# Računalni vid

## Konvolucijski modeli za klasifikaciju slike

od piksela do semantike

Siniša Šegvić

# Sadržaj

# Arhitekture: problem

Klasifikacija slike: temeljni problem raspoznavanja

Problem je težak jer ne znamo gdje je objekt koji definira razred

Posebno je težak kad imamo veliku raznolikost unutar razreda, a pojedini objekti različitih razreda su slični:

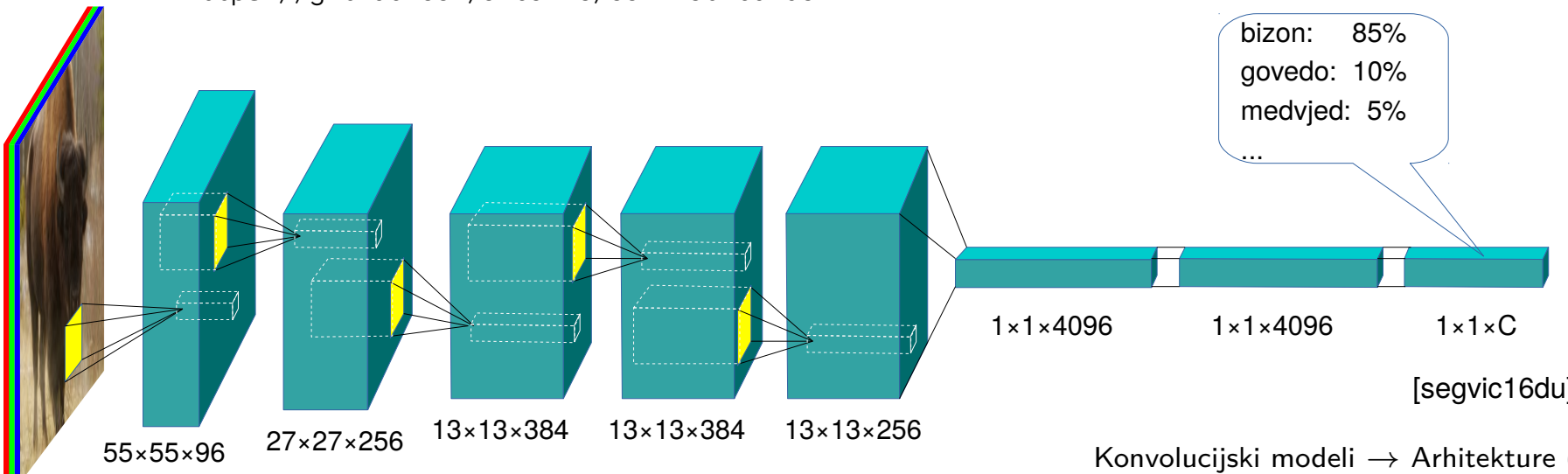
[image-net.org]

# Arhitekture: klasifikacija

Prvi duboki konvolucijski model za klasifikaciju na velikom podatkovnom skupu [krizhevsky12nips]

- **ulaz**: slika; **izlaz**: distribucija preko 1000 razreda
- **funkcija cilja**: prosječna log-izglednost točnog razreda
- **struktura**: slijed konvolucija i sažimanja
  - postepeno smanjenje rezolucije te povećanje semantičke dubine
- moderne arhitekture: O($10^2$) slojeva, O($10^6$) parametara!
  - za sliku 224x224: O($10^9$) množenja, O($10^8$) bajtova

https://github.com/albanie/convnet-burden

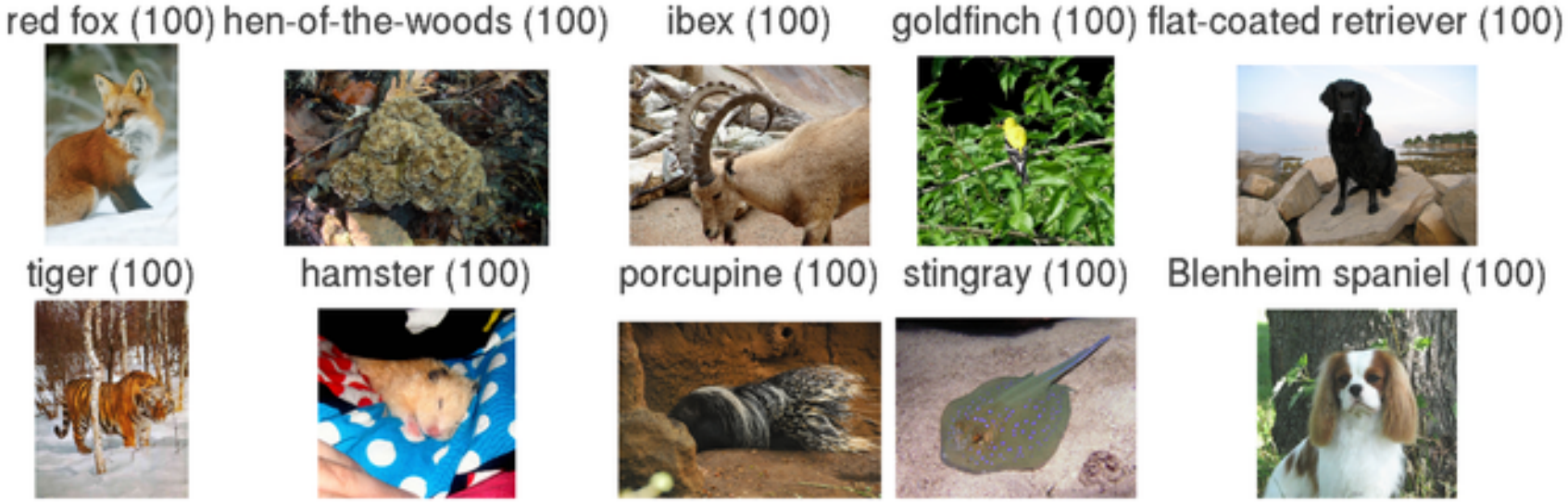



[segvic16du]

# Arhitekture: ImageNet

Jedan od najpopularnijih podatkovnih skupova [russakovsky15ijcv]

- godišnje natjecanje: klasifikacija, lokalizacija, detekcija u videu
- razmotrit ćemo klasifikacijsko natjecanje: $10^6$ slika, $10^3$ razreda
- životinje, objekti, materijali, sportovi, jela...
- evaluacijska metrika: top-5 pogreška predikcije (ljudi: 5%)




[image-net.org]

# Arhitekture: ImageNet teški primjeri

Razredi teški za **ljude** [russakovsky15ijcv]:

- vrste životinja (npr. 120 pasmina pasa!)
- egzotični razredi (kolotura, reflektor, majski stup)

Razredi teški za **GoogleNet** (2014, 6.7%):

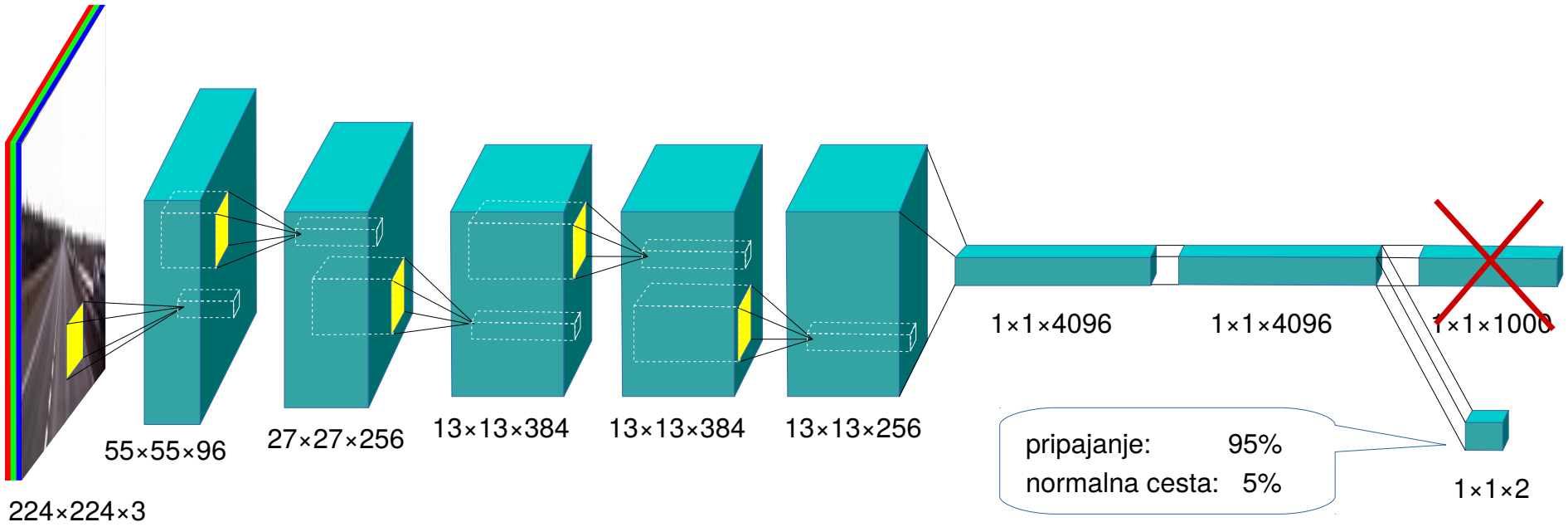
- maleni i tanki objekti, filtrirane i čudne slike
- apstrakcije (sjekira-igračka, slike s tekstom)
- razredi s velikom unutar-razrednom varijancom, slični razredi


[image-net.org]

