Reconocimiento holístico de partituras musicales

María Alfaro-Contreras, Jorge Calvo-Zaragoza y José M. Iñesta

Departamento de Lenguajes y Sistemas Informáticos, Universidad de Alicante, España
mac77@alu.ua.es, jcalvo@dlsi.ua.es, inesta@dlsi.ua.es
http://grfia.dlsi.ua.es

Resumen El reconocimiento de patrones con dependencia temporal es común en áreas como el reconocimiento del habla o el procesamiento del lenguaje natural. De manera análoga, encontramos el reconocimiento de texto o video en el campo de análisis de imágenes. Recientemente, las Redes Neuronales Recurrentes (RNN, por sus siglas en inglés) han sido ampliamente utilizadas para resolver estas tareas, arrojando buenos resultados, siguiendo un planteamiento conocido como reconocimiento *holístico*. Sin embargo, su aplicación en el campo del Reconocimiento Óptico de Música (OMR, por sus siglas en inglés) no es tan sencillo debido a la presencia de diferentes elementos en la misma posición horizontal, interrumpiendo así el flujo lineal temporal. En este artículo se estudia la capacidad de las RNN para aprender códigos que representan esta interrupción en partituras musicales homofónicas. Los resultados obtenidos demuestran que las formas serializadas para codificar el contenido musical propuestas son apropiadas para el OMR basado en RNN y que por tanto, merecen más estudio.

Palabras claves: Reconocimiento Óptico de Música · Aprendizaje Profundo · Reconocimiento holístico · Codificación musical

1. Introducción

El Reconocimiento Óptico de Música (OMR, por sus siglas en inglés) es el campo de investigación que estudia cómo decodificar computacionalmente la notación musical presente en imágenes de documentos. Su objetivo es convertir la gran cantidad de fuentes musicales escritas existentes en un formato codificado que permita su proceso computacional [2]. Existe una gran cantidad de documentos guardados en archivos privados y públicos, a menudo inaccesibles para el público general, a la espera de ser digitalizados. También hay muchos fondos digitalizados en portales especializados que solo están disponibles como imágenes, sin posibilidad de estudio o búsqueda de contenido. Dado que el grabado musical es un proceso tedioso y costoso, el OMR representa una alternativa para lidiar eficientemente con esta tarea.

Los enfoques tradicionales de OMR [2,14] se basan en una serie de etapas que caracteriza muchos sistemas de visión artificial adaptados a esta particular tarea: preprocesamiento de imágenes, detección individual de objetos musicales, reconstrucción de la semántica musical mediante el uso de conocimiento específico del campo y codificación de la salida en un formato simbólico adecuado.

Los recientes avances en el campo del Aprendizaje Automático, conocidos como Aprendizaje Profundo (DL, por sus siglas en inglés), han logrado excelentes resultados en tareas similares como el reconocimiento de texto, el reconocimiento del habla o la traducción automática. Esta mejora nos lleva a ser más optimistas sobre el desarrollo de sistemas OMR más precisos aprovechando dicha tecnología. La tendencia actual en estos campos es el uso de sistemas holísticos—también llamados de extremo a extremo o "end-to-end"—que lidian con el proceso de reconocimiento automático en una sola etapa, sin tener en cuenta explícitamente los pasos intermedios necesarios. Para llevar a cabo este enfoque, únicamente se necesitan pares de entrenamiento, los cuales consisten en imágenes acompañadas de sus correspondientes transcripciones [3,6].

Debido a razones de diseño, este enfoque, típicamente basado en Redes Neuronales Recurrentes, sólo puede formular la salida del sistema como secuencias unidimensionales. Esto encaja perfectamente en las tareas de procesamiento del lenguaje natural (reconocimiento de texto o voz, o traducción automática), ya que sus resultados consisten principalmente en secuencias de caracteres (o palabras). Sin embargo, su aplicación a la notación musical no es tan sencilla debido a la presencia de diferentes elementos compartiendo la misma posición horizontal. La distribución vertical de estos elementos interrumpe el flujo lineal cronológico (ver Figura 1). Este hecho complica la tarea de codificación, pudiendo causar importantes dificultades en el rendimiento de los sistemas de reconocimiento que hacen uso de las relaciones temporales de los elementos reconocidos.

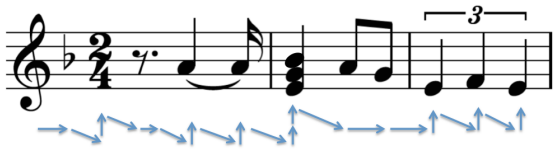


Figura 1: El proceso de reconocimiento de música no sigue un orden lineal de izquierda a derecha.

Aunque el problema podría simplificarse drásticamente al considerar que el proceso funcionaría al tratar cada pentagrama de manera independiente, un proceso que podría ser análogo al de los sistemas de reconocimiento de texto que descomponen el documento en una serie de líneas independientes, todavía habría que tratar con elementos que coinciden cronológicamente, como las notas que forman un acorde, los grupos de valoración especial o las marcas de expresión, por nombrar algunos.

Dentro del rango de complejidades de una partitura musical, una posible simplificación del problema, aplicable a muchas partituras, es asumir un contexto musical homofónico. En este caso, hay varias partes pero se mueven al mismo ritmo. De esta manera, pueden aparecer varias notas simultáneamente, pero únicamente como una sola voz. Así pues, todas las notas que comienzan al mismo tiempo duran lo mismo, por lo que la partitura se puede dividir en segmentos verticales que pueden contener uno o más símbolos musicales (ver Figura 2).

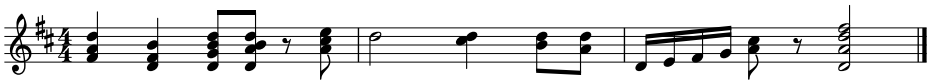


Figura 2: En la música homofónica, todas las notas que comienzan al mismo tiempo duran lo mismo.

