Modelo polifónico para seguimiento de tono musical

Lina Fernanda Prado Gamba [1], Javier Alejandro Jiménez Cañizares [2] y Duván Alberto Andrade Cuenca [3]
lfpradog@unal.edu.co [1], javajimenezcan@unal.edu.co [2] y daandradec@unal.edu.co [3]
Universidad Nacional de Colombia, Bogotá, Colombia

***Resumen*—En los últimos años se han desarrollado diferentes herramientas para el procesamiento y análisis del sonido, una de estas herramientas es el transcriptor automático musical. En este trabajo se propone un modelo que, por medio de una red neuronal convolucional, identifica las notas de audios polifónicos interpretados en piano.**
La red consta de dos conjuntos, compuestos de tres capas (*Batch normalization*, *Dropout* y *Convolution 2D*), un *Max pooling* y una capa de salida . Fue entrenada con 32 canciones del conjunto de datos *Maestro* de Google IA. El modelo reporta 0,9617 de *accuracy* de entrenamiento, 0,9633 de *accuracy* de validación y 0.2 de *accuracy* de testeo. Además reporta 0,0075 de *f1-score* macro, 0,079 *f1-score* micro y 0,016 *f1-score weighted*. Como trabajo futuro se propone generalizar el modelo para poderlo utilizar en audios de diferentes instrumentos y generar una aplicación capaz de identificar las notas a partir de audios polifónicos, para facilitarle a los usuarios la posibilidad de interpretar canciones sin tener acceso a la partitura.

***Index Terms*—RIM, transcriptor automático musical, espectrograma, pitch tracking.**

I. Introducción

La recuperación de información musical (RIM) es un área interdisciplinar en la que diferentes ramas del conocimiento extraen información de la música y trabajan en herramientas de análisis de audio. Entre las aplicaciones encontramos clasificadores de género, reconocimiento de canciones, identificación de acordes, detección de eventos de sonido, detección de humor, extracción de características, entre otros [NZ19]. Una de las aplicaciones de RIM es la transcripción automática de la música, procesar el audio y generar su respectiva notación simbólica. Para lograr este objetivo es necesario llevar a cabo diferentes tareas de análisis de audio: identificación de instrumentos, estimación de duración, extracción de la armonía, de la melodía y del ritmo, detección de tonos, entre otros, es importante resaltar que entre mayor sea el número de instrumentos mayor es la complejidad de esta tarea.

En el presente documento se propone la construcción de un modelo de *Deep learning* de RIM para la transcripción automática de los tonos musicales de un audio, es decir, un modelo que procesa el audio e identifica la secuencia de tonos correspondientes. Se propone el uso de una red neuronal convolucional que consta de dos conjuntos, compuestos de tres capas (*Batch normalization*, *Dropout* y *Convolution 2D*), un *Max pooling* y una capa de salida. En primera instancia se preprocesan los audios: se dividen las canciones en segmentos de 128 ms de duración, luego se genera un Mel-espectrograma, y se aplica el clasificador multi-etiqueta (red neuronal) para que identifique las notas que suenan simultáneamente.

Este proyecto busca consolidarse más adelante como una aplicación móvil, por medio de la cual sea posible que estudiantes, profesionales o personas que disfrutan mucho de una canción y que desean interpretarla, pero no tengan la partitura ni el entrenamiento musical para transcribirla, consigan una transcripción automática con solo ingresar el audio.

