

Proyecto 4

Transferencia de Estilo Neural con Estilo Monet

[Repositorio del Proyecto](#)

21016 Javier Chavez
21085 Andres Quezada
21600 Javier Ramirez
21631 Mario Cristales

Guatemala, 06 de junio de 2025

Introducción

La Transferencia de Estilo Neural (*Neural Style Transfer*, NST) es una técnica del campo del aprendizaje profundo que permite aplicar el estilo visual de una imagen (típicamente una pintura) al contenido de otra (como una fotografía). Esta técnica ha ganado popularidad por su capacidad para generar imágenes visualmente impactantes, combinando el contenido semántico de una escena real con las características estilísticas de obras de arte reconocidas.

En este proyecto se emplea un modelo preentrenado de red neuronal convolucional GG-19 para capturar y combinar de manera diferenciada el contenido de una fotografía con el estilo de una obra del pintor impresionista Claude Monet. Monet es conocido por sus trazos suaves, su uso de la luz natural y sus colores vibrantes, por lo que es un candidato ideal para este tipo de experimentos artísticos computacionales.

El objetivo general del proyecto es generar nuevas imágenes que mantengan la estructura y composición espacial del contenido original, pero que adopten las características texturales, cromáticas y compositivas propias del estilo artístico de Monet. Para lograrlo, se diseñó un pipeline en PyTorch que procesa ambas imágenes, extrae sus representaciones internas a través de la red neuronal, y optimiza una tercera imagen (inicialmente una copia del contenido) hasta lograr una fusión coherente entre ambos dominios visuales.

Esta técnica no solo permite explorar el cruce entre arte y tecnología, sino que también tiene aplicaciones potenciales en áreas como el diseño gráfico automatizado, la creación de contenido artístico digital y la síntesis de imágenes creativas para medios interactivos.

Objetivos

- Desarrollar una herramienta funcional para aplicar Transferencia de Estilo Neural.
- Utilizar una red convolucional preentrenada (VGG-19) para extraer características semánticas y estilísticas de imágenes.
- Implementar pérdidas específicas de estilo, contenido y suavidad (TV loss).
- Generar imágenes que combinen el contenido de una imagen base con el estilo de otra (Monet).

Fundamento Teórico

¿Qué es Transferencia de Estilo Neural?

NST es una técnica propuesta por Gatys et al. en 2015. Dado:

- Una imagen de contenido C , que representa la estructura general que se quiere conservar.
- Una imagen de estilo S , que representa el patrón visual (colores, texturas, pinceladas) que se desea aplicar.
- Se optimiza una tercera imagen G (generada), que busca tener el contenido de C y el estilo de S .

¿Por qué usar un modelo preentrenado?

Usamos VGG-19 preentrenado por varias razones:

- Fue entrenado en ImageNet, un dataset con más de 14 millones de imágenes.
- Ha aprendido a reconocer patrones visuales complejos como bordes, formas, texturas y objetos.
- Sus capas profundas abstraen información de alto nivel (semántica), y las capas superficiales capturan información de bajo nivel (estilo).
- No necesitamos reentrenar el modelo. Lo usamos como extractor de características (feature extractor), lo que reduce significativamente el coste computacional y el tiempo de desarrollo.

¿Cómo fue entrenado VGG-19?

VGG-19 es una red convolucional profunda desarrollada por el Visual Geometry Group de la Universidad de Oxford. Fue entrenada en el conjunto de datos ImageNet, que contiene más de 1.2 millones de imágenes clasificadas en 1000 categorías. El entrenamiento utilizó backpropagation, con técnicas como data augmentation, regularización L2, ReLU como función de activación y un optimizador SGD con *learning rate decay*.

Esta red está compuesta por capas convolucionales de pequeño filtro (3x3), seguidas de capas de pooling (reducción de resolución), y finaliza con capas totalmente conectadas (no utilizadas en este caso).

Arquitectura y Funcionalidad del Script

Flujo General:

1. Carga de imágenes: Se cargan la imagen de contenido (fotografía) y la de estilo (pintura de Monet).
2. Preprocesamiento: Se escalan, normalizan y transforman a tensores compatibles con PyTorch.
3. Modelo: Se construye un modelo secuencial usando las capas de VGG-19 hasta ciertas capas específicas.
4. Cálculo de pérdidas:
 - a. Contenido: Se compara una capa profunda del modelo entre la imagen generada y la de contenido.
 - b. Estilo: Se comparan varias capas poco profundas entre la imagen generada y la de estilo mediante matrices de Gram.
 - c. Regularización TV: Reduce artefactos pixelados manteniendo la suavidad.
5. Optimización: Se actualiza la imagen generada, no el modelo. Se usa el optimizador LBFGS.
6. Resultados: Se guarda el resultado cada 50 iteraciones para monitoreo visual.

