Escuela Superior
de Ingeniería y Tecnología
Universidad de La Laguna

Trabajo de Fin de Grado

Grado en Ingeniería Informática

Clasificación de documentos con Inteligencia Artificial

Document classification with Artificial Intelligence

Diego Rodríguez Pérez

La Laguna, 11 de Julio de 2023

D. **José Luis Roda García**, con N.I.F. 43.356.123-L profesor Titular de Universidad adscrito al Departamento de Ingeniería Informática y de Sistemas de la Universidad de La Laguna, como tutor

D. **José Carlos González González**, con N.I.F. 45.451.553-B Jefe del Servicio TIC de la Universidad de La Laguna, como cotutor

C E R T I F I C A (N)

Que la presente memoria titulada:

"Clasificación de documentos con Inteligencia Artificial"

ha sido realizada bajo su dirección por D. **Diego Rodríguez Pérez**,

con N.I.F. 54.064.733-J.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 11 de julio de 2023

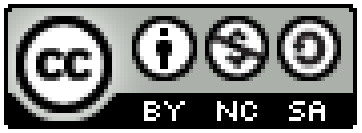
Agradecimientos

A mis padres, que me lo han dado todo.

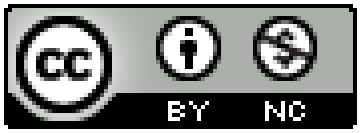
Licencia

* Si NO quiere permitir que se compartan las adaptaciones de tu obra y NO quieres permitir usos comerciales de tu obra indica:

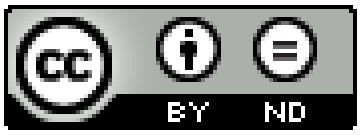
* Si quiere permitir que se compartan las adaptaciones de tu obra mientras se comparta de la misma manera y NO quieres permitir usos comerciales de tu obra indica:



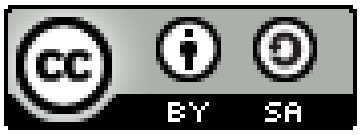
* Si quiere permitir que se compartan las adaptaciones de tu obra y NO quieres permitir usos comerciales de tu obra indica:



*Si NO quiere permitir que se compartan las adaptaciones de tu obra y quieres permitir usos comerciales de tu obra indica:



* Si quiere permitir que se compartan las adaptaciones de tu obra mientras se comparta de la misma manera y quieres permitir usos comerciales de tu obra (licencia de Cultura Libre) indica:



* Si quiere permitir que se compartan las adaptaciones de tu obra y quieres permitir usos comerciales de tu obra (licencia de Cultura Libre) indica:

Resumen

La Inteligencia Artificial es uno de los campos que más cambios y más avances ha tenido en los últimos años, pues ha pasado de ser un concepto muy confuso para todos a convertirse en uno de los términos más escuchados hoy en día. Además, hay herramientas desarrolladas a partir de IA que son esenciales en el día a día de muchas personas, algo que hace muy pocos años era impensable.

Este es un campo muy amplio, que abarca una infinidad de temas, dentro de los cuales están el Procesamiento del Lenguaje Natural (NLP) y el tratamiento de imágenes, es decir, la Visión por Computador. Son estos campos los que van a copar todo este Trabajo de Fin de Grado.

El objetivo de este trabajo, es principalmente la clasificación de documentos a través de la Inteligencia Artificial, utilizando diferentes estrategias, tanto de Machine Learning como de Deep Learning.

Se mostrarán todos los resultados que se han obtenido, de manera que se va a poder hacer una comparación completa de las estrategias, de manera que se van a poder ver los puntos fuertes y las debilidades de cada una de ellas.

Palabras clave: Inteligencia Artificial, Machine Learning, Deep Learning, dataset, modelo, clasificación, accuracy, precision, recall, f1, matriz de confusión, OCR ...

Abstract

Artificial Intelligence is one of the fields that has undergone the most changes and advances in recent years, as it has gone from being a very confusing concept for everyone to becoming one of the most commonly heard terms nowadays. Moreover, there are tools developed from AI that are essential in many people's daily lives, something that was unthinkable just a few years ago.

This is a very large field, so it covers a myriad of topics, among which are Natural Language Processing (NLP) and image processing, i.e. Computer Vision. It is these fields that will be the focus of this Final Degree Project.

The aim of this work is mainly the classification of documents through Artificial Intelligence, using different strategies, both Machine Learning and Deep Learning.

All the results that have been obtained will be shown, so that a complete comparison of the strategies can be made, so that the strengths and weaknesses of each of them can be seen.

Keywords: *Artificial Intelligence*, Machine Learning, Deep Learning, dataset, model, classification, accuracy, precision, recall, f1, confusion matrix, OCR, ...

Índice general

Índice de figuras

Índice de tablas

Capítulo 1 - Introducción

Cuando se habla de documentos, no se suele tener en cuenta la importancia de los mismos, ya que se han vuelto tan cotidianos en el día a día que se tiende a infravalorar y menospreciar la importancia de los mismos. Generalmente, un documento, además de ser una herramienta muy útil, es un sitio donde se puede almacenar el conocimiento humano, haciendo así que la información plasmada en dicho documento perdure mientras que este documento siga existiendo.

Como absolutamente todo en la sociedad, el uso de la documentación ha sufrido un gran avance tecnológico, pues cada vez más se puede almacenar más cantidad de información en menor espacio. Es por esto, que el conocimiento hoy en día es prácticamente infinito, pues gracias a tecnologías como Internet, se puede acceder a muchísima información almacenada en los documentos, algo que tiene sus ventajas y desventajas, pues al haber tanta información disponible, se tarda mucho tiempo en procesarla y en encontrar el contenido que se está buscando.

El desarrollo de este proyecto se busca, a través de diferentes técnicas y estrategias como el NLP, el OCR y el reconocimiento de imágenes, el poder procesar, analizar y clasificar los documentos de una manera automática, optimizando una tarea que podría considerarse como sistemática para una persona.

El contexto de este proyecto es que en la actualidad, la ULL tiene un sistema de clasificación documental que no es lo efectivo que le gustaría, por lo que el abordar diferentes soluciones va a permitir saber cuál es el mejor enfoque posible a la hora de enfrentarse a problemas similares, además de desarrollar algoritmos que mejoren el sistema actual.

1.1 Antecedentes y estado actual del tema

En los últimos años, se está haciendo un esfuerzo a nivel mundial, por ir modernizando los puestos de trabajo. Es aquí donde aparece la automatización de los trabajos sistemáticos, como puede ser el empaquetar un producto, responder siempre las mismas preguntas en un servicio de atención al cliente o la propia clasificación de documentos. La constante repetición de estas tareas puede llegar a afectar a la salud tanto física como psicológica de las personas, por lo que la transformación de la vida laboral es clave e indispensable para la evolución de la sociedad.

La clasificación de documentos es un proceso de asignación de categorías o clases a los documentos para facilitar su procesado, gestión, búsqueda, filtrado o análisis. En este caso, un documento es un elemento de información que tiene un contenido relacionado con alguna categoría específica. Las fotos de

productos, los comentarios, las facturas, los escaneos de documentos y los correos electrónicos pueden considerarse documentos, por lo que su relevancia es enorme, ya que son acciones que se realizan en múltiples ocasiones día tras día. La clasificación de documentos puede formar parte de una iniciativa más amplia denominada procesamiento inteligente de documentos (PID). En general, las tareas de clasificación de documentos se suelen dividir en clasificaciones textuales y visuales.

Por un lado, la clasificación textual consiste en definir el tipo, el género o el tema del texto a partir de su contenido. Dependiendo de la tarea, se pueden utilizar técnicas complejas como el NLP (Procesamiento del Lenguaje Natural) para analizar palabras y frases en su contexto y comprender su semántica (significado). Para poner un ejemplo actual, el detector de spam de Gmail de Google, por ejemplo, emplea NLP para encontrar mensajes basura y colocarlos en la carpeta correspondiente. En 2015, Google también implementó una red neuronal que mejoró las capacidades del NLP de su filtro de spam. Básicamente, las redes neuronales se aplican en caso de necesitar una tecnología más sofisticada para detectar el spam menos obvio.

En cuanto a la clasificación visual, se centra en la estructura visual de los documentos, empleando tecnologías de visión por ordenador. El reconocimiento de imágenes puede utilizarse en la clasificación de documentos para detectar objetos, su ubicación o su comportamiento en el contenido visual. Un caso de uso de esta tecnología, podría ser a la hora de tratar con documentos sanitarios, pues se debería ser capaz de diferenciar gracias a la visión por computador las diferentes partes de los documentos, como los datos del paciente, por un lado, el diagnóstico del doctor, por otro, y por último los detalles de la cita.

Capítulo 2 - Estado del arte

2.1 ¿Qué es la Inteligencia Artificial?

La Inteligencia Artificial *[1]* es una rama de la informática que se enfoca en la creación de máquinas con la capacidad de razonar e imitar el comportamiento de las personas, es decir, intenta transferir a las máquinas conocimiento humano para que estas puedan comportarse como tales. Esta tecnología, con el paso de los años, se ha convertido en una de las áreas de la computación de mayor crecimiento, pues cada vez más se aplica en situaciones cotidianas para mejorar distintas situaciones como puede ser la experiencia del usuario a la hora de comprar, ya que gracias a la información que tiene sobre el usuario el dispositivo, es capaz de proporcionar conocimientos y recomendaciones de forma más precisa. Esta tecnología está cambiando la forma en la que trabajamos, vivimos y nos relacionamos con el mundo.

2.2 Aplicaciones de la Inteligencia Artificial

La Inteligencia Artificial (IA) es un campo que tiene diversas aplicaciones y, actualmente, está transformando la forma en que las empresas y las organizaciones están abordando los desafíos a los que se enfrentan. A medida que la tecnología de IA sigue evolucionando, es muy probable que se vayan a ver más aplicaciones innovadoras en los próximos años. Algunas de las más comunes son las siguientes:

1. Automatización de procesos: La Inteligencia Artificial se ha convertido en una herramienta extremadamente útil para optimizar procesos. En esta categoría se pueden incluir actividades como la contabilidad de una empresa, su producción, realizar un análisis de los datos de la misma, la facturación y la gestión de inventario. Esto ahorra tiempo y recursos para las empresas, además de aumentar la eficiencia y la rentabilidad, pues los recursos que empleaba la compañía en que alguien hiciese una tarea de manera repetitiva, lo ha dejado en manos de la IA, mientras que puede destinar a dicho trabajador a una tarea más compleja.

2. Reconocimiento de voz y lenguaje: La IA ha convertido este aspecto casi que en una herramienta obligatoria para los dispositivos digitales, especialmente los móviles. Gracias a este avance, los usuarios pueden interactuar con los diferentes dispositivos, mediante la voz, algo que para unas personas puede resultar una simple funcionalidad más, mientras que para otras con discapacidades visuales es algo que facilita su día a día de una manera exponencial.

3. Reconocimiento de imágenes: El reconocimiento de imágenes *[2]* es otra de las grandes aplicaciones de la IA, ya que por ejemplo podría llegar a mejorar la seguridad de un edificio, pues si se coloca un detector facial a la entrada de este, se podía abrir la puerta solo a las personas registradas en el sistema. Otro gran ejemplo de uso podría ser la temprana detección de incendios, pues si se enseña a la máquina a detectar conatos de incendio y esto se instala en un satélite, se podrían detectar de una manera muy temprana y evitar así su propagación, lo que salvaría muchas vidas, tanto humanas como de animales.

2.3 Aprendizaje Automático

El aprendizaje automático es un campo de la Inteligencia Artificial que se centra en algoritmos y modelos que son capaces de adquirir conocimiento a través de los datos. Estos algoritmos son capaces de detectar patrones en esos datos y utilizarlos para elaborar predicciones. Este campo está transformando la manera en la que las empresas actúan, ya que ofrecen a las compañías posibilidades como automatizar procesos, incrementar la eficacia y mejorar la experiencia del cliente.

Una de las maneras en que una empresa puede hacer uso de esta tecnología es la toma de decisiones en tiempo real. Por ejemplo, estos modelos son capaces de predecir el comportamiento del cliente y así analizarlo y mejorar la experiencia del mismo con una serie de recomendaciones personalizadas.

En cuanto a la automatización de procesos, uno de los casos más claros es en el sistema sanitario. Como se ha mencionado anteriormente, estos sistemas se encargan de detectar patrones en los datos, de manera que en el caso de que si detectan algo anómalo en los datos de un paciente puede avisar al doctor, por lo que para un primer análisis resultan muy útiles.

Existen diversos tipos de aprendizaje automático, pero tres son los que predominan actualmente. Estos son el supervisado, no supervisado y por refuerzo.

- Aprendizaje supervisado: consiste en utilizar datos que han sido etiquetados previamente para entrenar un modelo. Este modelo va a ser utilizado para predecir un resultado o realizar algún tipo de clasificación. Ejemplos de este tipo de aprendizaje son la regresión, la clasificación y los sistemas de recomendación.
- Aprendizaje no supervisado: es lo contrario del supervisado, por lo que los datos con los que se trata en estos casos no están previamente etiquetados. Los modelos construidos tienen el fin de detectar patrones en los datos, ya sean agrupaciones o anomalías.
- Por refuerzo: utiliza un sistema de recompensas y castigos. En este caso el objetivo del modelo es poder realizar una serie de acciones en un entorno determinado para así poder maximizar su recompensa.

2.3.1 Algoritmos utilizados de Machine Learning

A continuación se va a profundizar en todos los algoritmos que han sido utilizados en algunas de las fases de este trabajo.

1. Random Forest: *[3]* es un tipo de algoritmo de aprendizaje automático que es utilizado habitualmente para problemas de clasificación, principalmente para aquellos en los que se cuenta con un gran conjunto de datos. Consiste en generar diversos árboles de decisión, los cuales individualmente generan una salida, la cual se combina con el resto de las salidas generadas por los demás árboles y se crea una predicción final basada en la combinación de todas estas predicciones individuales.

Este algoritmo tiene diversas ventajas. Una de las principales es la capacidad que tiene para reajustar el sobreajuste y hacer así mejores predicciones, ya que los diferentes árboles pueden tener diferentes debilidades y fortalezas, por lo que al combinarlos los puntos fuertes sobresalen. Otro de los grandes beneficios es que es un algoritmo que se puede implementar tanto en problemas de regresión como de clasificación.

Por otro lado, una de las grandes desventajas es la alta capacidad de computación que requiere, pues al generar tantos árboles individuales, lleva mucho tiempo de entrenamiento, además de que es un algoritmo que soporta grandes conjuntos de datos.

A modo de conclusión, el Random Forest, es un algoritmo que genera diversos árboles de decisión, los cuales generan una salida individual que se combina con los demás árboles generados para realizar una predicción final.