I-A. *Preliminares*

Notas musicales: Las notas musicales son un tipo de notación para los sonidos, se componen de altura y duración. La altura es la percepción de la frecuencia del sonido, este nos permite ubicar los sonidos en la escala musical y generar un sistema universal. Este atributo está altamente correlacionado con la frecuencia, frecuencias altas son equivalentes a sonidos agudos y frecuencias bajas son equivalentes a sonidos graves. Hay doce clases de tonos en la escala musical: C, C#, D, D#, E, F, F#, G, G#, A, A# y B, cada una de estas se compone por todas las notas equivalentes al variar las octavas, en este proyecto tenemos en cuenta las siete octavas del piano. En total son 88 notas (etiquetas).[Ope][Kiv10]

Espectrograma: Los espectrogramas son una herramienta de visualización, en la cual se muestran diferentes características del sonido: frecuencia, amplitud y duración de la onda. Son los gráficos generados al variar el espectro de frecuencia en relación al tiempo, los cambios en el eje *y* corresponden a los cambios de tono (cambios de alturas) y los colores hacen referencia al volumen. Es importante resaltar que los espectrogramas de la misma nota interpretada por diferentes instrumentos varían debido al cambio de timbre, dado por los armónicos propios del instrumento (sonidos cuyas frecuencias son múltiplos de la frecuencia de la nota fundamental). [Cen15] Formalmente es el gráfico de la magnitud de la

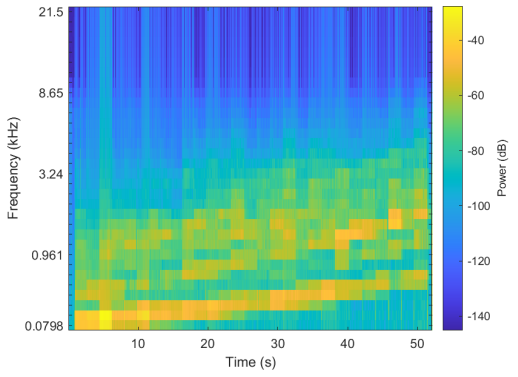

Figura 1. Espectrograma clarinete

transformada de Fourier a corto plazo:[Ell13]

$$X(\omega, \tau) = \int_{-\infty}^{\infty} x(t) w(t-\tau) e^{-it\omega} dt \quad (1)$$

Donde $x(t)$ es la onda correspondiente al audio, $w(\tau)$ es una función ventana centrada en cero. ω corresponde a la frecuencia y τ al tiempo.[Wik20b]

Los Mel-espectrogramas son aquellos en los que los audios de entrada están en escala Mel, esta escala se basa en la percepción auditiva humana; la distancia entre sonidos de frecuencias bajas son mayores en la escala Mel comparados con los sonidos de frecuencias altas, reflejando el hecho de que las personas identifican mejor la diferencia entre sonidos de bajas frecuencias. La ecuación 2 permite pasar de Hertz a escala Mel:

$$m(f) = 1127,01048 \log_{10}(1 + \frac{f}{700}) \quad (2)$$

[Wik20a]

MIDI: *Musical Instrument Digital Interface*, es una interfaz digital que posibilita la conexión entre instrumentos musicales electrónicos y ordenadores. Este sistema permite comunicar las notas musicales, el timbre y la intensidad entre otras características del sonido. Este protocolo se creó a finales del siglo XX con el fin de estandarizar la comunicación.[Sam18]

I-B. Estado del arte

En la actualidad se han llevado a cabo numerosas investigaciones sobre la generación y clasificación de música haciendo uso del *Deep learning*. En [BLBV12], los autores investigan el problema de modelar secuencias simbólicas de música polifónica en una representación de piano roll. Piano roll es un formato de almacenamiento de música que se asemeja a una partitura de una pieza musical, es la matriz de las alturas musicales contra el tiempo [Hao11]. En este trabajo se presenta un modelo probabilístico basado en *restricted Boltzmann machine* (RBM), llamado RNN-RBM, que les permite mayor precisión a la hora de realizar una transcripción polifónica. Hacen uso de representaciones de música simbólicas, compuestas por el tiempo explícito, el tono, la velocidad y la información instrumental que se encuentra generalmente en un archivo MIDI.

Por otro lado los autores [HSR+18] señalan que la mayoría de la música está altamente estructurada y se puede representar como eventos de notas discretas. Muestran, por medio del entrenamiento de notas, que un conjunto de modelos son capaces de transcribir, componer y sintetizar formas de audio. También mencionan que generar audio musical directamente con redes neuronales es notoriamente difícil porque requiere una estructura de modelado coherente en muchas escalas de tiempo diferentes.

