



**MÉTODO DE IDENTIFICACIÓN AUTOMÁTICA PARA RECONOCIMIENTO DE
ELEMENTOS MUSICALES PRESENTES EN IMÁGENES 2D DE PARTITURAS
USANDO TÉCNICAS DE APRENDIZAJE PROFUNDO.**

**CLAUDIA PAOLA TORRES RODRÍGUEZ
JUAN CAMILO MONTAÑEZ OSORIO**

Universidad Católica de Colombia
Facultad de Ingeniería
Programa de Ingeniería de Sistemas
Bogotá, Colombia

2019

**MÉTODO DE IDENTIFICACIÓN AUTOMÁTICA PARA RECONOCIMIENTO DE
ELEMENTOS MUSICALES PRESENTES EN IMÁGENES 2D DE PARTITURAS
USANDO TÉCNICAS DE APRENDIZAJE PROFUNDO.**

**CLAUDIA PAOLA TORRES RODRÍGUEZ
JUAN CAMILO MONTAÑEZ OSORIO**

**ESTE TRABAJO DE GRADO ES PRESENTADO COMO REQUISITO PARA OPTAR AL
TÍTULO DE: INGENIERO DE SISTEMAS**

**ASESOR ROGER GUZMÁN
M. SC. (C) INGENIERÍA DE SISTEMAS Y COMPUTACIÓN**

**UNIVERSIDAD CATÓLICA DE COLOMBIA
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA DE SISTEMAS
TRABAJO DE INVESTIGACIÓN TECNOLÓGICA
BOGOTA D.C.
2019**



Atribución-NoComercial 2.5 Colombia (CC BY-NC 2.5)

La presente obra está bajo una licencia:
Atribución-NoComercial 2.5 Colombia (CC BY-NC 2.5)

Para leer el texto completo de la licencia, visita:
<http://creativecommons.org/licenses/by-nc/2.5/co/>

Usted es libre de:



Compartir - copiar, distribuir, ejecutar y comunicar públicamente la obra
hacer obras derivadas

Bajo las condiciones siguientes:



Atribución — Debe reconocer los créditos de la obra de la manera especificada por el autor o el licenciante (pero no de una manera que sugiera que tiene su apoyo o que apoyan el uso que hace de su obra).



No Comercial — No puede utilizar esta obra para fines comerciales.

El presente trabajo se dedica a
Dios, a Roger Montañez,
Maryoribeth Osorio, Oscar
Montañez y Anamaria Montañez
con todo mi amor, sin ustedes y su
apoyo como mayor motivación no
hubiera sido posible.

Nota de aceptación.

Aprobado por el comité de grado en cumplimiento de los requisitos exigidos por la Facultad de Ingeniería y la Universidad Católica de Colombia para optar al título de ingeniero de sistemas

Manuel Báez
Jurado 1

Juan Carlos Barrero
Jurado 2

Roger Enrique Guzmán Avendaño, Msc.
Asesor

AGRADECIMIENTOS

Agradecemos en primer lugar a Dios por permitir la realización de este proyecto, a Roger Guzmán director y amigo, que brindó los lineamientos necesarios para el desarrollo del proyecto, a Nicolas Diaz, Brayan Criollo y Nicolas Cortés amigos quienes apoyaron la materialización de los experimentos, a Roger Montañez y Maryoribeth Osorio, Padres por su apoyo incondicional en este proceso formativo de pregrado y con quienes estaré infinitamente agradecido

Tabla de Contenido

RESUMEN	11
PALABRAS CLAVES.....	11
ABSTRACT.....	12
INTRODUCCIÓN.....	13
1. GENERALIDADES	15
1.1. LÍNEA DE INVESTIGACIÓN O ALTERNATIVA	15
1.2. PLANTEAMIENTO DEL PROBLEMA.....	15
1.2.1. PREGUNTA DE INVESTIGACIÓN	16
1.2.2. ALCANCES Y LIMITACIONES	16
1.3. JUSTIFICACIÓN	17
1.4. OBJETIVOS.....	18
1.4.1. OBJETIVO GENERAL.....	18
1.4.2. OBJETIVOS ESPECÍFICOS.....	18
2. MARCOS DE REFERENCIA.....	19
2.1. MARCO CONCEPTUAL	19
2.1.1. PREPROCESAMIENTO.....	19
2.1.2. NOTACIÓN MUSICAL.....	19
2.1.2.1. Partitura	19
2.1.2.2. Pentagrama.....	20
2.1.2.3. Figuras Musicales.....	20
2.1.3. EXTRACCIÓN DE CARACTERÍSTICAS	20
2.1.3.1. Estandarización	21
2.1.3.2. Normalización	21
2.1.3.3. Muestreo.....	21
2.1.3.4. Inception V2.....	21
2.1.4. mAP.....	22
2.1.6. IoU.....	23
2.2. ESTADO DEL ARTE	24
2.3. MARCO TEÓRICO	25
2.3.1. SMARTSCORE – NOTEREADER	25
2.3.2. MUSICPAL.....	26

2.3.3.	SHEETVISION	27
2.3.4.	Imágenes 2D.....	28
2.3.5.	Aprendizaje automático.....	28
2.3.6.	Inteligencia artificial.....	28
2.3.7.	Aprendizaje profundo o Deep Learning.....	28
2.3.8.	Redes neuronales.....	29
2.3.9.	Aprendizaje supervisado.....	29
2.4.	MARCO JURÍDICO O NORMATIVO	29
2.5.	MARCO GEOGRÁFICO	30
3.	METODOLOGÍA	30
3.1.	FASES DEL PROYECTO	31
4.	CRONOGRAMA DE ACTIVIDADES.....	33
5.	PRESUPUESTO	34
6.	INSTRUMENTOS O HERRAMIENTAS UTILIZADAS.....	36
6.1.1.	Tensorflow	36
6.1.2.	LabelImg.....	36
6.1.3.	GPU NVIDIA GEFORCE GTX 1050	37
6.1.4.	Python 3.6	37
6.1.5.	COCO.....	37
6.1.6.	Faster-RCNN Inception V2	38
6.1.7.	Github	39
7.	RECOLECCIÓN DE DATOS	39
7.1.1.	TIPOS DE DATOS.....	40
8.	DESARROLLO DE LA PROPUESTA	41
	Etiquetado de imágenes:	41
	División de conjunto de datos:	42
	Conversión de XML a CSV:.....	43
	Conversión de archivos CSV a TFRecord:	43
	Configuración de entrenamiento:.....	44
	Entrenamiento del modelo:.....	46
	Evaluación del modelo:	48
9.	RESULTADOS	49
10.	CONCLUSIONES.....	51

11.	RECOMENDACIONES.....	51
12.	TRABAJOS FUTUROS	52
13.	ESTRATEGIA DE COMUNICACIÓN Y DIVULGACIÓN	52
14.	REFERENCIAS.....	52

Tabla de Tablas

TABLA 1 RECURSO VS PRECIO	34
TABLA 2 ACTIVIDAD VS RECURSO	35
TABLA 3 SERVICIOS	35
TABLA 4 COMPARACIÓN DE EXPERIMENTOS	49

Tabla de Ilustraciones

Ilustración 1: Fases del proyecto	14	
Ilustración 2: Partitura Musical ¹⁴	19	
Ilustración 3: Elementos musicales.....	20	
Ilustración 4 Extracción de características.....	22	
Ilustración 5: Overlap/Union.....	23	
Ilustración 6: Interfaz NoteReader ¹⁰	25	
Ilustración 7: Interfaz 1 MusicPal 11	Ilustración 8: Interfaz 2 MusicPal ¹¹	26
Ilustración 9: SheetVision ¹²	27	
Ilustración 10 :Aprendizaje supervisado	29	
Ilustración 11 Metoología	32	
Ilustración 12 arquitectura Faster RCNN	39	

Ilustración 13 Imágenes del Conjunto de datos	40
Ilustración 14 Etiquetado de Imágenes	42
Ilustración 15 Proceso de entrenamiento	47
Ilustración 16 Medidas de desempeño CMD.....	48
Ilustración 17 Resultado de la identificación	49
Ilustración 18 Salida del Algoritmo	50

RESUMEN

Este proyecto surge de la necesidad de mejorar el proceso de aprendizaje musical basado en la lectura de partituras. Se realiza este enfoque teniendo en cuenta que la lectura musical es el fundamento de todas las habilidades tanto físicas como cognitivas necesarias para la interpretación de cualquier obra musical en un instrumento.

Con el fin de apoyar y fomentar el interés de las personas que desean o que han iniciado un proceso de aprendizaje musical, se busca desarrollar un algoritmo que facilite la identificación y clasificación de los elementos que componen una partitura.

Teniendo en cuenta que se busca identificar figuras musicales dentro de partituras, es necesario que la máquina pueda obtener los caracteres encontrados en ella a través del procesamiento de imágenes, por lo tanto, el experimento consta de la implementación de un algoritmo enfocado al reconocimiento de elementos, en este caso figuras musicales en imágenes de partituras en dos dimensiones usando aprendizaje profundo.

Se decide hacer uso del aprendizaje profundo buscando simular la acción humana de aprender por medio del entrenamiento del algoritmo con unas características determinadas. La implementación de la arquitectura Faster-RCNN se hace con el fin de disminuir tiempos de ejecución, y aumentar la precisión en los resultados que serán evaluados con las métricas de desempeño precision y recall, esto con el fin de monitorear y cuantificar que el experimento funcione de manera rápida y completa.

El experimento se realiza con diferentes porcentajes de muestreo para prueba y entrenamiento, pero se concluye que los mejores resultados se obtuvieron con 70% de los datos como entrenamiento y 30% como prueba, las métricas de Average Precision y Average Recall reflejan una precisión del 71.8% y un AR75 del 69.4% con un intervalo de rigurosidad de las pruebas IoU del 0.75, y se concluye que puede obtenerse mejores resultados bajo parámetros diferentes, como la forma de etiquetado y el tipo de imágenes del dataset.

PALABRAS CLAVES

APRENDIZAJE PROGRAMADO, RECONOCIMIENTO DE CARACTERES, INTELIGENCIA ARTIFICIAL, ALGORITMO, MÚSICA, APRENDIZAJE DE MAQUINA, TÉCNICA DE APRENDIZAJE.

ABSTRACT

This project arises from the need to improve the process of musical learning based on the reading of scores. This approach is carried out bearing in mind that musical reading is the foundation of all the physical and cognitive abilities necessary for the interpretation of any musical work in an instrument.

In order to support and promote the interest of people who want or have started a musical learning process, an algorithm is sought to facilitate the identification and classification of the elements that make up a score.

Considering that it is sought to identify musical figures within scores, it is necessary that the machine can obtain the characters found in it through image processing, therefore, the experiment consists of the implementation of an algorithm focused on the recognition of elements, in this case musical figures in images of scores in two dimensions using deep learning.

It is decided to make use of deep learning seeking to simulate the human action of learning by training the algorithm with certain characteristics. The implementation of the Faster-RCNN architecture is done in order to decrease execution times, and increase the accuracy of the results that will be evaluated with precision and recall performance metrics, this in order to monitor and quantify that the experiment works quickly and fully.

The experiment is performed with different percentages of sampling for trial and training, but it is concluded that the best results were obtained with 70% of the data as training and 30% as test, the metrics of Average Precision and Average Recall reflect an accuracy of 7.18% and an AR75 of 6.94% with an lou test severity range of 0.75, and it is concluded that better results can be obtained under different parameters, as the form of labelling and the type of images in the dataset.

INTRODUCCIÓN

En la actualidad el bajo interés y conocimiento de la música da lugar al termino de analfabetismo musical que es el resultado de altos niveles de dificultad en su proceso de aprendizaje, en especial para quien desea aprender música y no reconoce y tampoco identifica los elementos dentro de una partitura, ya que este es un proceso que requiere un alto nivel de experiencia en el campo de las artes musicales.¹

Este proyecto diseña e implementa un experimento basado en inteligencia artificial para el reconocimiento de elementos musicales en imágenes de partituras de dos dimensiones usando aprendizaje profundo; en los últimos años se ha incrementado la aplicación de aprendizaje profundo en diferentes ámbitos, especialmente en aquellos que involucran la identificación de elementos.

El tratamiento de las imágenes consta de tres etapas: procesamiento, segmentación-detección y por último la clasificación de los elementos musicales, este trabajo se desarrollará y ejecutará únicamente dentro del margen de las dos primeras fases: el procesamiento de imágenes y la segmentación-detección de elementos musicales en la partitura logrado así procesar la imagen para aislar los elementos que la componen y posteriormente detectar la figura musical dentro del pentagrama, esto servirá como base para trabajos futuros que busquen una clasificación e interpretación completa de los elementos musicales, es decir los detalles específicos de los tiempos asignados a cada nota o el nombre de las mismas.

