UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Universidad Politécnica de Valencia

Departamento de Sistemas Informáticos y Computación

Detección de signos de puntuación en documentos de texto manuscrito

Trabajo Final de Máster

Máster en Inteligencia Artificial, Reconocimiento de Formas e Imagen Digital

Alumno: Miguel Hernández Salmerón

Profesor: Dr. Carlos David Martínez Hinarejos

Valencia, Septiembre 2017

Agradecimientos:

A mi tutor,
por darme la oportunidad de realizar un proyecto tan interesante.

A mi familia,
por el apoyo que he recibido.

A mi novia,
por ayudarme en los momentos difíciles.

A mi hermano,
por prestarme su ordenador para realizar el proyecto.

Índice de contenido

Resumen

Reconocer texto manuscrito es una tarea compleja y no siempre se obtienen buenos resultados, ya sea por el estado del texto o por el tipo de escritura. Es por ello por lo que sigue siendo objeto de investigación, donde cada vez se van obteniendo nuevas técnicas y métodos diferentes.

Actualmente la segmentación del texto se realiza por línea, pues los sistemas de reconocimiento extraen cada línea del texto para después analizarla. Sin embargo, las líneas extraídas no tienen por qué ser coherentes, pues normalmente no coincide que cada línea sea una frase completa.

El objetivo del proyecto es localizar los signos de puntuación que se encuentran en el texto para fragmentar éste en frases completas que tengan un significado coherente. Para ello se ha decidido emplear una técnica utilizada para clasificar diferentes tipos de imágenes, utilizando una red neuronal convolucional. Se ha entrenado la red neuronal con imágenes de los diferentes signos de puntuación, con el fin de reconocerlos a lo largo del texto y almacenar la posición en la que se encuentran.

Para la realización del proyecto se ha utilizado un manuscrito del año 1853, del cuál se han obtenido las imágenes tanto de los puntos como de las comas para generar los diferentes corpus que se han empleado.

Abstract

Handwriting recognition is a complex task and not always we obtain good results, either by the text conditions or by the kind of writing. That is why it remains research topic, where every time new techniques and methods are obtained .

Currently text segmentation is done by line, as the recognition systems extract each line of the text and then analyze it. However, the extracted lines do not have to be coherent, since is normally does not coincide with a complete sentence.

The goal of this project is to locate the punctuation marks found in the text to fragment it into complete sentences that have a coherent meaning. For this purpose, it has been decided to use a technique used to classify different types of images by using a convolutional neural network. The neural network has been trained with images of the different punctuation marks in order to recognize them throughout the text and to store the position in which they are.

To make this project a manuscript of the year 1853 was used, from which the images of both the points and the commas have been obtained to generate the different corpus that have been used.

Capítulo 1. Objetivo del trabajo

El objetivo de este trabajo es la detección de signos de puntuación en texto manuscrito con el fin de detectar frases completas para una segmentación adecuada, ya que uno de los problemas del reconocimiento de texto manuscrito es que las segmentaciones del texto se realizan por líneas, en lugar de por frases. Esto se debe a que, en los textos digitalizados, lo primero que se detecta en una hoja son las diferentes líneas de texto. Estas líneas se aíslan para, posteriormente, analizar las palabras que contienen esas líneas, y de esta manera poder formar una frase para su análisis.

El problema es que una frase no tiene por qué comenzar al principio de cada línea ni terminar al final, pues puede comenzar y terminar en cualquier posición. Esto hace más difícil el análisis sintáctico y semántico de las oraciones. Sin embargo, sí en lugar de dividir el texto por líneas se dividiese por signos de puntuación, las frases tendrían mayor sentido y los análisis posteriores serían mucho más sencillos y precisos. De ahí la importancia de este trabajo a la hora de detectar los signos de puntuación y poder dividir el texto en frases coherentes.

El texto manuscrito escrito con letra cursiva es muy difícil de reconocer debido a la variedad de estilos de escritura a mano, a la caligrafía y también a que, en la escritura cursiva, las letras se superponen e incluso se pueden llegar a tocar unas con otras. También afecta la distancia que separa unos caracteres de otros e incluso la separación entre palabras que, al no ser siempre igual, suele producir errores en el reconocimiento y hace aún más difícil el proceso.

Los propios aparatos usados afectan al reconocimiento de textos manuscritos, pues los dispositivos que se encargan de obtener la imagen puede introducir niveles de grises en el fondo que no pertenecen a la imagen original. Otro problema que puede darse en la imagen es que el aparato introduzca ruido, afectando a los píxeles y distorsionando la imagen.

Si se traza una vertical entre una palabra y la siguiente, es muy probable que, en el hueco entre ambas, una parte de una de las palabras o ambas crucen esa línea, ya sea por arriba o por debajo. Es por ello que es difícil tender una línea vertical donde decidir la frontera de separación entre una palabra y la siguiente.

Otro problema que tiene la detección de los signos de puntuación es que, al no tener un patrón predecible en cuanto a la probabilidad de que pueda aparecer uno de ellos, hace que sea muy difícil utilizar técnicas como los modelos ocultos de Markov, pues la misma frase puede tener, por ejemplo, una coma en diferentes posiciones, dando un significado distinto en cada caso.

Los modelos de estados finitos son útiles para el reconocimiento de frases, ya que las palabras llevan una relación sintáctica y semántica entre ellas, pero para la detección de puntos y comas no es muy apropiado, pues un signo de puntuación puede ir colocado en diferentes sitios de una frase modificando el significado. Es por ello que se hace más difícil calcular la probabilidad que hay de que, después de una palabra, aparezca un signo de puntuación. El siguiente ejemplo muestra la misma frase con significados diferentes:

Los soldados, cansados, volvieron al campamento.

Los soldados cansados volvieron al campamento.

En el primer caso, todos volvieron al campamento y todos estaban cansados.

En el segundo, sólo regresaron los que estaban cansados.

Esto refleja en valor que tienen los signos de puntuación en una oración.

1.1. Motivación

Se trata de un proyecto que pretende hacer uso de técnicas que actualmente están siendo empleadas en otros ámbitos, como es la clasificación de imágenes, para aportar un nuevo enfoque a las técnicas de reconocimiento de la escritura, con el fin de obtener buenos resultados y así poder abrir nuevos frentes de investigación.

No hay mayor motivación que la de avanzar en la ciencia y aportar mejoras que sirvan para obtener nuevos métodos y nuevas técnicas con las que progresar.

Capítulo 2. Estado del arte

Tras varias búsquedas se han encontrado diversos artículos donde se describe el estado del arte en cuanto a reconocimiento de texto manuscrito. También se describe el estado del arte en cuanto a visión por computador, pues parte de las técnicas que se emplean en visión por computador se han utilizado para realizar este proyecto.

2.1. Estado del arte en reconocimiento de texto manuscrito

Con respecto al reconocimiento de la escritura, existen dos ramas diferentes, reconocimiento de la escritura en tiempo real (*on-line*) y reconocimiento de la escritura (*off-line*), en el cual se trata de reconocer el texto desde una imagen digitalizada, de un documento ya escrito en papel con anterioridad.

Uno de los artículos, en el cuál se ha basado este proyecto es [1], que recoge las diferentes técnicas de transcripción automática de la escritura, entre las cuáles se encuentra el análisis de diseño y métodos de extracción de líneas de texto, técnicas de preprocesamiento de imágenes, del modelado léxico y de lenguaje y de los modelos ocultos de Markov o HMM (por sus siglas en inglés, *Hidden Markov Model*). El artículo habla de la adaptación y aplicación de estas técnicas en documentos históricos, con diferentes estilos de manuscritos de diversos lugares y periodos de tiempo.

El reconocimiento *on-line* se encarga de identificar los caracteres conforme se van escribiendo. Se trabaja sobre una superficie activa, normalmente una pantalla táctil o tableta, que recoge el movimiento trazado sobre ella. Los caracteres son analizados al tiempo que van siendo escritos.

Para el análisis de este tipo de escritura se utilizan los modelos ocultos de Markov, ya que se trata de una potente herramienta que va calculando la probabilidad de la siguiente palabra según se termina la anterior, agilizando el proceso.

Es el modelo que se utiliza actualmente en reconocimiento de la escritura en tiempo real, por ejemplo en los móviles, *tablets*, …, pues su funcionamiento es rápido y los algoritmos están muy optimizados.

El reconocimiento *off-line* se encarga de reconocer la escritura en documentos digitalizados donde, previamente, se ha escrito a mano sobre papel. Por tanto, ésta será la rama base desde donde brotará, los métodos que se traten en este proyecto, ya que los textos que se van a trabajar son textos manuscritos del 1853 ya digitalizados.

El artículo hace mención de los sistemas de reconocimiento óptico de caracteres u OCR (por sus siglas en inglés, *Optical Character Recognition*) que, a pesar de reconocer texto escrito a máquina e impreso con muy buena precisión, no son muy eficientes cuando se trata de texto manuscrito, ya que no es nada fácil aislar automáticamente los caracteres, especialmente en documentos históricos, pues muchos textos son de complicada lectura incluso para el ojo humano.

También se menciona que el reconocimiento de texto manuscrito o HTR (por sus siglas en inglés, *Handwritten Text Recognition*) se puede comparar a la tarea de reconocer el habla continua en un archivo poco nítido. Se comenta también que la tecnología actual para el reconocimiento de texto emplea técnicas del campo del reconocimiento automático del habla, utilizando para ello los modelos ocultos de Markov.

2.2. Estado del arte en visión por computador

El estado del arte en visión por computador se encuentra en el uso de redes neuronales convolucionales para el reconocimiento y clasificación de imágenes [2]. Esta técnica sin duda ha superado a técnicas anteriores en las que el preprocesado de las imágenes era más de la mitad del trabajo. Por ello se emplean actualmente en los concursos de clasificación de imágenes[1].

Con el uso de redes neuronales no es necesario utilizar muchas de las técnicas de preprocesado, las cuales se utilizaban anteriormente en reconocimiento de imágenes, como, por ejemplo, la detección de contornos basados en la primera derivada (conocido en inglés como *gradient masks*), para encontrar los puntos característicos, …

Ya no son necesarios todos esos pasos previos al utilizar redes neuronales; con un pequeño preproceso, en algunos casos ninguno, las imágenes se envían a la red y ella se encarga de su clasificación. Esto aporta una gran ventaja al no tener que realizar operaciones matemáticas previas al entrenamiento de la red, reduciendo la potencia de

1 https://www.cs.toronto.edu/~kriz/cifar.html

cálculo y aumentando la rapidez.

