



DETECCIÓN ANOMALÍAS EN RADIOGRAFÍAS DE TÓRAX TRABAJO FIN DE MÁSTER BIG DATA & DATA SCIENCE

Aránzazu García Boronat

Tutor: Pablo Ramos Criado



- Descripción / Definición Problema /Objetivo.
- Estado de Cuestión Conceptos a destacar.
- Desarrollo del Problema.
- Conclusiones y Líneas Futuras.





- Participo en la competición de Kaggel, organizada por la Sociedad de Informática por Imágenes en Medicina (SIIM)
- Neumonía por COVID-19 se puede diagnosticar mediante la reacción en cadena: polimerasa o una radiografía de tórax.
- Modelo clasificación. Ayude a los radiólogos a diagnosticar millones de pacientes con la patologías con confianza y rapidez.



- Propia metodología utilizada por en las anotaciones SIIM para esta clasificación.
- Cuatro categorías, que son mutuamente excluyentes:

Aspecto típico/ patrón Clásico Neumonía COVID-19: Opacidades multifocales bilaterales, periféricas, de morfología redondeada, de distribución predominantemente pulmonar inferior.

Aspecto indeterminado: Alteraciones presentes, pero que no cumplen con los criterios de un patrón clásico ni tampoco de un patrón No-COVID-19. Ausencia de hallazgos típicos Y distribución unilateral, central o de predominio pulmonar superior

Aspecto atípico: Neumotórax, derrame pleural, edema pulmonar, consolidación lobar, nódulo o masa pulmonar solitaria, nódulos diminutos difusos, cavidad, entre otra posible enfermedad torácicas incluidas en esta categoría.

Negativo: No hay opacidades pulmonares.





- El propio sistema utilizado no identifica si una persona tiene o no tiene Covid-19.
- Otro problema no se han marcado los recuadros en todos los casos de radiografías con opacidad.
- Elevado número de patologías que recoge la categoría atípica.
- No disponer de suficientes imágenes. **Proceso de generalización**. Clases correctamente balanceadas Sobreaprendizaje.
- Las herramientas utilizadas.



Crear un modelo de clasificación binario que detecte una radiografía de tórax con patología /opacidad de una radiografía de tórax sin patología / sin opacidad por ello se realizará la agrupación (típica, atípica e indeterminada) versus negativa.



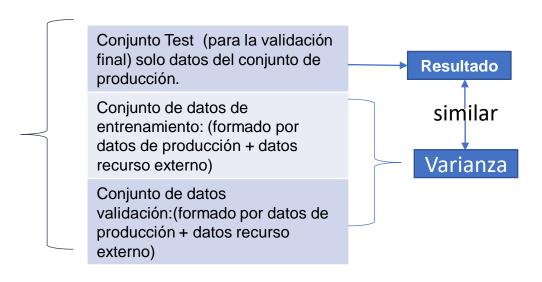
Deep learning

- Visión por computadora.
- CNN están formadas por: Capas convolucionales, capas de reducción o pooling.
- Otras aplicaciones.



Pocos datos

- Técnicas para solucionar problemas de pocos datos:
 - Data Augmentation. Aumento de imágenes en el entrenamiento.
 - Uso conjunto de Recursos externos:
 - ✓ Que sean datos del mismo dominio.
 - ✓ Ver si existen diferencias inapreciables.
 Nuevo conjunto de entrenamiento es representativo de nuestro conjunto de datos de producción.





DETECCIÓN ANOMALÍAS EN RADIOGRAFÍAS DE TÓRAX - Estado de Cuestión – Conceptos a destacar

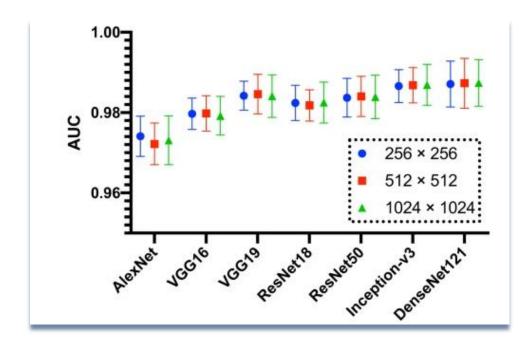
Artículo médico



- ✓ Conjunto de datos a gran escala:
 - ✓ NIH ChestX-ray 14: (112.120 x-ray, 30.805 pacientes (1992-2015), 14 etiquetas con PNL.)
 - ✓ CheXpert : (224.316 x-ray, 65.240 pacientes).
- ✓ Se demostró que es capaz de igualar el desempeño de los radiólogos en la tarea de clasificación binaria.



