles para el cometido del trabajo, estos datos suponen mayor carga innecesaria en los archivos y no proporcionan ningún tipo de información válida para el uso que haremos de ellos.

3.4.6. Preprocesado de los meta-datos.

Para poder usar correctamente los meta-datos que tiene el *dataset* primero se debe dar una buena estructura, hacer limpieza de datos innecesarios y realizar un estudio profundo sobre ellos. Este proceso se divide en diferentes iteraciones a través de las cuales se van obteniendo diferentes versiones de los archivos cada vez más óptimas para el propósito que se busca.

Formateo json de los archivos. Los archivos .txt de meta-datos tienen un formato a simple vista parecen *json*, sin embargo a la hora de intentar procesarlos se muestra de que no es así debido a que los diferentes objetos json correspondientes a cada una de las imágenes están concatenados formando una amalgama de objetos json unos detrás de otros.

¹⁸El *overfitting* o sobreajuste es el efecto de sobreentrenar una red de neuronas artificiales provocando que la misma aprenda características demasiado concretas del problema a solucionar impidiéndole a la red generalizar para enfrentarse a problemas no vistos anteriormente.

Para poder hacer lecturas correctas y formar un *array* json correcto basta con envolver cada uno de los elementos separando cada uno por comas y añadiendo corchetes al inicio y final del archivo. Una vez realizado este proceso el resultado final es una estructura json correcta en la que hay un *array* en el que se encuentra la información correspondiente a las diferentes imágenes.

Con este procedimiento se obtienen los archivos que forman el conjunto que se denominará “*MetadataStructured*” el cual tiene una estructura json correcta.

Limpieza de campos 1. El primer paso para aligerar la carga de cada archivo de meta-datos es eliminar la información que se almacena que es innecesaria para nuestro cometido. Para realizar esto primero se decide eliminar los campos que sin un estudio previo consideramos que no nos son útiles bajo ninguna circunstancia.

La tabla 3 recoge los campos eliminados junto a los motivos por los que no son útiles:

Nombre del campo	Motivo de su eliminación
created_at	No es útil para el entrenamiento de una inteligencia artificial.
uploader_id	No es útil para el entrenamiento de una inteligencia artificial.
score	Pese a que se podría estudiar su uso para diferenciar entre imágenes mejores votadas desestimamos finalmente su uso porque esta no es la intencionalidad del proyecto.
source	La fuente de la mayoría de las imágenes no es accesible porque ha sido eliminada dicha url.
md5	No es útil para el entrenamiento de una inteligencia artificial.
last_commented_at	No es útil para el entrenamiento de una inteligencia artificial.
is_note_locked	No es útil para el entrenamiento de una inteligencia artificial.
last_noted_at	No es útil para el entrenamiento de una inteligencia artificial.
is_rating_locked	No es útil para el entrenamiento de una inteligencia artificial.
approver_id	No es útil para el entrenamiento de una inteligencia artificial.
file_size	El tamaño descrito no es el de las imágenes del dataset, sino el de las imágenes originales de la página web por lo tanto no se corresponde con lo almacenado.

is_status_locked	No es útil para el entrenamiento de una inteligencia artificial.
up_score	Igual que el campo score no es útil para el entrenamiento de una inteligencia artificial.
down_score	Igual que el campo score no es útil para el entrenamiento de una inteligencia artificial.
is_pending	El estado de la imagen original no es útil para nuestro proyecto.
is_flagged	El estado de la imagen original no es útil para nuestro proyecto.
is_deleted	El estado de la imagen original no es útil para nuestro proyecto.
updated_at	No es útil para el entrenamiento de una inteligencia artificial.
pixiv_id	Igual que el campo source no es útil para el entrenamiento de una inteligencia artificial.
favs	No es útil para el entrenamiento de una inteligencia artificial.

Tabla 3: Motivos de la eliminación de los campos de los archivos de meta-datos.

Con este procedimiento se obtiene el conjunto de meta-datos que denominaremos “MetadataCleaned” que, al tener menos información almacenada, es más ligero para poder tratarlo.

Limpieza de datos sin imagen. Observando la estructura del *dataset* se ve que las imágenes están ordenadas en carpetas desde 0000 hasta 0150 donde esos cuatro dígitos corresponden a los últimos cuatro dígitos del id de la imagen. Sin embargo se puede observar que en los diferentes archivos de meta-datos se almacena información de imágenes con id superior a 0150.

Se pasa a eliminar la información de todas estas imágenes puesto que almacenar información de imágenes que no tenemos es innútil y sólo genera mayor carga en los archivos.

Con este procedimiento se obtiene el conjunto de meta-datos que denominaremos “MetadataStored”.

Una vez realizado este procedimiento se vuelve a estudiar la información almacenada y se observa que no todas las imágenes que tienen información en los meta-datos están almacenadas pese a tener un id en el rango 0000-0150. De la misma manera que se ha hecho anteriormente se pasan a eliminar dichas imágenes.

Con este procedimiento se obtiene el conjunto de meta-datos que denominaremos “MetadataImg” el cual contiene datos exclusivamente de imágenes que se almacenan en el dataset.

Estudio en profundidad de los campos. Para poder seguir eliminando campos se necesita realizar un estudio más profundo de la información que almacenan algunos de ellos. De esta forma se obtendrá más información de los diferentes campos que pueden ser eliminados para poder así tomar una decisión justificada.

La información obtenida de cada campo es la siguiente:

- **rating:** En los archivos de meta-datos se encontraban campos con imágenes cuestionables (q) y explícitas (e) sin embargo ninguna de esas imágenes estaba finalmente almacenada en el *dataset*. Por lo tanto todas las imágenes almacenadas en el *dataset* tienen como valor de *rating* seguras (s).
- **file_ext:** Se encuentran como extensiones .jpg, .png y .gif sin embargo todas las imágenes del *dataset* tienen extensión .jpg. Se observa que las imágenes cuyos datos indican que sus extensiones son .png y .gif sí están almacenadas en el dataset sin embargo su extensión en él es .jpg.
- **parent_id:** Se encuentran dentro del *dataset* imágenes con relaciones de parentesco, en algunos casos toda la familia de imágenes está almacenada, en otros sólo algunas de ellas.
- **is_banned:** Se estudia si hay algún tipo de correlación entre si una imagen tiene este campo a *True* con el *rating* de la misma pero no encontramos ninguna relación. Hay imágenes seguras (s) con este campo a *True*. Finalmente se llega a la conclusión de que la información de este campo es, como se había discutido en un inicio, el estado de restricción o no del usuario que subió la imagen original y no de la imagen en sí misma.
- **pools:** Igual que con los grupos de parentescos, los grupos de *pool* contienen imágenes que no siempre están almacenadas en el *dataset*.

Limpieza de campos 2. Una vez realizado el estudio de los campos se puede pasar a tomar la decisión de qué campos de los estudiados se pueden eliminar y cuáles se mantienen para un posible futuro uso.

La tabla 4 recoge la decisión tomada sobre los campos estudiados junto al motivo de la misma:

Nombre de tag	Decisión	Justificación
rating	Eliminar	Para todas las imágenes almacenadas es s.
file_ext	Eliminar	No concuerda para todos los casos puesto que todas las imágenes tienen formato .jpg en el <i>dataset</i> .
parent_id	Mantener	Puede ser útil para algún uso futuro.
is_banned	Eliminar	No es útil para el entrenamiento de una inteligencia artificial.
pools	Mantener	Puede ser útil para algún uso futuro.

Tabla 4: Decisiones tomadas sobre los campos de meta-datos tras el estudio realizado.

Con este procedimiento se obtiene el conjunto de meta-datos que se denominará “MetadataImgCleaned” el cual forma el conjunto de meta-datos completamente limpio del cual haremos uso para nuestro entrenamiento.

3.4.7. Resumen de la limpieza de meta-datos

La figura 16 contiene un esquema del preprocesado de los meta-datos realizado:

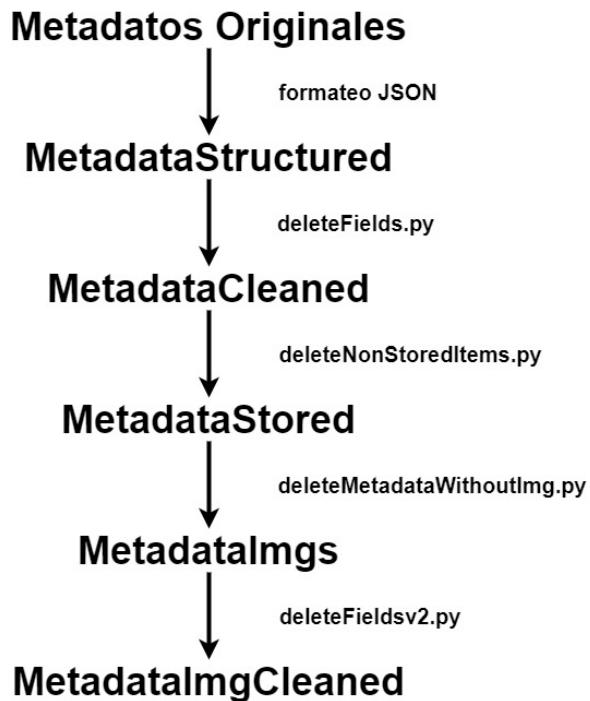


Figura 16: Proceso de limpieza de meta-datos.

Se puede observar en la gráfica como los meta-datos han ido sufriendo modificaciones hasta obtener el resultado buscado. Cada una de las iteraciones de limpieza de datos es útil para un cometido, la elección de realizar el proceso de manera iterativa es una decisión estudiada pues de esta manera en un futuro cabe la posibilidad de realizar cambios diferentes a los meta-datos para buscar un objetivo distinto.

En la figura 17 se puede observar cómo se reduce¹⁹ el tamaño de cada archivo de meta-datos con las diferentes iteraciones del preprocesado:

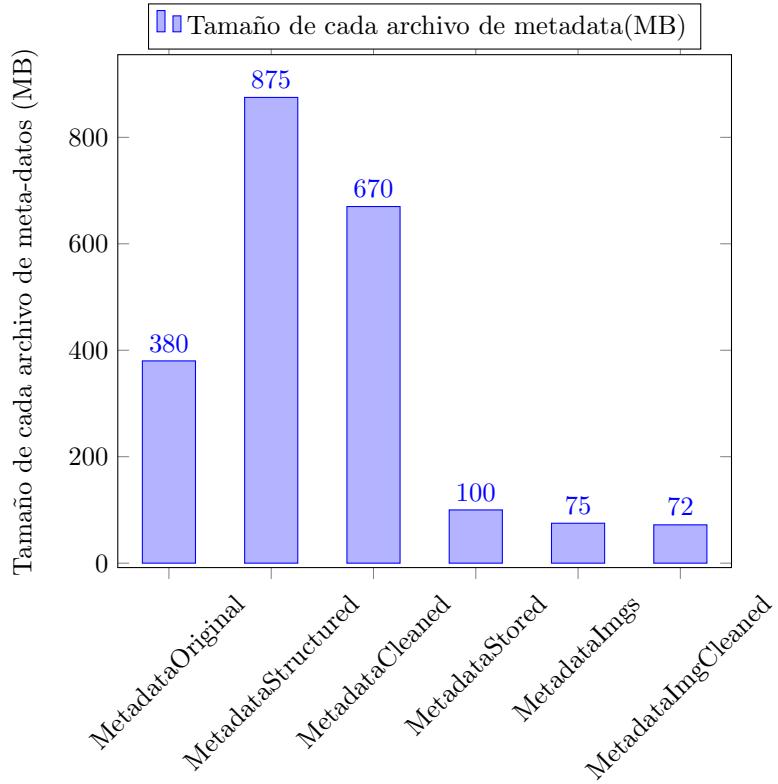


Figura 17: Reducción del tamaño de los archivos a medida que se producen las transformaciones en los mismos.

Como se observa en la figura el tamaño de los archivos a disminuido drásticamente consiguiendo de esta manera una mayor sencillez y optimización al leerlos. Como el proceso de lectura se realizará de manera masiva a la hora de realizar los entrenos, cada pequeña optimización en los meta-datos significa una gran diferencia a la hora de procesarlos.

¹⁹Se puede observar un aumento considerable de tamaño en los archivos después de darles formato *json*, esto se debe principalmente a que los archivos originales tenían un objeto *json* por fila lo cual dificultaba enormemente la comprensión del archivo. Al darle un formato legible cada objeto pasa a ocupar más filas ocupando más espacio físico al mismo tiempo.

3.5. Limpieza de imágenes del dataset

Una vez realizado el preprocesado de los meta-datos se pasa a intentar realizar un proceso similar con las propias imágenes del *dataset*. Para ello se han identificado las imágenes de las que se pueden prescindir en el apartado 3.4.3.

Se pasa a estudiar cada caso por separado:

- **Páginas de cómic:** Estas imágenes se pueden identificar a través de la información que almacenan en el campo *pools* de meta-datos. Sin embargo, al observar estas imágenes se encuentran numerosos casos que, pese a formar parte de una página de cómic, son útiles para el cometido pues contienen ilustraciones completas y comprensibles para una inteligencia artificial como se puede observar en la figura 18. Además los archivos que pertenecen a un *pool* forman parte de un 14,8 % del *dataset* y eliminarlas sería una pérdida de gran peso. Por ello decidimos que incluir este tipo de imágenes es beneficioso para el resultado final.



Figura 18: Imagen de cómic útil para el entrenamiento.

- **Títulos:** Este tipo de imágenes son muy difíciles de filtrar puesto que no se identifican directamente con ninguno de los campos de meta-datos previamente estudiados. Incluso pudiendo identificar los títulos se estarían perdiendo imágenes útiles para el entrenamiento ya que hay ciertos títulos que aparecen acompañados de una ilustración válida para el entrenamiento como se puede observar en la figura 19.



Figura 19: Imagen de título útil para el entrenamiento.

- **Imágenes inservibles:** Estas imágenes se caracterizan por tener originalmente unas dimensiones que han provocado que, tras el procesado para formar el *dataset* original, han quedado inservibles. Para estudiar el peso de estas imágenes se hace un recuento de las imágenes con dimensiones dispares.

El resultado de este estudio puede ser observado en la tabla 5 que recoge el numero de imágenes respecto a la relación alto/ancho o viceversa:

Relación alto/ancho	Número de imágenes	Porcentaje sobre el total
4	1811	0,53 %
3	5367	1,59 %

Tabla 5: Resultados de la contabilización de imágenes de proporciones desiguales.

Además se encuentran casos de ilustraciones con relaciones de alto/ancho elevadas que suponen ilustraciones perfectamente válidas para el uso que se harán de ellas como podemos observar en la figura 20.



Figura 20: Imagen de proporciones descompensadas útil para el entrenamiento.

Después del estudio se decide que no se pueden filtrar correctamente las imágenes inservibles con este procedimiento, también se observa que el peso que puedan tener este tipo de imágenes en el conjunto del *dataset* es mucho menor del esperado en un principio.

Tras realizar el estudio sobre cada uno de los tipos de imágenes que en un principio se habían considerado poco útiles se puede concluir que no merece la pena eliminarlas. Lo que se lograría eliminando las imágenes de estas categorías no se compensa con la información que se perdería con el mismo proceso. Por otra parte se considera incluso útil el hecho de contar con este tipo de imágenes pues algunas de ellas suponen ruido para el entrenamiento de la red de neuronas, lo cual puede mejorar el desempeño a la hora de entrenar la misma.

3.6. Estudio de *tags*

Previamente al entrenamiento de la inteligencia artificial se necesita realizar un estudio sobre la información de los elementos visuales de las imágenes. Para ello se debe conocer en profundidad la información que se almacena en el campo *tags* de los meta-datos.

3.6.1. Estructura del objeto *tags*

Primero se debe conocer la estructura de las etiquetas, para ello se estudian los diferentes elementos que conforman una tag:

- **id**: Número identificativo de la etiqueta.
- **name**: Nombre de la etiqueta.
- **category**: Número entre 0 y 5 que define el tipo de tag que es. Se estudia la correspondencia entre el número de categoría y el tipo de imagen:
 - **0 (General)**: Identifica elementos de la imagen como pueden ser complementos de los personajes, la ropa que llevan, expresiones faciales, etc. Se puede decir que las etiquetas con esta categoría son las más valiosas para nuestro cometido pues suponen la descripción de una imagen por sus elementos visuales.
 - **1 (Artista)**: Creador de la ilustración.
 - **2**: No hay ninguna imagen en nuestro dataset con alguna tag con esta etiqueta.
 - **3 (Copyright)**: Manga, serie o anime al que pertenece la ilustración.
 - **4 (Personaje)**: Nombre de un personaje que se aparece en la ilustración.
 - **5 (Meta-information)**: Meta-information de la imagen, tags con esta categoría pueden ser *highres* (alta resolución), *official_art* (arte oficial) o *translation_request* (petición de traducción).

3.6.2. Recuento de *tags*

Para el uso de las tags en los entrenos primero se debe filtrar qué *tags* no son útiles para el trabajo. Debido a que para el entreno de una red neuronal hacen falta gran cantidad de imágenes, el primer filtrado de etiquetas lo hacemos por el número de imágenes con esa *tag* presente.

