Trabajo de la asignatura

Aprendizaje Profundo (AP)

Autor: Antonio Gómez Giménez
Email: i72gogia@uco.es

Córdoba
(2022/2023)

UNIVERSIDAD D CÓRDOBA

Índice:

1. Introducción:

En este trabajo se pretende reforzar los conocimientos obtenidos en la asignatura, para ello, se pretende realizar la elección de un artículo científico para realizar el análisis del mismo y poder entender de una forma más práctica la aplicación de las redes neuronales profundas aplicadas en un entorno real.

A lo largo de este trabajo se explicará el por qué de la elección del artículo que se pretende analizar (beneficios a la sociedad, gustos personales, origen de la investigación, proyección de la misma, etc).

Se comentarán ciertos aspectos del propio artículo como por ejemplo, qué problema se pretende resolver, acotando y especificando el mismo. Para dicho problema se comentarán las distintas opciones disponibles para su resolución y casos anteriores para resolverlos que pueden servir de base para dicha investigación.

Finalmente, se comentará como se ha abordado el problema y la implementación llevada a cabo, explicando a su vez los resultados observados de la misma y si son aceptables o no.

Se realizará una conclusión sobre el artículo científico y qué posibles mejoras se podría realizar sobre la implementación y sobre el problema en general a abordar.

2. Elección del artículo:

Tras revisar distintos artículos científicos, se optó por el siguiente artículo científico **"Aplicación de técnicas de aprendizaje profundo al reconocimiento óptico de partituras SATB"**, basado en el reconocimiento de partituras. Este artículo fue realizado por Martin Morita Hernandez y Francisco Fernandez de Vega (Departamento de Tecnología de los Computadores y de las Comunicaciones Universidad de Extremadura Mérida, España), en conjunto con Juan Villegas Cortez (Departamento de Sistemas Universidad Autónoma Metropolitana, Cd. de México, México).
Algunas palabras claves de dicho artículo son Aprendizaje profundo, OMR, Reconocimiento óptico.

La principal elección de dicho artículo se basa en distintas razones:

- **Origen de la investigación.** Dicha investigación tiene origen español y latino, de tal forma que me parecía interesante comprobar y comprender un poco mejor cómo se realizaba dicha investigación (aunque fuera de manera resumida en un artículo).

- **Beneficios para la sociedad.** El problema que se pretende resolver me parecía interesante y con cierta relevancia para la sociedad, ya que el reconocimiento de partituras conlleva ciertas mejoras, como puede ser por ejemplo, una mejora en la enseñanza en los conservatorios, entre otras muchas aplicaciones posibles.

- **Proyección de la misma.** Este objetivo que se pretende resolver, tiene gran proyección de futuro, ya que sirve de base para permitir resolver problemas más complejos. Por ejemplo, si se consigue resolver el problema de reconocimiento de partituras, se podrían abarcar problemas como por ejemplo, corrección y mejora de partituras escritas a mano por alumnos, o incluso la creación de partituras nuevas basándose en melodías encontradas, etc.

- **Gustos personales.** Este tema me llamó personalmente la atención, ya que actualmente, estoy aprendiendo por mi cuenta a usar el piano, aprendiendo solo a entender el funcionamiento de la partitura (notas, armadura, tiempos, etc). Por ello pensé que si existiera una aplicación basada en este concepto de investigación, la gente amateur tendría más opciones para aprender sobre música, ya que toda ayuda para el aprendizaje es siempre bienvenida.

3. Problema planteado en el artículo:

Ahora sí centrándonos en el artículo, se pretende explicar el problema que se plantea resolver en dicho artículo, por ello, primeramente vamos a analizar el título del mismo.

El título es **"Aplicación de técnicas de aprendizaje profundo al reconocimiento óptico de partituras SATB",** lo que nos lleva a deducir que se pretende aplicar redes neuronales profundas para el reconocimiento de partituras, cabe destacar que cuando se refiere a SATB, se refiere a Soprano, Contralto, Tenor y Bajo, siendo SATB su acrónimo. Este acrónimo se usa para clasificar coros dependiendo de las voces (mixto, niños, adultos, etc).

Una vez entendemos el título del artículo, se procede a explicar el resumen y la introducción de dicho artículo donde se explica en detalle el problema global y el objetivo que se pretende resolver en dicho artículo.

El problema más general al que nos encontramos es el reconocimiento de partituras, esto entra dentro del ámbito OMR (Optical Music Recognition). Cabe destacar que este no es un problema sencillo ya que dentro de una partitura nos podemos encontrar muchas figuras como por ejemplo, el pentagrama, las notas musicales, la posición de las mismas, las armaduras, con las claves y tempos, y otras estructuras que se usan de apoyo a la partitura. De hecho, para ver la complejidad a la que pueden llegar algunas partituras, se muestra a continuación dos tipos distintos de partituras, una más sencilla y otra más compleja, hay que tener en mente siempre que deben poder ser analizadas por un computador. Los ejemplos son los siguientes:

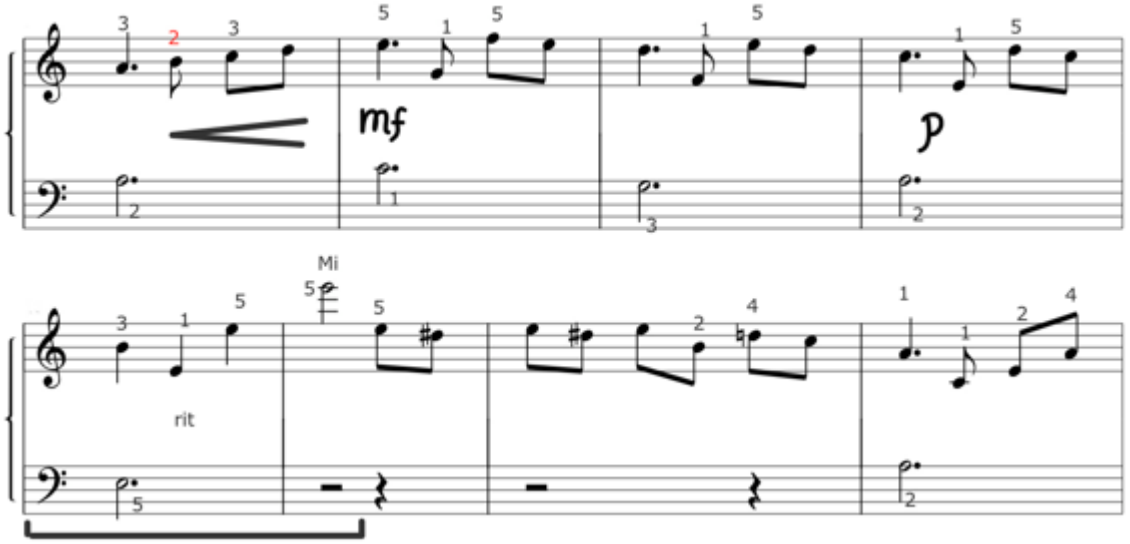

Figura 1: Fragmento de Für Elise (Arreglo por Ilsy Sánchez/ Piano Allegro Academia Virtual)

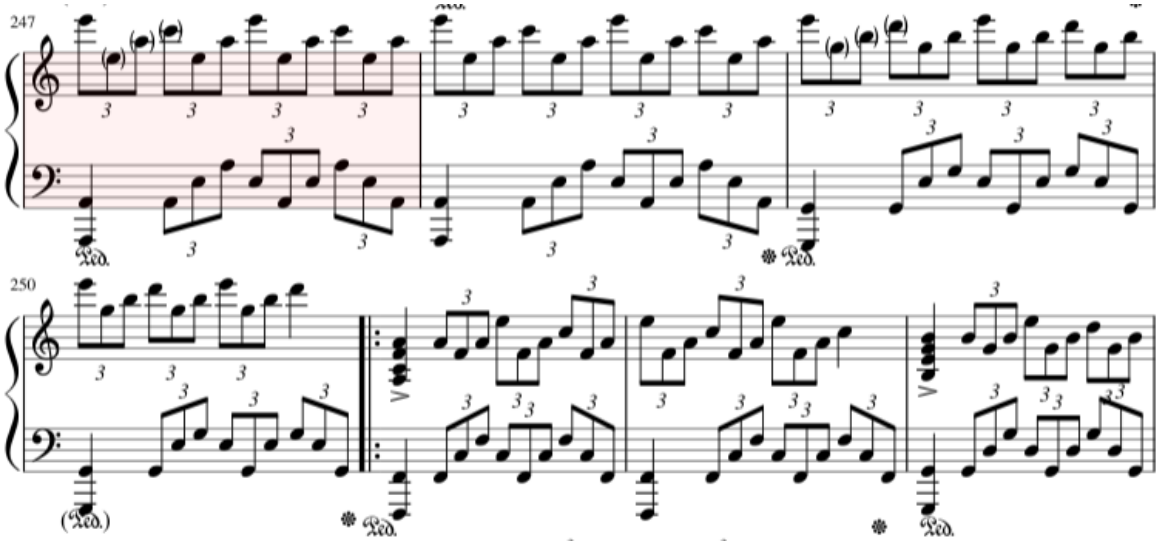

Figura 2: Interstellar-Suite (Hans Zimmer)

Como podemos observar, incluso ya en la primera figura, puede ser complejo el análisis de una partitura para una máquina, siendo necesario detectar todo lo explicado anteriormente. De hecho, si nos centramos en la figura 2, el grado de complejidad es muy elevado, pudiendo encontrar corcheas unidad, notas enlazadas entre sí, acentos, repeticiones de parte del pentagrama, etc. Por lo tanto, nos podemos dar cuenta así que abarcar todo el problema de golpe puede ser un trabajo un tanto inviable.

Por ello, en este artículo se analiza el este problema y llegan a la conclusión de querer ayudar a **La Sociedad Internacional para la Recuperacion de Informacion Musical** (ISMIR, International Society for Music Information Retrieval) considerándolo como una de las áreas relevante, y difícil de abordar.

