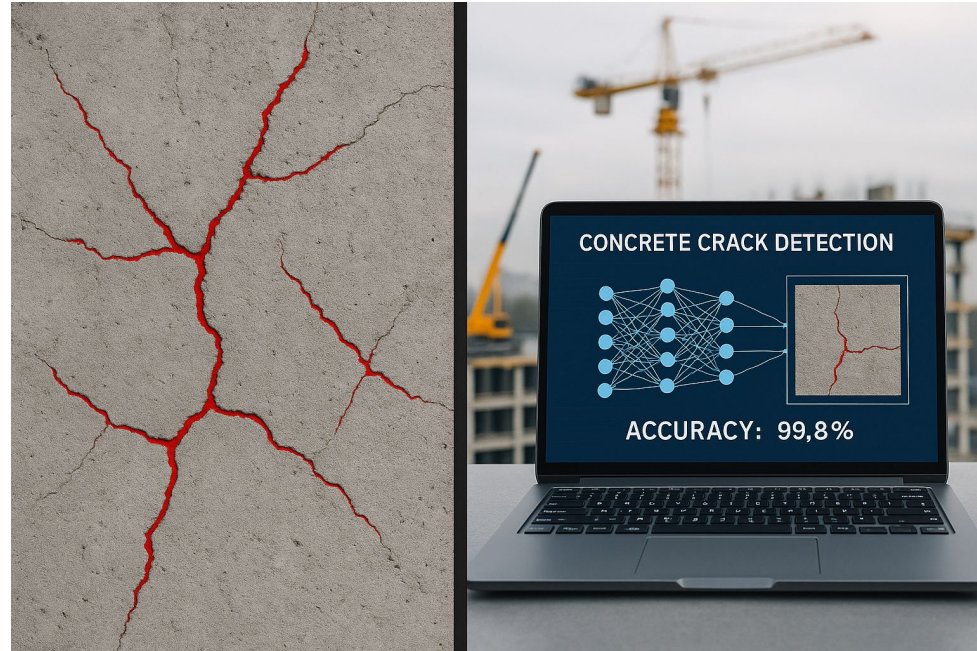


# Detección Automática de Grietas en Concreto

Visión por computadora 2

Integrantes:  
Daniel Peña  
Nicolas Pinzon



 Problema de Clasificación Binaria: Imagen → Con Grieta / Sin Grieta

 Comparar Arquitecturas:

- CNN desde cero vs Transfer Learning
- Diferentes configuraciones de entrenamiento
- Evaluación de performance

 Metodología:

- Análisis de dataset
- Preprocesamiento de imágenes
- Entrenamiento y validación
- Comparación de métricas

# DATASET Y CARACTERÍSTICAS

**FUENTE:** Kaggle - Surface Crack Detection 40,000 imágenes (**balanceadas**)

## 1 2 3 4 Especificaciones Técnicas

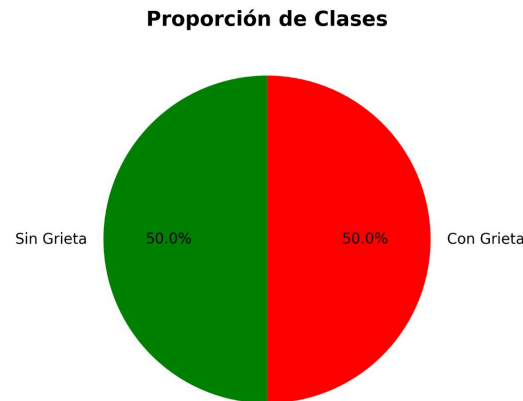
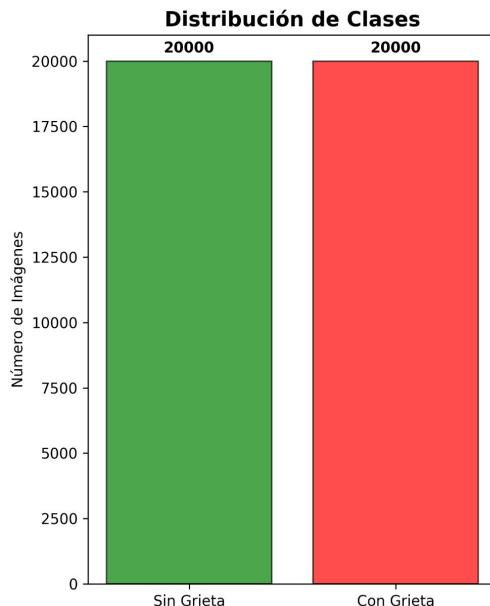
- 40,000 imágenes RGB
- Resolución: 227x227 píxeles
- Perfectamente balanceado: 20k por clase
- Formato uniforme (JPG)