Podemos observar la relación de número de imágenes y etiquetas en la figura 21:

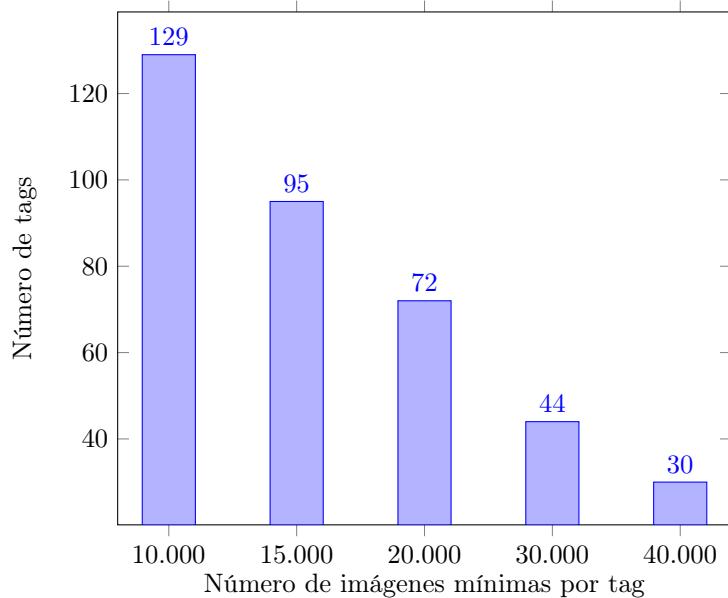


Figura 21: Resultado de la contabilización del número de *tags* con muchas imágenes.

Tras el recuento se considera que para una primera aproximación se tomarán en cuenta las tags con más de 10.000 imágenes y más adelante cuando se realice el entreno se podrá ajustar el conjunto añadiendo etiquetas con mayor número de imágenes si fuese necesario.

3.6.3. Descripción de *tags*

Para poder usar las diferentes etiquetas en un futuro primero se debe comprender en profundidad la diferente información que guardan. Para ello se decide, a partir de la lista de *tags* que cuentan con más de 10.000 imágenes, eliminar las *tags* que resulten innecesarias para nuestro uso. De esta forma se consigue reducir el tamaño de la lista para poder tratar con ella más fácilmente.

A la hora de eliminar *tags*, excepto para casos más específicos, se tendrá como criterio general eliminar aquellas que representen información que no represente a la imagen a la que pertenecen en concreto ya que se pretende hacer una clasificación por los elementos en concreto de la imagen y no su meta-information.

En general se eliminarán las etiquetas asociadas a meta-information puesto que no describen ningún elemento de la ilustración. Por otra parte se encuentran *tags* que corresponden a personajes, series de mangas etc. las cuales se deciden eliminar pues no se quiere realizar una clasificación por pertenencia a una serie ya que se considera que complicaría mucho el entrenamiento de la inteligencia artificial al ser categorías muy generales.

La tabla 6 recoge los motivos por los que decidimos eliminar de la lista las diferentes *tags*:

Nombre de tag	Descripción	Motivo de su eliminación
highres	Imagen en alta definición.	Representa meta-information de la imagen.
Touhou	Nombre de saga de videojuegos.	Representa que la imagen pertenece a una saga.
looking_at_viewer	Un personaje se encuentra mirando al lector	es demasiado general, prácticamente cualquier dibujo con los ojos mirando al frente podría entrar en esta categoría.
bad_id	El id original no era adecuado.	Representa meta-information de la imagen.
bad_pixiv_id	El id original de la web de <i>pixiv.net</i> .	Representa meta-information de la imagen.
translated	Imagen traducida.	Representa meta-information de la imagen.
original	Ilustración original.	Representa meta-information de la imagen.
kantai_collection	Nombre de juego de cartas.	Representa que la imagen pertenece a un juego, no algo concreto de la imagen.
male_focus	Imagen centrada en un hombre.	Podría ser utilizada para reconocer que la imagen tiene personajes masculinos, sin embargo para este cometido hay tags específicas como <i>1boy</i> .

translation_request	Imagen con petición de traducción.	Representa meta-information de la imagen.
commentary_request	Imagen con petición de comentario.	Representa meta-information de la imagen.
official_art	Arte oficial de la serie que representa.	Representa meta-information de la imagen.
fate_series	Nombre de una saga.	Representa que la imagen pertenece a una saga.
vocaloid	Cantante virtual. ²⁰	Representa el ámbito al que pertenece el personaje lo cual no es descriptivo de la ilustración.
fang	Presencia de colmillos.	La manera de representar colmillos en el estilo de cómic japonés son colmillos muy pequeños. Consideramos muy difícil para una red de neuronas identificar elementos tan pequeños.
idolmaster	Nombre de una saga.	Representa meta-information de la imagen.
absurdres	Imagen en alta definición (más que highres).	Representa meta-information de la imagen.
alternate_costume	Personaje vestido con ropa diferente a la que habitualmente lleva puesta.	Los trajes alternativos pueden ser de cualquier tipo y por tanto no representan ningún tipo de traje concreto.
comic	Imagen perteneciente a comic	Representa meta-information de la imagen.
monochrome	Imagen monocromática.	Representa meta-information de la imagen.
greyscale	Imagen en escala de grises.	Representa meta-information de la imagen.

²⁰ *Vocaloid* es un programa de síntesis de voz con el cual se desarrollan personajes ficticios que producen canciones

chibi	Estilo de dibujo caracterizado por cuerpos pequeños y cabezas grandes.	Es un tipo de dibujo muy general, no representa ninguna característica concreta fácilmente identificable.
stripped	Ropa con rayas	Elemento demasiado concreto de una imagen para nuestro caso.

Tabla 6: Descripción de los motivos por los que se deciden eliminar ciertas etiquetas de nuestra lista.

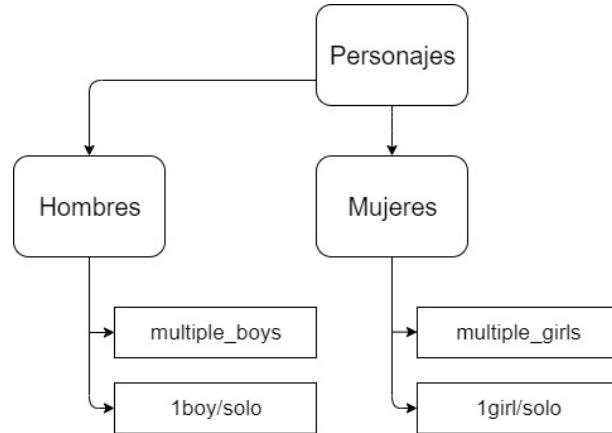
Se vuelve a revisar la nueva lista de etiquetas y se observa que hay algunas de ellas que se pueden agrupar, bien porque describen el mismo elemento o bien porque las diferencias entre sí son mínimas. Se pasa a agrupar ciertas *tags* para simplificar el conjunto, de esta forma se evita la repetición de etiquetas innecesarias. Los grupos que formamos están recogido en la tabla 7:

Nombre de las tags del grupo	Descripción
solo, 1girl	Un único personaje femenino en la escena.
2girls, multiple_girls, 3girls	Varios personajes femeninos en la escena.
open_mouth, :d	Personaje con la boca abierta.
navel, midriff	Personaje enseñando el vientre.
school_uniform, seraifuku	Personaje con uniforme escolar.
2boys, multiple_boys	Varios personajes masculinos en la escena.
swimsuit, bikini	Personaje con traje de baño.
pleated_skirt, skirt	Personaje con falda.
gloves, elvow_gloves	Personaje con guantes.
pantyhose, panties	Personaje con panties.
ribbon, hair_ribbon	Personaje con cinta.
hair_ornament, hairclip	Personaje con accesorios en el pelo.

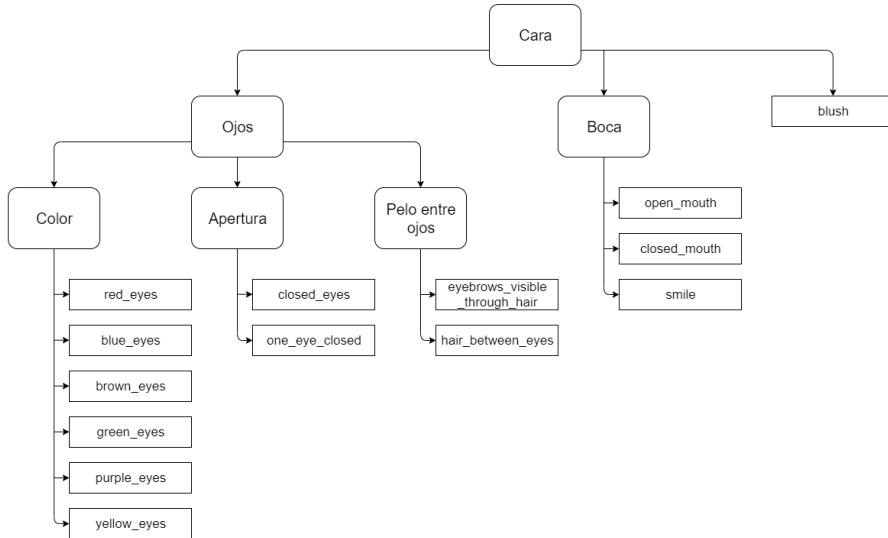
Tabla 7: Descripción de los motivos por los que se deciden agrupar ciertas etiquetas de nuestra lista.

Esquema de tags. Para poder estudiar con mayor facilidad las etiquetas con las que contamos se decide realizar un esquema en el que se desglosen las *tags* de nuestra lista. De esta forma se tendrá una forma visual de reconocer cómo se pueden agrupar las etiquetas y en última instancia se tendrá toda la información recogida de una manera fácil de comprender.

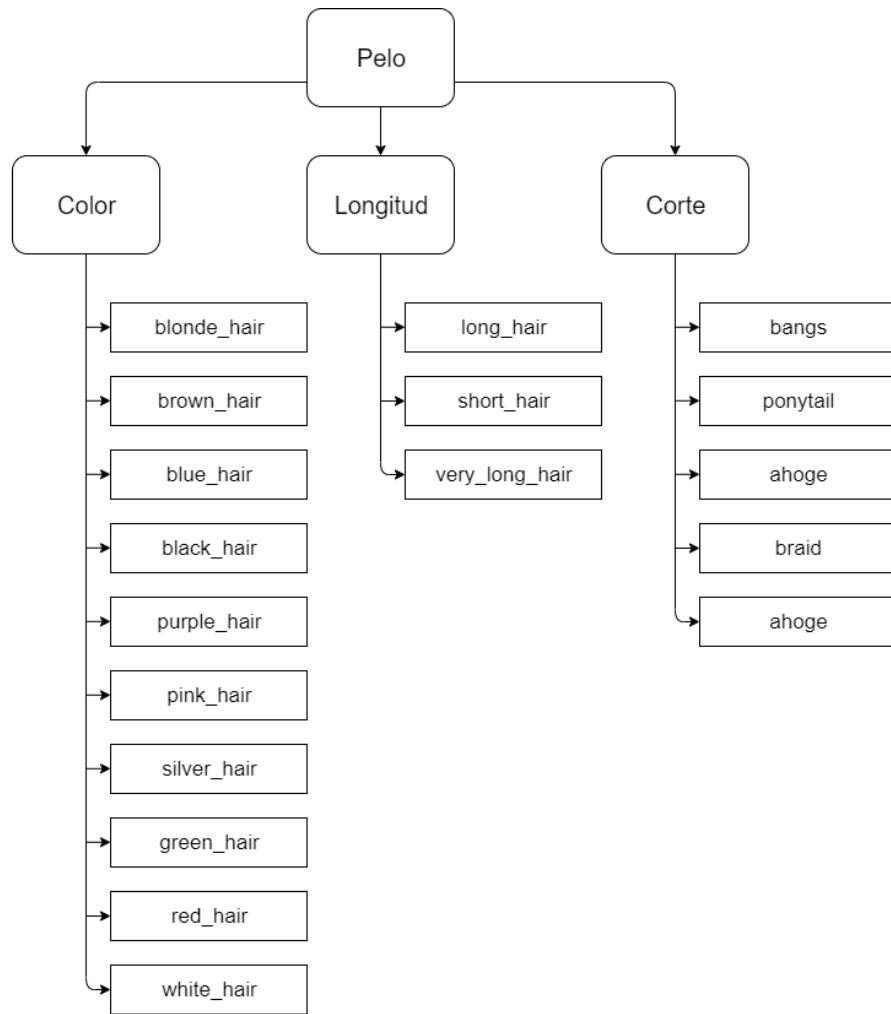
El esquema de la figura 22 recoge la clasificación realizada:



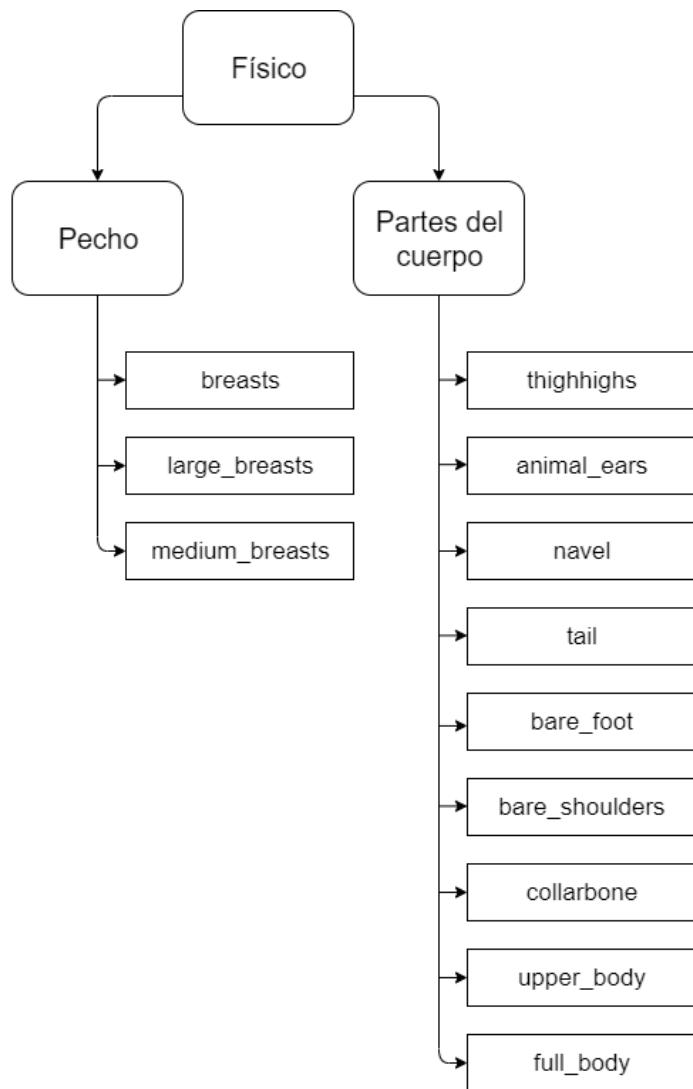
(a) Esquema de las etiquetas referentes a los personajes.



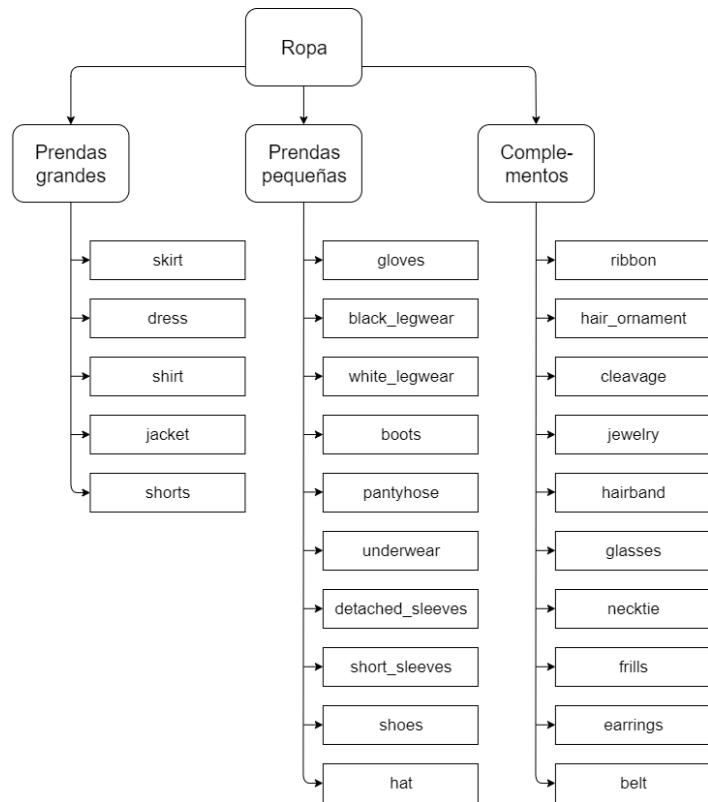
(b) Esquema de las etiquetas referentes a la cara de los personajes.



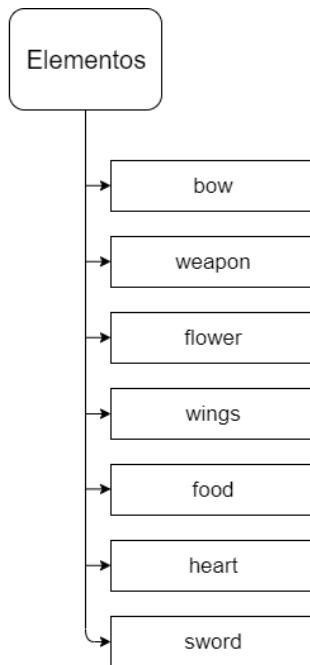
(c) Esquema de las etiquetas referentes al pelo de los personajes.