Artículo médico



Rendimiento de diferentes arquitecturas de CNN con diferentes tamaños de imagen de entrada en el conjunto de datos "ChestX-ray 14, Las barras de error representan las desviaciones estándar de los valores medios."

- ✓ Un modelo basado en CNN logró un AUC de **0,9824** ± 0,0043 (con una precisión de 94,64 .
- ✓ El entrenamiento previo, ayuda en el entrenamiento posterior de un conjunto de imágenes de tamaño más moderado.



- Conjunto de datos.
- Carga & EDA.
- Visualización radiografías tórax con cajas patologías.
- Preprocesamiento / Aumento de datos / partición.
- Modelos / métrica de evaluación & conclusión.
- Modelo con uso conjunto de datos externo.
- Mejora de resultados.
- Visualización de resultados.





Competición

SIIM-FISABIO-RSNA COVID-19 Detection



7.597 imágenes radiografías de tórax en formato DICOM, 6.334 imágenes etiquetadas.

SIIM COVID-19: Resized to 512px JPG



jpg. tamaño 512x512pixeles.

Recursos externos

COVID-19 Radiography Database



21.165 imágenes de radiografías de tórax en formato png de tamaño 299x299 pixeles.

SIIM COVID-19: Resized to 256px PNG



png de tamaño 299x299pixeles



DETECCIÓN ANOMALÍAS EN RADIOGRAFÍAS DE TÓRAX - Desarrollo del problema - Carga & EDA

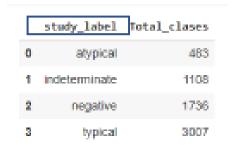
Carga Competición

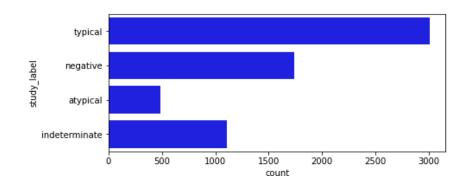


✓ Solo 6.334 imágenes.

Código notebook TFM: data frame final df_train. (Se utiliza Pandas: pd.read_csv(). Funciones: .loc(), .drop(), .str.replace(), merge(), entre otras para su creación)

EDA Competición



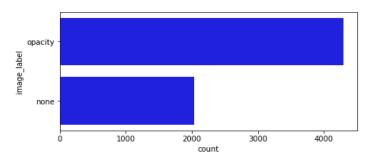


- ✓ Clases están totalmente desbalanceadas.
 - √ 8% atípica,
 - √ 17% indeterminada,
 - √ 27% negativa y
 - √ 47% típica



EDA Competición



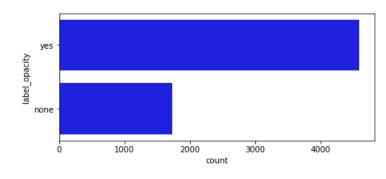


✓ Error en la columna "image_label".

"study_label", la clasificación de sin opacidad es de (negative=1.736 casos) mientras que aquí sin opacidad (none=2.040) por lo que hay un error de 304 casos con opacidad que estarían mal clasificados.

Nueva etiqueta de salida para la clasificación de con opacidad /sin opacidad

	label_opacity	Total	clases
0	none		1736
1	yes		4598



✓ Clases desbalanceada, solo se dispone de un 27% de los datos sin patología /sin opacidad.

