UNIVERSITAT POLITÈCNICA DE VALÈNCIA

etsinf

Escola Tècnica Superior d'Enginyeria Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Detección por imágenes de menores en actitudes inadecuadas

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática

Autor: Celia Gómez Sancho

Tutor: Carlos David Martínez Hinarejos

Curso 2019-2020

Resum

En aquest treball hem plantejat un sistema per detectar situacions inapropiades de menors d'edat en imatges, tant en situacions de violència com en situacions sexuals. Per aconseguir això treballarem amb tres xarxes neuronals. La primera s'empra per aconseguir distinguir persones adultes de persones menors d'edat en les diferents imatges. Una vegada obtingut això passarem a la detecció de situacions inapropiades. Finalment s'ha diferenciar si es troba en una situació de sexualització. Per poder fer realitzar això hem recopilat diferents conjunts de dades que compleixin els requisits plantejats. L'objectiu d'aquest treball és detectar imatges tants en xarxes socials com a internet de menors d'edat en situacions inapropiades, i que després aquestes imatges puguin ser retirades.

Paraules clau: Detecció d'images, xarxa neuronal, aprenentatge profund, violència, sexualitzar, menors d'edat

Resumen

En este trabajo hemos planteado un sistema para detectar situaciones inapropiadas de menores de edad en imágenes, tanto en situaciones de violencia como en situaciones sexuales. Para conseguir esto trabajaremos con tres redes neuronales. La primera de ellas se usa para distinguir a personas adultas de personas menores de edad en las diferentes imágenes. Una vez obtenido esto pasaremos a la detección de situaciones inapropiadas. Por último, se trata de diferenciar si se encuentra en una situación de sexualización. Para poder realizar esto hemos recopilado distintos conjuntos de datos que cumplan los requisitos planteados. El objetivo de este trabajo es detectar imágenes tanto en redes sociales como en internet de menores de edad en situaciones inapropiadas, y que después dichas imágenes puedan ser retiradas.

Palabras clave: Detección en imágenes, redes neuronales, aprendizaje profundo, violencia, sexualizar, menores de edad

Abstract

In this project we propose a system to detect inaproppiate situations of children and youngers in images, in violence situations as well as sexual situations. To achieve this we will create three neural networks. The first one will difference between adults and children/youngers in the different images. The next step will be to distinguish inappropiate situations. The last one will be able to distinguish violent situations. In order to do this, we have collected different data sets that fulfil the stated requirements. The objective of this work is to detect images on social networks and on the Internet of children/youngers in inappropriate situations, and then these images can be removed.

Key words: Image detection, neural network, deep learning, violence, sexualization, under age

Índice general

Índice de figuras

Índice de tablas

CAPÍTULO 1
Introducción

Vivimos en una época en la que el mundo de la informática y los sistemas inteligentes están cambiando la forma en la que vivimos. Podemos decir que estamos viviendo una cuarta revolución industrial. Con la inteligencia artificial se nos abren infinidad de posibilidades; éstas las podemos encontrar en el día a día, como puede ser sugerir una nueva canción dependiendo de la música que escuchas, o en las propias redes sociales, que nos ofrecen contenido dependiendo de nuestros gustos. Esta gran tecnología está disponible para todos los públicos y la podemos ver en las manos de los niños y cómo juguetean con ella como si no hubiese ningún peligro en ello. Sin embargo, hay distintos peligros contra los que debemos de luchar.

1.1 Motivación

Las redes sociales se han colado en nuestras vidas y estamos acostumbrados a utilizarlas en nuestro día a día, por lo que se han convertido en un escaparate de todo lo que hacemos. Aplicaciones ya tan necesarias para nuestra comunicación como puede ser WhatsApp también puede ser utilizadas por menores de edad para compartir fotografías y/o vídeos de contenido violento o sexual. Muchos padres y muchas madres se enfrentan a este problema, y aunque intenten controlar todo lo que sus hijos o hijas, comparten en las redes sociales, muchas veces es una tarea muy ardua, y en la mayoría de los casos sus criaturas consiguen salirse con la suya y comparten todo tipo de mensajes. Distintos estudios apuntan a que las personas menores de edad destinan más de tres horas diarias en sus perfiles sociales, en aplicaciones como WhatsApp, Instagram, Facebook o Snapchat [14]. El 50 % de las personas más jóvenes se limita a ejercer el papel de observador en sus perfiles sociales. Sin embargo, el 40 % participa de manera activa a través de mensajes, fotos o vídeos. El dato más alarmante, en cuanto a tipo de contenido publicado, es que un 20 % de menores ha publicado contenido propio de carácter íntimo. Además, un 10 % ha difundido dicho contenido de terceras personas [14].

Además, con la aparición de internet y las redes sociales, han aparecido nuevos tipos de violencia, y se crean nuevos retos virales como puede ser el *happy slapping* (bofetada feliz, en español); es un término que nace en Reino Unido y que se ha ido extendiendo alrededor del mundo durante los últimos años. Este término consiste en la grabación de una agresión, física, verbal o sexual, hacia una persona, que se difunde posteriormente. La agresión suele ser publicada en una red social. En el 61 % de los casos, la agresión proviene del círculo cercano de amistades o colegas académicos [15]. Lo que se percibe como un juego por parte de quien agrede es una grave forma de violencia física.

Gracias a la inteligencia artificial podemos identificar las señales que indican un posible abuso significativo de las nuevas tecnologías y poder evitar que esas personas meno-

res de edad compartan contenido que después puede ser utilizado por terceras personas que pueden poner en peligro su seguridad y su reputación en un futuro.

1.2 Objetivos

Con estas ideas, los objetivos a conseguir con este trabajo son:

- Analizar las posibles situaciones de violencia o sexualización.
- Crear sistemas de redes neuronales capaces de detectar situaciones de violencia o sexualización.
- Combinar los sistemas de detección de menores con los de situaciones de violencia o sexualización.
- Seleccionar datos correctos para la obtención de buenos resultados.
- Explorar las distintas maneras de obtener los mejores resultados.

1.3 Impacto esperado

Con este trabajo se pretende ofrecer una herramienta que pueda ser usada en distintas plataformas, haciendo que contenido sexual o violento no sea publicado. Además, se pretende que sea una herramienta que sirva también a los progenitores con criaturas menores de edad para que que tengan tranquilidad en lo que la seguridad de sus hijos e hijas se refiere, ya que no habrá contenido audiovisual suyo indeseado. También se busca proporcionar una herramienta que proteja a las personas menores de edad y cree una red segura para que no estén bajo amenaza y para que suban contenido propio de su edad. Además, se pretende identificar a aquellas personas mayores de edad que utilizan a menores en sus redes añadiendo contenido no deseado.

1.4 Metodología

La primera tarea para realizar este trabajo ha sido investigar y buscar distintos conjuntos de datos que satisfacieran los distintos requisitos que se estaban buscando, como puede ser imágenes de menores o mayores de edad en situaciones violentas o con objetos violentos. Una vez obtenidos estos datos se han planteado las distintas soluciones que había disponibles, y se ha indagado cómo solucionarlo de la mejor manera para obtener los mejores resultados posibles.

Con todo esto claro, y con una solución planteada, hemos continuado con el desarrollo del trabajo.

1.5 Estructura de la memoria

En este trabajo se puede encontrar una estructura tradicional, en la que tenemos siete capítulos además de la bibliografía al final del mismo.

En el capítulo 2 se ha hecho un análisis del problema del que trata este trabajo; además, lo hemos contextualizado en la época actual, y finalmente hemos explicado las distintas posibles soluciones que se pueden abordar.

En el capítulo 3 se habla sobre la estructura del sistema que hemos implementado, además de especificar el diseño que hemos realizado para tratar el problema.

A continuación, en el capítulo 4 se habla con profundidad de algunas de las tecnologías que se han utilizado para poder desarrollar el trabajo.

En el capítulo 5 se explica el desarrollo del proyecto y cómo se han abordado los distintos problemas con los que hemos trabajado.

En el capítulo 6 se comentan los resultados obtenidos en este proyecto y las conclusiones a las que se llega.

Por último, en el capítulo 7 se habla de trabajos futuros que puede implicar este proyecto, además de la relación que tiene este proyecto con el grado cursado.

CAPÍTULO 2
Problema

2.1 Análisis del problema

Uno de los grandes problemas en la creación de una herramienta capaz de detectar imágenes en las que aparecen menores de edad en situación de violencia y/o sexualización es encontrar datos que podamos usar, ya que son imágenes muy sensibles y que no se pueden, afortunadamente, encontrar fácilmente. Este no es el único problema, si no que encontrar la mejor manera de desarrollar este proyecto también es un problema al que nos debemos de enfrentar, ya que hoy en día existen muchas herramientas que podemos usar.

2.2 Estado del arte

Ahora mismo nos encontramos en un momento en el que las redes neuronales han crecido mucho y siguen haciéndolo. Un uso bastante extendido de las redes neuronales es el reconocimiento en imágenes. Muchas empresas de todo tipo de sectores están incorporando el reconocimiento de imágenes en sus procesos. El reconocimiento de imágenes consiste en identificar un objeto en un vídeo o una imagen. Hoy en día grandes empresas, como puede ser Google, han sacado herramientas y aplicaciones que permiten identificar objetos y hacer búsquedas de objetos similares para poder ofrecer un producto [11].

Esta tecnología nos rodea cada día más, ya que podemos encontrarla en los teléfonos. En éstos podemos encontrar herramientas que organizan imágenes dependiendo de las personas que aparecen en ella, ya sean amistades o familiares. Esto lo hacen detectando si hay personas en las imágenes y si es así comparan los parámetros detectados y si coinciden es que se trata de la misma persona, por lo que agrupan todas las imágenes en las que aparece esa persona en una carpeta. Con esta tecnología se facilita encontrar imágenes si queremos buscar por personas que aparecen.

La clasificación de imágenes se basa en la búsqueda y comparación de patrones en los datos de entrada, pero ésta no es la única utilidad que podemos obtener (y que las empresas han usado), ya que también podemos encontrar cámaras que te hacen una aproximación de tu edad y además te indican tu género. Sobre esto, Microsoft desarrolló en 2015 una página web que es how-old [8], en la que se introduce una imagen y detecta la edad que tiene la persona que aparece en ella. La Figura 2.1 muestra el aspecto de dicha página.