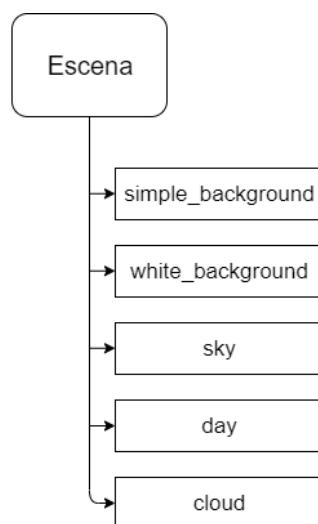
(d) Esquema de las etiquetas referentes al físico de los personajes.



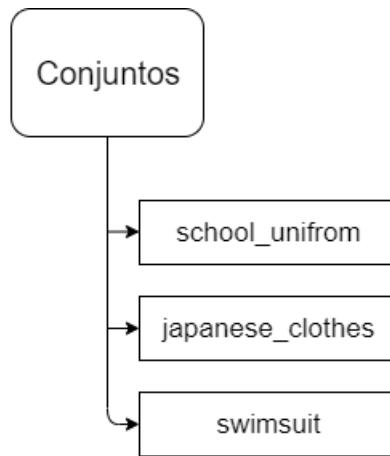
(e) Esquema de las etiquetas referentes a la ropa de los personajes.



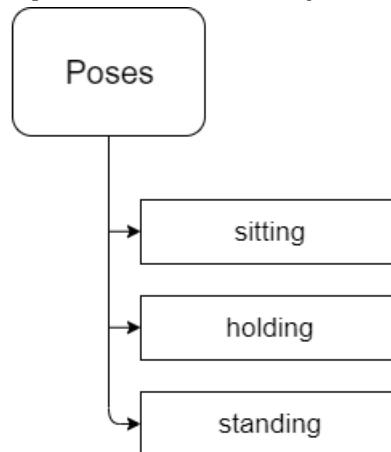
(f) Esquema de las etiquetas referentes a los elementos presentes en la escena.



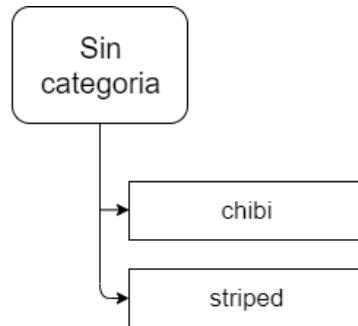
(g) Esquema de las etiquetas referentes a la forma del escenario de la ilustración.



(h) Esquema de las etiquetas referentes a los conjuntos de ropa de los personajes.



(i) Esquema de las etiquetas referentes a la pose que tienen los personajes.



(j) Esquema de las etiquetas referentes a las etiquetas chibi y striped que no se identifican en otro conjunto.

Figura 22: Agrupaciones de las etiquetas del *dataset*.

3.7. Descripción de las redes neuronales clasificadoras

En esta sección se describirá en profundidad el objetivo que se busca con la realización de las redes neuronales que nos permitan clasificar las imágenes atendiendo a las etiquetas estudiadas anteriormente.

3.7.1. Objetivo de las redes neuronales

Las redes neuronales que diseñemos recibirán como entrada una de las imágenes del *dataset* estudiadas. Estas imágenes serán procesadas por las redes para realizar una predicción en la que se obtengan sus meta-datos. Como se ha definido anteriormente las imágenes de entrada serán las que forman parte del *dataset*, y los meta-datos de salida serán las etiquetas definidas anteriormente.

3.7.2. Salida de la red

A través de la aplicación de las redes se quiere hacer una clasificación de n elementos de una imagen donde n es el número de etiquetas que se pretende diferenciar en la imagen. Esta clasificación puede ser múltiple pues en una misma imagen pueden estar presentes más de una etiqueta. Esto presenta un problema a la hora de realizar el entrenamiento ya que es mucho más complicado obtener varias salidas para una misma clasificación.

Ante esta dificultad a la hora de obtener la salida de la red se plantean dos posibilidades, obtener una única etiqueta como salida de la red u obtener varias clasificaciones para una misma imagen. En el caso de obtener varias etiquetas las redes neuronales deberían identificar todas las etiquetas de una imagen, si sólo se pretende obtener una salida por cada imagen la salida buscada debería ser la etiqueta más significativa de la ilustración. Más adelante estudiaremos mecanismos para elegir la etiqueta más significativa.

Podemos ver un ejemplo de los distintos tipos de salidas en la figura 23.



Única salida

- Uniforme escolar

Varias salidas

- Una chica
- Pelo corto
- Uniforme escolar
- Lazo

Figura 23: Ejemplo de la salida buscada de las redes neuronales, con una única clasificación o con varias clasificaciones.

Ahora se pasa a estudiar las implicaciones de ambas posibilidades para saber cual es mas conveniente para la situación del trabajo.

Una única salida en las redes. En la última capa de las redes se pondrá una neurona de salida por cada una de las etiquetas que se quieren clasificar, luego se aplicará una función *softmax* sobre dicha salida, con esta *softmax* se escogerá la salida de la neurona con más activación y cambiará su valor a 1 (presente en la imagen). La salida del resto de neuronas se quedará a 0 (no presentes en la imagen). Gracias a este mecanismo se consigue obtener como salida de las redes la característica más significativa de la imagen descartando las demás.

A la hora de realizar el entrenamiento con este mecanismo surge como problema la elección de las *tags* de cada imagen. Como se obtiene una única etiqueta de salida de la red la imagen original sólo puede tener una única etiqueta. Si dejásemos la imagen original con todas sus etiquetas podría darse el caso en que se hace una predicción de una imagen en la que se acierta en la predicción realizada, sin embargo se producen varios errores por las predicciones que no se han hecho pero sí estaban presentes en la imagen de entrada.

Para solventar este problema se plantea la posibilidad de simplemente juzgar si la etiqueta de salida es correcta, y en ese caso valorar la salida como buena, o si la salida de la red no corresponde con ninguna etiqueta de las que tiene la imagen, en ese caso valorar la salida como mala. Esta solución tiene como problema que si sólo se juzga si la salida está bien o mal y puede darse el caso en el que las redes aprendan a sacar siempre una de las salidas más populares²¹, obteniendo la mayor parte de las veces un resultado válido y haciendo que la red caiga en un mínimo local²² en el que siempre identificaría las etiquetas más comunes.

Finalmente se llega a la conclusión de que la mejor solución para obtener una única salida es que las imágenes de entrada tengan una única etiqueta presente, la más significativa de la ilustración. De esta forma al realizar la predicción con las redes de neuronas se buscará el elemento más característico de la imagen, para ello se debe definir qué se entiende como la característica más significativa de una imagen.

Definimos como la característica más significativa de una imagen como la etiqueta presente en ella menos común respecto al resto de etiquetas de la clasificación. Esta definición creemos que es la más idónea porque atendiendo a la composición de las imágenes de nuestro *dataset* se puede observar que hay elementos que se repiten constantemente. En esos casos lo mejor es identificar la imagen por otros elementos menos comunes. Por ejemplo en la figura 24 podemos observar una ilustración en la que el elemento más significativo de ella es el lazo, pues que la chica presente en ella tenga los ojos azules es un elemento prescindible y si cambiase el color de sus ojos tendría mucho menos impacto en la composición que si el lazo desapareciese.

²¹ Debido a la distribución de las imágenes del *dataset* hay etiquetas que aparecen con mucha más frecuencia que otras, por ejemplo 1girl supone un 60 % de las imágenes totales.

²² Un mínimo local es un punto del entrenamiento donde el error cometido no puede ser menor a través de ningún cambio inmediato ya que ninguna de las derivadas de la función de error en ese punto son negativas, sin embargo no supone el mínimo error absoluto de la función.

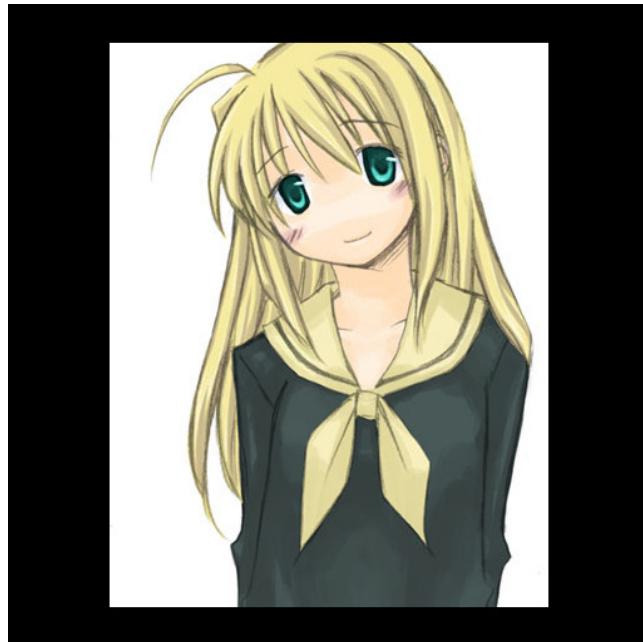


Figura 24: Imagen de una chica con un lazo, en la que el lazo es el elemento principal de la ilustración.

Varias salidas en las redes. Para obtener esta salida sería necesario modificar la estructura general de las redes neuronales. Este modelo se basa en que después de obtener la salida de la última capa de las redes escoger las m etiquetas con mayor valor, y poner sus valores a 1 y el resto de etiquetas a 0. Esta salida sería idónea para nuestro entrenamiento sin embargo elegir el número m de etiquetas supone una gran complicación pues no es siempre el mismo, puede haber imágenes con 5 etiquetas mientras que otras tengan 10. Para solventar este problema la única solución que se considera es aplicar otras redes para que como salida se obtenga el número de etiquetas m . En la figura 25 se puede observar un ejemplo de este tipo de arquitectura, en el se puede ver que aunque la solución sea válida supone la creación de una segunda red lo que supone añadir gran complejidad a nuestro modelo.

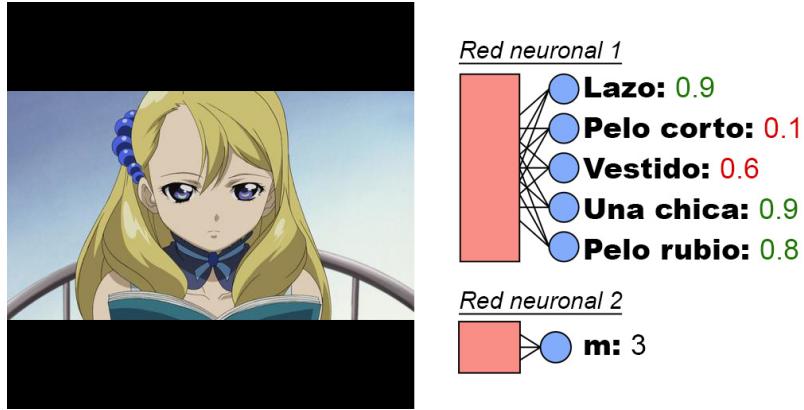


Figura 25: Ejemplo de combinación de redes para elegir el número de elementos presentes en una imagen.

Una vez estudiados ambos tipos de salidas se decide repasar las ventajas e inconvenientes que supone la aplicación de cada modelo. La tabla 8 recoge el resultado de este estudio:

Tipo de salida de tag	Ventajas	Inconvenientes
Única salida	<ul style="list-style-type: none"> ■ Simplicidad del modelo. ■ Salida sencilla. 	<ul style="list-style-type: none"> ■ Necesidad de preprocesar los meta-datos para obtener una única tag. ■ Salida poco explicativa.
Varias salidas	<ul style="list-style-type: none"> ■ Salida natural de la red. ■ Archivos de meta-datos válidos sin preprocessar. 	<ul style="list-style-type: none"> ■ Complejidad del modelo. ■ Necesidad de implementar una segunda red.

Tabla 8: Descripción de los motivos por los que se deciden eliminar ciertas etiquetas de nuestra lista.

Finalmente se decide escoger la opción de una única salida por imagen. Con esta opción se gana cierta simplicidad en la red permitiendo centrar los esfuerzos en optimizar la estructura de la red para obtener los mejores resultados posibles. Más adelante se estudiará cómo realizar el preprocesado de las imágenes para equilibrar la carga del entreno y obtener una única etiqueta por imagen.

3.8. Procesado de las etiquetas para el entreno

Para usar las etiquetas de los archivos de meta-datos primero se debe tener un formato más sencillo para simplificar la carga de imágenes de cara al entreno de las redes. Para ello se decide procesar el campo *tags* y formar con su información un nuevo campo llamado *tagsArray*.

El campo *tagsArray* guardará un *array* de números cuyos valores pueden ser 1 o 0, cada posición corresponderá con la presencia (valor a 1) o no presencia (valor a 0) de la etiqueta correspondiente a dicha posición en la imagen que contiene el *array*. Una vez creado dicho campo para saber si una imagen tiene presente la etiqueta de la posición *x* basta con leer el valor de *tagArray[x]*.

3.9. Ahorro de recursos en el entreno

Las características del ordenador con el que se realizarán los entrenos son las siguientes:

- Procesador: AMD Ryzen 5 3500U con Radeon Vega Mobile Gfx, 2.10GHz.
- Memoria RAM: 8GB.
- Sistema operativo: Windows 10 Home 64 bits.
- Disco duro: 256GB disco SSD.

Como se puede observar el equipo tiene recursos limitados, caben resaltar los 4GB de almacenamiento de la memoria principal, esta memoria será el lugar donde se almacenen las imágenes a la hora de cargarlas para el entreno. Debido a que la información de las imágenes ocupa mucho espacio debemos idear mecanismos para optimizar al máximo el entreno.

Para ello decidimos que a la hora de cargar las imágenes estas se carguen de *batch* en *batch*, de modo que un *batch* de imágenes se cargue, se entrene con dicho *batch* y sea sustituido por el siguiente. De esta forma se conseguirá optimizar el uso de recursos de nuestro modelo ya que cada vez que se cambia de *batch* se libera el espacio que ocupaba este y se reemplaza por el siguiente.

Por una parte este modelo de carga de imágenes conlleva que los diferentes conjuntos de imágenes tengan que ser continuamente leídos del disco y sobreescritos continuamente generando mucho tráfico de entrada/salida en la memoria principal. Este problema se toma en cuenta, pero sin embargo se considera que el beneficio de liberar la mayor parte de la memoria posible es más importante que la ralentización que pueda suceder por el tráfico en la memoria principal.

Para normalizar los datos de las imágenes se debe tratar la información de los píxeles que forman la imagen del entrenamiento, pasando su rango de valores de $[0, 255]$ a $[0, 1]$ ya que a la hora de entrenar modelos es más sencillo tratar con valores en el rango de 0 a 1. A la hora de realizar este proceso en los diferentes entrenamientos decidimos utilizar la librería *numpy*, pues con ella las operaciones se realizarán de manera matricial consiguiendo mayor velocidad en la operación, para ello utilizaremos la función *true_divide*.

3.10. Entreno con una red de neuronas densa

Para realizar una primera aproximación a al objetivo decidimos crear una red de neuronas densa. Debido a su sencillez se pueden controlar los posibles errores que se puedan suceder. El objetivo principal con este modelo no es obtener un buen resultado si no un resultado lo suficientemente bueno como para usarlo de base en las futuras arquitecturas que puedan surgir.

Con esta red se quiere hacer una clasificación de 3 etiquetas:

- **1girl y short _ hair:** Este grupo de etiquetas corresponden a las ilustraciones que cuentan con un único personaje femenino con el pelo corto.
- **1girl y long _ hair:** Este grupo de etiquetas corresponden a las ilustraciones que cuentan con un único personaje femenino con el pelo largo.
- **hat:** Ilustración con un sombrero en ella.

Para diseñar la arquitectura de la red se decide que cada neurona de la capa de entrada le corresponda cada uno de los píxeles de la imagen. Luego se sucederán 6 capas ocultas a través de las cuales se irá reduciendo el número de neuronas de cada capa para ir acercándose más a la salida. Finalmente la capa de salida estará compuesta por 3 neuronas, una por cada una de las clasificaciones que se buscan. La capa de entrada y las capas ocultas tendrán como función de activación una *Relu*²³ y a la capa de salida se le aplicará una función *softmax* que elija la salida única.

²³La función de activación *relu* es una función lineal en la salida es directamente la entrada para resultados positivos y cero para resultados negativos.