Incluso dentro de este contexto simplificado, existe la necesidad de un sistema claro y estructurado de codificación de salida que evite las ambigüedades que la representación de una salida lineal puede mostrar en presencia de estructuras verticales en los datos (ver Figura 3).

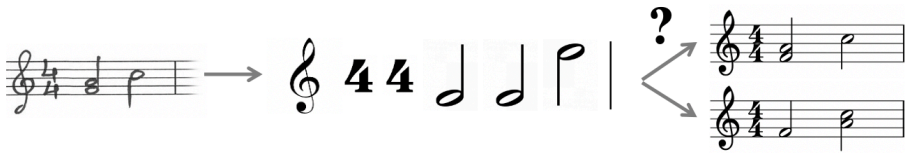


Figura 3: Las ambigüedades aparecen cuando los símbolos están apilados. Cuando aparecen dos notas juntas, que deben reproducirse al mismo tiempo, una secuencia lineal de símbolos sin marcas específicas puede ser interpretada de más de una manera.

Ya existen varios formatos estructurados para la representación y codificación de música, como los formatos de música basados en XML [8,11] que se centran en cómo debe codificarse la partitura para almacenar adecuadamente todo su contenido. Esa aplicación hace que sea inapropiado adoptarlos como salida para un sistema de reconocimiento óptico, ya que el formato XML contiene muchas marcas irrelevantes que dificultan que el sistema las genere al reconocer el contenido de la partitura (principalmente qué símbolos hay y dónde están en la partitura).

Por lo anterior, se propone el diseño de un lenguaje específico que represente una salida apropiada para el OMR holístico, basado en la serialización de los símbolos musicales que se encuentran en la música homofónica. La naturaleza secuencial de la lectura musical debe ser compatible con la representación de las alineaciones verticales de algunos símbolos, como los acordes. Además, esta representación debe ser fácil de generar por el sistema de reconocimiento, que analiza la entrada secuencialmente y produce una serie lineal de símbolos.

2. Materiales y métodos

2.1. Marco de reconocimiento

Para llevar a cabo la tarea de OMR de manera holística, se utiliza una Red Neuronal Recurrente Convolucional (CRNN, por sus siglas en inglés) que permite modelar la probabilidad posterior de generar símbolos de salida, dada una imagen de entrada. Se supone que las imágenes de entrada son pentagramas individuales, de igual manera que ocurre en el reconocimiento de texto al asumir líneas independientes [15]. Esta suposición no conlleva inconvenientes en la práctica, ya que los pentagramas de una partitura pueden aislarse fácilmente mediante métodos ya existentes [7].

Una CRNN consiste en un bloque de capas convolucionales seguido de un bloque de capas recurrentes [16]. El bloque convolucional es responsable de aprender a procesar la imagen de entrada, es decir, de extraer características de imagen relevantes para la tarea en cuestión, de modo que las capas recurrentes interpreten estas características en términos de secuencias de símbolos musicales. En el presente trabajo, las capas recurrentes se implementan como unidades del tipo "Bidirectional Long Short Term Memory" (BLSTM) [9].

Las activaciones de la última capa convolucional pueden ser interpretadas como una secuencia de vectores de características que representan la imagen de entrada, $\mathbf{x}$. Estas características alimentan la primera capa BLSTM y las activaciones unitarias de la última capa recurrente se consideran estimaciones de las probabilidades posteriores para cada vector:

$$P(\sigma|\mathbf{x}, f),\ 1 \leq f \leq F,\ \sigma \in \Sigma \tag{1}$$

donde F es el número de vectores de características de la secuencia de entrada y Σ es el conjunto de símbolos considerados, el cual debe incluir un símbolo "no-símbolo" requerido para separar convenientemente dos instancias consecutivas del mismo símbolo musical [9].

Dado que tanto los bloques convolucionales como los recurrentes pueden entrenarse a través del Descenso del Gradiente, utilizando el conocido algoritmo de Propagación hacia atrás [17], una CRNN puede entrenarse conjuntamente. Sin embargo, un conjunto convencional de entrenamiento de OMR holísitico solo proporciona, para cada imagen de pentagrama, su transcripción correspondiente, sin proporcionar ningún tipo de información explícita sobre la ubicación de los símbolos en la imagen.

Se ha demostrado que la CRNN se puede entrenar convenientemente sin esta información mediante el uso de la llamada función de pérdida de "Connectionist Temporal Classification"(CTC) [10]. El procedimiento de entrenamiento de CTC resultante es una forma de Esperanza-Maximización, similar al algoritmo de avance-retroceso utilizado para entrenar Modelos Ocultos de Márkov [13]. En otras palabras, la función CTC proporciona un medio para optimizar los parámetros de la CRNN de modo que es probable obtener la secuencia correcta dada una entrada. El uso del símbolo "no-símbolo" mencionado anteriormente para indicar una separación entre símbolos se considera esencial para un entrenamiento de CTC adecuado [10].

Una vez que la CRNN ha sido entrenada, una imagen de entrada puede ser decodificada en una secuencia de símbolos musicales $\hat{\mathbf{s}} \in \Sigma^*$. Primero, se calcula el símbolo más probable por fotograma:

$$\hat{\sigma}_i = \arg \max_{\sigma \in \Sigma} P(\sigma|\mathbf{x}, i),\ 1 \leq i \leq F \tag{2}$$

Después, se obtiene una secuencia de salida pseudo-óptima:

$$\hat{\mathbf{s}} = \arg \max_{s \in \Sigma^*} P(\mathbf{s}|\mathbf{x}) \approx \mathcal{D}(\hat{\sigma}_1, \ldots, \hat{\sigma}_F) \tag{3}$$

donde $\mathcal{D}$ es una función que primero combina todos los fotogramas consecutivos que contienen el mismo símbolo, y luego elimina el símbolo "no-símbolo" [9].

