



APRENDIZAJE DE MÁQUINA

PROYECTO FINAL

Reconocimiento de caracteres escritos a mano

Autores:

Hector Miguel Rodríguez Sosa (C411)
Karen Danelis Cantero López (C411)
Luis Alejandro Rodríguez Otero (C411)
Sebastián Suárez Gómez (C411)

zealot.algo@gmail.com
karen.canterolopez@gmail.com
luisalejandrorodriguezotero@gmail.com
sebastiansuarezgomez01@gmail.com

2024

Contents

1	Introducción	3
1.1	Motivación	3
1.2	Problemática	4
1.2.1	Características Específicas del Lenguaje:	4
1.2.2	Deterioro y Variaciones en la Escritura:	4
1.2.3	Escasez de Datos de Entrenamiento:	4
1.2.4	Relevancia Histórica y Cultural:	4
1.3	Objetivos generales y específicos	5
1.3.1	Objetivo General	5
1.3.2	Objetivos Específicos	5
2	Estado del Arte	6
2.1	Técnicas de preprocesamiento de imágenes	6
2.1.1	Procedimiento general	6
2.1.2	Combinando técnicas	7
2.1.3	Otras técnicas	7
2.2	Reconocimiento de caracteres	8
2.2.1	CNN	8
2.2.2	Los RNN y LSTM son prometedores:	8
2.2.3	Aparición de Transformers:	8
2.2.4	Uso de capas CTC:	8
2.3	Bases de datos mencionadas	9
2.3.1	IAM	9
2.3.2	NIST	9
2.3.3	UW-3	10
2.3.4	Urdu	10
3	Propuestas de Solución	11
3.1	Propuesta del workflow	11
3.2	Dataset Usado	12
3.3	Propuesta de preprocesamiento de las imágenes	13
3.3.1	Filtros de suavizado	13
3.3.2	Binarización	14
3.3.3	Preprocesamientos utilizados	14
3.4	Propuesta de modelo	16
3.4.1	Modelo Encoder-Decoder	16
3.4.2	Capas de Input	18
3.4.3	Capas convolucionales	18
3.4.4	Capas de pooling	18
3.4.5	Capa de Reshape	18
3.4.6	Capas Densas	18
3.4.7	Capa de Dropout	18
3.4.8	Capas de memoria bidireccional larga a corto plazo (LSTM)	18
3.4.9	Capa de salida	19
3.4.10	Capa CTC (Connectionist Temporal Classification)	19
4	Experimentación y Resultados	20
4.1	Métricas (WER y CER)	20
4.2	Distribución del dataset	20
4.3	Experimentación y discusión de resultados	20
4.3.1	Epochs vs Loss	21
4.3.2	Cambiando tasa de aprendizaje	22
4.3.3	Aumentando el tamaño de los lotes	22

4.3.4	¿Cómo afectaron los preprocesamientos a los resultados obtenidos?	23
4.3.5	Ejemplificación	24
4.4	Comparaciones	25
4.4.1	Random	25
4.4.2	Gemini	25
5	Conclusiones	28
5.1	Desafíos enfrentados	28
5.2	Trabajo Futuro	28
6	Bibliografía	29

1 Introducción

1.1 Motivación

La conservación del conocimiento histórico es una tarea fundamental para comprender nuestro pasado y guiar nuestro futuro. Los documentos manuscritos históricos, en particular, representan una fuente invaluable de información sobre diversas culturas, eventos y descubrimientos. Sin embargo, estos documentos a menudo se encuentran en condiciones frágiles debido al paso del tiempo, la exposición a elementos y las prácticas de almacenamiento inadecuadas.

Este proyecto tiene como objetivo desarrollar un sistema de OCR (Reconocimiento Óptico de Caracteres) para *transcribir documentos manuscritos históricos pertenecientes a la Sociedad Económica Amigos del País*. La motivación principal reside en la urgente necesidad de preservar este patrimonio cultural antes de que se deteriore irremediablemente.

La transcripción de estos documentos mediante OCR presenta varios beneficios:

- **Preservación digital:** Al convertir los documentos físicos en archivos digitales, se crea una copia duradera que puede almacenarse y consultarse de forma segura, incluso si los documentos originales se deterioran.
- **Acceso mejorado:** Los documentos digitales son fácilmente accesibles a través de internet, lo que permite que investigadores, estudiantes y el público en general exploren y analicen estos valiosos recursos históricos desde cualquier lugar del mundo.
- **Investigación facilitada:** La transcripción de los documentos facilita la búsqueda de información específica, la identificación de patrones y el análisis de contenido, lo que contribuye a una mejor comprensión de la historia y el desarrollo cultural.
- **Conservación a largo plazo:** La digitalización de documentos históricos ayuda a garantizar su conservación a largo plazo, protegiéndolos de daños físicos y preservando su contenido para las generaciones futuras.

En resumen, este proyecto de OCR para documentos manuscritos históricos se enmarca en un esfuerzo crucial para salvaguardar nuestro patrimonio cultural y garantizar su accesibilidad para las generaciones presentes y venideras. La digitalización de estos documentos no solo preserva su contenido físico, sino que también abre nuevas posibilidades para la investigación, la educación y la difusión del conocimiento histórico.

1.2 Problemática

La transcripción de documentos manuscritos históricos cubanos de la Sociedad Amigos del País del siglo XVIII mediante OCR presenta desafíos particulares que se suman a las problemáticas generales descritas anteriormente:

1.2.1 Características Específicas del Lenguaje:

1. Los documentos de esta época utilizan un español antiguo con ortografía, gramática y vocabulario diferentes al español actual, lo que puede dificultar el reconocimiento de palabras por parte de los sistemas de OCR convencionales.
2. La presencia de términos específicos relacionados con la Sociedad Amigos del País, la cultura y la historia cubana del siglo XVIII requiere un conocimiento especializado del contexto para una correcta interpretación.

1.2.2 Deterioro y Variaciones en la Escritura:

1. Los documentos de este período pueden presentar un grado aún mayor de deterioro debido a su antigüedad, incluyendo daños físicos, manchas, decoloración y fragilidad del papel.
2. La escritura manual del siglo XVIII puede ser más caligráfica y ornamentada que la escritura moderna, lo que aumenta la complejidad del reconocimiento de caracteres individuales.

1.2.3 Escasez de Datos de Entrenamiento:

1. La cantidad de datos de entrenamiento disponibles para el OCR de documentos manuscritos cubanos del siglo XVIII es extremadamente limitada debido a la rareza de este tipo de material.
2. La falta de datos de entrenamiento de alta calidad específicos para este contexto histórico puede afectar significativamente la precisión del OCR, especialmente en la identificación de palabras y frases poco comunes.

1.2.4 Relevancia Histórica y Cultural:

1. La transcripción precisa de estos documentos es crucial para preservar y difundir el conocimiento sobre la Sociedad Amigos del País, su papel en el desarrollo de Cuba y las ideas de la Ilustración en la isla.
2. Errores en la transcripción pueden llevar a interpretaciones erróneas de la historia y obstaculizar la comprensión del legado cultural de este período.

En resumen, la transcripción de documentos históricos cubanos de la Sociedad Amigos del País del siglo XVIII mediante OCR exige un enfoque especializado que considere las características lingüísticas, paleográficas e históricas únicas de este tipo de material. La escasez de datos de entrenamiento y la relevancia cultural de estos documentos hacen de este un desafío complejo que requiere soluciones innovadoras y adaptadas a las necesidades específicas de este contexto.

Abordar estas problemáticas de manera efectiva permitirá rescatar y preservar este valioso patrimonio documental para las generaciones presentes y futuras, contribuyendo a un mejor entendimiento de la historia de Cuba.

1.3 Objetivos generales y específicos

1.3.1 Objetivo General

Desarrollar un sistema de OCR (Reconocimiento Óptico de Caracteres) robusto y preciso para transcribir documentos manuscritos históricos cubanos de la Sociedad Amigos del País del siglo XVIII, con el fin de preservar y difundir este patrimonio cultural invaluable. La no preparación de un dataset adecuado con estos documentos nos impide trabajar con ellos directamente, por lo que se va a trabajar con un dataset de textos similares para hacer más fácil la reutilización de los resultados de esta investigación en un futuro.

1.3.2 Objetivos Específicos

1. Preprocesar las imágenes digitales para mejorar su calidad y facilitar el reconocimiento de caracteres.
2. Investigar sobre las técnicas de OCR que se han utilizado en el mundo y encontrar el modelo con mejores resultados para este problema en específico.
3. Evaluar el modelo utilizando métricas que permitan medir la eficiencia de este en la transcripción de los textos del dataset.
4. Comparar los resultados obtenidos con otros modelos como lo son Gemini y una función random.

2 Estado del Arte

El Reconocimiento Óptico de Caracteres (OCR) para documentos históricos es un campo de investigación en constante evolución, con el objetivo de transcribir con precisión textos manuscritos e impresos de diferentes épocas y estilos. El estado del arte en esta área abarca diversos aspectos, desde técnicas de preprocesamiento de imágenes y algoritmos de segmentación de texto, hasta el reconocimiento de caracteres y el post-procesamiento de las transcripciones.

2.1 Técnicas de preprocesamiento de imágenes

Excepto algún caso muy especial, las imágenes de un dataset de texto manuscrito necesitan pasar por una capa de preprocesamiento de imágenes, antes de pasar al conjunto de entrenamiento del modelo.

