

# 컴퓨터비전(AI응용)

신경망 스타일 전이

01

### NST 파이프라인 이해

Neural Style Transfer(NST)의 전체 파이프라인(콘텐츠/스타일 입력, 특징 추출기, 최적화 대상 이미지, 옵티마이저)의 구조와 데이터 흐름을 설명할 수 있다.

02

### 손실 함수 계산 원리

VGG의 특징 맵을 기반으로 콘텐츠 손실(Content Loss)과 스타일 손실(Style Loss)을 추출하는 원리를 이해한다.

03

### PyTorch 구현 및 튜닝

PyTorch에서 사전학습된 VGG19를 고정 특징 추출기로 구성하고, 하이퍼파라미터를 튜닝해 나만의 신경망 스타일 전이를 구현할 수 있다.

이미지 C를 만들기 위해 이미지 B의 스타일을 이미지 A로 옮기는 것



이미지 A

콘텐츠 (Content)

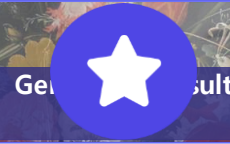
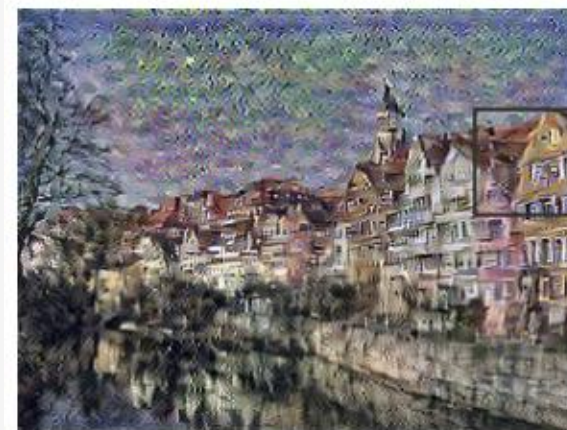
+



이미지 B

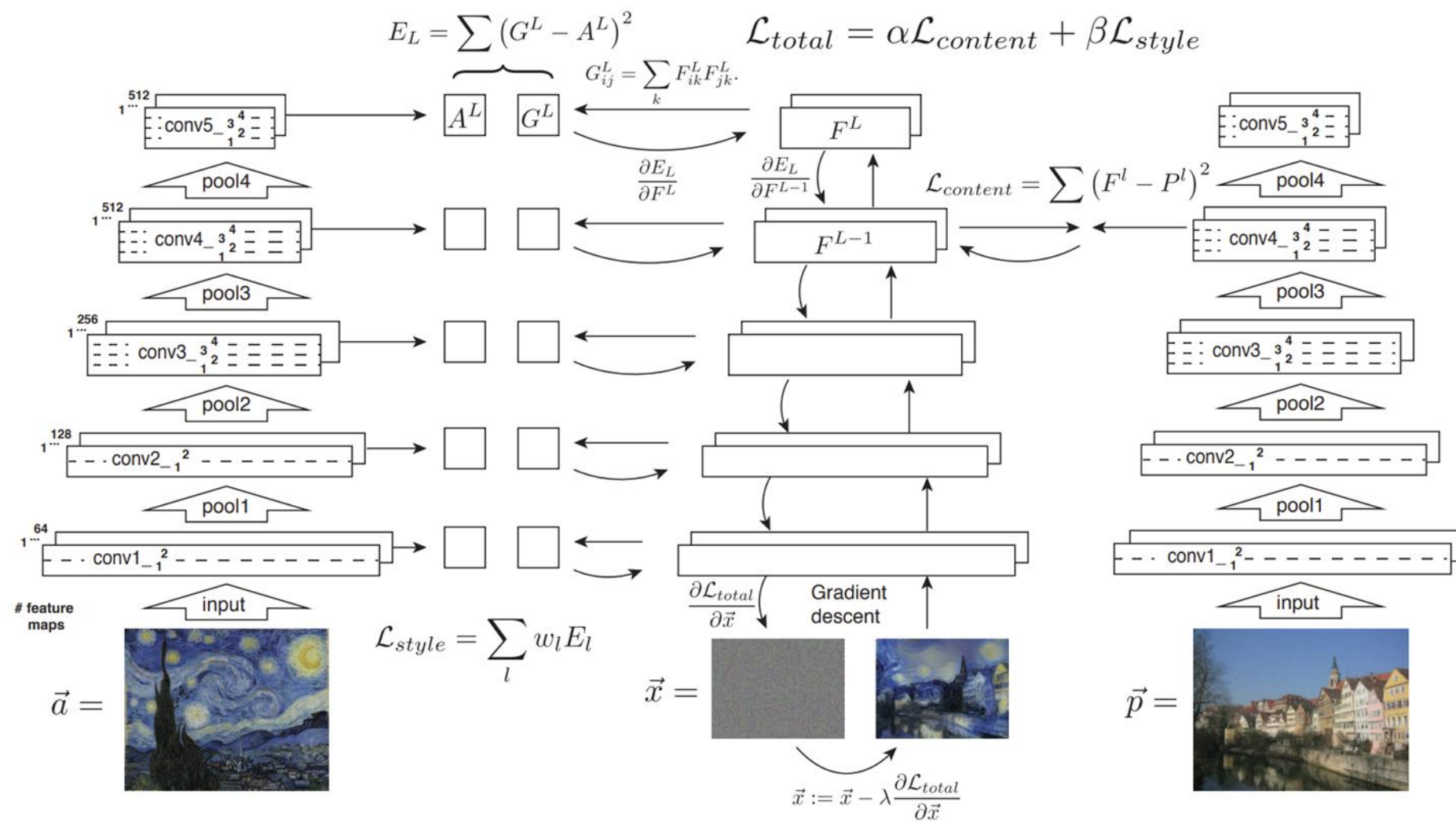
스타일 (Style)