Uno de los artículos en el que se ha basado este proyecto es [2]. En él se hace referencia al uso de redes neuronales convolucionales con diferentes variaciones en cuanto a la profundidad de la red, topología y variación de filtros. Se parte de una topología de red básica y se varían los parámetros de profundidad. Se trata de una red convolucional profunda llamada *OxfordNet*, también conocida por las siglas VGG (por sus siglas en inglés, *Visual Geometry Group*), que se basa en el uso de pequeños filtros convolucionales de 3x3 que han demostrado una mayor rapidez y mejora. La idea es llegar a tener un gran número de capas de profundidad, pues la red original llega incluso hasta las 19 capas.

Estas diferentes topologías han sido utilizadas en la clasificación de imágenes, donde sus modelos se han dejado públicos para facilitar la investigación sobre el uso de redes convolucionales profundas usadas en visión por computador. Esto abre un mundo de posibilidades, permitiendo que las futuras investigaciones no tengan que partir desde cero, sino que pueden continuar a partir de una base sólida y obtener mejores resultados y mejores modelos.

Capítulo 3. Metodología

Para este proyecto se ha decidido realizar un sistema que reconozca los signos de puntuación utilizando para ello redes neuronales convolucionales. Esta técnica está funcionando muy bien en reconocer objetos en imágenes y es por ello por lo que se ha decidido emplearla en la detección de signos de puntuación.

En primer lugar se ha de crear la red neuronal, cuya estructura y funcionamiento se describe con más detalle en apartados posteriores. Se trata de una red que utiliza filtros convolucionales en cada capa de pesos y una estructura profunda, cuya topología de partida está basada en el modelo de red neuronal que aparece descrito en [2].

Con ello se pretende que el programa sea capaz de encontrar los signos que se encuentran a lo largo del texto y almacena su posición en un vector para un análisis posterior.

De esta forma, al conocer la posición donde se encuentra cada signo dentro de la página mediante sus coordenadas en píxeles, se puede realizar una segmentación por línea que comience en uno de estos signos y llegue hasta el siguiente. La combinación de este método junto con la segmentación por líneas puede ser una potente herramienta de análisis de textos manuscritos.

La red debe de pasar por diferentes fases hasta que se consigue un resultado final. Estas fases se describen en los siguientes apartados.

3.1. Fase de aprendizaje

El siguiente paso es el entrenamiento de la red. Para poder entrenar la red es necesario un corpus de entrenamiento que contenga una gran diversidad de signos de puntuación. El problema es que ese corpus no existe y se ha de crear desde cero, incrementando la dificultad, puesto que existe un gran número de variables detrás de una selección adecuada de los diferentes signos de puntuación, como la variedad en el tamaño de las ventanas que contengan el signo, en la posición en la que se encuentre el signo dentro de la imagen, luminosidad de la misma, …

Para ello se ha ido haciendo pruebas con un corpus de tamaño reducido, comparando los resultados obtenidos hasta encontrar un tamaño óptimo.

Una vez se obtiene el corpus, se procede al entrenamiento de la red, donde para ello se ha de realizar una serie de pruebas modificando los diferentes parámetros de la red, así como su estructura con el fin de minimizar el error y obtener un buen resultado.

Con ello ya estaría la red neuronal entrenada y lista para funcionar. Ya no sería necesario volver a entrenar la red, puesto que una vez entrenada ya se puede usar infinidad de veces para detectar y almacenar los signos de puntuación de las diferentes páginas escaneadas del corpus original.

3.2. Fase de extracción de características

Una vez se tiene entrenada la red neuronal, el siguiente paso es recorrer una página utilizando una ventana deslizante a lo largo de ella, con el fin de ir detectando en cada ventana si en su interior se encuentra un signo de puntuación o si se trata de información no relevante (o *background*).

La ventana comienza en la parte superior izquierda de la página y se va desplazando hacia la derecha hasta llegar al final, donde regresa otra vez a la parte de la izquierda, pero esta vez con un desplazamiento hacia abajo. Cada vez que la ventana se desplaza, extrae un recorte de la imagen (del inglés, *crop*) que se va almacenando en un vector junto a la posición, tanto la del eje x como la del eje y, de dicha imagen. De esta forma se puede saber la posición exacta que ocupa cada *crop* en la imagen original y, en el caso de ser positiva, se dibujará un cuadrado donde se encuentra el signo detectado.

Entre otros parámetros que se han de calcular, se encuentra el tamaño de la ventana y el número de píxeles de desplazamiento. El desplazamiento será como máximo de la mitad del tamaño de la ventana, ya que si se escoge más grande los signos de puntuación pueden quedar divididos en varias ventanas, llegando a ser imposible el reconocimiento. Sin embargo, cuanto menor sea el desplazamiento, más probabilidades hay de que un signo de puntuación quede totalmente recogido dentro de la ventana y más centrado puede aparecer, haciendo posible y más fácil su reconocimiento.

Se ha de tener en cuenta que si el desplazamiento es muy pequeño ralentiza el proceso y hay más probabilidades de obtener falsos positivos, ya que el número de *crops* aumenta el doble cada vez que se divide por dos el desplazamiento.

Tanto para el tamaño de la ventana como para el desplazamiento se han utilizado potencias de dos, pues la red trabaja mejor con potencias de dos, aumentando la velocidad de la red a la hora de calcular las convoluciones de los filtros. La solución es conseguir un equilibrio entre rapidez y eficacia, donde se ha de seleccionar un tamaño que sea suficiente para poder recoger todos los signos de puntuación que se encuentran en el documento, pero sin realizar un sobremuestreo.

3.3. Fase de test

Con la red entrenada se pasa a la fase de test, donde se recorre la página que se desea analizar para extraer los *crops* y almacenarlos en sus correspondientes vectores. De esos vectores se hace cargo el clasificador, que procesa las imágenes y determina a qué clase pertenece cada una de ellas.

La manera en la que se clasifican los *crops* es utilizando la función argmax(c|x), función que devuelve el mayor valor cuya probabilidad sea la más elevada, dándole la etiqueta de la clase a la que pertenece ese valor de entre las diferentes clases.

Cuando la función argmax(c|x) da como resultado la etiqueta a la que pertenece ese valor, bien se trate de un punto o la etiqueta de coma, se dibujará en el documento un recuadro blanco marcando dónde se ha encontrado esa etiqueta.

3.4. Medidas de calidad

Para obtener los valores de los resultados, y puesto que la medida de *accuracy* para este proyecto no es muy acertada, ya que no aporta información concluyente sobre los resultados obtenidos, se han de emplear diferentes tipos de medidas que aportan información más relevante. Dos de las medidas que se han empleado son la precisión y la exhaustividad.

Se trata de métricas empleadas para la medida del rendimiento de los sistemas de reconocimiento de patrones y búsqueda y recuperación de información.

Exhaustividad:
También llamada sensibilidad (o también conocida del inglés como, *recall*), es la fracción de instancias relevantes que han sido recuperadas. En el numerador se coloca el número de signos de puntuación que ha detectado correctamente el sistema, partido por el número total de signos de puntuación que hay en el documento.

Si el resultado se multiplica por 100 se obtiene el porcentaje de signos de puntuación que ha sido capaz de detectar el sistema, del total de signos de puntuación que contiene el documento.

$$Exhaustividad = \frac{Identificaciones\ correctas}{Total\ de\ signos\ de\ puntuación\ en\ el\ documento}$$

Precisión:
Fracción de instancias recuperadas que son relevantes. En el numerador se coloca el número de signos de puntuación que ha detectado correctamente el sistema, partido por el número total de identificaciones que ha realizado el sistema, tanto los aciertos como los errores.

Si el resultado se multiplica por 100 se obtiene, de entre todas las detecciones realizadas, que porcentaje de esas detecciones corresponden a signos de puntuación.

$$Precisión = \frac{Identificaciones\ correctas}{Total\ de\ identificaciones\ realizadas}$$

Estos dos sistemas de medida aportan información sobre cuántos elementos ha sido la red capaz de detectar y cuántos elementos ha fallado.

Otro parámetro que también se utiliza, y para el cual se utiliza los resultados de las medidas anteriores, es el llamado valor-F. Aporta una información global del sistema, realizando para ellos una media entre ambos valores.

Valor-F:
También denominado F-*score* o media-F, es la medida de precisión que tiene un test. Se emplea en la determinación de un valor único ponderado de la precisión y la exhaustividad.
El valor-F se considera como una media armónica que combina los valores de la precisión y de la exhaustividad.

$$Valor-F = (1 + \beta^2) \frac{Precisión \cdot Exhaustividad}{(\beta^2 \cdot Precisión) + Exhaustividad}$$

Sí → β = 1, se está dando la misma ponderación a la precisión y a la exhaustividad.

Sí → β > 1, se está dando más importancia a la exhaustividad que a la precisión.

Sí → β < 1, se está dando más importancia a la precisión que a la exhaustividad.

Para este proyecto se da la misma importancia tanto a la precisión como a la exhausitivdad, por lo tanto, la fórmula queda de la siguiente manera:

$$Valor-F = 2 \frac{Precisión \cdot Exhaustividad}{Precisión + Exhaustividad}$$

Capítulo 4. Corpus

La finalidad de este trabajo es la de reconocer signos de puntuación en texto manuscrito. Por lo tanto, es necesario un corpus con el que trabajar y poder realizar diferentes pruebas. Para ello se ha hecho uso de un corpus cedido que está compuesto por 53 páginas digitalizadas de un antiguo texto manuscrito que data del año 1853.

El título del manuscrito es el siguiente: “Noticia histórica de las fiestas con que Valencia celebró el siglo sexto de la venida á esta Capital de la milagrosa imagen del Salvador”.

El texto está escrito a pluma utilizando letra cursiva, donde las páginas tienen un característico color amarillento debido a la degradación del papel con el paso del tiempo. También la escritura del reverso se transparenta creando sombras en el anverso y dificultando la lectura.

Al estar escrito a pluma, el texto posee borrones de tinta y letras borrosas debido a la manipulación del papel antes de secarse correctamente. Otro problema de la pluma es que la intensidad de la tinta decae según se va escribiendo, hasta que termina y se vuelve a mojar en el tintero. Esto provoca que una palabra aparezca débil y acto seguido la siguiente palabra esté muy cargada de tinta, produciendo un contraste muy fuerte entre ambas palabras. Todo ello hace más costosa la detección de la escritura.

Para este proyecto, las imágenes que se necesitan son imágenes que contengan puntos, imágenes que contengan comas e imágenes que no contengan ninguna de las dos. Por lo tanto, no se puede trabajar directamente con el corpus original, puesto que son páginas completas. Por tanto, se ha de extraer de él imágenes de menor tamaño donde solo aparezca los signos de puntuación y algo de información extra de lo que les rodea.

La detección de los dos puntos y la detección del punto y coma está implícita en las anteriores, pues los dos puntos, al tener un punto debajo, lo detectará como un punto. Igual pasa con el punto y coma, al tener una coma debajo, la red la detectará con la etiqueta de coma.

4.1. Preprocesado de las imágenes

Las técnicas de preprocesado facilitan la tarea de reconocimiento. Por ello, se han realizado varias pruebas para determinar la técnica más adecuada para este proyecto.