2.1.1 Procedimiento general

Hay muchas técnicas y el pipeline del preprocesamiento de imágenes varía de investigación a investigación pero en [1] y [2] se define el procedimiento más general que se lleva a cabo:

1. **Eliminación de ruido:** En la gran mayoría de los casos las imágenes contienen diferentes tipos de ruido. El ruido se define en una imagen como una deformación que afecta la calidad de la imagen, y una forma común de manifestarse es mediante granos o puntos de colores, los cuales no aportan información, en el caso de los textos antiguos una forma común de ruido son los rayones en la hoja y otras manchas producto del paso de los años. La manera de lidiar con esto son los llamados filtros de suavizado o filtros de Kernel, los cuales están hechos para eliminar distintos tipos de ruidos usando distintos tipos de desenfoques, y se aplican a la imagen llevada a escala de grises. Algunos de los filtros más utilizados son los siguientes:
 - *Filtro Gaussiano:* Utiliza una función gaussiana para ponderar los valores de píxeles vecinos, conservando bordes mientras suaviza la imagen.
 - *Filtro bilateral:* Combina información espacial y de intensidad para suavizar sin perder bordes.
 - *Filtro de mediana:* Reemplaza cada pixel con la mediana de sus vecinos. Es efectivo para eliminar ruido impulsivo.
 - *Filtro de Media:* Calcula el promedio de los valores de píxeles en una ventana local. Es muy útil en la reducción de pequeños detalles innecesarios en la imagen.
2. **Segmentación:** Es el proceso de dividir una imagen en regiones o segmentos, más concretamente en el contexto del reconocimiento de caracteres se usa para separar una imagen de texto en varias imágenes de líneas de texto o separar a su vez las imágenes de líneas de texto en imágenes de palabras. La segmentación además incluye la separación del fondo de la imagen con el texto.
3. **Binarización:** Un paso casi que obligatorio para cualquier investigación de este tipo. La binarización consiste en una función que se aplica a la imagen con un umbral predeterminado, los valores de píxeles por debajo del umbral son cambiados a 0, mientras que los valores que están por encima se cambian a 255, de esta manera la imagen resultante idealmente tiene todo el texto de color negro y el resto de la imagen es blanca, sin ningún otro valor de píxeles. Por supuesto, escoger el mejor umbral para la binarización depende por completo del problema en específico, aunque existen variantes de umbrales adaptativos, los cuales seleccionan automáticamente el mejor umbral para cada imagen, y en algunos casos para cada región de una imagen.
4. **Normalización:** Es el proceso de adaptar el tamaño de las imágenes o el texto a un tamaño que sea aceptable para el modelo a utilizar. Por supuesto este paso es un poco más abstracto ya que depende enteramente del modelo que se vaya a utilizar.
5. **Clasificación:** Es el proceso en que las imágenes procesadas son finalmente enviadas al modelo de OCR.

Hay otra técnica de preprocesamiento muy utilizada pero que no se menciona en estas investigaciones, la **Dilatación**. Es una operación morfológica que añade píxeles a los límites de los objetos en una imagen. Se usa

para hacer los objetos de una imagen más visibles y el procedimiento generalmente consiste en comparar cada píxel de la imagen con un entorno definido por un elemento estructurante, que generalmente es una matriz o una ventana.

Esto define el procedimiento general de muchas investigaciones para la parte del preprocesamiento de imágenes, pero la manera de llevarlo a cabo depende mucho de factores como el modelo, el dataset, o puede que incluso el idioma. Por ejemplo en [3] se utilizan datasets de *Urdu*, un idioma hablado principalmente en India y Pakistan, y por la forma del dataset solo fue necesario aplicar procesos de binarización y normalización de los datos para su modelo.

2.1.2 Combinando técnicas

Por otra parte, el orden del preprocesamiento no tiene que ser el descrito anteriormente, incluso, cada paso no tiene por que ser independiente del resto. En [4], luego de algunos ajustes de resolución, se sometieron las imágenes del dataset a una segmentación por líneas y, posteriormente, a una segmentación por palabras. Pero el proceso de segmentar en líneas incluye a su vez su propio preprocesamiento, se lleva la imagen a escala de grises, luego se binariza en invierte, quedando una imagen de fondo negro y texto blanco. Posteriormente se dilata la imagen y se utilizan métodos para hallar cuadros delimitadores (Bounding Boxes como aparece en la literatura) los cuales no son más que el área de la imagen que abarca un objeto en específico, una línea de texto en este caso, y sobre estos entonces se segmenta en líneas la imagen original. Para segmentar en palabras se lleva a cabo un proceso similar, con la diferencia de que se aplican filtros de kernel en un primer paso ya que, dado que las palabras son objetos más grandes en una línea, que una línea en todo el texto, es necesario que la dilatación sea más descriptiva para hallar los cuadros delimitadores, cosa que se obtiene aplicando dilatación sobre una imagen previamente desenfocada.

2.1.3 Otras técnicas

También hay investigaciones que utilizan otras técnicas como en [5]. En esta investigación se utilizó un algoritmo conocido como *slant method correction* una técnica de preprocesamiento de imágenes propuesta por los científicos Changming Sun y Deyi Si en el paper “*Skew and slant correction for document images using gradient direction*”[6]. Este método consiste en un algoritmo rápido para corregir la inclinación y la desviación en imágenes de documentos impresos utilizando únicamente información del gradiente. Se calcula la orientación del gradiente en la imagen en escala de grises y luego se busca un pico en el histograma de las orientaciones de gradiente para determinar el ángulo de inclinación. Por último corrige la inclinación y la desviación aplicando transformaciones geométricas.

En resumen, es necesaria una capa de preprocesamiento de imágenes antes de pensar en los modelos de reconocimiento de caracteres. Las distintas investigaciones consultadas dan ideas y formas de aplicar las técnicas existentes, pero siempre la manera más efectiva depende de la experimentación que se haga con el problema en específico en cuestión.

2.2 Reconocimiento de caracteres

Los idiomas, por naturaleza, varían en la forma de sus letras y en las conexiones de las palabras. El proceso de reconocimiento de texto escrito a mano es complicado debido a muchos impedimentos, como la variedad de estilos de escritura, la mala calidad de los documentos, el ruido, las manchas en el papel y la alineación del texto.

Los enfoques de aprendizaje profundo han revolucionado el reconocimiento de texto escrito a mano (HTR) en los últimos años. En la bibliografía consultada, se vieron propuestas utilizando varios tipos de modelos como CNN, SVM, LSTM, CRNN, BLSTM, Transformers y un sinfín de combinaciones de estas.

2.2.1 CNN

Las redes neuronales convolucionales (CNN) son un pilar de HTR y demuestran una capacidad excepcional para extraer características de imágenes de personajes escritas a mano. Estudios como [7] lograron una precisión del 98,85 utilizando CNN. De manera similar, [8] combinó CNN con RNN para lograr una tasa de error de 0,4 en el dataset UW3. En [2], se logró una accuracy de 94% usando solamente CNN en el dataset NIST.

2.2.2 Los RNN y LSTM son prometedores:

Las redes neuronales recurrentes (RNN), en particular las redes de memoria a corto plazo (LSTM), están ganando terreno en HTR debido a su capacidad para manejar datos secuenciales como texto escrito a mano. En [5] se logró una tasa de error de palabras (WER) del 10,5 y una tasa de error de caracteres (CER) del 3,6 utilizando una arquitectura MDLSTM especializada. Mientras, [3] empleó LSTM bidireccionales para reconocer texto en cursiva Urdu, logrando un CER del 16,31. En otras investigaciones como [8] y [4], fueron usados LSTM junto a capas CNN y CTC para lograr una tasa de error de 0.4% y un WER de 10.62% respectivamente.

2.2.3 Aparición de Transformers:

Aunque son menos frecuentes, las arquitecturas basadas en Transformers están dejando una huella en HTR. Estos modelos destacan por capturar dependencias de largo alcance dentro del texto, lo que es potencialmente valioso para tareas de reconocimiento complejas. Por ejemplo [9] utilizó transformers, logrando un CER del 5,07 y un WER del 21,47 en el conjunto de datos IAM.

2.2.4 Uso de capas CTC:

Las capas CTC (Connectionist Temporal Classification) son un tipo de capa de red neuronal recurrente (RNN) que se utilizan comúnmente en tareas de reconocimiento de voz, pero también tienen un gran potencial para ser utilizadas en el Reconocimiento Óptico de Caracteres (OCR). Las capas CTC pueden ser útiles para esta tarea porque pueden modelar las dependencias temporales entre los caracteres en una imagen. Esto significa que pueden tener en cuenta el orden de los caracteres a la hora de realizar la predicción, lo que puede mejorar la precisión del reconocimiento. Los resultados de usar capas CTC en el modelo pueden ser apreciados en [4] con un WER de 10.62 y una accuracy de 98% y en [4] donde se obtuvo una tasa de error de 0.4% haciendo uso de capas CTC.

Los desafíos persisten: A pesar de los avances, persisten los desafíos en el manejo de escenarios específicos. Los textos históricos escritos a mano, a menudo degradados y con estilos de escritura únicos, pueden resultar difíciles para los modelos actuales. La investigación llevada a cabo en [9] lo demuestra, logrando una menor precisión en comparación con los conjuntos de datos modernos. De manera similar, el reconocimiento de la escritura a mano en cursiva sigue siendo un área de investigación en curso, como lo muestra el CER del 16,31 para el texto en Urdu en [3].

En conclusión, las técnicas de aprendizaje profundo como CNN, RNN y Transformers están impulsando avances en la precisión de HTR. Si bien los modelos actuales logran resultados impresionantes en conjuntos de datos estándar, se necesita más investigación para abordar los desafíos asociados con los documentos históricos y los diversos estilos de escritura.

2.3 Bases de datos mencionadas

En la sección anterior, se mencionaron cuatro bases de datos de documentos que son los mencionados en la bibliografía. A continuación presentamos las características principales de cada dataset y una ejemplificación.

