# Aplicación de técnicas de aprendizaje profundo al reconocimiento óptico de partituras SATB

Martin Morita-Hernández*, Francisco Fernandez de Vega†
Departamento de Tecnología de los Computadores
y de las Comunicaciones
Universidad de Extremadura
Mérida, España
Email: *martin.morita.13@gmail.com †fcofdez@unex.es

Juan Villegas Cortez
Departamento de Sistemas
Universidad Autónoma Metropolitana, Unidad Azcapotzalco
Cd. de México, México
Email: juanvc@azc.uam.mx

***Resumen*—El reconocimiento óptico de partituras es un problema que forma parte del ámbito OMR (Optical Music Recognition). La gran variedad de elementos presentes en una partitura, que incluye clave, compás, tempo, dinámica, articulación, repeticiones, además de las notas, duración y alteraciones, hacen de su reconocimiento óptico un problema interesante en el dominio de la inteligencia artificial. Este artículo aborda este difícil problema utilizando técnicas de aprendizaje profundo y contextualizando en el aprendizaje de estudiantes de armonía, que realizan sus ejercicios de forma manuscrita sobre papel. Los resultados preliminares obtenidos, con unos porcentajes de acierto de alrededor del 95 % en la clasificación correcta de los elementos de la partitura en la fase de entrenamiento, nos hacen vislumbrar la posibilidad de crear una herramienta útil tanto para profesores como para alumnos de armonía.**

***Index Terms*—Aprendizaje profundo, OMR, Reconocimiento óptico.**

## I. Introducción: OMR

El reconocimiento óptico de información musical (OMR) es un área que goza de bastante interés entre las aplicaciones de la inteligencia artificial a la música.

La Sociedad Internacional para la Recuperación de Información Musical (ISMIR, International Society for Music Information Retrieval) lo considera como una de las áreas relevantes, y difíciles de abordar. Frente al reconocimiento óptico de caracteres, en el que cada carácter tiene un valor independientemente de su posición dentro de un documento, (e.g., la letra "a" siempre representa la misma información en un texto), las notas musicales poseen un valor que depende no solo de la grafía de la misma, sino de su posición sobre el pentagrama. Pero además de esto, una partitura posee mucha información adicional que se refieren al modo en que la nota musical debe emitirse con el instrumento, incluyendo dinámica (*fuerte, piano,...*), articulación (*ligado, stacatto,...*), repeticiones (*saltos, codas, da capo...*), claves (*Sol, Fa, Do...*), compás (2 *por* 4, 4 *por* 4, 6 *por* 8, ...), etc.

Así, y aunque en principio las técnicas disponibles para el reconocimiento óptico de caracteres puedan ser aplicables en el nuevo marco, la realidad es que se necesitan técnicas adicionales para poder obtener toda la información relevante, si el objetivo es convertir una partitura en papel a un formato de partitura estándar, tal como MIDI[1] o MusicXML[2].

Este artículo explora técnicas disponibles para realizar el reconocimiento de partituras manuscritas en el contexto de los ejercicios de armonía a cuatro voces, cuyas reglas son aprendidas por los estudiantes de tercer y cuarto curso de los conservatorios profesionales de música en España. Tal como describimos más adelante, el objetivo es desarrollar un módulo para que la aplicación y herramienta Sharpmony [3] permita a profesores y estudiantes capturar la información de cualquier ejercicio de armonía escrito a mano sobre una libreta pautada.

En la sección II describimos el tipo de ejercicio manuscrito al que nos enfrentamos, y revisamos las propuestas disponibles en OMR que puedan servir como punto de partida al trabajo que hemos desarrollado. En la sección 3 presentamos en detalle la metodología aplicada, y su implementación con los resultados los discutimos en la sección 8. La sección V describe las mejoras previstas y su aplicación a la herramienta *Sharpmony*. Finalmente en la sección VI compartimos las conclusiones y el trabajo futuro.