Uno de los problemas en el procesamiento de partituras es el tiempo respuesta del algoritmo en la identificación de cada uno de los elementos musicales, para mitigar los tiempos de respuesta se plantea un experimento con las 3 etapas anteriormente mencionadas, el primer paso consta del procesamiento de la imagen en donde “se adaptará la imagen para el funcionamiento de los pasos siguientes”² teniendo en cuenta que se maneja un conjunto de datos aproximadamente de 2093 figuras musicales en formato digital extraídas de fotos de partituras que fueron previamente escaneadas o descargadas.

¹ Mercé, Vilar i Monmany. «Educación y desarrollo musicales desde un punto de vista cognitivo.» En *Educación y desarrollo musicales desde un punto de vista cognitivo*, de L Davinson,, & L Scripp, 5-6. Barcelona: LEEME(Lista Electronica Europea de Música en la Educación, 2004.

² Carretero De la Rocha, D. (Junio de 2009). *Universidad Carlos III de Madrid*. Obtenido de Sistema de Reconocimiento de Partituras Musicales: <https://e-archivo.uc3m.es/handle/10016/7207>

Como segundo paso, en la segmentación-detección se extraerán las características de la imagen y se detectarán los elementos musicales presentes en el pentagrama. La definición aplicada aísla los elementos dentro de la partitura para su análisis en la fase de reconocimiento, donde según sus características principales se clasifica, y ahí es relacionado con su nombre o notación musical.

Etapas del reconocimiento de partituras

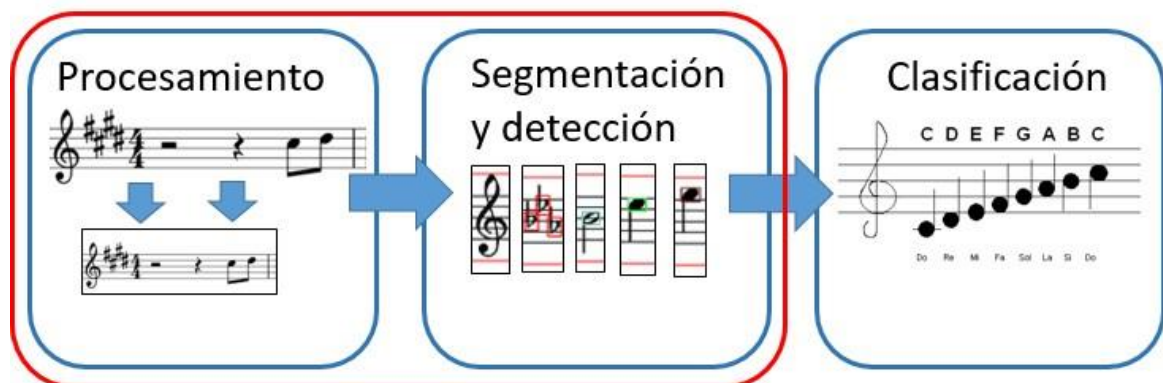


Ilustración 1: Fases del proyecto

Fuente: Autores

La culminación completa y satisfactoria de estas etapas depende de la evaluación del algoritmo por medio de un análisis de los resultados utilizando las métricas de desempeño precisión y recall. El uso de aprendizaje profundo ha generado grandes avances en técnicas de procesamiento de imágenes como se menciona en el libro "Python Deep Learning"³. Se resalta por su potencial y utilidad en distintos tipos de aplicaciones como el procesamiento de imágenes y en este caso en específico aporta un valor adicional al hecho que puede aplicarse con éxito a grandes volúmenes de datos para el descubrimiento de patrones tanto observados como no observados, buscando de esta manera obtener que los resultados superen las expectativas por los niveles de precisión.

³ Vasilev, Ivan, Daniel Slater, y Gianmario Spacagna. *Python Deep Learning*. Birmingham: Packt Publishing Ltd., 2019

1. GENERALIDADES

1.1. LÍNEA DE INVESTIGACIÓN O ALTERNATIVA

Trabajo de investigación tecnológica.

1.2. PLANTEAMIENTO DEL PROBLEMA

En el transcurso de la historia y atravesando todos los periodos de la humanidad, la música se ha fundamentado en un sistema de escritura de gran complejidad. Este sistema se refleja en documentos manuscritos o impresos que indican como debe interpretarse una composición musical por medio de elementos, notas y tiempos. Actualmente el no adquirir conocimiento o interés en la música como parte de la cultura, es algo común porque en la diversión y en la educación diaria prima la música de consumo industrial y no cultural⁴ y es aquí donde aparece el término “Analfabetismo musical”,

Una de las razones del analfabetismo musical es la sensación de incapacidad de las personas al intentar entender la música a través de la escritura y comunicarse con ella, debido a la complejidad que ésta demanda. Según la aplicación de la encuesta realizada a músicos y publicada en la “Revista de ciencias sociales”⁵ la partitura es la estrategia de adquisición más alta del repertorio con un 73%, y según los resultados del cuestionario realizado en el artículo “Métodos de lectura musical y el aprendizaje de nuevos repertorios”⁶ muestran que el 85,3% utilizan como principal método de lectura musical la partitura, logrando así demostrar la importancia de la misma y la alta necesidad de su correcta lectura al momento de iniciar un estudio musical.

El libro “Psychology of music”⁷ demuestra la influencia de la música en el incremento del uso de la inteligencia principalmente en aquellas personas que son capaces de leer partituras, así que la frustración que se puede generar frente a esta tarea, definitivamente es uno de los obstáculos más grandes en el proceso de aprendizaje musical, junto con los bajos niveles de disciplina, perseverancia y coordinación, teniendo en cuenta también los largos periodos de tiempo y concentración que se requiere.

De acuerdo con el estudio presentado en el artículo “La Notación Musical.

⁴ Marco, T. (Abril 2017). Analfabetos musicales. Melómano, 229

⁵ Valverde, X. (2017). Contexto de aprendizaje musical y de transmisión cultural. Revista de Cienciassociales (CI) núm. 39, 110

⁶ Diego Calderón, J. G. (2018). Métodos de lectura musical y el aprendizaje de nuevos repertorios. Sonograma magazine num 037.

⁷ Welch, G. (Julio 2004). Psychology of music. United Kingdom: Graham Welchsempre

Dificultades De Lecto-Escritura En Alumnos”⁸ Que consiste en analizar series melódicas para posteriormente reconocer los elementos y leerlos por su nombre. Los resultados evidencian que, de diez (10) sujetos sometidos a la prueba, ocho (8) confunden la lectura de alguna de las series presentadas. Por lo tanto, se demuestra una gran dificultad en el reconocimiento de elementos musicales.

Actualmente se encuentran herramientas capaces de reconocer, clasificar e interpretar una partitura con notas extraídas únicamente del piano a través de imágenes. Herramientas como NoteReader, SheetVision y MusicPal utilizan métodos de reconocimiento de imágenes y aprendizaje de máquina, aun así, se encuentran problemas como los tiempos de respuesta de las herramientas nombradas anteriormente y en ocasiones no se reconoce en su totalidad los símbolos de la partitura, por eso se busca solucionar estos problemas por medio de un algoritmo de identificación de elementos musicales en partituras de tal manera que apoyen el proceso de lectura de las mismas.

Se decide utilizar la técnica de aprendizaje profundo, ya que como se nombra en el libro “Python Deep Learning”³, el aprendizaje profundo basado en redes neuronales convolucionales, es capaz de aprender por sí mismo sobre las características de un objeto determinado sin que requieran ser establecidas previamente. Según el libro “Python Deep Learning”³, esta técnica presenta una ventaja frente a la necesidad de identificar caracteres dentro de imágenes, ya que se encarga de obtener las características más relevantes dentro de la imagen y luego por medio de la correlación entre pixel y pixel se puede llegar a determinar la forma de un elemento.

1.2.1. PREGUNTA DE INVESTIGACIÓN

¿Cómo apoyar el proceso de identificación de elementos musicales en partituras usando aprendizaje profundo?

1.2.2. ALCANCES Y LIMITACIONES

ALCANCES: El Proyecto se encargará de diseñar e implementar una estrategia metodológica para detectar los elementos musicales como, claves, compás y notas en partituras utilizando aprendizaje de máquina mediante el lenguaje de programación Python 3 dentro del segundo periodo académico del 2019

⁸ Gallo, Eva Maria, y María Isabel Reyzábal. «LA NOTACIÓN MUSICAL. DIFICULTADES DE LECTO-ESCRITURA EN ALUMNOS DE 2º CICLO DE E.S.O.» Red de Revistas Científicas de América Latina, el Caribe, España y Portugal, 2005: 465-485.

LIMITACIONES: Este trabajo se desarrollará y ejecutará únicamente dentro del margen de desarrollo de las dos primeras fases: el procesamiento de imágenes y la segmentación-detección de los elementos musicales en las partituras.

1.3. JUSTIFICACIÓN

Según el modelo propuesto por Howard Gardner existen diferentes tipos de inteligencia⁹, una de estas corresponde a la inteligencia musical, la cual se relaciona directamente con la capacidad de percibir, transformar y expresarse mediante las formas musicales. Este experimento se realiza con el fin de fortalecer este tipo de inteligencia y fomentar un desarrollo integral y completo de cualquier persona.

A largo plazo, la música juega un papel importante en el bienestar de la persona, así que este experimento será una línea base la cual podrá ser utilizada para llegar a mejorar habilidades verbales, aumentar la creatividad y desarrollar capacidades de autonomía; incluso puede ser la base para potenciar un talento y desarrollar una pasión junto con la autoestima.

La lectura es la interpretación de lo que el lector es capaz de comprender y aprender, esto depende fuertemente de lo que conoce con anterioridad, por tanto este experimento va dirigido específicamente a las personas con deseos de iniciar un proceso de aprendizaje musical empírico, en este caso lectores o intérpretes de la música, logrando ofrecer numerosos conocimientos y destrezas que la lectura musical implica, teniendo en cuenta los distintos tipos de información que contiene como: altura, duración e intensidad.

Este experimento está dirigido a quienes inician un proceso de aprendizaje musical teniendo en cuenta que la lectura de partituras es una actividad que implica una serie de habilidades complejas tales como: la decodificación de los símbolos del sistema de escritura, la construcción de una representación mental, la interpretación como proceso de significación, el control de una ejecución ajustada y el conocimiento del marco métrico y del sistema tonal de referencia¹⁰, entre otras, existe una

⁹ Gardner, H. (2011). *Inteligencias múltiples, la teoría en la práctica*. PAIDOS IBERICA.

¹⁰ Pereira Ghiena, A. (2014). *ACADEMICA*. Obtenido de [file:///C:/Users/Juan/AppData/Local/Packages/Microsoft.MicrosofEdge_8wekyb3d8bbwe/TempState/Downloads/Tesis_Maestria_PereiraGhiena\(final\)%20\(1\).pdf](file:///C:/Users/Juan/AppData/Local/Packages/Microsoft.MicrosofEdge_8wekyb3d8bbwe/TempState/Downloads/Tesis_Maestria_PereiraGhiena(final)%20(1).pdf)

necesidad real a la que este proyecto responde, y es facilitar la adquisición de conocimientos musicales por medio de un algoritmo que brinda apoyo al proceso de decodificación de símbolos con el fin de fortalecer las habilidades implicadas en el proceso de aprendizaje musical restante.

Se propone este experimento con el fin de que sea implementado para agilizar el proceso de reconocimiento de figuras musicales en el pentagrama y posteriormente poder ser utilizado para lectura de partituras, disminuyendo así el tiempo requerido para el fortalecimiento y desarrollo de la técnica musical de la persona y utilizando aprendizaje de máquina para lograr bajos tiempos de ejecución, mejorar la percepción computacional al encontrar caracteres en datos aparentemente inexistentes u ocultos y altos niveles de precisión en los resultados.

1.4. OBJETIVOS

1.4.1. OBJETIVO GENERAL

Implementar un método de identificación automática para reconocimiento de elementos musicales presentes en imágenes 2D de partituras usando técnicas de aprendizaje profundo.

