

INTRODUCCION

La inteligencia artificial generativa ha supuesto un cambio profundo en el campo de la visión artificial, ya que no se limita únicamente a reconocer patrones en imágenes, sino que también es capaz de crearlas, mejorarlas o modificarlas. Este avance abre nuevas posibilidades en sectores tan relevantes como la salud, la industria o el entretenimiento.

En el ámbito médico, donde ya se utilizan redes neuronales para tareas de diagnóstico —como la detección de fracturas en radiografías—, los modelos generativos pueden ir un paso más allá. Por ejemplo, permiten **mejorar la resolución de imágenes de baja calidad**, lo cual es especialmente útil cuando las radiografías o escáneres no tienen la nitidez suficiente. También pueden **generar imágenes sintéticas realistas** para ampliar conjuntos de datos y entrenar modelos de detección más robustos, algo muy necesario cuando se dispone de pocas imágenes de ciertas patologías.

Entre las distintas familias de modelos, los **Modelos de Difusión** (como *Stable Diffusion*) se han consolidado como una de las opciones más potentes. A diferencia de las GANs clásicas, estos modelos ofrecen mayor estabilidad y logran un nivel de detalle muy cercano al de imágenes reales. En este trabajo se ha optado por usar un modelo de difusión porque permite tanto la **generación desde texto o bocetos** como la **superresolución de imágenes**, dos funciones que resultan especialmente relevantes para el campo de la visión artificial aplicada a la medicina.

En resumen, explorar la IA generativa en este contexto no solo es un ejercicio académico, sino una forma de vislumbrar cómo estas tecnologías pueden convertirse en herramientas de apoyo para los profesionales de la salud, facilitando diagnósticos más rápidos y fiables, o ayudando a preparar datos de entrenamiento más completos para futuros sistemas inteligentes.

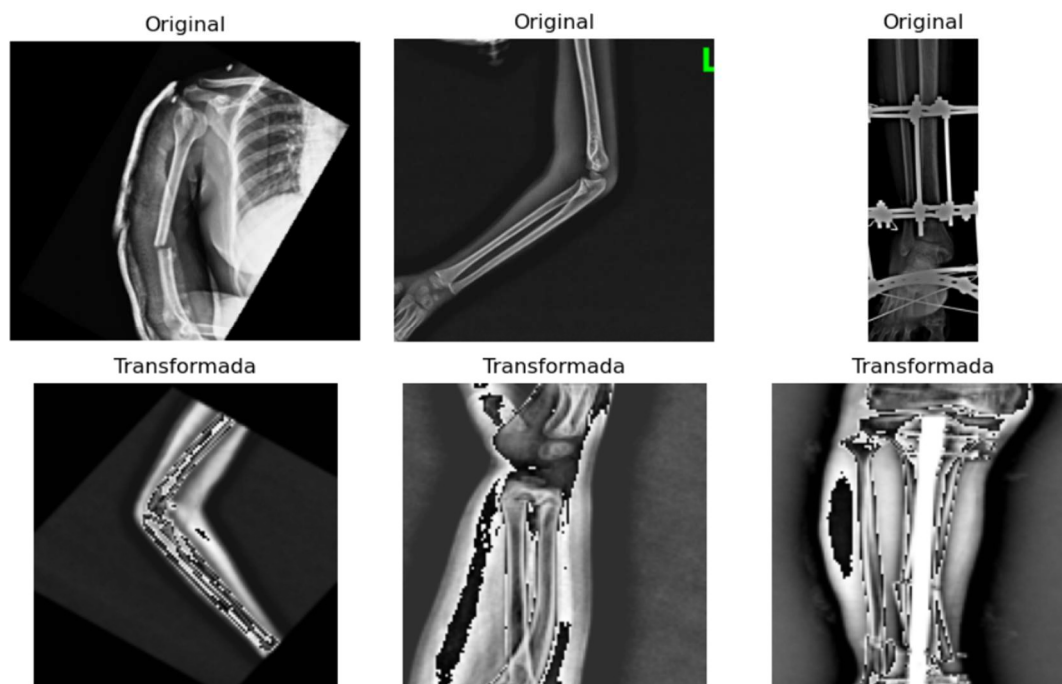
DESARROLLO TECNICO

Preparación de los datos

Primero organicé las radiografías en dos carpetas (fracturado / no fracturado). Para que el modelo pudiera trabajar con ellas, apliqué transformaciones básicas: redimensionar, normalizar y aumentar los datos con rotaciones y giros. Esto evita que el modelo memorice las imágenes y le ayuda a generalizar.

```
transformacion_entrenamiento = transforms.Compose([\n    transforms.Resize((128, 128)),\n    transforms.RandomHorizontalFlip(),\n    transforms.RandomRotation(10),\n    transforms.ToTensor(),\n    transforms.Normalize([0.5], [0.5])\n])\n\ntransformacion_prueba = transforms.Compose([\n    transforms.Resize((128, 128)),\n    transforms.ToTensor(),\n    transforms.Normalize([0.5], [0.5])\n])
```

