UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona

FIB

FACULTAD DE INFORMÁTICA DE BARCELONA
GRADO EN INGENIERÍA INFORMÁTICA
ESPECIALIDAD DE COMPUTACIÓN

Análisis del uso de cookies en servicios web utilizando aprendizaje automático

Autor: Pablo Fonoll Soto

Director y tutor: Pere Barlet Ros
Departamento de Arquitectura de Computadores

22 de junio de 2021

Abstract

English

The use of non-essential cookies by companies is a common practice. If explicit permission is not granted by the user, websites cannot use these types of cookies in the European Union. This permission is obtained with a consent form. The main objective of this project is to implement, using computer vision and machine learning techniques, a software capable of automatically accepting the consent form. This will be used to analyze if the websites comply with European legislation or, on the other hand, are making an improper use of this type of cookies.

This work is part of a larger one, known as ORM. This project brings together researchers and students to develop a framework that informs users of the main aspects of privacy on the Internet. This software can analyze specific web domains to find out their level of intrusion into the user's privacy. It also offers general statistics about these intrusion levels, which have been generated thanks to the individual statistics obtained from each domain.

At a technical level we will see the artificial vision algorithms that have been applied. We will also see machine learning techniques, with special emphasis on deep learning. Finally, the results obtained are evaluated.

Finally, the last pages contain an explanation of future improvements to the project and a conclusion containing a summary of what I've been learned throughout this work.

Castellano

El uso de cookies no esenciales por parte de las empresas es una práctica habitual. Si no se pide permiso explícito al usuario, los sitios web no pueden usar este tipo de cookies en la Unión Europea. Este permiso lo obtienen con un formulario de consentimiento. El objetivo principal de este proyecto es implementar, mediante técnicas de visión artificial y machine learning, un programa capaz de aceptar de forma automática dicho formulario. Esto servirá para analizar si los sitios web cumplen con la legislación europea o, por otro lado, están haciendo un uso indebido de este tipo de cookies.

Este trabajo se encuentra dentro de otro más grande. Este se conoce como ORM y une a investigadores y estudiantes para el desarrollo de un framework que informa a los usuarios de los aspectos principales de la privacidad en internet. Este framework permite analizar dominios web específicos para averiguar su nivel de intrusión en la privacidad del usuario. También ofrece estadísticas generales acerca de estos niveles de intrusión, que han sido generados gracias a las estadísticas individuales obtenidas de cada dominio.

A nivel técnico veremos, a grandes rasgos, los algoritmos y técnicas de visión artificial que se han aplicado. También veremos técnicas de machine learning, haciendo especial énfasis al aprendizaje profundo. Finalmente, se evalúan los resultados obtenidos.

Para terminar, en las últimas páginas de esta memoria se explican futuras mejoras del proyecto y se concluye con un resumen de lo aprendido a lo largo de este trabajo.

Català

L'ús de cookies no essencials per part de les empreses és una pràctica habitual. Si no es demana permís explícit a l'usuari, els llocs web no poden usar aquest tipus de cookies a la Unió Europea. Aquest permís el solen obtenir amb un formulari de consentiment. L'objectiu principal d'aquest projecte és implementar, mitjançant tècniques de visió per computador i machine learning, un programa capaç d'acceptar de forma automàtica dit formulari. Això servirà per analitzar si els llocs web compleixen amb la legislació europea o, d'altra banda, estan fent un ús indegut d'aquest tipus de cookies.

Aquest treball es troba dins d'un altre més gran. Aquest es coneix com ORM, i uneix a investigadors i estudiants per al desenvolupament d'un framework que informa els usuaris dels aspectes principals de la privacitat a internet. Aquest framework permet analitzar dominis web específics per a esbrinar el seu nivell d'intrusió en la privacitat de l'usuari. També ofereix estadístiques generals sobre aquests nivells d'intrusió, que han estat generats gràcies a les estadístiques individuals obtingudes de cada domini.

A nivell tècnic veurem, a grans trets, els algoritmes i tècniques de visió artificial que s'han aplicat. També veurem tècniques de machine learning, fent especial èmfasi a l'aprenentatge profund. Finalment, s'avaluen els resultats obtinguts.

Per acabar, en les últimes pàgines d'aquesta memòria s'expliquen futures millores del projecte i es conclou amb un resum del que s'ha après al llarg d'aquest treball.

Contenidos

Lista de Figuras

Lista de Tablas

1. Introducción

El uso actual de recopilación de datos para el seguimiento web, también conocido como web tracking, representa hoy en día una amenaza para la privacidad de muchos usuarios. Tanto es así, que en el año 2018, se aprobó el reglamento general de protección de datos en la Unión Europea, conocido como *GDPR [1]*, que pretende regular todos los aspectos de la privacidad en el mundo de internet. Por otro lado, en California, Estados Unidos, se aprobó en el mismo año la Ley de Privacidad del Consumidor de California, conocida como *CCPA [2]*. Esta es la primera ley de privacidad completa en Estados Unidos y comparte muchos puntos con el *GDPR* ya mencionado. Uno de estos puntos comunes, exige a las páginas web de un consentimiento explícito por parte de los usuarios para poder recopilar algunos datos. Más en concreto, todos los datos que no sean esenciales para navegar por la página web con normalidad requerirán de dicho consentimiento. Este tipo de datos no esenciales son almacenados por parte de las empresas de forma recurrente, como por ejemplo con el uso de la mayoría de cookies. Desde la aprobación de esta regulación, son muchas las páginas web que, mediante un formulario de consentimiento, obtienen dicho permiso.

Pese al uso extendido de este formulario por parte de muchos sitios web, son muchas las empresas que no cumplen con la actual regulación. Esto se ve reflejado en algunas de las últimas noticias que han salido a la luz, las cuales informan de multas por parte de algunos gobiernos sobre grandes corporaciones por hacer un uso indebido de las cookies *[3]*.

El incumplimiento del reglamento se debe a varios motivos: muchos de estos formularios no piden permiso para la recopilación de datos, simplemente informan al usuario de que lo están haciendo. Hay formularios que piden permiso pero no existe la opción de denegarlo. Existen también formularios en los que se pide permiso, pero este es completamente ignorado por el sitio web. Por otro lado, hay páginas en las que sus formularios se encuentra en un vacío legal, pues es cierto que te piden el permiso explícito y puedes rechazarlo, pero te dificultan con varios métodos que puedas llegar a encontrar esta última opción, provocando que, para la mayoría de los usuarios, sea prácticamente imposible llegar a rechazarlo, tal y como se ve reflejado en *[4]*. Finalmente, también existen casos en los que directamente no existe ningún tipo de formulario de consentimiento, aunque este caso es el menos frecuente, sobretodo cuando se trata de sitios web con cierta popularidad y reputación.

Así pues, este trabajo tiene como objetivo desarrollar una herramienta que contribuya a un análisis estadístico sobre el uso indebido de las cookies por parte de las páginas web. Más en concreto, se pretende que esta herramienta, mediante técnicas de Machine Learning y visión por computación, sea capaz de manipular el formulario de privacidad mencionado anteriormente de forma automática para más tarde poder analizar cómo influye la aprobación o el rechazo de este en el uso de cookies.

1.1. Contexto

Este trabajo de fin de grado pertenece a la rama de computación del grado en ingeniería informática de la Universidad Politécnica de Barcelona (UPC). Más concretamente, este trabajo está centrado en la disciplina de la inteligencia artificial, especialmente en el subcampo de Machine Learning, ya que el grueso del trabajo recae en técnicas que pertenecen a este campo: algoritmos de visión artificial y deep learning.

Este trabajo, en realidad, forma parte de un proyecto más ambicioso creado por el grupo de investigación de la UPC llamado Broadband Communications Research Group, más conocido como CBA *[5]*. Este proyecto une a estudiantes e investigadores para desarrollar en conjunto una herramienta orientada a concienciar a los usuarios sobre la privacidad en internet. Esta herramienta es conocida como ORM *[6],* un programa open source en desarrollo que se centra en el ámbito de la privacidad. Su objetivo es trazar una relación entre cada página web y los recursos que esta usa, haciendo un análisis de estos recursos para identificar cuál es su finalidad. Su objetivo es identificar cuántos recursos usa cada página web y analizar cuáles son sus causas. La finalidad de esta herramienta es concienciar a los usuarios de qué páginas están invadiendo su privacidad y de qué forma lo están haciendo.

1.2. Conceptos

A continuación, se definen los principales conceptos empleados en este trabajo con el objetivo de aclararlos para evitar posibles confusiones y malinterpretaciones.

Web Tracking

El concepto web tracking hace referencia a todo rastreo del usuario. Esto se consigue recopilando información acerca de sus movimientos, los sitios web que visita, como se comporta en estos, etc.

Dentro del web tracking podemos encontrar el uso de cookies, pero existen muchas otras técnicas de rastreo como el canvas fingerprinting. Este es un proceso en el que se identifica y monitorea al usuario gracias a un tipo de elemento que se encuentra en la estructura HTML.

Cookies

Las cookies son los datos almacenados en el navegador de internet por parte de cada una de las páginas web. Estos datos son usados con varios propósitos, siendo los principales los siguientes:

- Recordar acciones realizadas previamente por el usuario, ayudando a que este no tenga que volver a repetirlas.
- Recopilar información sobre el comportamiento y las preferencias del usuario: esta información suele ser utilizada por terceros con fines publicitarios.

Lo cierto es que aunque la gran mayoría de cookies ayudan a recordar acciones previas realizadas por el usuario, el principal objetivo es ser usadas con el fin publicitario.

Formulario de consentimiento o de privacidad

Este formulario es un requisito legal para que las páginas web puedan recopilar los datos no esenciales del usuario, como muchas de las cookies que se usan. La regulación europea, exige que, si el usuario no acepta explícitamente este formulario, el sitio web no podrá recopilar ningún tipo de dato que no sea imprescindible para un funcionamiento correcto de la página.

Por lo tanto, no sirve con tan solo informar al usuario de que se están recopilando datos (sería una aceptación implícita por parte del usuario), el sitio web requiere de una autorización explícita. La regulación, además, obliga a que estos formularios tengan tanto la opción de consentir como la opción de no consentir. Además, este consentimiento se ha de renovar una vez al año como mínimo, según dicta la regulación europea.

Machine y Deep Learning

El Machine Learning es una rama de la inteligencia artificial que tiene como objetivo el de predecir datos mediante el reconocimiento de patrones. Existen muchas disciplinas dentro del Machine Learning, pero en este proyecto nos centraremos en el Deep Learning o aprendizaje profundo. Este subcampo del Machine Learning se basa en el uso de redes neuronales para modelar la abstracción de datos. En el capítulo de desarrollo se explica de una forma más detallada el concepto de redes neuronales y sus características.

Visión artificial

Conjunto de algoritmos que se encargan de obtener, procesar y analizar imágenes para generar información que se pueda tratar por el ordenador.

Intenta imitar a la visión humana, es usada para detectar objetos, saber cómo actuar en determinados escenarios visuales, etc. Un ejemplo práctico de uso de esta disciplina está en los coches autónomos, que mediante sensores ópticos reconocen a peatones, vehículos, señales, etc. para poder actuar convenientemente. En este trabajo se propone usar estos algoritmos para detectar el formulario de consentimiento y distinguir entre las partes que lo componen.

Preproceso

En visión artificial, el preprocesamiento de una imagen o preprocessing, se refiere al conjunto de operaciones que tienen como objetivo mejorar una imagen mediante la eliminación de distorsiones o resaltando las características que te interesan de la imagen.

OCR

Un *OCR* (del inglés **O**ptical **C**haracter **R**ecognition) es un conjunto de algoritmos que detecta texto en imágenes, de tal forma que dada una imagen es capaz de indicar qué palabras contiene y en qué parte de la imagen se ubican.

Espacio de color

Un espacio de color es un sistema de interpretación del color. Cada sistema de interpretación es conocido como modelo. Esto es un modelo matemático abstracto que describe la forma en la que los colores pueden representarse como tuplas de números o componentes de color.

El modelo más conocido es el RGB, en el cual cada píxel de la imagen se representa con una tupla de tres enteros. Cada uno de estos enteros indica la proporción de color rojo, verde y azul que contiene el píxel (respectivamente, cada número representa un único color). Otro modelo usado en este proyecto es el HSV, donde cada componente o canal en vez de representar la proporción de color, representa una característica. En concreto, la componente *H* representa el matiz del color, la componente *S* la saturación y la componente *V* la iluminación. De este modo, cada píxel es descrito por el conjunto de estos tres canales. Si el desarrollador lo desea, puede representar la imagen con tan solo uno de estos componentes. Por ejemplo, si se representa la imagen solo con la componente *H*, la imagen no se verá afectada por la iluminación ni la saturación. Esto también se puede aplicar en los demás modelos, por ejemplo, puedes representar la parte azul de una imagen mostrando exclusivamente el tercer componente de la tupla RGB de cada píxel.

1.3. Identificación del problema

Con el auge del uso de la informática y de internet, son cada vez más las empresas que usan ilegalmente los datos de sus usuarios para lucrarse. Esto se ve reflejado en los noticiarios actuales, donde cada vez es más frecuente leer noticias como *[3]* o *[10].*

Pese a esta intromisión en la privacidad de los usuarios, hay relativamente pocos datos que nos den una idea de la frecuencia con la que las compañías hacen un uso ilícito de nuestra privacidad. Es por esto que nace el proyecto de ORM, y, por consiguiente, este trabajo. Así, el principal objetivo de este trabajo de fin de grado es el de desarrollar una herramienta que contribuya a concienciar a los usuarios acerca de qué compañías no cumplen la legislación en cuanto al uso de cookies.

