Pontificia Universidad Católica del Perú

Escuela de Posgrado

"Segmentación automática de textos, mediante redes neuronales convolucionales en imágenes documentos históricos"

Tesis para optar el Grado Académico de

MAGÍSTER EN INFORMÁTICA CON MENCIÓN EN CIENCIAS DE LA COMPUTACIÓN

Autor

Franco Javier Ascarza Mendoza

Asesor

Dr. César Armando Beltrán Castañón

Lima – Perú, 2018

Resumen

Los manuscritos históricos contienen valiosa información, en los últimos años se han realizado esfuerzos para digitalizar dicha información y ponerla al alcance de la comunidad científica y público en general a través de imágenes en bibliotecas virtuales y repositorios digitales. Sin embargo, existen documentos y manuscritos históricos escritos en un lenguaje extinto en la actualidad y una cantidad limitada de profesionales expertos en la interpretación y análisis de dichos documentos.

Las imágenes de los documentos y manuscritos históricos poseen características particulares producto precisamente de su antigüedad como por ejemplo: La degradación del papel, el desvanecimiento de la tinta, la variabilidad en iluminación y textura, entre otros.

Tareas como recuperación de información o traducción automática de imágenes de manuscritos históricos requieren una etapa de pre-procesamiento importante debido a las características mencionadas en el párrafo anterior. Entre las tareas de pre-procesamiento se puede mencionar la binarización y la segmentación de la imagen en regiones de interés.

La presente tesis se enfoca en el procedimiento de segmentación en regiones de interés de las imágenes de manuscritos históricos. Existen métodos para segmentar imágenes de documentos históricos basados fundamentalmente en la extracción manual de características con previo conocimiento del dominio. La finalidad de la presente tesis es desarrollar un modelo general que automáti-

camente aprenda a extraer características de los píxeles de las imágenes de los documentos históricos y utilizar dichas características para clasificar los píxeles en las clases que previamente se definirán.

Abstract

Historical handwritten documents contain valuable information, in recent years efforts have been made to digitize this information and make it available to the scientific community and the general public through images in virtual libraries and digital repositories. However, there are historical documents and manuscripts written in an extinct language today and a limited number of professional experts in the interpretation and analysis of such documents.

The historical handwritten document images have particular features due to their age, such as: The degradation of paper, the fading of the ink, the variability in lighting and texture, among others.

Tasks such as information retrieval or automatic translation of handwritten historical images require an important stage of preprocessing due to the features mentioned in the previous paragraph. Among the preprocessing tasks we can mention image binarization and image segmentation in regions of interest.

This thesis focuses on the procedure of segmentation in regions of interest of historical manuscript images. There are methods to segment historical document images based primarily on handcraft feature extraction with prior knowledge of the domain. The purpose of this thesis is to develop a general model that automatically learns to extract features from the images of historical documents and use these features to classify the pixels in the predefined labels.

Dedicatoria

A mi hermano Diego

Agradecimientos

A mi familia porque siempre creyeron en mi.
A mi asesor Dr. César Beltrán por su apoyo y motivación.

Índice general

Índice de tablas

Índice de figuras

CAPÍTULO 1

GENERALIDADES

1.1. Introducción

Existe una gran cantidad de documentos manuscritos históricos en librerías y museos. Muchos de estos manuscritos incluyen datos históricos importantes que necesitan preservarse debido a la constante degradación de la que son objeto con el paso del tiempo. Poner en valor dichos documentos históricos significa no solamente expandir el contenido de dichos documentos en imágenes subidas a la web, sino también el procesamiento automático para su análisis.

La segmentación de páginas en imágenes de documentos históricos es un procedimiento de pre-procesamiento importante para el análisis de éste tipo de imágenes y su entendimiento. La finalidad es segmentar la imagen del documento histórico en regiones de interés. A comparación de la segmentación de imágenes de documentos escritos a máquina, la segmentación de imágenes de documentos históricos escritos a mano involucra considerar detalles propios de éste tipo de escritos como: Variación en estructura de diseño, decoración en los estilos de escritura, degradación de la tinta, deterioro del papel, variación en la iluminación y la probable existencia de manchas propias de la escritura de la época.

Algunos métodos de segmentación de páginas en imágenes de documentos

históricos han sido desarrollados recientemente. Éstos métodos se basan en características extraídas manualmente [1], [2], [3], [4], o conocimiento previo [5], [6], [7], [8], o modelos que combinan características extraídas manualmente con previo conocimiento del dominio [9], [10]. La finalidad de la presente tesis es desarrollar un método general que automáticamente aprenda a extraer características de los píxeles de las imágenes de documentos históricos. Elementos como trazos de palabras, palabras en oraciones, palabras en párrafos tienen una estructura jerárquica de bajo a alto nivel siempre que estos patrones se repitan en diferentes partes de los documentos. Basado en éstas propiedades, un algoritmo de aprendizaje de características puede ser aplicado para aprender a etiquetar los diferentes segmentos que componen las imágenes.

Las redes neuronales convolucionales (CNN por sus siglas en inglés) tienen una arquitectura y entrenamiento similar a la del Perceptrón Multicapa con la diferencia de que las CNN poseen capas convolucionales y de agrupamiento. Las capas convolucionales tienen la particularidad de aprender de la coherencia espacial local de las entradas al asumir que las entradas espacialmente cercanas están correlacionadas al compartir pesos entre neuronas de la misma capa y reforzar patrones de conectividad local entre neuronas de capas adyacentes, las capas de agrupamiento permiten reducir el número de parámetros conservando suficiente información [11], con múltiples capas convolucionales y capas de agrupamiento, las CNN han alcanzado resultados muy buenos en varios campos como: Reconocimiento de manuscritos [12], clasificación [13], reconocimiento de textos en imágenes naturales [14] y clasificación de sentencias [15].

El enfoque básico de un algoritmo de segmentación es asignar una etiqueta a cada píxel de la imagen. la clasificación semántica Píxel a píxel ha sido ampliamente usado (Ver por ejemplo [16] ó [17]). En la presente tesis se considera el problema de segmentación como el etiquetado de un grupo de píxeles al que denominaremos parches. Dichos parches de la imagen se generarán a partir de la subdivisión de la imagen usando un algoritmo de clustering denominado algoritmo simple de agrupamiento iterativo lineal (SLIC por sus siglas en inglés), el cual divide a la imagen en regiones donde cada región se identificará como un super píxel [18], los parches que se procesarán vendrán a ser el cuadrilátero central de dicho super píxel. La motivación de emplear super píxeles en el presente trabajo es reducir la complejidad computacional sin degradar la calidad de la segmentación, con el propósito de incrementar la velocidad del procesamiento, en lugar de etiquetar cada píxel de la imagen, se etiquetará cada super píxel de la imagen. En la presente tesis se propone una red neuronal convolucional profunda que realizará tanto la tarea de extracción de características y clasificación. Los parches encontrados a partir de los super píxeles producidos

con el algoritmo SLIC se usarán como entradas para entrenar la red neuronal convolucional, la etiqueta que se asignará a cada parche será la misma que la etiqueta de su píxel central. Se contará con un conjunto de imágenes de documentos históricos con píxeles etiquetados para el proceso de entrenamiento. Durante el entrenamiento, las caracterísicas usadas para predecir las etiquetas de las piezas de las imágenes son aprendidas en las capas de convolución de la red neuronal.

1.2. Problemática

Los documentos y manuscritos históricos son fuentes primarias que contienen valiosa información de diversa índole como por ejemplo: El relato de los primeros cronistas españoles que llegaron al Tahuantinsuyo sobre el imperio Incaico, los primeros descubrimientos médicos, piezas literarias, documentos contables, glosas, etc. En años recientes se han realizado esfuerzos para poner a disposición de la comunidad científica y público en general dicha información de primera mano utilizando para ello la Web en forma de bibliotecas virtuales y repositorios digitales, y publicando el contenido generalmente en forma de imágenes, por ejemplo en [19] se describen las bases de datos Parzival y St. Gall, los cuales consisten de imágenes de manuscritos escritos con tinta sobre pergaminos, las imágenes están a colores, la base de datos Parzival data del siglo XIII y la base de datos St. Gall data del siglo IX y en [20] encontramos otras base de datos denominadas CB55 que data del siglo XIV y CSG18, CSG863 que datan del siglo XI.

Las imágenes de los documentos de manuscritos históricos poseen características particulares que aumentan la complejidad de su análisis y procesamiento, entre dichas caracterísicas podemos mencionar las siguientes: La degradación del papel, el desvanecimiento de la tinta, la variedad de estilos de escritura influenciada por la época en que fueron escritos, la variabilidad en iluminación y textura a lo largo de la imagen, entre otros.

Tareas como recuperación de información de imágenes de documentos históricos [21] o reconocimiento automático de caracteres de manuscritos históricos [22] requieren una etapa de pre-procesamiento importante debido a las características particulares de éste tipo de documentos mencionadas en el párrafo anterior. Entre las tareas de pre-procesamiento se puede mencionar la binarización que consiste en separar los píxeles que conforman el texto del resto de píxeles de la imagen, la segmentación de la imagen en regiones de interés que consiste en etiquetar a los píxeles para ver a qué región predefinida pertenece,

por ejemplo se puede requerir segmentar la imagen en 4 clases: fondo, texto, decoraciones y comentarios.

La presente tesis se enfoca en el procedimiento de segmentación en regiones de interés de imágenes de manuscritos históricos. Algunos métodos de segmentación de páginas en imágenes de documentos históricos han sido desarrollados recientemente. Éstos métodos se basan en características extraídas manualmente [1], [2], [3], [4] o conocimiento previo [5], [6], [7], [8] o modelos que combinan características extraídas manualmente con previo conocimiento del dominio [9], [10]. La finalidad de la presente tesis es desarrollar un método general que automáticamente aprenda a extraer características de los píxeles de las imágenes de los documentos históricos y utilizar dichas características para clasificar los píxeles en las clases predefinidas.

1.3. Objetivos

1.3.1. Objetivo General

Desarrollar un modelo basado en una red neuronal convolucional profunda para segmentación de imágenes de documentos históricos.

1.3.2. Objetivos Específicos

Los objetivos específicos son:

- Pre-procesar y construir los ejemplares de entrenamiento.
- Definir las métricas para medir la calidad de la segmentación.
- Implementar la red neuronal convolucional profunda para segmentación de imágenes de documentos históricos.
- Evaluar el desempeño de los algoritmos a nivel de calidad de segmentación.

1.4. Alcance

Se desarrollará durante la presente tesis el diseño, construcción, evaluación y optimización de un modelo computacional basado en una red neuronal convolucional profunda para segmentar páginas de imágenes de documentos históricos. Para ello se tiene el siguiente alcance:

- El trabajo busca explorar y aplicar la técnica de las redes neuronales convolucionales profundas para segmentar imágenes de documentos históricos en regiones de interés, se define en éste caso al proceso de segmentación como el proceso de etiquetado de los píxeles de las imágenes de documentos históricos. Las etiquetas estarán predefinidas y se contará con un conjunto de entrenamiento con sus respectivas etiquetas o clases. La arquitectura de red neuronal convolucional que se usa en la presente tesis es el modelo Resnet [23].

Por otro lado mencionamos a continuación las limitaciones en el desarrollo de la presente tesis:

- Se trabajará sobre la base de datos Parzival el cual está descritos en [19].
- Se asumen cuatro clases o etiquetas para el proceso de segmentación de las imágenes en regiones de interés los cuales son: Página, texto, decoración y comentarios.

CAPÍTULO 2

ESTADO DEL ARTE

Se revisó en la literatura, investigaciones relacionadas a la segmentación de imágenes de documentos manuscritos históricos, para ellos se realizaron búsquedas sobre las bases de datos: "IEEExplorer" y "ACM Digital Library", en ambos casos se realizó la búsqueda según las palabras claves de los autores, las palabras clave que se utilizaron para realizar las búsquedas son las siguientes:

- Historical
- Document
- Segmentation

Las investigaciones que se encontraron se pueden dividir en dos grandes grupos:

- Métodos basados en reglas
- Métodos basados en aprendizaje de máquina

2.1. Métodos basados en reglas

- Van Phan et al. [6] Propone una técnica de segmentación de caracteres de documentos manuscritos históricos, se trabajó sobre imágenes de documentos vietnamitas de los siglos X a XX. Se propuso un método basado

en análisis de componentes conectados para extracción de caracteres de imágenes de documentos históricos, se utilizaron diagramas de área de Voronoi para representar la vecindad y los límites de los componentes conectados. Basado en ésta representación, cada caracter se consideró como un grupo extraído de las regiones adyacentes de Voronoi.

- Panichkriangkrai et al. [7] Propone un sistema de extracción de líneas de texto y caracteres de imágenes de documentos de textos japoneses grabados en madera. Las líneas de texto son separadas usando proyecciones verticales en imágenes binarizadas. Para la extracción de los caracteres kanji, una integración basada en reglas es aplicada para fusionar o dividir los componentes conectados.

- Gatos et al. [8] Propone una segmentación de zonas y líneas de texto para imágenes de documentos manuscritos históricos. Basado en el previo conocimiento de la estructura del documento, las zonas de texto verticales son detectadas analizando las líneas de reglas verticales y las líneas verticales blancas del documento. Sobre las zonas detectadas, un método de segmentación de líneas de texto basado en transformaciones de Hough es usada para segmentar líneas de texto,

Todos los métodos descritos líneas arriba han alcanzado buenos resultados de segmentación sobre bases de datos de documentos específicos. Sin embargo, la limitación que acarran éstas técnicas radican en que una serie de reglas y características tienen que ser cuidadosamente definidas. En otras palabras, éstos métodos no se pueden aplicar directamente a otro tipo de imágenes de documentos históricos.

