Vizuelizacija CNN obeležja

Stanford University

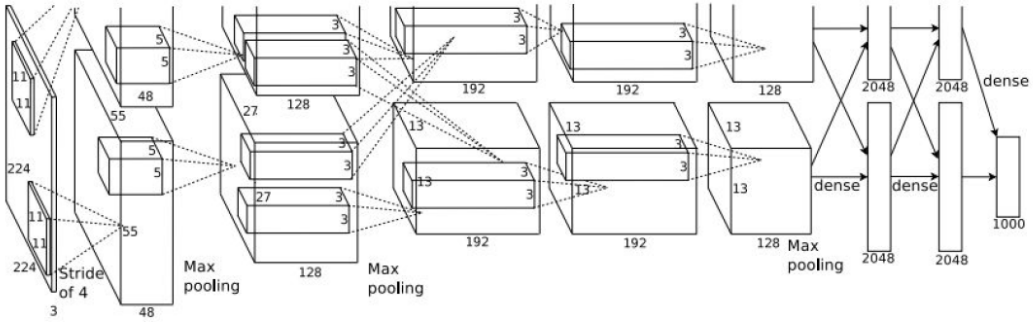
- https://www.youtube.com/watch?v=6wcs6szJWMY&list=PL3FW7Lu3i5JvHM8ljYj-zLfQRF3EO8sYv&index=12
- http://cs231n.github.io/understanding-cnn/

Šta rade središnji slojevi CNN?

What's going on inside ConvNets?

This image is CC0 public domain

Input Image:
3 x 224 x 224



What are the intermediate features looking for?

Class Scores:
1000 numbers

Krizhevsky et al, "ImageNet Classification with Deep Convolutional Neural Networks", NIPS 2012.
Figure reproduced with permission.

Prvi sloj

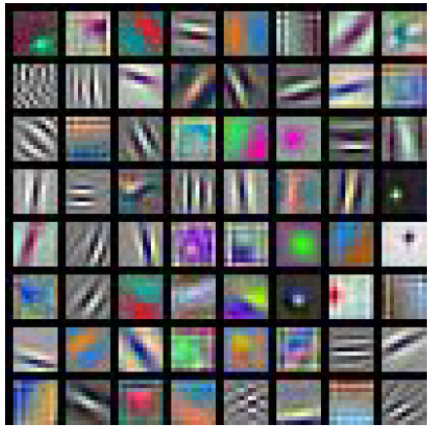
- Pošto vršimo direktan skalarni proizvod težina filtera sa ulazima (pikselima), možemo dobiti osećaj šta ovi filteri traže tako što prosto vizualizujemo težine filtera kao slike

- Npr. u AlexNet se prvi konvolucioni sloj sastoji od 64 filtera dimenzije $11 \times 11 \times 3$

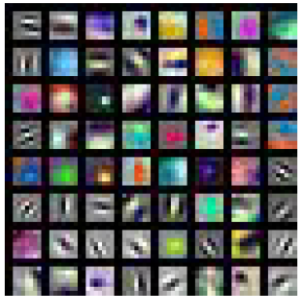
- Svaki od filtera dimenzije $11 \times 11 \times 3$ vizualizujemo kao sliku dimenzije 11×11 sa 3 kanala boja

Prvi sloj

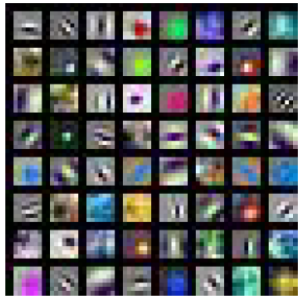
- Na slici su prikazani filteri prvog sloja uzeti iz različitih treniranih modela AlexNet, ResNet,...
- Interesantno je da, bez obzira na arhitekturu, prvi slojevi traže slične stvari na ulaznoj slici – orijentisane ivice i kontrastne boje (Hubel & Wiesel)



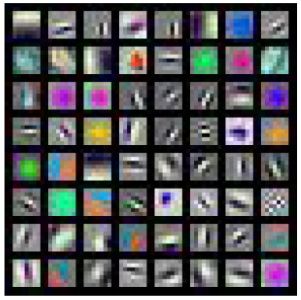
AlexNet:
64 x 3 x 11 x 11



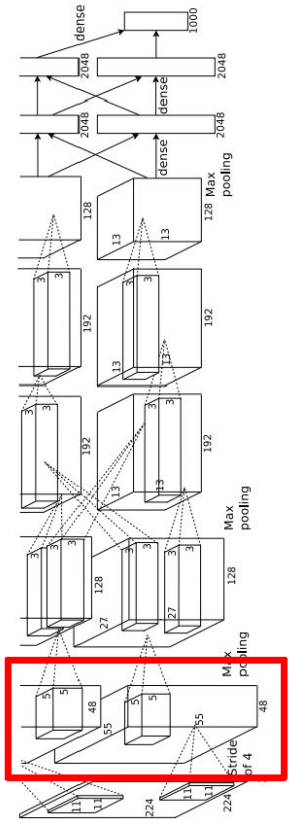
ResNet-18:
64 x 3 x 7 x 7



ResNet-101:
64 x 3 x 7 x 7



DenseNet-121:
64 x 3 x 7 x 7



Krizhevsky, "One weird trick for parallelizing convolutional neural networks", arXiv 2014
He et al, "Deep Residual Learning for Image Recognition", CVPR 2016
Huang et al, "Densely Connected Convolutional Networks", CVPR 2017

Središnji slojevi

- Dosta teži za interpretaciju jer ih ne možemo ih direktno interpretirati kao slike. Na primer,
  - Prvi sloj: 16 filtera dimenzije $7 \times 7 \times 3$
  - Drugi sloj: 20 filtera dimenzije $7 \times 7 \times \mathbf{16}$ – svaki filter ima dubinu 16
  - Težine možemo vizualizovati kao 16 *grayscale* slika dimenzije 7×7
  - Ali, ovo i dalje nije dobra intuicija šta filteri tačno traže *na ulazu*, jer filteri nisu konektovani direktno na ulaznu sliku
  - Ovo nam daje intuiciju kakav tip aktivacionog šablona *u prvom sloju* će uzrokovati maksimalnu aktivaciju neurona drugog sloja

- Napomena za vizualizacije
  - Težine su skalirane na 0-255 (inače ne moraju biti ograničene)
  - Bias nije uključen

Središnji slojevi su manje interpretabilni

Demo CNN na stanfordovom sajtu

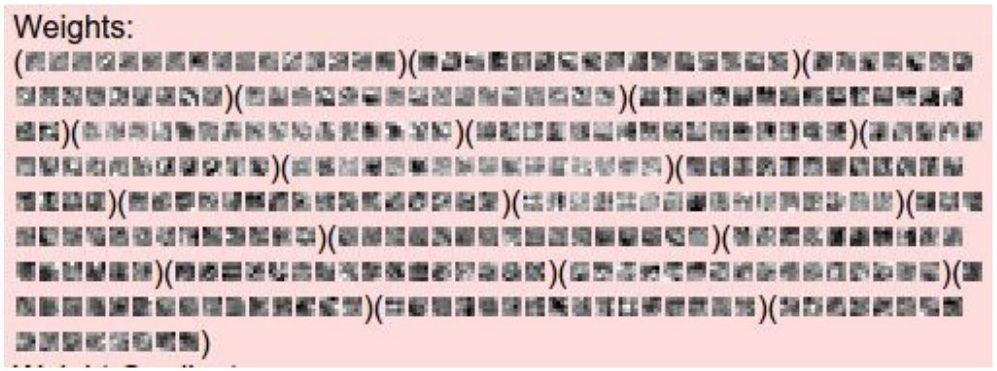
Visualize the filters/kernels (raw weights)

We can visualize filters at higher layers, but not that interesting

(these are taken from ConvNetJS CIFAR-10 demo)

layer 1 weights

16 x 3 x 7 x 7



layer 2 weights

20 x 16 x 7 x 7

layer 3 weights

20 x 20 x 7 x 7

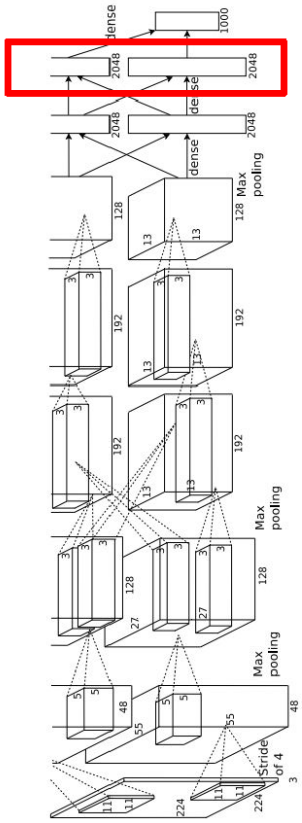
Poslednji sloj (pre izlaza)

- Izlazni sloj interpretiramo kao verovatnoće svake klase
  - Npr. na ImageNet imamo 1000 klasa, tako da bi mreža trenirana na ovom skupu imala 1000 neurona na izlazu
- Pre izlaznog sloja obično imamo *fully connected* sloj
  - Npr. u AlexNet imamo 4096 neurona koji se vezuju na 1000 neurona izlaza
- Još jedan način da se rešava problem vizuelizacije i razumevanja CNN jeste da se gleda šta se dešava sa tim poslednjim slojem od 4096 neurona

Poslednji sloj

Last Layer

FC7 layer



4096-dimensional feature vector for an image (layer immediately before the classifier)

Run the network on many images, collect the feature vectors

Razumevanje poslednjeg sloja

- *Nearest Neighbor* (NN) pristup

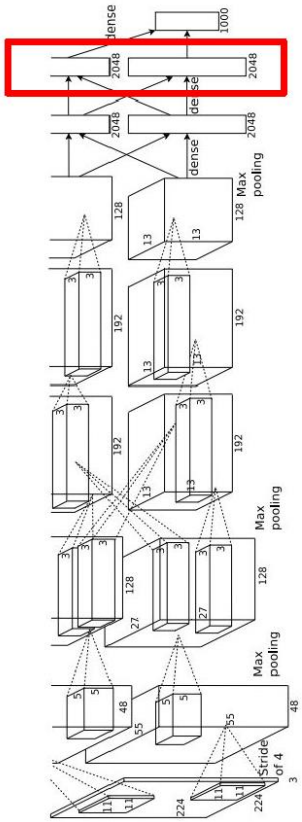
- Ranije smo prikazali da traženje najbližih suseda među slikama trening skupa koristeći *sirove piksele* ne daje zadovoljavajuće rešenje
  - Npr. za sliku belog psa – dobijemo slike sa belim „blobs“
  - Slike su slične po izgledu, ali nisu *semantički* slične