2.3.1 IAM

El dataset IAM [10] de documentos manuscritos es una colección a gran escala de imágenes de documentos manuscritos con sus respectivas transcripciones. El dataset contiene más de 300,000 imágenes de documentos escritos a mano en inglés, alemán y francés. Las imágenes provienen de una variedad de fuentes, incluyendo cartas, diarios, registros históricos y formularios. Las imágenes del dataset IAM de documentos manuscritos varían en calidad desde imágenes escaneadas de alta resolución hasta fotografías de baja resolución. La mayoría de las imágenes son en blanco y negro, pero algunas son en color.



Figure 1: dataset IAM

2.3.2 NIST

El dataset NIST [11] de documentos manuscritos es una colección de formularios manuscritos rellenos a mano. El dataset contiene 6,300 formularios de dos tipos:

- Formulario SF-2808: Este formulario es una solicitud de número de identificación de empleado federal.
- Formulario W-4: Este formulario es un formulario de retención de impuestos federales.

Los formularios están escritos a mano en inglés y provienen de una variedad de fuentes, incluyendo empleados del gobierno federal, contratistas y jubilados.

Los formularios del dataset incluyen una variedad de estilos de escritura, desde escritura cursiva formal hasta escritura a mano informal. Esto hace que el dataset sea desafiante para los modelos de reconocimiento de escritura a mano, pero también lo hace más realista y representativo de los documentos manuscritos del mundo real.

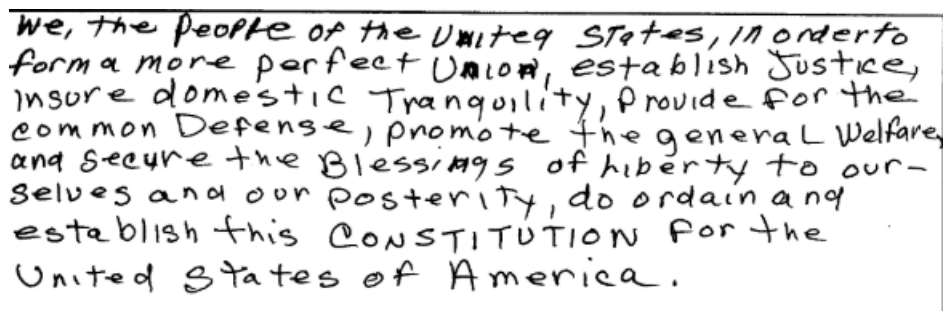
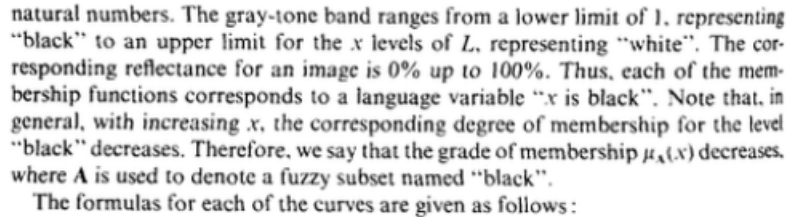


Figure 2: dataset NIST

2.3.3 UW-3

El conjunto de datos original consta de 1600 imágenes de documentos en inglés con corrección de páginas torcidas y cuadros delimitadores (bounding boxes) de entidades editados manualmente. Estos cuadros delimitadores encierran marcos de página, zonas de texto y sin texto, líneas de texto y palabras. También está marcado el tipo de cada zona (texto, matemáticas, tabla, medios tonos,...). Hay alrededor de 120 imágenes de documentos que contienen al menos una zona de tabla marcada.



natural numbers. The gray-tone band ranges from a lower limit of 1, representing "black" to an upper limit for the x levels of L , representing "white". The corresponding reflectance for an image is 0% up to 100%. Thus, each of the membership functions corresponds to a language variable " x is black". Note that, in general, with increasing x , the corresponding degree of membership for the level "black" decreases. Therefore, we say that the grade of membership $\mu_A(x)$ decreases, where A is used to denote a fuzzy subset named "black".

The formulas for each of the curves are given as follows :

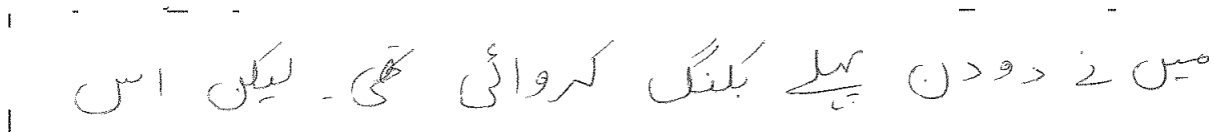
Figure 3: dataset UW-3

Como se puede observar, los documentos pertenecientes a este dataset no son manuscritos, sino que son mecanográficos.

2.3.4 Urdu

En la bibliografía mencionada [3] se hace uso de una mezcla de varios datasets, entre los que se encuentran IFN/ENIT [12], QUWI [13], KHATT [14], UCOM [15], UNHD [16]. En total, la base de datos usada en este paper consiste de:

1. 600 escritores
2. 6000 líneas totales
3. 86 400 palabras
4. 432 000 caracteres



میں نے دودن پہلے بکنگ کروائی تھی۔ لیکن اس

Figure 4: dataset Urdu

3 Propuestas de Solución

3.1 Propuesta del workflow

A continuación se enumeran las etapas seguidas para la elaboración de este proyecto:

1. Recolección y Preparación de Datos:

- **Recopilación de Datos:** Se logró reunir un conjunto de datos de documentos manuscritos en italiano cuyas características se asemejan lo suficiente al dataset de documentos históricos cubanos.
- **Preprocesamiento de Imágenes:** Se mejoró la calidad de las imágenes del conjunto de datos mediante técnicas las cuales van a ser discutidas en la próxima sección.

2. Modelo:

- **Selección del Modelo:** Una vez consultada la bibliografía y estudiado el estado del arte, se eligió el modelo de aprendizaje automático a usar. Este va a ser discutido en las próximas secciones.
- **Arquitectura del Modelo:** Se diseñó la arquitectura de la red neuronal que se ajusta a la tarea específica. Esto implicó definir el número de capas, el tipo de activaciones y las conexiones entre las neuronas. Cada capa usada, al igual que el objetivo de su uso va a ser explicado próximamente.
- **Entrenamiento del Modelo:** Se entrenó el modelo de aprendizaje automático utilizando los datos de entrenamiento preprocesados y etiquetados. Esto implicó ajustar los hiperparámetros del modelo para minimizar el error entre las predicciones del modelo y el texto de referencia.

3. Evaluación y Mejora del modelo:

- **Evaluación del Modelo:** se evaluó el rendimiento del modelo entrenado en un conjunto de datos de prueba independiente. Se usaron métricas como la accuracy, WER (Word Error Rate) y CER (Character Error Rate).

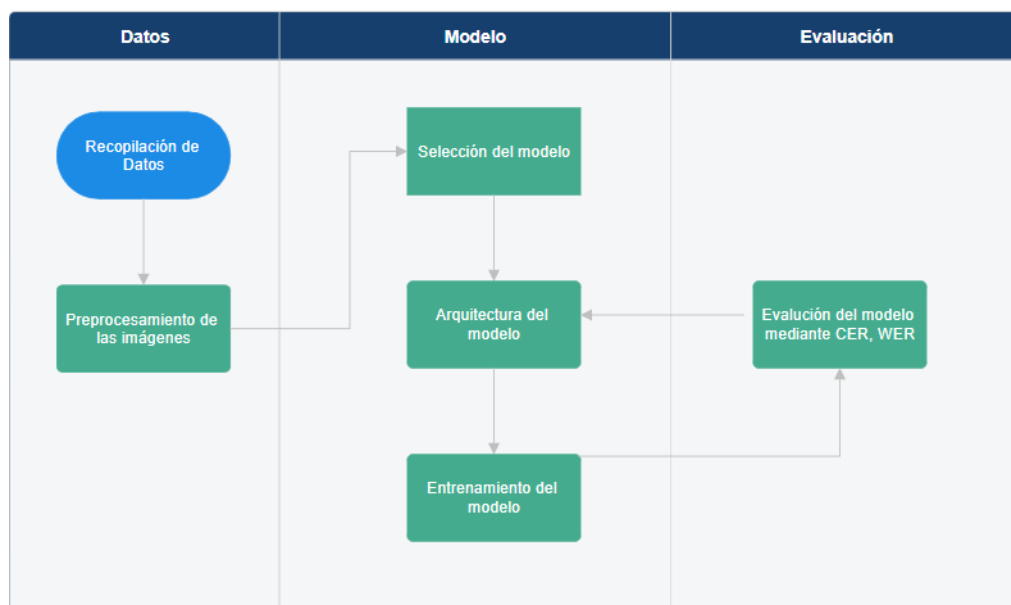


Figure 5: Propuesta del workflow

3.2 Dataset Usado

Para el entrenamiento de esta red neuronal, se decidió hacer uso del dataset LAM: Handwritten Italian Dataset [17].

El conjunto de datos de Ludovico Antonio Muratori (LAM) es el conjunto de datos HTR (Reconocimiento de texto manuscrito) a nivel de línea más grande hasta la fecha y contiene 25.823 líneas de manuscritos antiguos italianos editados por un solo autor durante 60 años.