3.10.1. Estructura de la red

La figura 26 representa un esquema de la estructura de la red creada:

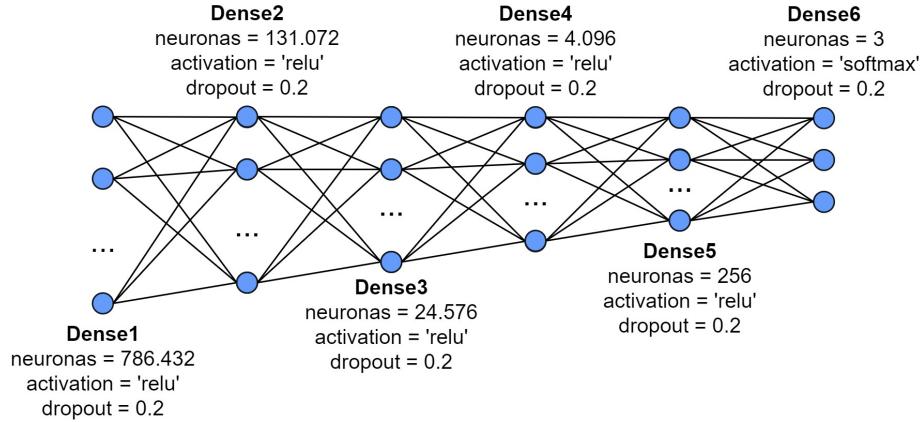


Figura 26: Esquema de la estructura de la red de neuronas densa.

3.10.2. Estructura del entreno

Para hacer este entreno se decide entrenar a la red con un tamaño de *batch* de 1 imagen, este tamaño es el mínimo posible para intentar encontrar resultados y más adelante, si fuese posible, aumentar el tamaño del *batch*.

La elección de los *batches* será aleatoria en cada entreno, escogiendo un *batch* de todos los posibles previamente definidos en la carga de la información de las imágenes. Una vez elegido el *batch* se cargará en memoria la imagen del mismo y se realizará una iteración del entreno con el mismo.

3.10.3. Resultados del entrenamiento

A la hora de hacer el entreno encontramos que con la arquitectura actual es imposible la realización del mismo. Debido a los escasos recursos del equipo que tenemos al cargar la red densa y el primer *batch* de imágenes el programa termina debido a que la memoria no puede almacenar tanta información.

Las causas de que esto suceda no son solo las limitaciones en el tamaño de la memoria, a esto se le suman el gran tamaño que ocupa la red de neuronas y el tamaño de las imágenes. Por una parte el tamaño de las imágenes podría verse reducido disminuyendo su resolución o combinando sus canales para pasar las imágenes a blanco y negro como más adelante estudiaremos. Esta reducción supondría también una disminución en el tamaño de la red pues el tamaño de las imágenes esta directamente relacionado con el número de neuronas necesarias para tratar una imagen.

3.11 Transformación de las imágenes a blanco y negro 3 METODOLOGÍA

En este punto se decide desestimar estos posibles cambios, si bien se podría llegar a entrenar con un modelo como el actual pero reducido, sus resultados no serían lo suficientemente buenos como para dar una visión rápida de los posibles resultados de este modelo. Como el objetivo de esta aproximación era obtener resultados rápidos y sin mucha complicación se decide abandonar ese proyecto pues los cambios necesarios para que funcione son demasiado costosos como para que tenga sentido hacerlos.

3.11. Transformación de las imágenes a blanco y negro

Como se ha observado en nuestro entreno fallido las imágenes del *dataset* ocupan demasiado espacio, lo cual colapsa la memoria principal del equipo impiendiendo el entreno. Para evitar esto como primera medida se decide reducir la dimensionalidad de las imágenes, combinando los canales de color en uno único formando imágenes en blanco y negro. Con ello se conseguirá reducir las dimensiones de cada una de las imágenes de (512, 512, 3) a (512, 512, 1).

Después de hacer este proceso se decide observar los cambios producidos en las imágenes, pues al pasarlas a blanco y negro pueden haber casos en los que se hayan producido pérdidas de información al perder los colores. Un caso concreto puede ser la figura 27 que originalmente contaba en sus etiquetas la de blue_hair pero tras haber pasado a blanco y negro el color de su pelo ha quedado como un tono grisáceo, debido a esto no se podría adivinar su color original.



Figura 27: Imagen de una chica con el pelo azul antes y después de ser transformada a blanco y negro. Se puede observar que el color de su pelo no puede ser extraído de la imagen en blanco y negro.

Como hemos podido observar, las imágenes en blanco y negro no podrán ser clasificadas por el color de los elementos de la imagen. A partir de ahora desestimaremos el uso de tags como *blonde_hair*, *white_hair*, *red_eyes*, etc.

3.12. Entreno con una red de neuronas convolucional

Como el entreno con la red de neuronas densa fue descartado por su alta carga en memoria se decide pasar directamente a un entreno usando la arquitectura de las redes convolucionales. Con el uso de estas capas se consigue reducir el tamaño de la red, pues el tratamiento de las capas convolucionales se hace a través de operaciones matriciales y además contamos con las capas de *MaxPooling* que consiguen reducir la dimensión de la información para aligerar la carga en la red.

3.12.1. Salida de la red

Con esta red se quiere hacer una clasificación de 3 etiquetas:

- 1girl y short _ hair: Este grupo de etiquetas corresponden a las ilustraciones que cuentan con un único personaje femenino con el pelo corto.
- 1girl y long _ hair: Este grupo de etiquetas corresponden a las ilustraciones que cuentan con un único personaje femenino con el pelo largo.
- hat: Ilustración con un sombrero en ella.

Para conseguir esta salida se necesitan identificar las etiquetas que tengan menos imágenes, pues estas etiquetas serán mas significativas si están presentes en una imagen como previamente hemos estudiado. Con ello la carga se realizará de la manera siguiente. Al recibir una imagen se comprueba si esta tiene como etiqueta la de hat y si es así se le asignará esa como su etiqueta, si no es así se comprobará con 1girl y short _ hair y por último con 1girl y long _ hair, finalmente si no cuenta con ninguna de las 3 etiquetas se descartará para el entreno.

De esta forma se asegurará que las imágenes con una chica y pelo largo no tengan etiquetas de una chica y pelo corto ni de sombrero y las imágenes de una chica y pelo corto no cuenten con la etiqueta de sombrero. El resultado de este proceso es que si una imagen cuenta con sombrero se clasifique como tal independientemente de sus otros elementos, de la misma manera si no tiene sombrero pero sí el pelo corto y una única chica se clasifique como tal, por último una chica y el pelo largo.

Además de esto se controlarán las imágenes cargadas de cada una de las categorías impidiendo una diferencia de más de 10 imágenes entre categorías. Esto unido a la clasificación de etiquetas anteriormente nombrada permite que la carga se equilibre impidiendo que haya muchas imágenes de una misma categoría y la red pueda caer en un mínimo local clasificando todas las imágenes como la categoría con más imágenes.

Con todo esto se generará un conjunto de *batches* para el entrenamiento. Estos *batches* se elegirán de manera aleatoria para cada iteración del entrenamiento.

3.12.2. Estructura de la red

La red contará con un conjunto de capas convolucionales seguidas de capas de *MaxPooling* en las que como entrada de la primera capa se recibirá la imagen de entrada y poco a poco se irá extrayendo su información mientras se reduce su dimensionalidad.

Después de producirse las convoluciones de la red habrá una capa de *flatten* a partir de la cual habrán ciertas capas de neuronas densas hasta reducir el tamaño a las 3 neuronas de salida buscadas. Allí se realizará como función de activación la función *softmax* que dará una única salida como resultado.

La figura 28 representa un esquema visual de la estructura de la red creada:

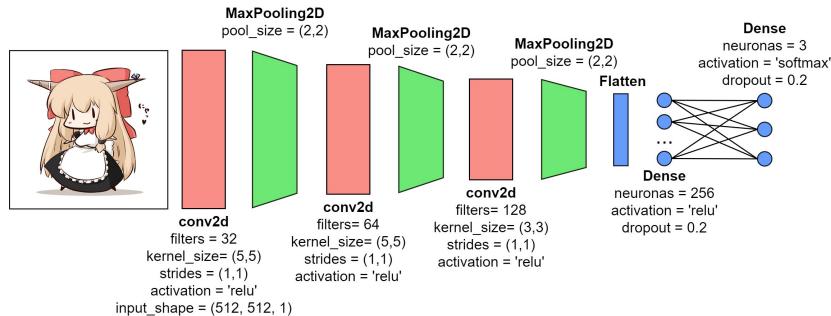


Figura 28: Representación visual de la red

La figura 29 contiene una lista más extensa de los parámetros de la red, dicha lista corresponde a la salida proporcionada por la función *summary* de la librería de *python Keras*:

Model: "sequential"		
Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 508, 508, 32)	832
max_pooling2d (MaxPooling2D)	(None, 254, 254, 32)	0
conv2d_1 (Conv2D)	(None, 250, 250, 64)	51264
max_pooling2d_1 (MaxPooling2 (None, 125, 125, 64)	0	
conv2d_2 (Conv2D)	(None, 123, 123, 128)	73856
max_pooling2d_2 (MaxPooling2 (None, 61, 61, 128)	0	
flatten (Flatten)	(None, 476288)	0
dense (Dense)	(None, 256)	121929984
dropout (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 3)	771
dropout_1 (Dropout)	(None, 3)	0
<hr/>		
Total params: 122,056,707		
Trainable params: 122,056,707		
Non-trainable params: 0		

Figura 29: Resumen de la red proporcionado por la función *summary* librería *Keras*.

3.12.3. Resultados del entreno

Mientras se produce el entreno se observa que el equipo cuenta con grandes dificultades para proseguir. Por una parte la velocidad a la que se suceden los diferentes *batches* es muy lenta; unido a esto la velocidad a la que se la red aprende es demasiado lenta. Como resultado no se llega a observar ninguna mejora en los resultados de la red por lo tanto este modelo no supone ningún resultado. La precisión de la red no mejora de manera consistente en ningún punto del entreno.

La conclusión de por qué sucede esto es porque la red esta sobredimensionada y por lo tanto le cuesta procesar tanta información. Imaginamos que si el entreno siguiese llegaría un punto en el que se comenzarían a obtener resultados pero preferimos simplificar el modelo para obtener resultados de manera más rápida. Debido a esto se decide finalmente realizar más cambios en la arquitectura de nuestro modelo para obtener resultados.

3.13. Cambios en el modelo de entrenamiento

En este punto tras dos entrenamientos fallidos se considera que lo mejor es realizar cambios en la forma de entrenamiento de las redes del proyecto. Con ello pretendemos solucionar los problemas vistos en los entrenamientos fracasados aprendiendo de ellos.

3.13.1. Reducción de dimensionalidad de las imágenes

Como hemos observado en el entrenamiento anterior la capacidad de nuestro equipo de procesar las imágenes es demasiado limitada como para poder realizarlo con la suficiente velocidad. Para solventar este problema decidimos reducir el tamaño de las imágenes de nuestro *dataset* para así aligerar la carga total de la red.

Con la reducción de tamaño conseguiremos un conjunto de mejoras en nuestras redes:

- **Reducción del tamaño de la red:** Como los datos de entrada ocupan menos espacio las capas que procesan dicha información serán de menor tamaño. Con esto se conseguirá que la reducción del tamaño entre capas sea menos brusco ya que anteriormente la diferencia de tamaños entre capas podría impedir la extracción de cierta información. Además de esto la velocidad de procesamiento de cada imagen por la red será mucho más rápida pues es menor la información a procesar.
- **Mayor velocidad de carga:** Como el tamaño de cada una de las imágenes va a reducirse, la velocidad a la que estas se cargan en memoria será más rápida permitiendo realizar más *batches* en menos tiempo.
- **Disminución del tamaño en disco:** Como las imágenes ocupan menos espacio en memoria principal no sólo se liberará espacio durante los entrenamientos sino que a la hora de realizar los entrenamientos se podrá aumentar el tamaño de las imágenes por cada *batch* para aumentar al máximo la eficiencia de los entrenamientos.

Estudio de la reducción de dimensión. A la hora de reducir la dimensión de las imágenes decidimos pasar de 512x512 píxeles a 128x128 píxeles de imágenes en blanco y negro, esto es una dimensión de (512,512,1). Al realizar dicho cambio debemos estudiar qué cambios pueden haber afectado a las imágenes del *dataset*, en concreto estudiamos la pérdida de información que se pueda haber producido con esta reducción.

La figura 30 muestra una imagen antes y después del cambio de dimensionalidad. Originalmente la imagen contaba con elementos como los lazos de las coletas del personaje principal, tras el cambio dichos lazos son apenas visibles. Sin embargo todos los elementos originales aún son diferenciables, los lazos no han desaparecido totalmente y otros elementos como las coletas o los ojos del personaje aún son visibles.



Figura 30: Imagen antes y después de la reducción de dimensionalidad.

Se considera que la reducción de dimensionalidad es idónea. La dimensión se ha reducido a un cuarto del original. Pese a poderse haber perdido información la pérdida es mínima pues todos los elementos originales aún siguen siendo visibles. Aunque la visibilidad haya sido reducida recordamos que el objetivo es que las ilustraciones puedan ser procesadas por una red neuronal y siempre y cuando sigan presentes podrán ser identificadas.

3.13.2. Desequilibrio de carga de imágenes en los *batches*

Hasta ahora la manera por la cual se decidían las imágenes con las que se entrenaba la red era la siguiente:

1. Se forma un *array* con pares de imágenes/etiquetas. El *array* estará formado por dos campos en cada posición, el primero correspondiente a la imagen y el segundo campo a la etiqueta correspondiente a dicha imagen. La manera por la cual se escogen dichos pares es leyendo una a una las imágenes de manera aleatoria y almacenando su información a medida que estas aparecen en los archivos de meta-datos.
2. Una vez se forma el *array* del apartado anterior las imágenes se formatean y normalizan sus valores al rango [0,1].
3. Por último se realiza el entreno con el *batch* guardado.
4. Este proceso se repite para el siguiente *batch*.

Los problemas derivados de este modelo de carga son los siguientes:

- Al elegir las imágenes de manera aleatoria no podemos asegurar de que haya ciertas imágenes con las que se entrenen muchas veces mientras que otras imágenes no aparezcan en ningún entreno. Esto puede provocar *overfitting* en la red pues el número de imágenes de entreno es menor al total.
- No se define un *epoch* en el entreno. Al no realizar un entreno ordenado sobre todas las imágenes no se puede llevar un recuento de cuándo se ha entrenado con todas las imágenes y por lo tanto no se puede definir un *epoch*. La función del *epoch* definido no es más que hacer una división de grupos de *batches* y por lo tanto no es verdaderamente un *epoch*. Debido a esto no se puede reservar un subconjunto del total para hacer testing sobre el entreno de la red.
- La distribución de carga en cada *batch* está controlada de manera parcial. Al cargar las imágenes de cada *batch* se impide que la diferencia entre las imágenes cargadas de cada categoría sea mayor a cierto número, 10 imágenes de diferencia.

Ante estos problemas decidimos crear un nuevo modelo de carga de imágenes para solucionar los problemas encontrados.

3.13.3. Carga de imágenes basada en *epoch*

Para solventar los errores del modelo anterior decidimos basarnos en los *epoch* para nuestra nueva carga de imágenes. Para nuestro entreno tenemos que cargar un número de imágenes por cada una de las etiquetas a entrenar, para ello hacemos el siguiente recorrido:

1. Recorremos los diferentes archivos de meta-datos uno a uno completamente. Por cada una de las imágenes que almacena información comprobaremos si contiene alguna de las etiquetas del entreno. En caso afirmativo guardamos la ruta completa de dicha imagen en un *array*, tendremos un *array* por cada una de las etiquetas que queremos clasificar de modo que se generen tantos *array* como etiquetas queramos clasificar. De la misma manera tendremos un *array* por cada una de las etiquetas en el que guardaremos el número de la etiqueta correspondiente a la imagen de la misma posición. Este proceso se realiza mediante la función *get_imgs*.

2. Una vez formados todos los *arrays* estos serán mezclados formando dos *arrays*, uno de imágenes y otro de etiquetas. Ambos *arrays* estarán divididos en el número de *batches* del entrenamiento y en ellos estarán situadas las imágenes alternando una a una imágenes por cada una de las etiquetas. Con esto se conseguirá que cada uno de los *batches* esté formado exactamente por el mismo número de imágenes de cada categoría y la carga total esté completamente equilibrada. Cada *epoch* corresponde a un entrenamiento por cada uno de los *batches* definidos durante la carga. Este proceso se realiza mediante la función *mix_dataset*.
3. Antes de realizar un entrenamiento se escoge el *batch* correspondiente al número del entrenamiento, dicho *batch* corresponde con el *subarray* formado anteriormente. Las imágenes de dicho *array* se cargan en memoria, se formatean y se normaliza su valor al rango [0,1] para prepararlas para el entrenamiento.
4. Se realiza el entrenamiento con el conjunto de imágenes cargado.
5. Se carga el siguiente conjunto de imágenes.

En la figura 31 podemos observar un esquema de cómo se realiza la nueva carga de imágenes para un ejemplo en el que se usan 3 etiquetas, 9 imágenes por cada clase y el entrenamiento se divide en 3 iteraciones o *batches*:

get_imgs

Array X

Etiqueta 1	Img1	Img2	...	Img9	[0]
Etiqueta 2	Img10	Img11	...	Img18	[1]
Etiqueta 3	Img19	Img 20	...	Img27	[2]

Array Y

Etiqueta 1	0	0	...	0	[0]
Etiqueta 2	1	1	...	1	[1]
Etiqueta 3	2	2	...	2	[2]

(a) Contenido de los *arrays* devueltos por la función *get_imgs* de la carga del *dataset* de entrenamiento.

mix_dataset

X_train

batch 1	Img1	Img10	Img19	Img2	...	Img21	[0]
batch 2	Img4	Img13	Img22	Img5	...	Img24	[1]
batch 3	Img7	Img16	Img25	Img8	...	Img27	[2]

Y_train

batch 1	1	2	3	1	...	3	[0]
batch 2	1	2	3	1	...	3	[1]
batch 3	1	2	3	1	...	3	[2]

(b) Contenido de los *arrays* devueltos por la función *mix_dataset* de la carga del *dataset* de entrenamiento.

