UNIVERSITAT POLITÈCNICA DE VALÈNCIA

etsinf

Escola Tècnica Superior d'Enginyeria Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Mejoras en el reconocimiento de música manuscrita mediante la re-interpretación de modelos de lenguaje para notación mensural

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática

Autor: Villarreal Ruiz, Manuel

Tutor: Sánchez Peiró, Joan Andreu

Curso 2019-2020

Resum

El reconeixement de música manuscrita estudia tècniques perquè els ordinadors siguen capaços de transcriure notació musical manuscrita que es troba registrada en imatges a format electrònic, i fer esta música disponible al públic. Recents acostaments d'intel·ligència de màquina basats en Xarxes Neuronals Recurrents i Profundes han mostrat que funcionen significativament millor en aquest problema que l'acostament tradicional basat en models ocults de Markov, especialment en el cas de Notació Mensural.

Aquestes investigacions basades en Xarxes Neuronals han investigat la tasca de reconéixer Notació Mensural com una altra tasca de reconeixement de text , però no han explotat les característiques dels elements musicals en profunditat. Altres treballs han intentat aprofundir a analitzar elements musicals i en l'extracció de les característiques des de símbols segmentats, sense reflectir això de manera holística.

En aquest treball tractarem de fer un sistema de reconeixement complet directament des dels pentagrames, utilitzant tècniques que enalteixen la informació obtinguda a partir dels símbols. Explorem altres interpretacions de model de llenguatge i provem la nostra proposta en un conjunt de dades disponible de manera pública. En el nostre experiments hem fet una millora del 31% en referència a l'error a nivell de símbol. Amb això, hem anat d'un 3,91% de ràtio d'error, usant tecnologies basades en Xarxes Neuronals, a un 2,7% de ràtio d'error, usant re-interpretacions del model de llenguatge.

Paraules clau: música manuscrita, notació mensural, aprenentatge màquina, xarxes neuronals, xarxes profundes, model de llenguatge

---

Resumen

El reconocimiento de música manuscrita estudia técnicas para que los ordenadores sean capaces de transcribir notación musical manuscrita que se encuentra registrada en imágenes en formato electrónico, y hacer esta música disponible al público. Recientes aproximaciones de inteligencia de máquina basados en Redes Neuronales Recurrentes y Profundas han mostrado que funcionan significativamente mejor en este problema que el acercamiento tradicional basado en modelos ocultos de Markov, especialmente en el caso de Notación Mensural.

Estas investigaciones basadas en Redes Neuronales han investigado la tarea de reconocer Notación Mensural como otra tarea de reconocimiento de texto , pero no han explotado las características de los elementos musicales en profundidad. Otros trabajos han intentado profundizar en analizar elementos musicales y en la extracción de las características desde símbolos segmentados, sin reflejar esto de manera holística.

En este trabajo vamos a tratar de hacer un sistema de reconocimiento completo directamente desde los pentagramas, utilizando técnicas que ensalzan la información obtenida a partir de los símbolos. Exploramos otras interpretaciones de modelo de lenguaje y probamos nuestra propuesta en un conjunto de datos disponible de forma pública. En nuestro experimentos hemos hecho una mejora del 31 % en referencia al error a nivel de símbolo. Con esto, hemos ido de un

3,91 % de ratio de error, usando tecnologías basadas en Redes Neuronales, a un 2,7 % de ratio de error, usando re-interpretaciones del modelo de lenguaje.

Palabras clave: música manuscrita, notación mensural, aprendizaje máquina, redes neuronales, redes profundas, modelo de lenguaje

Abstract

Handwritten Music Recognition studies techniques for computers to transcribe handwritten musical notation that is registered in document images into electronic format, and to make this music available to the public. This task has been of great interest lately, as the technologies improve and can get better and better results on this problem. Recent machine intelligent approaches based on Recurrent and Deep Neural Networks have already shown how they work significantly better in the problem than traditional hidden Markov model based approaches, especially when we are talking about Mensural Notation.

These Neural Network-based researches have researched the task of recognizing Mensural Notation as another written text recognition task, but have not explored the characteristics of musical elements in depth. Other papers have tried to dig deeper into analyzing musical elements and the extraction of their characteristics from segmented symbols, without reflecting this in holistic way.

In this paper, we will try to make a complete recognition system directly from the scores, using techniques that enhance information obtained from symbols. We explore other language model interpretations and test our proposal on a publicly available dataset. In our experiments, we have made a 31% relative improvement in regards to error at the symbol level. With this, we have gone from a 3.91% absolute error rate, using Neural Network-based technology, to a 2.70% absolute error rate, by using language model re-interpretations.

Key words: handwritten music, mensural notation, machine learning, neural networks, deep networks, language model

Índice general

Índice de figuras

Índice de tablas

CAPÍTULO 1
Introducción

La notación mensural escrita se utilizó desde finales del siglo XIII hasta aproximadamente el año 1600, lo que supone muchos años de obras que utilizan dicha notación. La primera edición de una obra musical impresa data del 1501 e incluso entonces no estaba extendida.

Todo esto nos lleva a la conclusión de la existencia de una gran cantidad de obras manuscritas en notación mensural, la mayoría de las cuales se encuentran únicamente en documentos originales, sin acceso al público.

Las pocas obras en notación mensural a las que se tiene acceso son imágenes de las distintas partituras que conforman la obra, pero no se tiene su transcripción. Esto es un problema ya que hoy en día se utilizan tecnologías que dependen de archivos MIDI o al menos de la transcripción digitalizada de la obra.

Para poder formar estos archivos se necesita algo más que la imagen de una partitura, se necesita una transcripción de la misma. Esto nos lleva al problema del reconocimiento de la notación musical manuscrita o OMR (por su traducción al inglés *Optical Music Recognition*).

Este problema forma parte del conjunto del OMR, dentro de los cuáles nos encontramos con problemas más sencillos como el reconocimiento de notación impresa, la cuál es más fácil de reconocer dada su uniformidad, y otros más complejos como la música manuscrita.

Ya se ha abordado este problema desde el punto de vista tradicional, utilizando técnicas análogas a las que se usan en el reconocimiento de texto manuscrito. Varios intentos con distintas tecnologías han ido mejorando los resultados obtenidos, siendo las técnicas que hacen uso de redes neuronales profundas las que permiten obtener resultados más satisfactorios.

En los experimentos realizados en investigaciones previas se trataba la notación musical como texto plano, lo cuál no es ideal. Otros acercamientos recientes se han centrado en explotar las características de las notas, duración y altura, para tratar de averiguar el mejor uso de estas en el reconocimiento.

En este trabajo combinamos las técnicas aplicadas al reconocimiento de texto manuscrito, junto con la explotación de características musicales, basándonos en el uso de modelos de lenguaje, para mejorar el reconocimiento de la notación mensural manuscrita.

1.1 Motivación

Este trabajo tiene dos motivaciones principales, las cuáles se reflejan en los objetivos primarios del mismo.

La primera motivación es el aprendizaje de tecnologías avanzadas en inteligencia artificial, el uso de redes profundas y modelos de lenguaje, así como su aplicación en un caso del mundo real. Esto se centra principalmente en Técnicas aplicadas en el mundo de la investigación que son de gran utilidad en el campo de trabajo de la inteligencia artificial.

La segunda está relacionada con la herencia cultural, hacer tecnologías que permitan que obras de arte de la antigüedad, en este caso musicales, no se pierdan en las estanterías y puedan llegar a todo el mundo.

Esta segunda motivación tiene un carácter menos técnico, pero no por ello es menos válida, la tarea de mantener la cultura a lo largo del tiempo es considerada de gran importancia.

La combinación de estas motivaciones es lo que da forma al trabajo realizado. Teniendo en este una motivación no sólo por el aprendizaje y la aplicación de lo ya aprendido, sino también por el objetivo que se busca lograr en el mismo en cuanto a la preservación de la cultura.

1.2 Objetivos

Los objetivos primarios de este trabajo, con sus respectivos objetivos secundarios asociados, son los siguientes:

–Aprendizaje y aplicación de técnicas de inteligencia artificial.

Esto incluye:

-Aprendizaje y uso de redes neuronales profundas.

-Aplicación de modelos ocultos de Markov a modelos de lenguaje.

-Aprendizaje y aplicación de diversos conjuntos de herramientas adecuados a la tarea.

–Realización de un trabajo de investigación en el campo del reconocimiento de música manuscrita.

Esto incluye:

-Realización de un estudio crítico sobre el problema a tratar.

-Realización de una interpretación adecuada de una posible solución.

-Aplicación de las tecnologías adecuadas para el cumplimiento de dicha solución.

Para valorar esto se compararán los resultados obtenidos con nuestra nueva aproximación a los que se han obtenido en investigaciones previas.

–Realización de un análisis crítico de resultados.

Esto incluye:

-Uso de técnicas adecuadas para el análisis de los mismos.

-Realización de una comparación objetiva con otros trabajos en el campo.

1.3 Metodología

Para lograr estos objetivos se va a proceder de la siguiente manera:

Se han reproducido otros experimentos realizados hasta la fecha, utilizando herramientas de confianza. Si los resultados obtenidos son iguales o las diferencias no son estadísticamente significativas con los anteriores se considerará como logrado el primer objetivo.

Para reproducir estos resultados se han utilizado herramientas de aprendizaje profundo y modelos de lenguaje a partir de modelos ocultos de Markov. La reproducción de estos resultados es la que determina si la forma de proceder es la correcta para el resto del trabajo.

Una vez se tiene esta línea base para el trabajo se ha actuado sobre los datos conforme a la re-interpretación de estos. Con esto cambia la forma del modelo de lenguaje y el como trabaja el mismo.

El modelo de lenguaje es un modelo estadístico que representa la probabilidad de aparición de un símbolo teniendo en cuenta los que preceden al mismo. De esta manera las propiedades lingüísticas pueden tenerse en cuenta.

Una vez esto ha sido realizado se ha completado el proceso de aprendizaje con estos nuevos datos utilizando las técnicas aprendidas respecto al primero de los objetivos y se han obtenido nuevos resultados. Esto incluye tanto el trabajo con redes profundas como el trabajo con modelos de lenguaje.

Estos nuevos resultados después han sido analizados con detenimiento y comparados con lo obtenido hasta la fecha. Para el análisis crítico de resultados se ha estudiado en detalle el producto de los mismos, así como las distintas partes generadas en el proceso, análisis de transcripciones y de los *lattices* generados por el modelo de lenguaje.

En esta última fase se han utilizado *scripts* propios que nos permitirán observar las tendencias en los resultados, así como visualizar las mismas en imágenes para tratar de ver patrones. Con esto el trabajo de investigación se da concluido.

1.4 Impacto esperado

En este trabajo, al ser más experimental que relacionado con el desarrollo de una aplicación, el estudio de impacto esperado no está completamente dirigido a un posible consumidor del producto, sino que también toma otras direcciones.

El primer punto de impacto de este trabajo es la comunidad investigadora. Dado que vamos a explorar técnicas que no se habían utilizado hasta la fecha en este campo, en caso de que las mismas tengan éxito, esto habrá sido de interés en dicha comunidad, por lo tanto este es uno de los puntos donde el trabajo puede afectar y tener impacto.

El segundo punto de impacto esperado es dentro de la comunidad musicológica. Esta comunidad se dedica al estudio científico de los elementos relacionados con la música, tanto a nivel histórico como la relación que esta tiene con el ser humano y la sociedad.

El impacto en esta comunidad ocurre si es el caso de que los resultados obtenidos hasta la fecha se mejoran, pues esto implica que se consigue acercar más y más las capacidades de los reconocedores de notación musical manuscrita a un punto en el que se puedan utilizar de forma general.

Si esto se consigue muchas obras que no se tienen transcritas podrían transcribirse, y esto supondría un gran paso para el estudio de la música y, junto a esto, una gran ayuda para esta comunidad musicológica.

Finalmente, el tercer grupo o punto de impacto a considerar es, aunque no menos importante, mucho más general. Este tercer grupo supone cualquier persona con interés, aunque este no sea en pos de la investigación o del estudio, en la música antigua de la cuál no se tienen transcripciones.

Esto es debido al último punto ya expresado para la comunidad musicológica, aunque en menor escala, pues es de esperar que la cantidad de personas con interés general en la materia que no están dedicadas a la misma es reducida.

Es interesante mencionar también el plazo de tiempo en el que se espera que los resultados de este trabajo tengan impacto, pues difiere entre unos y otros puntos de impacto.

En cuanto a la comunidad investigadores, el impacto debería suceder a corto plazo, pues en esta comunidad únicamente se necesitan los resultados del trabajo para poder trabajar con ellos y por lo tanto que estos tengan impacto.

En cuanto a la comunidad musicológica y el público general el impacto esperado es a más largo plazo, pues si bien vamos a desarrollar un sistema reconocedor, el mismo no va a estar implementado en una aplicación o demostrador a lo largo de este trabajo.

Es en el momento en el que se desarrollara dicha aplicación o demostrador que este trabajo vería su impacto en estas comunidades, ya que la mejora obtenida habría nacido del mismo.

En general, el impacto esperado es positivo, pues incluso aunque las técnicas a utilizar no sean exitosas esto sigue siendo un avance en la investigación de la materia, por lo tanto al menos en uno de los tres puntos de impacto esperado sea cuál sea el resultado obtenido el impacto es positivo.

Aunque estos tres puntos de impacto definidos en los que se espera que nuestro trabajo influya son bastante específicos, al tratarse el mismo de un trabajo de investigación, esto no le resta importancia al efecto que el desarrollo del mismo pueda lograr en ellos.

1.5 Estructura de la memória

En este primer capítulo del trabajo hemos introducido tanto el tema a tratar como el curso que va a seguir el mismo. Se han establecido los objetivos y procedimientos que se van a seguir a lo largo del mismo para lograr esto. En esta sección vamos a ver resumidamente el resto de segmentos que tiene este trabajo.

En el segundo capítulo de este trabajo, en la parte de Contexto Tecnológico, vamos a establecer como se ha trabajado hasta el momento en este campo. Comentaremos los avances hasta la fecha y trataremos de extraer de ellos el conocimiento de utilidad aplicable a nuestro trabajo.

En esta sección también haremos una crítica a este contexto, tratando de encontrar los puntos de mejora del mismo con la finalidad de establecer una propuesta que trataremos de cumplir en este trabajo.

En el tercer capítulo del trabajo hablaremos sobre el marco de trabajo, el entorno y la arquitectura sobre la que nuestra investigación se sustenta. Explicaremos los distintos componentes que conforman un sistema de reconocimiento.

También explicaremos como funcionará el modelo de lenguaje y las interpretaciones del mismo sobre las que se experimentará, siendo estas el punto central de la investigación.

En el cuarto capítulo hablaremos de las características de estas tecnologías a utilizar respecto a nuestros experimentos. Los diferentes parámetros establecidos así como los conjuntos de herramientas utilizados que nos permiten llevar los datos que se tienen al mejor punto posible para poder trabajar con ellos en la práctica.

Este capítulo también incluye de forma detallada una descripción sobre el conjunto de datos con el que trabajamos así como el proceso de creación de los distintos elementos que conforman el sistema.

En el quinto capítulo trataremos los experimentos realizados, el como se han llevado a cabo los mismos, los resultados que se han obtenido, realizando con ello un análisis detallado acerca de lo obtenido con respecto a otros trabajos en la materia.

Esto incluye también una fase de interpretación de resultados, donde trataremos de discernir por qué estos tienen las características que tienen y trataremos de justificar los mismos.

En el sexto capítulo concluiremos el trabajo, haremos un repaso a lo logrado y comentaremos otras vías sobre las que este trabajo podría expandirse o técnicas que se podrían aplicar a otros trabajos en este campo.

También relacionaremos el resultado de este trabajo con los objetivos propuestos al principio del mismo, así como estableceremos la relación entre este trabajo y los estudios cursados a lo lago de la titulación.

Finalmente, nos encontramos con un séptimo capítulo donde se comentan distintos trabajos con los que continuar con lo obtenido en este.

CAPÍTULO 2
Contexto Tecnológico

Un consenso entorno a un marco de trabajo general fue establecido en el pasado a lo largo de distintas investigaciones en el campo del reconocimiento óptico de música. Este proponía dividir el trabajo en diferentes fases donde la imagen se corregía, líneas y símbolos eran separados y el modelo era entrenado con los símbolos de forma independiente [13].

Estas investigaciones tempranas que dividían el problema en varias fases tenían problemas derivados de esta misma separación. La corrección de imágenes es una tarea que mejora el reconocimiento sin añadir directamente errores de reconocimiento significativos. Pero la separación de líneas y símbolos sí los produce, por lo que el marco de trabajo no parece ser ideal.

El uso de modelos ocultos de Markov ha mostrado mejoras en esta tarea utilizando acercamientos holísticos [3]. Estos primeros acercamientos holísticos proporcionan un camino adecuado hacia la mejora en el reconocimiento de música manuscrita.

Trabajos posteriores evitan completamente la fase de eliminación de líneas, que introducía errores dada su complejidad. Con una precisión de clasificación del 92 % de media, estas investigaciones muestran que este paso puede ser evitado sin una gran pérdida en precisión [1].

Más recientemente, métodos basados en redes neuronales han ido en aumento en el campo del reconocimiento de música manuscrita. Esto es debido a los buenos resultados que estos métodos obtienen en muchos problemas distintos [2]. Para reconocimiento de texto y música manuscritos la mayor contribución de estos modelos ha sido la disponibilidad para el modelado y captura de los modelos ópticos de gran variabilidad que se presentan en los estilos de escritura.

La investigación introducida en [2] muestra que una normalización simple como la corrección del sesgo o el centrado de las líneas consigue mejoras, ya que hacen que las imágenes sean más sencillas al trabajar con estos métodos basados en redes neuronales. Esta investigación hacía uso de un modelo de lenguaje basado en n-gramas, donde, a partir de los datos que se tienen, se determina un modelo estadístico que influye en la probabilidad de aparición de un símbolo.

Dado que la altura y la duración de una nota representan dos tipos distintos de información parece razonable representar ambos separados en el modelo de lenguaje. Un acercamiento similar ha sido probado [11] para reconocimiento

de símbolos aislados y sin el uso de un modelo de lenguaje. En nuestro caso lo hacemos considerando el reconocimiento completo de cada pentagrama.

2.1 Crítica al contexto tecnológico

Se han utilizado técnicas basadas en modelos ocultos de Markov y mixturas de gausianas previamente, pero estas no han obtenido los resultados deseados aún suponiendo una mejora respecto al marco de trabajo más clásico. Las capacidades de las redes neuronales y profundas han dejado atrás estás técnicas dada su capacidad para obtener características de las imágenes.