Carga Competición + Recursos Externos

Dataframe – Unión todos

<class 'pandas.core.frame.DataFrame'> Int64Index: 27499 entries, 0 to 1344 Data columns (total 4 columns): # Column Non-Null Count Dtype -----27499 non-null obiect study_label 27499 non-null object

object

2 2L opacity no opacity 27499 non-null 2L_covid_no_covid 27499 non-null object dtypes: object(4)

memory usage: 2.1+ MB

27.499 = 6.334 + 21.165

Dataframe - Nuevo conjunto Entrenamiento

<class 'pandas.core.frame.DataFrame'> Int64Index: 26232 entries, 0 to 26231 Data columns (total 4 columns): # Column Non-Null Count Dtype ----id Θ 26232 non-null object study label 16040 non-null object 2L opacity no opacity 16040 non-null object 2L covid no covid 16848 non-null object dtypes: object(4) memory usage: 1.0+ MB

26.232 = 5.067 + 21.165

Dataframe – Conjunto Test Final

<class 'pandas.core.frame.DataFrame'> Int64Index: 1267 entries, 0 to 1266 Data columns (total 4 columns): # Column Non-Null Count Dtype -------id 0 1267 non-null object 1 study label 1267 non-null object 2 2L opacity no opacity 1267 non-null object 3 2L covid no covid 1267 non-null object

dtypes: object(4) memory usage: 49.5+ KB

1267 Competición

Conjunto inicial Competición

	study_label	Total_clases
0	atypical	483
1	indeterminat e	1.108
2	negative	1.736
3	typical	3.007

Conjunto test final

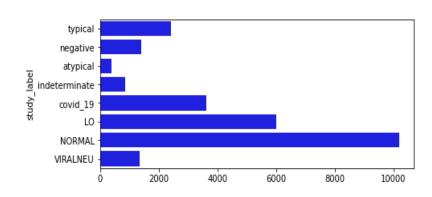
	study_label	Total clases	
)	atypical	97	17%
1	indeterminate	242	18%
2	negative	342	16%
3	typical	586	16%



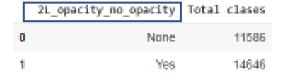


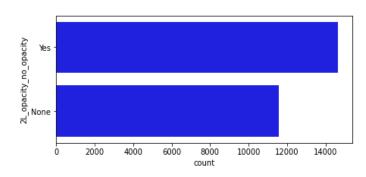
EDA Competición + Recursos Externos

	study label	Total clases
	Study_Tabel	TOTAL CLASES
0	LO	6012
1	NORMAL	10192
2	VIRALNEU	1345
3	atypical	386
4	covid_19	3616
5	indeterminate	866
6	negative	1394
7	typical	2421



√ 7 categorías resultantes de la unión , con las clases muy desbalanceadas.

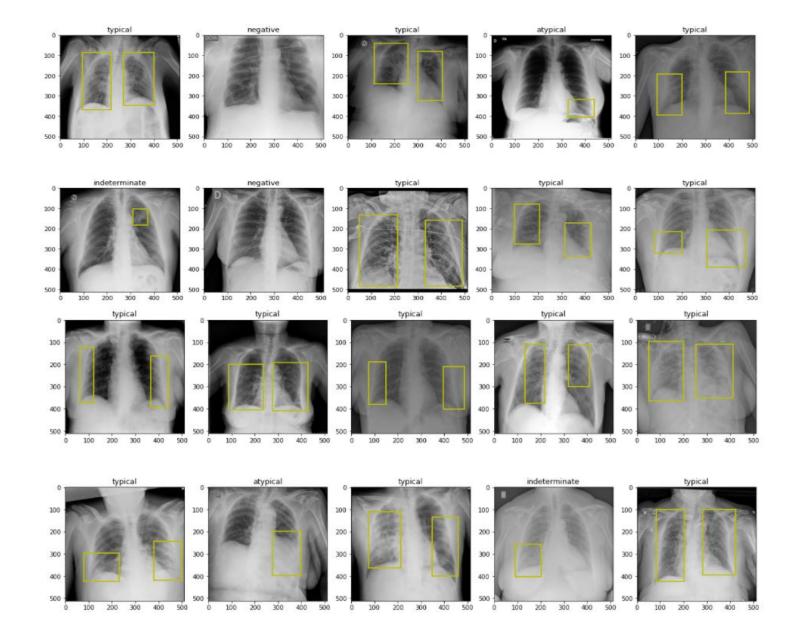




✓ Entre la agrupación binaria con opacidad / con anomalía (56% de los datos) versus sin opacidad/ sin anomalía (44% de los datos), Clases más balanceada.



