

# Clasificación de imágenes antiguas

Autores: Álvaro Gutiérrez y Rubén Escobedo.

# Índice de contenidos

- Contexto
- Problemas previos
- Solución 1: clasificación
- Solución 2: regresión
- Cómo usar estos modelos
- Siguiendo pasos

# Índice de contenidos

- **Contexto**

- Problemas previos
- Solución 1: clasificación
- Solución 2: regresión
- Cómo usar estos modelos
- Siguiendo pasos

# Contexto

[La Rioja en la memoria](#) es un archivo gráfico digital que almacena fotos y vídeos de La Rioja de los años 1860 hasta 1990.

Para cada fotografía (que es lo que a nosotros nos interesa), se muestra un identificador, un título, un titular, el año y el lugar donde fue tomada.

La idea es construir un modelo que permita clasificar las imágenes en categorías. En nuestro caso, por décadas (80', 90' ...).

# Índice de contenidos

- Contexto
- **Problemas previos**
- Solución 1: clasificación
- Solución 2: regresión
- Cómo usar estos modelos
- Siguiendo pasos

# Problemas previos

Antes de comenzar a construir el modelo de clasificación, debemos dar respuesta a varios problemas:

- ¿Dónde guardamos los datos?
- ¿Todas las categorías están representadas?
- ¿Los datos están balanceados?
- ¿Cuántos datos debemos usar para entrenar el modelo?

# Índice de contenidos

- Contexto
- Problemas previos
- **Solución 1: clasificación**
- Solución 2: regresión
- Cómo usar estos modelos
- Siguiendo pasos

# Solución 1: clasificación

La primera aproximación fue crear, con las herramientas del entregable 2, un modelo de clasificación.

Para ello, separamos todas las fotografías en carpetas según el año en el que habían sido tomadas.

Después, construimos el modelo usando la arquitectura resnet18 (fine-tuning).



# Solución 1: clasificación

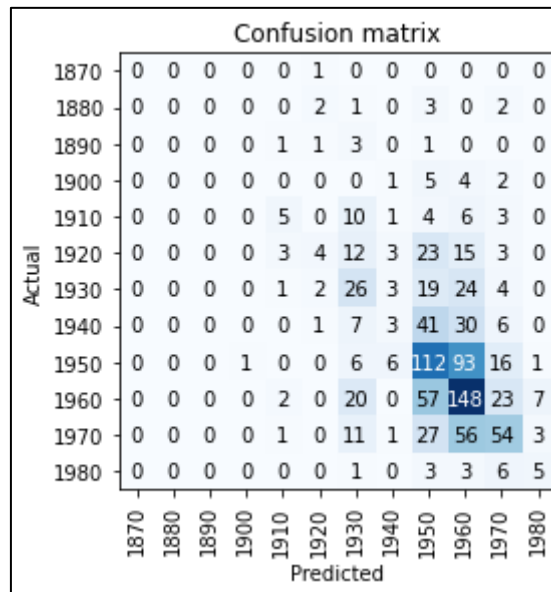
Los resultados del primer modelo son mejorables. 18% de precisión.

	1870	1880	1890	1900	1910	1920	1930	1940	1950	1960	1970	1980
1870	0	0	0	0	0	1	0	0	0	0	0	0
1880	0	0	2	0	0	2	0	0	1	0	0	0
1890	0	0	2	0	0	4	0	1	2	0	0	0
1900	0	0	2	2	0	2	3	0	1	1	0	0
1910	0	0	1	0	8	10	8	1	1	1	1	0
1920	0	0	2	2	10	13	5	2	7	2	4	0
1930	0	0	0	1	4	27	33	8	15	8	3	0
1940	0	0	1	1	1	7	2	16	29	22	2	1
1950	0	0	1	1	3	9	2	16	107	80	13	0
1960	0	0	0	0	2	8	18	7	46	146	32	1
1970	0	0	0	3	1	0	5	1	9	40	80	2
1980	0	0	0	0	1	1	1	0	0	2	22	4
	1870	1880	1890	1900	1910	1920	1930	1940	1950	1960	1970	1980