2. Support Vector Machine: *[4]* es un algoritmo que se utiliza tanto en tareas de regresión como en tareas de clasificación, el cual es muy eficaz en problemas complejos. Cada muestra de datos se representa como un vector, y, a partir de todos los vectores, se crea un hiperplano en un espacio multidimensional que tiene el objetivo de mejorar la separación de dos clases.

El principal objetivo es maximizar el margen que hay entre dos clases, pues lo que consigue este algoritmo es encontrar el hiperplano óptimo que mejor separa las dos clases, es decir, maximizar el margen que hay entre ellas.

La manera en la que funciona este algoritmo es que mapea los puntos del conjunto de datos en un espacio de mayor dimensión para que las clases puedan ser separadas mediante un hiperplano. Una vez se ha realizado este primer paso, calcula el hiperplano óptimo que separa las dos clases. Por último, una vez se conoce este plano, se utiliza para hacer predicciones con las observaciones futuras.

En resumen, el algoritmo SVM, se utiliza para tareas de regresión y de clasificación que optimizan la distancia existente entre la separación de elementos de dos clases diferentes, gracias a calcular el hiperplano que mejor las separa.

2.4 Aprendizaje Profundo

El aprendizaje profundo *[5]* es una de las ramas del aprendizaje automático, la cual utiliza algoritmos inspirados en la estructura y el funcionamiento del cerebro humano para poder aprender de grandes cantidades de datos, lo que va a permitir a las máquinas ser capaces de identificar patrones y realizar predicciones

Los algoritmos de aprendizaje profundo son utilizados para resolver tareas que pueden llegar a resultar demasiado complejas para los modelos de programación más tradicionales. Estos modelos son aplicados en diversas áreas, como la visión por ordenador, el procesamiento del lenguaje natural, la robótica y los juegos.

El éxito de los algoritmos de aprendizaje profundo se basa en la capacidad que tienen para aprender de grandes cantidades de datos. Este tipo de modelos se componen de una o varias capas de neuronas conectadas entre sí que han sido entrenadas. Cada capa de neuronas extrae un determinado tipo de información de los datos. Esta información pasa a la siguiente capa para su posterior procesamiento y análisis. Este proceso de extracción y transformación de datos se denomina "aprendizaje profundo" debido a que requiere múltiples capas de neuronas para que el modelo sea capaz de interpretar con precisión los datos utilizados.

El potencial de estos algoritmos es inmenso, y las posibilidades de sus aplicaciones, infinitas, pues a medida que se va aumentando la cantidad de datos disponibles, también lo hace simultáneamente el potencial de los algoritmos de aprendizaje automático para proporcionar conocimientos y permitir a las máquinas hacer cosas que antes no eran posibles.

2.4.1 Redes Neuronales

Las redes neuronales son un tipo IA que basa su funcionamiento en la estructura del cerebro humano. Como su propio nombre indica, estas redes están compuestas por neuronas conectadas entre sí y son utilizadas para detectar relaciones no lineales complejas entre entradas y salidas.

Por lo general, una red neuronal se compone de varias capas, las cuales a su vez están formadas por un conjunto de neuronas. Estas capas están conectadas entre ellas, de manera que la entrada de una capa va a ser la salida de la anterior, al igual que la salida de una capa va a ser la entrada de la siguiente capa. Cada una de las neuronas de la capa procesa los datos recibidos con un conjunto de pesos y sesgos. Una vez los datos han sido procesados, pasan a la siguiente capa para repetir el proceso. Esto se va a realizar desde la capa de entrada de la red hasta la de salida.

Al igual que la gran mayoría de los algoritmos de Inteligencia Artificial, las redes neuronales contienen una gran cantidad de parámetros y variaciones como pueden ser los pesos y sesgos. Los pesos son valores numéricos que representan la fuerza con la que las neuronas se conectan entre sí, es decir, la salida de una neurona con mayor peso que otra va a ser más significativa a la hora de procesar dicha información. Estos pesos se van ajustando durante el entrenamiento de la red, de manera que optimizan los resultados de la red. El sesgo de una red neuronal es un concepto muy importante, pues generalmente una red muy sesgada no suele dar buenos resultados. Para ponerlo en contexto, una persona que tenga un sesgo, por ejemplo, a una marca de coches, va a llevar a esta persona a no considerar comprarse un coche de otra marca, lo que puede hacer que la decisión final que tome no sea la más adecuada. Enfocando esto a la inteligencia artificial, un modelo sesgado va a tender a clasificar las instancias siempre de la misma manera. Es por esto que regular este sesgo va a permitir que el modelo o la red neuronal alcance una mayor precisión.

Las redes neuronales pueden ser usadas para resolver una innumerable cantidad de problemas como pueden ser predecir acciones futuras, identificar objetos en imágenes en tiempo real y también para clasificar texto. Además, tienen la capacidad de mejorar sistemas de inteligencia artificial ya existentes, ya que al tener un funcionamiento similar al del cerebro humano son capaces de tomar decisiones más humanas. Han ayudado a la revolución de la inteligencia artificial, pues se ha demostrado que son muy prometedoras para resolver problemas complejos.

2.4.2 Tipos de Redes Neuronales

Las redes neuronales artificiales son redes extremadamente útiles, puesto que son utilizadas para resolver problemas en temas relacionados con la informática y la ingeniería. Están formadas por un conjunto de neuronas conectadas entre sí a través de enlaces. Una de las grandes ventajas es la polivalencia de las mismas, puede pueden ser aplicadas en tareas de clasificación, regresión, predicción y reconocimiento de patrones. El funcionamiento es muy similar al de un cerebro humano, y es aquí donde reside uno de sus grandes poderes, y, es que tienen una alta capacidad para identificar patrones y sacar conclusiones y encontrar soluciones a partir de los datos con los que han sido entrenadas. Los tipos más conocidos son los siguientes:

1. Redes neuronales recurrentes: son redes neuronales que tienen la peculiaridad que pueden recordar datos de pasos anteriores, información que puede ser utilizada para la elaboración de las predicciones. Esto las hace ideales para tareas secuenciales como puede ser el reconocimiento del habla o el procesamiento del lenguaje natural. Esto se debe a que cuando los humanos hablan o escriben, el orden en el que se encuentran las palabras es determinante para el significado final, por lo que una red que sea capaz de recordar el orden de las palabras es ideal. En los humanos, sucede que algunos tiene una memoria más desarrollada que otros, por lo que son capaces de recordar durante más tiempo que otros, como en este tipo de redes, donde

existen diversos tipos que tienen diferentes capacidades de almacenamiento.

2. Redes neuronales convolucionales: son aplicadas principalmente en tareas de visión por computador y han sido las principales responsables de los grandes avances en este campo. La estructura de estas redes se basa en que cada una de las capas es responsable de detectar un patrón característico en la imagen. Por ejemplo, en una imagen, la primera capa de la red detectaría los bordes que existan en la imagen. Esta información aprendida pasaría a la siguiente capa, la cual a partir de los bordes que ha detectado la primera, podría detectar por ejemplo formas, y así sucesivamente se irían aprendiendo cada vez patrones más complejos, el cual es el gran potencial de este tipo de redes.

2.4.3 Algoritmos utilizados de Deep Learning

1. Perceptrón Multicapa: *[6]* el MLP(Multilayer Perceptron) es un tipo de red neuronal artificial que tiene la capacidad de comprender relaciones no lineales entre entradas y salidas. Está dentro de la categoría de aprendizaje supervisado que utiliza la retropropagación(back-propagation) para ajustar los pesos de la red, para así minimizar el error. Este tipo de redes neuronales pueden ser utilizados para tareas de clasificación, de regresión y de agrupación.

Estas redes neuronales constan de una capa de entrada, una o varias capas ocultas y por último una capa de salida. Cada una de estas capas contiene neuronas, las cuales reciben entradas de las neuronas de las capas anteriores y realiza una suma ponderada de todas estas entradas para determinar la salida de la neurona. Estas salidas de las capas de entrada pasan hacia las capas ocultas, y así sucesivamente hasta llegar a la última capa de la red.

Como se ha comentado, se utiliza la técnica de back-propagation, la cual consiste en calcular el error que hay entre la salida prevista y la salida deseada. Entonces, según estos cálculos, se reajustan los pesos de las conexiones para poder minimizar el error.

2. Red Recurrente Simple: *[7]* como su propio nombre indica, es una de las formas más sencillas de las redes recurrentes, pues utilizan una única capa recurrente para procesar la información. La arquitectura de estas redes es la siguiente: una capa de entrada, una capa recurrente y la capa de salida, de manera que la capa de entrada recibe los datos, la recurrente es la encargada de procesarlos y la de salida produce la salida de la red.

Como es lógico, la parte más importante de este tipo de redes, es la capa recurrente, la cual está formada por un conjunto de “celdas de memoria”, las cuales almacenan cada una un valor y están conectadas entre sí. Cada vez que se procesa una nueva entrada, los valores de las diferentes celdas se van actualizando, lo que permite a este tipo de redes captar el contexto de la secuencia de entrada y ayudarse de él para producir la salida de la red.

En general, es un tipo de red sencillo de entrenar, se utilizan principalmente en el campo del Procesamiento del Lenguaje Natural y para predecir series

temporales. Son una gran herramienta para el modelado de datos secuenciales y proporcionan una forma robusta de capturar el contexto de las secuencias de entrada.

3. Memoria a Largo Plazo: *[8]* (LSTM-Long Short Term Memory) son redes que están diseñadas para ser capaces de aprender información a largo plazo en secuencias de datos de entrada. La gran diferencia con las Redes Recurrentes Simples, es que mientras unas son solo capaces de aprender las secuencias a corto plazo, este tipo de redes pueden recordar y utilizar la información en grandes secuencias de datos.

Al igual que en las SRNN, se utilizan las células de memoria para ir almacenando valores que se van actualizando, pero además, se añade el uso de las denominadas células de compuerta. La estructura se basa en tres tipos de puerta, una de entrada, una de salida y una de olvido. La de entrada se encarga de controlar el flujo de datos que accede a la célula, mientras que la de salida controla el flujo que sale. La puerta de olvido es la encargada de manejar la cantidad de información que va a conservar la célula y de la información que se va a olvidar.

Como es tónica en las redes recurrentes, se aplican en una de gran cantidad de tareas, siendo algunas de las más conocidas la creación de asistentes virtuales como pueden ser Siri y Alexa. Otro de los grandes usos ha sido el diagnóstico de enfermedades y la predicción de respuestas a fármacos.

4. Unidad Neuronal Recurrente: [9] (GRU-Gated Recurrent Unit) este tipo de redes tienen como objetivo simular de manera muy precisa el comportamiento del cerebro humano. Son un tipo de red que se podría considerar que están a medio camino de las RNN Simples y las LSTM, pues son capaces de almacenar conocimiento más a largo plazo que las SRNN, pero no tanto como las LSTM.

Como es tónica en las redes recurrentes, tienen una capa de oculta que uno de sus componentes son las células de compuerta, pero con una diferencia de las LSTM y, es que, en este tipo de redes, están compuestas por dos puertas en vez de tres. Una de las puertas es denominada de actualización y se encarga de controlar la cantidad de información que va a almacenar y el otro tipo son las de reseteo, las cuales son responsables de controlar la información que va a ser descartada.

Se utilizan en diversas aplicaciones, desde el tratamiento de imágenes y el procesamiento del lenguaje natural hasta la robótica y los vehículos autónomos, además de ayudar a crear sistemas que puedan interactuar con los humanos de una forma más natural. En definitiva, son un importante paso adelante en el campo de la Inteligencia Artificial, y, es que están allanando el camino para que las máquinas puedan ser capaces de pensar como los humanos, además de mejorar la interacción con las mismas.

5. Convolucional 2D: *[10]* es el tipo más popular de las redes convolucionales. Este tipo de red se compone de un conjunto de filtros aprendibles, los cuales se aplican a una imagen de entrada o a un mapa de características. Estos filtros pueden ser considerados como un conjunto de núcleos que se aplican a la imagen. Entonces, se produce un mapa de características que va a contener las características extraídas de la imagen. Una vez se ha hecho esto, la capa convolucional es capaz de aprender entonces a combinar estas características para producir una salida.

En términos generales, la capa Conv2D suele ser la primera capa de una red convolucional y suele ser utilizada para extraer características de una imagen de entrada como pueden ser bordes, esquinas o texturas. Estas características se transmiten a las siguientes capas de la red, que aprenden características más complejas a partir de las características extraídas. Este proceso de extracción y aprendizaje de características continúa hasta que la CNN es capaz de clasificar imágenes con precisión. En caso de tener una imagen de un animal, esta primera red sería la encargada de detectar los ojos, las orejas y la nariz, etc. Cuando esta información se pasa a la siguiente capa, esta capa la usará para detectar que si existen estas tres cosas juntas, podrá detectar que existe una cabeza. Algo similar sucederá con el tronco del animal, que a su vez se combinarán tanto el saber que existe el tronco y la cabeza para saber que existe dicho animal.

Este tipo de capas son muy eficientes tanto en términos de complejidad computacional como de uso de memoria. Por eso son adecuadas para aplicaciones en tiempo real como la conducción autónoma, el reconocimiento facial, la clasificación de imágenes y la detección de objetos.

6. Convolucional 2D Separable: *[11]* es una red neuronal convolucional que se utiliza para reducir el número de parámetros, ya que su principal concepto es descomponer un núcleo de convolución en dos núcleos separados, un núcleo horizontal y un núcleo vertical. Al descomponer un núcleo de convolución, se puede obtener el mismo resultado aplicando los dos núcleos separados sucesivamente, en lugar de aplicar el núcleo original más grande de una sola vez para así poder mejorar la interacción con las mismas.

La ventaja principal de utilizar es reducir el número de parámetros, por lo tanto, la eficiencia. Esto resulta especialmente beneficioso a la hora de utilizar redes más grandes o cuando varias capas de convolución son empleadas en la red. Como consecuencia de esta reducción, se pueden ir añadiendo y utilizar más capas de convoluciones, de manera que se mejorarían los resultados finales

Capítulo 3 - Planteamiento del problema

Como se ha visto en la introducción, la clasificación de documentos *[12]* es un problema cotidiano al que nos enfrentamos en el día a día. Es por ello que, automatizar esta tarea es algo fundamental, pues el número de documentos cada vez es mayor, por lo que en poco tiempo hacer esta tarea de forma manual sea impensable debido a las posibles dimensiones del problema. Es aquí donde entran diferentes estrategias para poder enfrentarnos al problema.

En este caso, se plantean diversos problemas, por lo tanto, diversas soluciones. Es por ello que se van a dividir en diferentes fases, dentro de las cuales se van a utilizar diferentes estrategias para resolver cada una de ellas. Estas fases estarán ordenadas de manera que la primera de ellas tiene un objetivo más general y en la última de ellas el objetivo final está bastante más acotado.

