

# Entrenamiento de un modelo de clasificación de bajos y guitarras con dataset de cosecha propia

Trabajo Final de Asignatura - Programación Automática

Máster en Ciencia y Tecnología Informática

2019-2020

Alejandro Rey López

# Contenidos

- 1 Introducción
- 2 Descripción del problema
- 3 Diseño de la solución
- 4 Conclusiones
- 5 Limitaciones y trabajo futuro

# Introducción

1

Motivación del trabajo e introducción al problema

# Introducción.

**Algunos instrumentos musicales son difíciles de distinguir visualmente** (e.g. la *guitarra* y el *bajo*), especialmente para personas no familiarizadas con la música.

¿Bajo o guitarra?



Guitarra



Bajo

¿Bajo o guitarra?



Bajo



Guitarra

¿Bajo o guitarra?



Bajo



Guitarra

# Descripción del problema

2

Clasificación de imágenes en bajos o guitarras

# Problema de clasificación en 2 clases:

**No es trivial extraer unas características que clasifiquen inequívocamente ambas clases de instrumentos.**

**Reglas más básicas no siempre aplican (e.g. número de cuerdas)**

**Redes convolucionales son capaces de extraer características de forma autónoma, por lo que parecen adecuadas para un problema de clasificación de este tipo.**

**Problema de clasificación con  
modelo convolucional.  
Guitarra | Bajo (CNN)**

**¿Será el modelo capaz de abstraer el concepto general de bajo y guitarra desde el punto de vista visual con gran precisión?**

# Descripción del problema: Requisitos de solución

**Datos masivos de entrada para entrenar la red.**



**Guitarras y bajos**

**Muy pocos datos de bajos en datasets públicos**



**Creación de dataset propio**

**Implementación de la red / Uso de modelo preentrenado**

**Uso de modelo implementado en PyTorch basado en práctica 1**

**Implementación y ejecución de ciclo de entrenamiento y comparación de arquitecturas e hiperparámetros.**

**Codificación de arquitectura fácilmente configurable para realizar batería de pruebas.**

**Inferencia**

# Diseño de la solución

3

*Web scraper, pre-procesado, y pruebas tras entrenamiento con diversas arquitecturas.*



# Objetivo del trabajo y metodología.

## Objetivo del trabajo

Entrenar un modelo de clasificación de imágenes de guitarras y bajos con buena tasa de aciertos basado en redes convolucionales y estudiar cómo de sencillo es este problema para un sistema computacional de este tipo.

## Metodología:

1) Creación del dataset de guitarras y bajos

*Web scraping app*

2) Pre-procesado para adecuar datos a entrada del modelo

Python & Google Collab

3) Entrenamiento, validación y test.

Python & Google Collab

\* 4) Pruebas en inferencia

Mini-juego contra modelo entrenado

# 1) Creación de *dataset* : scraping de *Thomann*



Web scraping para obtención de imágenes de productos (guitarras y bajos).

## Proceso:

1

Exploración recursiva de páginas enlazadas desde una página de categorías de instrumentos + Filtrado de lo que no son guitarras ni bajos.

Obtención de urls a páginas de *listado de productos* listas de productos con

2

Iteración sobre las páginas de resultados y extracción de thumbnails

Urls a imágenes thumbnail con etiqueta.

3

Descarga iterativa de imágenes thumbnail en carpeta “Bajos” o “Guitarras” y filtrado de thumbnails “default”

Dataset

## Resultados:

8000 imágenes (150x150)

6924 guitarras

1076 bajos



Código accesible en



## 2) Pre-procesado: adaptación de entradas

### Pre-procesado necesario:

- El modelo no admite imágenes directamente, sino la información de sus píxeles.
- Realizar la carga iterando sobre los directorios cada vez que queremos entrenar el modelo es ineficiente.
- Partimos de etiquetado no explícito.
- Conjunto no dividido

### Procesos realizados

**1** Carga y encapsulación de dataset en archivo único. (.npy)

2 numpy arrays persistentes (datos rgb, etiqueta asociada a fila/imagen)

**2** Barajado de imágenes y creación de conjuntos para entrenamiento, validación y test

**3** Adecuación de entrada al modelo. Intercambio de dimensiones de información RGB

```
training: (4500, 150, 150, 3), (4500,)
validation: (1500, 150, 150, 3), (1500,)
testing: (2000, 150, 150, 3), (2000,)
Training set made of: 3880 guitars and 620 bass
Validation set made of: 1311 guitars and 189 bass
Testing set made of: 1733 guitars and 267 bass
```