Se decidió hacer uso de este dataset por varias razones:

- **Semejanza con el dataset de documentos manuscritos cubanos:** Esta es la razón principal por la que se eligió este dataset. Se eligió un dataset en idioma italiano debido a la cercanía entre los idiomas español e italiano ya que ambas lenguas provienen del latín. Otro aspecto que fue decisivo para la elección de este dataset es el estado de los documentos y la legibilidad de la letra; ya que el conjunto de documentos manuscritos cubano se encuentra en muy mal estado de conservación, se intentó buscar un dataset que se asemejara a esas condiciones.
- **Tamaño del conjunto de datos:** Como ya se mencionó, es uno de los conjuntos de datos de manuscritos más extensos que existen.
- **Calidad de las imágenes:** Las imágenes del conjunto de datos son de alta calidad y están cuidadosamente etiquetadas. Esto las hace ideales para entrenar modelos de aprendizaje automático.
- **Acceso gratuito:** El conjunto de datos es gratuito y de código abierto. Esto lo hace accesible para cualquier persona que quiera trabajar con él.

A continuación, se presentan ejemplos del conjunto de datos. Estos ejemplos han sido seleccionados para ilustrar la diversidad y las condiciones de los elementos del dataset.

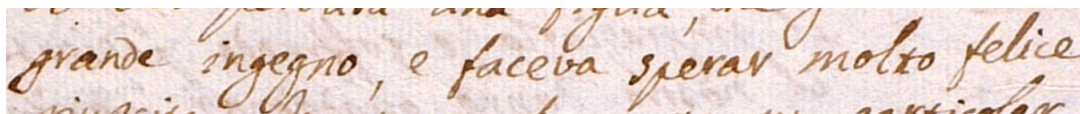


Figure 6: Aquí se demuestra la colisión de las letras en las líneas superiores e inferiores

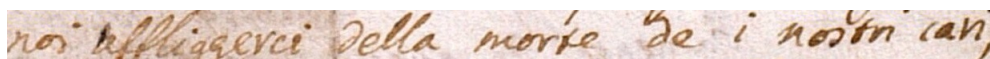


Figure 7: Aquí se muestra como el color del papel se puede mezclar con la escritura y se complejiza la lectura de estas

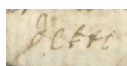


Figure 8: Aquí de igual manera se puede apreciar el ruido de la foto

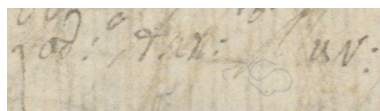


Figure 9: Aquí se puede apreciar el ruido de la foto haciendo la escritura casi ilegible

3.3 Propuesta de preprocesamiento de las imágenes

Como se vió anteriormente, no hay una manera única de llevar a cabo un preprocesamiento de imágenes en un problema de este tipo, por lo que se decidió experimentar con las herramientas que brinda la bibliografía consultada, y otras más que se encontraron a lo largo de la investigación, para encontrar las mejores formas de preprocesamiento de cara al dataset escogido.

3.3.1 Filtros de suavizado

Aparte de los filtros de suavizado vistos en la sección del estado del arte (Filtro Gaussiano, Filtro de Media, Filtro de Mediana y Filtro Bilateral) fueron utilizados otros 2 filtros para la experimentación:

- **Filtro Laplaciano:** Especializado en la detección de bordes. Se basa en la segunda derivada espacial de la imagen, lo que significa que resalta las áreas donde hay cambios rápidos de intensidad.
- **Filtro de sal y pimienta (Salt and Pepper):** El ruido sal y pimienta se refiere a una forma de ruido visual que se manifiesta como píxeles blancos y negros dispersos, o de alto y bajo brillo, que aparecen de manera aleatoria en la imagen. Un filtro de Sal y pimienta se especializa en eliminar este tipo de ruido en específico.

Mostrar como afectan estos filtros a las imágenes del dataset no es muy útil usando solo las imágenes originales, o incluso llevadas a escala de grises, ya que no se aprecia una diferencia marcada entre cada uno, por lo tanto se va a aplicar una binarización con umbral adaptativo (se explicará esta técnica más adelante) luego de aplicar el filtro para poder definir cuál es el más útil, ya que de esta manera si se aprecian mejor los cambios.



Figure 10: Filtro Gaussiano



Figure 11: Filtro de Mediana



Figure 12: Filtro de Media



Figure 13: Filtro Bilateral



Figure 14: Filtro Laplaciano



Figure 15: Filtro de Sal y Pimienta

Evidentemente el filtro con mejor resultado es el bilateral, ya que es el que elimina mayor cantidad de ruido de la imagen. Por tanto, este será el que se usará en las propuestas de preprocesamiento que se enunciarán más adelante.

3.3.2 Binarización

Como se dijo anteriormente, este es posiblemente el paso más importante del preprocesamiento, sin embargo el primer problema que surgió aquí fue el tema de escoger el mejor umbral para la binarización. Al igual que otros parámetros que utilizan la mayoría de filtros en procesamiento de imágenes, la mejor manera de encontrar el mejor valor para este umbral es mediante la experimentación, probar con varios valores de umbrales y elegir el que dé mejor resultado, sin embargo, el problema aquí es la propia composición del dataset. En la sección anterior aparecen ejemplos de como a veces no hay distinción entre el color de la hoja y el de una sección de texto, o como algunas veces este es casi ilegible. Esto hace prácticamente imposible encontrar un valor de umbral en el que se logre un resultado óptimo para todas las imágenes, por lo que se hizo necesario buscar una alternativa: La binarización con umbral adaptativo.

La binarización con umbral adaptativo tiene el mismo objetivo que la binarización con umbral global (O sea fijo), la diferencia está en que esta ajusta el umbral para cada píxel basándose en la información local de la imagen. Su objetivo es tratar de manera más efectiva las variaciones de iluminación en diferentes partes de la imagen. El procedimiento, de manera resumida, consiste en definir un tamaño de ventana o Kernel y luego para cada píxel en la imagen se calcula un valor de umbral basado en las estadísticas de los píxeles dentro de la ventana. Esto puede ser la media, la mediana, o cualquier otra medida que refleje las características locales de la imagen. Hay una notable diferencia entre aplicar ambos tipos de binarización al dataset:

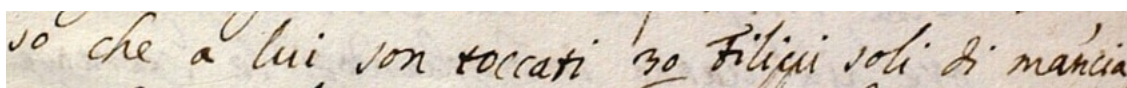


Figure 16: Imagen original del dataset



Figure 17: Imagen luego de una binarización con umbral global



Figure 18: Imagen luego de una binarización con umbral adaptativo

Como se puede observar, usando el umbral global el final de la línea de texto se vuelve bastante ilegible, mientras que el umbral adaptativo da un resultado mucho más aceptable. Aún así, el enfoque del preprocesamiento usando una binarización con umbral global no se va a descartar todavía.

3.3.3 Preprocesamientos utilizados

A partir de los filtros enunciados anteriormente, la binarización con distintos umbrales y otras técnicas como la dilatación (explicada en secciones anteriores), se probaron varias combinaciones para componer distintos pipelines de preprocesamiento, de los cuales se eligieron los 4 que mejor resultado devolvieron. El resto de combinaciones de técnicas o bien no eran factibles, o devolvían resultados desfavorables, o bien no devolvían un resultado lo suficientemente diferente de los ya obtenidos como para considerarlos.

- **Procesamiento 1 (Umbral adaptativo básico):** Se lleva la imagen a escala de grises y luego se le aplica un filtro de suavizado bilateral para eliminar ruido. Posteriormente se aplica una binarización con umbral adaptativo a la imagen resultante.
- **Procesamiento 2 (Umbral adaptativo y dilatación):** Se lleva a cabo el mismo proceso que en el procesamiento 1, pero luego de la binarización se le aplica una dilatación a la imagen resultante.
- **Procesamiento 3 (Umbral adaptativo y sal y pimienta):** Posteriormente se va a ilustrar como luce una imagen del dataset luego de ser sometida a cada uno de estos procesamiento, pero como adelanto, un problema que tienen los 2 procesamiento anteriores es que algunos píxeles aislados logran pasar el umbral de la binarización, por lo que en este procesamiento vamos a intentar eliminarlos utilizando un filtro de sal y pimienta luego de aplicar el procesamiento 1, es decir, luego de llevar a escala de grises, aplicar un filtro bilateral y utilizar una binarización con umbral adaptativo.
- **Procesamiento 4 (Umbral global):** Como su nombre lo indica, aquí no se utilizará un umbral adaptativo, sino un umbral global para la binarización, luego de llevar la imagen a escala de grises y aplicar un filtro bilateral. El valor del umbral fue elegido luego de probar con varios umbrales y, aunque sigue sin ser un umbral que devuelva el mejor resultado para la totalidad de las imágenes del dataset, si es cierto que brinda un enfoque distinto para atacar el problema, por lo que se decidió que sería el modelo quien diría la última palabra sobre la efectividad de esta técnica de procesamiento.



Figure 19: Imagen original del dataset



Figure 20: Procesamiento 1 (Umbral adaptativo básico)



Figure 21: Procesamiento 2 (Umbral adaptativo y dilatación)



Figure 22: Procesamiento 3 (Umbral adaptativo y sal y pimienta)



Figure 23: Procesamiento 4 (Umbral global)

El modelo se entrenará por separado con cada uno de estos preprocesamientos, y se analizará la efectividad que tuvo con cada uno más adelante.

3.4 Propuesta de modelo

Luego de haber consultado la bibliografía mencionada, decidimos implementar un modelo que combine capas convolucionales, recurrentes de tipo LSTM y una capa CTC, ya que esta combinación es la que mostró mejores resultados en documentos manuscritos con letra torcida.