Figura 31: Contenido de los *arrays* donde se almacenan la información de las rutas de las imágenes y sus respectivas etiquetas.

3.13.4. Entrenamiento de nuestro modelo

El modelo se basa en los diferentes *batches* previamente definidos, cuando el modelo es entrenado con todos los *batches* se realiza una época o *epoch* del entrenamiento. Durante todo el entrenamiento se controlan las métricas de *loss* y *accuracy* de cada iteración del mismo. Estas medidas son las que nos servirán para comprobar el estado de aprendizaje del modelo.

Cada cierto número de *batches* se realiza una predicción sobre un *batch* aleatorio de los reservados para test. Estos *batches* están compuestos por imágenes que se reservan para comprobar el estado de aprendizaje de nuestra red ante imágenes con las que no se había encontrado antes, es decir, las imágenes de test son un conjunto de imágenes cuya función es comprobar el estado del modelo ante un caso real nunca antes visto. Por ello estas imágenes no se usan para entrenar, sólo se usan para comprobar la precisión de las predicciones.

3.13.5. Métricas del entrenamiento

Para controlar nuestro entrenamiento se debe llevar un registro de cómo evoluciona el aprendizaje del mismo. Para ello se debe recoger de alguna forma las diferentes métricas del mismo. Se recogerán de alguna manera los resultados de las predicciones de nuestro modelo a lo largo del entrenamiento. Actualmente las medidas del entrenamiento que se encuentran son las siguientes:

- **Accuracy:** La métrica de *accuracy* o precisión devuelve el porcentaje de aciertos que ha tenido la red con las imágenes del *batch* de imágenes de entrada.
- **Loss:** La métrica de *loss* o pérdida da un número que hace referencia al estado del entrenamiento en el que se encuentra el modelo. Cuanto menor sea ese número la red estará más entrenada. Del conjunto de funciones de pérdida que podemos escoger decidimos elegir la función de *categorical crossentropy* pues es la más idónea para un problema de clasificación multi-etiqueta como el que nos enfrentamos.

Estas métricas deben ser recogidas para tener una visión de cómo evolucionan a lo largo del entrenamiento, para ello recogemos las mismas en las siguientes gráficas:

- **Métricas de cada batch:** En una gráfica por cada métrica recogemos su valor para cada uno de los *batches* de nuestro entrenamiento. Pese a ser un conjunto muy completo presenta como principal inconveniente el gran número de valores de cada entrenamiento. Como por cada entrenamiento se realizarán un gran número de *batches* el número de métricas será muy alto y por lo tanto la legibilidad de este tipo de gráficas será baja como se puede observar en la figura 32. Este problema lo tenemos en cuenta pero decidimos conservar estos gráficos pues en ellos se guarda la información de los entrenamientos sin ningún tratamiento previo y por tanto sirve por si en un futuro queremos hacer cálculos derivados de estos valores.

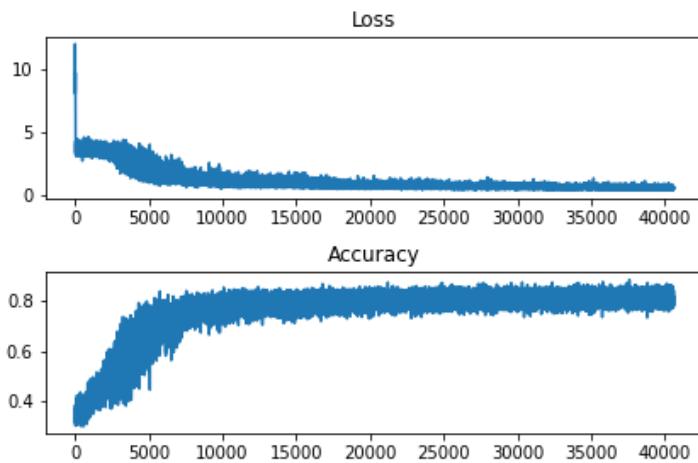


Figura 32: Gráfico de métricas de *accuracy* y *loss* para los *batches* a lo largo de un entrenamiento.

- **Métricas de cada epoch:** En una gráfica por cada métrica recogemos su valor medio para cada uno de los *epoch* del entrenamiento. A diferencia de las métricas por *batch* esta gráfica tiene mejor visibilidad, como se puede observar en la figura 33 donde se puede diferenciar el valor de cada métrica en cada uno de los puntos del entrenamiento ya que al ser una media de cada *epoch* se recogen varios valores en un mismo punto. Un efecto negativo de esto es que el valor de cada punto es una media y por lo tanto no da tanta información como todo el conjunto de valores.

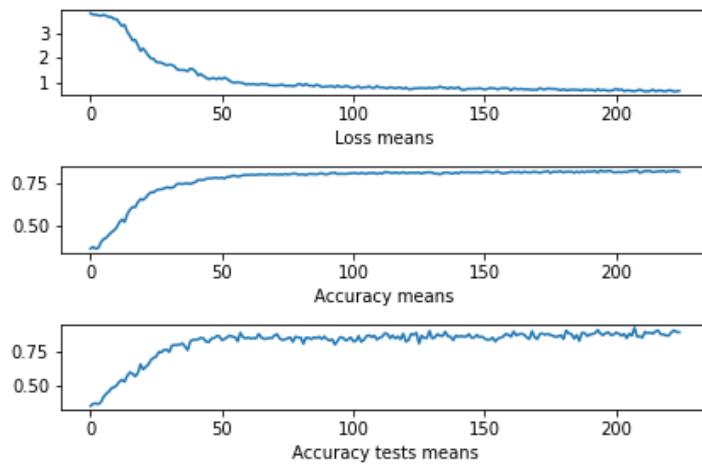


Figura 33: Gráfico de métricas de las medias de *accuracy* y *loss* para los *epoch* a lo largo de un entrenamiento.

- **Métricas de cada test:** En una gráfica por cada métrica recogemos su valor para cada predicción sobre los datos de test realizadas durante el entrenamiento. La visibilidad de esta gráfica es buena como se puede observar en la figura 34 debido a que el número de predicciones es mucho menor al de *batches*. Para los test sólo tendremos como métrica la precisión ya que el *loss* es un valor dependiente del entrenamiento. Esta gráfica nos dará información de cómo evoluciona nuestro modelo a lo largo del entrenamiento cuando se enfrenta a ilustraciones que no ha visto para entrenar.

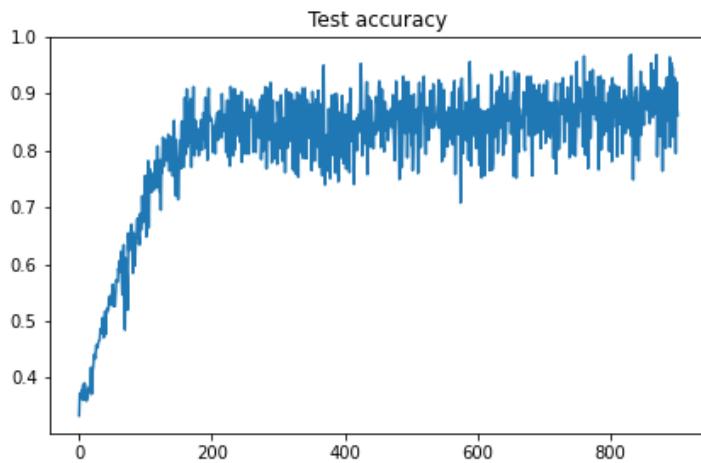


Figura 34: Gráfico de métricas del *accuracy* a lo largo de las predicciones de un entrenamiento.

De todas las gráficas se guardarán tanto las gráficas a lo largo del entrenamiento como los valores de sus diferentes puntos. De esta forma respectivamente se podrá ver cómo se van formando las gráficas y se podrán tratar los datos más adelante si fuese necesario obtener otro tipo de información de ellos.

Por otra parte se decide guardar la matriz de confusión²⁴ de las diferentes predicciones de test a lo largo del entrenamiento. En la figura 35 se puede observar un ejemplo de predicciones realizadas sobre un conjunto de x imágenes con 3 etiquetas posibles. En la tabla se pueden observar cómo se agrupan las predicciones siendo la mayoría de ellas aciertos pues la etiqueta real y la predicha coinciden. Las dos tablas corresponden con los valores absolutos del número de imágenes predichas (arriba) y el valor normalizado (abajo) corresponde con el porcentaje de imágenes con ese par de etiqueta/predicción.

²⁴La matriz de confusión es una matriz que permite la visualización de la precisión de las predicciones realizadas sobre un conjunto de imágenes. En ella se recogen el número de veces que una imagen con la etiqueta de la fila ha sido identificada con la etiqueta de la columna, por tanto las veces que se acertó corresponden con los números de la diagonal principal de la matriz.

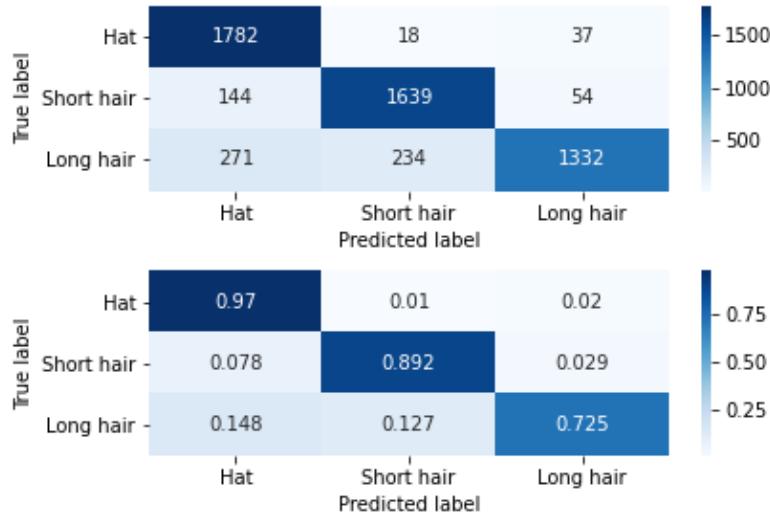


Figura 35: Matriz de confusión para la predicción de las etiquetas de las etiquetas sombrero, pelo corto y pelo largo.

3.14. Entreno tras los cambios realizados

Después de solucionar los problemas detectados en los anteriores entrenos se pasa a realizar un entrenamiento con el objetivo de obtener resultados válidos. Con el entrenamiento de este modelo se pretende realizar la clasificación de las 3 etiquetas de los anteriores entrenos:

- **1girl y short _ hair:** Este grupo de etiquetas corresponden a las ilustraciones que cuentan con un único personaje femenino con el pelo corto.
- **1girl y long _ hair:** Este grupo de etiquetas corresponden a las ilustraciones que cuentan con un único personaje femenino con el pelo largo.
- **hat:** Ilustración con un sombrero en ella.

El entrenamiento se realizará con un total de 37.500 imágenes de cada etiqueta, siendo 30.000 las dedicadas al entrenamiento de la red y 7.500, un 20 % de las imágenes totales, al test. A medida que se realice el entrenamiento se irán guardando las métricas anteriormente mencionadas.

3.14.1. Resultados del entrenamiento

A medida que se sucede el entrenamiento se observa cómo los resultados obtenidos mejoran llegando a un punto en el que se consigue una precisión en entorno al

50 %. Esta precisión es significativa debido a que la clasificación se realiza sobre tres etiquetas y la precisión para predicciones aleatorias es del 33 %; por tanto se ha conseguido una mejora del 17 %.

En este punto se confirma que nuestra nueva red es capaz de aprender las características de las diferentes ilustraciones. Para seguir con el entrenamiento se decide cambiar de entorno para obtener los resultados finales. Antes de seguir usando nuestro equipo se deciden buscar alternativas de uso remoto de equipos preparados para entrenos de inteligencias artificiales. Si se logra hacer uso de equipos remotos se aumentará en gran medida la velocidad de entrenamiento y optimizaremos nuestro tiempo consiguiendo mejores resultados en menor tiempo.

3.15. Búsqueda de entornos de entrenamiento remotos

Para realizar los entrenamientos se decide buscar algún sistema que nos permita usar los recursos de otra máquina para llevar a cabo completamente nuestro proyecto. Para ello se busca algún entorno que permita el uso de máquinas que tengan tarjetas gráficas con la mayor potencia posible para acelerar al máximo nuestros entrenamientos.

3.15.1. *Google Colaboratory*

La herramienta de Google llamada *Google Colaboratory* o *Google Colab* es un entorno de desarrollo remoto que permite ejecutar código en los servidores de la nube de Google. En concreto para nuestro proyecto nos interesa ejecutar código en *python* haciendo uso de equipos en la nube que cuenten con una tarjeta gráfica lo más potente posible para acelerar al máximo nuestros entrenamientos. A través de esta herramienta se puede acceder al uso de tarjetas gráficas que dependiendo de la sesión se hará uso de una gráfica diferente pues no se puede controlar el equipo concreto que se va a usar.

Importado del dataset. Al realizar los entrenamientos se debe contar con las imágenes y los meta-datos de los que hace uso el modelo del proyecto. Para poder entrenar de manera remota se deben importar a la máquina remota dichos archivos y posteriormente estos podrán ser cargados en la memoria de la máquina remota tal y como si fuese en nuestra máquina local.

Para importar a la máquina el conjunto de datos se decide crear un repositorio en *git* el cual pueda ser importado en la máquina remota y una vez importado, usado sin problema. De esta forma a la hora de realizar un entrenamiento el primer paso será cargar en el equipo nuestro conjunto de datos a través de *git* haciendo una copia del repositorio en la máquina remota. Una vez contemos con el repositorio el programa en *python* hará uso del mismo para realizar los entrenamientos deseados.

Salidas de los entrenos. Desde *Google Colab* se puede acceder al sistema de ficheros de *Google Drive*, por lo tanto todos los archivos que se generan como resultado de los diferentes entrenamientos serán almacenados en la cuenta de *Google Drive* asociada a *Google Colab*. Estos resultados se dividen en dos grupos:

- **Métricas del entrenamiento:** Las métricas definidas anteriormente se almacenarán en carpetas del *drive*. Para ello sólo debemos controlar que las rutas de las carpetas donde almacenar los archivos se encuentren presentes en nuestro sistema de ficheros de *drive*.
- **Red de neuronas:** Para guardar el estado de entrenamiento del modelo entrenado se guardará por una parte la estructura de la red de neuronas en un fichero .json y los pesos de la red en ficheros con extensión .hdf5. Con estos dos archivos se podrá recuperar tanto la red en sí como sus pesos para poder exportarla de la máquina remota.

El proceso de guardado de la red se realizará con una periodicidad igual que las predicciones por simplificar el entrenamiento, pero si fuese necesario se podría hacer con otro periodo. Al guardar periódicamente la evolución de la red se podría sobreentrenar a la red de modo que de este modo se asegura no caer en mínimos locales ya que posteriormente se podrá elegir una versión menos entrenada de la red si así fuese necesario.

Resultados

4.1. Estructura del capítulo

Como hemos indicado en la sección 3.15.1 los resultados del trabajo se plasman en los resultados del entrenamiento de las diferentes redes que se configuren para nuestro cometido. De esta forma en esta sección recogeremos los resultados y gracias a las métricas que ya hemos definido podremos obtener conclusiones de los resultados y la evolución en cada uno de nuestros entrenamientos.

4.2. Objetivos de los diferentes entrenos

El objetivo es desarrollar hasta su punto óptimo 3 redes de neuronas, diferenciando en cada una de ellas el número de etiquetas que diferencia:

- **3 clases:** La red más sencilla de todas, en un principio se considera el objetivo de esta red como una base que confirme que la red es capaz de obtener resultados buenos en su clasificación. Debido al poco número de clases la diferenciación no será especialmente interesante, sin embargo será lo suficientemente sencilla como para formar una base sólida sobre la cual construir las sucesivas redes de mayor tamaño.
- **5 clases:** Esta red es de un tamaño medio, se considera esta como la red más estándar de las que vamos a construir. Al tener un número de clases no demasiado elevado creemos poder obtener buenos resultados con el entrenamiento, pero ligeramente inferiores a la red clasificadora de 3 tipos de etiquetas. Al contar con 5 etiquetas la clasificación será lo suficientemente compleja para probar la capacidad de nuestras redes ante situaciones de una gran complejidad.
- **7 clases:** Gracias a esta red se conseguirá estudiar cómo se comporta el rendimiento de las redes cuando se aumenta en gran cantidad el número de etiquetas a diferenciar. El objetivo principal de esta red es comprobar hasta dónde se pueden dilatar el número de etiquetas clasificadas. Gracias a ella podremos comprobar y hacer estimaciones de cómo se pierde precisión a medida que aumentan el número de clases a clasificar.

Pese a querer obtener 3 redes no se desprecia la posibilidad de en un futuro realizar más redes para estudiar otras posibles arquitecturas. Esto conforma una base que más adelante podemos extender si fuese necesario.

La forma de elegir las etiquetas o grupos de etiquetas será acumulativa, es decir, las 3 etiquetas de la primera red serán junto a 2 etiquetas nuevas las etiquetas de la red de 5 etiquetas, de la misma manera el entrenamiento de 7 etiquetas estará formado por 2 etiquetas más.