=



이미지 C

스타일 전이 결과





$$L_{\text{content}}(\tilde{p}, \tilde{x}, l) = \frac{1}{2} \sum_{i,j} (F_{lij} - P_{lij})^2.$$



### 목적 (Objective)

입력된 이미지의 **객체(Object)**와 **배치(Layout)**에 대한 자세한 특징을 추출하는 것을 목표로 합니다.



### 시각화 방법 (Visualization)

인코딩된 이미지 정보를 시각화하기 위해 다음 과정을 수행함

흰색 노이즈 이미지(White Noise)에서 시작  
경사하강법(Gradient Descent)을 수행하여 최적화  
원본 이미지의 특징과 일치하는 이미지를 탐색

$$L_{style} = \sum_l w_l E_l \quad \text{where} \quad E_l = \frac{1}{4N_l^2 M_l^2} \sum (G_{ij}^l - A_{ij}^l)^2$$



### 목적 (Objective)

스타일 이미지의 **질감(Texture)**, **색상(Color)**, **패턴(Pattern)** 등 시각적 표현 특징을 유지하는 것을 목표로 합니다.



### 계산 원리 (Gram Matrix)

CNN의 여러 계층에서 얻은 특징 맵 간의 **상관관계(Correlation)**를 비교하여 계산합니다.

특징 간의 **상호작용**을 수치화

스타일 이미지의 패턴이 생성 이미지에 반영되도록 유도

여기서  $G_{ij}^l$ 와  $A_{ij}^l$ 는 무엇인가?

각각 생성 이미지와 스타일 이미지의 Gram Matrix (특징 간의 상호작용을 나타냄)

이 손실은 스타일 이미지의 패턴이 생성 이미지에 잘 반영되도록 한다.

## 정의 (Definition)

Gram Matrix는 주어진 행렬의 **특징 벡터(feature vector)**들 간의 **내적(dot product)**을 통해 생성된 상관관계 행렬(correlation matrix)입니다.

$$G_{ij}^l = \sum_k F_{ik}^l F_{jk}^l$$

**i번째** 특징 맵과 **j번째** 특징 맵 사이의 내적 값

### 표기 (Notation)

$F_l$ : 레이어  $l$ 의 특징 맵 (Feature Map)

$N_l$ : 필터(채널)의 개수

$M_l$ : 공간적 크기 (가로 × 세로 픽셀 수)

\*  $F_l \in \mathbb{R}^{N_l \times M_l}$



### 상관관계 해석

두 특징 맵의 값이 함께 클 때 내적 값이 커지며, 이는 해당 특징들이 **동시에 활성화**됨을 의미합니다.



### 핵심 의미

"이 이미지의 어떤 시각적 특징들이 **얼마나 자주 함께 나타나는가?**"를 수치적으로 표현합니다.