

# Segmentación Automática de Tumores Cerebrales mediante IA

Arquitectura U-Net + ResNet50 para diagnóstico asistido por inteligencia artificial

**Autores:** Adrian Herrera, Brian Chiru

Universidad Latina de Panamá · Ingeniería Biomédica



Made with GAMMA

# El Desafío del Diagnóstico Manual



## Segmentación manual

Proceso lento y tedioso para radiólogos

## Variabilidad diagnóstica

Resultados inconsistentes entre especialistas

## Tiempo crítico

La detección temprana salva vidas



Solución Tecnológica

# Deep Learning en Imágenes Biomédicas

01

## Automatización del proceso

Segmentación píxel a píxel mediante redes neuronales

02

## Segunda opinión objetiva

Asistencia rápida y consistente para médicos

03

## Transfer Learning

Aprovechamiento de conocimiento pre-entrenado



# Metodología: Cuatro Etapas Clave



## Preprocesamiento

Normalización de imágenes MRI



## Arquitectura

U-Net + ResNet50



## Entrenamiento

Optimización con Dice Loss



## Aplicación

Software de escritorio



# Arquitectura Híbrida U-Net + ResNet50

## Encoder

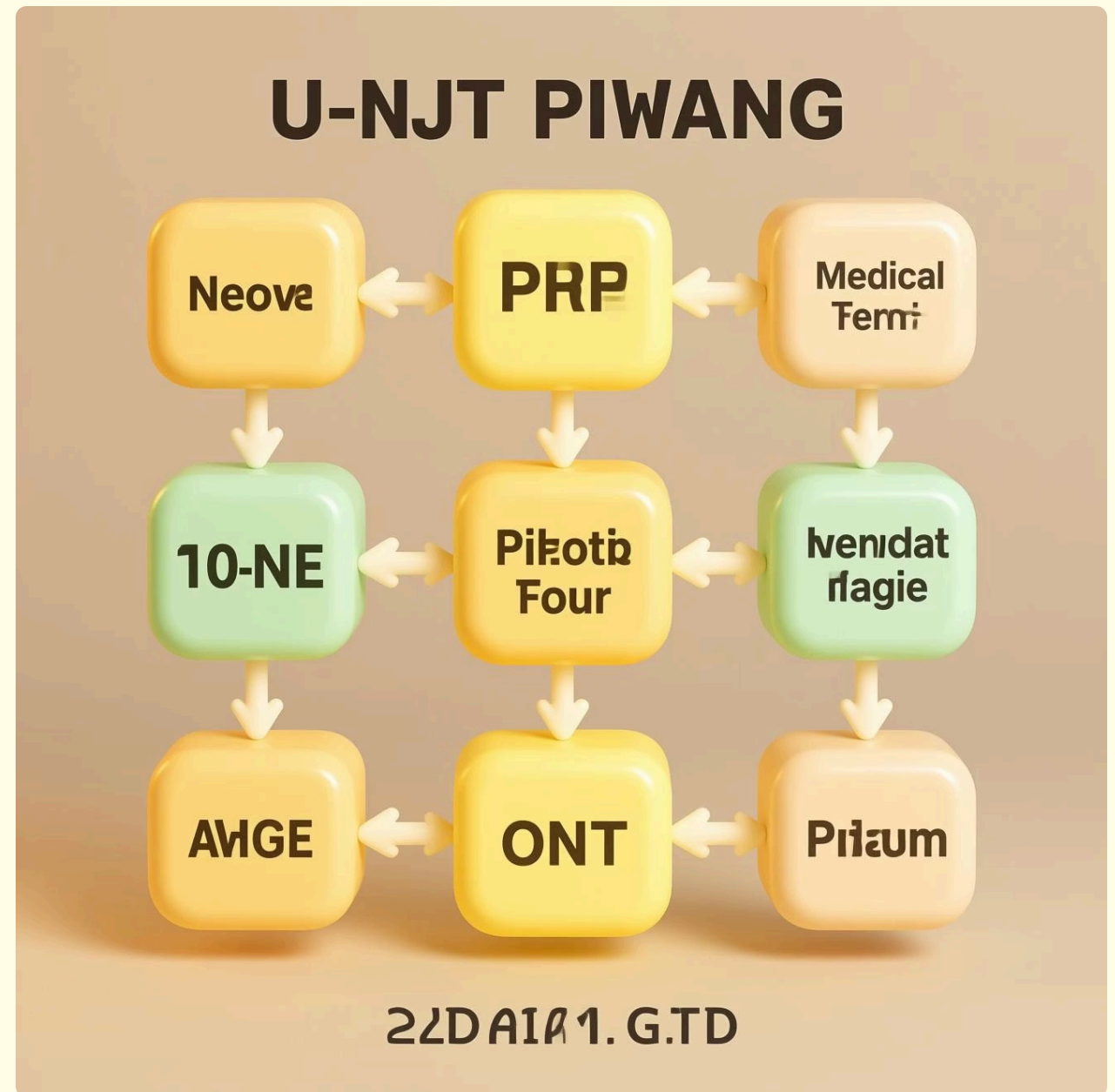
**ResNet50 pre-entrenada** con pesos ImageNet