Una de las pruebas que se han hecho ha sido binarizar las imágenes para que solo posean blanco y negro; para ello se han utilizado diferentes procedimientos de binarización.

El primer método empleado ha sido el de umbralización (en inglés, *thresholding*). Este método consiste en seleccionar un valor x de brillo (intensidad) que sea capaz de dividir la imagen en dos regiones cuyas intensidad son mayores y menores que el valor x.
Este método transforma una imagen de niveles de grises en una imagen binaria, tal como se muestra en las figuras 4.1.1 y 4.1.2:

Figura 4.1.1: Imagen donde se muestra un punto en escala de grises

Figura 4.1.2: Imagen donde se muestra un punto binarizado

En los ejemplos anteriores se puede observar que las imágenes contienen un punto, donde la primera (figura 4.1.1) se muestra en escala de grises y la segunda (figura 4.1.2) binarizada. La segunda imagen crea confusión, puesto que en ella se puede observar como, al binarizar, aparece un punto al lado del original. Este punto en la primera imagen no aparece, ya que forma parte de la letra, pero al binarizar se ha separado.

El problema de este método de binarización está en que, pese a que en la misma página la iluminación es muy parecida, no todas las páginas están igual de iluminadas. Otro problema es el citado anteriormente con respecto a la tinta de la pluma, pues no se escribe con la misma intensidad durante todo el texto. Es por ello por lo que se ha decidido probar otro método de binarización llamado umbral adaptativo (en inglés, *adaptive thresholding*).

Para el cálculo del umbral adaptativo se han empleado dos métodos diferentes, un método mediante el cálculo de la media y el otro mediante el cálculo de la Gaussiana.

La idea básica que estos métodos proporcionan es utilizar un enfoque alternativo para encontrar el umbral local, examinando estadísticamente los valores de intensidad del vecindario que rodea a cada píxel.

Se trata de obtener un valor umbral x en cada sección de la imagen, el cual divida la imagen en dos tonos de color, blanco y negro.

A continuación se muestran los diferentes resultados que se han obtenido binarizando una coma.

La figura 4.1.3 muestra la coma en escala de grises, tal y como se ha utilizado en este proyecto.

Figura 4.1.3: Imagen de una coma en escala de grises

Como se ha mencionado anteriormente, se ha realizado la binarización de la imagen anterior utilizando diferentes técnicas, mostrando a continuación los diferentes resultados que se han obteniendo.

Binarización estática:
En una binarización estática se utiliza un umbral fijo, el cual ha de utilizarse a lo largo de toda la página. Esto presenta una gran dificultad a la hora de hacer esa selección, pues si el valor del umbral es muy elevado, aparece ruido en la imagen que no pertenece a ninguna palabra, pero sí pertenece al fondo de la imagen y puede crear confusión, como se muestra en la figura 4.1.4, pues para que la coma aparezca completamente dibujada se ha de seleccionar un umbral tan elevado que muestra ruido del fondo, de ahí la unión con el borde o el punto que aparece al lado de la coma.

Figura 4.1.4: Imagen de una coma binarizada utilizando un umbral fijo grande

Sin embargo, para que no aparezca ese ruido en las imágenes se ha de seleccionar un umbral menos elevado, lo que provoca que los tonos de grises más débiles aparezcan como blanco, quitando todo el fondo pero también quitando parte de la información que se necesita. De esta manera lo que antes era una coma, ahora aparece como dos puntos separados, pues el centro de la coma, al ser de un gris más débil, ha desaparecido. También ha fraccionado la letra que acompañaba a la coma, haciendo que parezca dos puntos también. En la figura 4.1.5 se muestra el resultado de utilizar un umbral más bajo.

Figura 4.1.5: Imagen de una coma binarizada utilizando un umbral fijo pequeño

Para solucionar este problema se puede utilizar umbrales adaptativos, como el caso de umbrales adaptativos Gaussianos y de la media. Estos umbrales, a pesar de eliminar el ruido que se ha producido anteriormente y adaptarse según se va recorriendo la página, no llegan a generar buenos resultados, pues la propia naturaleza del texto, al estar escrita a pluma, no siempre consigue un trazo uniforme y con la misma intensidad, originando el mismo problema que se ha descrito anteriormente: tanto la coma como la parte de la letra que la acompaña se ve afectada.

Un ejemplo de binarización utilizando la media se muestra en la figura 4.1.6.

Figura 4.1.6: Imagen de una coma binarizada utilizando la media

Un ejemplo de binarización utilizando Gauss se muestra en la figura 4.1.7.

Figura 4.1.7: Imagen de una coma binarizada utilizando Gauss

Ambos resultados son muy similares y mejores que cuando se ha utilizado un umbral fijo. Sin embargo, la binarización no ha dado buenos resultados para este proyecto, por lo que se ha decidido utilizar las imágenes en escala de grises.

4.2. Selección del tamaño

Otra cuestión fruto de investigación es la selección del tamaño de las imágenes que formarán el corpus de entrenamiento. Las imágenes han de contener los signos de puntuación, pero también han de contener información extra que indique si se trata de un punto o una coma. Esto se debe a que el modelo ha de ser capaz de diferenciar un punto de final de frase del punto de una 'i', como también ha de ser capaz de distinguir entre una coma o un acento.

Al tratarse de letra cursiva, esta tarea se hace más complicada, ya que parte de las palabra anterior y parte de la palabra posterior invaden el espacio del signo de puntuación, haciendo más complicado aislarlo para su reconocimiento y creando confusión a la hora del reconocimiento.

La manera de diferenciar el punto o la coma como signo de puntuación es incluir información extra que indique en qué posición se encuentra. La imagen que los contenga ha de poseer parte de la palabra que le precede y, si no es punto y final, parte de la palabra que le sucede. De esta forma se puede diferenciar una coma o un punto de un acento o el punto de una 'i'.

Por ejemplo, si se trata de un punto de una 'i', éste estará situado en la parte superior de la palabra; por tanto, en la imagen en la que aparezca un punto perteneciente a una 'i' habrá parte de la palabra en la parte baja de la imagen.

El problema es que no es fácil obtener un punto o una coma en la que solo se encuentre texto a ambos lados, pues muchas veces aparece texto por encima o por debajo. Esto se debe a que las prolongaciones de las letras invaden el espacio de la palabra anterior o posterior. Con ello también invaden la parte superior o inferior del punto o la coma.

Puesto que el tamaño de las imágenes que forman el corpus es un parámetro importante, en este proyecto se ha realizado un estudio con dos tipos de tamaño distintos.

Por un lado, se ha seleccionado el tamaño de imagen más pequeño que sea capaz de contener el signo de puntuación y que recoja parte de las letras que acompañan al signo, con el fin de añadir información de la posición y de si se trata de un signo o de otra cosa.

Por otro lado, se ha seleccionado un tamaño de imagen, el tamaño más grande, que recoja la altura completa de la palabra que acompaña al signo de puntuación.

Los tamaños se seleccionan en potencia de dos, donde el primer tamaño mencionado son imágenes de 32x32 píxeles, mientras que las segundas son imágenes de 64x64 píxeles. A continuación se realizará un análisis más detallado de los diferentes corpus y sus imágenes.

4.2.1 Corpus con un tamaño de imagen pequeño

El primer corpus que se ha creado está formado por las imágenes de 32x32 píxeles. Contiene 500 imágenes de puntos, 500 imágenes de comas y 100000 imágenes que no pertenecen a ninguna de las clases anteriores, sino que pertenecen al fondo (en inglés, *background*). Esto es así puesto que la relación de comas y de puntos en el texto, en comparación con las imágenes que no lo son, es aproximadamente de 1-1-200, donde por cada punto y por cada coma hay 200 imágenes de fondo aproximadamente.

Las imágenes se han obtenido de las páginas comprendidas entre la cuatro y la veintinueve, ambas incluidas. A partir de la página veintinueve, las restantes han sido utilizadas para la parte de test.

En la figura 4.2.1.1 se muestra una coma centrada, donde se puede apreciar parte de la palabra anterior y posterior a ella. La coma se encuentra entre dos palabras, aunque ambas invaden la vertical de la coma tanto por arriba como por abajo.

Figura 4.2.1.1: Imagen de una coma centrada situada entre dos palabras

Sin embargo, no todas las comas se encuentran bien situadas, pues muchas de ellas se encuentran en lugares que pueden causar confusión. En la figura 4.2.1.2 aparece una coma que se encuentra en la parte inferior de la palabra, donde en la parte superior de la imagen se puede ver la terminación de la palabra a la que precede.

Figura 4.2.1.2: Imagen de una coma centrada situada debajo de la palabra que acompaña

En este caso, la coma se encuentra debajo de la palabra anterior. Esto se puede confundir con un acento que pertenezca a una palabra de la línea de abajo, e incluso con la prolongación de los rabillos de la palabra siguiente, que invade el espacio de la palabra anterior.

En la figura 4.2.1.3 se muestra un ejemplo de la prolongación del rabillo de la letra 'g' que pertenece a la siguiente palabra, pero que puede dar a confusión con una coma, tal como se muestra en la imagen.

Figura 4.2.1.3: Imagen de el rabillo de la letra 'g' que pertenece a la palabra siguiente

La única diferencia entre las dos imágenes que se han visto anteriormente es que, en la primera, la coma se encuentra aislada completamente de los bordes; sin embargo, en la segunda imagen el rabillo de la letra 'g' se ve cortado por el propio recorte de la ventana. Esa es la única manera que el ordenador tiene de distinguir una coma de cualquier otra cosa. A continuación, en la figura 4.2.1.4, se muestra un ejemplo de la imagen anterior y la siguiente, donde se aprecia mejor como la prolongación de la letra queda truncada.

Figura 4.2.1.4: Dos imágenes seguidas donde se muestra que el rabillo de la letra 'g' puede confundirse con una coma

La terminación de algunas palabras también puede ser causa de confusión, pues algunas tienen la misma forma y posición que una coma. Al igual que sucedía anteriormente, la manera de diferenciar una terminación una letra de una coma es que no parezca aislada de los bordes, pero no siempre es tan fácil, pues hay situaciones en que la intensidad de la tinta y el paso de los años hace que la línea sea muy apagada y delgada, llegando a dar un error de clasificación.

Figura 4.2.1.5: Imagen de la terminación de la letra 'a' que puede dar a confusión con una coma

En la figura 4.2.1.5 y la figura 4.2.1.6 se muestra cómo la terminación de la letra 'a', se puede confundir con una coma, pues tiene una curvatura y una posición muy parecida a algunas de ellas.

Figura 4.2.1.6: Dos imágenes seguidas donde se muestra la terminación de la letra 'a' que se prolonga hasta la siguiente palabra

Anteriormente se ha comentado que la forma de distinguir coma y la prolongación de una palabra es porque la coma se encuentra completamente aislada de los bordes del recorte. Sin embargo, ésto no sirve para el caso de los acentos, pues también se encuentran aislados.