## II. Armonía SATB

En música clásica, el estudio de armonía es uno de los elementos claves para que los alumnos desarrollen sus habilidades en composición. La música occidental ha desarrollado y evolucionado una serie de reglas armónicas que son trabajadas por todos los estudiantes matriculados en los conservatorios profesionales de música que estudian la música occidental. Estas reglas, que comenzaron a desarrollarse más o menos

[1]Acrónimo de Musical Instrument Digital Interface(Interfaz digital para instrumentos musicales). Protocolo para la transmisión de datos musicales entre componentes digitales, como los sintetizadores y las tarjetas de sonido de las computadoras. Britannica Escolar, s.v. "MIDI", consultado el 6 de mayo, 2021, https://bidi.uam.mx:8429/levels/academica/article/MIDI/421779.

[2]MusicXML es un formato abierto de notación musical basado en XML. Consultado el 6 de mayo, 2021, https://www.musicxml.com/

[3]*Sharpmony* es la primera herramienta basada en Inteligencia Artificial para ayudar a los estudiantes de Armonía y sus profesores. Utilizando las últimas tecnologías de IA desarrolladas en la Universidad de Extremadura, integra una aplicación móvil y un portal web que permitirá al estudiante revisar ejercicios de armonía clásica (SATB), y realizar armonización automática de un tiple o bajo cifrado. Consultado el 6 de mayo, 2021, https://sharpmony.unex.es

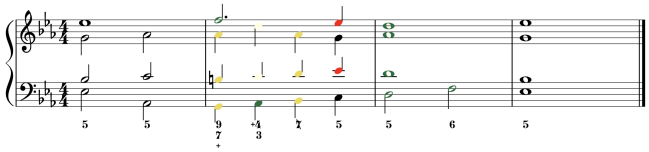

Figura 1. Ejercicio corregido.

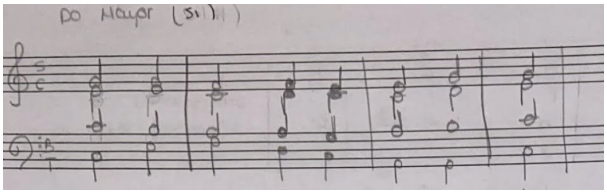


Figura 2. Ejercicio manuscrito de armonía a cuatro voces.

formalmente en el renacimiento, se establecen de forma definitiva en el Barroco, con el estudio del contrapunto y la fuga, siendo Bach la figura de referencia del periodo. Además de música instrumental, Bach desarrolla toda una serie de composiciones corales, en las que la superposición de voces requiere de habilidad y maestría en el desarrollo armónico.

Así, en el periodo que le sigue, el clasicismo, se afianza el estudio de la armonía coral a cuatro voces, integrada por cuatro voces: Soprano, Contralto, Tenor y Bajo (SATB); y surgen nuevas reglas que marcan la novedad de la música del periodo frente a lo que se considera antiguo, por pertenecer a estilos pasados, como el Barroco o Renacimiento.

Es de este periodo, el clasicismo, del que se toma el conjunto de reglas que actualmente se estudia en las asignaturas Armonía 1 y 2, aplicadas a corales SATB en los curso tercero y cuarto de las enseñanzas profesionales de música, y que continúan perfeccionando posteriormente los estudiantes con el estudio del contrapunto y otras técnicas de composición más modernas en cursos posteriores del grado profesional, así como en el grado superior de música.

El método de estudio consiste por tanto en tomar como punto de partida una melodía que se asigna a la Soprano, o un bajo cifrado, y el estudiante debe componer el resto de las voces siguiendo unas reglas que indican lo que está permitido hacer y lo que no. Las reglas afectan tanto a los intervalos entre voces, como a los movimientos melódicos que cada voz desarrolla en combinación con el resto de voces.

La Figura 1 muestra un ejemplo de ejercicio de armonía coral desarrollado por un estudiante, con algunas notas marcadas con colores, correspondiente a notas que no cumplen las reglas de armonía clásica. Concretamente aparecen marcados errores correspondientes a *Distancias mayor que octava entre voces (amarillo claro) Cifrado Incorrecto (amarillo mostaza), Quintas-Octavas paralelas (rojo) y acordes incorrectos (verde oliva)*.