Una de las características que hacen que las redes neuronales profundas sean más competitivas que los modelos ocultos de Markov está en la forma en que se realiza el aprendizaje. En el caso de de las redes neuronales profundas este aprendizaje se realiza discriminativamente, esto es, se utilizan muestras positivas y negativas. Por contra, la forma habitual de entrenar los modelos ocultos de Markov es utilizando únicamente muestras positivas.

Algunas de las últimas investigaciones en este campo han tratado de resolver el problema utilizando las tecnologías punteras, como estas redes profundas, pero no han aprovechado las características de este lenguaje, y lo han tratado como texto manuscrito.

Dado que a la hora de trabajar con modelos de lenguaje no existen métodos basados puramente en redes neuronales el uso de los mismos todavía depende de modelos de estados finitos y modelos ocultos de Markov, por lo que aunque estas tecnologías en parte se quedan atrás siguen siendo de utilidad.

Otras investigaciones han atacado esa parte de interpretación lingüística. Se han tratado de distintas maneras las características de las notas, pero no se han utilizado las tecnologías ideales, y no se han modelado estas características a nivel de lenguaje, sino a nivel óptico.

Para poder tratar el problema como este merece tenemos que considerar ambas cuestiones. El uso de las tecnologías y metodologías más adecuadas y novedosas que permitan sacar el máximo potencial a los datos que se tienen, y combinar esto con un tratamiento adecuado del campo en el que se trabaja, haciendo una correcta interpretación de la notación musical y sus características.

2.2 HTR y HMR

Dentro de este contexto tecnológico es interesante destacar las diferencias entre HTR y HMR. El campo del HTR está notablemente más desarrollado que el del HMR, y muchas de las técnicas que se han aplicado al primero de estos son las mismas utilizadas para estudiar el segundo.

Si bien muchas de estas técnicas son directamente aplicables, como el uso de redes neuronales profundas, es interesante tener en cuenta las diferencias que presentan los datos en HTR y en HMR, pues estas pueden ser decisivas a la hora

de tratar el reconocimiento de los datos. Además, el correcto tratamiento de estas características es uno de los pilares centrales de este trabajo.

Cabe destacar que la notación musical que vamos a estudiar sigue un orden de izquierda a derecha y de arriba hacia abajo, igual que la escritura a la que estamos acostumbrados. Esta similitud es importante a la hora de presentar los datos debido al orden que estos deben seguir.

Una diferencia clave viene dada por la disposición de cada página cuando comparamos música y texto. Aquí vamos a observar las diferencias en la composición de las obras de texto y música.

Si observamos la figura 2.1 podemos ver que en el texto la disposición de los elementos a leer está en dos columnas, pero esto podría estar en una única o esta disposición podría alterarse en distintos textos.

Cuando en la misma figura observamos la disposición de los elementos musicales estos están separados en pentagramas, lo cuál hace que el orden de lectura de la pieza musical sea mucho más claro. La tarea de obtener las distintas piezas de las obras para estudiar las mismas debería ser más sencilla en estas obras musicales dada su estructuración.

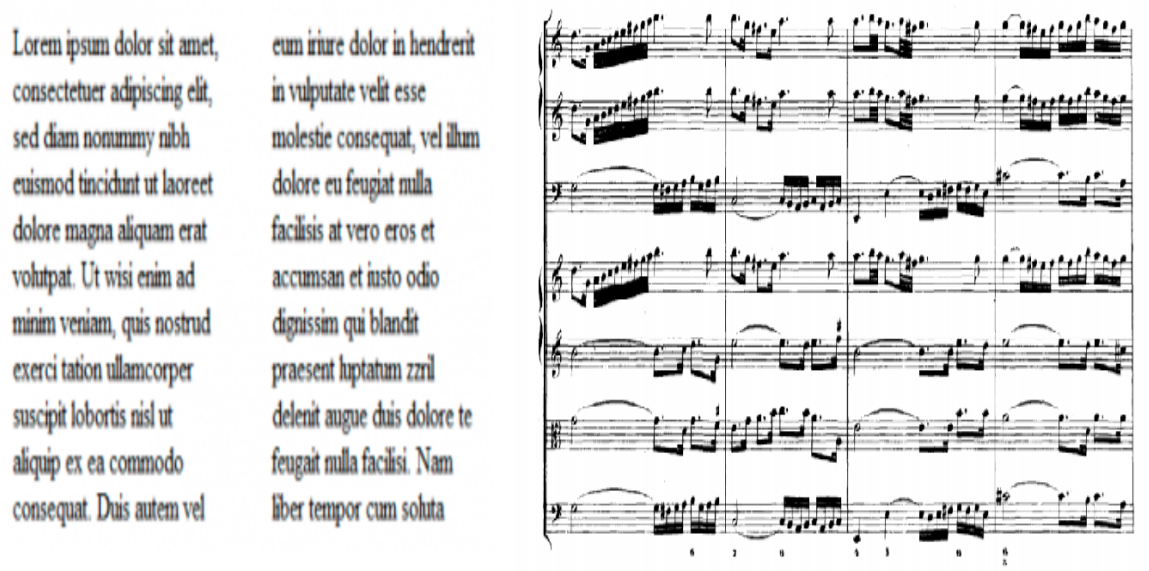

Figura 2.1: Comparación estrucutral entre texto y música.

Una facilidad que presenta el HMR es que, a la hora de obtener los fragmentos de datos a estudiar, en este caso los distintos pentagramas de forma independiente, estos son sencillos de extraer.

Al haber una clara separación entre los distintos pentagramas y al estar estos siempre compuestos de la misma forma, una serie de líneas horizontales, es fácil discriminar de forma individual cada uno de ellos y por tanto obtener los mismos para su estudio de forma automática.

Esta tarea es equivalente a la tarea de extracción de líneas en HTR, la cuál es notablemente más compleja debido a la irregularidad entre líneas, que es notablemente mayor que la irregularidad entre pentagramas.

Otra de las diferencias a destacar es la dimensionalidad de los datos. En el texto nos encontramos con datos que tienen una única dimensión que debe ser reconocida, su forma. Para reconocer cada una de las letras que encontramos en un texto nos basamos en la forma de las mismas, así como en las probabilidades dadas por los modelos de lenguaje pertinentes, pues esto es característica suficiente para determinar qué letra es.

Sin embargo, los datos en música comprenden dos dimensiones a reconocer:

- La primera es la misma que tienen las letras, la forma, que determina duración del sonido. Determinar la forma de una nota musical es lo que define si ésta es una *minima* o una *brevis*, de la misma manera que se determina si una letra es una *a* o una *b*.

- La segunda es la posición vertical de estas notas, que determina el tono de la nota. Una nota musical no representa el mismo sonido si está en posiciones verticales distintas del pentagrama.

Por lo tanto, para reconocer adecuadamente la notación musical hay que reconocer con precisión no sólo cuánto va a durar una nota sino también su tonalidad.

Esta bidimensionalidad es el reto principal al que nos enfrentamos en este trabajo, donde trataremos de explotar estas características en el reconocimiento mediante un modelo de lenguaje como el descrito en la sección 1.3.

Cabe decir que hay elementos en el contexto musical, como la armadura, que pueden afectar al reconocimiento debido a cómo se realizarían los distintos modelados. En este trabajo nos centramos en la bididemsionalidad de los datos y por tanto no hemos tratado con ellos, pero sería interesante tenerlos en cuenta ya que marcan otra diferencia con respecto al reconocimiento de texto.

2.3 Propuestas

En nuestras propuestas de mejora hemos tomado ideas de distintos trabajos realizados previamente en este campo. Vamos a utilizar las tecnologías más establecidas en la cuestión del reconocimiento tanto de texto como de música manuscritos, las redes profundas.

También vamos a combinar éstas con una distinta interpretación de los elementos musicales, tratando las características de las notas musicales como dimensiones distintas y por ende como elementos distintos en la transcripción.

En nuestra primera propuesta estudiaremos completamente por separado estas dimensiones, donde tendremos un modelo óptico con su apropiado modelo de lenguaje para el reconocimiento de cada una de las dimensiones de los datos.

Esta aproximación es más simple pero es un primer paso para acercarnos al tratamiento de la condición de bidimensionalidad en los datos que tenemos.

En nuestra segunda propuesta, para que se pueda obtener información de estas características utilizaremos modelos de lenguaje que buscan específicamente

extraer información de las características separadas, algo que no se había realizado hasta la fecha.

La combinación de un modelo óptico apropiado, junto con un modelo de lenguaje capaz de extraer características de las notas musicales deberían suponer un avance en lo logrado hasta la fecha, pues ambas cuestiones son prometedoras en la investigación a tratar.

Finalmente, nuestra última propuesta es un intento de mejora sobre los resultados obtenidos en nuestra segunda propuesta, donde trataremos de atacar el problema de la concentración de errores en los extremos y presentaremos distintas maneras de tratar de resolver el mismo.

Con estas tres propuestas de experimentación buscamos obtener una mejora sobre los resultados obtenidos hasta el momento en el campo, ya que estamos haciendo uso de las mejores técnicas utilizadas hasta el momento combinándolas con un tratamiento de datos adecuado.

CAPÍTULO 3
Marco de Trabajo

En esta sección vamos a explicar en detalle los diferentes pasos a seguir y las distintas técnicas que se deben aplicar para construir un sistema de reconocimiento, en este caso, aplicado a notación mensural manuscrita.

3.1 Modelo Óptico

El primer componente para conformar un reconocedor es el modelo óptico. El modelo óptico es la parte encargada de obtener las características visuales de la imagen, y asociar a ellas un valor.

El modelo óptico más utilizado en la actualidad es un modelo matemático cuya parte fundamental es un modelo oculto de Markov. Los parámetros de este modelo se entrenan actualmente con redes neuronales de varias capas por las que pasa la imagen, las cuáles obtienen las características de la misma.

Un modelo oculto de Markov o HMM (por sus siglas del inglés) es un modelo estadístico en el que se pretende modelar un sistema de parámetros desconocidos, una cadena de Markov. El objetivo es determinar los parámetros desconocidos de dicha cadena a partir de los parámetros observables, en este caso las muestras de entrenamiento. Para determinar estos parámetros utilizamos las redes neuronales.

En este caso utilizamos redes neuronales convolucionales y recurrentes para determinar la probabilidad de generar un símbolo dada una imagen. Estas capas siguen la estructura de la figura 3.1.

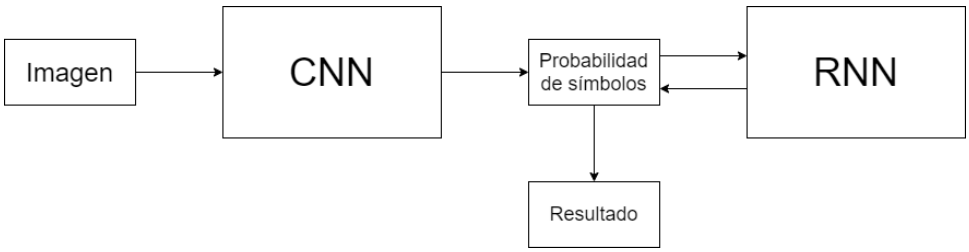

Figura 3.1: Estructura general de generación del modelo óptico.

Toda esta estructura en conjunto se denomina RCNN. Esto nos permite obtener en la capa de salida un vector $P(\mathbf{s}|\mathbf{x}_t)$ donde $\mathbf{s}$ representa los posibles símbolos

musicales y $\mathbf{x}_t$ es el *frame* de la imagen en tiempo t. Para poder hacer el proceso de decodificación con un automata de estados finitos e implementar el uso de HMM el valor de $P(\mathbf{x}_t|\mathbf{s})$ se necesita para calcular $P(\mathbf{s}|\mathbf{x}_t)$.

Esto se puede lograr estimando $P(\mathbf{s})$ a partir de las transcripciones, seguido de $P(\mathbf{x}_t|\mathbf{s}) = P(\mathbf{s}|\mathbf{x}_t)P(\mathbf{x}_t)/P(\mathbf{s})$. Si asumimos que $P(\mathbf{x}_t)$ es equiprobable, se puede ignorar en el proceso de maximización que se lleva a cabo con el algoritmo de Viterbi [12].

De forma general, un HMM es utilizado para cada símbolo, que tiene en cuenta los vectores $P(\mathbf{x}_t|\mathbf{s})$. En este caso utilizaremos dos HMM por símbolo, uno para cada una de sus componentes, altura y duración.

3.1.1. Capas Convolucionales

Las capas convolucionales son versiones regularizadas de un perceptrón de multiples capas, la principal ventaja de estas capas es que trabajos con matrices bidimensionales, que es la forma ideal de representar imágenes, y por tanto son más adecuadas para la tarea.

Este tipo de capas se benefician de los patrones jerárquicos en los datos y son capaces de, a partir de patrones sencillos, comprender patrones más complejos, de manera similar a la composición.

Estas capas aplican una operación matemática llamada convolución a las imágenes a lo largo de las distintas capas ocultas, así como en la capa de entrada y de salida. Cada capa aplica su operación de convolución y pasa el resultado a la entrada de la siguiente capa. La entrada de estas capas es un tensor.

Estas capas utilizan una función de activación en las mismas, en este caso, utilizamos la función *LeakyReLu*. Las ventajas de utilizar una función de la familia *softmax*, en lugar de una función [0-1], es que esta permite, primero, la clasificación en más de dos clases, y segundo, permite expresar las probabilidades obtenidas por las redes neuronales en su salida.

En la figura 3.2 podemos observar como esta función tiene un valor inferior a cero cuando X toma valores inferiores a cero, esto se denomina *leak*, y valores mayores a 0 cuando X es superior. Esto se debe a que la función es de la siguiente forma:

$$f(\ x\ ) = max(\ 0{,}1x\quad ,\quad x\ ) \tag{3.1}$$

Tras la aplicación de cada una de estas capas se aplican una operaciones de *Max-pooling* y de *down-sampling*. Esto reduce el coste computacional y ayuda a evitar que se produzca un sobreajuste a las muestras de entrenamiento.

Max-pooling y *down-sampling*

La operación de *Max-Pooling* ayuda de dos maneras distintas. Por un lado esta operación reduce el coste computacional a la hora de entrenar la red ya que reduce el número de parámetros de los que hay que aprender manteniendo la

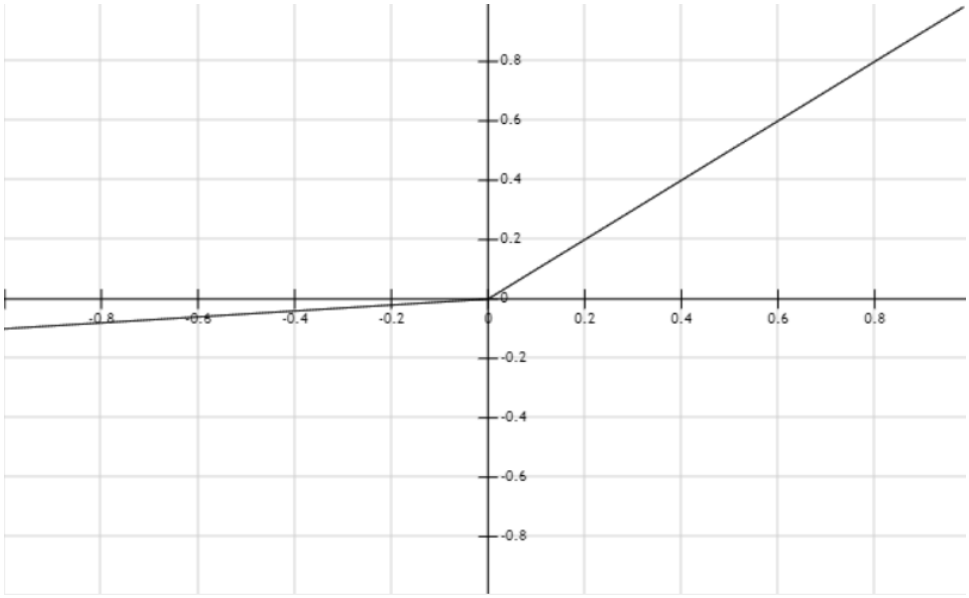

Figura 3.2: Forma que toma la función *LeakyReLu*.

representación interna, por otro, esta operación reduce la cantidad de valores dependientes de los datos y ayuda a generalizar. En la figura 3.3 se observa un ejemplo del funcionamiento de esta operación.

La operación de *down-sampling* es sinónima de la operación de compresión, y las ventajas que se obtienen de esta están relacionadas con una cuestión de espacio requerido por los elementos del sistema.

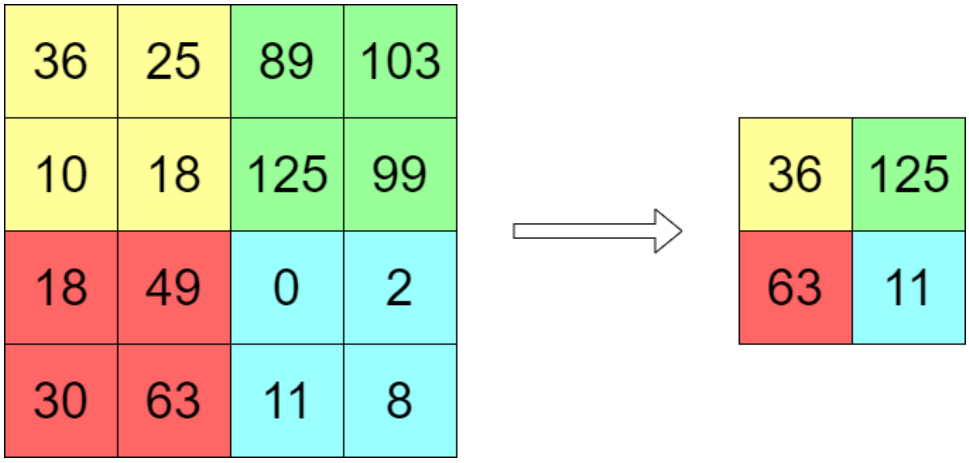

Figura 3.3: Ejemplo de una operación de *Max-Pooling*.

3.1.2. Capas Recurrentes

Las capas recurrentes buscan formar conexiones a lo largo de una secuencia temporal. En nuestro caso estas están formadas por unidades LSTM para poder tener en cuenta información contextual. La función de este tipo de capas es ayudar al reconocedor a relacionar la información en función del cuando se ha obtenido, en función del contexto.

Las unidades LSTM resuelven el problema del desvanecimiento del gradiente que tienen las capas recurrentes clásicas [9]. Para resolverlo las unidades LSTM tienen lo que podríamos denominar memoria.

Este problema de desvanecimiento del gradiente sucede de la siguiente manera, como se observa en la imagen 3.4, conforme se introducen nuevas entradas, la red es menos sensible a las entradas originales y más dependiente de las nuevas entradas por lo que se pierde parte de esa información original.