La Fase 1 va a consistir principalmente en saber si un documento se corresponde con un justificante de pago o no. El motivo de realizar esta fase es que hay muchas ocasiones en las que las personas se registran para algún evento, como puede ser la convocatoria de unas oposiciones y para completar dicho registro es necesario subir el justificante de pago. Bien, pues puede suceder que la persona en cuestión no lo tenga disponible en ese momento y decida subir un documento que nada tiene que ver o simplemente una hoja en blanco. Es por ello que es conveniente detectar si el documento que ha sido subido se trata de un justificante de pago para así poder saber qué personas han pagado y quiénes faltan por hacerlo.

La Fase 2 es similar a la primera, pero tiene una meta mucho más determinada. Y es que al igual que antes se quería saber si se había subido un recibo bancario o no, el enfoque se va a hacer sobre la cantidad pagada. Es decir, una vez ya se sabe que se está tratando con un recibo, hay que detectar si se ha abonado la cantidad correcta o no, ya que se puede dar el caso de que alguien haya pagado una cantidad menor a la debida. Gracias a desarrollar esta fase se va a poder complementar la primera fase del proceso, pues si una persona subiese un justificante de pago, la fase uno diría que esa persona ha pagado, pero no sería capaz de decir si el importe era el correcto, lo cual si se va a saber gracias a esta fase.

La Fase 3 va a ser muy similar a la segunda, con la diferencia de en vez de centrarse en el importe abonado, lo va a hacer en el concepto de la transferencia. Puede suceder que una misma persona se haya registrado en, por ejemplo, dos eventos distintos, los cuales son organizados y gestionados por la misma empresa, entonces la empresa va a tener que gestionar ambos recibos y tenga que diferenciar automáticamente que recibo es de cada curso. Para ello, una de las maneras de hacerlo es con el concepto de la transferencia bancaria., algo que a simple vista puede resultar muy fácil de hacer en dos documentos,

pero en el caso de que fuesen miles y miles de los mismos sería una tarea muy tediosa, además de que la cantidad de horas empleadas sería enorme.

Inicialmente, como no se tenía una idea clara de la manera correcta en la que se debería resolver el problema, se ha hecho una profunda investigación de problemas similares y la manera de resolverlos y se han ido apuntando diferentes soluciones. Una vez con varias propuestas, se han discutido las ventajas y desventajas y como se podrían adaptar a este caso y se ha decidido que se van a seguir dos estrategias para ir resolviendo cada una de las fases. Una de ellas será mediante OCR (Optical Character Recognition) *[13]*, la cual consiste en extraer el texto de una imagen y después mediante técnicas de Procesamiento de Lenguaje Natural, construir modelos tanto de Machine Learning como de Deep Learning para clasificar los diferentes documentos. La segunda técnica empleada va a ser la de tratar los documentos como imágenes *[14]*, es decir, al igual que hay modelos construidos que son capaces de detectar objetos en una imagen, se va a intentar resolver cada una de las fases de una manera en que se pueda clasificar en base a la imagen, no al propio texto de la imagen.

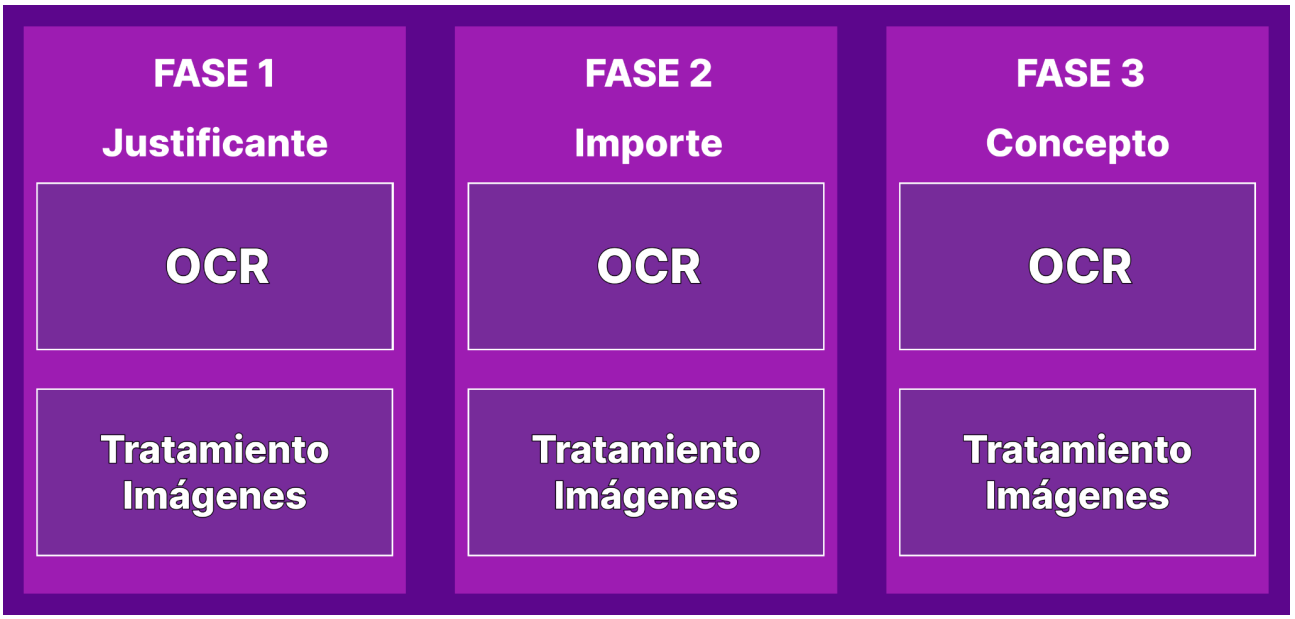


Figura 1 - Estructura inicial del desarrollo

La imagen superior representa la idea inicial para la resolución del problema, pero ni mucho menos es definitiva, ya que según vaya avanzando el desarrollo del proyecto y se vayan obteniendo resultados de las diferentes aproximaciones realizadas, se podrá ir cambiando de idea en caso de que las planteadas al inicio no estén dando los frutos esperados. Por otro lado, también cabe la posibilidad de que si se refleja que se ha acertado en la idea se pueda profundizar en ella con la intención de ver hasta donde se puede llegar, teniendo en cuenta tanto el acierto como lo óptimo que sea lo que se ha llevado a cabo

Uno de los elementos más influyentes en los problemas de Machine Learning es el conjunto de datos utilizado, por lo que su calidad va a resultar determinante. En este caso, los datos a utilizar se encuentran en una estructura de directorios en la que cada persona tiene su propia carpeta.

Figura 2 - Formato en crudo de los datos

En total, hay 200 carpetas con este mismo formato. Dentro de ellas se almacena la información y los documentos relacionados con la persona en cuestión. En general, el contenido existente dentro de cada una de las carpetas suele ser similar, pues hay diferentes tipos de archivos. Se muestra un ejemplo del contenido en la siguiente imagen.

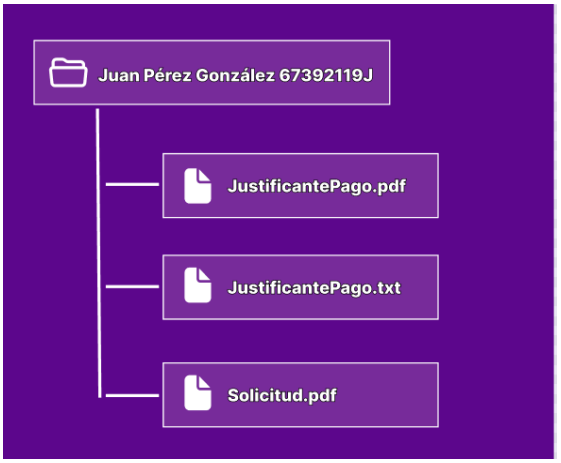


Figura 3 - Estructura de ficheros por persona

Como se observa en la imagen. Se tiene un archivo PDF que supuestamente corresponde con un justificante de pago, pero hasta que no se abre el archivo, no se puede saber a ciencia cierta. Para la construcción del conjunto de datos se han realizado varios pasos, algunos de ellos de manera manual, mientras que para otros se han hecho programas en *Python* que realizarán una tarea determinada.

El primer paso ha sido realizado con un ejecutable de *Python*, que itera sobre el directorio donde se encontraban todas las carpetas de las personas. Después entraba en cada una de las carpetas y buscaba el fichero que se correspondía con el Justificante de Pago en formato *PDF*. Una vez se obtenía este fichero mediante la librería *pdf2image* se convertía este archivo en una imagen para obtener mejores resultados, es decir, en *JPG*. Una vez convertido el archivo se almacenaban todas estas nuevas imágenes en un directorio para su posterior clasificación, quedando de la siguiente manera.

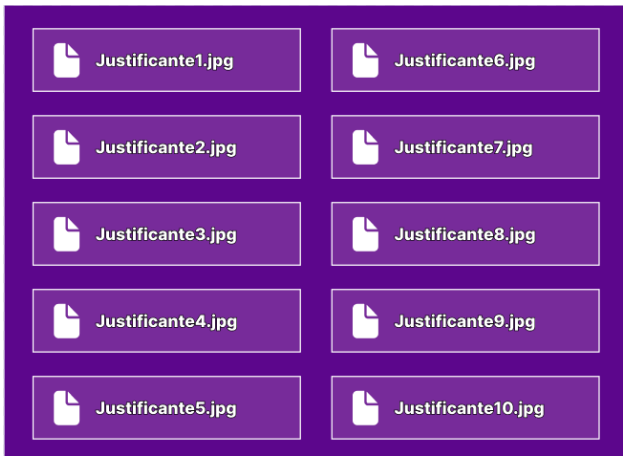


Figura 4 - Carpeta con todos los justificantes convertidos en imágenes

Una vez se tienen todos los archivos en el formato adecuado dentro de una misma carpeta, es momento de la "creación" del Dataset de manera manual. Para ello se ha ido revisando de manera individual cada uno de los doscientos documentos, quedando el conjunto de datos listo para usar.

Figura 5 - Estructura de los directorios con los justificantes clasificados

Capítulo 4 - Fase 1 Clasificación primaria

Como se ha explicado en el capítulo anterior, esta primera fase va a consistir en detectar si un documento se corresponde con un justificante de pago o no. Para ello se van a utilizar estrategias de Machine Learning, y dentro de este campo se usará también Deep Learning, según se crea conveniente. Para todos los modelos, las métricas usadas han sido *Accuracy, Precision, Recall* y *F1*.

4.1 OCR (Optical Character Recognition)

Esta estrategia denominada *OCR*, es una tecnología que permite la digitalización de documentos y el reconocimiento de textos en ellos, ya sean impresos o escritos a mano. El OCR es una herramienta muy valiosa en diferentes campos, como la industria editorial, la administración de documentos, la banca, la medicina y la educación, entre otros. Una de las grandes ventajas de esta técnica es que permite la accesibilidad de los documentos para personas con discapacidades visuales, ya que puede convertirlos tanto en un audio como en el formato Braille.

Para poder llevar esta tarea a cabo, se ha tenido que construir primero el *dataset* sobre el que se va a trabajar. Para ello se ha aplicado esta técnica en cada uno de los documentos, tanto los justificantes como los que no lo eran. Una vez obtenido el texto, se ha guardado en un fichero *csv* y se ha etiquetado según a la categoría a la que pertenecía, un 1 si era un justificante y un 0 si no lo era. Al final ha quedado el siguiente conjunto de datos:

	texto	justificantePago
0	orden transferencia cajasiete oficina dtelefon...	1
1	transferencia nacional vy operacion realizada ...	1
2	™n imagin vank transferencias traspasos operac...	1
3	wk ¢ aixabank ff operacion realizada correctam...	1
4	NaN	1
...	...	...
194	informe inscripcion servicio canario empleo in...	0
195	— ij ministerio ht rc economia social direccio...	0
196	como renuevo demanda empleo ecuando fecha prox...	0
197	informe inscripcionrechazo ofertas acciones or...	0
198	gobierno consejeria empleo ba asuntos sociales...	0

199 rows × 2 columns

Figura 6 - DataFrame inicial Fase 1 OCR

En los problemas de *Machine Learning*, la limpieza de los datos es algo fundamental, ya que influyen directamente en la calidad del modelo final. Para ello existen diferentes técnicas. En este caso, al ser un problema de

Procesamiento de Lenguaje Natural, se ha hecho uso de la librería Preln, la cual deja los datos de una manera muy limpia, pues elimina los signos de puntuación, los acentos y muchos más aspectos. Una vez realizado este proceso es momento de ver si existen datos vacíos o nulos, pues no es conveniente tenerlos porque no aportan información relevante. Concretamente, se han encontrado 9 datos vacíos y se han eliminado, quedando el conjunto de datos final así:

	texto	justificantePago
0	orden transferencia cajasiete oficina dtelefon...	1
1	transferencia nacional vy operacion realizada ...	1
2	™n imagin vank transferencias traspasos operac...	1
3	wk ¢ aixabank ff operacion realizada correctam...	1
5	™n imagin vank transferencias traspasos operac...	1
...	...	...
194	informe inscripcion servicio canario empleo in...	0
195	— ij ministerio ht rc economia social direccio...	0
196	como renuevo demanda empleo ecuando fecha prox...	0
197	informe inscripcionrechazo ofertas acciones or...	0
198	gobierno consejeria empleo ba asuntos sociales...	0

190 rows × 2 columns

Figura 7 - DataFrame final Fase 1 OCR

Una vez los datos están listos para ser utilizados se han seguido los pasos más comunes para la resolución de problemas de clasificación de *Machine Learning*. El primero de ellos es separar los datos en entrenamiento y *testeo*:

```
X = df.texto
Y = df.justificantePago
```

Figura 8 - Ejemplo separación en datos entrenamiento y de testeo

Al tratarse de un problema de NLP es necesario la vectorización del texto y para ello se ha hecho uso de TF-IDF *[15]*, la cual es una técnica que tiene en cuenta la frecuencia de un término dentro del documento:

```
tfidf = TfidfVectorizer()
X = tfidf.fit_transform(X)
```

Figura 9 - Ejemplo vectorización TF-IDF

Este proceso se ha repetido en todas las ocasiones que se han usado algoritmos de ML, además de haberse hecho también en el caso del perceptrón multicapa. Por último, queda construir cada modelo individualmente. A continuación se muestran los resultados *[16]* de cada modelo y sus matrices de confusión.