Lo que buscan es el reconocimiento de partituras pero centrándose en partituras escritas a mano, en vez de en ordenador, aumentando en mayor medida el grado de dificultad. Este objetivo se debe a que buscan apoyar en la educación en conservatorios profesionales, buscando una mejora en la aplicación Sharpmony, usando esta herramienta como apoyo a los estudiantes a la hora de componer.

Especificando un poco más pero sin entrar en mucho detalle, se comenta que dichos estudiantes de conservatorios deben ser capaces de realizar armonías, en concreto, armonías basadas en armonía clásica (SATB). Este tipo de armonías aparecen gracias a Batch debido a las composiciones corales, donde existe superposición de voces. Por lo tanto, a este tipo de armonía, también se le puede llamar armonía coral a cuatro voces (Soprano, Contralto, Tenor y Bajo (SATB)).

Un ejemplo de ejercicio donde estos estudiantes aplican este tipo de armonía es el siguiente:

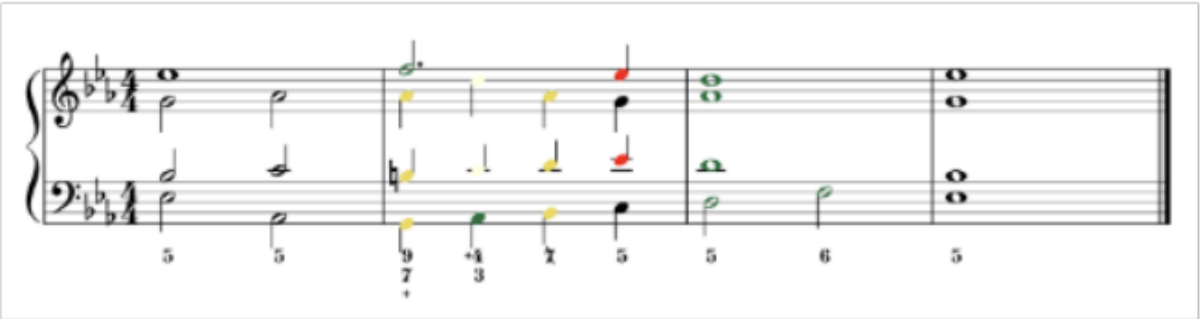

Figura 3: Ejemplo de ejercicio hecho por alumnos de conservatorio (corregido por sharpmony)

Esta figura, nos enseña un ejemplo de armonía desarrollada por el estudiante, donde se pueden apreciar ciertos fallos, en amarillo claro se muestra un error con distancias mayores de octava entre voces, en rojo quintas-octava en paralelo, amarillo oscuro error de cifrado y verde acordes incorrectos.

El ejercicio fue corregido con la herramienta sharpmony, cuya herramienta incorpora una IA que se utiliza por varias instituciones para la docencia de la armonía.

Este ejemplo previo nos deja entrever, porque puede ser útil una herramienta que permita reconocer las partituras realizadas a mano por los alumnos para llegar a detectar estos errores y que el alumno pueda corregirlos, apoyándose en sharpmony.

Para tener una mayor idea de como se puede dividir el problema de SATB aplicando técnicas de Inteligencia Artificial, en el artículo los clasifican de la siguiente manera:

- Corrección automática de ejercicios SATB.
- Composición automática de corales SATB.
- Reconocimiento automático de ejercicios SATB escritos a mano (OMR).

Como se puede apreciar, el problema que se pretende abarcar es el último donde se deja de lado las partituras a ordenador y la corrección y composición teniendo en cuenta SATB.

La partitura vista en la figura tres no es un buen ejemplo del problema a analizar ya que está realizada a ordenador, se muestra en la siguiente figura un mejor ejemplo, donde un estudiante realiza un ejercicio en una libreta pautada:

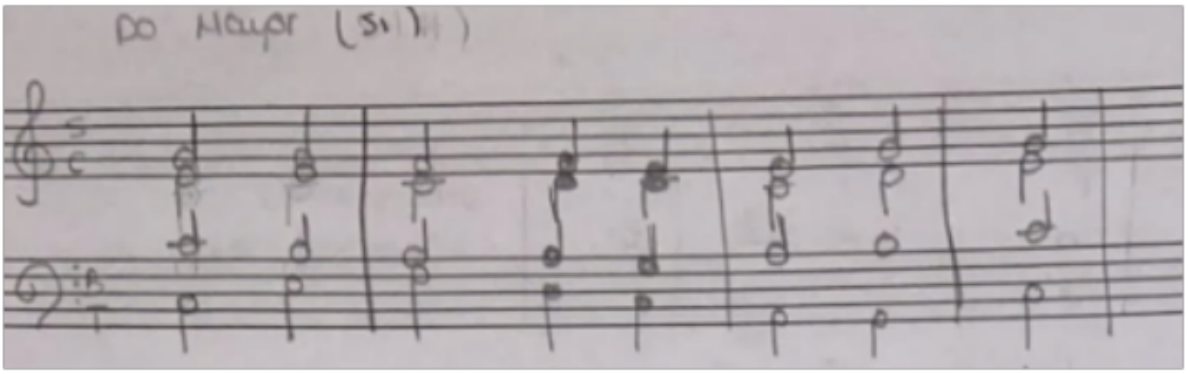

Figura 4: Ejemplo de ejercicio hecho por alumnos de conservatorio a mano en libreta pautada

Antes de comenzar con la resolución del problema y de la implementación, en dicho artículo, se realizó un estudio previo para saber cómo abordar este problema, es decir, como la imagen de arriba puede ser detectada y extraer la información necesaria de la partitura a un formato estándar reconocido como **MIDI** o **MusicXML**.

