Deep Dream i Neuralni prenos stila

Stanford University

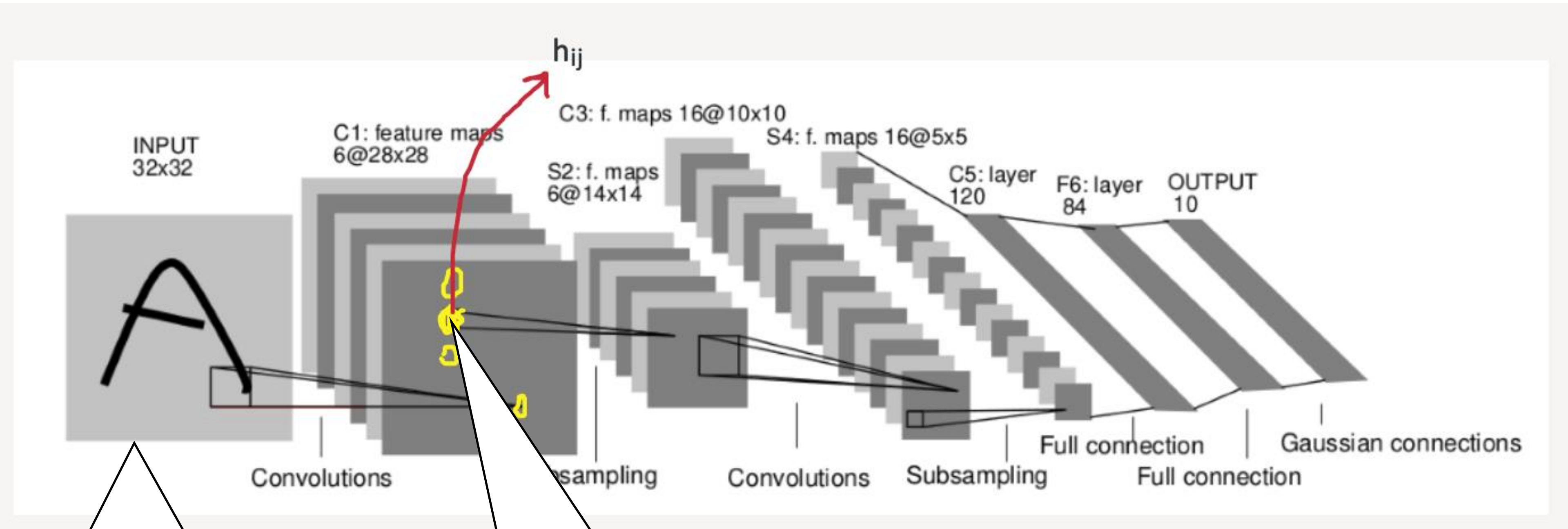
- https://www.youtube.com/watch?v=6wcs6szJWMY&list=PL3FW7Lu3i5JvHM8ljYj-zLfQRF3EO8sYv&index=12
- http://cs231n.github.io/understanding-cnn/

DeepDream

- Vizuelizacija šablona koje je CNN model naučio

Similar to when a child watches clouds and tries to interpret random shapes, DeepDream over-interprets and enhances the patterns it sees in an image

DeepDream



Propuštamo sliku kroz obučenu mrežu

Želimo da uvećamo aktivaciju h_{ij}

(menjamo sliku da aktivacija bude veća)

$$L(I) = h_{ij}^2$$

Magnituda h_{ij}

$$\max_I (L(I))$$

Izračunamo gradijent po I i menjamo sliku

DeepDream

DeepDream: Amplify existing features

https://github.com/google/deepdream

Code is very simple but it uses a couple tricks:


```
def objective_L2(dst):
    dst.diff[:] = dst.data

def make_step(net, step_size=1.5, end='inception_4c/output',
              jitter=32, clip=True, objective=objective_L2):
    '''Basic gradient ascent step.'''

    src = net.blobs['data'] # input image is stored in Net's 'data' blob
    dst = net.blobs[end]

    ox, oy = np.random.randint(-jitter, jitter+1, 2)
    src.data[0] = np.roll(np.roll(src.data[0], ox, -1), oy, -2) # apply jitter shift

    net.forward(end=end)
    objective(dst)  # specify the optimization objective
    net.backward(start=end)
    g = src.diff[0]
    # apply normalized ascent step to the input image
    src.data[:] += step_size/np.abs(g).mean() * g

    src.data[0] = np.roll(np.roll(src.data[0], -ox, -1), -oy, -2) # unshift image

    if clip:
        bias = net.transformer.mean['data']
        src.data[:] = np.clip(src.data, -bias, 255-bias)
```