1-**Random Forest**:

N_estimators = 100

Bootstrap = True

RandomState = 25

Accuracy	0.975
Precision	0.973
Recall	0.973
F1	0.974

Tabla 1 - Métricas Random Forest

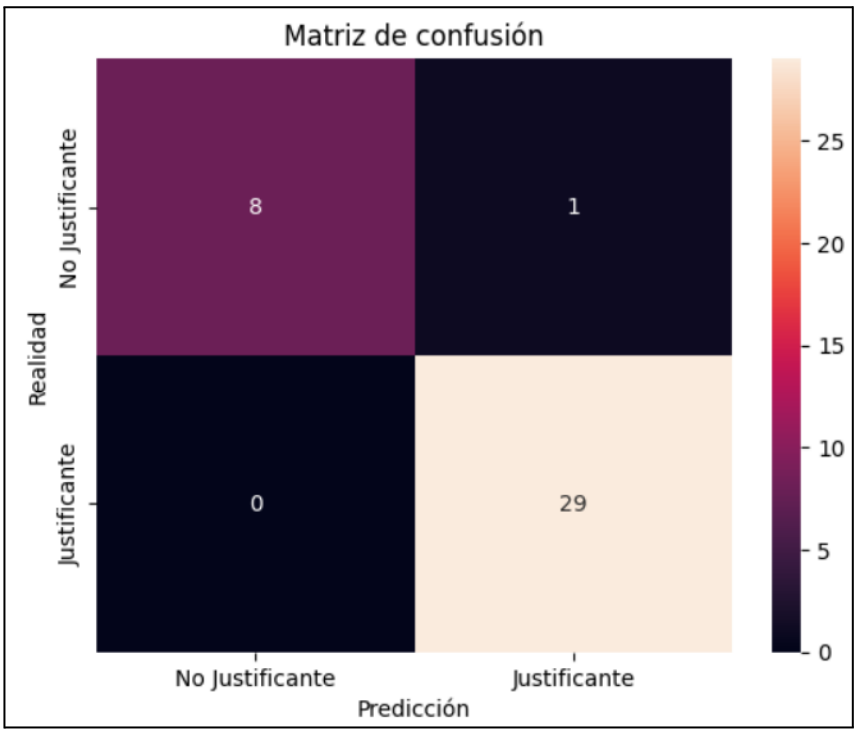


Figura 10 - Matriz de confusión Random Forest

2-**Support Vector Machine**:

Kernel = Sigmoid

Probability= True

BreakTies = True

RandomState = 25

Accuracy	0.982
Precision	0.979
Recall	0.983
F1	0.982

Tabla 2 - Métricas SVM

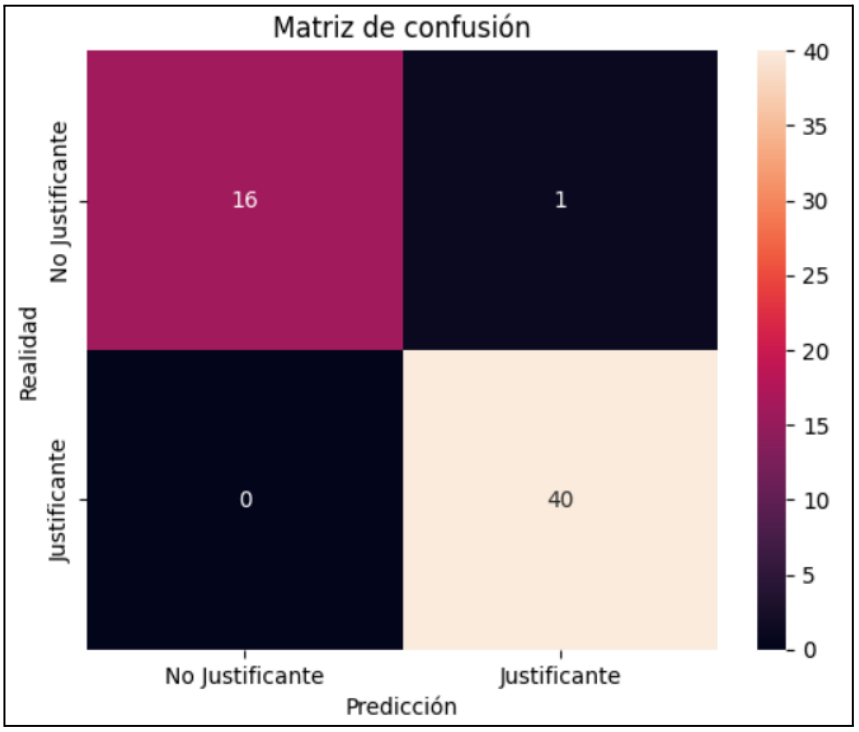


Figura 11 - Matriz de confusión SVM

3-**MultiLayer Perceptron**:

HiddenLayers = 128

Activation = Relu

MaxIter= 35

RandomState = 25

Accuracy	0.982
Precision	0.975
Recall	1
F1	0.987

Tabla 3 - Métricas Multilayer Perceptron

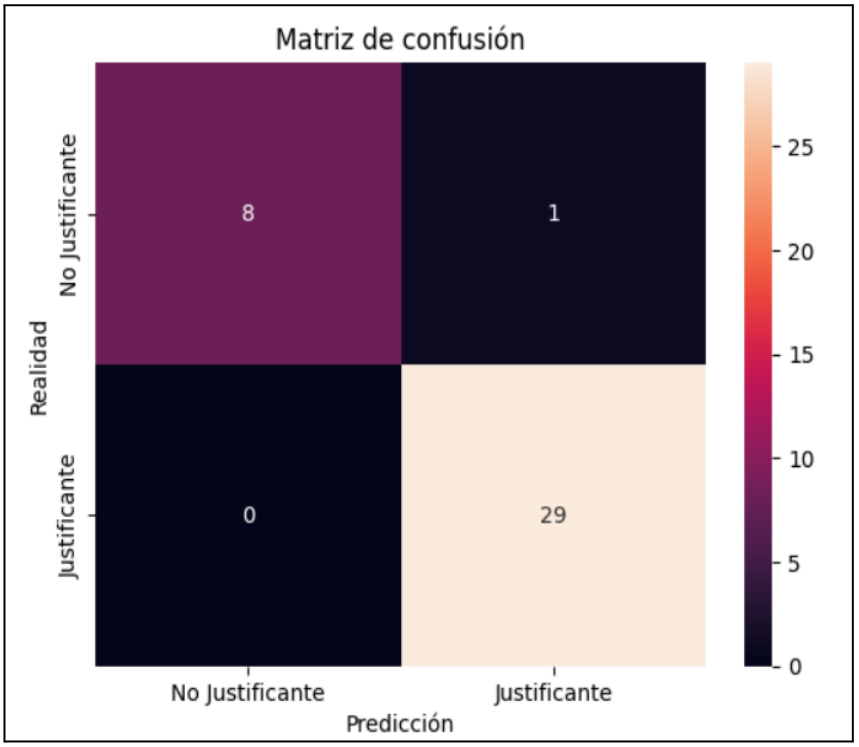


Figura 12 - Matriz de confusión MP

4-**Red Recurrente Simple**:

Optimizer = Rmsprop

Loss = Binary_crossentropy

Metrics = Accuracy

Accuracy	0.894
Precision	0.91
Recall	0.894
F1	0.9

Tabla 4 - Métricas SimpleRNN

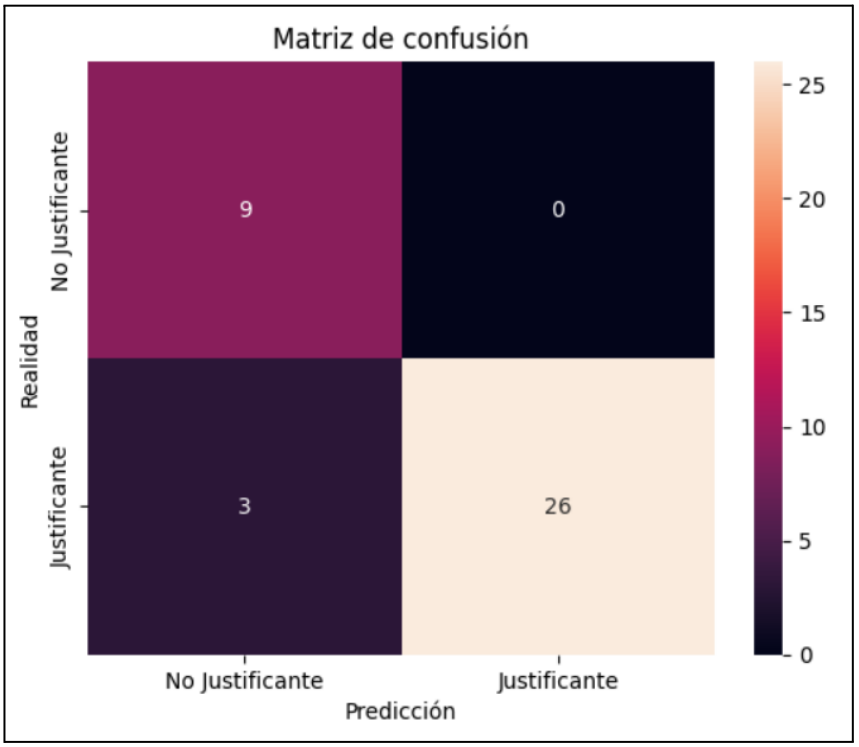


Figura 13 - Matriz de confusión SimpleRNN

5-**Red Recurrente Long-Short Term Memory**:

Optimizer = Rmsprop

Loss = Binary_crossentropy

Metrics = Accuracy

Accuracy	0.996
Precision	0.99
Recall	0.99
F1	0.995

Tabla 5 - Métricas LSTM

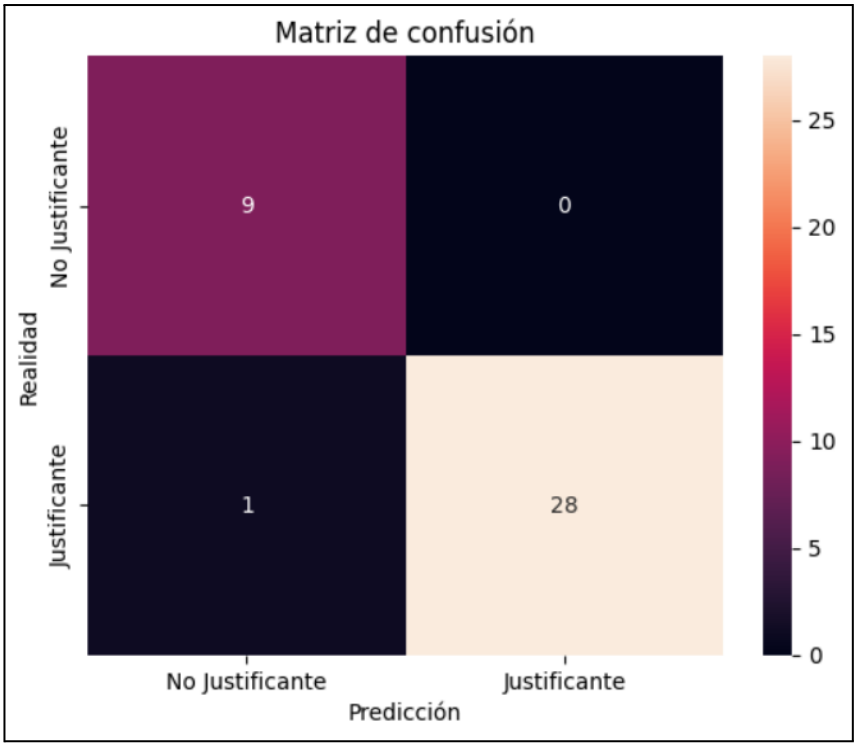


Figura 14 - Matriz de confusión LSTM

4.2 Procesamiento de Imágenes

El procesamiento de imágenes es una de las tareas principales en el campo de la visión por computador. Esto es muy común a la hora de usar *Deep Learning*, ya que las imágenes se representan como matrices de píxeles, lo que las convierte en un formato ideal para ser procesadas por redes neuronales convolucionales. La tarea principal del procesamiento de imágenes en el aprendizaje profundo es ser capaz de extraer características relevantes de la imagen, como pueden ser bordes, formas, texturas, etc. Esto se logra mediante el uso de filtros convolucionales, los cuales se aplican sobre la imagen para detectar patrones específicos en la misma. Gracias al entrenamiento de una red neuronal, estos filtros se van ajustando automáticamente para así poder identificar las características más relevantes de las imágenes.

Para resolver este problema, hay que tener en cuenta los datos disponibles y, en este caso, son documentos bancarios, por lo que su color no va a resultar relevante. Por ello lo primero que se hace es transformar una imagen en color a una escala de grises, ya que en este caso los colores no influyen. Una vez se ha hecho la transformación hay que asegurarse que todas las imágenes tengan exactamente el mismo tamaño, sin perder información, ya que la entrada a la red neuronal ha de ser de la misma forma. Estos pasos se han hecho de la siguiente manera:

```
cv2.imread(os.path.join(path,img) ,cv2.IMREAD_GRAYSCALE)
cv2.resize(img_array, (IMG_SIZE, IMG_SIZE))
```

Figura 15 - Transformación de la imagen a escala de grises

Ya en este punto, el siguiente paso es normalizar las imágenes y, es que, en este momento, tienen un valor numérico de 0 a 255, por lo que se dividen todos los valores entre 255 para así tenerlo en el rango de 0 a 1 y evitar que la red neuronal se sature:

```
X_train = X_train/255.0
X_test = X_test/255.0
```

Figura 16 - Normalización de los datos

Ya con los datos separados en entrenamiento y testeo, se obtiene la forma final de los datos, algo fundamental para la construcción de la red:

```
print(X_train.shape, y_train.shape, X_test.shape, y_test.shape)
```

```
(159, 50, 50) (159,) (40, 50, 50) (40,)
```

Figura 17 - Forma de los datos para la construcción de la red

Esto indica que se tienen 159 imágenes de entrenamiento con 50 píxeles de ancho y 50 de alto(se escoge 50 por los resultados obtenidos), 159 etiquetas

relacionadas con cada imagen y ocurre lo mismo con los datos de *testeo*.

1-**Red Recurrente Simple**:

Optimizer = Adam

Loss = Sparse_cat_cross...