1.4.2. OBJETIVOS ESPECÍFICOS

- Construir un conjunto de datos con partituras musicales para el entrenamiento del algoritmo.
 - Diseñar una estrategia metodológica basada en aprendizaje profundo para identificar los elementos presentes en las partituras musicales.
 - Desarrollar un algoritmo basado en aprendizaje profundo, para la identificación de elementos musicales en imágenes de partituras 2D.
 - Evaluar la precisión y exactitud del algoritmo de identificación sobre imágenes 2D para medir el rendimiento del algoritmo usando métricas como precision y recall.
-

2. MARCOS DE REFERENCIA

2.1. MARCO CONCEPTUAL

2.1.1. PREPROCESAMIENTO

En las operaciones de imágenes aparecen diferentes aspectos físicos y visuales con una alta complejidad a la hora de controlar, entre estos se encuentran los niveles bajos de iluminación, la textura y el ruido que la mayoría posee. Por estos motivos es necesario realizarles un proceso a dichas imágenes, utilizando técnicas con la cual se le pueda reducir el ruido y realzar el contraste.

Estos procesos se aplican con el fin de obtener un mayor porcentaje de acierto a la hora de obtener las características de la imagen y para realizar el proceso de clasificación.¹¹

2.1.2. NOTACIÓN MUSICAL

Este sistema de escritura es utilizado para representar una pieza musical de forma gráfica, permitiendo ser ejecutada por el intérprete, de manera deseada por el compositor, la notación es representar sobre un pentagrama una serie de signos. El elemento básico de cualquier sistema de notación musical es la nota, que representa un único sonido y sus características básicas: duración y frecuencia. Los sistemas de notación también permiten representar otras características diversas, tales como variaciones de intensidad, expresión o técnicas de ejecución instrumental. No obstante, existen muchos otros sistemas de notación y muchos de ellos también se usan en la música moderna¹²

2.1.2.1. Partitura

Una partitura es un documento manuscrito o impreso que indica cómo debe interpretarse una composición musical, mediante un lenguaje propio formado por signos musicales y el llamado sistema de notación.



Ilustración 2: Partitura Musical¹⁴

¹¹ Gutiérrez, José Antonio Taquía. «El procesamiento de imágenes y su potencial.» Universidad de Lima, 2017

¹² GUEVARA SANIN, JUAN SEBASTIAN. Teoria Musical. 2015.

2.1.2.2. Pentagrama

El término pentagrama se refiere a cinco líneas paralelas sobre las cuales se escriben las diferentes figuras musicales. Según el músico Juan Sebastian Guevara el quien define la música como “el arte de combinar sonidos de manera armónica”¹¹ habla acerca de que el uso del pentagrama es aquel que permite a los músicos descifrar el sonido, el tiempo de las figuras musicales y la correcta interpretación de una obra.

2.1.2.3. Figuras Musicales

Son aquellos símbolos comúnmente utilizados para informar acerca de la duración de las notas musicales y la tonalidad. La ubicación de la cabeza o núcleo en el pentagrama proporciona información acerca de la altura a la cual se debe tocar la figura musical.¹³

Figura	Nombre	Valor
	Cuadrada	El doble de una redonda
	Redonda	El doble de una blanca Mitad de una cuadrada
	Blanca	El doble de una negra Mitad de una redonda
	Negra	El doble de una corchea Mitad de una blanca
	Corchea	El doble de una semicorchea Mitad de una negra
	semicorchea	El doble de una fusa Mitad de una corchea
	fusa	El doble de una semifusa Mitad de una semicorchea
	semifusa	Mitad de una fusa

Ilustración 3: Elementos musicales

2.1.3. EXTRACCIÓN DE CARACTERÍSTICAS

¹³ Sosa , Roberto, y Mariela Dominguez. Conceptos Básicos del Lenguaje Musical. San Luis: Universidad de La Punta, 2014.

Una característica es un atributo que se identifica de cada imagen. Las características pueden ser almacenadas en vectores bidimensionales ($x=x_1, x_2, x_n$)

Entre los procesos para la realizar un proceso correcto de extracción de características, necesitamos tener en cuenta las siguientes transformaciones

2.1.3.1. Estandarización

En este proceso se pueden encontrar diferencias entre variables, es decir, pueden existir variables similares, pero con diferentes escalas, es por esto que se debe aplicar un proceso de estandarización. Lo siguiente se puede considerar como un ejemplo, $x = [x_1, x_2]$ en donde x_1 representa el ancho y x_2 representa el largo. Se puede hacer una comparación de x_1 Vs x_2 pero el proceso correcto sería realizar primero una normalización de estos datos.

2.1.3.2. Normalización

Un ejemplo de normalización es el caso donde x es una imagen y las x_i son el número de píxeles que poseen color i . Tiene sentido común realizarle un proceso de normalización a la variable x , dividiéndola por el número total de conteos que se tienen para codificar la distribución y eliminar la característica de dependencia que se tiene con el tamaño de la imagen. Esto se puede entender o traducir con la siguiente fórmula:

$$x' = \frac{x}{||x||}$$

2.1.3.3. Muestreo

El muestreo puede ser particularmente útil con conjuntos de datos que resultan muy grandes para analizarlos de manera eficiente en su totalidad, esta metodología se puede evidenciar en procesos de Big Data.

El muestreo permite que los científicos de datos, los modeladores de predicciones, entre otros tipos de analistas de datos trabajen con una cantidad pequeña o grande de datos para construir y ejecutar modelos analíticos de manera rápida y efectiva, mientras se realizan resultados más precisos

2.1.3.4. Inception V2

Inception V2 es una arquitectura usada para la extracción de características basada en redes neuronales convolucionales. Esta arquitectura tiene como objetivo extraer las características de una imagen evitando la pérdida masiva de características, lo cual ocasiona conflictos a la hora de identificar los elementos.

Inception V2 cuenta con la siguiente ecuación:

$$G[m, n] = (F * h)[m, n] = \sum_j \sum_k h[j, k] F[m - j, n - k]$$

Siendo G la salida convolucional para cada uno de los valores, F la imagen de entrada, h el filtro determinado por la arquitectura y m,n las dimensiones del filtro.

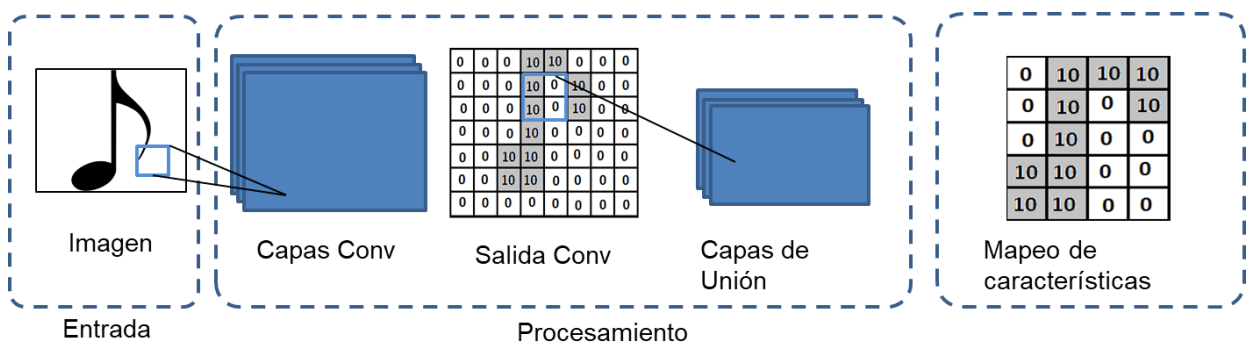


Ilustración 4 Extracción de características

Fuente: Autores

2.1.4. mAP

El mean Average Precision (mAP) es una herramienta enfocada a la fácil detección de elementos con una medición de 0 a 1, en cuanto más tiende a 1 el resultado es mayor su precisión.

La precisión mide cuán precisas son sus predicciones. es decir, el porcentaje de sus predicciones son correctas.

Recordar mide qué tan bueno encuentra todos los aspectos positivos. Por ejemplo, podemos encontrar el 80% de los posibles casos positivos en nuestras predicciones principales K.

Aquí están sus definiciones matemáticas:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

TP= True Positive

TN= True Negative

FP= False Positive

FN= False Negative

2.1.6. IoU

IoU es la forma de medir un elemento determinando, un "overlap" el cual es el área a reconocer, posteriormente se establece el porcentaje umbral de falla con el fin de asignar la precisión de la identificación. En este método se fijan dos áreas delimitadas una a reconocer y la otra es la predicción, de allí arroja la precisión de identificación con un área de "Unión".

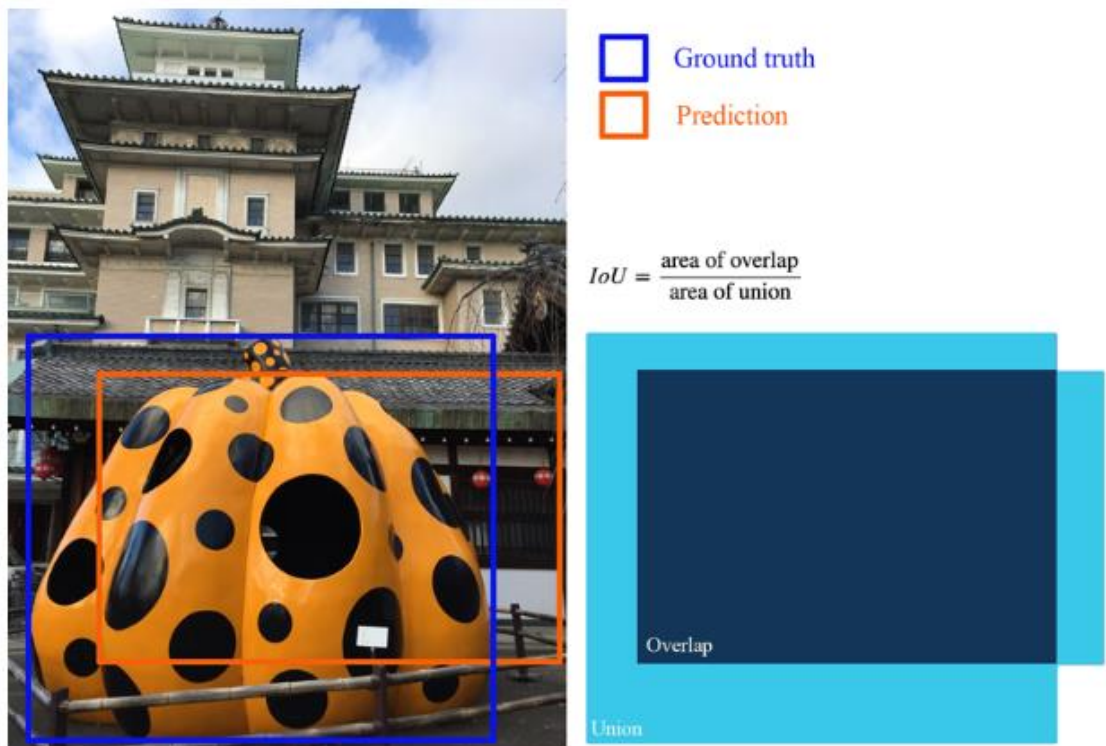


Ilustración 5: Overlap/Union

2.2. ESTADO DEL ARTE

Según el Artículo “Reconocimiento Automático De Partituras Para Guitarra Por Medio De Visión Artificial Y Redes Neuronales Artificiales”¹⁴ se implementa un algoritmo con el objetivo de brindarle al estudiante herramienta que permita facilitarle la lectura de partituras usando visión por computadora, esta herramienta fue desarrollada en MATLAB y buscaba leer las líneas de la partitura e interpretarlas en la posición de la mano en el diapasón de la guitarra. Para la identificación se implementaron dos algoritmos diferentes OCR y RNA y como resultado se analizó cuál de los dos tiene mejor desempeño. En el informe de desempeño se pudo evidenciar que la técnica RNA identificó más del 60% de símbolos una vez ejecutado el entrenamiento varias veces, mientras que la técnica OCR logró identificar satisfactoriamente un 85% de los signos en la primera ejecución, teniendo en cuenta de que son partituras digitales. En cuanto a las partituras manuscritas, se encuentra que las RNA superan a la técnica de OCR debido a las diferencias morfológicas de los símbolos teniendo en cuenta de que cada persona tiene una manera particular de escribir

En el artículo “A System for Optical Music Recognition and Audio Synthesis” (Wallner, 2014) aparecen el término OMR que es implementado a través de una metodología que se divide en procesamiento de la imagen identificación y eliminación de las líneas de la partitura y por último se hace una clasificación de las características. El método OMR es basado en la comparación de plantillas, la cual, mediante unas plantillas determinadas de símbolos musicales como clave de sol, negra, redonda entre otras, procede a buscar dichos elementos en los elementos encontrados en la partitura. La debilidad con la que cuenta el modelo es que en ocasiones cuando no encuentra un parecido bastante alto entre la plantilla y el objeto encontrado, procede a ignorarlo, lo cual hace que este modelo frente a partituras escritas a mano presente deficiencias al no estar entrenado con símbolos manuscritos.