Los acentos, al estar aislados igual que las comas, generan confusión a la hora de su identificación, pues no llegan a tocar el borde del recorte. La manera de diferenciar una coma de un acento es por la información extra que la rodea, pues para una coma lo normal es que se encuentre parte de la palabra que la acompaña por uno o ambos lados. En cambio, en el caso de los acentos lo normal es encontrar por debajo de ellos parte de la palabra a la que pertenecen, como se puede ver en la figura 4.2.1.7.

Figura 4.2.1.7: Imagen de un acento con información de la palabra a la que pertenece por debajo de él

También se pueden encontrar acentos pegados al rabillo de una letra que pertenece a una palabra que se encuentra en la línea superior, creando confusión, pues puede parecer una coma que acompaña a la palabra de la frase de arriba, en vez de un acento de la palabra de debajo como se puede ver en la figura 4.2.1.8.

Figura 4.2.1.8: Imagen de un acento que puede causar confusión con una coma que pertenezca a la fila superior

Otro factor de confusión se produce cuando el acento se encuentra separado de la palabra a la que pertenece; por tanto, la palabra no aparece en la imagen, pero esa palabra posee letras con terminaciones hacia arriba (como por ejemplo una 't' o una 'l') de forma que el acento aparece al lado de esa terminación, dando lugar a confusión con una coma, tal como se muestra en la figura 4.2.1.9.

Figura 4.2.1.9: Imagen de un acento que se confunde con una coma

En la figura 4.2.1.10 se muestra un conjunto de imágenes que rodean el acento mostrado anteriormente en la figura 4.2.1.9, donde se puede ver que pertenece a la letra 'a'.

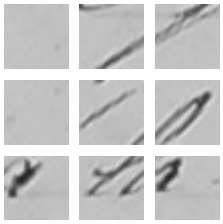


Figura 4.2.1.10: Conjunto de imágenes que rodean el acento que acompaña a la letra 'a'

No todas las comas del texto son iguales, pues hay una gran variedad de tamaños y formas que difieren. También existe diferencia en la cantidad de tinta que posee cada una, pues existen comas fuertemente marcadas con trazo grueso y existen comas con muy poca tinta y un trazo muy fino, lo que dificulta la detección de las mismas.

En la figura 4.2.1.11 se muestra un conjunto de imágenes con diferentes tipos de comas que se pueden encontrar en el texto manuscrito.

Figura 4.2.1.11: Conjunto de imágenes donde se muestra la variedad de comas que aparecen en el texto

Al igual que con las comas, con los puntos ocurre lo mismo; muchas veces se encuentra un punto con parte por arriba y parte por debajo de las palabras que lo rodean, creando confusión sobre si se trata de un punto perteneciente a una 'i' o de un punto y seguido. En la figura 4.2.1.12 se puede ver un ejemplo de un punto.

Figura 4.2.1.12: Imagen de un punto y seguido centrado entre dos palabras

Al tener por debajo del punto parte de la palabra siguiente se hace más complicado diferenciar qué clase de punto se trata. A continuación, en la figura 4.2.1.13, se muestra la imagen de un punto que pertenece a una 'i', que se parece a la imagen vista en la figura 4.2.1.12 y que da confusión con un punto y coma.

Figura 4.2.1.13: Imagen de un punto de una 'i'

Sin embargo se trata de un punto de una 'i', como se muestra en la figura 4.2.1.14.

Figura 4.2.1.14: Dos imágenes que muestran el punto de una 'i', que se confundo con un punto y coma

Otra confusión que aparece bastante en el texto es la marca de la pluma al finalizar o iniciar una palabra, ya que se produce una acumulación de tinta que da origen a puntos indeseados, que se confunden con puntos finales o puntos y seguidos.

El trazo que continua el punto en ocasiones es tan fino y débil que no se aprecia y da lugar a confusiones. En la figura 4.2.1.15 se muestra una 'o' que va acompañada de una 's' final, donde el trazo de terminación de la 's' se ha quedado marcado con un punto, dando confusión a un punto y final que acompaña a la 'o'.

Figura 4.2.1.15: Imagen del inicio de la letra 's' que se confunde con un punto y final

Sin embargo, en la figura 4.2.1.16 se aprecia cómo el punto pertenece a la letra 's' en lugar de tratarse de un punto y final.

Figura 4.2.1.16: Dos imágenes donde se muestra que no es un punto y final, sino que es un punto de la propia letra 's'

Como se ha comentado anteriormente, el punto de la 'i' muchas veces da lugar a confusión. Esto ocurre también cuando, delante del punto de la 'i', se encuentra una letra cuya terminación sobresale por arriba, quedando a la misma altura que el punto. Sí la ventana se desliza sin llegar a mostrar la parte de la palabra a la que pertenece, da lugar a confusión, pues parece el punto y final de una frase.

En la figura 4.2.1.17 se muestra un punto que pertenece a una 'i' pero da lugar a confusión porque va acompañado de la parte de arriba de una letra de la palabra a la que pertenece la 'i'.

Figura 4.2.1.17: Imagen donde se muestra un punto de una 'i'

En la figura 4.2.1.18 se muestra el grupo de imágenes al que pertenece el punto visto en la figura 4.2.1.17, donde se muestra parte de la palabra que contiene la 'i' junto con el punto.

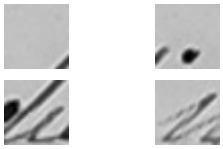


Figura 4.2.1.18: Imágenes donde se muestra el punto de la 'i' y la palabra a la que pertenece ese punto

Al igual que sucede con las comas, existe una gran variedad de puntos. Estos varían tanto en el tamaño como en la cantidad de tinta que poseen. A ello se le ha de sumar que no todos los puntos son iguales, pues existen puntos que se han alargado un poco y parecen una coma, normalmente porque la tinta no se secó a tiempo y se emborronó al manipular del texto.

En la figura 4.2.1.19 se muestra un conjunto de imágenes con diferentes tipos de puntos que se pueden encontrar en el texto manuscrito.

Figura 4.2.1.19: Conjunto de imágenes donde se muestra la variedad de puntos que hay en el texto

La suma de todo lo descrito anteriormente hace que reconocer texto manuscrito *off-line* sea una tarea muy complicada. Es por ello que en la actualidad sigue siendo fruto de investigación.

Una vez se han visto las peculiaridades que poseen las imágenes de tamaño pequeño, se procede en el apartado 4.2.2 a explicar las características que poseen las imágenes de mayor tamaño.

4.2.2 Corpus con un tamaño de imagen grande

El segundo corpus que se ha creado esta formado por las imágenes de 64x64 píxeles. Al igual que el anterior, contiene 500 imágenes de puntos, 500 imágenes de comas y 100000 imágenes de *background*.

A diferencia del corpus anterior, al ser las imágenes más grandes contienen mucha más información extra del entorno que rodea al signo de puntuación, eliminando así mucha de la problemática descrita anteriormente, sobre todo a la hora de confundir las comas con los acentos y confundir los puntos con un punto de una 'i'. Sin embargo, al ser las imágenes de mayor tamaño, la red necesita de un mayor tiempo de aprendizaje y el signo de puntuación ya no se encuentra resaltado, puesto que ya no ocupa casi toda la imagen, sino que ahora ocupa entre un 15 y un 20% de ella.

A continuación, en la figura 4.2.2.1, se muestra una imagen que contiene una coma, donde ésta se encuentra centrada en la imagen. Se puede observar que la cantidad de información que aporta esta imagen es muchísimo más que las imágenes del corpus anterior.

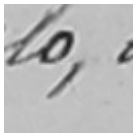


Figura 4.2.2.1: Imagen de una coma centrada situada entre dos palabras

Se distingue claramente la coma del resto de letras que la acompañan. También se puede ver cómo aparecen las letras que acompañan a la coma prácticamente completas. Esto da una ventaja, como se verá más adelante.

La terminación de ciertas palabras, que suponía un problema en el corpus anterior, ahora se aprecia claramente que no se trata de una coma, como muestran la figura 4.2.2.2 y la figura 4.2.2.3.

Figura 4.2.2.2: Imagen de la terminación de la letra de la palabra anterior

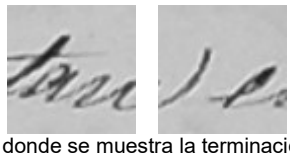


Figura 4.2.2.3: Dos imágenes seguidas donde se muestra la terminación de la letra 'n' y el inicio de la siguiente

Al igual que las terminaciones, los acentos, que anteriormente daban lugar a confusión ahora presentan más información sobre la palabra a la que pertenecen y la posición en la que se encuentran, pues en la imagen aparece la letra a la que pertenece el acento completo como muestra la figura 4.2.2.4.

Figura 4.2.2.4: Imagen de un acento

En la figura 4.2.2.5 se muestra un conjunto de imágenes con diferentes tipos de comas que se pueden encontrar en el texto manuscrito.

Figura 4.2.2.5: Conjunto de imágenes donde se muestra la variedad de comas que aparecen en el texto

Al igual que con las comas, con los puntos pasa algo parecido; la confusión que se creaba con el punto de la 'i' ya se distingue claramente que no se trata de un punto y seguido o punto y final, pues debajo de él aparece la palabra a la que pertenece como se muestra en la figura 4.2.2.6.

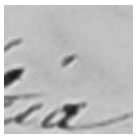


Figura 4.2.2.6: Imagen de un punto de una 'i'

Si se compara la figura 4.2.2.6 con la de un punto y seguido, (figura 4.2.2.7), se puede ver claramente la diferencia entren ambas imágenes.

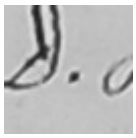


Figura 4.2.2.7: Imagen de un punto y seguido

En la figura 4.2.2.8 se muestran imágenes de la variedad de puntos que pueden aparecer en el corpus, de diferentes tamaños y formas.

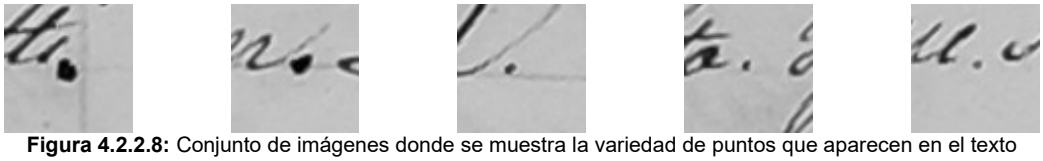


Figura 4.2.2.8: Conjunto de imágenes donde se muestra la variedad de puntos que aparecen en el texto

Capítulo 5. Red neuronal convolucional

Para la creación de la red neuronal se ha utilizado la librería de TensorFlow[2]. Se trata de una librería de cálculo numérico de flujo de datos. Fue desarrollada por Google para satisfacer las necesidades de construir y entrenar redes neuronales para detectar y descifrar patrones y correlaciones.