### *II-A. Inteligencia artificial y armonía SATB*

Son varios los problemas asociados a la armonía SATB en los que pueden aplicarse diferentes técnicas de Inteligencia Artificial. Podemos destacar:

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

Como ejemplo del primer punto, lo que mostramos en la figura 1 corresponde a una corrección realizada de forma automática con la herramienta *Sharpmony*, herramienta que incorporando Inteligencia Artificial está ya siendo usada por varias instituciones para ayuda a la docencia de armonía, y que esperamos pueda incorporar en breve las metodologías que presentamos a continuación.

En este trabajo nos vamos a ocupar del tercero de los puntos detallados anteriormente, y que corresponde con el reconocimiento automático de partituras con ejercicios SATB. En la figura 2 mostramos un ejercicio típico desarrollado por un estudiante en su libreta pautada. Como puede observarse el ejercicio contiene indicaciones del compás, la tonalidad (en este caso al ser Do Mayor el ejercicio no contiene armadura) y las diferentes figuras de cada compás y voz, incluyendo redondas, blancas y negras.

### *II-B. Reconocimiento automático de partituras*

El problema OMR es bien conocido en la comunidad de Recuperación Automática de Información Musical (Music Information Retrieval, MIR). Aunque existen dos versiones del mismo, el que corresponde a partituras impresas y aquel dedicado a partituras manuscritas, el más interesante y a la vez difícil es el segundo caso, que corresponde con lo que mostramos anteriormente en la figura 2.

Las recientes investigaciones sobre redes neuronales convolucionales (CNN, Convolutional Neuronal Networks) por medio del aprendizaje profundo (Deep Learning), han dado pie a que se propongan varios enfoques OMR. Durante este período, han surgido diversos algoritmos de detección de objetos (Object Detection) que compiten entre sí, como YOLO [1], SSD [2] y RetinaNet [3], siendo clasificados como modelos de detección de una etapa, por lo que tienen una velocidad de detección más rápida. También existen modelos de detección de dos etapas, como Faster R-CNN [4], R-CNN [5], y R-FCN [6], tienen niveles de precisión de detección más altos pero velocidades más lentas. Mientras que algunos optimizan la precisión, otros se esfuerzan por lograr un alto rendimiento.

Se ha demostrado que los procedimientos tradicionales de segmentación y clasificación de símbolos funciona bien en partituras musicales impresas [7]. Pero cuando se consideran imágenes de partituras escritas a mano, estos sistemas tienden a fallar, porque los errores se propagan a lo largo del entrenamiento, por ejemplo, un error de segmentación podría causar objetos detectados incorrectamente. Algunos investigadores han hecho intentos para superar esta limitación haciendo uso de CNN, por ejemplo, en [8] Pacha y sus colegas para detectar notas utilizan tres conjuntos de datos distintos

DeepScores [9], MUSCIMA++ [12] y Capitan collection [10], además de sugerir una línea base para la detección general de objetos dentro de la partitura, haciendo uso de tres modelos distintos de detección de objetos Faster R-CNN, U-Net [11] y RetinaNet. Los resultados de las pruebas de cada modelo se dieron en términos de precisión promedio.

Además Hajič y sus colegas proponen la detección directa de objetos musicales con CNN [13], quienes sugieren una adaptación de Faster R-CNN con un mecanismo de propuesta de región personalizada basado en el esqueleto morfológico para detectar las cabezas de las notas, no requiere la eliminación del pentagrama y es aplicable a una variedad de estilos de escritura y niveles de complejidad musical. En investigaciones posteriores sugieren identificar las notas en dos etapas, haciendo uso de la arquitectura U-Net [14]. Durante la primera etapa, la imagen de la puntuación de entrada es segmentada como imagen binaria utilizando el modelo de segmentación semántica, y el problema de detección general se descompuso en un conjunto de problemas de clasificación de píxeles binarios, y luego se utilizó el detector de componentes conectados para derivar la propuesta de detección final. El experimento tuvo como objetivo el conjunto de datos MUSCIMA++, que muestra los resultados de detección de símbolos en términos de puntuaciones.

Tuggener y sus colegas hicieron uso de ResNet [15] para predecir mapas de energía con el fin de obtener la ubicación, la clase y el cuadro delimitador de cada símbolo contenido dentro de una partitura, sin necesidad de recortar por secciones las imágenes de pentagramas [16]. Este método es efectivo detectando símbolos pequeños, pero genera cuadros delimitadores inexactos y clases no contenidas en el sistema.