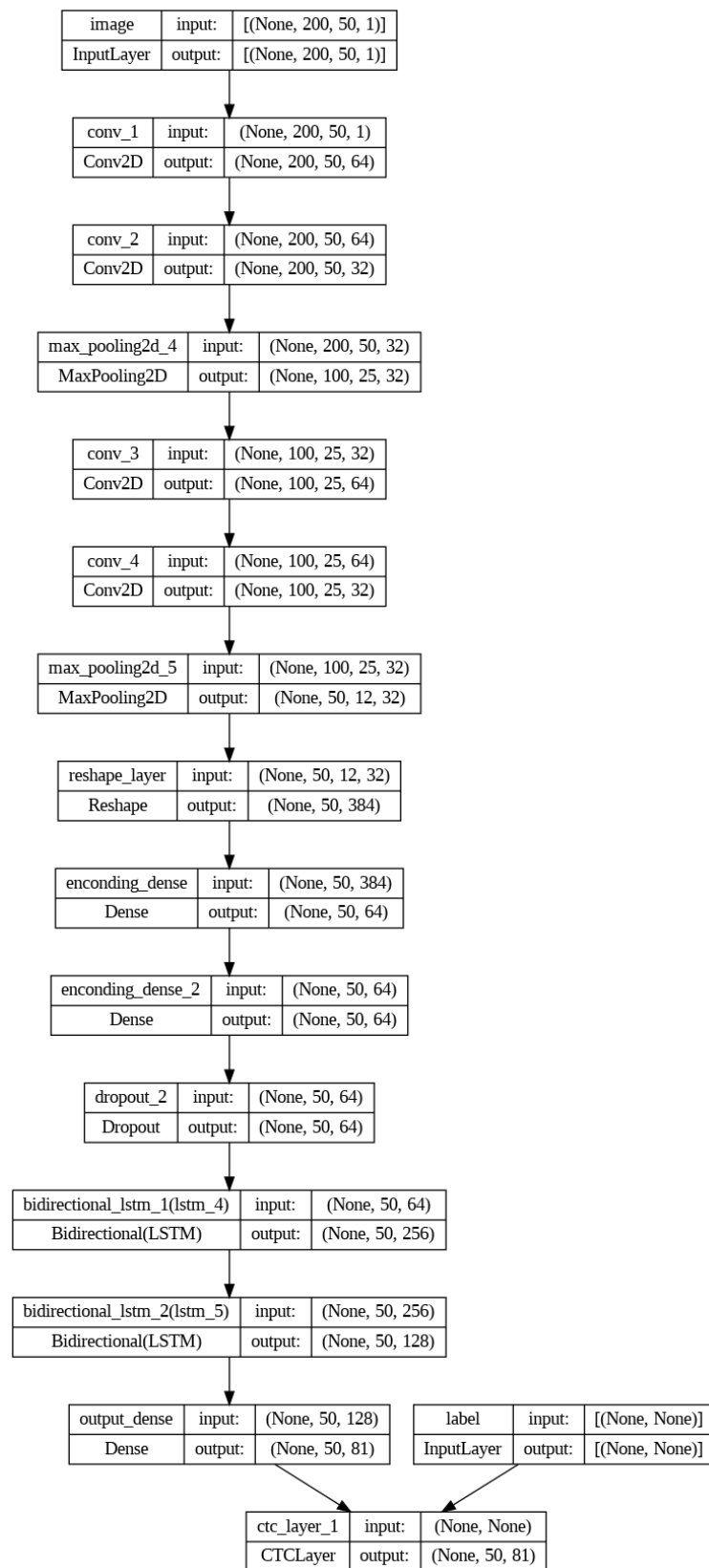
3.4.1 Modelo Encoder-Decoder

El modelo que se propone es un modelo encoder-decoder[18][19] o Seq2Seq. Esta consiste de 2 partes : el codificador y el decodificador. El codificador procesa una secuencia de input y produce un grupo de vectores contextuales, los cuales son usados por el decodificador. La idea de este modelo es que se pueda recibir los datos en un formato (imagen) y convertirlo a otro formato (texto).

- **Encoder:** El encoder se encarga de procesar la imagen de entrada y transformarla en una representación vectorial que captura las características del texto embebido en la imagen. Para ello, se emplea una red convolucional profunda, que extrae características locales de la imagen a través de convoluciones y funciones de agrupación. Estas características se combinan luego en capas posteriores para obtener una representación global más abstracta de la imagen.
- **Decoder:** El decoder, por otro lado, toma la representación vectorial generada por el encoder y la utiliza para predecir la secuencia de caracteres que conforman el texto presente en la imagen. El decoder en este caso es una red neuronal recurrente, en nuestro caso hacemos uso de capas bidireccionales de LSTM, que son capaces de procesar secuencias de información de manera efectiva. A medida que el decoder procesa la representación vectorial, genera predicciones de caracteres uno a la vez, construyendo gradualmente la secuencia de texto completa.

Una limitación asociada con el uso de una arquitectura encoder-decoder son los problemas de preprocesamiento de datos. Dado que la mayoría de las aplicaciones modernas requieren conjuntos de datos de entrada en formatos numéricos en lugar de texto sin formato o imágenes, esto significa que se deben tomar medidas adicionales antes de que pueda comenzar cualquier capacitación. En nuestro caso, como el modelo recibe dos inputs (imagen y su transcripción), cada uno de estos datos deben ser convertidos en valores numéricos que sean representativos; de lo contrario, el sistema simplemente estaría adivinando aleatoriamente sin ningún contexto mientras intenta encontrarle sentido a la secuencia de entrada proporcionada por su usuario/programador. Como tal, tener especial cuidado durante este paso es esencial para garantizar resultados precisos una vez que comienza el entrenamiento, algo que hace que el uso de encoders-decoders sea un poco más complicado en comparación con otros sistemas de IA disponibles en la actualidad.

El modelo implementado luce de la siguiente manera:

**Figure 24:** Modelo implementado

Aclarar que una dimensión en 'None' significa que esta es variable. El primer parámetro de las capas

neuronales representan el tamaño del batch, el cual es variable.

3.4.2 Capas de Input

Tenemos dos capas de entrada:

1. **imagen:** Esta capa toma la imagen de entrada como un tensor 4D. La primera dimensión ("None") representa el tamaño del lote, que puede variar. Las siguientes tres dimensiones representan la altura de la imagen (200 píxeles), el ancho (50 píxeles) y el número de canales (1 para escala de grises).
2. **label:** Esta capa agregada durante el entrenamiento toma la etiqueta de texto real (secuencia de caracteres) para la imagen. Su dimensión es ('None', 'None'), donde el primer 'None' representa el tamaño del lote y el segundo 'None' representa la longitud variable de la secuencia de texto, ambas pueden variar.

3.4.3 Capas convolucionales

Nuestro modelo cuenta con 4 capas convolucionales.

conv_1, conv_2, conv_3, conv_4: Estas capas son los encargados de la extracción de características. Aplican filtros a la imagen de entrada, identificando bordes, formas y patrones que son relevantes para el reconocimiento de caracteres. Cada capa genera un nuevo mapa de características con un tamaño reducido (tanto alto como ancho) en comparación con la capa anterior.

La cantidad de filtros (64, 32 en este caso) determina la cantidad de mapas de características producidos por cada capa convolucional. Más filtros permiten capturar una variedad más amplia de funciones.

3.4.4 Capas de pooling

max_pooling2d, max_pooling2d_1: Estas capas realizan una reducción de resolución en los mapas de características. Toman el valor máximo de una pequeña región rectangular (ventana) en la entrada, lo que reduce efectivamente el tamaño de la imagen y conserva las características más destacadas. Esto ayuda a controlar la complejidad del modelo y reducir el sobreajuste.

3.4.5 Capa de Reshape

reshape_layer: Esta capa toma la salida aplanada de la capa de agrupación y la reforma en un tensor 3D adecuado para alimentar las capas recurrentes.

3.4.6 Capas Densas

enconding_dense, enconding_dense_2: Estas son capas completamente conectadas que procesan aún más las características extraídas. Pueden aprender relaciones no lineales entre las características y contribuir a la representación de características de nivel superior.

3.4.7 Capa de Dropout

dropout: Esta capa elimina aleatoriamente un cierto porcentaje de activaciones (salidas) de la capa anterior durante el entrenamiento. Esto ayuda a evitar que el modelo se sobreajuste a los datos de entrenamiento al obligarlo a aprender características sólidas que no dependen de activaciones específicas.

3.4.8 Capas de memoria bidireccional larga a corto plazo (LSTM)

bidireccional_lstm_1, bidireccional_lstm_2: Estos son los componentes principales para el reconocimiento de secuencias. Los LSTM son un tipo de red neuronal recurrente que puede aprender dependencias entre caracteres en una secuencia. La parte "bidireccional" significa que los LSTM procesan la secuencia en ambas direcciones, capturando información contextual de los caracteres anteriores y siguientes. El resultado de los LSTM representa una comprensión de alto nivel de la secuencia de caracteres en la imagen.

3.4.9 Capa de salida

output_dense: Esta capa transforma la salida final de LSTM en una distribución de probabilidad sobre los caracteres posibles (incluido un carácter en blanco para manejar los espacios entre caracteres). Tiene 81 unidades ya que considera letras mayúsculas y minúsculas, números y algunos signos de puntuación.

3.4.10 Capa CTC (Connectionist Temporal Classification)

ctc_layer: Esta capa está diseñada específicamente para tareas de reconocimiento de secuencias como OCR. Toma tanto la etiqueta fundamental (si se proporciona durante el entrenamiento) como la distribución de probabilidad de salida de la capa anterior y calcula la probabilidad de que la secuencia predicha sea la correcta. Permite que el modelo maneje secuencias de diferentes longitudes y posibles espacios entre caracteres.

