

“Detección de Neumonía, a través de las imágenes de radiografías con CNN”



Fuente de datos

kaggle

Este proyecto está basado en un dataset extraído de la plataforma Kaggle, la cual, es una plataforma abierta que cuenta con múltiples data sets públicos.

El dataset se trata de 3 carpetas cargadas con radiografías normales y radiografías con Neumonía, corresponde a pacientes pediátricas de 1 a 5 años, son radiografías reales.

PAUL MOONEY · UPDATED 7 YEARS AGO

7035 Code Download

Chest X-Ray Images (Pneumonia)

5,863 images, 2 categories

Data Card Code (3170) Discussion (65) Suggestions (0)

About Dataset

Context
[http://www.cell.com/cell/fulltext/S0092-8674\(18\)30154-5](http://www.cell.com/cell/fulltext/S0092-8674(18)30154-5)

Usability
7.50

License
Other (specified in description)

Expected update frequency
Not specified

Tags
Health Biology
Online Communities
Image Medicine

Normal Bacterial Pneumonia Viral Pneumonia

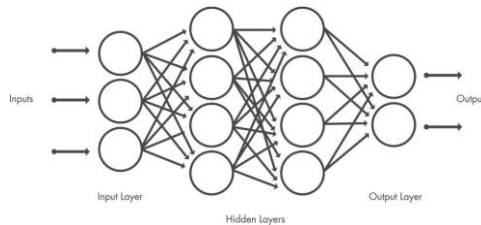
Figure S6. Illustrative Examples of Chest X-Rays in Patients with Pneumonia, Related to Figure 6
The normal chest X-ray (left panel) depicts clear lungs without any areas of abnormal opacification in the image. Bacterial pneumonia (middle) typically exhibits a focal/lobar consolidation, in this case in the right upper lobe (white arrows), whereas viral pneumonia (right) manifests with a more diffuse "interstitial" pattern in both lungs.
[http://www.cell.com/cell/fulltext/S0092-8674\(18\)30154-5](http://www.cell.com/cell/fulltext/S0092-8674(18)30154-5)

View more



Detalles generales del proyecto

Para el desarrollo del proyecto se implementó una red neuronal convolucional (CNN), de clasificación binaria, utilizando Keras con TensorFlow, aplicando técnicas como data augmentation, early stopping y ajuste de umbral basado en la curva de ROC para mejorar el rendimiento en el conjunto de test. También se realizó Tuning y Ensemble.



DEEP LEARNING



EDA Análisis exploratorio de datos

El dataset estaba compuesto por 3 carpetas, train, test y val, en el EDA se evaluó con dataframe la etiqueta de las radiografías, y se evidencio que las radiografías con neumonía habían muchas de una misma persona, pero además había una misma persona con causal viral y con causal bacteriano, por lo que se dejo una imagen por persona máximo, excepto si una persona tenia una imagen de origen viral y bacteriano, quedando una de cada una. A partir de ese principio se dejo el dato balanceado.



Resultado de predicción real

Caso 1: Se cargo una foto pediátrica, sacada de internet, con Neumonía, y se pone a prueba la app, da una respuesta correcta que se puede visualizar en la imagen que se ve a continuación.

Guardar como

MÉTODO
POST

ESQUEMA: // HOST [":" PUERTO] [RUTA ["?" CONSULTA]]
http://127.0.0.1:5000/predict
length: 29 byte(s)

EnviarAbortarEnviarAbortarEnviarAbortar (2 s)Abortar (1 s)AbortarEnviarEnviarEnviarAbortarEnviarEnviarEnviarEnviarEnviarAbortarEnviar

PARÁMETROS DE CONSULTA

ENCABEZADOS
Content-Type : multipart/form-data
+ Agregar encabezado Agregar autorización

CUERPO
file [Archiv] = 003-radiografia-neumonia.jpg.webp (type: "image/webp"; size: 15944 bytes)
+ Agregar parámetro de formulario multipart/form-data multipart/form-data

Respuesta

Caché Detectada - Tiempo Transcurrido: 393 Ms

200 OK

ENCABEZADOS
Servidor: Herramienta/3.1.3 Python/3.10.11
Fecha: Vie, 18 jul 2025 23:39:06 GMT
Tipo de contenido: aplicación/json
Longitud del contenido: 62 bytes
Conexión: cerca
ENCABEZADOS DE SOLICITUD COMPLETOS