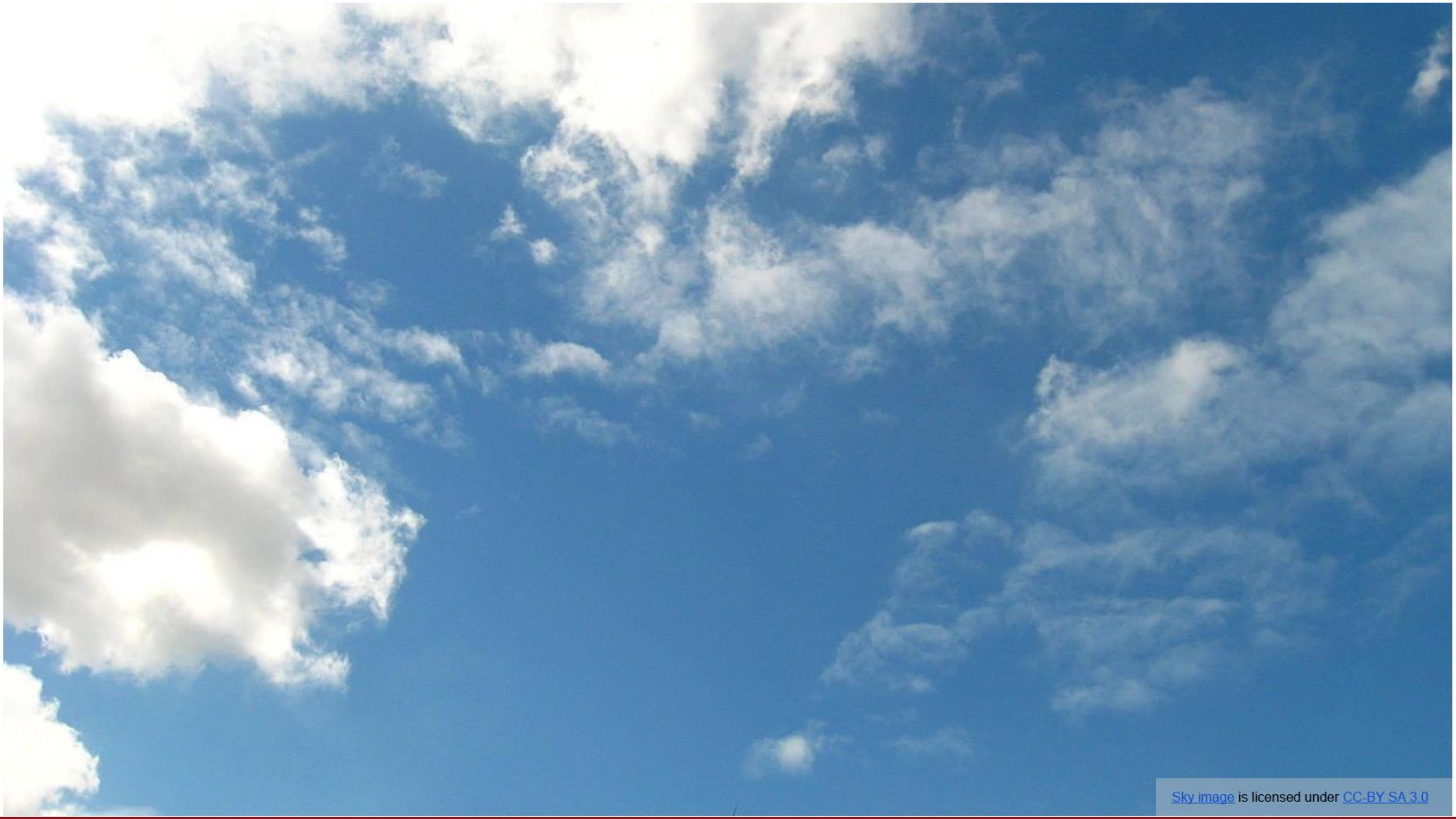
Jitter image

L1 Normalize gradients

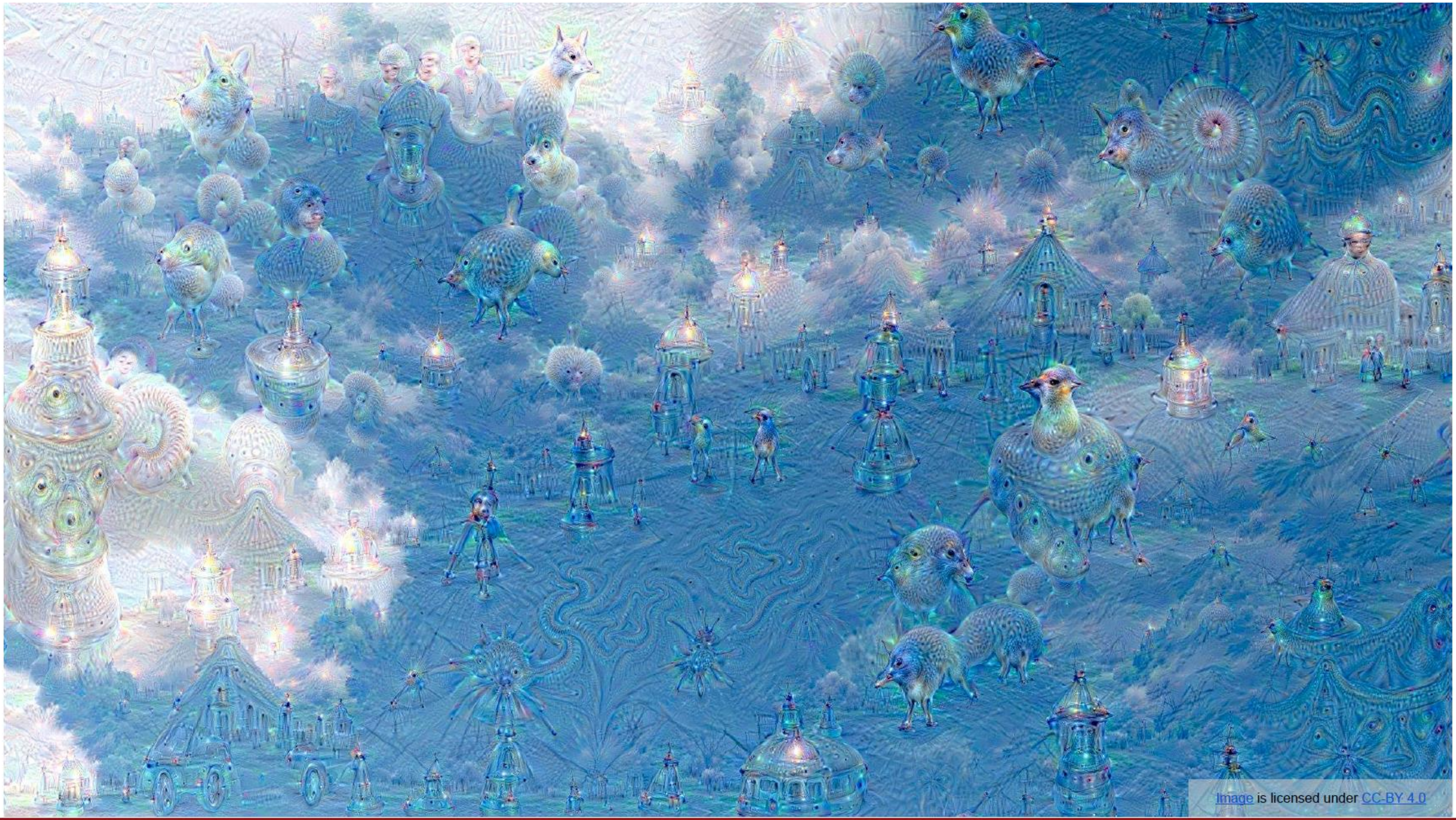
Clip pixel values

Also uses multiscale processing for a fractal effect (not shown)

DeepDream

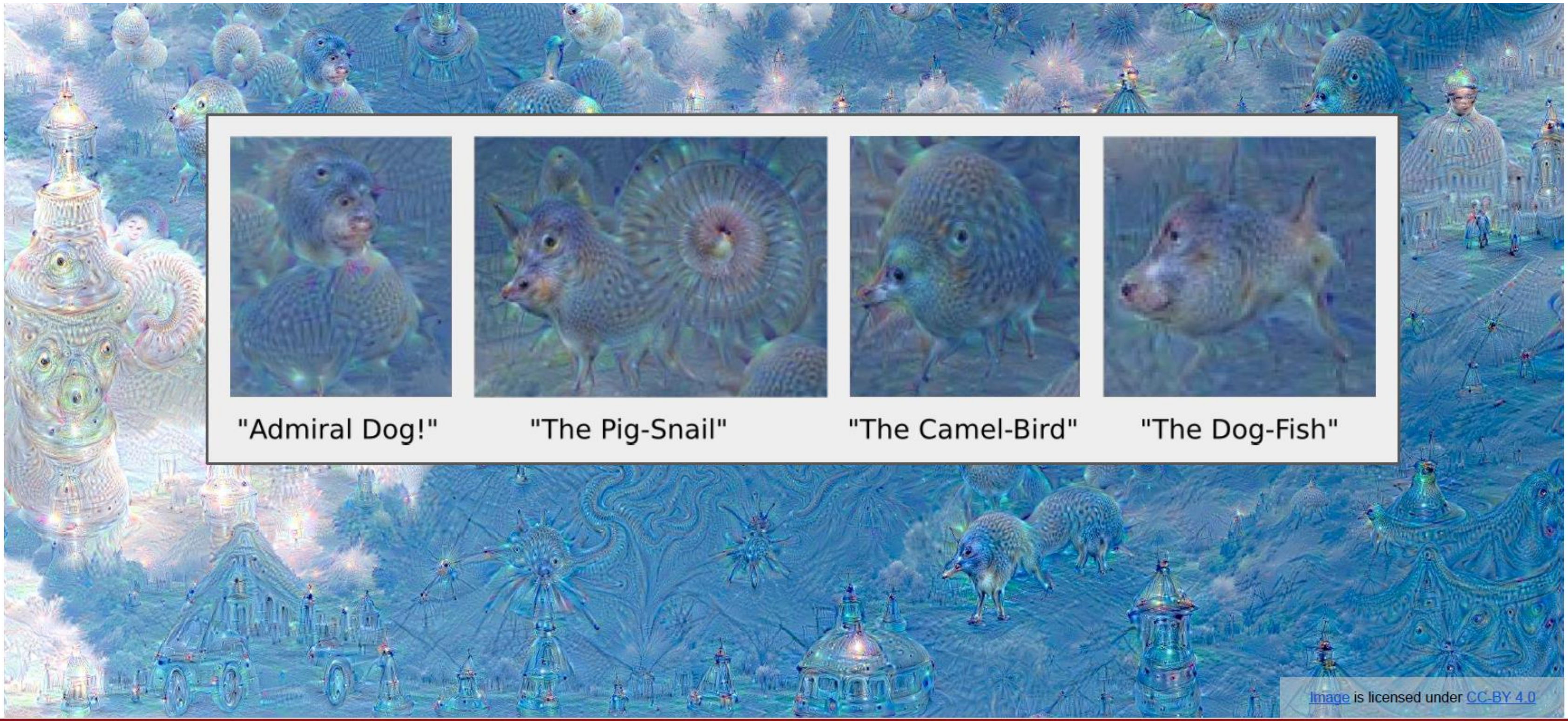


DeepDream

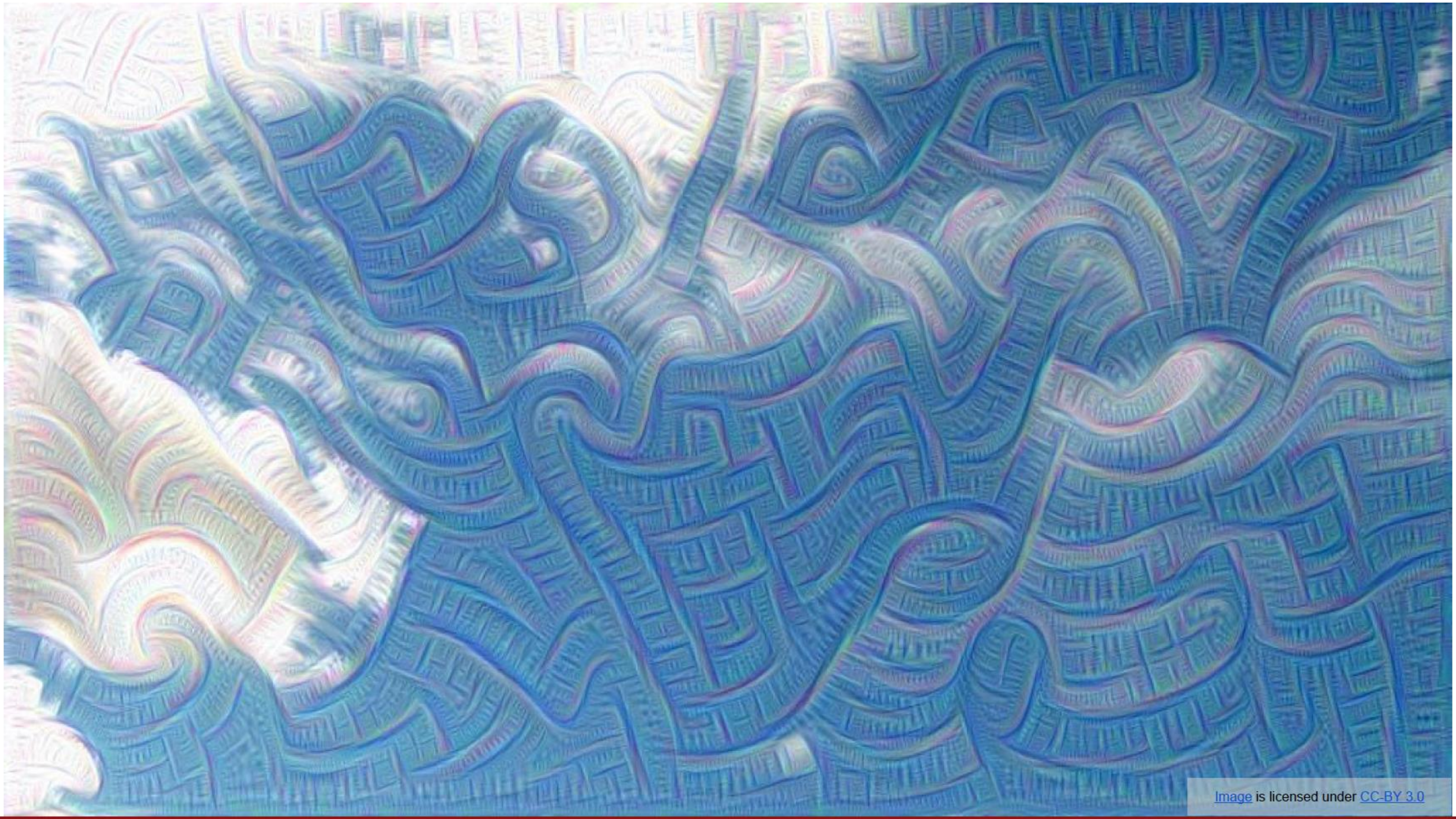


DeepDream

- Činjenica da se pas toliko puta pojavljuje u ovim vizuelizacijama nam zapravo govori nešto o podacima na kojima je mreža trenirana
- Ova mreža je trenirana na 1000 kategorija ImageNet, ali 200 od ovih kategorija su psi



DeepDream



Na drugim (nižim) slojevima mreže rezultati izgledaju drugačije

DeepDream – ImageNet

DeepDream – MIT Places Dataset

200 tipova scena (spavaće sobe, kuhinje,...)