En otro trabajo realizado por [MSB18], se propuso un método para la transcripción de notas polifónicas que hace uso de una versión adaptada de WaveNet: una arquitectura de red neuronal diseñada para el modelado de audio sin formato. Este trabajo supone un referente para el problema de transcripción de notas dado que WaveNet no se había usado para esta tarea antes.

II. Conjunto de datos[HSR+19]

Se usó el conjunto de datos *MAESTRO* V1.0.0 (MIDI and Audio Edited for Synchronous Tracks and Organization) que hace parte de la librería para Python *MAGENTA* desarrollada por Google, este es un proyecto que explora el rol del aprendizaje de máquina como herramienta del proceso creativo. [ID]

Este conjunto de datos cuenta con más de 172,3 horas de interpretaciones en piano junto con sus respectivas etiquetas de notas (con una alineación aprox. 3 ms). Este conjunto es la recompilación de las grabaciones de las presentaciones de nueve años de la competencia de piano *International Piano-e-Competition*. Los audios corresponden a interpretaciones de pianos *Yamaha Disklaviers*, para la adquisición de datos se usa el sistema integrado de captura y reproducción MIDI de alta precisión.

MAESTRO cuenta con las velocidades de pulsación de teclas, posiciones de pedal de sostenido, compositor, el título y el año de ejecución. El audio sin comprimir es de calidad CD o superior (44.1–48 kHz estéreo PCM de 16 bits). En su mayoría los audios pertenecen al género de musica clásica, y sus compositores son del siglo XVII hasta inicios del siglo XX.

En este trabajo se siguió las sugerencias de división de los conjuntos de entrenamiento, validación y testeo propuesto en [ID], debido a que esta partición garantiza que no se repita, en diferentes conjuntos, la misma composición.

Se seleccionaron 32 canciones de *MAESTRO*, con duración de entre 2,5 min y 35 min, con una media de 13,45 mins. Para los datos de entrenamiento se seleccionaron 22 canciones, para testeo 7 y para validación 3. En las figuras 2, 3 y 4 se muestran las frecuencias de sus respectivas etiquetas.

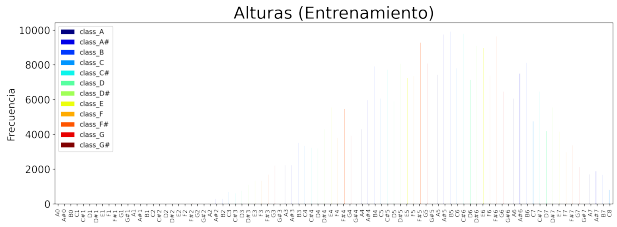

Figura 2. Frecuencias conjunto de entrenamiento

En primera instancia se divide los audios en archivos de duración de 128 ms y se generan sus respectivos Mel-espectrogramas en escala de grises, en total se generaron 200.043 imágenes, de las cuales 158.790 se usaron para entrenar el modelo, 16.026 para validación y 25.227 para testeo. Los parámetros usados fueron: radio de muestreo igual a 16kHz, *fft window* igual a 2048, *hop length* igual a 512 y *n mels* igual a 256.