Como el problema trata sobre reconocimiento de objetos, se optó por el enfoque de las redes neuronales convolucionales (CNN, Convolutional Neuronal Networks), basándose en el aprendizaje profundo. Hay diversos algoritmos de este tipo que compiten entre sí como puede ser YOLO, SSD o RetinaNet, clasificados como modelos de detección de una etapa, de tal forma que, su detección es mucho más rápida. Existen también modelos que se basan en detección en dos etapas, como Faster R-CNN, R-CNN o R-FCN que tienen precisiones más altas a costa de la velocidad del mismo. Resumiendo de forma más clara:

Mayor velocidad/Menor precisión:

- YOLO [1]
- SSD [2]
- RetinaNet [3]

Mayor precisión/Menor velocidad:

- Faster R-CNN [4]
- R-CNN [5]
- R-FCN [6]

El problema de reconocimiento de partituras realizadas a ordenador se suele resolver con procedimientos tradicionales de segmentación y clasificación, pero para el caso en el que nos encontramos, esto es un problema, ya que si hay un error en la parte de segmentación, dicho error se retransmite a la clasificación. Por ello, en este artículo, tras realizar una investigación sobre otros trabajos que intentan resolver este problema, llegan a la conclusión de crear una nuevo clasificador de redes neuronales convolucionales llamado **Mask R-CNN** [7] que busca una mejor precisión de reconocimiento de partituras escritas a mano donde se busca simplificar el proceso, prediciendo una máscara binaria para cada clase de forma independiente, en los casos investigados donde se usan otras redes, deben de dividir el proceso en una primera red para la detección de las cabezas de las notas y posteriormente especificar el tipo de nota encontrada.

En el siguiente apartado se explica con mayor profundidad la red **Mask R-CNN**.

4. Resolución del problema y metodología implementada:

Para poder solucionar el problema, se pretende realizar una metodología en cuatro pasos. De forma resumida, se basa en la obtención de los datos en un repositorio con imágenes de partituras escritas a mano, a las cuales se le aplica una segmentación. Tras obtener dichas imágenes procesadas, se aplican para el entrenamiento de la red neuronal convolucional profunda(la cual está pre-entrenada). Así, finalmente se obtienen los símbolos clasificados de las imágenes. Este proceso es una primera etapa con datos de entrenamiento, posteriormente se realiza otra etapa con imágenes desconocidas para la red.
En la siguiente imagen se puede apreciar la estructura general de la metodología que se pretende implementar:

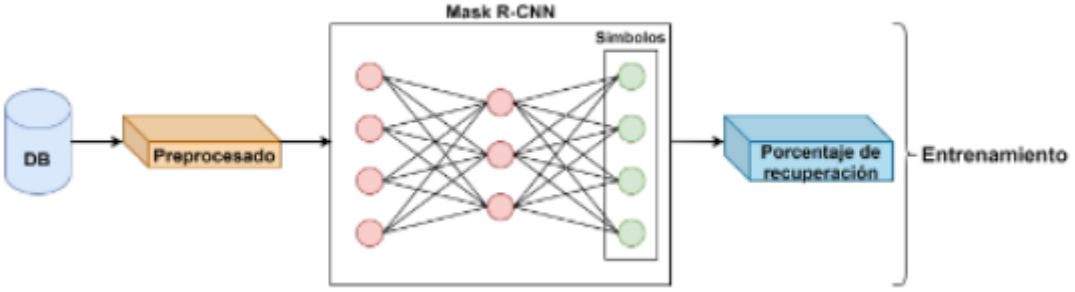

Figura 5: Metodología a implementar

Para mayor entendimiento, se procede a explicar cada etapa de una manera más detallada:

- **Conjunto de datos (DB).** Para poder entrenar la red neuronal convolucional profunda, es necesario tener datos para ello. Por ende, se creó un conjunto de datos basándose en fotografías realizadas sobre los cuadernos pautados de los estudiantes de conservatorio (conteniendo partituras SATB), dichas imágenes fueron suministradas por los profesores de armonía que participaban en dicho proyecto, obteniendo un total de **100 imágenes** donde se pueden observar más de **3000 anotaciones** a nivel de símbolo, donde por ejemplo nos podemos encontrar blancas, negras, corcheas, sostenidos, bemoles o incluso la armadura del pentagrama, del cual se puede extraer información como el compás, figuras de cada compás o la voz.

Cabe destacar, que sobre dicha cantidad de información, se aplicaron técnicas de data augmentation para obtener mayor cantidad de patrones, como puede ser escalar o recortar imágenes.