4.3. Estructura general de las redes

La estructura de la red es la misma que la utilizada en el entrenamiento en nuestro equipo local. El único cambio que iremos realizando para las diferentes arquitecturas será adecuar la salida de la red al número de etiquetas que deseamos clasificar teniendo una neurona por cada clasificación que deseemos hacer.

La figura 36 muestra la estructura final que tendrán las redes durante los diferentes entrenamientos. La única diferencia existente entre cada entrenamiento es el número de neuronas presentes en la última capa de la red siendo este el mismo al número de etiquetas a clasificar.

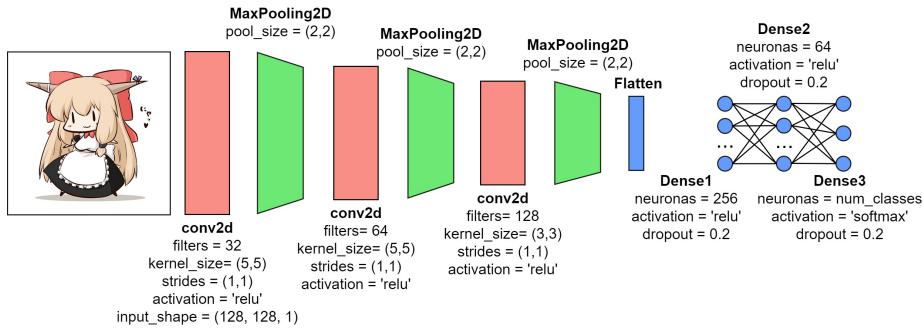


Figura 36: Arquitectura de las diferentes redes que usaremos para nuestros entrenamientos.

4.4. Entreno con 3 clases

Para realizar esta red se deciden escoger 3 etiquetas que sean sencillas de clasificar, pues como hemos indicado en la sección 4.2 el objetivo es obtener los mejores resultados posibles y formar una base con los mejores resultados que podamos obtener.

4.4.1. Etiquetas a clasificar

Las etiquetas a clasificar elegidas para la realización del entrenamiento son:

- **1girl y short _ hair:** Este conjunto de dos etiquetas representa unos elementos muy claros y comunes en el *dataset*. Al combinar los dos conseguimos reducir el número de imágenes con estas etiquetas pero a su vez conservamos el carácter general del conjunto. El cometido de la etiqueta 1girl es diferenciar elementos de la imagen muy sencillos de identificar a priori como pueden ser simplemente los bultos que representan los personajes, aún así hay cierta dificultad en saber diferenciar una persona de lo que pueden ser elementos decorativos o accesorios.

Por otra parte short_hair permite hacer diferenciación en el pelo de los personajes, el cual puede tener muchos estilismos y ser de diferentes maneras de modo que la red debe aprender una clasificación ligeramente subjetiva sobre en qué punto de comienza a considerar a un pelo como corto.

- **1girl y long_hair:** Este conjunto es muy parecido al anterior y las razones por las que se escogen estas etiquetas son las mismas pero ninguna de las dos clasificaciones tendría sentido sin la otra pues casi todos los personajes tendrán pelo. Al introducir la longitud como un factor que diferenciar buscamos que la inteligencia artificial sea capaz de tomar decisiones sobre el mismo elemento, en este caso el pelo, juzgando en qué estado está el mismo.
- **hat:** Esta etiqueta tiene como fin valorar la capacidad de la red de clasificar accesorios de los personajes de las imágenes. El sombrero además puede ser confundido con facilidad con el pelo por lo tanto con esta etiqueta seremos capaces de juzgar si la red diferencia de accesorios en lugares en los que en otras situaciones más comunes hay otro tipo de elementos parecidos, es decir, queremos saber si se puede llegar a diferenciar un sombrero del pelo que en situaciones pueden llegar a tener formas y colores parecidos. Además los sombreros pueden ser de muchos tipos y las imágenes con esta etiqueta cuentan con gorras, gorros, sombreros de copa, pamelas, etc. como podemos observar en la figura 37 que cuenta con varias imágenes del dataset con un gorro presente²⁵.



Figura 37: Imágenes con gorros de formas muy dispares.

²⁵Para una mayor legibilidad a la hora de ver las imágenes mostraremos las imágenes del dataset antes de la bajada de resolución pese a que desde los entrenos hagamos uso de las imágenes con baja resolución

El conjunto de etiquetas es fácilmente clasificable, sin embargo se debe observar cómo hemos escogido que la mayoría de etiquetas hagan referencia a elementos presentes en la cabeza de los personajes. Con esto se busca enfrentar a la inteligencia artificial a situaciones complicadas en las que se encuentre con varias clasificaciones posibles donde tiene que elegir el elemento más significativo de la misma. Si por ejemplo se confundiese frecuentemente el pelo corto con sombreros aparecería este problema al observar la matriz de confusión ya que se agruparían muchos fallos en esa situación por lo tanto podremos hacer conciencia de los problemas a los que pretendemos enfrentar nuestra arquitectura.

Estas situaciones descritas se deben a que hay cierta intersección de imágenes que comparten etiquetas por ejemplo puede haber una ilustración en que cuente con una chica con el pelo corto pero que además lleve puesto un sombrero. El esquema de la figura 38 recoge el conjunto de situaciones que se pueden dar.

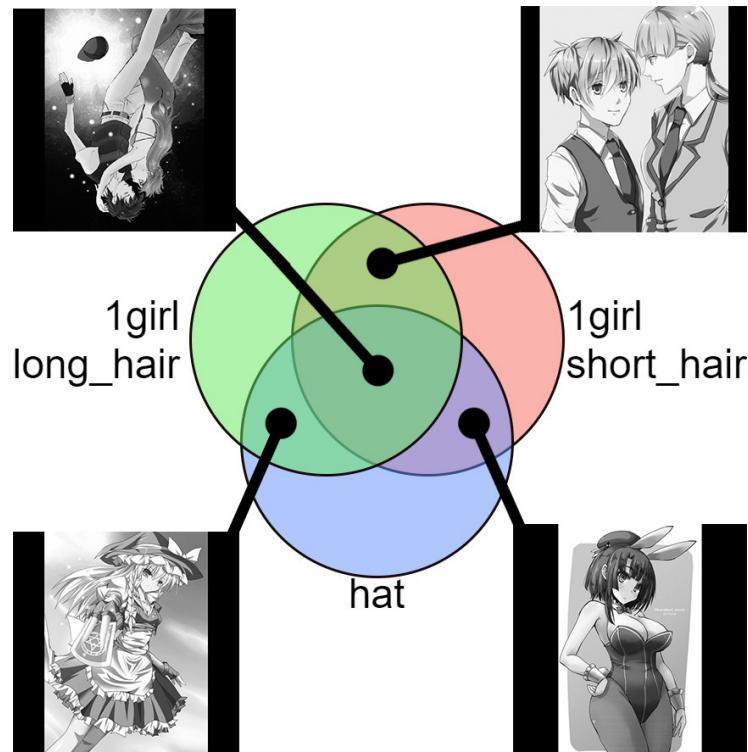


Figura 38: Diagrama de conjuntos con las imágenes posibles entre las intersecciones de las etiquetas 1girl y short_hair, 1girl y long_hair y hat.

4.4.2. Resultados del entrenamiento

Se realizará un entrenamiento de 225 *epoch* con un tamaño de *batch* de 501²⁶ imágenes y con 37.500 imágenes por cada clase, 112.500 imágenes en total de las cuales un 20 % del total (22.500 imágenes) se reservan para el testeo.

Observamos la figura 39 que contiene la evolución media del *loss* y la precisión del entrenamiento así como la precisión de los tests. Dichos datos están agrupados formando la media de cada *epoch*:

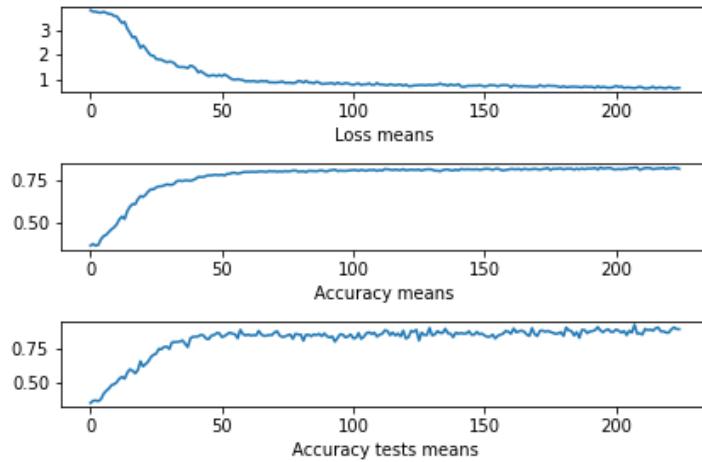


Figura 39: Evolución de las medias de las métricas medidas para la clasificación de 3 clases.

Como se puede observar a medida que avanza el entrenamiento se estabiliza una precisión de en torno al 85 % tanto para la precisión de imágenes de test como para las predicciones de los entrenamientos. Este dato es muy alto pues cabe recalcar que la precisión aleatoria para la clasificación de 3 clases es del 33 %.

A medida que avanza el entrenamiento se observa cómo la precisión aumenta hasta llegar al máximo en el *epoch* número 50, a partir de ese punto no se observa una pérdida de precisión en las imágenes del test, lo cual es buena señal pues indica que no se produce *overfitting*.

Por su parte se puede observar cómo el *loss* disminuye de manera exponencial hasta llegar a más o menos un 0.6 lo que indica que la red ha entrenado lo suficiente como para tener en cuenta sus resultados.

²⁶El número de imágenes por *batch* ha de ser múltiplo del número de clases para poder equilibrar exactamente el número de imágenes de cada clase en un *batch*

4.4.3. Evolución de las predicciones

Una vez hemos observado que el entrenamiento se ha realizado de manera correcta decidimos observar la evolución de la red y para ello decidimos observar la matriz de confusión generada para los tests en 3 puntos distintos del entrenamiento.

La figura 40 muestra la matriz de confusión generada para el *batch* 150 del *epoch* 0. En ella se puede observar cómo la red no ha aprendido absolutamente nada, debido a ello la predicción es prácticamente aleatoria escogiendo casi siempre como etiqueta de predicción *hat*.

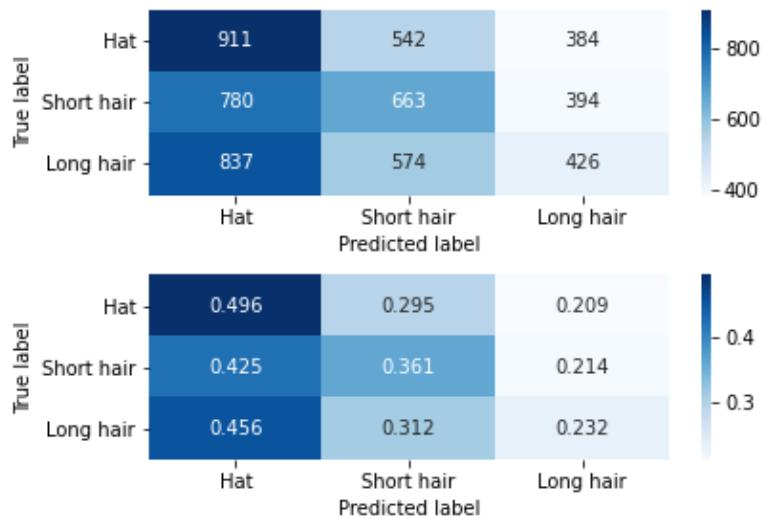


Figura 40: Matriz de confusión para el *batch* 150 del *epoch* 0.

La figura 41 muestra la matriz de confusión para un punto de mayor madurez de la red en el *epoch* 20 y *batch* 0, en ella se puede observar cómo la mayoría de los resultados se comienzan a agrupar en la diagonal principal de la matriz, dichos valores corresponden con los aciertos de la red.

Aún así se observa cómo los resultados se siguen agrupando aunque en menor medida en la etiqueta long_hair²⁷, esto se debe a que la red comienza a aprender características de la imagen pero aún tiene un gran factor aleatorio.

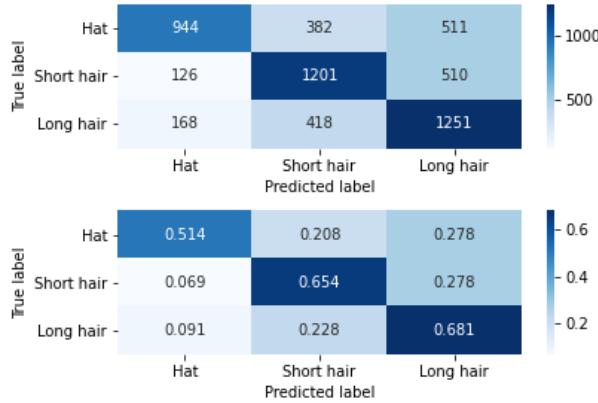


Figura 41: Matriz de confusión para el *batch* 0 del *epoch* 20.

La figura 42 muestra la matriz de confusión en el estado de mayor madurez de la red, en el *epoch* 225 y el *batch* 0. En la matriz se puede observar claramente como casi todos los resultados están agrupados en los aciertos habiendo simplemente ciertos casos aislados de fallos.

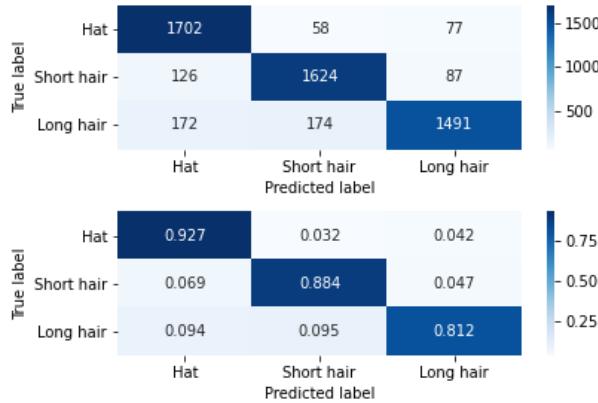


Figura 42: Matriz de confusión para el *batch* 0 del *epoch* 225.

²⁷La etiqueta long_hair es el conjunto de etiquetas de long_hair y 1girl que por mayor simplicidad de lectura se decide eliminar

4.4.4. Predicciones sobre imágenes

Para completar el estudio de los resultados decidimos estudiar las salidas producidas por la red al realizar ciertas predicciones sobre imágenes que consideramos interesantes.

La figura 43 contiene el resultado del uso de nuestro modelo entrenado para clasificar los elementos de una imagen que consta de una única chica con el pelo corto por la nuca pero con flequillo largo. Dicha situación presenta una clasificación complicada para la red ya que el pelo pese a ser largo podría ser también identificado como corto. Como se puede observar la clasificación de la red es pelo largo, dicho resultado se considera como inválido debido a que el pelo del personaje en general es corto pero tiene el flequillo largo. Es remarcable que la etiqueta correspondiente a pelo corto se activa ligeramente pues la red es capaz de reconocer el pelo corto, sin embargo predomina la parte larga del pelo.



Figura 43: Resultados de la predicción de una imagen con el pelo de longitud media.

La figura 44 contiene la salidas del modelo para una imagen que presenta dos personajes, uno de ellos cuenta con un yelmo medieval. Con los resultados se puede observar cómo la red es capaz de clasificar de manera correcta sombreros de formas raras como en este caso un yelmo.

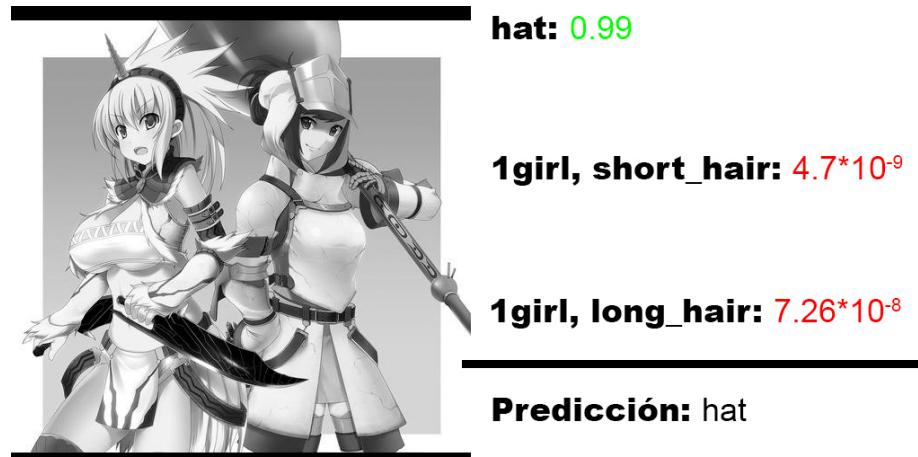


Figura 44: Resultados de la predicción de una imagen con un yelmo como sombrero.

La figura 45 presenta una situación de muchos personajes con el pelo corto. Como se puede observar la predicción del modelo es 1girl, short _hair lo cual indica que la red sólo clasifica si el pelo es corto o largo siendo la predicción final incorrecta. Pese a no ser válida la predicción es la más correcta de las 3 posibles por lo tanto consideramos como un éxito la predicción.