Muchas empresas, entre las que se encuentra Microsoft, no han publicado el algoritmo utilizado. Sin embargo, al tratarse de un tema que ya ha sido tocado por muchas empresas y por muchas personas, podemos ver que una de las herramientas más usa-

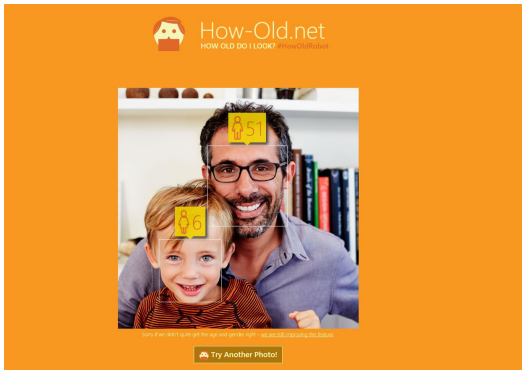

Figura 2.1: Ejemplo de una imagen de la página how-old

das para el reconocimiento de objetos o personas en imágenes son las redes neuronales convolucionales, al tratarse de una herramienta muy poderosa y rápida.

Por otro lado la detección de imágenes se ha aplicado a la detección de situaciones violentas. Un ejemplo de esto es Scylla [9], una inteligencia artificial desarrollada por Develandoo. Este sistema de seguridad es capaz de detectar situaciones potencialmente violentas y avisa a emergencias. Tiene una gran precisión, ya que es capaz de detectar si una persona está apuntando a otra persona (como puede verse en la Figura 2.2), o incluso si un grupo de personas tiene algún elemento que pueda ocasionar daño a otra persona. Esto se hace gracias a la detección de imágenes, ya que va comprobando las imágenes en tiempo real y el comportamiento de las personas que hay, y si cree que hay algún movimiento raro efectúa una llamada a emergencias. No solo eso, si los agentes de seguridad lo califican de un falso positivo, el sistema es capaz de aprender de ello para que no vuelva a pasar. Todo esto es gracias a la inteligencia artificial y a las redes neuronales, que permiten que detecte y aprenda si hay situaciones de peligro o no. Según los desarrolladores, la efectividad de su plataforma para detectar la violencia tiene una precisión del 96 %. Por ello, este sistema de seguridad está siendo usado por escuelas.

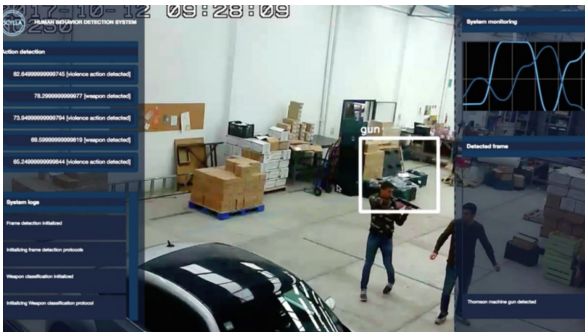

Figura 2.2: Ejemplo de una imagen del sistema de Scylla

Por último, no hemos podido encontrar ninguna herramienta capaz de detectar situaciones de sexualiazción. Esto puede ser debido a la sensibilidad del tema, y aunque las redes neuronales sí que son capaces de detectar estas situaciones no hay ninguna herramienta conocida sobre esto. Lo que sí hemos podido encontrar es un proyecto que quería detectar en imágenes la intención sexual o erótica de una persona en ella. Esto se puede medir con distintos parámetros, como puede ser la posición el cuerpo, la cantidad de ropa que lleva en la imagen, las expresiones faciales...

Por lo tanto podemos observar proyectos que se han realizado en las tres áreas, tanto para reconocer la edad de las personas, como para detectar situaciones violentas e incluso

algún proyecto para detectar personas en situaciones sexuales. Sin embargo, no hemos encontrado ninguna tecnología actual que combine las tres áreas.

2.3 Posibles soluciones

Hay varias maneras de resolver este problema. Vamos a exponer solo dos de ellas relacionadas con la obtención de los datos necesarios. Además, exploraremos algunas de las opciones que vamos a plantear, con sus puntos fuertes y problemáticas.

2.3.1. Un solo conjunto de datos

En cuanto a los datos se pueden diferenciar dos maneras de enfocar el problema. La primera opción sería obtener buenos datos los cuales fueran representativos de todas las situaciones. Sería un conjunto de datos constituídos por una parte de imágenes de menores de edad en situaciones de violencia y/o sexualización, otra parte con imágenes de menores de edad en situaciones propias de su edad, y otra parte de personas mayores de edad en cualquier tipo de situación. Por simplificar, podemos suponer que en cada imagen sólo habría una persona, pero si hubiera más personas podrían concurrir en una misma imagen todas las categorías.

Si lo hiciésemos de esta manera se ve claramente que sería una buena solución, ya que tenemos los datos necesarios que cumplen todo lo que queremos detectar en las imágenes. Sin embargo, encontramos una gran problemática en esta solución, y es la ya comentada, la sensibilidad de estos datos, que no podemos encontrar en internet (al menos en los sitios habituales y legales) por razones entendibles, lo que dificulta mucho reflejar dichas situaciones. Una solución a esto es crear nosotros las imágenes recortando caras de niños y poniéndolas en este tipo de situaciones. Pero incluso haciendo esto se debería de realizar de una buena manera para que el resultado fuese realista.

En cuanto al desarrollo de esta solución, solo tendríamos que hacer una red neuronal, ya que tendríamos los datos diferenciados en las dos situaciones (menores en situación inadecuada y cualquier otra combinación) claramente. Probablemente, la complejidad de esta red neuronal tendría que ser bastante alta, debido a lo complejo que puede llegar a ser detectar las distintas situaciones en una imagen de manera clara.

2.3.2. Dividir el problema inicial en tres

Por otro lado, otra posible solución puede ser analizar cada situación por separado. Es decir, encontrar datos de menores de edad y mayores de edad y detectar de qué tipo se trata. Esto puede hacerse igualmente con situaciones de violencia, donde bastaría encontrar imágenes en las que aparezca situaciones violentas o incluso con armas y, por otro lado, situaciones en las que haya personas en situaciones opuestas, pero sin diferenciar la edad. Por último, sería necesario encontrar imágenes de situaciones en las que aparezcan personas en situaciones de sexualización o no sexualización, y también aquí no hacer diferencia en la edad de la persona (teniendo en cuenta que imágenes de menores sexualizadas es un contenido que no es accesible por motivos éticos).

Esta solución puede ser mejor, puesto que se soluciona la problemática de encontrar datos, ya que aunque siguen siendo de carácter sensible se hace más fácil encontrar unas imágenes en las que aparezcan mayores de edad en esa situación.

En este caso, la problemática la encontramos en el desarrollo de las redes neuronales, debido a que no solo tendríamos que hacer una sino tres distintas, que detectasen cada

situación por separado. En principio, no podríamos usar el mismo modelo en todas si queremos obtener buenos resultados, ya que estaríamos tratando situaciones totalmente distintas y con distinta complejidad, pues no es lo mismo diferenciar si en una imagen aparece un menor de edad o un mayor de edad, que detectar si en la imagen hay una situación violenta. Esto motivaría que la complejidad de cada una de las redes neuronales creadas tendría que ser distinta.

2.4 Solución propuesta

En nuestro caso, hemos optado por la solución en la que se analiza cada uno de los problemas (la detección de una persona menor de edad, una situación violenta y una situación de sexualización) por separado.

Hemos elegido esta solución puesto que obtener los datos era muy complejo, debido a que la sensibilidad de éstos y la complejidad de obtenerlos estaba a la par de crear los nuestros. Sin embargo, sí que vamos a poder unir dos de los casos, el de menores de edad en posesión de armas. Por lo tanto, cuando lleguemos a la creación de una red neuronal que distinga si una persona está en situación violenta, añadiremos datos en los que aparezcan niños en posesión de armas, ya que desgraciadamente esto pasa en la actualidad y sí es fácil encontrar este tipo de datos por internet.

CAPÍTULO 3
Diseño de la solución

3.1 Fase de preproceso de datos

Como ya hemos comentado en el capítulo 2, la solución por la que hemos optado ha sido la de dividir los diferentes problemas para que sea más fácil encontrar imágenes y datos adecuados.

Para poder encontrar dichos datos hemos buscado tanto en imágenes de Google, a través de distintas búsquedas, como en proyectos y bases de datos creadas especialmente para casos muy similares a los estudiados en este proyecto.

En algunas situaciones no hemos encontrados muchos datos para elegir, como es el caso de situaciones de sexualización; por ello hemos optado por una librería de datos que hemos adecuado a nuestro caso, lo cual se explicará más adelante en el capítulo 5.

3.2 Arquitectura de redes usadas

Antes de explicar el tipo de arquitectura usada, se van a describir otros tipos de redes sobradamente conocidas y los motivos por los cuales se han descartado para la realización de este trabajo.

El primer tipo de arquitectura es perceptrón (Figura 3.1). Es el tipo más sencillo, y por esa razón no se ha escogido [6]. Su arquitectura consta con dos celdas de entrada y una de salida y funciona de la siguiente manera: los datos de entrada llegan a la capa de entrada y pasan a la celda de salida, se les aplica la media ponderada, seguida de la función de activación correspondiente, y se devuelve el valor resultante en la capa de salida. La sencillez de este modelo la hace inviable para este proyecto, debido a la complejidad de los datos que vamos a utilizar.

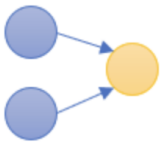

Figura 3.1: Diagrama de una red tipo perceptrón

Por otro lado tenemos el perceptrón multicapa [6] . Este tipo de arquitectura es una extensión del perceptrón, como podemos observar en la figura 3.2, pero aquí se introducen las capas ocultas, mostradas en rojo en la figura. Este tipo de arquitectura consta de

varias capas de neuronas (por lo que puede haber varias capas ocultas) además de una capa de entrada y otra de salida. Todas las celdas de una capa están conectadas con las celdas de la siguiente capa. La información circula de izquierda a derecha. Debido al uso de las capas ocultas y a la posibilidad de que pueda haber varias de ellas se pueden procesar datos más complejos. Sin embargo, también encontramos problemas, ya que puede ser muy difícil entrenar modelos para grandes conjuntos de datos. Por lo tanto, aunque puede llegar a computar datos más complejos, actualmente existen otras opciones más apropiadas para el caso de imágenes, con lo que no es aconsejable usar este tipo de arquitectura para el procesamiento de imágenes.

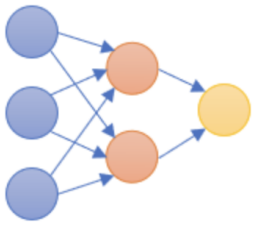

Figura 3.2: Diagrama de un perceptrón multicapa

Por último, llegamos a la arquitectura que hemos usado a la hora de desarrollar las redes neuronales de este proyecto, que son las redes convolucionales, o también denominadas CNN (*convolutional neural network*) [6]. Este tipo de arquitectura es la más usada para el procesamiento de imágenes [6]. Podemos dividir la red en dos bloques distintos, el primero compuesto por las capas convolucionales y las de *pooling*, las cuales tienen como objetivo la identificación de patrones gráficos, y el segundo bloque, que se encarga de la clasificación de los datos que se reciben.