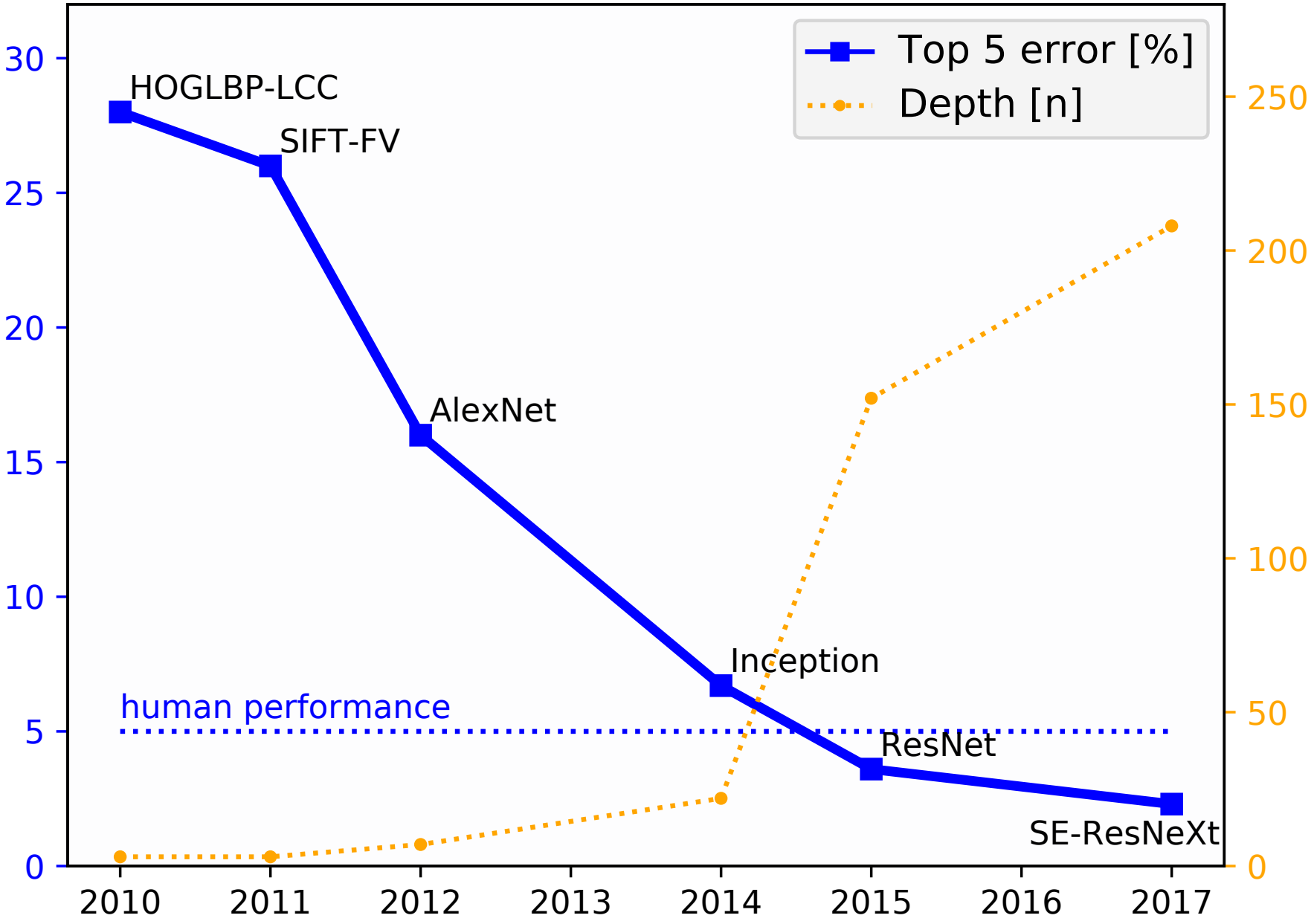
# Arhitekture: prijenos znanja

Naučeni model možemo ugoditi (eng. fine tune) za novi (lakši) zadatak

- odrezati posljednjih nekoliko slojeva
- spojiti preostale slojeve s prednjim krajem za novi zadatak
- trenirati (ugoditi) dobiveni model za novi zadatak
- naslijeđeni slojevi već su naučeni pa sada možemo učiti s manje podataka (nekoliko tisuća slika)



# Arhitekture: napredak



# Arhitekture: napredak

| naziv | godina | dubina | param | GFLOP | top-5 |
|---|---|---|---|---|---|
| Le Net 5 | 1998 | 6 | $60 \cdot 10^3$ | - | - |
| SIFT-FV | 2011 | 3 | $500 \cdot 10^3$ | - | 26.0% |
| AlexNet | 2012 | 7 | $60 \cdot 10^6$ | 0.7 | 15.0% |
| VGG-E | 2014 | 19 | $144 \cdot 10^6$ | 20 | 8% |
| Inception v1 | 2014 | $22 \times 4$ | $6 \cdot 10^6$ | 2 | 10.1% |
| ResNet | 2015 | 152 | $60 \cdot 10^6$ | 11 | 5.5% |
| DenseNet | 2016 | 161 | $30 \cdot 10^6$ | 8 | 6.2% |
| MobileNetv2 | 2018 | 20 | $3.4 \cdot 10^6$ | .3 | 9.5% |
| EfficientNet-B7 | 2019 | 264 | $66 \cdot 10^6$ | $37 \cdot 10^3$ | 2.9% |
| ConvNext-XL | 2022 | 110 | $350 \cdot 10^6$ | 60.9 | 1.8% |

Članci: AlexNet, VGG, GoogleNet, ResNet, DenseNet, EfficientNet.

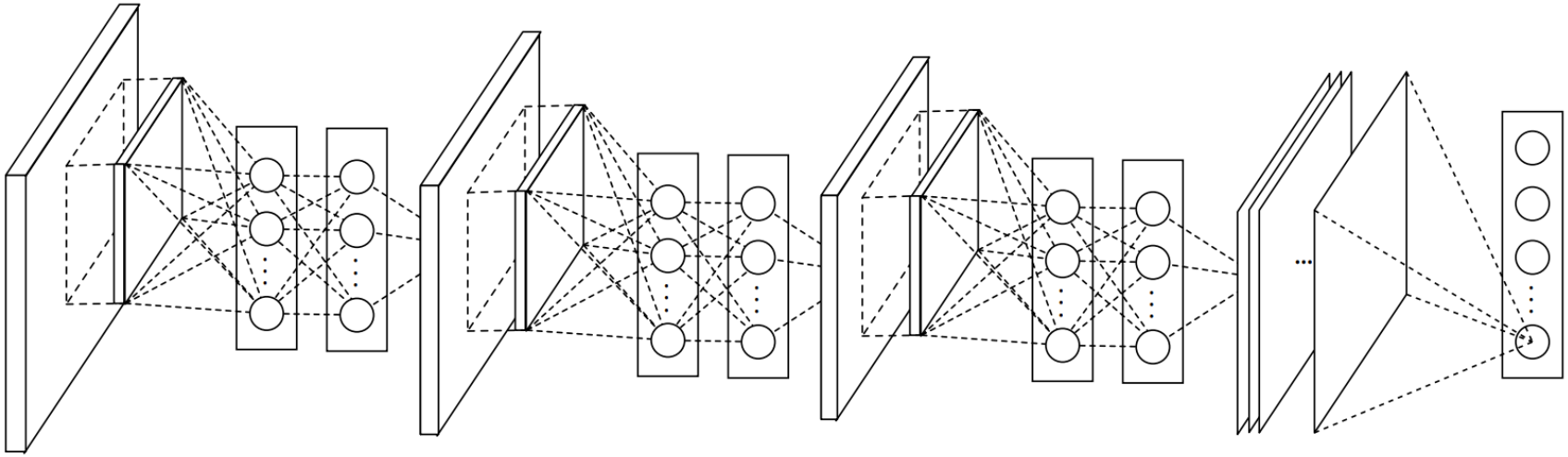
# Arhitekture: VGG

| ConvNet Configuration | | | | | |
|---|---|---|---|---|---|
| A | A-LRN | B | C | D | E |
| 11 weight layers | 11 weight layers | 13 weight layers | 16 weight layers | 16 weight layers | 19 weight layers |
| input (224 × 224 RGB image) | | | | | |
| conv3-64 | conv3-64<br>**LRN** | conv3-64<br>**conv3-64** | conv3-64<br>conv3-64 | conv3-64<br>conv3-64 | conv3-64<br>conv3-64 |
| maxpool | | | | | |
| conv3-128 | conv3-128 | conv3-128<br>**conv3-128** | conv3-128<br>conv3-128 | conv3-128<br>conv3-128 | conv3-128<br>conv3-128 |
| maxpool | | | | | |
| conv3-256<br>conv3-256 | conv3-256<br>conv3-256 | conv3-256<br>conv3-256 | conv3-256<br>conv3-256<br>**conv1-256** | conv3-256<br>conv3-256<br>**conv3-256** | conv3-256<br>conv3-256<br>conv3-256<br>**conv3-256** |
| maxpool | | | | | |
| conv3-512<br>conv3-512 | conv3-512<br>conv3-512 | conv3-512<br>conv3-512 | conv3-512<br>conv3-512<br>**conv1-512** | conv3-512<br>conv3-512<br>**conv3-512** | conv3-512<br>conv3-512<br>conv3-512<br>**conv3-512** |
| maxpool | | | | | |
| conv3-512<br>conv3-512 | conv3-512<br>conv3-512 | conv3-512<br>conv3-512 | conv3-512<br>conv3-512<br>**conv1-512** | conv3-512<br>conv3-512<br>**conv3-512** | conv3-512<br>conv3-512<br>conv3-512<br>**conv3-512** |
| maxpool | | | | | |
| FC-4096 | | | | | |
| FC-4096 | | | | | |
| FC-1000 | | | | | |
| soft-max | | | | | |

[simonyan15iclr]

# Arhitekture: network in network

Idea 1: convolutional application of a small fully connected model

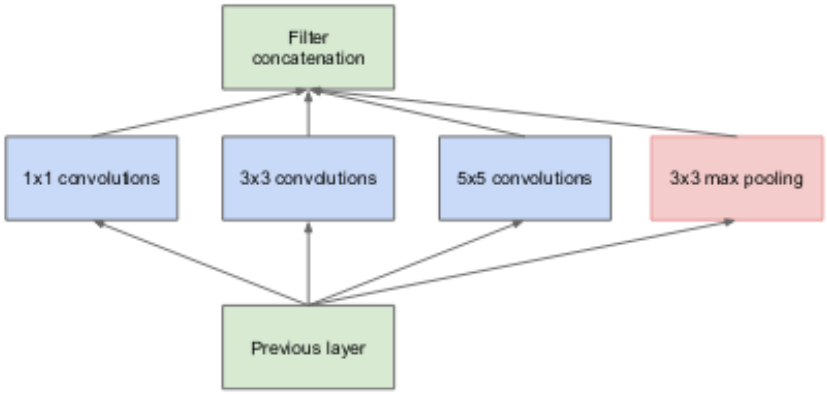


[lin14iclr]

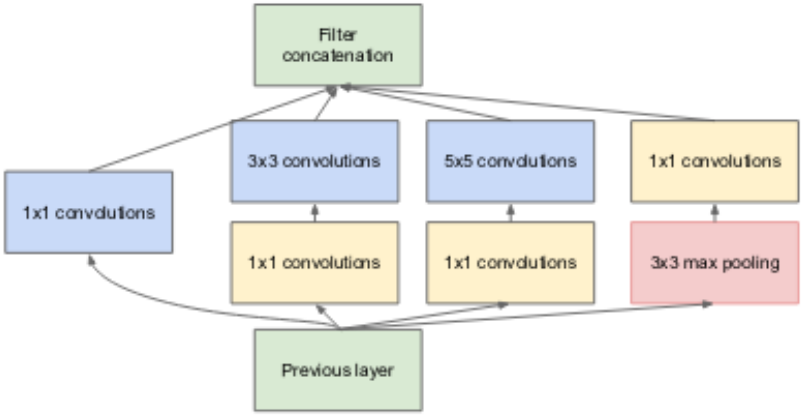
Idea 2: replace flatten with global average pool

