Image Style Transfer (Görüntü Stili Aktarımı)

Rabia Eda Yılmaz

Yıldız Teknik Üniversitesi, Biyomedikal Mühendisliği

Bu yazıda Gatys et. al., 2016 yazdığı "Image Style Transfer Using Convolutional Neural Networks" makalesini başlık başlık inceleyeceğiz. Daha sonra, bu modeli kullanarak birlikte sanat yapacağız. Türkçe bir kaynak olmasını istedim, yine de terimleri aslı ile yazmanın daha faydalı olacağını düşünüyorum.

Bu konu hakkında, hazırladığım study-source sayfasına bu Notion linkinden ulaşabilirsiniz:

https://muddy-theory-8d3.notion.site/Image-Style-Transfer-bac6d6ee5e484861a9a1ed31c2e47ad4

İncelediğim makaleye bu linkten ulaşabilirsiniz:

https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Gatys_Image_Style_Transfer_CVPR_2016_paper.pdf

1. Introduction

Önceki araştırmalarda kayda değer sonuçlar edinildi ama sınırlayıcı bir yönü var: Hedef resmin sadece low-level image features (üst katmanları) texture-transfer için kullanıldı.

- Bir style-transfer algoritması, hedef resimden (target image) semantic image content çıkarabilmelidir. (obje veya arkaplan gibi) ve sonrasında hedef resmin semantic content'i, kaynak resmin (source image) stili ile, texture-transfer yoluyla bilgilendirilmeli.
- Content (içerik) ve style (stil) birbirinden ayırmak için, high-level semantic information (daha derin katmanlardaki anlamsal bilgiler) Convolutional Neural Networks (CNNs) kullanıldı.
- Bu makalede, CNNs tabanlı parametreli "A Neural Algorithm of Artistic Style" modeli tanıtılıyor.

2. Deep Image Representations

- Temelini object recognition & localization (obje tanımlama ve yer saptama) için olan VGG network'ünden alıyor. Bu makalede, VGG-19 kullanıldı: 16 convolutional layers & 5 pooling layers.
- Network, ağırlıkları (weights) ölçekleyerek (scaling) normalize edildi. Ayrıca, resim üzerinde her convolution filtresinin ortalama aktivasyonları ve konumları bire eşit.
- Rescaling (yeniden ölçeklendirme), output'u değiştirmeden, VGG ağı için yapılabilir. Bunun nedeni, sadece ReLU (Rectifying Linear Activation) içermesi ve feature map üzerinde normalization ile pooling olmamasıdır.
- Ağdaki hiçbir fully-connected layer kullanılmadı.
- Max pooling'ten biraz daha iyi olması ile resim sentezi için average pooling, tercih edildi.

2.1.Content Representation

- Ağdaki her katman non-linear filtre bankası tanımlar. Karmaşıklığı, katmanın konumuna bağlı olarak artar. Bu yüzden, input image (girdi resim), $\vec{x}$, CNN'in her katmanında filter responses ile encode edilmiştir.
- Orijinal resmin feature responses(ona has özellikler) ile eşleşen diğer resmi bulmak için, white noise image (TV ekranındaki sinyal yok ekranı gibi) üzerine gradient descent uygulanıyor.
- Orijinal resimleri temsil etmeleri için $\vec{p}\ (photo)\ ve\ \vec{x}$ (white noise olan) kullanalım ve P^l ile F^l , sırasıyla, l katmanındaki (üslerinde layer kelimesinin baş harfi var) feature representations olsun. O zaman, squared-error loss (karesi alınmış hata kaybı):

$$\mathcal{L}_{content}(\vec{p}, \vec{x}, l) = \frac{1}{2}\sum_{ij}\left(F_{ij}^l - P_{ij}^l\right)$$

Bu ifadenin l katmanındaki aktivasyonlarına bağlı olarak türevi ile:

$$\frac{\partial \mathcal{L}_{content}}{\partial F_{ij}^l} = \begin{cases}\left(F_{ij}^l - P_{ij}^l\right) & if\ F_{ij}^l > 0 \\ 0 & if\ F_{ij}^l < 0\end{cases}$$

- $\vec{x}$, yani White noise resmimiz, standard error back-propagation (geri yayılım, öğrenmenin gerçekleştiği yer) ile hesaplanabilir.
- Low-layers, orijinal resim ile aynı pixel'leri ürettiğinden, content (içerik) gösterimi için; high-layers tercih edildi.

2.2.Style Representation

- Stil için, texture bilgisi yakalanması gerektiği için, feature space kullanıldı.
- Bu feature map, ağdaki herhangi bir filter responses katmanının üzerine eklenebilir.
- Farklı filter response'ların ilişkilerini (correlation) gösterir ve bu Gram matrisi ile ifade edilir:

$$G_i^l = \sum F_{ij}^l \cdot F_{jk}^l$$

Layer l'deki feature map i ve j'lerin arasındaki vektörize inner product (iç çarpımına) eşittir.

