Texture Networks: Feed-forward Synthesis of Textures and Stylized Images

Dmitry Ulyanov, Vadim Lebedev, Andrea Vedaldi, Victor Lempitsky

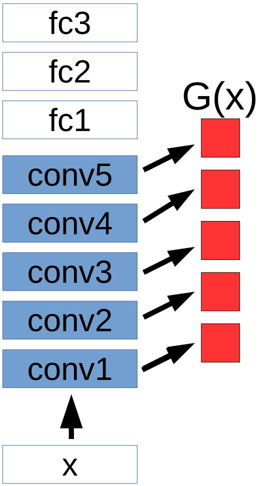
June 5, 2016

Structure

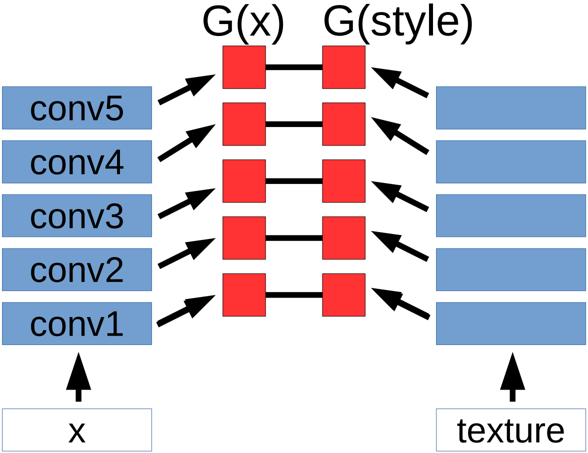
	Texture generation	Stylization
Optimization (Gatys et al.)	1	2
Feedforward (Ulyanov et al.)	3	4

- And more!

Texture Synthesis



- Input image *x* is fed to VGG19
- Activation maps $F_i^{(l)}$ contain full description of the image
- Gram matirices $G_{ij}^{(l)} = (F_i^{(l)}, F_j^{(l)})$ describe texture, spatial information is lost



Distance from image x to a fixed texture sample is defined by

$$\mathcal{L}_{\text{texture}}(x) = \sum_{l \in L_t} \|G_{ij}^{(l)}(x) - G_{ij}^{(l)}(\text{texture})\|_F^2$$

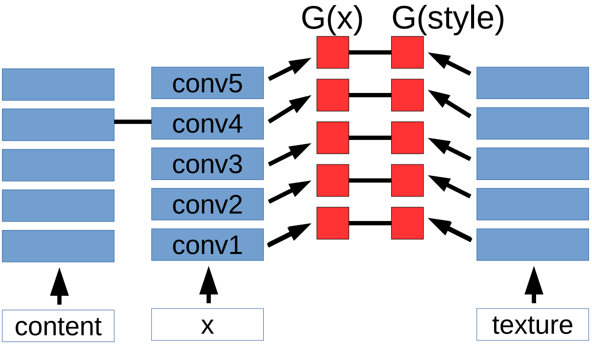
Texture Generation

- Constuct texture x_0 by minimization

$$x_0 = \operatorname{argmin} L(x)$$

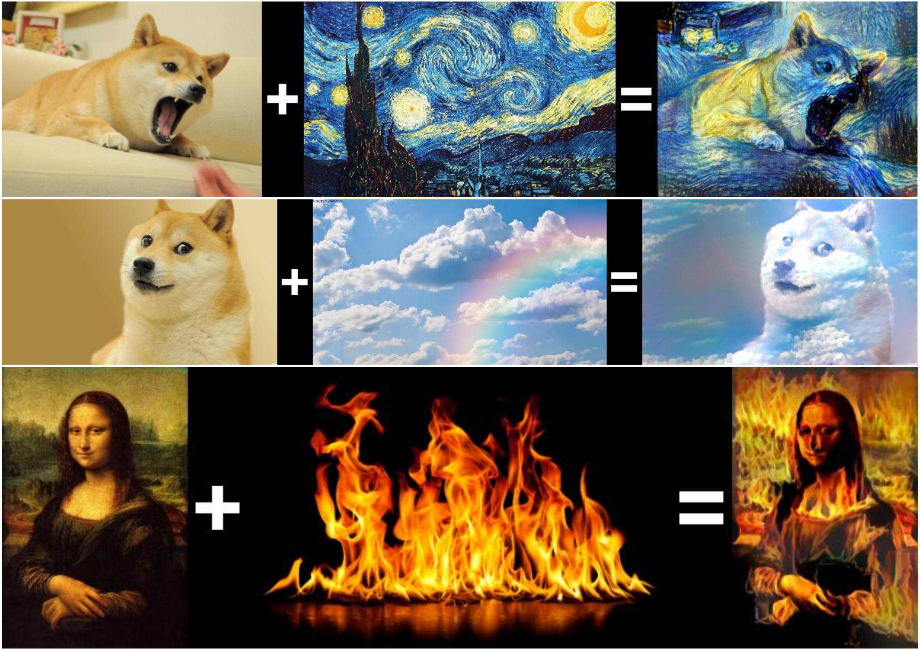
- L-BFGS, several hundreds iterations
- pixels of x_0 are initialized with noise $N(0, \sigma)$