- use only one fully-connected layer
- less parameters, better generalization

# Arhitekture: inception




(a) Inception module, naïve version




(b) Inception module with dimension reductions

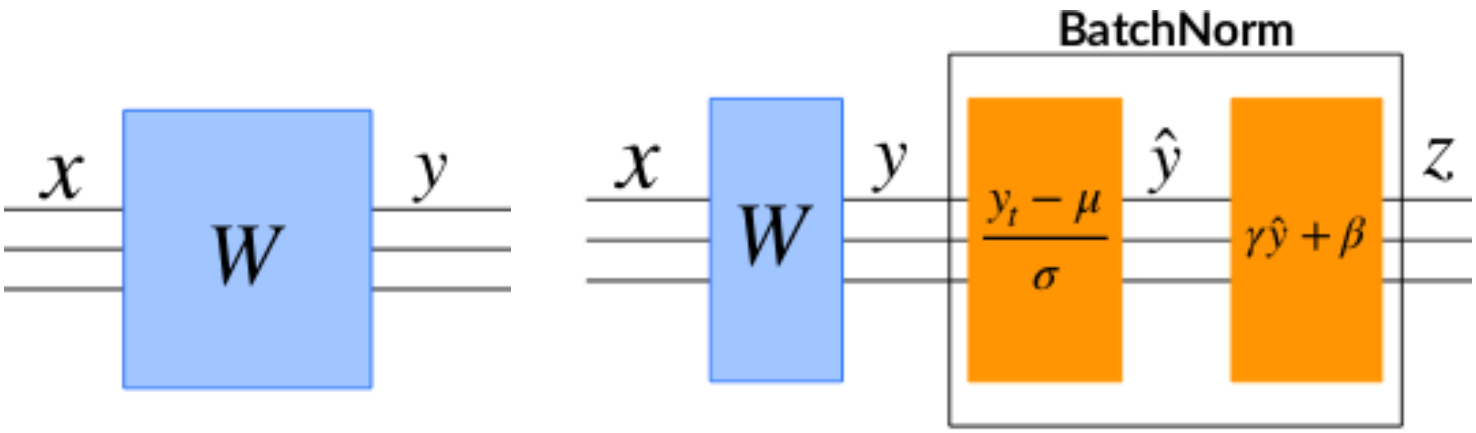
[szegedy15cvpr]

| Number of models | Number of Crops | Cost | Top-5 error | compared to base |
|---|---|---|---|---|
| 1 | 1 | 1 | 10.07% | base |
| 1 | 10 | 10 | 9.15% | -0.92% |
| 1 | 144 | 144 | 7.89% | -2.18% |
| 7 | 1 | 7 | 8.09% | -1.98% |
| 7 | 10 | 70 | 7.62% | -2.45% |
| 7 | 144 | 1008 | 6.67% | -3.45% |

[szegedy15cvpr]

# Tehnike: BatchNorm

Important idea: normalize activations produced by each layer




[santurkar18neurips]

During training:

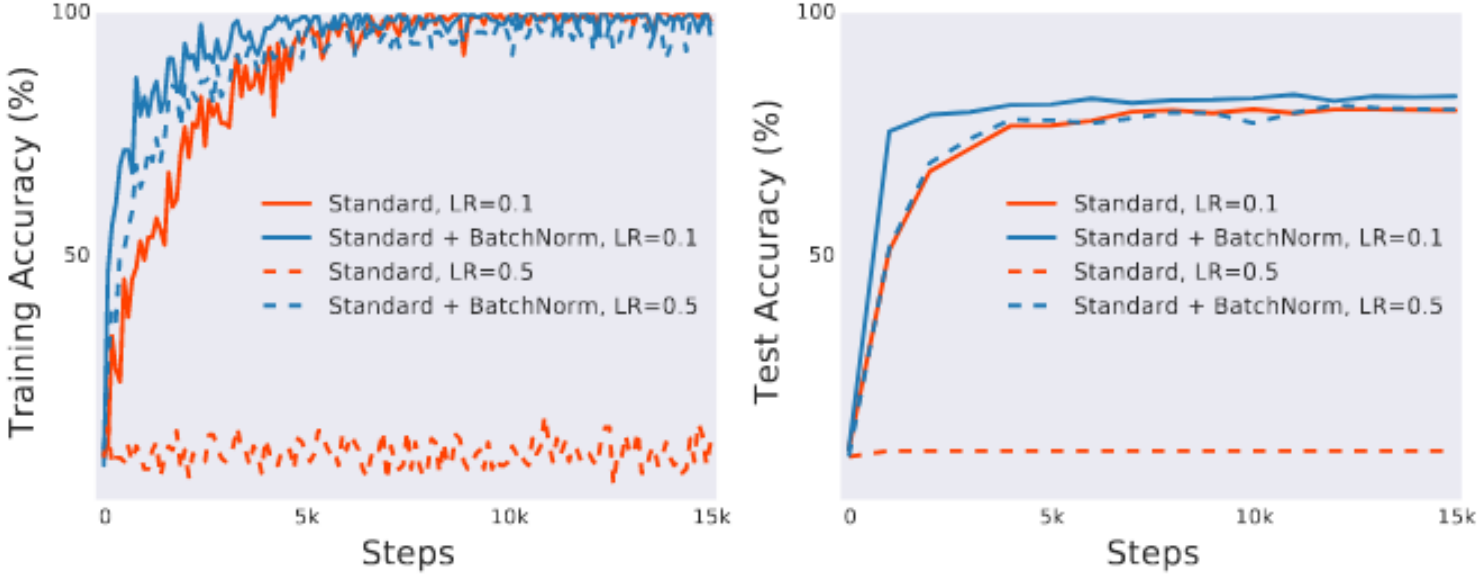
- the activations in the batch are $\mathcal{N}(0, 1)$
  - ideal conditions for ReLU (no vanishing gradients!)
- all activations are equally important
  - stable learning, tolerance for high learning rates
- each particular image is jittered in an epoch-specific manner
  - this makes it hard for the model to overfit (=regularization)

# Tehnike: BatchNorm (2)

During inference:

- the activations are normalized according to population statistics
- often batchnorm can be fused with the preceding convolution in order to optimize the inference time
- sometimes we finetune the model with freezed population statistics

Advantages: faster learning, improved generalization

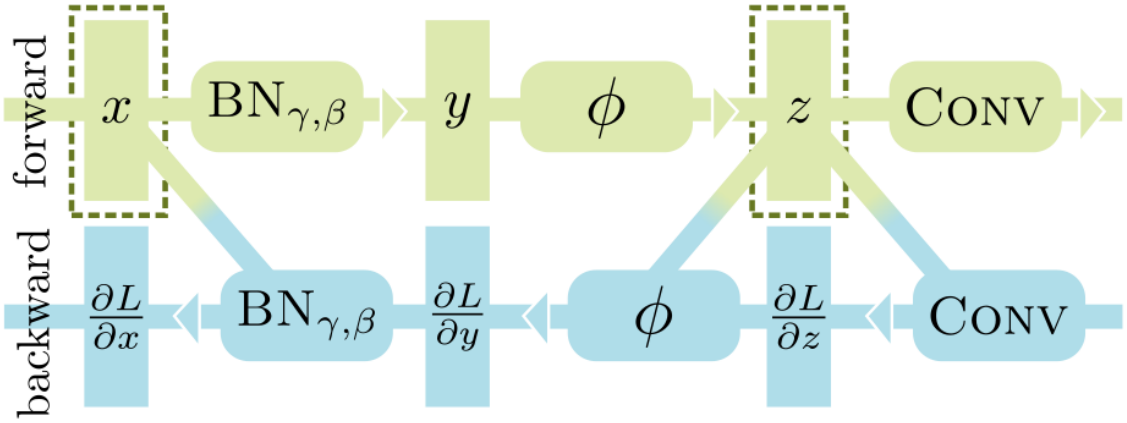



[santurkar18nips]

# Tehnike: BatchNorm (3)

Downsides:

- stable training requires large batches
- increased training footprint unless custom backprop is used
- problematic with non-standard training and multiple datasets




(a) Standard building block (memory-inefficient)

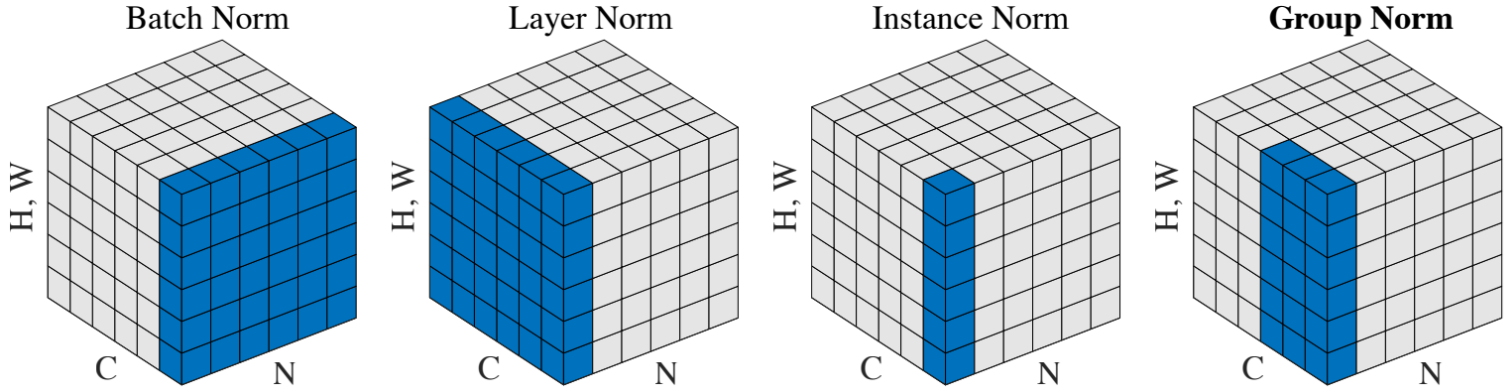
[bulo18cvpr]

Batchnorm replacements:

- weight standardization, gradient clipping [brock21icml].
- layernorm, groupnorm