- Birden fazla katman arasındaki özellik korelasyonlarını ve texture bilgisini saptamamıza yarar.
- Elde edilen style feature spaces bilgisi görselleştirilmek için gradient descent white noise resim üzerine uygulanır. Amaç, hem orijinal resim ile hem de resimlerin Gram matrisleri arasındaki, mean-squared distance küçültmek, minimal değere indirgemektir.
- $\vec{a}$ (artwork) ve $\vec{x}$ (white noise) orijinal ve üretilen resmi temsil etsin ve A^l ile G^l, onların l layer(katmanındaki) style representation (stil gösterimi) olsun. O zaman, total loss (toplam kayıp) için her katmanın katkısı şu şekilde ifade edilebilir:

$$E_l = \frac{1}{4N_l^2 M_l^2}\sum_{i,j}\left(G_{ij}^l - A_{ij}^l\right)$$

Total loss (toplam kayıp) ise:

$$\mathcal{L}_{content}(\vec{a}, \vec{x}) = \sum_{l=0}^{l} \omega_l E_l$$

Her katmanın toplam kayba katkısı olan El türevini alırsak:

$$\frac{\partial E_l}{\partial F_j^l} = \begin{cases} \frac{1}{4N_l^2 M_l^2} \left((F^l)^T \cdot (G^l - A^l)\right)_{ij} & if\ F_{ij}^l < 0 \\ 0,\ if\ F_{ij}^l < 0 \end{cases}$$

2.3.Style Transfer

Minimize edilen loss function (kayıp fonksiyonu):

$$\mathcal{L}_{content}(\vec{p}, \vec{a}, \vec{x}) = \alpha \cdot \mathcal{L}_{content}(\vec{p}, \vec{x}) + \beta \mathcal{L}_{style}(\vec{a}, \vec{x})$$

- α ve β ağırlık (weighting) faktörleri. Bunları değiştirerek, content ve style arasındaki trade-off (yani içerik ve stil ters orantılı, içeriği koruduğunuz miktarda stilden kaybedersiniz ve tam tersi şekilde) oranına karar verebilirsiniz.
- $\frac{\partial \mathcal{L}_{content}}{\partial \vec{x}}$ ise, sayısal optimizasyon stratejisi için, input (girdi) olarak kullanılabilir.
- Ama bu makalede, L-BFGS optimizasyon yöntemi resim sentezi için en iyi olması nedeniyle tercih edildi.
- Her zaman stil resimleri (style images) resize (yeniden boyutlandırma) edilerek, içerik resmi (content images) ile aynı size (boyuta) eşitlendi.
- Sentez sonuçları, image priors ile (daha iyi sonuç için resim hakkında modele ipuçları babında verilen bilgi olarak düşünebilirsiniz) regularize edilmedi.

3. Results

3.1.Trade-off Between Content & Style Matching

3.2.Effect of Different Layers of CNN

- Style representations (stil gösterimlerini) ağdaki higher-layers (daha derindeki katmanlar ile) eşleştirmek, local (yerel) resim yapılarının artan geniş bir ölçekte korunmasını sağladı.
- Yani lower-layers kullanmak, orijinal resimden daha fazla içerik alınmasını sağlıyor. (Ve stilden de kaybediyoruz, trade-off nedeniyle)

3.3.Initialization of Gradient Descent

- Bu makalede, white noise resim kullanıldı ama isteyen başlangıç resmini istediği bir resim olarak seçebilir. Mesela, content veya style resimlerden biriyle başlatılabilir ama bu bias olmasına yol açacaktır, yani o resme daha çok benzeyecektir.

3.4.Photorealistic Style Transfer

4. Discussion

- En çok sınırlayıcı olan faktör sentez edilmeye çalışılan resmin çözünürlüğü (resolution).
- Optimizasyon probleminin boyutsallığı ve CNN'deki birimlerin sayısı, pixel sayısı ile birlikte, lineer bir şekilde artıyor. Bu yüzden, hız çözünürlüğe bağlıdır.
- Bu makalede, 512x512 pixel resimler ve Nvidia K40 GPU kullanıldı ve eğitimi bir saat sürdü.
- Bu algoritma, hız sorunundan sebep, online ve interaktif uygulamalarda henüz kullanılmaya hazır değil.

- Resmin içerik ve stil olarak ikiye ayrılması durumu iyi tanımlanmış bir problem değil ve bu yüzden evrensel fikir birliğine varılacak bir cevaba sahip değil.