4 Experimentación y Resultados

4.1 Métricas (WER y CER)

Word Error Rate (**WER**) y Character Error Rate (**CER**) son métricas comúnmente utilizadas en la evaluación de sistemas del habla o de traducción automática, así como en la evaluación de modelos de OCR. Calculan el número mínimo de inserciones, eliminaciones y sustituciones de: una palabra por otra en el caso del WER, y de un carácter por otro en el CER. Se basan en la distancia de edición o distancia de Levenshtein, pero a diferencia de esta última, el WER se calcula a nivel de palabra en lugar de letra, y la diferencia con CER es principalmente de contexto, es decir, mientras que la distancia de Levenshtein es una medida general de similitud entre cadenas, CER se enfoca en la evaluación de sistemas de reconocimiento de caracteres.

Para calcular estas métricas se utiliza la siguiente fórmula:

$$\text{Métrica} = \frac{S + I + D}{N}$$

Donde:

- **S** representa sustituciones.
- **I** representa inserciones.
- **D** representa eliminaciones.
- **N** es el número total de: palabras en el texto original en el caso de WER, letras en el texto original en el caso de CER.

Se compara el texto generado por el OCR con el texto original, un WER y CER más bajos indican una mayor precisión del modelo en la transcripción.

4.2 Distribución del dataset

En total, nuestro dataset cuenta con 23 868 líneas. Estas fueron separadas en tres conjuntos : el conjunto de entrenamiento, el conjunto de validación y el conjunto de pruebas de la siguiente manera:

- **Train:** El 80% del 80%(15 275) de los elementos del dataset fueron destinados a la etapa de entrenamiento
- **Validación:** El 20% del 80% (3819) de los elementos del dataset fueron destinados a la etapa de validación.
- **Prueba:** El 20% (4774) del total de los elementos del dataset fueron destinados a la etapa de prueba del modelo

4.3 Experimentación y discusión de resultados

El objetivo de esta sección es describir los experimentos realizados para evaluar el rendimiento del modelo de OCR desarrollado. Se probará el modelo en el conjunto de datos de documentos manuscritos discutido anteriormente utilizando los 4 preprocesamientos discutidos para analizar su comportamiento y determinar la configuración óptima para obtener las mejores métricas de WER (Word Error Rate) y CER (Character Error Rate).

Épocas	80		100		150	
Métrica	CER	WER	CER	WER	CER	WER
dataset sin preprocesamiento	0.19	0.51	0.17	0.48	0.19	0.51
procesamiento 1	0.14	0.42	0.14	0.42	0.14	0.42
procesamiento 2	0.24	0.59	0.22	0.57	0.22	0.57
procesamiento 3	0.21	0.55	0.21	0.55	0.21	0.55
procesamiento 4	0.33	0.65	0.31	0.66	0.30	0.65

Table 1: Resultados

Como se puede observar en la sección de Resultados, el dataset que recibió una mejor evaluación generalmente fue el que se sometió al procesamiento 1. Vamos a profundizar los resultados obtenidos de este dataset.

4.3.1 Epochs vs Loss

Primeramente veamos la gráfica de "Epochs vs Loss" para ver como se comporta la pérdida en la etapa de entrenamiento:

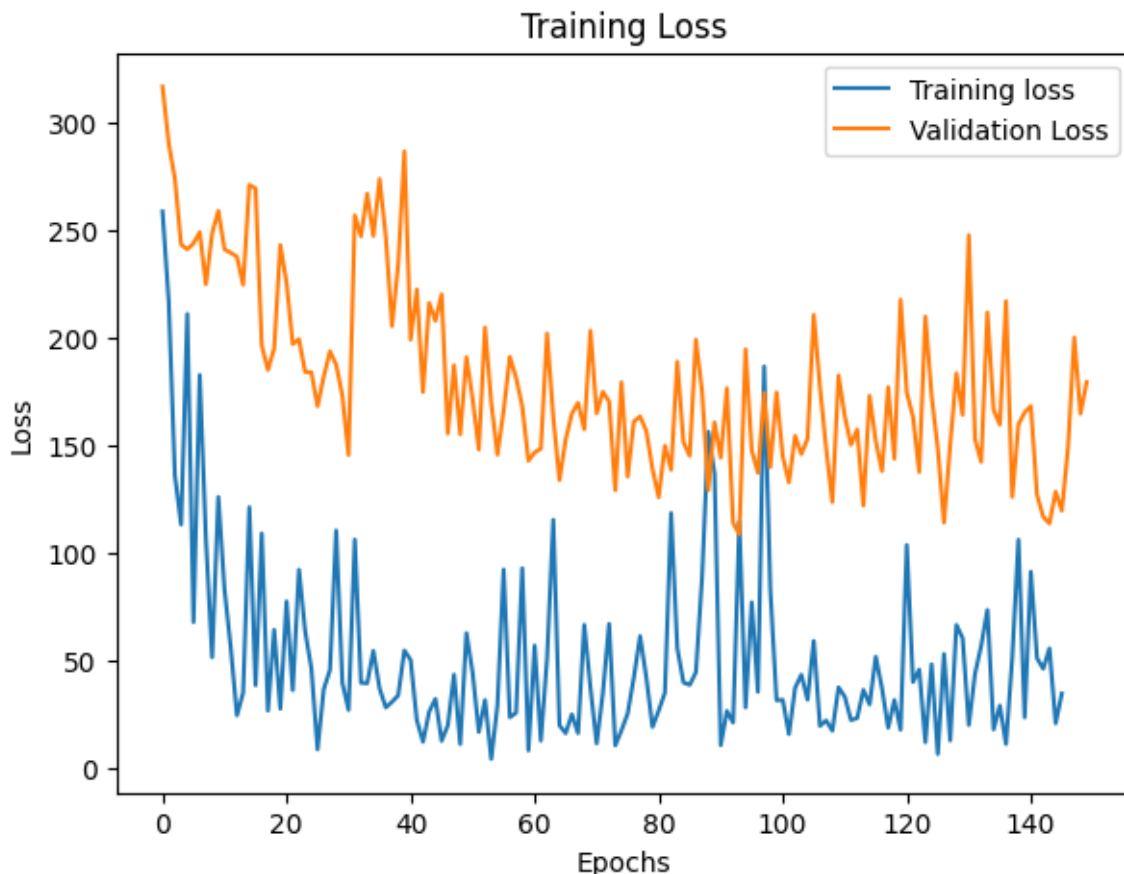


Figure 25: Epoch vs Loss (procesamiento 1)

De esta gráfica se puede observar lo siguiente:

- Pérdida de entrenamiento: La pérdida de entrenamiento parece ser algo errática, con picos a lo largo del entrenamiento. Este comportamiento podría ser debido a varias razones como: entrenamiento en mini lotes, optimización estocástica, fluctuaciones en la tasa de aprendizaje.
- Pérdida de validación: la pérdida de validación también tiene algunos picos, pero en general muestra una ligera tendencia a la baja a lo largo de las épocas. Esto sugiere que el modelo se generaliza bastante bien a datos no vistos a pesar de las fluctuaciones en la pérdida de entrenamiento.
- Brecha entre la pérdida de entrenamiento y la pérdida de validación: la brecha entre las curvas de pérdida de entrenamiento y de pérdida de validación varía a lo largo del entrenamiento. En algunas épocas, la pérdida de validación es incluso menor que la pérdida de entrenamiento. Esto podría ser una señal de desajuste, donde el modelo no aprende bien de los datos de entrenamiento.
- Épocas: Es difícil decir definitivamente cuántas épocas se necesitan basándose únicamente en este gráfico. Idealmente, veríamos que la pérdida de validación continuara disminuyendo y estabilizándose. La Época 80 podría ser un posible punto de parada. La pérdida de validación parece ser relativamente estable en este

punto y continuar con el entrenamiento más allá de esta época podría conducir a un sobreajuste. También es posible que el entrenamiento pueda continuar durante más épocas. La pérdida de validación todavía muestra una ligera tendencia a la baja, por lo que el modelo podría beneficiarse de una capacitación adicional.

4.3.2 Cambiando tasa de aprendizaje

La tasa de aprendizaje es un hiperparámetro que controla cuanto el modelo ajusta su peso basado en la pérdida calculada. En los resultados anteriores, la tasa de aprendizaje se encontraba fija en $1e-3$. Se experimentó aumentándola y disminuyéndola para comprobar si está era la causante de los picos en la gráfica.

Se obtuvieron las siguientes gráficas y resultados:



(a) Learning Rate: $1e-1$
WER:0.47
CER:0.15



(b) Learning Rate: $1e-5$
WER:0.45
CER:0.15

Figure 26: Experimentando con la tasa de aprendizaje

Estos son muy semejantes a los obtenidos con el learning rate de $1e-3$, lo cual sugiere que este factor no es el causante de los picos.

4.3.3 Aumentando el tamaño de los lotes

Otro posible causante de los picos en la gráfica son los mini lotes. En cada época, el modelo es actualizado basado en la pérdida calculada en cada lote. Si hay algún lote que contenga algún outlier o un ejemplo difícil de aprender, la pérdida para esa época puede ser representada como un pico.

En la experimentación pasada, los lotes estaban fijos de tamaño 8.