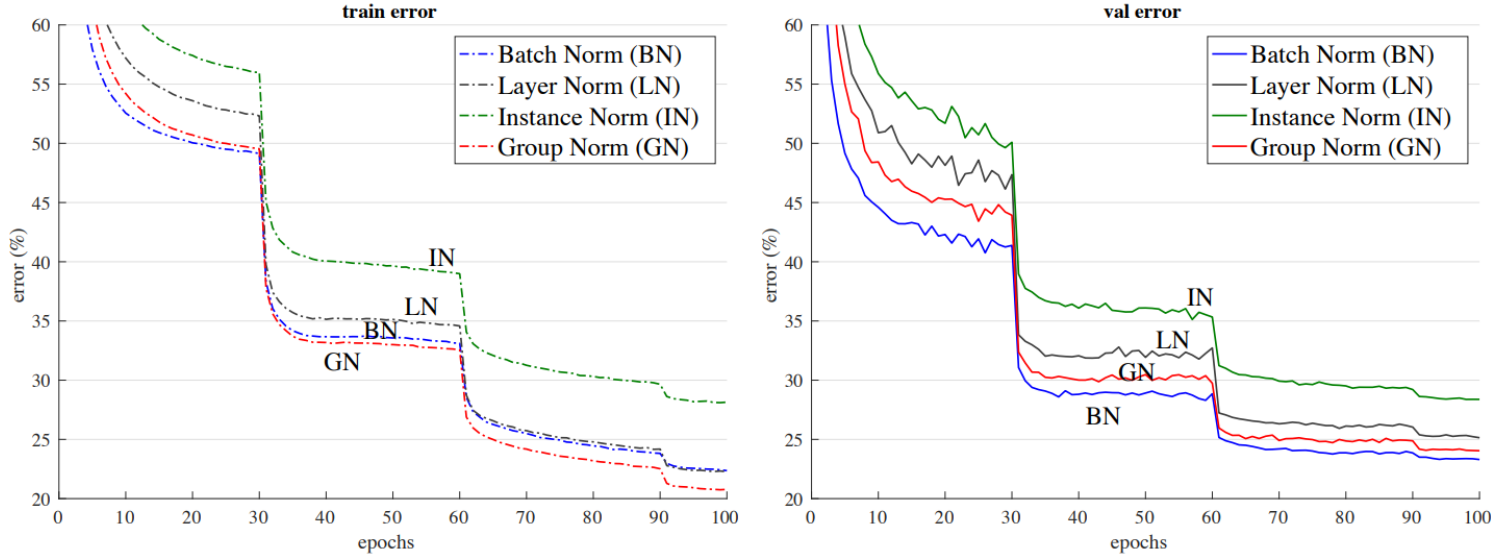
# Tehnike: Other normalizations

Idea: normalize over pixels instead of over examples




[wu18eccv]

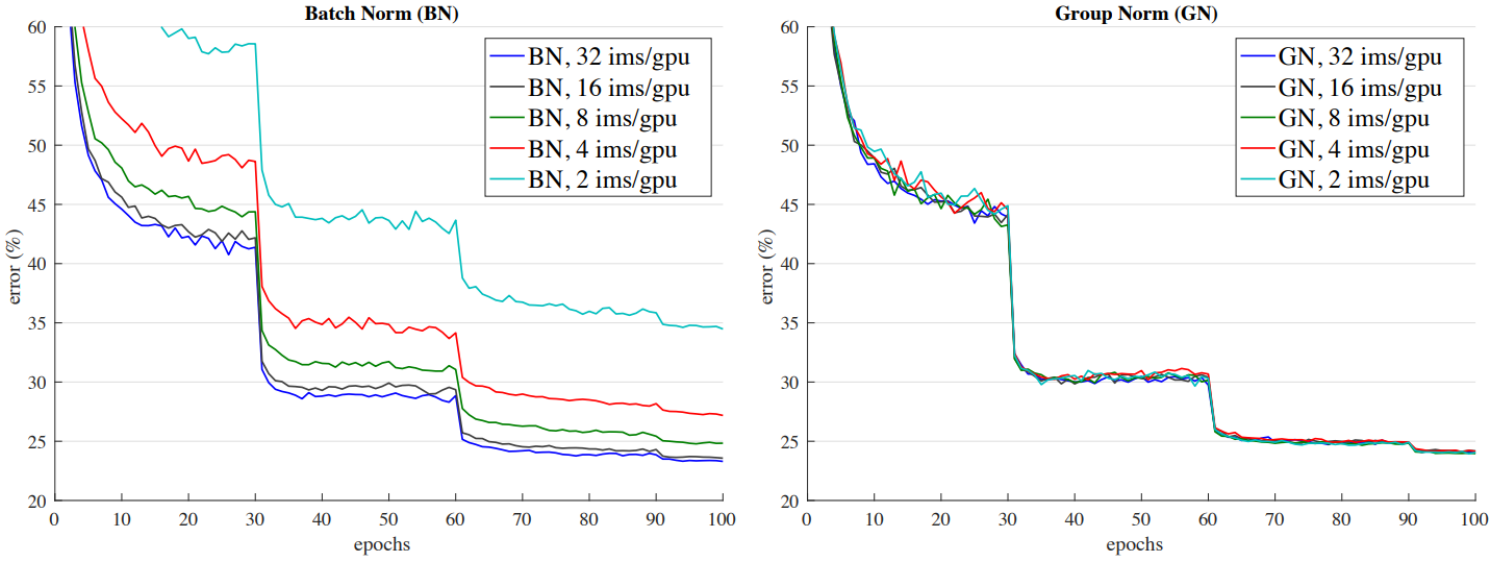
Group-norm outperforms layernorm, but batchnorm generalizes better:




[wu18eccv]

# Tehnike: Other normalizations (2)

Batchnorm requires large batches:




[wu18eccv]

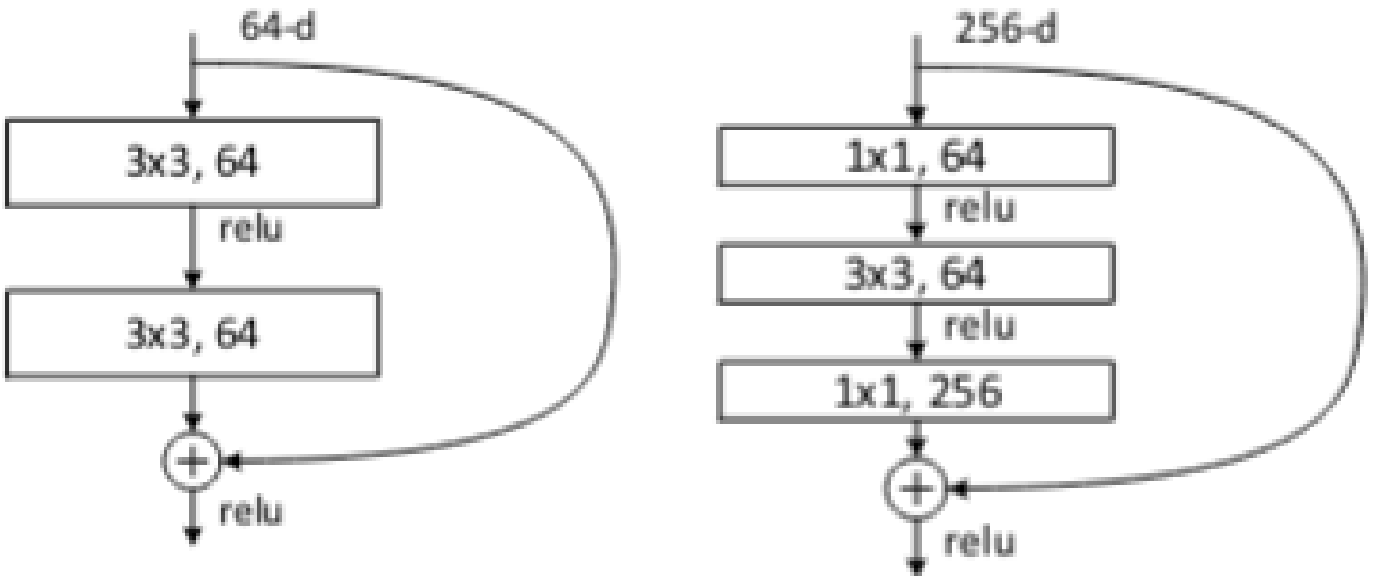
Actnorm [kingma18neurips] - poor man-s batchnorm

- initialize population statistics on the first mini-batch
- afterwards, treat them as regular parameters!
- can learn with batch size 1

# Tehnike: ResNet - ideja

Duboki slojevi aditivno popravljaju približnu predikciju prethodnika

- sloj modelira aditivnu korekciju (=rezidual)



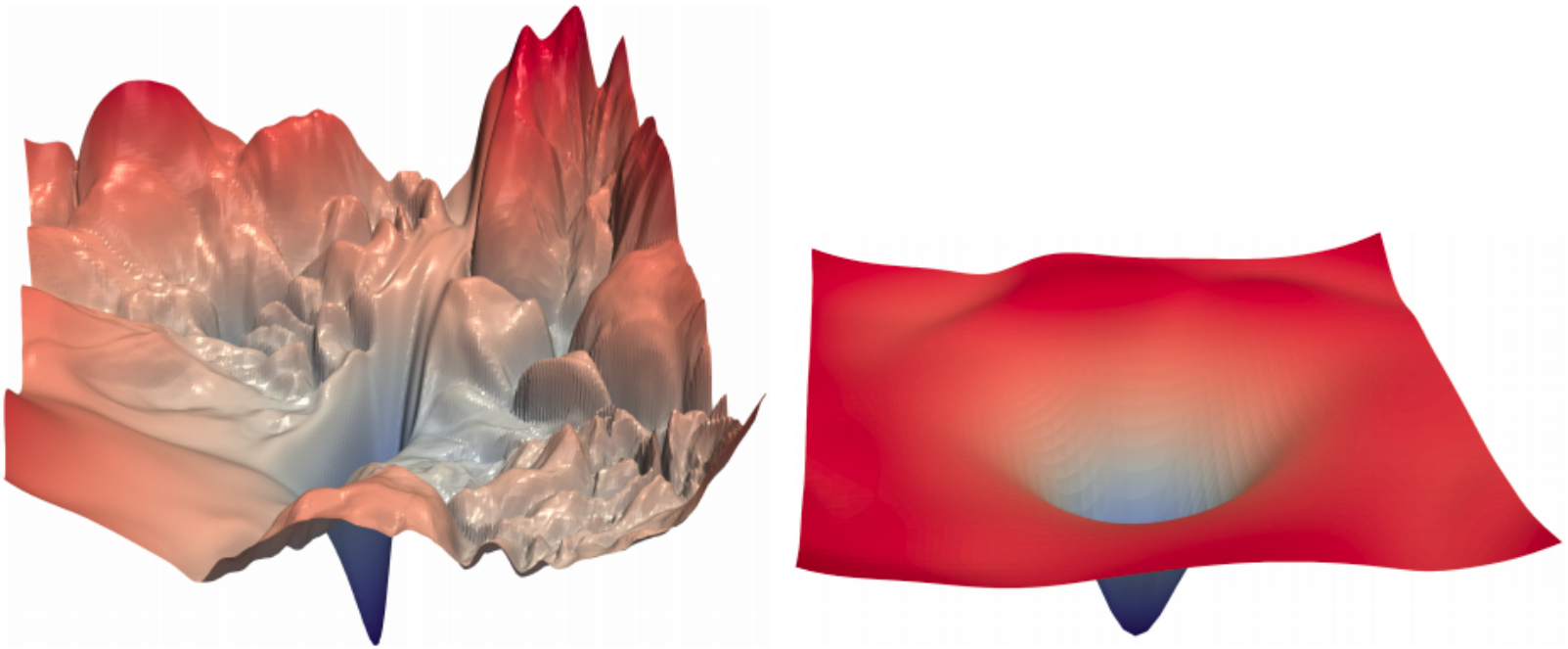

[he15cvpr]