Este tipo de arquitectura, como ya hemos dicho, es la más usada para el procesamiento de imágenes, ya que gracias a su configuración, visible en la Figura 3.3, es más sencilla la detección de objetos en las imágenes. Por este motivo hemos optado por realizar las tres redes neuronales con esta arquitectura.

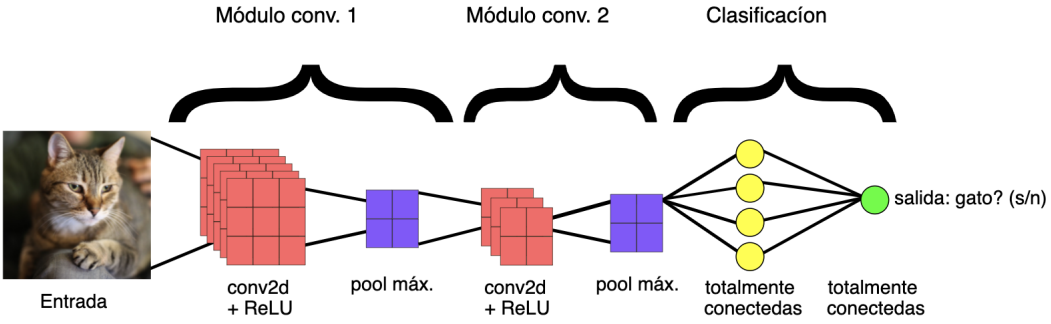

Figura 3.3: Diagrama de una red convolucional

3.3 Combinación empleada

Como ya hemos comentado previamente, para obtener mejores resultados vamos a mezclar algunas de las situaciones. Por lo tanto, vamos a dividir el problema inicial en tres:

- Diferenciar en una imagen a menores de edad y a mayores de edad.
- Detectar en una imagen si existe una situación de sexualización o no.
- Detectar no solo si es una situación de violencia por la postura de las personas, si no que vamos a introducir niños en posesión de armas y en contraposición niños felices o en otras situaciones más propias de su edad.

De esta manera y con esta configuración esperamos conseguir buenos resultados, ya que no solo se estudiaría cada situación por separado si no que se estudiaría en conjunto alguna de ellas.

CAPÍTULO 4
Tecnologías usadas

Para poder desarrollar este trabajo hemos utilizado el lenguaje de programación Python, además de diversas librerías como son Keras [3] y Tensorflow [4]. Además, hemos utilizado redes neuronales para poder llevar a cabo el proyecto, por lo que a continuación vamos a explicar cada una de estas tecnologías

4.1 Python

Python es un lenguaje de programación interpretado, dinámico y multiplataforma. Es un lenguaje que soporta orientación a objetos y cuya filosofía se basa en obtener un código legible que ayude a la programación con una sintaxis sencilla.

En los últimos años el lenguaje se ha hecho muy popular debido a diferentes motivos. El primero de ellos es la gran cantidad de librerías que tiene, además de los tipos de datos y funciones incorporadas; esto ayuda a los programadores a realizar muchas tareas de forma fácil y sencilla sin tener que programarlas desde el principio. En este trabajo hemos utilizado las librerías `cv2`, `tensorflow` y `keras`.

Otro motivo es la sencillez con la que se puede programar, ya que una función en Python puede contener muchas menos líneas que en otros lenguajes como Java o C. Además, es compatible con muchas plataformas, lo que nos permite desarrollar programas en Unix, Windows, OS/2, Mac y otros.

El último motivo es que Python tiene una licencia de código abierto, denominada Python Software Foundation License. Todo esto lo hace una muy buena opción para empezar a programar con él.

4.2 Redes neuronales

Las redes neuronales son un modelo matemático basados en el comportamiento de las neuronas del cerebro humano [6] . Una red neuronal está compuesta por neuronas (nodos), que se interconectan entre sí a través de enlaces, transmitiéndose la información. Los nodos están distribuidos en distintas capas; el número de éstas puede variar dependiendo de la complejidad de la operación que queramos realizar. La información entra por la capa de entrada y atraviesa la red neuronal, generando valores a través de funciones de activación y obteniendo los valores de salida.

Debido a que se necesita una gran cantidad de recursos de un ordenador para entrenar y ejecutar una red neuronal, no han tenido gran éxito hasta hace poco. Una característica muy importante de este sistema, común a otros modelos basados en aprendizaje

automático, es que aprende por sí mismo en lugar de ser programado, por lo que han supuesto una gran ayuda para el reconocimiento de patrones complejos.

Cada neurona está conectada con otras a través de unos enlaces. En estos enlaces el valor de salida de la neurona anterior es multiplicado por un valor de peso. Estos pesos en los enlaces pueden incrementar o inhibir el estado de activación de las neuronas adyacentes. Además, a la salida de la neurona podemos añadir una función que modifica el valor resultado o que impone un límite que no se puede superar antes de pasar a otra neurona; esta función se denomina función de activación.

Perceptrón [6] es el tipo de neurona más clásico; puede tener varias entradas con un peso cada una, proporcionando un valor de salida binario. Este valor se obtiene de la multiplicación de las distintas entradas por cada peso, que indican cómo de importante es cada dato. Los resultados de estas operaciones se suman, añadiendo un *bias* (sesgo), que nos indica la dificultad de obtener un uno en la salida, (si el valor es grande será muy sencillo obtener un uno en la salida). Si el resultado es mayor que un determinado número (umbral de activación), la salida es un uno; si no es así, la salida es cero. La Figura 4.1 muestra gráficamente este proceso para un sistema perceptrón con cinco unidades de entrada.

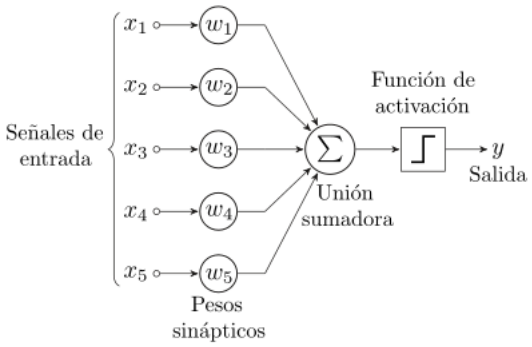

Figura 4.1: Perceptrón con cinco unidades

Como hemos podido ver, perceptrón nos ayuda a analizar unos valores de entrada y tomar decisiones a partir de los resultados obtenidos durante toda la red neuronal, y aunque sí que es verdad que podemos crear redes más complejas añadiendo más capas, tiene el problema de que si variamos un poco los pesos o los *bias* podemos cambiar mucho el resultado, cosa que no queremos. Lo que se desea es poder ir añadiendo pequeñas modificaciones y así poder modificar el comportamiento de la red, por lo que vamos a utilizar otro tipo de redes neuronales más modernas como las redes neuronales convolucionales.

Las llamadas CNN son un tipo de redes neuronales que contienen varias capas ocultas. Estas capas ocultas se van especializando y van aprendiendo hasta llegar a las capas más profundas que reconocen formas más complejas, como puede ser el rostro de una persona o la silueta de distintos objetos como armas. A continuación vamos a explicar con más detenimiento cómo funcionan estas redes.

Estas redes se componen por la capa de entrada, de salida y las capas ocultas. En estas últimas es donde se lleva a cabo gran parte del cálculo de la red neuronal. La última de estas capas se conecta con la capa de salida, donde tenemos una neurona por cada posible resultado de salida.

Como en todas las redes neuronales, hay una parte de pre-procesamiento en la que, antes de pasarle datos a la red neuronal, tenemos que normalizar los valores. Una vez hecho esto haremos las llamadas convoluciones, que consisten en tomar grupos de píxeles e ir operando con una pequeña matriz denominada kernel. Este kernel va recorriendo

toda las neuronas de entrada y obtiene valores de salida que conformarán nuestra nueva capa de neuronas ocultas, como podemos ver en la Figura 4.2.

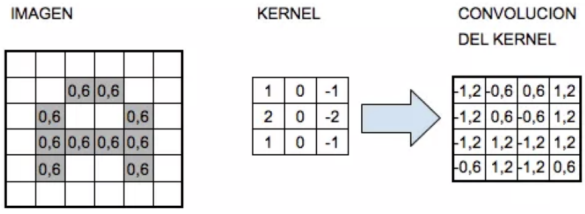

Figura 4.2: Demostración de cómo funciona el kernel

A continuación vamos a explicar la función de activación ReLu, que es la que usamos en este proyecto. Usamos esta función debido a que cuando procesamos una imagen los valores negativos que se producen en una capa no son importantes, por lo que se establecen en 0; sin embargo, los valores positivos sí que lo son, por lo que queremos que pasen a la siguiente capa, y si usamos las funciones sigmoide o tangente hiperbólica (tanh) esta información se pierde. Con esto, podemos expresar matemáticamente la función ReLu

Cuando procesamos una imagen, cada capa de convolución debe capturar algún patrón en la imagen y pasarlo a la siguiente capa de convolución. Los valores negativos no son importantes en el procesamiento de imágenes y, por lo tanto, se establecen en 0. Pero los valores positivos después de la convolución deben pasar a la siguiente capa. Es por eso que ReLu se utiliza ampliamente en proceso de imágenes como una función de activación. Si utilizamos sigmoide o tanh, la información se pierde, ya que ambas funciones modificarán las entradas a un rango muy cerrado.

Después de aplicar esto realizamos una reducción de la cantidad de neuronas, ya que al realizar una convolución la capa que obtenemos tiene todavía una gran cantidad de neuronas, y si siguiésemos sin realizar esta reducción el número de neuronas sería muy grande y eso implicaría mayor procesamiento, además de que no se recogería la información discriminativa apropiadamente. Una vez realizado esto se repite el proceso tantas veces como consideremos que requiere la complejidad de nuestro problema.

Queda por explicar dos tipos de funciones de reducción como son la *max pool* y *average pool*. El *pooling*, también denominado *pool*, reduce la altura y anchura de los datos de entrada de activación. Esto no solo ayuda a reducir la computación, sino que también contribuye a aprender características de invariancia espacial. Hay dos tipos principales de *pooling*.