Feature inversion

- Još jedan postupak koji služi da nam pruži uvid koji tipovi elemenata slike su „uhvaćeni“ na različitim slojevima mreže

- Postupak:
  - Sliku propustimo kroz mrežu
  - Snimimo vrednosti obeležja
  - Pokušamo da rekonstruišemo tu sliku od dobijene reprezentacije obeležja (sintetišemo sliku koja bi rezultovala istim obeležjima kao ona koja smo snimili)
  - Bazirano na tome kako izgleda rekonstruisana slika, dobićemo osećaj koji tip informacija o slici je „uhvaćen“ datim obeležjima

Feature inversion

Given a CNN feature vector for an image, find a new image that:

- Matches the given feature vector
- "looks natural" (image prior regularization)

$$\mathbf{x}^* = \underset{\mathbf{x} \in \mathbb{R}^{H \times W \times C}}{\operatorname{argmin}} \ \ell(\Phi(\mathbf{x}), \Phi_0) + \lambda \mathcal{R}(\mathbf{x})$$

$$\ell(\Phi(\mathbf{x}), \Phi_0) = \|\Phi(\mathbf{x}) - \Phi_0\|^2$$

$$\mathcal{R}_{V^\beta}(\mathbf{x}) = \sum_{i,j} \left((x_{i,j+1} - x_{ij})^2 + (x_{i+1,j} - x_{ij})^2 \right)^{\frac{\beta}{2}}$$

Total Variation regularizer (encourages spatial smoothness)

Penalizuje razlike između susednih piksela (levo/desno i ispod/iznad) da pobojša prostornu glatkoću generisane slike

Mahendran and Vedaldi, "Understanding Deep Image Representations by Inverting Them", CVPR 2015

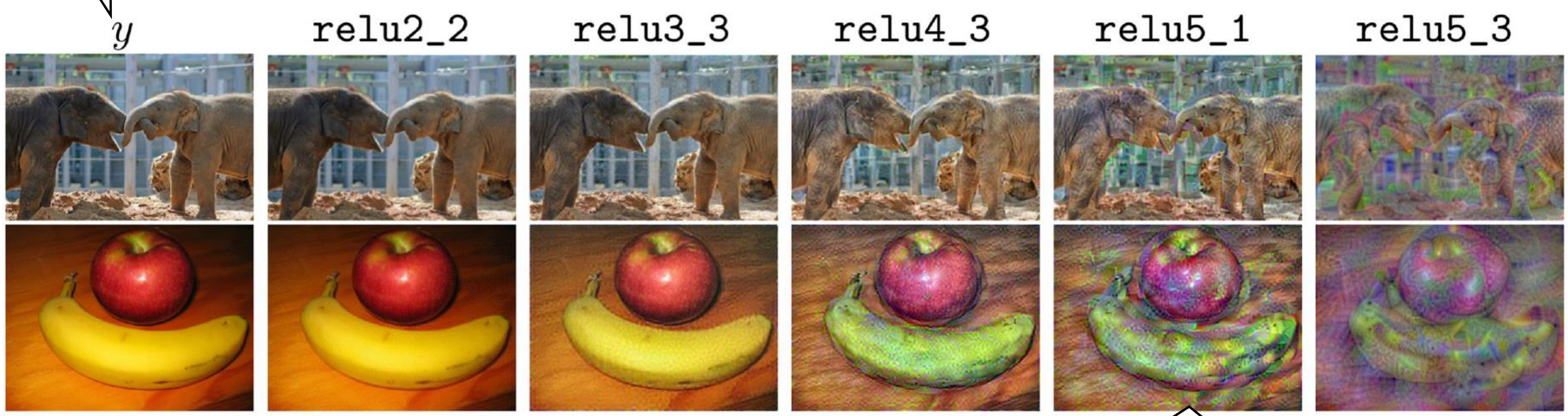
Feature inversion

Originalna slika

Propustimo sliku kroz VGG16 mrežu

Snimimo obeležja te slike izračunata u datom sloju. Sintetišemo novu sliku koja odgovara snimljenim obeležjima

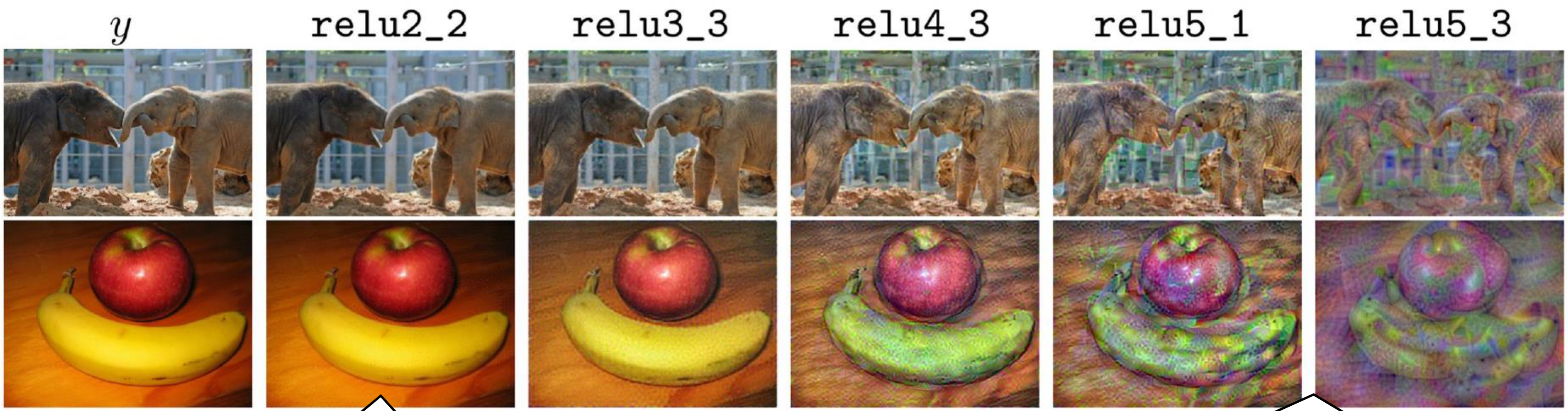
Reconstructing from different layers of VGG-16



Daje nam osećaj koliko je informacija snimljeno u datim obeležjima

Mahendran and Vedaldi, "Understanding Deep Image Representations by Inverting Them", CVPR 2015
Figure from Johnson, Alahi, and Fei-Fei, "Perceptual Losses for Real-Time Style Transfer and Super-Resolution", ECCV 2016. Copyright Springer, 2016.
Reproduced for educational purposes.

Feature inversion



Niži slojevi:

- Slike su gotovo perfektno rekonstruisane
- Ne odbacujemo mnogo informacija o „sirovim“ vrednostima piksela

Dublji slojevi:

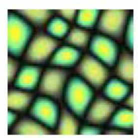
- Rekonstruisana slika čuva generalnu prostornu strukturu slike (možemo još da kažemo da je u pitanju jabuka, banana ili slon)
- Ali, fini detalji (niskog nivoa) su izgubljeni (slike nisu ono što su bile u pogledu boje, teksture,...)

Kako se krećemo kroz slojeve mreže, odbacujemo informacije niskog nivoa i čuvamo samo bitnije informacije kako bismo bili više invarijantni na male promene u boji i teksturi

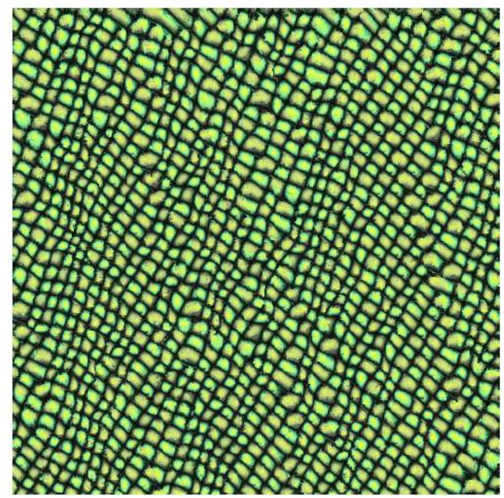
Sinteza teksture

Texture Synthesis

Given a sample patch of some texture, can we generate a bigger image of the same texture?



Input

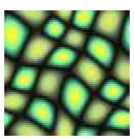


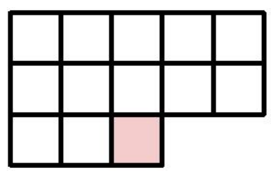
Output

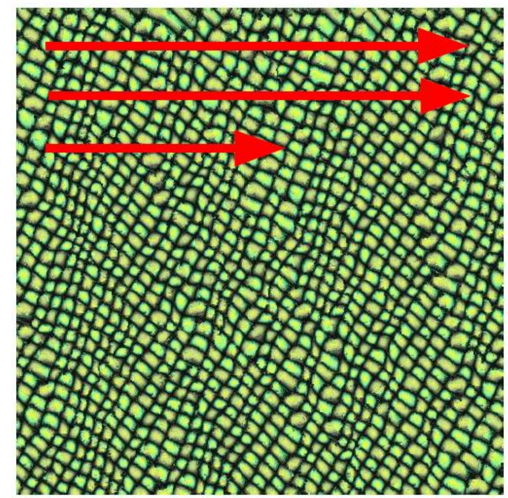
Sinteza teksture – klasični algoritmi

Texture Synthesis: Nearest Neighbor

Generate pixels one at a time in scanline order; form neighborhood of already generated pixels and copy nearest neighbor from input