A través de un experimento realizado en el artículo “Implementation of exemplar- based learning model for music cognition”¹⁵ se busca crear una herramienta basada en un modelo de aprendizaje. Se utilizaron 2 modelos, El primero Clasificador K- Vecinos cercanos que utiliza vectores para calcular su distancia a un centroide y de esta manera por medio de K-Clusters poder clasificar los elementos. Por último, se implementa el algoritmo genético, un algoritmo evolutivo el cual consiste en la representación genética de una cadena de cromosomas que a su vez

¹⁴ Varela Cañón, Jeferson Camilo. «Reconocimiento automatico de partituras para guitarra por medio de vision artificial y redes neuronales artificiales.» Universidad La Salle, 2018.

¹⁵ Fujinaga, I., Moore, S., & Sullivan Jr, D. S. (1998). Implementation of exemplar-based learning model. Universidad Johns Hopkins

tienen 3 operaciones llamadas, mutación, inversión y mezcla. El experimento arroja como resultado reconoce un 99% de los símbolos musicales.

En el documento "Deep Neural Networks for Document Processing of Music Score Images"¹⁶ se realiza un experimento basado en aprendizaje profundo utilizando 3 enfoques diferentes de redes neuronales convolucionales las cuales mediante la técnica de OCR se encargan de realizar un reconocimiento a manuscritos de partituras musicales de la época medieval. El experimento que cuenta con un enfoque de red pre entrenado, arroja excelentes resultados al mostrar que requiere una menor cantidad de datos de entrenamiento para ser efectivo

2.3. MARCO TEÓRICO

En el marco teórico se mostrarán las tecnologías que han incursionado en el ámbito de la identificación de partituras musicales y se mostrarán las teorías demostradas que se emplearán de alguna manera dentro del desarrollo del proyecto.

2.3.1. SMARTSCORE – NOTEREADER



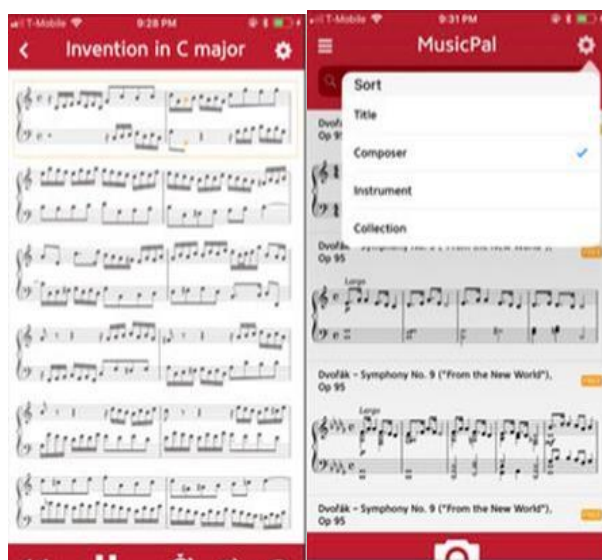
Ilustración 6: Interfaz NoteReader¹⁰

Smart Score es una aplicación móvil disponible para Android en tienda GooglePlay ofrecida por Musitek que se encarga de la lectura de partituras musicales por medio de un Smart Phone. Esta aplicación captura canciones con la cámara del SmartPhone y de vuelta reproduce la partitura

¹⁶ Calvo Zaragoza, Jorge, Francisco J. Castellanos, Gabriel Vigliensoni, y Ichiro Fujinaga. «Deep Neural Networks for Document Processing of Music Scores Images.» *MDPI*, 2018

musical. Como se presenta en la Ilustración 1, la interfaz de NOTE Reader permite navegar en la aplicación a través de las opciones de capturar imagen, por medio de la cámara del dispositivo móvil, también permite cargar un archivo pdf o importar una imagen desde la galería luego de realizar este procedimiento, la aplicación empieza a hacer el reconocimiento de la partitura y posterior a ello reproduce el sonido de la partitura. La aplicación presenta una calificación de 2.1 estrellas en PlayStore por que tarda demasiado al momento de reproducir la canción, cuenta con problemas para identificar los elementos musicales confundiéndonlos o pasándolos por alto¹⁷

2.3.2. MUSICPAL



MusicPal 11

Ilustración 8: Interfaz 2 MusicPal¹¹



Ilustración 7: Interfaz 1

Esta aplicación para IOS permite tomar una fotografía a cualquier partitura y reproducirla utilizando visión por computadora. La aplicación con el avanzar del tiempo aprende con el uso que se le da gracias al aprendizaje de máquina y a los usuarios que desean contribuir con el desarrollo de esta aplicación. Como se ve en la Ilustración 2 la aplicación funciona activando la cámara en la aplicación y posteriormente, colocar la partitura frente a la cámara y esperar hasta que la aplicación reconozca los elementos presentes y, por último, reproducirá la canción. También tiene la opción de ayudar a entrenar el modelo como se muestra en la ilustración 3, seleccionando cada elemento musical y clasificándolo. Aunque la

¹⁷ MuseAmi Inc. (Enero de 2015). iTunes. Obtenido de <https://itunes.apple.com/es/app/musicpal-snap-hear-play/id976261614#?platform=ipad>

aplicación tiene una bastante utilidad, esta tiene una calificación de 2.5 en iTunes por el tiempo que tarda en reconocer una partitura y que en ocasiones no reconoce la totalidad de elementos musicales dañando así la melodía.¹⁸

2.3.3. SHEETVISION

Es un repositorio en Github el cual contiene un algoritmo que utiliza filtrado de imágenes, escalamiento de plantillas, un umbral clasificador e identificación de notas en secuencia, que se encarga de convertir partituras musicales (que estén en un formato que sea legible por computadoras) en archivos MIDI para poder reproducirlo (Pratt 2018). Este algoritmo fue desarrollado en Python 3.6 y utiliza la técnica OCR (Reconocimiento óptico de caracteres) el cual es una aplicación de la visión por computadora que se utiliza para agilizar el procedimiento de reconocer elementos en imágenes a primera vista. Esta técnica, se divide en 4 etapas, pre procesamiento, segmentación, extracción de características y un reconocimiento, en este caso, basado en redes neuronales¹⁹

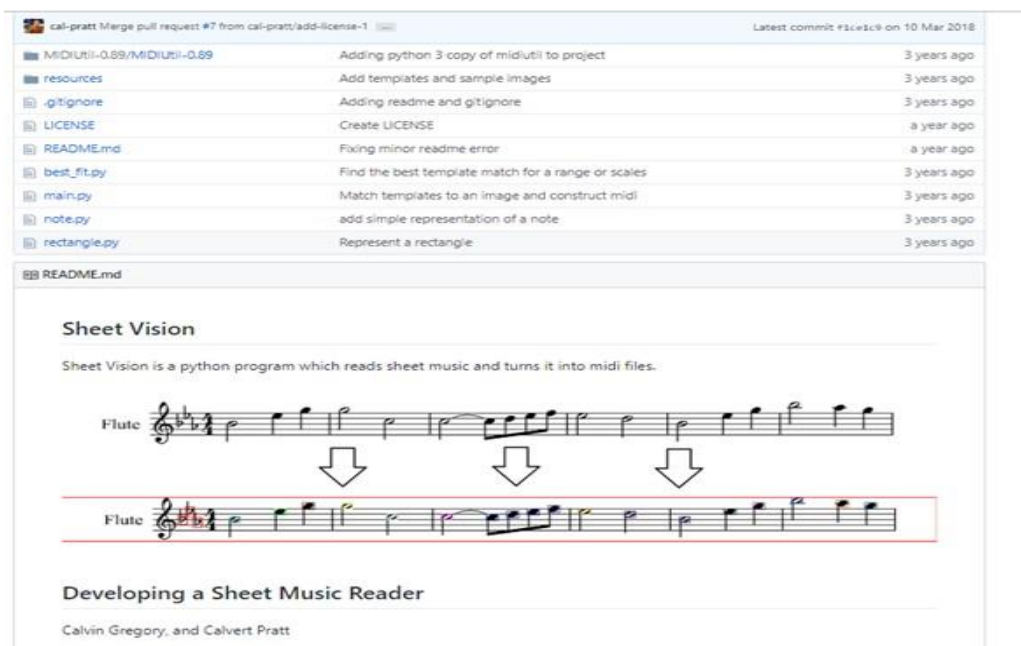


Ilustración 9: SheetVision¹²

¹⁸ Musitek. (17 de 6 de 2014). Google Play. Obtenido de https://play.google.com/store/apps/details?id=com.musitek.notereader&hl=es_C

¹⁹ Sánchez Fernández, C. J., & Sandonís Consuegra, V. (2008). Universidad Carlos III De Madrid. Obtenido de Reconocimiento Optico de Caracteres(OCR): <http://www.it.uc3m.es/jvillena/irc/practicas/08-09/09.pdf>

2.3.4. Imágenes 2D

Las dimensiones definen el espacio en que un objeto puede existir. Imagina una línea muy fina, que tiene sólo una dimensión, longitud. Si agregas una segunda dimensión de ancho, tienes un espacio 2D o bidimensional. Las formas 2D o bidimensionales tienen longitud y ancho, pero no profundidad. Son planas, como un cómic o un dibujo. Los círculos, cuadrados, triángulos, rectángulos —estas son todas figuras de dos dimensiones.²⁰

2.3.5. Aprendizaje automático

Es el sub-campo de las ciencias de la computación y una rama de la inteligencia artificial, cuyo objetivo es desarrollar técnicas que permitan que las computadoras aprendan. Se dice que un agente aprende cuando su desempeño mejora con la experiencia; es decir, cuando la habilidad no estaba presente en su genotipo o rasgos de nacimiento. De forma más concreta, los investigadores del aprendizaje de máquinas buscan algoritmos y heurísticas para convertir muestras de datos en programas de computadora, sin tener que escribir los últimos explícitamente. Los modelos o programas resultantes deben ser capaces de generalizar comportamientos e inferencias para un conjunto más amplio de datos²¹

2.3.6. Inteligencia artificial

Es un conjunto de algoritmos de aprendizaje automático que intenta modelar abstracciones de alto nivel en datos usando arquitecturas computacionales que admiten transformaciones no lineales múltiples e iterativas de datos expresados en forma matricial o tensorial.²²

2.3.7. Aprendizaje profundo o Deep Learning

El aprendizaje profundo trata del uso de redes neuronales para mejorar cosas tales como el reconocimiento de voz, la visión por ordenador y el procesamiento del lenguaje natural.²³

²⁰ Edwards, C. (s.f.). Techlandia. Obtenido de https://techlandia.com/3d-2d-info_231195/

²¹ Rouse, M. (Enero de 2017). Aprendizaje Automático (Machine Learning). Obtenido de <https://searchdatacenter.techtarget.com/es/definicion/Aprendizaje-automatocomachine-learning>

²² Brumfiel, G. (20 de Febrero de 2014). Deep Learning: Teaching Computers To Tell Things Apart. Obtenido de

<https://www.npr.org/sections/alltechconsidered/2014/02/20/280232074/deeplearning-teaching-computers-to-tell-things-apart>

²³ Regalado, A. (2014). Is google Cornering the Market on Deep Learning? Obtenido de <http://www.technologyreview.com/news/524026/is-google-cornering-the-market-ondeep->

2.3.8. Redes neuronales

El aprendizaje profundo trata del uso de redes neuronales para mejorar cosas tales como el reconocimiento de voz, la visión por ordenador y el procesamiento del lenguaje natural.²⁴

2.3.9. Aprendizaje supervisado

La primera modalidad de aprendizaje que tiene el machine learning es la de aprendizaje supervisado. Usándola, se entrena al algoritmo otorgándole las preguntas, denominadas características, y las respuestas, denominadas etiquetas. Esto se hace con la finalidad de que el algoritmo las combine y pueda hacer predicciones.²⁵