2.2. Métodos basados en aprendizaje de máquina

- Wick et al. [24] Propone una técnica de segmentación basada en la arquitectura U-Net [25], el cual procesa una página entera en un solo paso, tiene la habilidad de aprender directamente de los píxeles de la imagen.

- Chen et al. [26] Propone una red neuronal convolucional simple para segmentar imágenes de páginas de documentos históricos, las entradas de dicha red son parches extraídos de los super píxeles generados en cada imagen.

- Bukhari et al. [2] Propone un método de segmentación para imágenes de documentos históricos arábigos. Se considera la altura normalizada

de las imágenes, área en primer plano (foreground area), la distancia relativa, la orientación e información del vecindario de los componentes conectados como características del modelo. Luego éstas características son usadas para entrenar una red neuronal perceptrón multicapa (MLP por sus siglas en inglés) para clasificar dichos componentes conectados en clases relevantes de texto.

- Cohen et al. [9] Propone un modelo donde se aplican laplacianos y gaussianos en la escala múltiple de la imagen binarizada para extraer los componentes conectados. Basado en conocimiento previo, se eligen los valores umbrales apropiados para eliminar ruido de los componentes conectados. Con un método de minimización de energía y las características como el tamaño del cuadrilátero delimitador, el área, el ancho del trazo y la distancia de las l´ıneas de texto estimadas, cada componente conectado se etiqueta como texto o no texto.

- Asi et al. [10] Propone un método de segmentación en dos pasos para imágenes de documentos históricos arábigos. se extrae el área de texto principal con filtros de Gabor. Luego la segmentación es refinada a través de la minimización de una función de energía.

Como se puede ver, comparado con los métodos basados en reglas, la ventaja de los métodos basados en técnicas de aprendizaje de máquina requieren menos conocimiento previo del dominio. Sin embargo los métodos existentes basados en técnicas de aprendizaje de máquina excepto la técnica basada en redes neuronales convolucionales, se basan en una ingeniería de extracción de caracter´ısticas hechas a mano lo cual demanda tiempo y esfuerzo.

En la tabla 2.1 se muestran todas las investigaciones revisadas donde ”BR” significa ”Basado en reglas” y ”AM” significa ”Basado en aprendiza- je de máquina”.

Tabla 2.1: Investigaciones relacionadas a segmentación de imágenes de documentos históricos

BD	**Título**	Autor	Año	Cit.	Tipo
ACM DL	Development of Nom character segmentation for collecting patterns from historical document pages	Truyen Van Phan, Bilan Zhu, Masaki Nakagawa	2011	2	BR

Continúa en la siguiente página

Tabla 2.1 – Continuado de la página anterior

BD	Título	Autor	Año	Cit.	Tipo
ACM DL	Char. segmentation and retrieval for learning support system of Japanese historical books	C. Panichkriangkrai L. Li K. Hachimura	2013	4	BR
IEEExplorer	Segmentation of Historical Handwritten Documents into Text Zones and Text Lines	B. Gatos G. Louloudis N. Stamatopoulos	2014	6	BR
IEEExplorer	Fully Convolutional Neural Networks for Page Segmentation of Historical Document Images	C. Wick F. Puppe	2018	1	AM
IEEExplorer	Convolutional Neural Networks for Page Segmentation of Historical Document Images	K. Chen M. Seuret J. Hennebert	2017	2	AM
IEEExplorer	Layout Analysis for Arabic Historical Document Images Using Machine Learning	S. Bukhari T. Breuel A. Asi	2012	13	AM
ACM DL	Robust text and drawing segmentation algorithm for historical documents	R. Cohen A. Asi K. Kedem J. El-Sana I. Dinstein	2012	5	AM
IEEExplorer	A Coarse-to-Fine Approach for Layout Analysis of Ancient Manuscripts	A. Asi R. Cohen K. Kedem	2014	15	AM

CAPÍTULO 3

MARCO CONCEPTUAL

A continuación se hará una revisión de la parte teórica relacionada al tema de la presente tesis.

3.1. Segmentación de imágenes

La segmentacion subdivide una imagen en sus regiones u objetos constituyentes, el nivel de detalle de la subdivisión depende del problema que se pretende resolver, es decir, la segmentacion debe detenerse cuando los objetos o regiones de interés en una aplicación han sido detectados. Por ejemplo en la inspección automática de ensamblajes electrónicos interesa analizar las imagenes de los productos con el objetivo de detectar la presencia o ausencia de anomalías específicas, como por ejemplo componentes faltantes o alguna conexion defectuosa entre las piezas, no tiene sentido llevar ésta segmentacion más all del nivel de detalle requerido para identificar esos elementos. La segmentacion no trivial de imagenes es una de las tareas más difíciles en el procesamiento de imagenes, la precisión de la segmentación determina el eventual éxito o fracaso del proceso de analisis computarizado [27].

3.2. Técnicas de segmentación de imágenes

Según Chauhan et al. [28] las técnicas de segmentación de imágenes se dividen en 4 grandes grupos:

- Métodos de detección de bordes.
- Métodos basados en umbrales.
- Segmentacion orientada a regiones
- Métodos basados en clustering

3.2.1. Métodos de detección de bordes

Los métodos de detección de bordes son mayormente usados para detectar discontinuidades en imagenes. Es una forma popular de detectar p´ıxeles de borde y enlazarlos para crear bordes en la imagen. Los p´ıxeles de borde son los p´ıxeles en los cuales existe una transicion en valores de intensidad. Bordes son un conjunto de p´ıxeles de borde conectados y representan los l´ımites entre dos regiones que tienen distintos niveles de intensidad [27]. Entre éste tipo de métodos de detección de bordes tenemos:

- Detección de bordes basado en gradientes: Un gradiente es un vector bidimensional que apunta a la dirección en la que la intensidad de la imagen crece más rápidamente. Las dos funciones que pueden ser expresadas en términos de las derivadas direccionales son la magnitud de la gradiente y la orientación de la gradiente. La magnitud de la gradiente está definido por [28]:

$$g(x, y) = (\Delta x^2 + \Delta y^2)^{1/2} \tag{1}$$

Donde:

$$\Delta x = f(x + n, y) - f(x - n, y) \tag{2}$$

y

$$\Delta y = f(x, y + n) - f(x, y - n) \tag{3}$$

Este valor nos da la maxima tasa de incremento de f(x,y)por unidad de distancia en la orientacion del gradiente de g(x,y). La orientación del gradiente es también un valor importante, la orientación del gradiente está dado por:

$$\Theta(x, y) = atan(\Delta y / \Delta x) \tag{4}$$

Aquí el ángulo es medido con respecto al eje x. La dirección del borde en (x,y) es perpendicular a la dirección del vector gradiente en ese punto [29].

- Detección de bordes basado en laplacianos: Los métodos basados en laplacianos buscan cruces cero en la segunda derivada de la imagen para encontrar bordes, usualmente los cruces de cero del Laplaciano o los cruces de cero de una expresión diferencial no lineal [29].

3.2.2. Métodos basados en umbrales

Es el método mas sencillo para segmentar una imagen en regiones. El valor de intensidad de cada p´ıxel es comparado con un valor de umbral dado. Dependiendo del resultado de la comparacion entre dicho p´ıxel y el umbral, el p´ıxel es etiquetado. Éste método es utilizado cuando la imagen está conformada de objetos claros y fondo oscuro [13]. Si el valor de intensidad de un p´ıxel f(x,y) es mayor o igual al valor del umbral T, entonces el píxel pertenece al objeto, caso contrario éste pertenece al fondo. Según Chauhan et al. [28] la clasificacion de los métodos basado en umbrales puede ser resumido de acuerdo al siguiente concepto:

$$T = T[(x, y), p(x, y), f(x, y)] \tag{5}$$

donde $f(x, y)$ es la intensidad de gris del punto (x, y) y $p(x, y)$ denota alguna propiedad de vecindad local de (x, y).

Cuando T depende solamente de $f(x, y)$, entonces el umbral es llamado global. Si T depende de ambos $f(x, y)$ y $p(x, y)$, el umbral es llamado local. La umbralización local es computacionalmente mas costosa que la umbralización global. Es bastante útil para segmentar estructuras de variado fondo, y para extracción de regiones que son pequeñas y dispersas. Si T depende de (x, y), $p(x, y)$ y $f(x, y)$ entonces el método se llama umbralización adaptativo. En iluminaciones no uniformes, es una tarea tediosa encontrar el valor global del umbral. En éstos casos la imagen es particionada en sub-imágenes y umbrales adaptativos son aplicados en cada sub-imagen y luego el resultado es combinado [27].

Una imagen umbralizada g(x,y) está definida por:

$$g(x, y) = \begin{cases} 1, & \text{si } f(x, y) > T. \\ 0, & \text{si } f(x, y) < T. \end{cases} \tag{6}$$

3.2.3. Segmentación orientada a regiones

Los métodos de detección de regiones son usados para particionar la imagen en regiones que son similares en caracter´ısticas como color, textura, etc. Se agrupan p´ıxeles en regiones homogéneas, dentro de éste tipo de segmentación se tiene el método de crecimiento de la región y el métodos de dividir-combinar regiones[27]:

- Método de crecimiento de la región: En éste enfoque, inicialmente se selecciona un píxel semilla y la región crece fusionando píxeles vecinos de semilla con criterios de similitud satisfactorios como color y valor de intensidad. Este proceso continúa hasta que ningún píxel cumple con el criterio de similitud.

- Método de dividir-fusionar regiones: En éste método inicialmente toda la imagen es tratada como región semilla. Si no se cumplen criterios de similaridad predefinidos entonces la imagen se divide en cuadrantes. Esta división continúa hasta que subregiones homogéneas son obtenidas. Las regiones homogéneas subdivididas son luego fusionadas para extraer objetos de interés de la imagen de acuerdo a caracter´ısticas similares. El procedimiento culmina cuando ya no es posible realizar mas fusiones [30].

3.2.4. Métodos basados en clustering

Clustering es una división de datos en grupos de similares características. Cada grupo está formado por objetos que son similares entre ellos y distintos a los que pertenecen a otros grupos. Las técnicas de clusterización se pueden dividir en clusterización supervisada y clusterizacion no supervisada. Clusterización supervisada involucra interaccion humana y clusterizacion no supervisada decide el criterio de clustering por sí mismo. Los métodos de clusterización pueden también ser divididos como duros o difusos dependiendo si el patrón pertenece exclusivamente a un solo cluster o a varios con diferentes valores. [31].

El método $K-Means$ es un método popular de clusterización dura que particiona una imagen en K clusters [32]. Los siguientes pasos conforman el método $K-Means$:

- Seleccionar k p´ıxeles como centroides iniciales.
- Formar k clusters asignando todos los píxeles al centroide más cercano.
- Recalcular el centroide de cada cluster y reasignar los p´ıxeles.
- Repetir el proceso hasta que los centroides no cambien más o hasta un criterio de parada.

3.3. Clusterización iterativa lineal simple (SLIC)

Los super p´ıxeles SLIC corresponden a agrupamientos en el espacio de caracter´ısticas $labxy$. Los super p´ıxeles capturan redundancia en la imagen y reduce la complejidad de las subsiguientes tareas de procesamiento de imágenes. SLIC tiene básicamente dos parámetros de entrada, el número de super píxeles deseado de tamaños aproximadamente iguales k, y un parámetro m para ofrecer control sobre su compacidad. La complejidad de éste algoritmo es lineal en el número de píxeles N, e independiente del número de super píxeles k [18]. Para imágenes a colores, el algoritmo SLIC tiene las siguientes etapas:

- La imagen de entrada es convertida al espacio de colores $CIELAB$.
- Un total de k centros de agrupamiento $c_i = [l_i a_i b_i x_i y_i]^T$ iniciales son mostrados en una grilla regular espaciada en $S = \sqrt{N/k}$ p´ıxeles separados.
- Opcionalmente, los centros pueden ser movidos hacia la posición más baja de gradiente en una vecindad de 3x3, para evitar la inicialización de un píxel con mucho ruido.

- Cada píxel es asociado con el centro de agrupamiento más próximo de acuerdo a una medida de distancia D pero considerando solo los centros cuya región de $2Sx2S$ se sobrepone a su localizacion.
- Un paso de actualizacion ajusta los centros de los grupos a un vector $[labxy]^T$
- Los pasos de atribucion y de actualización son entonces repetidos por 10 iteraciones.
- Finalmente, algunos p´ıxeles disjuntos que no pertenecen al mismo componente conexo que su centro de agrupamiento pueden permanecer. Por ello, una etapa de post-procesamiento para forzar las conexiones es aplicado a través de la atribucion de un rótulo distinto para cada componente conexo.
  La distancia D ésta dada por:

$$d_c = \sqrt{(l_i - l_j)^2 + (a_i - a_j)^2 + (b_i - b_j)^2} \tag{7}$$

$$d_s = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{8}$$

$$D = \sqrt{d_c^2 + (\frac{d_s}{S})^2 m^2} \tag{9}$$

Donde m da una importancia relativa entre disimilaridad de color (d_c) y distancia espacial (d_s). Cuando m es grande, los super píxeles resultantes son más compactos y cuando m es pequeño, los super píxeles tienen una mejor adherencia para los bordes de los objetos presentes en la imagen, pero con tamaño y forma irregulares [33]. Para muchas tareas de visión computacional, super p´ıxeles altamente uniformes y compactos que respeten los l´ımites de la imagen, como los super p´ıxeles generados en la figura 3.1 [18]

Figura 3.1: Imágenes segmentadas usando SLIC con 64, 256 y 1024 super píxeles

3.4. Redes neuronales convolucionales

En esta seccion se describirá brevemente las redes neuronales artificiales y las redes neuronales convolucionales (CNN).