Un esquema gráfico del marco explicado anteriormente se da en la Figura 4.

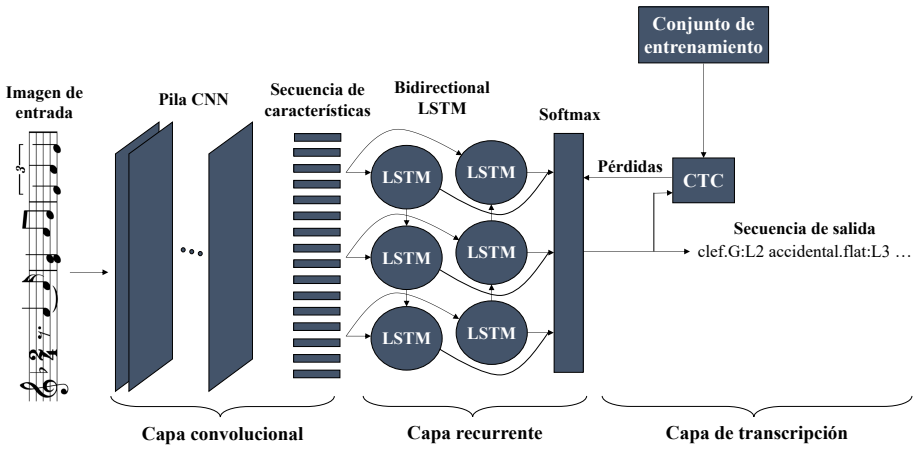


Figura 4: Esquema gráfico de la CRNN considerada para el enfoque holístico. La red se entrena utilizando la función de pérdida de CTC.

Este marco de trabajo es equivalente al utilizado en las tareas de reconocimiento de texto [16], cuya expresividad podría ser suficiente cuando se trabaja con partituras sencillas donde todos los símbolos tienen un orden lineal de izquierda a derecha. Sin embargo, tal y como se ha nombrado anteriormente, se pretende ampliar este enfoque para ser capaz de modelar partituras más complejas, como las de un contexto musical homofónico. En tal caso, pueden aparecer problemas como los acordes, donde varios símbolos comparten una posición horizontal. Como se ve en la Figura 3, una secuencia unidimensional no es lo suficientemente expresiva para esto. Es por ello que a continuación se presentan y describen cuatro propuestas de codificación para realizar el OMR holístico de partituras homofónicas.

Propuestas de serialización La investigación plasmada en este trabajo consiste en el estudio de cuatro representaciones deterministas, inequívocas y serializadas diferentes para codificar los escenarios que suceden en la música homofónica de manera que el sistema OMR sea más efectivo a la hora de reconocer imágenes de partituras musicales complejas. Para ello, se proponen cuatro tipos diferentes de representaciones musicales que difieren no en la codificación de los símbolos musicales en sí, sino en la forma en que se representan las distribuciones horizontales y verticales de estos símbolos. La gramática de estas codificaciones musicales debe ser determinista y sin ambigüedades, permitiendo analizar un documento dado de una sola manera.

La representación propuesta no hace suposiciones sobre el significado musical de lo que está representado en el documento que se está analizando, es decir, los elementos se identifican en un catálogo de símbolos musicales por la forma que tienen y dónde se colocan en la partitura. Esto se ha denominado "representación agnóstica", a diferencia de una representación semántica en la que los símbolos musicales se codifican de acuerdo con su significado musical real [5].

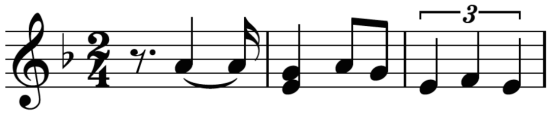
Como se mencionó anteriormente, la única diferencia entre los cuatro códigos musicales propuestos es cómo representan las dimensiones horizontal y vertical. Cada uno de los cuatro códigos tendrá uno o dos caracteres que indican si, al transcribir la partitura, el sistema debe avanzar hacia delante, de izquierda a derecha, o hacia arriba, de abajo hacia arriba.

Los cuatro códigos propuestos se describen a continuación:

- **Codificación de parada** (*Remain*): al transcribir la partitura, se supone que los diferentes símbolos musicales siguen un orden lineal de izquierda a derecha, excepto cuando están en la misma posición horizontal. En ese caso, están separados por una barra inclinada, "/". Este carácter indica al sistema que no debe avanzar hacia adelante sino que debe hacerlo hacia arriba (ver Figura 5.a). Este comportamiento es similar al retroceso de las máquinas de escribir. El carro avanza después de escribir y si queremos alinear dos símbolos necesitamos mantener el carro en una posición fija (moviéndolo hacia atrás una posición).

- **Codificación de avance** (*Advance*): este tipo de codificación utiliza un signo "+" para forzar al sistema a avanzar. De esta manera, cuando falta ese signo, se está codificando una distribución vertical (ver Figura 5.b).
- **Codificación entre paréntesis** (*Parenthesized*): cuando aparece una distribución vertical en la partitura, el sistema genera una estructura parentizada, como **vertical.start** musical_symbol . . . musical_symbol **vertical.end** (ver Figura 5.c).
- **Codificación verbosa** (*Verbose*): esta última codificación es una combinación de las dos primeras. Utiliza el signo "+" para indicarle al sistema que tiene que avanzar hacia delante, y el signo "/" para indicarle al sistema tiene que hacerlo hacia arriba (ver Figura 5.d). De manera que, en esta codificación, cada dos símbolos adyacentes hay un símbolo de separación que indica si el sistema debe permanecer en la misma posición horizontal o si debe avanzar a la siguiente.