Image Stylization



$$L_{\text{content}}(x) = \sum_{l \in L_c} \|F^{(l)}(x) - F^{(l)}(\text{content})\|_F^2$$

$$L(x) = \alpha L_{\text{texture}}(x) + \beta L_{\text{content}}(x)$$

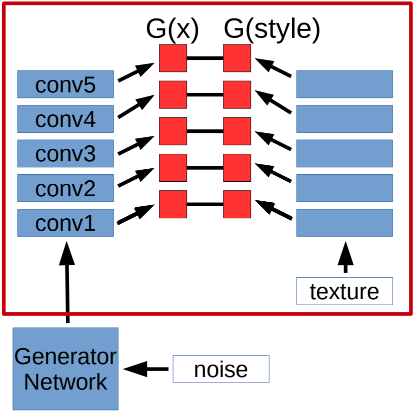


Timing

2cm 5m 7m 10m 15m

- Single image takes several minutes on GPU and hours on CPU
- We can make it faster, both for texture generation and image stylization tasks

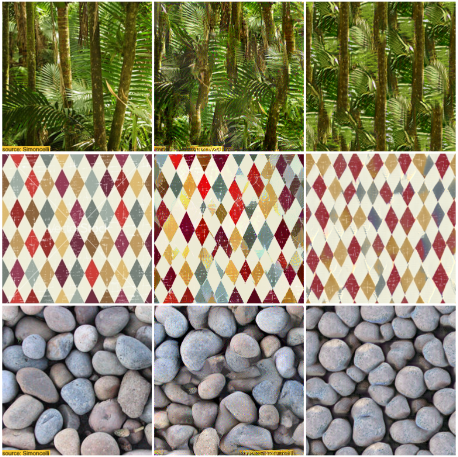
Feedforward Texture Generation



- Train generator network with loss L
- After that, produce images in single pass instead of hundreds iterations