3.4.1. Red neuronal pre-alimentada (feed-forward)

Antes de describir las redes neuronales convolucionales, es necesario ver primero el modelo básico o red neuronal pre-alimentada. Para mayores detalles revisar [34]. Considerando un escenario de aprendizaje supervisado donde tenemos un conjunto etiquetado de datos (x^i, y^i), donde x^i e y^i denotan las características y etiquetas respectivamente para el i^{th} ejemplar de entrenamiento. A alto nivel, las redes neuronales proveen una forma compleja de representación, una función no lineal $h_W(x)$ de nuestra variable de entrada x. La función $h_W(x)$ es parametrizada por una matriz de pesos W cuyos valores podemos encontrar para ajustar nuestros datos [35].

En la figura 3.2 [35] se muestra una red neuronal simple que consiste de tres neuronas de entrada x_{11}, x_{12} y x_{13} y una sola neurona de salida $y = h_W(x_{11}, x_{12}, x_{13})$

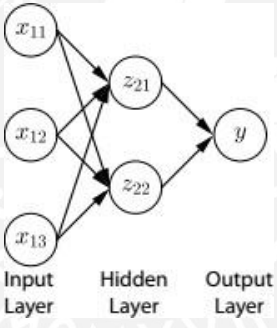

Figura 3.2: Una red neuronal pre-alimentada simple

Una red neuronal está generalmente organizada en múltiples capas. Por ejemplo, la red neuronal mostrada en la figura 3.2 consiste de tres capas: La capa de entrada, la capa oculta y una capa de salida, también se muestra un conjunto de aristas conectando las neuronas entre capas adyacentes. Mientras que la figura 3.2 muestra una red neuronal completamente interconectada donde cada neurona está conectada a todas las neuronas de la capa anterior, no es una condición necesaria en la construccion de la red neuronal. El patrón de conectividad de una red neuronal está generalmente relacionada a la arquitectura de red neuronal [35].

Aparte de las neuronas en la capa de entrada, cada neurona x_i en la red neuronal es una unidad computacional que toma una entrada de valores de las neuronas de la

capa predecesora. Por ejemplo, en la figura 3.2, la neurona z_{21} tiene como entradas a (x_{11}, x_{12}, x_{13}) y la entrada de y es z_{21} y z_{22}, dadas las entradas, las neuronas primero calculan una combinación lineal simple de sus entradas, de forma general, dado $x_1, \ldots, x_n$ que denotan las entradas de la neurona z_j [35]. Entonces primero se calcula:

$$a_j = \sum_{i=1} w_{ij} x_i + b_j \tag{10}$$

Donde $w_j i$ es el parámetro que describe la interacción entre z_j y la neurona de entrada x_i. Los términos b_j es un bias o un término de intercepción asociado a la neurona z_j. Luego se aplica una función de activación no lineal [36] a a_j. Algunas funciones de activacion comunes incluyen la función sigmoide y la tangente hiperbolica. En particular, la activación o valor de la neurona z_j está definida por

$$z_j = h(a_j) = h(\sum_{i=1} w_{ij} x_i + b_j) \tag{11}$$

Donde h es la función de activacion no lineal. Dado un conjunto de entradas x y pesos W , podemos calcular la activacion de cada neurona siguiendo los pasos descritos líneas arriba. La activación de cada neurona depende solo de los valores de las neuronas en las capas predecesoras, se calculan las activaciones comenzando de la primera capa oculta (las cuales a su vez dependen de los valores de sus entradas) y se procede capa por capa de atras hacia adelante. Este proceso donde la información se propaga a través de la red se denomina paso de propagación hacia adelante [35]. Al final de la propagación hacia adelante, obtenemos un conjunto de salidas $y = h_W(x)$. y viene a ser el resultado de clasificar una entrada x.

Los parámetros W en la red neuronal estan compuestos por los términos de peso para cada uno de las aristas, así como un término de sesgo o bias para cada una de las neuronas, con exclusión de los que están en la capa de entrada. Dado nuestro conjunto de entrenamiento etiquetado $\{(x^i, y^i)\}$, el objetivo es aprender o encontrar un valor óptimo para los parámetros W de tal manera que se minimice una función de pérdida [35]. El enfoque estándar para encontrar los valores óptimos de W para minimizar la función de pérdida es el algoritmo de propagación del error hacia atrás (error backpropagation algorithm) [34]. Debido a que el algoritmo de propagación del error hacia atras es un enfoque estándar en la literatura sobre redes neuronales y no es esencial para el entendimiento del trabajo presentado en ésta tesis, no se entrará a detallar dicho algoritmo.

3.4.2. Red neuronal convolucional (CNN)

Una red neuronal convolucional es un tipo de red multicapa y jerárquica. Hay tres principales factores que diferencian a una CNN de la red neuronal pre-alimentada descrita en la seccion anterior: Campos receptivos locales, pesos compartidos y capas de

agrupamiento o submuestreo.[35]

En la arquitectura simple, una red neuronal pre-alimentada (feed-forward) descrita en la seccion anterior, cada neurona está completamente interconectada a cada uno de las neuronas de las capas siguientes, cada neurona de la capa oculta realiza calcula una función que depende de los valores de cada nodo en la capa de entrada, sin embargo, En visión computacional, es frecuente y ventajoso explorar subestructuras locales en la imagen. Por ejemplo, los p´ıxeles que conforman una vecindad (p´ıxeles adyacentes) tienden a estar fuertemente correlacionados mientras que los p´ıxeles que están más lejos unos de otros en la imagen tienden a estar débilmente correlacionados o simplemente no correlacionados [35]. No es sorpresa entonces que muchas representaciones estándares de características usadas en problemas de visión computacional están basados en características locales de vecindades en la imagen [37] [38]. En una arquitectura CNN, se capturan dichas subestructuras locales de la imagen al limitar cada neurona de tal manera que depende solo de un subconjunto espacialmente local de las variables de la capa anterior. Por ejemplo, si la entrada de la CNN es una imagen de 32 x 32, una neurona en la primera capa oculta podría depende solo de un parche de 8 x 8 del total que es 32 x 32. El conjunto de neuronas en la capa de entrada que afecta la activacion de una neurona es denominada como el campo receptivo de la neurona, en una CNN cada neurona individual generalmente tiene un campo receptivo local y no global [35].

La segunda caracter´ıstica que distingue a las CNNs de las redes neuronales simples es el hecho de que los pesos están compartidos a través de diferentes neuronas en las capas ocultas. Se debe recordar que cada neurona en la red primero calcula una combinación lineal de sus entradas, podemos ver este proceso como la evaluación de un filtro lineal sobre los valores de entrada, en este contexto, compartir los pesos entre múltiples neuronas en la capa oculta se traduce a evaluar el mismo filtro sobre múltiples subventanas de la imagen de entrada, entonces podemos decir que la CNN aprende un conjunto de filtros $F = \{F_i | i = 1, ..., n\}$, cada uno de los cuales es aplicado a todas las subventanas de la imagen de entrada. Usando el mismo conjunto de filtros sobre la imagen entera, la red aprende una codificación o representación general de los datos subyacentes. Restringiendo los pesos para que sean iguales a través de diferentes neuronas también tiene un efecto de regularización sobre la CNN, permitiendo que la red pueda generalizar mejor. Otro beneficio del compartimiento de pesos es el hecho que se reduce significativamente el número de parámetros libres de la CNN, haciendo el entrenamiento de manera mas eficiente. Como nota final, evaluando un filtro F sobre cada ventana en la imagen de entrada I equivale a realizar una convolución de la imagen I con el filtro F, entonces en el paso convolucional de una CNN, se toma la imagen de entrada y se convoluciona con cada filtro F para obtener el mapa de respuesta convolucional (mapa de filtros) [35].

El componente distintivo final en una CNN es la presencia de capas de agrupamiento cuyo objetivo aqu´ı es doble: reducir la dimensionalidad de las respuestas

convolucionales y conferir un pequeño grado de invarianza traslacional al modelo. El enfoque estándar es a través de agrupamiento espacial [39]. En el agrupamiento espacial, el mapa de respuestas convolucionales es primero dividido en un conjunto de *mxn* bloques (generalmente disjuntos), luego se evalúa una función de agrupamiento sobre las respuestas en cada bloque. Este proceso da como resultado un mapa de respuestas mas pequeño de dimensión *mxn* (una respuesta por cada bloque). En el caso de agrupamiento maximo (max pooling), la respuesta para cada bloque es el valor máximo del bloque de respuestas, y en el caso del agrupamiento promedio (average pooling), la respuesta es el valor promedio del bloque de respuestas [35]. En la figura 3.3 [35] se muestra un ejemplo de agrupamiento promedio. En éste caso, el mapa de respuestas convolucionales es una entrada de 4 x 4, realizamos una función de agrupamiento promedio sobre cuatro bloques de 2 x 2, la respuesta agrupada es el promedio de los valores en el bloque. Después de aplicar el procedimiento de agrupamiento promedio, la respuesta es un mapa de respuesta de agrupamiento de 2 x 2, comparado al original que era de 4 x 4, ésto representa una reducción significativa de la dimensionalidad del mapa de respuesta [35].

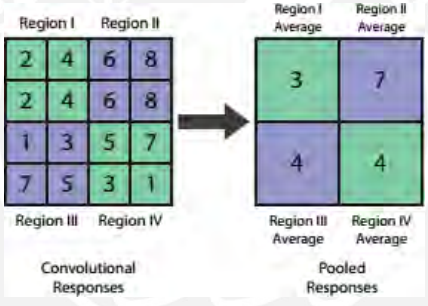

Figura 3.3: Agrupamiento promedio en una CNN

En una CNN típica, tenemos múltiples capas, alternando entre capas de convolución y capas de agrupamiento.Por ejemplo, podemos apilar otra capa de convolución-agrupamiento en la parte superior de las salidas de la primera capa de convolución-agrupamiento, en este caso, simplemente tratamos las salidas del primer conjunto de capas de convolucion-agrupamiento como la entrada al segundo conjunto de capas. De esta manera, podemos construir una arquitectura multicapa o profunda. Intuitivamente, los filtros convolucionales de bajo nivel,como aquellos en la primera capa convolucional, se puede pensar que proporcionan un nivel bajo codificación de los datos de entrada. En el caso de los datos de una imagen, estos filtros de bajo nivel pueden consistir en filtros de borde simples. A medida que nos movemos a capas más profundas en la red neuronal, el modelo comienza a aprender más y más estructuras complicadas. Al usar múltiples capas y grandes números de filtros, la arquitectura CNN puede proporcionar un poder de representación de gran complejidad. Para en-

trenar una CNN, podemos usar la técnica estándar de propagación del error hacia atrás utilizada para entrenar redes neuronales pre-alimentadas [34]

3.5. Entrenamiento de una red neuronal

Antes de iniciarse el proceso de aprendizaje, algunos parámetros son fijados como las funciones de activacion que se usarán así como el tipo y la cantidad de capas. El entrenamiento de una red neuronal artificial es un proceso de actualización de los pesos de la red. Ésto es realizado usando un proceso denominado propagación del error hacia atras [34], el objetivo es minimizar una función de pérdida, en otras palabras lo que se desea es actualizar los pesos de la red neuronal de tal manera que las clases o predicciones de los ejemplares de entrenamiento se aproximen a las clases o predicciones predefinidas en el conjunto de entrenamiento.

3.5.1. Funciones de pérdida

- Entropia cruzada: Es una función de pérdida multi clase, representada por:

$$Loss(x, y) = -\sum_i y_i * log(\frac{e^{x_i}}{\sum_j e^{x_j}}) \tag{12}$$

  Con x como vector de entrada e y como vector binario con valores 0 excepto en la posicion que indica la clase predecida correctamente [34].

3.5.2. Algoritmos de optimización

Las funciones de pérdida de las redes neuronales convolucionales son altamente no convexas pero derivables, los algoritmos basados en gradientes pueden ser aplicados. Sin embargo, las redes neuronales convolucionales son usualmente hechas de decenas de miles de parámetros. Solo la derivada de primer orden son usadas en la prácti- ca. Usar la segunda derivada resulta demasiado costoso en términos de memoria y complejidad computacional [40].

- Gradiente descendente estocastico (SGD): Es uno de los algoritmos de optimi- zación mas populares, consiste en usar un subconjunto pequeño del conjunto de entrenamiento para calcular la gradiente de los parámetros con respecto a la función de pérdida:

$$\Theta_{t+1} = \Theta_t - \lambda \nabla_{\Theta_t} L(f_{\Theta_t}(x_i), y_i) \tag{13}$$

  Donde λ es denominado tasa de aprendizaje (learning rate) [40]

3.5.3. Tasa de aprendizaje cíclico para entrenar redes neuronales

La tasa de aprendizaje determina cuan rapido o lento se actualizarán los pesos (o parámetros) de una red neuronal. De hecho la tasa de aprendizaje es uno de los parámetros mas importantes y mas difíciles de elegir debido a que afecta significativamente el performance del modelo. Smith [41] define una técnica para encontrar una tasa de aprendizaje óptima, esto se logra al incrementar poco a poco la tasa de aprendizaje iniciando en un valor muy pequeño, hasta que la pérdida deje de decrecer. Graficando la tasa de aprendizaje versus los lotes (batches) podemos ver que tasa de aprendizaje nos conviene utilizar. Por ejemplo en la figura 3.4, se tiene una gráfica de iteraciones versus tasa de aprendizaje, conforme se va iterando, se va aumentando gradualmente la tasa de aprendizaje, y en la figura 3.5 se puede observar que eligiendo una tasa de aprendizaje de 0.01, la pérdida aún está disminuyendo, si eligiéramos 0.1, la pérdida deja de disminuir. Por lo tanto la tasa de aprendizaje óptima en éste caso es 0.01.