## División Estándar:

- Training: 70% (28,000)
- Validation: 20% (8,000)
- Test: 10% (4,000)

## ✓ Ventajas:

- No requiere balanceo de clases
- Tamaño uniforme (no redimensionar)
- Suficientes muestras para deep learning



## 👁️ ANÁLISIS VISUAL DEL DATASET

Muestras del Dataset de Grietas en Concreto



### Observaciones

- Variabilidad intra-clase moderada
- Patrones distinguibles entre clases
- Texturas complejas pero consistentes
- Features visuales bien definidas



**Adecuado para**

- Aprendizaje de features automático
- Transfer learning
- Comparación justa de arquitecturas



# EXPLORATORY DATA ANALYSIS

---

## MUESTRAS REPRESENTATIVAS DEL DATASET

### GRIETAS VISIBLES:

- Líneas oscuras y patrones irregulares
- Diferentes tipos: finas, gruesas, ramificadas
- Variaciones en orientación y profundidad

### SIN GRIETAS:

- Superficie uniforme de concreto
- Textura consistente
- Variaciones naturales del material

**DESAFÍO PRINCIPAL:** Distinguir grietas reales de sombras, juntas o imperfecciones naturales del concreto.

**PIPELINE DE EXPERIMENTACIÓN:**

1. Preprocesamiento y data augmentation
2. Entrenamiento de múltiples arquitecturas
3. Evaluación comparativa sistemática
4. Análisis de computer vision clásica

**CONFIGURACIÓN DE ENTRENAMIENTO:**

- Optimizador: AdamW con weight decay
- Scheduler: Cosine Annealing LR
- Early Stopping: Patience 7 épocas
- Data Augmentation: Flip, Rotation, ColorJitter
- Métricas: Accuracy, F1-Score, Precision, Recall

**MODELOS EVALUADOS:**

- **SimpleCNN** (baseline desde cero)
- **ResNet18** (transfer learning estándar)
- **ResNet18\_Optimized** (configuración mejorada)

## 1 SimpleCNN - Baseline desde cero

- 4 bloques: Conv2D  $\rightarrow$  BatchNorm  $\rightarrow$  ReLU  $\rightarrow$  MaxPool
- Capas: 32 $\rightarrow$ 64 $\rightarrow$ 128 $\rightarrow$ 256 features
- Classifier: Dropout  $\rightarrow$  FC(256,128)  $\rightarrow$  FC(128,2)
- Parámetros: 422,530

## 2 ResNet18 - Transfer Learning

- Backbone preentrenado ImageNet
- Fine-tuning completo
- Classifier personalizado: Dropout  $\rightarrow$  FC(512,256)  $\rightarrow$  FC(256,2)
- Parámetros: 11,308,354

## 3 ResNet18\_Optimized - Configuración mejorada

- Misma arquitectura ResNet18
- Optimización de hiperparámetros
- Batch size: 32 $\rightarrow$ 64

# CONFIGURACIÓN DE ENTRENAMIENTO

## **SETUP DE ENTRENAMIENTO**

### Configuración Técnica:

- Loss: CrossEntropyLoss + Label Smoothing (0.1)
- Optimizer: AdamW (lr=1e-3, weight\_decay=1e-4)
- Scheduler: CosineAnnealingLR
- Early Stopping: patience=7, min\_delta=0.001