## Solución 1: clasificación

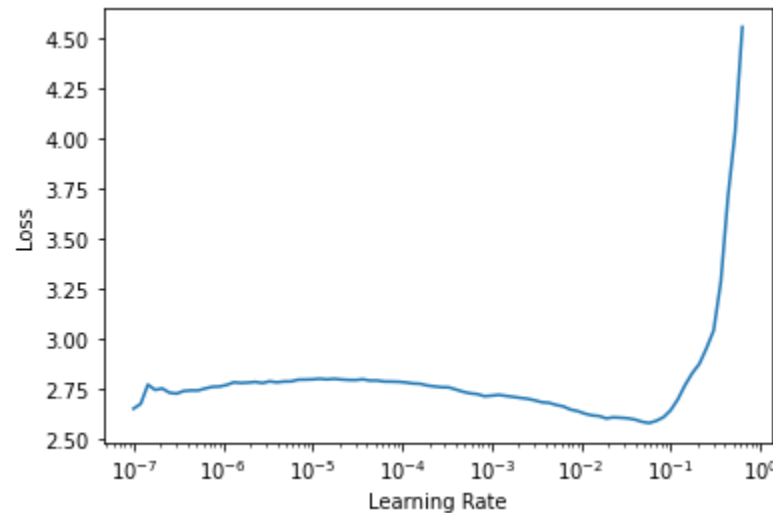
La primera mejora que proponemos es utilizar otra estructura para realizar el fine-tuning del modelo. En este caso, usamos `xse_resnet_18_deeper` (**NO** lo busques en Google Imágenes).

Obtenemos mejores resultados. Aumentamos la precisión del modelo hasta 35%. Reducimos la varianza.



# Solución 1: clasificación

Otra mejora a la clasificación ha sido utilizar la estructura efficientnet.

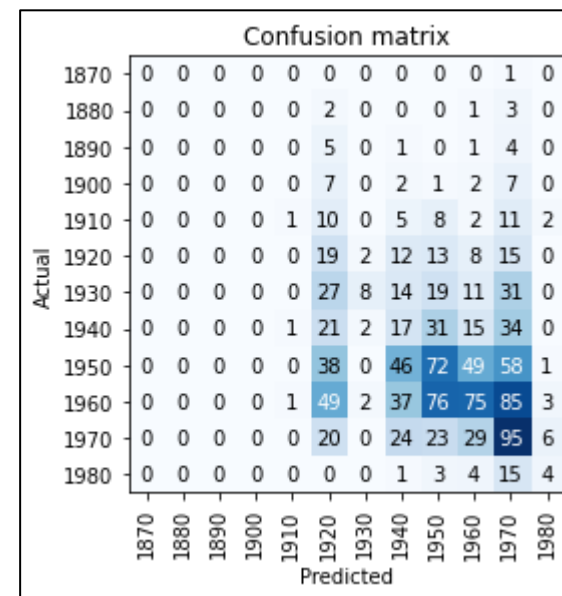


El mejor learning rate es para esta estructura es 0.05.

## Solución 1: clasificación

## Entrenamiento con la librería Timm y la arquitectura Efficentnet.

epoch	train_loss	valid_loss	time
0	2.836477	2.781734	06:01
epoch	train_loss	valid_loss	time
0	1.945647	1.787350	06:08
1	1.710518	1.746221	06:01
2	1.581159	1.792405	06:00
3	1.439653	1.794751	06:01
4	1.201044	1.970757	06:02
5	0.928315	2.311417	06:00
6	0.644542	2.436093	06:01
7	0.428995	2.566288	06:00
8	0.304632	2.625626	05:59
9	0.245358	2.631128	06:03



# Solución 1: clasificación

Entrenamiento con la librería Timm y la arquitectura Efficentnet.