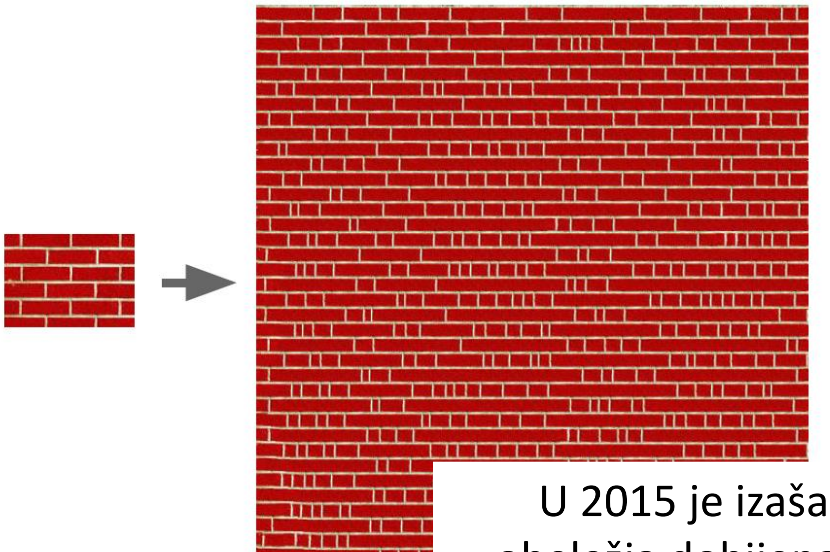


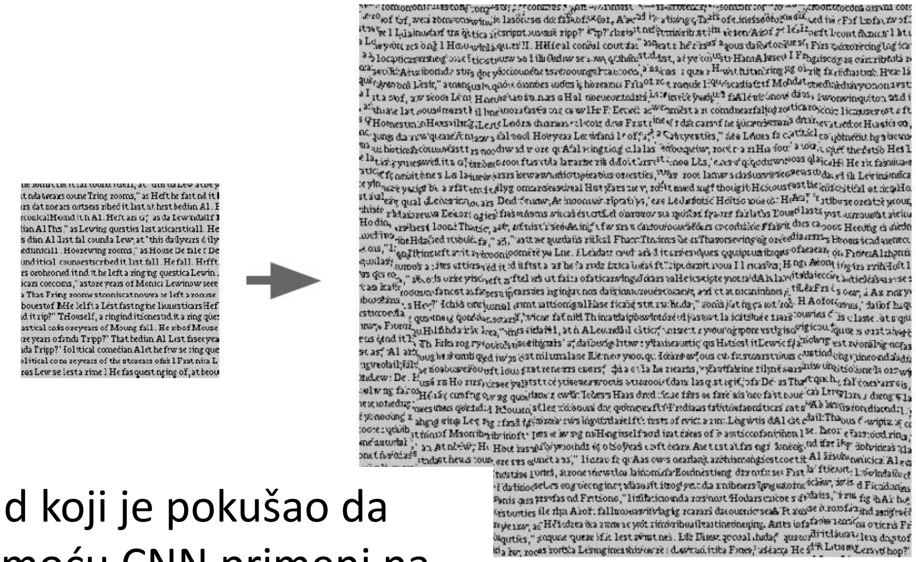


Wei and Levoy, "Fast Texture Synthesis using Tree-structured Vector Quantization", SIGGRAPH 2000
Efros and Leung, "Texture Synthesis by Non-parametric Sampling", ICCV 1999

Sinteza teksture – klasični algoritmi

Texture Synthesis: Nearest Neighbor

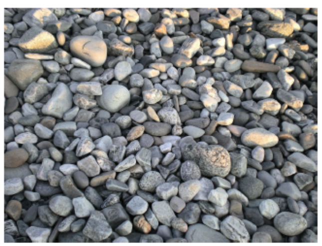




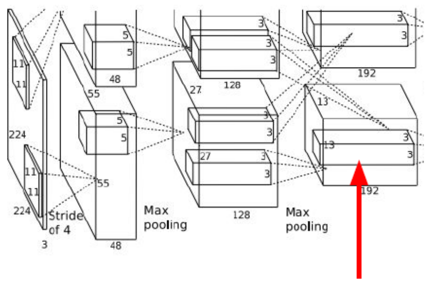
U 2015 je izašao rad koji je pokušao da obeležja dobijena pomoću CNN primeni na ovaj problem sinteze teksture

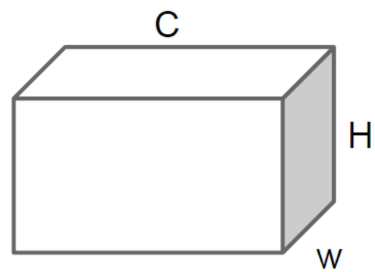
Sinteza teksture pomoću neuronskih mreža

Neural Texture Synthesis: Gram Matrix



This image is in the public domain.





Korističemo ovu aktivacionu mapu da izračunamo deskriptor teksture ove ulazne slike

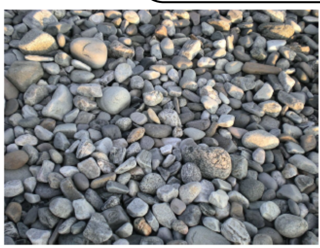
Each layer of CNN gives C x H x W tensor of features; H x W grid of C-dimensional vectors

C-dimenzioni vektor obeležja koji opisuje (grubi) izgled ulazne slike

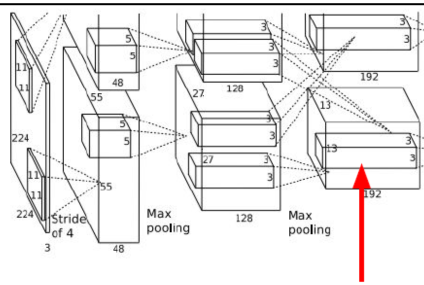
Sinteza teksture pomoću neuronskih mreža

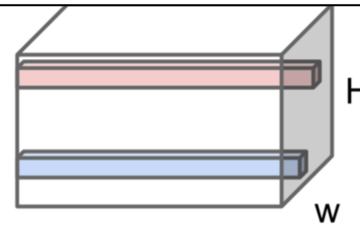
Ova matrica nam kaže nešto o istovremenoj pojavi dva različita obeležja na date dve tačke slike

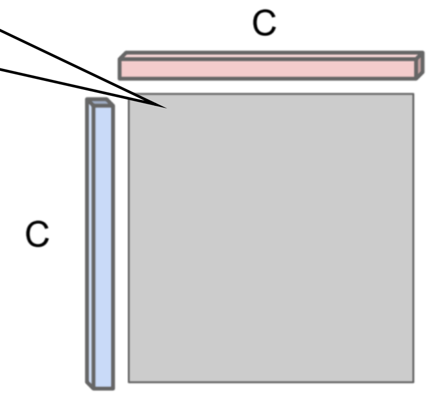
koja obeležja imaju tendenciju da se istovremeno aktiviraju na različitim prostornim pozicijama



This image is in the public domain.







Gram Matrix

Each layer of CNN gives C x H x W tensor of features; H x W grid of C-dimensional vectors

Outer product of two C-dimensional vectors gives C x C matrix measuring co-occurrence

Average over all HW pairs of vectors, giving **Gram matrix** of shape C x C

$$\mathbf{u} \otimes \mathbf{v} = \mathbf{u}\mathbf{v}^{\mathsf{T}} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} \begin{bmatrix} v_1 & v_2 & v_3 \end{bmatrix} = \begin{bmatrix} u_1v_1 & u_1v_2 & u_1v_3 \\ u_2v_1 & u_2v_2 & u_2v_3 \\ u_3v_1 & u_3v_2 & u_3v_3 \\ u_4v_1 & u_4v_2 & u_4v_3 \end{bmatrix}$$

Gram matrica

- Ponovićemo ovu proceduru koristeći sve kombinacije parova vektora obeležja
  - (vektora na različitim pozicijama H x W mreže)

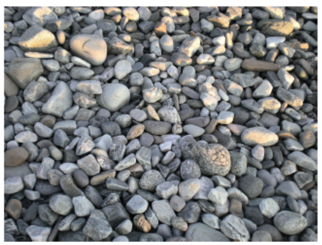
- Uprosečićemo sve dobijene matrice i to će nam vratiti konačnu gram matricu veličine $C \times C$

- Na ovoj gram matrici su odbačene sve prostorne informacije iz ove ulazne zapremine C x H x W. Umesto toga čuvamo samo statistiku o istovremenoj pojavi obeležja (na različitim pozicijama)

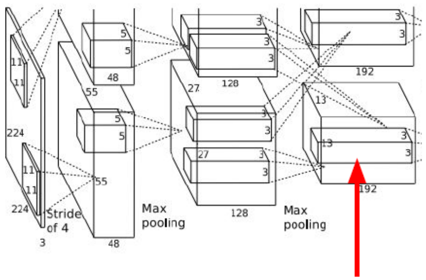
- Ova gram matrica se može koristiti kao deskriptor koji opisuje kakve se teksture nalaze na toj ulaznoj slici

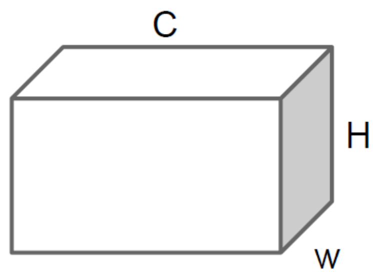
Sinteza teksture pomoću neuronskih mreža

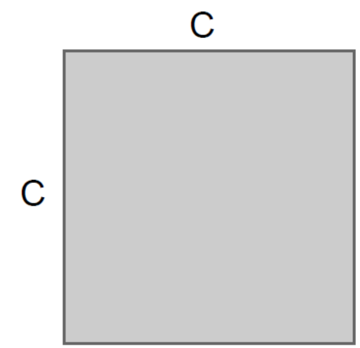
Neural Texture Synthesis: Gram Matrix



This image is in the public domain.







Each layer of CNN gives C x H x W tensor of features; H x W grid of C-dimensional vectors

Outer product of two C-dimensional vectors gives C x C matrix measuring co-occurrence

Average over all HW pairs of vectors, giving **Gram matrix** of shape C x C

Efficient to compute; reshape features from

C x H x W to =C x HW

then compute $G = FF^T$

Sinteza teksture

- Kada imamo ovaj „neuralni“ deskriptor strukture, možemo koristiti *gradient ascent* proceduru da sintetišemo novu sliku koja odgovara teksturi

- Ovo dosta liči na *feature reconstruction* koji smo videli na prethodnim slajdovima

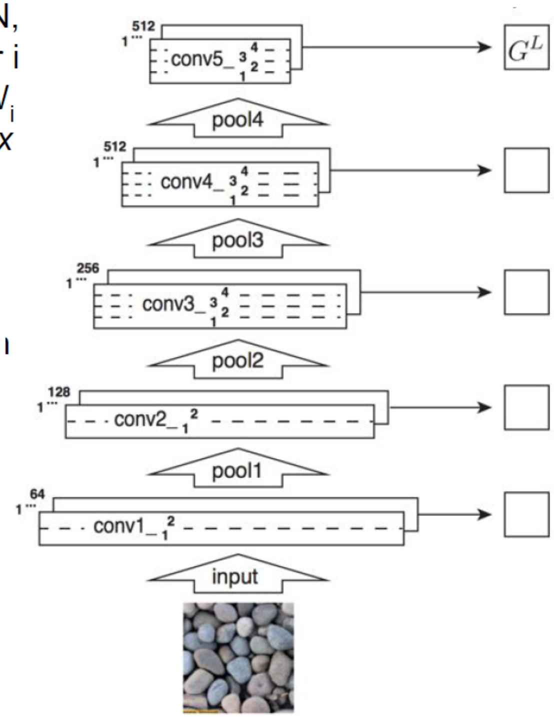
- Ali, umesto što pokušavamo da rekonstruišemo celu mapu obeležja, pokušaćemo da rekonstruišemo ovaj deskriptor (gram matricu) ulazne slike