Desde un punto de vista más práctico, el objetivo de este trabajo es el de desarrollar una herramienta que manipule automáticamente el formulario de consentimiento. La intención es que esta herramienta se incorpore al ORM para contribuir a un análisis estadístico. En concreto, servirá para cuantificar el número de páginas que no cumplen con la regulación del GDPR, que, entre otras cosas, prohíbe a los proveedores web recopilar información de los usuarios si estos no lo consienten explícitamente. Para poder comprobar si los servicios webs cumplen con esta legislación, el ORM tiene que analizar, para cada página web, qué información recopila en función de si se le concede permiso para el uso de cookies o no.

Así, el ORM es capaz de sacar conclusiones sobre si esta decisión del usuario influye como debería. Para ello, el proceso de otorgar permiso tiene que ser automático y aquí, es donde nace este trabajo de fin de grado.

Este trabajo, por lo tanto, tiene una parte de investigación, que consiste en estudiar e investigar los métodos de visión artificial y machine learning ya existentes para llevar a cabo el desarrollo de la herramienta de automatización. Y una parte práctica, donde se desarrollará dicha herramienta, que consiste en manipular el formulario de consentimiento de forma automática.

1.4. Actores implicados

Como ya se ha mencionado, este programa será incorporado al ORM, el cual pretende dar una visión general a los consumidores de internet acerca de la privacidad que realmente poseen, con la intención de hacerles conscientes de la intromisión por parte de algunas páginas web. Además, al ser un proyecto de código abierto, se espera que otros grupos de investigación usen este proyecto como herramienta para indagar más en el mundo de la privacidad en internet.

En conclusión, este proyecto tiene como beneficiarios a los usuarios de internet, que gracias a la herramienta ORM, serán más conscientes de que páginas no respetan su privacidad. Además, futuros grupos de investigación podrán verse beneficiados con el ORM, ya que podrán hacer uso tanto del código fuente como de la herramienta en sí, dado que esta es de código abierto.

2. Justificación

Como ya se ha explicado anteriormente, este trabajo consta de dos partes: la parte de investigación y la parte de desarrollo. Esta primera parte consiste en estudiar los métodos actuales de machine learning y visión artificial para poder determinar cuales son los más adecuados para este proyecto concreto. Finalmente, en la parte desarrollo, se usan estos métodos para la implementación del programa. Por lo tanto, es en la etapa de investigación dónde estudiaré los trabajos y las técnicas ya existentes para evaluar si es necesario el desarrollo de este proyecto.

2.1. Situación actual

En primer lugar, cabe destacar que ya existen algunas soluciones con las mismas intenciones que este proyecto: la automatización en la manipulación del formulario de consentimiento para el uso de cookies.

Aún así, estos objetivos se quieren alcanzar de una forma muy distinta de la que lo hacen las otras soluciones, puesto que en estas se manipula el formulario de consentimiento modificando la estructura HTML del sitio web. Por otro lado, en este proyecto, se quiere lograr este objetivo mediante el uso de visión artificial combinado con técnicas de Machine Learning.

Se quieren usar estas técnicas para averiguar si este problema se puede abordar con este nuevo enfoque. También sería interesante averiguar si con el uso de estas técnicas, la solución de este proyecto es más efectiva que las soluciones ya existentes, dado que las soluciones actuales pierden efectividad si el sitio web tiene una estructura HTML muy compleja y diferente a la estándar, ya que es difícil de encontrar la estructura asociada al formulario de consentimiento. En cambio, este problema no debería existir con el enfoque de este proyecto, puesto que el formulario de consentimiento, tenga la estructura que tenga, tiene que ser visible para que el usuario pueda manipularlo y, por lo tanto, debería poderse detectar con técnicas de visión artificial.

Por otro lado, la existencia de este proyecto también es justificada porque es necesario incorporar esta solución al ORM y, dado que muchas de las soluciones existentes no se podrían incorporar porque no son de código abierto, es necesario el desarrollo de la herramienta. Además, una herramienta propia permitirá adaptarse mejor al ORM en cuanto a funcionalidades específicas.

2.2. Soluciones existentes

Como ya se ha explicado, existen herramientas que se dedican a bloquear el formulario de consentimiento. A continuación, presentaré brevemente las soluciones más relevantes.

La mayoría de estas aplicaciones se basan en identificar el formulario dentro de la estructura HTML de la página para mostrarte el sitio web sin cargar dicho elemento, como es el caso de *[7]*, un plugin para navegador que bloquea el formulario de consentimiento, eso sí, sin manipular las opciones de rechazar ni de consentir, tan solo impide que se muestre el formulario.

Por otro lado, existen herramientas que intentan aceptar siempre las cookies como es el caso de *[8]* . También las hay que intentan rechazarlas, como es el caso de *[9]*. La mayoría de estas herramientas también hacen uso de la estructura HTML del sitio web para identificar el botón que acepta o rechaza el uso de cookies.

Cabe destacar que la complejidad de manipular el formulario de consentimiento es muy diferente según cual sea su fin. Por ejemplo, si lo que se quiere es aceptar todas las cookies, la complejidad será mucho menor que la de querer rechazarlas todas, esto es debido a que estos formularios suelen dar mucha más visibilidad al botón de aceptar todas las cookies que al de rechazarlas, el cual suele estar escondido a nivel visual o puede aparecer tras completar unos pasos. Esto es en el caso de que exista dicho botón, ya que en algunos sitios web no aparece directamente. De hecho, existen estudios que intentan demostrar como las páginas web hacen esto deliberadamente para hacer desistir al usuario. Más en concreto, existe un estudio *[4]* que, mediante visión por computación, extrae los formularios y evalúa la dificultad de rechazarlos midiendo el número de clicks que conlleva. Este trabajo, para la identificación y clasificación de los botones del formulario hace uso de técnicas de Machine Learning supervisadas, que requieren de esfuerzo manual para la creación de un conjunto de soluciones conocidas(conocido como datos de entrenamiento). Este esfuerzo manual no es viable con los recursos tanto temporales como económicos de los que consta este TFG, por lo que a priori queda descartado usar técnicas similares. Además, este estudio no ha publicado los algoritmos ni modelos empleados, por lo que no podré aprovecharlo.

Finalmente, quiero destacar un conjunto de trabajos llevados a cabo por otros grupos de investigación que consisten en, a partir de una captura de pantalla de una aplicación móvil, generar el código fuente asociado. Pese a que estas herramientas tienen objetivos distintos al mío, para lograrlos también tienen que identificar mediante visión artificial, los diferentes objetos en la imagen, que, al ser elementos de interfaz, comparten ciertas características con el formulario de consentimiento de los sitios web.

Uno de estos trabajos es el caso de *[11]*, se usa el algoritmo de *Canny* para detectar las aristas que forman los diferentes objetos para poder identificarlos. Esto lo combina con técnicas para la detección de caracteres y palabras que puedan aportar información. Más tarde, clasifican los elementos encontrados mediante el uso de redes neuronales, que también requieren de datos de entrenamiento. Creo que podré adaptar las técnicas de visión artificial de este estudio para este TFG, dado que usan librerías de Open Source. Por ejemplo, usan una librería muy popular de visión por computación que incluye un gran abanico de técnicas, esta se llama OpenCV y está disponible tanto para el lenguaje de programación C++ como para Python. Aun así, no podré hacer uso de técnicas de Machine Learning supervisadas para este trabajo por falta de recursos, como ya se ha mencionado anteriormente.

En conclusión, no podré aprovechar nada de las soluciones que usan la estructura HTML del sitio web para manipular el formulario de consentimiento, puesto que estas tienen un enfoque muy distinto al de este proyecto. Además, la mayoría de estos proyectos no son de código abierto y son de dudosa efectividad. Por otro lado, de los dos últimos trabajos que he mencionado podré adaptar algunas de las técnicas de visión por computación que usan, como el algoritmo de Canny y el uso de OCR. Aunque a priori no podré aprovechar la parte de Machine Learning, ya que usan sistemas supervisados que requieren de recursos que este proyecto no posee.

3. Alcance

3.1. Objetivos

El principal objetivo de este trabajo es el de crear una herramienta que sea capaz de automatizar la manipulación del formulario de consentimiento que aparece en muchas páginas web. Esto se conseguirá mediante visión artificial y machine learning. Al incorporarse a la herramienta ORM, este proyecto estará centralizado en un repositorio remoto, por lo tanto, en la nube.

A continuación, se exponen los principales subobjetivos de este trabajo:

- Estructurar la gestión del proyecto. Esto consiste en planificar, concretar y documentar este trabajo de fin de grado, que quedará reflejado en esta memoria.

- Estudio y elección de soluciones existentes y herramientas. Este objetivo pretende investigar proyectos y estudios ya existentes que puedan ser usados para este trabajo. Por otro lado, también consiste en encontrar las herramientas de trabajo más apropiadas para el desarrollo de este proyecto, facilitando así el desarrollo.

- Investigar y desarrollar técnicas de detección de elementos visuales: se pretende encontrar los mejores métodos para detectar elementos, qué propiedades pueden ser las más útiles para usar como criterio para el caso específico de este proyecto. Esto es imprescindible para detectar no sólo dónde se sitúa el formulario de consentimiento dentro de la página web, sino también para identificar dónde se encuentran los diferentes botones a manipular para otorgar o no los permisos de privacidad. La visión artificial será la encargada de decidir cuales son las partes de la pantalla a pulsables para poder manipular el formulario de consentimiento.

- Investigar y desarrollar técnicas de machine learning: Dado que existen diferentes algoritmos de machine learning, el objetivo es encontrar el más eficiente tanto en tiempo como en ratio de acierto para este trabajo particular. Este algoritmo será el encargado de decidir en qué partes de la pantalla pulsar para manipular el formulario de consentimiento.

- Validación de la herramienta. Una vez implementada la herramienta, se tendrá que comprobar y validar su funcionamiento y efectividad.

- Facilitar la integración de la herramienta a ORM: como ya se ha mencionado, esta herramienta se incorporará a otra mayor, el ORM. Se tendrá que adaptar parte del código para facilitar la integración con ORM.

3.2. Requerimientos

3.2.3. Requerimientos funcionales

Estos son los principales requerimientos funcionales a tener en cuenta:

- Desarrollar una herramienta que sea capaz de identificar y aceptar, dentro de una página web, el formulario de consentimiento. Lo ideal sería que también pudiese rechazar o cambiar la configuración de dicho formulario, pero esto puede ser muy complicado puesto que muchos formularios tienen muchos pasos intermedios para poder realizar estas acciones. De hecho, ya hemos visto en el capítulo de soluciones existentes como esto aún no se ha logrado con suficiente eficacia.

- Integración con la herramienta principal: como ya se ha mencionado varias veces, este proyecto se integrará con la herramienta ORM, por lo tanto, se tiene que diseñar y desarrollar para que sea posible esta futura integración.

3.2.4. Requerimientos no funcionales

- Este trabajo tiene que ser muy eficiente, puesto que tanto las técnicas de Machine Learning como las de visión por computador suelen ser muy costosas en recursos, tanto en tiempo como en capacidad de procesamiento. Además, este proyecto está pensado para examinar una gran cantidad de páginas web, por lo que una buena gestión de los recursos es vital.

- Este trabajo también tiene que ser reusable, puesto que irá integrado al ORM y esta herramienta es de código abierto, por lo que otros usuarios pueden adaptar esta herramienta para darle un uso específico. Es por esto que es necesario que este trabajo se pueda integrar con facilidad a diferentes proyectos.

3.3. Obstáculos y riesgos

Todos los proyectos constan de dificultades y este, no es menos. A continuación se enumeran los posibles obstáculos que puedan aparecer y cómo se solventarían.

- Como ya se ha mencionado, este trabajo necesita una investigación previa, por lo que, en función de los resultados de esta, puede que la implementación de la herramienta lleve mucho tiempo y trabajo. Como solución, se seguirá una metodología flexible, que permita modificar los objetivos rápidamente y que impida grandes bloques de progreso sin revisar, evitando así posibles pérdidas de tiempo indagando en callejones sin salida.

- Existen muchos trabajos relacionados con la privacidad, pero, al ser una área relativamente nueva, no está muy claro qué métodos son mejores, por lo que asegurarse de usar el método más efectivo será difícil. Como solución a esto, se

investigarán cada uno de los principales métodos en profundidad, lo que llevará tiempo pero asegurará que el trabajo tenga unas bases fiables y efectivas.

- Al existir muchos tipos de formularios de consentimiento, es posible que sea muy complejo detectarlos todos. En el caso que así sea, el trabajo sería reducido a un ámbito más concreto. Por ejemplo, la herramienta se podría reducir a tratar solo algunos tipos concretos de formularios, como por ejemplo, los más frecuentes. Por otro lado, también se podría reducir la funcionalidad de la herramienta, como por ejemplo, que solo fuera capaz de indicar dónde está el botón de aceptar las cookies (en vez del de aceptar y rechazar).

4. Metodología

En este proyecto hay una gran parte de investigación acerca de qué estrategias seguir y qué técnicas de Machine Learning usar. Es por ello que se requiere de una metodología muy flexible que permita probar estas técnicas y, en función de los resultados, decidir si indagar más en ellas o descartarlas y buscar alternativas. Esta metodología tan flexible requiere de revisiones muy frecuentes, por tal de dar margen de maniobra, siendo esta una metodología muy similar a la metodología ágil.