Observamos algunos inconvenientes en los métodos anteriores. Después de los enfoques basados en la detección de objetos, se requieren pasos adicionales para detectar los símbolos y obtener el tono de la nota.

En este artículo, proponemos el uso de un nuevo clasificador de redes neuronales convolucionales (CNN) denominado Mask R-CNN [17], que tiene el potencial de lograr una mejor precisión de reconocimiento en las partituras escritas a mano, simplificando el proceso y prediciendo una máscara binaria para cada clase de forma independiente.

## III. Metodología

A continuación presentamos la metodología propuesta para reconocimiento de partituras escritas a mano, tal como se aprecia en la Figura 3, esta consta de cuatro etapas generales.

En primer lugar hemos procedido a crear nuestro repositorio formado por fotografías de las partituras escritas a mano, a las que se aplica el proceso de segmentación detallado más abajo.

A continuación estas partituras ya segmentadas han sido utilizadas para entrenar una red neuronal convolucional profunda, pre-entrenada, y finalmente se obtiene una clasificación de los símbolos varios contenidos en cada imagen del repositorio. Este procedimiento se realiza en una primera etapa de entrenamiento para nuestro problema, posteriormente se realiza la etapa de prueba sobre imágenes desconocidas para la red.

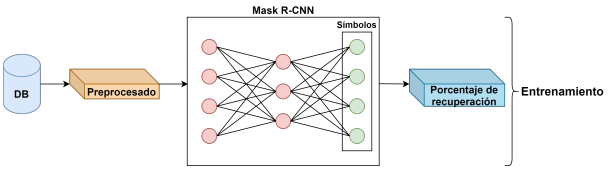


Figura 3. Metodología.

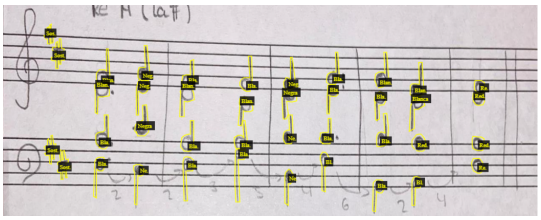

Figura 4. Ejemplo de preprocesamiento del conjunto de datos, para cada partitura SATB.

En las siguientes líneas damos detalles de las etapas de la metodología para facilitar la comprensión de nuestro trabajo.

### *III-A. Conjunto de datos (DB)*

Para entrenar el detector de notas musicales, se elaboró nuestro propio conjunto de datos utilizando algunas técnicas de *data augmentation*, como los son recortar y escalar, aplicadas a imágenes de partituras SATB suministrada por los profesores de armonía que colaboran en el proyecto, obteniendo 100 imágenes que contienen más de 3000 anotaciones a nivel de símbolo, que incluyen redondas, blancas y negras, así como la armadura de la partitura, que sirve para detección de la tonalidad, indicaciones del compás, y las diferentes figuras de cada compás y voz, realizadas por estudiantes de conservatorios en sus libretas pautadas.

### *III-B. Preprocesamiento*

A partir del conjunto de datos creado con las imágenes de partituras obtenidas de los alumnos, fueron adaptadas de modo que todas tuvieran el mismo tamaño y formato. El proceso se dividió en dos partes. Primero, para entrenar eficientemente el detector de objetos la implementación evaluada coge imágenes en color como entrada, las cuales deben tener el mismo tamaño, por tal motivo han sido redimensionadas en $1280 \times 720$ píxeles. Segundo, se utilizó VGG Image Annotator (VIA), para identificar dentro de la imagen de la partitura las ubicaciones de cada nota, así como el símbolo al que pertenecen. Con toda la información generada, disponemos de imágenes de partituras acompañadas por un archivo en formato JSON, que contiene la ubicación de cada una de las notas dentro de la imagen y sus respectivas etiquetas, que se representan con un polígono delimitador adaptado a la forma de cada símbolo, como se muestran en la Figura 4.

### *III-C. Aplicación de la CNN*

Para el experimento, se evaluó la arquitectura Mask R-CNN, presentada en 2017, es una extensión de Faster R-CNN

que añade una rama para predecir máscaras segmentadas en cada región de interés, siendo esta paralela a las tareas de identificación y localización. Esto es, si Faster R-CNN tenía como salida las clases y los cuadros delimitados, Mask R-CNN añade a estos una máscara binaria haciendo uso de una operación *RoI Align* (RoI, Region of Interest) para cada región de interés.