Por encima de TensorFlow corre una librería llamada Keras[3], de código abierto escrita en Python, que fue diseñada para la experimentación rápida de redes neuronales profundas. Con estas herramientas se ha desarrollado la red neuronal utilizada en este proyecto y que parte del artículo [2].

5.1. Funciones que componen la red neuronal

Se trata de una red neuronal convolucional, cuyo funcionamiento es realizar la convolución de los píxeles de una capa con un filtro (o *kernel*), para obtener los píxeles de la capa siguiente. El problema de la convolución es que se obtiene una imagen menor, ya que los bordes no llegan a convolucionarse.

Para evitar este problema se ha de realizar la función de *padding*, que consiste en rellenar los bordes con ceros para aumentar el tamaño y así poder realizar la convolución con los bordes originales. De esta manera se consigue que la imagen que resulta de la convolución tenga el mismo tamaño que la original. Los ceros que se añaden solo se usan a la hora de hacer la convolución. Al terminar la convolución los ceros desaparecen, volviendo al tamaño original.

A continuación, en la figura 5.1.1, se muestra un ejemplo de la convolución de una ventana de pixeles de 5x5 con un filtro de 3x3, dando como resultado una ventana de tamaño 3x3.

2 https://www.tensorflow.org
3 https://keras.io/

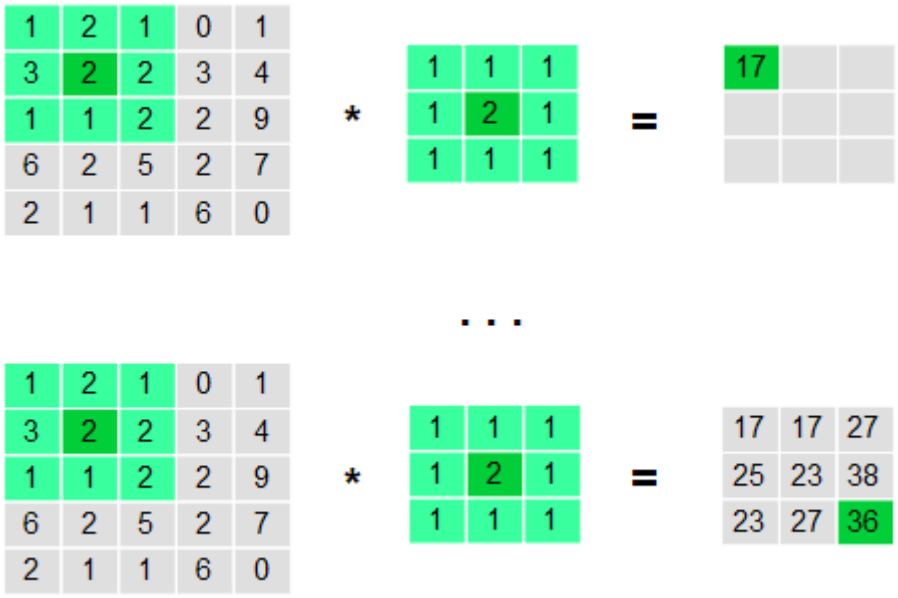


Figura 5.1.1: Ejemplo de una convolución sin *padding*

En este proyecto, y para esta red, el tamaño del *kernel* que se ha utilizado ha sido de 3x3, donde las imágenes de entrada, en lugar de ser de 5x5 como en el ejemplo anterior, son de 32x32 y de 64x64, puesto que son los tamaños que tienen las imágenes de los dos corpus empleados en este proyecto.

La función de *padding* que se ha utilizado es la de “SAME”[4], que indica que la salida tiene las mismas dimensiones que la entrada, donde para ello se rellena los bordes con ceros.

Continuando con el ejemplo anterior, al realizar *padding* antes de la convolución, se obtiene como resultado una ventana del mismo tamaño que la entrada, como se muestra en la figura 5.1.2.

4 https://keras.io/activations

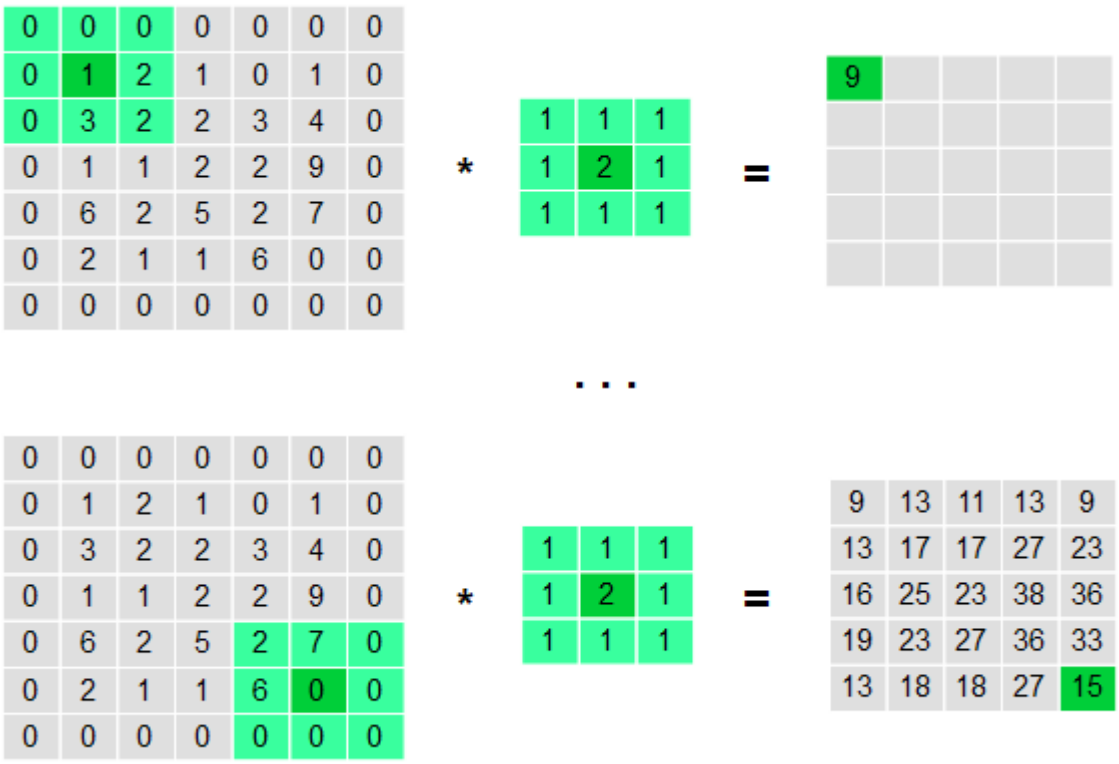


Figura 5.1.2: Ejemplo de una convolución en la que se ha realizado previamente *padding* de ceros

El *padding* que se ha utilizado en el ejemplo anterior es el mismo que se ha utilizado en la red neuronal; se trata de añadir un marco de ceros a la imagen original antes de realizar la convolución, para que el resultado sea del mismo tamaño que la imagen original.

Otra de las funciones que posee la red neuronal es la activación ReLU (por sus siglas en inglés, *Rectified Linear Unit*). Es una de las funciones de activación más utilizadas en las redes neuronales profundas.

Se trata de una función de activación no lineal muy sencilla. Cuando la entrada es positiva, la función es creciente; sin embargo, cuando la entrada es negativa, la función toma el valor cero. A continuación, en la figura 5.1.3, se muestra una imagen donde se representa la función ReLU (en color azul) y una variante a la que se le ha aplicado un suavizado (en color verde).

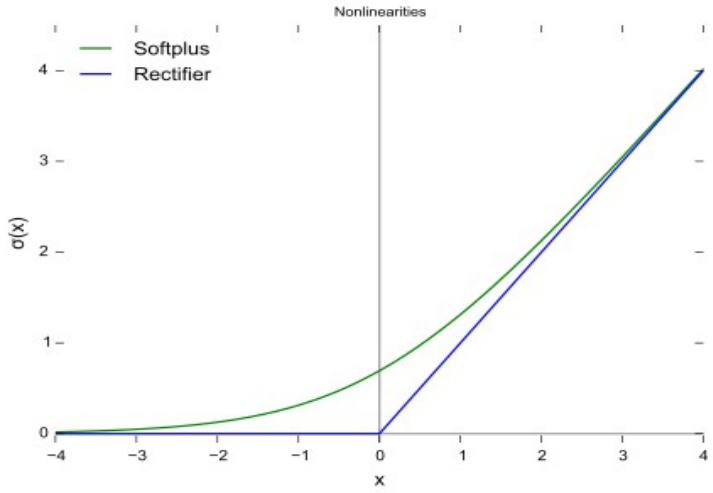


Figura 5.1.3: Gráfica donde se muestra una función de activación ReLU (línea azul) y una variante con suavizado (línea verde)[5]

Otra de las funciones que incorpora la red es la de reducción de tamaño de los datos, utilizando la función *max-pooling*. Para ello se coloca sobre la ventana un filtro de un tamaño determinado y, de entre todos los píxeles que quedan dentro del filtro, se escoge el de mayor tamaño.

Esta función lo que hace es reducir el tamaño a la mitad, ya que el tamaño del *max-pooling* que se ha utilizado es de 2x2 píxeles. Esto significa que, de cada grupo de 2x2 píxeles, se selecciona el de valor más elevado. De cuatro píxeles obtenemos un único valor, en este caso el mayor.

El ejemplo de la figura 5.1.4 muestra con más detalle el funcionamiento de la función *max-pooling,* donde gracias a esta función se consigue reducir la resolución de la ventana y obtener un resumen de las características de la región por la que va pasando el filtro. Esto se hace para ir simplificando en cada capa el tamaño de las imágenes sin perder las características principales. Es muy útil, pues la red utilizada va incrementando el número de pesos por capa a la vez que va decrementando el tamaño de las imágenes que van entrando a esas capas, hasta conseguir que a la última capa entren ventanas de 1x1 píxel.

5 https://en.wikipedia.org/wiki/Rectifier_(neural_networks)

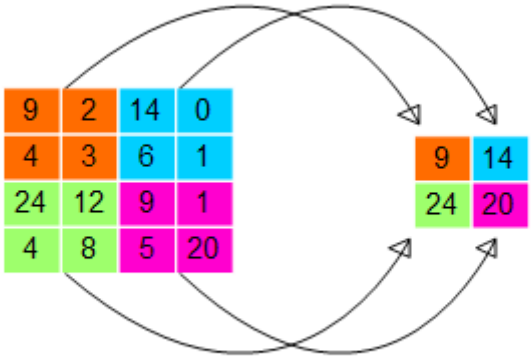


Figura 5.1.4: Ejemplo de *max-pooling* de 2x2 utilizado en la red neuronal

Tras aplicar la función *max-pooling* en el ejemplo anterior, el tamaño de la ventana pasa de ser de 4x4 píxeles a 2x2 píxeles, quedándose con los valores de mayor tamaño.