Metrics = Accuracy

Accuracy	0.8
Precision	0.85
Recall	0.8
F1	0.78

Tabla 6 - Métricas SimpleRNN

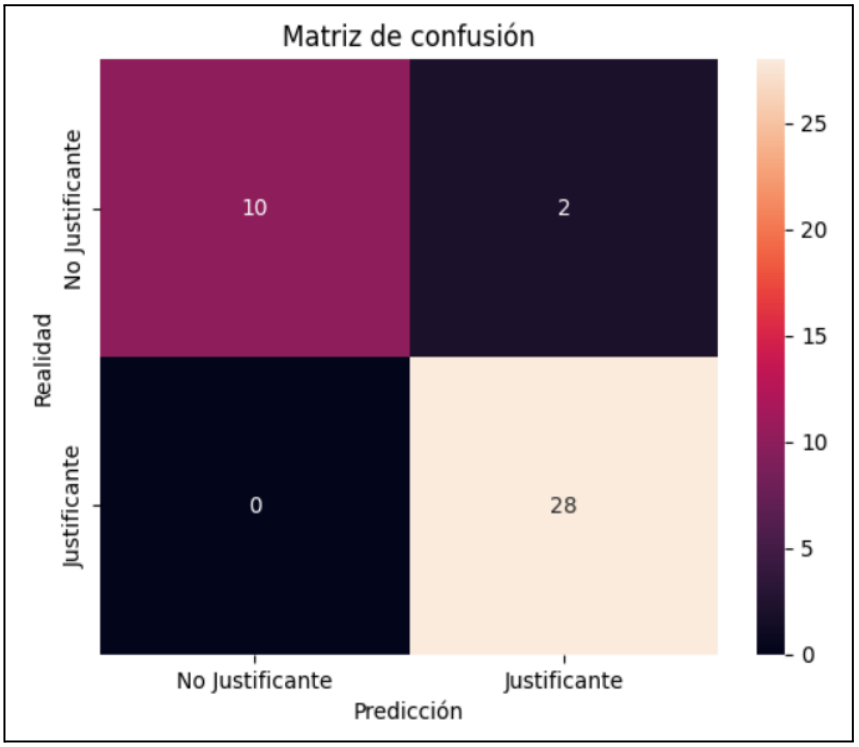


Figura 18 - Matriz de confusión SimpleRNN

2-**Red Recurrente Long-Short Term Memory**:

Optimizer = Adam

Loss = Sparse_cat_cross...

Metrics = Accuracy

Accuracy	0.825
Precision	0.81
Recall	0.82
F1	0.81

Tabla 7 - Métricas LSTM

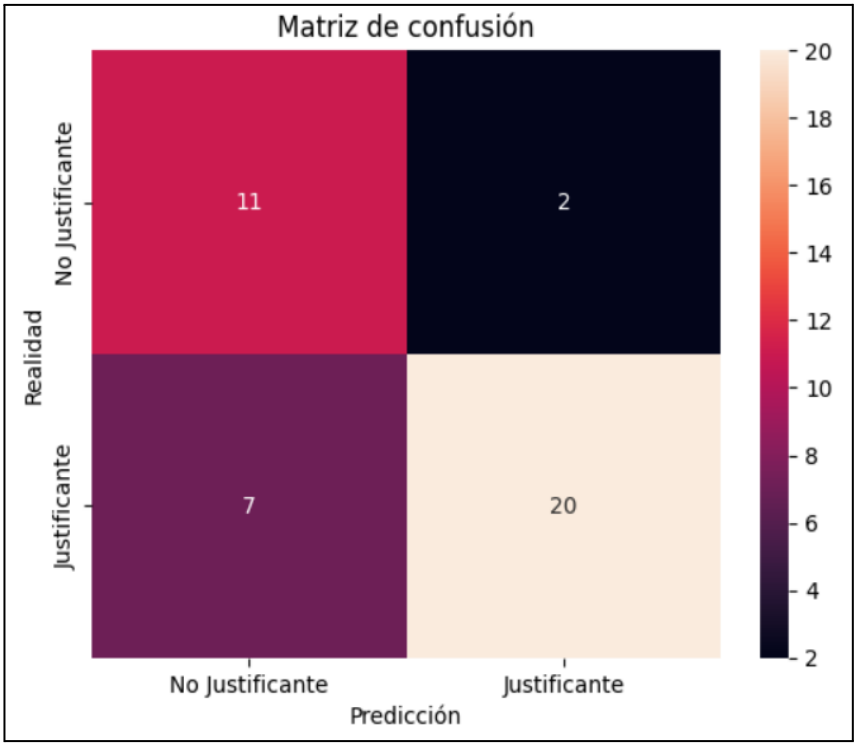


Figura 19 - Matriz de confusión LSTM

3-**Red Recurrente Gated Recurrent Unit**:

Optimizer = Adam

Loss = Sparse_cat_cross...

Metrics = Accuracy

Accuracy	0.78
Precision	0.75
Recall	0.766
F1	0.758

Tabla 8 - Métricas GRU

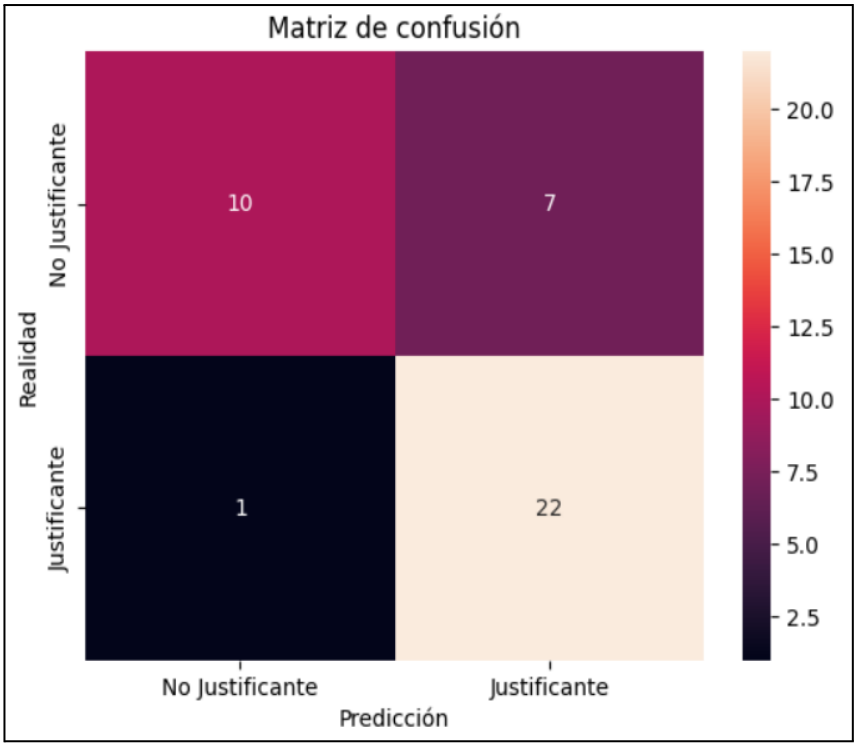


Figura 20 - Matriz de confusión GRU

4-**Red Convolucional Conv2D**:

Optimizer = Rmsprop

Loss = Binary_crossentropy

Metrics = Accuracy

Accuracy	0.925
Precision	0.92
Recall	0.93
F1	0.93

Tabla 9 - Métricas Conv2D

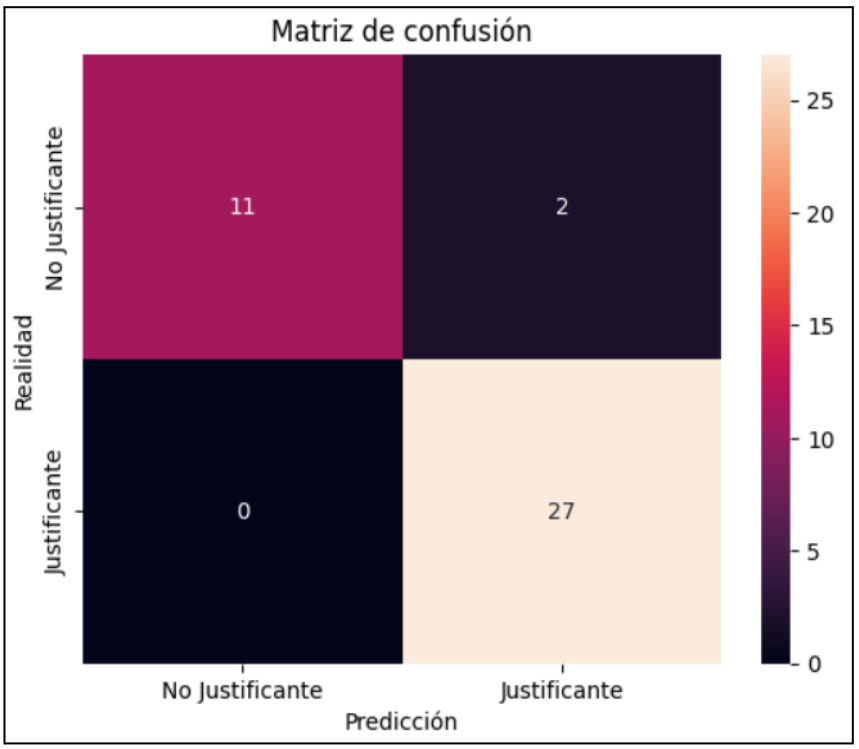


Figura 21 - Matriz de confusión C2D

5-**Red Convolucional Conv2D Separable**:

Optimizer = Rmsprop

Loss = Binary_crossentropy

Metrics = Accuracy

Accuracy	0.95
Precision	0.97
Recall	0.95
F1	0.94

Tabla 10 - Métricas Separable Conv2D

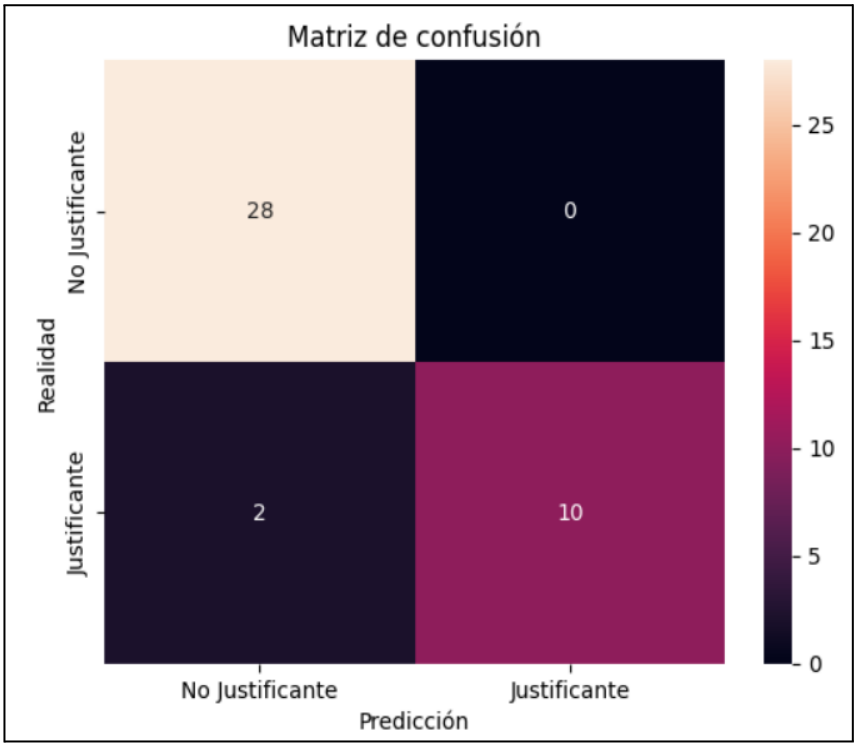


Figura 22 - Matriz de confusión Separable C2D

Capítulo 5 - Fase 2 Clasificación del importe

Esta fase tiene como objetivo final saber si se ha pagado el importe adecuado o no, es decir, una vez ya se sabe que el documento es un justificante de pago gracias a la primera fase, es momento de comprobar si se ha pagado lo debido.

5.1 OCR (Optical Character Recognition)

En este caso, el problema tiene algunas similitudes con el de la primera fase, y es que hay que seguir los mismos pasos para construir un modelo. Como es lógico, los datos son diferentes, por lo que una vez, se han hecho los mismos procesos que en la fase anterior, el dataset queda así:

	texto	justificantePago
0	importe 1560€	1
1	importe con gastos incluidos 1560€	1
2	importe “ 1560	1
3	import a transfer 1560 euros	1
4	moneds » imporis de ls operacien ra sae	1
...	...	...
257	6l0lz08zv “ 4d o€028 — w efoy ‘ goes cwo ‘ z0z...	0
258	transferencias emitidas orden de transferencias	0
259	openbank santander group	0
260	019162	0
261	redes sociales	0

236 rows × 2 columns

Figura 23 - DataFrame final Fase 2 OCR

Ya con el dataset inicial listo, es momento de exponer el problema más en detalle. Como en cualquier problema de clasificación, hay que construir un modelo para poder hacer predicciones. Para ejemplificar esto se va a usar la primera estrategia empleada, es decir, la de *Random Forest*, con la cual queda un modelo con la siguiente matriz de confusión:

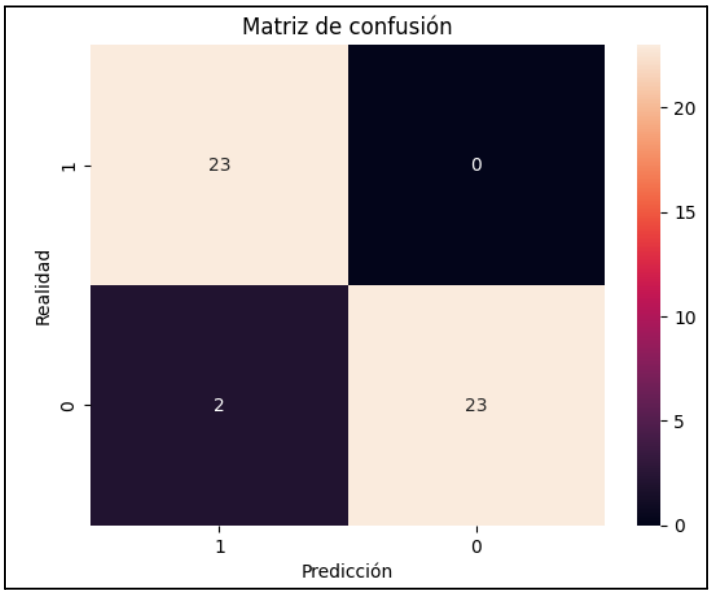


Figura 24 - Matriz de confusión modelo inicial

Como se aprecia, se trata de un modelo muy fiable, por lo que se puede considerar apto para realizar predicciones. Ya con un modelo construido, es momento de "resolver" de verdad el problema. Para ello, se va a buscar los bloques de texto en cada uno de los documentos:

Figura 25 - Ejemplo detección de texto

Como se ve en la imagen (distorsionada por privacidad de los datos), ya se tienen todas las porciones de texto de un documento. Es ahora cuando se va a recorrer cada uno de estos trozos y se va a hacer lo siguiente:

1-Se extrae el texto de cada recuadro.

2-Se le aplica un pre procesado.

3-Se hace la predicción utilizando el modelo anterior.