Hay que tener en cuenta que los cuatro códigos son representaciones inequívocas de los mismos datos, por lo que son intercambiables y se pueden traducir entre ellos.



clef.G:L2 accidental.flat:L3 digit.4:L2 / digit.2:L4 rest.eighth:L3 dot:S3 slur.start:S2 / note.quarter:S2 slur.end:S2 / note.sixteenth:S2 verticalLine:L1 note.quarter:L1 / note.quarter:L2 note.beamedRight:S2 note.beamedLeft:L2 verticalLine:L1 note.quarter:L1 / bracket.start-S6 note.quarter:S1 / digit.3-S6 note.quarter:L1 / bracket.end-S6 verticalLine:L1

clef.G:L2 + accidental.flat:L3 + digit.4:L2 digit.2:L4 + rest.eighth:L3 + dot:S3 + slur.start:S2 note.quarter:S2 + slur.end:S2 note.sixteenth:S2 + verticalLine:L1 + note.quarter:L1 note.quarter:L2 + note.beamedRight:S2 + note.beamedLeft:L2 + verticalLine:L1 + note.quarter:L1 bracket.start-S6 + note.quarter:S1 digit.3-S6 + note.quarter:L1 bracket.end-S6 + verticalLine:L1

clef.G:L2 accidental.flat:L3 vertical.start digit.4:L2 digit.2:L4 vertical.end rest.eighth:L3 dot:S3 vertical.start slur.start:S2 note.quarter:S2 vertical.end vertical.start slur.end:S2 note.sixteenth:S2 vertical.end verticalLine:L1 vertical.start note.quarter:L1 note.quarter:L2 vertical.end note.beamedRight:S2 note.beamedLeft:L2 verticalLine:L1 vertical.start note.quarter:L1 bracket.start-S6 vertical.end vertical.start note.quarter:S1 digit.3-S6 vertical.end vertical.start note.quarter:L1 bracket.end-S6 vertical.end verticalLine:L1

clef.G:L2 + accidental.flat:L3 + digit.4:L2 / digit.2:L4 + rest.eighth:L3 + dot:S3 + slur.start:S2 / note.quarter:S2 + slur.end:S2 / note.sixteenth:S2 + verticalLine:L1 + note.quarter:L1 / note.quarter:L2 + note.beamedRight:S2 + note.beamedLeft:L2 + verticalLine:L1 + note.quarter:L1 / bracket.start-S6 + note.quarter:S1 / digit.3-S6 + note.quarter:L1 / bracket.end-S6 + verticalLine:L1

Figura 5: Extracto musical que presenta diferentes situaciones donde se producen alineaciones verticales y su transcripción utilizando las codificaciones propuestas. De arriba abajo: a) codificación de parada, b) codificación de avance, c) codificación entre paréntesis y d) codificación verbosa.

2.2. Marco experimental

A continuación, se describirá el marco experimental, dentro del cual se detalla el proceso de creación del corpus musical y el protocolo de evaluación considerado.

Generación del corpus Tal y como se nombró anteriormente, la tendencia actual para el desarrollo de sistemas OMR es utilizar técnicas de aprendizaje automático que pueden inferir la transcripción a partir de ejemplos correctos de la tarea conocidos como conjunto de pares (imagen, transcripción). Dada la complejidad de la notación musical, para que estas técnicas produzcan resultados satisfactorios es necesario usar un conjunto de tamaño suficiente. Para lograr esto, se ha desarrollado un sistema de generación automática de datos etiquetados [1] mediante el uso de técnicas de composición algorítmica [12]. El sistema desarrollado proporciona dos salidas: por un lado, la imagen de la partitura en formato PDF; por otro lado, la transcripción esperada de la partitura generada en cualquiera de las codificaciones descritas anteriormente. Con ambas salidas, se obtienen los pares necesarios para el algoritmo de aprendizaje automático.

Para el sistema de generación, se han implementado tres métodos diferentes de composición algorítmica, lo que permite obtener composiciones con características musicales dispares. Además, se ha limitado el rango de alturas codificables según la clave con el fin de lograr una partitura con la mayor coherencia musical posible, ya que al fin y al cabo, en las partituras, según la clave, hay un rango de alturas más común que otro. Se ha optado por un rango de 18 alturas diferentes para cada una de las claves. Las claves codificables por el sistema son: clave de sol, clave de fa en cuarta y clave de do en tercera.

A continuación, se describen los detalles técnicos de cada uno de los posibles métodos de composición algorítmica.

1. Generación aleatoria según la distribución normal. La distribución normal, distribución de Gauss, distribución gaussiana o distribución de Laplace-Gauss, es una distribución de probabilidad de variable continua. Es de gran aplicación en los campos de ingeniería, física y ciencias sociales debido a que permite modelar numerosos fenómenos naturales, sociales y psicológicos.
La gráfica de su función de densidad tiene una forma de campana y es simétrica respecto a la media (Figura 7). Esta curva se conoce como campana de Gauss y es el gráfico de una función gaussiana.
Se creará una distribución gaussiana con las 18 posibles alturas del rango determinado según la clave.