Ejemplos antes y después de transformaciones



Arquitectura de la red

El modelo que construí es una red neuronal convolucional con tres capas convolucionales seguidas de funciones ReLU y max pooling, y al final dos capas densas.

```
class ModeloRayosX(nn.Module):
    def __init__(self):
        super(ModeloRayosX, self).__init__()
        self.conv1 = nn.Conv2d(3, 16, kernel_size=3, padding=1)
        self.conv2 = nn.Conv2d(16, 32, kernel_size=3, padding=1)
        self.conv3 = nn.Conv2d(32, 64, kernel_size=3, padding=1)
        self.pool = nn.MaxPool2d(2, 2)
        self.fc1 = nn.Linear(64 * 16 * 16, 128)
        self.fc2 = nn.Linear(128, 2)

    def forward(self, x):
        x = self.pool(torch.relu(self.conv1(x)))
        x = self.pool(torch.relu(self.conv2(x)))
        x = self.pool(torch.relu(self.conv3(x)))
        x = x.view(-1, 64 * 16 * 16)
        x = torch.relu(self.fc1(x))
        return self.fc2(x)
```

Diagrama simplificado de la arquitectura CNN

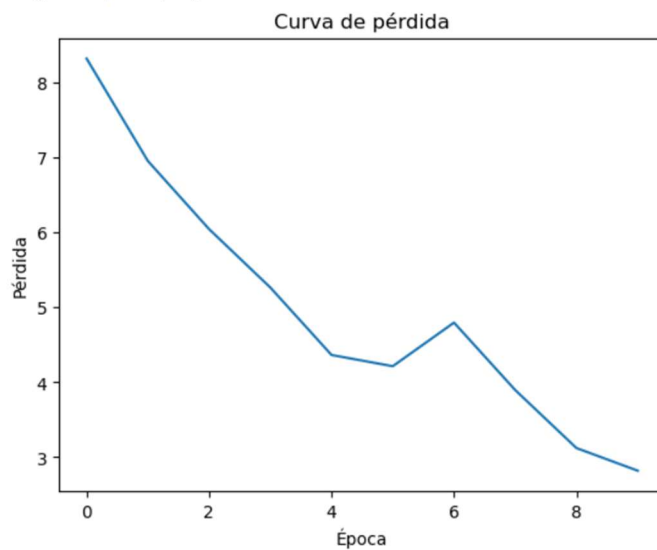


Entrenamiento del modelo

Se entrenó con la función de pérdida de entropía cruzada y el optimizador Adam. Durante cada época se calculaba la pérdida total y se ajustaban los pesos de la red.

```
def entrenar(modelo, cargador, criterio, optimizador, epocas=10):
    historial = []
    for epoca in range(epocas):
        modelo.train()
        perdida_total = 0
        for imagenes, etiquetas in cargador:
            imagenes, etiquetas = imagenes.to(dispositivo),
            etiquetas.to(dispositivo)
            optimizador.zero_grad()
            salida = modelo(imagenes)
            perdida = criterio(salida, etiquetas)
            perdida.backward()
            optimizador.step()
            perdida_total += perdida.item()
        historial.append(perdida_total)
        print(f"Época {epoca+1}/{epocas}, pérdida:
        {perdida_total:.4f}")
    return historial
```

Elige una opción (1-5): 3



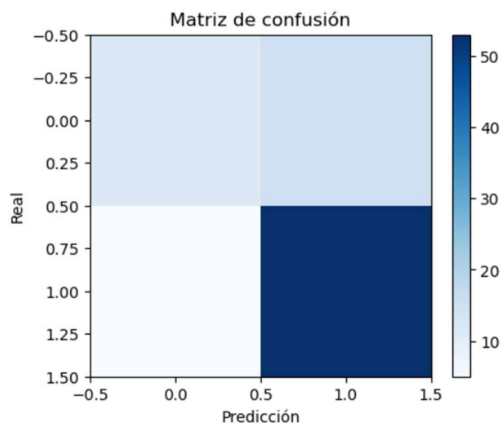
Evaluación del modelo

Al evaluar el modelo con imágenes nuevas, la precisión rondó el 70–75 %, lo que muestra que el sistema aprendió a diferenciar fracturas de no fracturas.