Para realizar un seguimiento adecuado del proyecto, se ha creado un pequeño equipo de investigación de unos ocho miembros. Este grupo está formado por algunos profesores, entre los cuales se encuentra mi tutor, y por estudiantes que están haciendo un trabajo de fin de estudios sobre la temática de la privacidad en internet. Todos los miembros de este equipo estamos contribuyendo al desarrollo del proyecto ORM y cada dos semanas nos reunimos de forma telemática durante dos horas aproximadamente para que cada uno exponga su progreso hasta la fecha. En estas reuniones, se discute en grupo cuáles son los objetivos de cada miembro de cara a la próxima reunión, dando más valor a la opinión del autor del trabajo y del tutor. A parte de estas reuniones bisemanales, adicionalmente convocamos reuniones con mi tutor para evaluar de una forma más detallada mi progreso.

Así pues, el método de seguimiento de este trabajo consiste en reuniones periódicas vía Google Meet con mi tutor y reuniones con los desarrolladores de ORM. En estas reuniones, mi tutor es quien acaba validando el trabajo hecho en función de si se han cumplido los objetivos propuestos en la anterior reunión.

5. Planificación temporal

En esta sección se realiza una planificación temporal del proyecto, con el objetivo de cumplir con los plazos establecidos para poder finalizar el proyecto en la fecha estimada. Esta planificación se divide en las diferentes tareas a realizar y se usan horas como unidad temporal.

Este trabajo de fin de grado tiene como fecha de inicio el día 1 de febrero de 2021 y como fecha de finalización el día 28 de junio de 2021, aunque esta última no es una fecha exacta, tan solo una previsión. Durante este periodo se estima un trabajo de unas 365 horas, que, divididas entre los aproximadamente 150 días de duración, nos da unas 2,5 horas diarias de media, incluidos fines de semana.

5.1. Descripción de las tareas

A continuación, se procede a explicar en qué consisten las tareas a realizar. En estas tareas se incluyen las reuniones, así como la gestión de la documentación que este proyecto conlleva.

Las tareas se van a dividir en Gestión (G), Investigación y estudio (IE), Desarrollo y testeo (DT).

Con el fin de evitar repeticiones, se entiende que, para todas las tareas, es necesario los siguientes recursos materiales: Ordenador e internet. Los recursos adicionales para cada tarea concreta se especifican en la descripción de estas, que se exponen a continuación.

1. Tareas de gestión del proyecto

Estas tareas consisten en planificar, concretar y documentar este trabajo de fin de grado, que quedará reflejado en esta memoria. También se incluyen otras tareas de gestión, como por ejemplo las reuniones de planificación. Se estima que está subdivisión de las tareas conlleva unas 160 horas. El hito de esta tarea es tener redactada la memoria del TFG, así como tener una presentación hecha.

G1 - Contextualización y alcance

Consiste en definir el contexto en el que se encuentra el proyecto, así como especificar los siguientes puntos que definen el alcance: objetivos del trabajo, requerimientos mínimos y posibles obstáculos y riesgos. En esta tarea se justifica también la existencia de este trabajo.
Duración: 25h.
Dependencias: Ninguna.
Recursos humanos: Gestor del proyecto.
Recursos Materiales: Microsoft Office.

G2 - Planificación temporal

Consiste en concretar y estructurar la planificación temporal del proyecto, la cual va ayudar a cumplir los objetivos en los plazos temporales estipulados, ya que se describe en cuánto tiempo conlleva cada tarea a realizar.
Duración: 15h.
Dependencias: G1.
Recursos humanos: Gestor del proyecto.
Recursos Materiales: Microsoft Office, GanttProject.

G3 - Gestión económica y de sostenibilidad

En esta tarea se pretende realizar un presupuesto para cuantificar el coste que comporta realizar este proyecto, tanto desde el punto de vista material como por parte de personal. También se analiza el impacto medioambiental, económico y social de este trabajo.
Duración: 20h.
Dependencias: G2.
Recursos humanos: Gestor del proyecto.
Recursos Materiales: Microsoft Office.

G4 - Organización de documentos

Esta tarea consiste en agrupar los documentos realizados en las tareas previas, de manera que se mantenga la concordancia y cohesión entre las diferentes partes a unificar.
Duración: 10h.
Dependencias: G3.
Recursos humanos: Gestor del proyecto.
Recursos Materiales: Microsoft Office.

G5 - Reuniones

Es necesario reunirse periódicamente con el director de este proyecto para analizar el progreso del proyecto y, en función de esto, volver a planificar o acotar las tareas a realizar. Estas reuniones sirven como métodos de validación y seguimiento, dado que se realizan casi semanalmente. Su duración varía mucho en función del progreso alcanzado. Además, también se incluyen las reuniones con los otros desarrolladores de ORM. Estas reuniones son bisemanales y tienen una duración de unas 2 horas por reunión. El hito de esta tarea es realizar un seguimiento y validación del progreso realizado.
Duración: 30h.
Dependencias: Ninguna.
Recursos humanos: Gestor del proyecto.
Recursos Materiales: Google Meet.

G6 - Memoria

Esta tarea consiste en la redacción de la memoria de este proyecto. Esto incluye la unión del documento generado en la tarea G4 con la redacción del resto de la memoria.
Duración: 50h.
Dependencias: Se realizará en paralelo mientras se avanza con el desarrollo del proyecto. Aún así, es requerido haber finalizado la tarea G4.
Recursos humanos: Gestor del proyecto.
Recursos Materiales: Microsoft Office.

G7 - Presentación

Aquí se prepara una presentación oral, apoyada por diapositivas, para el tribunal de evaluación del TFG.
Duración: 10h.
Dependencias: G6.
Recursos humanos: Gestor del proyecto.
Recursos Materiales: Microsoft Office.

2. Investigación y estudio

Estas tareas consisten en investigar soluciones, métodos existentes y artículos científicos relacionados con este trabajo. También se incluye el estudio de las herramientas apropiadas para desarrollar este proyecto. Se estima que está subdivisión de las tareas conlleva unas 115 horas.

IE1 - Estado del arte

Consiste en investigar soluciones existentes que puedan ser usados para el desarrollo de este proyecto. Esta tarea también incluye la lectura de artículos científicos para tener conocimiento de la situación general del problema para poder redirigir el proyecto. Se intentarán aprovechar estas soluciones existentes, ya sea en su totalidad o en algunos métodos concretos que usan, para el desarrollo del proyecto, que se especifica a continuación. El hito de esta tarea es orientar y concretar el desarrollo de trabajo en función del estado del arte.
Duración: 35h.
Dependencias: Ninguno.
Recursos humanos: Investigador y gestor del proyecto.
Recursos Materiales: Ninguno adicional.

IE2 - Estudio y elección de las herramientas de trabajo

Consiste en encontrar las mejores herramientas de trabajo. Se estudia qué librerías y algoritmos concretos de Machine Learning y visión artificial son las más apropiadas para usar en este proyecto. También se investiga cuál es el mejor entorno de programación para usar estas librerías y algoritmos en términos de usabilidad y eficiencia. Como se puede apreciar, esta tarea es más práctica y menos teórica que IE1, ya que se buscan métodos,

librerías y herramientas específicas que serán usadas para el desarrollo. El hito de esta tarea es encontrar herramientas específicas para que el desarrollador las pueda utilizar.
Duración: 80h.
Dependencias: Ninguno.
Recursos humanos: Programador e investigador.
Recursos Materiales: Ninguno adicional.

3. Desarrollo del proyecto

Estas tareas consisten en el desarrollo de la herramienta a realizar, así como la validación mediante testeo del código programado. Se divide en dos fases que están claramente separadas. La parte de visión artificial, que consiste en detectar el formulario de consentimiento, y la parte de Machine Learning, que consiste en identificar las diferentes partes del formulario de forma automática. Se estima que esta subdivisión de las tareas conlleva unas 90 horas.

DT1 - Diseño de la herramienta

Consiste en diseñar cómo estará estructurada la herramienta a implementar. Se decidirán cuáles de las herramientas de IE2 son finalmente usadas y, con esta información, se concretará cómo será la herramienta en detalle.
Duración: 20h.
Dependencias: IE1, IE2.
Recursos humanos: Programador.
Recursos Materiales: Visual Studio.

DT2 - Programación de detección de formulario

Consiste en programar la parte de la herramienta que reconoce tanto el formulario de consentimiento como sus diferentes partes. Esto se conseguirá gracias a las técnicas de visión artificial ya vistas en las tareas de estudio (IE), por lo que se ahorrará mucho tiempo y esfuerzo que se perdería en la programación. El hito de esta tarea es haber implementado la detección del formulario y sus partes, sin identificar que es cada parte ni para qué sirve.
Duración: 20h.
Dependencias: DT1.
Recursos humanos: Programador.
Recursos Materiales: Visual Studio.

DT3 - Programación de la parte de Machine Learning

Consiste en programar la parte de la herramienta que identifica qué función realiza el pulsar cada parte del formulario de consentimiento. Esto se conseguirá gracias a las técnicas de Machine Learning, más concretamente de Machine Learning, ya vistas en las tareas de estudio (IE). El hito de esta tarea es completar un método de identificación de las partes del

formulario, sabiendo para cada objetivo, ya sea el de rechazar, aceptar o cerrar el formulario, que partes del formulario se han de manipular para lograrlo.
Duración: 20h.
Dependencias: DT2.
Recursos humanos: Programador.
Recursos Materiales: Visual Studio.

DT4 - Validación de la herramienta

Consiste en validar y comprobar el funcionamiento y la efectividad de la herramienta en conjunto. Esto se conseguirá mediante pruebas exhaustivas que se determinarán en un futuro. Este esfuerzo se reducirá ligeramente ya que los otros desarrolladores de ORM también validarán el funcionamiento de esta herramienta probándola en sus respectivos entornos de trabajo. El hito de esta tarea es asegurar el correcto funcionamiento en todos los aspectos de la herramienta implementada, además de calcular y valorar su efectividad.
Duración: 30h.
Dependencias: DT3.
Recursos humanos: Programador.
Recursos Materiales: Visual Studio.

5.2. Recursos

5.2.1. Recursos humanos

En este proyecto se distinguen tres roles distintos: gestor del proyecto, investigador y programador. Todos estos roles son realizados por el autor de este trabajo de fin de grado. A continuación se explican estos roles y qué papel desempeñan en este trabajo.

- **Gestor del proyecto**: se encarga de orientar y supervisar el proyecto, tiene como función establecer los objetivos y dirigir con criterio tanto el proceso de desarrollo como el de investigación, es decir, es el responsable de tomar las decisiones importantes que acaban determinando el proyecto. También se encarga de concretar objetivos y documentar el proyecto.

- **Investigador**: se encarga de estudiar los métodos y soluciones existentes para acotar una buena solución. También se encarga de evaluar las posibles herramientas para que el programador haga uso de las más óptimas.

- **Programador**: se encarga de implementar la herramienta a desarrollar, aplicando las soluciones y herramientas aportadas por el investigador.

- **Tester**: se encarga de validar y comprobar el funcionamiento correcto de la herramienta, así como estudiar la efectividad de esta tanto en términos temporales como en aciertos.

5.2.2. Recursos materiales

En este apartado se listan y describen los recursos materiales necesarios para realizar este proyecto, distinguiendo entre los materiales de software y de hardware.

Recursos Hardware:

- **Ordenador portátil**: este recurso es esencial para este proyecto, puesto que es la herramienta que se usará para documentar el proyecto, investigar, desarrollar y probar la herramienta.

Recursos Software:

- **Microsoft Office**: conjunto de aplicaciones ofimáticas que serán usadas para la documentación del proyecto, así como para realizar la presentación del trabajo.

- **Visual Studio:** entorno de desarrollo de software, será el entorno donde se desarrollará la aplicación.

- **Google Meet:** servicio web para organizar reuniones telemáticas. Mediante este servicio se llevan a cabo todas las reuniones de este proyecto.

- **GanttProject:** herramienta de software libre que se utilizará para realizar el diagrama de Gantt.

Acabamos de ver las tareas a realizar, los roles del proyecto y los recursos tanto de hardware como de software. Para un mejor entendimiento y, a modo de resumen, se sintetiza toda esta información de forma conceptual en la siguiente tabla:

Id. Tarea	Tiempo	Dependencias	Roles	Recursos
G1	25h	-	GP	P, MO
G2	15h	G1	GP	P, MO, GP
G3	20h	G2	GP	P, MO
G4	10h	G3	GP	P, MO
G5	30h	-	GP	P, GM
G6	50h	G4	GP	P, MO
G7	10h	G6	GP	P, MO
IE1	35h	-	I, GP	P
IE2	80h	-	I, P	P
DT1	20h	IE1, IE2	P	P, VS
DT2	20h	DT1	P	P, VS
DT3	20h	DT2	P	P, VS
DT4	30h	DT3	P	P, VS

***Tabla 1**: Tabla de tareas, con el identificador de tarea, tiempo previsto, dependendencias, roles y recursos materiales. Roles: GP - gestor del proyecto, I- Investigador, P- Programador.. Recursos: P- Portátil, MO- Microsoft Office, GP- GanttProject, GM- Google Meet, VS- Visual Studio. **Fuente:** Elaboración Propia.*

5.3. Gestión de riesgo

Al ser un proyecto de investigación y desarrollo, la previsión del riesgo es crucial para no llevarse sorpresas de última hora. A continuación se explican los posibles riesgos y se detallan planes alternativos para solventarlos.

1. Estudio del arte insuficiente: es posible que las soluciones existentes actuales no se puedan aprovechar completamente para la realización de este proyecto. Para prevenir esto, se han añadido hasta 40 horas adicionales en la tarea DT1, que corresponde al trabajo adicional que tendría que emplear el programador para desarrollar todo lo que no se haya podido aprovechar de soluciones existentes. De no existir este riesgo, se usarían estas horas para mejorar la eficacia y usabilidad de la herramienta.