Componentes Técnicos

Preprocesamiento:

transforms.Resize, transforms.ToTensor, transforms.Normalize

Normaliza las imágenes a la media y desviación estándar usadas durante el entrenamiento de VGG-19.

Selección de Capas

STYLE_LAYERS = ["conv1_1", "conv2_1", "conv3_1", "conv4_1", "conv5_1"]

CONTENT_LAYERS = ["conv4_2"]

- Las capas de estilo capturan diferentes niveles de textura y color.
- La capa conv4_2 captura la semántica o estructura de la imagen.

Funciones de Pérdida

- Contenido: Diferencia MSE entre activaciones de la imagen generada y la imagen de contenido.
- Estilo: Se calcula la matriz de Gram para capturar correlaciones entre filtros. Luego se aplica MSE entre las matrices de Gram del estilo y la generada.
- TV Loss (Total Variation): Promueve suavidad, eliminando ruido entre píxeles contiguos.

*loss = style_w * s_loss + content_w * c_loss + tv_w * t_loss*

Donde:

- *style_w = 1e6*
- *content_w = 1*
- *tv_w = 1e-6*

Optimización

Se usa el optimizador L-BFGS, recomendado por Gatys et al., debido a su estabilidad en este tipo de problemas.

```
optimizer = optim.LBFGS([input_img.requires_grad_()], lr=0.03)
```

Guardado de Progreso

Cada `SAVE_EVERY = 50` pasos, se guarda una imagen intermedia para poder observar la evolución del proceso.

Parámetros Clave

Parámetro	Valor	Explicación
IMSIZE	512 px	Tamaño de entrada
STYLE_LAYERS	5 capas	Capas para capturar estilo visual
CONTENT_LAYERS	1 capa	Capa para preservar estructura
STYLE_WEIGHT	1e6	Peso alto para influir más el estilo
CONTENT_WEIGHT	1	Peso bajo para preservar contenido base
TV_WEIGHT	1e-6	Peso bajo para suavidad visual
NUM_STEPS	400	Iteraciones totales de optimización
LR	0.03	Learning rate del optimizador LBFGS
SAVE_EVERY	50	Frecuencia de guardado de imagen

Resultados Esperados

- Imágenes generadas que mantienen la estructura y semántica del contenido original.
- Pinceladas, colores, patrones y texturas inspiradas en el estilo de Monet.
- Evolución visible del estilo en las imágenes cada 50 iteraciones (progress_XXXX.png).
- Gráfica de pérdida que muestra la disminución progresiva de la suma total de pérdidas (contenido, estilo y variación total), reflejando la optimización y convergencia del modelo.

Resultados Obtenidos

Resultado 1:

Contenido original: Campo abierto bajo un cielo azul claro.



Estilo aplicado: Pintura de Monet caracterizada por pinceladas sueltas, texturas en las plantas y juego de verdes y azules.



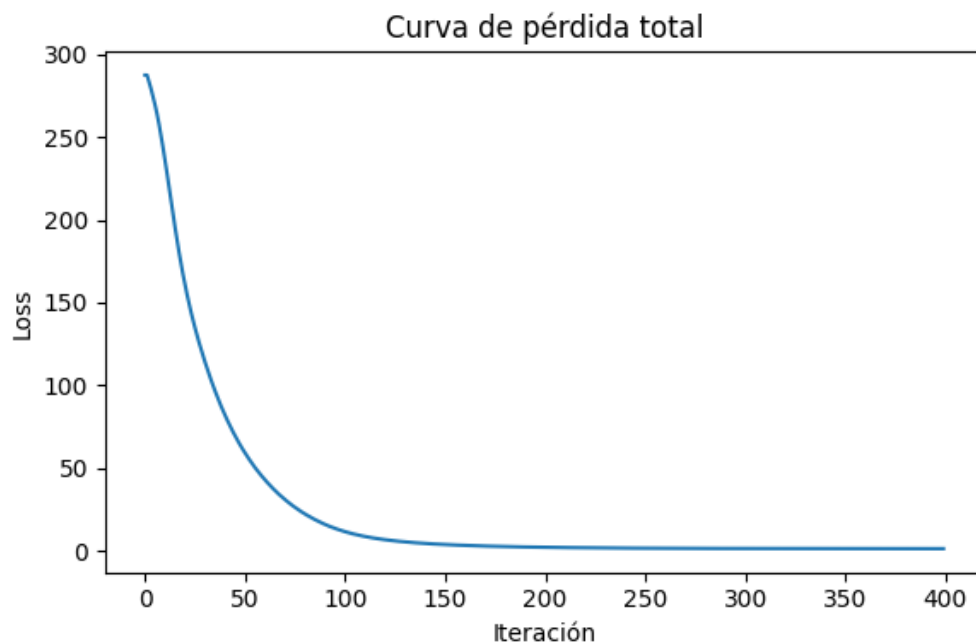
Resultado: Se logró una síntesis que mantiene la estructura del campo, mientras que el cielo muestra trazos de pincel suaves y texturas pictóricas.