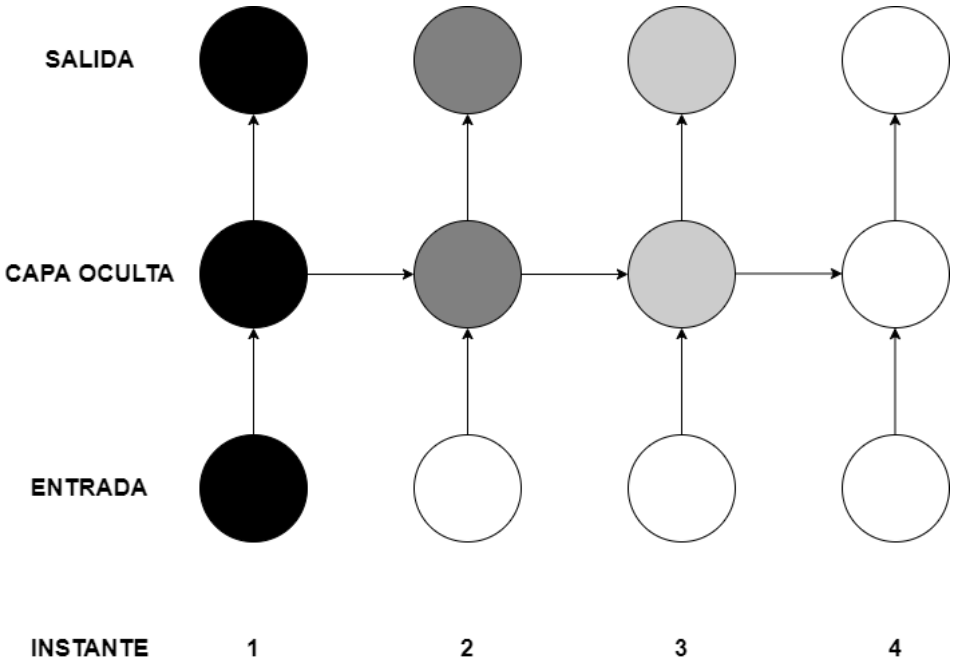

Figura 3.4: Desvanecimiento del gradiente en RNN. La intensidad del color indica como de sensible es el gradiente a la entrada en el momento inicial.

Dada la extensión en la explicación de las unidades LSTM nos limitamos a dirigir al lector a un estudio donde se realiza un análisis mucho más detallado de estas así como se describe en detalle como las mismas resuelven estos problemas planteados [5]. Como en las capas convolucionales, la salida de una capa recurrente se utiliza como entrada de la siguiente.

Estas capas tratan de relacionar la salida obtenida a través de las capas convolucionales con el resultado real, el que se debería haber obtenido. Con esto se consigue que el modelo aprenda a partir no sólo de las características visuales sino también de las características contextuales y del propio resultado.

3.2 Entrenamiento

Las capas convolucionales se entrenan utilizando el algortimo de *Back Propagation* [8]. Este algoritmo está establecido como el algoritmo de entrenamiento de redes neuronales por excelencia, donde se aplican las operaciones a través de las distintas capas de la red hasta obtener una salida.

Este proceso se llama descenso por gradiente, un algortimo de optimización iterativo, que busca un mínimo local en una función diferencial. En las distintas

iteraciones de este se aplica una función de pérdida. Como ya se ha dicho en nuestro caso es la función *LeakyReLU*.

Durante este entrenamiento se generan distorsiones que ayudan a generalizar las imágenes. Con esto se obtienen muestras distintas de las que aprender. Esto ayuda a que no se haga un sobre ajuste acorde a las muestras de entrenamiento, logrando una mayor generalización del modelo óptico.

Las capas recurrentes, sus unidades LSTM, son entrenadas con el algoritmo *Back Propagation Through Time* [19]. Este algoritmo es una variación del algoritmo *Back Propagation*. Para poder aplicarse necesita información respecto al instante de tiempo en el que se debe predecir un símbolo.

Para cada pentagrama, el *frame* en el que se encuentra un símbolo concreto no se proporciona. Para resolver este problema utilizamos la función de pérdida CTC [6], que permite realizar este entrenamiento de forma directa, esto es, sin necesidad de incluir explícitamente en el proceso de entrenamiento la información temporal del frame.

Para realizar este alineamiento entre símbolos y *frames* el algoritmo no requiere que los datos estén previamente alineados. Esto se debe a que lo que este hace es utilizar la probabilidad de todos los posibles alineamientos entre la entrada y la etiqueta objetivo para obtener un resultado.

Esto se observa en la figura 3.5, donde como ejemplo se pone el reconocimiento de la palabra «hola» y su respectiva alineación. En esta figura nos encontramos con las distintas predicciones para una línea, donde se tiene la probabilidad de aparición de cada una de las letras. Aquellas letras idénticas consecutivas se consideran el reconocimiento del mismo símbolo y se combinan.

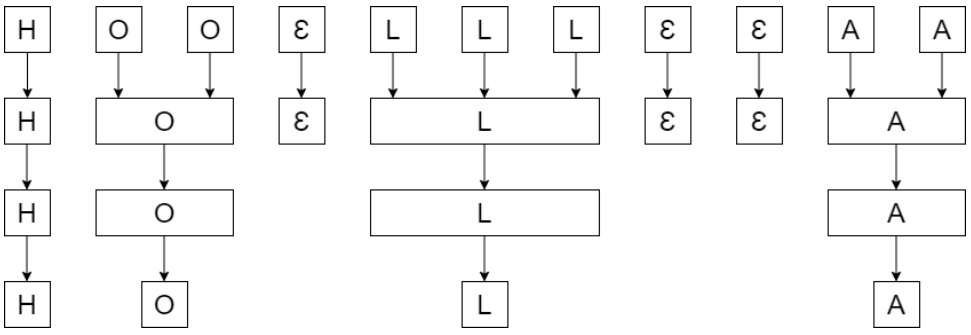

Figura 3.5: Función CTC en 3 pasos: Se mezclan los símbolos iguales, se eliminan los símbolos inútiles (símbolo *dummy*) y se obtiene el alineamiento.

Esta alineación es necesaria ya que se debe tener en cuenta el instante en el tiempo en el que algo ocurre, y esto va determinado por la posición que ocupa de izquierda a derecha. Lo que se encuentra primero según este orden se considera primero en la ordenación.

3.3 Decodificación y modelo de lenguaje

De forma general, para realizar el proceso de decodificación se resuelve la ecuación por optimización local. Cuando únicamente se utiliza un modelo óptico, la mejor hipótesis de decodificación $\hat{\mathbf{s}}$ se obtiene para cada imagen $\mathbf{x}$.

En nuestro caso incorporamos también el uso de un modelo de lenguaje, que apoya al modelo óptico basándose en las características de, en este caso, el lenguaje musical. Dado esto, la ecuación que debemos resolver en el proceso de decodificación es la siguiente:

$$\hat{\mathbf{s}} = \arg\max_{\mathbf{s}} P(\mathbf{s}|\mathbf{x}) = \arg\max_{\mathbf{s}} P(\mathbf{s})P(\mathbf{x}|\mathbf{s}) \tag{3.2}$$

En la ecuación (3.2) $\mathrm{P}(\mathbf{x}\,|\,\mathbf{s})$ representa la probabilidad obtenida con el modelo óptico. En la ecuación (3.2) $\mathrm{P}(\mathbf{s})$ representa la probabilidad a priori, que en el problema que estamos tratando en este trabajo se le conoce como *probabilidad del modelo de lenguaje*.

En este trabajo hemos utilizado modelos de lenguaje basados en n-gramas. Estos buscan reflejar el conocimiento previo basado en el lenguaje, asignar una probabilidad a cada posible secuencia de palabras y limitar el espacio de búsqueda a la búsqueda de la secuencia más probable de palabras.

La probabilidad de una palabra $\mathbf{w}$ se aproxima de la siguiente manera:

$$P(\mathbf{w}) \approx \prod_{i=1}^{m} P(\mathbf{w}|\Phi_n(\mathbf{w}_1 \dots \mathbf{w}_{i-1})) = \prod_{i=1}^{m} P(\mathbf{w}|\mathbf{w}_{i-n+1} \dots \mathbf{w}_{i-1}) \tag{3.3}$$

La estimación de la probabilidad de $P(\mathbf{w}|\mathbf{w}_{i-n+1} \dots \mathbf{w}_{i-1})$ se calcula usualmente a partir de las cuentas de frecuencia relativas $f(\cdot|\cdot)$:

$$P(\mathbf{w}|\mathbf{w}_{i-n+1} \dots \mathbf{w}_{i-1}) = f(\mathbf{w}|\mathbf{w}_{i-n+1} \dots \mathbf{w}_{i-1}) = \frac{C(\mathbf{w}|\mathbf{w}_{i-n+1} \dots \mathbf{w}_{i-1}\mathbf{w}_i)}{C(\mathbf{w}|\mathbf{w}_{i-n+1} \dots \mathbf{w}_{i-1})} \tag{3.4}$$

Estos modelos de n-gramas son los utilizados en el caso del HTR.

En HMR la probabilidad obtenida por el modelo de lenguaje es más difícil de atacar que en un sistema de HTR. Esto se debe a que el modelo de lenguaje para HMR tiene que tener en cuenta dos tipos de información, altura y duración, aunque hasta este momento estos se hayan tratado como una única dimensión conjunta.

En investigaciones anteriores este modelo de lenguaje ha sido modelado como una secuencia de símbolos únicos donde cada símbolo contiene información tanto de la altura como de la duración de la nota que representa. Este modelo de lenguaje se computa como se observa en la expresión (3.5), donde h se refiera a la altura y d a la duración.

$$P(\mathbf{s}) = P(h_1\_d_1 \dots h_{|s|}\_d_{|s|}) \,. \tag{3.5}$$

Esta expresión generalmente se modela mediante modelos de n-gramas:

$$\begin{aligned} & P(h_1\_d_1 \dots h_{|s|}\_d_{|s|}) \\ & \approx \prod_{i=1}^{|s|} P(h_i\_d_i \mid h_{i-n+1}\_d_{i-n+1} \dots h_{i-1}\_d_{i-1}) \ . \end{aligned} \tag{3.6}$$

Para obtener una secuencia de símbolos dada una imagen de entrada, proponemos aproximar el modelo de lenguaje de dos maneras distintas. Este se puede descomponer de la siguiente manera:

$$P(h_1\_d_1 \dots h_{|s|}\_d_{|s|}) \approx P(h_1\ d_1\ h_2\ d_2\ \dots h_{|s|}\ d_{|s|}) \tag{3.7}$$

$$\approx P(d_1\ h_1\ d_2\ h_2\ \dots d_{|s|}\ h_{|s|}) \, . \tag{3.8}$$

Tanto la expresión (3.7) como la expresión (3.8) son propuestas para aproximar la expresión (3.5). En estas expresiones h_i y d_i se tratan como palabras en un sistema HTR, pero en HMR estas palabras están ambas relacionadas con un único símbolo. Son como palabras de un único carácter[1].

En principio, considerar primero altura o duración es arbitrario y no debería estar motivado por ninguna razón física. La principal ventaja de este acercamiento es que el vocabulario del modelo de lenguaje se reduce notablemente, por lo que se reduce el número de parámetros a estimar y el ratio de datos de entrenamiento por parámetros aumenta. Cada palabra toma cuenta de algunos *frames* de la imagen independientemente de si es una altura o una duración.

Cada dos palabras deben tomar cuenta de un símbolo físico. No hemos aplicado ninguna restricción respecto al número de *frames* que cada palabra puede consumir. Algunas restricciones que se podían definir como modelar los distintos modelos ocultos de Markov para cada tipo de palabra con diferente número de estados han sido probadas. Esto no han afectado a los resultados de forma significativa, por ello no se han considerado estas restricciones.

La secuencia de salida que se obtiene con un sistema como este es o bien una secuencia de símbolos musicales construida (si se utiliza la expresión (3.5)), o una secuencia deconstruida (si se utilizan las expresiones (3.7) o (3.8)). En el caso de la segunda se necesitará realizar una reconstrucción de la misma para componer una secuencia de símbolos.

Cuando hablamos de una secuencia de símbolos musicales construida nos referimos a una secuencia donde altura y duración están combinados como una única palabra para representar un símbolo, generada por la expresión (3.5)).

Cuando hablamos de una secuencia deconstruida nos referimos a una secuencia donde altura y duración son palabras distintas y tienen que combinarse a posteriori para representar símbolos únicos (independientemente de la ordenación). Generadas por las expresiones (3.7) o (3.8)).

[1] Utilizamos el término *palabra* cuando nos referimos a d_i o h_i como elementos del modelo de lenguaje.

3.4 Planificación y estructura de trabajo

En este capítulo hemos hablado del marco de trabajo que se nos presenta con respecto a los experimentos que se van a realizar, donde se han establecido los elementos en los que estos se fundamentan. Sin embargo, no hemos mencionado nada acerca del marco de trabajo desde el punto de vista de la planificación y estructura del mismo, con respecto a sus fases y sus horas de trabajo.

En esta sección vamos a comentar esto mismo, estableciendo como se ha desarrollado este trabajo, cuantas horas hombre se han estimado para cada una de las fases que se establecieron para el mismo y como se distribuye el esfuerzo a lo largo de estas.

3.4.1. Fases del proyecto

Primero determinamos cuáles son las fases en las que se sustenta un trabajo como este. Hay que destacar que, al ser un trabajo experimental y no es desarrollo de una aplicación dentro de un ámbito empresarial, este no necesariamente sigue las pautas habituales en el campo de la informática.

La primera fase es una fase de estudio, en la que se analizan las distintas técnicas aplicadas a este campo, las mismas que se reflejan en el capítulo dos de este trabajo, en el contexto tecnológico del mismo.

Se necesita conocer el contexto en el que se trabaja para poder desarrollar un nuevo proyecto. Esto sería el equivalente a un estudio de mercado, en el que observas que se ha desarrollado previamente a tu producto, exceptuando que la competitividad económica no es el factor clave, sino el desarrollo de nuevas tecnologías.

La segunda fase es una fase de lluvia de ideas. Una vez se ha realizado el estudio detallado de otras investigaciones se puede establecer una base sobre la que trabajar, con esto establecido es el momento de determinar donde se introducen las mejoras o novedades.

En esta fase se comparan las distintas opciones que se tienen y se trata de llegar a ideas que, sin un análisis en profundidad, parecen razonables y dignas de su estudio.

La tercera fase de este proyecto es una fase de diseño. Esta fase tiene una duración mayor a las anteriores, junto con una mayor complejidad. En esta fase se estudiarán más en detalle las propuestas a las que se ha llegado en la fase de lluvia de ideas,

En esta fase se establece como debe funcionar y como se debe desarrollar cada una de esas propuestas, determinando desde como se estructura cada una de las partes del trabajo a realizar hasta como debe de funcionar el reconocedor.

La cuarta fase a desarrollar es una fase de experimentación. Aquí se ponen a test los distintos métodos diseñados en la fase anterior. Esta experimentación es exhaustiva en la medida de lo posible, pues se busca dar resultados con una alta confianza.

Esta fase es la que tiene mayor carga de trabajo, pues el desarrollo de estos experimentos es notablemente largo en el tiempo. Estas medidas de tiempo se desarrollarán más adelante en un diagrama de Gantt.

La quinta fase para el desarrollo de este proyecto es una fase de análisis de resultados. Esta fase sería el equivalente a un control de calidad, donde se trata de establecer y determinar el valor de nuestro proyecto.

En este caso no es tan sencillo como asignar un valor económico, pues este no es el objetivo de nuestro trabajo, por lo tanto se tratará de establecer como de válidos son nuestros resultados y que pueden aportar a la comunidad investigadora.

Por último, tenemos una fase de redacción que incluye la generación de un documento en el que se reporte todo lo obtenido, nos referimos a este mismo. Esto sería el equivalente a la generación de un reporte del proyecto que se realizaría en el desarrollo de un producto.

3.4.2. Paquetes de actividades

En la figura 3.6 se pueden observar cada una de estar fases ya comentadas con las respectivas actividades asociadas a cada una de ellas. Estas actividades son aquellas a desarrollar para considerar cada una de estas fases como completadas.

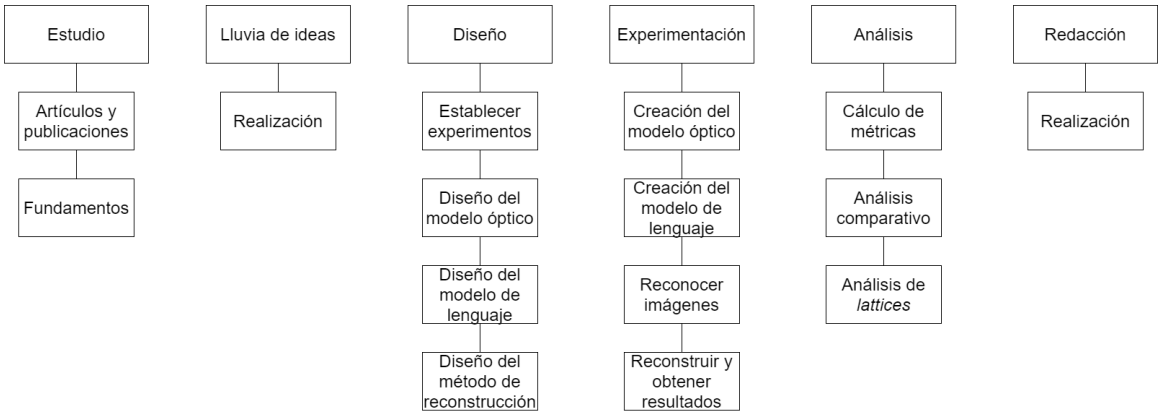

Figura 3.6: Distintas fases del proyecto y sus paquetes de actividades asociados.

Ahora vamos a comentar con mayor detalle los distintos paquetes de actividades asociados a cada fase del proyecto. De nuevo, hay que tener en cuenta que al estar desarrollándose un proceso experimental estos van a diferir de los típicos paquetes de actividades asociados a un proyecto de desarrollo de aplicación.

Los paquetes asociados a la primera fase incluyen:

- El estudio de artículos y publicaciones recientes para conocer los últimos avances y las tecnologías más punteras en el campo.

- El estudio de los fundamentos teóricos de las distintas tecnologías a que son de utilidad en este proyecto, muchas de las cuáles ya se han estudiado a lo largo de la titulación.

La realización de estas actividades da soporte a todas y cada una de las siguientes fases, pues se debe conocer tanto aquello que ya ha tenido éxito en la materia como en que se fundamenta, para poder tratar de aportar nuevas ideas.

En la segunda fase únicamente encontramos un paquete de actividades, esto encaja con el planteamiento de la misma fase, pues únicamente envuelve la realización de esta actividad. Dado que únicamente comprende una actividad se podría plantear incluirla en otra fase, pero dada la importancia de la misma y ya que sirve de punto de inflexión entre la parte más teórica y la parte más práctica del proyecto esto no se ha hecho.