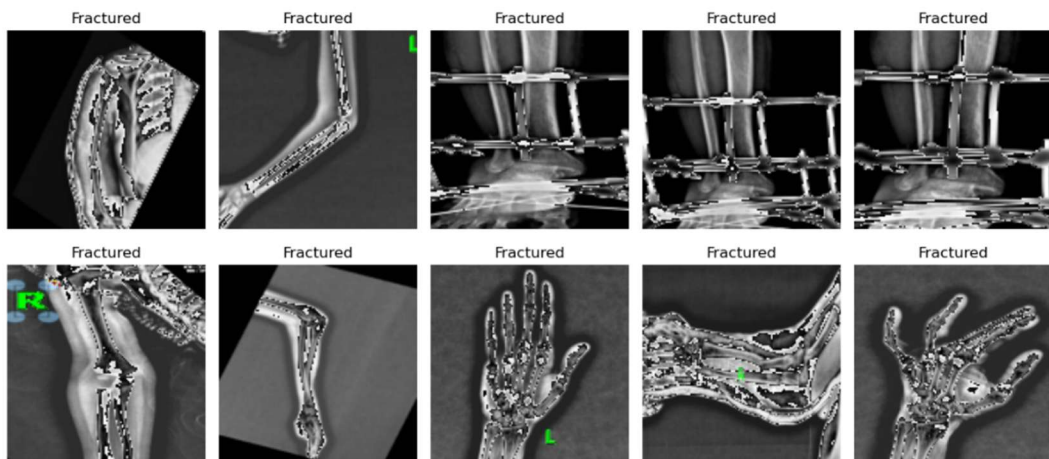
```
def evaluar(modelo, cargador):  
    modelo.eval()  
    correctos, total = 0, 0  
    with torch.no_grad():  
        for imagenes, etiquetas in cargador:  
            imagenes, etiquetas = imagenes.to(dispositivo),  
            etiquetas.to(dispositivo)  
            salida = modelo(imagenes)  
            _, pred = torch.max(salida, 1)  
            total += etiquetas.size(0)  
            correctos += (pred == etiquetas).sum().item()  
    print(f"Precisión en prueba: {100 * correctos / total:.2f}%")
```

Reporte de clasificación:

	precision	recall	f1-score	support
0	0.69	0.42	0.52	26
1	0.78	0.91	0.84	58
accuracy			0.76	84
macro avg	0.73	0.67	0.68	84
weighted avg	0.75	0.76	0.74	84



Ejemplos de radiografías del dataset



EVALUACION Y REFLEXION

El modelo que hemos montado para distinguir radiografías de huesos sanos y fracturados ha mostrado un rendimiento bastante bueno, aunque no perfecto. Durante las pruebas, vimos que la red neuronal era capaz de reconocer patrones claros en la mayoría de los casos, pero también cometía errores, sobre todo con imágenes donde la fractura era muy leve o difícil de distinguir a simple vista.

Una de las primeras evaluaciones que hicimos fue la curva de pérdida durante el entrenamiento. En ella se observaba cómo la red iba aprendiendo de forma progresiva, aunque con altibajos típicos de los datos que no están del todo equilibrados.

```
plt.plot(historial)
plt.xlabel("Época")
plt.ylabel("Pérdida")
plt.title("Curva de pérdida durante el entrenamiento")
plt.show()
```

La gráfica mostraba que después de unas pocas épocas la pérdida ya bajaba bastante, señal de que la red estaba captando bien los patrones. Sin embargo, si alargábamos demasiado el entrenamiento, empezaba a memorizar demasiado algunos ejemplos (sobreajuste).

Al evaluar el modelo en el conjunto de prueba, la matriz de confusión nos dio una idea más clara de dónde se equivocaba. Los fallos más comunes estaban en confundir huesos fracturados muy pequeños con huesos sanos, algo lógico si pensamos que incluso para un humano sin formación médica no es tan fácil detectarlos.

```
from sklearn.metrics import confusion_matrix
import seaborn as sns

matriz = confusion_matrix(etiquetas, predicciones)
sns.heatmap(matriz, annot=True, fmt="d", cmap="Blues")
plt.xlabel("Predicción")
plt.ylabel("Real")
plt.title("Matriz de confusión")
plt.show()
```

En cuanto a métricas, la precisión y el F1 macro fueron buenos, lo cual significa que no solo acertaba en general, sino que también mantenía un equilibrio aceptable entre las dos clases. No obstante, la sensibilidad en fracturas era algo menor, y esto es crítico porque en un contexto médico lo más peligroso es no detectar una fractura que sí existe.