DETECCIÓN ANOMALÍAS EN RADIOGRAFÍAS DE TÓRAX - Desarrollo del problema - Visualización radiografías tórax con cajas patologías



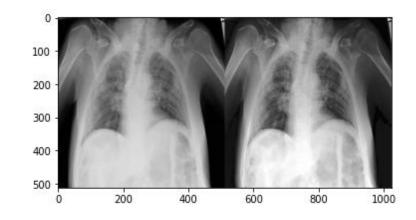
✓ Lo que podemos ver en estos ejemplos es que en todas las imágenes con neumonía de Covid-19/ típica son bilaterales los recuadros, mientras con en las patologías atípicas e indeterminadas los recuadro son unilaterales.



DETECCIÓN ANOMALÍAS EN RADIOGRAFÍAS DE TÓRAX - Desarrollo del problema - Preprocesamiento / Aumento de datos / partición

Preprocesamiento

- Módulo "exposure" de scikit-image: equalize_hist(). "mejora una imagen con bajo contraste, distribuye los valores de intensidad más frecuentes en la imagen"
- Modelo "keras.preprocessing" + np.array(). Conversión de las imágenes de entrada y de la etiqueta de salida a un vector



Aumento de datos

Módulo "ImageDataGenerator" de "tensorflow.keras.preprocessing.image"

horizontal_flip = True. zoom_range= 0.1. shear_range=0.05. brightnees_range=[0.8, 1.1]. validation_split=0.2.

Partición de datos

Solo datos competición

6224	80% train (5.068)
6334	20% valid (1.266)

Datos competición + Recursos externos

26.232	80% train (20.986)
5.067 (80%) + 21.165	20% valid (5.246)
1.267 (20%)	Test Final

Conjunto train lo aplicamos método ".flow_from_dataframe()" y La partición de los datos la marcamos dentro las transformaciones de la función "ImageDataGenerator()".



Varios modelos de redes neuronales convolucionales
La realización de cada uno de los modelos tiene como objetivo
mejorar rendimiento sobre los modelos anteriores, así como
subsanar los problemas obtenidos en los anteriores
desarrollos.





2-3 Finalidad

Mejora del rendimiento del modelo anterior con la utilización de transfer learning Xception...

				-1 10 11	
N.º	Datos de Entrada	Modelo	Parámetros de interés	Clasificación	Rendimiento
notebk.					
6.2	Se utilizan los datos	Se utiliza transfer	1º Entrenamiento	4 clases de	1º Entren.:
У	de la competición.	learning, se importa el		salida:	
6.3		modelo Xception .	optimizer= SGD(Ir=0.2,	- atypical.	51,74 % de
	La entrada es carpeta de	como modelo base:	momentum=0.9, decay=0.001)	-	accuracy en
	imágenes train del	Utilizamos los pesos de	- epochs=5.	Indeterminate	datos de
	directorio:./Fuentes/train	la propia red:	- base_model.trainable:False.		validación.
	y data frame con	weights="imagenet"		- negative	2º Entren.:
	columna de etiqueta de	No se incluye la capa de	2º Entrenamiento, Ajuste	- typical	
	salida.	salida, ya que he de	Fino cambio:		59,32 % de
		cambiarla por mi capa de			accuracy en
	Total, imágenes:	salida: include_top=False.	- base_model.trainable:True		datos de
	5068 para train	- Se Añade al modelo base	- Se importa de Keras, modulo		validación.
	1266 para valid	una capa Global Average	keras.callbacks.EarlyStopping		
		Pooling.	. Me permite detener el		
	Con Data	- Se añade la capa de	entrenamiento si la función de		
	Augmentation	salida, con una capa dense	pérdidas de validación aumenta		
		de dimensión 4 y función	con un patience=10.		
		de activación	-optimizer=Adam(lr=0,001).		
		sigmoidea: "softmax".	- epochs=100		