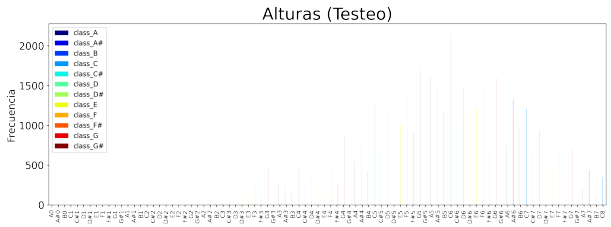

Figura 3. Frecuencias conjunto de testeo

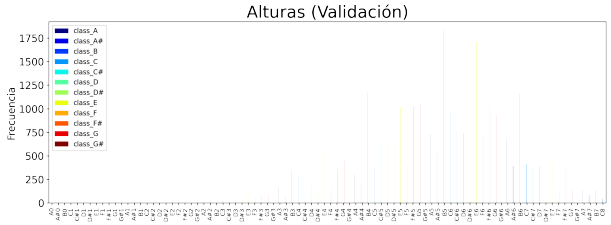

Figura 4. Frecuencias conjunto de validación

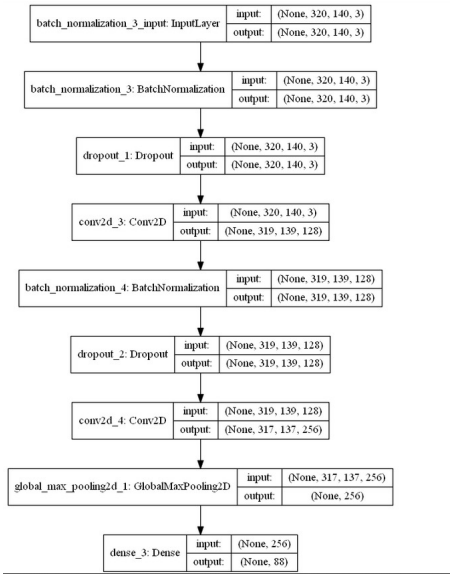

Figura 6. Segunda arquitectura de RNC

III. Metodología

Para este trabajo se plantea el uso de redes neuronales convolucionales (RNC) dado el éxito que han tenido para la tarea de clasificar imágenes. Estos resultados han inspirado el uso de RNC para RIM porque las representaciones bidimensionales de frecuencia versus tiempo (e.g., la transformada de Fourier) han demostrado ser una buena representación del audio.[SVI+15]

Para llevar a cabo el proyecto propuesto se plantearon dos arquitecturas diferentes de RNC. En la figura 5 podemos ver la primera arquitectura usada.

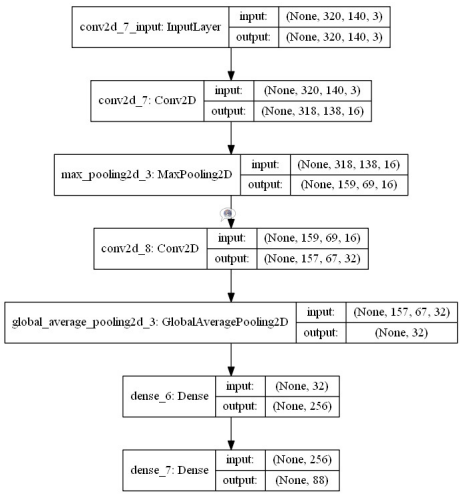

Figura 5. Primera arquitectura de RNC

La segunda propuesta usada se ilustra en la figura 6.

La primera red propuesta al igual que la segunda consta de dos conjuntos, compuestos a su vez por tres capas (*Batch normalization, Dropout y Convolution 2D*) un *Max pooling* y una capa de salida. La capa de *Batch normalization* busca producir redes cuyas activaciones por capa sean de media cero y de varianza unitaria. Esta capa tiene efectos de regularización a medida que se agregan capas a la red. También evita que los pesos de las capas siguientes deban cambiar haciendo que el entrenamiento de la red sea más acelerado mejorando los tiempos de ejecución [IS15].

Por otro lado, al aplicar *dropout* a una capa se mejora la capacidad de generalización de las redes y se reduce el riesgo de sobreajuste. Con ayuda de esta capa la robustez de la red aumenta. [SHK+14].

La capa de *convolution 2D* tiene el efecto de filtrar la imagen de entrada con un *kernel* previamente entrenado. Esto transforma los datos de tal manera que ciertas características (determinadas por la forma del *kernel*) se vuelven más dominantes en la imagen de salida al tener estas un valor numérico más alto asignados a los píxeles que las representan.