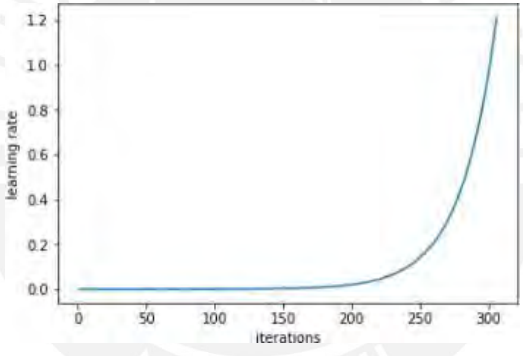

Figura 3.4: Iteraciones vs. Tasa de aprendizaje. Se va incrementando el valor de la tasa de aprendizaje conforme aumentan las iteraciones

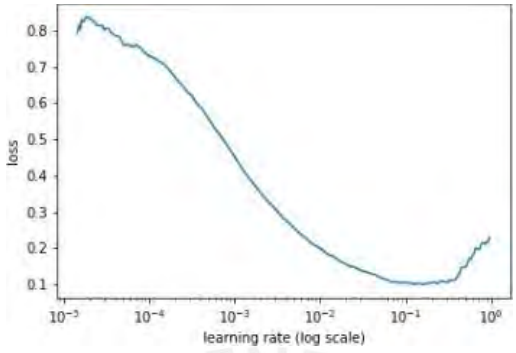

Figura 3.5: Tasa de aprendizaje vs. Pérdida

3.5.4. Gradiente descendente estocástico con reinicios (SGDR)

La técnica definida por Loshchilov et al. [42] busca gradualmente disminuir la tasa de aprendizaje conforme el entrenamiento progresa. Esto es útil porque conforme nos vamos acercando al valor óptimo de los pesos, se buscará tener desplazamientos más pequeños hacia el valor mínimo. Sin embargo podemos encontrarnos en un escenario en el cual dentro del espacio formado por el valor de los pesos, nos encontremos en un lugar poco resiliente en el que pequeños cambios a los valores de los pesos podría resultar en variaciones grandes en el valor de la pérdida. Lo que queremos es encontrar partes en el espacio formado por los pesos que son precisos y estables. Para ello cada cierto tiempo se incrementa la tasa de aprendizaje (re inicios en SGDR) los cuales causaran que el modelo de un salto a un lugar diferente en el espacio de los pesos si dicha área es demasiado empinado (pico). En la figura 3.7 [43] se puede observar en la parte izquierda como los pesos se inicializan y se localizan en el punto azul, después comienza el descenso hasta terminar en un mínimo local, sin embargo no sabemos si dicho mínimo es resiliente, es decir, si un cambio pequeño hace que el valor de la pérdida aumente demasiado. Por otro lado en la parte derecha se tiene que cada vez que se llega a un m´ınimo local, se realizan saltos y se comienza de nuevo el descenso hasta encontrar un área resiliente y estable.

3.5.5. Enfoques de regularización

Las redes neuronales pueden memorizar cualquier data. Durante el entrenamiento, su precisión sobre el conjunto de entrenamiento usualmente converge hacia la perfección mientras se degrada la precisión sobre el conjunto de prueba o test, éste fenómeno se denomina overfitting o sobreajuste [40].

Aumentación de data (data augmentation)

Como su nombre lo dice, es un método para aumentar el conjunto de entrenamiento para que el modelo no caiga en sobreajustes u overfitting. La aumentación de data puede tomar varias formas dependiendo del conjunto de datos. Por ejemplo si los objetos son invariantes a la rotación como las galaxias, entonces es adecuado generar imágenes aplicando rotación a la imágenes originales [40]. En la figura ?? se muestran seis imagenes del mismo gato, se aplicó rotación y acercamiento para generar las imagenes.

Figura 3.6: Ejemplo de data augmentation

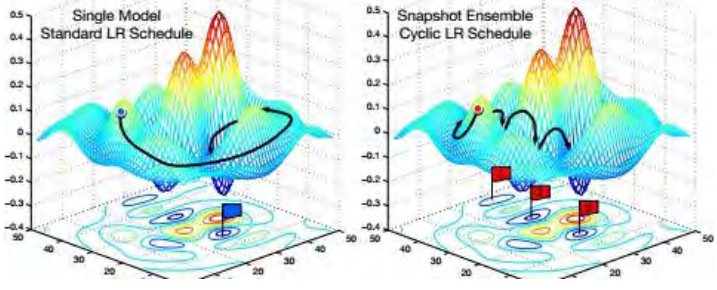

Figura 3.7: Izquierda. Descenso convencional. Derecha. Descenso en ciclos (reinicios)

3.6. Redes residuales (Resnet)

El entrenamiento de las redes neuronales profundas tienen una mayor complejidad. El aprendizaje residual facilita el entrenamiento de éste tipo de redes al solucionar el problema de la degradación de la precisión, conforme se aumenta la profundidad de la red, se tiene el problema de que se va distorsionando el aprendizaje y esto produce que la precisión empeora conforme se agregan más capas a la red. La

solución para este problema consiste en aplicar lo que se denomina como aprendizaje residual al replicar la entrada de la capa anterior a la salida de la capa actual, de este modo se conserva lo aprendido en capas anteriores y se agrega lo aprendido en las capas subsiguientes mediante bloques residuales [44]. En la figura 3.8 [44] se tiene un bloque residual, se replica la entrada de la capa anterior a la salida de la capa actual para evitar la degradación de la precisión. En la figura 3.9 [44] se muestra una red Resnet con sus respectivos bloques residuales(derecha) y una red plana sin bloques residuales(izquierda).

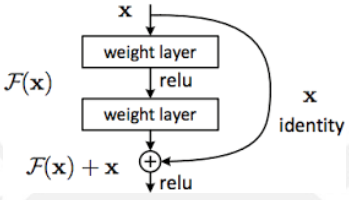

Figura 3.8: Bloque residual

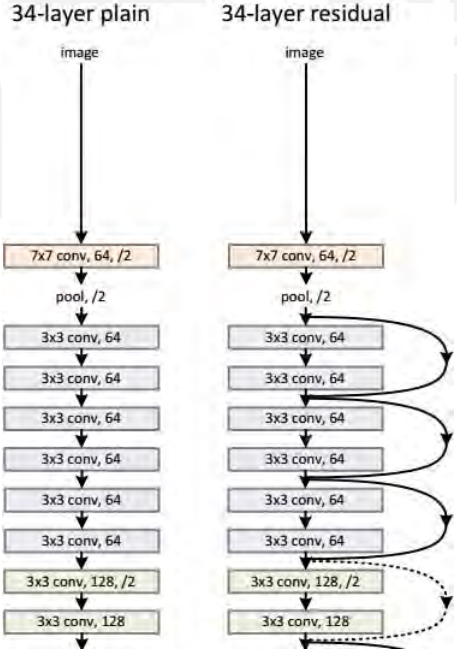

Figura 3.9: Arquitectura Resnet

CAPÍTULO 4

Metodología

Se ha considerado en el presente trabajo al problema de segmentación de páginas de documentos históricos como un problema de etiquetado de los píxeles de las imágenes. En la presente tesis se propone el uso de una red neuronal convolucional profunda con arquitectura Resnet para la tarea de etiquetado de los píxeles. La idea principal es obtener un conjunto de detectores de características y entrenar un clasificador no lineal sobre las características extraídas por los detectores. Con el conjunto de detectores de características y el clasificador, se podrá etiquetar los píxeles de las imágenes de documentos históricos que no forman parte del conjunto de entrenamiento ni de validación.

Se proponen los siguientes pasos para llevar a cabo el proceso de segmentación:

1. Pre-procesamiento.
    a) Segmentacion de las imagenes en super píxeles.
    b) Extracción de los parches por cada super píxel.
    c) Elaboracion del conjunto de entrenamiento y etiquetado.
2. Definición de la arquitectura de la red neuronal convolucional.
3. Entrenamiento de la red neuronal convolucional.

En la figura 4.1 se muestra la secuencia de pasos para la creación y clasificación del conjunto de entrenamiento: Segmentacion de la imagen en super píxeles, extracción del parche central por cada super píxel, entrada y clasificación de cada parche extraído mediante la CNN.

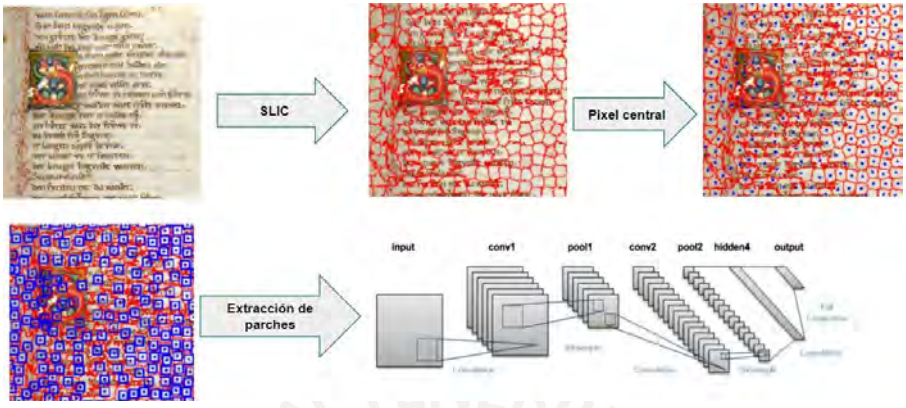

Figura 4.1: Pasos para realizar el entrenamiento de la CNN propuesta

4.1. Pre-procesamiento

Segmentación de la imagen en super p**íxeles**

Con la finalidad de mejorar la velocidad del procesamiento de etiquetado de los píxeles de las imagenes de documentos históricos, dada una imagen, el primer paso es aplicar el algoritmo de clustering iterativo lineal simple (SLIC) [18] para la obtención de subregiones de la imagen denominados super píxeles. Un super píxel es un segmento de imagen que contiene píxeles que guardan una relación, ya sea nivel de intensidad parecido, posicion, entre otros. Por lo tanto en lugar de etiquetar cada píxel de la imagen, se etiquetará solo el píxel central de cada super píxel y el resto de p´ıxeles que pertenecen al super p´ıxel son etiquetados con la misma clase o etiqueta. La superioridad del enfoque de etiquetado de super píxeles para la segmentación de imágenes de páginas de documentos históricos ha sido demostrado en [45].

En la figura 4.2 se puede observar la imagen original y la imagen segmentada en super p´ıxeles.

Figura 4.2: Imagen de documento histórico dividido en super píxeles. Izquierda, la imagen original. Derecha, la imagen segmentada en super p´ıxeles (cada segmento está con borde de color rojo).

Extracción de los parches por cada super píxel en la imagen

Una vez segmentada la imagen en super p´ıxeles, el siguiente paso consiste en extraer el cuadrilatero central (denominado parche) de cada super p´ıxel, para ello primero se calcula el punto central de cada super p´ıxel, tal como se muestra en la figura 4.3. Una vez calculado el punto central de cada super p´ıxel, se extrae el cuadrilátero central cuyo centro es precisamente el punto central extraído, tal como se muestra en la figura 4.4. Se debe resaltar que todos los parches extraídos tendrán el mismo tamaño.

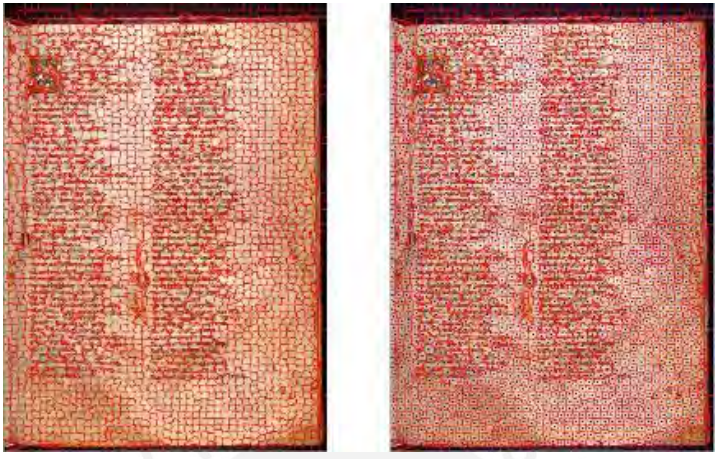

Figura 4.3: Izquierda: Imagen segmentada en superpíxeles. Derecha: Imagen segmentada, cada segmento con su respectivo p´ıxel central en azul.

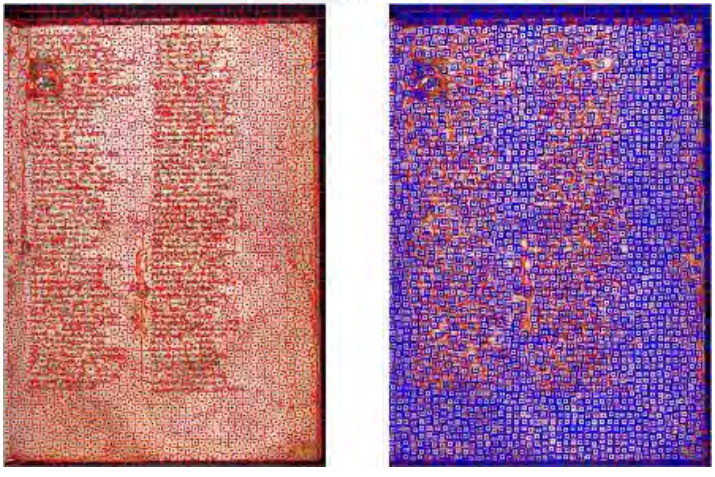

Figura 4.4: Izquierda: Imagen segmentada y sus respectivos p´ıxeles centrales. Derecha: Imagen segmentada y sus respectivos parches centrales.