Figura 45: Resultados de la predicción de una imagen con varios personajes con el pelo corto.

4.5. Entreno con 5 clases

En este punto se quiere estudiar cómo se ve afectado el rendimiento de nuestras redes cuando se aumentan de manera leve el número de etiquetas a clasificar. Para ello se decide conservar las 3 etiquetas de el anterior entreno y añadir al conjunto 2 etiquetas más que aumenten la complejidad de nuestro sistema.

4.5.1. Etiquetas a clasificar

Las etiquetas a clasificar elegidas para la realización del entreno son:

- **1girl y short_hair:** Se elige por el mismo motivo que en el entreno anterior.
- **1girl y long_hair:** Se elige por el mismo motivo que en el entreno anterior.
- **hat:** Se elige por el mismo motivo que en el entreno anterior.
- **dress:** La etiqueta de vestido supone clasificar elementos de tamaños y formas muy variables, con ello se pondrá a la inteligencia artificial ante situaciones en las que un vestido puede ser de formas, tamaños y colores muy diferentes. Con ello se pretende seguir la línea de testear los límites de nuestra arquitectura.

- **skirt:** Si bien la etiqueta del vestido supone una clasificación complicada esta se complementa perfectamente con la falda pues ambas etiquetas hacen referencia a elementos difícilmente diferenciables. Distinguir entre una falda y un vestido puede ser complicado pues en la mayoría de ocasiones los vestidos están abiertos por la parte inferior en forma de falda como se puede observar en la figura 46.



Figura 46: Imágenes con un vestido abierto en forma de falda.

Con el conjunto de estas etiquetas se quiere expandir las posibilidades del modelo llegando a más clasificaciones y estudiando cómo añadir etiquetas puede afectar al desempeño de nuestra red. Las etiquetas elegidas interactúan bien entre sí pero a su vez están diferenciadas de las 3 etiquetas del anterior entrenamiento. Esta decisión de elegir un conjunto de elementos que normalmente se encuentran en otra posición respecto a los anteriores se debe a que así se puede estudiar claramente la acción de expandir el número de etiquetas. Si se escogiesen elementos relacionados con la cabeza enturbiaríamos los resultados pues estaríamos estudiando muchos elementos muy parecidos mientras que el objetivo de este entrenamiento es estudiar el aumento de etiquetas. De esta forma aprovechamos todas las posibilidades que nos ofrecen las imágenes del *dataset*.

En cuanto a las intersecciones que se puedan producir entre las 2 nuevas etiquetas se observa que son ilustraciones en las que hay varios personajes con alguno de ellos llevando un vestido mientras que otro lleva una falda. Esta situación evaluará cómo es capaz el modelo de elegir siempre el vestido antes que la falda pues como hay menos imágenes de vestidos consideramos que es el elemento más significativo. En la figura 47 podemos observar un diagrama que muestra esta situación.

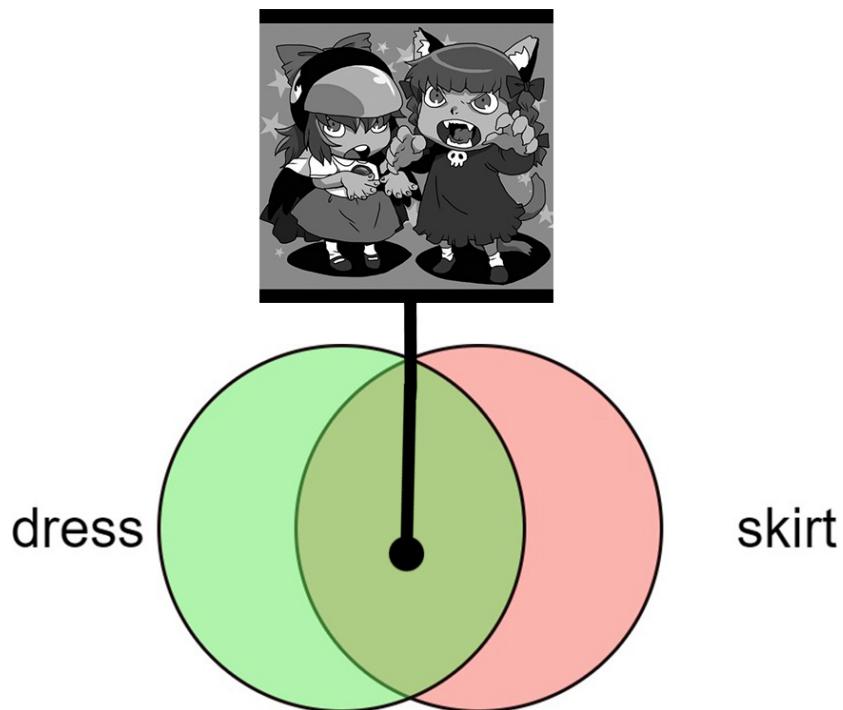


Figura 47: Diagrama de conjuntos con las imágenes posibles entre las intersecciones de las etiquetas dress y skirt.

4.5.2. Resultados del entrenamiento

Se realizará un entrenamiento de 204 *epochs* con un tamaño de batch de 500 imágenes y con 25.000 imágenes por cada clase, 125.000 imágenes en total de las cuales un 20 % del total (25.000 imágenes) se reservan para el testeo.

Se observa en la figura 48 un comportamiento similar al de el entrenamiento de la sección 4.4:

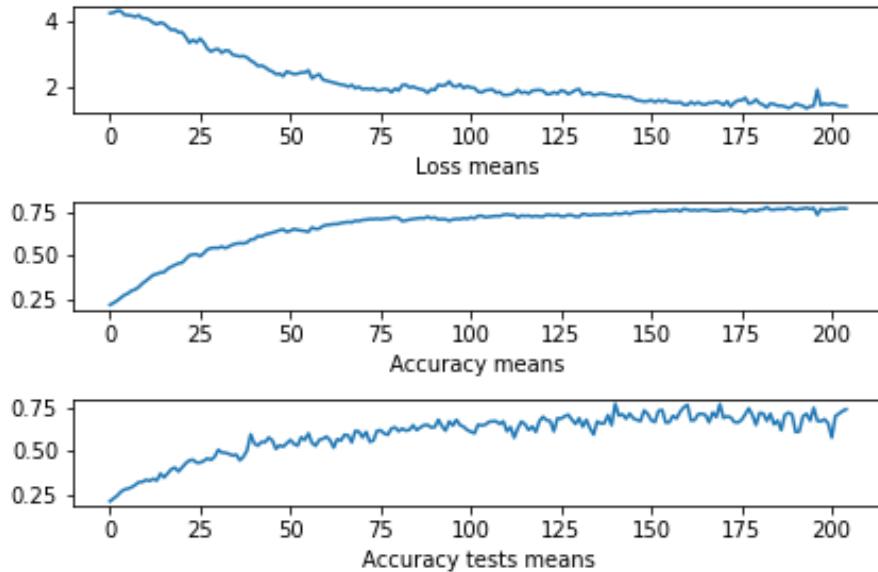


Figura 48: Evolución de las medias de las métricas medidas para la clasificación de 5 clases.

Como se puede observar en la figura la precisión de las predicciones ligeramente superior a un 75 %, recordamos que la precisión aleatoria para la clasificación de 5 clases es del 20 %.

De la misma manera que en el entrenamiento anterior se puede ver cómo se estabiliza la precisión en torno al *epoch* número 50.

4.5.3. Evolución de las predicciones

El comportamiento de las predicciones del entrenamiento es muy similar al del entrenamiento de la sección 4.4. En la figura 49 correspondiente al *batch* 0 del *epoch* 1 se puede observar cómo la red comienza haciendo predicciones aleatorias.

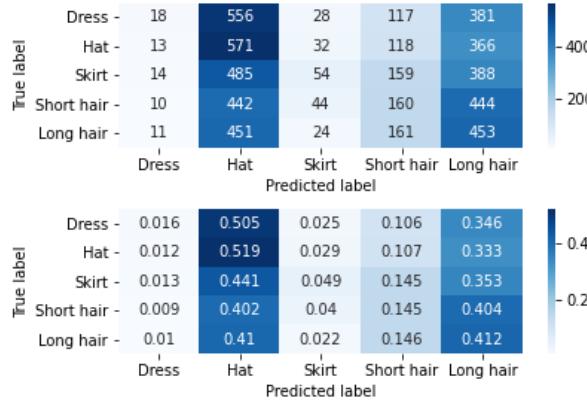


Figura 49: Matriz de confusión para el *batch* 0 del *epoch* 1.

En la figura 50 se observa la matriz de confusión para el *batch* 100 del *epoch* 205, en ella se puede observar cómo no hay ningún problema con las predicciones, llegando estas a una precisión del 80 % y distribuyéndose uniformemente.

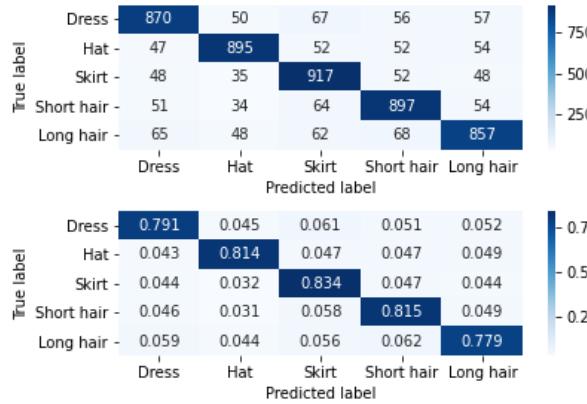


Figura 50: Matriz de confusión para el *batch* 100 del *epoch* 225.

4.5.4. Predicciones sobre imágenes

En la figura 51 se pueden observar las salidas del modelo para una imagen en la que el personaje lleva una falda de un color muy similar al de su camiseta. La clasificación de la red es la de una chica con el pelo corto cuando debería haber sido la de falda por ser un elemento más significativo. Pese a este fallo la red se activa un 80 % para vestido ya que no consigue diferenciar que son dos prendas distintas la falda y la camiseta y un 20 % de clasificación de falda que es la salida correcta.



Figura 51: Resultados de la predicción de una imagen con una chica de falda y camiseta difícilmente diferenciables.

El origen de la mala clasificación puede deberse a que al pasar a una imagen de baja resolución se ha perdido la resolución necesaria para distinguir la separación entre falda y camiseta como se puede observar en la figura 52 que cuenta con la imagen antes (izquierda) y después (derecha) de la bajada de resolución.

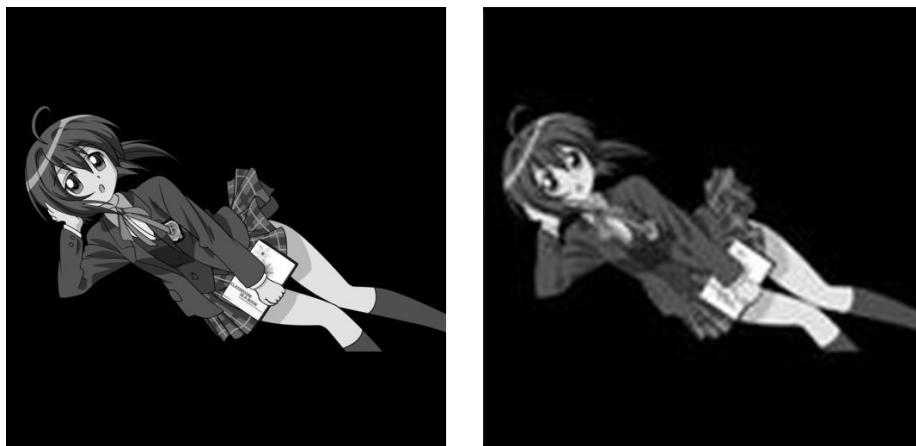


Figura 52: Imagen de la predicción antes y después de bajar su resolución.

La figura 53 muestra la predicción para una imagen de una chica con falda en la que se puede observar que pese a ser correcta la predicción la red cuenta con problemas para realizarla pues la salida para la etiqueta correcta es del 9 %. Se puede ver que ante casos dudables el modelo da predicciones de manera no muy segura pero sí correcta.



Figura 53: Resultados de la predicción de una imagen con una chica de falda.

4.6. Entreno con 7 clases

El último entreno que se realizará será con un modelo para clasificar 7 tipos de clasificaciones diferentes. Con este entreno se pretende enfrentar a una situación muy compleja a nuestra arquitectura y con ella estudiar cuál podría ser el futuro de versiones muy complejas de clasificadores similares al nuestro. Para ello añadiremos 2 últimas etiquetas a nuestro conjunto previo de etiquetas.

4.6.1. Etiquetas a clasificar

Las etiquetas a clasificar elegidas para la realización del entreno son:

- **1girl y short_hair:** Se elige por el mismo motivo que en el entreno anterior.
- **1girl y long_hair:** Se elige por el mismo motivo que en el entreno anterior.
- **hat:** Se elige por el mismo motivo que en el entreno anterior.
- **dress:** Se elige por el mismo motivo que en el entreno anterior.
- **skirt:** Se elige por el mismo motivo que en el entreno anterior.
- **breasts:** La etiqueta de pechos hace referencia a un concepto tan general que se considera especialmente interesante elegirla pues enfrenta a nuestro modelo a una característica difícil de diferenciar de una imagen. En concreto se considera que los pechos de un personaje pese a parecerse pueden ser muy distintos en dos imágenes y eso podría enfrentar a la arquitectura a situaciones complicadas como podemos observar en las ilustraciones de la figura 54 en las que el pecho se presenta de maneras muy diferentes.



Figura 54: Imágenes con la etiqueta breasts en la que el pecho se presenta de maneras muy diferentes.

- **blush:** Esta etiqueta presenta el elemento más difícil de identificar hasta ahora pues en la mayoría de las imágenes viene representado por una sombra a la altura de las mejillas, a priori dicha sombra sería muy fácilmente de identificar pero en la mayoría de casos la sombra apenas es percibible como podemos observar en la figura 55. Además se considera especialmente interesante el rubor pues normalmente puede ser identificado como una zona roja en las mejillas sin embargo al tratar las imágenes y pasárlas a blanco y negro se ha perdido la información de dicho color y con ello se complica más aún la identificación del rubor. Con ello se quiere estudiar si la pérdida de información por la pérdida de color puede afectar gravemente a nuestros resultados.



Figura 55: Imagen con la etiqueta blush en la que la presencia de rubor es muy difícil de ver.

Las nuevas etiquetas añadidas al conjunto presentan los elementos más difíciles de clasificar a priori. Con ello se quiere observar cómo se comporta nuestro modelo ante la peor situación a la que podemos enfrentarle. Esta es la última versión del modelo y con ella se quiere llegar a cubrir la situación más complicada a la que podamos llegar y por ello se ha tratado de elegir las etiquetas que consideramos más problemáticas.

En cuanto a las imágenes que puedan tener tanto la etiqueta de blush como la de breasts son muchas y no suponen ninguna situación especial pues se considera obvio que hay imágenes que cuentan con rubor en las mejillas y a su vez pechos. De la misma manera que se ha indicado anteriormente la red deberá ser capaz de clasificar con mayor prioridad las imágenes de pechos pues es más significativa. En la figura 56 se puede ver un ejemplo de imagen que cuenta con ambas etiquetas.

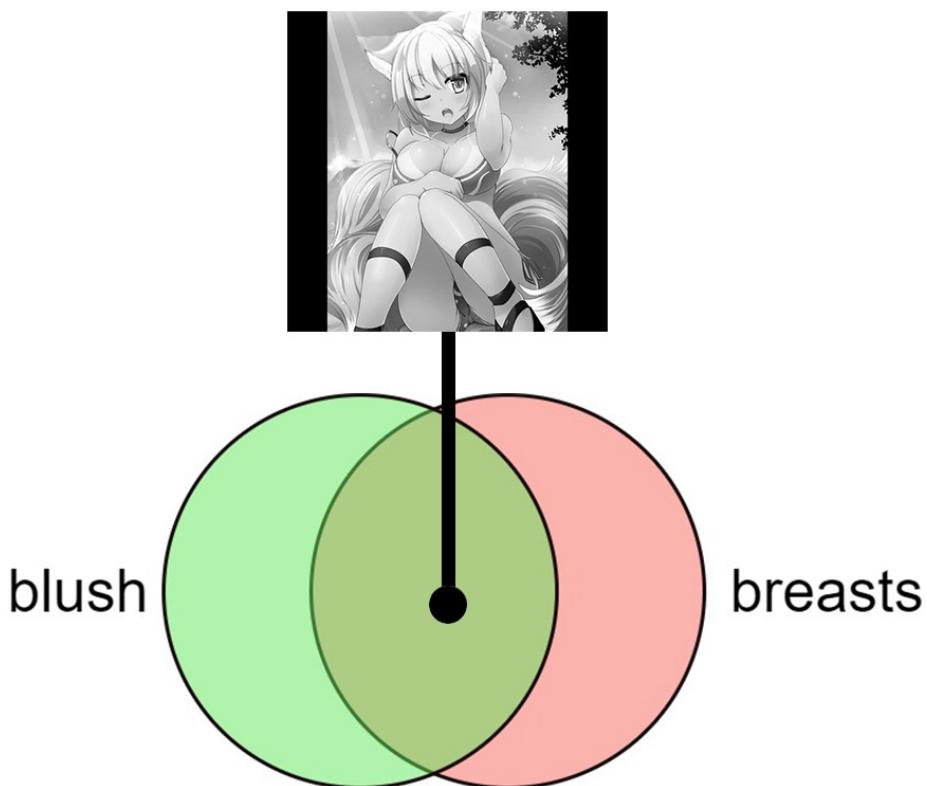


Figura 56: Diagrama de conjuntos con las imágenes posibles entre las intersecciones de las etiquetas breasts y blush.