En el caso de *max pool* se recorre la entrada usando una ventana de n x n y se elige el máximo valor dentro de la ventana. En cambio, *average pool* usa el mismo tipo de ventana sobre los datos de entrada pero realiza una media sobre ellos. Las ventanas realizan el recorrido según un valor de desplazamiento (*stride*) de un cierto número de píxeles en cada dimensión. La Figura 4.3 muestra un ejemplo de los tipos de *pooling* descritos para una ventana de 3x3 y un *stride* de 3x3 píxeles.

Antes de llegar a la capa de salida tenemos que aplicar una función para que se encargue de pasar los cálculos realizados durante toda la red neuronal a las neuronas de salida.

Al entrenar nuestro modelo queremos saber qué valores de los pesos y los *bias* van a conseguir que se minimice el coste lo máximo posible. Para optimizar estos valores podemos utilizar diferentes algoritmos, como puede ser el algoritmo de descenso por gradiente [2].

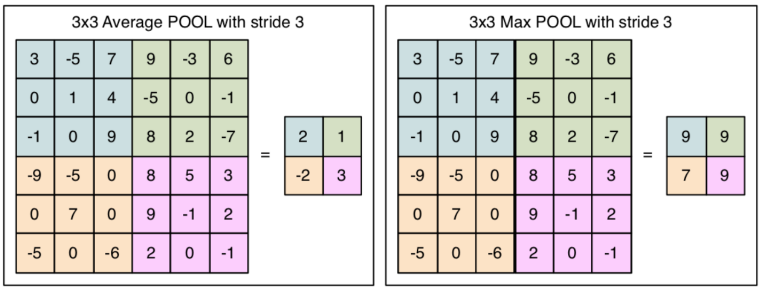

Figura 4.3: Operación de *pool* (*average pool* a la izquierda, *max pool* a la derecha) con una ventana de 3 por 3 y un *stride* de 3 por 3

Para poder poder explicar estrategias de optimización más elaboradas primero vamos a explicar en qué consiste el descenso por gradiente. Esta función consiste en elegir un punto aleatorio de nuestra función de error de salida y, mediante el cómputo de derivadas, vamos variando este valor para ir encontrando los mínimos locales e ir mejorando nuestro error. Pero no solo hay que tener en cuenta esos valores, sino que debemos de añadir el ratio de aprendizaje. Este es un valor muy importante, ya que define cuánto afecta el gradiente a la actualización de nuestros parámetros en cada iteración. Este valor define el comportamiento de nuestro algoritmo, ya que si elegimos un valor pequeño nuestro punto se irá aproximando muy lentamente a la zona del mínimo, lo que lleva a un gran número de operaciones; además, cabe la posibilidad de que el punto se quede atrapado en un punto local ineficiente. Por otro lado, si elegimos un valor grande puede hacer que nuestro punto dé grandes saltos y nunca llegue a alcanzar un mínimo local ya que no lo detecta; esto también puede causar que el proceso de optimización quede en un bucle infinito. Todo este proceso queda representado en la Figura 4.4 Hay diferentes técnicas que sirven para ajustar este parámetro de forma dinámica.

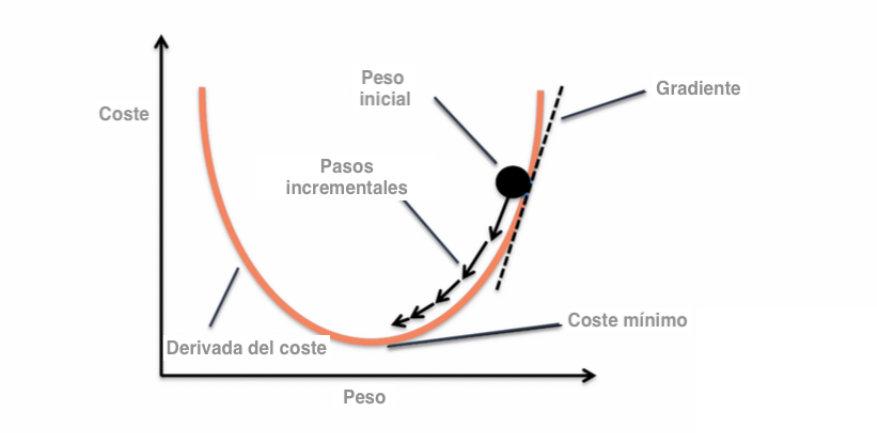

Figura 4.4: Descenso por gradiente

El objetivo de un optimizador es minimizar el valor de una función de pérdida usando un algoritmo iterativo. Obtener buenos valores de mínimo local puede llegar a ser difícil.

A veces el descenso por gradiente trae consigo diferentes problemas como puede ser

- *Overfitting* (sobreentrenamiento): consiste en que el modelo memoriza los ejemplos muy bien pero generaliza muy mal.
- El descenso por gradiente es muy lento: como ya hemos comentado, si la actualización de nuestro valor es muy lenta hace que se compute muchos cálculos.
- *Local minima*: debido a que no sabemos si el valor obtenido es el mínimo global podemos obtener un mínimo local pensando que es el global pero que no sea así, y esto puede hacer que no obtengamos el mejor valor posible.

Existen diferentes alternativas al optimizador por descenso por gradiente original. Es el caso de las funciones de optimización Adam o RMSProp [6].

Vamos a explicar con más detenimiento la función de optimización de Adam. Adam es uno de los optimizadores más populares, ya que combina ideas de otras funciones de optimización como son RMSProp y *momentum* [6]. Adam produce un entrenamiento efectivo, evaluando las siguientes cantidades para realizar las actualizaciones de parámetros:

- Promedio ponderado exponencialmente de gradientes pasados almacenados en v y $\bar{v}$, donde v es el diccionario que contiene estimaciones actuales del primer momento, (con corrección del *bias*).
- Promedio ponderado exponencialmente de los cuadrados de los gradientes pasados almacenados s y $\bar{s}$, donde s es diccionario que contiene estimaciones actuales del segundo momento, (con corrección del *bias*).

4.3 TensorFlow

TensorFlow es una librería de código abierto, desarrollada por Google Brain, que ayuda a implementar y entrenar modelos de redes neuronales de manera sencilla, ya que ofrece una gran variedad de modelos y algoritmos [4].

Esta librería es compatible con distintos lenguajes de programación, entre los que se incluye Python, que es el que hemos usado nosotros; esto hace que su uso sea muy sencillo.

TensorFlow es hoy en día una de las librerías más usadas y conocidas en el campo de *deep learning* debido a que se puede usar en múltiples áreas como ayudar al diagnóstico médico, mejorar la fotografía en los dispositivos móviles o procesamiento de imágenes, entre otros. Todo esto se realiza de forma rápida y eficiente.

4.4 Keras

Keras es una librería de Google para redes neuronales capaz de ejecutarse en varios entornos como TensorFlow. Está creada para ayudar a la hora de explorar con las redes neuronales, ya que es muy fácil de entender y usar. Esto es debido a que hacen mucho más sencilla la forma de interactuar a la hora de desarrollar nuevos modelos de aprendizaje profundo.

Keras también contiene soporte para las redes neuronales convoluciones y recurrentes y contiene una API con la que podemos crear modelos y depurarlos, además de crear modelos dinámicos con herramientas estándar.

Como ya hemos comentado, una de las características que nos ofrece la librería Keras es que tiene implementadas muchas herramientas que nos ayudan a crear y mejorar las redes neuronales que creemos. A continuación se van a explicar algunas de esas herramientas que se han usado.

Lo primero que hemos empleado ha sido el tipo de modelo que hemos usado para crear nuestro modelo. Éste ha sido *secuencial*, el tipo de modelo más simple, ya que nos permite ir agregando capas en secuencia, lo que nos permite añadir los distintos tipos de capas que queramos en el orden que consideremos; además, nos permite jugar con estas capas hasta encontrar la mejor combinación.

Por otro lado hemos utilizado distintos *layers*. Aquí hemos hecho uso de una gran cantidad de éstos para poder ir añadiendo las distintas capas a nuestro modelo. La primera que hemos usado ha sido Conv2D como mostramos en la Figura 4.5 . Con este tipo de capa creamos kernels convolucionales que cogen los datos de entrada y crean los datos de salida, calculados a través de la función de activación que hemos añadido (ReLu en nuestro caso).

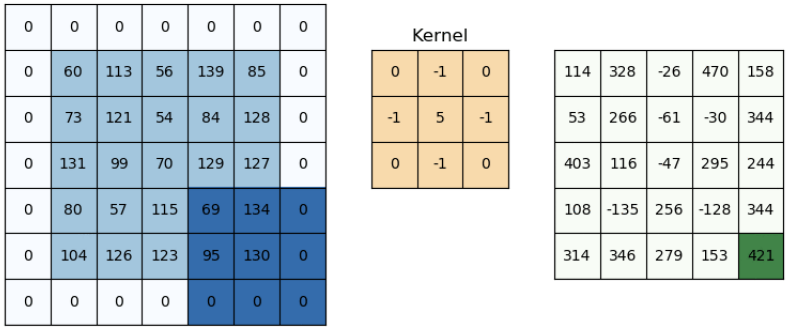

Figura 4.5: Ejemplo fotos de uso de capas Conv2D

Otro tipo de capa que hemos usado ha sido AveragePooling2D, como mostramos en la Figura 4.6. Este tipo de capa se usa para reducir la muestra de entrada. Esta función de reducción la hemos explicado anteriormente en el apartado de redes neuronales (sección 4.2) . De esta manera reducimos los cálculos necesarios para la siguiente capa.

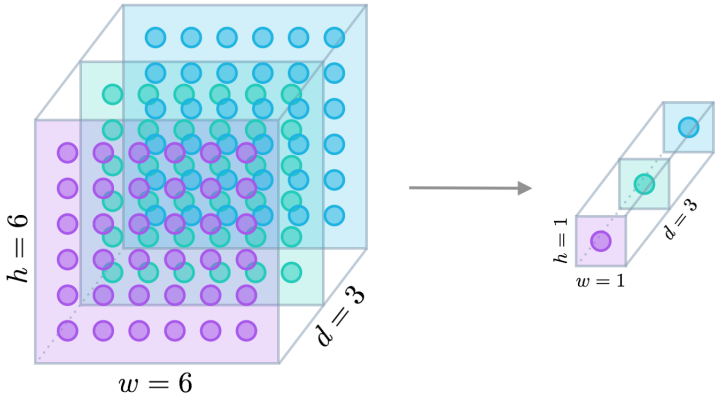

Figura 4.6: Ejemplo fotos de uso de capas AveragePooling2D

En nuestra red neuronal hemos optado por usar el optimizador Adam, el cual hemos explicado en el apartado 4.2.

La última función que hemos utilizado es Dropout; lo que hace es desconectar aleatoriamente un número de neuronas para generalizar para un número mayor de muestras.