La capa Max pooling se utiliza en redes convolucionales para proporcionar una pequeña cantidad de invariancia traslacional, esto quiere decir que hace que el transcriptor de notas sea inmutable a pequeños cambios de afinación.

A pesar de que se probaron 6 configuraciones de arquitecturas para este problema las 2 mejores son las presentadas en este reporte. La primera arquitectura es más sencilla que la segunda, consta de 2 conjuntos de capas convolucionales y 2 capas densas, fue obtenida a partir de buscar las capas que la gente usaba normalmente en proyectos de transcripción musical, su complejidad es baja con sólo 36 mil parámetros para entrenar. La segunda arquitectura es más compleja en sus 2 conjuntos de capas convolucionales ya que incluye *batch normalization*, *dropouts* y filtros, además requiere del entrenamiento de 319 mil parámetros.

Se utilizó la librería sklearn para calcular las métricas, exactitud y F1 score, que evalúan el modelo, además se usó

la función de perdida *Categorical crossentropy*.

IV. Resultados

En la figura 7 y 8 podemos observar los resultados de validación y entrenamiento para ambas arquitecturas. Los resultados para la primera arquitectura sugieren valores altos para el *accuracy* y una mejoría del modelo a medida que aumentas los *epochs*. Sin embargo, para la arquitectura número dos, observamos que la métrica de *accuracy* para el conjunto de validación no presenta ningún tipo de mejora.

Figura 7. Resultados validación y entrenamiento primera arquitectura

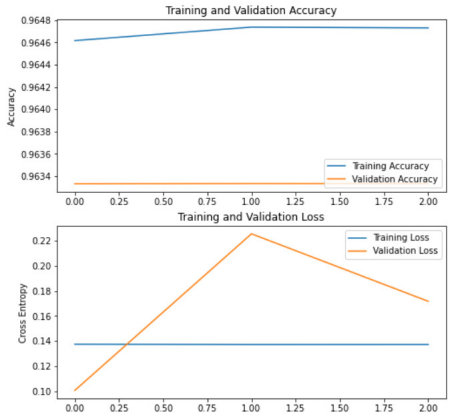

Figura 8. Resultados validación y entrenamiento segunda arquitectura

El resumen de las métricas obtenidas para ambos modelos se presentan a continuación:

	1ra Arq. 88 clases	2da Arq. 88 clases
accuracy entrenamiento	0,9647	0,9647
accuracy testeo	0,0	0,0
accuracy validación	0,9633	0,9633
F1-score macro Testeo	0,0093	0,0054
F1-score micro Testeo	0,1163	0,0743
F1-weighted	0,0368	0,0133

Cuadro I
Métricas obtenidas

En la figura 9 se presenta un ejemplo de predicción usando el modelo entrenado con la primera arquitectura.

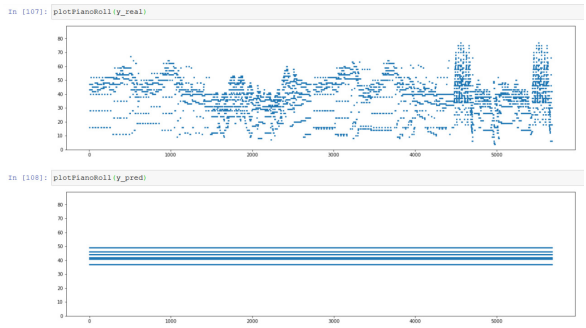

Figura 9. Ejemplo valor real vs predicción

Podemos observar en la figura anterior que el modelo logra hallar una muestra representativa que se encuentra en un promedio de las alturas interpretadas a pesar de no hacerlo con cada nota.