- Umesto da tražimo najbliže susede u prostoru piksela, tražimo ih u 4096 dimenzionom prostoru obeležja izračunatom od strane CNN

Razumevanje poslednjeg sloja

Last Layer: Nearest Neighbors

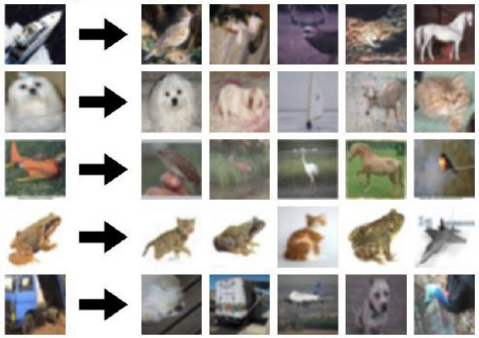
4096-dim vector

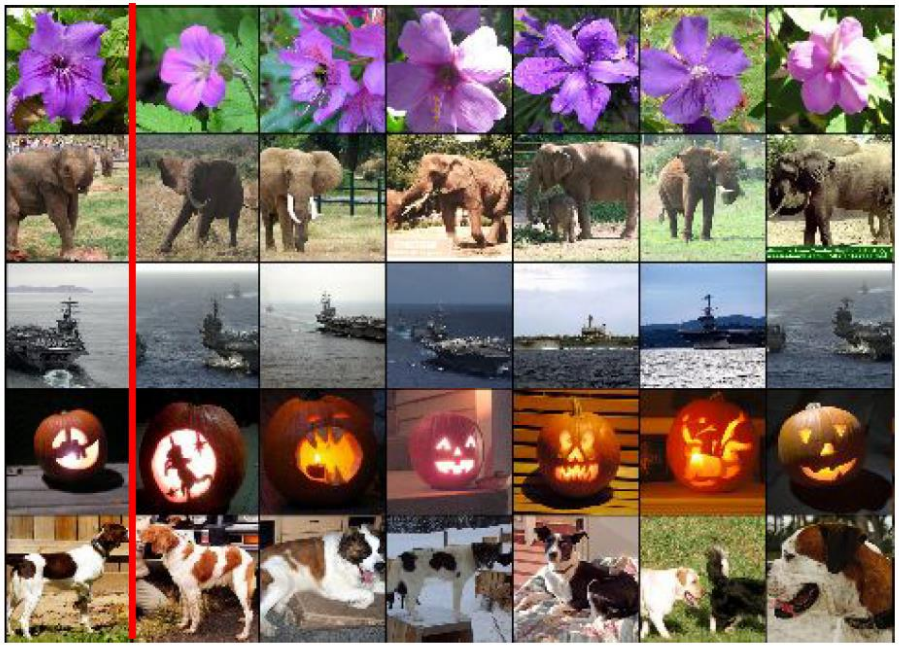


Test image

L2 Nearest neighbors in feature space

Recall: Nearest neighbors in pixel space





Krizhevsky et al, "ImageNet Classification with Deep Convolutional Neural Networks", NIPS 2012.
Figures reproduced with permission.

Razumevanje poslednjeg sloja

- Rezultati
  - Sirovi pikseli su dosta različiti kod slika i njihovih najbližih suseda, ali je semantički sadržaj veoma blizak
  - Npr. ulazna slika je slon okrenut na desno, ali 4. najbliži sused slon okrenut na levo – pikeseli ove dve slike su gotovo u potpunosti različiti, ali su ove slike bliske u prostoru obeležja kojeg je mreža naučila i zadovoljavajući rezultat je da su semantički slične

Razumevanje poslednjeg sloja

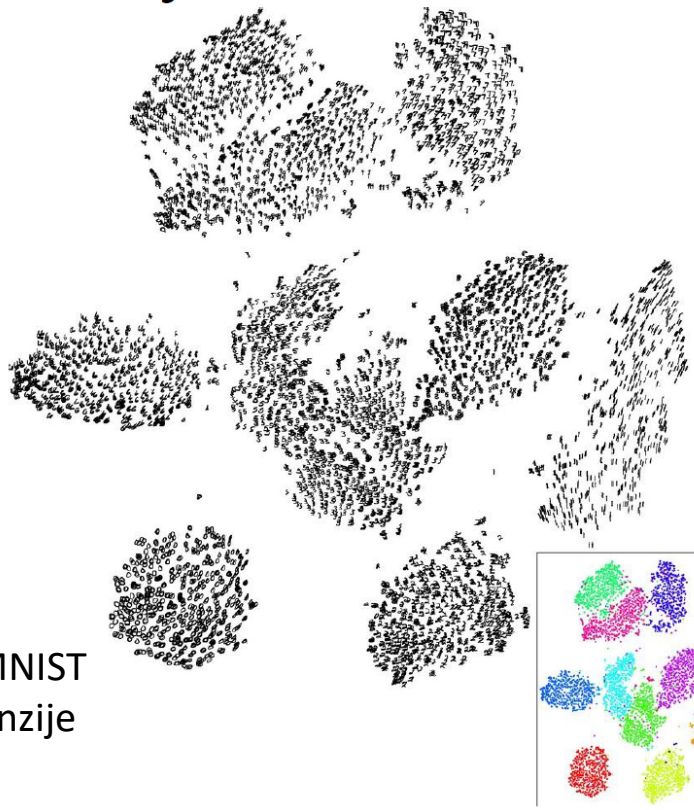
Last Layer: Dimensionality Reduction

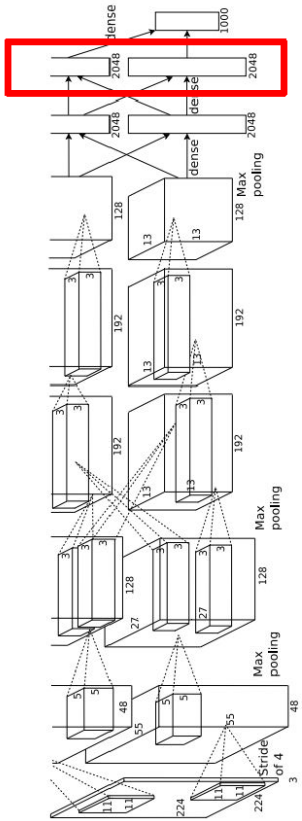
Visualize the “space” of FC7 feature vectors by reducing dimensionality of vectors from 4096 to 2 dimensions

Simple algorithm: Principle Component Analysis (PCA)

More complex: **t-SNE**

- t-SNE primenjen na sirove piksele MNIST
- 28 × 28 slike prikazujemo u 2 dimenzije

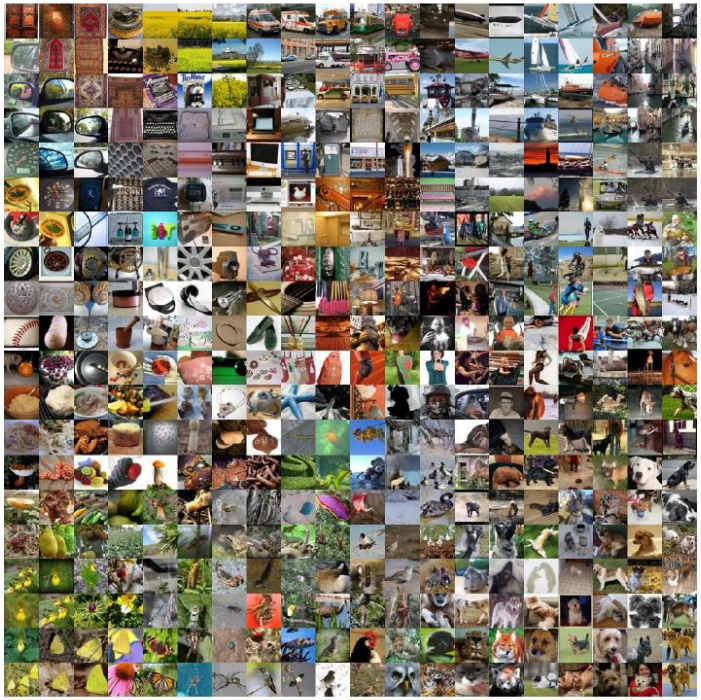


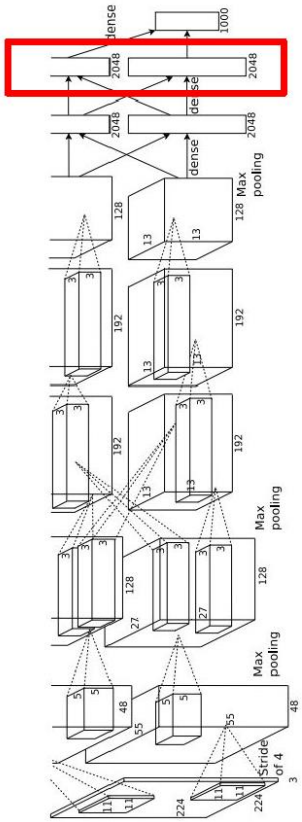


Van der Maaten and Hinton, “Visualizing Data using t-SNE”, JMLR 2008
Figure copyright Laurens van der Maaten and Geoff Hinton, 2008. Reproduced with permission.

Razumevanje poslednjeg sloja

Last Layer: Dimensionality Reduction





Van der Maaten and Hinton, "Visualizing Data using t-SNE", JMLR 2008
Krizhevsky et al, "ImageNet Classification with Deep Convolutional Neural Networks", NIPS 2012.
Figure reproduced with permission.