Rezidualna jedinica (engl. residual unit [he16eccv])

- temeljna građevna jedinica većih modela
- za male slike: 2 $\times$ conv3x3
- za velike slike: c1x1, c3x3, c1x1 (usko grlo $\rightarrow$ manje FLOPS)

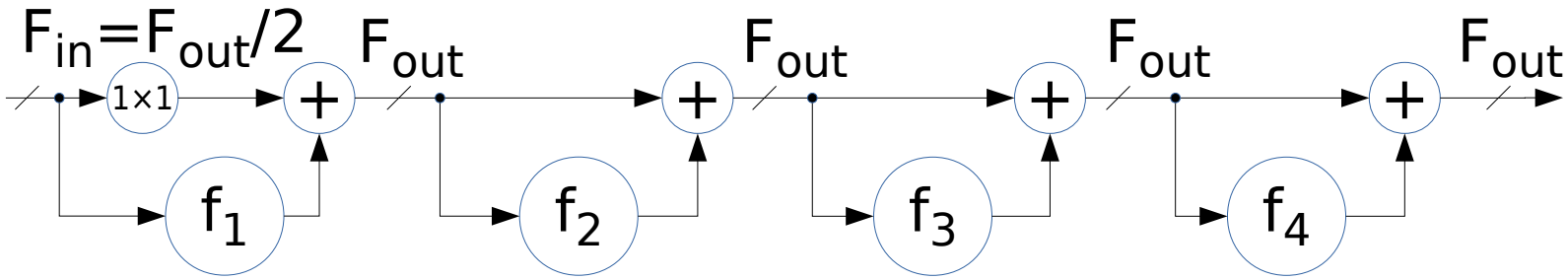
# Tehnike: ResNet - učinak

Rezidualne veze glade gubitak i poboljšavaju tok gradijenta



(a) without skip connections (b) with skip connections

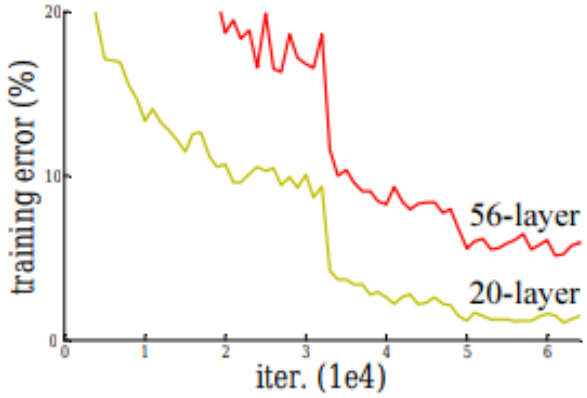
[li18nips]

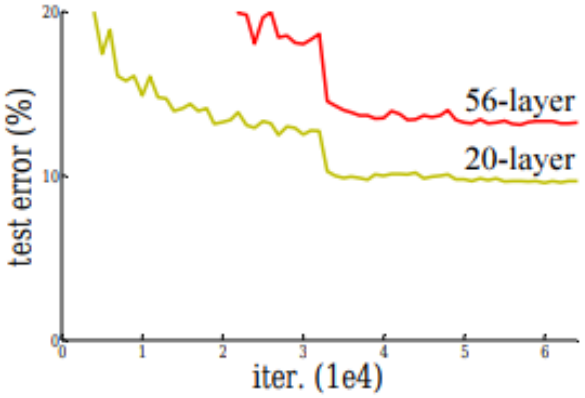


# Tehnike: ResNet - dubina

Slijedne modele možemo učiti do dubine cca 20

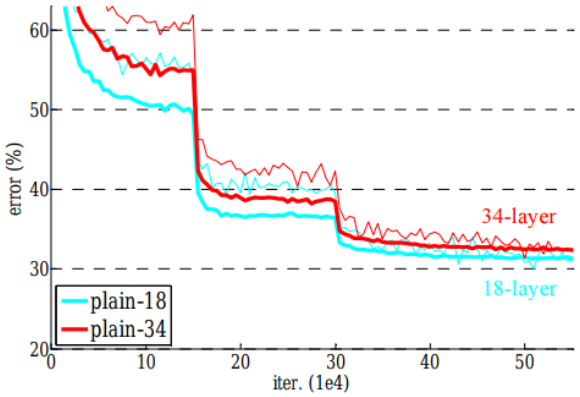
- zatim, *empirijska* greška raste (C10):

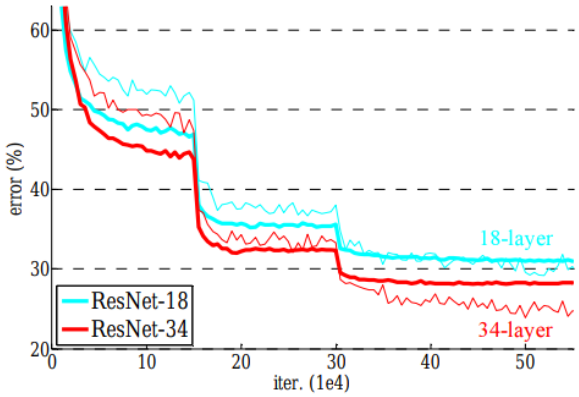



[he15cvpr]

Rezidualni modeli nemaju problema s dubinom:




[he15cvpr]


[he15cvpr]

(tanko=train, debelo=test)

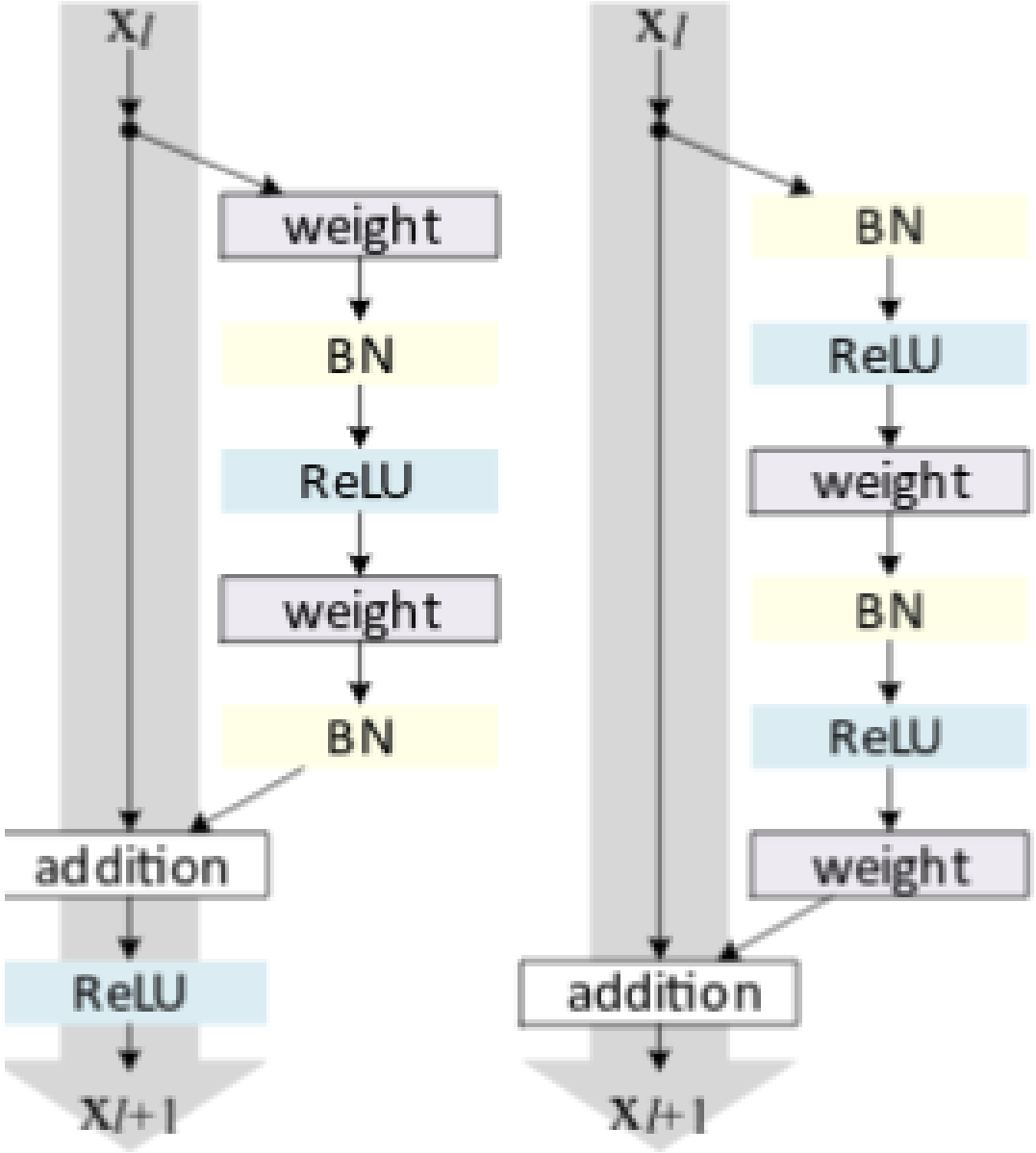
# Tehnike: ResNet - struktura

Rane rezidualne jedinice: conv-BN-ReLU [he15cvpr]

- gradijent može zapeti na ReLU

Poboljšana izvedba: BN-ReLU-conv [he16eccv]

- gradijent slobodno teče od izlaza prema ulazu
- koristi se za modele dublje od 100 slojeva (npr. RN-152)