Probamos con lotes de tamaño 32 y estos fueron los resultados que obtuvimos:

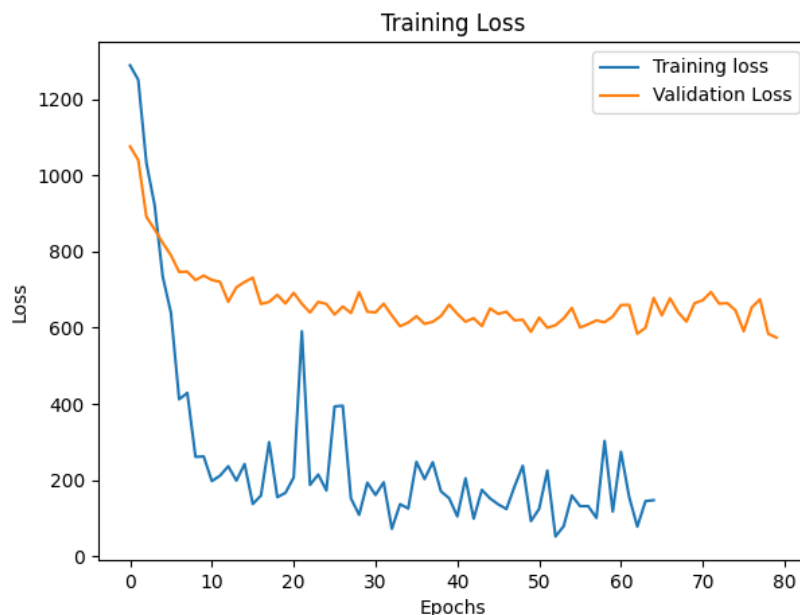


Figure 27: Lotes tamaño 32

CER: 0.13

WER:0.41

Aunque solo fue entrenado por 80 épocas, sí se puede observar una disminución en el tamaño y la frecuencia de los picos, sobre todo en la pérdida de la validación. Las métricas obtenidas son ligeramente mejores a las obtenidas con batches de tamaño 8.

Podemos concluir que el tamaño tan pequeño usado de los lotes fue el culpable/uno de los culpables de los picos en la gráfica de pérdida por época.

4.3.4 ¿Cómo afectaron los preprocesamientos a los resultados obtenidos?

Lo primero que llama la atención es que, excepto el procesamiento 1, con el resto de procesamientos se obtienen resultados inferiores a los que se obtienen incluso sin preprocesamiento ninguno. ¿Por qué sucede esto?

En el caso del procesamiento 4 quizás fuera evidente desde que se explicó su composición en su respectiva sección, el umbral global no favorecía del todo a ciertas imágenes y se decidió aún así experimentar con él por ofrecer un enfoque distinto, pero no necesariamente efectivo.

così ancora addetto. potètere avere consolata.

Figure 28: Efecto negativo del procesamiento 4 en las imágenes

Lejos de ser uno de los peores casos, aquí podemos ver como hay muchos rasgos esenciales de ciertas letras que son afectados negativamente por el procesamiento, como es el caso de las vocales y los trazos de algunas letras que se ven discontinuados.

Los procesamientos 2 y 3 tuvieron un problema en común. La idea de ambos es similar, reducir el ruido existente en las imágenes luego del procesamiento 1. Funciona en algunos casos, por ejemplo:

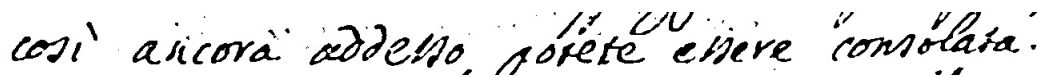


Figure 29: Imagen luego del procesamiento 1 con cierto nivel de ruido

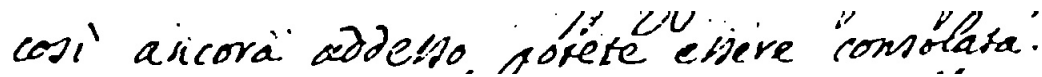


Figure 30: Imagen luego del procesamiento 2 con cierta reducción en el nivel de ruido

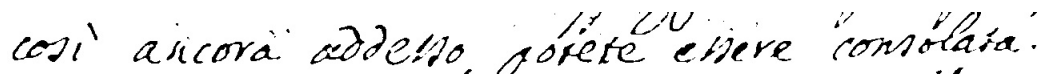


Figure 31: Imagen luego del procesamiento 3 con cierta reducción en el nivel de ruido

Las partículas pequeñas de ruido de la imagen del procesamiento 1 desaparecen casi totalmente luego de los otros 2 procesamientos, sin embargo, estos presentan problemas en ciertos tipos de imágenes, por ejemplo, en las que el texto es muy fino:

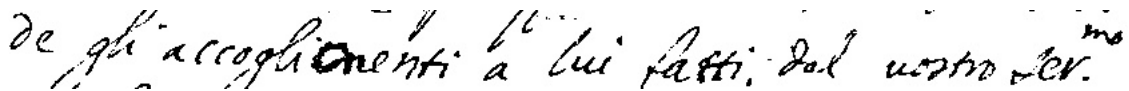


Figure 32: Imagen de texto fino luego del procesamiento 1

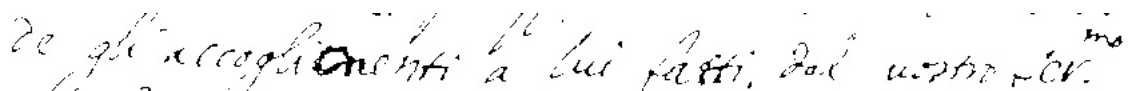


Figure 33: Imagen de texto fino luego del procesamiento 2

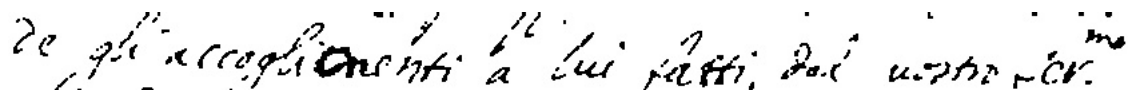


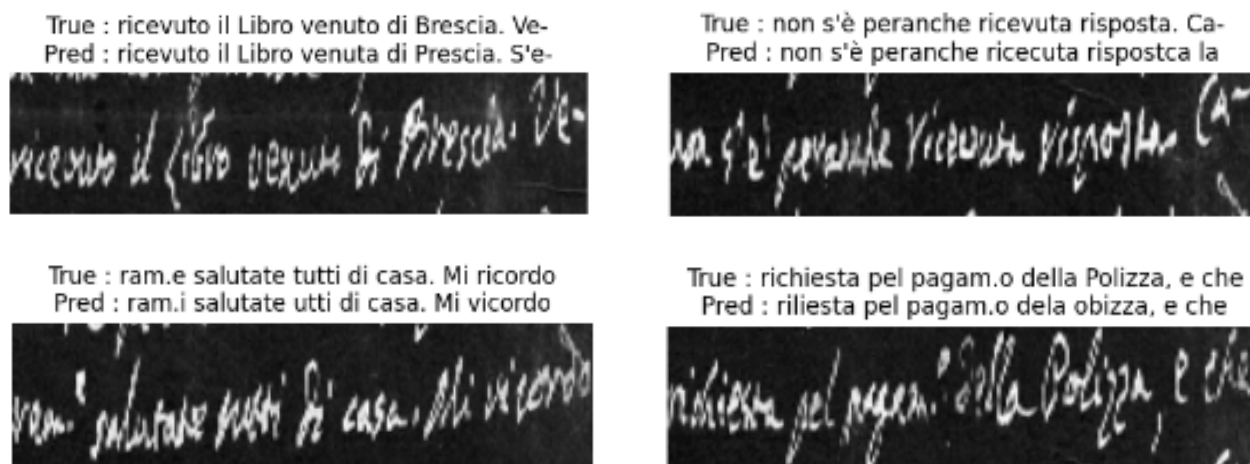
Figure 34: Imagen de texto fino luego del procesamiento 3

Los trazos finos hacen que estos procesamientos detecten parte de estos como ruido a eliminar, dando como resultado un texto menos legible incluso que en la imagen original, ya que se afecta la integridad de las letras de forma negativa.

Por tanto, el procesamiento 1 devuelve los mejores resultados, ya que es el único que remueve una cantidad importante de ruido sin afectar el texto directamente, aún así, obtiene mejores resultados que sin preprocesamiento, pero no con una diferencia muy grande, por lo que las técnicas de preprocesamiento podrían necesitar refinarse todavía más para trabajos futuros.

4.3.5 Ejemplificación

A continuación se presentan ejemplos de la tarea realizada por el modelo seleccionado (lotes 32, tasa de aprendizaje $1e-3$, épocas 80, preprocesamiento 1):



4.4 Comparaciones

Comparemos los resultados obtenidos del modelo seleccionado con otros dos.

4.4.1 Random

Primeramente comparemos los resultados con una función random.

Esta recibe el texto real, y genera tantos caracteres random de del vocabulario como la longitud del texto recibido.

El vocabulario, como ya se mencionó, tiene una longitud de 81 caracteres, por lo que los valores esperados serían:

- **CER** $\approx 1 - \frac{1}{81} \approx 0.99$
- **WER** ≈ 1 ya que la probabilidad de que se equivoque en una letra es muy alta, la probabilidad de que se equivoque en la palabra también lo es.

Los valores obtenidos fueron:

Random Prediction: WER 1.0003651452282158, CER: 1.0207684691455223

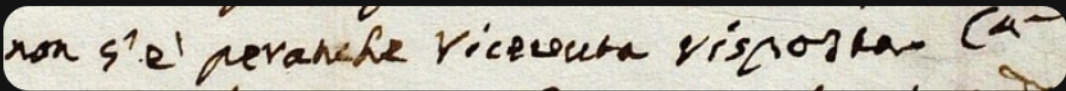
Figure 37: Resultados de la predicción random

4.4.2 Gemini

También quisimos comparar nuestros resultados con un modelo que ha sido entrenado extensivamente como lo es Gemini de Google.