Mask R-CNN se divide en tres etapas, tal como se muestra en la Figura 5. Primero, una red se encarga de generar los mapas de características de las imágenes de entrada. En segundo lugar, los mapas de características generados son enviados a la red de propuesta de región (RPN) para generar regiones de interés (RoI). En tercer lugar, las RoI generadas por RPN se mapean para extraer las características correspondientes para la clasificación de objetos y la segmentación de instancias. Este proceso genera las puntuaciones de clasificación, los cuadros delimitadores y las máscaras de segmentación.

El objetivo de nuestro modelo no es detectar todos los símbolos en la partitura, sino proponer un nuevo método de reconocimiento de objetos musicales, centrándose en el reconocimiento de notas, utilizando los símbolos más comunes dentro de una partitura SATB. Aquí, no necesitamos detectar simbología compleja. Los símbolos representativos se muestran en la Figura 6.

### *III-D. Porcentaje de recuperación*

La pérdida de entrenamiento de Mask R-CNN consta de tres partes principales: la pérdida de clasificación ($L_{cls}$) ecuación 1, localización ($L_{box}$) ecuaciones 2, 3, y máscara de segmentación ($L_{mask}$) ecuación 4. La pérdida total de entrenamiento se puede calcular mediante la ecuación 5.

$$L_{cls}(p_i^*\ ,\ p_i) = -\ log(p_i^*\ p_i) \tag{1}$$

$$L_{box}(t_i\ ,\ t_i^*) = L_1^{smooth}(t_i^* - t_i) \tag{2}$$

$$L_1^{smooth}\ (x)\ =\ \begin{cases} 0{,}5x^2 & if\ \ |x| < 1 \\ |x| - 0{,}5 & en\quad otro\quad caso \end{cases} \tag{3}$$

$$L_{mask}(s_i\ ,\ s_i^*) = -(s_i^* \log s_i + (1 - s_i^*) \log(1 - s_i)) \tag{4}$$

$$L(p_i,\ p_i^*,\ t_i,\ t_i^*,\ s_i,\ s_i^*)\ =\ \frac{1}{N_{cls}} \sum_i L_{cls}\ (p_i\ ,\ p_i^*) + \frac{\lambda}{N_{box}} \sum_i\ p_i^*\ L_{box}\ (t_i\ ,\ t_i^*)\ +\ \frac{\gamma}{N_{mask}} \sum_i L_{mask}\ (s_i\ ,\ s_i^*) \tag{5}$$

En la ecuación 5, $N_*$ representa el número de cuadros delimitadores correspondientes. Los parámetros $\lambda$ y $\gamma$ equilibran las pérdidas de entrenamiento de la regresión y la rama de máscara.

Además, $p_i$ representa la probabilidad predicha de que el cuadro delimitador $i$ sea un objeto, y $p_i^*$ representa la probabilidad de verdad básica (binaria) de si el cuadro delimitador