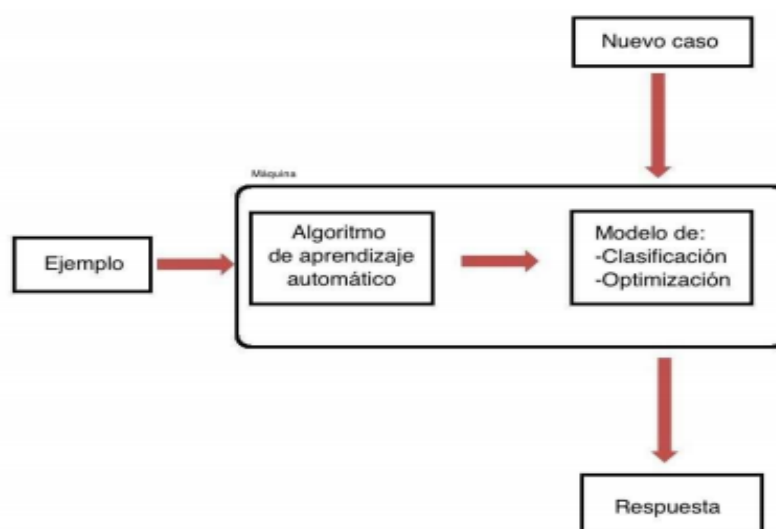


Ilustración 10 :Aprendizaje supervisado

2.4. MARCO JURÍDICO O NORMATIVO

LEY 397 de 1997 Artículo 1: Esta norma permite argumentar y justificar el proyecto, ya que respalda los procesos, proyectos y actividades en un marco de reconocimiento y respeto por la diversidad y variedad cultural de la Nación Colombiana, impulsa la investigación en áreas musicales y culturales.

learning

²⁴ Rouse, M. (Agosto de 2019). Artificial Neural Network. Obtenido de <https://searchenterpriseai.techtarget.com/definition/neural-network>

²⁵ Zambrano, J. (30 de Marzo de 2018). ¿Aprendizaje supervisado o no supervisado? Conoce sus diferencias dentro del machine learning y la automatización inteligente. Obtenido de <https://medium.com/@juanzambrano/aprendizajesupervisado-o-no-supervisado-39ccf1fd6e7b>

“10. El Estado garantizará la libre investigación y fomentará el talento investigativo dentro de los parámetros de calidad, rigor y coherencia académica.” (Ley General de Cultura, 1997)

Al compararse el algoritmo con otros medios o programas, se busca evitar plagios o dañar e interferir en propiedades intelectuales, además al utilizar partituras con diferentes fuentes, se debe mantener la integridad de los derechos de autor.

“1. Por cualquier medio o procedimiento, reproduzca una obra de carácter literario, científico, artístico o cinematográfico, fonograma, videograma, soporte lógico o programa de ordenador, o, quien transporte, almacene, conserve, distribuya, importe, venda, ofrezca, adquiera para la venta o distribución, o suministre a cualquier título dichas reproducciones.”

2.5. MARCO GEOGRÁFICO

El área geográfica en la cual se realizará la investigación es en la ciudad de Bogotá, Distrito Capital de Colombia. El desarrollo del trabajo y pruebas se realizará en una única ubicación:

Universidad Católica de Colombia

La primera ubicación es exactamente en la Dg. 46a #15-10 en donde se realizarán las pruebas del algoritmo y el desarrollo del mismo, además de toda la documentación y manuales necesarios.



Ubicación de la Universidad Católica de Colombia

(Google Maps, s.f.),²⁶

3. METODOLOGÍA

²⁶ Google Maps. (s.f.). Obtenido de [google.com/maps/place/Universidad+Católica+de+Colombia+Sede+El+Claustro/@4.6297126,-74.0790761,15z/data=!4m8!1m2!2m1!1sunivversidad+catolica!3m4!1s0x8e3f9a2e545a44b1:0x4cf0b8f62e304521!8m2!3d4.634626!4d-74.0689315](https://www.google.com/maps/place/Universidad+Católica+de+Colombia+Sede+El+Claustro/@4.6297126,-74.0790761,15z/data=!4m8!1m2!2m1!1sunivversidad+catolica!3m4!1s0x8e3f9a2e545a44b1:0x4cf0b8f62e304521!8m2!3d4.634626!4d-74.0689315)

En el procedimiento de identificación de los elementos musicales presentes en imágenes de partituras, es necesario la realización de 5 etapas como se puede ver en la ilustración 10.

3.1. FASES DEL PROYECTO

- a. Se reunirá un conjunto de imágenes 2D que contengan partituras musicales, dando origen al **Conjunto de datos**. El cual será usado como entrada y punto de partida del experimento.

- b. Una vez obtenido el conjunto de datos, se requiere agilizar el proceso de identificación utilizando un **Pre- procesamiento** de las imágenes reunidas anteriormente, aplicando una reducción a las dimensiones de la imagen.

- c. Teniendo como entrada las imágenes del conjunto de datos ya pre procesadas, la aplicación de una Faster-CNN(Faster-Convolutional Neural Network) realiza un **Procesamiento** a las imágenes lo cual tendrá como resultado la determinación de en qué lugares de la imagen se pueden ubicar los elementos musicales, haciendo uso de los siguientes subprocesos:
 - La Faster-CNN en su etapa inicial se encarga de realizar un mapeo de las características principales encontradas en la imagen haciendo uso de una **Capa Convolutiva** como se muestra en la Ilustración 10 en la sección de procesamiento.
 - El segundo subproceso dividirá el conjunto de datos en un subconjunto encargado del entrenamiento del modelo y un segundo subconjunto que se encargará de probar el modelo, dando como origen a un **Muestreo** de los datos.
 - La implementación de una **RPN** (Region Proposal Network) mostrará un conjunto de **Regiones Propuestas** que contiene lugares en donde posiblemente existan caracteres de interés en este caso, elementos musicales.
 - Una vez reunido el conjunto de regiones propuestas, para producir la salida de las Faster-CNN se aplica una **Agrupación ROI** que consta en la agrupación de las zonas de interés anteriormente identificadas

- d. Como primera salida la **Detección de Elementos Musicales** se obtiene por medio de una imagen con cajas delimitadoras, que se usan para mostrar los lugares en donde se encuentran los elementos musicales.

- e. Como etapa final, se utiliza un **Reporte de Rendimiento** basado en técnicas de medidas de desempeño que buscarán evaluar que tan eficaz es el Algoritmo

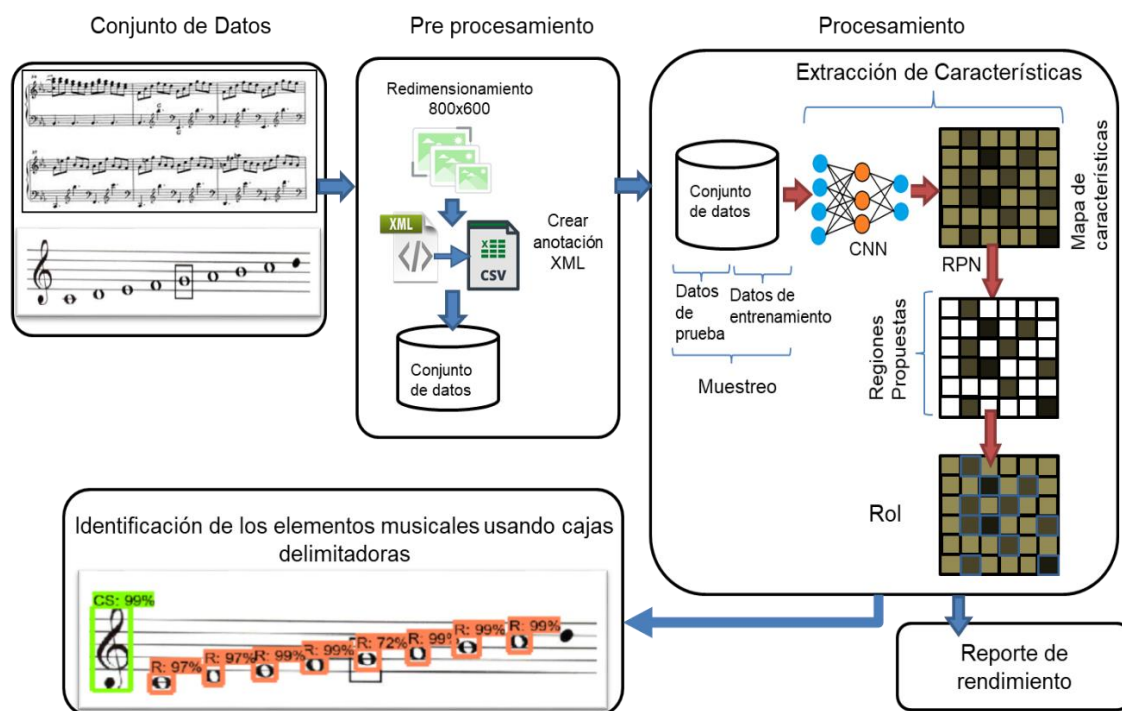


Ilustración 11 Metodología
Los autores

4. CRONOGRAMA DE ACTIVIDADES

En esta sección se mostrarán los cronogramas de actividades del anteproyecto y cada etapa fue realizada basado en el diagrama de Gantt durante los meses de febrero y marzo, posterior a esto se han venido haciendo correcciones y modificaciones.

Actividad	Fecha Inicio	Fecha Final
ANTEPROYECTO		
Definir tematica y titulo del proyecto		
Redactar Introduccion	10/02/2019	14/02/2019
Realizar el planteamiento de problema	15/02/2019	19/02/2019
Formular problemática y pregunta problema	20/02/2019	24/02/2019
Justificacion	25/02/2019	27/02/2019
Definir Objetivos	02/03/2019	05/03/2019
Objetivo general	02/03/2019	03/03/2019
Objetivos específicos	03/03/2019	05/03/2019
Redactar Marcos	05/03/2015	14/03/2015
Marco de referencia	05/03/2015	06/03/2015
Marco teórico	07/03/2015	10/03/2015
Marco conceptual	11/03/2015	14/03/2015
Definir alcances y limitaciones	15/03/2019	19/03/2019
Plantear metodología	20/03/2019	25/03/2019
Analizar costos	26/03/2019	26/03/2019
Precisar productos a entregar-entregables	27/03/2019	31/03/2019
Entrega y correcciones	01/04/2019	03/04/2019
Institucionalización	04/04/2019	04/04/2019
Aval de director de trabajo de grado	31/05/2019	31/05/2019
Entrega de Anteproyecto	03/06/2019	03/06/2019
Primera Revisión y concepto de jurados	06/06/2019	12/06/2019
Ajuste de propuesta	13/06/2019	17/06/2019
Aval de director de trabajo de grado	18/06/2019	19/06/2019
Segunda Revisión y concepto de jurados	21/06/2019	29/06/2019
Aprobacion o rechazo de la propuesta	05/07/2019 -	
PROYECTO		
Creacion de Conjunto de Datos	12/06/2019	09/07/2019
Recoleccion de partituras Musicales	12/06/2019	25/06/2019
Escaneo de Partituras Musicales	25/06/2019	27/06/2019
Creacion de repositorio	27/06/2019	09/07/2019
Realizar Pre Procesamiento de las imágenes	10/07/2019	16/07/2019
Redimensionamiento de las imágenes	10/07/2019	16/07/2019
Realizar Procesamiento de las imágenes	16/07/2019	26/09/2019
Aplicación de CNN	16/07/2019	13/08/2019
Extraccion de características	13/08/2019	27/08/2019
Muestreo	27/08/2019	30/08/2019
Generar regiones propuestas	30/08/2019	10/09/2019
Agrupacion de regiones de interes	10/09/2019	24/09/2019
Generar Reporte de rendimiento	24/09/2019	26/09/2019
Deteccion de elementos musicales	24/09/2019	26/09/2019
Pruebas y correcciones	26/09/2019	01/10/2019
Entrega	01/10/2019	11/10/2019

5. PRESUPUESTO

15. Recursos Vs. Precio		
Recurso	Cantidad	Precio
Computador	1 Computador	2800000
Recurso Humano	1 Persona	50000 Hora por persona
Escáner	1 Escáner	800000
Transmilenio	Día	4600 Por integrante
Energía Eléctrica	Día	1500 Por integrante
Internet	Día	2500 Por Integrante
Imprevisto	Por Ocurrencia	25000

*Tabla 1 Recurso Vs Precio
Fuente: Los autores*

Actividad Vs. Recurso					
Actividad	Recurso	Cantidad Recurso	Tiempo por unidad de recurso	Tiempo total	Costo Total
Construcción de conjunto de datos.	Recurso Humano	2 Personas	40 Horas	80 Horas	4000000
	Computador	2 Unidades			
	Escáner	1 Unidad			
Pre procesamiento de Imágenes	Recurso Humano	2 Personas	9 Horas	18 Horas	900000
	Computador	2 Unidades			
Procesamiento de Imágenes	Recurso Humano	2 Personas	108 Horas	216 Horas	10800000

	Computador	2 Unidades			
Generar Reporte de Rendimiento	Recurso Humano	2 Personas	7 Horas	14 Horas	700000
	Computador	2 Unidades			
Detectar elementos Musicales	Recurso Humano	2 Personas	7 Horas	14 Horas	700000
	Computador	1 Unidad			
Pruebas y Correcciones	Recurso Humano	2 Personas	4 Horas	7 Horas	350000
	Computador	2 Unidades			
Transporte	Transmilenio	484 Pasajes		121 Días	1161600
Imprevistos	Varios	3 Ocurrencias			75000
Recurso fijo	Computador	2 Computadores			5600000
	Escáner	1 Escáner			800000
Total					25086600

Tabla 2 Actividad Vs Recurso
Fuente: Los autores

Servicios			
Nombre del servicio	Cantidad	Costo Por Mes	Costo Total
Energía Eléctrica	4 Meses	63000	252000
Internet	4 Meses	75000	300000
Total			552000

Tabla 3 Servicios
Fuente: Los autores

El costo total del proyecto está dado por la suma de los costos de la tabla Actividad Vs. Recurso “Tabla 3 Actividad Vs Recurso” y el total de la tabla de servicios “Tabla 4 Servicios” dando como resultado de 25638600 \$.