Sinteza teksture pomoću neuronskih mreža

Neural Texture Synthesis

1. Pretrain a CNN on ImageNet (VGG-19)
2. Run input texture forward through CNN, record activations on every layer; layer i gives feature map of shape $C_i \times H_i \times W_i$
3. At each layer compute the *Gram matrix* giving outer product of features:

$$G^l_{ij} = \sum_k F^l_{ik} F^l_{jk} \text{ (shape } C_i \times C_i\text{)}$$



Gatys, Ecker, and Bethge, "Texture Synthesis Using Convolutional Neural Networks", NIPS 2015

Sinteza teksture pomoću neuronskih mreža

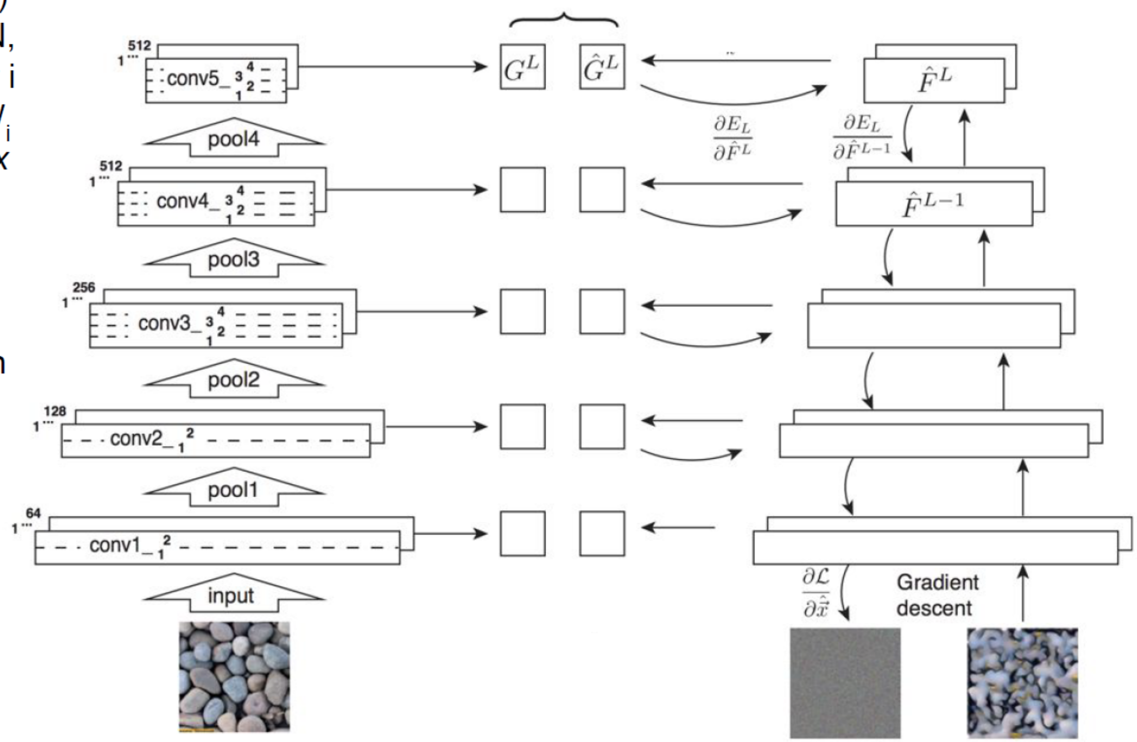
Neural Texture Synthesis

$$E_l = \frac{1}{4N_l^2 M_l^2} \sum_{i,j} \left(G_{ij}^l - \hat{G}_{ij}^l \right)^2 \qquad \mathcal{L}(\vec{x}, \hat{\vec{x}}) = \sum_{l=0}^{L} w_l E_l$$

1. Pretrain a CNN on ImageNet (VGG-19)
2. Run input texture forward through CNN, record activations on every layer; layer i gives feature map of shape $C_i \times H_i \times W_i$
3. At each layer compute the *Gram matrix* giving outer product of features:

$$G_{ij}^l = \sum_k F_{ik}^l F_{jk}^l \text{ (shape } C_i \times C_i)$$

4. Initialize generated image from random noise
5. Pass generated image through CNN, compute Gram matrix on each layer
6. Compute loss: weighted sum of L2 distance between Gram matrices
7. Backprop to get gradient on image
8. Make gradient step on image
9. GOTO 5



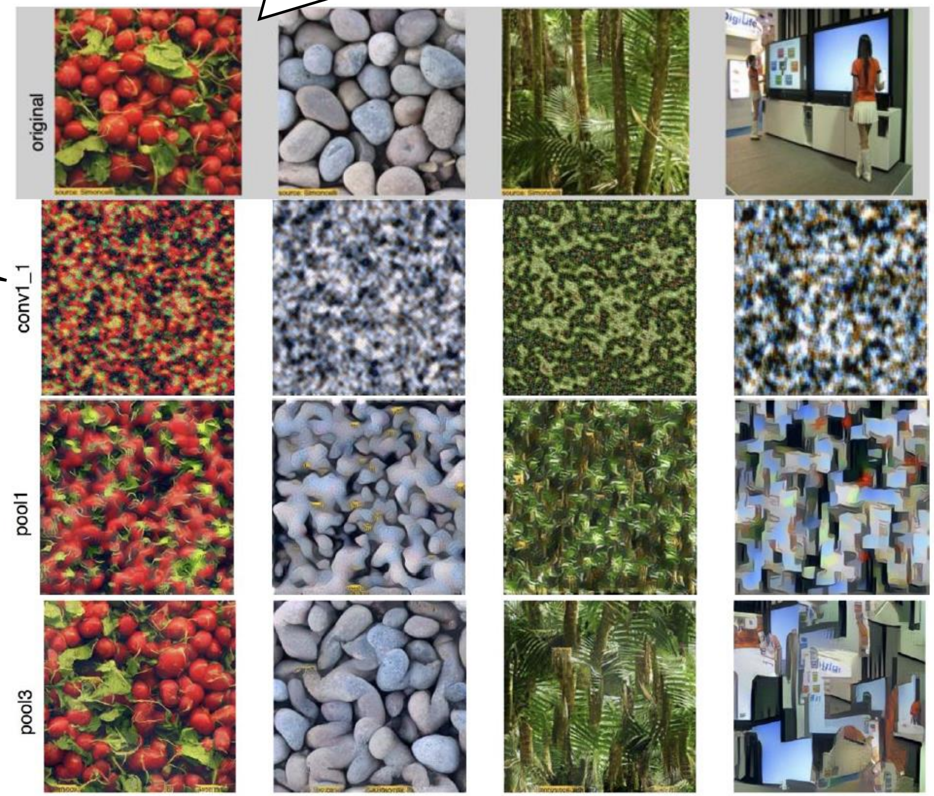
Sinteza teksture pomoću neuronskih mreža

Ispod: generisanje strukture određenog sloja pretrenirane CNN mreže

(za računanje gubitka korišćen je samo dati sloj)

Ulazne teksture

Reconstructing texture from higher layers recovers larger features from the input texture



Gatys, Ecker, and Bethge, "Texture Synthesis Using Convolutional Neural Networks", NIPS 2015

Sinteza teksture pomoću neuronskih mreža

- Možemo sintetisati nove slike koji odgovaraju generalnoj prostornoj statistici ulaza, ali su dosta drugačije po vrednostima pojedinačnih piksela od same ulazne slike

Umetnička dela umesto teksture

Neural Texture Synthesis: Texture = Artwork

Texture synthesis (Gram reconstruction)

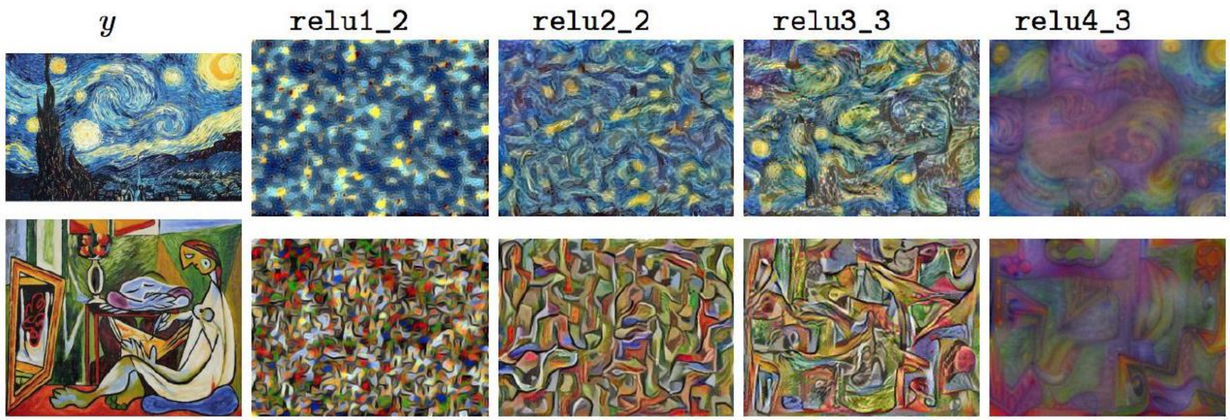
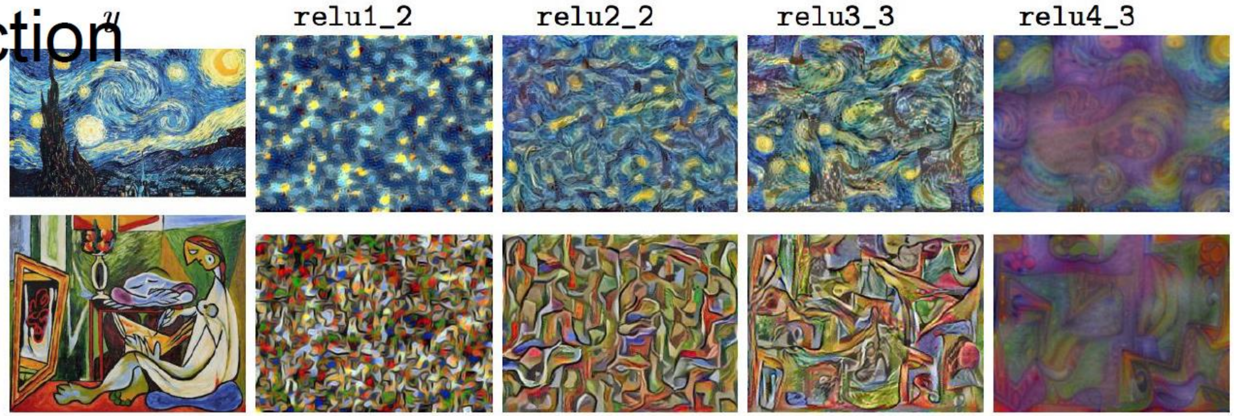


Figure from Johnson, Alahi, and Fei-Fei, "Perceptual Losses for Real-Time Style Transfer and Super-Resolution", ECCV 2016. Copyright Springer, 2016. Reproduced for educational purposes.

