1. Resumen del Proyecto

El objetivo es entrenar una Red Neuronal Convolucional (CNN) para detectar cáncer de piel y, posteriormente, **garantizar matemáticamente** su fiabilidad clínica. A diferencia de la validación tradicional (test accuracy), utilizaremos **Verificación Formal** para probar tres propiedades críticas de seguridad (Safety Properties):

1. **Robustez al Sensor:** Ruido L_inf (Cámara de baja calidad).
2. **Robustez Ambiental:** Cambios de Iluminación/Contraste (Condiciones de luz variables).
3. **Robustez Geométrica:** Invarianza a la Rotación (Orientación del dermatoscopio) -> *Puede ser bastante original*

---

2. Paso a Paso

FASE 1: Preparación del Entorno y Datos

- **Dataset:** Descargar **HAM10000** (imágenes de nevos y melanomas).
- **Preprocesamiento:**
    - Redimensionar las imágenes a **32x32** o **48x48** píxeles. (Fundamental para que la verificación no tarde días). → No sé si está hecho ya
    - Normalizar los tensores (valores entre 0 y 1).
- **Split:** Dividir en 80% Entrenamiento y 20% Test.

FASE 2: Entrenamiento del Modelo

- **Arquitectura:** Diseñar una CNN compacta en PyTorch → La tenemos!!!
- **Entrenamiento:** Entrenar hasta obtener una precisión decente (>80%).
- **Selección para Verificación:**
    - Pasar el set de Test por el modelo.
    - Seleccionar **20 imágenes** (10 benignas, 10 malignas) que el modelo clasifique **correctamente**.
    - Guardar estas 20 imágenes en un archivo .npy (NumPy) para alimentarlas a los verificadores.

FASE 3: Verificación

Ejecutamos los 3 experimentos principales usando las herramientas.

A. Experimento 1: Ruido de Cámara

- **Herramienta:** alpha-beta-CROWN (mediante archivo .yaml).
- **Configuración:** Perturbación epsilon = 0.005 (o similar).
- **Objetivo:** Certificar que pequeños cambios en los píxeles no alteran el diagnóstico.

B. Experimento 2: Iluminación y Contraste (Bias)

- **Herramienta:** Script de Python con auto_LiRPA.
- **Configuración:** Perturbación global (sumar un valor delta a todos los canales).
- **Objetivo:** Certificar que el modelo funciona igual con luz tenue o luz fuerte.

C. Experimento 3: Geometría (Rotación)

- **Herramienta:** Script de Python con auto_LiRPA (usando transformaciones geométricas).
- **Configuración:** Verificar estabilidad en un rango de rotación pequeño.
- **Objetivo:** Certificar que la orientación del dermatoscopio no cambia el diagnóstico.
- Si sale *Unsafe* (falla), presentamos el contraejemplo como vulnerabilidad de la red o aplicamos *Data Augmentation* y reentrenamos.

FASE 4: Informe y Entrega

- **Redacción:** Máximo 5 páginas.
    - *Pág 1:* Introducción médica y definición del modelo.
    - *Pág 2:* Definición matemática de las 3 propiedades.
    - *Pág 3-4:* Tablas de resultados (Imágenes verificadas vs. fallidas).
    - *Pág 5:* Conclusiones (¿Es la IA segura para uso clínico?).
- **Código:** Organizado, comentado y ejecutable en Linux/Unix.

---

3. Entregables Técnicos

Al final tendremos estos archivos en nuestra carpeta:

1. train_model.py: Entrena la red y guarda model.pth.
2. prepare_data.py: Genera data_X.npy (las 20 imágenes).
3. config_noise.yaml: Configuración para CROWN (Propiedad 1).
4. verify_light.py: Script para Auto-LiRPA (Propiedad 2).
5. verify_rotation.py: Script para Auto-LiRPA (Propiedad 3).