Métrica de evaluación

```
Test Accuracy: 0.5932069510268563

Confusion matrix: tf.Tensor(

[[ 21     5     14     49]

[ 38     8     64     129]

[ 32     20     242     62]

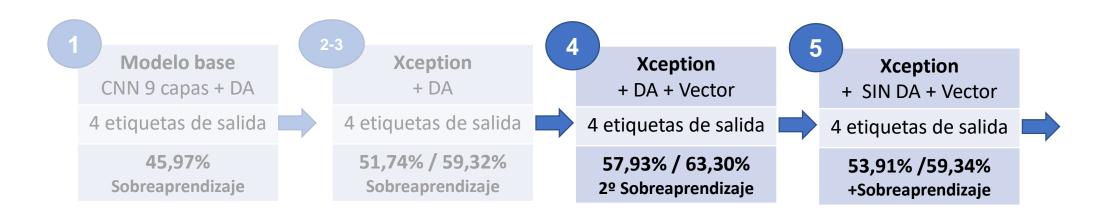
[ 36     12     54     480]], shape=(4, 4), dtype=int32)
```

Conclusión

Sobreaprendizaje, clasifica mejor las dos clases con mayor peso: negative y typical, sin embargo, las otras dos tienen malos resultados.

... se podría mejorar el resultado si se modifican los datos de entrada con preprocesamiento



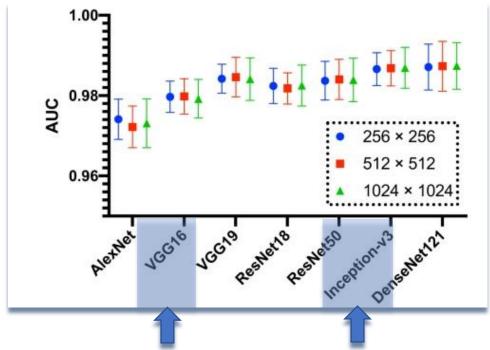






DETECCIÓN ANOMALÍAS EN RADIOGRAFÍAS DE TÓRAX - Estado de Cuestión – Conceptos a destacar

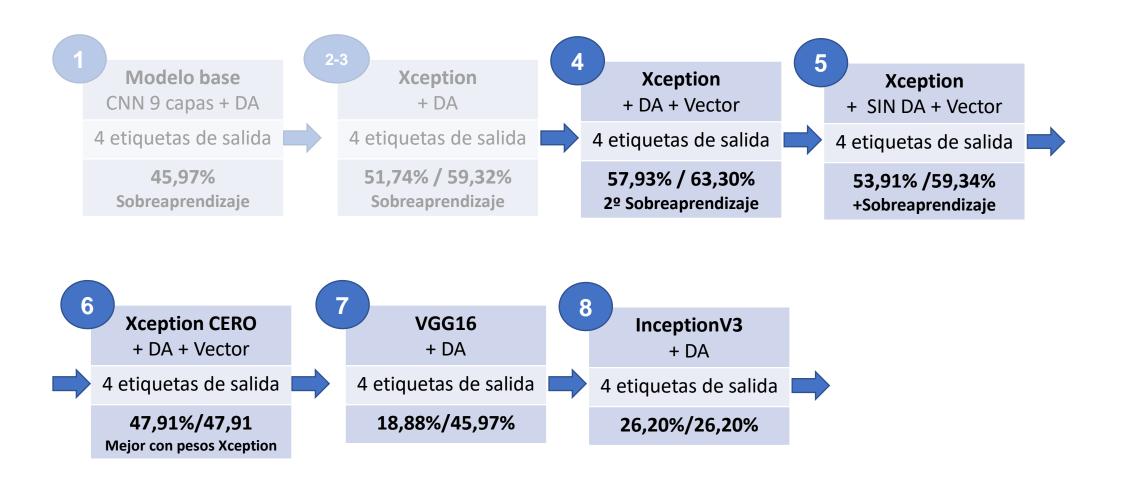
Artículo médico



Rendimiento de diferentes arquitecturas de CNN con diferentes tamaños de imagen de entrada en el conjunto de datos "ChestX-ray 14, Las barras de error representan las desviaciones estándar de los valores medios."