epoch	train_loss	valid_loss	time
0	1.482337	2.436817	06:02
epoch	train_loss	valid_loss	time
0	1.394743	1.796886	06:05
1	1.311121	1.821014	06:07
2	1.015368	1.932837	06:07
3	0.705872	2.036676	06:06

Confusion matrix													
Actual	1870	1880	1890	1900	1910	1920	1930	1940	1950	1960	1970	1980	
	0	0	0	0	0	0	0	0	0	1	0	0	
	0	0	0	0	0	1	2	1	1	0	1	0	
	0	0	0	0	0	1	1	5	0	3	1	0	
	0	0	0	0	0	4	4	2	3	5	1	0	
	0	0	0	0	0	5	17	1	6	6	2	2	
	0	0	0	0	0	7	21	5	15	16	5	0	
	0	0	0	0	0	7	41	7	24	21	3	7	
	0	0	0	0	0	2	36	27	21	28	5	2	
	0	0	0	0	0	5	57	17	93	72	15	5	
	0	0	0	0	0	11	54	33	119	98	10	3	
	0	0	0	0	0	5	37	13	38	59	34	11	
	0	0	0	0	0	0	6	0	3	2	8	8	
Predicted	1870	1880	1890	1900	1910	1920	1930	1940	1950	1960	1970	1980	

# Solución 1: clasificación

Esta es una evaluación de los dos modelos de efficientnet.



# Índice de contenidos

- Contexto
- Problemas previos
- Solución 1: clasificación
- **Solución 2: regresión**
- Cómo usar estos modelos
- Siguiendo pasos

## Solución 2: regresión

Otra opción es intentar hacer regresión sobre los años en los que las fotografías fueron tomadas en vez de hacer clasificación a sus décadas. El rango de años va entre 1863 y 1991.

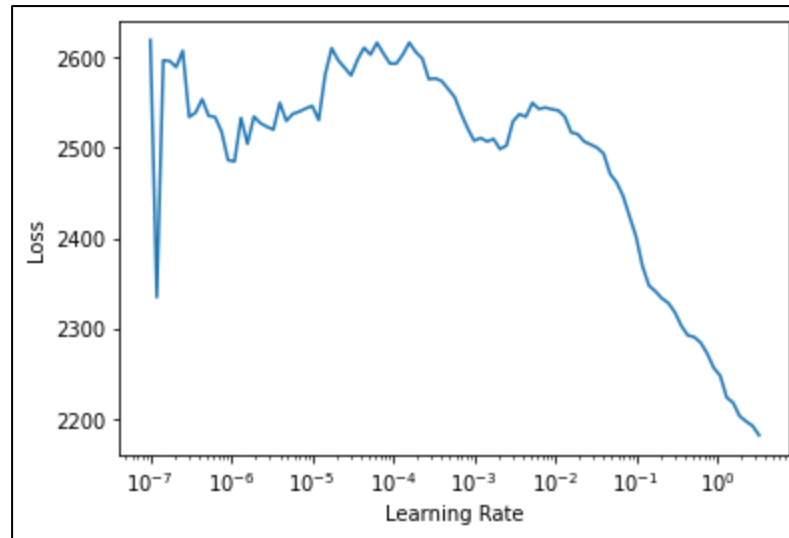
En este caso, no podemos usar la precisión como métrica a maximizar, sino que debemos usar otra (en nuestro caso, el RMSE –raíz cuadrada del valor medio de la suma del cuadrado de los errores-).