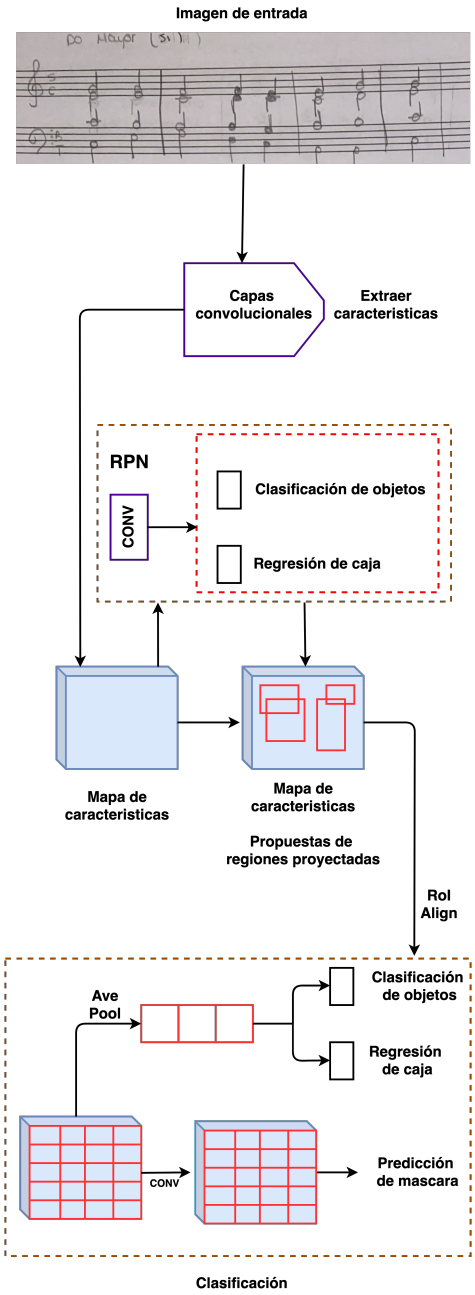


Figura 5. Arquitectura de Mask R-CNN.

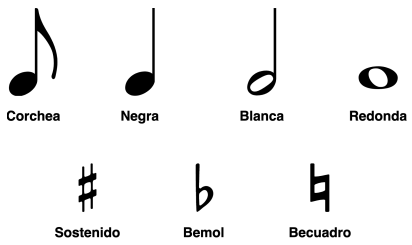


Figura 6. Símbolos utilizados en una partitura SATB.

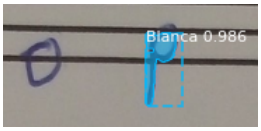


Figura 7. Símbolo correctamente identificado (derecha) y no identificado (izquierda).

$i$ es un objeto; La variable $t_i$ 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, y $t_i^*$ indica la diferencia entre el cuadro de la etiqueta verdadera y el cuadro delimitador positivo; Y $s$ representan respectivamente las matrices binarias de la máscara de predicción y de la etiqueta verdadera.

Para este trabajo de investigación, se aplicaron las tasas de precisión $(P)$ y recuperación $(R)$, se pueden calcular mediante las ecuaciones 6 y 7 respectivamente, para evaluar el rendimiento de Mask R-CNN en la detección de los símbolos, donde $TP$ es el número de casos que son positivos y se detectan como positivos, $FP$ es el número de casos que son negativos pero que se detectan como positivos y $FN$ es el número de casos que son positivos pero que se detectan como negativos [18].

$$P = \frac{TP}{TP + FP} \tag{6}$$

$$R = \frac{TP}{TP + FN} \tag{7}$$

De manera visual también es posible detectar los símbolos que han sido correctamente identificados, y cuales no han sido reconocidos por la CNN. Aquellos símbolos que son detectados de manera correcta están delimitados por un cuadro y una mascara que se adapta a su forma, también contienen la etiqueta de la categoría a la que pertenecen y el porcentaje de acierto con el que se ha evaluado, en la Figura 7 podemos observar estas características, también se observa un símbolo que no ha sido detectado.

## IV. Implementación y resultados

Nuestra metodología la implementamos bajo el marco de desarrollo de aprendizaje profundo de TensorFlow[4] y Keras[5], programado en lenguaje Python, con una doble GPU, las cuales son NVIDIA GTX 1080 Ti y NVIDIA GTX 1080, memoria RAM de 64GB y 8TB de disco duro.

Para la investigación, se seleccionaron 100 imágenes de partituras para el entrenamiento (80 % del conjunto de entrenamiento y 20 % del conjunto de validación). Con el fin de verificar la estabilidad y confiabilidad del modelo entrenado, se utilizaron las mismas imágenes del conjunto de entrenamiento

[4]TensorFlow es una plataforma de código abierto para construir y entrenar redes neuronales, que permiten detectar y descifrar patrones análogos al aprendizaje utilizado por los humanos, https://www.tensorflow.org/?hl=es-419.

[5]Keras es una biblioteca de código abierto, tiene como objetivo de acelerar la creación de redes neuronales. Keras no funciona como un framework independiente, sino como una interfaz de uso intuitivo (API), https://keras.io/.