- ✓ Un modelo basado en CNN logró un AUC de **0,9824** ± 0,0043 (con una precisión de 94,64 .
- ✓ El entrenamiento previo, ayuda en el entrenamiento posterior de un conjunto de imágenes de entrenamiento de tamaño moderado







Mejora del rendimiento con modelo Xception y clasificación binaria con anomalía/con opacidad versus sin anomalía/sin opacidad. "Sólo 27% sin patología"

N.º	Datos de Entrada	Modelo	Parámetros de interés	Clasificación	Rendimiento
notebk.	0 1 1 .		105.	2 1	10.5
6.10	Se utilizan los datos	Se utiliza transfer	1º Entrenamiento	2 clases de	1º Entren.:
	de la competición.	learning y se importa el		salida:	
		modelo Xception .	- optimizer= Nadam(lr=0,001)	- Yes (typical,	74,96 % de
	La entrada es carpeta de	como modelo base:	- epochs=10.	atypical e	accuracy en
	imágenes train del	Se utilizan los pesos de	- base_model.trainable:False.	indeterminate	datos de
	directorio:./Fuentes/train	la propia red:)	validación.
	y data frame con	weights="imagenet"	2º Entrenamiento, Ajuste	- None	2º Entren.:
	columna de etiqueta de	No se incluye la capa de	Fino cambio:	(Negative)	
	salida.	salida, ya que he de		`	80,17 % de
		cambiarla por mi capa de	- base model.trainable:True		accuracy en
	Total, imágenes:	salida: include top=False.	- Se importa de Keras, modulo		datos de
	5068 para train	- Se añade al modelo base	keras.callbacks.EarlyStopping		validación.
	1266 para valid	una capa Global Average	.Nos permite detener el		randadioni
	1200 para rana	Pooling y otra Bach	entrenamiento si la función de		
	Con Data	Normalization.	pérdidas de validación aumenta		
	Augmentation	- Se añade la capa de	con un patience=10.		
	Augmentation				
		salida, con una capa dense	-optimizer=Nadam(lr=0.001)		
		de dimensión 2 y función	- epochs=100		
		de activación			
		sigmoidea: "softmax".			

... No se puede considerar solución a este problema, por ello se analiza la otra posibilidad de clasificación binaria

Métrica de evaluación

Test Accuracy: 0.80173775671406
Confusion matrix: tf.Tensor(
[[226 130]
 [121 789]], shape=(2, 2), dtype=int32)

Conclusión

Mejor resultado 74,96% en 1º Entrenamiento.

Indicios sobreaprendizaje en el 2º Ent con bajadas + pronunciadas en los epochs 4, 8, 11y 15.





... Podemos mejorar el rendimiento con uso de un conjunto de datos externo.

... importante cumplir requisitos de representatividad.



Mejora del rendimiento <u>con uso del conjunto de datos externo</u> con modelo Xception y clasificación binaria con anomalía/con opacidad versus sin anomalía/sin opacidad. "Clases +

balanceadas"

N.º	Datos de Entrada	Modelo	Parámetros de interés	Clasificación	Rendimiento
N.º notebk.	Datos de Entrada Se utiliza la unión de los conjuntos de datos: Total, 26.232 imágenes. La entrada es carpeta de imágenes: "train_nuevo_final" del directorio:"./Fuentes_C" y data frame: "df_new_train_final" con columna de etiqueta de salida.	Modelo Se utiliza transfer learning y se importa el modelo Xception. como modelo base:Se Utiliza los pesos de la propia red: weights="imagenet" - No se incluye la capa de salida, ya que he de cambiarla por mi capa de salida: include_top=False Se añade al modelo base una capa Global Average Pooling y una capa Bach Normalization.	Parámetros de interés 1º Entrenamiento - optimizer= Nadam(lr=0,001) - epochs=20 - base_model.trainable:False.	Clasificación 2 clases de salida: - Yes (con opacidad) - None (sin opacidad)	Pendimiento 1º Entren.: 76,67% de accuracy en datos de validación.
	Total, imágenes: 20.986 para train 5.246 para valid Con Data Augmentation	Normalization Se añade la capa de salida, con una capa dense de dimensión 2 y función de activación sigmoidea: "softmax".			

... Realizamos ajuste fino con un 2º entrenamiento para ver si mejora el resultado

Métrica de evaluación

Matriz Confusión - 1º Entrenamiento

Valid Accuracy: 76,67%			
0 - None	965	429	69%
1 - Yes	795	3057	79%

Matriz Confusión - Datos Test Final

Test Final Accuracy: 75,20%			
0 - None	238	98	71%
1 - Yes	213	705	77%

Conclusión

Mejora resultado.