Elaboración del conjunto de entrenamiento y etiquetado

A cada parche extraído se le asigna una clase o etiqueta según el ground truth del dataset elegido, en éste caso las secciones o segmentos de las imágenes de documentos

históricos están en formato XML como se muestra en la figura 4.5. Cada sección se representa como una coleccion de puntos cuyas coordenadas son los vértices del polígono que encierra a un segmento de la imagen, por ejemplo en la figura 4.5, se tiene que la seccion “comment” de la imagen “d-144.png” está encerrada en el polígono cuyos vértices son (137,234) ; (266,231)...(137,237). Así es posible etiquetar todos los píxeles de la imagen según a qué polígono definido en su ground truth pertenezca, en éste caso solo se etiquetó a los píxeles centrales de cada parche y para entrenar la red neuronal convolucional, la etiqueta de dicho p´ixel central del parche, será también la etiqueta del parche. En la figura 18 se tiene una muestra de los parches que formarán el conjunto de entrenamiento. En la figura 19 se tiene un ejemplar original de imagen original y la misma imagen con los segmentos [’text’, ’decoration’, ’comment’, ’page’] definidos en su correspondiente ground truth.

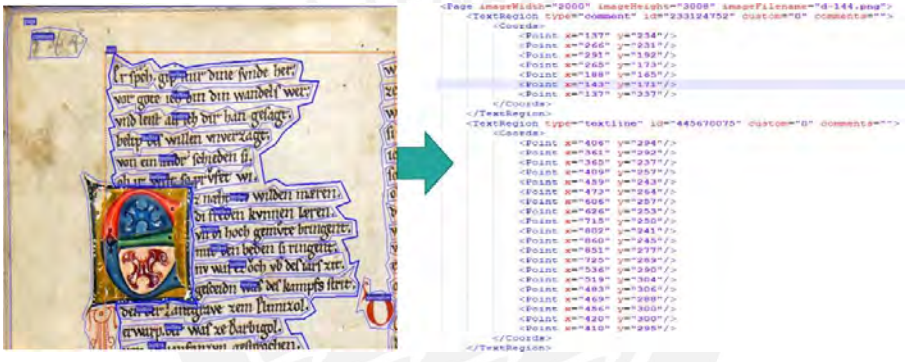

Figura 4.5: Ground truth en formato XML

En la figura 4.6 se tiene una muestra de los parches que formarán el conjunto de entrenamiento. En la figura 4.7 se tiene un ejemplar original de imagen original y la misma imagen con los segmentos [’text’, ’decoration’, ’comment’, ’page’] definidos con su correspondiente ground truth.

Figura 4.6: Ejemplo de parches de 224 x 224 que formarán parte del conjunto de entrenamiento.

Figura 4.7: Imagen original y los segmentos ['página', 'texto', 'decoración', 'comentario'] en blanco, azul, rojo y verde respectivamente.

4.2. Definición de la arquitectura de la CNN

En la presente tesis se utilizará la arquitectura Resnet [23] para el proceso de clasificación de los parches extraídos de los super píxeles de las imágenes de documentos históricos, a diferencia de la arquitectura planteada por Chen et al.[26], se ha optado por explorar las bondades de utilizar una arquitectura mucho más compleja y profunda para mejorar el proceso de clasificación de los parches, especificamente se plantea utilizar la arquitectura Resnet-18, como sabemos, éste tipo de arquitectura de red neuronal cuenta con los denominados bloques residuales, los cuales permiten que la precisión no se deteriore en el proceso de entrenamiento. Como se muestra en la figura 4.8, la entrada de la red tendrá dimensión 224 x 224 x 3 y estará conformado por los parches extraídos de los super píxeles, a partir del píxel central de cada super píxel.

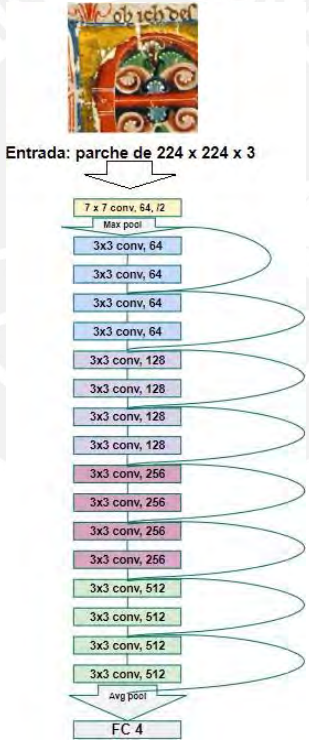

Figura 4.8: Ejemplo de parches de 224 x 224 que formarán parte del conjunto de entrenamiento.

La última capa de la red tendrá 4 salidas y representarán las 4 clases predefinidas

que son: *Comentario, decoración, página y texto*. Esta capa consiste de una regresión logística con softmax cuyas salidas serán la probabilidad de ocurrencia de cada clase:

$$P(y = i|x, W_1, ..., W_c, b_1, ..., b_c) = \frac{e^{W_i x + b_i}}{\sum_{j=1}^{M} e^{W_j x + b_j}} \quad (14)$$

Donde x es la salida de la capa completamente interconectada, W_i y b_i son los pesos y biases de la $i$ava neurona en la capa, y M es el número de clases. La clase predecida $\hat{y}$ es la clase que ha alcanzado la máxima probabilidad la cual se expresa por:

$$\hat{y} = argmax(P(y = i|x, W_1, ..., W_M, b_1, ..., b_M)) \quad (15)$$

Tanto en la capa convolucional como en la capa completamente interconectada de la red, la función de activacion que se usará será el de las unidades lineales rectificadas (ReLUs por sus siglas en inglés). Una función de activación ReLU está dada por:

$$f(x) = max(0, x) \quad (16)$$

donde x es la entrada de la neurona. La superioridad de usar ReLUs como función de activación en las neuronas en una red neuronal convolucional sobre otras funciones de activación como la Sigmoid está demostrado en [46].

4.3. Entrenamiento

Utilizaremos un modelo pre-entrenado, es decir un modelo creado para resolver un problema distinto (clasificacion de animales por ejemplo). En lugar de construir el modelo desde cero para resolver un problema de clasificación, utilizaremos un modelo entrenado con el dataset ImageNet [47] el cual cuenta con 1.2 millones de imágenes y 1000 clases como punto de partida. En este caso, reemplazamos la última capa de 1000 neuronas de salida por una capa de 4 neuronas correspondientes, conservamos los pesos del modelo entrenado para resolver el problema de Imagenet y al inicio entrenamos solamente la última capa de la red (la capa completamente interconectada).

Para entrenar la red neuronal convolucional, por cada super píxel, se generará un parche como resultado de extraer la parte central del super p´ıxel. El parche es considerada como entrada de la red neuronal convolucional. El tamaño de cada parche es de 224 x 224 píxeles. La etiqueta de cada parche será la etiqueta de su píxel central. Los parches de las imagenes de entrenamiento son usadas para entrenar la red. En la red neuronal convolucional, la función de costo está definida como la pérdida de entropía cruzada (cross-entropy loss) [34] según:

$$\Gamma(X, Y) = \frac{\sum}{n} \quad (\ln a(x^{(i)}) + (1 \quad y^{(i)}) \ln (1 \quad a(x^{(i)}))) \quad (17)$$

Donde $X = \{x^{(1)}, ..., x^{(n)}\}$ ses el conjunto de entrenamiento conformado por los parches de las imágenes (parte central de cada super píxel) e $Y = \{y^{(1)}, ..., y^{(n)}\}$ son las correspondientes etiquetas. El número de parches de imagen que conforman los ejemplares de entrenamiento es n. Para cada $x^{(i)}$, $a(x^{(i)})$ es la salida de la red definida en la ecuación 14. La red neuronal convolucional es entrenada con el método del gradiente descendente estocastico con reinicio(SGDR por sus siglas en inglés). Después de cada capa convolucional se aplic normalización por lotes (batch normalization) [42] y después de la capa completamente conectada se aplicó la técnica del valor de regularización (dropout) [48]. El objetivo de aplicar la técnica del valor de marginación es evitar sobreajuste del modelo a los ejemplares de entrenamiento(overfitting) al introducir ruido aleatorio a los ejemplares de entrenamiento.

Se siguieron los siguientes pasos para entrenar la red neuronal convolucional profunda propuesta en la presente tesis:

1. Inicializar el modelo con los pesos pre-cargados cuando se entrenó con el dataset ImageNet [47], excepto la última capa, se reemplaza las 1000 neuronas de salida del modelo original de Imagenet y se reemplaza por las 4 neuronas de salida correspondientes a los segmentos de interés de la presente tesis los cuales son: ['página', 'texto', 'decoracion', 'comentario'].
2. Utilizar la técnica descrita en [41] para encontrar la tasa de aprendizaje óptima el cual denominaremos lr.
3. Entrenar solo la última capa completamente interconectada por 10 épocas utilizando la tasa de aprendizaje hallada en el paso 2.
4. Entrenar la última capa con aumento de data (data augmentation) por 5 épocas para evitar overfitting.
5. Entrenar toda la red neuronal en tres bloques, es decir, desde la capa de entrada, hasta la capa de salida, se divide el número de capas en tres bloques; el primer bloque se entrena con una tasa de aprendizaje $lr/3$, el segundo bloque con una tasa de aprendizaje de $lr/2$ y el último bloque con una tasa de aprendizaje lr. Donde lr es la tasa de aprendizaje óptima encontrada en el paso 2.

CAPÍTULO 5

Experimentación

5.1. Bases de datos (datasets)

En la presente tesis se utilizó la base de datos de imágenes de documentos manuscritos históricos de dominio público Parzival el cual se detalla a continuación:

Parzival: La base de datos Parzival descrita en [49] contiene imágenes de manuscritos históricos que tienen las siguientes características:

- Datan del siglo XIII.
- Está en lenguaje medieval aleman.
- Están escritos por tres autores.
- Tienen estilo gotico.
- Para la escritura se utilizó tinta sobre pergamino.

En la figura 4.7 se muestra una imagen de la base de datos Parzival y su ground truth.

En la tabla 5.2 se muestra como se ha distribuido la base de datos: Tamaño de cada imagen, número de imagenes de entrenamiento (Training), número de imágenes de validación(Validation) y número de imagenes de pruebas(Test).

Tabla 5.2: Distribución de las imágenes.

Base de dato	Dimensión	Training	Validation	Test	Total
Parzival	2000x3008x3	25	6	4	35

Las regiones de interés que se ha considerado en la presente tesis son las que se mencionan a continuación:

- Texto (text): Representada con color azul.
- Decoración (decoration): Representado con color rojo.
- Página (page): Representado con color blanco.
- Comentario (comment): Representado con color verde.

5.2. Métricas de evaluación

Para evaluar la calidad de la segmentacion de las imágenes de documentos históricos, se usarán las métricas utilizadas por Chen et al. [26]. Los cuales son:

- Precisión de píxeles(pixel accuracy).
- Precisión media de píxeles(mean pixel accuracy).
- Interseccion sobre la unión promedio(mean IU).
- Frecuencia ponderada de interseccion sobre la unión (frequency weighted IU).

As´ı, se definen las siguientes variables:

- n_c: Número de clases o etiquetas.
- n_{ij}: Número de píxeles de la clase i cuya predicción es la clase j. Para la clase i:
  - n_{ii}: Número de píxeles correctamente clasificados(verdaderos positivos).
  - n_{ij}: Número de píxeles incorrectamente clasificados(falsos positivos).
  - n_{ji}: Número de píxeles incorrectamente no clasificados(falsos positivos).
- t_i: Número total de píxeles en la clase i tal que:

$$t_i = \sum_j n_{ji} \tag{18}$$

Dadas las variables, se definen cada una de las métricas según las siguientes ecuaciones:

- pixel accuracy:

$$acc = \frac{\sum_i n_{ii}}{\sum_i t_i} \tag{19}$$

- mean accuracy:

$$acc_{mean} = \frac{1}{n_c} * \sum_i \frac{n_{ii}}{t_i} \tag{20}$$

- mean IU:

$$iu_{mean} = \frac{1}{n} * \sum_i \frac{n_{ii}}{t_i + \sum_j n_{ji} - n_{ii}} \tag{21}$$

- f.w. IU:

$$iu_{weighted} = \frac{1}{\sum_k t_k} * \sum_i \frac{t_i * n_{ii}}{t_i + \sum_j n_{ji} - n_{ii}} \tag{22}$$

5.3. Configuración de los experimentos

Se realizaron en total 4 experimentos, por cada experimento se generaron los conjuntos de entrenamientos como resultado de dividir cada imagen en 3000, 6000, 9000 y 12000 super píxeles. En la tabla 5.3 se muestra la distribución aproximada(aproximada porque la técnica SLIC vista en la sección 3.3 recibe un parámetro que indica el número de super píxeles en las cuales será dividida la imagen, pero eso no significa que la imagen sea dividida exactamente en ese número de super píxeles) de parches para los grupos de entrenamiento, validación y pruebas:

Tabla 5.3: Distribución de los parches sobre el dataset Parzival.

Base de datos	Parche	Training	Validation	Test	Total
Parzival	224*224*3	≈25x3000	≈6x3000	≈4x3000	≈105000
		≈25x6000	≈6x6000	≈4x6000	≈210000
		≈25x9000	≈6x9000	≈4x9000	≈315000
		≈25x12000	≈6x12000	≈4x12000	≈420000

En todos los experimentos se utilizó la técnica de aumentación de data(data augmentation), ésta técnica no ayuda a mejorar la precisión del modelo (accuracy),

pero si es útil para que el modelo tenga una mejor capacidad de generalización y por lo tanto evitar overfitting. En la figura 5.1 se muestra un ejemplo de imágenes generadas a partir de un parche.