Código accesible en



### 3) Resultados: Precisión de modelos obtenidos

#### Consideraciones generales

- 1000 ciclos de entrenamiento.
- Ejecución con GPU en Google Colab.
- Random seed = 0.
- Guardado del mejor modelo generado para cada prueba realizada con una arquitectura.

hyperparams				Performance					Conv 1					
ID	batchsize	lr	Momentum	Min Loss	Best epoch	Test Accuracy(%)	Time to best (min)	Total Training time(min)	Conv #	Kernel size	out maps	batch norm	DP	Conv2
T1	512	0.005	0.6	0.38	965	86.65	22.68	23.48	CONV 1	5x5	5	NO	NO	N/A
T2	256	0.01	0.6	0.38	695	86.65	16.55	23.85	CONV 1	5x5	5	NO	NO	N/A
T3	64	0.01	0.6	0.38	448	86.65	10.98	24.45	CONV 1	5x5	5	NO	NO	N/A
T4	64	0.04	0.6	0.35	30	87.95	0.80	25.72	CONV 1	5x5	10	NO	NO	N/A
T5	256	0.01	0.6	0.07	21	98.45	0.63	28.22	CONV 1	5x5	10	YES	NO	N/A
<b>T6</b>	32	0.003	0.3	0.04	561	<b>99.25</b>	17.16	<b>30.60</b>	CONV 1	5x5	10	YES	0.5	N/A
T7	256	0.01	0.3	0.06	52	98.75	1.67	29.68	CONV 1	3x3	20	YES	NO	N/A
T8	512	0.005	0.3	0.24	19	95.3	0.44	21.55	CONV 1	3x3	5	NO	NO	N/A
T9	512	0.005	0.3	NF	NF	NF	NF	NF	CONV2	5x5	10	NO	NO	(CPU EXEC) *30 out 11 k SIN Batchnorm
T10	512	0.01	0.6	0.05	215	98.86	9.60	<b>70.21</b>	CONV2	5x5	10	YES	NO	*30 out 11 k CON Batchnorm
hyperparams				Performance										

Código accesible en 

**Mejor modelo: T6**  
**99.25%** de acierto en test.

**1 CONV, 5K, 10out,**  
**BatchNorm y Dropout**

## 4) Pruebas con inferencia:



Ver al final de notebook

**Desarrollo de carga de modelos desde archivo +  
minijuego de clasificación Humano/CNN**



What is this, a guitar (type 0) or a bass (type 1) ?

0

This image is a GUITAR:

You guessed GUITAR

CNN guessed GUITAR

KEEP IT UP! <3

```
*****  
[GAME COMPLETED] USER SCORE = 100/100 100/100  
WOW, A TIE!  
*****
```

# Conclusiones

4

Resultados obtenidos y aportaciones

# Conclusiones

## Aportaciones

1

**Se ha desarrollado un sistema de clasificación de imágenes de bajos y guitarras con alta precisión basado en redes convolucionales y con relativa flexibilidad.**

2

**Se ha desarrollado una aplicación de web scraping que puede ser fácilmente configurable para la creación de datasets más amplios y de mayor resolución.**

3

**Se ha recopilado un amplio dataset (800 imágenes) de instrumentos musicales pertenecientes a las 2 categorías.**

## Conclusión general

**CNNs adecuadas para resolver el problema de clasificación incluso con arquitecturas sencillas**

# Limitaciones y trabajo futuro

5

Lacras, defectos y direcciones de investigación futura



# Limitaciones y trabajo futuro

## Limitaciones

- **Problema de clasificación muy simple**  
(2 etiquetas únicamente)
- **¿Poca resolución de imágenes == Poca usabilidad?**
- **¿Calidad del dataset?**
  - a) No solamente aparecen guitarras ( a veces aparecen junto con fundas, amplificadores, en el thumbnails
  - b) ¿Bias? Pocas perspectivas de los instrumentos.
  - c) Descompensación del dataset
- **¿Qué características ha extraído? Explicabilidad**

## Trabajo futuro

- ***Pruebas con downscaling***
- ***Problema de clasificación multi-instrumento.***
- ***Mejora del dataset***
  - a) data augmentation (más perspectivas)
  - b) más bajos
  - c) filtrado más fino
  - d) expansión del dataset con scraping a otras páginas
- ***Abordar el problema de la explicabilidad***