Esta técnica se utiliza debido a que hay veces que sobreentrenamos nuestra red neuronal, cosa que no que queremos ya que los resultados obtenidos no serán igual de precisos.

Además de todas estas funciones, la librería Keras ofrece un conjunto de modelos ya creados. Hemos utilizado VGG16 [5] ; este modelo contiene un conjunto de pesos ya preentrenados con ImageNet. El uso de estos modelos con pesos ya preentrenados ayuda a la hora de entrenar la red, ya que obtenemos mejores resultados y además más rápido.

CAPÍTULO 5
Desarrollo

5.1 Obtención de datos

Hemos dividido el proyecto en tres redes neuronales para poder atacar cada problema por separado y conseguir mejores resultados. A continuación describimos cada uno de esos modelos junto con los conjuntos de datos empleados para su entrenamiento.

Menores de edad

Primero hemos creado una red neuronal que detecte si en una imagen aparece una persona menor de edad o una persona mayor de edad. Lo primero de todo ha sido encontrar los datos adecuados para esto. Esta ha sido una tarea un poco compleja, ya que se han encontrado datos de todo tipo. Primero se intentó hacer esta tarea con imágenes de grupo, pero dado que en una misma imagen podían aparecer menores y mayores de edad era bastante complejo abordar el problema desde este punto. Por tanto, se optó por imágenes en las que apareciese solo una persona. Además, se asumió que estas imágenes se adecuaban más al problema, ya que las imágenes que se suelen subir a los perfiles de redes sociales son de personas solas. En todo caso, para imágenes grupales podría adoptarse una estrategia de segmentar en subimágenes donde hubiera una única persona y, a partir de esa segmentación, aplicar esta aproximación que asume una única persona en la imagen.

Figura 5.1: Ejemplo fotos de la librería UTK

Para este tipo de imágenes hay diversas librerías que se pueden utilizar, ya que es un tema bastante tratado y podemos obtener datos. Al final se optó por utilizar los da-

tos proporcionados por la librería UTK [12], ya que ofrece una gran variedad de fotos. Pueden verse ejemplos de las fotos disponibles en la Figura 5.1 . Además cada foto está etiquetada con la edad de la persona que aparece en ella, por lo que para tratar dichas imágenes se optó por diferenciarlas en dos carpetas distintas: una que contuviese todas las imágenes de menores de edad y otra que contuviese las imágenes de mayores de edad.

Este conjunto de datos está compuesto por 24108 imágenes. De ellas, 19318 pertenecen a personas mayores de edad, y las 4790 imágenes restantes son de personas menores de edad. Al haber tanta diferencia entre los dos conjuntos se decidió acotar los resultados a 4790 en el caso de las imágenes de personas menores de edad y a 4800 en el caso del conjunto de datos de las personas mayores de edad. Por lo tanto, el conjunto final de datos es de 9590 imágenes La resolución y el tamaño de las imágenes varían, ya que no son aspectos controlados. Además, las imágenes son de todo tipo de carácter, es decir, tanto los fondos, como la iluminación, así como la postura de las personas que hay en ellas, varía.

El siguiente paso es guardar todas esas imágenes, por lo que se optó por etiquetar cada imagen. Para ello, creamos una función para que añadiese en un *array* los datos, a fin diferenciar si en una imagen aparecía un menor de edad o un mayor de edad. Las etiquetas que hemos utilizado son: *menor* y *mayor*.

Acciones sexuales

La siguiente red neuronal que construimos fue una que detectase el carácter sexual de las imágenes. Encontrar datos para esta parte fue bastante difícil por la sensibilidad de las imágenes. Se estuvo investigando cómo podrían obtenerse dichos datos, e incluso se pensó en crear datos propios. Sin embargo , se encontró un directorio que podía adaptarse a esta tarea. Este directorio contiene 1146 imágenes de 203 diferentes personas famosas obtenidas de la página `people.com`. En estas imágenes hay 892 imágenes en las que aparecen mujeres y 245 hombres. 646 imágenes pertenecen a situaciones sexuales, mientras que 500 imágenes pertenecen a situaciones no sexuales. Además, en este conjunto tenemos muchas variables que varían como son: la posición de la persona, la iluminación, la parte de la persona que se ve en la imagen, el sitio en el que se encuentra, etc. Se pueden ver varios ejemplos en la Figura 5.2.

Esta librería proporcionaba una serie de características para detectar la intención de la persona que aparece en la imagen. Una de estas características medía la cantidad de piel que enseña la persona de la imagen.

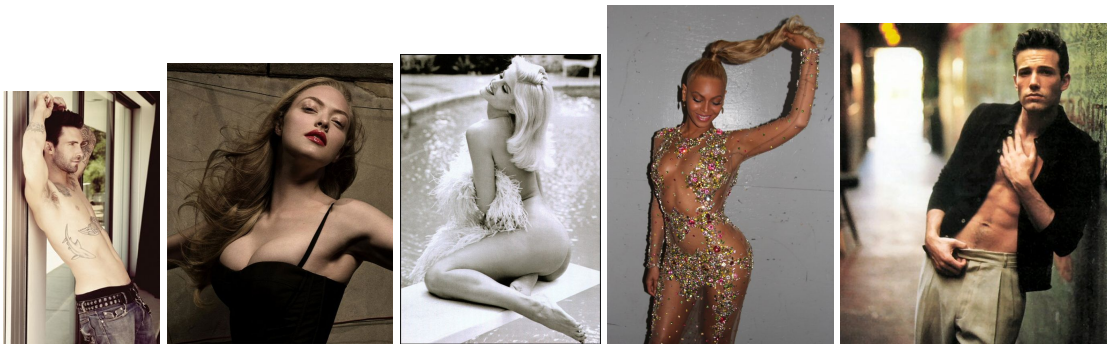

Figura 5.2: Ejemplo de fotos usadas para la detección de actitudes sexualizadas

En esta parte teníamos los datos de otra forma distinta, ya que teníamos un fichero en formato CSV en el que nos indicaba, con una escala (-1, 0, 1), si se trataba de una imagen

en la que no se muestra nada de piel (-1), que la foto era sugerente (0) o definitivamente estaba claro que en esa imagen sí que se muestra partes íntimas (1). Así, se agruparon las imágenes de los conjuntos con etiquetas 0 y 1, y se catalogaron con la etiqueta *sí*, mientras que el otro conjunto de datos está catalogado con *no*.

A la hora de obtener las imágenes he procedido de la misma manera, utilizando la herramienta `cv2`.

Violencia

La última red neuronal que hemos creado ha sido una que detecte si en la imagen hay violencia. Puede que esta búsqueda de datos haya sido la más complicada de las tres debido tanto a la sensibiliad de los datos como a la complejidad de reflejar en una imagen situaciones asociadas a la violencia. Esto es debido a que es más fácil encontrar violencia en un vídeo, ya que refleja mejor si hay violencia contra una persona, debido al carácter dinámico que tienen en general las actitudes violentas. Sin embargo, en una imagen tenemos que detectar ciertas posturas y de ellas decir si esas posturas reflejan violencia o no. Pero además la violencia no solo está en la postura de una persona, si no en la actitud de esta; por lo tanto hemos dividido los datos en dos.

Por una parte hemos cogido imágenes de menores que reflejen una actitud feliz; por otro lado, tendremos menores en actitud triste, o incluso con armas a su alrededor o sujetándola ellos mismos. De esta manera, la red neuronal será capaz de diferenciar estos dos casos. Estas imágenes las hemos obtenido de Google, realizando búsquedas como 'niños felices', 'niños en tenencia de armas'. La Figura 5.3 nos muestra algunos ejemplos de las imágenes obtenidas.

Figura 5.3: Ejemplo de fotos usadas para la detección de actitudes violentas

Por otra parte, para poder plasmar la parte en la que una situación se puede catalogar como violenta, hemos añadidos datos de capturas de vídeos en los que había una situación violenta, como puede ser dar patadas, señalar a alguien, empujando a la otra persona, o incluso pegando a alguien. Por contraposición, hemos elegido situaciones en las que había dos personas hablando una frente a otra, dándose la mano, o dándose un abrazo. Estos datos los hemos obtenido de la librería UT [13] , y pueden verse ejemplos de las imágenes extraídas en la Figura 5.4.

El conjunto de datos con el que se va a trabajar está compuesto por 8266 imágenes, en las que la mitad de ellas corresponden a personas menores de edad en las situaciones descritas anteriormente, y la otra mitad corresponden a personas mayores de edad, igualmente en las situaciones descritas anteriormente. Tanto el tamaño como la calidad de las imágenes varía dependiendo de la imagen. Además, hay imágenes en las que aparecen personas en un primer plano, mientras que hay otras en las que la situación se da más lejos de la cámara, por lo que también varía mucho las posiciones y la iluminación de las imágenes.

Para tratar estos datos hemos procedido de la misma manera que en la parte de diferenciar la edad, creando dos carpetas: una con los datos violentos y otra con las imágenes

Figura 5.4: Ejemplo de fotos usadas para la detección de actitudes violentas

en las que aparecen menores contentos o situaciones no violentas. Hemos etiquetado cada imagen como: *violence*, *non-violence*.

Edad-Violencia

Como ya hemos comentado antes también se ha probado a juntar dos casos, el caso de diferenciar la edad y el caso de diferenciar situaciones de violencia en imágenes. Para poder realizar este caso hemos cogido datos que ya teníamos de las librerías anteriores. Estos datos están compuestos por personas tanto menores de edad como mayores de edad en situaciones de violencia y de no violencia. La Figura 5.5 muestra ejemplos de las imágenes utilizadas.

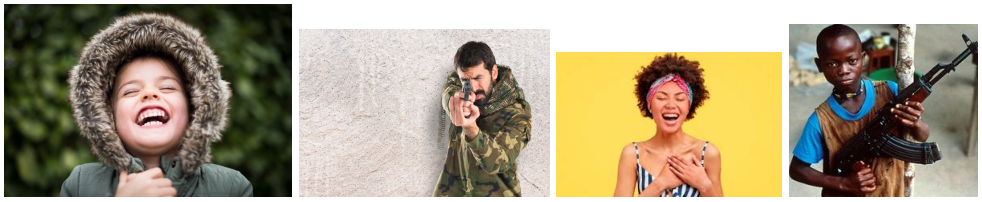

Figura 5.5: Ejemplo de fotos usadas para la detección de actitudes violentas y distinción de personas mayores y menores de edad

El conjunto de datos con el que se va a trabajar está compuesto por 504 imágenes: 116 imágenes corresponden a situaciones violentas en las que aparece un menor de edad, 129 imágenes son de menores de edad en una situación no violenta, 137 corresponden a personas mayores de edad en situaciones violentas y por último 122 son de personas mayores de edad en situaciones no violentas. Pero hay que reagrupar estas imágenes para utilizarlas en ambas redes neuronales, por lo que se obtiene un conjunto de datos en el que 253 imágenes corresponden a situaciones de violencia tanto de personas adultas, como de menores de edad y 251 son imágenes de situaciones no violentas, igualmente de menores y mayores de edad. Por otro lado, este mismo conjunto de datos está compuesto por 245 imágenes de menores de edad en situaciones violentas y no violentas y 259 imágenes de adultos en situaciones violentas y no violentas.