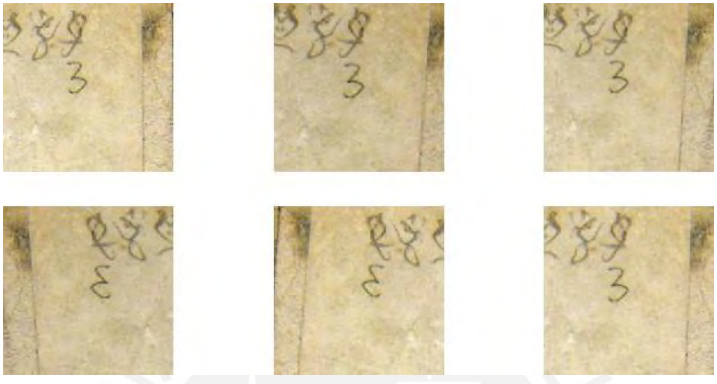

Figura 5.1: Ejemplo de data augmentation de un parche. Las 6 imágenes son de un mismo parche.

5.4. Análisis de resultados.

En todos los experimentos se utilizó la técnica para encontrar la tasa de aprendizaje óptima descrita en la seccion 3.5.3. No en todos los casos se encontró la misma tasa óptima. En las figuras 5.2, 5.3, 5.4 y 5.5 se muestran las gráficas que nos ayudarán a determinar la tasa de aprendizaje óptima por experimento. Por ejemplo en la figura 5.2, con una tasa de aprendizaje de 10^{-2}, vemos que la pérdida aún disminuye, a diferencia de los demás casos en los que nos conviene utilizar una tasa de aprendizaje de 10^{-3}

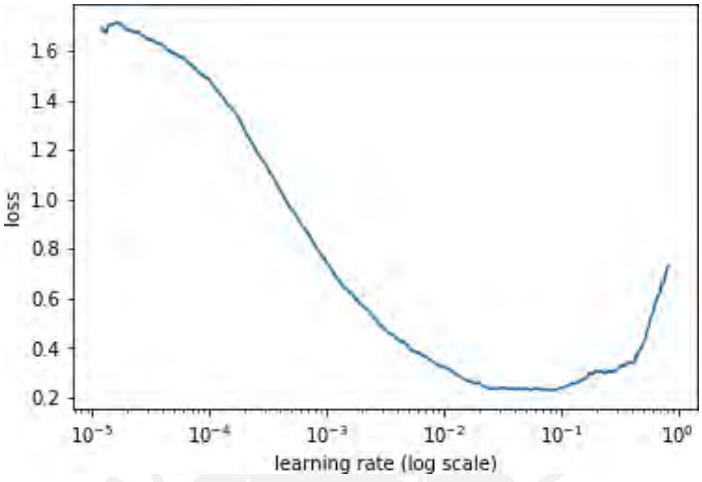

Figura 5.2: Pérdida vs. Tasa de aprendizaje para el experimento de 3000 super p´ıxeles.

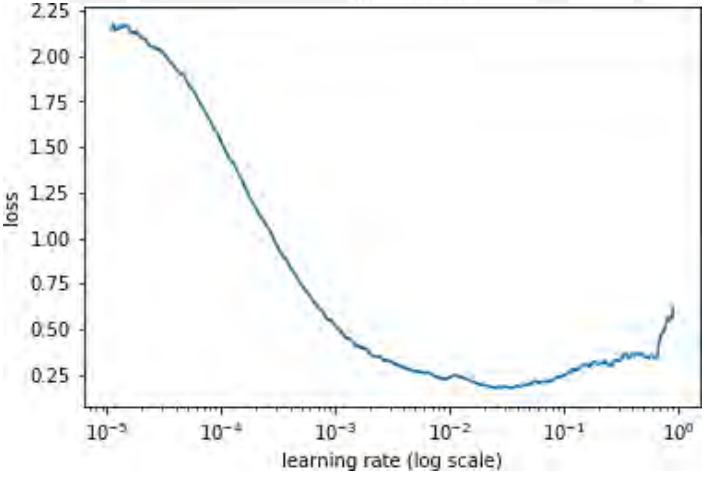

Figura 5.3: Pérdida vs. Tasa de aprendizaje para el experimento de 6000 super p´ıxeles.

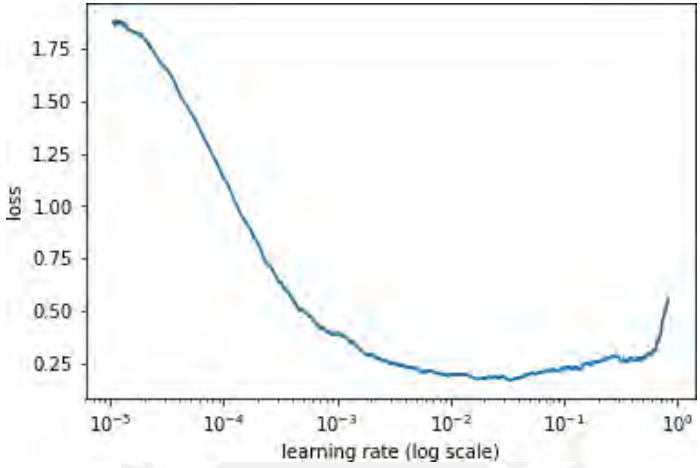

Figura 5.4: Pérdida vs. Tasa de aprendizaje para el experimento de 9000 super p´ıxeles.

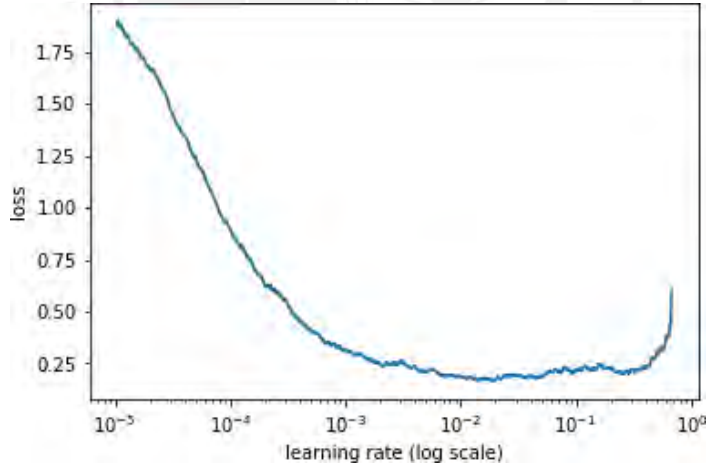

Figura 5.5: Pérdida vs. Tasa de aprendizaje para el experimento de 12000 super p´ıxeles.

Una forma común de analizar el resultado de clasificación de los modelos utilizados es ver la matriz de confusión (confusion matrix), en las figuras 5.6 - 5.7, 5.8 - 5.9, 5.10 - 5.11 y 5.12 - 5.13 se muestran las matrices de confusión y matriz de

confusión en porcentajes para los experimentos de 3000, 6000, 9000 y 12000 super p´ıxeles respectivamente:

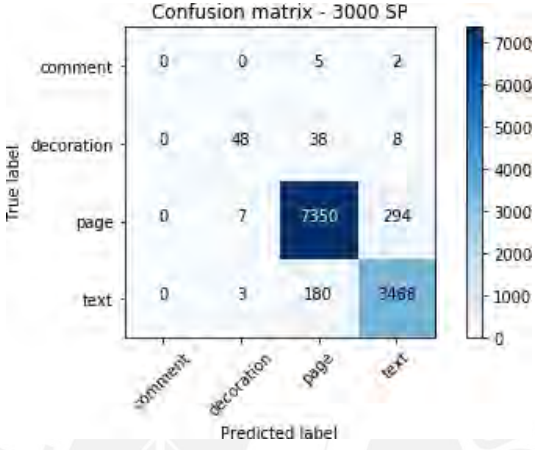

Figura 5.6: Matriz de confusion para el experimento de 3000 super p´ıxeles.

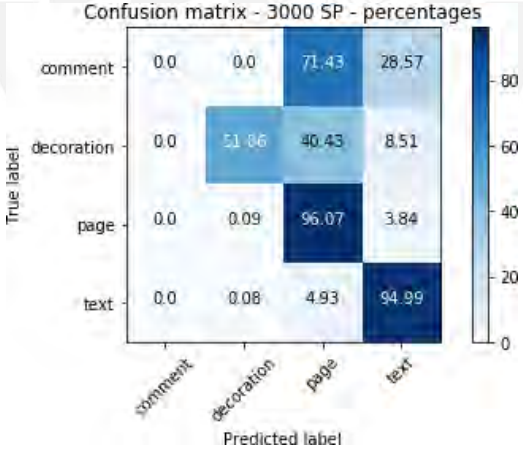

Figura 5.7: Matriz de confusion para el experimento de 3000 super p´ıxeles - En porcentajes.

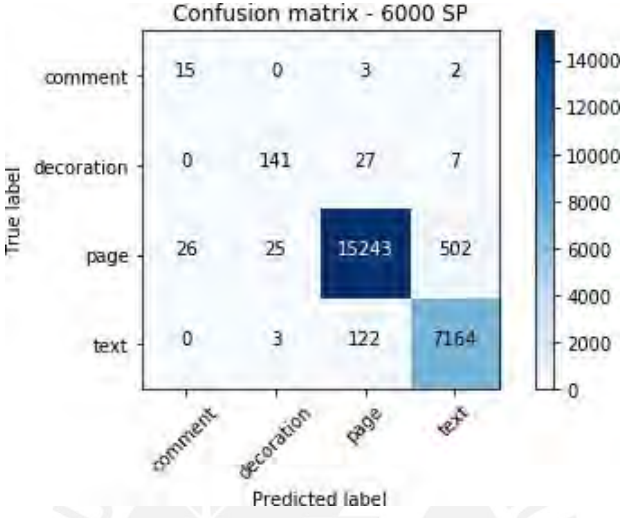

Figura 5.8: Matriz de confusion para el experimento de 6000 super p´ıxeles.

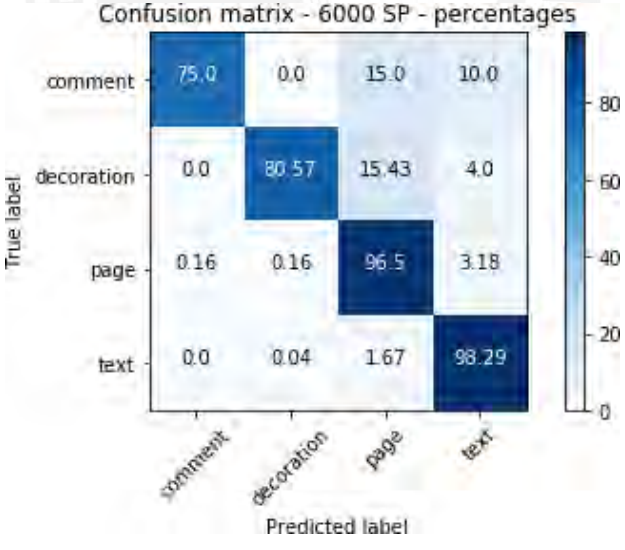

Figura 5.9: Matriz de confusion para el experimento de 6000 super p´ıxeles - En porcentajes.

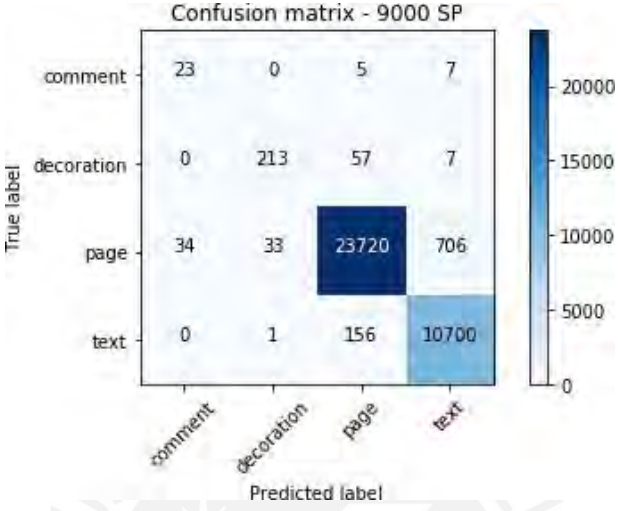

Figura 5.10: Matriz de confusion para el experimento de 9000 super p´ıxeles.

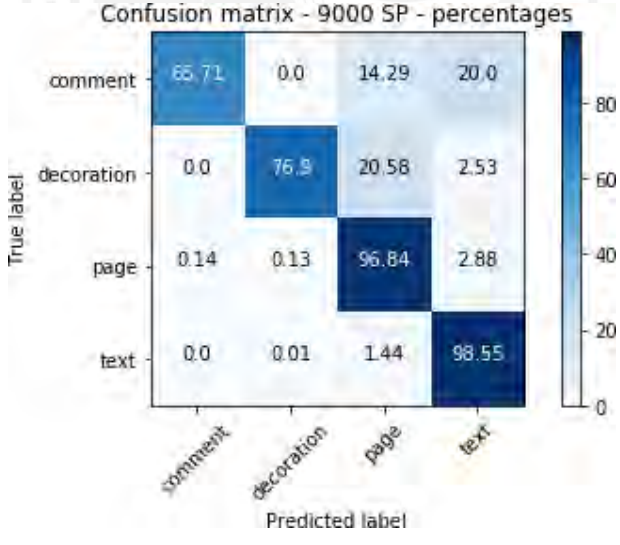

Figura 5.11: Matriz de confusion para el experimento de 9000 super p´ıxeles - En porcentajes.

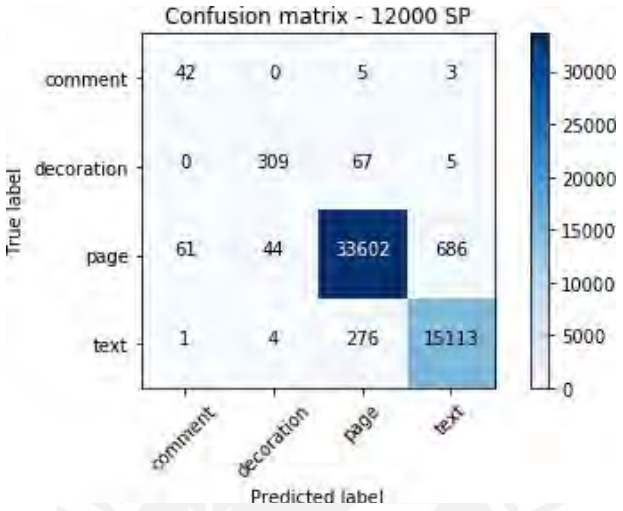

Figura 5.12: Matriz de confusion para el experimento de 12000 super p´ıxeles.