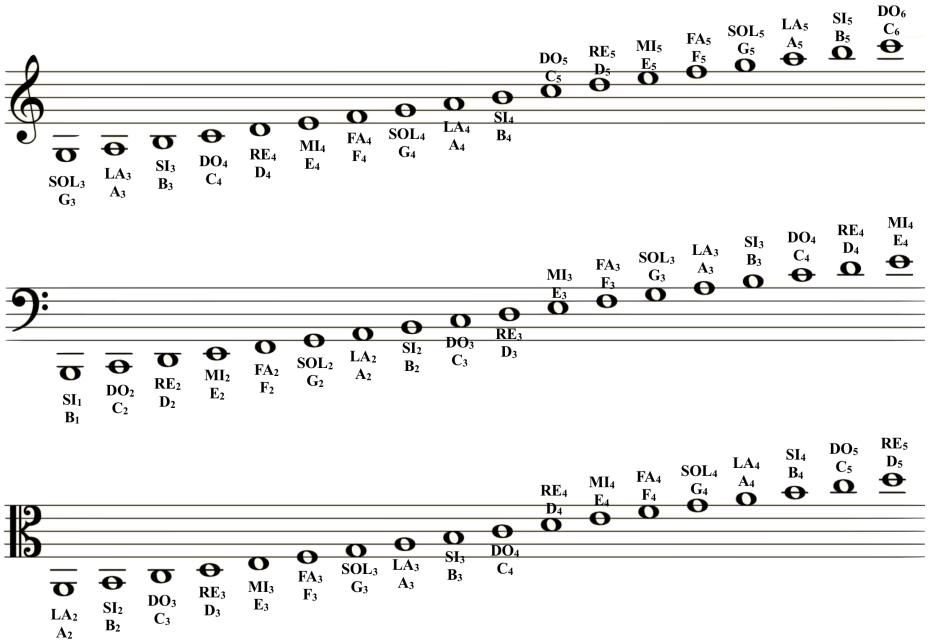
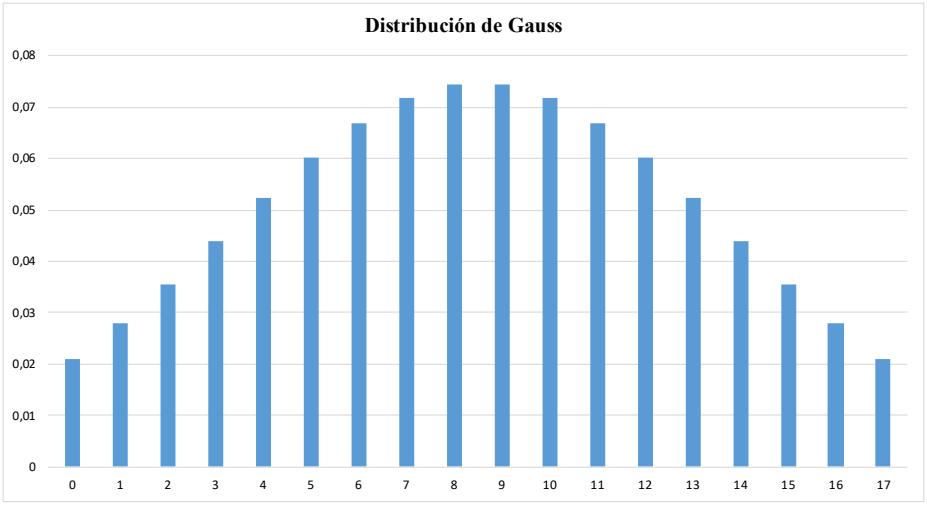


Figura 6: Rango de alturas codificables para cada clave. De arriba a abajo: a) clave de sol, b) clave de fa en cuarta y c) clave de do en tercera.



con $\bar{x} = 8,5$ y $\sigma = 5,34$

Figura 7: Distribución gaussiana para el rango de 18 alturas.

2. Paseo aleatorio. El paseo aleatorio, conocido como "random walk", es una formalización matemática de la trayectoria que resulta de hacer sucesivos pasos aleatorios.
En este trabajo, el paseo aleatorio partirá siempre desde la altura central del rango de alturas determinado para cada clave. Se podrán dar tres posibles pasos aleatorios (todos igual de probables):

1. Un paso hacia adelante. Es decir, se sumará uno a la altura de la nota anterior, avanzando a la siguiente altura del rango, la cual será la altura de la nota actual.
2. Un paso hacia atrás. Es decir, se restará uno a la altura de la nota anterior, retrocediendo a la anterior altura del rango, la cual será la altura de la nota actual.
3. No se dará ningún paso. Es decir, la altura actual será igual a la altura de la nota anterior.

Aparecerán situaciones en las que se desee avanzar o retroceder más de lo permitido, es decir, la altura a decidir quede fuera del rango marcado. En estas situaciones, se establecerán dos soluciones:

- Límite reflexivo: como su propio nombre indica, funciona como un espejo haciendo que el paso a dar sea el reflejo del que inicialmente se quería dar. Si se quería avanzar, se retrocederá, y viceversa. Es decir, la altura de la nota actual será la segunda del rango empezando o bien por el límite superior o bien por el inferior, según corresponda.
- Límite absorbente: la altura de la nota actual será la correspondiente al límite superior o inferior, según corresponda, del rango de alturas.

La solución a tomar se elegirá al azar, siendo ambas igual de probables.

3. Sonificación de la ecuación logística. La ecuación logística está definida por la ecuación 4.

$$x_{n+1} = r\ x_n(1 - x_n) \quad \text{donde } n = 0, 1, 2, 3, ... \tag{4}$$

Esta ecuación define una iteración, donde x_0 es igual a 0 y el parámetro r es un valor entre 0 y 4. El valor resultante siempre estará contenido en [0, 1].

Si el usuario decide este método, se le pedirá que introduzca un valor para el parámetro r entre 3,5 y 4, pues es el rango de valores para el cuál se generan las secuencias de notas más interesantes. Las secuencias para $3 \leq r \leq 3,5$ producen alternancias entre 2 o 4 alturas periódicamente, con poca variabilidad. Valores para $r < 3$ generan sencuencias de alturas constantes (unísonas) después de un corto periodo de transición.

El primer y último método generan composiciones musicales "estridentes", en las cuales es posible encontrar intervalos más grandes (disjuntos). Mientras que el segundo método, produce composiciones más melódicas, donde los intervalos siempre son conjuntos (unísonos o segundas).

Los tres métodos se han utilizado por igual al generar el conjunto de entrenamiento para que este no esté sesgado en favor de ningún estilo en particular.