La tercera fase tiene un mayor número de paquetes de actividades, los paquetes asociados a la misma son los siguientes:

- Establecimiento de los experimentos, donde se deja marcado con claridad como se debe realizar cada uno de los experimentos así como los elementos necesarios asociados a los mismos.
- Diseño del modelo óptico, donde se establece como debe ser el modelo óptico a utilizar, en nuestro caso generado por una red RCNN.
- Diseño del modelo de lenguaje, aquí determinamos como van a ser los distintos modelos de lenguaje asociados a los distintos experimentos, se clarifica cada modelo en función del experimento a realizar.
- Diseño del método de reconstrucción, en este último paquete se realiza el diseño de los métodos de reconstrucción que serán necesarios para cada experimento.

La cuarta fase consta mayoritariamente de la test de las distintas técnicas desarrolladas en la fase de diseño. Los paquetes de actividades a realizar en la misma son los que se presentan a continuación:

- Creación y entrenamiento del modelo óptico, aquí se utilizan los datos de entrenamiento y validación para generar los que serán los modelos ópticos asociados a cada uno de los experimentos.
- Creación y entrenamiento del modelo de lenguaje, en este paquete se utilizan los mismos datos que en el anterior para generar un modelo de lenguaje en base al diseño adecuado para cada experimento.
- Realización del reconocimiento, utilizando tanto el modelo óptico como el modelo de lenguaje apropiados en función del experimento en el que nos encontremos se reconocerán las muestras de test para obtener una transcripción.
- Reconstrucción y obtención de resultados, finalmente se aplica el proceso de reconstrucción que sea conveniente dependiendo del experimento y se obtienen los resultados para cada uno.

Finalmente, pasamos a ver los paquetes asociados a la última fase. Estas actividades son las siguientes:

- Cálculo de métricas, donde se obtienen los distintos valores de SER, GER y HER en los distintos experimentos para poder compararlos entre sí y con otros.
- Análisis comparativo, aquí realizamos un análisis de los resultados obtenidos haciendo especial énfasis en observar si estos han supuesto o no una mejora con respecto a los resultados de investigaciones previas.
- Obtención y análisis de *lattices*, finalmente se realiza una útlima actividad de análisis donde se computan los grafos de palabras de los datos de test a partir de sus *lattices* y se analizan los mismos.

La fase final del proyecto contiene una única actividad que da nombre a la misma, esta es la redacción del documento final en el que se establece todo lo obtenido en las fases anteriores.

3.4.3. Diagrama de Gantt y análisis de tiempo

En este apartado vamos a definir el coste temporal de las distintas fases y actividades del proyecto, mostrando las mismas en un diagrama de Gantt como el que se observa en la figura 3.7.

Nos encontramos con una división de horas hombre asociadas a las distintas fases y actividades similar a las que corresponderían en un proyecto cualquiera dentro de esta materia, lo cuál es señal de que, de forma general, están bien establecidas.

Hay que destacar que hay una serie de horas de trabajo necesarias en el desarrollo del proyecto asociadas a los procesos de entrenamiento. Como estos mismos se hacen de forma automática y no requieren atención de forma activa el tiempo que se les ha asociado es inferior al tiempo real necesario ya que no suponen horas hombre en su totalidad.

A la hora de hablar de coste del proyecto hay que hacer una diferencia entre las clásicas horas hombre, las cuáles son las reflejadas en el diagrama de Gantt, y las horas de cómputo necesarias que, como ya hemos dicho, pueden no tener horas hombre asociadas a las mismas.

Estas se pueden considerar de coste más reducido, por tanto tienen menos impacto en el coste final del proyecto. El hecho de que también puedan ser tareas simultáneas a otras hace que se puedan superponer actividades en la planificación.

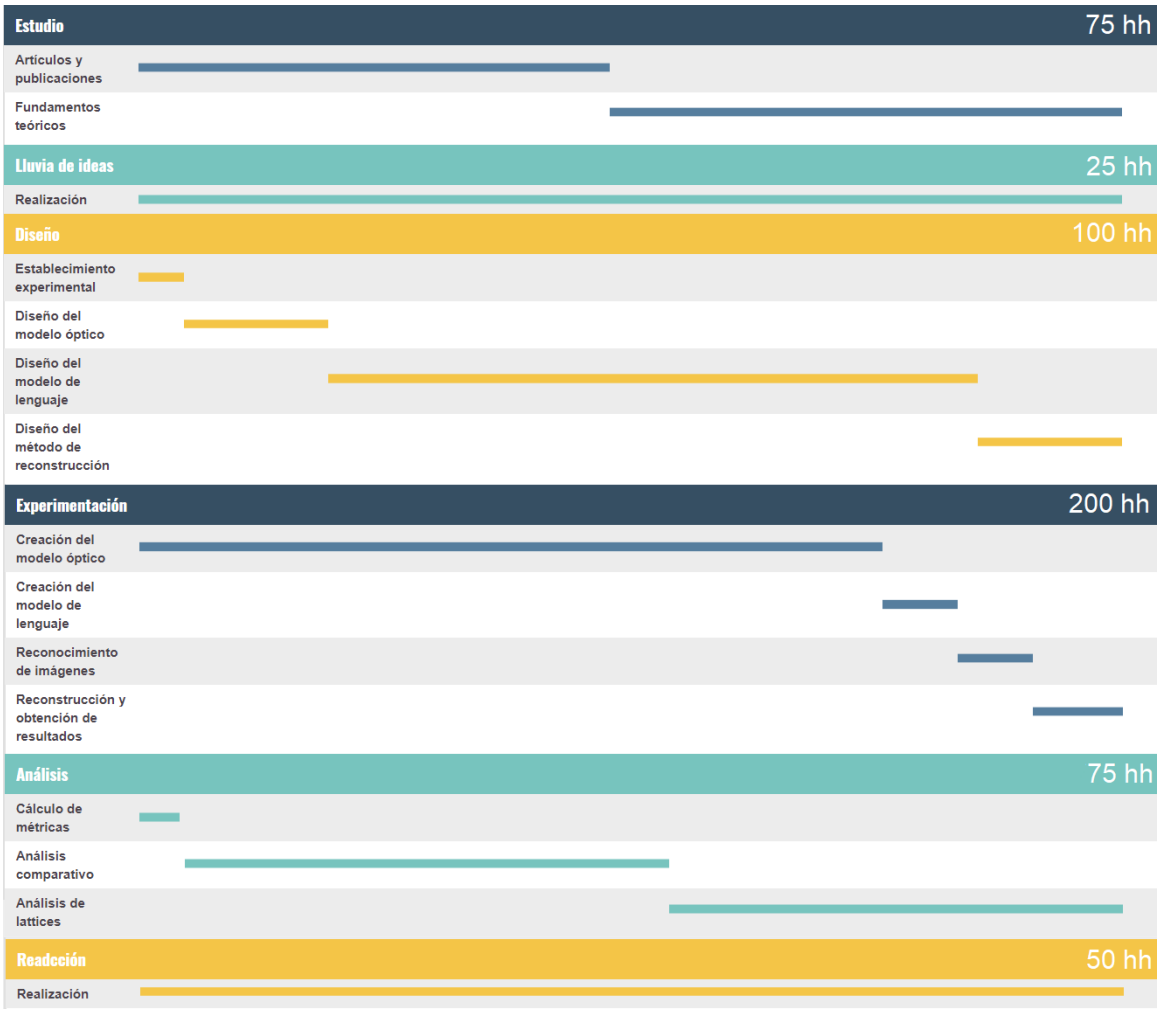

Figura 3.7: Diagrama de Gantt con las horas hombre asociadas a cada fase.

CAPÍTULO 4
Diseño

4.1 Diseño del sistema de HMR

En esta sección vamos a ver el sistema diseñado, este sistema está formado por un modelo óptico compuesto por redes neuronales convolucionales y recurrentes seguido de un modelo de lenguaje basado en n-gramas que da forma a las distintas interpretaciones del lenguaje que hacemos.

La primera parte de nuestra red, la referente a la red neuronal convolucional, consiste en cuatro capas convolucionales con función de activación *LeakyReLu* [20], operaciones de *Max-Pooling* y *down-sampling*. Tras esto se utilizan tres capas recurrentes basadas en unidades LSTM.

Configuraciones de red similares han mostrado buenos resultados en este campo y son notablemente exitosas en el campo del HTR. La tabla 4.1 muestra los parametros que hemos utilizado.

En investigaciones previas [2] se muestra que, a mayor es la altura de la imagen utilizada, se tienen resultados mejores. Esto se explica por el hecho de que, al tener la imagen mayor altura, esta es más similar a la original y se pierde menos información.

En anteriores investigaciones sobre el mismo conjunto de datos que utilizamos en este trabajo, debido a la tecnología que utilizaban, la altura de la imagen estaba fija. Esto no permite el uso de imágenes de altura variable, las cuáles se encuentran en el conjunto de datos. Para resolver esto utilizamos *adaptive pooling*.

También en estas investigaciones previas se muestra que los parametros a la hora de determinar el *horizontal pooling* no afectan notablemente al error obtenido [2], por lo tanto usaremos los mismo que se obtuvieron empíricamente en estas. Estos parametros son $(2, 2, 1, 1)$.

4.1.1. *Adaptive Pooling*

El uso de *adaptive pooling* nos permite no tener que fijar esta altura. Esta tecnología adapta la matriz de parámetros de las imágenes que se utilizan para permitir trabajar con imágenes de distinta altura.

Con esto no se pierde la información al fijar la altura de la imagen y con ello se evita la pérdida de precisión. Esto no debería afectar de forma excesiva dado que la altura de las imágenes, aunque afecta al resultado, no es el elemento más decisivo. Sin embargo, cualquier mejora, aunque pequeña, es interesante.

Tabla 4.1: Configuración de la red utilizada en los experimentos.

Capa	Características
Convolucional (16)	*Max-Pooling* (2)
Convolucional (24)	*Max-Pooling* (2)
Convolucional (48)	*Max-Pooling* (1)
Convolucional (96)	*Max-Pooling* (1)
Rnn LSTM (256)	*Dropout* (0.5)
Rnn LSTM (256)	*Dropout* (0.5)
Rnn LSTM (256)	*Dropout* (0.5)
Salida	*Lin. Dropout* (0.5)

4.2 *Corpus* de datos

El *corpus* utilizado en este trabajo es el *corpus* CAPITAN [4]. Este *corpus* se corresponde a una *missa* (Música sagrada), y está compuesto por 576 pentagramas ya segmentados que han sido extraídos de la obra musical original, de 96 páginas.

Cada símbolo musical tiene dos características: su altura, que determina la tonalidad, y su forma, que determina la duración del sonido. Aunque en música moderna los símbolos que representan silencios sólo vienen representados por su forma, en notación mensural también pueden situarse a diferentes alturas.

Considerando cada combinación de duraciones y alturas como una palabra el *corpus* tiene un vocabulario de 183 palabras. La tabla 4.2 muestra las estadísticas generales de este conjunto de datos. Las particiones de entrenamiento, test y validación son las mismas utilizadas en investigaciones anteriores [4] para que una comparación con estas sea más acertada. Algunos errores en las transcripciones fueron encontrados y corregidos. A su vez se añadieron separadores a reconocer entre notas.

Tabla 4.2: Número de pentagramas, de símbolos distintos y de símbolos totales para cada partición del conjunto de datos CAPITAN con representación estándar.

	Entrenamiento	Validación	test
Pentagramas	462	57	57
Símbolos distintos	176	123	115
Total de símbolos	10 275	1 279	1 267

Información interesante a tener en cuenta es el número medio de símbolos musicales por pentagrama. Esto es útil para considerar como se comportan los distintos modelos dadas diferentes composiciones del pentagrama. La tabla 4.3 muestra esta información.

Tabla 4.3: Media y desviación típica para el número de símbolos en cada pentagrama para cada partición del conjunto de datos.

	Entrenamiento	Validación	test
Media	22.2	22.4	22.2
Desviación típica	6.4	6.4	7.1

Observamos una elevada desviación típica, que indica que los pentagramas varían notablemente en longitud. Esto puede no resultar en ningún problema notable dado que esta variación es similar en las tres particiones.

Para nuestros experimentos hemos modificado la forma de representar estos datos para cada símbolo musical, ya que nuestro objetivo era desarrollar nuevas técnicas de interpretación para el modelo de lenguaje.

En lugar de una combinación de altura y duración, hemos separado estas tratándolas como dos palabras distintas del vocabulario. Cabe decir que hay dos interpretaciones posibles como se muestra en las expresiones (3.7) y (3.8).

Las notas que no tienen una altura asociada se dejan intactas. Hay que destacar que con estos cambios el número de símbolos para entrenamiento, validación y test cambia. Las nuevas estadísticas en este caso se muestran en la tabla 4.4.

Cabe decir que aunque se han introducido palabras para remarcar las separaciones entre distintos símbolos únicos éstos no están considerados en las tablas representativas de cada *corpus*.

Es importante remarcar la gran diferencia en el número medio de muestras de entrenamiento por cada símbolo en las tablas 4.2 y 4.4. Este valor es de media 58 en la tabla 4.2 y 346 en la tabla 4.4.

Este valor afecta notablemente y de manera positiva en la forma de entrenar los modelos ópticos y de lenguaje, tanto po el aumento en entrenamiento como la reducción en vocabulario . Somos conscientes de que esto puede afectar también negativamente porque el mismo símbolo musical en un pentagrama no es visto como idéntico al mismo en otro pentagrama por el sistema, pero esperamos que el modelo de lenguaje sea capaz de corregir estos errores.

Tabla 4.4: Número de pentagramas, de símbolos distintos y de símbolos totales para cada partición del conjunto de datos CAPITAN con representación múltiple para altura y duración.

	Entrenamiento	Validación	test
Pentagramas	462	57	57
Símbolos distintos	59	45	46
Total de símbolos	20 415	2 527	2 513

Otra cuestión a destacar en las tablas 4.2 y 4.4 es la diferencia en el número de símbolos que afecta directamente al modelo de lenguaje. Un número más reducido de palabras significa un menor número de n-gramas y por tanto el modelo de lenguaje puede ser entrenado mejor.

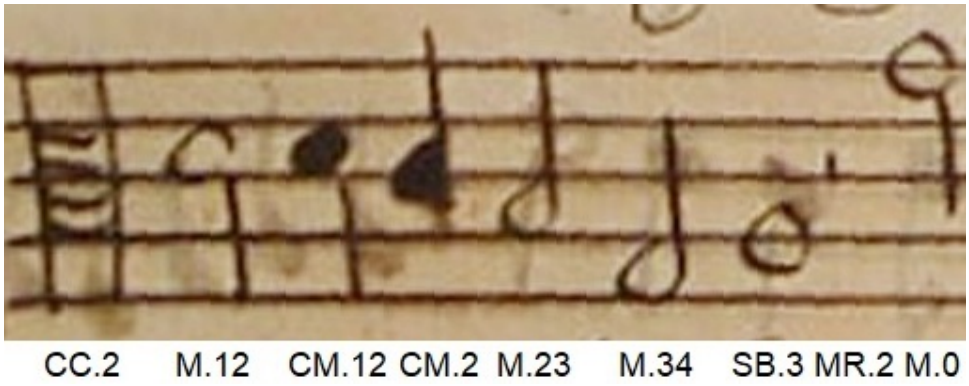

Figura 4.1: Imagen extraída del *corpus* y su correspondiente transcripción (sin incluir separadores) con interpretación estándar, el punto implica que ambas características van unidas.

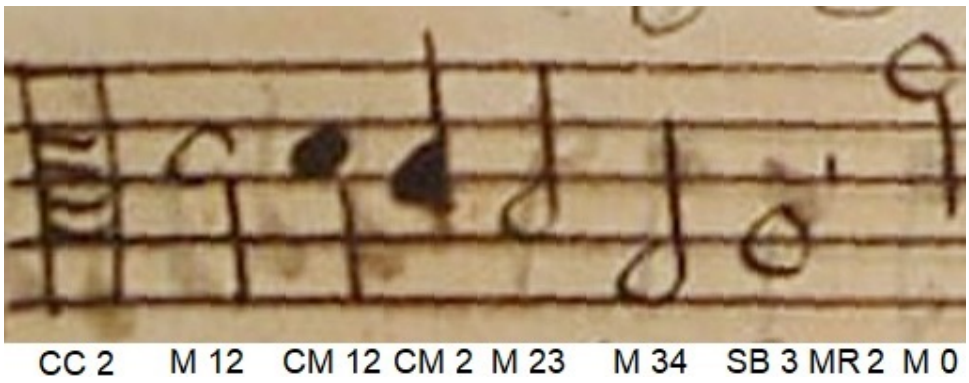

Figura 4.2: Imagen extraída del *corpus* y su correspondiente transcripción (sin incluir separadores) según la expresión (3.7).

En este conjunto de datos también nos encontramos con imágenes de calidades muy distintas. Como se puede observar en la figura 4.4, hay imágenes de este conjunto con notablemente más ruido que otras.

De forma general esto no supone un problema. Aunque las imágenes tengan ruido el modelo debería ser capaz de reconocerlas, pues estas imágenes de distinta visibilidad se encuentran tanto en las particiones de entrenamiento como en las de test.

4.3 Modelo de lenguaje

A continuación vamos a ver como se crea el modelo de lenguaje descrito en la sección 3.3, el cuál se compone principalmente por un modelo de n-gramas.

Para reconocer las transcripciones sin modificar se ha utilizado un modelo de n-gramas donde n varía de 3 a 8, mientras que en trabajos previos [2] se ha probado un 3-grama. Pensamos que la razón de esto es la reducida cantidad de datos. Cabe destacar que la unión de altura y duración de cada símbolo disminuye significativamente el número de posible n-gramas debido a que el vocabulario es mayor (véase la tabla 4.2).

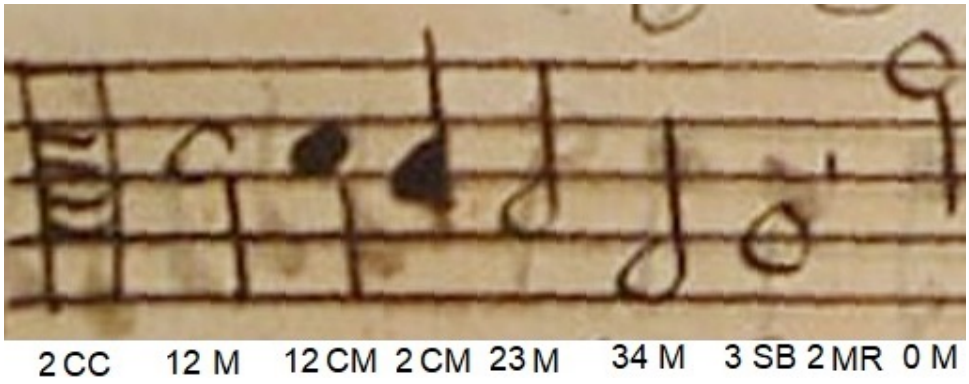

Figura 4.3: Imagen extraída del *corpus* y su correspondiente transcripción (sin incluir separadores) según la expresión (3.8).