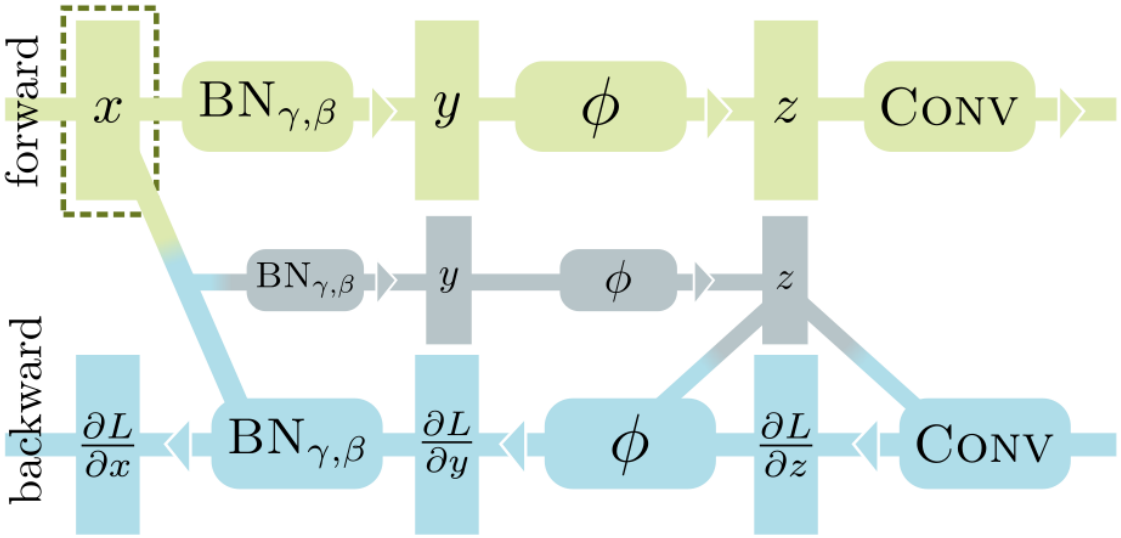

[he16eccv]

# Tehnike: ResNet - modeli

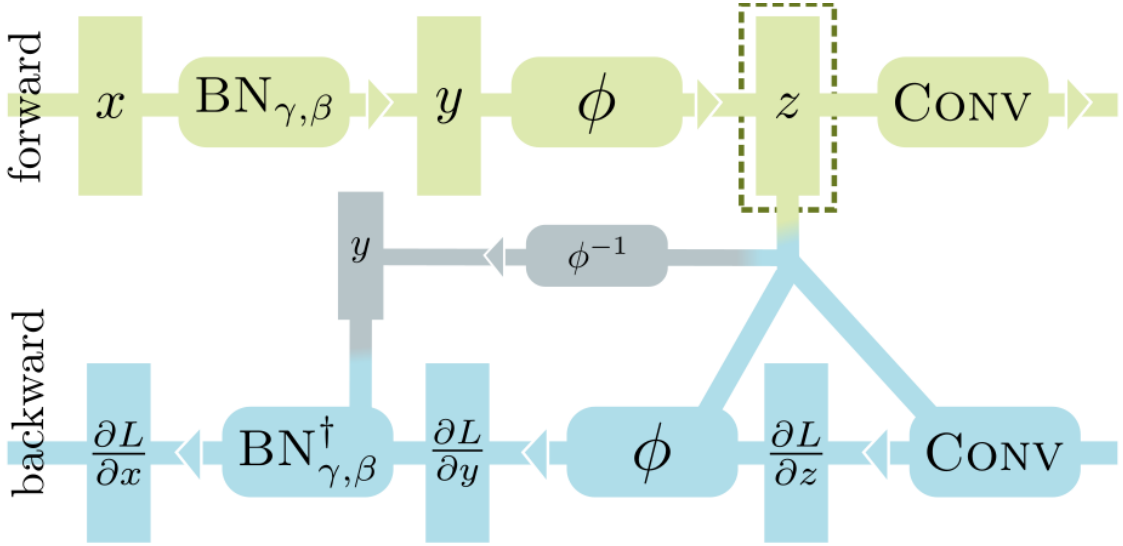
| layer name | output size | 18-layer | 34-layer | 50-layer | 101-layer | 152-layer |
|---|---|---|---|---|---|---|
| conv1 | 112×112 | 7×7, 64, stride 2 | | | | |
| conv2_x | 56×56 | 3×3 max pool, stride 2 | | | | |
| | | $\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$ | $\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$ | $\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$ | $\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$ | $\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$ |
| conv3_x | 28×28 | $\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$ | $\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$ | $\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$ | $\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$ | $\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$ |
| conv4_x | 14×14 | $\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$ | $\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$ | $\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$ | $\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$ | $\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$ |
| conv5_x | 7×7 | $\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$ | $\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$ | $\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$ | $\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$ | $\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$ |
| | 1×1 | average pool, 1000-d fc, softmax | | | | |
| FLOPs | | $1.8\times10^9$ | $3.6\times10^9$ | $3.8\times10^9$ | $7.6\times10^9$ | $11.3\times10^9$ |

[he15iclr]

# Tehnike: ResNet - BatchNorm




(b) Checkpointing [4, 19]

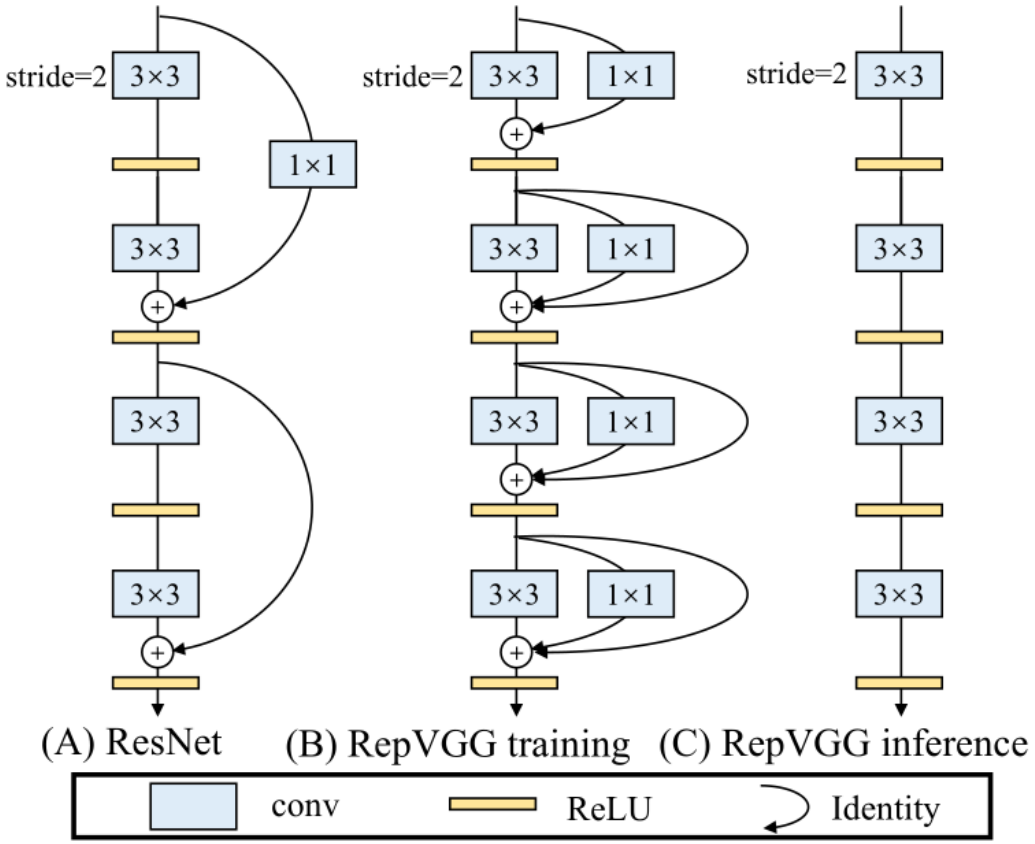



(e) In-Place Activated Batch Normalization II (proposed method)

[bulo18cvpr]

# Tehnike: ResNet - RepVGG

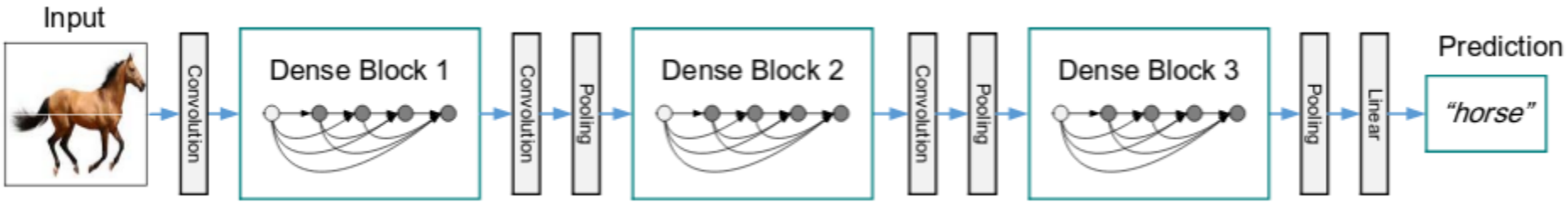
Rezidualne veze mogu se simulirati strukturnom reparametrizacijom slijednog modela:



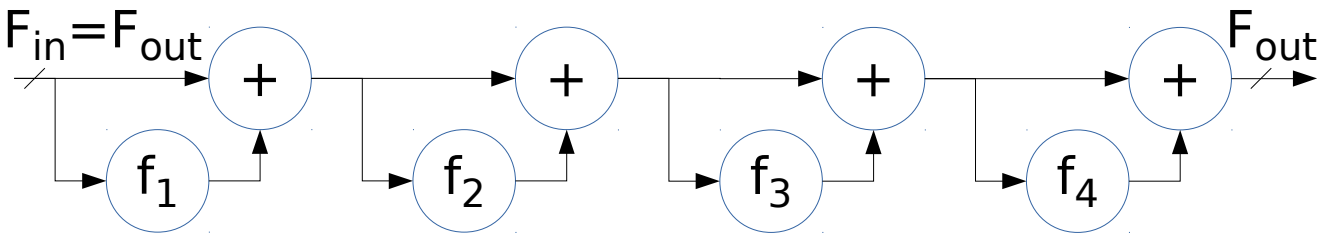

[ding21cvpr]

Tijekom zaključivanja, tri paralelne grane izvode se jednom 3x3 konvolucijom.