Estas funciones que se han comentado son las que componen la red neuronal convolucional utilizada en este proyecto, variando la estructura de la red, la composición de cada capa y el entrenamiento realizado, pero no se han añadido más funciones que las descritas. La disposición de las mismas y el entrenamiento se comentan más adelante.

En el apartado 5.2 se detalla la red completa con el número de pesos en cada capa y las capas que se han utilizado.

5.2. Estructura de la red neuronal

La red neuronal que se ha utilizado está compuesta por 8 capas de pesos, donde en cada capa se ha ido incrementado la profundidad en un factor de potencia de dos. La entrada a la red es una imagen en escala de grises (0 – 255) de un tamaño fijo de 32x32 píxeles.

Esa imagen pasa por un conjunto de capas convolucionales, donde se utilizan unos filtros muy pequeños de 3x3 píxeles vistos en los ejemplos del apartado anterior. Se utilizan estos filtros ya que es el valor más pequeño capaz de capturar la noción de izquierda, derecha, arriba, abajo y centro. Como se ha mencionado, antes de realizar cada convolución se realiza la función de *padding,* manteniendo el tamaño de la salida igual al original, conservando así la resolución espacial.

La agrupación espacial de cada imagen la realizan las capas de *max-pooling,* en este caso de 2x2 píxeles, que se encuentran después de cada cambio de una capa con menor número de pesos hacia una capa de mayor número de pesos. Con ello se consigue dividir el tamaño de la imagen a la mitad, obteniendo las características más destacadas de cada imagen. El total de capas en la red de *max-pooling* es de 5, puesto que se incrementa en potencia de dos el número de pesos por capa, excepto el último *max-pooling* que pasa de una capa de 512 pesos a otra de 512 pesos.

Por último se encuentran las capas que van todas conectadas (del inglés, *Fully-Connected*), donde las dos primeras capas tienen un total de 4096 canales y la última contiene 500 canales para cada clase. En la capa final se aplica una función de *Soft-Max*, siendo ésta la función suavizada de la función ReLU.

Tabla 5.2.1: Esquema de las capas que componen la red neuronal utilizada en el proyecto

Capas de la Red	Tamaño de la imagen para *crops* de 32x32	Tamaño de la imagen para *crops* de 64x64	Profundidad
Input	32x32	64x64	1
Convolucional 2D	32x32	64x64	64
Max-pooling 2D	16x16	32x32	64
Convolucional 2D	16x16	32x32	128
Max-pooling 2D	8x8	16x16	128
Convolucional 2D	8x8	16x16	256
Convolucional 2D	8x8	16x16	256
Max-pooling	4x4	8x8	256
Convolucional 2D	4x4	8x8	512
Convolucional 2D	4x4	8x8	512
Max-pooling	2x2	4x4	512
Convolucional 2D	2x2	4x4	512
Convolucional 2D	2x2	4x4	512
Max-pooling	1x1	2x2	512
Flatten	1x1	1x1	512 para 32x32 2048 para 64x64
Fully-Connected	1x1	1x1	4096
Fully-Connected	1x1	1x1	4096
Fully-Connected	1x1	1x1	500
Output (Soft-Max)	1x1	1x1	3

En la tabla 5.2.1 se muestra la descomposición de la red en sus diferentes capas, donde cada capa contiene el tamaño de ventana que se utiliza y la profundidad. La configuración anterior ha sido fruto de una serie de pruebas que se han ido realizando con diferentes topologías, diferentes valores de profundidad y de los filtros.

Se ha partido de la red base a la cual se le han ido haciendo modificaciones. Una de las modificaciones que se ha tenido en cuenta ha sido la de añadir ruido aleatorio Gaussiano a las imágenes de entrada.

5.3. Entrenamiento de la red neuronal

Una vez se ha creado la red neuronal el siguiente paso es entrenarla. Para ello se han de ajustar los diferentes parámetros, de forma que se vaya incrementando la precisión y minimizando el error.

La tasa de aprendizaje ha sido uno de los parámetros que se ha ido ajustando durante el entrenamiento, pues primero se ha entrenado con una tasa de aprendizaje alta y se ha ido disminuyendo, dividiendo el valor por 10 cada modificación, hasta que la red converge.

Valor de inicio:

$$Lr = 0{,}00001$$

Valor final:

$$Lr = 0{,}00000001$$

La frecuencia con la que se ha modificado el valor ha sido cada 20 *epochs,* hasta llegar a un total de 80 *epochs* de entrenamiento. La red ha sido entrenada en bloques de 10 *epochs* para hacer el proceso más sencillo y dinámico ya que, por sus dimensiones, el proceso de entrenamiento es muy costoso en cuanto a coste computacional.

Otro parámetro importante a tener en cuenta es el tamaño del *batch*, pues es un parámetro peculiar, ya que cuanto más pequeño sea el valor, más precisión a la hora del entrenamiento, a costa de incrementar el tiempo de entrenamiento. En este caso, el tamaño del *batch* ha sido el mismo durante todo el entrenamiento de la red.

$$Batch = 512$$

El tercer parámetro que es fruto de estudio y de pruebas durante el entrenamiento es el que se encarga de regular el valor con el que decae los pesos (en inglés, *decay*), pues en un principio la formula para el cálculo de este valor era la siguiente:

$$Decay = \frac{Learning\ Rate}{N^o\ Epochs}$$

Aplicando esta formula se ha obtenido un valor de partida con el que se han hecho diferentes pruebas y modificaciones del mismo, hasta que se ha obtenido un valor con el que trabajar.

$$Decay = 1 \cdot e^{-6}$$

Capítulo 6. Resultados obtenidos

Una vez finaliza el entrenamiento de la red neuronal, el siguiente paso es realizar el test. Para ello, el programa que contiene la red entrenada carga una página del corpus, la cual se recorre y se almacenan las imágenes obtenidas con las que se hace el test.

Una vez se ha cargado la imagen y se ha realizado el test, el programa dibuja un cuadrado, de color blanco y del tamaño de la ventana que se ha utilizado, en la posición donde considere la red neuronal que ahí se encuentra un signo de puntuación. Estas ventanas quedan dibujadas en la página y su posición almacenada en un vector, tanto la posición del eje x como la posición del eje y, con el fin de futuros usos a la hora de fragmentar el texto.

Para poder comparar, se muestran los resultados de varias páginas, donde cada página posee un número diferente de signos de puntuación.

Antes de cada imagen se muestran las fórmulas con los valores de las medidas de calidad sobre cada página.

6.1. Resultados utilizando imágenes pequeñas

Los siguientes resultados se han realizado utilizando el corpus que contiene las imágenes pequeñas, cuyo tamaño es de 32x32 píxeles.

Estos resultados se han obtenido entrenando la red utilizando la topología anteriormente descrita y con las siguientes etapas:

- Los 20 primero *epochs* se han entrenado con un *Learning-Rate* = 0,00001
- Los 30 siguientes *epochs* se han entrenado con un *Learning-Rate* = 0,000001
- Y para finalizar el entrenamiento, los 10 últimos *epochs* se han entrenado con un *Leraning-Rate* = 0,0000001

Tras estos pasos, se llega a un punto en el que la red converge y los resultados ya no mejoran. Para mostrar los resultados se ha realizado el test con las páginas que no han sido empleadas en la composición del corpus de entrenamiento.

Los resultados obtenidos tras realizar el test a las páginas comprendidas entre la treinta y la cuarenta y nueve, ambas incluidas, utilizando una ventana deslizante de 32x32 píxeles, donde el deslizamiento entre una ventana y la siguiente es de 8 píxeles, se representan en la tabla 6.1.1

Tabla 6.1.1: Resultados obtenidos al realizar el test a las páginas comprendidas entre la 30 y 49

Medida	**Valor**
Total de identificaciones realizadas	557
Identificaciones correctas	335
Identificaciones incorrectas	222
Nº de signos de puntuación sin detectar	268
Total de signos de puntuación en los documentos	603
Exhaustividad	0,55
Precisión	0,6
Valor-F	0,57

Las medidas de los resultados obtenidos, siendo éstos el total de la suma de los valores obtenidos en cada página, son:

$$Exhaustividad = \frac{Identificaciones\ correctas}{Total\ de\ signos\ de\ puntuación\ en\ los\ documentos} = \frac{335}{603} = 0{,}55$$

$$Precisión = \frac{Identificaciones\ correctas}{Total\ de\ identificaciones\ realizadas} = \frac{335}{557} = 0{,}6$$

$$Valor-F = 2\ \frac{Precisión \cdot Exhaustividad}{Precisión + Exhaustividad} = 2\ \frac{0{,}6 \cdot 0{,}55}{0{,}6 + 0{,}55} = 0{,}57$$

A continuación, en la figura 6.1.2 y figura 6.1.3, se muestran ejemplos de los resultados obtenidos en diferentes páginas.