2. Prolongación del desarrollo de la herramienta: al ser una herramienta compleja que hace uso de disciplinas que prácticamente no he visto en este grado, como es la visión artificial o machine learning, es posible que el desarrollo de la herramienta lleve más tiempo de lo previsto. Para poder manejar con éxito este riego, se han añadido 20 horas adicionales a la tarea DT1, que implicarían más horas de trabajo para el programador. También se han reservado 10h adicionales en reuniones, puesto que al haber más dificultades se emplearía más tiempo en la organización y validación del trabajo hecho. De nuevo, de no existir este riesgo, se usarían estas horas para mejorar la herramienta en términos de eficacia.

Estos riesgos no conllevan penalizaciones en el material, dado que todos los recursos materiales usados son gratuitos o ya han sido adquiridos. Por lo tanto no se les aplica restricciones temporales.

6. Gestión económica

En este capítulo se calcula una estimación de los recursos económicos necesarios para llevar a cabo este proyecto, es decir, los costes. Estos recursos se pueden clasificar en recursos humanos, recursos materiales, gastos generales e impuestos. A continuación, se explica y se detalla cada uno de ellos.

6.1. Recursos humanos

Los costes destinados a recursos humanos son los entendidos como gastos del personal implicado directamente en este proyecto, que contribuyen a la resolución de las tareas descritas en la sección anterior. Estos costes se expresan en el coste por hora y viene determinado por el rol que ocupa cada agente en relación al proyecto. Con la intención de que estos costes sean objetivos, se ha consultado el salario medio de cada rol en el servicio web *PayScale [15]*. Además, a estos gastos se tiene que añadir el coste de la seguridad social, que, a niveles prácticos, supone aproximadamente un 30% del salario bruto del empleado.

Así pues, se distinguen los roles de investigador científico, programador de software y de jefe de proyecto:

Rol	**Coste por hora (incluido Seguridad Social)**
Gestor de proyecto (GP)	25.09 €
Investigador (I)	22.48€
Desarrollador Software (P)	19.72€

Tabla 2: *Costes de personal elaborada a partir de los datos obtenidos de PayScale.*
Fuente: *elaboración propia.*

Una vez sabemos el coste por hora del personal y el número de horas por tarea, donde las tareas están especificadas en el Gantt del Anexo, podemos calcular un coste aproximado de cada tarea, el cual se ve reflejado en la siguiente tabla:

Identificador de la tarea	**Tiempo**	**Roles involucrados**	**Coste Jefe Proyecto** (incluida Seguridad Social)	**Coste Investigador** (incluida Seguridad Social)	**Coste Desarrollador Software** (incluida Seguridad Social)	**Coste Total** (incluida Seguridad Social)
Gestión			**4329€**			**4329€**
G1	25h	GP			493€	493€
G2	15h	GP			295€	295€
G3	20h	GP			394€	394€
G4	10h	GP			197€	197€
G5	30h	GP	752€		591€	1344€
G6	50h	GP	376€		986€	1326€
G7	10h	GP	83€		197.2€	280€
Investigación			**878€**	**2585€**	**1577€**	**5040€**
IE1	35h	I, GP	878€	787€		1665€
IE2	80h	I, P		1798€	1577€	3375€
Desarrollo					**1773€**	**1773€**
DT1	20h	P			394€	394€
DT2	20h	P			394€	394€
DT3	20h	P			394€	394€
DT4	30h	P			591€	591€
Total	**365h**					**11142€**

Tabla 3: *Costes por tarea.* ***Fuente:*** *elaboración propia.*

6.2. Costes materiales

Estos costes son los asociados a las herramientas necesarias para poder desarrollar este proyecto. Este trabajo, al consistir en el desarrollo de una aplicación informática, se podría dividir en costes de hardware y de software, como se ve reflejado a continuación.

6.2.1. Hardware

En cuanto al hardware, que hace referencia a los componentes físicos usados para el desarrollo del proyecto, el principal gasto proviene del requerimiento de un ordenador, pues es necesario tanto para la investigación como para el desarrollo software, así como para redactar la documentación. En este caso particular se usa el modelo de *Lenovo IdeaPad Duet 3i* y el precio de este se ve reflejado en su página web oficial *[12].* Como se puede ver en la tabla de la siguiente página, se refleja el coste hardware, junto a la vida útil y a la amortización, en la que se tiene en consideración un uso diario de 2,5 horas en los 150 días de duración del proyecto. Para calcular la amortización se aplica la fórmula de la figura 1:

$$Amortización = \frac{Coste\ (euros) * Duración\ del\ proyecto\ (horas)}{Vida\ útil(años) * \left(\frac{Días\ laborables}{Año}\right) * Dedicación\ \left(\frac{horas}{Día}\right)}$$

Figura 1: Fórmula amortización. **Fuente:** Elaboración propia.

Equipo	Unidades	Precio	Vida útil	Amortización
IdeaPad Duet 3i	1	476.10€	4 años	119€

Tabla 4: Costes de Hardware. **Fuente:** Elaboración propia.

6.2.2. Software

Se conoce como costes de software al coste del conjunto de programas usados para poder llevar a cabo este proyecto. Entre ellos destacan los programas del entorno de windows, como los programas de elaboración de documentos. Se resumen en la siguiente tabla:

Producto	Precio	Vida útil	Amortización
Microsoft Windows 10 pro	259€	8 años	32.37€
Microsoft Office 2019	299€	4 años	74.75€

Tabla 5: Costes de Software. **Fuente:** Elaboración propia.

6.2.3. Costes Directos

Este coste se refleja mediante la suma de los costes anteriores, dado que son los gastos explícitos necesarios para llevar a cabo este proyecto. La suma de estos costes nos aproxima una cifra de **11367€**.

6.2.4. Costes indirectos

Estos son los costes implícitos que conlleva el proyecto. En el caso de este trabajo, el único uso explícito es el de la electricidad, que tiene un precio aproximado de 0.1270€/kWh *[13]*. La potencia media de este portátil es de 220Wh, según el propio fabricante, por lo que, si multiplicamos esta cifra en las unidades adecuadas por el número de horas totales del proyecto obtenemos un consumo de 82.5KWh, que, al precio aproximado anterior nos sale la electricidad por unos 10,4€. Si a esto le sumamos el coste de luz, que es de aproximadamente unos 110Wh, por lo que nos sale por unos 5,2€. Así pues, el coste total en electricidad es de unos **15,6€**.

Por último, se tiene que tener en cuenta el uso de internet. Como estimación, calculo que con unos 50MB de internet es más que suficiente para cubrir las necesidades que requiere este proyecto. El precio mensual aproximado de la conexión es de unos 30€/mes *[14]*, por lo que implica un gasto de unos **150€** aproximadamente.

Así pues, se estima que los costes indirectos suman un total de **165€**.

6.3. Plan de contingencia

Como se ha explicado en anteriores secciones, un plan de contingencia es necesario para afrontar cualquier tipo de problema. Pese a esto, el plan de contingencia no supondrá un gasto adicional más que el tiempo de dedicación del personal y del uso de material. Es por ello que se añaden unos **1739€** aproximados a los posibles retrasos, que corresponden a un 15% del total de los costes. Este coste se ve justificado en la siguiente tabla.

Tipo de coste	**Coste**	**Coste de contingencia**
Personal	11142	1671€
Software	106€	15€
Hardware	119€	17€
Costes Indirectos	165€	27€
Total	**11532€**	**1730€**

Tabla 6: Tabla de contingencia. **Fuente:** Elaboración propia.

6.4. Imprevistos

Existen varios imprevistos que pueden afectar temporal y económicamente al coste del proyecto. Es por esto que se ha previsto una reserva de un 10% de los gastos totales para cubrirlos, lo que equivale a unos **1150€** aproximadamente.

6.5. Presupuesto final

Así pues, el presupuesto final se eleva a la cifra de 12273.58€, que no es más que un reflejo de la suma de todas las cifras mencionadas anteriormente.

Tipo de coste	Coste
Costes Directos	11367€
Costes Indirectos	166€
Contingencias	1730€
Imprevistos	1150€
Total	**14413€**

Tabla 7: Tabla de presupuesto final. **Fuente:** Elaboración propia.

6.6. Control de gestión

Para asegurarse de una buena gestión del proyecto, se realizarán controles de forma periódica del uso de los recursos. Cada dos semanas se calcularán los costes del tiempo transcurrido para ver si son fieles a las estimaciones anteriores. En el caso de resultar estimaciones no realistas, se reajustarán para adaptarse a la realidad. Aun así, se calcula, en el peor de los casos, una extensión adicional de un 25% del proyecto, lo que supone un total de 91 horas aproximadamente, lo que supone un desvío en gasto en mano de obra de unos 2800€ y un desvío en materiales de unos 56€. Como aproximación, se calcula unos 3500€ aproximados de gasto adicional total.

7. Sostenibilidad

Es importante calcular la sostenibilidad del proyecto desde el punto de vista económico, ambiental y social. En el siguiente apartado escribo una autoevaluación sobre el conocimiento de la competencia de sostenibilidad, seguido de un enfoque en las tres dimensiones mencionadas anteriormente.

7.1. Autoevaluación del conocimiento de la competencia de sostenibilidad

Como ya se ha explicado anteriormente, existen tres ámbitos en los que se puede evaluar la sostenibilidad, desde el punto de vista medioambiental, social y económico.

El punto de vista medioambiental, según mi entender, es uno de los puntos cruciales hoy en día para la sociedad y con razón, puesto que usamos muchos más recursos de los que la tierra puede producir. Con lo que, si seguimos a este ritmo de consumo de recursos, el desastre es inevitable. Es por esto que en este trabajo se han intentado consumir exclusivamente los recursos necesarios, intentando ahorrar los máximos recursos posibles.

Por otro lado, nos encontramos con el punto de vista social, el cual es sin duda el más importante de este trabajo, puesto que este proyecto tiene un fin social y no uno económico, como es el de mantener a la población informada acerca de la privacidad en internet. Esto, desde mi punto de vista, beneficia a la sociedad, aportando al individuo información para poder tomar decisiones críticas al respecto.

Por último, las dimensiones del ámbito económico pasan prácticamente desapercibidas, puesto que este trabajo no tiene fines lucrativos y, en este aspecto, solo se preocupa de gestionar los gastos económicos de la forma más eficiente posible, es decir, evitando gastos superfluos.

En resumen, este trabajo consta de grandes dimensiones sociales, puesto que su objetivo se podría considerar ético. Este ámbito es tan relevante en este trabajo que eclipsa los otros dos ámbitos, pese a que también son cruciales.

7.2. Dimensión Económica

El coste que conlleva este proyecto me parece adecuado, teniendo en cuenta que no es un coste excesivo y que se es muy previsor con los posibles riesgos que conllevan costes adicionales.

Además, este proyecto podría dar a conocer los sitios web que no cumplen con la regulación de protección de datos y, por lo tanto, representaría multas de gran tamaño para

las empresas responsables. Estas multas irían principalmente destinadas a la indemnización de las víctimas.

7.3. Dimensión Ambiental

El coste ambiental de este proyecto es muy bajo, dado que solo se consumen los mínimos recursos de hardware que cualquier proyecto informático debe considerar. Aun así, en la realización de este proyecto se intenta minimizar el impacto ambiental al mínimo.

De hecho, los servidores usados para mantener el ORM han sido completamente reutilizados. Creo que esto muestra el compromiso de los integrantes de este grupo de desarrollo con el planeta.

A pequeña escala, creo que hago todo lo que está en mi mano para no hacer daño al medio ambiente. Por ejemplo, las reuniones las hacemos de forma remota para evitar desplazamientos innecesarios, ahorro energía del ordenador cuando no lo uso, etc.

7.4. Dimensión Social

Esta es la dimensión en la que este trabajo cobra más peso, pues el objetivo de este es concienciar a los usuarios, mediante el uso de datos, del cumplimiento de la privacidad por parte de las empresas, ya que, como se ha mencionado en capítulos anteriores, es sabido el abuso que estas ejercen sobre el usuario. Por lo tanto, por cuestiones éticas, creo que este trabajo es imprescindible y puede ser determinante en un futuro, puesto que puede contribuir al conocimiento de los usuarios acerca de estas prácticas y al fin y al cabo somos los usuarios los responsables de provocar cambios en la regulación y en las políticas de empresa. Creo que es muy importante este concepto, ya que creo que gracias a este proyecto el usuario tendrá más poder de decisión y, gracias al conocimiento que tendrán disponible, habrá muchos aspectos que al navegar por internet entenderán y, por lo tanto, serán menos débiles o manipulables.

Por otro lado, en el aspecto personal, este proyecto ha enriquecido mis conocimientos de visión artificial y de deep learning, aspectos que encuentro apasionantes y con mucho potencial. Este conocimiento adquirido ha despertado en mí las ganas de dedicarme profesionalmente a la Inteligencia artificial. Además, al ser un trabajo desde mi punto de vista ético, trabajo con más entusiasmo.

8. Evolución de la planificación

Tras haber explicado la planificación que se quería seguir para realizar este proyecto, este capítulo va destinado a analizar la situación actual, con la parte de desarrollo ya finalizada. Así, en este capítulo se explica la evolución del proyecto y las desviaciones con respecto a las planificaciones iniciales .