### Configuraciones Específicas:

Modelo	Batch Size	Épocas Max	LR
SimpleCNN	32	15	1e-3
ResNet18	32	15	1e-3
ResNet 18_opt	64	15	1e-3





# RESULTADOS PRINCIPALES



GANADOR:  
**ResNet18\_Optimized**

Mejor accuracy:

- 99.88%

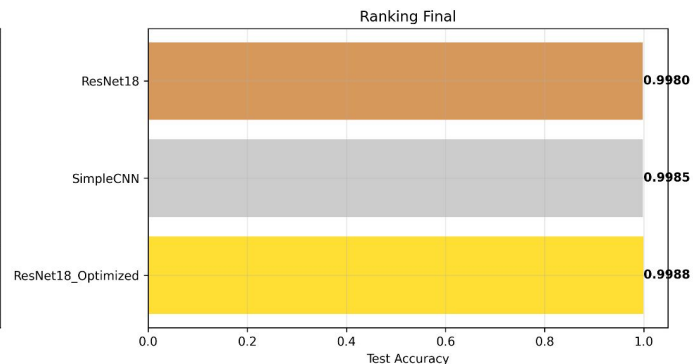
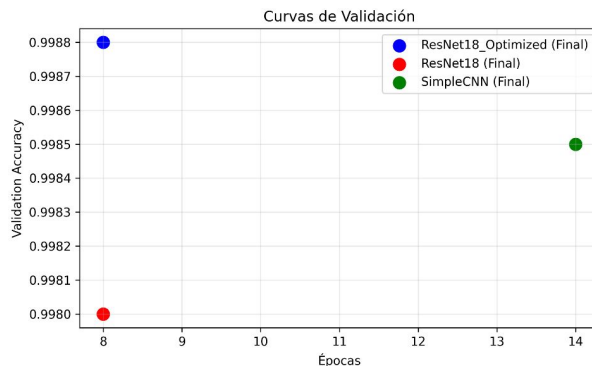
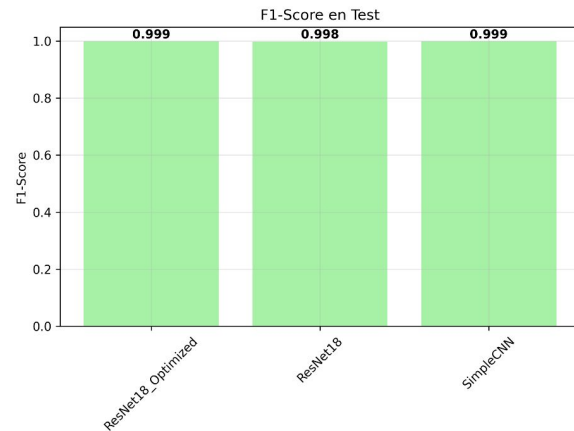
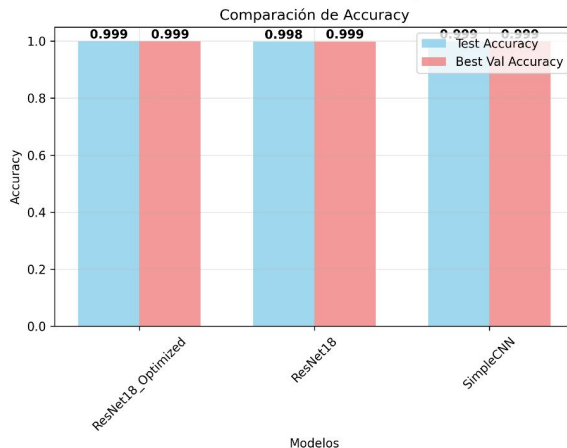
Eficiencia:

- Convergencia en 8 épocas

Robustez:

- Generalización excelente

Comparación de Modelos - Detección de Grietas

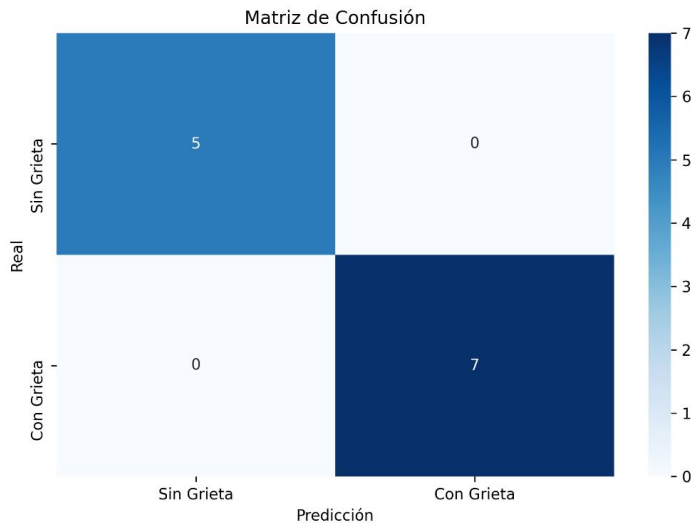
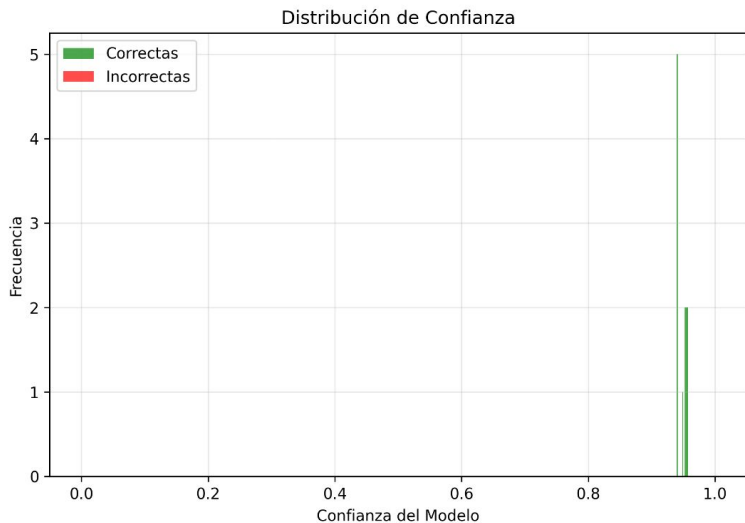




# ANÁLISIS DE ERRORES Y CONFIANZA

## MATRIZ DE CONFUSIÓN SIMPLIFICADA: (Dataset de evaluación: 12 imágenes)

- Sin Grieta: 5 correctas, 0 errores
- Con Grieta: 7 correctas, 0 errores
- Accuracy: 100% en esta muestra