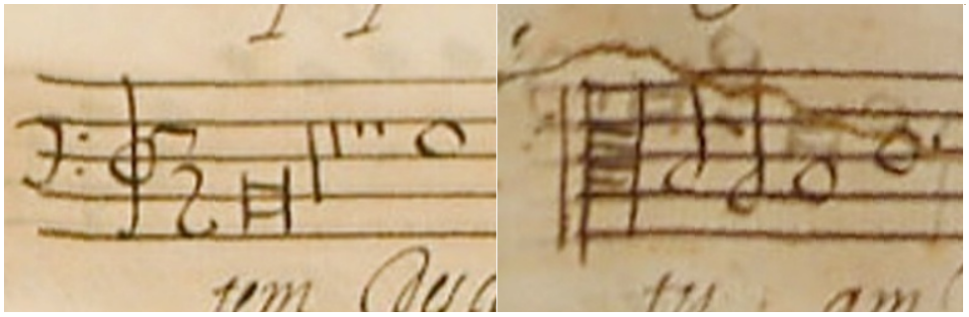

Figura 4.4: Comparativa de claridad en las imágenes del conjunto de datos.

En el caso de utilizar las expresiones (3.7) o (3.8) el tamaño del n-grama probado ha sido más grande. Este tamaño se obtiene de duplicar la cantidad de palabras a considerar. Esto es debido a que el número de símbolos a reconocer aumenta significativamente al hacer uso de estas expresiones, dado que cada nota pasa a ser dos elementos distintos. Cuando hablamos de estos tamaños de n-grama no contamos con la palabra separador, aunque esta sí ha sido considerada en el mismo.

Para la creación de estos modelos de lenguaje que se utilizan en la fase de decodificación junto con el modelo óptico primero se necesita tener un conjunto de datos a partir del cuál entrenar el mismo.

Este conjunto de datos se prepara a partir de las muestras de entrenamiento. Para hacer esto se cogen estas mismas muestras y se obtienen los distintos n-gramas que en ellas encontramos.

La talla de estos n-gramas es en función de n. Si vamos a probar, por ejemplo, un 8-grama, debemos obtener todos los n-gramas que se encuentran en las transcripciones de talla 8. Estos n-gramas obtenidos son los que se utilizan para entrenar nuestro modelo de lenguaje.

A la hora de asignar los pesos a los n-gramas de este conjunto aplicamos un suavizado que ayuda a generalizar. En nuestro caso hemos probado dos descuentos distintos: descuento Witten-Bell y descuento Kneser-Ney.

Para poder aplicar los mismos necesitamos, a parte de obtener los n-gramas de talla n del texto, obtener también los n-gramas de talla 1 a $(n-1)$. Estos se utilizan ya que, al aplicar un descuento con *back off*, cuando nos encontramos con la

ausencia de un n-grama dado, recurrimos a los $(n-1)$-gramas para asignar una probabilidad, si no hay $(n-1)$-grama válido buscamos en los $(n-2)$-gramas, y así sucesivamente, asignando una probabilidad incluso en el caso de que no exista posible n-grama que ocupe ese lugar.

Existen comparaciones entre estos descuentos, donde se prueban ambos para distintas tareas tratando de explicar cuál es mejor dependiendo del trabajo que se realiza [7] [17]. En nuestro caso vamos a determinar de forma empírica cuál es mejor para esta tarea.

Primero, vamos a ver cada uno de ellos:

- Descuento Witten-Bell: está técnica de descuento se diseñó para la tarea de la compresión de texto, y podría considerarse una instancia del descuento Jelinek-Mercer. El descuento al modelo de n-gramas se define recursivamente como una interpolación entre el n-grama de mayor verosimilitud y el $(n-1)$-grama ya suavizado [18].

- Descuento Kneser-Ney: esta técnica se introduce como una extensión del algoritmo de descuento absoluto donde distribuciones de menor orden se combinan con distribuciones de mayor orden. El uso de esta distribución de menor orden es un factor importante en el modelo combinado especialmente cuando hay pocas o ninguna cuentas presentes en la distribución de mayor orden [14].

Para comprobar cuál de los dos descuentos es más apropiado para esta tarea hemos realizado la creación del modelo de lenguaje al completo, así como el reconocimiento, utilizando ambos descuentos por separado y comparando los resultados obtenidos con cada uno de ellos.

El descuento Witten-Bell ha dado unos resultados mejores que el descuento Kneser-Ney, por lo que en los resultados presentados considerar que este es el descuento que se ha utilizado para el suavizado del modelo de n-gramas.

4.4 Modelos ocultos de Markov

Ahora vamos a ver como son los modelos ocultos de Markov que dan forma a las distintas unidades de reconocimiento básicas, las asociadas a las palabras que representan las alturas y duraciones, conformando así el grueso del modelo óptico.

En nuestro caso estos modelos ocultos de Markov suponen directamente las palabras completas, aunque habitualmente estos representen los carácteres que forman las palabras, ya que tenemos palabras de un único carácter.

La cantidad de modelos ocultos de Markov a generar depende de la cantidad de palabras del vocabulario y por tanto este valor es paramétrico en función de si se utiliza la interpretación clásica del lenguaje musical o si utilizamos alguna de nuestras reinterpretaciones.

En la figura 4.5 se muestra el aspecto de cada uno de estos modelos. Cada uno de ellos se compone de tres estados, esto sucede sólo para aquellos que representan una palabra real del vocabulario.

También se debe modelar la palabra *DUMMY*. Esta palabra no tiene una función en el vocabulario más allá de representar aquellos espacios de la imagen para los cuáles no se ha detectado ninguna palabra. En el caso del *DUMMY* el modelo oculto de Markov únicamente tiene un estado.

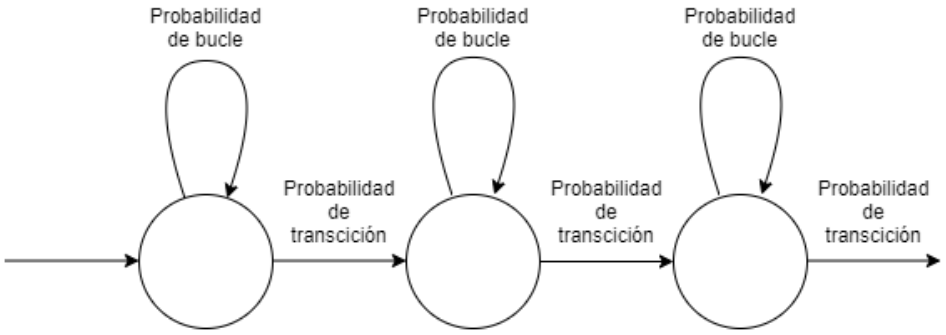

Figura 4.5: Modelo oculto de Markov para aquellas palabras que representan símbolos distintos del *DUMMY*.

En este punto se ha considerado y probado a modificar la cantidad de estados asociada al modelo oculto de Markov de las palabras dependiendo de si estas describen una altura o una duración con la finalidad de observar si esto afecta a los resultados obtenidos.

Para hacer esto se establecen dos topologías distintas para el modelo oculto de Markov en función del tipo de palabra. Se ha probado a modificar el número de estados asociados a cada tipo de palabra entre 3 y 5 en sus distintas combinaciones.

Esto no ha afectado de forma significativa a los resultados por lo tanto hemos optado por la opción más sencilla y mantener tres estados para cada modelo de Markov oculto independientemente de la palabra a reconocer, exceptuando el *DUMMY*.

4.5 Transductor de estados finitos

El siguiente paso es la generación del transductor de estados finitos. A partir modelo de lenguaje y de los modelos ocultos de Markov que suponen el modelo óptico se genera este transductor de estados finitos.

Dado que tanto el modelo de lenguaje generado como los distintos modelos ocultos de Markov son modelos de estados finitos homogéneos estos se pueden combinar en un único modelo de estados finitos de tamaño enorme. Este transductor es el que se utilizará en la fase de decodificaicón para determinar la probabilidad de aparición de una palabra.

Para hacer esta combinación es necesario que el modelo de lenguaje y el modelo óptico estén en la misma escala. Para esto vamos a optimizar los valores del factor de escala del grafo o GSF, que afecta directamente a la probabilidad de

transición en los valores del modelo de n-gramas, y de la penalización por inserción de palabra o WIP, cuyo efecto es de notar conforme más transcripciones se realicen en el mismo.

Estos valores son los que permiten que modelo de lenguaje y modelo óptico se combinen de forma sencilla. Para optimizar estos valores recurrimos al algoritmo simplex, este es un establecido algoritmo de optimización.

Hay que destacar que este transductor no se encuentra de forma completa en memoria, dado el tamaño del mismo del orden de millones de nodos, se trabaja sobre el mismo de forma dinámica utilizando haces de distinto tamaño para recorrer el mismo de forma mucho menos costosa ya que únicamente se representa aquella parte por la que cruzan estos haces. En nuestro caso utilizamos 12 haces, valor que se ha determinado de forma empírica.

4.6 Herramientas

Se han repetido los experimentos que han dado mejores resultados en investigaciones anteriores [2] utilizando la herramienta PyLaia[1] [10]. Estos se han repetido sobre las transcripciones corregidas para establecer una línea base.

La herramienta PyLaia es una herramienta para aprendizaje profundo basada en PyTorch. Esta está especializada en análisis de documentos manuscritos, por lo que es ideal para la tarea.

Es un sucesor de la herramienta Laia[2], de implementación en python y con distintas mejoras sobre esta.

La razón para cambiar la herramienta respecto a las utilizadas en trabajos previos sobre esta misma materia es que PyLaia viene con varios *scripts* que nos permiten una mejor manipulación del modelo de lenguaje.

En concreto utilizamos esta herramienta primero para la creación de nuestro modelo óptico. Segundo para el entrenamiento del mismo. Y finalmente para extraer este modelo en un formato en el que se puedan aplicar modelos de lenguaje.

Esto último es clave debido a que no todas las herramientas te permiten obtener el modelo en un formato idóneo para esta tarea.

Kaldi[3] junto con OpenFST[4] son las herramientas utilizadas para realizar la decodificaicón.

La herramienta Kaldi está diseñada para realizar reconocimiento de voz. En este caso, tanto el habla como la música comparten muchas características, como el hecho de que estas suceden a lo largo del tiempo, o que la música escrita representa componentes sonoros. Por esto, la herramienta es aplicable a este campo sin mayor problema.

[1] https://github.com/jpuigcerver/PyLaia
[2] https://github.com/jpuigcerver/Laia
[3] https://kaldi-asr.org/
[4] https://openfst.org/

Es Kaldi la herramienta utilizada para gestionar los modelos ocultos de Markov y la mayoría de elementos referentes a la creación y aplicación de nuestro modelo de lenguaje.

La herramienta OpenFST es una librería para crear, combinar, optimizar y buscar en transductores de estados finitos ponderados o FSTs. Los FSTs son autómatas donde cada transición tiene una etiqueta de entrada, una etiqueta de salida y una ponderación. Estos son muy utilizados en tareas de reconocimiento del habla, reconocimiento de texto, etc...

En nuestro caso esta herramienta es utilizada en las últimas fases de creación del modelo de lenguaje y, dado su diseño, es muy útil para la tarea que aquí se ha desarrollado.

4.7 *Hardware*

El *hardware* esencial en el contexto de las redes neuronales es la tarjeta gráfica. Este componente permite realizar el proceso de entrenamiento de la red de forma rápida y eficiente, dada su potente memoria así como su gran capacidad de paralelización.

En general, la tarjeta gráfica utilizada en este proceso no debe afectar a los resultados obtenidos, sino más bien al tiempo de entrenamiento. Sin embargo, diferentes tarjetas gráficas pueden trabajar de una u otra forma en referencia al software utilizado.

Sea o no el caso, es importante dar las especificiaciones del *hardware* utilizado ya que este supone una pieza clave en la realización de los experimentos pertinentes al trabajo.

En nuestros experimentos utilizamos una tarjeta gráfica GeForce GTX 1080 con memoria de 8 GB. En la siguiente figura se observan las especificaciones directamente obtenidas de la página web del fabricante:

	GeForce GTX 1080
Fabrication Node	16nm FinFET
Architecture	Pascal
Die Size	314 mm2
GPU	GP104-400
Transistors	7.2 b
Transistors per mm2	~22.9 m
Streaming Multiprocessors	20
CUDA Cores	2560
TMUs	160
ROPs	64
TFLOPs	8.2 TFLOPs
Memory Type	8GB GDDR5X
Base Clock	1607 MHz
Boost Clock	1733 MHz
Memory Clock	1250 MHz
Effective Memory Clock	10000 MHz
Memory Bus	256-bit
Memory Bandwidth	320 GB/s
TDP	180W
Power Connectors	1x 8pin

Figura 4.6: Especificaciones técnicas del *hardware* utilizado, obtenidas del fabricante.

CAPÍTULO 5
Experimentos

En esta sección vamos a detallar los experimentos llevados a cabo para verificar la validez de nuestra propuesta.

5.1 Protocolo de evaluación

Para la evaluación de los resultados utilizamos una medida similar al clásico error a nivel de palabra o WER (del inglés *Word Error Rate*). Debemos tener en cuenta que las transcripciones de los distintos pentagramas tienen dos partes para cada símbolo: una asociada a la altura y otra a la duración del mismo. Ambas representadas en las transcripciones como se muestra en la figura 4.1.

El sistema de reconocimiento tiene que proveer ambos y el proceso de evaluación tiene que hacerse con ambos unidos de forma adecuada. Esto es, para cada símbolo en la imagen, se ha de proporcionar una altura y una duración. En investigaciones previas ambos tipos de información están unidos y se consideran como una única palabra [2].

La métrica para evaluar este sistema es el nivel de error de símbolo diplomático o SER, que es idéntico a la definición del error a nivel de carácter utilizado en HTR, pero trasladado a HMR.

Introducimos nuevas maneras de tratar el modelo de lenguaje donde la duración y la altura son generadas por el sistema de reconocimiento de manera separada. Por esto, utilizamos métricas adicionales para evaluar el sistema. Error a nivel de glifo o GER y error a nivel de altura o HER son introducidas para este propósito.

El GER representa el error causado únicamente por los errores a la hora de reconocer la duración asociada a un símbolo. El HER representa el error causado únicamente por los errores asociados a reconocer la altura a la que se encuentra un símbolo.

Como el análisis de resultados considera pares de alturas y duraciones necesitamos crear un método de reconstrucción para poder realizar está evaluación. Adicionalmente, necesitamos este método de reconstrucción para poder compararnos con otros trabajos realizados previamente en esta misma materia [2].

5.2 Proceso de entrenamiento del modelo óptico

Como ya hemos mencionado previamente en este trabajo, el proceso experimental del mismo consiste en el entrenamiento del modelo óptico, seguido de la creación de un modelo de lenguaje y finalmente una decodificación de unas muestras de test.

En el capítulo 3 se explicaba el fundamento teórico de nuestros procesos tanto de entrenamiento como de creación del modelo de lenguaje y decodificación. En el capítulo 4 se concretó acerca de este modelo de lenguaje, pues el diseño del mismo va ligado al proceso de creación de este.

Sin embargo no se ha concretado cómo se realiza el entrenamiento más allá de los algoritmos utilizados para éste, cosa que haremos en esta sección.

Otro de los factores importantes en este proceso de entrenamiento es el factor de aprendizaje. El factor de aprendizaje es un parámetro del algoritmo de *Back Propagation* que determina cuanto cambia la función obtenida en cada iteración.

Un factor de aprendizaje muy elevado hace que la función que se tiene cambie mucho en cada iteración, haciendo más difícil encontrar un mínimo en esta y por tanto en muchos casos imposibilitando la convergencia.

Un factor de aprendizaje muy reducido hace que la función apenas cambie entre iteraciones, ralentizando notablemente la convergencia y haciendo que un sobre ajuste respecto al conjunto de datos sea más probable.

Este factor se determina de forma empírica, en nuestro caso el valor que ha dado mejores resultados sin ralentizar el proceso de entrenamiento es de $3E-4$.

Tabla 5.1: Factor de aprendizaje y número de iteraciones hasta la detención del entrenamiento.

Tipo de experimento	$1E-4$	$3E-4$	$5E-4$	$7E-4$
Símbolos únicos	307 iter	242 iter	126 iter	14 iter
Símbolos separados (H D	386 iter	282 iter	168 iter	19 iter
Símbolos separados (D H)	375 iter	297 iter	181 iter	17 iter

Cuando el factor de aprendizaje es $1E-4$ o $5E-4$ los resultados obtenidos son peores que cuando este valor es $3E-4$, esto puede deberse tanto a un sobre ajuste como a una generalización demasiado extensa según el caso.

En el caso del factor de aprendizaje de $7E-4$ el resultado es de un error superior al 90 %, es decir, aunque el sistema se detiene rápidamente, esto no implica que se haya logrado un resultado positivo. En este caso al tener un factor de aprendizaje más elevado la función no mejora lo suficiente a lo largo del entrenamiento.

El último componente a determinar en este proceso de entrenamiento es cuando finalizar el mismo. Para marcar esto lo que haremos es, tras cada iteración de entrenamiento del modelo óptico sobre el conjunto de entrenamiento calcularemos el SER de reconocer las muestras del conjunto de validación.

Este valor de SER debería ser decreciente, conforme el modelo óptico se entrene y mejore, el SER sobre el conjunto de validación disminuirá. Llega un punto a

partir del cuál este SER no mejora, por lo tanto es en ese momento cuando detendremos el entrenamiento.

Dado que la mejora del SER puede no ocurrir a lo largo de dos o tres iteraciones, pero sí ocurrir en la cuarta hay que determinar de alguna manera que el SER ya no va a mejorar. Para esto se utiliza un parámetro llamado número de épocas no decrecientes.

Como su nombre indica, este parámetro es el número de épocas que deben pasar, es decir el número de iteraciones de entrenamiento y reconocimiento del conjunto de validación, sin que el valor del SER mejore para detener el entrenamiento.

Con estos parámetros y procesos tenemos suficiente para realizar el entrenamiento de nuestro modelo óptico.

5.3 Pre-procesado y experimento base

Las redes convolucionales son fundamentales a la hora de extraer características de las imágenes, pero algunas técnicas de pre-procesado se pueden aplicar para ensalzar cómo estas redes son capaces de obtenerlas.

Este pre-proceso es especialmente útil cuando no se tiene una gran cantidad de datos de entrenamiento, ya que el pre-proceso reduce la variabilidad. Por esto, en este trabajo usamos algunas de las técnicas de pre-proceso descritas en trabajos anteriores [4], técnicas como corrección de sesgo usando el pentagrama [15] y centrado de la imagen [2].