Umetnička dela umesto teksture

Neural Style Transfer: Feature + Gram Reconstruction

Texture synthesis (Gram reconstruction)



Feature reconstruction

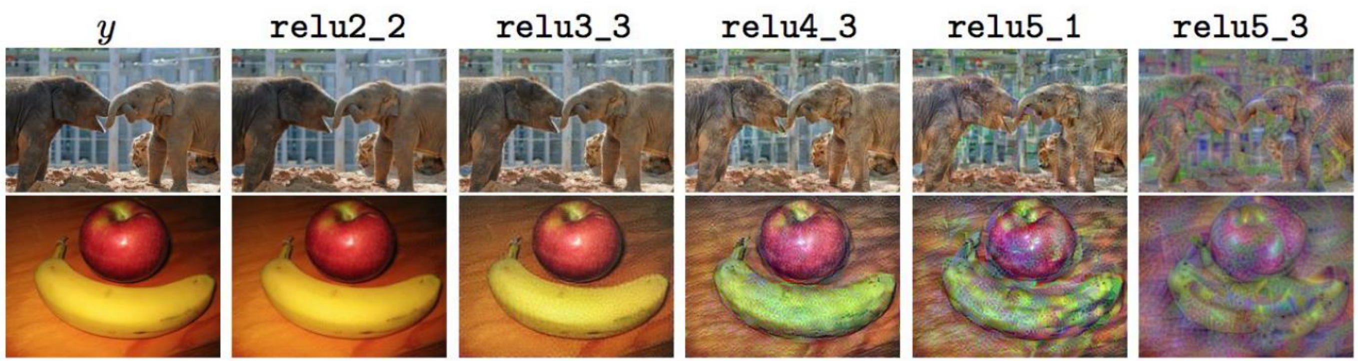


Figure from Johnson, Alahi, and Fei-Fei, "Perceptual Losses for Real-Time Style Transfer and Super-Resolution", ECCV 2016. Copyright Springer, 2016. Reproduced for educational purposes.

Neuralni prenos stila

Neural Style Transfer

Generisaćemo novu sliku minimizacijom
- *feature reconstruction loss* slike sadržaja i
- gubitka izračunatog pomoću gram matrica slike stila

Content Image

+

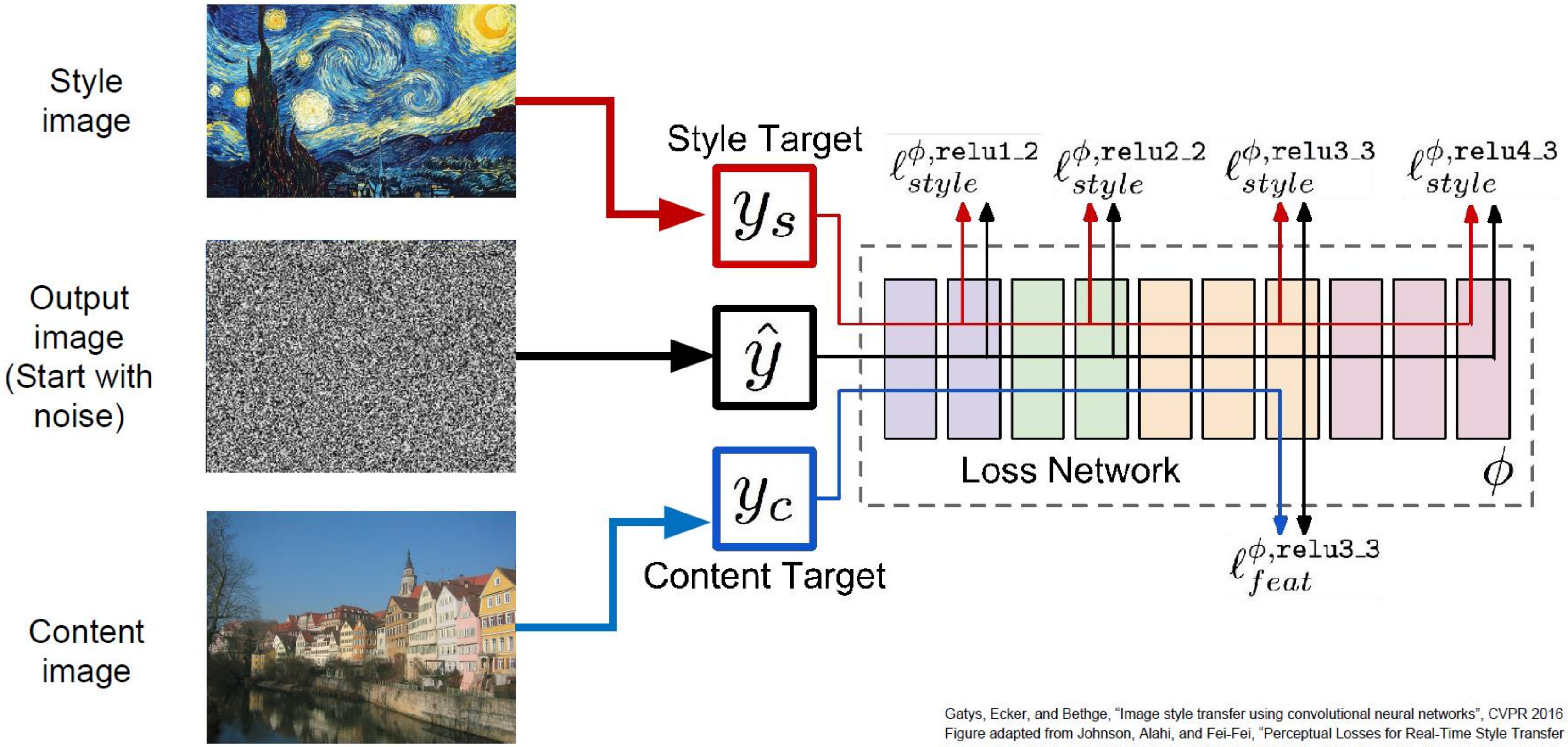
Style Image

=

Style Transfer!

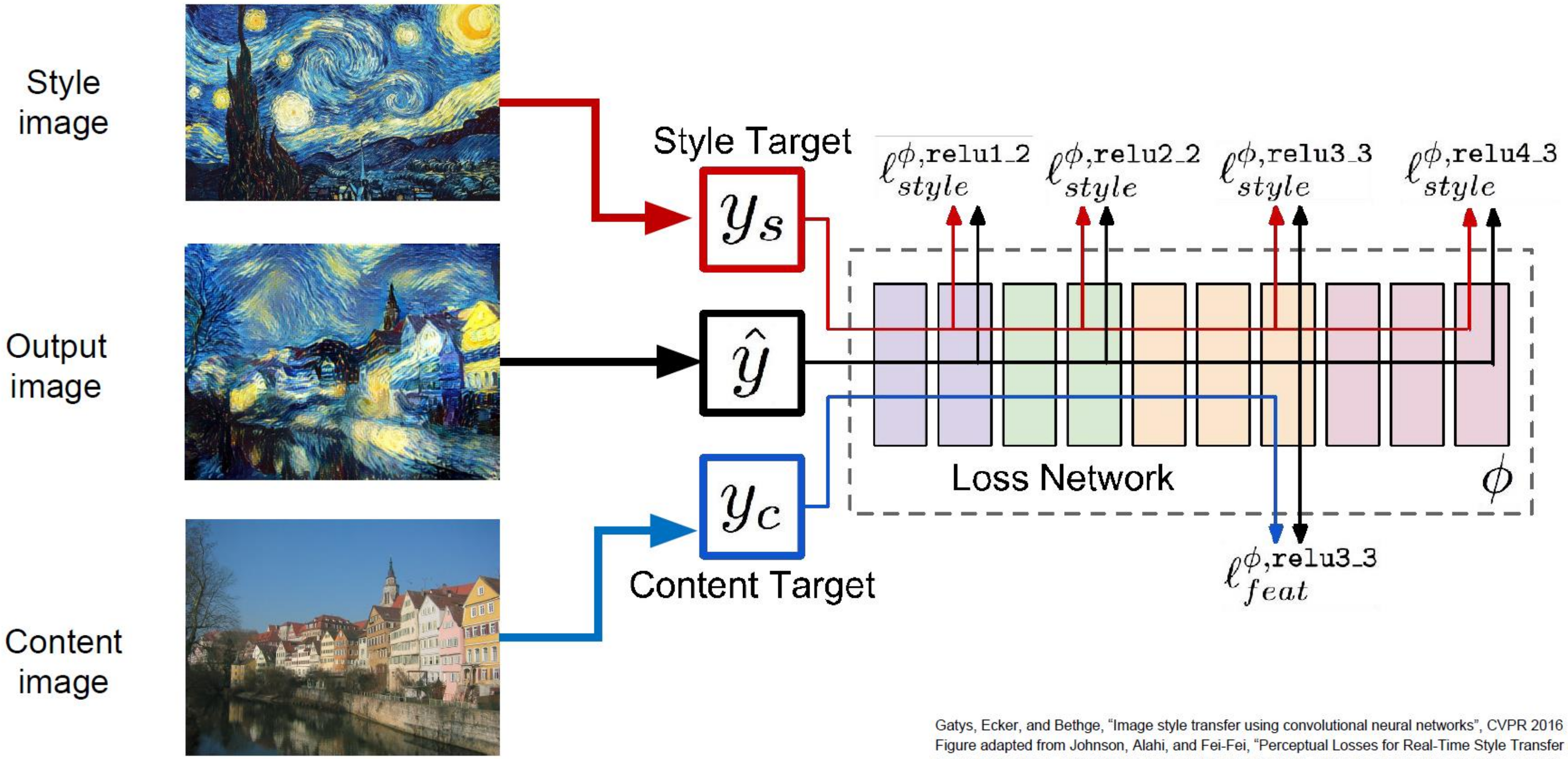
Gatys, Ecker, and Bethge, "Image style transfer using convolutional neural networks", CVPR 2016

Neuralni prenos stila



Gatys, Ecker, and Bethge, "Image style transfer using convolutional neural networks", CVPR 2016
Figure adapted from Johnson, Alahi, and Fei-Fei, "Perceptual Losses for Real-Time Style Transfer and Super-Resolution", ECCV 2016. Copyright Springer, 2016. Reproduced for educational purposes.