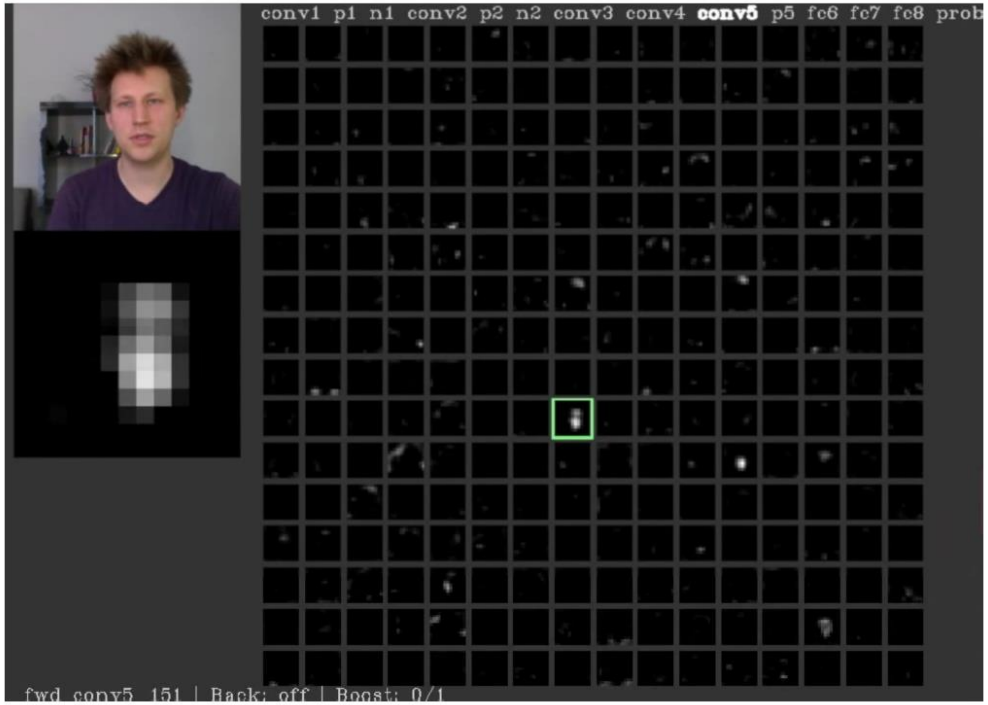
See high-resolution versions at
http://cs.stanford.edu/people/karpathy/cnnembed/

Razumevanje: aktivacije srednjih slojeva

- Vizuelizacija aktivacionih mapa srednjih slojeva jeste interpretabilna u nekim slučajevima
- Alat pušta CNN u realnom vremenu na sliku dobijenu putem kamere i vizualizuje aktivacije odabranog sloja

conv5 feature map is 128x13x13; visualize as 128 13x13 grayscale images

Označeno obeležje deluje kao da se aktivira na delove koji odgovaraju licu osobe

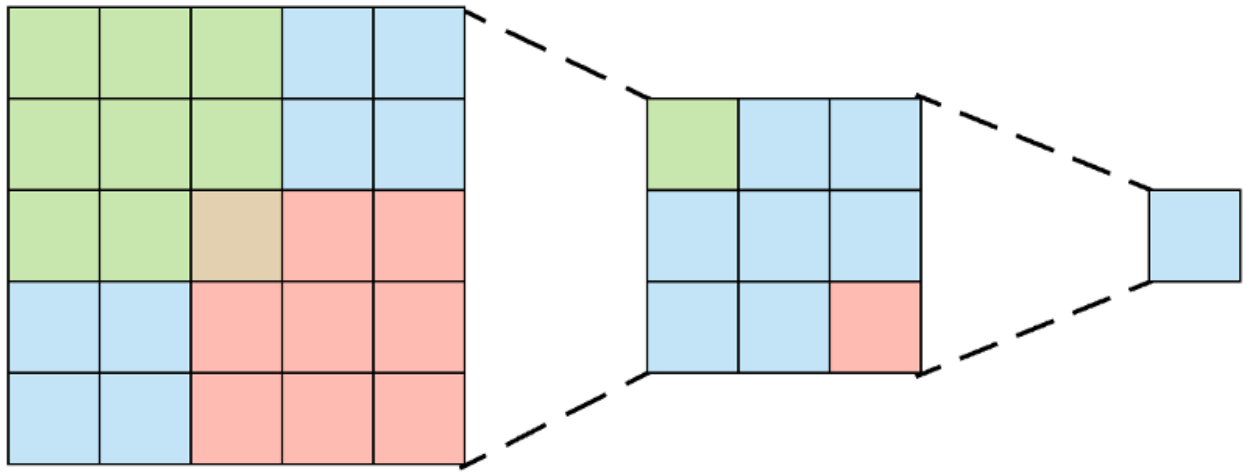


Yosinski et al, "Understanding Neural Networks Through Deep Visualization", ICML DL Workshop 2014.
Figure copyright Jason Yosinski, 2014. Reproduced with permission.

Maximally Activating Patches

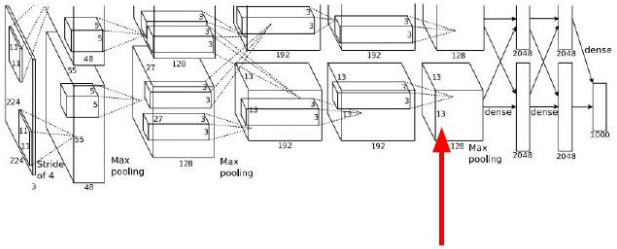
- Još jedan način vizuelizacije je da vidimo koji tipovi isečaka ulazne slike uzrokuju maksimalnu aktivaciju u različitim neuronima

- Neuroni srednjih slojeva ne gledaju celu sliku već neko manje receptivno polje. Zato vizualizujemo isečke koji ih maksimalno aktiviraju



Maximally Activating Patches

Maximally Activating Patches

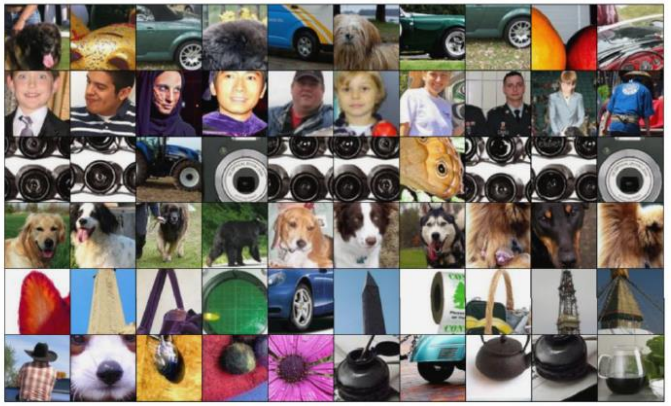


Pick a layer and a channel; e.g. conv5 is 128 x 13 x 13, pick channel 17/128

Run many images through the network, record values of chosen channel

Visualize image patches that correspond to maximal activations

Svaki red predstavlja jedan sloj mreže (jedan neuron) i ovo su isečci iz skupa podataka koji su maksimalno aktivirali neuron, sortirani po jačini aktivacije



Springenberg et al, "Striving for Simplicity: The All Convolutional Net", ICLR Workshop 2015
Figure copyright Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, Martin Riedmiller, 2015; reproduced with permission.

Eksperiment sa zaklanjanjem

- Ideja:
  - želimo da vidimo koji delovi ulazne slike uzrokuju da CNN napravi klasifikacionu odluku
  - Ako zaklanjanje određenog dela slike uzrokuje drastičnu promenu verovatnoće, onda je taj deo ulazne slike verovatno veoma važan za klasifikaciju

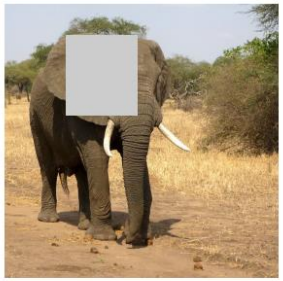
- Postupak:
  - Deo slike ćemo zakloniti (npr. zameniti srednjom vrednošću piksela skupa podataka) i propustiti modifikovanu sliku kroz mrežu. Zabeležićemo verovatnoću klasifikacije (modifikovane) slike u odgovarajuću klasu
  - Ponavljamo proces tako što pomeramo zaklanjajući isečak preko cele slike
  - Iscrtavamo *heath map* koji pokazuje sa kolikom verovatnoćom je CNN klasifikovala sliku kao funkciju toga koji deo slike je zaklonjen

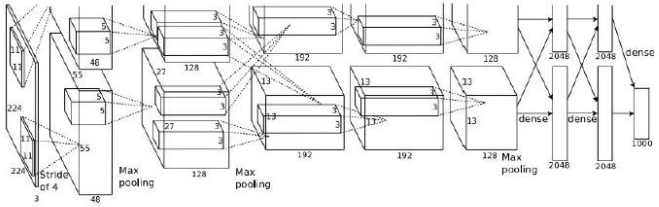
Eksperiment sa zaklanjanjem

Crvena boja: niska verovatnoća

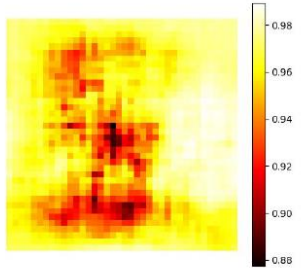
Occlusion Experiments

Mask part of the image before feeding to CNN, draw heatmap of probability at each mask location

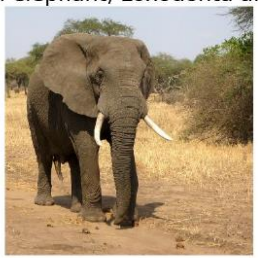


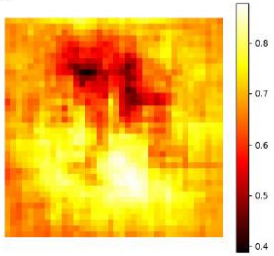


schooner

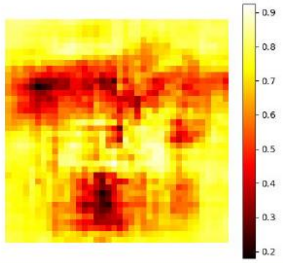


African elephant, Loxodonta africana





go-kart



Zeiler and Fergus, "Visualizing and Understanding Convolutional Networks", ECCV 2014

Saliency Maps

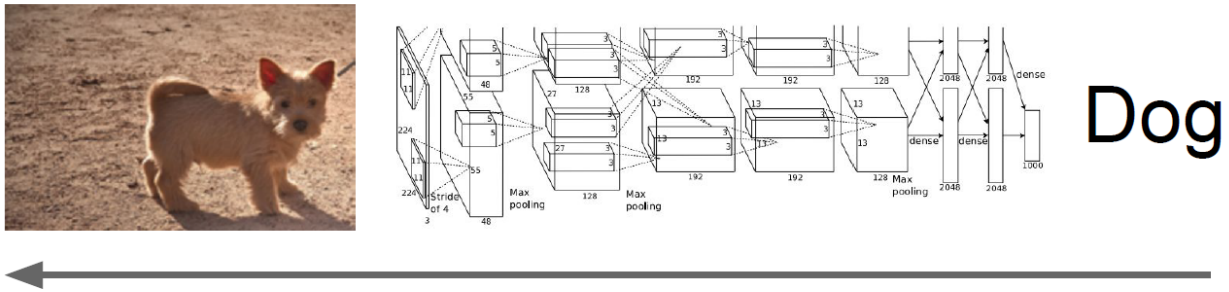
- Slična ideja: ako imamo ulaznu sliku psa i prediktovanu labelu „pas“, želimo da znamo koji pikseli u ulaznoj slici su važni za klasifikaciju

- Postupak:
  - Računamo gradijent od verovatnoće prediktovane klase po vrednostima piksela ulazne slike
  - Uvid: ako malo promenimo vrednost piksela ulazne slike, koliko će se promeniti verovatnoća klasifikacije slike?