5.3.1. Corrección de sesgo

La corrección del sesgo es un problema sencillo de resolver. Lo que se busca con esta corrección es que todas las imágenes sean uniformes para el reconocimiento en cuanto a estructura. Eliminando del modelo óptico la dificultad de reconocer imágenes torcidas.

Para esto se utiliza el algoritmo de detección de líneas estables desarrollado por Jaime S. Cardoso descrito en [16]. Una vez se detectan las líneas consideradas estables se corrige la imagen usando las mismas como referencia. Lo que se busca con esto es que estas líneas estables sean paralelas al eje horizontal y, con ello, también lo sea la imagen.

5.3.2. Centrado de la imagen

El pre-proceso para conseguir el centrado de la imagen también es sencillo. En este caso se coge el pentagrama de cada imagen y para este se selecciona la línea central. Lo que se busca es que esta línea sea el centro de la imagen para todas las imágenes.

La imagen se recorta para que tenga una altura de 1.5 veces el tamaño del pentagrama desde la primera hasta la última línea. Cabe decir que, dado que no

todos los pentagramas son iguales, no todas las imágenes resultantes tienen el mismo tamaño.

En otros trabajos aquí se aplica un proceso de normalización de la altura, pero nosotros resolveremos el problema de las imágenes de distinta altura mediante *adaptive pooling*, como se explicará más adelante.

Otros pasos de pre-proceso probados, como por ejemplo transformaciones de color, pero no se obtuvieron diferencias significativas, por lo que han sido descartados. La figura 5.1 muestra un ejemplo de la imagen después de aplicar las técnicas que se han considerado adecuadas.

Nótese que las técnicas aplicadas son aquellas que buscan dar estabilidad a las imágenes para que el reconocimiento sea más sencillo. Se han evitado aquellas técnicas que modificaban de forma más notable la imagen, pues se buscaba evitar las modificaciones en los datos tanto como fuera posible.

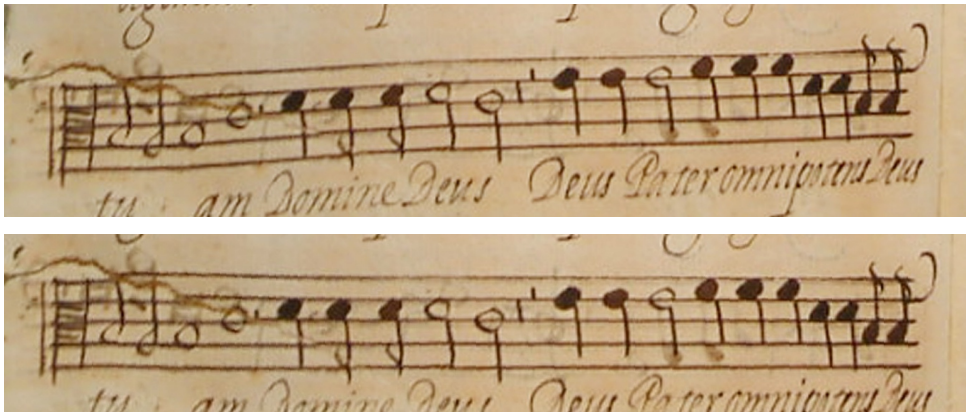

Figura 5.1: La figura superior muestra la imagen original, la figura inferior muestra la imagen resultante tras el pre-proceso.

El error obtenido bajo estás condiciones es de $7{,}38 \pm 0{,}32$ sin hacer uso de modelo de lenguaje. En caso de usar este modelo de lenguaje los resultados mejoran ligeramente.

Estos resultados son lo bastante similares a los obtenidos previamente y por ello podemos considerar que nuestro sistema base es equivalente al utilizado en otras investigaciones.

Bajo estas condiciones y teniendo en cuenta las modificaciones que se han hecho sobre las transcripciones debido a los errores encontrados en las mismas repetimos los experimentos realizados ahora considerando también el modelo de lenguaje.

El nuevo error obtenido es de $3{,}91 \pm 0{,}25$, este error lo consideramos equivalente al que fue obtenido en las últimas investigaciones realizadas en este campo y es con el que compararemos el error obtenido con nuestras nuevas interpretaciones del modelo de lenguaje.

5.4 Experimento con éxito

5.4.1. Reconstrucción

En esta sección vamos a describir como la secuencia de palabras producida por el sistema de reconocimiento ha sido tratada para obtener pentagramas donde cada símbolo está compuesto con una altura y una duración.

Un error que puede derivar del proceso de decodificación es en el que se reconocen consecutivamente varias alturas o duraciones, que no permiten formar un símbolo musical completo (un par duración.altura como se ilustra en la figura 4.1).

Este problema se resuelve con una heurística sencilla: cuando varias alturas se reconocen de forma consecutiva sólo la última se considera. Lo mismo se aplica cuando sucede para las duraciones. En el conjunto de figuras 5.2 se muestran algunos ejemplos de esto.

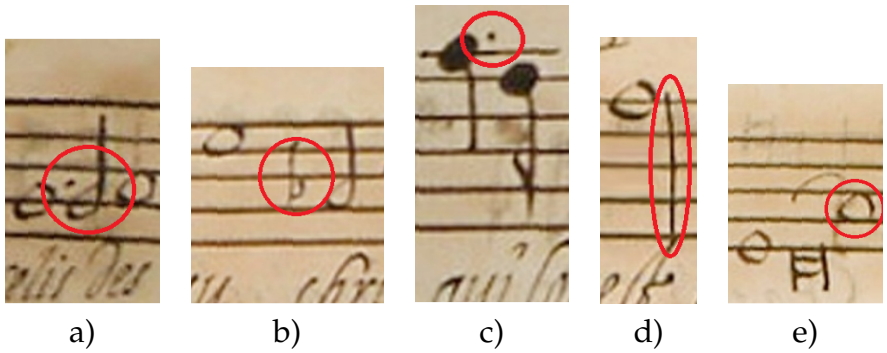

Figura 5.2: Ejemplos para transcripciones producidas por el sistema HMR.

En el conjunto de figuras 5.2 figura (a) el sistema ha generado `3 3 M` para el símbolo central y esta secuencia se ha convertido en `M.3`. En la figura (b) el sistema ha generado `23 3 F` para el símbolo central y esta secuencia se ha convertido en `F.3`. En la figura (c) el sistema ha generado `-2-1 -10 -2-1 D` para el símbolo central y esta secuencia se ha convertido en `D.-2-1`. En la figura (d) el sistema ha generado `M B` para el símbolo de la derecha y esta secuencia se ha convertido en `B`. En la figura (e) el sistema ha generado la secuencia `B SB 23` para el símbolo de la derecha y esta secuencia se ha convertido en `SB.23`.

Hay que decir que aunque está heurística es frágil funciona bastante bien dado que el número de errores es reducido.

5.4.2. Resultados

Comparamos los resultados obtenidos con nuestro método con aquellos reportados en otras investigaciones. La tabla 5.2 muestra los resultados cuando los símbolos musicales han sido reconstruidos como se ha explicado en la sección anterior. La primera fila muestra los resultados obtenidos con la expresión (3.5) como modelo de lenguaje.

El SER de 3,91 es comparable al 7,38 mencionado anteriormente. La diferencia es debida a los cambios aplicados a las transcripciones, pero es equivalente como línea base.

La segunda fila muestra los resultados cuando la expresión (3.7) ha sido utilizada como modelo de lenguaje, y la tercera fila muestra los resultados cuando el modelo de lenguaje utiliza como expresión (3.8). Una mejora clara es observada cuando la expresión utilizada para el modelo de lenguaje es (3.7).

La tabla 5.3 muestra la evaluación cuando los símbolos musicales no son reconstruidos. Las filas en esta tabla siguen la misma interpretación que se ha seguido para la tabla 5.2.

Esta tabla se incluye para mostrar el efecto de la reconstrucción. Se observa según la segunda fila de las tablas que la heurística de reconstrucción tiene sentido.

Como se muestra en las tablas 5.2 y 5.3, ambas formas de re-interpretar el modelo de lenguaje mejoran sobre resultados previos, aunque se observan diferencias entre comenzar con altura o duración.

Como era esperado estos resultados no son iguales pero comparten una gran parte de su margen de error. Dados los resultados obtenidos se podría concluir que el orden en que estas características son obtenidas de los símbolos es relevante de alguna manera.

La diferencia en comportamiento entre los resultados en las tablas 5.2 y 5.3 es debido a que al computar el SER, la distancia de Levenshtein, el denominador de la expresión es aproximadamente el doble de los resultados reportados en la tabla 5.3 respecto a la tabla 5.2.

Tabla 5.2: Media ± Desviación Típica e Intervalo de Confianza al 95 % para el SER considerando elementos reconstruidos.

	SER (M ± DT)	SER (IC 95 %)
Salida única	3,91 ± 0,25	[3.66, 4.15]
Salida múltiple (H D)	2,70 ± 0,32	[2.40, 3.01]
Salida múltiple (D H)	3,19 ± 0,40	[2.81, 3.58]

Tabla 5.3: Media ± Desviación Típica e Intervalo de Confianza al 95 % para el SER considerando elementos separados.

	SER (M ± DT)	SER (IC 95 %)
Salida única	2,73 ± 0,23	[2.51, 2.95]
Salida múltiple (H D)	2,22 ± 0,32	[1.91, 2.53]
Salida múltiple (D H)	2,58 ± 0,34	[2.25, 2.91]

También hemos computado el HER y el GER para analizar si los errores se centraban en alguna de las partes. La tabla 5.4 muestra una pequeña diferencia entre estos cuando el experimento es de salida única para cada símbolo, pero dada su desviación típica este diferencia no es significativa.

Sin embargo, cuando analizamos los experimentos de salida múltiple se denota un incremento en esta diferencia. Estos aún comparten parte de sus márgenes

pero la diferencia es más clara. Esto es algo a tener en cuenta en el futuro, ya que muestra que es posible que sean más fáciles de reconocer las alturas que las duraciones.

Tabla 5.4: Media ± Desviación típica para HER y GER.

	HER	GER
Salida única	2,59 ± 0,12	2,69 ± 0,10
Salida múltiple (H D)	1,51 ± 0,15	1,79 ± 0,19
Salida múltiple (D H)	1,81 ± 0,17	2,05 ± 0,22

5.4.3. Análisis de resultados

Ahora que ya hemos establecido los resultados obtenidos y está claro que el sistema funciona adecuadamente y supone una mejora respecto a lo que previamente se había obtenido en este campo vamos a realizar un análisis y una crítica de los mismos para poder esclarecer lo que en ellos se observa.

Primero, hemos encontrado una diferencia entre los resultados obtenidos al utilizar la expresión (3.7) respecto a los obtenidos al utilizar la expresión (3.8). Para esto vamos a analizar detenidamente los *lattices* obtenidos y el grafo de palabras generado a partir de los mismos, aunque antes de hacer esto, vamos a describir brevemente lo que son los *lattices* y el grafo de palabras.

Un *lattice* es una tabla interpolada que aproxima las relaciones de entrada-salida en los datos. En nuestro caso para realizar el análisis obtenemos el grafo de palabras a partir de estos *lattices* y con ello pasamos a analizar el mismo. Un grafo de palabras es una representación en forma de grafo de las n mejores hipótesis obtenidas que contiene además la información temporal asociada a las mismas.

Lo que buscamos es observar las diferencias entre aquellos generados cuando se ha reconocido mediante la expresión (3.7) y aquellos generados cuando se ha reconocido mediante la expresión (3.8).

A partir de los *lattices* y del grafo de palabras obtenemos cada una de las palabras asociadas a cada conjunto de *frames*, esto nos permite saber cuál es la palabra asociada por el modelo de lenguaje a cada parte de la imagen.

La primera diferencia que observamos es al realizar un análisis estadístico de los mismos. Como se observa en la tabla 5.5 los *frames* consumidos por los símbolos que representan alturas respecto a los consumidos por los símbolos que representan duraciones cambian considerablemente de usar una expresión a la otra. Esto es un indicador más de que la ordenación sí afecta de forma apreciable al reconocimiento.

Observamos como en el modelo que mejores resultados nos ha dado, el que utiliza la expresión (3.7) las alturas consumen más *frames* que las duraciones, sin embargo, en el otro modelo alturas y duraciones consumen una cantidad de *frames* similares. Esto podría ser un indicativo de que las alturas están tratando de captar información de la parte delantera de las notas en el segundo modelo, y que no tienen problema para hacerlo en el primero, de ahí la diferencia en resultados y comportamiento.

Tabla 5.5: Media ± Desviación típica para la cantidad de *frames* que consumen alturas y duraciones con ambas ordenaciones.

	Alturas	Duraciones
Modelo H D ((3.8))	4,84 ± 0,29	5,38 ± 0,84
Modelo D H ((3.7))	10,92 ± 1,26	2,18 ± 0,37

Al observar las figuras 5.3 y 5.4 se ve claramente como ambos modelos extraen la información de partes diferentes de la imagen, por lo que no resulta extraño que se obtengan resultados diferentes. Dada esta observación es muy posible que haya condiciones físicas acorde a altura y duración, por lo que esto es algo a considerar en investigaciones futuras, ya que debería influir en el resultado.

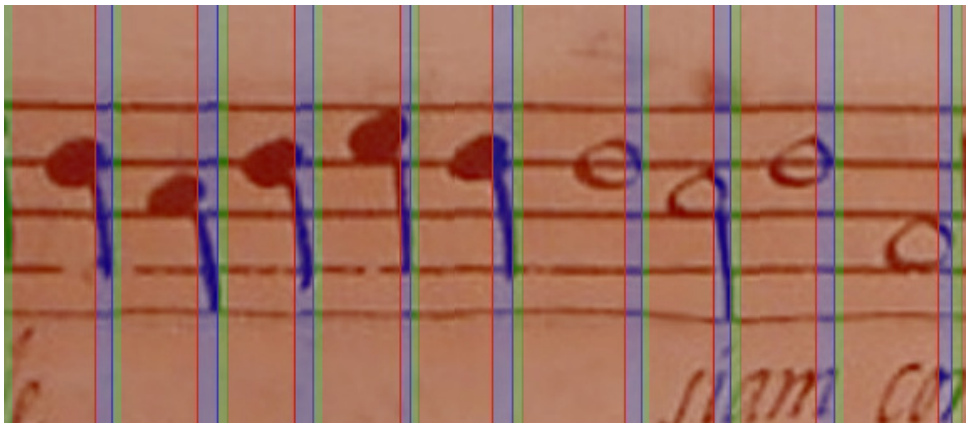

Figura 5.3: Imagen extraída del *corpus* donde cada tipo de símbolo se representa con color, rojo para alturas, azul para duraciones y verde para el separador, siguiendo la expresión (3.7).

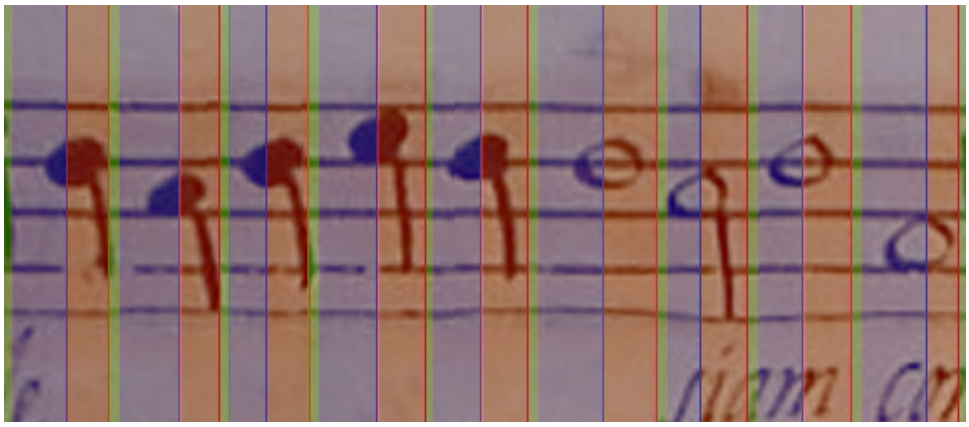

Figura 5.4: Imagen extraída del *corpus* donde cada tipo de símbolo se representa con color, rojo para alturas, azul para duraciones y verde para el separador, siguiendo la expresión (3.8).

Al analizar la posición general de los errores en los pentagramas nos encontramos con el histograma 5.5. Aquí se observa la distribución de los errores a lo largo de los pentagramas.

Como se observa en este histograma la mayoría de errores se concentra al comienzo y al final de los pentagramas. Esto puede deberse al funcionamiento

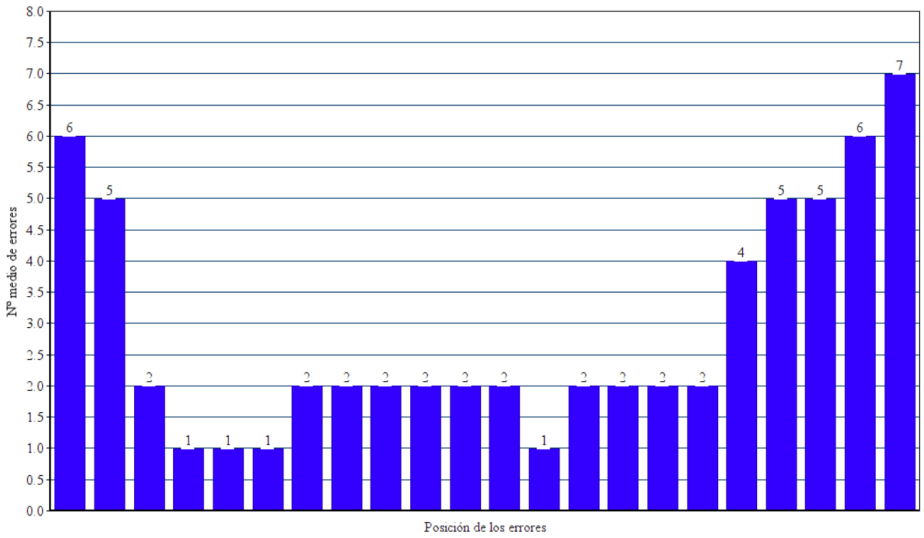

Figura 5.5: Histograma que representa el número medio de errores por cada posición del pentagrama considerando elementos combinados.

del algoritmo de Viterbi, donde al ser un algoritmo que itera hacia delante y hacia atrás la parte central concentra más información, por lo tanto esta parte central se reconoce mejor.

Estos errores también pueden deberse a la calidad de las imágenes hacia los extremos, pues en algunas imágenes debido a que con el tiempo se han deteriorado las partes finales de las mismas han sufrido degradados y son más díficiles de reconocer.

En otro experimento trataremos de solventar estos problemas dando contexto a las imágenes para tratar de resolver el problema si este estuviera causado por el funcionamiento del algoritmo de Viterbi.