Inicialmente la planificación se dividió en dos partes muy diferenciadas: la parte de investigación y la de desarrollo. Esta división se ha mantenido hasta el día de hoy. Por otro lado, la parte de desarrollo se podía dividir en dos partes, la parte de visión por computación y la parte de machine learning. Esta última parte tenía como objetivo mejorar la precisión de la detección de botones realizada por la parte de visión artificial. Para esto, se había planteado el uso de técnicas como el reinforcement learning, ya que no son algoritmos supervisados.

Finalmente, por motivos de practicidad, he implementado una red neuronal convolucional para obtener una mejora en la precisión en vez de utilizar algoritmos no supervisados, como se había acordado en un principio. Esto se debe a que inicialmente era inviable el uso de aprendizaje automático supervisado, ya que este requiere de largos conjuntos de datos que no podía generar por falta de recursos, dado que estos conjuntos de datos se tenían que generar manualmente. Pese a esto, gracias a los algoritmos de visión artificial, he sido capaz de automatizar este proceso y el único proceso manual es la supervisión de los datos generados, ya que estos contienen errores. Aún así, al ser la generación de datos automática, se reduce mucho el tiempo, por lo que esta opción no sólo ha pasado a ser viable, sino también la más adecuada, puesto que otras técnicas como el reinforcement learning conllevan un mayor tiempo de estudio y una mayor complejidad.

Por otro lado, aunque ya se explicó que la parte importante de este proyecto era la de aceptar el formulario, se planteó la posibilidad de extender la funcionalidad del programa para ser capaz de rechazar el formulario de consentimiento. Tras un estudio más exhaustivo, vimos que este planteamiento del proyecto suponía varios problemas.

En concreto, observamos que la mayoría de formularios no tienen opción de rechazarlo o al menos no tienen la opción directa de hacerlo, dado que muchos formularios tienen una opción para configurar las cookies y allí, normalmente después de seguir unos pasos, existe la posibilidad de desactivar las cookies que el usuario desee. Esto supone un problema para el proyecto ya que, si el botón de rechazar no existe no se puede detectar y, si existe pero es muy complejo llegar hasta él porque hay pasos intermedios, el conjunto de posibles escenarios se multiplica, haciendo el problema inviable.

Como solución, se ha decidido, que la herramienta se encargue únicamente de identificar la opción de aceptar el formulario. Pese a esto, el algoritmo también es capaz de rechazarlo en los casos que se muestre esta opción como una opción directa, es decir, que con un solo click al botón adecuado se pueda rechazar el formulario, sin necesidad de seguir más pasos. Esta funcionalidad, aunque existe, no se ha usado en prácticamente ninguna ocasión, ya que no es válida para la mayoría de casos.

Además, tras un estudio por parte del equipo de desarrollo del ORM, se descubrieron métodos alternativos para el bloqueo de cookies, que se basaban en bloquear la obtención de estos recursos. También se podría solucionar el rechazo automático del formulario de consentimiento con el plugin Ninja Cookies (explicado en el capítulo de soluciones existentes), el cual tiene esta misma finalidad, pese a que no es muy efectiva.

Aún así, insisto en que la parte importante es la de aceptar el formulario de consentimiento, ya que lo que se quiere lograr con esta herramienta es averiguar si los sitios web hacen uso de cookies no esenciales aún cuando el usuario no ha aceptado el formulario, ya que en principio, las opciones de rechazar y no actuar son equivalentes a ojos de la regulación y el único caso en el que se puede recopilar esta información es cuando el usuario da su consentimiento explícito.

En cuanto al diagrama de Gantt, donde se especifica el reparto de tareas en el tiempo, se determinó que combinaría las tareas de visión artificial y Machine Learning. Finalmente no ha sido así, ya que primero se ha realizado el grueso de la parte de visión artificial para más tarde mejorarla con deep learning. Aun así, me gustaría destacar que esto no ha modificado las tareas ni el grueso de estas, por lo que tampoco supone una alteración en los costes, simplemente se ha alterado el orden de estas. Este diagrama se puede ver en el anexo, donde ya ha sido actualizado.

En definitiva y, a modo de resumen, este proyecto ha cambiado ligeramente y los objetivos de este se han visto actualizados. Más en concreto, el objetivo final de este proyecto es el de automatizar la manipulación del formulario de consentimiento, haciendo énfasis en el proceso de aceptación del formulario, dado que el proceso de rechazo es en muchas casos inexistente o demasiado complejo como para ser tratado con el enfoque de este proyecto. Esta herramienta se ha conseguido implementar gracias a la generación de un dataset etiquetado que sirve para entrenar un modelo de redes neuronales que será el encargado de la toma de decisiones de la herramienta.

9. Entorno de trabajo

En este capítulo comentaremos las principales herramientas utilizadas para poder llevar a cabo este trabajo.

Este trabajo ha sido realizado en el sistema operativo **Windows** por motivos de compatibilidad con algunas de las herramientas. Además, este cuenta con **Visual Studio**, un entorno de desarrollo integrado de Microsoft que facilita la producción de software.

Todo el trabajo ha sido desarrollado en el lenguaje de programación **Python**. Inicialmente se empezó a programar en C++, ya que es un lenguaje con el que me siento muy cómodo y, a priori, todas las librerías que pretendía usar estaban disponibles para este lenguaje. Finalmente, me acabé decantando por python por los siguientes motivos:

- Las principales librerías de redes neuronales estaban escritas en python, y, aunque en muchos casos existen versiones de estas librerías en C++, son más complejas de integrar y usar.

- Existen muchas librerías de Visión artificial de python que no están desarrolladas en C++.

- Existen métodos muy sencillos que permiten manipular un navegador web desde python, también los hay para hacer capturas de pantalla. Esto no sucede en C++, donde conseguir esto conlleva una mayor complejidad.

Para el mantenimiento del programa y el control de versiones, he utilizado el gestor de repositorios git (para más información acerca de la gestión de versiones, consultar *[16]*). Más en concreto, he utilizado el repositorio Github *[17]* y su programa Github Desktop para gestionar las versiones. He utilizado Github porque es una herramienta que ya conocía y he usado tanto en mi puesto laboral como para proyectos personales.

Las principales librerías utilizadas para la parte de visión artificial son las siguientes:

- **OpenCV** *[18]*: esta es una de las librerías más utilizadas de Visión artificial, es de código abierto y fue desarrollada por Intel. He decidido utilizar esta librería ya que la conocía de antiguos proyectos que he realizado. Además, creo que es una de las más completas que existen. Esta librería tiene funciones para marcar imágenes, manipular los píxeles de la imagen uno a uno, aplicar filtros y otros algoritmos conocidos, como es el algoritmo de detección de contornos de Canny, el cual es primordial para este proyecto. Para saber más acerca de este algoritmo, véase *[19]*.

- **Tesseract** *[20]*: este es el *OCR* implementado por Google. Como veremos en el capítulo de Desarrollo, esta herramienta ha sido vital para este trabajo. He escogido Tesseract dado que es de código abierto y ha sido implementado por Google, por lo que es uno de los mejores OCR que existen en el momento.

Por otro lado, las principales librerías que he utilizado para la parte de Machine Learning han sido las siguientes:

- **YOLO**: este es un sistema de código abierto enfocado a la detección de objetos en tiempo real. Se encuentra contenido dentro del framework **Darknet** *[21]*, el cual es uno de los frameworks más rápidos de redes neuronales. El algoritmo de YOLO consiste en crear una red neuronal convolucional para detectar los objetos deseados en imágenes, en este caso los botones del formulario de consentimiento.

- **TensorFlow** *[22]*: este es un framework necesario para poder ejecutar los algoritmos de YOLO, es una de las principales librerías de aprendizaje automático que existen.

Estos algoritmos de Deep learning son muy costosos, ya que gastan muchos recursos. Para atenuar este inconveniente, he entrenar la red neuronal en **Google Collab**, el cual es un entorno virtual en el que puedes usar los recursos que ofrece Google, los cuales son mucho más potentes que los que poseo en mi ordenador particular.

Finalmente, he usado otras herramientas que me parecen de especial relevancia pese a que no intervengan directamente en los algoritmos de detección:

- **Pyautogui**: esta es una librería diseñada exclusivamente para Python. Tiene muchas funcionalidades, de las cuales yo he usado principalmente las de captura de pantalla y las que permiten mover y seleccionar con el ratón los elementos deseados.

- **WebBrowser**: es una librería para Python que permite abrir y cargar direcciones URL en el navegador deseado.

10. Marco y contexto del proyecto

Como ya se ha explicado anteriormente, este trabajo forma parte de el ORM, un proyecto promovido por el grupo de investigación CBA en el que participan otros estudiantes e ingenieros. El ORM muestra estadísticas y otros tipos de datos acerca de la información personal que almacenan los dominios web.

Este capítulo pretende ser una guía para el lector de las funcionalidades del ORM, además de ofrecer una concreción acerca de cuál es el papel que juega este TFG dentro del proyecto.

Pero antes de empezar, me gustaría aclarar que el ORM es un programa de consola, por lo que no tiene interfaz. Es por esto que el equipo de desarrollo CBA implementó el ePrivacy Observatory *[26]*, que tiene como fin presentar adecuadamente los resultados del ORM. Además, el ePrivacy Observatory presta sus servicios a todos los usuarios que dispongan de acceso a internet, sin necesidad de tener que instalar y configurar el ORM en sus máquinas, lo que hace que esta herramienta sea usable para el público general, que es a quién va enfocado.

El ePrivacy Observatory es un dominio web que ofrece información acerca de la privacidad del usuario. En concreto, ofrece dos tipos de datos: datos genéricos, que se han recopilado a partir del conjunto de webs que los usuarios de ORM han buscado y datos específicos, que se obtienen de consultar el sitio web que el usuario desee.

10.1. Interfaz y guía de uso

En cuanto a la interfaz, la página principal ofrece en su parte superior una bóveda de estadísticas que van cambiando. Estas estadísticas se obtienen del conjunto de análisis que realiza ORM a las páginas que los usuarios buscan. Esta parte de la interfz se puede apreciar en la figura 2.

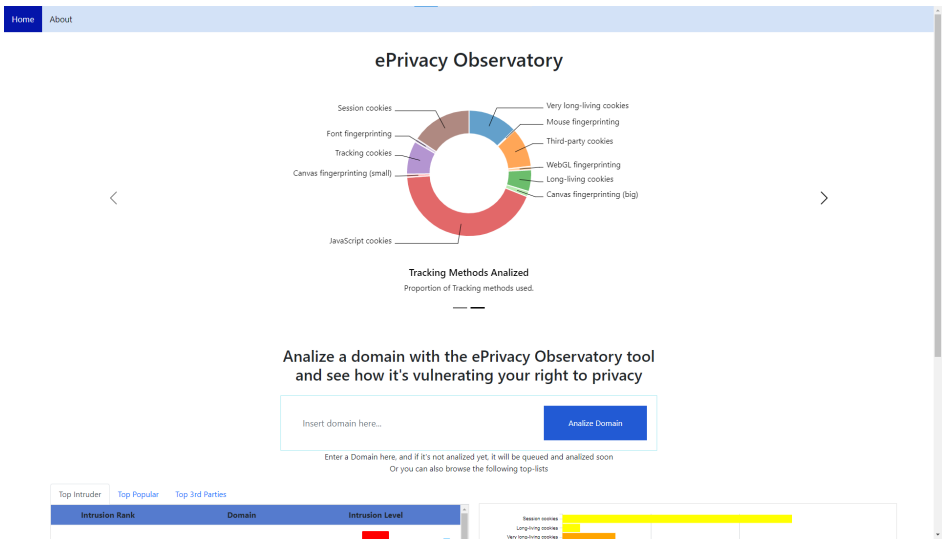

***Figura 2**: Página principal del ePrivacy Observatory. **Fuente**: http://tars.cba.upc.edu.*

Justo debajo de esta bóveda, nos encontramos con un buscador de dominios, donde el usuario tiene la opción de introducir el enlace de la página web que desee analizar. Una vez buscamos el dominio, nos aparece una pantalla de análisis que nos ofrece información acerca de los recursos que recopila este dominio (véase la figura 3). Se entiende por recursos toda la información que almacena un sitio web acerca de un usuario, no tiene porqué tratarse de información sensible. En esta pantalla, se muestran datos relativos al dominio visitado, como el nivel de intrusión (puntuación que refleja el número de recursos que son invasivos para el usuario y son usados por el dominio), el tipo de cookies que usa, así como otra información relevante, como la proporción de los recursos que son de terceros.

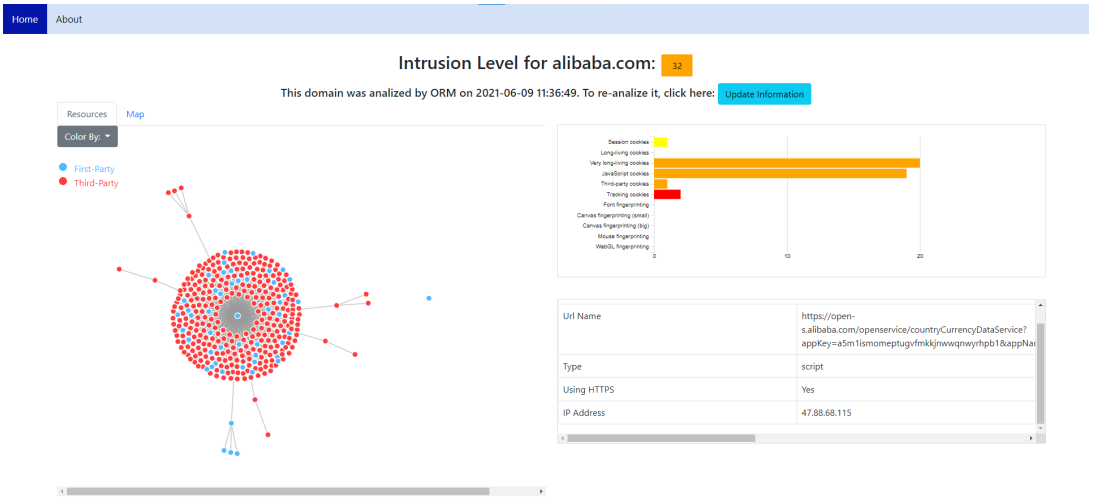

***Figura 3**: Resultado del análisis de ORM al dominio alibaba.com. **Fuente**: http://tars.cba.upc.edu.*

Finalmente, si nos desplazamos por la página principal, veremos otro tipo de información genérica, como el ranking de páginas más invasivas o el ranking de las páginas que usan más recursos de terceros.