Ya conocidos los tres pasos aplicados a cada bloque, la manera de saber si se ha pagado el importe correcto es que desde que una de las predicciones realizadas en todo el documento sea que el importe es correcto, ya se sabrá que se ha abonado la cantidad adecuada. Por el otro lado, para saber que un importe no es correcto, se tendrá que comprobar que todas las diferentes predicciones en cada uno de los bloques sea negativa. A primera vista esto puede parecer relativamente sencillo, pero es un problema mucho más complicado, y, es que al tener tantos documentos diferentes con unos formatos tan dispares, es complicado que la librería opencv acierte a la hora de dibujar los distintos rectángulos, por lo que en caso de que no se haga de manera correcta, puede dar una predicción que no sea la esperada. Por otro lado, también se podría dar un falso positivo si se detectan los números del importe en otro lado. Además, en este caso también se depende mucho de la calidad de la imagen, ya que no es lo mismo hacer este proceso en un justificante que se haya enviado automáticamente al correo, a uno en papel que haya sido generado por un banco, y la persona le haga una foto y esa foto sea el archivo que adjunte.

A continuación, lo que se va a generar con los diferentes algoritmos va a ser el modelo, pues realmente a la hora de hacer el testeo se va a hacer siempre con

los mismos datos, por lo que los resultados que se van a enseñar van a ser los segundos, es decir, con los datos de testeo.

1-**Random Forest**:

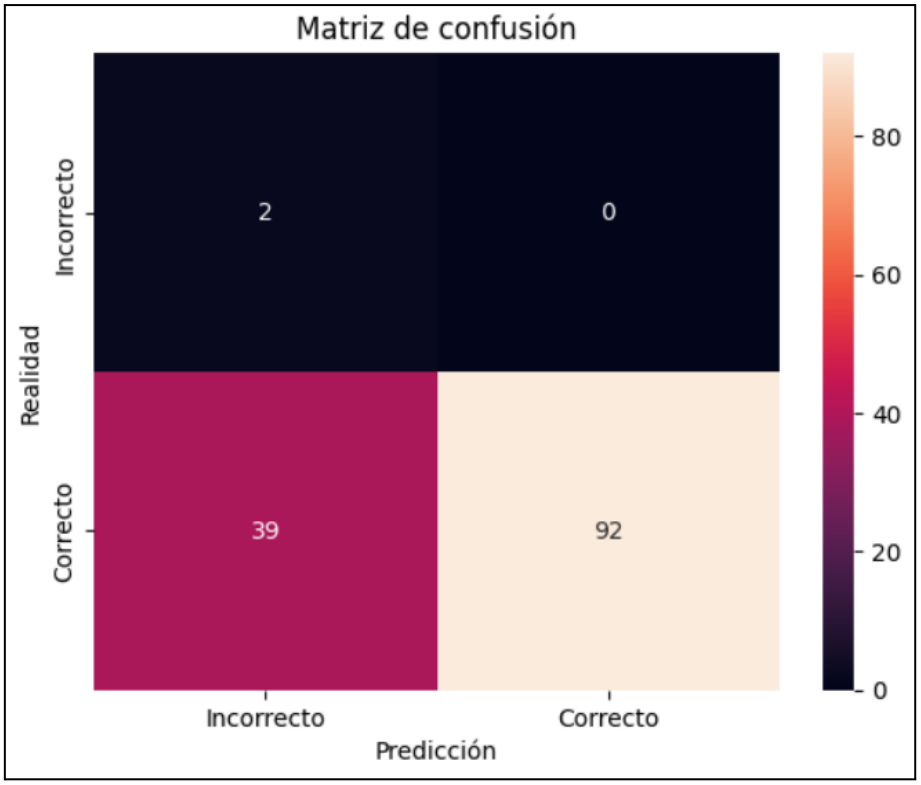


Figura 26 - Matriz de confusión Random Forest

2-**Support Vector Machine**:

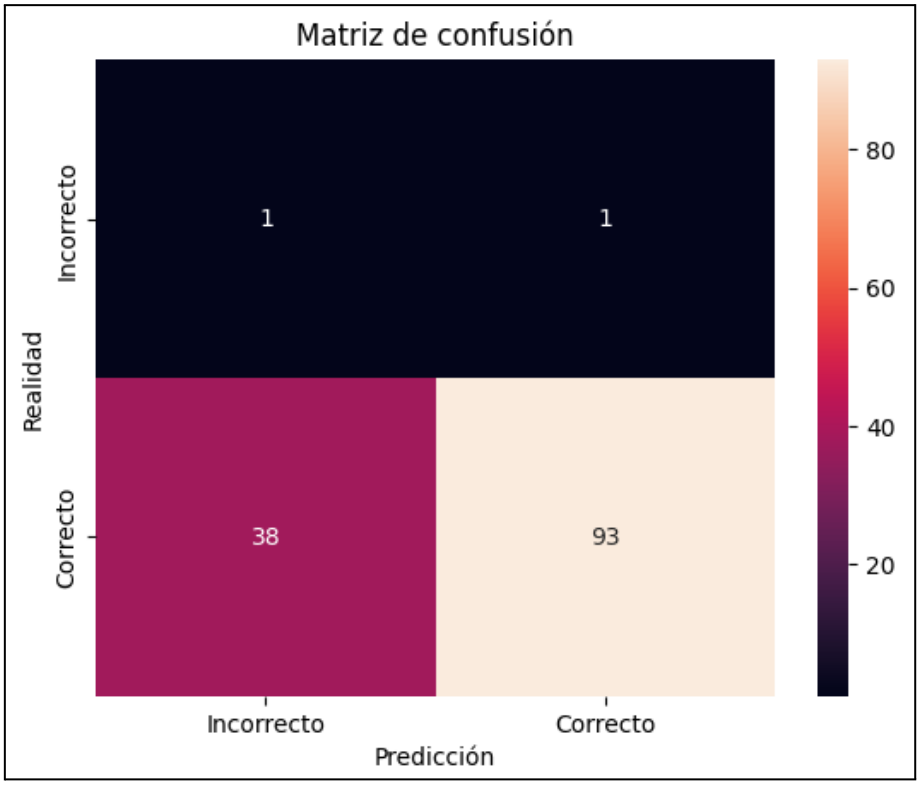


Figura 27 - Matriz de confusión SVM

3-**MultiLayer Perceptron**:

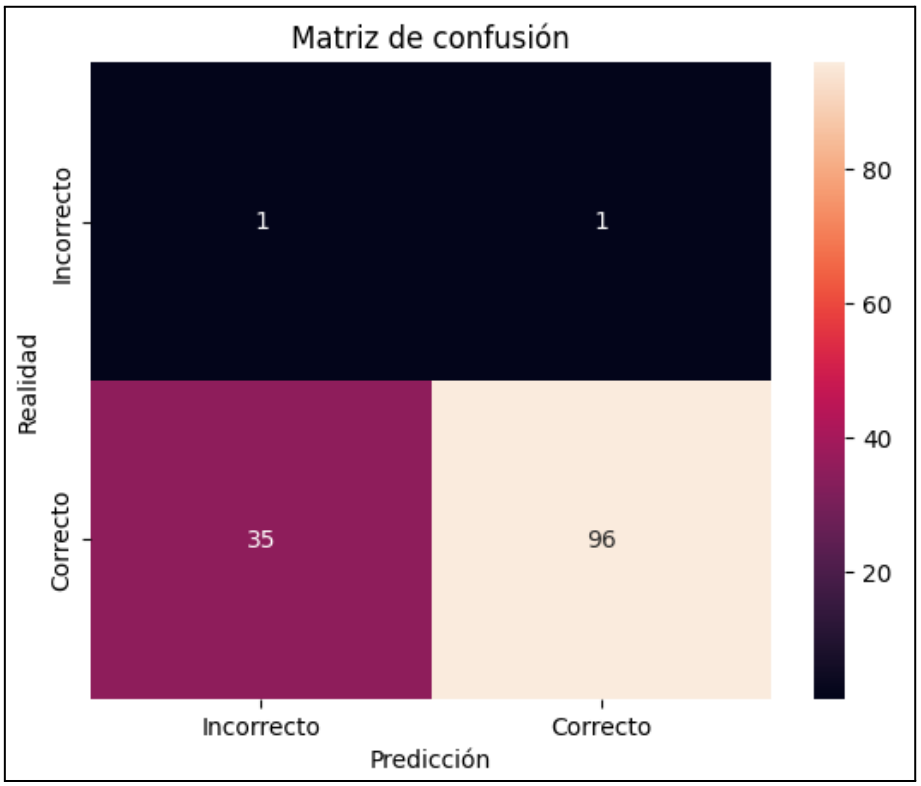


Figura 28 - Matriz de confusión Perceptrón Multicapa

4-**Red Recurrente Simple**:

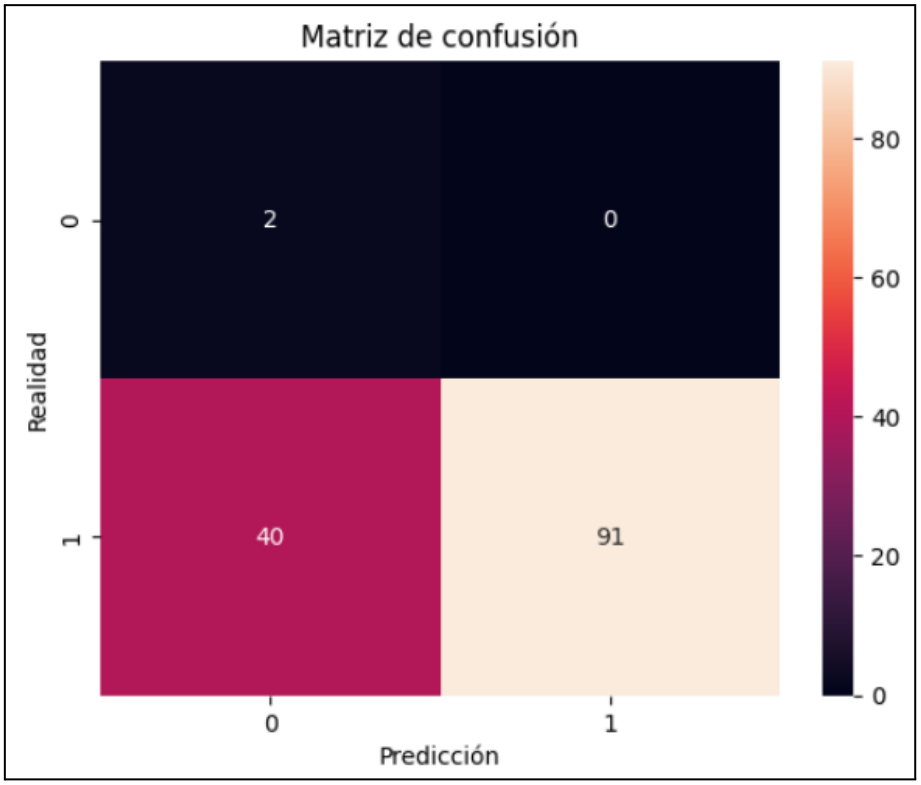


Figura 29 - Matriz de confusión SimpleRNN

5-**Red Recurrente Long-Short Term Memory**:

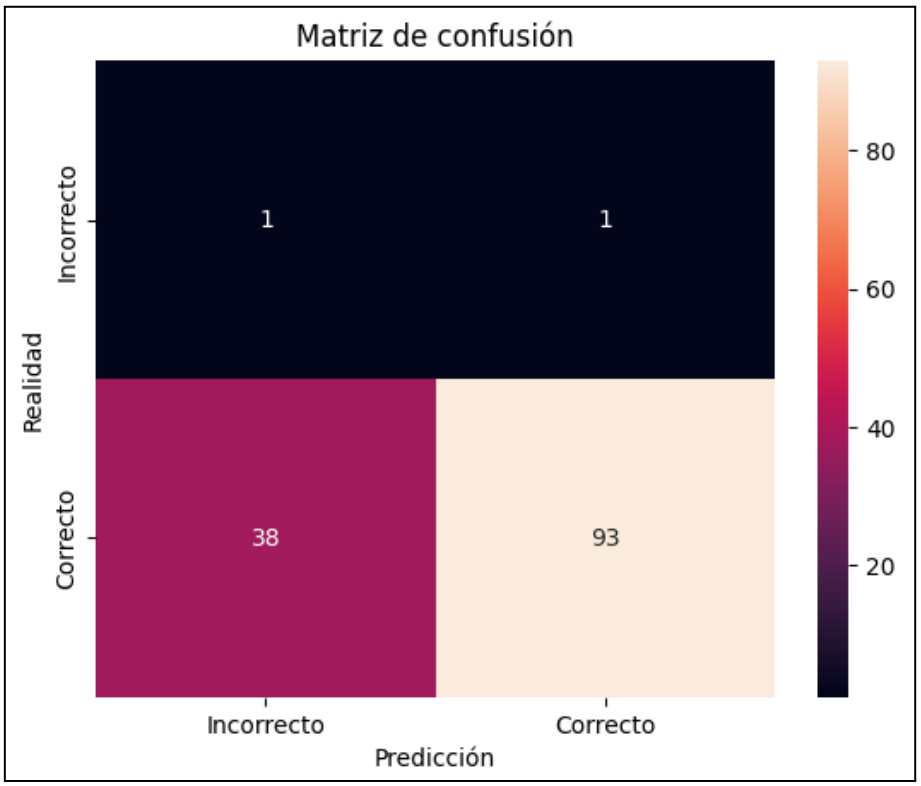


Figura 30 - Matriz de confusión LSTM

En general, se observa que los resultados no son perfectos, ya que hay muchas ocasiones en los que no da un importe correcto cuando si lo es. El motivo de esto es lo que se ha comentado anteriormente, y, es que en muchas ocasiones no se distinguen los rectángulos de manera precisa. De todas maneras, sí que es cierto que en un alto porcentaje de veces se aciertan los justificantes en los que el importe sí que es correcto.

5.2 Procesamiento de Imágenes

Esta estrategia comparte, en gran medida, los pasos de la estrategia anterior de OCR, pero, lógicamente, se centra en tratar los datos como imágenes. Como los pasos a seguir son bastante similares, no se va a profundizar mucho en ellos, pero si se van a nombrar. Lo primero que se va a tener que realizar es crear un modelo para poder realizar las predicciones. Una vez se tiene este modelo, para hacer las predicciones de sí el importe es correcto, en cada documento hay que detectar las partes del documento donde se encuentra el texto, tal y como se muestra en la *Figura 25*. Por último, ya con los rectángulos detectados, se va a aplicar el pre procesado habitual en las imágenes y se va a preguntar al modelo si el importe es correcto o no, y, en caso de serlo, se va a saber que se ha abonado la cantidad adecuada y se pasará al siguiente documento.

En este caso, no se han obtenidos resultados convincentes, a pesar de que se han explorado e implementado diversas estrategias, las cuales se van a exponer a continuación y se van a explicar los problemas encontrados, aunque en varios casos el inconveniente principal era el mismo.