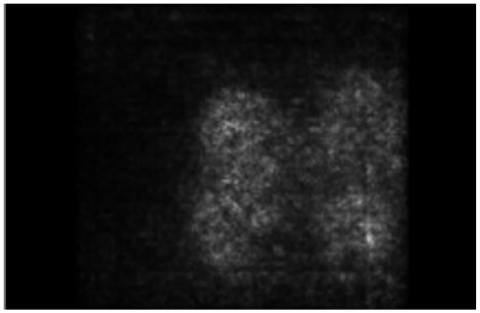
Saliency Maps

Saliency Maps

How to tell which pixels matter for classification?



Compute gradient of (unnormalized) class score with respect to image pixels, take absolute value and max over RGB channels



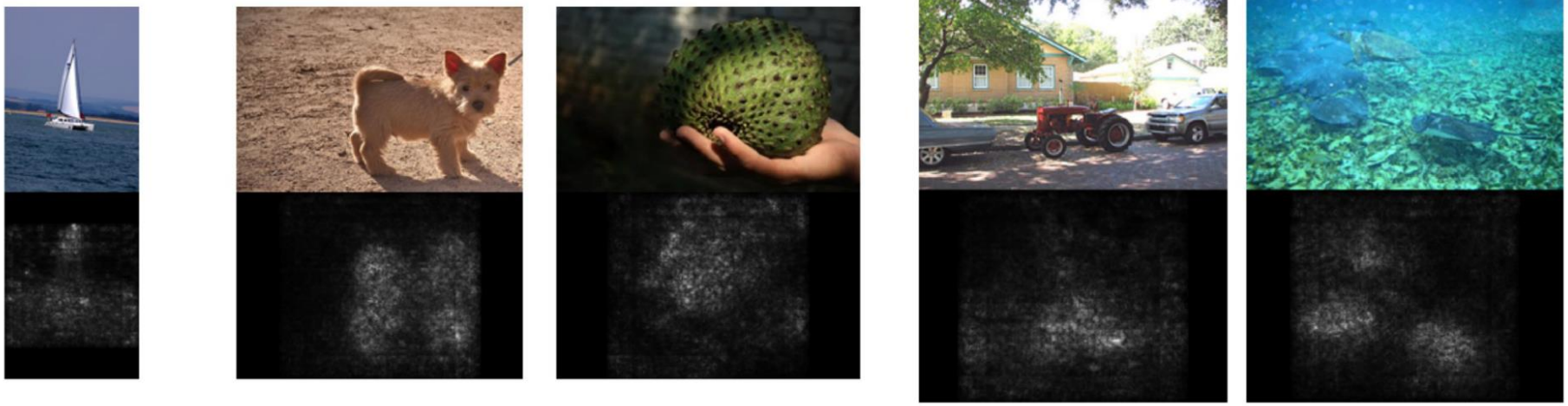
Pikseli važni za klasifikaciju odgovaraju pikselima psa

Simonyan, Vedaldi, and Zisserman, "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps", ICLR Workshop 2014.

Saliency Maps

Saliency Maps

Deluje da mreža prilikom klasifikacione odluke „gleda dobre regije“

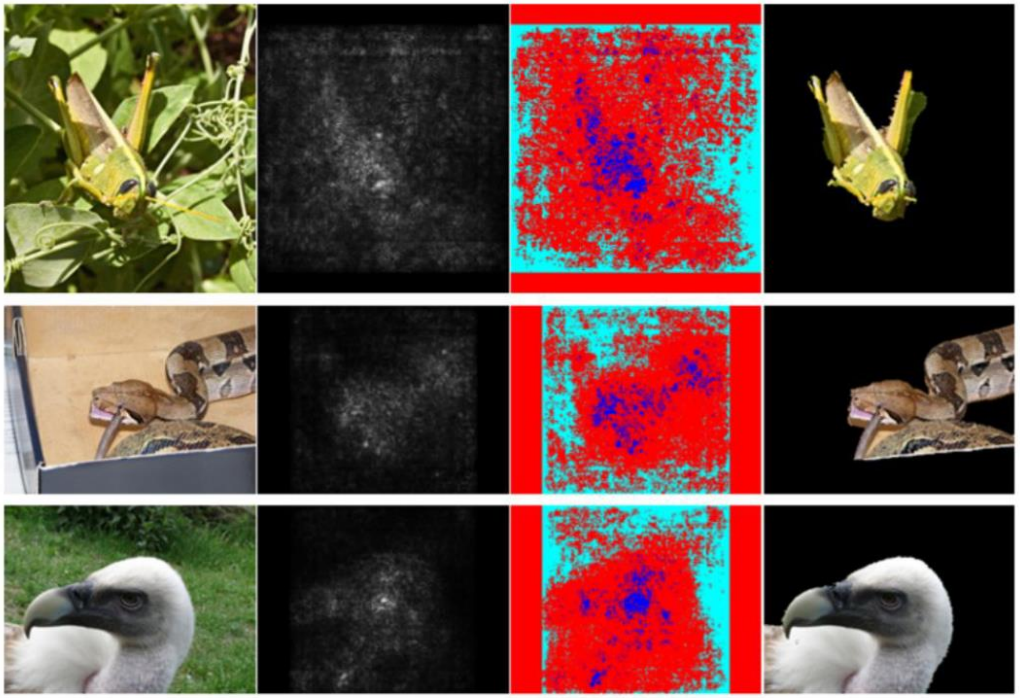


Simonyan, Vedaldi, and Zisserman, "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps", ICLR Workshop 2014.
Figures copyright Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman, 2014; reproduced with permission.

Saliency Maps

Saliency Maps: Segmentation without supervision

Use GrabCut on saliency map



Simonyan, Vedaldi, and Zisserman, "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps", ICLR Workshop 2014.
Figures copyright Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman, 2014; reproduced with permission.
Rother et al, "Grabcut: Interactive foreground extraction using iterated graph cuts", ACM TOG 2004

Guided backprop

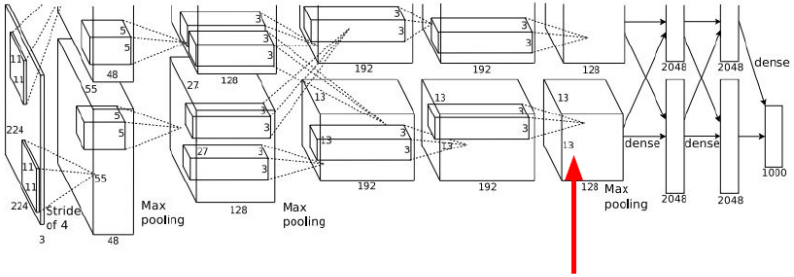
- Za konkretnu sliku, umesto da gledamo *class score*, želimo da odaberemo neki središnji neuron u mreži i da vidimo koji delovi ulazne slike imaju uticaja na aktivaciju tog neurona

- Takođe računamo *saliency map*, ali, umesto da računamo gradijent *class score* u odnosu na piksele slike, računamo gradijent aktivacije (izlaza) određenog neurona u odnosu na vrednosti piksela slike

- Dobijamo uvid koji tačno pikseli slike imaju uticaja na konkretan neuron
  - Pritom koristimo standardnu propagaciju u nazad

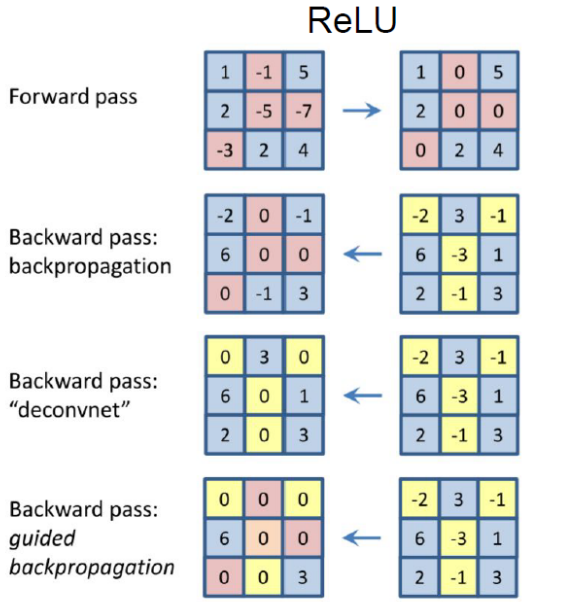
Guided backprop

Intermediate features via (guided) backprop



Pick a single intermediate neuron, e.g. one value in 128 x 13 x 13 conv5 feature map

Compute gradient of neuron value with respect to image pixels



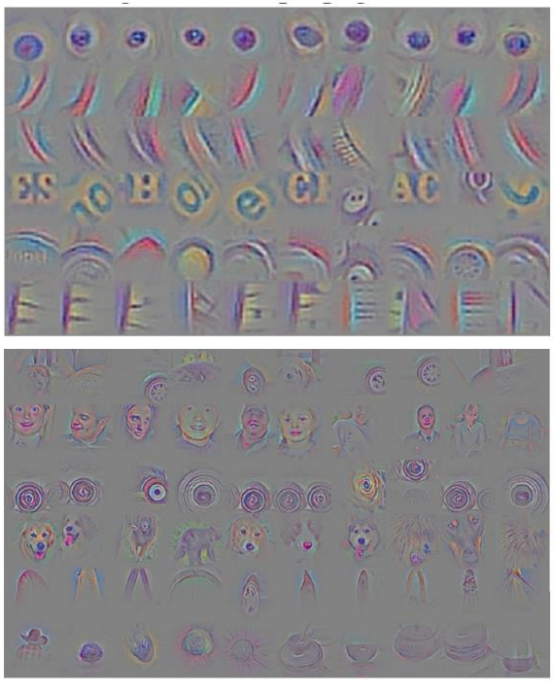
Images come out nicer if you only backprop positive gradients through each ReLU (guided backprop)

Zeiler and Fergus, "Visualizing and Understanding Convolutional Networks", ECCV 2014
Springenberg et al, "Striving for Simplicity: The All Convolutional Net", ICLR Workshop 2015

Guided backprop

Intermediate features via (guided) backprop

(guided) backprop



Maximally Activating Patches

Zeiler and Fergus, "Visualizing and Understanding Convolutional Networks", ECCV 2014
Springenberg et al, "Striving for Simplicity: The All Convolutional Net", ICLR Workshop 2015

Gradient Ascent

- *Saliency maps* i *guided backprop* su funkcije fiksiranog ulaza
  - To jest, za fiksiranu sliku nam kažu koje vrednosti piksela imaju uticaja na neuron
  - Voleli bismo da uklonimo oslonac na konkretnu sliku i da vidimo koji tip ulaza bi generalno uzrokovao da se određeni neuron aktivira