Neuralni prenos stila



Gatys, Ecker, and Bethge, "Image style transfer using convolutional neural networks", CVPR 2016
Figure adapted from Johnson, Alahi, and Fei-Fei, "Perceptual Losses for Real-Time Style Transfer and Super-Resolution", ECCV 2016. Copyright Springer, 2016. Reproduced for educational purposes.

Kod

Neural Style Transfer

Example outputs from

https://github.com/jcjohnson/neural-style

Gatys, Ecker, and Bethge, “Image style transfer using convolutional neural networks”, CVPR 2016
Figure copyright Justin Johnson, 2015.

Igra sa hiper-parametrima

Neural Style Transfer

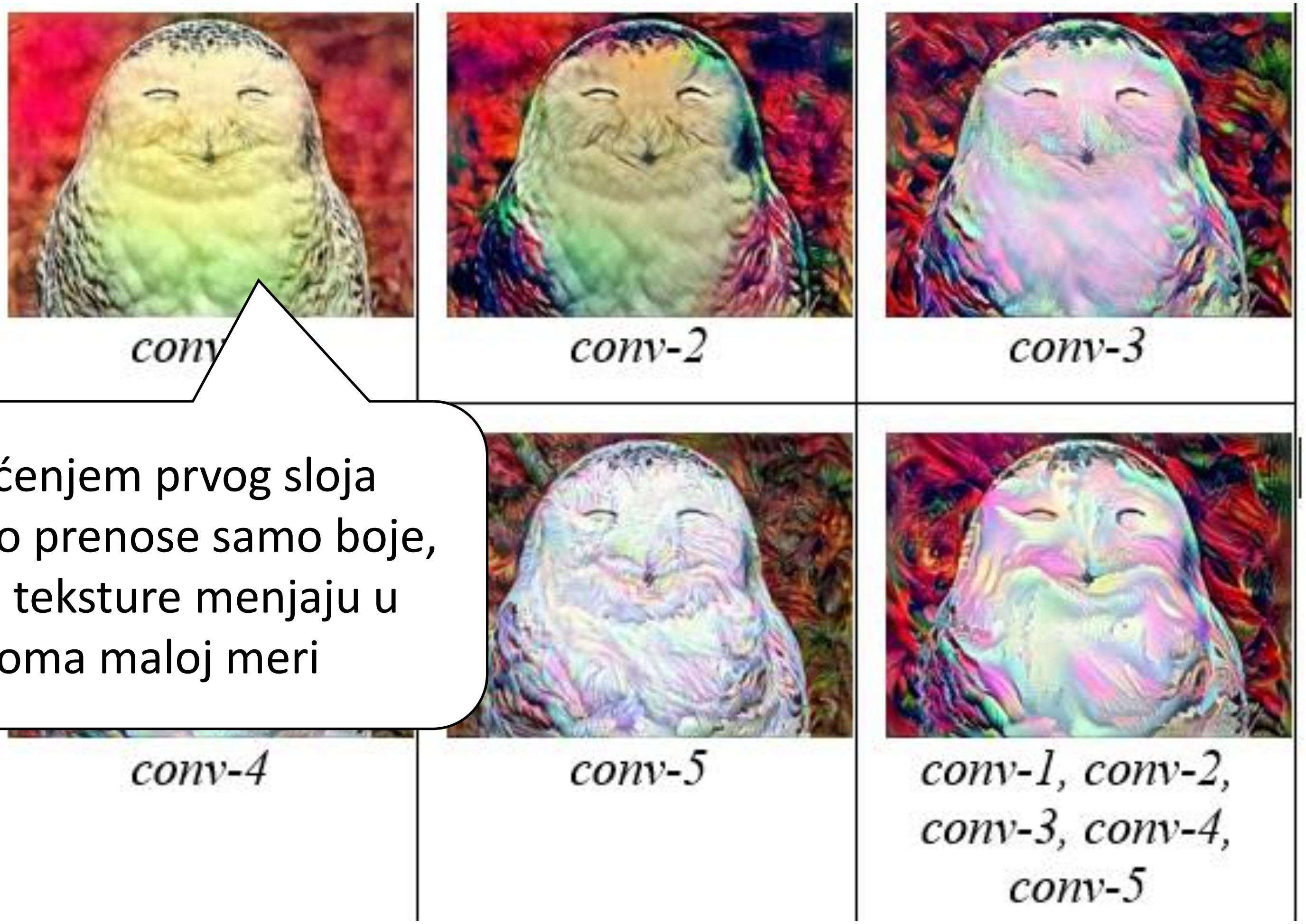
Uticaj odabira sloja na sadržaj slike

- Vizuelno gledano, najbolje slike se dobijaju tako što se za stilsku reprezentaciju iskoriste viši slojevi mreže

Na nižim slojevima mreže algoritam rekonstruiše detaljno informaciju o pikselima sadržajne slike.

Generisana slika je veoma slična sadržajnoj (izgleda kao „zamućena“ sadržajna slika)

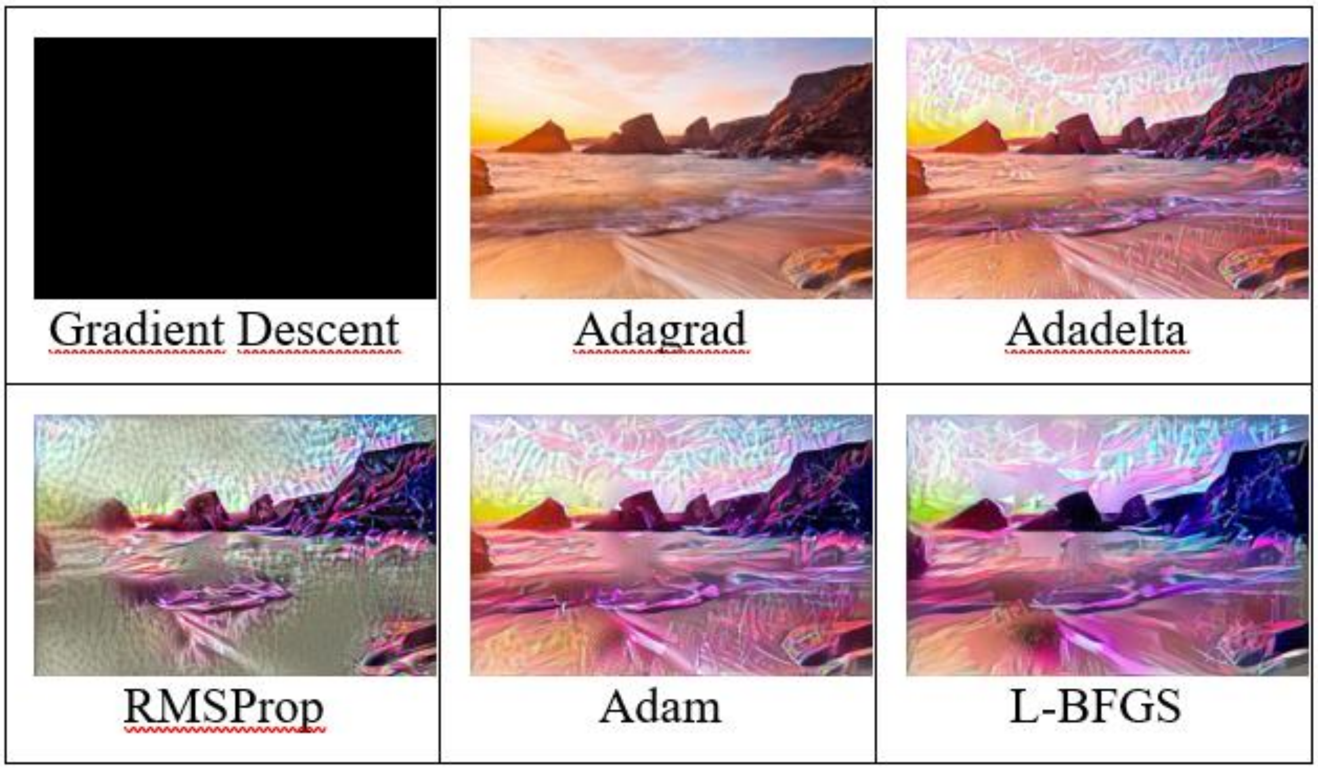
caj odabira sloja na sadržaj slike



Korišćenjem prvog sloja efektivno prenose samo boje, dok se teksture menjaju u veoma maloj meri

Poređenje različitih reprezentacija stila (sloja konvolucione neuronske mreže) za prenos stila

Izbor algoritma za iterativnu optimizaciju slike



Slika 4. *Poređenje različitih optimizera za prenos stila.*

Broj iteracija

 10 iteracija	 30 iteracija	 50 iteracija
 100 iteracija	 150 iteracija	 200 iteracija

Očuvanje boja

Slika 6. *Slika sadržaja (levo) i slika stila (desno)*

Slika 7. *Rezultat prenosa stila bez očuvanja originalnih boja slike sadržaja (levo) i sa očuvanjem originalnih boja (desno)*

Master rad Milan Keča

L. A. Gatys, M. Berthge, A. Hertzmann, E. Shechtman, Preserving Color in Neural Artistic Style Transfer, 2016.

Igra sa hiper-parametrima

Neural Style Transfer

Resizing style image before running style transfer algorithm can transfer different types of features

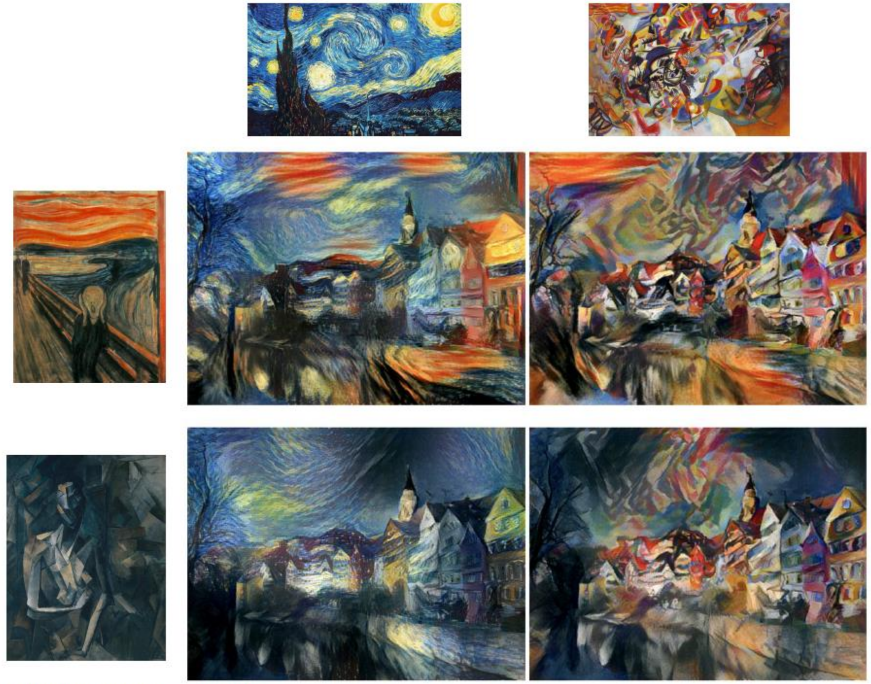
Larger style image ⟷ Smaller style image

Gatys, Ecker, and Bethge, "Image style transfer using convolutional neural networks", CVPR 2016
Figure copyright Justin Johnson, 2015.