Lo primero intentado ha sido resolver este problema con el MNIST Dataset *[17] [18],* el cual es un conjunto de 60000 imágenes de dígitos escritos a mano:

Figura 31 - MNIST Dataset

El motivo del uso de este dataset fue el que inicialmente se pensó en que si se entrenaba una red que reconociese los patrones de los diferentes números *[19] [20]*, luego podría detectar cada uno de los números del importe. Al construir un modelo con estos datos, los resultados del modelo fueron muy buenos, acercándose casi al 100% de acierto con los datos de testeo, pero a la hora de hacer las predicciones con los justificantes los resultados no fueron buenos, pues los dígitos de los documentos son escritos por ordenador, lo que hace que no sean muy similares a los del *dataset* que eran escritos a mano, por lo que los resultados no fueron los esperados.

Tras haber discutido los resultados en diferentes reuniones, se llegó a la idea de que como había fallado el que los dígitos escritos por un ordenador no eran

similares a los escritos a mano, se decidió intentar buscar algún dataset que fuera más similar a los números de los justificantes. Tras una búsqueda exhaustiva se encontró el DIGITS Dataset, el cual contenía más de 10000 imágenes que correspondían a los diferentes números. La ventaja inicial que se creía que tenía este conjunto de datos respecto al anterior, es que los números no eran simplemente escritos a mano, y, es que se compone de todo tipo de imágenes de dígitos: escritos a mano, a ordenador, ilustraciones, etc. En este caso se hizo lo mismo que con el anterior, pues se construyó el modelo, se comprobó que era un modelo efectivo con sus datos de testeo y se probó con los justificantes, lo que no dio buenos resultados. Tras ver que el problema no mejoraba, se volvió a probar a construir un modelo con estos datos, con la diferencia que se aplicó el mismo pre procesado tanto a las imágenes del *dataset* como a cada uno de los rectángulos de cada justificante que se usaba para hacer la predicción. En este momento, se mejoró un poco respecto al anterior modelo, pero los resultados seguían sin ser convincentes y, tras un análisis profundo, se concluyó que el problema no era tanto la construcción del modelo, sino la detección del texto. A continuación se va a mostrar gráficamente el motivo de que ninguna de las estas tres estrategias haya sido un éxito. Primero se va a enseñar un ejemplo en el que funciona correctamente:

1-Se identifica el texto:

Figura 32 - Detección correcta del texto

2-Se pre procesa y sé separa cada dígito:

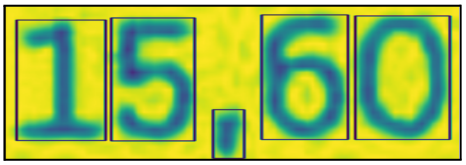


Figura 33 - Separación de los dígitos

Entonces se realiza la predicción de qué número es a cada uno y en caso de que las predicciones sean 1, 5, 6 y 0 en ese orden se considera que el importe es correcto, cosa que en este caso sí que ocurre. Ahora se van a mostrar dos casos en el que la detección del texto no es correcta, y es algo que no se puede solucionar, pues la gran variedad de formatos de los diferentes justificantes hace que cada uno sea diferente, por lo que el importe no se encuentra ni en el mismo sitio ni de la misma manera.

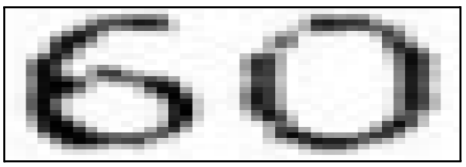


Figura 34 - Detección de texto incorrecto 1

Figura 35 - Detección de texto incorrecto 2

Las dos imágenes anteriores ejemplifican perfectamente el problema principal, y, es que, desde que la detección no sea perfecta, ya va a resultar imposible que la predicción sea correcta, pues mientras que en el primer caso solo podría ser 6 y 0, en el segundo sería 5, 6 y 0, lo que hace que ninguna de las dos predicciones coincidan con el importe correcto completo.

Al descubrir que el problema persiste a la hora de intentar predecir cada número, se ha optado por seguir una estrategia totalmente diferente. Para ello se ha construido un *dataset* personalizado con imágenes del importe correcto, las cuales han sido extraídas de cada uno de los justificantes. Para la categoría incorrecta se han extraído partes aleatorias de los documentos que no correspondían con el importe, como por ejemplo un nombre o una fecha.

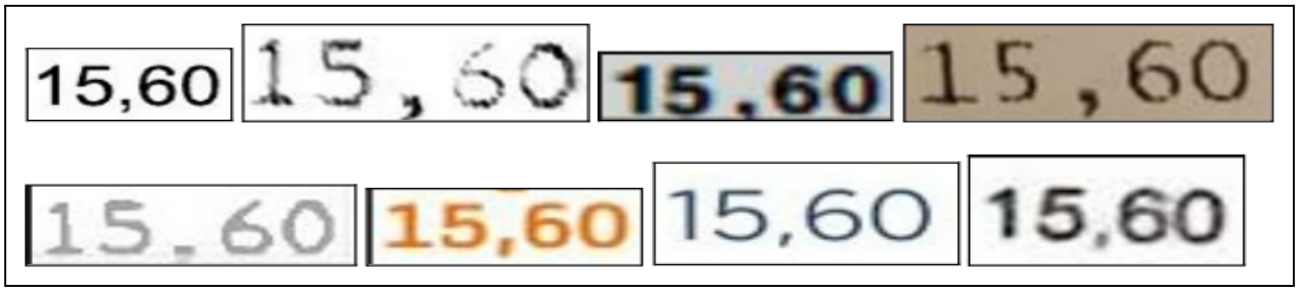


Figura 36 - Dataset Personalizado Fase 2 Imágenes

Ya con este *dataset* se ha construido el modelo y de nuevo se ha obtenido un modelo fiable. Para testearlo se ha recorrido cada documento por trozos, y no detectando el texto debido a los errores anteriores, lo que ha mejorado los resultados, aunque han seguido siendo insuficientes, además que el consumo computacional era extraordinariamente elevado. La manera de probar cada documento es la siguiente:

Figura 37 - Recorrido documento para testeo

Como se aprecia, se va recorriendo la imagen por “trozos” de izquierda a derecha y de arriba hacia abajo, y se va realizando una predicción en cada rectángulo hasta que una de ellas dé el importe como correcto. Como se dijo anteriormente, los resultados aun así no han sido los esperados, y como se evidencia en la imagen, computacionalmente el consumo es enorme.

Tras haber probado todas estrategias y tras haber investigado otras que no se estimaba que fueran a mejorar lo obtenido, se ha decidido no continuar con esta fase, además de que con la técnica de OCR ya había buenos resultados.

Capítulo 6 - Fase 3 Clasificación del concepto

En esta última fase, se va a proceder de una manera muy similar, pues el objetivo es muy parecido, pero en vez de saber si se ha pagado el importe correcto, ahora el enfoque será en sí el concepto de la transferencia es el adecuado.

6.1 OCR (Optical Character Recognition)

Como se comentó en la introducción de esta fase, el procedimiento va a ser muy parecido, con la diferencia de que el dataset va a ser bastante diferente, ya que antes se trataba de números y ahora de una frase bastante más compleja, pues a la hora de poner un mismo concepto pueden existir formas muy diferentes, por el hecho de que en muchas ocasiones hay límite de caracteres, por lo que las abreviaturas de las palabras van a ser muy dispares.

El conjunto de datos, una vez hechos todos los pasos comentados en las fases anteriores, es el siguiente:

	texto	justificantePago
0	['pruebas', 'selectivas', 'ingreso', 'grupo', ...	1
1	['prueb', 'selec', 'ina', 'prof', 'aux', 'cons...	1
2	['asas', 'examen', 'pruebas']	1
3	['pruebas', 'selec', 'ingreso', 'pro', 'auxili...	1
4	['tasa', 'ull', 'oposicion', 'auxiliar', 'cons...	1
...	...	...
195	['mporte', 'tofal']	0
196	['caja', 'kural', 'sc']	0
197	['debwnteaioe']	0
198	['operacien', 'realizada', '«', 'fecha', 'hor']	0
199	['precio', 'aplicado', 'servicio']	0

200 rows × 2 columns

Figura 38 - DataFrame final Fase 2 OCR

En cuanto a la detección de los bloques de texto, va a ser exactamente igual que en la segunda fase (se muestra en la figura 25), al igual que la manera de construir los modelos para luego ponerlos a prueba con otros datos. Estos últimos resultados son los que se van a mostrar en las siguientes páginas.

1-**Random Forest**:

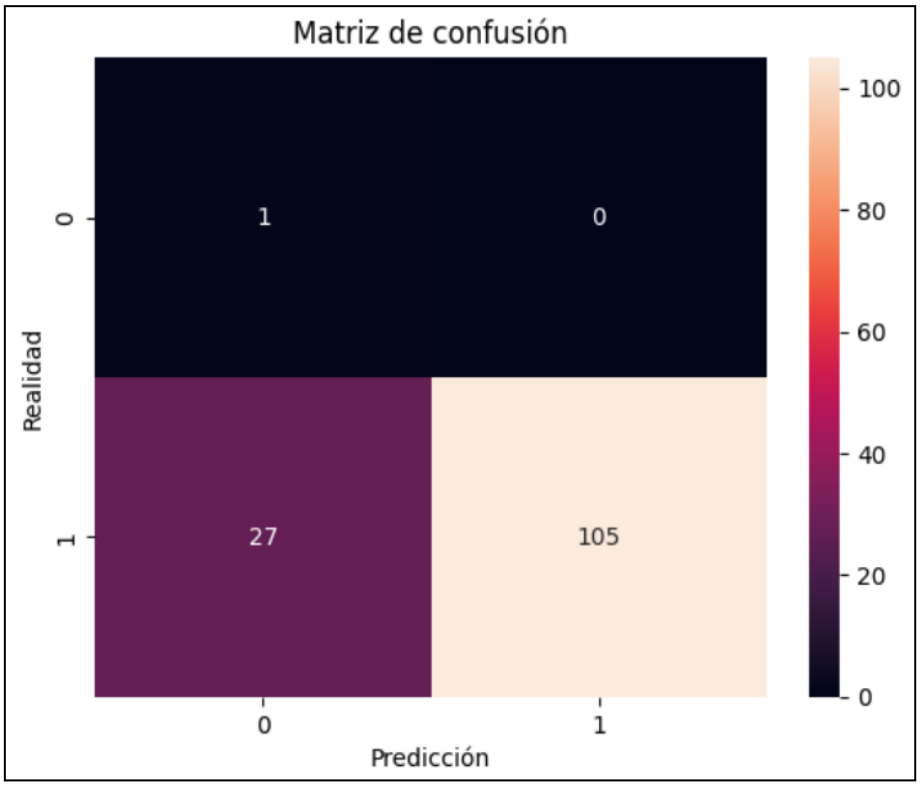


Figura 39 - Matriz de confusión Random Forest

2-**Support Vector Machine**:

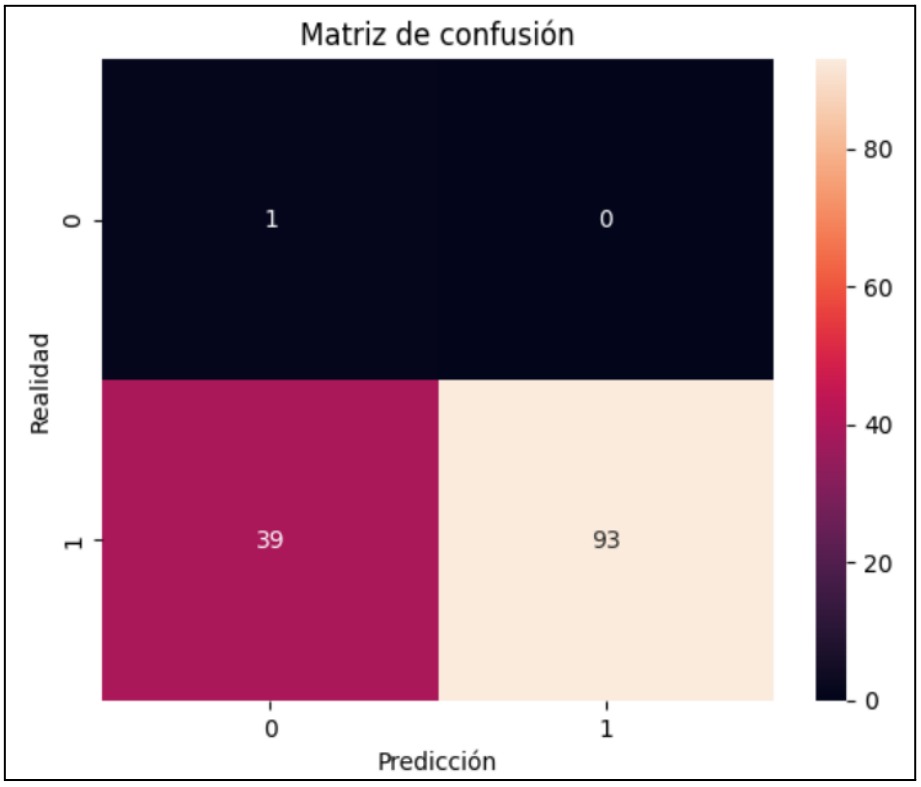


Figura 40 - Matriz de confusión SVM

3-**MultiLayer Perceptron**:

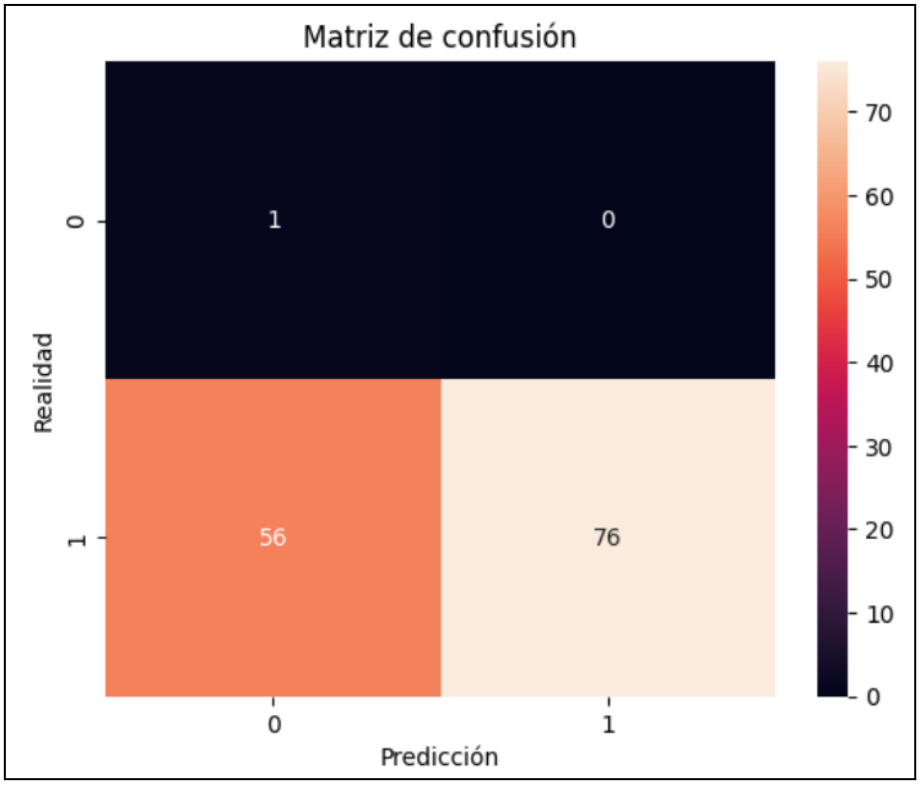


Figura 41- Matriz de confusión Perceptrón Multicapa

4-**Red Recurrente Simple**:

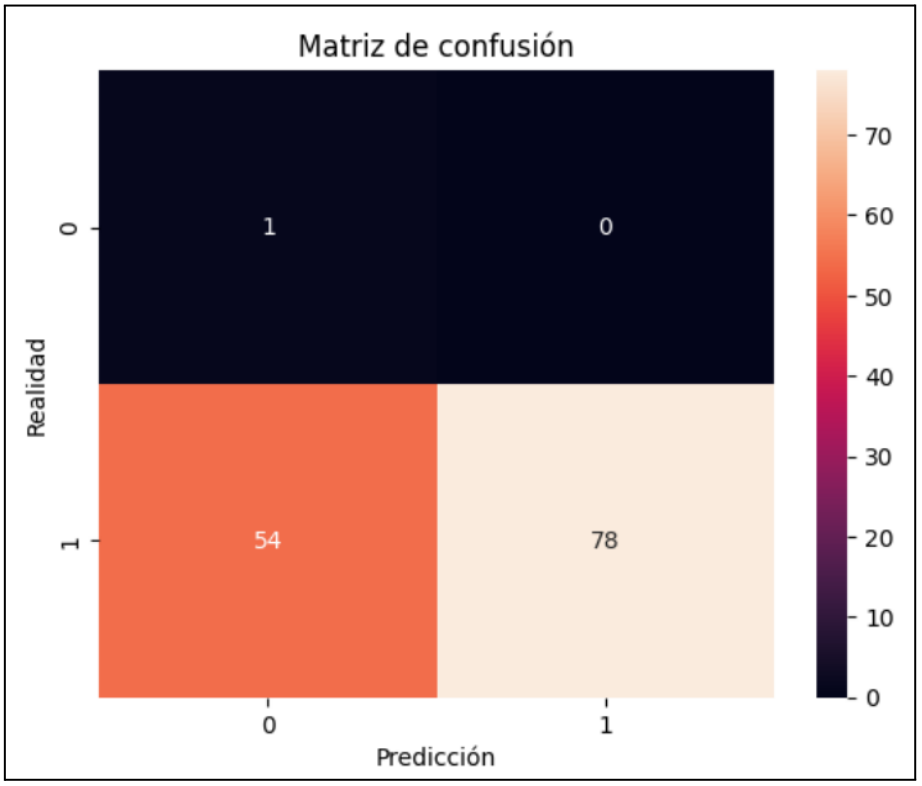


Figura 42 - Matriz de confusión SimpleRNN

5-**Red Recurrente Long-Short Term Memory**:

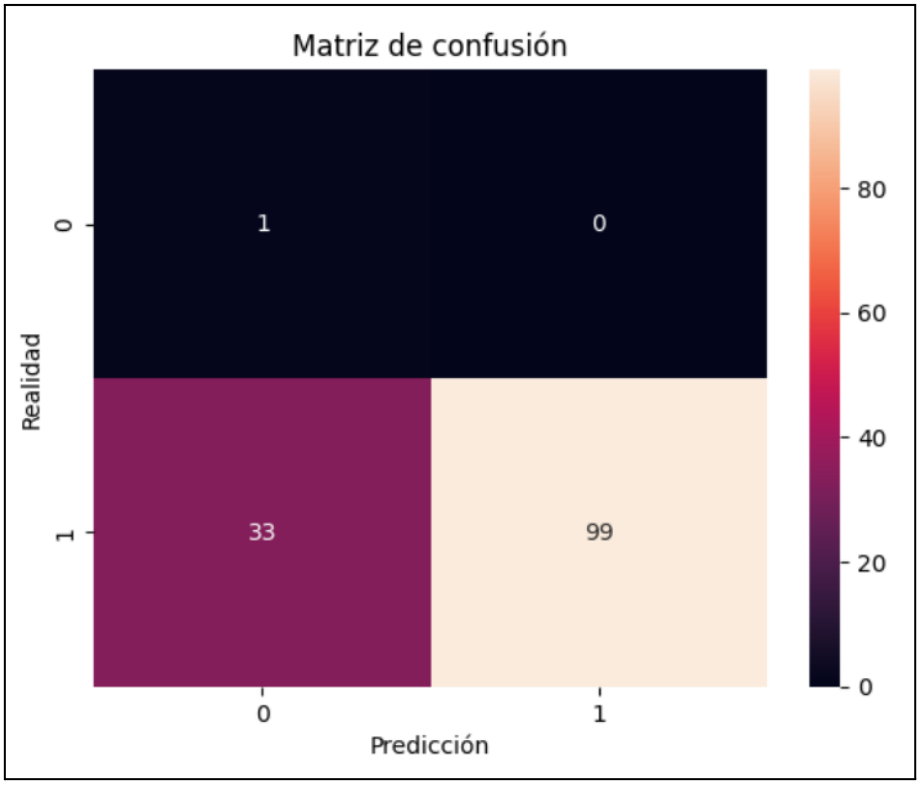


Figura 43 - Matriz de confusión LSTM

6.2 Procesamiento de Imágenes

En esta última estrategia de la última fase, el objetivo es claro, y es intentar detectar si el concepto del documento es correcto tratando los datos como imágenes.

En un principio, esto se iba a intentar resolver como se había hecho en la fase anterior con el importe. Pero como se ve a simple vista, el problema es mucho más complejo, pues antes se querían clasificar 9 variables diferentes correspondientes a cada uno de los dígitos, y ahora no solo habría que clasificar cada una de las veintinueve letras del abecedario, pues al estar la posibilidad de que estas letras estuviesen escritas en mayúscula el problema ya se multiplica por dos, además de tener que incluir las vocales con tildes.

Una vez se es consciente de la complejidad computacional del problema, sería momento de resolverlo, pero gracias a que es la última fase y se ha intentado con un problema muy similar, pero más sencillo y no ha sido exitoso, se pueden intuir los resultados que se van a obtener. Es por esto que, tras consultarlo con el profesorado, se ha decidido que no se va a llevar a cabo esta fase, pues se van a encontrar los mismos problemas que antes, pero aumentados.

En el primer caso, en el que se clasificaba dígito a dígito, ahora habría que hacerlo letra a letra, algo que es totalmente inviable, pues se podría dar la casualidad que un concepto estuviese perfecto y tuviese todas las letras, pero por norma general los conceptos suelen estar llenos de abreviaciones y en la inmensa mayoría de los casos, estas abreviaturas no están reconocidas en el lenguaje, sino que cada persona las hace de manera diferente, por lo que las diferencias que se van a encontrar en los conceptos van a ser enormes.

Por otro lado, antes cabía la posibilidad de construir un dataset personalizado e intentar recorrer la imagen, y, es que al ser simplemente cuatro dígitos, se puede entender que si se va recorriendo la imagen por trozos se podría llegar a tener buenos resultados, pues al fin y al cabo el tamaño y la forma de presentar estos dígitos va a ser similar por norma general. Esto no ocurre en este caso, ya que el concepto, dependiendo del justificante que sea, va a tener una forma muy diferente a otro. A continuación se van a mostrar imágenes que resalten la diferencia:

Figura 44 - Forma Concepto 1

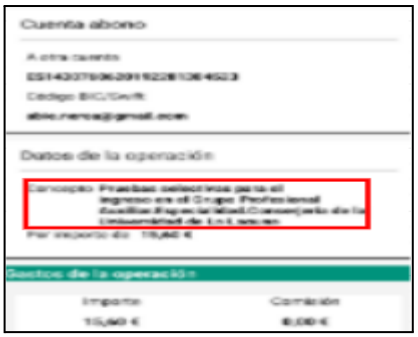


Figura 45 - Forma Concepto 2

Como se ve, las formas de los conceptos no tienen nada que ver, por lo que sería inviable encontrar una forma que englobase a todos los conceptos de todos los diferentes justificantes. Es por esto y todo lo comentado anteriormente que, finalmente, esta fase no se va a llevar a cabo.

Capítulo 7 - Summary and Conclusions

It can be affirmed that the objective of the Final Degree Project has been satisfactorily completed, since at the end of it has been possible to achieve a reliable model for each of the phases. Results have also been obtained in accordance with the strategies used, since in all the phases the models made with NLP techniques have been considerably better than those made with image processing, which is totally logical, since the whole time it was a problem of understanding the text.

On a personal level, expectations have been far exceeded, as it has been seen how from nothing, it has been possible to reach a final solution with very good results. Of course, the road has been very hard and not at all easy, as countless difficulties have had to be overcome as they arose during the development of the project. In addition, there is also a high degree of satisfaction in seeing that the initial personal estimates have been fulfilled after having resolved the different phases.

On the other hand, it is evident that time plays a fundamental role in the development of a project, as it is very likely that if there had not been a deadline, the results could have been improved in some cases.

As for future improvements, the main one would be to find an OCR (Optical Character Recognition) tool that would be much more accurate, since it is believed that if it had been possible to detect the desired text at each moment, the confusion matrices and the metrics of the models would have been much more favourable.

To conclude, it is thought that this work has been a very good end to such an enriching formative stage, since without all the knowledge acquired during these years, all this could not have been carried out. It is also a very clear example of the importance of Computer Engineering in the world, as it has been seen how what was initially born as an idea to solve an existing problem, has ended up being a process to find a solution to this problem, something that has finally been achieved.

Capítulo 8 - Presupuesto

El presupuesto se ha calculado teniendo en cuenta dos factores principales, el equipo de trabajo necesario para el desarrollo y las horas totales invertidas, tanto en investigación, desarrollo y en documentación. Se va a estipular que el salario de un ingeniero informático son 20 euros la hora. El presupuesto total se desglosa en le siguiente tabla:

Tipos	**Descripción**	**Precio**
Equipo de trabajo	Dell Inspiron 15 300 Dell S Series S2721HS 27"	640 € 225 €
Honorarios	550 horas de trabajo	11000 €
Total		11865 €

Tabla 11 - Presupuestos

Bibliografía

[1] AriaBarnes. (2023, marzo 31). Machine Learning (ML) vs Artificial Intelligence (AI) — crucial differences. Data Science Central; TechTarget. https://www.datasciencecentral.com/machine-learning-ml-vs-artificial-intelligence-ai-crucial-differences/

[2] Voulodimos, A., Doulamis, N., Doulamis, A., & Protopapadakis, E. (2018). Deep learning for computer vision: A brief review. Computational Intelligence and Neuroscience, 2018, 7068349. https://doi.org/10.1155/2018/7068349

[3] Svetnik, V., Liaw, Tong, C., Culberson, J. C., Sheridan, & Feuston, B. P. (2003). Random forest: a classification and regression tool for compound classification and QSAR modeling. Journal of Chemical Information and Computer Sciences, 43(6), 1947–1958. https://doi.org/10.1021/ci034160g

[4] Hsu, C.-W., Chang, C.-C., & Lin, C.-J. (s/f). A practical guide to support vector classification. 2023, de https://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf

[5] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436–444. https://doi.org/10.1038/nature14539

[6] Lecun, Y., Eon Bottou, L., Bengio, Y., & Ha, P. (s/f). GradientBased learning applied to document recognition. Stanford.edu. Recuperado el 11 de mayo de 2023, de http://vision.stanford.edu/cs598_spring07/papers/Lecun98.pdf

[7] Pascanu, R., Mikolov, T., & Bengio, Y. (2012). On the difficulty of training Recurrent Neural Networks. En arXiv [cs.LG]. https://arxiv.org/pdf/1211.5063.pdf

[8] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

[9] RNNs, GRUs, LSTMs, Atención, Seq2Seq, y Redes de Memoria · Aprendizaje Profundo 2023 https://atcold.github.io/pytorch-Deep-Learning/es/week06/06-2/

[10] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. Communications of the ACM, 60(6), 84–90. https://doi.org/10.1145/3065386

[11] Chollet, F. (2016). Xception: Deep learning with depthwise separable convolutions. En arXiv [cs.CV]. https://arxiv.org/pdf/1610.02357.pdf

[12] Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification. En arXiv [cs.CL]. https://arxiv.org/pdf/1408.5882.pdf

[13] Liu, X., Liang, D., Yan, S., Chen, D., Qiao, Y., & Yan, J. (2018). FOTS: Fast Oriented Text Spotting with a unified network. En arXiv [cs.CV]. https://arxiv.org/pdf/1801.01671.pdf

[14] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. En arXiv [cs.CV]. https://arxiv.org/pdf/1409.1556.pdf

[15] Bafna, P, Pramod, & Vaidya, A. (2016). Document clustering: TF-IDF approach. 2016 International Conference on Electrical, Electronics, and Optimization Techniques https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7754750

[16]Davis, J., & Goadrich, M. (2006). The relationship between Precision-Recall and ROC curves. Proceedings of the 23rd international conference on Machine learning - ICML '06. https://dl.acm.org/doi/10.1145/1143844.1143874

[17] Burges, C. J. C. (s/f). MNIST handwritten digit database, Yann LeCun, Corinna Cortes and Chris Burges. http://yann.lecun.com/exdb/mnist/

[18] Ahlawat, S., Choudhary, A., Nayyar, A., Singh, S., & Yoon, B. (2020). Improved handwritten digit recognition using convolutional neural networks (CNN). Sensors (Basel, Switzerland), 20(12), 3344. https://doi.org/10.3390/s20123344

[19] Goodfellow, I. J., Bulatov, Y., Ibarz, J., Arnoud, S., & Shet, V. (2013). Multi-digit number recognition from Street View imagery using deep convolutional neural networks. En arXiv [cs.CV]. https://arxiv.org/pdf/1312.6082.pdf

[20] Vetriselvi, T., Laxmi Lydia, E., Nandan Mohanty, S., Alabdulkreem, E., Al-Otaibi, S., Al-Rasheed, A., & F. Mansour, R. (2022). Deep learning based license plate number recognition for smart cities. Computers, Materials & Continua, 70(1), 2049–2064. https://www.techscience.com/cmc/v70n1/44448/pdf