V. Conclusiones

- El modelo propuesto presenta overfitting debido a que alcanza un buen rendimiento con los conjuntos de validación y de entrenamiento, pero no logra buenos resultados en testeo, consideramos que la razón de esto, es posiblemente, la naturaleza del conjunto de datos. En las composiciones usualmente hay un patrón de notas que se repite en ciertas ocasiones (por ejemplo en los coros), por lo cual el conjunto de entrenamiento es bastante desbalanceado. Sería importante aumentar los datos, considerando divisiones más finas, más composiciones y géneros musicales, aunque vale la pena aclarar que el costo computacional es bastante alto.
- Aunque no se obtuvieron los resultados esperados, de los piano roll que produce nuestro modelo podemos observar que a pesar de no conseguir identificar cada nota, logra hallar una muestra representativa que se encuentra en un promedio de las alturas interpretadas.
- Para trabajos posteriores se recomienda probar otros tipos de espectrogramas y variar las calidades del mismo. También poder cambiar la partición de cada audio para obtener mejores resultados, creemos que el modelo logrará conseguir su objetivo si se juega más con los atributos que lo componen, arquitectura, particiones, dataset, etc.

VI. Apéndice

Adicionalmente se desarrolló un modelo para identificar las alturas en audios monofónico de diferentes instrumentos, figura 10, usamos una red neuronal basada en le red lenet-5 [Gaz26] para clasificar los audios en 52 categorías (correspondientes a las notas del piano, sin tener en cuenta las notas sostenidas). Se usaron los audios de otro conjunto de datos *NSynth* de *MAGENTA*. Se tomaron 18.496 espectrogramas en entrenamiento, 6.842 de testeo y 7.376 de validación, es importante resaltar que los conjuntos no comparten interpretaciones de los mismos instrumentos.

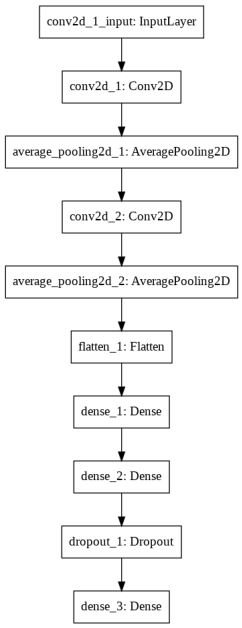

Figura 10. Modelo monofónico

Al implementar el modelo usamos la función de pérdida *Categorical crossentropy*, métrica *accuracy* y optimizador *adam*. También se probó el clasificador reduciendo el numero de clases a las notas de la cuarta octava del piano (sin contar los sostenidos). Los puntajes de los modelos se encuentran la siguiente tabla II.

	52 clases	7 clases
accuracy entrenamiento	0,937	0,998
accuracy testeo	0,021	0,15
accuracy validación	0,742	0,988
F1-score macro Testeo	0,019	0,143
F1-score micro Testeo	0,021	0,15

Cuadro II
PUNTAJES MODELO MONOFÓNICO

A partir de estos resultados llegamos a la conclusión de que el modelo sufre de *overfitting* puesto que alcanza puntajes muy altos en entrenamiento y validación, pero apenas alcanza un 0.2 de *accuracy* de testeo, en el mejor de los casos, probablemente por al diferencias de timbre entre instrumentos.

REFERENCIAS

[BLBV12] BOULANGER-LEWANDOWSKI, Nicolas ; BENGIO, Yoshua ; VINCENT, Pascal: *Modeling Temporal Dependencies in High-Dimensional Sequences: Application to Polyphonic Music Generation and Transcription*. 2012

[Cen15] CENTRE, National M.: *Spectrograms: an Introduction*. https://www.youtube.com/watch?v=_FatxGN3vAM. Version: 2015. – [Online; accessed 12-July-2020]

[Ell13] ELLIS, D: *Spectrograms: Constant-Q (Log-frequency) and conventional (linear)*. https://www.ee.columbia.edu/~dpwe/resources/matlab/sgram/. Version: 2013. – [Online; accessed 12-July-2020]

