✧ Image Style Transfer ✧

Gatys et. al. 2016

1. Introduction

Texture analysis → Efros & Freeman (correspondence map)
→ Hertzman et. al. (image analogies)
→ Ashikhmin (transferring the high-frequency texture info + preserve coarse scale)
→ Lee et. al. improved ← + edge orientation info.

remarkable results but limitation = only low-level image features of the target img to inform the texture transfer.

a style transfer algo. should be able to extract the semantic img content from the target img. (e.g. objects & scenery.) and then inform a texture transfer procedure to render the semantic content of the target img. in the style of the source img.

→ to seperate content from style, extract high-level semantic info. by using Convolutional Neural Networks (CNNs).

introduced: A Neural Algo. of Artistic Style.
+ combines a parametric texture model based on CNNs with a method to invert their img. ~~represtat~~ representations.

2. Deep Image Representations

- basis → VGG network which is for object recognition & localization.

VGG-19 } 16 convolutional & 5 pooling layes
total 19 layers = 16 conv + 3 fully connected

network is normalized by scaling the weights
→ the mean activation of each conv. filter over imgs. and positions is equal to one.

re-scaling can be done for the VGG net. without changing its output; because it contains only rectifying linear activation (ReLU) funcs. and no normalization or pooling over feature maps.

- not used any of fully connected layers.
- for img. synthesis avg. pooling is slightly better than max. pooling.

2.1 Content Representation

- each layer in network defines a non-linear filter bank
  - complexity of layer increases with the position of layer.
  - hence, input img. $\vec{x}$ is encoded in each layer of CNN by filter responses.

N_l → distinct layers
M_l → size of each feature map (height x width)
F^l → respons in layer l
↓
matrix $F^l \in R^{N_l \times M_l}$, F_{ij}^l → activation of the i^{th} filter at position j in layer l.

perform gradient descent on a white noise img. to find another img. that matches the features responses of original img.

original img. → white noise img. ← style img.

let $\vec{p}$ & $\vec{x}$ (white-noise) be original img., P^l & F^l their respective feature representations in layer l.

Then, squared-error loss:

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

Derivative of loss func. wrt. activations in layer l:

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

$\frac{\partial \mathcal{L}}{\partial \vec{x}}$ can be computed using standard error back-propagation

> low layers reproduce exact pixel values of original img.
> so, higher layers are preferred → as content representation.

2.2. Style Representation

- for style, a feature space to capture texture info.
  - this feature map can be built on top of the filter responses in any layer of net.
  - consists of correlations between different filter responses ⇒ represented by Gram matrix.

$G^l \in R^{N_l \times M_l}$, G_{ij}^{l} → inner product between vectorised feature map i and j in l.

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

• feature correlations of multiple layers → stationary,
+ texture info, not global arrangement (multiscale representati

◆ visualize info. by style feature spaces:

→ by using gradient descent from a white noise img. to minimise the mean-squared distance between entries of Gram matrices from original img. & Gram matrices of img. to be generated.

let $\vec{a}$ & $\vec{x}$ be original img. and generated img.
A^l & $\vec{G}$ respectively style representation in layer l.
Then, contribution of layer l to total loss:

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

Total style loss:

$$\mathcal{L}_{style}(\vec{a}, \vec{x}) = \sum_{l=0}^{L} \overset{\text{weighting factor}}{w_l} \cdot E_l$$

Derivative of E_l:

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

2.3. Style Transfer

The loss function that is minimized:

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

- α & β are weighting factors.
- $\frac{\partial \mathcal{L}_{total}}{\partial \vec{x}}$ can be used as input for some numerical optimisation strategy.

- here L-BFGS is used (best for img. synthesis)

- always resized style imgs to same size as the content.

- not regularised synthesis results with img. priors.

3. Results

3.1. Trade-off between content & style matching

3.2. Effect of different layers of the CNN

- matching the style representations up to higher layers in the network preserves local imgs. structures on increasingly large scale. ⊛ 'conv1_1', 'conv2_1', 'conv3_1', 'conv4_1' and 'conv5_1'

- lower layers → more content from original img.

3.3. Initialisation of Gradient Descent

- white noise is used but one can also initialize it with content or style img; but it causes bias.

3.4. Photorealistic Style Transfer

4. Discussion

- most limiting factor is resolution of the synthesised img.

- dimensionality of the optimisation problem + the number of units in the CNN } grow linearly with # of pixels.

  so, speed depends on resolution

512x512 pixels, Nvidia K40 GPU } in this paper → prohibits online & interactive applications
one hour

- seperation of img. content from style is not well-defined problem.

Bonus: L-BFGS : Limited-memory BFGS

- an optimization algorithm that approximates BFGS* algorithm (Broyden-Fletcher-Goldfarb-Shanno) using a limited amount of computer memory. family: quasi-Newton methods.

  * an iterative method for solving unconstrained non-linear optimization. $O(n^2)$

- popular algo. for parameter estimation in ML.
- target problem is to minimize $f(x)$ over constrained values.

Quasi-Newton Method: used to either find local maxima & minima of functions or zeroes. They can be used if the Jacobian or Hessian is unavailable or too expensive.

- L-BFGS aka "the algo. of choice" for fitting log-linear (MaxEnt) and conditional random fields with ℓ_2-regularization.

a way of finding a (local) minimum of an objective func., making use of objective func. values and the gradient of the obj. func.

CamScanner ile tarandı