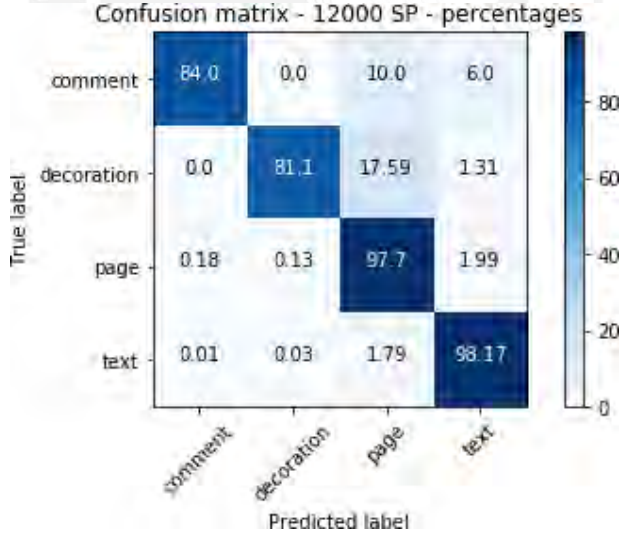

Figura 5.13: Matriz de confusion para el experimento de 12000 super p´ıxeles - En porcentajes.

Se observa para todos los casos que el modelo se equivoca con mayor frecuencia cuando clasifica parches de la clase `page` y parches de la clase `text`. Por otro lado, conforme se aumenta el número de super píxeles, se incrementa la precisión de la clasificación.

Se tiene que tener en cuenta que las matriz de confusión solo nos muestra el resultado de clasificación de los parches, pero no nos indica la calidad de la segmentación de todas las imagenes del conjunto de entrenamiento.

A continuación se muestran algunos resultados de clasificación del modelo entrenado con imágenes del experimento cuyas imágenes fueron divididas con 12000 super p´ıxeles. En las figuras 5.14 y 5.15 se muestran ejemplares clasificados correctamente e incorrectamente por el modelo respectivamente por el modelo entrenado con imágenes segmentadas con 12000 super píxeles.

Figura 5.14: Ejemplares correctamente clasificados por el modelo cuyas imágenes se segmentaron con 12000 super p´ıxeles.

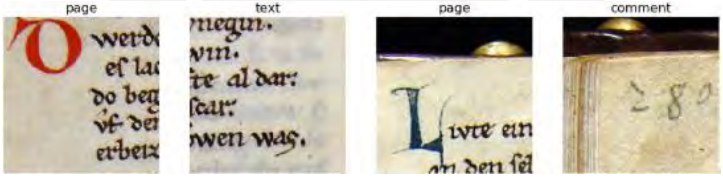

Figura 5.15: Ejemplares clasificados de manera errónea por el modelo cuyas imagenes se segmentaron con 12000 super p´ıxeles.

5.4.1. Calidad de la segmentación

Para realizar el analisis de la calidad de la segmentación, utilizaremos las métricas definidas en la seccion 5.2, las cuales son: precisión de píxeles(pixel accuracy), media de precisión de píxeles(mean accuracy), media de la intersección sobre la unión(mean IU) e interseccion sobre la unión ponderado(f.w. IU). En las figuras 5.16, 5.17, 5.18, 5.19 se muestran los resultados para las métricas descritas en ese orden.

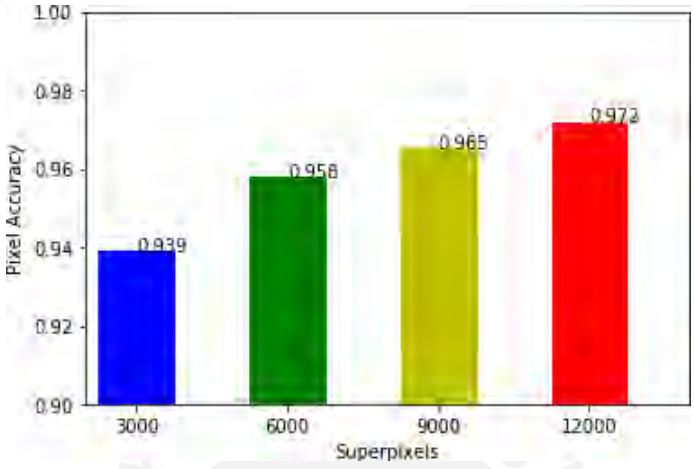

Figura 5.16: Precisión de píxeles por experimento.

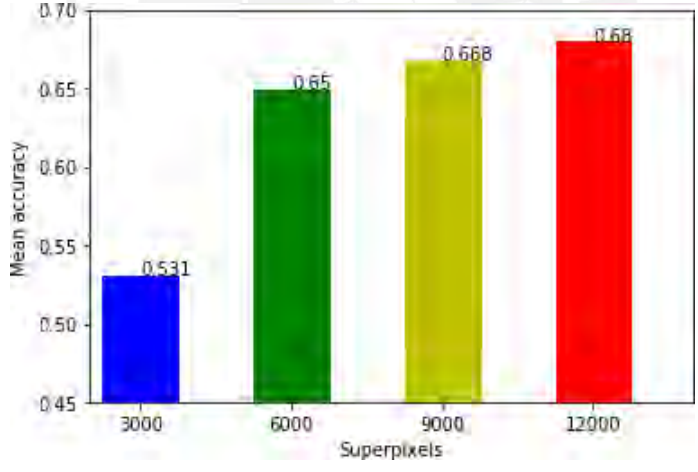

Figura 5.17: Media de precisión de píxeles por experimento.

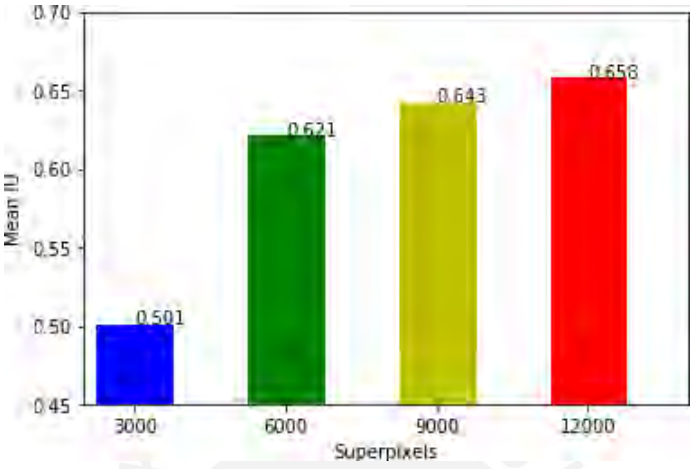

Figura 5.18: Media de intersección sobre la unión por experimento.

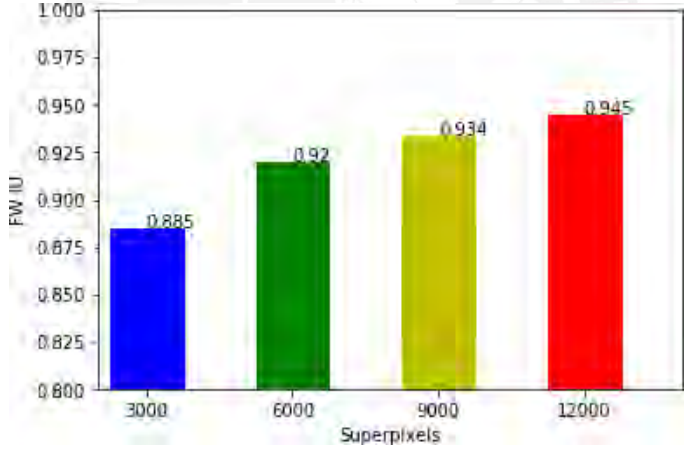

Figura 5.19: Interseccion sobre la unión ponderado.

Tabla 5.5: Resultados de la calidad de segmentación en cada experimento.

Num. Superpixels	Acc.	Mean. acc.	Mean. IU	F.W IU
3000	0.9390048205	0.5309275138	0.501467207	0.884907323
6000	0.9579387051	0.6497137159	0.6213659093	0.9195909031
9000	0.9654836686	0.6677013391	0.642635418	0.9336178901
12000	0.9716704205	0.6798185386	0.65798644	0.9450567032

En la tabla 5.5 se resume los valores obtenidos en cada experimento. Finalmente,

en las figuras 5.20, 5.21, 5.22, 5.23 se tienen imágenes no etiquetadas segmentadas por el modelo entrenado con los parámetros del experimento de 12000 super p´ıxeles.

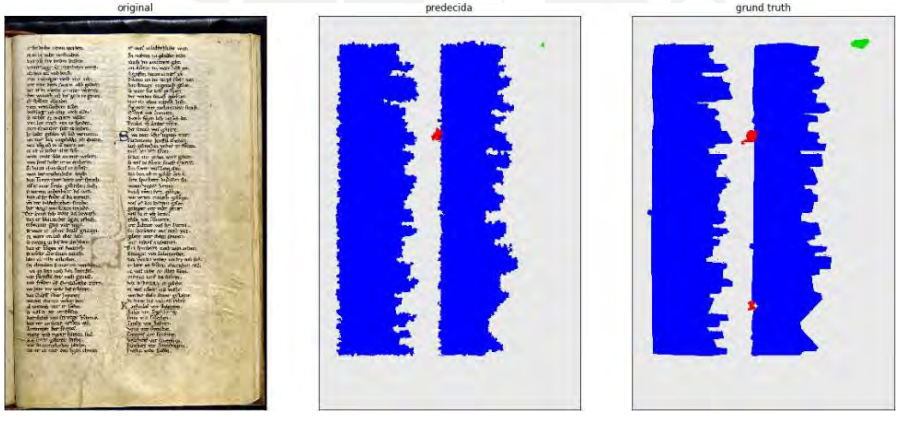

Figura 5.20: Resultado 1 - Izquierda: Imagen original. Centro: Imagen segmentada por el modelo, Derecha: Ground truth.

Figura 5.21: Resultado 2 - Izquierda: Imagen original. Centro: Imagen segmentada por el modelo, Derecha: Ground truth.

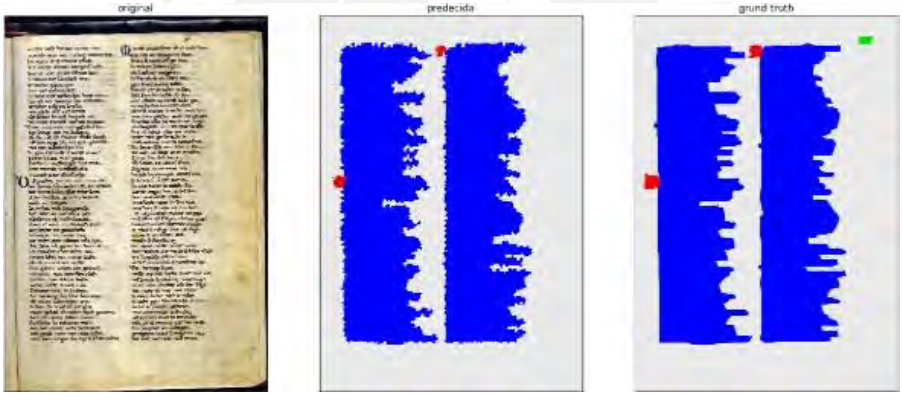

Figura 5.22: Resultado 3 - Izquierda: Imagen original. Centro: Imagen segmentada por el modelo, Derecha: Ground truth.

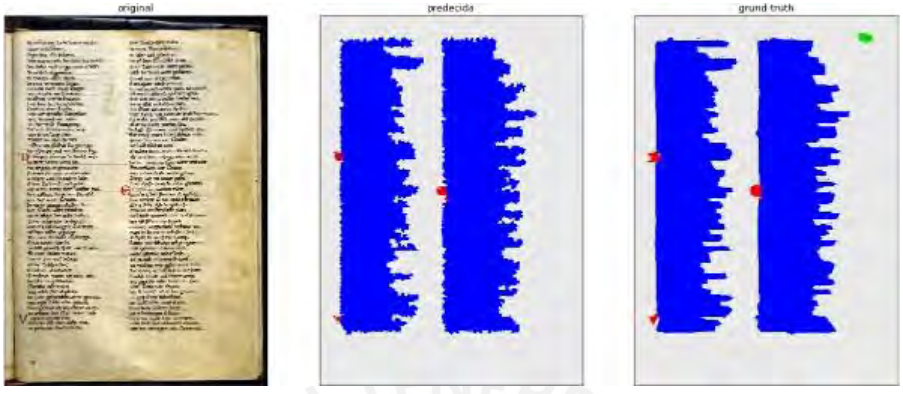

Figura 5.23: Resultado 4 - Izquierda: Imagen original. Centro: Imagen segmentada por el modelo, Derecha: Ground truth.

5.4.2. Ejecución

La red neuronal convolucional profunda propuesta se implementó utilizando la librer´ıa fastai [50] que utiliza a su vez la librer´ıa pytorch. Los experimentos fueron ejecutados en una maquina virtual con un procesador Intel Xeon E5-2623 v4 de 2.60 GHz y frecuencia maxima de 3.20 GHz, tarjeta de video Nvidia P5000 de 16GB GDDR5X y con una memoria ram de 30GB.