- Ovo možemo postići tehnikom *gradient ascent*
  - Standardno koristimo *gradient descent* da *minimizujemo* gubitak tako što menjamo težine
  - Umesto toga, ovde fiksiramo težine i sintetišemo sliku korišćenjem *gradient ascent* da probamo da *maksimizujemo* score nekog središnjeg neurona (ili klase)

Gradient Ascent

Visualizing CNN features: Gradient Ascent

(Guided) backprop:
Find the part of an image that a neuron responds to

Gradient ascent:
Generate a synthetic image that maximally activates a neuron

$$I^* = \arg\max_I f(I) + R(I)$$

Neuron value

Natural image regularizer

Gradient Ascent

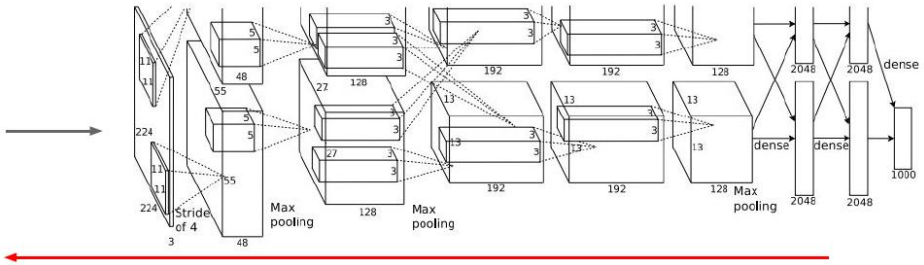
Visualizing CNN features: Gradient Ascent

$$\arg\max_I S_c(I) - \lambda \|I\|_2^2$$

score for class c (before Softmax)

1. Initialize image to zeros

zero image



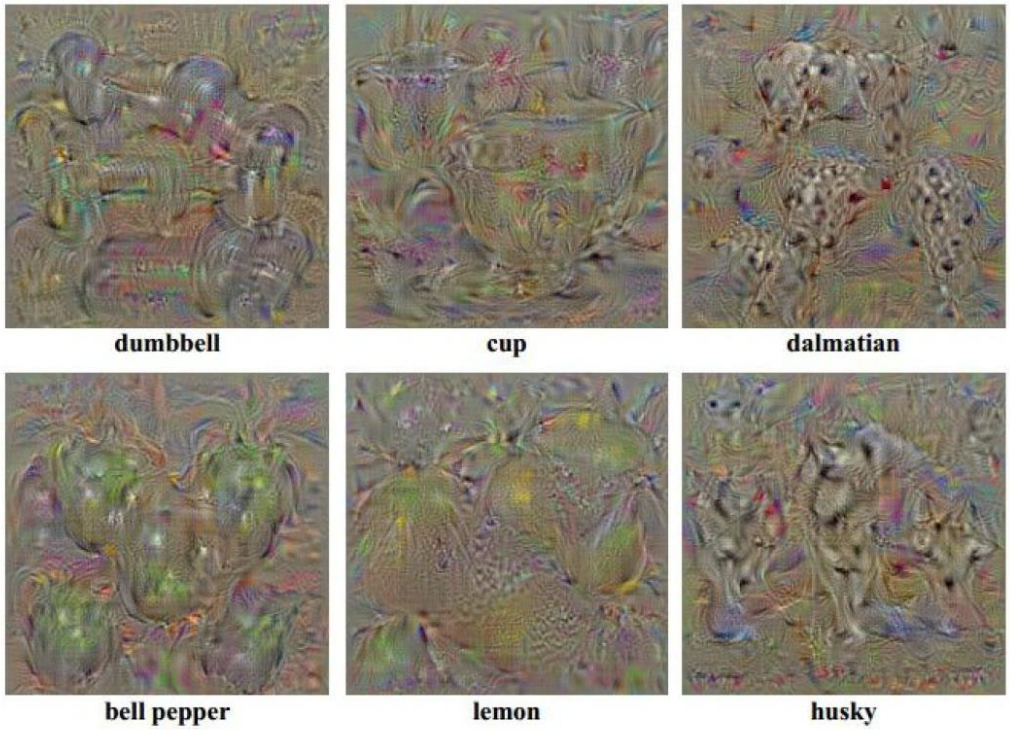
Repeat:
2. Forward image to compute current scores
3. Backprop to get gradient of neuron value with respect to image pixels
4. Make a small update to the image

Regularizacija – L2 norma

Visualizing CNN features: Gradient Ascent

$$\arg\max_I S_c(I) - \lambda \|I\|_2^2$$

Simple regularizer: Penalize L2 norm of generated image



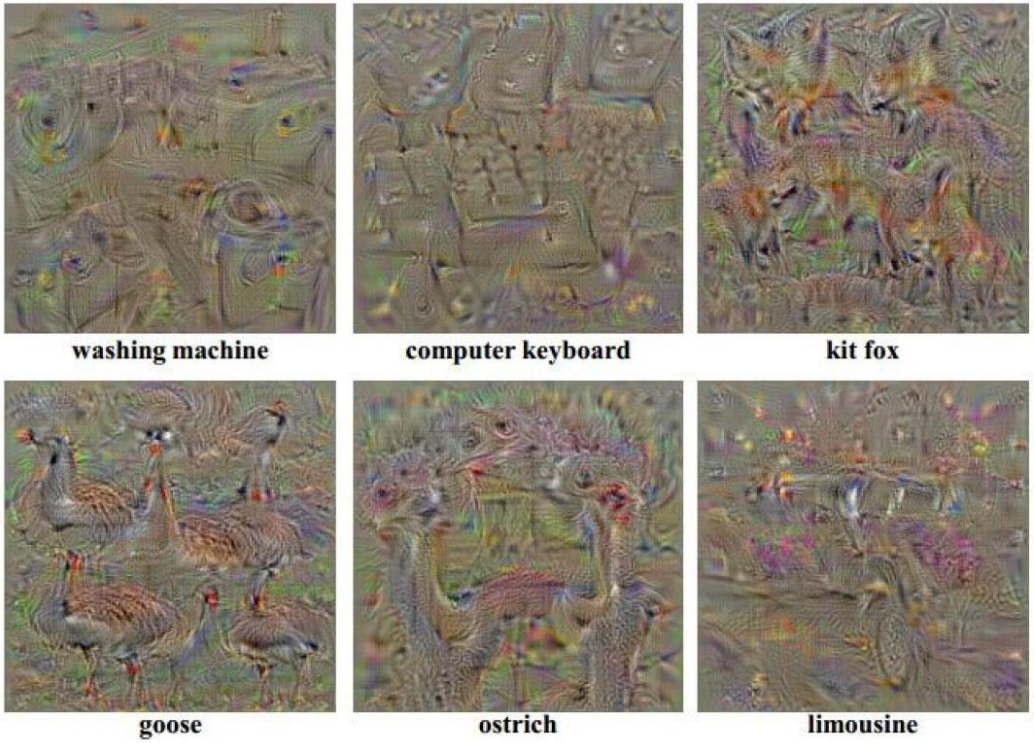
Simonyan, Vedaldi, and Zisserman, "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps", ICLR Workshop 2014.
Figures copyright Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman, 2014; reproduced with permission.

Regularizacija – L2 norma

Visualizing CNN features: Gradient Ascent

$$\arg\max_I S_c(I) - \lambda \|I\|_2^2$$

Simple regularizer: Penalize L2 norm of generated image



Yosinski et al, "Understanding Neural Networks Through Deep Visualization", ICML DL Workshop 2014. Figure copyright Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson, 2014. Reproduced with permission.

Popravka regularizacije

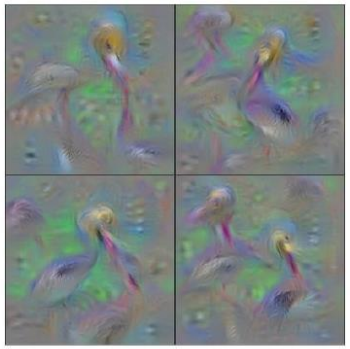
Visualizing CNN features: Gradient Ascent

$$\arg\max_I S_c(I) - \lambda \|I\|_2^2$$

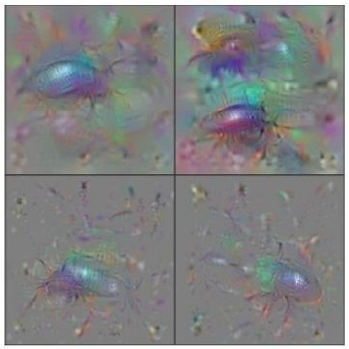
Better regularizer: Penalize L2 norm of image; also during optimization periodically

(1) Gaussian blur image
(2) Clip pixels with small values to 0
(3) Clip pixels with small gradients to 0

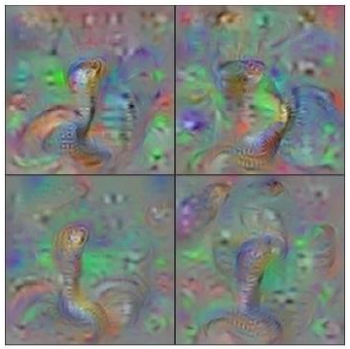
Flamingo



Pelican



Ground Beetle



Indian Cobra

Yosinski et al, "Understanding Neural Networks Through Deep Visualization", ICML DL Workshop 2014.
Figure copyright Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson, 2014. Reproduced with permission.