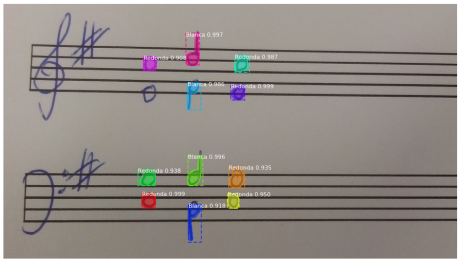

Figura 8. Predicciones de los símbolos.

para la evaluación del modelo. En este caso la arquitectura se configuro con una tasa de aprendizaje de 0,001, y se ajustó a 700 épocas de entrenamiento con un factor de ajuste de 0,90 % y 100 pasos por cada una de ellas, evaluando 1 imagen por GPU. El modelo generó las puntuaciones de las categorías, los cuadros delimitadores y las máscaras individuales de los símbolos para cada imagen de entrada. El tiempo de entrenamiento para 700 épocas fue de 11 horas, aproximadamente 1 minuto por época, y la función de pérdida del modelo alcanzó un estado de convergencia.

Los resultados de 100 imágenes de prueba mostraron que la precisión general y las tasas de recuperación fueron 95,40 % y 94,50 %, respectivamente. En la Figura 8 se presenta una detección de símbolos clasificados, junto con el porcentaje individual de precisión.

Es notable que la detección de algunos símbolos funciona mejor que otros. Las principales razones de los errores en el reconocimiento de los símbolos se deben a que el tamaño de las muestras de algunos símbolos, como son corcheas, sostenidos, bemoles y becuadros, las cuales representan el 2 %, 6 %, 5 % y 2 %, respectivamente, no son suficientes para producir resultados confiables, en comparación con blancas, redondas y negras, que tienen un promedio del 25 %, 40 % y 20 %, respectivamente. Además, varios símbolos se juzgaron mal entre sí, ya que incluso para el ojo humano eran indistinguibles, debido a la manera en que escriben los alumnos, y los resultados de la clasificación del modelo se ven afectados por errores humanos al etiquetar el conjunto de entrenamiento.

## V. Discusión y trabajo futuro

Los resultados obtenidos hasta el momento nos hacen confiar en obtener una metodología que pueda ponerse a disposición de los usuarios interesados. Primero, la herramienta *Sharpmony*, actualmente disponible en Google Play Store para descarga, está siendo utilizada por más de 1800 usuarios registrados. Adicionalmente, esperamos poder incluir en *Sharpmony* la opción de captura fotográfica descrita anteriormente. No obstante, a la metodología anterior habrá que añadir la etapa que permita capturar no solo las figuras (redondas, blancas...), sino también la posición concreta sobre el pentagrama que nos permita identificar el nombre de la nota (*do, re, mi, ...*).

En pruebas preliminares estamos añadiendo a la metodología una segunda CNN, con el objetivo de identificar el tono

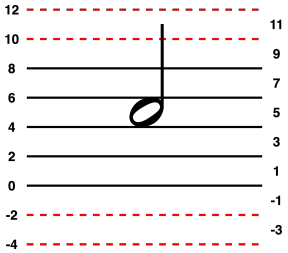


Figura 9. Clasificación de posiciones dentro en el pentagrama.

de una nota, ya que es información vital en el reconocimiento de partituras. Utilizamos la posición para representar el tono, asignando como punto de partida la quinta linea del pentagrama, identificando con números pares las lineas y con números impares los espacios. En la Figura 9, se representa de manera gráfica un ejemplo donde una nota blanca es clasificada en la posición 5. Aunque aún estamos en una etapa preliminar de análisis, en fase de entrenamiento estamos obteniendo una tasa de acierto del 85 %. Esperamos poder mejorar este valor y llegar a niveles superiores al 90 % en fase de prueba para mostrar un análisis completo de la metodología e incluir la misma en la herramienta *Sharpmony*.

## VI. Conclusiones

En este trabajo hemos presentado una aproximación factible al reconocimiento de partituras escritas a mano, como una aplicación de visión por computador y aprendizaje profundo, para asistir en el proceso de enseñanza-aprendizaje de las personas que realizan estudios de música.