Protocolo de evaluación Se ha generado un corpus de 8.000 partituras etiquetadas, cada una compuesta por un único pentagrama, utilizando el sistema de generación automática de datos etiquetados explicado anteriormente. El objetivo de la investigación es evaluar en qué medida la CRNN puede aprender las no linealidades en la línea temporal, y qué tipo de codificación produce los mejores resultados en la tarea de reconocimiento. Este corpus se usará para entrenar la red neuronal holística descrita en el marco de reconocimiento. Cada muestra será un par compuesto por la imagen con un pentagrama renderizado y su representación correspondiente con el formato impuesto por una de las cuatro codificaciones musicales propuestas, como en el ejemplo que se muestra en la Figura 5.

Se consideran las siguientes métricas de evaluación para medir el rendimiento del reconocimiento:

- **Tasa de error de secuencia, Seq-ER** (%): ratio de secuencias predichas incorrectamente (la secuencia de símbolos reconocidos tiene al menos un error).
- **Tasa de error de símbolo, Sym-ER** (%): se calcula como el número promedio de operaciones de edición elementales (inserciones, eliminaciones o sustituciones) necesarias para hacer hacer coincidir la secuencia predicha por el modelo con la secuencia de referencia.

La Seq-ER nos da una evaluación más justa porque no depende de la cantidad de símbolos necesarios para la codificación, lo que podría sesgar la Sym-ER a favor de las codificaciones menos verbosas. Por otro lado, la Seq-ER es una estimación mucho más pesimista del rendimiento porque un solo error arruina la salida. Un modelo puede tener una Sym-ER muy baja, por ejemplo del 1 %, pero si los símbolos incorrectos se distribuyen equitativamente, podemos tener una Seq-ER muy alta.

Debido a esto, en la siguiente sección, la Seq-ER se utilizará para comparar el rendimiento del modelo neural utilizando las diferentes codificaciones (aunque también se estudian los resultados proporcionados por la Sym-ER), y luego se utilizará la mejor representación para una evaluación cualitativa del OMR en algunos fragmentos de música real seleccionados.

3. Resultados

Se han entrenado cuatro modelos diferentes, cada uno correspondiente a una de las cuatro propuestas de codificación, utilizando el corpus de 8.000 partituras generado. De todo el conjunto, 7.000 se han utilizado para el entrenamiento y las 1.000 partituras restantes se han usado para la validación. Hay que tener en cuenta que el objetivo principal de esta experimentación es comparar las diferentes codificaciones, por lo que estamos interesados en sus límites de error.

Primero, se muestra la convergencia de los modelos aprendidos, es decir, cuántas épocas de entrenamiento necesitan los modelos para ajustar sus parámetros adecuadamente. Esto proporciona pistas sobre la complejidad del proceso de aprendizaje de cada codificación llevado a cabo por la CRNN. Las curvas obtenidas por cada tipo de codificación se muestran en las Figuras 8 (Seq-ER) y 9 (Sym-ER). A partir de las curvas podemos observar que los cuatro modelos convergen relativamente rápido, necesitando menos de 20 épocas para llegar al punto en el que cambia la pendiente.

	Remain	*Advance*	*Parenth.*	*Verbose*
Tasa de error de secuencia (%)	35.8	29.9	33.1	48.9
Tasa de error de símbolo (%)	1.18	0.74	0.77	1.79

Tabla 1: Análisis de los límites de error: la mejor precisión alcanzada por cada codificación sobre el conjunto de validación. **Remain** denota la codificación de parada, **Advance** la de avance, **Parenth.** es para la parentizada y **Verbose** para la verbosa.

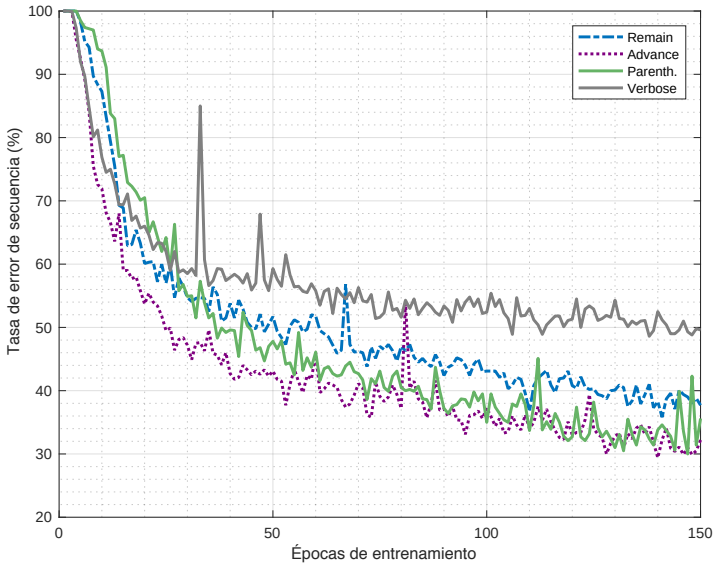


Figura 8: Análisis de convergencia: tasas de error de secuencia (Seq-ER) sobre el conjunto de validación, con respecto a la época de entrenamiento de la red. Se han usado las mismas abreviaturas que en la Tabla 1.