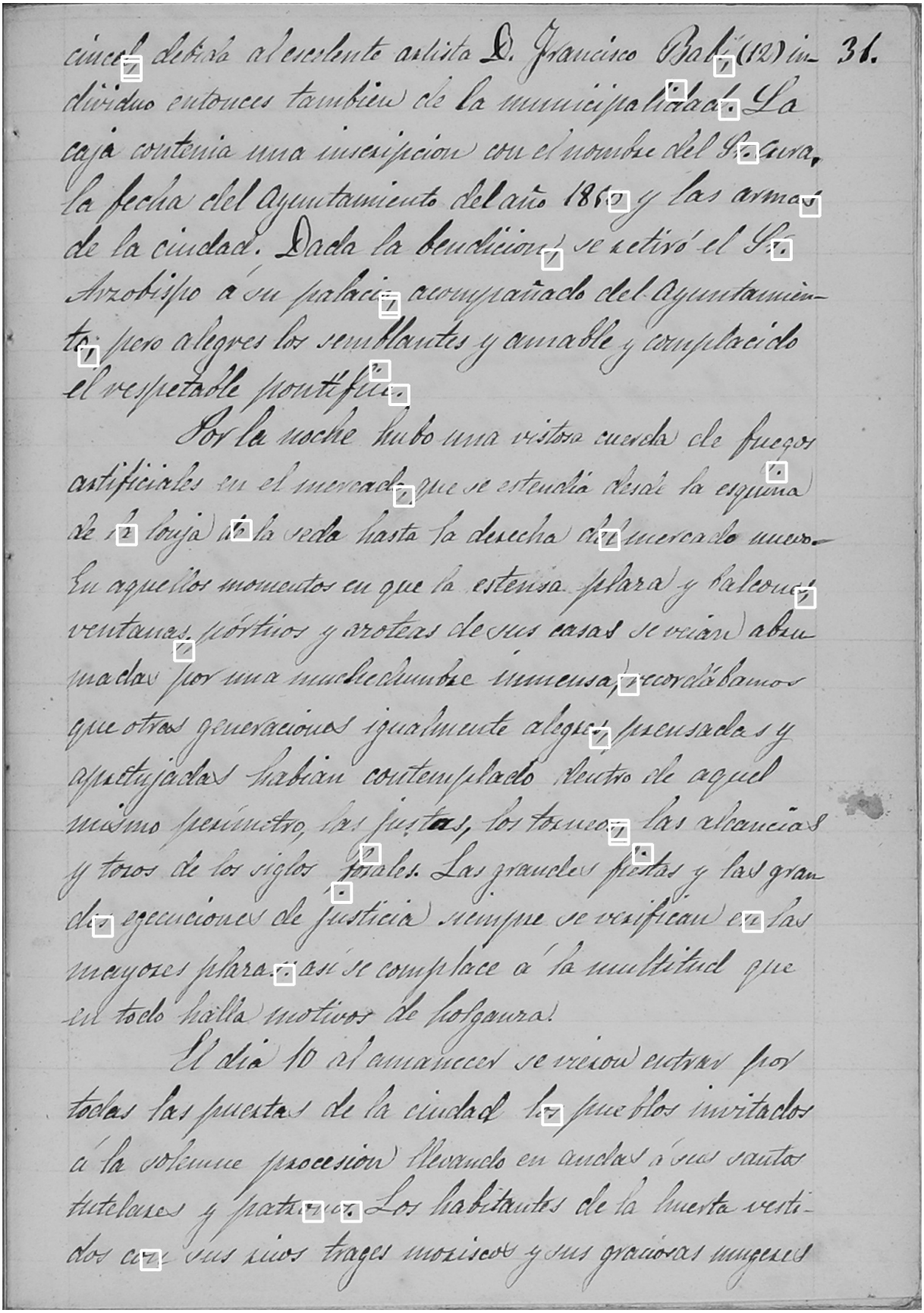
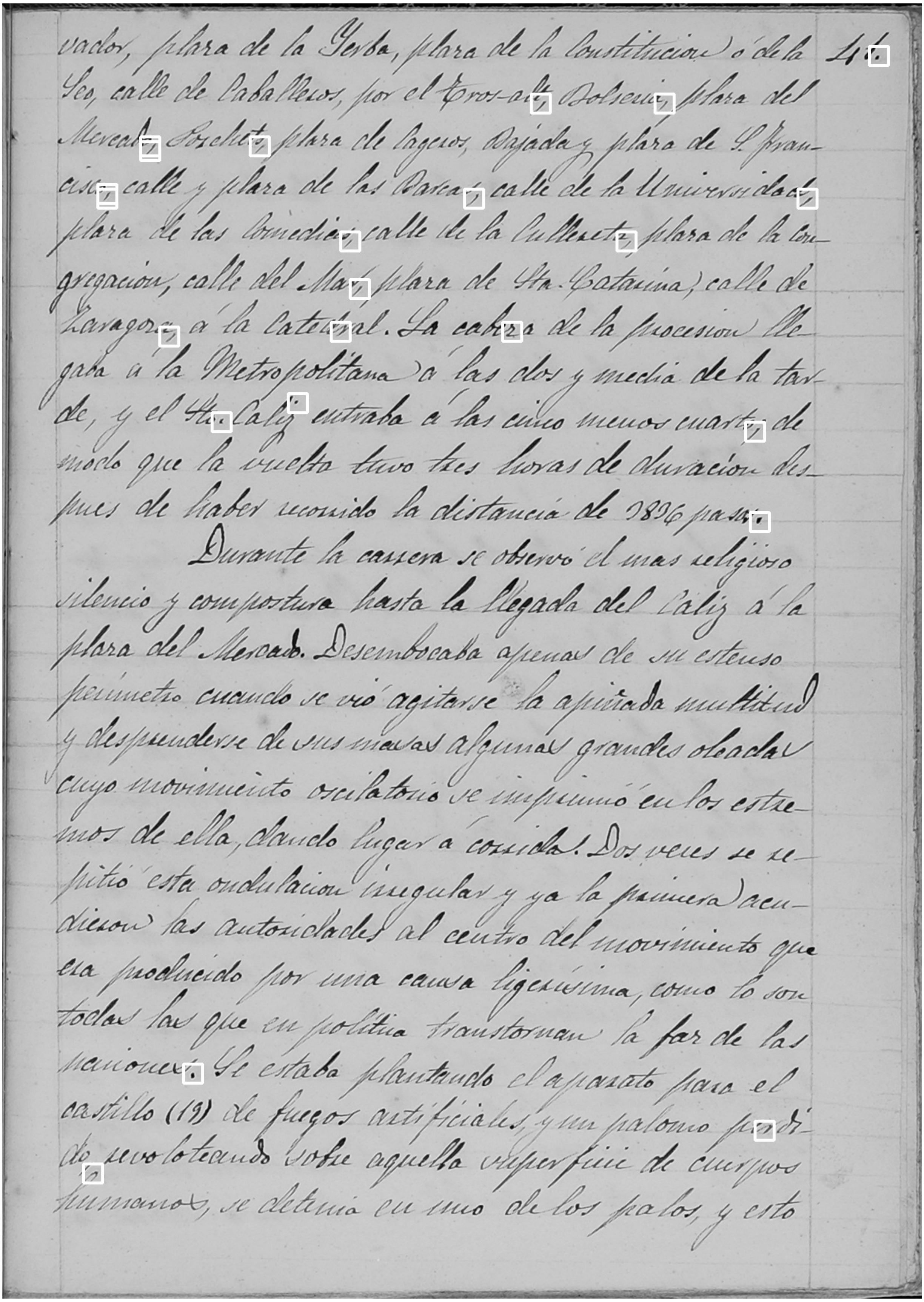


Figura 6.1.2: Imagen de la página 34 del corpus usando *crops* de 32x32 píxeles



vador, plaza de la Yerba, plaza de la Constitucion ó de la 44
Seo, calle de Caballeros, por el Tros-alt, Bolsería, plaza del
Mercado, Bosch, plaza de Lagueros, Bajada y plaza de S. Fran-
cisco, calle y plaza de las Barcas, calle de la Universidad,
plaza de las Comedias, calle de la Culleretas, plaza de la Con-
gregacion, calle del Mar, plaza de Sta. Catalina, calle de
Zaragoza, á la Catedral. La cabeza de la procesion lle-
gaba á la Metropolitana á las dos y media de la tar-
de, y el Sto. Cáliz entraba á las cinco menos cuarto, de
modo que la vuelta tuvo tres horas de duracion des-
pues de haber recorrido la distancia de 2896 pasos.

Durante la carrera se observó el mas religioso
silencio y compostura hasta la llegada del Cáliz á la
plaza del Mercado. Desembocaba apenas de su estenso
perímetro cuando se vió agitarse la apiñada multitud
y desprenderse de sus masas algunas grandes oleadas
cuyo movimiento oscilatorio se imprimió en los estre-
mos de ella, dando lugar á corridas. Dos veces se re-
pitió esta ondulacion irregular y ya la primera acu-
dieron las autoridades al centro del movimiento que
era producido por una causa ligerísima, como lo son
todas las que en política transtornan la paz de las
naciones. Se estaba plantando el aparato para el
castillo (19) de fuegos artificiales, y un palomo perdi-
do revoloteando sobre aquella superficie de cuerpos
humanos, se detenia en uno de los palos, y esto

Figura 6.1.3: Imagen de la pagina 44 del corpus usando *crops* de 32x32 píxeles

6.2. Resultados utilizando imágenes grandes

Los siguientes resultados se han realizado utilizando el corpus que contiene las imágenes grandes, cuyo tamaño es de 64x64 píxeles.

Estos resultados se han obtenido entrenando la red utilizando la topología anteriormente descrita, la misma topología que se ha empleado con las imágenes pequeñas.

Las etapas de entrenamiento son las siguientes:

- Los 10 primeros *epochs* se han entrenado con un *Learning-Rate* = 0,00001
- Y para finalizar el entrenamiento, los 10 últimos *epochs* se han entrenado con un *Learning-Rate* = 0,000001

Tras estos pasos, se llega a un punto en el que la red converge y los resultados ya no mejoran. Para mostrar los resultados se ha realizado el test a diferentes páginas y analizado los resultados obtenidos.

Los resultados obtenidos tras realizar el test a las páginas comprendidas entre la treinta y la cuarenta y nueve, ambas incluidas, utilizando una ventana deslizante de 64x64 píxeles, donde el deslizamiento entre una ventana y la siguiente es de 16 píxeles, son los mostrados en la tabla 6.1.2

Tabla 6.2.1: Resultados obtenidos al realizar el test a las páginas comprendidas entre la 30 y 49

Medida	**Valor**
Total de identificaciones realizadas	428
Identificaciones correctas	280
Identificaciones incorrectas	148
Nº de signos de puntuación sin detectar	323
Total de signos de puntuación en los documentos	603
Exhaustividad	0,46
Precisión	0,65
Valor-F	0,53

Las medidas de los resultados obtenidos, siendo éstos el total de la suma de los valores obtenidos en cada página, son:

$$Exhaustividad = \frac{Identificaciones\ correctas}{Total\ de\ signos\ de\ puntuación\ en\ los\ documentos} = \frac{280}{603} = 0{,}46$$

$$Precisión = \frac{Identificaciones\ correctas}{Total\ de\ identificaciones\ realizadas} = \frac{280}{428} = 0{,}65$$

$$Valor-F = 2\ \frac{Precisión \cdot Exhaustividad}{Precisión + Exhaustividad} = 2\ \frac{0{,}46 \cdot 0{,}65}{0{,}46 + 0{,}65} = 0{,}53$$

A continuación, en la figura 6.2.2 y figura 6.2.3, se muestran ejemplos de los resultados obtenidos en diferentes páginas.