5.5 Experimentos sin éxito

En esta sección nos encontramos con dos variantes de los experimentos descritos hasta este punto.

El primero es una versión simplificada de la separación de los componentes de los elementos musicales. Este experimento se realizó previo al experimento con éxito.

El segundo es un intento en la mejora de los resultados obtenidos en el experimento que ha tenido éxito, dado el análisis realizado de sus resultados. Este experimento se realizó posteriormente al experimento con éxito.

5.5.1. Primer experimento

En este caso se ha simplificado la tarea de combinar los elementos duales de cada símbolo musical, altura y duración. Para esto se entrenan dos modelos ópticos distintos y se utiliza un modelo de lenguaje para cada uno de ellos.

Estos dos modelos ópticos y de lenguaje nacen de la idea inicial de separar los elementos musicales en su altura y su duración.

Primero, para entrenar dos modelos ópticos utilizamos dos transcripciones distintas. En este caso tendremos un modelo óptico que reconoce alturas y un modelo óptico que reconoce duraciones. Para cada uno de estos modelos se generan unas transcripciones que únicamente contienen los elementos asociados al mismo. En el modelo de alturas se ignoran las duraciones de las notas, en el modelo de duraciones se ignoran las alturas de las notas.

Al separar por un lado alturas y por otro duraciones esto nos permite modelar las dependencias entre alturas independientemente de las duraciones asociadas a los símbolos, y las dependencias entre duraciones independientemente de las alturas asociadas a los símbolos.

La expresión (3.6) se separa en las dos siguientes expresiones:

$$P(h_1 \dots h_{|s|}) \approx \prod_{i=1}^{|s|} P(h_i \mid h_{i-n+1} \dots h_{i-1}) \,. \tag{5.1}$$

$$P(d_1 \dots d_{|s|}) \approx \prod_{i=1}^{|s|} P(d_i \mid d_{i-n+1} \dots d_{i-1}) \,. \tag{5.2}$$

Para cada uno de estos modelos ópticos y su modelo de lenguaje asociado el proceso ha sido el mismo que en el experimento con éxito. Se ha entrenado el modelo óptico con las imágenes de entrenamiento y sus transcripciones asociadas (en este caso aquellas que contienen sólo las palabras adecuadas) y se ha utilizado el mismo junto a su modelo de lenguaje adecuado para la decodificación.

Como se observa en la figura 5.6 a partir de una misma imagen se obtendrían dos transcripciones distintas que deberían ser combinadas a posteriori.

En este caso tras hacer ambas decodificaciones se obtienen dos ficheros de resultados. Uno de ellos contiene únicamente las alturas reconocidas a partir de las partituras de test, el otro contiene las duraciones reconocidas a partir de estas mismas partituras. De alguna manera hay que combinar estos ficheros para obtener un resultado.

Reconstrucción

En este caso la tarea de la reconstrucción es menos sólida. Al tener que utilizar distintos documentos nuestro único indicador de que una altura y una duración están relacionadas es el orden en el que se reconocen en el documento.

Esto es, por ejemplo, dada una línea obtenida por ambos reconocedores tenemos que asumir que los elementos se relacionan uno a uno en ambos reconocimientos. El problema de esta manera de relacionarlos es que, en el caso en el que

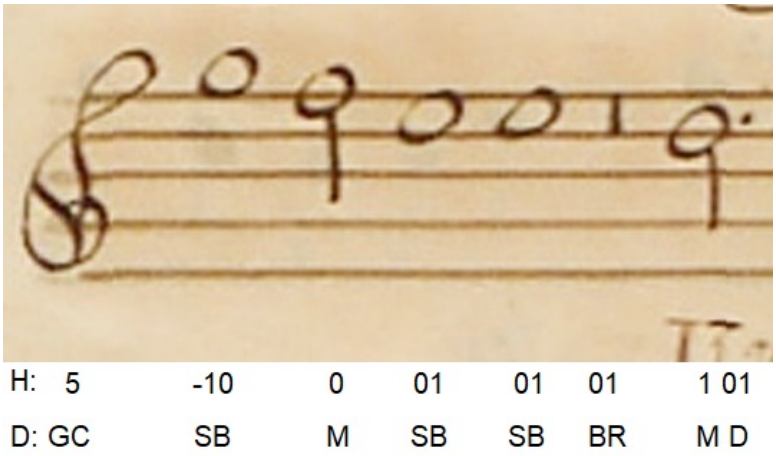

Figura 5.6: Imagen extraida del *corpus* con las transcripciones que obtendrían ambos sistemas, H para el sistema que reconoce alturas y D para el sistema que reconoce duraciones.

una nota no se reconozca en la posición correcta o directamente no se reconozca por cualquiera de los modelos la línea a partir de ese punto es incorrecta.

Dado que esta heurística en la que asumimos que todos los símbolos se reconocen en la posición correcta es demasiado optimista los resultados esperados no son especialmente prometedores, en cualquier caso, si el reconocedor funcionase con una precisión del 100 % esta heurística sería suficiente.

Resultados

Para obtener los resultados de este experimento calcularemos, por un lado, HER y GER en los respectivos documentos donde únicamente encontramos el elemento asociado a ellos, y calcularemos el SER en el documento resultante de realizar la reconstrucción.

La tabla 5.6 muestra estos valores. Como se puede observar al compararlos con los resultados obtenidos en el experimento con éxito, los valores de HER y GER no son tan distintos, sin embargo, sí se observa como son ligeramente peores.

Esto puede deberse a que al eliminar completamente la relación entre alturas y duraciones se pierde información que las relaciona en el modelo de lenguaje y, aunque la relación de símbolos con ellos mismos es más fuerte que con los distintos, esta relación sigue existiendo.

En cuanto al valor del SER, este es notablemente peor que en el experimento considerado de éxito. Esto se puede deber mayoritariamente a la heurística utilizada para la reconstrucción que, dada la situación de los datos, no es buena.

Tabla 5.6: Media $\pm$ Desviación típica para HER, GER y SER para elementos combinados.

	HER	GER	SER
Salida única	$2{,}59 \pm 0{,}12$	$2{,}69 \pm 0{,}10$	$3{,}91 \pm 0{,}25$
Dos salidas	$2{,}67 \pm 0{,}15$	$2{,}80 \pm 0{,}10$	$7{,}88 \pm 0{,}36$

El valor del SER no sólo es peor que aquel que hemos conseguido mejorar a partir de nuestras nuevas técnicas de interpretación, sino que también es notablemente peor que los resultados obtenidos hasta la fecha en otros trabajos.

En este caso la situación de los datos, y que no haya ninguna relación posicional entre alturas y duraciones tal y como se han distribuido, hace que la heurística de reconstrucción suponga un problema que no se puede resolver.

Dado que los resultados de este experimento son peores a nivel de SER que los obtenidos en últimas investigaciones previas a este trabajo no podemos considerar que estos sean de éxito.

5.5.2. Segundo Experimento

En este caso vamos a tratar de resolver el problema observado en el análisis de resultados del experimento con éxito, en el cuál los errores tienden a concentrarse tanto al comienzo como al final de las líneas.

Lo que vamos a tratar de hacer en este experimento es concatenar las imágenes de manera que, para cada imagen, esta tenga una imagen a izquierda y derecha. De esta manera a la hora de entrenar cada imagen central tendrá contexto en ambos lados, al eliminar los extremos de la misma debería perderse menos información.

Para hacer esto de forma adecuada vamos a concatenar las imágenes de forma cíclica, es decir, a la imagen dos se le añadirá a la izquierda la imagen uno y a la derecha la imagen tres, a la imagen tres se le añadirá a la izquierda la imagen dos y a la derecha la imagen cuatro, ...

Haremos esto para todas las imágenes dentro de los distintos conjuntos del *corpus*, sin combinar imágenes de un conjunto con las de otro. Como se observa en la figura 5.7 a una imagen del conjunto de validación se le han añadido otras dos para que esta tenga contexto.

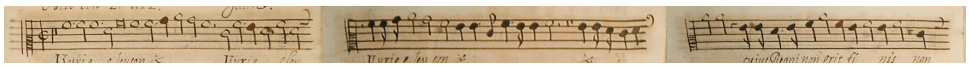

Figura 5.7: Imágenes 96, 97 y 98 del *corpus* concatenadas.

La idea detrás de esto es que ahora todos los símbolos e imágenes de entrenamiento tienen contexto cuando se utilizan para entrenar. Con esto buscamos que en la decodificación estos hayan tenido contexto al ser reconocidos. Lo que haremos es, utilizando este modelo entrenado con contexto, reconocer las imágenes del conjunto de test sin modificar.

Reconstrucción

En este experimento las transcripciones y los distintos elementos del conjunto de datos han sufrido las mismas modificaciones que se han realizado en el experimento con éxito, por lo tanto los métodos de reconstrucción utilizados serán los mismos que en ese.

Resultados

Los resultados obtenidos se presentan en la tabla 5.7. Como se puede observar estos resultados son muy similares a los obtenidos en el experimento con éxito, de hecho son estadísticamente iguales.

Tabla 5.7: Media ± Desviación Típica e Intervalo de Confianza al 95 % para el SER considerando elementos reconstruidos.

	SER (M ± DT)	SER (IC 95 %)
Salida única	4,02 ± 0,32	[3.79, 4.24]
Salida múltiple (H D)	2,67 ± 0,35	[2.41, 2.92]
Salida múltiple (D H)	3,23 ± 0,36	[2.97, 3.48]

No hay nada que comentar especialmente respecto a ellos pues todo lo que se podía decir de los mismos, al ser prácticamente idénticos a lo ya obtenido, ya se ha comentado en la sección de resultados del experimento con éxito.

Si analizamos la posición de los errores, para observar si, aunque el ratio de error no ha mejorado, quizá los errores ya no se concentran hacia los extremos de las imágenes lo que se observa es que la distribución de los errores es muy similar a la obtenida en el experimento con éxito.

Al no haber cambiado nada de forma notable al entrenar con contexto esto nos lleva a pensar que los cambios para añadir contexto deberían haberse hecho en el conjunto de test. En el capítulo de trabajos futuros establecemos como se podría hacer esto a nivel teórico, aunque no hemos podido ponerlo a prueba para este trabajo.

5.6 Limitaciones y problemas encontrados a lo largo de la experimentación

En esta sección vamos a comentar los problemas y limitaciones principales que se han encontrado a lo largo de la realización de estos experimentos, indicando si los mismos se han podido resolver o circumventar.

Errores en las transcripciones

El primero y más claro de estos son los errores encontrados en las transcripciones del conjunto de datos. Estos errores se encontraron en las transcripciones de test, y por eso se volvió a calcular un experimento base a partir del que trabajar.

Estos errores se encontraron durante las primeras fases de análisis de resultados, donde los mismos eran equivalentes a investigaciones anteriores. Al analizar los errores de forma individual para su estudio se encontraron discrepancias que fueron corregidas, tras las cuáles se repitieron los experimentos hechos hasta ese momento.

Tamaño del conjunto de datos

Una de las limitaciones encontradas más claras es el tamaño del conjunto de datos, nos encontramos con un conjunto de datos notablemente pequeño. Esto afecta por un lado a la generación del modelo de lenguaje, el cuál no tiene tanta base de la que partir dado que se genera a partir de las muestras que se tienen. Esto se ha podido circumventar ajustando todo lo posible el modelo de n-gramas y la generación del mismo de forma empírica.

Por otro lado esto afecta al impacto que tienen los errores. El número de errores con el que nos encontramos es bajo, pero al no tener tantas muestras que en una test de entrenamiento y reconocimiento haya dos o tres errores más o menos, que en un conjunto más grande puede no ser notable, aquí si puede notarse más.

Por esto es que los resultados se han presentado con un intervalo de confianza al 95 %, para asegurar que los resultados presentados no son basados en un buen entrenamiento general, sino que se cuentan todo tipo de resultados posibles.

Variabilidad en el tamaño de las imágenes

Un limitación menos importante encontrado es el hecho de la variabilidad en el tamaño de las imágenes, si bien estas son generalmente consistentes en cuanto a altura, la longitud de las mismas es mucho más variable.

Esto no ha supuesto un problema que afecte a los resultados, sino al propio tiempo de experimentación y a la realización de los experimentos. Dado que las imágenes se seleccionan de forma aleatoria para el entrenamiento se ha dado el caso en el que las imágenes de mayor tamaño coincidían en el mismo *batch* y excedían la memoria de la tarjeta gráfica.

Esto ha resultado en la necesidad de ajustar adecuadamente el tamaño de *batch* y ha limitado notablemente el tiempo de entrenamiento, que ha sido mayor del quizá esperado debido a esto.

Por último decir que se han encontrado otros problemas en la ejecución de esta fase experimental, pero estos estaban relacionados con la falta de conocimiento respecto a las herramientas utilizadas que se han aprendido y solventado en el mismo proceso.

CAPÍTULO 6
Conclusiones

6.1 Conclusiones respecto al procedimiento

El procedimiento realizado ha sido exitoso, basta con observar los resultados positivos para saber que el mismo ha funcionado debidamente.

Si lo comparamos con la experimentación más clásica, hemos seguido los pasos de esta al pie de la letra. Hemos observado un problema existente, el del reconocimiento de música manuscrita.

Tras esto hemos realizado una hipótesis basándonos en el conocimiento adquirido, el como la interpretación del modelo de lenguaje es clave para la investigación.

Hemos realizado la experimentación adecuada y pertinente, buscando resultados sólidos y consistentes. Y finalmente hemos analizado los mismos de forma crítica, paso final que fundamenta este proceso de experimentación que ha resultado tan positivamente.

Además, el procedimiento seguido cumple con lo que suele ser un estándar en la materia de la inteligencia artificial y el reconocimiento de texto, el uso de tecnologías punteras y adecuadas y la aplicación del conocimiento que se tiene en la materia.

Las diferencias respecto a procesos más clásicos son el tratamiento de las transcripciones no habitual que se ha hecho, el cuál era clave en esta investigación.

Esto es interesante ya que marca que en futuras investigaciones podría reconsiderarse el tratamiento que se hace de los datos, que aunque es clave establecer un consenso entre investigadores, es importante tener en cuenta que distintas interpretaciones pueden llevarnos a mejores resultados.

No obstante, no podemos obviar que hay errores generados en estos distintos procedimientos. Por ejemplo, en nuestro caso al interpretar los datos de forma distinta a lo habitual ha surgido el problema de la generación de símbolos imposibles.

Aunque hemos podido resolver el mismo sin mayor problema, no hay que despreciar el hecho de que estos errores con los que nadie se ha topado antes pueden surgir al salirse del estándar.

En general la combinación de procedimientos ya establecidos, así como la aplicación de nuestras propias metodologías han dado lugar a un proceso sólido que nos ha permitido obtener buenos resultados.

6.2 Conclusiones respecto a los resultados

Nuestros experimentos muestran que con una reducción del nivel de error de un 31 % el uso de una interpretación con salidas múltiples mejora los últimos resultados obtenidos hasta la fecha.

No olvidar la solidez de los resultados presentados al hacer estas consideraciones, pues los mismos están tratados con un intervalo de confianza al 95 %.

Algunos errores derivan de este proceso que no habían sido tenidos en cuenta en otros trabajos que usaban técnicas similares, dado que estos no utilizaban pentagramas completos, pero han sido resueltos por nuestro método de reconstrucción.

Esto nos permite un sistema de fin a fin que puede evitar procesos como segmentación de símbolos para obtener un pentagrama interpretable.

Es muy posible que la reducción en la complejidad dimensional haya sido un factor importante en obtener estos buenos resultados. Bajar de dos dimensiones a una en nuestros datos puede haber ayudado a la red a generalizar mejor y por tanto hacerla mejor transcribiendo los manuscritos.

Sería interesante probar este acercamiento en otros campos donde los datos tienen múltiples dimensiones/interpretaciones, para ver si una reducción en complejidad dimensional también ayuda a la generalización de los datos en esos campos.

Otras razones para esta mejora puede ser que, con alturas y duraciones separadas sus características lingüísticas independientes son más fáciles de modelar por los componentes recurrentes de nuestra red, dando mejores resultados al final.

Esto tiene sentido dado que las distintas dimensiones en un símbolo musical tienen diferentes dependencias, teniendo una correlación más fuerte con su misma dimensión en otros símbolos que con la dimensión distinta, es más difícil de modelar con ambas dimensiones representadas de forma conjunta.

En general, el uso de estas nuevas interpretaciones ha mejorado notablemente los resultados obtenidos hasta el momento, y es interesante considerarlas para otros trabajos tanto dentro como fuera de la misma materia.

Por último destacar que estos resultados no sólo se encuentran presentados en este trabajo sino también en una publicación aceptada por el ICFHR 2020 (INTERNATIONAL CONFERENCE ON FRONTIERS IN HANDWRITING RECOGNITION 2020).

6.3 Cumplimiento de objetivos

En esta sección vamos a repasar uno a uno los objetivos planteados al comienzo de este trabajo y considerar si se ha logrado el cumplimiento de los mismos.

El primero de los objetivos planteados era el siguiente: .Aprendizaje y aplicación de técnicas de inteligencia artificial". Este incluía como objetivos secundarios el aprendizaje y/o aplicación técnicas específicas en el ámbito de la inteligencia artificial.

Tanto respecto al objetivo primario como a los secundarios consideramos que se han cumplido ampliamente. Se han aprendido y aplicado las distintas técnicas descritas, así como se ha obtenido conocimiento sobre su fundamento teórico para poder trabajar adecuadamente.

Se observa que estas técnicas han sido aplicadas de forma adecuada al ser capaces de reproducir otros experimentos realizados en la materia tratada obteniendo resultados estadísticamente idénticos.

El segundo de los objetivos planteados era el siguiente: Realización de un trabajo de investigación en el campo del reconocimiento de música manuscrita". Este incluía las distintas fases desde el análisis del problema en cuestión hasta lograr una solución para el mismo.

Respecto a ambos objetivos consideramos que estos se han cumplido en su totalidad. Se ha realizado un análisis adecuado donde se ha identificado el camino a seguir, cómo las mejoras a nivel óptico eran complicadas de lograr y se tenía que trabajar con el propio lenguaje.

Tras esto se ha realizado una hipótesis sobre una solución que ha sido llevada a cabo gracias a las distintas técnicas tanto aprendidas a lo largo de los cursos como a aquellas aprendidas específicamente para este proyecto. También se han resuelto aquellos problemas que han surgido a lo largo de la realización de la misma, haciendo uso de las técnicas que tenemos a nuestro alcance.

Esta solución demuestra ser buena al mejorar los resultados obtenidos previamente en este mismo conjunto de datos, así como al ser estadísticamente consistente, con diferencias en resultados que van más allá de una y dos desviaciones típicas.