Više stilova

Neural Style Transfer: Multiple Style Images

Mix style from multiple images by taking a weighted average of Gram matrices



Gatys, Ecker, and Bethge, "Image style transfer using convolutional neural networks", CVPR 2016
Figure copyright Justin Johnson, 2015.

Visoka rezolucija – *multi-scale processing*

Visoka rezolucija

Istovremeni prenos stila i *DeepDream*

Problem sa prenosom stila

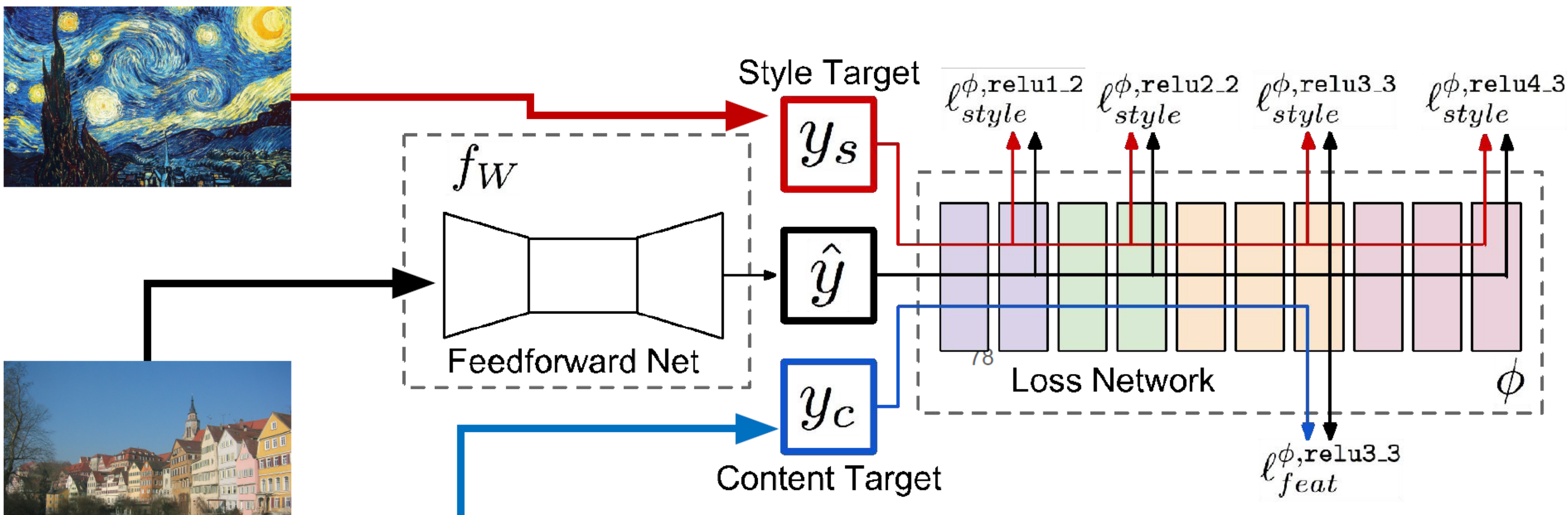
- Zahteva mnogo iteracija (forward/backward prolaza kroz VGG mrežu)
- Ovo je veoma sporo!

- Rešenje: trenirati drugu mrežu da radi prenos stila za nas

Brz prenos stila

Fast Style Transfer

(1) Train a feedforward network for each style
(2) Use pretrained CNN to compute same losses as before
(3) After training, stylize images using a single forward pass



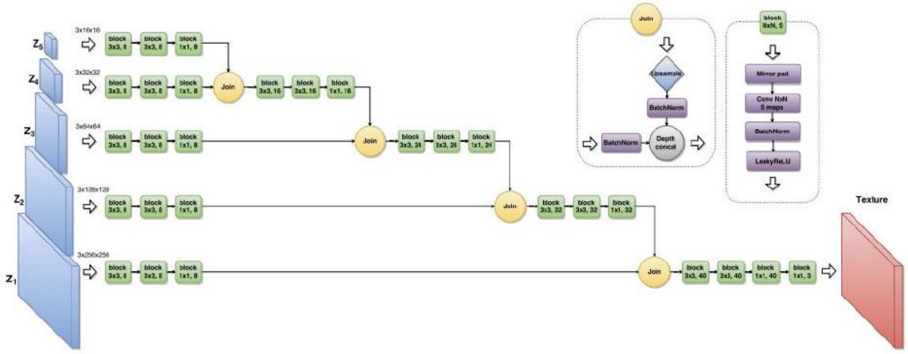
Johnson, Alahi, and Fei-Fei, "Perceptual Losses for Real-Time Style Transfer and Super-Resolution", ECCV 2016
Figure copyright Springer, 2016. Reproduced for educational purposes.

Brz prenos stila

https://github.com/jcjohnson/fast-neural-style

Brz prenos stila

Fast Style Transfer



Concurrent work from Ulyanov et al, comparable results

Ulyanov et al, "Texture Networks: Feed-forward Synthesis of Textures and Stylized Images", ICML 2016
Ulyanov et al, "Instance Normalization: The Missing Ingredient for Fast Stylization", arXiv 2016

Brz prenos stila

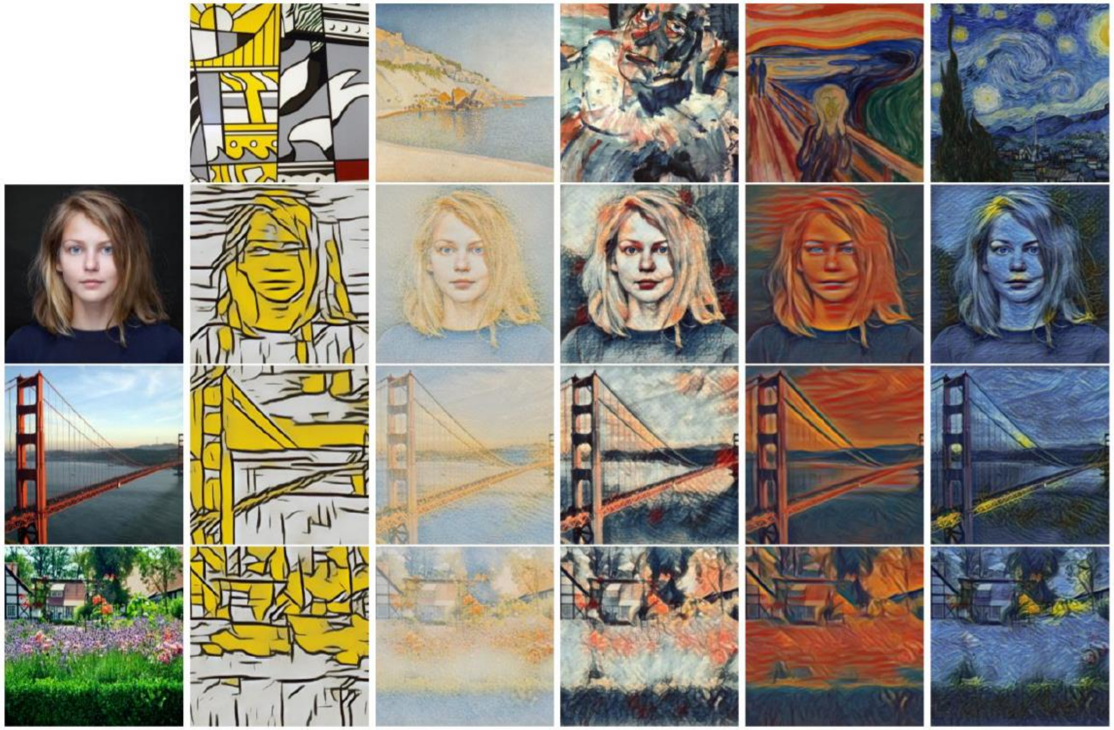
Fast Style Transfer

Replacing batch normalization with Instance Normalization improves results

Ulyanov et al, "Texture Networks: Feed-forward Synthesis of Textures and Stylized Images", ICML 2016
Ulyanov et al, "Instance Normalization: The Missing Ingredient for Fast Stylization", arXiv 2016

Brz prenos stila

One Network, Many Styles

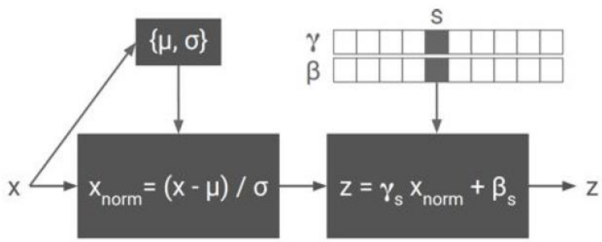


Dumoulin, Shlens, and Kudlur, "A Learned Representation for Artistic Style", ICLR 2017.

Style blending

One Network, Many Styles

Use the same network for multiple styles using *conditional instance normalization*: learn separate scale and shift parameters per style



Single network can blend styles after training

Dumoulin, Shlens, and Kudlur, "A Learned Representation for Artistic Style", ICLR 2017.