Este conjunto de datos, como ya hemos comentado, está constituido por imágenes de distintos tamaños y las personas que aparecen en ellas están en todo tipo de posiciones y no en todas ellas las personas aparecen en un primer plano, ni la iluminación es la misma.

Por último, queda comentar cómo se va a proceder en la clasificación de los datos. Dado que vamos a juntar los resultados obtenidos en dos redes neuronales vamos a etiquetarlas de una manera distinta. Si las redes neuronales detectan que hay un menor de edad en ella y hay violencia, se etiquetará con (1, 1); si detecta que hay un mayor de edad y hay violencia se etiquetará con (0, 1); si detecta que hay un menor de edad y no hay violencia la etiqueta será (1, 0); y por último, si detecta que es mayor de edad y no hay violencia se etiquetará como (0, 0)

5.2 Convertir las imágenes

Una vez que teníamos todas las imágenes etiquetadas, el siguiente problema con el que nos encontramos era obtener el formato adecuado de las imágenes, ya que para procesarlas necesitamos que todas ellas estén de la misma manera. Además, para poder usarlas no podemos guardarlas con formato de imagen, si no que debemos de guardarlas como una matriz. Todo esto lo hemos hecho gracias a la herramienta `cv2`. Además, hemos guardado todas las imágenes con el mismo formato, RGB y con el mismo tamaño, 128x128 píxeles.

Esto lo hemos hecho en todas las redes neuronales que hemos creado, es decir, en las redes neuronales de edad, violencia y sexualización. El código empleado para la conversión se puede ver en el Listing 5.1.

```
def load_images(data):
    keys = list(data.keys())
    values = list()
    result = []

    for key in keys:
        if data[key] == 'menor':
            imagePath = '/UTKFace/Photos/All_photos/minors/' + key
        else:
            imagePath = '/UTKFace/Photos/All_photos/adults/' + key

        image = cv2.imread(imagePath)
        image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
        image = cv2.resize(image, (128, 128))
        result.append(image)

    return result
```

Listing 5.1: Extracto de código conversión de las imágenes

5.3 Conversor de datos

El siguiente paso es dividir los datos obtenidos y crear datos de entrenamiento y datos de testeo. Para realizar esto hemos decidido usar el 75 % de los datos para el entrenamiento y el 25 % restante para testear.

Por otra parte, la librería Keras nos proporciona un método que es `ImageDataGenerator`. Este método nos permite hacer transformaciones sobre las imágenes que tenemos. De esta manera podemos generar más datos incluso, modificando los que ya tenemos. Podemos rotar las imágenes, voltearlas horizontalmente o verticalmente, redimensionarlas...

Este proceso de *data augmentation* se ha realizado con los parámetros que se muestran en el Listing 5.2. En concreto, se ha reescalado la imagen para normalizar todos los datos,

además de hacer un *zoom* de un 0.2, se ha rotado la imagen 5 grados y por último se ha volteado horizontalmente las imágenes.

```
trainAug = ImageDataGenerator(
    rescale=1. / 255,
    zoom_range=0.2,
    rotation_range = 5,
    horizontal_flip=True)
```

Listing 5.2: Proceso de *data augmentation*

Todos estos procesos los hemos realizado también en las tres redes neuronales. Como resultado, los conjuntos de datos finales presentan las características presentadas en la Tabla 5.1.

Tabla 5.1: Características de los datos finales

Categoría	75 %	25 %	Menores/Sexual/ Violencia	Mayores/No sexual/ No violencia	Total
Edad	7192	2397	4790	4800	9590
Violencia	859	286	646	500	1146
Sexual	6199	2066	4133	4133	8266

5.4 Diseño de los modelos de redes neuronales

Para cada uno de los casos (edad, actitudes sexuales y violencia) hemos creado una red neuronal distinta, como ya hemos comentado anteriormente. Hemos creado un modelo distinto para cada uno de los casos, ya que la complejidad de las imágenes no es la misma. Los resultados obtenidos se describen en el capítulo 6.

Edad

Para el diseño de este modelo hemos probado diferentes distribuciones de las capas ocultas; además hemos ido variando los parámetros del número de neuronas y de capas usadas, y con la distribución mostrada en la Figura 5.7 hemos obtenido los mejores resultados, esta distribución la podemos observar mejor en el esquema gráfico de la Figura 5.6

Como hemos podido observar, las fotos no son muy complejas, ya que lo único que suele aparecer en primer plano es la cara de la persona; sin embargo, sí que hay bastante complejidad en detectar si es un menor de edad entorno a los 12-16 o si es un mayor de edad entre 18-20 años. Esto se verá más específicamente en los resultados que se muestran en el capítulo 6.

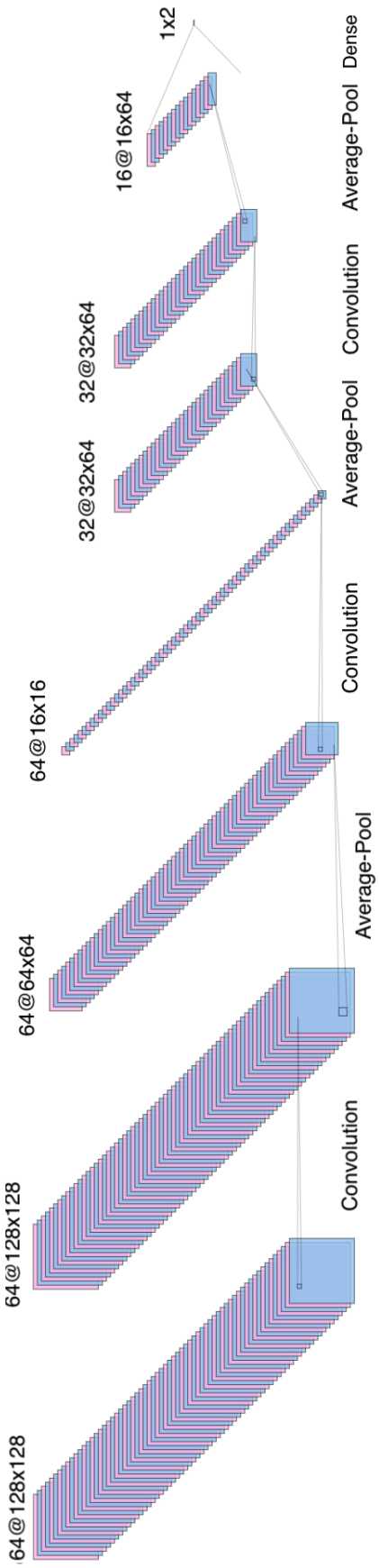

Figura 5.6: Esquema gráfico de la red neuronal para detectar la edad

```
Model: "sequential_7"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_43 (Conv2D)           (None, 128, 128, 64)      1792
_________________________________________________________________
conv2d_44 (Conv2D)           (None, 128, 128, 64)      36928
_________________________________________________________________
average_pooling2d_30 (Averag (None, 64, 64, 64)        0
_________________________________________________________________
conv2d_45 (Conv2D)           (None, 64, 64, 64)        36928
_________________________________________________________________
average_pooling2d_31 (Averag (None, 32, 32, 64)        0
_________________________________________________________________
conv2d_46 (Conv2D)           (None, 32, 32, 64)        36928
_________________________________________________________________
average_pooling2d_32 (Averag (None, 16, 16, 64)        0
_________________________________________________________________
flatten_7 (Flatten)          (None, 16384)             0
_________________________________________________________________
dense_7 (Dense)              (None, 2)                 32770
=================================================================
Total params: 145,346
Trainable params: 145,346
Non-trainable params: 0
_________________________________________________________________
```

Figura 5.7: Modelo de la red neuronal para detectar la edad

Esta red se compone por una capa de entrada, con un tamaño de 128x128, al igual que las imágenes usadas. Además, hemos usado tres capas ocultas, usando como función de activación ReLu, y por último una capa de salida de 2 valores que corresponden a la puntuación de que sea un menor de edad o un mayor de edad; esta última capa tiene la función de activación Softmax.

Acciones sexuales

Al realizar el modelo de esta red neuronal nos encontramos con un gran problema, y es que al realizar una red neuronal cambiando el número de capas ocultas y el número de neuronas usadas no obteníamos buenos resultados. Viendo las imágenes usadas podemos entender el por qué; se debe a la complejidad del problema que intentamos abordar, ya que el conjunto de los datos es muy variado, y lo que intentamos aprender es muy complejo, ya que es que detecte la cantidad de piel que aparece en la imagen.

Por todo eso se utilizó el modelo que podemos encontrar en la librería Keras; se trata del modelo VGG16. Al utilizar este modelo los resultados mejoraron notablemente, y esto es debido a que el modelo posee un conjunto de pesos ya preentrenados, y esto ayuda a la hora de entrenar la red neuronal.

Esta red contiene una capa de entrada, con un tamaño de 128x128, al igual que las imágenes usadas. Además, posee una gran cantidad de capas ocultas, lo que ayuda al aprendizaje. Por último, hemos añadido una capa de *dropout* y la capa de salida con función de activación Softmax. Todo esto lo podemos ver en la Figura 5.9. También podemos observar el esquema gráfico de esta red neuronal en la Figura 5.8.