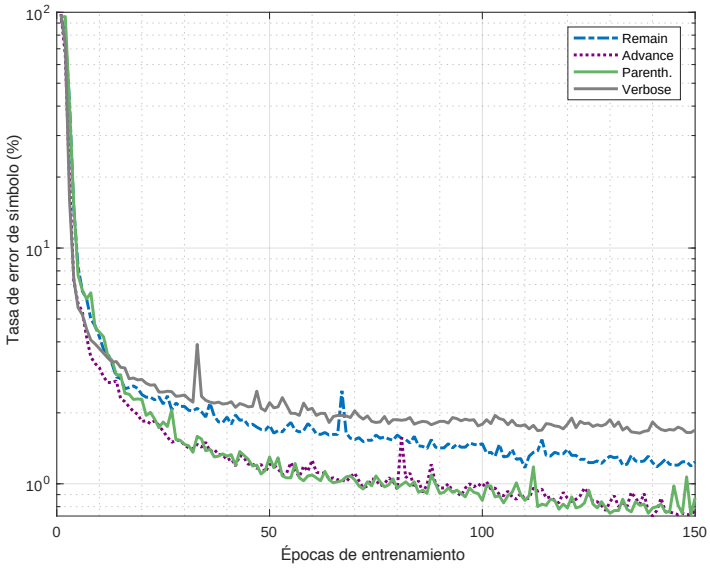


Figura 9: Tasas de error de símbolo (Sym-ER) sobre el conjunto de validación, con respecto a la época de entrenamiento de la red. Se ha usado unidades logarítmicas para representar los valores de error. Se han usado las mismas abreviaturas que en la Tabla 1.

4. Discusión

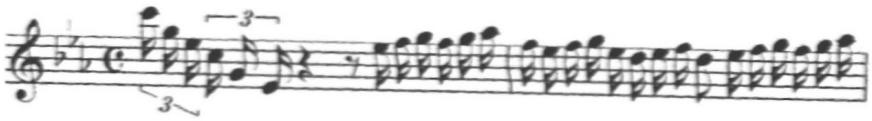
Observando las Figuras 8 y 9, se comprueba una clara correlación entre los errores Sym-ER y Seq-ER para las cuatro codificaciones propuestas: la codificación con la mayor Seq-ER es también la que tiene mayor Sym-ER, y viceversa. De este modo, podemos descartar que la Sym-ER esté sesgando la evaluación en función de la verbosidad de la codificación que se use, por lo que podemos utilizarla para una evaluación adecuada del comportamiento del sistema.

También se observa que la codificación de la salida para esta tarea OMR tiene un impacto en el entrenamiento y, en consecuencia, en el rendimiento del reconocimiento. La codificación de avance logró los mejores resultados para ambas métricas: alcanza la Sym-ER y la Seq-ER más bajas. Los resultados fueron muy alentadores, ya que alrededor del 70 % de las partituras de validación fueron reconocidas sin errores, y además, la tasa de error de reconocimiento de símbolos es inferior al 1 % de los símbolos predichos.

4.1. Evaluación cualitativa

Todas las partituras usadas en la experimentación anterior son sintéticas. Es por ello que se ha llevado a cabo un experimento adicional para transcribir el contenido de partituras reales tomadas de un repositorio de partituras históricas (RISM, *Répertoire International des Sources Musicales*[1]). Dos fragmentos musicales de RISM se introdujeron en el sistema como imágenes independientes (nunca antes vistas) para la evaluación cualitativa del enfoque propuesto. Las imágenes de este experimento han sido distorsionadas para así hacer frente a los desafíos de un escenario real, como en [4].

Las Figuras 10 y 11 muestran las imágenes y las secuencias predichas usando el modelo de codificación *Advance*. A partir de ellas, podemos decir que los resultados obtenidos (Sym-ER = 5.3 % y 9.5 %, respectivamente) son bastante precisos, incluso teniendo en cuenta que las imágenes pertenecen a una base de datos diferente (nunca antes vista) que la utilizada para el entrenamiento, se renderizaron utilizando diferentes métodos, presentan distorsiones y son de baja calidad. Estos hechos explican que las tasas de error sean más altas que las presentadas en la Tabla 1, aún así el rendimiento muestra una alta precisión en el reconocimiento pudiendo mejorarse agregando imágenes distorsionadas al conjunto de entrenamiento.



(a) Imagen de entrada.

```
clef.G:L2 + accidental.flat:L3 + accidental.flat:S4 + accidental.flat:S2 + metersign.C:L3
+ [bracket.start:S0] note.sixteenth:L7 + digit3:S6 note.sixteenth:S4 + note.sixteenth:S4 +
[bracket.end:S-1] note.sixteenth:S3 bracket.start:S6 + note.quarter:L2 + note.sixteenth:L1
bracket.end:S6 + rest.quarter:L3 + rest.eighth:L3 + note.sixteenth:S4 + note.sixteenth:L5
+ note.sixteenth:S5 + note.sixteenth:L5 + note.sixteenth:S5 + note.sixteenth:L6
+ verticalLine:L1 + note.sixteenth:L5 + note.sixteenth:S4 + note.sixteenth:L5 +
note.sixteenth:S5 + note.sixteenth:S4 + note.sixteenth:L4 + note.sixteenth:S4
+ note.sixteenth:L5 + note.eighth:L4 + note.sixteenth:S4 + note.sixteenth:L5 +
note.sixteenth:S5 + note.sixteenth:L5 + note.sixteenth:S5 + note.sixteenth:L6 +
verticalLine:L1
```

(b) Secuencia predicha con errores en negrita y símbolos faltantes entre paréntesis.

Figura 10: Evaluación cualitativa del enfoque OMR para el fragmento *incipit RISM ID no. 110003911-1_1_1*, produciendo un Sym-ER de 5,3 %.

[1] http://www.rism.info/home.html

(a) Imagen de entrada.

clef.G:L2 + accidental.sharp:L5 + accidental.sharp:S3 + accidental.sharp:S5 + digit.8:L2 digit.6:L4 + note.beamedRight2:S2 + **note.beamedRight1**:S3 + note.beamedLeft1:L4 + note.quarter:S4 + note.eighth:S3 + verticalLine:L1 + note.quarter:L4 note.quarter:L5 + dot:S4 *[dot:S5]* + *[note.quarter:S3]* note.quarter:S4 + *[dot:S3]* dot:S4 + verticalLine:L1 + rest.eighth:L3 + note.beamedRight1:L4 + note.beamedLeft1:S3 + note.quarter:L3 + note.eighth:S2 + verticalLine:L1 + note.beamedRight1:L2 + **note.beamedRight1**:S2 + note.beamedLeft1:L3 + note.quarter:L1 note.quarter:L2 note.quarter:L3 + dot:S1 dot:S2 dot:S3

(b) Secuencia predicha con errores en negrita y símbolos faltantes entre paréntesis.