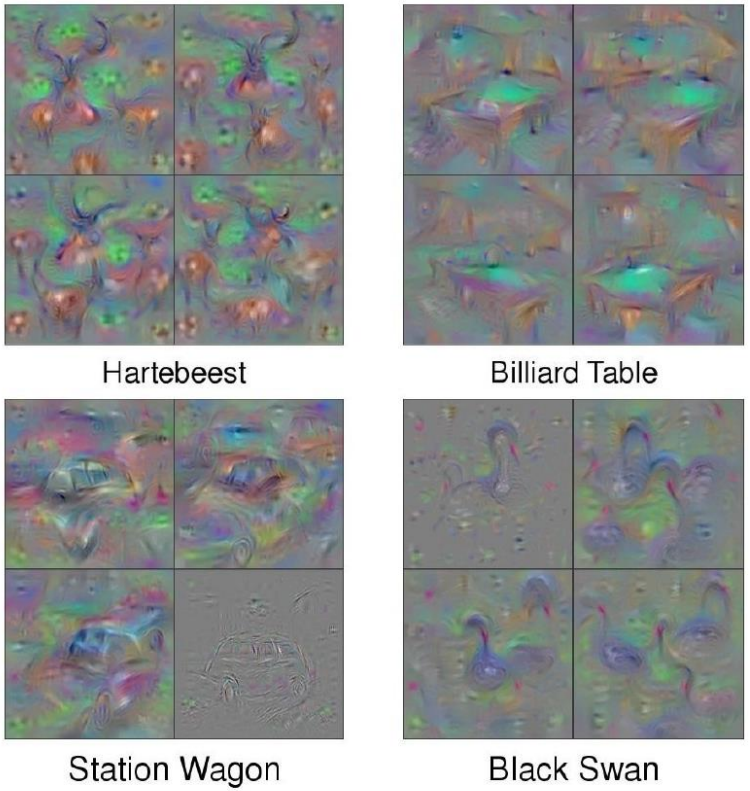
Popravka regularizacije

Visualizing CNN features: Gradient Ascent

$$\arg\max_I S_c(I) - \lambda \|I\|_2^2$$

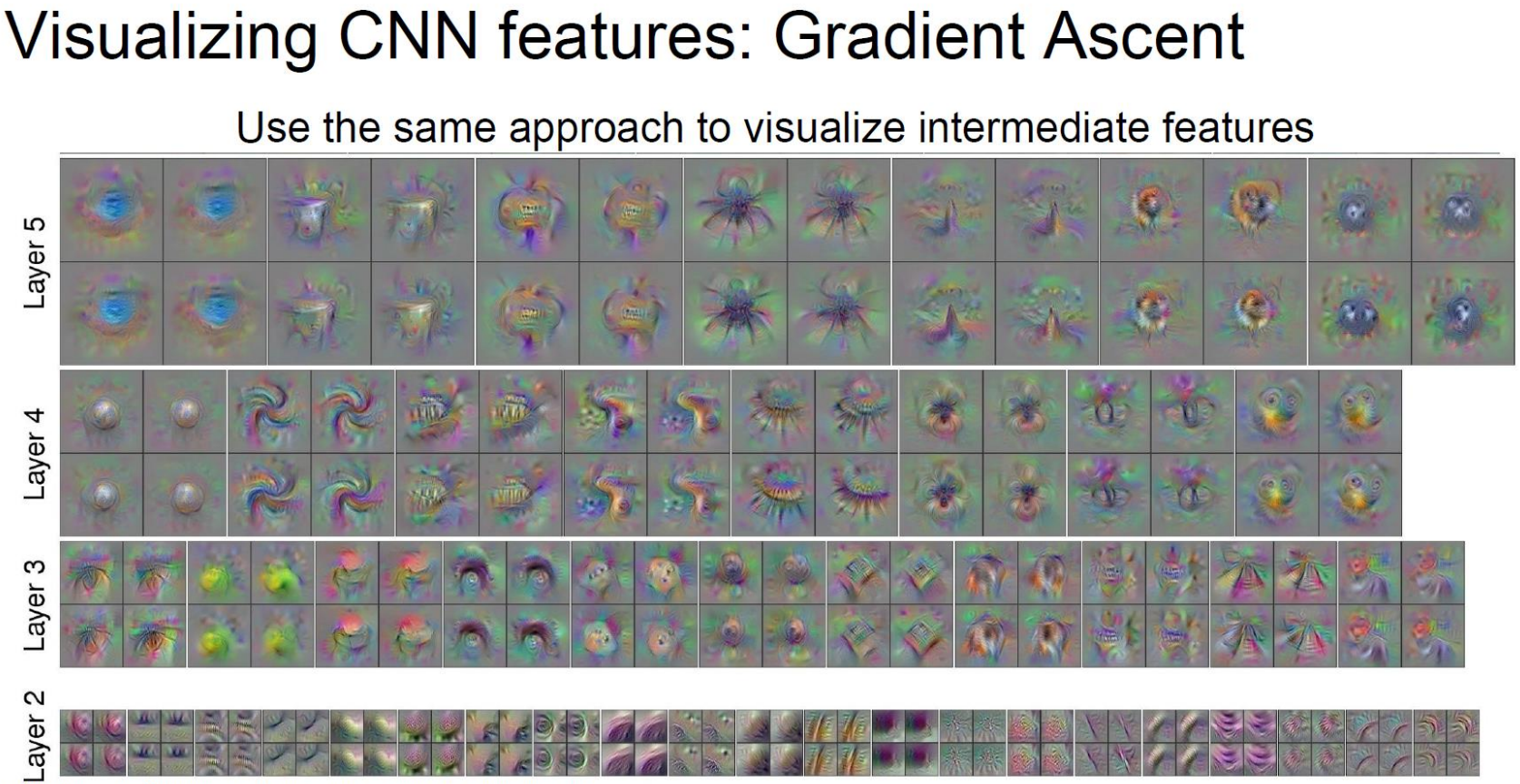
Better regularizer: Penalize L2 norm of image; also during optimization periodically

(1) Gaussian blur image
(2) Clip pixels with small values to 0
(3) Clip pixels with small gradients to 0



Yosinski et al, "Understanding Neural Networks Through Deep Visualization", ICML DL Workshop 2014.
Figure copyright Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson, 2014. Reproduced with permission.

Popravka regularizacije



Yosinski et al, "Understanding Neural Networks Through Deep Visualization", ICML DL Workshop 2014.
Figure copyright Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson, 2014. Reproduced with permission.

Popravka regularizacije

Visualizing CNN features: Gradient Ascent

Adding “multi-faceted” visualization gives even nicer results:
(Plus more careful regularization, center-bias)

Reconstructions of multiple feature types (facets) recognized by the same “grocery store” neuron

Corresponding example training set images recognized by the same neuron as in the "grocery store" class

Nguyen et al, “Multifaceted Feature Visualization: Uncovering the Different Types of Features Learned By Each Neuron in Deep Neural Networks”, ICML Visualization for Deep Learning Workshop 2016.
Figures copyright Anh Nguyen, Jason Yosinski, and Jeff Clune, 2016; reproduced with permission.

Popravka regularizacije

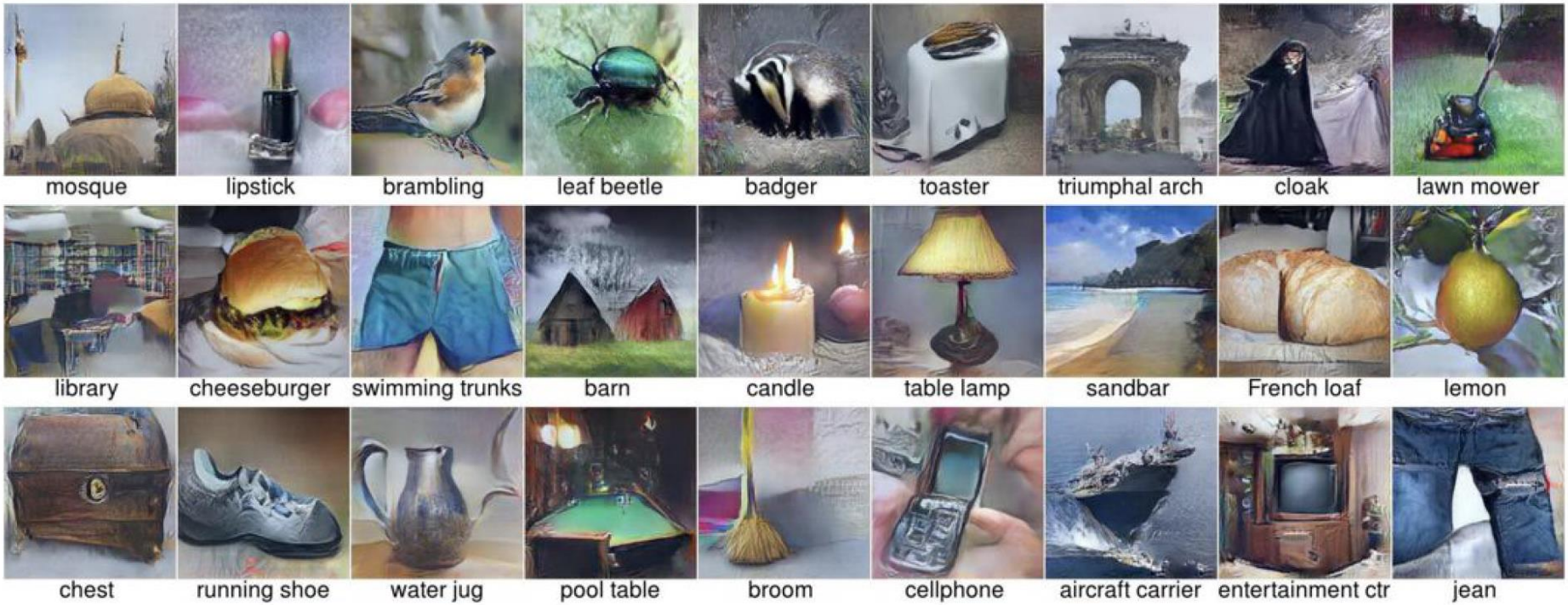
Visualizing CNN features: Gradient Ascent

Nguyen et al, "Multifaceted Feature Visualization: Uncovering the Different Types of Features Learned By Each Neuron in Deep Neural Networks", ICML Visualization for Deep Learning Workshop 2016.
Figures copyright Anh Nguyen, Jason Yosinski, and Jeff Clune, 2016; reproduced with permission.

Popravka regularizacije

Visualizing CNN features: Gradient Ascent

Optimize in FC6 latent space instead of pixel space:



Nguyen et al, "Synthesizing the preferred inputs for neurons in neural networks via deep generator networks," NIPS 2016
Figure copyright Nguyen et al, 2016; reproduced with permission.