CUERPO

```
{
  predicted_class : "NEUMONÍA",
  probabilidad : 0,6359
}
```

líneas numéros de líneas

length: 62 bytes



Resultado de predicción real

Se replica el ejercicio en Google colab, dando la misma respuesta (se comparte foto de la predicción).

```
1 # === Cargar los modelos entrenados ===
2 from tensorflow.keras.models import load_model
3 import numpy as np
4 from tensorflow.keras.preprocessing.image import img_to_array
5 from PIL import Image
6 import matplotlib.pyplot as plt
7 import io
8
9 # Cargar modelos guardados
10 modelo1 = load_model("modelo_afinado_tuning.keras")
11 modelo2 = load_model("modelo_2_ensamble.keras")
12
13 # === Función de ensamble con probabilidad ===
14 def predict_ensemble(model1, model2, image_array, threshold=0.5):
15     pred1 = model1.predict(image_array)[0][0]
16     pred2 = model2.predict(image_array)[0][0]
17     avg_pred = (pred1 + pred2) / 2
18     class_label = 'PNEUMONIA' if avg_pred > threshold else 'NORMAL'
19     return class_label, avg_pred
20
21 # === Probar con una imagen ===
22 # Ruta de una imagen de prueba
23 image_path = "/content/003-radiografia-neumonia.jpg.webp" # Cambia por la ruta de tu imagen
24
25 # Cargar y preprocesar imagen
26 image = Image.open(image_path).convert("L")
27 image = image.resize((256, 256))
28 img_array = img_to_array(image) / 255.0
29 img_array = np.expand_dims(img_array, axis=0)
30
31 # Predicción con ensamble
32 class_label, prob = predict_ensemble(modelo1, modelo2, img_array)
33
34 # Mostrar resultado
35 print(f"Diagnóstico: {class_label} ({prob * 100:.2f}% de certeza)")
36 plt.imshow(image, cmap='gray')
37 plt.title(f"Predicción: {class_label} ({prob * 100:.2f}%)")
38 plt.axis('off')
39 plt.show()
```

Diagnóstico: PNEUMONIA (63.59% de certeza)

Predicción: PNEUMONIA (63.59%)



Resultado de predicción real

Caso 2: Se cargo una imagen de radiografía pediátrica normal sacada de internet, y se pone a prueba en Talent API tester y da Normal.

ADOR

Guardar como

MÉTODO
POST

ESQUEMA // HOST ["*" PUERTO] [RUTA ["?" CONSULTA]]
http://127.0.0.1:5000/predict

length: 29 byte(s)

EnviarAbortarEnviarAbortar (2 s)Abortar (1 s)AbortarEnviarEnviarEnviarAbortarEnviarEnviarEnviarEnviarAbortarEnviar

PARÁMETROS DE CONSULTA

ENCABEZADOS
Content-Type : multipart/form-data

+ Agregar encabezado

Agregar autorización

CUERPO
file [Archiv] = Normal 2.jpeg (type: "image/jpeg", size: 463259 bytes)

+ Agregar parámetro de formulario

multipart/form-data multipart/form-data

Respuesta

Caché Detectada - Tiempo Transcurrido: 411 Ms

200 OK

ENCABEZADOS
Servidor: Herramienta/3.1.3 Python/3.10.11
Fecha: Vie, 18 jul 2025 23:33:28 GMT
Tipo de contenido: aplicación/json
Longitud del contenido: 59 bytes
Conexión: cerca

ENCABEZADOS DE SOLICITUD COMPLETOS

CUERPO
{
 clase_predicha : "NORMAL",
 probabilidad : 0,0186
}

lines nums números de líneas

copiar

length: 59 bytes



Resultado de predicción real

Se replica el ejercicio en Google colab, dando la misma respuesta (se comparte foto de la predicción.)

```
1 # === Cargar los modelos entrenados ===
2 from tensorflow.keras.models import load_model
3 import numpy as np
4 from tensorflow.keras.preprocessing.image import img_to_array
5 from PIL import Image
6 import matplotlib.pyplot as plt
7 import io
8
9 # Cargar modelos guardados
10 modelo1 = load_model("modelo_afinado_tuning.keras")
11 modelo2 = load_model("modelo_2_ensamble.keras")
12
13 # === Función de ensamble con probabilidad ===
14 def predict_ensemble(model1, model2, image_array, threshold=0.5):
15     pred1 = model1.predict(image_array)[0][0]
16     pred2 = model2.predict(image_array)[0][0]
17     avg_pred = (pred1 + pred2) / 2
18     class_label = 'PNEUMONIA' if avg_pred > threshold else 'NORMAL'
19     return class_label, avg_pred
20
21 # === Probar con una imagen ===
22 # Ruta de una imagen de prueba
23 image_path = "/content/Normal 2.jpeg" # Cambia por la ruta de tu imagen
24
25 # Cargar y preprocesar imagen
26 image = Image.open(image_path).convert("L")
27 image = image.resize((256, 256))
28 img_array = img_to_array(image) / 255.0
29 img_array = np.expand_dims(img_array, axis=0)
30
31 # Predicción con ensamble
32 class_label, prob = predict_ensemble(modelo1, modelo2, img_array)
33
34 # Mostrar resultado
35 print(f"Diagnóstico: {class_label} ((prob * 100:.2f)% de certeza)")
36 plt.imshow(image, cmap="gray")
37 plt.title(f"Predicción: {class_label} ((prob * 100:.2f)%)")
38 plt.axis('off')
39 plt.show()
```

1/1 ————— 1s 790ms/step
1/1 ————— 1s 521ms/step
Diagnóstico: NORMAL (1.86% de certeza)

Predicción: NORMAL (1.86%)



Desafíos a futuro

(Desafíos)

- Se trabajó con baja capacidad computacional, pero se solucionó utilizando colores grises e imágenes con el tamaño de 256 x 256 pixeles
- A futuro se sugiere la incorporación de fotografías más pixeladas, con mayor contenido en la descripción del diagnostico en la etiqueta.
- Hacer la prueba con Segmentación Semántica con U-net, que esa herramienta usada en medicina.



Conclusión

Herramientas como el Deep Learning, pueden servir para interpretar exámenes imagenológicos, esto puede ser un primer paso, frente a un sin número de proyectos que se pueden llevar a cabo en salud.