- **Preprocesamiento**. Una vez que tenemos los datos, es necesario procesarlos antes de usarlos como entrenamiento en la red neuronal convolucional profunda. Para realizar dicho procesamiento, este se dividió en dos etapas:

    - **Primera etapa.** Para mejorar la eficiencia del entrenamiento de la red, se ajustaron todas las imágenes de entrada al mismo tamaño y formato, es decir, se cogieron todas las imágenes de entrada a color y se redimensionaron a imágenes de 1280x720 píxeles. Esto permite así a la red ajustarse a dichos parámetros de entrada (a la hora de pasarle una imagen de test debe estar en la misma resolución). Ya que sería muy complejo para la red trabajar con distintas resoluciones.

    - **Segunda etapa.** Para que pueda entrenar la red, es necesario especificarle las figuras que debe reconocer y donde se encuentran. Para ello algún ser humano debe clasificar dichos patrones.
      Para poder lograrlo se usó VGG Image Annotator(VIA) [8]. VGG Image Annotator es un software de anotación manual independiente y simple que se utiliza para imagen, audio y video, además se utiliza sobre el navegador web de tal forma que no se requiere ninguna configuración o incluso instalación. Este software nos permite entonces marcar el lugar de cada nota a mano y especificar incluso el símbolo al que pertenece (blanca, negra, corchea, etc). Entonces aparte de las imágenes de entrada, estas tendrán también como acompañamiento un fichero con formato JSON, que contiene la ubicación de cada nota de la imagen y una etiqueta que especificará el tipo de figura, esta información se representará con un polígono sobre la imagen que marca la ubicación de la nota y tiene asociado la etiqueta de la misma.

      En la siguiente figura se puede apreciar una imagen de entrada ya procesada de una manera más clara.

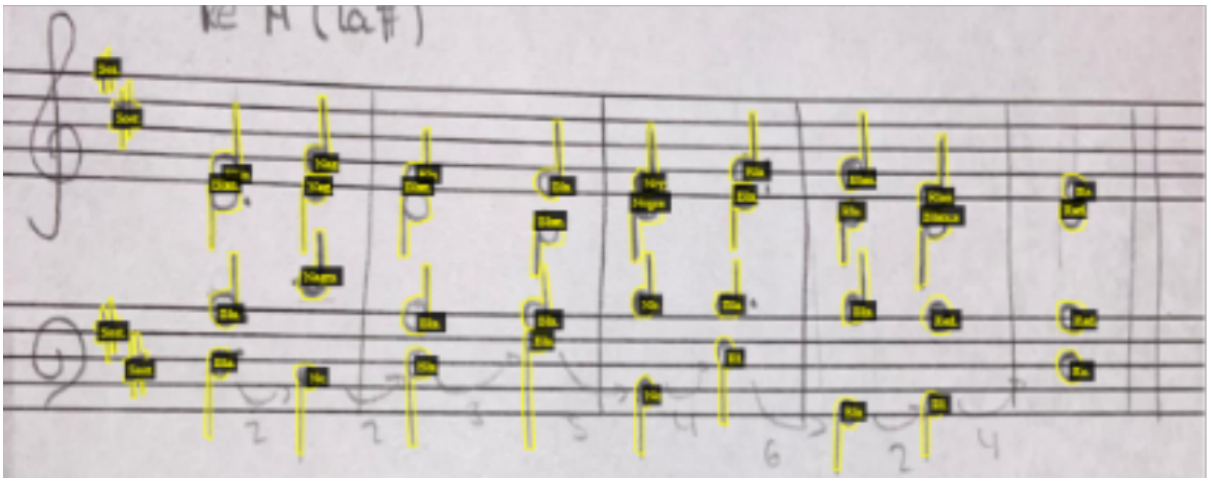

Figura 6: Ejemplo del preprocesamiento del conjunto de datos, para cada partitura SATB

- **Aplicación de la CNN**. Para la realización de este trabajo, como se comentó anteriormente, se aplicó una arquitectura Mask R-CNN, dicha arquitectura fue presentada en el 2017, siendo una extensión de la anteriormente nombrada arquitectura Faster R-CNN. La diferencia es que para la Mask R-CNN se añade una rama para predecir máscaras segmentadas para cada región de interés, de tal forma que es paralela a las tareas de identificación y localización. Lo que quiere decir es que, en vez de dar como resultado la etiqueta de clasificación con los cuadros delimitados como hacía la arquitectura Faster R-CNN, se añaden máscaras binarias aplicando RoI Align (RoI, Region of Interest), para cada región de interés. A continuación se explica el por qué de esta adicción y qué ventajas nos ofrece.

  La red Mask R-CNN está formada por tres etapas, en la siguiente figura se pueden apreciar la estructura completa de dicha red.