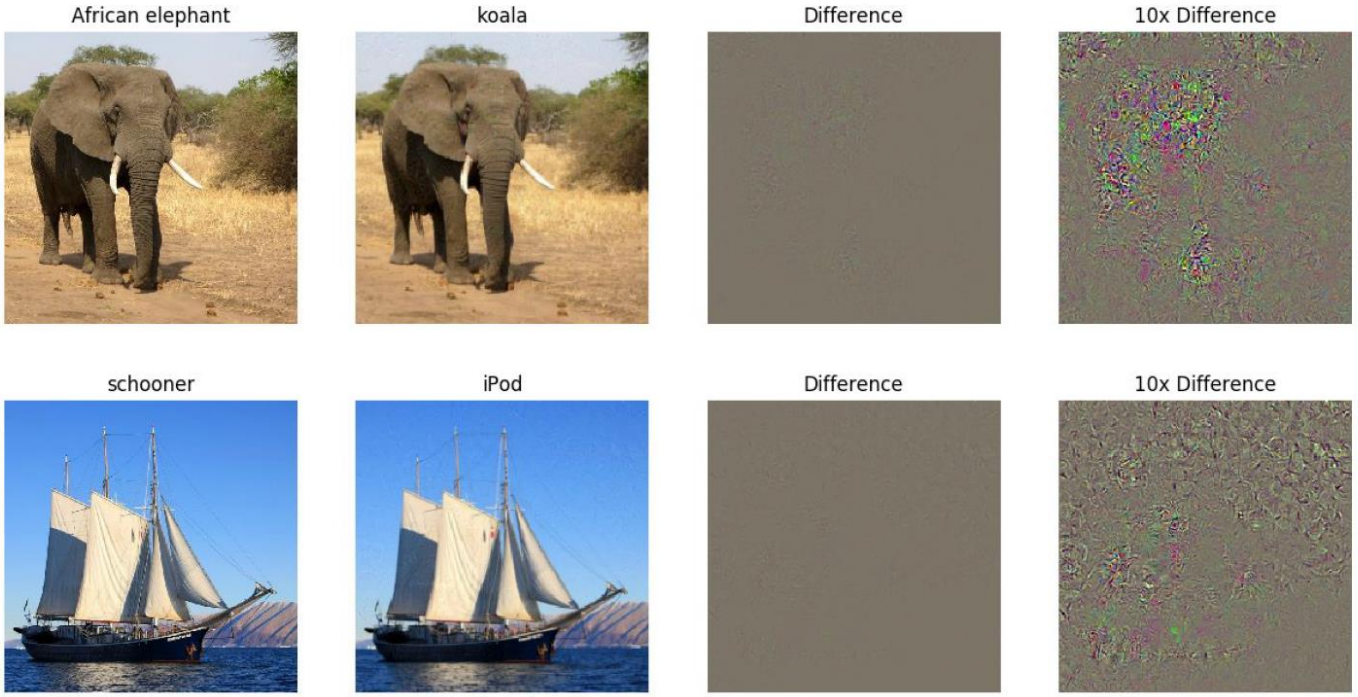
Prevara neuronske mreže

Fooling Images / Adversarial Examples

(1) Start from an arbitrary image
(2) Pick an arbitrary class
(3) Modify the image to maximize the class
(4) Repeat until network is fooled

Prevara neuronske mreže

Fooling Images / Adversarial Examples

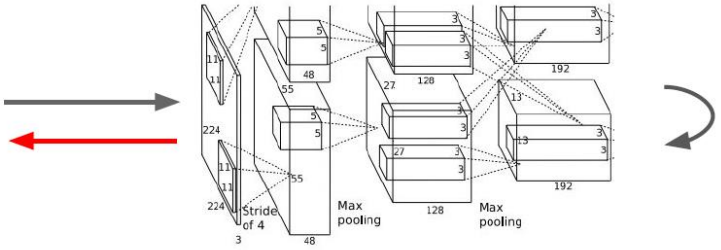


Boat image is CC0 public domain
Elephant image is CC0 public domain

DeepDream

DeepDream: Amplify existing features

Rather than synthesizing an image to maximize a specific neuron, instead try to **amplify** the neuron activations at some layer in the network



Choose an image and a layer in a CNN; repeat:

1. Forward: compute activations at chosen layer
2. Set gradient of chosen layer *equal to its activation*
3. Backward: Compute gradient on image
4. Update image

Equivalent to:

$$I^* = \arg\max_I \sum_i f_i(I)^2$$

Mordvintsev, Olah, and Tyka, "Inceptionism: Going Deeper into Neural Networks", Google Research Blog. Images are licensed under CC-BY 4.0

DeepDream

DeepDream: Amplify existing features

https://github.com/google/deepdream

Code is very simple but it uses a couple tricks:

(Code is licensed under Apache 2.0)

```
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)
```