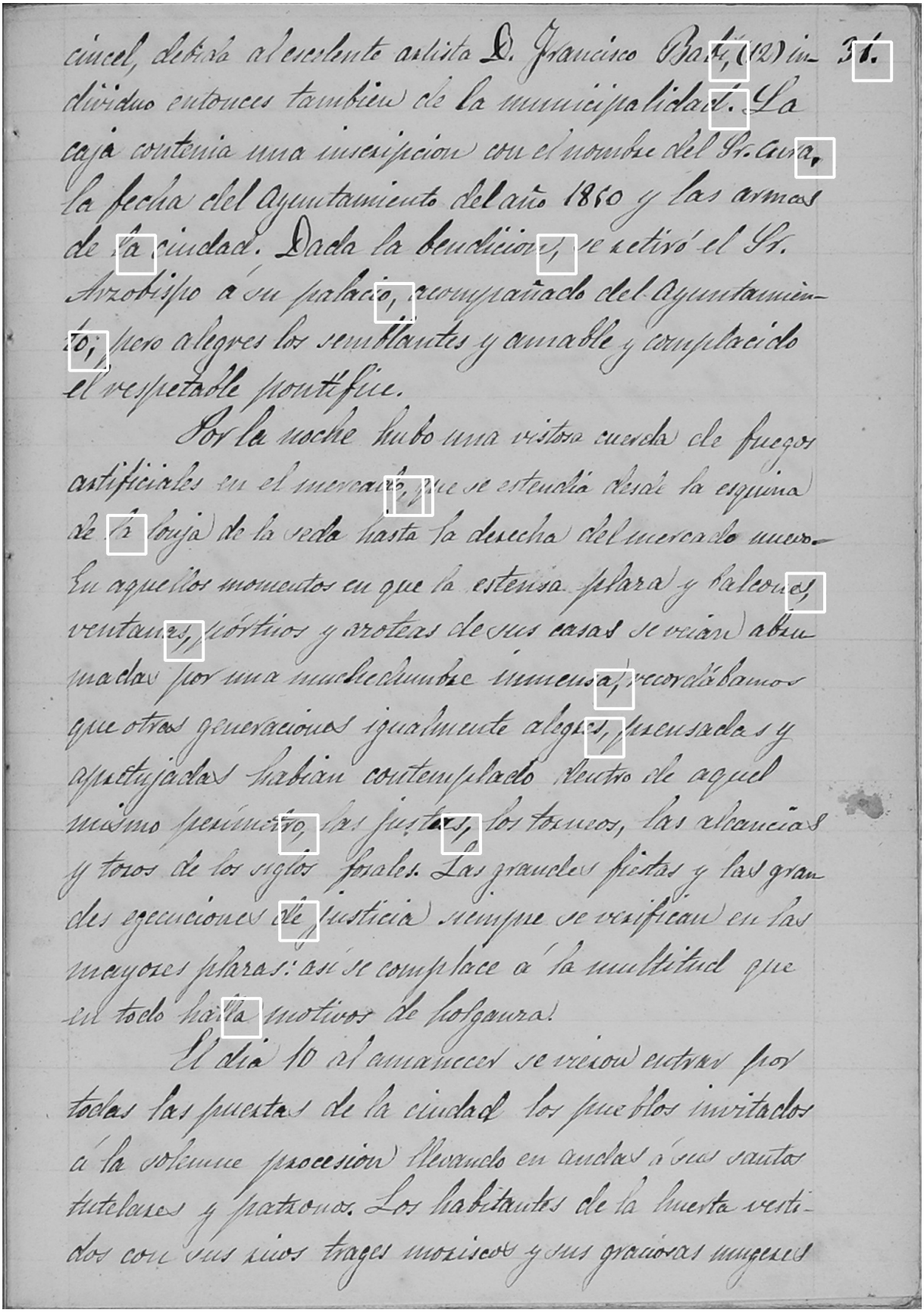


Figura 6.2.2: Imagen de la pagina 34 del corpus usando *crops* de 64x64 píxeles

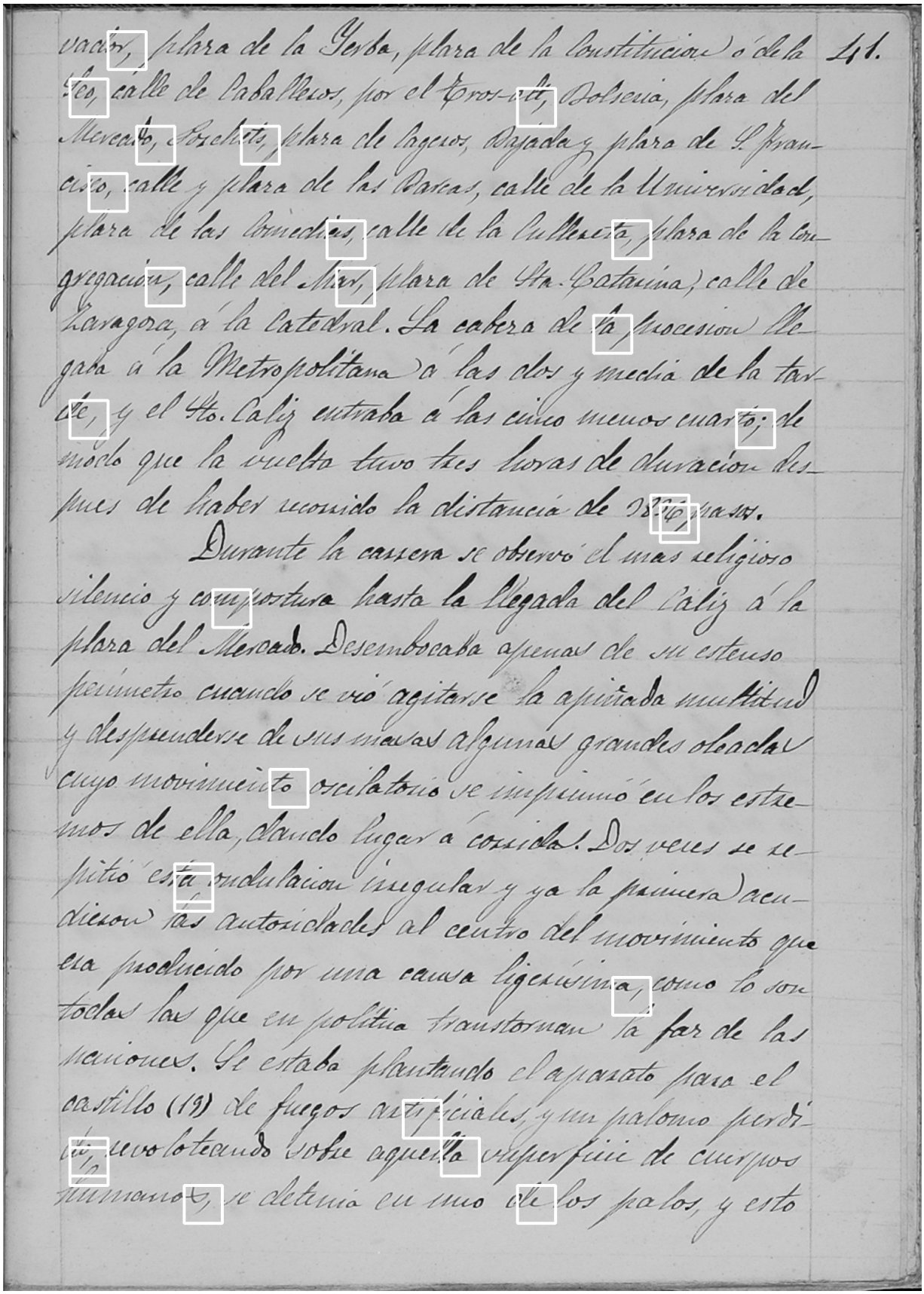


Figura 6.2.3: Imagen de la página 44 del corpus usando *crops* de 64x64 píxeles

6.3. Comparativa de los resultados

Este apartado hace una comparativa de los resultados que se han obtenido en este proyecto utilizando los diferentes tamaños del *crops*. Estos se muestran en la tabla 6.3.1

Tabla 6.3.1: Comparativa de resultados por tamaño de ventana

Técnica	Tamaño de Ventana	Exhaustividad (%)	Precisión (%)	Valor-F (%)
CNN	32x32	55	60	57
CNN	64x64	46	65	53

Cuando se utilizan *crops* de 32x32 se obtiene un mayor número de identificaciones en el texto que cuando se utiliza *crops* de 64x64. Sin embargo, con éstos últimos se obtiene una mayor precisión.

Pese a que teóricamente se obtendría un mejor resultado al utilizar los tamaños de *crops* de 64x64, no ha sido así.

Uno de los posibles motivos por los cuales no se han obtenido mejores resultados es que al aumentar el tamaño de la imagen donde aparecen los signos de puntuación, éstos no destaca en la imagen, sino que ocupan una parte reducida, disminuyendo su importancia.

Otra de los posibles motivos es que al trabajar con imágenes de mayor tamaño, se incrementa el número de pesos que contiene la red, haciendo necesario una mayor cantidad imágenes de entrenamiento para entrenar y hacer converger dichos pesos.

Capítulo 7. Conclusiones

Las conclusiones a las que se ha llegado, tras la realización del proyecto, son las que se comentan continuación.

Una de las primeras conclusiones a las que se ha llegado es sobre la importancia de tener un buen corpus. Uno de los problemas ha sido el tamaño del corpus, pues para un buen funcionamiento es necesario una gran cantidad de imágenes que hay que obtener y clasificar a mano. Esto ha sido un trabajo muy laborioso y de muchas horas de dedicación. Horas de trabajo que, una vez realizadas, se pueden utilizar en futuras investigaciones.

Otra conclusión importante que se ha obtenido ha sido la dificultad de trabajar con textos manuscritos, pues hay muchos factores externos al proyecto que dependen del estado de los documentos y que hacen muy costoso el trabajo de reconocimiento. Sin embargo, y como se ha visto aquí, con el uso de redes neuronales mucho del trabajo de preprocesado de imágenes que antes componía la mitad del trabajo, ahora queda simplificado y prácticamente no hay que realizar nada.

Sin embargo, el problema del uso de redes neuronales es que no existe una fórmula concreta que indique cuántas capas hay que utilizar, qué funciones componen cada capa, o la cantidad de neuronas que debe tener cada capa, así como diversos parámetros más que componen la red. Por ello, se ha de realizar una gran cantidad de pruebas hasta conseguir buenos resultados. Para agilizar este proceso, lo ideal es partir de una topología y una configuración de los parámetros base; de ahí ya se hacen pequeñas modificaciones hasta conseguir minimizar el error.

De todo ello se puede decir que las redes neuronales son potentes herramientas, las cuales se pueden utilizar en infinidad de aplicaciones y complicados proyectos en los cuáles otras técnicas serías muy costosas o insuficientes.

Como trabajos futuros se puede aumentar el tamaño del corpus incorporando mayor variedad de puntos y de comas. También se puede probar otro tipo diferentes de redes neuronales que no sean convolucionales.

También se puede utilizar la red para encontrar algún tipo de palabra concreta o de letra que se desee, pues solo habría que entrenarla con un corpus adecuado.

Bibliografía

[1] V. Romero, N. Serrano, A. H. Toselli, J. A. Sánchez & E. Vidal. 2011. Handwritten Text Recognition for Historical Documents ITI, Universitat Politècnica de València, Spain

[2] K. Simonyan & A. Zisserman. 2014. Very Deep Convolutional Networks for Large Scale Image Recognition. Visual Geometry Group, Department of Engineering Science, University of Oxford. Published as a conference paper at ICLR 2015.