Partiendo de ejercicios realizados por estudiantes de armonía de conservatorios profesionales de música, hemos desarrollado una metodología que nos permite identificar las notas escritas con una tasa de acierto superior al 95 %. Además, algunas pruebas preliminares ya desarrolladas nos permiten ver que la metodología puede mantener esas tasas de reconocimiento a la hora de identificar la posiciones de las notas sobre el pentagrama, y obtener así toda la información necesaria, junto con la armadura y el compás, para poder generar la partitura en formato MusicXML, que pueda ser procesada posteriormente por cualquier software de edición musical. El objetivo final es mejorar la herramienta *Sharpmony* con las metodología aquí descrita.

El nivel de reconocimiento logrado permite dar un paso hacia adelante en el entendimiento de problemas de visión por computador, ya que una imagen puede contener información contextual acorde a un propósito específico, que para nuestro interés es el lenguaje escrito de la música.

Nuestro desarrollo ya está en operación experimental y esperamos contar con nuevas imágenes de partituras para enriquecer el reconocimiento, así como orientar a nuevos propósitos este reconocimiento.

## Agradecimientos

Agradecemos el apoyo del Ministerio de Economía y Competitividad de España en el marco de los proyectos TIN2017-85727-C4-{2,4}-P y RTI2018-095180-B-I00, TIN2017-85727-C4-{2,4}-P y PID2020-115570GB-C21, de la Junta de Extremadura, Consejería de Comercio y Economía, del Fondo Europeo de Desarrollo Regional, una manera de construir Europa, bajo el proyecto IB16035 y de la Junta de Extremadura, proyecto GR15068 y GR18049. Este trabajo es resultado de la colaboración de la Universidad de Extremadura con la Universidad Autónoma Metropolitana en la Cd. de México, dentro del proyecto “Evolución artificial de descriptores estadísticos de textura de superficie para implementación en clasificación de imágenes digitales”, clave: EL006-18.

## Referencias

[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. Dollár, *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 real-time 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] F. Rossant y I. Bloch, *Robust and adaptive OMR system including fuzzy modeling, fusion of musical rules, and possible error detection*, EURASIP Journal on Advances in Signal Processing, vol. 2007, pp. 1-25, 2006.

[8] A. Pacha, J. Hajič, y J. Calvo-Zaragoza, *A baseline for general music object detection with deep learning*, Applied Sciences, vol. 8, n.o 9, p. 1488, 2018.

[9] L. Tuggener, I. Elezi, J. Schmidhuber, M. Pelillo, y T. Stadelmann, *Deepscores-a dataset for segmentation, detection and classification of tiny objects*, en 2018 24th International Conference on Pattern Recognition (ICPR), 2018, pp. 3704-3709.

[10] J. Calvo-Zaragoza, D. Rizo, y J. M. I. Quereda, *Two (Note) Heads Are Better Than One: Pen-Based Multimodal Interaction with Music Scores.*, en ISMIR, 2016, pp. 509-514.

[11] O. Ronneberger, P. Fischer, y T. Brox, *U-net: Convolutional networks for biomedical image segmentation*, en International Conference on Medical image computing and computer-assisted intervention, 2015, pp. 234-241.

[12] J. Hajič y P. Pecina, *The MUSCIMA++ dataset for handwritten optical music recognition*, en 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), 2017, vol. 1, pp. 39-46.

[13] J. Hajič Jr y P. Pecina, *Detecting noteheads in handwritten scores with convnets and bounding box regression*, arXiv preprint arXiv:1708.01806, 2017.

[14] J. Hajic Jr, M. Dorfer, G. Widmer, y P. Pecina, *Towards Full-Pipeline Handwritten OMR with Musical Symbol Detection by U-Nets.*, en ISMIR, 2018, pp. 225-232.

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

[16] L. Tuggener, I. Elezi, J. Schmidhuber, y T. Stadelmann, *Deep watershed detector for music object recognition*, arXiv preprint arXiv:1805.10548, 2018.

[17] K. He, G. Gkioxari, P. Dollár, y R. Girshick, *Mask r-cnn*, en Proceedings of the IEEE international conference on computer vision, 2017, pp. 2961-2969.

[18] Q. Yang, D. Xiao, y S. Lin, *Feeding behavior recognition for group-housed pigs with the Faster R-CNN*, Computers and Electronics in Agriculture, vol. 155, pp. 453-460, 2018.