10.2. ¿Cómo influye este proyecto en el ORM?

Como ya hemos visto, el ORM está siendo desarrollado por diversos estudiantes y profesionales. Muchos de estos estudiantes también están realizando su TFG o TFM sobre el ORM. En concreto, la alumna del máster en ciberseguridad Meritxell Basart tiene como trabajo de fin de máster estudiar hasta qué punto los sitios web cumplen con la regulación europea de privacidad. Esta regulación requiere a los dominios web de un consentimiento explícito por parte del usuario si quieren recopilar información del usuario. Así, el trabajo de mi compañera consiste en navegar de forma automática por la web y estudiar los siguientes tres casos para cada dominio que visita:

- En el primer caso, se visita el dominio sin hacer ninguna acción. El ORM analiza los recursos que ha usado.

- En este caso, se visita el dominio con el plugin Ninja Cookies, ya explicado en la sección de soluciones existentes. Este plugin intenta rechazar el formulario de consentimiento para que más tarde el ORM analize los recursos usados
- En el último escenario, se visita la página y se aplica el algoritmo que he desarrollado en este proyecto, el cual intenta aceptar el formulario. Luego, el ORM analiza los recursos que ha usado el dominio.

Una vez ha examinado estos tres escenarios de un dominio concreto, los compara para obtener conclusiones acerca de la legalidad de la empresa. Según el GDPR, los dos primeros casos mencionados anteriormente tendrían que recopilar la misma información, puesto que en ningún momento se ha aceptado explícitamente el formulario de privacidad. Por otro lado, en el último escenario se tendría que recopilar más información que en los dos primeros.

Las conclusiones de este estudio aún no se han publicado puesto que la autora aún no lo ha terminado. Aun así, una vez Meritxell haya finalizado el estudio, el ePrivacy Observatory mostrará para cada dominio si este cumple con el GDPR. Esta información también se mostrará a nivel general, indicando la proporción de dominios analizados que cumplen con la regulación.

11. Desarrollo de la herramienta

En este capítulo se describe por orden cronológico los avances y la evolución del proyecto, en la que se detalla la parte de desarrollo. El objetivo de explicarlo de este modo es, por un lado, asegurar un entendimiento de la evolución del proyecto por parte del lector y por otro lado, poder justificar con mayor facilidad la toma de decisiones en cada momento.

11.1. Primeros pasos

El primer paso de este proyecto fue intentar, a modo de toma de contacto y, mediante algoritmos de visión artificial, desarrollar un programa que sea capaz de identificar el formulario de consentimiento en una captura de pantalla de una página web (esto no incluye detectar los botones del formulario, solo el formulario en sí). Esta primera versión del programa recibía como parámetro la ubicación del equipo donde se almacenaba la captura de pantalla y tenía que modificar la imagen para señalar, con un rectángulo verde, el formulario de consentimiento. La intención de hacer esto es reducir el número de candidatos a ser botones para manipular el formulario. Por lo tanto, el paso a seguir sería buscar, dentro del formulario de consentimiento (la zona señalada), los botones de manipulación del formulario, que serán más fáciles de encontrar que si buscamos en toda la imagen, ya que aparecen muchos más botones.

La resolución de este problema era viable dado que tenía la hipótesis infundada de que la mayoría de sitios webs muestran el formulario de consentimiento de una forma llamativa, haciendo que destaque entre el resto de elementos (como es el caso de la figura 4). Esta hipótesis se me ocurrió gracias al estudio previo que hice, donde observé como se mostraba el formulario de consentimiento en las páginas web más visitadas del mundo. Dando por hecho que esta hipótesis sea cierta, la resolución por visión artificial del problema parecía viable y consistía en los siguientes pasos:

1. Leer la imagen y aplicar un preproceso. Este preproceso consiste en preparar la imagen para que al algoritmo de detección le sea más fácil obtener el resultado. En este caso, el algoritmo tiene que detectar los contornos o los bordes que contienen el formulario de consentimiento, por esto, uno de los pasos de preproceso consiste en aplicar sobre la imagen un filtro que acentúa y resalta los bordes de las figuras de la imagen, es decir todo lo contrario al proceso de suavizar una imagen, donde los contornos pierden su definición y la imagen se ve más borrosa.

2. Aplicar los algoritmos de detección. Esto consiste en aplicar técnicas de visión artificial para extraer características que nos puedan dar información sobre qué es un formulario de consentimiento y qué no. Inicialmente esta tarea fue sencilla, dado que el algoritmo de detección de contornos de Canny es muy eficiente para este tipo de problemas, donde existen unos contornos muy bien definidos en una imagen. A esto se le aplica un filtro para eliminar los posibles falsos positivos, cosa que no suponía ningún problema dado que en todas las imágenes probadas hasta el

momento tenían muy distinguido el formulario de consentimiento con respecto al resto de elementos con los que el algoritmo se podía confundir.

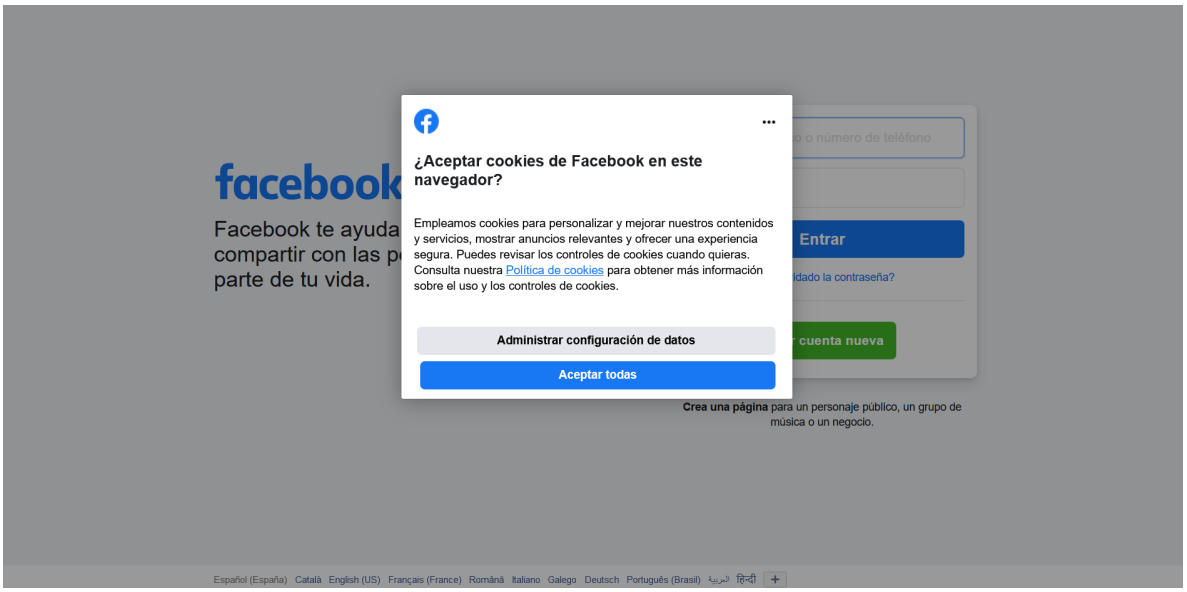

***Figura 4**: Página de inicio de Facebook.com. El formulario de consentimiento es lo que más resalta, es fácil detectar sus contornos. **Fuente**: Elaboración propia.*

Al final, este algoritmo se implementó con éxito, al menos sobre el conjunto de imágenes limitadas que probamos. El siguiente paso fue extrapolar este algoritmo a un número de imágenes superior, que contuviese tipos muy distintos de sitios web.

Para conseguir capturas de más sitios web, necesitaba automatizar el proceso, dado que hasta el momento obtenía las capturas manualmente. Este programa de automatización no fue muy complejo de implementar y los pasos que seguí para lograrlo son los siguientes:

- Investigación y análisis de repositorios que contienen listas extensas de páginas web. Acabe decantándome por el servicio de Amazon llamado Alexa Top Sites (ATS) *[23]*, el cual posee una lista que contiene las páginas más visitadas por el asistente virtual de Amazon. En concreto, obtuve una lista del millón de páginas más visitadas por Alexa en el último mes. Esta lista se tuvo que limpiar con un programa auxiliar que desarrollé, el cual examina la lista y descarta los dominios que no son españoles (acabados en “es”). Esto lo hice por dos motivos: el primero, asegurarse de que todas las páginas de la lista están sometidas a la regulación de la Unión Europea, puesto que las páginas no europeas legalmente no tienen porque presentar el formulario de consentimiento. Y, como segundo motivo, escoger páginas web que estén escritas en el alfabeto latino, para más tarde poder diferenciar entre las componentes del formulario gracias a la detección de texto y palabras clave. Destacar que esto no quiere decir examinar únicamente sitios web en español, dado que hay muchas páginas con este dominio que tienen el formulario en inglés o otros idiomas vecinos.

- Una vez obtenida la lista de páginas web, implementé un programa que leía esta lista y, para cada página que leía, la abría en el navegador y realizaba una captura de pantalla que guardaba. Todo este proceso automático lo conseguí gracias a las librerías webbrowser y pyautogui de Python.

Una vez fui capaz de tener un conjunto de datos más amplio, volví a probar el programa de detección de formulario. Para mi sorpresa, perdía efectividad. Esto se debe a que mi hipótesis era falsa: no todas las páginas web resaltan el formulario de consentimiento. Si bien es cierto que las principales páginas web lo hacen (que son en las que había probado inicialmente), existen algunas páginas menos conocidas que no, por lo que el algoritmo fallaba. Además, existen páginas más invasivas que resaltan otros mensajes, como un formulario de suscripción, confirmación de mayoría de edad, una solicitud para poder recibir notificaciones, etc. Por último, también había casos en los que estos tipo de mensajes resaltados compartían el mismo formato que el formulario de consentimiento, por lo que es imposible distinguirlos con las técnicas que había aplicado hasta el momento. Como se puede apreciar en la figura 5, se observa como destaca más el formulario de subscripción que el formulario de privacidad, apenas visible en la parte inferior de la imagen, por lo que el algoritmo falla en este caso.

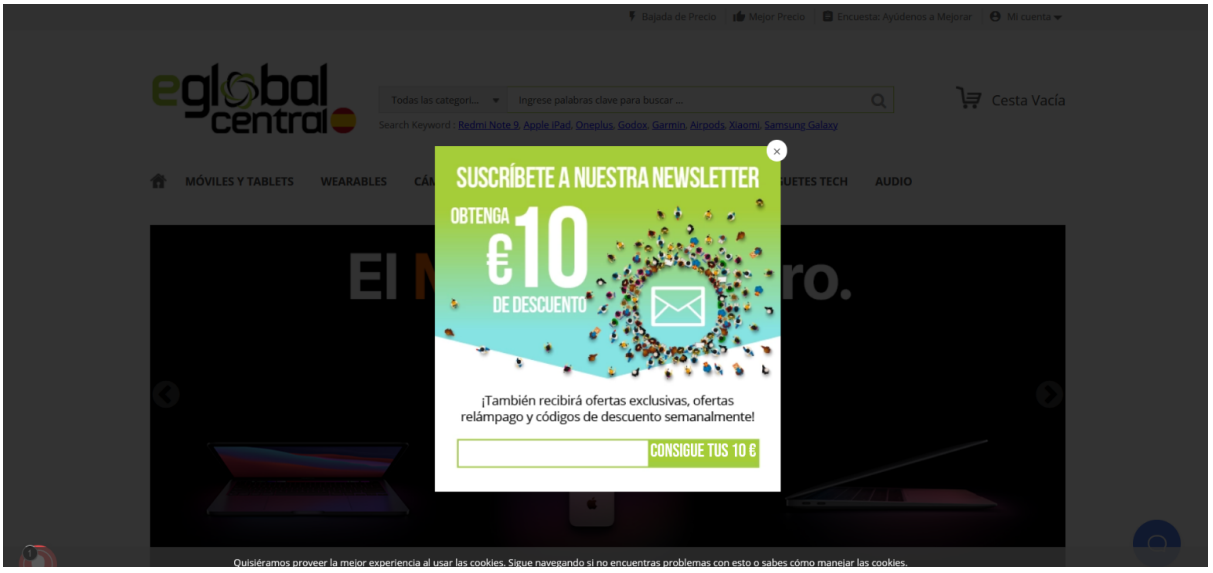

***Figura 5**: Página de inicio de Eglobalcentral.es. **Fuente**: eglobalcentral.es.*

Pese a que los casos problemáticos eran pocos, se decidió enfocar el algoritmo de otra forma ya que estos obstáculos eran intrínsecos de la solución que había propuesto y, por lo tanto, irremediables.

Aún así, cabe destacar que el objetivo de implementar esta primera solución era el de familiarizarme con el entorno de trabajo y orientar una primera solución. Por esto creo que los objetivos se alcanzaron.