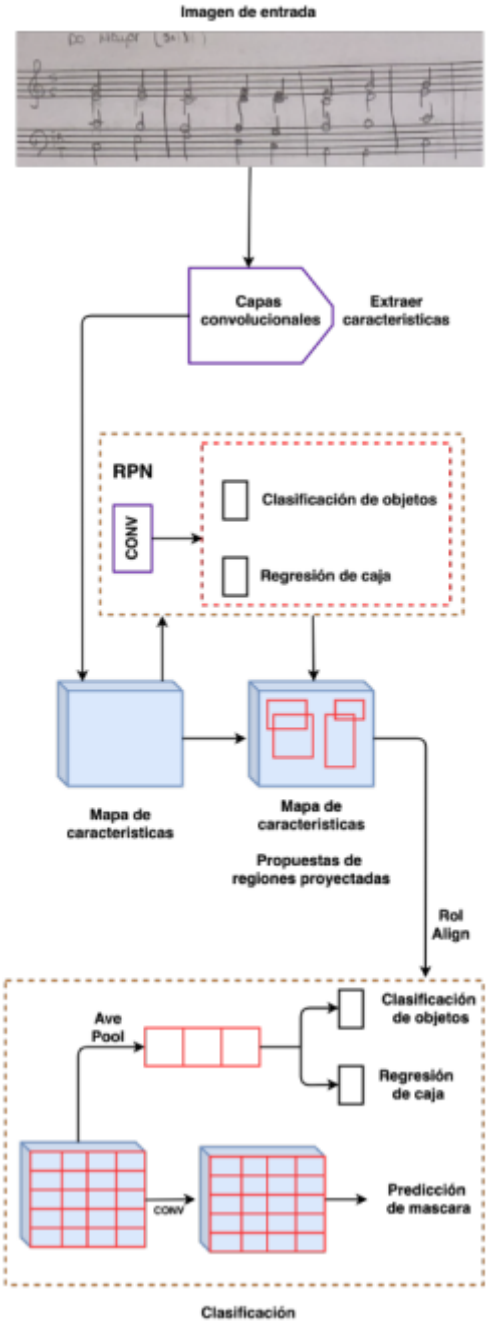

Figura 7: Arquitectura de la red Mask R-CNN

- **Primera etapa.** Mapa de características para la imagen de entrada.
- **Segunda etapa.** Las salidas de la primera etapa se utilizan para la red de propuesta de región (RPN) para generar regiones de interés (ROI)

- **Tercera etapa.** Aprovechando las regiones de interés generadas por el RPN, se mapean y se extraen las características necesarias para realizar las clasificaciones de los objetos, las máscaras de segmentación y los cuadros delimitadores.

De esta forma se consigue el reconocimiento de la partitura SATB, concretamente de aquellos objetos musicales sencillos con los que se ha entrenado, ya que no se busca ser capaz de detectar toda la simbología, sino proponer un nuevo método de reconocimiento de notas. Las notas que puede reconocer esta red son las vistas en la siguiente figura.

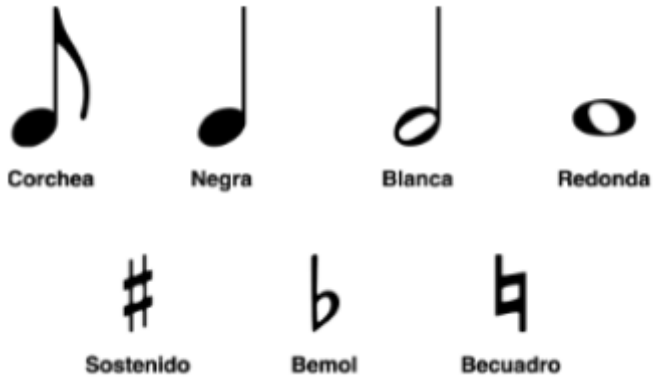

Figura 8: Notas clasificables en las partituras SATB

- **Porcentaje de recuperación**. Para que la red aprenda es necesario ir ajustando el error de dicha red. En este caso la pérdida obtenida tras el entrenamiento consta de tres partes:

  - La perdida de clasificación (Lcls).

$$L_{cls}(p_i^*\ ,\ p_i) = -\ log(p_i^*\ p_i)$$

  - Localización (Lbox).

$$L_{box}(t_i\ ,\ t_i^*) = L_1^{smooth}(t_i^* - t_i)$$

$$L_1^{smooth}\ (x)\ =\ \begin{cases} 0{,}5x^2 & if\ \ |x| < 1 \\ |x| - 0{,}5 & en\ \ otro\ \ caso \end{cases}$$

- Máscara de segmentación (L$mask$).

$$L_{mask}(s_i\ ,\ s_i^*) = -(s_i^* \log s_i + (1 - s_i^*) \log(1 - s_i))$$

Para la pérdida total del entrenamiento, se puede calcular mediante la siguiente fórmula:

$$L(p_i,\ p_i^*,\ t_i,\ t_i^*,\ s_i,\ s_i^*) = \frac{1}{N_{cls}} \sum_i L_{cls}\ (p_i\ ,\ p_i^*)$$
$$+ \frac{\lambda}{N_{box}} \sum_i\ p_i^*\ L_{box}\ (t_i\ ,\ t_i^*)\ +\ \frac{\gamma}{N_{mask}} \sum_i L_{mask}\ (s_i\ ,\ s_i^*)$$

Si nos fijamos en esta última fórmula, calcular la pérdida total consta del sumatorio de los tres errores anteriores donde N representa el número de cuadrados delimitadores correspondientes y λ y γ, equilibran las pérdidas de entrenamiento de la regresión y de la rama de máscara.