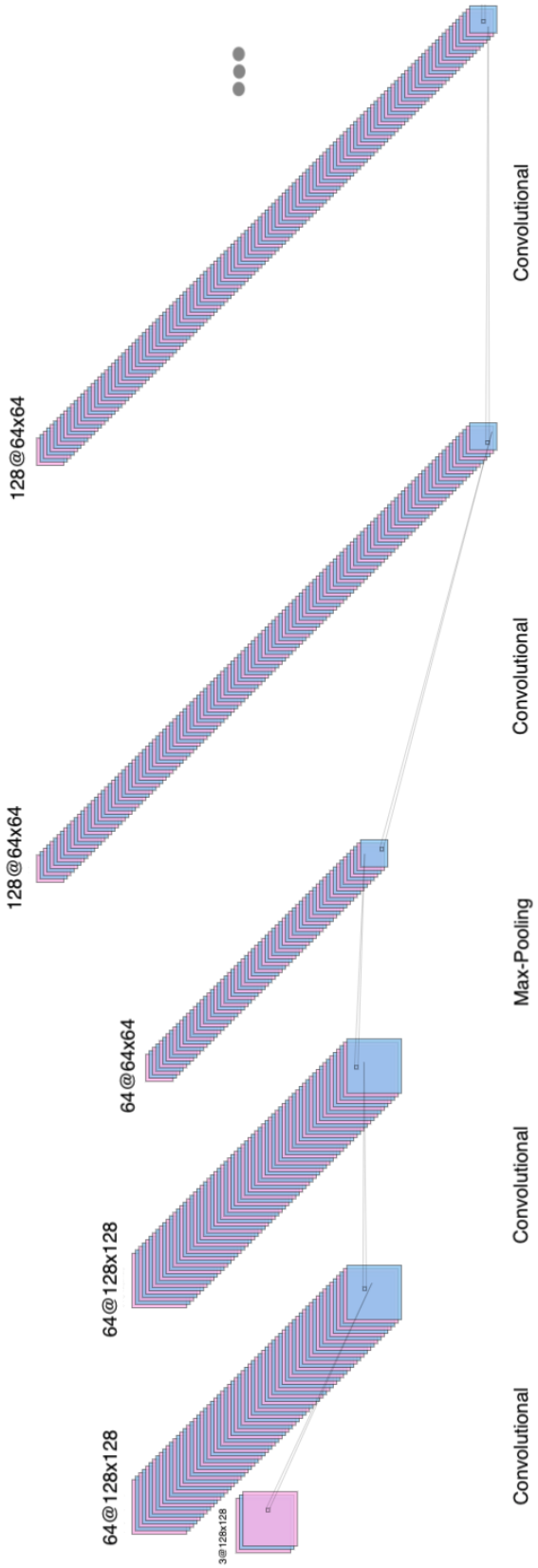

Figura 5.8: Esquema gráfico de la red neuronal para detectar sexualización

Violencia

Al realizar el modelo de esta red neuronal nos encontramos, igual que con el caso anterior, que la complejidad de los datos es muy grande, dado que no solo hay un tipo de imagen (niños felices, niños tristes o con armas), si no que además añadimos la complejidad de que en el mismo modelo metemos otro tipo de datos, como puede ser el detectar las posturas de una persona y comprobar si estas son situaciones de violencia o no. Por todo esto hemos procedido de la misma manera que en la red neuronal anterior y se ha optado por utilizar el modelo dado por la librería Keras VGG16.

La distribución del modelo de esta red neuronal la podemos ver en la Figura 5.9

5.5 Aprendizaje y entrenamiento de las redes neuronales

En todas las redes neuronales hemos procedido de la misma manera a la hora de entrenarlas. Se ha entrenado el modelo utilizando el optimizador de Adam, que ya explicamos en el capitulo 4.

Además, como ya hemos dicho, le pasamos el 75 % del conjunto de datos que tenemos. Cada muestra contiene la imagen que queremos que la red neuronal aprenda. El código de entrenamiento empleado se puede ver en el Listing 5.3.

```
1
2 model.compile(loss="binary_crossentropy", optimizer=opt, metrics=["accuracy"])
3
4 H = model.fit_generator(
5     trainAug.flow(X_train, y_train, batch_size=32),
6     steps_per_epoch=len(X_train) // 32,
7     validation_data=valAug.flow(X_test, y_test),
8     validation_steps=len(X_test) // 32,
9     epochs=epochs)
```

Listing 5.3: Código de entrenamiento de las redes empleadas.

```
Model: "vgg16"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_6 (InputLayer)         (None, 128, 128, 3)       0
_________________________________________________________________
block1_conv1 (Conv2D)        (None, 128, 128, 64)      1792
_________________________________________________________________
block1_conv2 (Conv2D)        (None, 128, 128, 64)      36928
_________________________________________________________________
block1_pool (MaxPooling2D)   (None, 64, 64, 64)        0
_________________________________________________________________
block2_conv1 (Conv2D)        (None, 64, 64, 128)       73856
_________________________________________________________________
block2_conv2 (Conv2D)        (None, 64, 64, 128)       147584
_________________________________________________________________
block2_pool (MaxPooling2D)   (None, 32, 32, 128)       0
_________________________________________________________________
block3_conv1 (Conv2D)        (None, 32, 32, 256)       295168
_________________________________________________________________
block3_conv2 (Conv2D)        (None, 32, 32, 256)       590080
_________________________________________________________________
block3_conv3 (Conv2D)        (None, 32, 32, 256)       590080
_________________________________________________________________
block3_pool (MaxPooling2D)   (None, 16, 16, 256)       0
_________________________________________________________________
block4_conv1 (Conv2D)        (None, 16, 16, 512)       1180160
_________________________________________________________________
block4_conv2 (Conv2D)        (None, 16, 16, 512)       2359808
_________________________________________________________________
block4_conv3 (Conv2D)        (None, 16, 16, 512)       2359808
_________________________________________________________________
block4_pool (MaxPooling2D)   (None, 8, 8, 512)         0
_________________________________________________________________
block5_conv1 (Conv2D)        (None, 8, 8, 512)         2359808
_________________________________________________________________
block5_conv2 (Conv2D)        (None, 8, 8, 512)         2359808
_________________________________________________________________
block5_conv3 (Conv2D)        (None, 8, 8, 512)         2359808
_________________________________________________________________
block5_pool (MaxPooling2D)   (None, 4, 4, 512)         0
=================================================================
Total params: 14,714,688
Trainable params: 0
Non-trainable params: 14,714,688
```

Figura 5.9: Modelo red neuronal acciones sexuales y violencia

CAPÍTULO 6
Resultados y conclusiones

6.1 Resultados

A continuación vamos a analizar y explicar los resultados obtenidos en cada una de las redes neuronales. Para ello primero vamos a explicar cómo analizar los resultados y qué parámetros vamos a usar para saber si hemos obtenido buenos resultados.

6.1.1. Diferencias entre accuracy, precision, recall y f1-score

En casi todos los resultados cuando se emplea una red neuronal se busca que el *accuracy* sea un valor lo más alto posible (mayor de un ochenta por cien si es posible). Esto es porque el *accuracy* mide el número de las predicciones correctas partido el total de las predicciones realizadas, por lo que con este valor puede saberse si la red neuronal está clasificando bien las muestras o no. Sin embargo, en nuestro caso, aunque también es importante este dato, es aún más importante no tener falsos negativos; es decir, es preferible que ante una imagen se clasifique primero como que existe una situación de violencia como que no, debido a que es más importante que se revisen todas las imágenes con contenido inapropiado a que se deban revisar de más (es decir, que se deban aprobar muchas con contenido apropiado), debido a la naturaleza tan sensible de los datos de la tarea. Esto lo medimos a través de los parámetros *precision*, *recall* y *f1-score*.

Para definir estos conceptos se parte del hecho de que un clasificador puede cometer dos tipos de acierto y dos tipos de error. Los aciertos en los que la clase objetivo se identifica correctamente se denominan verdaderos positivos (TP del inglés *true positive*), mientras que aquellos en los que la clase no objetivo se identifica correctamente son los verdaderos negativos (TN del inglés *true negatives*). Respecto a los fallos, están los falsos positivos (FP) y los falsos negativos (FN). FP corresponde a las muestras clasificadas como de la clase objetivo cuando no lo son, mientras que FN corresponde a las muestras clasificadas como que no son de la clase objetivo cuando sí lo son. Gráficamente, la Figura 6.1 ilustra esta situación.

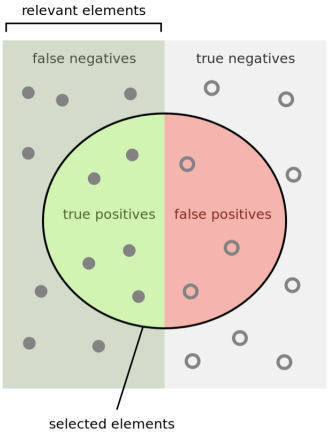

Figura 6.1: Resultados de un clasificador. *Selected elements* indica los clasificados en la clase objetivo, mientras que *relevant elements* indica los que realmente son de la clase objetivo.

La *precision* nos da el valor de cuántos de los aciertos son realmente verdaderos positivos. Podemos explicarlo también como cuántos de los datos seleccionados son relevantes, lo cual puede formularse como TP/(TP+FP). Lo podemos ver con mejor facilidad en la Figura 6.2.

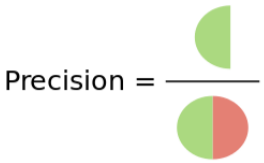

Figura 6.2: *Precision*

El *recall* indica cuántos datos positivos hemos encontrado del total de los posibles. También se puede explicar como cuántos datos relevantes hemos seleccionado, y es calculable como TP/(TP+FN). Lo podemos ver con mejor facilidad en la Figura 6.3.

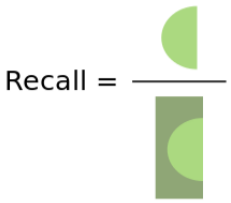

Figura 6.3: *Recall*

El *f1-score* lo utilizamos cuando queremos buscar el equilibrio entre *precision* y *recall*. Su fórmula es la siguiente:

$$F1 = 2\frac{Precision \cdot Recall}{Precision + Recall} \tag{6.1}$$

6.1.2. Explicación de los resultados obtenidos

Con estos parámetros claros vamos a pasar a explicar los datos obtenidos.

Primero vamos a explicar los resultados obtenidos en la red neuronal creada para diferenciar en una imagen menores de edad y mayores de edad. Los resultados los podemos observar en la Tabla 6.1. De los resultados podemos obtener mucha información; primero, podemos observar que el *accuracy* no es muy elevado, ya que obtenemos un 62 %; esto implica que poco más que la mitad de las muestras son clasificadas correctamente, lo que puede ser debido a la complejidad de los datos. Sin embargo, obtenemos muy buenos valores tanto en *precision* como en *recall*. Podemos observar que las imágenes etiquetadas con *menor* obtiene muy buen *recall*; esto significa que un gran porcentaje obtenido de los datos obtenidos son relevantes. Por el contrario, obtenemos que la precision es mayor en las imágenes etiquetadas como 'mayor'.