CAPÍTULO 6

Conclusiones y trabajos futuros

6.1. Conclusiones

En la presente tesis se propone una red neuronal convolucional profunda para segmentar imágenes de documentos manuscritos históricos, a diferencia de métodos tradicionales de segmentacion de páginas que usan clasificadores entrenados con vectores de características construidos manualmente, el método propuesto extrae características directamente de los parches extraídos de las imágenes, más aún, la extracción de caracterísitcas y la clasificacion se combinan en un solo paso, es un modelo end-to-end. Los experimentos en el dataset público Parzival muestran que la calidad de la segmentación mejora cuando se segmenta a las imágenes del conjunto de entrenamiento en un mayor número de super píxeles. En todos los experimentos se encontró que los modelos entrenados para clasificar los parches, confunden más los parches de tipo página y texto. Los parches de tipo ”decoración” y ”comentario” son los que han sido clasificados con menor precisión por el modelo, si observamos los resultados en las figuras 5.20, 5.21, 5.22, 5.23 podemos observar que los parches de tipo comentario no han podido ser detectados correctamente por el modelo y han sido confundidos en su mayoría por parches de tipo ”pagina” y que existe un desbalance entre el número de parches de tipo ”comentario” y ”decoracion” con respecto a la cantidad de parches de tipo ”texto” y ”pagina”. La métrica de calidad de segmentación que mejora considerablemente cuando aumentamos el número de super píxeles por imagen son las de precisión de píxeles promedio(Mean pixel accuracy) e intersección sobre la unión promedio(mean IU). El uso de modelos pre-entrenados, la aumentación de data (data augmentation), la búsqueda de la taza de aprendizaje óptima y el uso de gradiente

descendente estocastico con reinicio (SGDR) hicieron que nuestro modelo entrenado tenga la capacidad de generalizar mejor las predicciones y evitar el sobreajuste del modelo al conjunto de entrenamiento (overfitting), los experimentos muestran que el modelo propuesto ha alcanzado resultados interesantes dada la complejidad de las características de las imagenes de documentos manuscritos históricos descritos en la sección 1.1.

6.2. Trabajos futuros

Como trabajos futuros se propone explorar otras variantes de segmentación para la obtención de super píxeles, como por ejemplo el algoritmo IFT-SLIC propuesto por B. Alexandre et al. [33]. Para abordar el problema del desbalance de ejemplares por clase (por ejemplo se tienen pocos ejemplares de tipo ”comentario” con respecto a los ejemplares de tipo ”página”) se recomienda evaluar alguna alternativa para lidiar con éste problema como los ”balanceadores de clases” al momento de obtener los lotes (batches) cuando se entrena la red. Para los casos en los que se tienen parches cuyos p´ıxeles tienen similitudes a los parches de otras clases, como por ejemplo el parecido que existe entre parches de tipo ”comentario” con parches de tipo ”texto” ó ”página”, se recomienda como trabajo futuro explorar una heurística que permita clasificar mejor los parches como por ejemplo el hecho de que los ”comentarios” tienen una ubicación particular en las imágenes. Por último, se recomienda explorar otras arquitecturas de redes neuronales convolucionales como por ejemplo la arquitectura propuesta por Ronneberger et al. [25]

Bibliografía

[1] C. Grana, D. Borghesani, and R. Cucchiara, "Automatic Segmentation of Digitalized Historical Manuscripts," *Multimedia Tools Appl.*, vol. 55, no. 3, pp. 483–506, Dec. 2011. [Online]. Available: http://dx.doi.org/10.1007/s11042-010-0561-8

[2] S. S. Bukhari, T. M. Breuel, A. Asi, and J. El-Sana, "Layout analysis for arabic historical document images using machine learning," *2012 International Conference on Frontiers in Handwriting Recognition*, pp. 639–644, 2012.

[3] K. Chen, H. Wei, M. Liwicki, J. Hennebert, and R. Ingold, "Robust text line segmentation for historical manuscript images using color and texture," in *2014 22nd International Conference on Pattern Recognition*, Aug 2014, pp. 2978–2983.

[4] K. Chen, H. Wei, J. Hennebert, R. Ingold, and M. Liwicki, "Page segmentation for historical handwritten document images using color and texture features," in *2014 14th International Conference on Frontiers in Handwriting Recognition*, Sept 2014, pp. 488–493.

[5] M. Bulacu, R. van Koert, L. Schomaker, and T. van der Zant, "Layout analysis of handwritten historical documents for searching the archive of the cabinet of the dutch queen," in *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, vol. 1, Sept 2007, pp. 357–361.

[6] T. Phan, K. Nguyen, and M. Nakagawa, "A nom historical document recognition system for digital archiving," *Int. J. Doc. Anal. Recognit.*, vol. 19, no. 1, pp. 49–64, Mar. 2016. [Online]. Available: http://dx.doi.org/10.1007/s10032-015-0257-8

[7] C. Panichkriangkrai, L. Li, and K. Hachimura, "Character segmentation and retrieval for learning support system of japanese historical books," in

Proceedings of the 2Nd International Workshop on Historical Document Imaging and Processing, ser. HIP '13. New York, NY, USA: ACM, 2013, pp. 118–122. [Online]. Available: http://doi.acm.org/10.1145/2501115.2501129

[8] B. Gatos, G. Louloudis, and N. Stamatopoulos, "Segmentation of historical handwritten documents into text zones and text lines," in *2014 14th International Conference on Frontiers in Handwriting Recognition*, Sept 2014, pp. 464–469.

[9] R. Cohen, A. Asi, K. Kedem, J. El-Sana, and I. Dinstein, "Robust text and drawing segmentation algorithm for historical documents," in *Proceedings of the 2Nd International Workshop on Historical Document Imaging and Processing*, ser. HIP '13. New York, NY, USA: ACM, 2013, pp. 110–117. [Online]. Available: http://doi.acm.org/10.1145/2501115.2501117

[10] A. Asi, R. Cohen, K. Kedem, J. El-Sana, and I. Dinstein, "A coarse-to-fine approach for layout analysis of ancient manuscripts," in *2014 14th International Conference on Frontiers in Handwriting Recognition*, Sept 2014, pp. 140–145.

[11] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov 1998.

[12] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, Dec 1989.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, ser. NIPS'12. USA: Curran Associates Inc., 2012, pp. 1097–1105. [Online]. Available: http://dl.acm.org/citation.cfm?id=2999134.2999257

[14] T. Wang, D. J. Wu, A. Coates, and A. Y. Ng, "End-to-end text recognition with convolutional neural networks," in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, Nov 2012, pp. 3304–3308.

[15] Y. Kim, "Convolutional neural networks for sentence classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2014, pp. 1746–1751. [Online]. Available: http://www.aclweb.org/anthology/D14-1181

[16] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1915–1929, Aug 2013.

[17] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," in *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ser. ICML'14. JMLR.org, 2014, pp. I–647–I–655. [Online]. Available: http://dl.acm.org/citation.cfm?id=3044805.3044879

[18] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "Slic superpixels," p. 15, 2010.

[19] K. Chen, M. Seuret, H. Wei, M. Liwicki, J. Hennebert, and R. Ingold, "Ground truth model, tool, and dataset for layout analysis of historical documents," in *DRR*, 2015.

[20] F. Simistira, M. Seuret, N. Eichenberger, A. Garz, M. Liwicki, and R. Ingold, "Diva-hisdb: A precisely annotated large dataset of challenging medieval manuscripts," in *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, Oct 2016, pp. 471–476.

[21] Y. Liang, R. M. Guest, and M. Fairhurst, "Implementing word retrieval in handwritten documents using a small dataset," in *2012 International Conference on Frontiers in Handwriting Recognition*, Sept 2012, pp. 728–733.

[22] G. Vamvakas, B. Gatos, N. Stamatopoulos, and S. J. Perantonis, "A complete optical character recognition methodology for historical documents," in *2008 The Eighth IAPR International Workshop on Document Analysis Systems*, Sept 2008, pp. 525–532.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 770–778.

[24] C. Wick and F. Puppe, "Fully convolutional neural networks for page segmentation of historical document images," 11 2017.

[25] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," pp. 234–241, 10 2015.

[26] K. Chen, M. Seuret, J. Hennebert, and R. Ingold, "Convolutional neural networks for page segmentation of historical document images," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 01, Nov 2017, pp. 965–970.

[27] R. C. Gonzalez and R. E. Woods, *Digital Image Processing (3rd Edition)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2006.

[28] A. S. Chauhan, S. Silakari, and M. Dixit, "Image segmentation methods: A survey approach," in *2014 Fourth International Conference on Communication Systems and Network Technologies*, April 2014, pp. 929–933.

[29] S. E. S. Jayaraman and T. Veerakumar, *Digital Image Processing*. Tata McGraw Hill Education, 2009.

[30] S. L. Horowitz and T. Pavlidis, "Picture Segmentation by a directed split-and-merge procedure," *Proceedings of the 2nd International Joint Conference on Pattern Recognition, Copenhagen, Denmark*, pp. 424–433, 1974.

[31] S. Saraswathi and A. Allirani, "Survey on image segmentation via clustering," in *2013 International Conference on Information Communication and Embedded Systems (ICICES)*, Feb 2013, pp. 331–335.

[32] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: A k-means clustering algorithm," *Applied Statistics*, vol. 28, no. 1, pp. 100–108, 1979. [Online]. Available: http://dx.doi.org/10.2307/2346830

[33] E. B. Alexandre, A. S. Chowdhury, A. X. Falcão, and P. A. V. Miranda, "Ift-slic: A general framework for superpixel generation based on simple linear iterative clustering and image foresting transform," in *2015 28th SIBGRAPI Conference on Graphics, Patterns and Images*, Aug 2015, pp. 337–344.

[34] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[35] D. J. Wu, "End-to-end text recognition with convolutional neural networks," 2012. [Online]. Available: https://crypto.stanford.edu/~dwu4/papers/HonorThesis.pdf

[36] B. Karlik and A. V. Olgac, "Performance analysis of various activation functions in generalized mlp architectures of neural networks," *International Journal of Artificial Intelligence and Expert Systems*, vol. 1, no. 4, pp. 111–122, 2011.

[37] A. Krizhevsky, "Learning multiple layers of features from tiny images," 2009.

[38] Q. V. Le, W. Y. Zou, S. Y. Yeung, and A. Y. Ng, "Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis," in *CVPR 2011*, June 2011, pp. 3361–3368.

[39] Y.-L. Boureau, F. Bach, Y. LeCun, and J. Ponce, "Learning mid-level features for recognition," 2010.

[40] R. Cadène, N. Thome, and M. Cord, "Master's thesis : Deep learning for visual recognition," *CoRR*, vol. abs/1610.05567, 2016. [Online]. Available: http://arxiv.org/abs/1610.05567

[41] L. N. Smith, "Cyclical learning rates for training neural networks," *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 464–472, 2017.

[42] I. Loshchilov and F. Hutter, "Sgdr: Stochastic gradient descent with warm restarts," 2016.

[43] G. Huang, Y. Li, G. Pleiss, Z. Liu, J. E. Hopcroft, and K. Q. Weinberger, "Snapshot ensembles: Train 1, get M for free," *CoRR*, vol. abs/1704.00109, 2017. [Online]. Available: http://arxiv.org/abs/1704.00109

[44] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[45] K. Chen, C. Liu, M. Seuret, M. Liwicki, J. Hennebert, and R. Ingold, "Page segmentation for historical document images based on superpixel classification with unsupervised feature learning," in *2016 12th IAPR Workshop on Document Analysis Systems (DAS)*, April 2016, pp. 299–304.

[46] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the 25th International*

Conference on Neural Information Processing Systems - Volume 1, ser. NIPS'12. USA: Curran Associates Inc., 2012, pp. 1097–1105. [Online]. Available: http://dl.acm.org/citation.cfm?id=2999134.2999257

[47] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," *Int. J. Comput. Vision*, vol. 115, no. 3, pp. 211–252, Dec. 2015. [Online]. Available: http://dx.doi.org/10.1007/s11263-015-0816-y

[48] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, Jan. 2014. [Online]. Available: http://dl.acm.org/citation.cfm?id=2627435.2670313

[49] A. Fischer, M. Wuthrich, M. Liwicki, V. Frinken, H. Bunke, G. Viehhauser, and M. Stolz, "Automatic transcription of handwritten medieval documents," in *2009 15th International Conference on Virtual Systems and Multimedia*, Sept 2009, pp. 137–142.

[50] J. Howard *et al.*, "fastai," https://github.com/fastai/fastai, 2018.

[51] A. M. Hesham, M. A. Rashwan, H. M. Al-Barhamtoshy, S. M. Abdou, A. A. Badr, and I. Farag, "Arabic Document Layout Analysis," *Pattern Anal. Appl.*, vol. 20, no. 4, pp. 1275–1287, Nov. 2017. [Online]. Available: https://doi.org/10.1007/s10044-017-0595-x

[52] T. V. Phan, B. Zhu, and M. Nakagawa, "Development of nom character segmentation for collecting patterns from historical document pages," in *HIP@ICDAR*, 2011.

[53] L. N. Smith, "No more pesky learning rate guessing games," *CoRR*, vol. abs/1506.01186, 2015. [Online]. Available: http://arxiv.org/abs/1506.01186

[54] A. Fischer, V. Frinken, A. Fornés, and H. Bunke, "Transcription alignment of latin manuscripts using hidden markov models," in *Proceedings of the 2011 Workshop on Historical Document Imaging and Processing*, ser. HIP '11. New York, NY, USA: ACM, 2011, pp. 29–36. [Online]. Available: http://doi.acm.org/10.1145/2037342.2037348

[55] F. Simistira, M. Seuret, N. Eichenberger, A. Garz, M. Liwicki, and R. Ingold, "Diva-hisdb: A precisely annotated large dataset of challenging medieval manuscripts," in *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, Oct 2016, pp. 471–476.