6. INSTRUMENTOS O HERRAMIENTAS UTILIZADAS

Es un conjunto de algoritmos de aprendizaje automático que intenta modelar abstracciones de alto nivel en datos usando arquitecturas computacionales que admiten

6.1.1. Tensorflow

Es una biblioteca de código abierto para aprendizaje automático a través de un rango de tareas, y desarrollado por Google para satisfacer sus necesidades de sistemas capaces de construir y entrenar redes neuronales para detectar y descifrar patrones y correlaciones, análogos al aprendizaje y razonamiento usados por las personas.²⁷

6.1.2. Labellmg

Esta herramienta permite realizar el proceso de generar y colocar automáticamente texto descriptivo para las diferentes clases de los objetos dentro de la partitura por medio de etiquetado, su instalación se encuentra en <https://tzutalin.github.io/labellmg/>

²⁷ Definición obtenida de <https://es.wikipedia.org/wiki/TensorFlow>

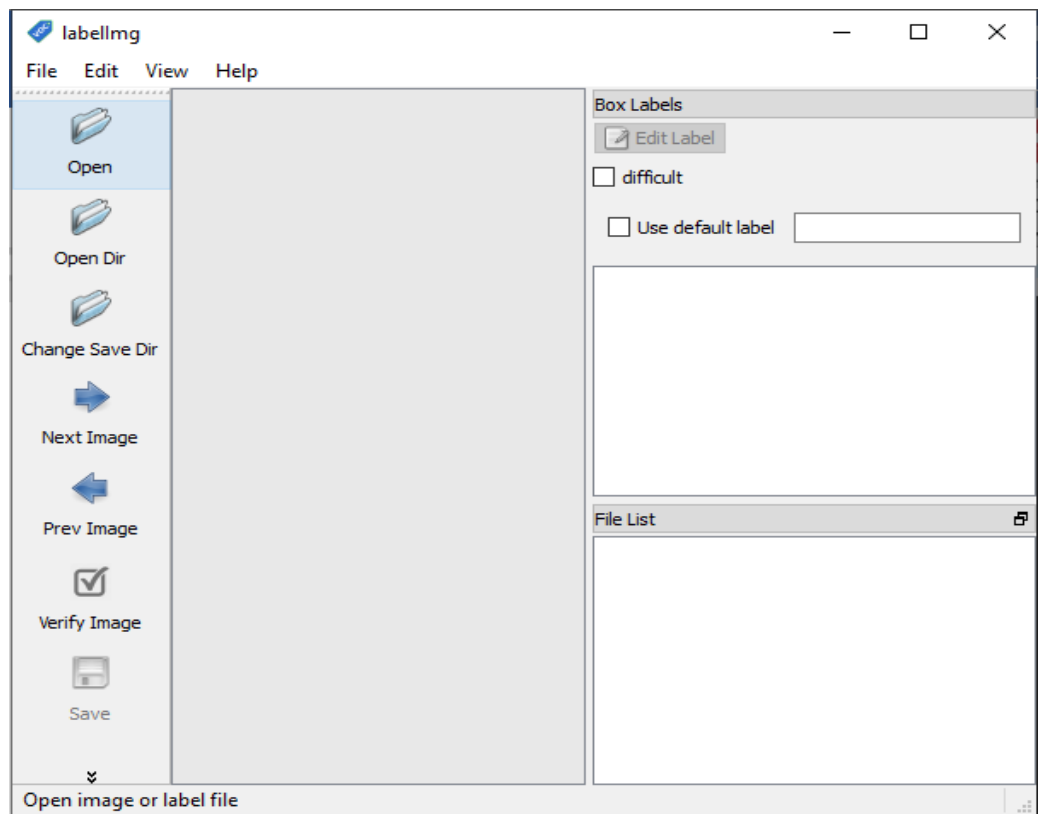


Imagen 8. Interface Labellmg

6.1.3. GPU NVIDIA GEFORCE GTX 1050

Esta unidad de procesamiento gráfico o coprocesador dedicado al procesamiento de gráficos u operaciones de coma flotante, para aligerar la carga de trabajo del procesador central permite manejar grandes volúmenes de imágenes y gráficos con altas resoluciones, buscando así mejores y más rápidos análisis y resultados.

6.1.4. Python 3.6

Se utilizó y se trabajó sobre este lenguaje de programación, buscando una sintaxis que favorece al código legible, además que se trata de un lenguaje de programación multiparadigma que soporta orientación a objetos y programación imperativa, lo que permite que el desarrollo del algoritmo utilizara API y herramientas adjuntas al mismo.

6.1.5. COCO

Es un conjunto de datos de detección, segmentación y etiquetado de objetos a gran escala. Que tiene dentro de sus características las más importantes que fueron utilizadas, como segmentación de objetos y Reconocimiento en contexto

6.1.6. Faster-RCNN Inception V2

La arquitectura Faster RCNN es una arquitectura usada para la detección y clasificación de objetos dentro de imágenes basada en la propuesta de regiones por medio de RPN(red de regiones propuestas).

Este modelo de arquitectura consta de dos etapas, la extracción de características y la predicción y agrupación de las regiones propuestas.

La **extracción de características** que utiliza la arquitectura convolucional **Inception V2** usa filtros que realizan operaciones de convolución mientras escanea la entrada con respecto a sus dimensiones dando como resultado un mapeo de características convolucional.

La predicción y agrupación de regiones propuestas está basada en la **RPN** (Red de regiones propuestas) la cual tiene como objetivo predecir en que partes de la imagen hay un objeto. Para la etapa del entrenamiento de la RPN se tiene en cuenta la siguiente función de pérdida:

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

Esta función iterativa sobre los puntos de anclaje consta de dos términos, el primero, una función de pérdida de la capa clasificadora, que depende de un parámetro p_i que es el valor predicho de si existe o no un objeto en un punto de anclaje y p_i^* que es el valor real de si existe o no, el segundo término consta de la multiplicación de la tasa de aprendizaje λ por la existencia de un objeto, por la función de pérdida de las regiones, que dependen de los valores reales y los predichos del modelo respectivamente. El Faster-RCNN es ahora un modelo canónico para la detección de objetos basada en el aprendizaje profundo y consiste en 3 pasos simples:

1. Escanee la imagen de entrada en busca de posibles objetos utilizando un algoritmo llamado Búsqueda selectiva, que genera ~ 2000 propuestas de región
2. Ejecute una red neuronal convolucional (CNN) sobre cada una de estas propuestas regionales²⁸

²⁸ Obtenido de <https://towardsdatascience.com/deep-learning-for-object-detection-a-comprehensive-review-73930816d8d9>

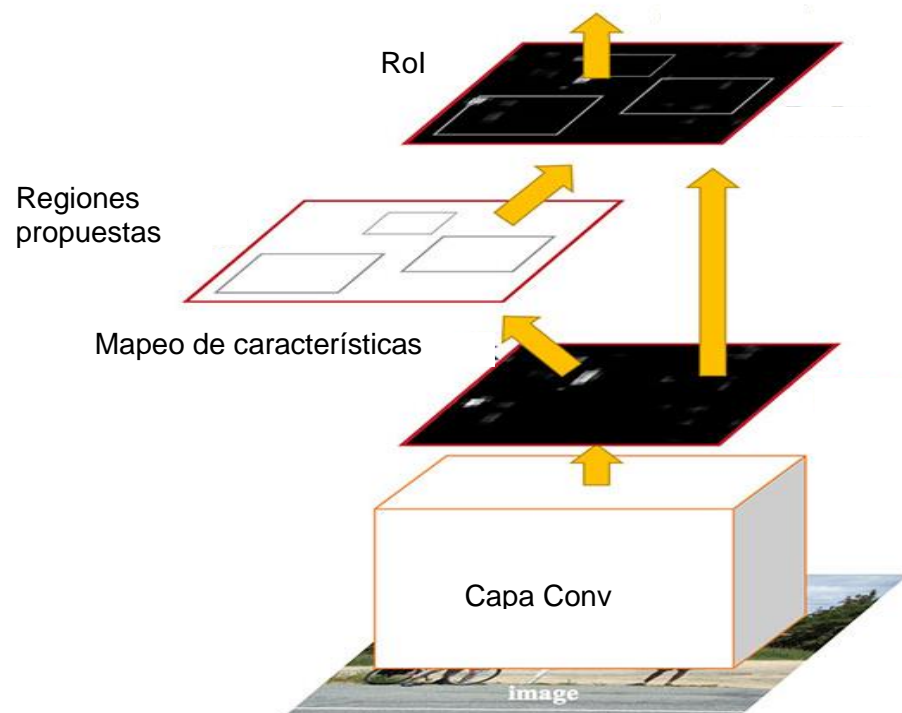


Ilustración 12 arquitectura Faster RCNN

6.1.7. Github

Es una forja (plataforma de desarrollo colaborativo) para alojar proyectos utilizando el sistema de control de versiones Git. Se utiliza principalmente para la creación de código fuente de programas de ordenador ²⁹ Por medio de esta herramienta es posible la portabilidad y manejo del proyecto.

7. RECOLECCIÓN DE DATOS

La recolección de datos se hace mediante la obtención de imágenes de partituras musicales que sean legibles. Se obtienen partituras del himnario adventista del séptimo día, las cuales se encuentran en formato pdf, del himnario Evangélico luterano, las cuales son escaneadas o fotografiadas y del banco virtual de partituras que ofrece el ministerio de cultura. También se buscar imágenes que contengan elementos musicales aislados.

²⁹ Obtenido de <https://es.wikipedia.org/wiki/GitHub>



Ilustración 13 Imágenes del Conjunto de datos

7.1.1. TIPOS DE DATOS

Los datos están conformados por Imágenes 2D de partituras musicales. Se tiene en cuenta que a las dimensiones de las imágenes recolectadas se les debe aplicar un redimensionamiento de su tamaño original a 800X600 para hacer una estandarización de tamaños y una conversión de tipo de la imagen exclusivamente a jpg. Se hará las respectivas anotaciones de las clases encontradas dentro de la imagen ya redimensionada y convertida a jpg,

generando así un archivo XML por cada imagen y se procede a hacer conversión de estas anotaciones a un formato csv el cual arroja un total de 2093 anotaciones.

Los resultados obtenidos de los diferentes experimentos, son de tipo cuantitativo los cuales están plasmados en una matriz descriptiva que busca mostrar los resultados globales por medio de porcentajes.

8. DESARROLLO DE LA PROPUESTA

En esta sección se mostrarán los pasos necesarios para crear un detector de objetos haciendo uso de la API de Tensorflow.

Para la implementación del modelo se requiere clonar el modelo `Identificacion-de-Figuras-musicales-usando-faster-rcnn.git` mediante el comando

```
git clone Identificacion-de-Figuras-musicales-usando-faster-rcnn.git
```

Los pasos necesarios para ejecutar el proyecto son:

- Etiquetado de imágenes.
- Seleccionar la proporción del conjunto de entrenamiento y conjunto de validación.
- Conversión de anotaciones XML a CSV.
- Conversión de archivos CSV a TFRecord.
- Configuraciones de entrenamiento.
- Entrenar modelo.
- Exportar modelo de inferencia.
- Evaluar el modelo.