[Gaz26] GAZAR, M.: *LeNet-5 in 9 lines of code using Keras*. https://medium.com/@mgazar/lenet-5-in-9-lines-of-code-using-keras-ac99294c8086. Version: 2018, November 26. – [Online; accessed 12-July-2020]

[Hao11] HAO-WEN DONG, WEN-YI HSIAO, LI-CHIA YANG, Yi-Hsuan Y.: *Lakh Pianoroll Dataset*. https://salu133445.github.io/lakh-pianoroll-dataset/representation.html. Version: 2011

[HSR+18] HAWTHORNE, Curtis ; STASYUK, Andriy ; ROBERTS, Adam ; SIMON, Ian ; HUANG, Cheng-Zhi A. ; DIELEMAN, Sander ; ELSEN, Erich ; ENGEL, Jesse H. ; ECK, Douglas: Enabling Factorized Piano Music Modeling and Generation with the MAESTRO Dataset. In: *CoRR* abs/1810.12247 (2018). http://arxiv.org/abs/1810.12247

[HSR+19] HAWTHORNE, Curtis ; STASYUK, Andriy ; ROBERTS, Adam ; SIMON, Ian ; HUANG, Cheng-Zhi A. ; DIELEMAN, Sander ; ELSEN, Erich ; ENGEL, Jesse ; ECK, Douglas: Enabling Factorized Piano Music Modeling and Generation with the MAESTRO Dataset. In: *International Conference on Learning Representations*, 2019

[ID] ID, Google: *MAGENTA, Make music and art using machine learning*. https://magenta.tensorflow.org/. – [Online; accessed 12-July-2020]

[IS15] IOFFE, Sergey ; SZEGEDY, Christian: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: *arXiv preprint arXiv:1502.03167* (2015)

[Kiv10] KIVUMBI: *Difference Between Tone and Pitch*. http://www.differencebetween.net/miscellaneous/difference-between-tone-and-pitch/. Version: 2010. – [Online; accessed 12-July-2020]

[MSB18] MARTAK, Lukas ; SAJGALIK, Marius ; BENESOVA, Wanda: Polyphonic Note Transcription of Time-Domain Audio Signal with Deep WaveNet Architecture, 2018, S. 1–5

[NZ19] NASRULLAH, Zain ; ZHAO, Yue: Music artist classification with convolutional recurrent neural networks. In: *2019 International Joint Conference on Neural Networks (IJCNN)* IEEE, 2019, S. 1–8

[Ope] OPENMUSICTHEORY: *Pitch(Class)*. http://openmusictheory.com/pitch(Class).html. – [Online; accessed 12-July-2020]

[Sam18] SAMPLING, A.: *¿Qué es el MIDI?: La guía del principiante para la herramienta musical más poderosa*. https://bit.ly/3j0skAe. Version: 2018. – [Online; accessed 12-July-2020]

[SHK+14] SRIVASTAVA, Nitish ; HINTON, Geoffrey ; KRIZHEVSKY, Alex ; SUTSKEVER, Ilya ; SALAKHUTDINOV, Ruslan: Dropout: A Simple Way to Prevent Neural Networks from Overfitting. In: *Journal of Machine Learning Research* 15 (2014), Nr. 56, 1929-1958. http://jmlr.org/papers/v15/srivastava14a.html

[SVI+15] SZEGEDY, Christian ; VANHOUCKE, Vincent ; IOFFE, Sergey ; SHLENS, Jonathon ; WOJNA, Zbigniew: Rethinking the Inception Architecture for Computer Vision. In: *CoRR* abs/1512.00567 (2015). http://arxiv.org/abs/1512.00567

[Wik20a] WIKIPEDIA CONTRIBUTORS: *Mel scale — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Mel_scale&oldid=950711154. Version: 2020. – [Online; accessed 12-July-2020]

[Wik20b] WIKIPEDIA CONTRIBUTORS: *Short-time Fourier transform — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Short-time_Fourier_transform&oldid=949310451. Version: 2020. – [Online; accessed 12-July-2020]