Para mayor entendimiento se explican las siguientes variables:

- pi -> representa la probabilidad predicha de que el cuadro delimitador i sea un objeto.
- p*i -> representa la probabilidad de verdad básica (binaria) de si el cuadro delimitador i es un objeto.
- ti -> representa cuatro coordenadas parametrizadas, que son: el valor de las coordenadas horizontales y verticales del punto central en el cuadro, la anchura y la altura del cuadro.
- t*i -> indica la diferencia entre el cuadro de la etiqueta verdadera y el cuadro delimitador positivo.
- s -> representan respectivamente las matrices binarias de la máscara de predicción y de la etiqueta verdadera.

Una vez especificado el cálculo del error, solo falta definir qué métricas se van a utilizar para calcular el rendimiento de la red. En este caso, se utilizaron las métricas de **precisión (P)** y **recuperación (R)**. Las fórmulas de la mismas son las siguientes:

$$P = \frac{TP}{TP + FP} \qquad R = \frac{TP}{TP + FN}$$

Donde **PT es** el número de casos positivos que correctamente se detectaron como positivos, **FP** es el número de casos que se dieron como negativos cuando en realidad eran positivo y **FN**, es el número de casos que son positivos y se dieron como negativos.

Aparte de dichas fórmulas, se puede comprobar la eficiencia de la red de forma visual, de tal forma que en la imagen resultante, encontraremos para cada nota encontrada, su cuadro que define su posición, la etiqueta que la clasifica con el porcentaje de acierto, y el contorno de dicha nota. En la siguiente imagen se puede observar de forma más clara un ejemplo de salida, donde no se detecta una nota.

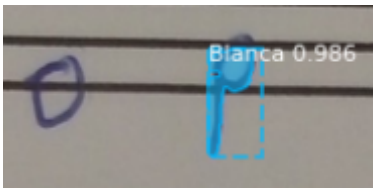

Figura 9: Ejemplo de clasificación con imagen nota detectada (derecha) y nota no detectada (izquierda) en partitura SATB

Una vez explicada la metodología a seguir y la estructura de la red, se comenta la implementación utilizada y los resultados obtenidos.

Respecto a la implementación, se utilizó como lenguaje de programación Python, apoyándose en TenzoFlow y Keras para la creación de la red. En términos de cómputo, se utilizó una doble GPU, siendo una NVIDIA GTX 1080 Ti y una NVIDIA GTX 1080. Respecto a la memoria, se utilizó una RAM de 64GB y un disco duro de 8TB.

A la hora de llevar a cabo todo el proceso explicado anteriormente, se utilizaron 100 imágenes de partituras SABT para el entrenamiento donde el 80% del conjunto se utilizó para entrenamiento y el 20% restante para validación del modelo, para comprobar la fiabilidad del modelo. Cabe destacar que se utilizaron las mismas imágenes para entrenar que para verificar la estabilidad y confiabilidad (evaluación) del modelo entrenado.

Centrándonos más en la red, se configuró la arquitectura con una tasa de aprendizaje de 0.001, ajustándose a 700 épocas de entrenamiento con un factor de ajuste 0.9% y 100 pasos para cada época, evaluando 1 imagen por cada GPU.

Una vez configurada la red, ya se empezaron a generar resultados una vez lanzada, de tal forma que para cada entrada se generaba la puntuación de categorías, los delimitadores y las máscaras individuales para cada símbolo. La ejecución de todo el entrenamiento duró 11 horas hasta converger, de tal forma que tardaba aproximadamente un minuto por época.

Respecto a los resultados obtenidos, para las 100 imágenes de prueba, la precisión general era de un 95.4%, y para las tasas de recuperación, del 94.5%. Dando unos resultados bastante razonables.

Cabe destacar que los resultados no van a ser perfectos por problemas de etiquetado, ya que algunas notas, incluso para el ojo humano, son difíciles de etiquetar correctamente. Además, hay diferencia de aparición entre distintos tipos de figuras musicales. Por ejemplo, la blancas tienen un 25% de aparición, las redondas un 40% y las negras un 20%, mientras que las corcheas, sostenidos, bemoles y becuadros tienen solo un 2%, 6%, 5% y 2% de aparición respectivamente. Por lo tanto, en este problema, hay un pequeño caso de imbalance de clases.

En la siguiente imagen, se muestra una salida de dicha red, donde se pueden apreciar, la etiqueta de cada nota con su probabilidad, la caja que la define y la silueta de la misma.