Finalmente, el tercero de los objetivos planteados era el siguiente: Realización de un análisis crítico de resultados". Este objetivo es sin duda el más difícil de medir, pues no hay un componente numérico que determina si se ha logrado o no el cumplimiento del mismo.

No obstante, consideramos que el cumplimiento del mismo sí se ha logrado. Se han aplicado las técnicas a nuestro alcance para realizar el análisis de los resultados obtenidos y estos se han comparado con los logrados en otras investigaciones. Asimismo se ha tratado de dar una explicación a los resultados obtenidos y a las diferencias entre estos de forma objetiva y crítica.

Aunque no se pueda medir con precisión numérica, podemos decir que se ha hecho todo lo posible por lograr su cumplimiento y que este ha sido conseguido.

En general, los objetivos para este trabajo se han cumplido sin mayor problema, logrando para todos ellos llegar al nivel deseado y encontrando soluciones a los problemas que han surgido en el proceso.

6.4 Relación con los estudios cursados

Para relacionar este trabajo con los estudios cursados basta con repasar las técnicas empleadas así como el fundamento teórico del mismo, ya que este nace de aquello que se ha aprendido a lo largo de la titulación.

Este trabajo hace una especial incisión en las materias cursadas a lo largo de los últimos cursos. Al ser un trabajo relacionado con inteligencia artificial y técnicas más complejas estas son enseñadas cuando el alumno tiene ya una mayor experiencia.

Este trabajo utiliza técnicas basadas en redes neuronales, si bien se utilizan tecnologías como las redes convolucionales y recurrentes, y estas no se han enseñado de forma directa en los estudios, sí que se ha tratado el fundamento teórico de estas, así como se ha dado formación suficiente para poder trabajar con ellas y aprender como aprovecharlas y utilizarlas. Esto se ha visto en la asignatura de Aprendizaje Automático.

El reconocimiento de caracteres sí ha sido estudiado a lo largo de los cursos, dado que es una importante rama de la investigación en informática, teniendo asignaturas como Percepción, que estudian esto de forma directa.

También se han estudiado los modelos de lenguaje, y con ello los modelos de Markov necesarios para conformar estos mismos. En la asignatura de Sistemas Inteligentes así como en la asignatura de Aprendizaje Automático se han estudiado estos elementos y algoritmos para tratarlos.

También se han utilizado técnicas más relacionadas con la rama algorítmica, para el diseño de nuestros algoritmos de reconstrucción, así como para realizar el análisis y crítica en profundidad de resultados, pues se han analizado ficheros de gran tamaño para los cuáles ha sido útil el conocimiento en algorítmica adecuado para poder tratarlos eficientemente.

En general se han utilizado desde conocimientos específicos aprendidos en los últimos cursos de la titulación, hasta técnicas de programación más sencillas pero eficientes que se han aprendido a lo largo de la misma.

CAPÍTULO 7
Trabajos futuros

En este capítulo final vamos a desarrollar los distintos caminos con los que podría continuar un trabajo como este, definiendo más en detalle algunas de las ideas ya mencionadas.

Repetir los experimentos sobre un *corpus* más completo

Uno de los caminos que sería interesante explorar y que no se ha podido tratar en este trabajo es la aplicación de estas técnicas desarrolladas a distintos conjuntos de datos.

Los resultados que hemos presentado son positivos y las mejoras obtenidas encajan con las hipótesis planteadas, sin embargo, es cierto que el conjunto de datos utilizado está muy cerca de ser escaso en cuanto a volumen.

Por esto, sería interesante repetir los experimentos utilizando un conjunto de datos de características similares, pero notablemente más extenso, para terminar de confirmar sin lugar a duda las hipótesis planteadas.

Creación de un indexador y demostrador

Otro paso en la continuación de un trabajo como este sería la creación de un indexador, que utilice los resultados obtenidos por el reconocedor para indexar a modo de base de datos los mismos.

Una vez realizada esta indexación, se utilizaría para crear un demostrador, que permita buscar en el mismo distintos elementos musicales a petición del usuario.

En el caso del conjunto de datos utilizado se presenta el problema de no tener acceso a las páginas completas de la obra, sino únicamente a los pentagramas extraídos de la misma, haciendo que la tarea de la creación de un demostrador pierda valor.

En la indexación de música también se presentan otros problemas como la dificultad para identificar patrones mélodicos a partir de los cuáles indexar, o el hecho de que muchos elementos musicales se repiten un elevado número de veces, haciendo la tarea de la búsqueda computacionalmente costosa para consultas complejas.

Estudio de separación de dimensiones para atacar el problema de la música polifónica

Otro notable problema en el campo del HMR es la música polifónica, música donde varias notas o sonidos ocurren simultáneamente. Este es un problema interesante a estudiar a partir de lo observado en este trabajo.

Aquí hemos separado las distintas dimensiones de un único elementos musical para reconocerlas por separado. Este tipo de modificaciones y estas separaciones en las dimensiones de los elementos que se estudian podrían intentar aplicarse a la música polifónica.

En ese caso la idea de reconocer dos dimensiones asociadas a un único elemento pasaría a ser el reconocimiento de dos o más notas distintas asociadas a un único momento en el tiempo.

Si bien es cierto que esta traducción no es inmediata, y se requeriría un estudio con respecto a la música polifónica, es posible que esta sea una manera de enfrentar la misma.

Introducción de elementos musicales más complejos al reconocimiento

Un último camino a plantear es el de introducir en este reconocimiento realizado otros elementos musicales que en este trabajo no se han considerado, como por ejemplo la armadura, y que podrían afectar al mismo.

A la hora de desarrollar nuestro modelo de lenguaje, que es el punto donde estos elementos musicales entrarían en juego, únicamente hemos considerado las notas que se observan y sus predecesoras.

La idea sería considerar elementos musicales que van más allá de simplemente las notas, así como tratar de incorporar al reconocimiento la consideración de las diferentes tonalidades musicales y de como estas afectan al lenguaje musical.

Si se consiguiera incorporar estos elementos adecuadamente se conseguiría un modelo de lenguaje mucho más preciso, que no sólo tendría en cuenta la obra tratada, sino que además consideraría componentes del lenguaje musical que hasta el momento no se han considerado.

Cualquiera de estas cuatro posibilidades planteadas sería tanto un interesante camino por el que continuar un trabajo como el que se ha realizado, como una parte de trabajo que habría sido interesante incluir en caso de haberse podido desarrollar.

Cabe decir que algunas partes notablemente más ligadas a este trabajo como podría ser el uso de un conjunto de datos diferente para reafirmar nuestros resultados son sencillas de realizar, y las razones por las que no se han hecho o incluido en este trabajo es debido a otros factores como no tener acceso a las mismas, no tanto por su complejidad.

Decodificación con contexto

Para realizar la decodificación con contexto no es tan sencillo como añadir una imagen a izquierda y derecha de cada una de las imágenes de test y reconocerlas como se ha hecho en el entrenamiento con contexto.

Si hiciésemos esto obtendríamos el reconocimiento de tres imágenes en una y, aunque la central tendría contexto a ambos lados, las otras no lo tendrían. Esto también nos llevaría a reconocer imágenes repetidas si la concatenación es cíclica como se ha descrito previamente.

Hay dos caminos que podrían ser explorados en este aspecto. En ambos consideramos que las imágenes se han concatenado de forma cíclica.

La primera ruta de exploración es también la más sencilla y menos sólida. Habría que realizar el entrenamiento con contexto incluyendo un símbolo separador entre imágenes a reconocer.

Se reconocerían las imágenes con contexto y se debería reconocer en ellas este símbolo que marca la separación entre las imágenes. Tras esto se extraerían las transcripciones de las imágenes centrales y se compararían estas únicamente con las transcripciones correctas.

Los problemas que esto plantea son los siguientes:

- Primero, el propio reconocimiento de este simbolo separador. Si esto no se reconoce con una precisión del 100 % en todos los casos deja de poder extraerse esa transcripción asociada a la imagen central, y por tanto se pierde el próposito del propio experimento.
- Segundo, la introducción de un nuevo símbolo afecta al modelo de lenguaje. Esto no tiene por que ser necesariamente negativo ya que este símbolo únicamente representa comienzo o final de una imagen, pero al no poder controlar su efecto hay que considerarlo un problema de cara al futuro.

La segunda ruta de exploración es más compleja, pero deja mucho menos trabajo a cargo del reconocedor, y se basa en los propios datos y el funcionamiento de las redes para obtener un resultado.

En este caso lo que vamos a buscar es extraer directamente de la confMatrix generada en la decodificación únicamente aquellos *frames* de la imagen que nos interesan.

Hay dos opciones que se pueden hacer en este proceso:

- La primera, extraer estos *frames* de la imagen central. Dado que para cada imagen reconocida conocemos el tamaño de cada una de las imágenes que la componen, podemos extraer únicamente aquellas que corresponden a la imagen central.

  Con estas probabilidades extraídas podemos hacer el reconocimiento de manera externa a las herramientas que tenemos, y obtener únicamente la transcripción asociada a esa imagen central sin considerar las que la acompañan. De esta manera dicha imagen sería considerada con contexto en el

reconocimiento. Este contexto es únicamente a nivel óptico, no a nivel de modelo de lenguaje.

- La segunda, extraer los *frames* de una imagen dadas todas sus posiciones, es decir, extraer estos *frames* cuando la imagen es la imagen de la izquierda, cuando la misma es la imagen central y cuando es la imagen de la derecha.

  Una vez se hubieran obtenido las probabilidades asociadas a estos *frames* a partir de la confMatrix se combinarían las mismas, por ejemplo haciendo la media entre ellas. En la figura 7.1 se observa de forma esquemática este proceso.

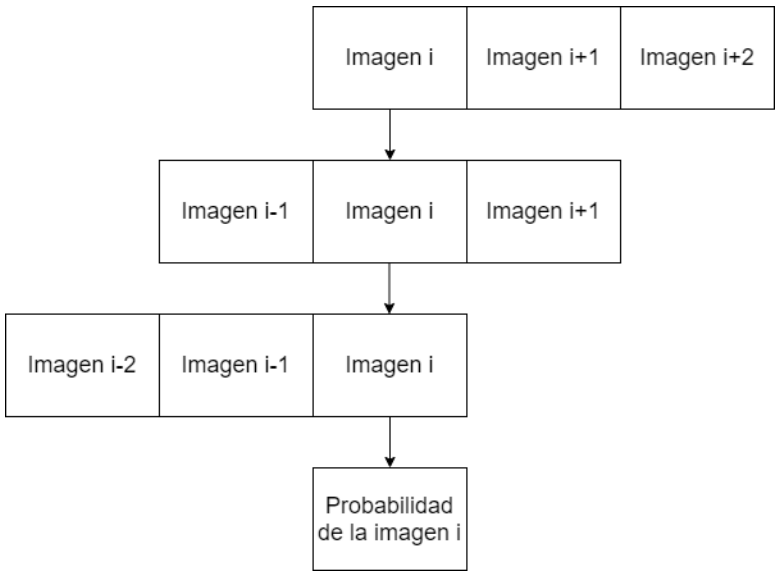

Figura 7.1: Esquema de la combinación de probabilidades de imágenes.

Esto tiene la finalidad de haber reconocido la imagen no sólo cuando esta es central, sino teniendo en cuenta cuando esta es también la imagen de la izquierda y de la derecha. Con esto se buscaría tener más solidez en el reconocimiento realizado.

Estos experimentos no se han llevado a cabo tanto por razones de tiempo como por razones tecnológicas, pues no hay ninguna herramienta ya diseñada que nos permita esta extracción de *frames* y nuestro trabajo se basaba en el tratamiento del modelo de lenguaje y el uso de herramientas de redes neuronales, no en la creación de una nueva o la modificación de las mismas.

Agradecimientos

Este trabajo ha sido parcialmente financiado por la Generalitat Valenciana con el proyecto PROMETEO/2019/121 (DeepPattern).

7.1 Glosario de términos

Este glosario incluye tanto términos técnicos como términos utilizados en un idioma extranjero cuya traducción no es adecuada o no existe en el contexto de la materia.

Aprendizaje automático - Conjunto de técnicas basadas en el aprendizaje de modelos de manera automática por parte de una computadora.

Red neuronal - Modelo basado en la estructura de las neuronas del cerebro humano.

Modelo de lenguaje - Modelo estadístico que representa las probabilidades de aparición de los elementos de un lenguaje.

Palabra - En el contexto de los modelos de lenguaje, unidad de medida del elemento único.

Modelo oculto de Markov - Modelo estadístico que asume que el sistema a modelar es un proceso de Markov de parámetros desconocidos.

HTR - Reconocimiento de texto manuscrito.

HMR - Reconocimiento de música manuscrita.

OCR - Reconocimiento óptico de carácteres, campo que estudia el reconocimiento de carácteres por medio de un computador.

OMR - Reconocimiento óptico de música, rama del campo del reconocimiento de carácteres dedicada al estudio de la música.

Lattice - Tabla interpolada que aproxima las relaciones arbitrarias de entrada-salida en los datos.

Script - Programa ejecutable de forma directa que contiene y aplica un algortimo.

Frame - Fragmento de una imagen que representa un instante en el tiempo en el proceso de reconocimiento.

LeakyReLu - Función de activación.

Leak - Pérdida asociada a la función de activación LeakyReLu.

Max-Pooling - Técnica de generalización utilizada en redes neuronales.

Down-Sampling - Técnica de reducción de muestras utilizada en redes neuronales.

Back Propagation - Algoritmo básico de entrenamiento que conforma las redes neuronales.

Back Propagation Through Time - Algoritmo de *Back Propagation* modificado para considerar el factor tiempo.

Dummy - Carácter que representa la ausencia de carácteres.

Adaptive pooling - Técnica dinámica para el tratamiento de muestras de diferentes tamaños en redes neuronales.

Dropout - Función de pérdida aplicada para conseguir una mejor generalización.

Corpus - Conjunto de datos de estudio.

Missa - Obra musical de carácter religioso.

Hardware - Componente físico de un sistema de computación.

Batch - Fragmento de un conjunto de datos utilizado para entrenar sistemas de forma paralela.

7.2 Anexo de código

A continuación vamos a mostrar en pseudocódigo el algoritmo de reconstrucción utilizado en el experimento con éxito.

SD = lista de duraciones sin altura

D = lista de duraciones

H = lista de alturas

Lista = {}

- *Desde* n = 0 *hasta* n = longitud de línea:

Para ordenación h d:

- - *Si* elemento [n] *pertenece a* H *y* elemento [n + 1] *pertenece a* D:

- - - *Formar símbolo* [n , n + 1], *añadir a* Lista

- - *Sino, Si* elemento [n] *pertenece a* SD:

- - - *Formar símbolo* [n], *añadir a* Lista

Para ordenación d h:

- - *Si* elemento [n] *pertenece a* D *y* elemento [n + 1] *pertenece a* H:

- - - *Formar símbolo* [n , n + 1], *añadir a* Lista

- - *Sino, Si* elemento [n] *pertenece a* SD:

- - - *Formar símbolo* [n], *añadir a* Lista

- *Devolver* Lista

Bibliografía

[1] J. Calvo-Zaragoza and I. Barbancho and L.J. Tardón and Ana M. Barbancho *Avoiding staff removal stage in optical music recognition: application to scores written in white mensural notation. Pattern Analysis and Applications, Volume 18, 933–943* 2015.

[2] J. Calvo-Zaragoza and A. H. Toselli and E. Vidal *Handwritten Music Recognition for Mensural notation with convolutional recurrent neural networks. Pattern Recognition Letters, Volume 128, Pages 115-121* 2019.

[3] J. Calvo-Zaragoza and A. H. Toselli and E. Vidal *Early Handwritten Music Recognition with Hidden Markov Models. 2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR), Pages 319-324* 2016.

[4] J. Calvo-Zaragoza and A. H. Toselli and E. Vidal *Handwritten Music Recognition for Mensural Notation: Formulation, Data and Baseline Results. 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), Pages 1081-1086* 2017.

[5] Graves A. *Long Short-Term Memory. In: Supervised Sequence Labelling with Recurrent Neural Networks. Studies in Computational Intelligence, Volume 385. Springer, Berlin, Heidelberg*, 2012.

[6] Graves, A. and Fernández, S. & Gomez, F. & Schmidhuber, J. *Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks. Proceedings of the 23rd International Conference on Machine Learning, Pages 369–376* 2006.

[7] Hasan, A. S. M and Islam, Saria and Rahman, M. *A Comparative Study of Witten Bell and Kneser-Ney Smoothing Methods for Statistical Machine Translation. JU Journal of Information Technology (JIT), Volume 1* 2012.

[8] R. Hecht-Nielsen *Theory of the Backpropagation Neural Network. Neural Networks for Perception, Pages 65-93* 1992.

[9] Hochreiter, Sepp and Schmidhuber, Jürgen *Long Short-Term Memory. Neural Computation, Volume 9, Pages 1735-1780* 1997.

[10] Mocholi C. *Development and experimentation of a deep learning system for convolutional and recurrent neural networks.* Universitat Politècnica de València. 2018.

[11] Nuñez-Alcover, A. & de León, P. J. & Calvo-Zaragoza, J. *Glyph and Position Classification of Music Symbols in Early Music Manuscripts. Pattern Recognition and Image Analysis, LNCS, Volume 11868* 2019.

[12] J. Puigcerver *A probabilistic formulation of keyword spotting.* [Tesis doctoral no publicada]. Universitat Politècnica de València. 2018.

[13] Rebelo, A. and Fujinaga, I. and Paszkiewicz, F. and Marçal, A. and Guedes, C. and Cardoso, J. *Optical music recognition: State-of-the-art and open issues. International Journal of Multimedia Information Retrieval, Volume 1, Pages 173–190* 2012.

[14] Reinhard Kneser and Hermann Ney *Improved backing-off for M-gram language modeling. 1995 International Conference on Acoustics, Speech, and Signal Processing, Volume 1, Pages 181-184* 1995.

[15] J. dos Santos Cardoso and A. Capela and A. Rebelo and C. Guedes and J. Pinto da Costa *Staff Detection with Stable Paths. IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 31, Number 6, Pages 1134-1139* 2009.

[16] J. S. Cardoso *Stable text line detection. 2009 Workshop on Applications of Computer Vision (WACV), Pages 1-5* 2009.

[17] Stanley F. Chena and Joshua Goodman *An empirical study of smoothing techniques for language modeling. Computer Speech & Language, Volume 13, Pages 359-394* 1999.

[18] Timothy C. Bell, John G. Cleary, Ian H. Witten *Text compression.* 1990.

[19] Williams, R. J and Zipser, D. *Gradient-based learning algorithms for recurrent networks and their computational complexity. Backpropagation: Theory, architectures, and applications, Chapter 13* 1995.

[20] B. Xu and N. Wang and T. Chen and M. Li *Empirical Evaluation of Rectified Activations in Convolutional Network.* 2015.