4.6.2. Resultados del entrenamiento

Durante el desarrollo de este entrenamiento se observa la poca velocidad de aprendizaje de la red, hasta tal punto en el que se realizan un total de 710 *epochs* con un tamaño de *batch* de 504 imágenes y con 18.750 imágenes por cada clase, 131.250 imágenes en total, de las cuales un 20 % del total (26.250 imágenes) se reservan para el testeo.

Mientras se realiza el entorno se encuentra como problema que la aplicación de *Google Colab* limita el uso de la plataforma, terminando la sesión de manera abrupta al llegar a una cantidad de tiempo que es variable. Esto supone que cada cierto tiempo se tiene que reiniciar el entrenamiento, recuperando el estado del modelo anterior y concatenando ambos entrenamientos. Gracias a la estructura de guardado del estado de evolución del modelo que se ha diseñado este proceso se puede realizar con una sencillez relativa.

De esta forma se concatenan hasta un total de 7 entrenamientos remotos, llegando hasta los 710 *epochs*. A partir de este punto se decide terminar este entrenamiento y estudiar los resultados.

Se observa en la figura 57 un comportamiento similar al de los entrenamientos de las secciones 4.4 y 4.5:

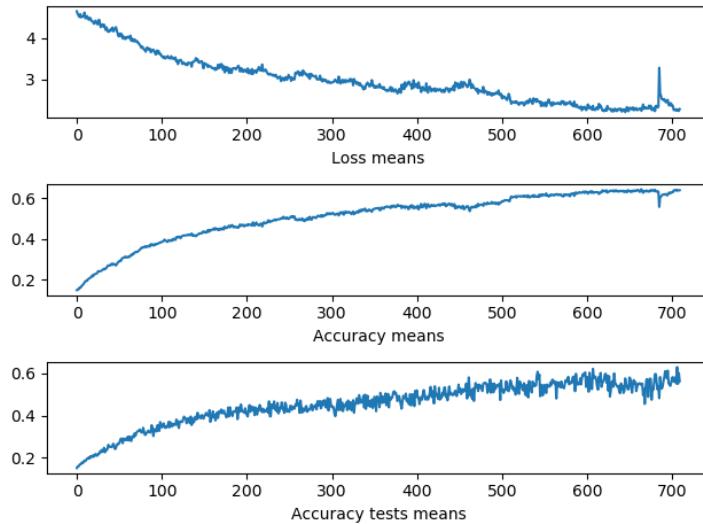


Figura 57: Evolución de las medias de las métricas medidas para la clasificación de 7 clases.

Como se puede observar se llega a una precisión de en torno al 60 %, recordamos que la precisión aleatoria para la clasificación de 7 clases es del 14 %.

Sin embargo en este entrenamiento no se aprecia una estabilización del aprendizaje pues este es siempre creciente. Esto significa que el entrenamiento podría previsiblemente seguir realizándose y consiguiendo mejores resultados. Debido a que con este proceso queremos estudiar sólo el comportamiento de la red decidimos parar el entrenamiento en este punto para ahorrar tiempo pese a poder haber obtenido resultados ligeramente mejores.

4.6.3. Evolución de las predicciones

El comportamiento de las predicciones es exactamente el mismo que en los entrenos de las secciones 4.4 y 4.5. En la figura 58 se observa la matriz de confusión para el *batch* 200 del *epoch* 35 donde se observa que pese a comenzar a agruparse predicciones en la diagonal principal, se concentran gran cantidad de predicciones en una etiqueta aleatoria.

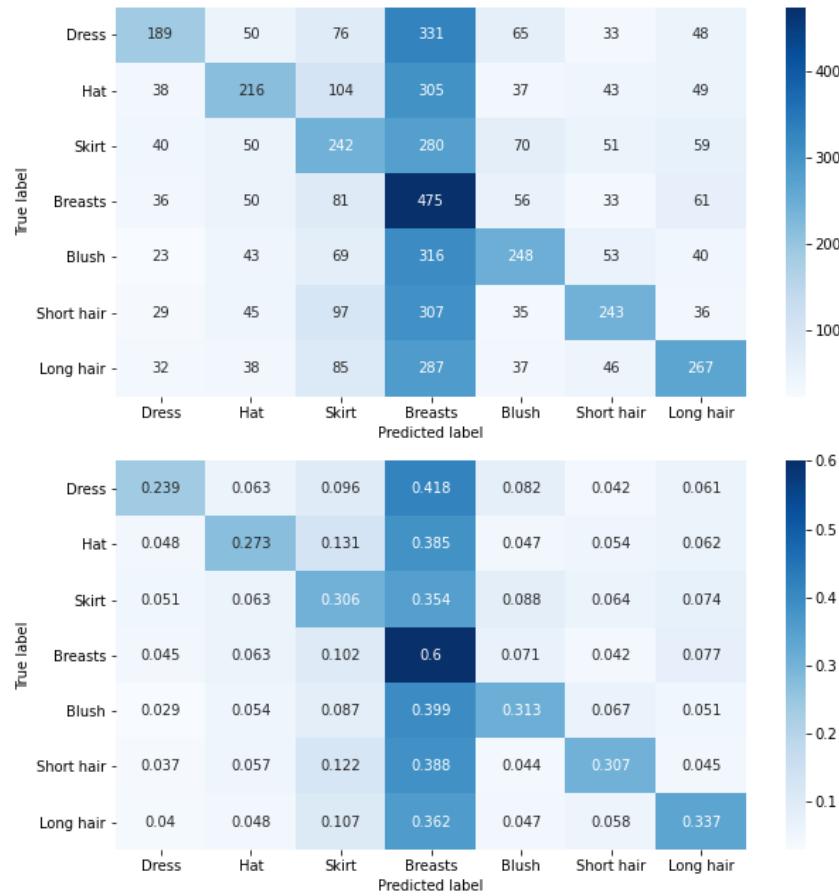


Figura 58: Matriz de confusión para el *batch* 200 del *epoch* 35.

La figura 59 contiene la matriz de confusión para el estado del entrenamiento más avanzado que tenemos, en el *batch* 150 del *epoch* 710, en la figura se puede observar cómo se ha avanzado en las predicciones pero aún hay cierto grado de aleatoriedad, agurpándose gran cantidad de resultados en la etiqueta de *breasts*.

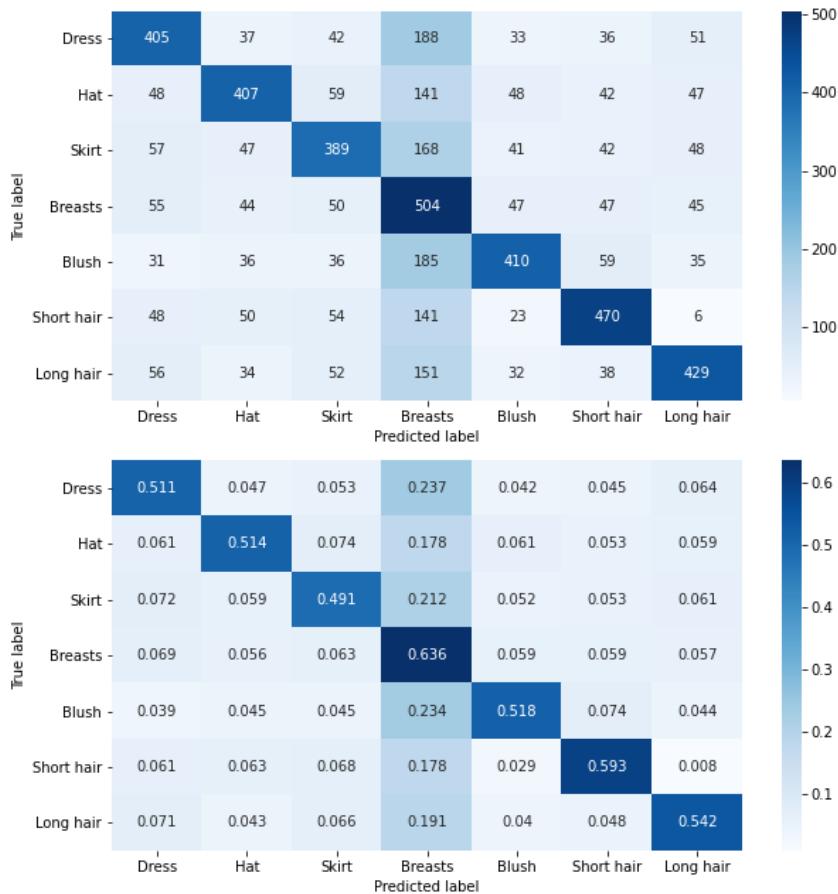


Figura 59: Matriz de confusión para el *batch* 150 del *epoch* 710.

4.6.4. Predicciones sobre imágenes

En la figura 60 se puede observar la predicción del modelo para una imagen con una chica que tiene el pelo largo, está ruborizada y tiene falda. Con las salidas se observa cómo la red se activa para la etiqueta de rubor y pelo largo, pues identifica ambos elementos en la imagen, sin embargo la etiqueta más activa es la falda ya que es el elemento más significativo de estos 3. Esto supone la identificación de los 3 elementos pues son los más activados mientras que el resto que no están presentes en la imagen apenas son activados en comparación. Esto supone la clasificación de una imagen de elementos muy complejos que constata los buenos resultados del modelo clasificador.



Figura 60: Resultados de la predicción de una imagen de una chica ruborizada, pelo largo y falda.

Impacto social y medioambiental

5.1. Impacto del proyecto

Debido al ámbito en el que nos encontramos el proyecto no tiene ningún impacto directamente medioambiental o social. Sin embargo los efectos indirectos del desarrollo del presente TFG sí son notables.

Desde el punto de vista social durante la última década se ha puesto en entredicho las funciones que tiene la informática, en concreto el desarrollo de sistemas de inteligencia artificial como el nuestro despierta un gran interés en el público general, en parte debido a creencias en que la inteligencia artificial podrá superar al ser humano. Sin embargo los proyectos actuales tienen un enfoque alejado de este objetivo teniendo como fin la resolución de tareas concretas y no la sustitución de la mente de una persona.

Con los resultados del trabajo se consigue dotar a una máquina de la capacidad de reconocer formas y patrones en imágenes lo cual pone en entredicho la creencia general de que la visión artificial está muy alejada de la humana pues se confirma que a través de inteligencia artificial se puede conseguir procesar un gran número de imágenes como lo haría una persona, pero a mayor velocidad gracias a ser un proceso automático. La aplicación de la inteligencia artificial del trabajo podría sustituir el etiquetado manual de ilustraciones de cómic, convirtiéndolo en un proceso automático con una gran tasa de acierto.

Los proyectos de visión artificial como el nuestro podrían suponer la sustitución de tareas que actualmente desarrollan humanos, resultando en la pérdida de empleos gracias a la automatización del trabajo. Sin embargo las tareas que son reemplazadas con sistemas como el nuestro son muy simples y tediosas para una persona y no suponen en ningún caso la sustitución completa de un trabajo, simplemente la sustitución de tareas repetitivas del mismo. En concreto el etiquetado de las imágenes originales de nuestros conjuntos de datos se realiza manualmente por los usuarios de la web donde se alojan las ilustraciones, con la aplicación del sistema desarrollado se conseguiría simplificar dicho proceso y aligerar las funciones manuales del usuario.

Desde el ámbito medioambiental el proyecto no tiene un gran impacto, sin embargo hay que tomar en cuenta el consumo de energía y recursos tecnológicos necesarios para el entrenamiento e implementación de una inteligencia artificial, pues los sistemas de visión artificial necesitan de gran capacidad de cómputo para su funcionamiento. A su vez el almacenamiento del *dataset* generado con el trabajo supone la ocupación de grandes volúmenes de datos.

En conclusión el impacto del proyecto en concreto no es muy grande, sin embargo el conjunto de proyectos del mismo ámbito puede tener gran impacto tanto en la sociedad como en el medioambiente.

Líneas de investigación

6.1. Posibles líneas de investigación

Tras la realización del trabajo se observan dos tipos de posibles avances en el proyecto. Intentar conseguir mejorar los resultados obtenidos para mejorar y optimizar el clasificador obtenido por otro lado el objetivo del trabajo es dejar una base teórica y un estudio previo para en un segundo trabajo de fin de grado realizar una inteligencia artificial capaz de generar imágenes.

6.2. Línea de investigación principal

Como se ha indicado durante todo el trabajo, el objetivo final de este es en un futuro poder realizar redes generadoras adversarias para la creación de cómics gracias al uso de la inteligencia artificial. Esto supone la principal posibilidad futura de seguir con el proyecto.

Esta investigación supondría aprovechar todo el estudio y preprocesado del *dataset* generado para usarlo en el entrenamiento de las redes GAN del futuro trabajo. Por otra parte los modelos creados podrán ser reutilizados sufriendo ligeras modificaciones para que sirvan como base sobre la cual construir las redes necesarias en el proyecto futuro.

Para todo esto nuestro trabajo actual es crucial pues su objetivo final está directamente relacionado con futuras posibilidades de investigación.

6.3. Líneas de investigación secundarias

En cuanto al presente trabajo hay grandes posibilidades de optimización de nuestros resultados en la parte de la creación de modelos clasificadores. Como posibles investigaciones futuras se podría replantear cambiar los hiperparámetros de las diferentes redes para que a través de ese *tunning* se consigan mejores resultados haciendo uso de la arquitectura de entrenamiento actual.

Por otra parte se encuentra como posibilidad hacer cambios en las imágenes usadas como aumentar o reducir la dimensiones para comparar y conseguir mejores resultados o también volver a añadir los canales de colores a las imágenes y observar si eso afecta a nuestros resultados.

Conclusiones

7.1. Conclusiones generales

Este trabajo ha supuesto la consolidación de los conocimientos adquiridos durante toda la vida académica. No sólo han sido importantes los conocimientos técnicos sino la metodología a seguir para realizar un proyecto de calidad y minimizar los riesgos. En ese sentido con este proyecto se ha podido poner en práctica todos los conocimientos relativos a una ingeniería obtenidos durante la carrera para obtener un resultado.

Por otra parte todo el periodo previo al desarrollo del trabajo durante el cual se adquirieron los conocimientos técnicos necesarios relativos a la inteligencia artificial fue un periodo muy importante, pues durante el trabajo se ha observado cómo sin aquel estudio previo nada de esto hubiese sido posible pues sin una base sólida no se pueden obtener buenos resultados. Asimismo cada vez que se ha encontrado un problema durante el desarrollo se ha aprendido que es mejor parar e informarse de qué puede estar causando ese problema que tomar una decisión rápida y posiblemente precipitada y errónea.

El desarrollo ha sido plenamente satisfactorio y pese a haber épocas en las que no se obtenían resultados y eso pudo ser frustrante, dichos contratiempos se pudieron solucionar llegando a un resultado que a todas luces supera las expectativas buscadas en un principio. El proyecto al surgir desde cero ha supuesto una buena toma de contacto con el “mundo real” pues los tutoriales seguidos antes de realizar el trabajo siempre funcionaban ya que estaban diseñados para que funcionasen, sin embargo en ningún momento había certeza de la posibilidad de éxito para esta aplicación concreta de la inteligencia artificial.

En cuanto a la metodología seguida en un principio se pensaba que el proceso sería más automático y el número de decisiones importantes sería menor, sin embargo se ha observado la importancia de cada detalle que antes podía pasar desapercibido como el paso a blanco y negro de las imágenes del *dataset* que podía suponer perdida de información relevante para la clasificación buscada. Cada detalle se ha observado que es crucial para el resultado final.

El trabajo supone un gran logro personal consiguiendo realizar un proyecto complejo de una manera correcta obteniendo resultados válidos, y gracias a este proyecto se considera que se ha aprendido cómo funciona en reglas generales la inteligencia artificial, aumentando la curiosidad por el campo y planteando la posibilidad de seguir formándose en él.

Referencias

- [1] Alan Turing. “Computing machinery and intelligence-AM Turing”. En: *Mind* 59.236 (1950), pág. 433.
- [2] David H Hubel y Torsten N Wiesel. “Receptive fields of single neurones in the cat’s striate cortex”. En: *The Journal of physiology* 148.3 (1959), págs. 574-591.
- [3] Kunihiko Fukushima. “Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position”. En: *Biological cybernetics* 36.4 (1980), págs. 193-202.
- [4] Marvin Minsky. *Society of mind*. Simon y Schuster, 1988, pág. 29.
- [5] Dan Claudiu Ciresan y col. “Flexible, high performance convolutional neural networks for image classification”. En: *Twenty-Second International Joint Conference on Artificial Intelligence*. 2011.
- [6] Ian Goodfellow y col. “Generative adversarial nets”. En: *Advances in neural information processing systems*. 2014, págs. 2672-2680.
- [7] Tero Karras, Samuli Laine y Timo Aila. “A style-based generator architecture for generative adversarial networks”. En: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, págs. 4401-4410.
- [8] Aristóteles. *tratados de lógica: El Órganon*. Editorial Porrúa, pág. 534. ISBN: 9789700726632.