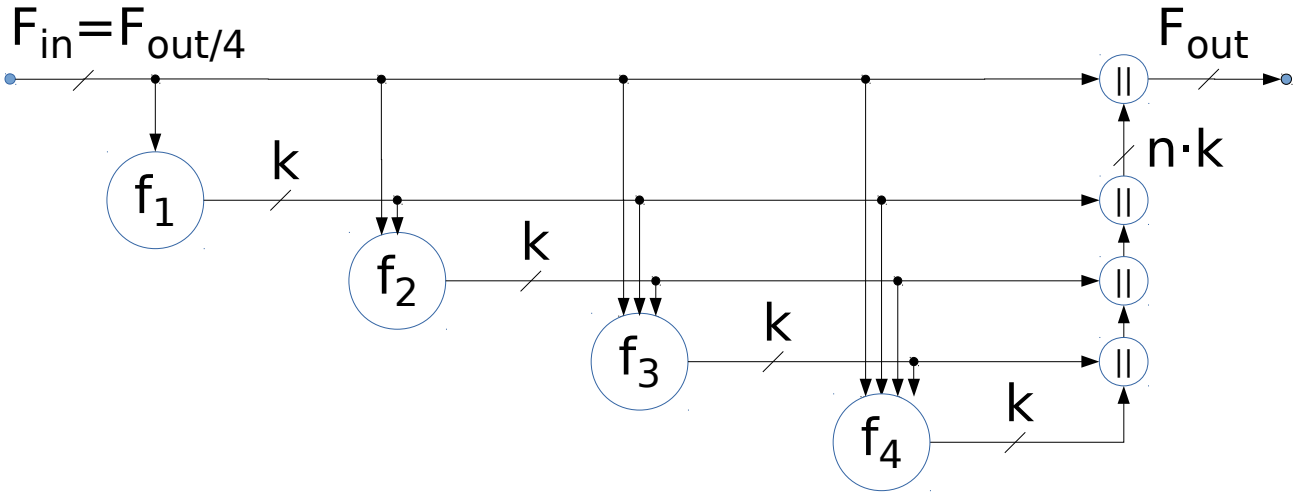
# Tehnike: DenseNet - ideja




[huang17cvpr]

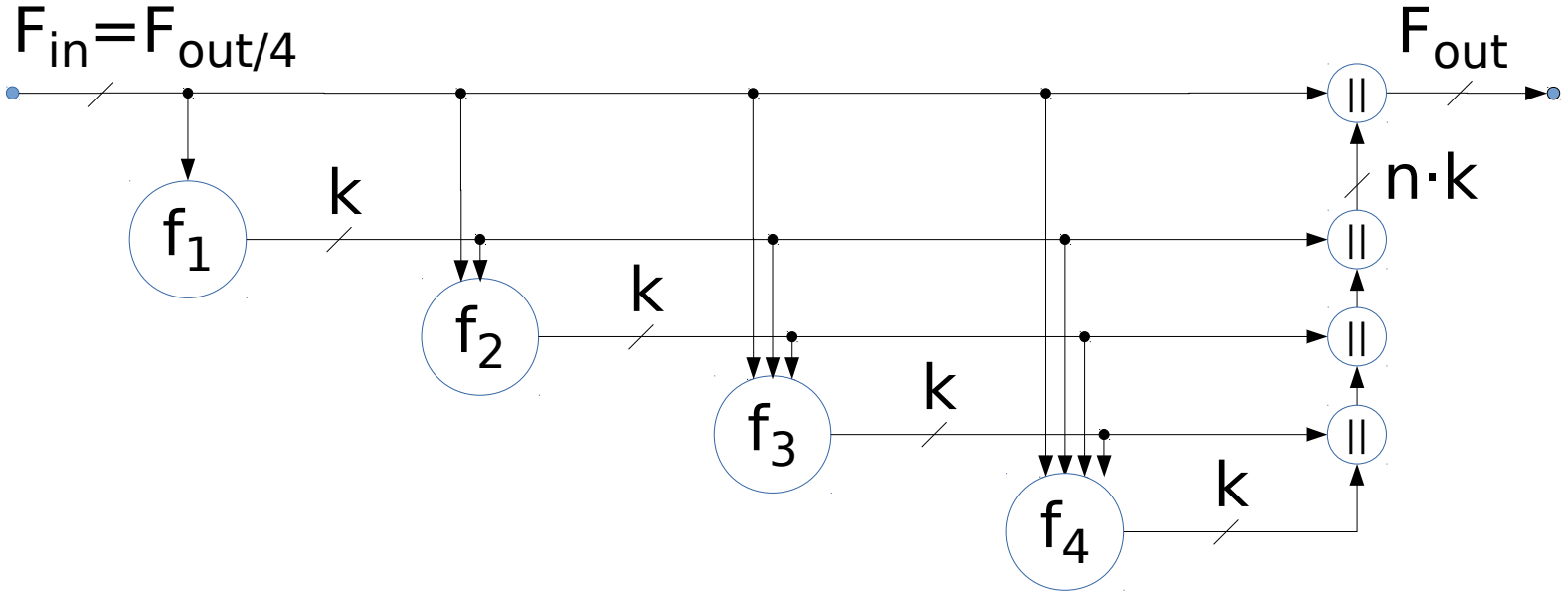


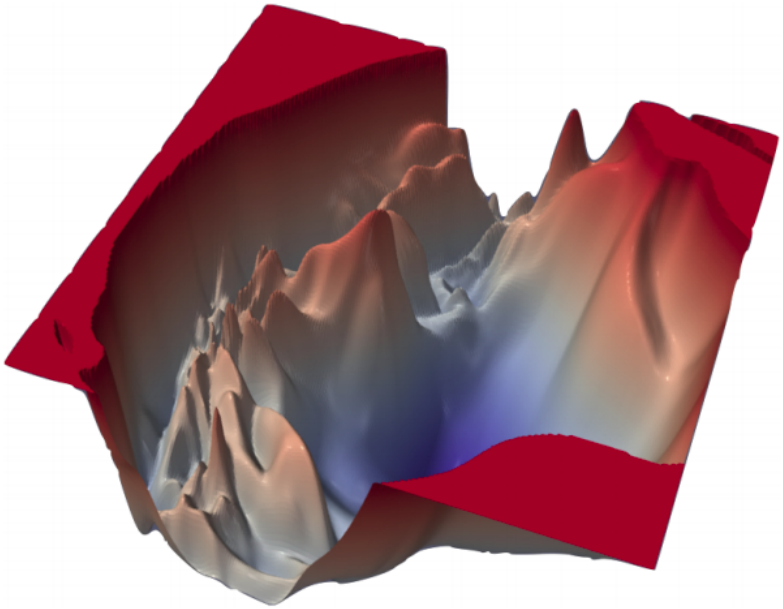

ResNet



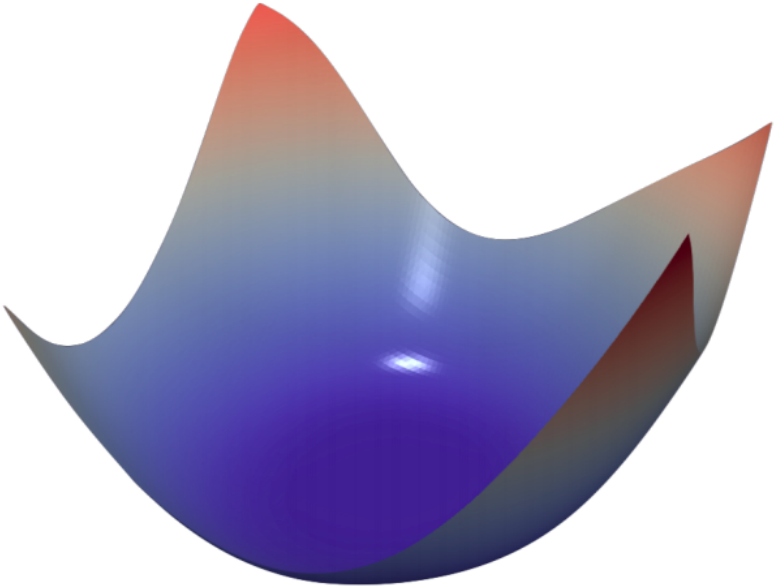

DenseNet

# Tehnike: DenseNet - učinak



(a) ResNet-110, no skip connections



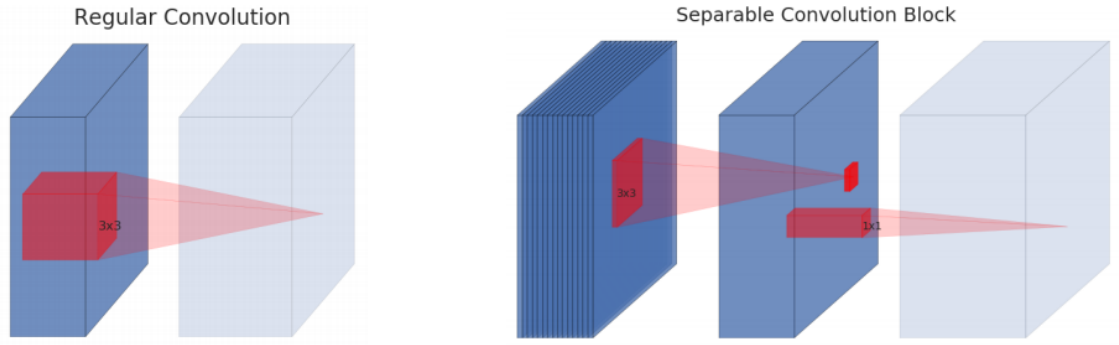
(b) DenseNet, 121 layers

# Tehnike: DenseNet - modeli

| Layers | Output Size | DenseNet-121 | DenseNet-169 | DenseNet-201 | DenseNet-264 |
|---|---|---|---|---|---|
| Convolution | $112 \times 112$ | $7 \times 7$ conv, stride 2 | | | |
| Pooling | $56 \times 56$ | $3 \times 3$ max pool, stride 2 | | | |
| Dense Block (1) | $56 \times 56$ | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$ | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$ | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$ | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$ |
| Transition Layer (1) | $56 \times 56$ | $1 \times 1$ conv | | | |
| | $28 \times 28$ | $2 \times 2$ average pool, stride 2 | | | |
| Dense Block (2) | $28 \times 28$ | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$ | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$ | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$ | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$ |
| Transition Layer (2) | $28 \times 28$ | $1 \times 1$ conv | | | |
| | $14 \times 14$ | $2 \times 2$ average pool, stride 2 | | | |
| Dense Block (3) | $14 \times 14$ | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 24$ | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$ | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$ | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 64$ |
| Transition Layer (3) | $14 \times 14$ | $1 \times 1$ conv | | | |
| | $7 \times 7$ | $2 \times 2$ average pool, stride 2 | | | |
| Dense Block (4) | $7 \times 7$ | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 16$ | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$ | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$ | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$ |
| Classification Layer | $1 \times 1$ | $7 \times 7$ global average pool | | | |
| | | 1000D fully-connected, softmax | | | |