Etiquetado de imágenes:

Para hacer el etiquetado de imágenes se procede a reunir un conjunto de datos el cual contenga figuras musicales, las dimensiones de las imágenes deben ser de 800x600 únicamente.

Se ha de tener en cuenta que existen 15 clases:

- CS=Clave de Sol
- CF=Clave de fa
- N=Negra
- B=Blanca
- C=Corchea
- F=Fusa
- SF=SemiFusa

- SC=Semicorchea
- SN=Silencio
- SB=Silencio de Blanca
- SR=Silencio de Redonda
- P=Puntillo
- R=Redonda
- Bem=Bemol
- SOS= Sostenido

Por medio de la herramienta Labelimg(<https://github.com/tzutalin/labelimg>) se procede a hacer el etiquetado encerrando cada nota en un recuadro el cual la relacionará con su respectiva clase.



Ilustración 14 Etiquetado de Imágenes
Los autores

Esto generará por imagen un archivo de extensión XML el cual contiene las anotaciones de cada imagen.

División de conjunto de datos:

Una vez clonado el repositorio en la carpeta /object_detection/images/ se procede a hacer la división del conjunto de datos en entrenamiento y validación y se ubican en las imágenes y los XML en su respectiva carpeta de train y test.

Conversión de XML a CSV:

Dentro de la carpeta /object_detection/images/se ejecutan los dos archivos xml_to_csv_train.py xml_to_csv_test.py esto generará un archivo csv para cada uno de los conjuntos en la misma carpeta que contienen las etiquetas filename, width, height, class,xmin,ymin,xmax,ymax de cada anotación.

Conversión de archivos CSV a TFRecord:

Una vez convertidos en csv las anotaciones para cada conjunto de datos, se procede a convertir los datos en un formato el cual el “API de tensorflow para la detección de objetos” es capaz de interpretar. Mediante el archivo “generate_tfrecord.py” encontrado en la carpeta object_detection el cual contiene el nombre de las clases y devuelve un valor en específico por cada una de ellas:

```
def class_text_to_int(row_label):
    if row_label == 'CS':
        return 1
    elif row_label == 'CF':
        return 2
    elif row_label == 'N':
        return 3
    elif row_label == 'B':
        return 4
    elif row_label == 'C':
        return 5
    elif row_label == 'F':
        return 6
    elif row_label == 'SC':
        return 7
    elif row_label == 'SF':
        return 8
    elif row_label == 'SB':
        return 9
    elif row_label == 'SR':
        return 10
    elif row_label == 'P':
        return 11
    elif row_label == 'SN':
        return 12
    elif row_label == 'R':
        return 13
    elif row_label == 'Bem':
        return 14
    elif row_label == 'SOS':
```

```
    return 15
else:
    None
```

Mediante los comandos:

```
python generate_tfrecord.py --csv_input=images\train_labels.csv
--image_dir=images\train --output_path=train.record
```

```
python generate_tfrecord.py --csv_input=images\test_labels.csv -
-image_dir=images\test --output_path=test.record
```

se crearán dos archivos TFRecord para cada uno de los csv anteriormente creados.

Configuración de entrenamiento:

En la carpeta /training/ se encuentra el archivo de mapeo de características el cual debe estar idéntico al a la clase class_text_to_int del archivo generate_tfrecord.py de la siguiente manera:

```
item {
  id: 1
  name: 'CS'
}
item {
  id: 2
  name: 'CF'
}
item {
  id: 3
  name: 'N'
}
item {
  id: 4
  name: 'B'
}
item {
  id: 5
  name: 'C'
}
item {
  id: 6
  name: 'F'
}
item {
  id: 7
  name: 'SC'
}
item {
```

```
    id: 8
    name: 'SF'
  }
  item {
    id: 9
    name: 'SB'
  }
  item {
    id: 10
    name: 'SR'
  }
  item {
    id: 11
    name: 'P'
  }
}

item {
  id: 12
  name: 'SN'
}
item {
  id: 13
  name: 'R'
}
item {
  id: 14
  name: 'Bem'
}
item {
  id: 15
  name: 'SOS'
}
```

Ahora se necesita crear un archivo de configuración que se puede descargar del repositorio de github (https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/detection_model_zoo.md)

Se utilizará el modelo que se encuentra en la carpeta /object_detection/samples/ llamado:

- faster_rcnn_inception_v2_pets.config

y se editaran las líneas:

Línea 85:

Batch_size: 2

Línea 106:

```
fine_tune_checkpoint: "<DIRECCION> /models-  
master/research/object_detection/faster_rcnn_nas_coco_2018_01_28  
/model.ckpt"
```

Línea 113

Se cambia el número de pasos a conveniencia por defecto se dejan en 20000.

Línea 123:

Se selecciona el directorio en donde se guardó el archivo train.record

```
input_path: "<DIRECCION>/models-  
master/research/object_detection/train.record"
```

Línea 125 y 137:

```
label_map_path:  
"C:/Tesis/modelsaster/research/object_detection/training/labelma  
p.pbtxt"
```

Línea 135:

Se selecciona el directorio en donde se guardó el archivo test.record

```
input_path: "C:/Tesis/models-  
master/research/object_detection/test.record"
```

Entrenamiento del modelo:

El entrenamiento del modelo cuenta con dos etapas, la extracción de características y la el entreno de la RPN con la unión de las regiones de interés. Se tiene como entrada una imagen de la cual se hará una extracción de características usan la arquitectura Inception V2 de la cual se extraerá un mapa de las características encontradas dentro de la imagen como se puede ver en la ilustración 10.

A continuación, una ventana deslizante de 4x4 px (debido al tamaño de los objetos en la imagen) teniendo como referencia los puntos de anclaje generados en el mapeo de características, una vez extraída la proporción de la imagen, se proponen cajas delimitadoras con diferentes dimensiones las cuales serán usadas para determinar el puntaje de IOU posteriormente. Luego una capa de clasificación se encarga de decidir si existe o no de acuerdo a que si el umbral del IOU es mayor a 0.5 se determina que existe una figura musical dentro de este punto de anclaje, de no existir, simplemente se pasa por alto este punto de anclaje y de si existir el objeto intervendrá una capa de regiones que se encarga de medir que tan exacta es la predicción sobre el objeto real

usando la métrica IOU.

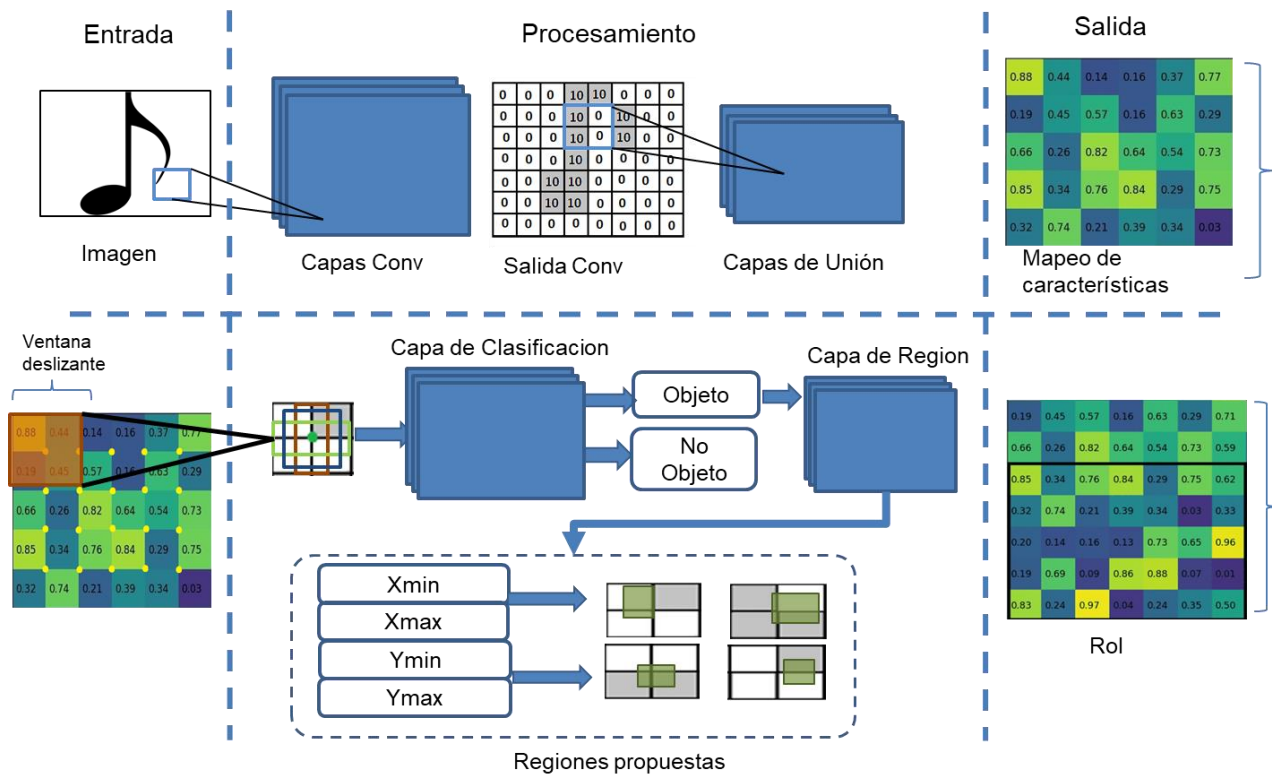


Ilustración 15 Proceso de entrenamiento

Fuente: Autores.

Para entrenar el modelo se utiliza el comando:

```
python train.py --logtostderr --train_dir=training/ --
pipeline_config_path=training/faster_rcnn_inception_v2_pets.conf
ig
```

y después de 5 minutos en otro cmd se ejecuta el comando:

```
tensorboard --logdir=training
```

el modelo se entrenará durante los pasos que se ha configurado en el archivo .config, pero se detendrán a conveniencia cuando la función de pérdida de "Total_Loss" sea menor que 0.1 y se procede a exportar el modelo de inferencia mediante el comando:

```
python export_inference_graph.py --input_type image_tensor --
pipeline_config_path
training/faster_rcnn_inception_v2_pets.config --
```

```
trained_checkpoint_prefix training/model.ckpt-XXXX --
output_directory inference_graph
```

en el comando anterior, en donde se encuentran “XXXX” se coloca el número mayor encontrado en la carpeta training

Evaluación del modelo:

Para hacer una validación utilizando la métrica de mAP que son usadas por “COCO DATASET” como las medidas de AP(Average Precision) y AR(Average Recall). Se hace uso del archivo eval ubicado en la carpeta legacy del proyecto, se copia y se pega en la carpeta de object_detection y se ejecuta el comando:

```
python eval.py --logtostderr --
pipeline_config_path=training/faster_rcnn_inception_v2_pets.conf
ig --checkpoint_dir=training/ --eval_dir=eval/
```

```
DONE (t=0.105).
Average Precision (AP) @[ IoU=0.50:0.95 | area= all | maxDets=100 ] = 0.648
Average Precision (AP) @[ IoU=0.50 | area= all | maxDets=100 ] = 0.846
Average Precision (AP) @[ IoU=0.75 | area= all | maxDets=100 ] = 0.770
Average Precision (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.733
Average Precision (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.735
Average Precision (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.250
Average Recall (AR) @[ IoU=0.50:0.95 | area= all | maxDets= 1 ] = 0.317
Average Recall (AR) @[ IoU=0.50:0.95 | area= all | maxDets= 10 ] = 0.689
Average Recall (AR) @[ IoU=0.50:0.95 | area= all | maxDets=100 ] = 0.694
Average Recall (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.771
Average Recall (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.772
Average Recall (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.315
```

Ilustración 16 Medidas de desempeño CMD

Los Autores

para poder tener una visualización de los resultados se hace el uso del comando de tensorboard en su modelo de evaluación así:

```
tensorboard --logdir=eval
```

Por ultimo obtendrá un resultado similar al siguiente:

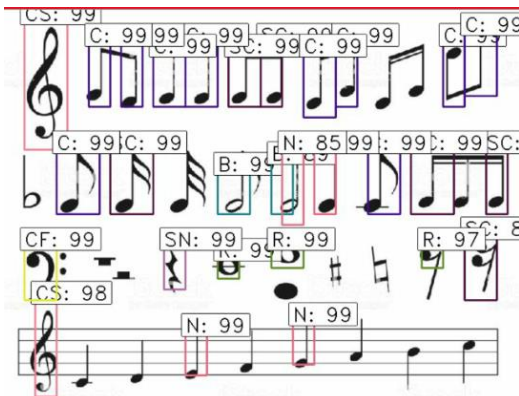


Ilustración 17 Resultado de la identificación
Los autores

9. RESULTADOS

Mediante el uso del API de Tensorflow para la detección de objetos, se hace la implementación del modelo Faster RCNN Inception V2, dividiendo el conjunto de datos en conjunto de entrenamiento y prueba con las siguientes proporciones de manera aleatoria:

- Experimento 1: 80% Conjunto de datos de entrenamiento y 20% Conjunto de datos de prueba.
- Experimento 2: 70% Conjunto de datos de entrenamiento y 30% Conjunto de datos de prueba.
- Experimento 3: 75% Conjunto de datos de entrenamiento y 25% Conjunto de datos de prueba.