Figura 11: Evaluación cualitativa del enfoque OMR para el fragmento *incipit RISM ID no. 000136642-1_1_1*, produciendo un Sym-ER de 9,5 %.

Como se puede observar en la Figura 11, en el último compás hay un acorde de tres negras con puntillo que el modelo interpreta como dos grupos claramente diferenciados: las tres negras comparten el mismo espacio horizontal y de igual manera lo hacen los tres puntillos, pero estos se colocan a la derecha de las negras, lo que significa que hay un avance horizontal, el cual está siendo codificado correctamente por el modelo al colocar este el carácter "+" (avance) entre estos dos grupos. En consecuencia, esto nos lleva a la conclusión de que el modelo puede interpretar las relaciones verticales y horizontales presentes en la partitura y aprender a codificarlas.

5. Conclusiones

En este trabajo, se ha estudiado la idoneidad del uso de redes neuronales para resolver la tarea de OMR mediante un enfoque holístico a través de un escenario controlado de partituras sintéticas homofónicas, presentando y analizando cuatro codificaciones diferentes para la salida.

Tal y como se mostró en los experimentos, las formas serializadas para codificar el contenido musical propuestas demuestran ser apropiadas para la tarea de OMR basada en DL, ya que el proceso de aprendizaje es exitoso y eventualmente se alcanzan cifras bajas para la tasa de error de símbolo. Además, se demuestra que la elección de la codificación tiene cierto impacto en el límite inferior que las tasas de error pueden alcanzar, lo que se correlaciona casi directamente con la tendencia de las curvas de aprendizaje. Estos hechos refuerzan la afirmación inicial presentada en este trabajo sobre una mayor consideración de la codificación de la salida para la tarea de OMR dentro del paradigma DL de enfoque holístico.

Información complementaria

Este trabajo cuenta con el apoyo del proyecto del ministerio español HISPAMUS TIN2017-86576-R, parcialmente financiado por la Unión Europea.

Referencias

1. Alfaro Contreras, M.: Construcción de un corpus de referencia para investigación en reconocimiento automático de partituras musicales. Trabajo de fin de grado, Universidad de Alicante (2018)
2. Bainbridge, D., Bell, T.: The challenge of optical music recognition. Computers and the Humanities **35**(2), 95–121 (2001)
3. Baró, A., Riba, P., Calvo-Zaragoza, J., Fornés, A.: From optical music recognition to handwritten music recognition: A baseline. Pattern Recognition Letters **123**, 1–8 (2019)
4. Calvo-Zaragoza, J., Rizo, D.: Camera-primus: Neural end-to-end optical music recognition on realistic monophonic scores. In: Proceedings of the 19th International Society for Music Information Retrieval Conference, ISMIR 2018, Paris, France, September 23-27, 2018. pp. 248–255 (2018)
5. Calvo-Zaragoza, J., Rizo, D.: End-to-end neural optical music recognition of monophonic scores. Applied Sciences (4), 606 (2018)
6. Calvo-Zaragoza, J., Toselli, A.H., Vidal, E.: Early handwritten music recognition with hidden markov models. In: 15th International Conference on Frontiers in Handwriting Recognition. pp. 319–324. Institute of Electrical and Electronics Engineers Inc. (2017)
7. Campos, V.B., Calvo-Zaragoza, J., Toselli, A.H., Vidal-Ruiz, E.: Sheet music statistical layout analysis. In: 15th International Conference on Frontiers in Handwriting Recognition. pp. 313–318 (2016)
8. Good, M., et al.: MusicXML: An internet-friendly format for sheet music. In: XML Conference and Expo. pp. 03–04 (2001)
9. Graves, A.: Supervised sequence labelling with recurrent neural networks. Ph.D. thesis, Technical University Munich (2008)
10. Graves, A., Fernández, S., Gomez, F., Schmidhuber, J.: Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In: Proceedings of the 23rd International Conference on Machine Learning. pp. 369–376. ICML '06, ACM, New York, NY, USA (2006)
11. Hankinson, A., Roland, P., Fujinaga, I.: The music encoding initiative as a document-encoding framework. In: Proceedings of the 12th International Society for Music Information Retrieval Conference. pp. 293–298 (2011)
12. Miranda, E.R.: Composing Music with Computers. Focal Press (2001)
13. Rabiner, L., Juang, B.H.: Fundamentals of Speech Recognition. Prentice-Hall, Inc., Upper Saddle River, NJ, USA (1993)
14. Rebelo, A., Fujinaga, I., Paszkiewicz, F., Marçal, A., Guedes, C., Cardoso, J.: Optical music recognition: State-of-the-art and open issues. International Journal of Multimedia Information Retrieval **1** (10 2012). https://doi.org/10.1007/s13735-012-0004-6
15. Romero, V., Sanchez, J.A., Bosch, V., Depuydt, K., de Does, J.: Influence of text line segmentation in handwritten text recognition. In: 2015 13th International Conference on Document Analysis and Recognition (ICDAR). pp. 536–540. IEEE (2015)

16. Shi, B., Bai, X., Yao, C.: An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence **39**(11), 2298–2304 (2017)
17. Williams, R.J., Zipser, D.: Gradient-based learning algorithms for recurrent networks and their computational complexity. In: Chauvin, Y., Rumelhart, D.E. (eds.) Back-propagation: Theory, Architectures and Applications, chap. 13, pp. 433–486. Hillsdale, NJ: Erlbaum (1995)