11.2. Primera solución al problema

Como consecuencia de los resultados no favorables del primer programa, pensé en otras formas de enfocar el problema, ya que la resolución propuesta no estaba bien encaminada. Al final decidimos orientar el nuevo programa como un problema de reconocimiento de texto.

Más en concreto, la intención de este programa es escanear las palabras de la imagen, para luego ser filtradas con una lista de palabras clave pre-establecidas. Estas palabras clave se han ido añadiendo a medida que se ha ido avanzando en el proyecto, aunque inicialmente se usaron las palabras clave de este estudio anteriormente mencionado *[4]*. Este escáner de texto es conocido como OCR, explicado en conceptos. Siendo más específicos, usé el OCR de Google, Tesseract, explicado en el capítulo de entorno de trabajo

Esta solución conlleva ventajas y desventajas. Por un lado, el programa ya no tiene que detectar el formulario por su forma, ni tampoco los botones que lo forman, ahora solo tendrá que detectar el texto de estos botones, que se pueden encontrar en toda la imagen. Por otro lado, esta solución tiene los siguientes inconvenientes:

- Dependiendo del lenguaje con el que esté configurado el OCR, se obtienen distintos resultados. Esto supondría un gran problema si se tuvieran que tratar textos en idiomas con alfabetos distintos. En este caso, no tuve mayor dificultad ya que prácticamente todas las páginas recopiladas están escritas con el alfabeto latino.

- La detección de texto es muy sensible, es difícil detectar textos en condiciones visuales complejas, por ejemplo, si es un texto blanco sobre colores oscuros, si la tipografía es muy fina, etc. Esto se puede solucionar con un algoritmo de preproceso que intensifique las propiedades de la imagen que deseadas.

- El programa no tiene porqué encontrar un resultado único, de hecho, en la mayoría de casos se detectaba más de una posible solución.

A continuación, explicaré de una forma más extensa la solución a los dos últimos inconvenientes recién explicados.

Para solucionar el primer inconveniente y hacer el algoritmo más robusto, en vez de tratar una única imagen, lo que hice fue transformar esta imagen de entrada para obtener imágenes ligeramente distintas y poderlas re-evaluar. En concreto, transformaba la imagen en escala de grises y acto seguido la binarizaba. Binarizar una imagen significa transformarla para que esta acabe conteniendo únicamente dos colores: blanco y negro. Esto lo hice a partir de la imagen en escala de grises, donde cada píxel está representado con un valor del 0 al 255 (en este sistema de representación sólo existe un canal, el cual determina el nivel de gris de cada píxel, como se puede apreciar en la figura 6). Sobre esta imagen, apliqué la siguiente condición: los píxeles que tengan un valor inferior a un número concreto, los pintaba de negro y el resto los pintaba de blanco. El valor del número concreto suele ser la mitad del rango de valores que puede adoptar un píxel (en este caso 127),

aunque este valor se puede modificar según el tipo de imágenes que estés tratando, como fué mi caso. También ayudó en algunos casos tratar la imagen en un sistema de representación distinto para trabajar sobre un único canal, es decir, transformar la imagen de RGB a HSV y trabajar sólo con el canal del matiz (los espacios de color están explicados en conceptos). Además de estas técnicas, apliqué otras que aprendí en la asignatura de Visión por Computación, de la FIB, como tratar de resaltar los detalles de la imagen o tratar de hacer más gruesas las letras (operación morfológica de dilatación *[24]*).

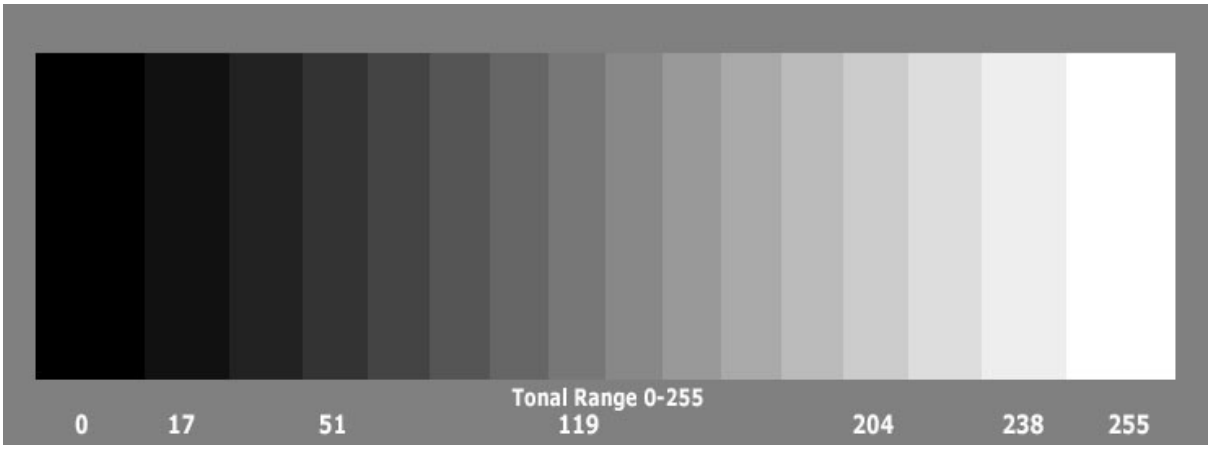

Figura 6: *Rango de la escala de grises*. ***Fuente****: https://improvephotography.com*.

Como solución al último inconveniente, en el que las palabras clave aparecen más de una vez en un sitio web, tuvimos que replantear el funcionamiento del programa por completo. Hasta ahora, el programa leía capturas de pantalla y devolvía la misma captura modificada, indicando las partes de interés. El inconveniente de este planteamiento es que no se puede interactuar con la página web y, se me ocurrió que, para descartar candidatos, se podría pulsar sobre todas las palabras clave para ver si la página cambiaba de aspecto, ya que pulsar en los botones del formulario de consentimiento hacen que este desaparezca.

Así, el nuevo programa recibía una lista de dominios que examinaba uno por uno y, para cada dominio, cargaba la página web, realizaba una captura de pantalla al cabo de unos segundos y la analizaba con el preproceso y el OCR ya mencionados. Acto seguido. el programa simula un click sobre todas las posiciones donde se encuentran las palabras clave y, para cada palabra que pulse, realiza una nueva captura en la que analiza que porcentaje ha cambiado con respecto a la imagen original. En función de esto se determina si es un botón o no. Si es un botón y además contiene una palabra clave, es muy probable que sea el botón del formulario de consentimiento. Pese a esto, hay casos en los que esto falla, dado que también existen botones que no son del formulario y que contienen estas palabras clave, pero estos casos eran mínimos.

A base de probar el algoritmo muchas veces, logré ajustar el programa con una configuración concreta que conseguía una efectividad de un 77% de acierto a la hora de encontrar el botón. Es decir, en el 77% de los casos donde existía el botón de aceptar, este se encontraba.

11.3. Mejorando la efectividad con Machine Learning

Una vez finalizado el programa que acabo de explicar, el siguiente paso era intentar mejorar esta efectividad del 77% con Machine Learning. Pero antes, me gustaría explicar algunos conceptos de esta disciplina, de modo que se entienda mejor la explicación que sigue.

- **Algoritmo supervisado**: es un algoritmo que requiere de ejemplos resueltos para poder aprender, estos ejemplos resueltos son conocidos como datasets etiquetados. Por ejemplo, en este problema concreto, un dataset etiquetado sería un conjunto de imágenes junto a la posición donde se encuentra el botón de cada imagen. De hecho, estos algoritmos estudian los ejemplos resueltos para, a partir de estos, encontrar un patrón que los resuelva. Este patrón se podrá extrapolar para resolver nuevos problemas cuya solución no sabemos.

- **Reinforcement Learning**: es un área del Machine Learning basada en un sistema de prueba y error. Al no ser supervisado, no tiene un dataset etiquetado del que estudiar, por lo que es el usuario quien determina cuando se está alejando de la solución deseada y cuando se está acercando. A partir de esto, el algoritmo trata de acercarse lo máximo posible a la solución.

- **Deep Learning**: es otra área del Machine Learning que puede ser supervisado o no supervisado. Estos algoritmos funcionan usando una red neuronal artificial que se compone por niveles jerárquicos.

- **Red Neuronal artificial**: se trata de un conjunto de nodos, conocidos como neuronas, que se comunican entre sí para transmitir información. En un problema de deep learning, la entrada del problema (en nuestro caso, la captura del sitio web sin marcar) atraviesa la red neuronal, sufriendo pequeñas alteraciones en cada neurona que atraviesa. Si nos fijamos en la figura 7, la entrada llega a los nodos de la parte izquierda y estos la alteran y la distribuyen hacia los demás nodos. Cuando llega a la neurona final, encontramos el resultado (en nuestro caso, la captura con los botones marcados). La información que transmiten las neuronas es ponderada, es decir, dependiendo de la neurona la información que esta transmita tendrá mayor o menor importancia. Estos pesos se ven reflejados por el grosor de las flechas de la figura 7. Si al llegar al final no se ha obtenido el resultado esperado, se hace una propagación hacia atrás del error y se redistribuyen los pesos para ver si la solución mejora. Cada propagación de error es un paso de aprendizaje. Cuando la red neuronal no puede aprender más, esa distribución de la red neuronal se guarda como un modelo. Este modelo es el que se usa para resolver nuevos problemas.

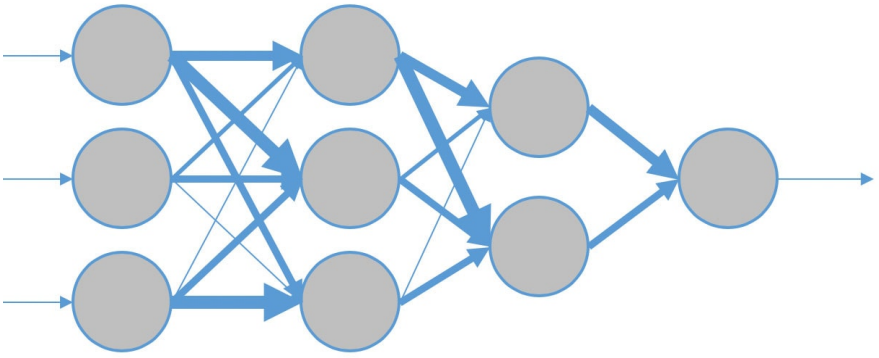

***Figura 7**: Esquema de una red neuronal.* ***Fuente:** https://tex.stackexchange.com*

Con estos conceptos ya explicados, detallaré el proceso de mejora del algoritmo con Machine Learning.

Inicialmente, la idea era utilizar técnicas de reinforcement learning, ya que estos tipos de algoritmos no requieren de un conjunto de datos etiquetados. La intención era evitar un dataset etiquetado, puesto que necesitábamos un dataset muy extenso debido al tipo de problema y este dataset lo teníamos que obtener de forma manual.

Aunque esta fue la idea inicial, no llegué a implementar ningún algoritmo de reinforcement learning, ya que pensamos que sería más interesante utilizar técnicas supervisadas con un dataset generado de forma automática. Así, se nos ocurrió aprovechar el programa de visión artificial implementado anteriormente para generar un conjunto de imágenes junto con las localizaciones de sus respectivos botones. Esto suponía dos inconvenientes, que explico a continuación:

- El programa anterior tiene una efectividad del 77%, esto supone que si queremos crear un dataset etiquetado con 3000 imágenes (que era la intención inicial), unas 690 contendrán errores. Esto lo solucionamos revisando manualmente todas las imágenes y etiquetando manualmente las erróneas. Quiero destacar que la acción de revisar imágenes es rápida, lo complejo es etiquetarlas manualmente y, al haber relativamente pocas imágenes a etiquetar (solo los errores), no suponía un mayor problema.

- Aunque para el programa anterior ya era suficiente con que encontrase la palabra clave, dado que tan solo se tenía que pulsar sobre esta, para generar el dataset etiquetado requería que el programa localizase el botón, no solo la palabra clave. Esto supone una mejora en el entrenamiento de la red neuronal, ya que no es lo mismo entrenar a una red para encontrar un botón (que en muchos casos se parecen), que entrenarla para encontrar una palabra concreta, que en muchos casos estará representada de una forma distinta (diferente fuente y tamaño de letra, sobre un fondo diferente, etc.). Además, habría casos donde detectaría la palabra pero esta no sería solución(recordemos que no todas las palabras clave son solución). Así, lo que se hizo fue modificar el programa de tal forma que, una vez encontrada la palabra clave que es solución, se busca el botón en el que está contenida con el

algoritmo de contornos de Canny. Una vez encontrado el botón que lo contien, se marca.

Una vez solucionados estos problemas, generé el dataset etiquetado, que estaba formado por unas 2700 imágenes. El siguiente paso fue crear y entrenar una red neuronal con el dataset obtenido. Para entrenar la red, que es computacionalmente muy costoso, usé la herramienta Colab, de Google. Esta herramienta, ya explicada en el apartado de entorno de trabajo, permite ejecutar tu código en una CPU o GPU remota que te ofrece Google.