Tras colocar el modelo en entrenamiento durante 200000 pasos haciendo uso de una aceleradora grafica Nvidia Geforce GTX 1050, y con una tasa de aprendizaje de 0.0003, la función de pérdida total da convergencias en 0.1. Se procede a analizar los resultados por medio de las pruebas del modelo de cada uno de los experimentos.

REPORTE DE RENDIMIENTO DEL ALGORITMO FASTER-RCNN INCEPTION							
# EXP	Muestreo	AP[0.50:0.95]	AP50	AP75	AR[0.50:0.95]	AR50	AR75
1	80-20	0,64	0,885	0,718	0,659	0,678	0,503
2	70-30	0,648	0,846	0,77	0,317	0,689	0,694
3	75-25	0,481	0,678	0,535	0,219	0,489	0,297

Tabla 4 Comparación de experimentos
Fuente: Autores

En la tabla de resultados se utilizan métricas de AP (Average precision) y AR (Average Recall) divididas por 3 intervalos de certeza o rigurosidad de las pruebas llamados IoU que nos indica el área de superposición entre dos cuadros delimitadores donde se muestra que si la predicción es perfecta el IoU=1 y si la predicción se pierde totalmente el IoU sería igual 0. De acuerdo a las se dividen en IoU= [0.50-0.95] IoU=0.50 y IoU= 0.75 para cada métrica según las métricas de COCO.

Se puede apreciar en la tabla que el experimento 2 el cual utilizo el muestreo de 70%-30% obtuvo el mejor puntaje en una precisión media con margen de rigurosidad (IoU) del 0.75 obteniendo una precisión del 71.8% y un AR75 del 69.4%.

En la evaluación del modelo se evidencia una alta identificación en las figuras musicales que no están dentro del pentagrama, mientras que cuando están en él y las figuras son muy pequeñas, presenta dificultades al momento de identificarlas, pero se puede evidenciar que, en cada una de las clases o categorías, el porcentaje de coincidencia entre la categoría y la clase realmente identificada es bastante alta para cada uno del elemento identificados.

A continuación, se presentan ejemplos de las salidas del algoritmo.

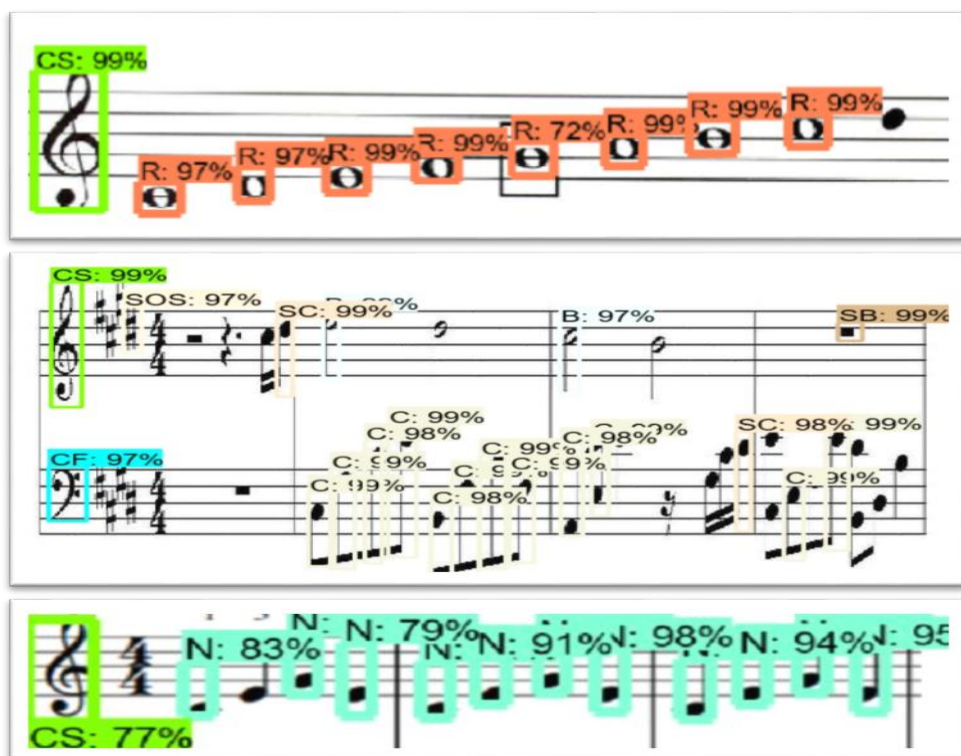


Ilustración 18 Salida del Algoritmo
Los autores

10. CONCLUSIONES

La identificación de figuras musicales utilizando la metodología propuesta, la cual es un modelo enfocado a la predicción de regiones por medio de la RPN, presenta buenos resultados al momento de identificar las figuras musicales.

Durante el desarrollo del proyecto se decidió manejar 15 clases de figuras, debido a las diferentes características que poseen los elementos y los inconvenientes que se generan al momento de quererlas agrupar en una sola.

En la etapa de construcción del conjunto de datos se concluye que deben usarse imágenes con gran variedad de elementos y con alto nivel de legibilidad para no tener problemas en el momento de la extracción de características.

Se tuvo en cuenta el redimensionamiento de las imágenes que evitaron la pérdida de características y se definió una dimensión estándar (800 px X 600 px) para poder entrenar el modelo.

A lo largo del desarrollo del proyecto se realizaron 3 experimentos, en los cuales varía la proporción del conjunto de datos de entrenamiento y validación, se evidencia que el experimento #2 presenta una precisión de 77,7% en el índice de rigurosidad más alto de precisión (AP75), siendo éste el índice más alto de los tres experimentos, cumpliendo con una proporción de 70% de entrenamiento y 30% de prueba, lo cual significa que el modelo utilizado para este proyecto proporciona un alto desempeño en su tarea de identificación de figuras musicales y adicionalmente realiza una clasificación de dichos elementos.

11. RECOMENDACIONES

Se recomienda hacer el uso de modelos orientados a la predicción de píxeles que posiblemente mejorarán el desempeño en cuanto a la detección de las diferentes categorías en el pentagrama.

El conjunto de datos debe contener imágenes con figuras musicales individuales y figuras plasmadas en el pentagrama, se debe contar con el tiempo necesario para poder hacer todo el etiquetado de las imágenes de forma adecuada.

Se recomienda al momento de estandarizar las imágenes en la etapa de pre-procesamiento, utilizar una dimensión de 600x800 la cual permite obtener de manera más fácil las características de la imagen.

Para la etapa de procesamiento se busca agilizar los tiempos de entrenamiento

que consume el algoritmo para aprender del conjunto de datos, se sugiere utilizar una GPU con una capacidad mayor a 4GB para tener un tiempo de entrenamiento más óptimo.

12. TRABAJOS FUTUROS

A partir del conjunto de datos creado, se puede implementar una categorización profunda que contenga mayor cantidad de elementos descriptores como tiempos y métricas de las figuras musicales respecto a la nota que cada una representa en el pentagrama.

Es posible instaurar un diseño móvil amigable para el usuario que permita su interacción con el algoritmo y facilite el etiquetado por medio de colaboraciones del usuario a través del aplicativo lo cual aportará a la construcción de un conjunto de datos más amplio y más robusto.

Por medio de esta línea base, se puede realizar implementaciones que se usen como herramienta que permitan el desarrollo musical tanto empírico como grupal en ambientes que fomenten el aprendizaje musical.

13. ESTRATEGIA DE COMUNICACIÓN Y DIVULGACIÓN

Como métodos de comunicación y divulgación se tendrán en cuenta los siguientes tres:

- La sustentación de Ante proyecto de grado ante los jurados.
- Mediante una publicación en formato digital en el repositorio RIUCaC de la Universidad Católica de Colombia en la URL <http://repository.ucatolica.edu.co/>.

14. REFERENCIAS

Mercé , Vilar i Monmany. «Educacion y desarrollo musicales desde un punto de vista cognitivo.» En Educacion y desarrollo musicales desde un punto de vista cognitivo, de L Davinson,, & L Scripp, 5-6. Barcelona: LEEME(Lista Electronica Europea de Música en la Educacion, 2004.

Carretero De la Rocha, D. (Junio de 2009). Universidad Carlos III de Madrid. Obtenido de Sistema de Reconocimiento de Partituras Musicales: <https://e-archivo.uc3m.es/handle/10016/7207>

Vasilev, Ivan, Daniel Slater, y Gianmario Spacagna. Python Deep Learning.

Birmingham: Packt Publishing Ltd., 2019

Marco, T. (Abril 2017). Analfabetos musicales. Melómano, 229.

Valverde, X. (2017). Contexto de aprendizaje musical y de transmisión cultural. Revista de Cienciassociales (CI) núm. 39, 110

Diego Calderón, J. G. (2018). Métodos de lectura musical y el aprendizaje de nuevos repertorios. Sonograma magazine num 037.

Welch, G. (Julio 2004). Psychology of music. United Kingdom: Graham Welchsempre

Gallo, Eva Maria, y María Isabel Reyzábal. «LA NOTACIÓN MUSICAL. DIFICULTADES DE LECTO-ESCRITURA EN ALUMNOS DE 2º CICLO DE E.S.O.» Red de Revistas Científicas de América Latina, el Caribe, España y Portugal, 2005: 465-485.

Gardner, H. (2011). Inteligencias múltiples, la teoría en la práctica. PAIDOS IBERICA.

MuseAmi Inc. (Enero de 2015). iTunes. Obtenido de <https://itunes.apple.com/es/app/musicpal-snap-hear-play/id976261614#?platform=ipad>

Musitek. (17 de 6 de 2014). Google Play. Obtenido de

https://play.google.com/store/apps/details?id=com.musitek.notereader&hl=es_C

Gamandé Villanueva, N. (9 de Junio de 2013). *Re-UNIR Repositorio Digital*. Obtenido de

<https://reunir.unir.net/bitstream/handle/123456789/2595/gamande%20villanueva.pdf?sequence=1&isAllowe>

Pereira Ghiena, A. (2014). *ACADEMICA*. Obtenido de

[file:///C:/Users/Juan/AppData/Local/Packages/Microsoft.MicrosoftEdge_8wekyb3d8bbwe/TempState/Downloads/Tesis_Maestria_PereiraGhiena\(final\)%20\(1\).pdf](file:///C:/Users/Juan/AppData/Local/Packages/Microsoft.MicrosoftEdge_8wekyb3d8bbwe/TempState/Downloads/Tesis_Maestria_PereiraGhiena(final)%20(1).pdf)

Sánchez Fernández, C. J., & Sandonís Consuegra, V. (2008). Universidad Carlos III De Madrid. Obtenido de Reconocimiento Optico de Caracteres(OCR): <http://www.it.uc3m.es/jvillena/irc/practicas/08-09/09.pdf>

Gutiérrez, José Antonio Taquíá. «El procesamiento de imágenes y su potencial.» Universidad de Lima, 2017

GUEVARA SANIN, JUAN SEBASTIAN. Teoria Musical. 2015.

Sosa , Roberto, y Mariela Dominguez. Conceptos Básicos del Lenguaje Musical.

San Luis: Universidad de La Punta, 2014.

Varela Cañón, Jeferson Camilo. «Reconocimiento automatico de partituras para guitarra por medio de vision artificial y redes neuronales artificiales.» Universidad La Salle, 2018.

Fujinaga, I., Moore, S., & Sullivan Jr, D. S. (1998). Implementation of exemplar- based learning model. Universidad Johns Hopkins

Calvo Zaragoza, Jorge, Francisco J. Castellanos, Gabriel Vigliensoni, y Ichiro Fujinaga. «Deep Neural Networks for Document Processing of Music Scores Images.» MDPI, 2018