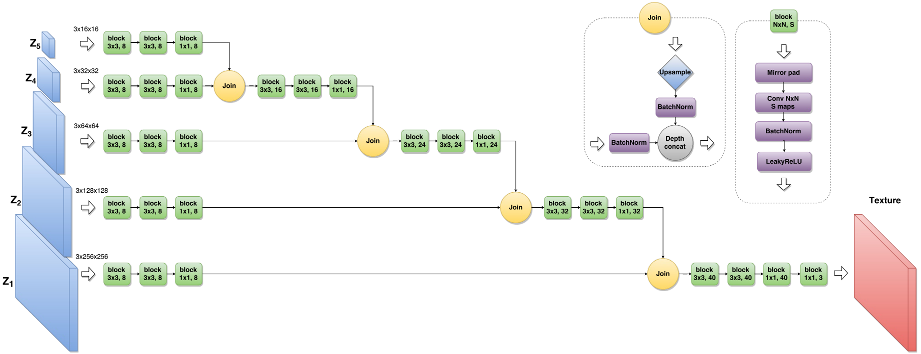
Generated Textures

Quality is similar to images obtained by otimization

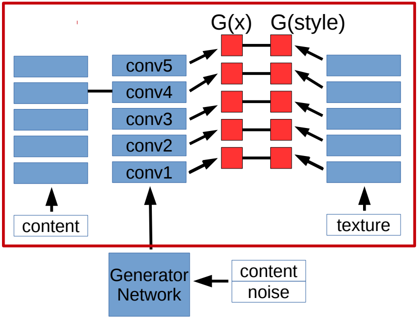


original optim feedforward

Generator Network

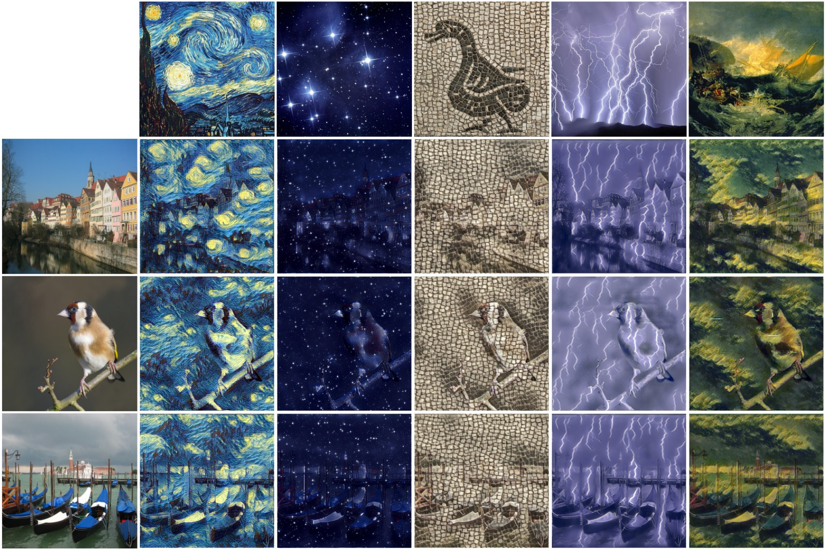


Feedforward Image Stylization



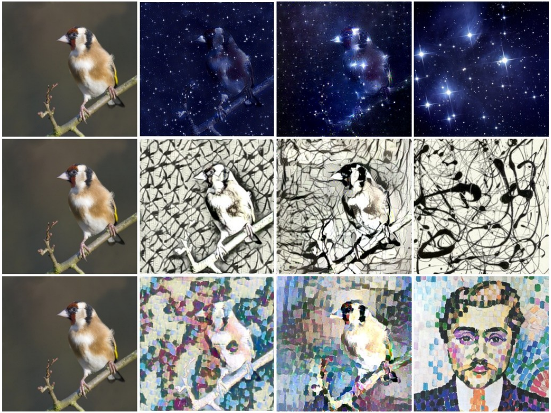
- Similar to texture generation
- + content loss
- Generator recieves noise and content image as inputs

Stylization Examples



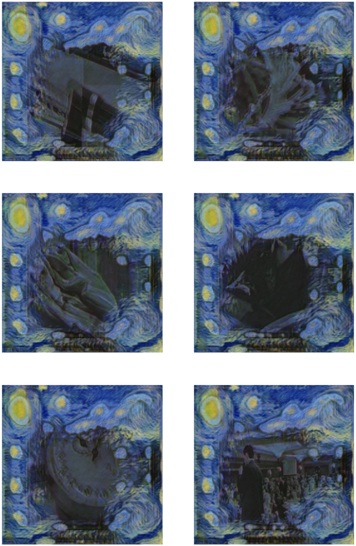
Stylization Examples

Feedforward generation results are mostly inferior to optimization results



content feedforward optim texture

Edge effects



- Generator will cheat the loss if it is possible
- Padding with zeros allows generator to infer position of a given pixel and produce fixed patterns
- We use circular padding to overcome this problem

Do we need pretrained weights?

- Gatys et al. say **yes**
- Contradicting results were published
- If pretrained weights are not needed, we can apply same methods to other domains, such as sound and speech

Code

- Optimizational image stylization is avaliable for every deep learning library, but Torch implementation is the most popular: `https://github.com/jcjohnson/neural-style`
- Our Torch implementation of feed-forward method: `https://github.com/DmitryUlyanov/texture_nets`
  Tensorflow is also available, lasagne is coming soon
- Check out our demo for feedforward neural doodling: `http://likemo.net/`

Conclusion

- We have presented a new approach for for texture synthesis and image stylization in a feed-forward way
- Complex loss function is used. It involves deep nets assessing the performance of the feed-forward generator
- Two orders of magnitude faster compared to optimisation approach by Gatys et al.
- Comparable quality for texture generation, slightly worse for image stylization
- Future work: better generation architectures, video, sound?