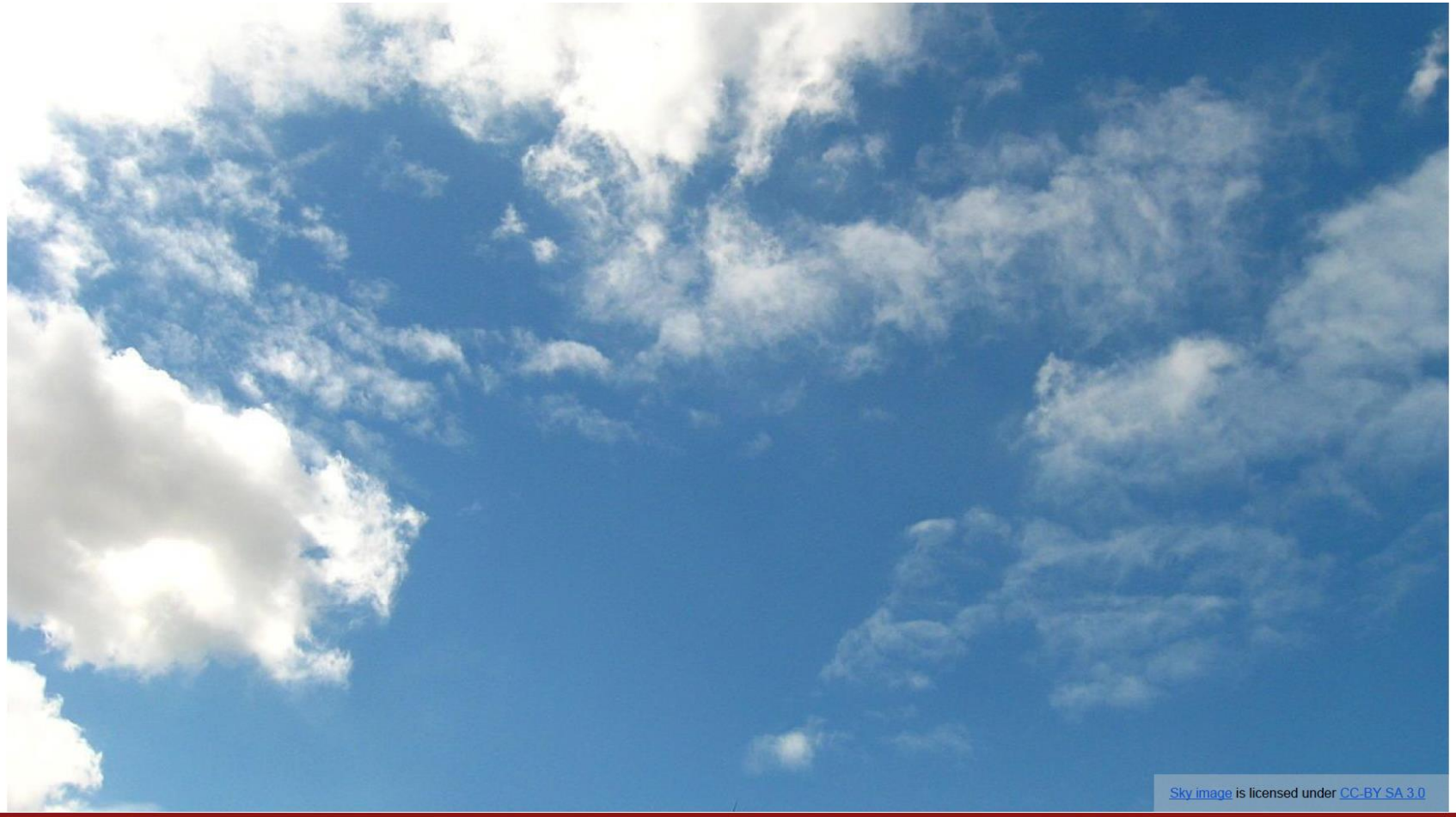
Jitter image

L1 Normalize gradients

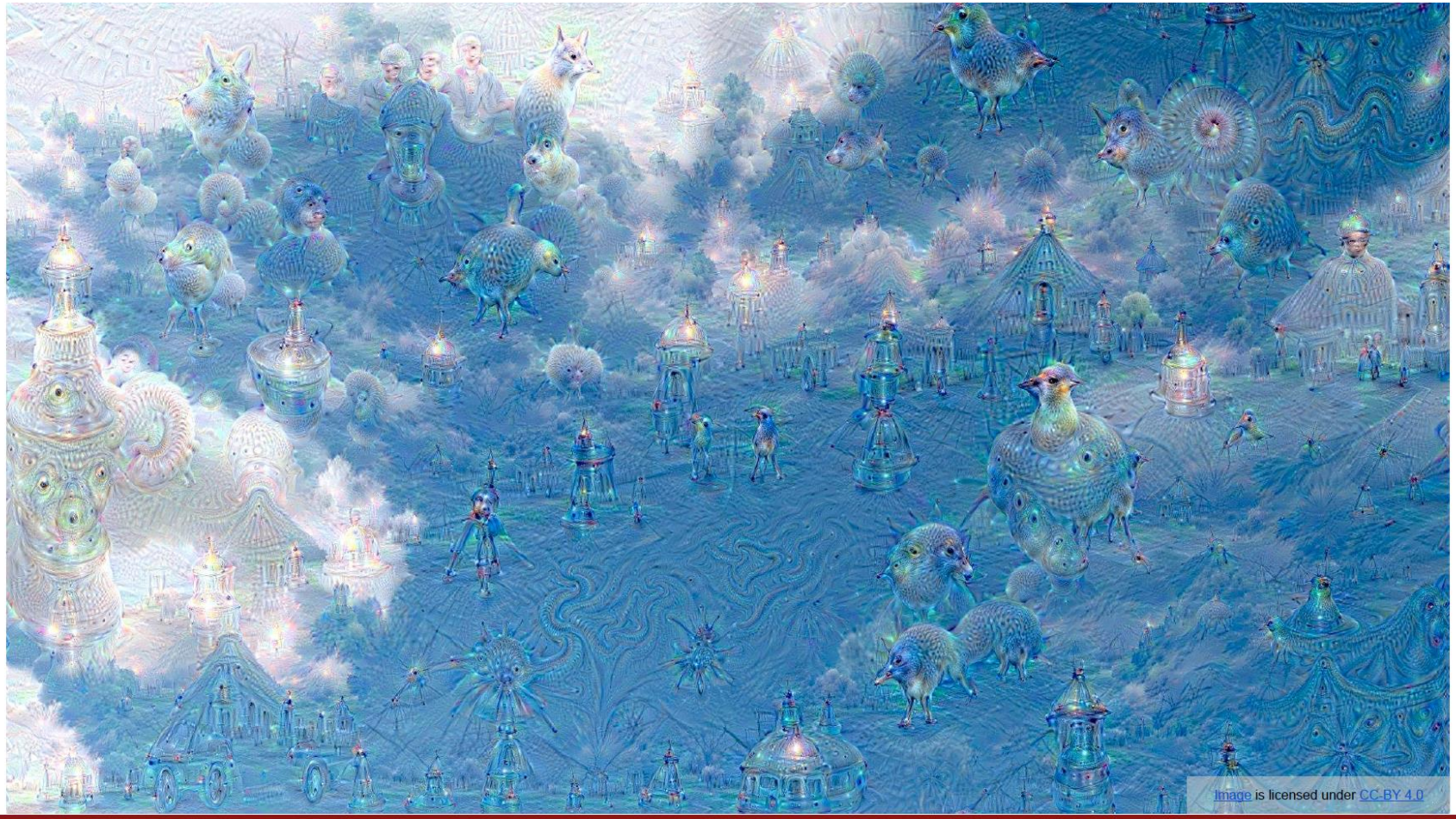
Clip pixel values

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

DeepDream

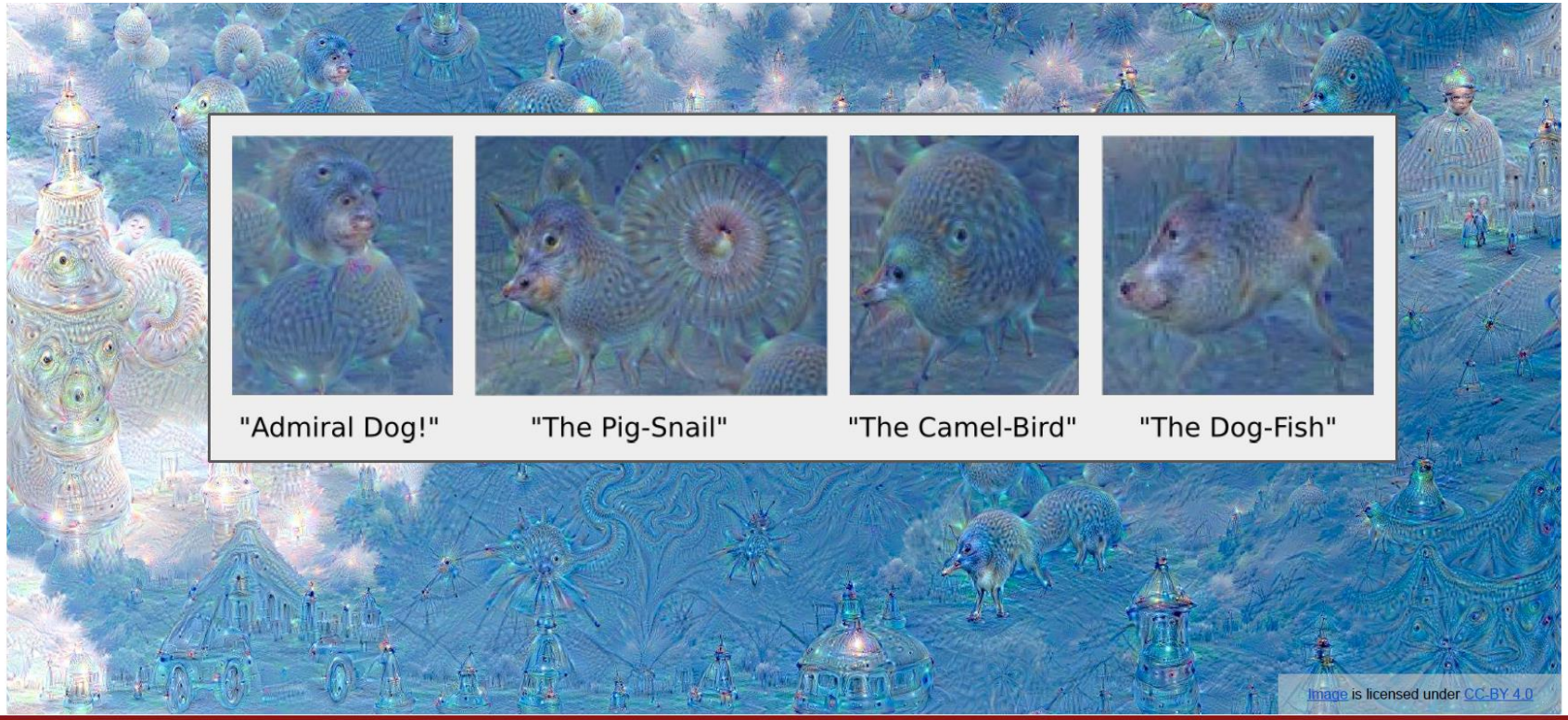


DeepDream

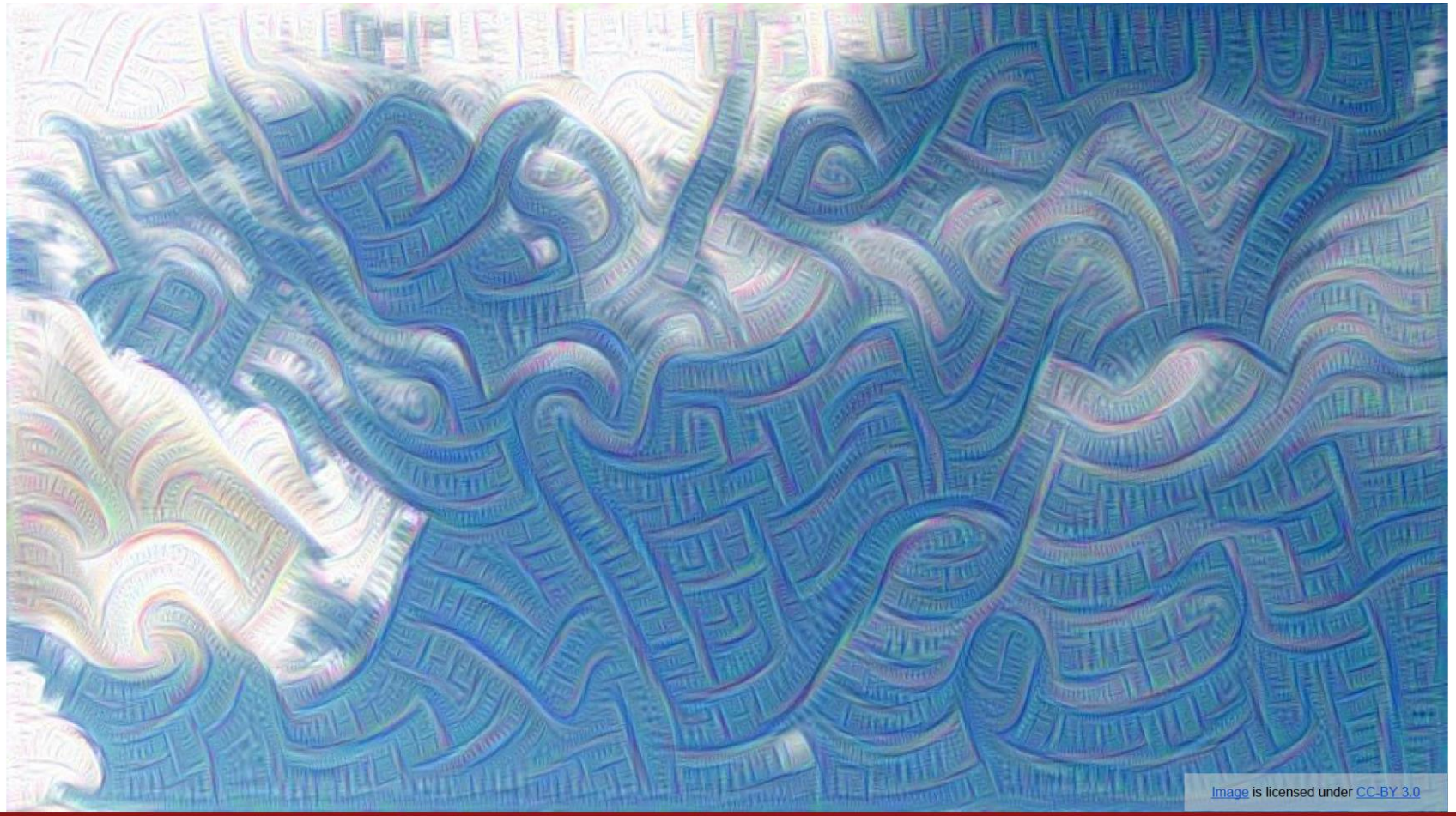


DeepDream

- 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

Feature inversion

- Još jedan postupak koji služi da nam pruži uvid koji tipovi elemanata 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

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)

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

Given feature vector

Features of new image

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

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

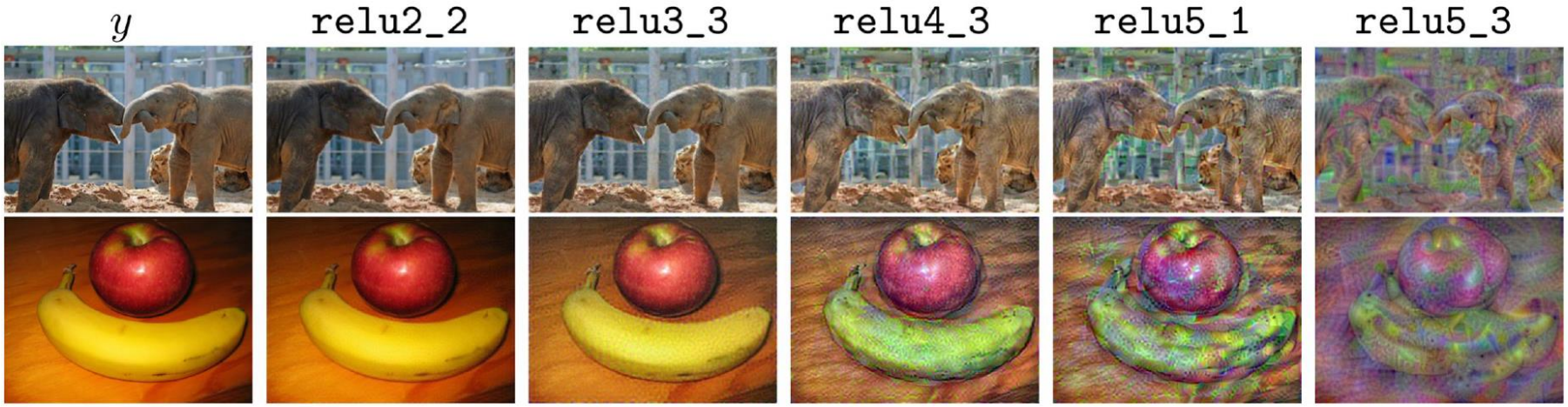
Total Variation regularizer
(encourages spatial smoothness)

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

Feature inversion

Feature Inversion

Reconstructing from different layers of VGG-16



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
  - Na ovom nivou 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, mnogi fini detalji (niskog nivoa) su izgubljeni
    - Nisu ono što su originalni pikseli tačno bili u pogledu boje, teksture,...
  - Ovo nam daje osećaj da, 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