Curva

de

pérdida:



La curva de pérdida muestra un descenso rápido en las primeras iteraciones y una estabilización progresiva alrededor de las 200 iteraciones, lo que indica una convergencia adecuada. Este comportamiento sugiere que el contenido de la imagen base fue preservado correctamente y que el estilo fue transferido de manera efectiva. Además, no se observan oscilaciones ni signos de sobreajuste. Los resultados visuales muestran una aplicación consistente del estilo, manteniendo la estructura original de la imagen y logrando un balance apropiado entre contenido y estilo, validando la correcta elección de los hiperparámetros.

Resultado 2:

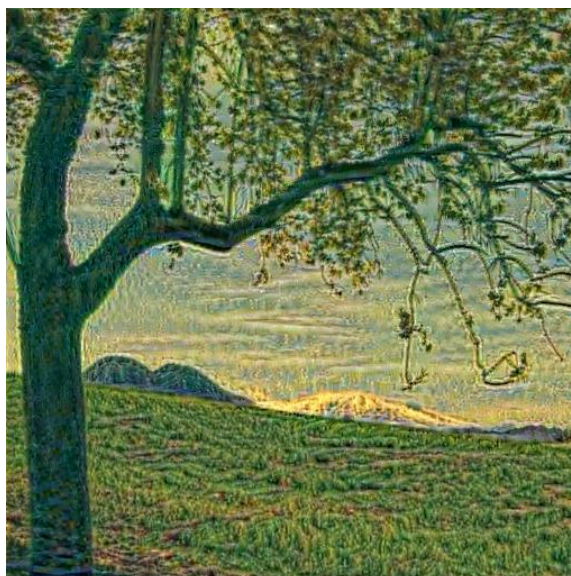
Contenido original: Primer plano de un árbol con montañas al fondo.



Estilo aplicado: Pinceladas vibrantes y colores cálidos, con bastantes tonos verdes del estilo Monet.



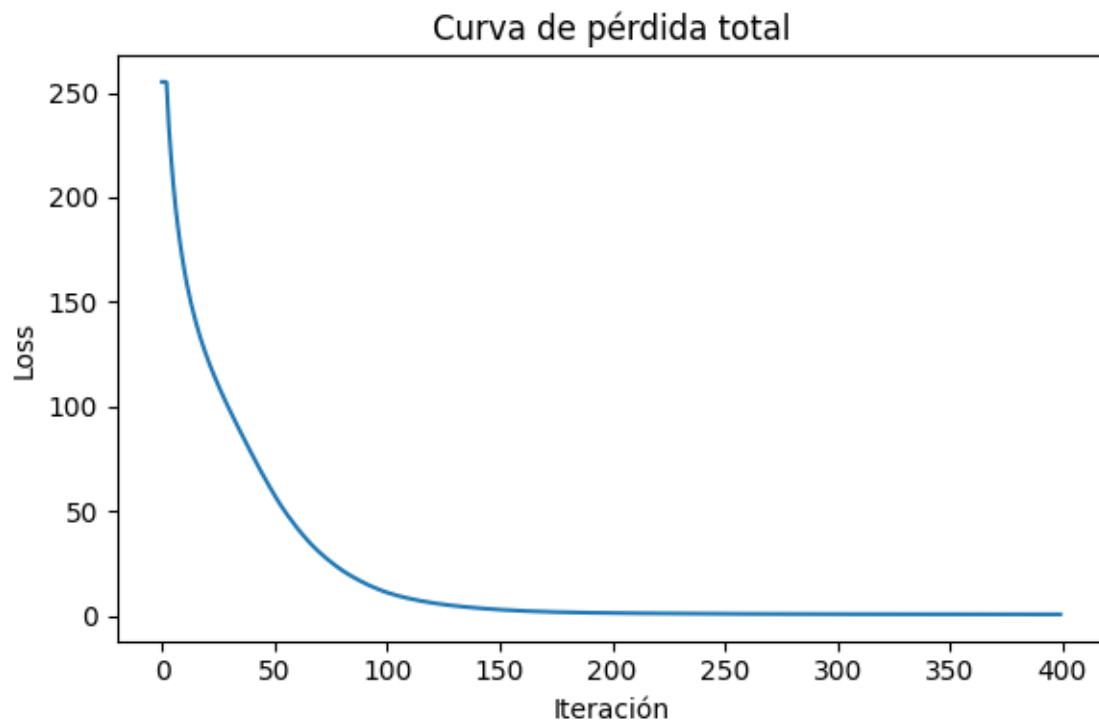
Resultado: Las ramas y la textura del árbol se funden con trazos más abstractos, y los colores amarillos y verdes del fondo evocan una atmósfera impresionista.