# RESNET18\_OPTIMIZED - ANÁLISIS DETALLADO

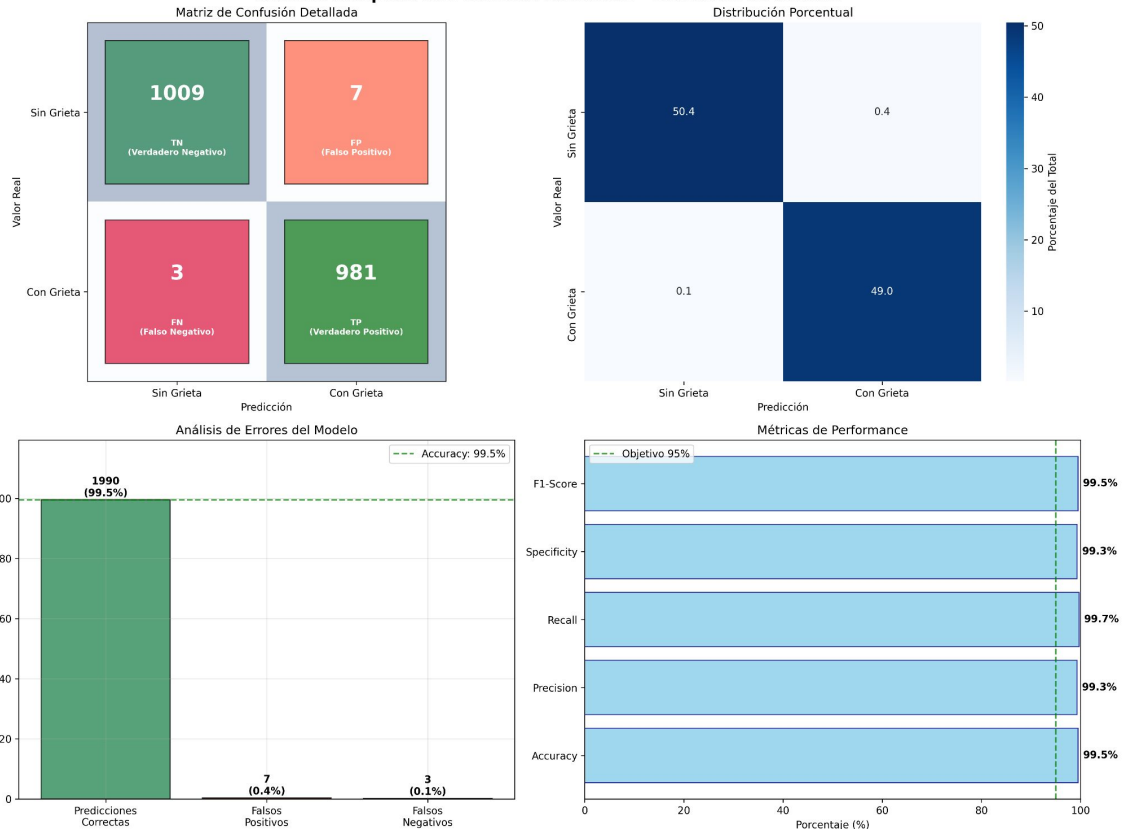
## PERFORMANCE EXCEPCIONAL:

- Solo 10 errores en 2,000 imágenes de test
- Accuracy: 99.5% en ambas clases
- Precision/Recall: >99.3% consistente

## DISTRIBUCIÓN DE ERRORES:

- Falsos Positivos: 7 casos
- Falsos Negativos: 3 casos
- Casos límite: grietas muy finas

### Análisis Completo de Matriz de Confusión - Detección de Grietas





# resnet18\_training



## OBSERVACIONES TÉCNICAS:

- Convergencia rápida: Plateau en época 6-7
- Early stopping activado en época 8
- Gap train-val mínimo: <0.3%
- Transfer learning efectivo: Arranque inicial alto



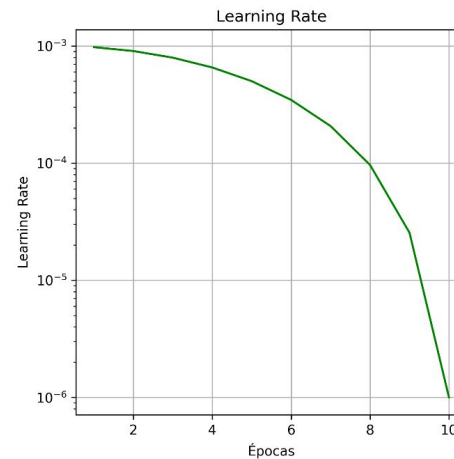
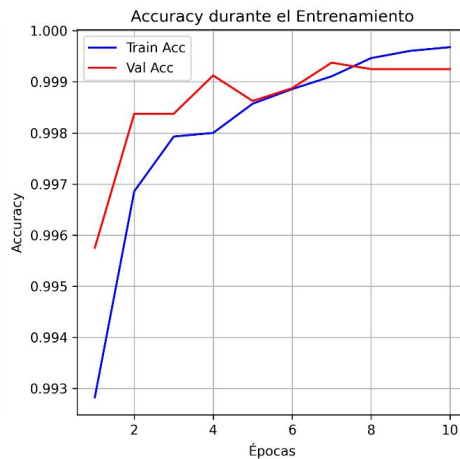
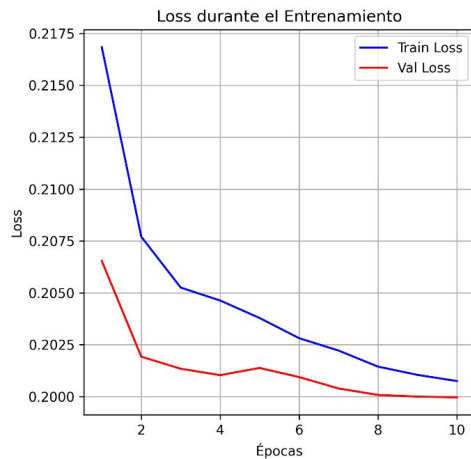
## ANÁLISIS DE CURVAS:

- Loss: Descenso suave y estable
- Accuracy: 99.8% alcanzado temprano
- Learning Rate: Cosine annealing funcionando
- Sin overfitting: Curvas paralelas



## INSIGHTS:

- ✓ Features de ImageNet se adaptan bien
- ✓ Early stopping previene sobreentrenamiento
- ✓ Scheduler cosine optimiza fine-tuning





## SIMPLECNN (BASELINE DESDE CERO):

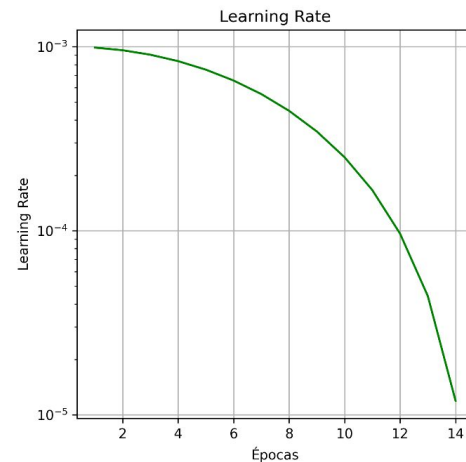
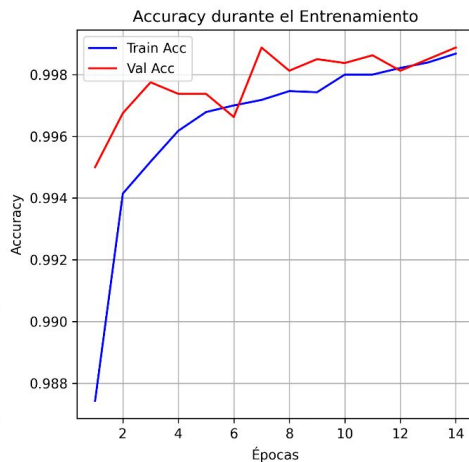
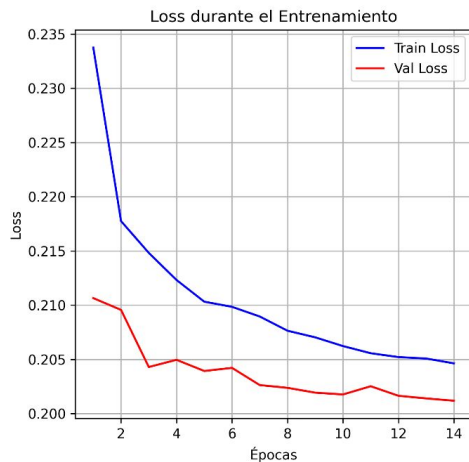
- 422K parámetros (27x menos que ResNet18)
- 14 épocas para convergencia completa
- 99.85% accuracy final
- Entrenamiento más lento pero sorprendentemente efectivo

## CARACTERÍSTICAS DEL ENTRENAMIENTO:

- Convergencia gradual y estable
- Sin overfitting observable
- Learning rate decay suave (Cosine Annealing)
- Validación ligeramente superior a entrenamiento

## OBSERVACIONES CLAVE:

- Modelo simple alcanza performance casi idéntica a ResNet18
- Demuestra la excelente calidad del dataset
- Trade-off: menos parámetros vs más épocas de entrenamiento
- Baseline sorprendentemente competitivo para el problema