Confirmamos que sí es representativo. Pasamos del 74,96% al 75,20%.



Mejora del rendimiento con uso del conjunto de datos externo y ajuste fino en el 2º entrenamiento.

N.º	Datos de Entrada	Modelo	Parámetros de interés	Clasificación	Rendimiento
notebk. 9.1	Utilizo unión de los conjuntos de datos: Total, 26.232 imágenes. La entrada es carpeta de imágenes: "train_nuevo_final" del directorio:"./Fuentes_C" y data frame: "df_new_train_final" con columna de etiqueta de salida. Total, imágenes: 20.986 para train 5.246 para valid Con Data Augmentation	Utilizo transfer learning e importo el modelo Xception. como modelo base: - Utilizo los pesos de la propia red: weights="imagenet" - No incluyo la capa de salida, ya que he de cambiarla por mi capa de salida: include_top=False Añado al modelo base una capa Global Average Pooling y una capa Bach Normalization Añado la capa de salida, con una capa dense de dimensión 2 y función de activación sigmoidea: "softmax".		2 clases de salida: - Yes (con opacidad) - None (sin opacidad)	74,17% de accuracy en datos de validación.

... El mejor modelo lo obtenemos con el primer entrenamiento con uso de datos externos. ¿Puede mejorar?

Métrica de evaluación

Matriz Confusión - 2º entrenamiento

Valid Accuracy:74,17%			
0 - None	62	1332	4%
1 - Yes	23	3829	99%

Matriz Confusión - Datos test Final

Test Final Accuracy:73,68%			
0 - None	16	320	5%
1 - Yes	10	908	99%

Conclusión

Empeora resultado. Sobreaprendizaje, clase mayoritaria.



Mejora del rendimiento <u>con uso del modelo preentrenado.</u> Mejor modelo con uso de datos externo. Entrenamos de nuevo los datos de la competición iniciales con uso del modelo preentrenado.

N.O	Datos de Entrada	Modelo	Parámetros de interés	Clasificación	Rendimiento
10	Se utilizan los datos solo de la competición 6.334 La entrada para train es carpeta de imágenes: "train_nuevo_ini" del directorio:"./Fuentes_C" y data frame: "df_new_train_final_ini" con columna de etiqueta de salida. Los datos de test se utiliza la carpeta de imágenes: "test_nuevo_fin!" del directorio:"./Fuentes_C" y data frame: "df_new_test_final" con columna de etiqueta de salida. 5.067 para train 1.267 para Test Final Con Data Augmentation	Utilizo Modelo preentrenado. Cargo los pesos del modelo obtenido en el primer entrenamiento de la unión del conjunto de datos de la competición más el conjunto de datos externo.	Entrenamiento - optimizer= Nadam(lr=0,01) - epochs=15 - base_model.trainable: True.	2 clases de salida: - Yes (con opacidad) - None (sin opacidad)	77,83% de accuracy en datos de validación.

Métrica de evaluación

79/79 [======] - 71s [0.46506696939468384, 0.7783094048500061]

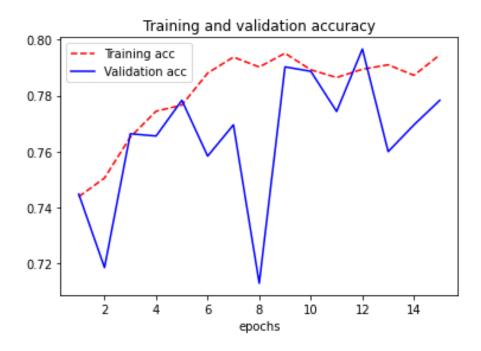
893ms/step - loss: 0.4651 - accuracy: 0.7783

Conclusión

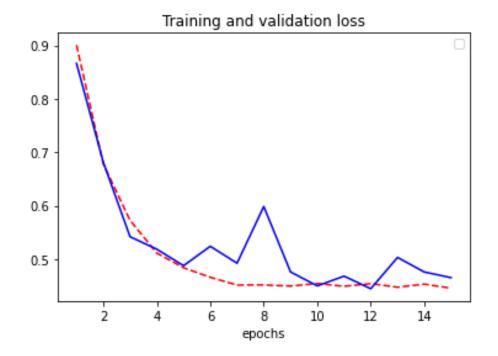
Mejoramos el resultado. Conseguimos pasar de una precisión en test del 75,20% al 77,83% (+2 puntos porcentuales)