Curva

de

pérdida:



La gráfica muestra una disminución rápida de la pérdida total durante las primeras 100 iteraciones, lo que indica una fase inicial de aprendizaje acelerado donde el modelo ajusta rápidamente la transferencia de estilo. Después de este punto, la pérdida continúa disminuyendo de manera más gradual y se estabiliza a partir de las 200 iteraciones, mostrando que el modelo converge correctamente. Este comportamiento confirma que las pérdidas de contenido, estilo y variación total se optimizan de manera conjunta y efectiva, logrando un balance adecuado entre preservar la estructura de la imagen original y aplicar el estilo objetivo sin caídas abruptas ni sobreajuste.

Consideraciones Técnicas Adicionales

- Se reemplazaron las capas de MaxPooling por AveragePooling para lograr una suavidad más consistente.
- Las capas de activación ReLU fueron configuradas con `inplace=False` para evitar la sobrescritura del gradiente.
- El modelo VGG se usó solo hasta las capas necesarias, para evitar computación innecesaria.
- Las matrices de Gram permiten capturar correlaciones espaciales entre filtros, representando patrones de textura.

Conclusiones

Este proyecto demuestra cómo un modelo de clasificación de imágenes preentrenado puede ser reutilizado para generar arte. La Transferencia de Estilo Neural aprovecha las representaciones profundas aprendidas por redes convolucionales para combinar imágenes de formas creativas y visualmente impresionantes.

El uso de VGG-19 preentrenado permite una implementación eficiente, ya que se aprovechan sus filtros y estructuras internas como mecanismos para medir diferencias de contenido y estilo. No es necesario reentrenar el modelo, lo que facilita su uso incluso en entornos con pocos recursos computacionales.

La implementación modular, con funciones claras y estructura definida, permite adaptar el estilo a otros artistas simplemente cambiando la imagen de estilo.

Este enfoque combina arte y ciencia, y puede utilizarse en campos como el diseño gráfico, la creación de contenido, el arte digital automatizado e incluso en el cine. Los resultados obtenidos muestran una fusión equilibrada entre contenido y estilo, confirmando la efectividad del enfoque y la sensibilidad artística que puede lograr una red neuronal cuando se guía adecuadamente.

Finalmente, las gráficas de pérdida evidencian un comportamiento de optimización estable y esperado: una rápida disminución inicial seguida de una convergencia progresiva. Esto indica que el modelo logra minimizar de manera efectiva la combinación de las pérdidas de estilo, contenido y variación total, asegurando una transferencia de estilo exitosa y consistente.

Referencias

Gatys, L. A., Ecker, A. S., & Bethge, M. (2016). Image style transfer using convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 2414-2423). <https://arxiv.org/abs/1508.06576>

Simonyan, K., & Zisserman, A. (2015). *Very deep convolutional networks for large-scale image recognition*. In International Conference on Learning Representations (ICLR). <https://arxiv.org/abs/1409.1556>

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... & Chintala, S. (2019). torchvision.models.vgg19. PyTorch. <https://pytorch.org/vision/stable/models/generated/torchvision.models.vgg19.html>