Exploración y análisis de modelos generativos en visión artificial

La validación cruzada con distintos pliegues reforzó esta idea: el modelo era estable, no dependía solo de un conjunto de imágenes, pero los mismos errores se repetían en varios pliegues. Eso nos indica que no se trata de mala suerte en un lote concreto de datos, sino de una limitación real del modelo.

Limitaciones encontradas

El dataset es relativamente pequeño, y eso hace que el modelo no generalice tan bien en casos raros o fracturas muy finas.

Las imágenes vienen de una única fuente, por lo que no cubren todas las variaciones que podríamos encontrar en radiografías reales (calidad, ruido, contraste, etc.).

El modelo es bastante simple en comparación con arquitecturas más potentes como ResNet o EfficientNet, lo que limita su capacidad de representación.

Posibles mejoras

Aumentar el dataset, ya sea con más imágenes reales o con técnicas de aumento más variadas.

Probar arquitecturas preentrenadas (transfer learning) que ya han aprendido a reconocer patrones generales en imágenes.

Ajustar mejor los hiperparámetros, como la tasa de aprendizaje o el número de filtros en las capas convolucionales.

Trabajar con imágenes en mayor resolución para no perder detalles finos en las fracturas.

Reflexión ética

Este tipo de modelos no puede sustituir a un médico ni tomarse como diagnóstico definitivo. Puede servir como apoyo para señalar posibles fracturas y ahorrar tiempo al radiólogo, pero nunca debería usarse de forma automática sin supervisión. Hay que tener mucho cuidado porque un falso negativo (no detectar una fractura que sí está) podría tener consecuencias graves para un paciente.

Al mismo tiempo, si se usa bien, este tipo de herramienta puede ser una gran ayuda, sobre todo en lugares donde no hay especialistas suficientes y se necesita una primera criba de imágenes. El reto está en no caer en la tentación de confiar ciegamente en el algoritmo y mantener siempre el criterio humano en la última palabra.

CONCLUSIOM

Con este trabajo he podido dar un paso más respecto al ejercicio anterior, en el que construí y evalué una red neuronal para detectar fracturas en radiografías. Allí me centré en que el modelo fuera capaz de reconocer patrones en imágenes ya existentes. Ahora, en cambio, he explorado cómo la inteligencia artificial generativa no solo analiza, sino que también puede crear y mejorar imágenes.

Los experimentos con modelos de difusión han demostrado que es posible dar más nitidez a imágenes de baja calidad, algo que en el ámbito médico puede marcar la diferencia cuando se trata de interpretar una radiografía poco clara. También se ha visto que se pueden generar imágenes nuevas a partir de texto o de bocetos, lo que abre la puerta a contar con datos sintéticos extra para entrenar otros modelos, algo muy útil cuando no se dispone de tantas imágenes reales.

Eso sí, también me he encontrado con limitaciones: el consumo de recursos, los fallos en detalles pequeños y la necesidad de ajustar bien los parámetros para que la salida sea coherente. Y, como en el trabajo anterior, vuelve a quedar claro que estos sistemas deben ser entendidos como herramientas de apoyo, no como sustitutos de la mirada experta de un médico.

En conjunto, tanto la red neuronal que entrené en el primer ejercicio como el modelo generativo de este proyecto muestran dos caras complementarias de la IA en visión artificial: una orientada a reconocer y clasificar, y otra a generar y mejorar. Juntas apuntan a un futuro donde estas tecnologías pueden aportar mucho, siempre que se usen con responsabilidad y con una reflexión ética detrás.

BIBLIOGRAFIA

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014).

Generative Adversarial Nets. NIPS Proceedings.

Dhariwal, P., & Nichol, A. (2021).

Diffusion Models Beat GANs on Image Synthesis.

Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A.,

Aitken, A., Tejani, A., Totz, J., Wang, Z., & Shi, W. (2017).

Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial

Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., Qiao, Y., & Loy, C. C. (2021).

Real-ESRGAN: Training Real-World Blind Super-Resolution with Pure Synthetic

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016).

You Only Look Once: Unified, Real-Time Object Detection (YOLO).

Ronneberger, O., Fischer, P., & Brox, T. (2015).

U-Net: Convolutional Networks for Biomedical Image Segmentation.

Chaudhary, A., & Suri, J. S. (2022).

Artificial Intelligence in Bone Fracture Detection: A Review. *Journal of Imaging*, 8(6), 143.

Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A., Ciompi, F., Ghafoorian, M., van der Laak, J. A., van Ginneken, B., & Sánchez, C. I. (2017).

A survey on deep learning in medical image analysis. *Medical Image Analysis*, 42, 60-88.