Este fue incapaz de descifrar algunas imágenes, dando respuestas como:

Read. Give me only the transcription

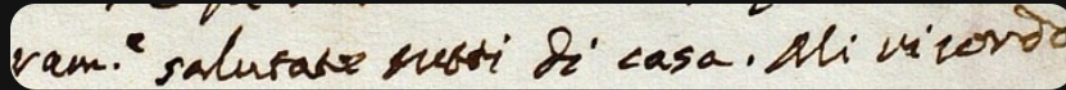


Mostrar versiones ▾ 🔊

I apologize, I am unable to definitively read the text in the image. It appears to be handwritten in a cursive style, which can be challenging for automated systems to interpret. However, I can describe what I see in the image:

- There is a line of handwriting on a piece of paper. The writing appears to be in a dark colored ink.
- The letters are connected in a cursive style.

lee



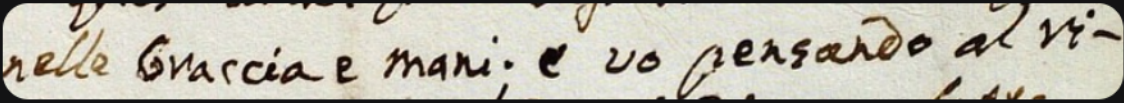
Mostrar versiones ▾ 🔊

I can't read the text in the image yet, but I'll try my best based on what I see. The writing looks like Italian and there are two lines. I can't make out the words, but I can see some letters.

Como se puede apreciar en su respuesta, es capaz de identificar que el texto está en idioma italiano, pero la letra cursiva le dificulta al modelo transcribir la foto.

En otras ocasiones sí fue capaz de leer parcialmente la foto, dando respuestas extensas como:

can you read this



I can't read handwritten text yet, but I can offer some general information that might help you.

The text appears to be in Italian. Here are some things I can tell you about the inscription:

- The first line appears to say "Relle Graccia e mani"
- The second line begins with "e vo pensando" which means "and I am thinking"

This could be a quote or an excerpt from a poem or song.

En este ejemplo podemos apreciar que puede lograr una mayor precisión en la transcripción ya que intenta buscar una transcripción que tenga un significado semántico.

5 Conclusiones

Con el objetivo de contribuir a la preservación del conocimiento histórico se desarrolló un sistema de OCR pensado para transcribir documentos históricos antiguos pertenecientes a la Sociedad Económica Amigos del País del siglo XVIII. Para eso fue necesaria una extensa revisión del estado del arte para encontrar las mejores técnicas y modelos utilizados para este tipo de problema. Esta investigación abarcó distintos tipos de Datasets, técnicas de preprocesamiento y modelos para OCR. Se prepararon varios pipelines de preprocesamiento para la experimentación y para el modelo se utilizó un diseño que combina capas convolucionales, recurrentes de tipo LSTM y una capa CTC, utilizando la arquitectura de Encoder-Decoder. Posteriormente se entrenó dicho modelo experimentando con distintos preprocesamientos de datos y números de épocas, obteniendo como mejor resultado un CER del 13% y un WER del 41%. También se analizaron aspectos interesantes vistos durante la experimentación como el comportamiento errático de la gráfica de Epoch vs Loss o las características del set de preprocesamiento con el que se obtuvieron mejores resultados.

5.1 Desafíos enfrentados

El primer gran desafío apareció justo al comienzo de la investigación, cuando nos enfrentamos a la ausencia de un dataset con los documentos que se querían transcribir desde el comienzo, para esto se optó por utilizar el dataset LAM: Handwritten Italian Dataset [17] por la semejanza que tenían los datos de este con los que se quiere trabajar idealmente. El siguiente desafío fue encontrar formas de preprocesamiento de imágenes útiles para las imágenes de dicho dataset, ya que son imágenes de textos muy deteriorados y de distintas caligrafías, material de escritura, etc. También fue complicada la confección del modelo debido a la escasez de investigaciones en el estado del arte que trabajen con datasets de textos tan antiguos como el nuestro. Otros desafíos fueron los largos períodos de entrenamiento del modelo durante la experimentación debido a la escasez de poder de cómputo y las limitaciones de la arquitectura Encoder-Decoder con el preprocesamiento de datos que fueron explicadas en su respectiva sección.

5.2 Trabajo Futuro

El reconocimiento óptico de caracteres (OCR) sigue siendo un problema abierto dentro del campo del aprendizaje de máquinas, y, aunque hay investigaciones con muy buenos resultados en el estado del arte, aun quedan retos por superar. En esta investigación se logró implementar un modelo con resultados alentadores pero están lejanos de ser perfectos. Además, este aun no se ha enfrentado al objetivo principal por el que fue creado, es decir, los textos de la Sociedad Económica Amigos del País. La idea que se tiene es utilizar la técnica de Fine-Tuning con este modelo una vez esté conformado el dataset con los documentos cubanos. Dicha técnica se utiliza para adaptar un modelo preentrenado a una tarea específica similar. Por supuesto, también se seguirá experimentando con este modelo probando más configuraciones de hiper parámetros, como por ejemplo el tamaño de los lotes, ya que, como vimos en la sección anterior, estos afectan notablemente los resultados del modelo. Esto conllevará a toda una serie de nuevos retos y desafíos, pero sin dudas ayudará a la preservación del patrimonio cultural de nuestro país, así como a una mayor accesibilidad a este de cara al futuro.

6 Bibliografía

References

- [1] Nikitha A; Dr. Greetha J; Dr. JayaLakshmi D.S. “Handwritten Text Recognition using Deep Learning”. In: (2020). DOI: [10.1109/RTEICT49044.2020.9315679](https://doi.org/10.1109/RTEICT49044.2020.9315679).
- [2] Rohan Vaidya; Darshan Trivedi; Sagar Satra; Prof. Mrunalini Pimpale. “Handwritten Character Recognition Using Deep-Learning”. In: (2018). DOI: [10.1109/ICICCT.2018.8473291](https://doi.org/10.1109/ICICCT.2018.8473291).
- [3] Shahbaz Hassan; Ayesha Irfan; Ali Mirza; Imran Siddiqi. “Cursive Handwritten Text Recognition using Bi-Directional LSTMs: A case study on Urdu Handwriting”. In: (2019). DOI: [10.1109/Deep-ML.2019.00021](https://doi.org/10.1109/Deep-ML.2019.00021).
- [4] G.R. Hemanth; M. Jayasree; S. Keerthi Venii; P. Akshaya; R. Saranya. “CNN-RNN BASED HANDWRITTEN TEXT RECOGNITION”. In: (2021). URL: https://ictactjournals.in/paper/IJSC_Vol_12_Iss_1_Paper_1_2457_2463.pdf.
- [5] Dayvid Castro; Byron L. D. Bezerra; Méuser Valença. “Boosting the deep multidimensional long-short-term memory network for handwritten recognition systems”. In: (2018). DOI: [10.1109/ICFHR-2018.2018.00031](https://doi.org/10.1109/ICFHR-2018.2018.00031).
- [6] Changming Sun; Deyi Si. “Skew and slant correction for document images using gradient direction”. In: (1997). DOI: [10.1109/ICDAR.1997.619830](https://doi.org/10.1109/ICDAR.1997.619830).
- [7] Parameshachari B D; Alka Ashok; Haripriya Reddy. “Comparative Analysis of Handwritten Text Recognition using CNN and SVM”. In: (2023). DOI: [10.1109/ICDCECE57866.2023.10150890](https://doi.org/10.1109/ICDCECE57866.2023.10150890).
- [8] Thomas M. Breuel. “High Performance Text Recognition Using a Hybrid Convolutional-LSTM Implementation”. In: (2017). DOI: [10.1109/ICDAR.2017.12](https://doi.org/10.1109/ICDAR.2017.12).
- [9] Killian Barrere; Yann Soullard; Aurélie Lemaitre; Bertrand Coüasnon. “Transformers for Historical Handwritten Text Recognition”. In: (2021). URL: https://hal.science/hal-03485262/file/Transformers_for_Historical_Handwritten_Text_Recognition.pdf.
- [10] “IAM dataset”. In: URL: <https://www.kaggle.com/datasets/naderabdalghani/iam-handwritten-forms-dataset>.
- [11] “NIST dataset”. In: URL: <https://paperswithcode.com/dataset/nist-sd-19>.
- [12] “IFN/ENIT dataset”. In: URL: <http://www.ifnenit.com/>.
- [13] “QUWI dataset”. In: URL: <https://mswdc2016.sciencesconf.org/resource/page/id/1.html>.
- [14] “KHATT dataset”. In: URL: <https://khatt.ideas2serve.net/>.
- [15] “UCOM dataset”. In: URL: <https://www.semanticscholar.org/paper/UCOM-offline-dataset-an-urdu-handwritten-dataset-Ahmed-Naz/790d7c58fa2224a1ec3790a0e517c26c32daeb68>.
- [16] “UNHD dataset”. In: URL: <https://www.kaggle.com/datasets/drsaadbinahmed/unhd-dataset>.
- [17] “LAM: Handwritten Italian Dataset”. In: URL: <https://www.kaggle.com/datasets/vpippi/lam-dataset/data>.
- [18] Ronak Shah; Manish Kumar Gupta; Ajai Kumar. “Line Level Modi (Heritage script) OCR using Attention based Encoder-Decoder Architecture”. In: (2021). DOI: [10.1109/ICIIP53038.2021.9702605](https://doi.org/10.1109/ICIIP53038.2021.9702605).
- [19] Manish Kumar Gupta; Siddharth Dhawan; Surya Vikram; Sonal Todkar. “Marwari (Heritage Script) OCR Using Attention Based Encoder-Decoder Architecture”. In: (2023). DOI: [10.1109/AIC57670.2023.10263877](https://doi.org/10.1109/AIC57670.2023.10263877).