Gráficas con la evolución de las métricas de "accuracy" y "función de pérdidas" tanto en el entrenamiento como en la validación final del subconjunto de test



... mantienen una pendiente ascendente, con una bajada más pronunciada en los datos de test en el epoch 8. Si vemos la escala vertical son rangos entre [0,72 y 0,80], por lo que no es una bajada muy pronunciada, y al ser puntual no podemos considerarlo como un problema de sobreaprendizaje

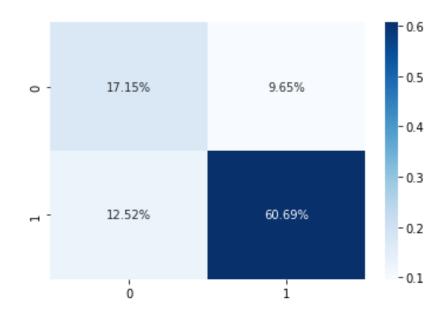


... mismo comportamiento. Mantienen la misma pendiente descendente. Con la subida puntual del epoch 8.



Matriz de confusión

Prest Accuracy: 0.7783094098883573
Matrix confusion: tf.Tensor(
[[215 121]
 [157 761]], shape=(2, 2), dtype=int32)



..., el 77,83% de accuracy en los datos de test se consiguen con un 60,69% de precisión de verdaderos positivos y un 17,15% de precisión de verdaderos negativos. La diagonal principal mantiene valores más elevados que la diagonal de los casos no acertados.



Evolución tasa de acierto en los datos de validación Test Final

Modelo resultante: 1º Entrenamiento unión bases de datos

Matriz Confusión - Datos test Final

Test Final Accuracy:75 20%

Test Fillal Acc	uracy.73,20/6	
0 - None	238	98
1 - Yes	213	705



Modelo resultante: Tras cargar modelo preentrenado y entrenar de nuevo

Matriz Confusión - Datos test Final

Test Final Accuracy:77,83%		
0 - None	215	121
1 - Yes	157	761



^{...,} El aumento de accuracy del modelo en más de dos puntos porcentuales de 75,20% a 77,83%, nos ha permitido mejorar la tasa de acierto de verdaderos positivos. Conseguimos aumentar en 6 puntos porcentuales la tasa de acierto de radiografías con opacidad/con anomalía. Pasamos de un 77% de tasa de acierto a un 83% de tasa de acierto.



Conclusiones Finales

- Pocos datos: 6.334 imágenes de radiografías de tórax y con categorías muy desbalanceadas: 8% atípica, 17% indeterminada, 27% negativa y 47% típica.
- Rendimiento máximo del 74,96% de precisión en datos de validación. Modelo de clasificación binario con opacidad versus si opacidad.
- Con el uso de recursos externos y tras confirmar que el nuevo conjunto de datos de entrenamiento es representativo del conjunto de datos original, conseguimos aumentar accuracy en test a 75,20%.
- Con uso de **modelo preentrenado en conjunto de datos original**, finalmente **mejora** hasta **77,83%** de accuracy en test final.
- Si nos centramos en las radiografías con patologías, los casos con opacidad, nuestro resultado final es una tasa de acierto del 83%. Se consigue aumentar 6 puntos porcentuales, con el último entrenamiento, tras cargar modelo preentrenado.



Líneas Futuras

- Este resultado no es suficientemente bueno como para considerarse como un caso de producción o un caso de éxito en la obtención de un modelo de clasificación binario de radiografías de tórax.
- El artículo médico reflejaba mejores resultados, llegando a un AUC superior al 98%.
- Para ello se deberá hacer uso de un conjunto de datos de mayor dimensión (NIH ChestX-ray 14, (+112K), CheXpert (+224K)).
- Además del número de imágenes elevado, las clases han de estar más balanceadas.
- Especial atención en cumplir requisitos de representatividad.
 - Precisión elevada . Hospital /máquina Rayos-X.
 - Solución ámbito general. Muestra lo más diversa.
- Limitación tiempo, herramientas, no hacen posible realizar las mejoras propuestas.