- Extracción de características complejas
- Detección de bordes sutiles
- Transfer Learning aplicado

## Decoder

**Conv2DTranspose** para recuperar resolución

- Reconstrucción espacial
- Skip connections residuales



📄 **Resolución:** 256 x 256 píxeles · **Normalización:** Valores escalados [0, 1] · **Framework:** TensorFlow/Keras



# Resultados: Métricas de Rendimiento

99.4%

Accuracy

Exactitud global de  
clasificación píxel

0.79

Dice Score

F1-Score de  
segmentación tumoral

85%

Recall

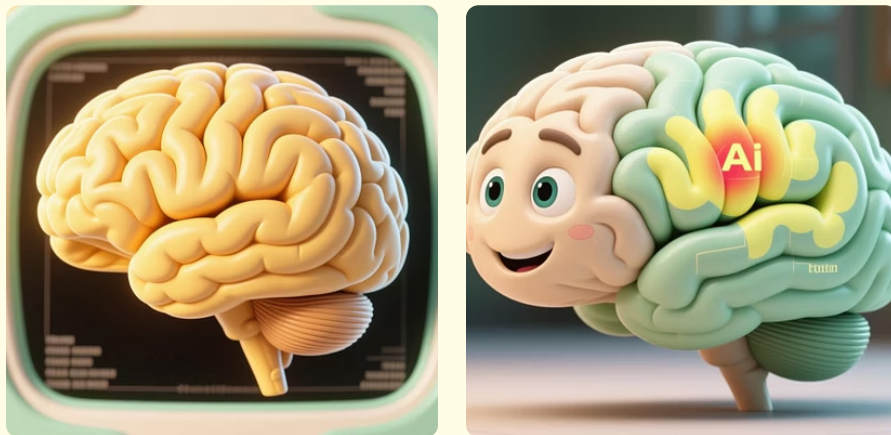
Sensibilidad en  
detección completa

0.71

IoU

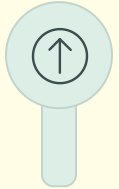
Intersección sobre  
Unión (Jaccard)

# Validación Visual: MRI vs Segmentación IA



**Imagen no incluida en entrenamiento** · Visualización comparativa con mapa de calor que resalta tejido tumoral en contraste con tejido sano

# Aplicación de Escritorio: Pipeline Técnico



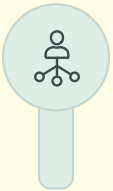
## Carga del Modelo

Inyección de función `dice_loss` personalizada



## Preprocesamiento

Redimensionamiento 256×256 y normalización [0,1]



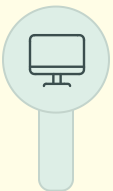
## Inferencia Neuronal

Matriz de probabilidades por píxel



## Generación Heatmap

Colorización `cv2.COLORMAP_JET` fusionada



## Visualización Comparativa

Interfaz lado a lado para validación médica



Stack Tecnológico

# Pila de Desarrollo y Herramientas



**Python 3.10**

Lenguaje base del sistema



**TensorFlow/Keras**

Carga e inferencia del modelo



**OpenCV**

Preprocesamiento de imágenes



**NumPy**

Operaciones tensoriales eficientes



**Pillow**

Gestión de formatos y rendering

# Conclusiones y Pasos Futuros

## Logros del Proyecto

- **Sensibilidad del 85%:** crucial para no omitir regiones patológicas
- **Sistema funcional** sin hardware costoso de servidor
- **Base sólida** para apoyo a decisión clínica en Panamá

---

## Requerimientos para Uso Clínico

- Validación prospectiva con casos reales
- Aprobación de comités éticos institucionales
- Certificación regulatoria oficial
- Estudios de impacto clínico documentado



📄 **Referencias clave:** Ronneberger et al. (U-Net, 2015) · He et al. (ResNet, 2016) · Isensee et al. (nnU-Net, 2021)