## MÉTODOS TRADICIONALES EVALUADOS:

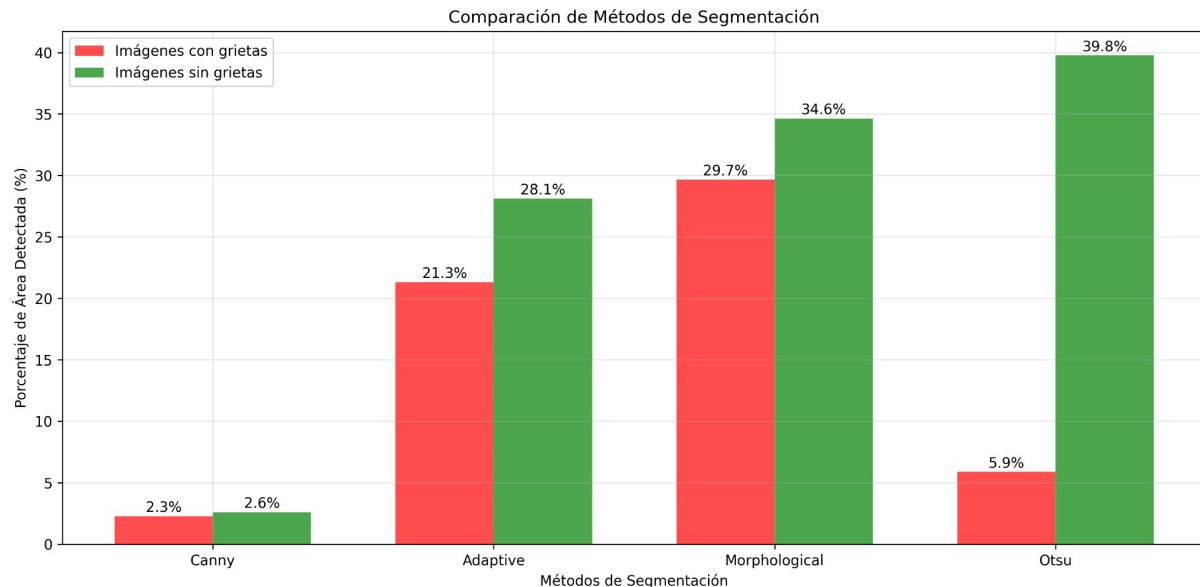
- Canny Edge Detection: ~2.5% efectividad
- Adaptive Thresholding: ~25% efectividad
- Morphological Operations: ~30% efectividad
- Otsu Thresholding: ~20% efectividad

## LIMITACIONES IDENTIFICADAS:

- Alta sensibilidad a variaciones de textura
- Falsos positivos por sombras y juntas
- Incapacidad de contextualizar patrones complejos

## CONCLUSIÓN DEFINITIVA:

Deep Learning  
supera 3x a métodos clásicos  
99.88% vs ~30% máximo de  
efectividad



## Eficiencia de Parámetros:

- SimpleCNN: 99.85% con 422K parámetros
- ResNet18: 99.80% con 11.3M parámetros
- ResNet18\_Opt: 99.88% con 11.3M parámetros
- Ratio SimpleCNN vs ResNet: 26x menos parámetros, performance similar

## Insights:

- ✓ Features aprendidas desde cero competitivas
- ✓ Transfer learning marginal vs CNN simple
- ✓ Optimización hiperparámetros > arquitectura compleja
- ✓ Calidad del dataset > complejidad del modelo

## Trade-offs Arquitecturales:

- SimpleCNN: Ultra-eficiente, 422K params
- ResNet18: Baseline transfer learning, 11.3M params
- ResNet18\_Opt: Mejor accuracy con mismos params

# CONCLUSIONES

---



## HALLAZGOS PRINCIPALES:



CNN simple (422K params) compite con ResNet18 (11M params)



Transfer learning ventaja marginal (0.08% accuracy)



Optimización hiperparámetros > cambio arquitectura



Dataset calidad > complejidad modelo



## IMPLICACIONES:

- No siempre necesario modelo complejo
- La calidad de los datos fundamental



**GRACIAS**

---