Existen muchos tipos de redes neuronales, pero para la detección de objetos en imágenes se suelen usar las CNN (**C**onvolutional **N**eural **N**etwork) *[25]*. Estas redes se suelen usar en este tipo de datos ya que usan capas convolucionales en 2D, lo que hace que esta arquitectura sea muy adecuada para procesar datos bidimensionales, como lo son las imágenes, que se representan en un plano de coordenadas bidimensional. Tras estudiar distintos algoritmos, me acabé decantando por YOLO, un algoritmo del framework Darknet, que usa TensorFlow para crear y entrenar las redes neuronales. El motivo de esta elección han sido principalmente dos:

- Es uno de los algoritmos más rápidos y precisos que existen para este propósito, además con YOLO no es necesario bajar la resolución de las imágenes, cosa que muchos otros algoritmos requieren, ya que bajar la resolución de una imagen no suele ser muy perjudicial para la detección de objetos y conlleva mucho más tiempo de entrenamiento. Pero en mi caso, bajar la resolución de las imágenes era una acción prohibitiva ya que los botones a detectar, al ser tan pequeños (y por lo tanto, ocupar pocos píxeles) pierden su forma si esta se ve más reducida todavía.

- Ya tenía experiencia previa con esta herramienta, dado que en mi actual puesto de trabajo lo he usado ocasionalmente para el reconocimiento facial, por lo que ya conocía su eficiencia y precisión.

Una de las claves de YOLO es que dispone de diversos modelos pre-entrenados en los que ya se han determinado los pesos de cada neurona. De este modo no se parte desde cero y el algoritmo tan solo tiene que redistribuir los pesos durante el entrenamiento para adaptarse al dataset etiquetado. Así y, tras probar varias ponderaciones de peso, conseguí crear un modelo con un 88.5% de aciertos.

Finalmente, se me ocurrió una forma de mejorar aún más el algoritmo: mezclar el modelo neuronal con el algoritmo de visión artificial. El programa final consiste en usar el modelo generado para determinar dónde se encuentran los botones del formulario. Si no detecta nada o detecta el botón pero con una confianza baja, volvemos a analizar el dominio con el programa que generaba el dataset etiquetado que, a modo de recordatorio, buscaba los botones por palabras clave.

Con esta fusión de algoritmos, he logrado una precisión del 92.23% de acierto en un conjunto de test de 386 imágenes. A continuación analizaremos en más detalle este resultado.

12. Análisis de resultados

Antes de empezar a hablar de resultados, me gustaría explicar algunos conceptos para poder interpretarlos correctamente:

- **Verdaderos/Falsos positivos**: se trata de los resultados en los que se ha marcado algo, es decir, en nuestro caso, cada supuesto botón que se haya detectado es un positivo. En este caso, se trataría de un falso positivo si se ha detectado un botón que realmente no existe, por lo que el algoritmo se ha equivocado. Decimos que un verdadero positivo es el caso en el que encuentra el botón correctamente.

- **Verdaderos/Falsos negativos**: caso contrario a los positivos. Son los casos en los que no se ha detectado ningún botón. En este caso, se trataría de un falso negativo si no ha detectado ningún botón cuando en la imagen realmente sí que había uno. Hablamos de un verdadero negativo cuando el algoritmo no detecta ningún botón, pero en este caso porque no hay ninguno en la imagen, por lo que el algoritmo ha actuado correctamente.

Una vez entendido estos conceptos, podemos observar la matriz de confusión, que no es más que una representación de los resultados:

Confusion Matrix	**Positivos**	**Negativos**
Predecido positivo	281	23
Predecido negativo	7	75

***Tabla 8**: Matriz de confusión del algoritmo final. **Fuente**: Elaboración propia.*

Como podemos apreciar, lo que más nos interesa de una matriz de confusión es tener los valores más altos posibles en la diagonal, puesto que en la diagonal se encuentran los casos correctos. Vemos que en esta matriz de confusión hay bastantes falsos negativos, es decir, en un 6% de los casos detecta un botón dónde no lo hay. Por otro lado, observamos como el número de falsos positivos es mucho más bajo, es decir, solo en un 2% de los casos no se ha detectado correctamente el botón existente.

Además, con esta tabla también podemos aproximar el número de páginas que usan un formulario de consentimiento con botón, que representa el 75% de las imágenes.

Con la matriz de confusión se pueden hacer muchos cálculos, como la accuracy, la precisión, el F1-score, etc. Estas fórmulas son más o menos significativas en función del problema se trate. Por ejemplo, un caso en el que hay muchos negativos y pocos positivos, es decir, en la mayoría de las muestras no hay que marcar nada, no se recomienda mirar la accuracy, puesto que esta es una suma de los verdaderos positivos y verdaderos negativos dividido entre el total de valores. Y vemos que en este caso, un algoritmo que nunca marque nada, tendrá una accuracy elevada, ya que la mayoría de resultados han sido verdaderos negativos. Esto solo es un ejemplo para mostrar que estos valores no tienen porque ser significativos. En mi caso, creo que la accuracy sí que es muy significativa, ya

que en mi juego de test la proporción de positivos es significativamente mayor que la de negativos. En la figura 8, muestro estos cálculos entre otros.

Measure	Value	Derivations
Sensitivity	0.9757	TPR = TP / (TP + FN)
Specificity	0.7653	SPC = TN / (FP + TN)
Precision	0.9243	PPV = TP / (TP + FP)
Negative Predictive Value	0.9146	NPV = TN / (TN + FN)
False Positive Rate	0.2347	FPR = FP / (FP + TN)
False Discovery Rate	0.0757	FDR = FP / (FP + TP)
False Negative Rate	0.0243	FNR = FN / (FN + TP)
Accuracy	0.9223	ACC = (TP + TN) / (P + N)
F1 Score	0.9493	F1 = 2TP / (2TP + FP + FN)
Matthews Correlation Coefficient	0.7885	TP*TN - FP*FN / sqrt((TP+FP)*(TP+FN)*(TN+FP)*(TN+FN))

***Figura 8**: Principales cálculos para evaluar la matriz de confusión. **Fuente**: https://onlineconfusionmatrix.com/*

13. Futuras mejoras y continuación del proyecto

En este capítulo, explico las futuras características que se pueden implementar en este proyecto si se decide seguir con el desarrollo.

En primer lugar, sería interesante comparar la efectividad de este algoritmo con las soluciones ya existentes. Esto es difícil de comparar puesto que es complejo de objetivar. Creo que la mejor aproximación a esta propuesta sería seguir con el proyecto de mi compañera, Meritxell Basart, que analiza los dominios manipulando el formulario de formas distintas para averiguar su efecto.

En segundo lugar, se podría aumentar la funcionalidad del programa para que abarcase la opción de rechazar el formulario. Creo que esto sería interesante para comparar su efectividad con otras soluciones existentes. Además sería curioso averiguar las diferencias entre el rechazo del formulario y el bloqueo directo de las cookies, donde no deberían encontrarse diferencias. Esto, en parte, lo ha estudiado Meritxell, pero, a causa de la ineficiencia de las soluciones existentes, los resultados no son del todo concluyentes.

Finalmente y, lo más importante, es incorporar esta herramienta de forma oficial al ORM. Este programa lo he diseñado con la intención de que su incorporación con el ORM sea sencilla y, en algunas versiones de prueba, ya se ha integrado. Aún así, a nivel oficial, no se ha publicado una versión con este algoritmo incorporado ya que debemos esperar a las conclusiones que obtendrá mi compañera, Meritxell Basart.

14. Conclusiones

Este trabajo ha resultado, honestamente, muy satisfactorio por diversos motivos. El primero es que, pese a los inconvenientes que hemos ido viendo a lo largo de esta memoria, creo que he sido capaz de superarlos con éxito y adaptarme. Creo que este era un punto clave de este proyecto, dado que se trataba de un trabajo que tenía una parte de investigación, por lo que era un proyecto poco definido y con un futuro bastante incierto.

En el aspecto social y ético, creo que he descubierto muchas cosas que desconocía, creo que en cierto modo he abierto mis ojos en el mundo de la privacidad, donde he descubierto que los sitios web son más invasivos de lo que me esperaba. Además, he tenido la oportunidad de trabajar con excelentes profesionales, así como con alumnos sobresalientes, de los que he aprendido mucho, tanto en el ámbito académico como en el ético.

En cuanto a las tareas realizadas, se han completado todas con éxito, es cierto que han habido ligeros repasos y cambios en el diagrama de Gantt, pero, pese a esto, creo que se planteó un escenario realista que se acabó cumpliendo.

En cuanto a los resultados, no podría estar más contento, ya que, con un 92.2% de acierto en el programa final, he superado todas mis expectativas. Aún así, se ha reducido la dimensionalidad del problema, dado que el algoritmo solo detecta el botón de “aceptar” y no los pasos para rechazar el formulario. Esto creo que era inevitable, puesto que rechazar el formulario es demasiado complicado como para resolverlo en un TFG con una duración de un semestre. Pese a esto, esta parte de la funcionalidad queda pendiente para un desarrollo futuro, como se ha explicado en el capítulo anterior.

Finalmente, me gustaría resumir las habilidades y conocimientos que he adquirido gracias a este proyecto, ya que me he sorprendido a mí mismo con mis capacidades de aprendizaje y adaptación.

En primer lugar, creo que he mejorado mi sentido crítico, puesto que a medida que se iba avanzando con el proyecto e iba cometiendo nuevos errores, me planteaba las cosas de forma diferente, aplicando nuevos puntos de vista en cada problema que me encontraba.

En segundo y último lugar, he mejorado con creces mis conocimientos de visión artificial, que hasta el momento eran más bien escasas. También he aprendido muchas técnicas nuevas de Machine Learning que desconocía. Además, al entrenar mi propio modelo de deep learning, he aprendido mucho sobre las redes neuronales y su funcionamiento. Creo que esta disciplina tiene mucho futuro por delante.

Referencias

[1] *Protección de datos | Comisión Europea.* Dirección: https://ec.europa.eu/info/law/law-topic/data-protection_es (visitado 02/02/2021)

[2] *California Consumer Privacy Act of 2018*. Dirección: https://leginfo.legislature.ca.gov/faces/codes_displayText.xhtml?division=3.&part=4.&lawCode=CIV&title=1.81.5 (visitado 02/02/2021)

[3] *Francia multa a Google con 100 millones de euros y a Amazon con 35 millones por sus 'cookies'*. Dirección: https://www.elconfidencial.com/tecnologia/2020-12-10/francia-multa-google-amazon-millones-euros-cookies_2865871/ (visitado 07/02/2021)

[4] *Dark Patterns after the GDPR: Scraping Consent Pop-ups and Demonstrating their Influence* .Dirección: https://arxiv.org/pdf/2001.02479.pdf (visitado 10/02/2021)

[5] *Broadband Communications Systems and Architectures Research Group*. Dirección: https://cba.upc.edu/ (visitado 26/02/2021)

[6] *Online Resource Mapper (ORM).* Dirección: https://github.com/CBA-UPC/ORM (visitado 02/02/2021)

[7] *I don't care about cookies.* Dirección: https://www.i-dont-care-about-cookies.eu (visitado 25/02/2021)

[8] *CookiesOK for Chrome and Opera*. Dirección: https://github.com/SharkWipf/CookiesOK-Chrome (visitado 25/02/2021)

[9] *Ninja Cookies*. Dirección: https://www.ninja-cookies.com (visitado 10/02/2021)

[10] *'Outrageous abuse of privacy': New York orders inquiry into Facebook data use*. Dirección: https://www.theguardian.com/technology/2019/feb/22/new-york-facebook-privacy-data-app-wall-street-journal-report (visitado 27/02/2021)

[11] *Widget detection on screenshots using computer vision and machine learning algorithms.* Dirección: http://koral.ise.pw.edu.pl/~rrom/SPIE/SPIE11176-Wilga2019/source/3-computat%20intellig/028-radzikowski.pdf (visitado 10/03/2021)

[12] Página oficial de Lenovo España. Dirección: https://www.lenovo.com/es/es/laptops/ideapad/d-series/IdeaPad-Duet-3-10IGL5/p/88IPD301447 (Visitado 13/03/2021).

[13] ¿Cuánto cuesta el kilovatio hora de luz(kWh) en España?. Dirección: https://tarifaluzhora.es/info/precio-kwh (Visitado 14/03/2021).

[14] Tabla comparativa de tarifas de internet. Dirección: https://selectra.es/internet-telefono/internet (Visitado 13/03/2021).

[15] PayScale - Salary Comparison, Salary Survey, Search Wages. Dirección: www.payscale.com (Visitado 14/03/2021).

[16] Version Control System: A Review. Dirección: https://www.sciencedirect.com/science/article/pii/S1877050918314819 (Visitado 03/06/2021)

[17] Página principal de Github. Dirección: https://www.github.com (Visitado 21/04/2021)

[18] Página principal de OpenCV. Dirección: https://www.opencv.org/ (visitado 15/03/2021)

[19] A Computational Approach to Edge Detection. Dirección: https://www.ieeexplore.ieee.org/document/4767851 (visitado 20/03/2021)

[20] Página principal de Tesseract OCR. Dirección: https://www.opensource.google/projects/tesseract (visitado 03/04/2021)

[21] Página principal de Darknet. Dirección: https://www.pjreddie.com/darknet/ (visitado 13/05/2021)

[22] Página principal de TensorFlow. Dirección: https://www.tensorflow.org/ (visitado 13/05/2021)

[23] Página principal de Alexa Top Sites. Dirección: https://ats.alexa.com/ (visitado 10/04/2021)

[24] S. Ravi, A.M. Khan. Morphological operations for image processing: understanding and its applications. Proceedings of the national conference on VLSI, signal processing & communications (2013), pp. 17-19

[25] Understanding of a convolutional neural network. Dirección: https://ieeexplore.ieee.org/abstract/document/8308186 (visitado 15/05/2021).

[26] ePrivacy Observatory. Dirección: *http://tars.cba.upc.edu* (visitado 04/03/2021)

Anexos

Anexo 1. Diagrama de Gantt