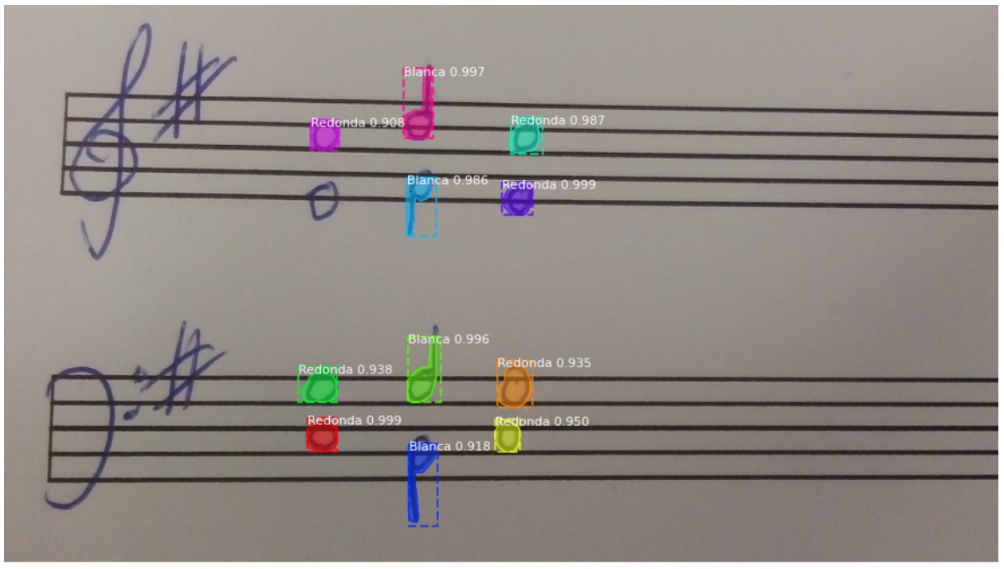

Figura 10: Ejemplo de salida (predicción)

5. Conclusión y posibles mejoras:

Teniendo en cuenta toda la información que hemos visto, podemos decir que es un trabajo muy interesante con un objetivo muy enfocado pero con muchas posibilidades de extrapolarlo. Pudiendo pasar de una demostración de una herramienta de apoyo sencilla a sharpmony, a otros problemas generales de reconocimiento de partituras.

Cabe destacar que personalmente ha sido muy interesante la parte donde se explica la red mask R-CNN y porque el uso de esta, de tal forma, que se ha podido aprender con este trabajo, la evolución de la detección de objetos. Pasando de las redes CNN sencillas por redes R-CNN o faster R-CNN hasta la red elegida para este problema y los beneficios que nos ofrece.

Todo este tema de la música, es un tema muy interesante y con proyección de futuro, permitiendo la creación de nuevas herramientas, ya sean para uso educativo u otros. Cabe destacar que se podrían realizar muchísimas mejoras sobre este proyecto. Como puede ser la capacidad de detectar mayor cantidad de figuras musicales, ajustar la red a un tipo concreto de instrumento, como puede ser el piano (se usa el pedal por ejemplo), introducción de nuevos concepto como pentagramas con más de cinco líneas o incluso aun más complejo, detectar la nota especificada, esto es algo más complejo ya que se debe analizar la armadura con la respectiva clave que se esté utilizando.

Aunque como idea es muy buena y tiene mucha proyección, si nos fijamos en el proyecto realizado como tal, sí que hay cierto margen de mejora. Cabe destacar, que al tener una cantidad de datos tan reducida, el modelo no puede llegar a entrenar correctamente. Esto se puede apreciar sobre todo, con el imbalance de patrones existente y lógico, que se puede encontrar en las imágenes de entrada, de tal forma que se ve perjudicada la clase minoritaria. De tal forma, que habría que solucionar este problema ya sea aplicando muestras artificiales, modificar el Dataset, ajustar los parámetros del modelo, etc.

En conclusión, es un trabajo muy interesante con cierto margen de mejora y con muchísima proyección, ya sea centrado en la idea que se plantea en este trabajo, o más a nivel general en reconocimiento de partituras.

Bibliografía:

[1] J. Redmon, S. Divvala, R. Girshick, y A. Farhadi, You only look once: Unified, real-time object detection, en Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 779-788.
[2] W. Liu et al., Ssd: Single shot multibox detector, en European conference on computer vision, 2016, pp. 21-37.
[3] T.-Y. Lin, P. Goyal, R. Girshick, K. He, y P. Dollar, ´ Focal loss for dense object detection, en Proceedings of the IEEE international conference on computer vision, 2017, pp. 2980-2988.
[4] S. Ren, K. He, R. Girshick, y J. Sun, Faster r-cnn: Towards realtime object detection with region proposal networks, arXiv preprint arXiv:1506.01497, 2015.
[5] R. Girshick, J. Donahue, T. Darrell, y J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, en Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 580-587.
[6] J. Dai, Y. Li, K. He, y J. Sun, R-fcn: Object detection via region-based fully convolutional networks, arXiv preprint arXiv:1605.06409, 2016.
[7] Odemakinde, E. (2022, 11 julio). Everything about Mask R-CNN: A Beginner's Guide. viso.ai. https://viso.ai/deep-learning/mask-r-cnn
[8] Visual Geometry Group - University of Oxford. (s. f.). https://www.robots.ox.ac.uk/%7Evgg/software/via/