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^n (P_i - O_i)^2}{n}}$$



# Solución 2: regresión

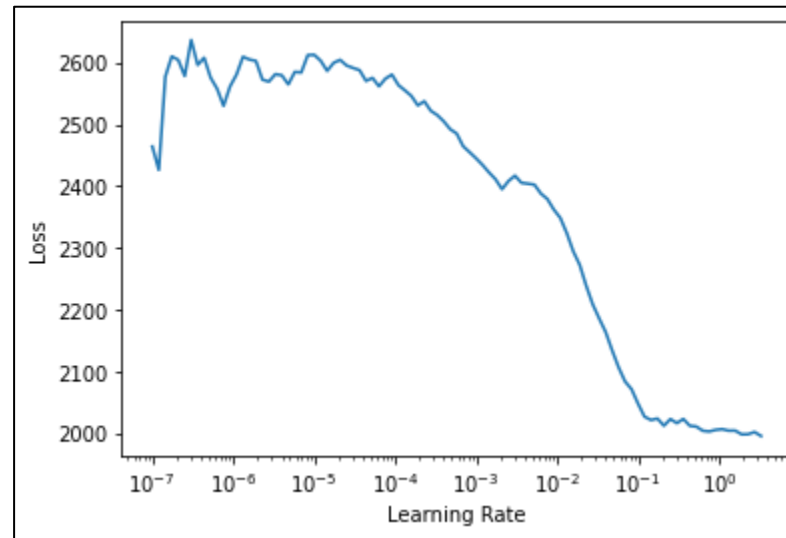
Al igual que antes, buscamos cuál es el mejor learning rate.



¿Qué learning rate usamos aquí? Nosotros elegimos 10. Sale mal: (+- 10 años).

# Solución 2: regresión

Probamos la mejor estructura de clasificación para hacer la regresión, pero los datos empeoran. El learning rate adecuado es 0.05.



# Solución 2: regresión

Mostramos una comparación entre las dos estructuras comentadas.

epoch	train_loss	valid_loss	mae	time
0	1989.104370	766.022400	21.889299	06:29
epoch	train_loss	valid_loss	mae	time
0	543.561951	303.249023	13.349410	06:30
1	358.539368	360.380157	12.571272	06:25
2	312.376251	378.835175	15.186860	06:23
3	278.344910	409.857849	14.083694	06:22
4	236.095428	226.490845	10.766935	06:24
5	200.048706	171.558777	9.644416	06:25
6	160.418320	167.743286	9.664773	06:23
7	139.942429	174.278259	9.562922	06:26
8	106.933258	170.353210	9.603662	06:24
9	88.164322	165.712662	9.475904	06:25

epoch	train_loss	valid_loss	mae	time
0	1239.443481	1409.311768	32.207619	06:26
epoch	train_loss	valid_loss	mae	time
0	797.918030	555.106567	17.123838	06:28
1	617.293335	527.643677	18.562073	06:21
2	561.430481	413.080475	14.708747	06:24
3	471.990784	343.654297	13.635862	06:22
4	402.748962	331.131592	13.145343	06:28
5	386.967529	310.700165	13.107221	06:24
6	355.108368	305.221313	13.279085	06:19
7	324.182739	289.824860	12.584779	06:17
8	319.385010	291.046967	12.773040	06:22
9	305.426392	290.726837	12.837651	06:21

# Índice de contenidos

- Contexto
- Problemas previos
- Solución 1: clasificación
- Solución 2: regresión
- **Cómo usar estos modelos**
- Siguiendo pasos

# Cómo usar estos modelos

Para usar estos modelos se puede usar [esta](#) aplicación creada a partir de un notebook que los recoge.

# Índice de contenidos

- Contexto
- Problemas previos
- Solución 1: clasificación
- Solución 2: regresión
- Cómo usar estos modelos
- **Siguientes pasos**

# Siguientes pasos

Para mejorar el trabajo, podríamos:

- Intentar obtener más imágenes para mejorar nuestros resultados.
- Buscar otras categorías que permitan hacer una mejor clasificación (grupos de veinte años, por ejemplo).
- Realizar data augmentation sobre las categorías poco representadas.
- Balancear el conjunto de datos.
- Crear un modelo de detección de objetos para encontrar personas en las imágenes.
- Crear un modelo de procesamiento del lenguaje para encontrar datos interesantes en las descripciones de las imágenes.