Tabla 6.1: Resultados obtenidos para la red neuronal de edad

Etiquetas	*precision*	*recall*	*f1-score*	Número de muestras
mayor	0.90	0.26	0.40	1195
menor	0.57	0.97	0.72	1198
accuracy	-	-	0.62	2397

A continuación vamos a explicar los datos obtenidos en la red neuronal creada para diferenciar situaciones sexualizadas de las que no lo son. Los resultados de esta red neuronal los podemos observar en la Figura 6.2. Al observar estos datos podemos ver que el *accuracy* se aproxima más a un resultado bueno, ya que obtenemos el 70 %. También podemos observar en este caso que tanto *precision* como *recall* son bastante elevados, lo que hace que se clasifiquen mejor las muestras cuando son situaciones sexualizadas que cuando no. Esto lo queremos de esta manera, ya que preferimos que ante una situación sexualizada se clasifique como sexualizada que como no sexualizada.

Tabla 6.2: Resultados obtenidos para la red neuronal de situaciones sexuales

Etiquetas	*precision*	*recall*	*f1-score*	Número de muestras
no	0.53	0.64	0.58	126
sí	0.81	0.74	0.77	160
accuracy	-	-	0.70	286

Ahora, vamos a explicar los datos obtenido por la red neuronal que diferenciaba si en una imagen había situación de violencia o no. Los resultados podemos observarlos en la Figura 6.3. Como podemos apreciar a primera vista, son los porcentajes más bajos que hemos obtenido de las tres redes neuronales, pero es bastante lógico debido a la complejidad de los datos aportados, ya que las imágenes eran muy variadas y complejas. Aún así, podemos observar que el *accuracy* supera el 50 %, por lo que más de la mitad de las muestras son clasificadas correctamente. Además, obtenemos un buen *recall* en el caso de situaciones de violencia; como ya hemos comentado antes esta es una buena situación, ya que ante una disyuntiva preferimos que la etiquete como violenta y evitar todos los falsos positivos posibles, ya que no queremos que una situación que es violenta la catalogue como no violenta.

Por último vamos a explicar los datos obtenidos por la unión de las redes neuronales correspondientes a la clasificación por edad y a la clasificación por violencia. Se cogió la red neuronal se entrenada con los datos anteriores y se predijeron los datos correspondientes a este apartado. Se hizo primero con la red neuronal que detecta si hay un menor o un mayor de edad en una imagen. Los resultados de esto se pueden observar en la Tabla 6.4. Se observa que se ha obtenido un 60 % de *accuracy*, por lo que más de la mitad de

Tabla 6.3: Resultados obtenidos para la red neuronal de situaciones de violencia

Etiquetas	*precision*	*recall*	*f1-score*	Número de muestras
no	0.67	0.26	0.37	1022
sí	0.54	0.87	0.67	1027
accuracy	-	-	0.57	2066

los datos están bien clasificados. Por otro lado observamos que obtenemos un muy buen *recall* en el caso de imágenes menores de edad.

Tabla 6.4: Resultados obtenidos para la red neurona de edad mezclados con situaciones de violencia

Etiquetas	*precision*	*recall*	*f1-score*	Número de muestras
mayor	0.67	0.18	0.28	259
menor	0.48	0.89	0.62	216
accuracy	-	-	0.60	475

Por otra parte se cogieron los mismos datos y se pasaron por la red neuronal que predice si hay o no violencia en un imagen. Los resultados de esto se pueden observar en la Tabla 6.5. En este caso obtenemos buen *accuracy*, ya que es de un 72 %. Nos datos de *precision* y *recall* no son tan buenos como en otras ocasiones pero aún así siguen siendo buenos, 72 % y 63 % respectivamente en el caso de imágenes en las que no aparece o sí aparece violencia.

Tabla 6.5: Resultados obtenidos para la red neuronal de situaciones de violencia mezclados con datos de edad

Etiquetas	*precision*	*recall*	*f1-score*	Número de muestras
no-violence	0.72	0.80	0.76	258
violence	0.72	0.63	0.67	217
accuracy	-	-	0.72	475

Los resultados que verdaderamente interesa son los del fruto de unir junta ambos resultados, ya que con esto se puede observar sobre las situaciones que realmente importan (imágenes en las que aparecen menores de edad y en situaciones violentas) . Estos resultados se ven reflejados en la Tabla 6.6. Cada porcentaje indica la cantidad de aciertos (*accuracy*) que ha habido en cada caso. Se observa que el porcentaje de 'menor-violencia' es de 73 %; esto quiere decir que del número total de imágenes que había de niños y en situaciones violentas, un 73 % de esas imágenes se han clasificado correctamente. Por tanto podemos decir que, aunque hay un margen de mejora, para estas situaciones hemos obtenido un porcentaje alto de aciertos.

Pero los resultados que verdaderamente nos interesa en este caso es cuando se juntas ambos resultados. Estos se ven reflejados en la Tabla 6.6

6.2 Conclusiones

En conclusión, este proyecto consta de tres fases diferenciadas: obtención de datos, creación de las redes neuronales y obtención de los resultados.

Tabla 6.6: Resultados (*accuracy*) obtenidos para la red neuronal de situaciones de violencia mezclados con datos de edad

Menor-Violencia	Mayor-Violencia	Menor-No violencia	Mayor-No violencia
0.73	0.68	0.76	0.72

En la primera fase hemos encontrado bastante dificultades a la hora de encontrar imágenes adecuadas; además, no hemos podido obtener datos de personas menores de edad en situaciones de violencia o sexuales, algo lógico debido a la delicada naturaleza de esos datos.

En cuanto a la segunda y tercera fases, han consistido en la creación de tres redes neuronales donde se han podido desarrollar los conocimientos aprendidos en clase y aumentar dichos conocimientos, así como experimentar la complejidad de crear una red neuronal.

Respecto a los objetivos que se propusieron al principio del proyecto, podemos observar que se han cumplido todos, aunque el objetivo de seleccionar datos correctos para la obtención de buenos resultados se podría mejorar. En particular, el hecho de que los datos obtenidos en algunas situaciones eran demasiado complejos, lo que hacía difícil la buena clasificación de las imágenes

Además, con este proyecto se ha aprendido a manejar nuevas herramientas, como son Keras o TensorFlow, que son muy útiles y usadas en el ámbito de las redes neuronales. Por tanto, decir que ha sido muy satisfactorio realizar este trabajo, ya que se ha podido aumentar los conocimientos en una rama de la informática que resulta muy atractiva y además se han podido poner en práctica conocimientos que se han impartido en la carrera.

CAPÍTULO 7
Trabajos futuros

7.1 Trabajos futuros

Al realizar este proyecto hemos encontrado varias mejoras que se podrían realizar.

- Mejorar resultados: en algunos de los tres casos no se ha llegado a buenos resultados, por lo que se podría ver qué configuración de capas en la red neuronal sería mejor, para obtener mejores resultados.
- Utilizar datos más complejos: el conjunto de datos utilizados en algunas situaciones puede ser considerado demasiado simple, ya que en la mayoría solo aparecen una persona; por tanto, se podría utilizar un conjunto de datos donde aparezcan más de una persona y en distintos escenarios y con distintas calidades de imágenes, para que los resultados obtenidos sean más generalizables.
- Utilizar datos que mezclen todos los casos: si bien es verdad que encontrar imágenes que mezclen todos los los escenarios (menores de edad, violencia y sexualización) es imposible, ya que por motivos obvios no podemos encontrar dichos datos en internet, podríamos crear nuestro grupo de datos; por ejemplo, se podría hacer modificando imágenes de adultos en dichas situaciones y cambiarlas por menores.

Una posible actualización puede ser crear una aplicación de móvil para que antes de subir una imagen se compruebe que dicha imagen no es considerada inapropiada para un menor de edad. También se podría realizar un programa que compruebe que una página no contiene fotos inapropiadas y de esta manera encontrar de una manera rápida y sencilla imágenes de menores que se encuentran en situaciones de violencia o de sexualización. El objetivo final sería integrarlo en redes sociales, de manera que si en la cuenta de una persona menor se detecta que se sube contenido inapropiado, se pueda avisar a los adultos responsables de dicha persona, a fin de que tomen las medidas pertinentes. Estas medidas pasan inicialmente por la retirada de dicho contenido, pero deben continuar con una adecuada pedagogía por parte de las personas adultas para que las personas menores a su cargo sean cada vez más responsables y conscientes de qué contenidos pueden compartir en sus redes sociales.

7.2 Relación con el grado

Este trabajo está relacionado con la rama de Computación que está dentro del Grado en Informática.. Dentro de esta rama podemos encontrar distintas asignaturas que están

relacionadas con el proyecto, como puede ser Aprendizaje Automático, ya que en esta asignatura hemos podido aprender sobre las redes neuronales y perceptrón. Además he realizado una estancia Erasmus en la cual tuve dos asignaturas: *Deep Learning* y *Connectionist Computing*. En estas asignaturas pude desarrollar y profundizar más en las redes neuronales y obtuve conocimientos que he podido poner en práctica en este proyecto.

Bibliografía

[1] A. Gallagher, T. Chen *Understanding Groups of Images of People*. IEEE Conference on Computer Vision and Pattern RecognitionWorkshops, CVPR Workshops 2009, pp. 256-263.

[2] Gradient Descent information January 24, 2012 Consultado en http://homes.sice.indiana.edu/classes/spring2012/csci/b553-hauserk/gradient_descent.pdf.

[3] Keras library Consultado en https://keras.io/.

[4] TensorFlow library Consultado en https://www.tensorflow.org/.

[5] VGG16 – Convolutional Network for Classification and Detection 20 November 2018 Consultado en https://neurohive.io/en/popular-networks/vgg16/.

[6] Ian Goodfellow and Yoshua Bengio and Aaron Courville *Deep Learning (Adaptive Computation and Machine Learning)*. 2016.

[7] Sapos y princesas Artículo sobre niños expuestos en las redes sociales *Tecnología para padres*, febrero, 2018.

[8] Página web how-old Consultado en https://www.how-old.net/themagic.

[9] Scylla Inteligencia Artificial para detectar situaciones de violencia Octubre de 2018 Consultado en https://www.casadomo.com/2018/10/16/scylla-aplica-tecnologia-inteligencia-artificial-detectar-situaciones-violencia.

[10] Detección de humanos en secuencias de videos 2005 Consultado en http://www.wisdom.weizmann.ac.il/~vision/SpaceTimeActions.html.

[11] Google lens Consultado en https://lens.google.com/.

[12] Librería UTK faces Consultado en https://susanqq.github.io/UTKFace/.

[13] Librería human interaction Consultado en http://cvrc.ece.utexas.edu/SDHA2010/Human_Interaction.html.

[14] Relación de los niños con la tecnología Consultado en http://educaryaprender.es/datos-uso-de-redes-sociales-ninos-y-adolescentes/.

[15] Violencia online en menores Consultado en https://www.savethechildren.es/actualidad/happy-slapping-violencia-online-menores.