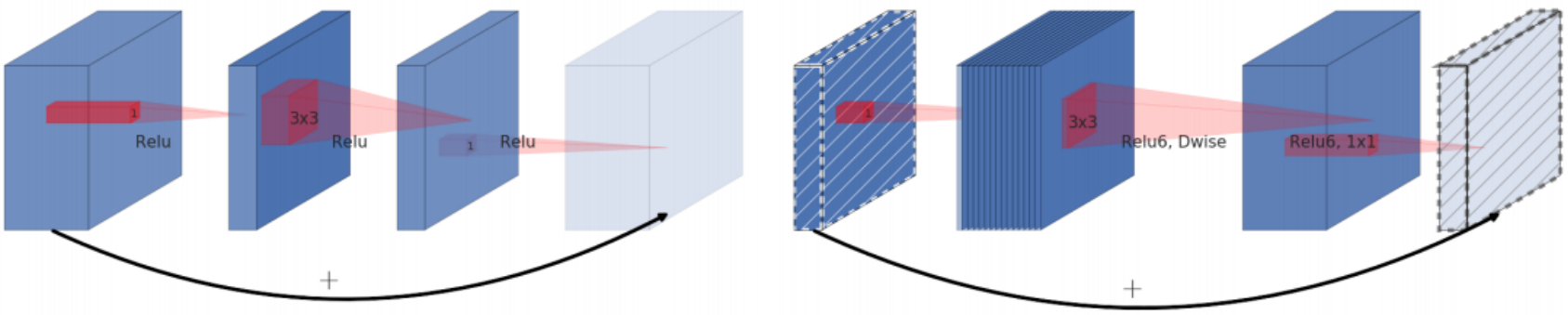
[huang17cvpr]

# Tehnike: MobileNet v2




[sandler18cvpr]

ideja 1: dubinski separirana konvolucija




[sandler18cvpr]

ideja 2: dubinski separirana konvolucija na t$\times$ "napuhanom" tenzoru

- lijevo: standardno rezidualno usko grlo
- desno: "invertirano" usko grlo (eng. inverted residual)
- "inverted residual": zbrajaju se tenzori sa smanjenim brojem

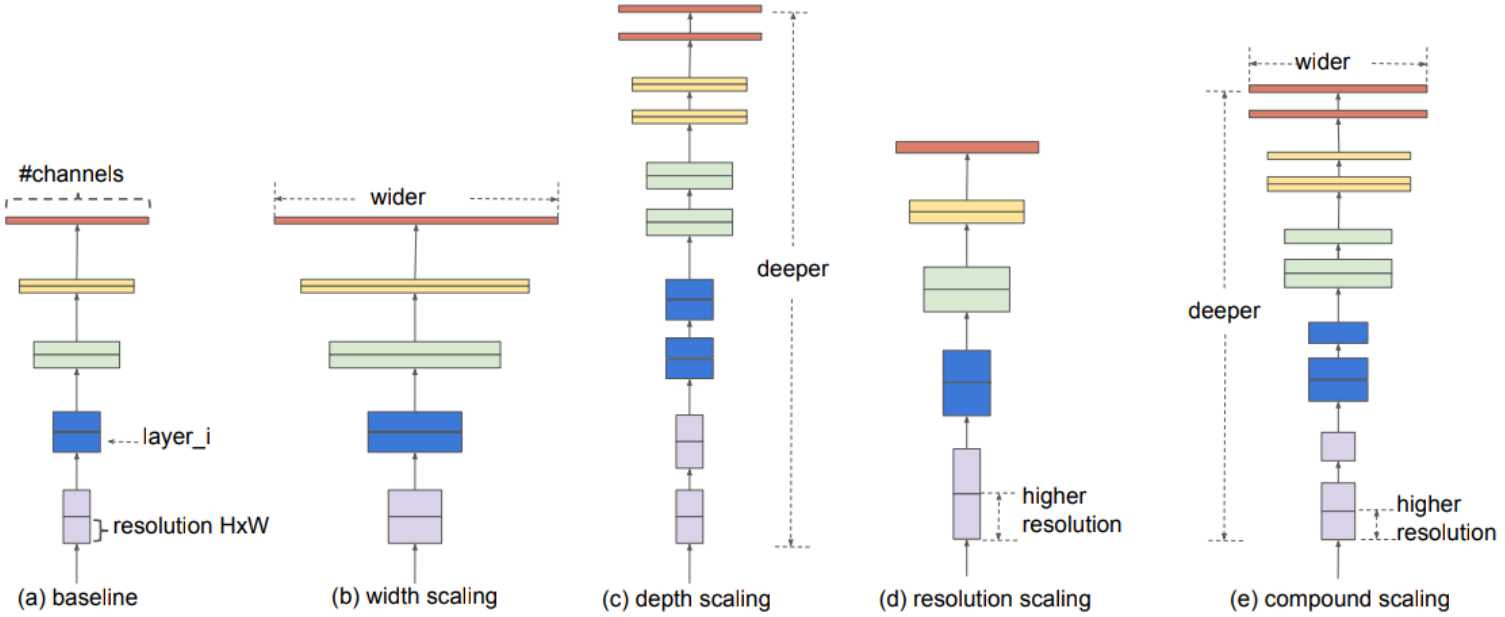
# Tehnike: MobileNet v2 (2)

| Input | Operator | $t$ | $c$ | $n$ | $s$ |
|---|---|---|---|---|---|
| $224^2 \times 3$ | conv2d | - | 32 | 1 | 2 |
| $112^2 \times 32$ | bottleneck | 1 | 16 | 1 | 1 |
| $112^2 \times 16$ | bottleneck | 6 | 24 | 2 | 2 |
| $56^2 \times 24$ | bottleneck | 6 | 32 | 3 | 2 |
| $28^2 \times 32$ | bottleneck | 6 | 64 | 4 | 2 |
| $14^2 \times 64$ | bottleneck | 6 | 96 | 3 | 1 |
| $14^2 \times 96$ | bottleneck | 6 | 160 | 3 | 2 |
| $7^2 \times 160$ | bottleneck | 6 | 320 | 1 | 1 |
| $7^2 \times 320$ | conv2d 1x1 | - | 1280 | 1 | 1 |
| $7^2 \times 1280$ | avgpool 7x7 | - | - | 1 | - |
| $1 \times 1 \times 1280$ | conv2d 1x1 | - | k | - | |

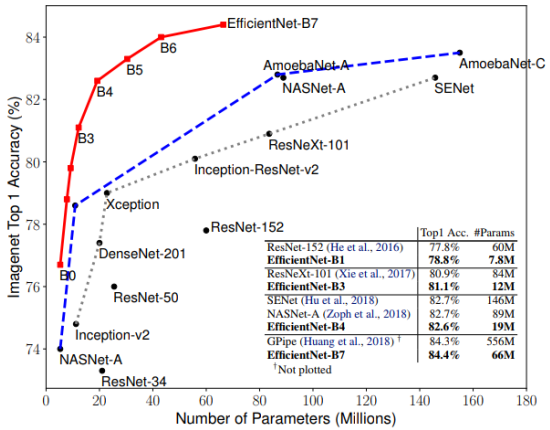
[sandler18cvpr]

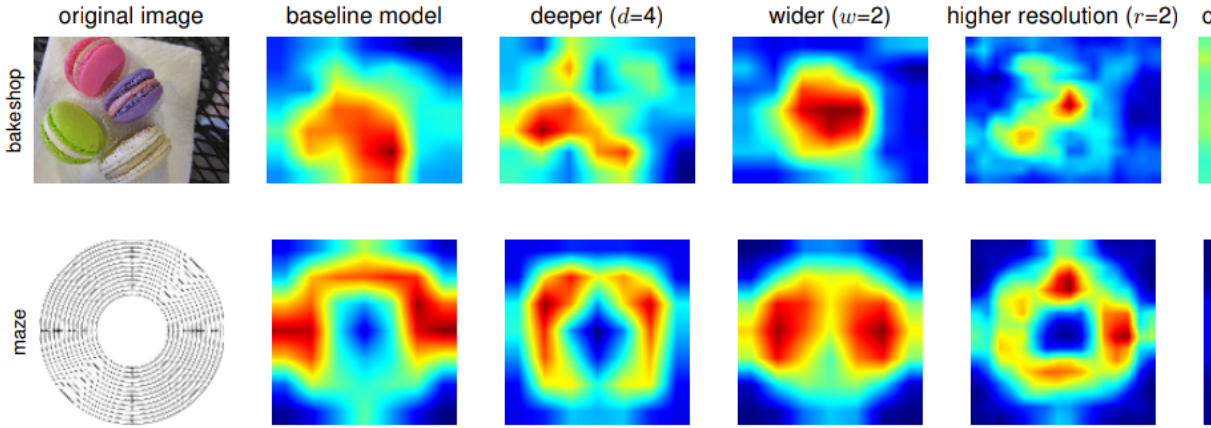
# Tehnike: EfficientNet

Ideja: dosljedno skaliranje po tri osi



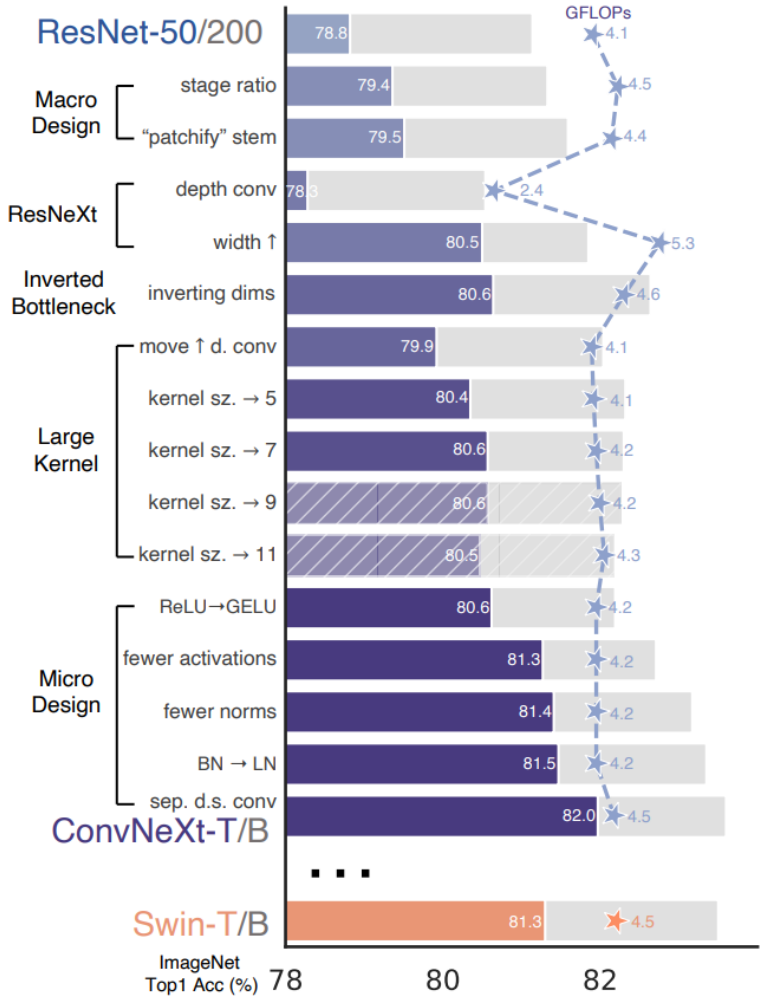

[tan19icml]

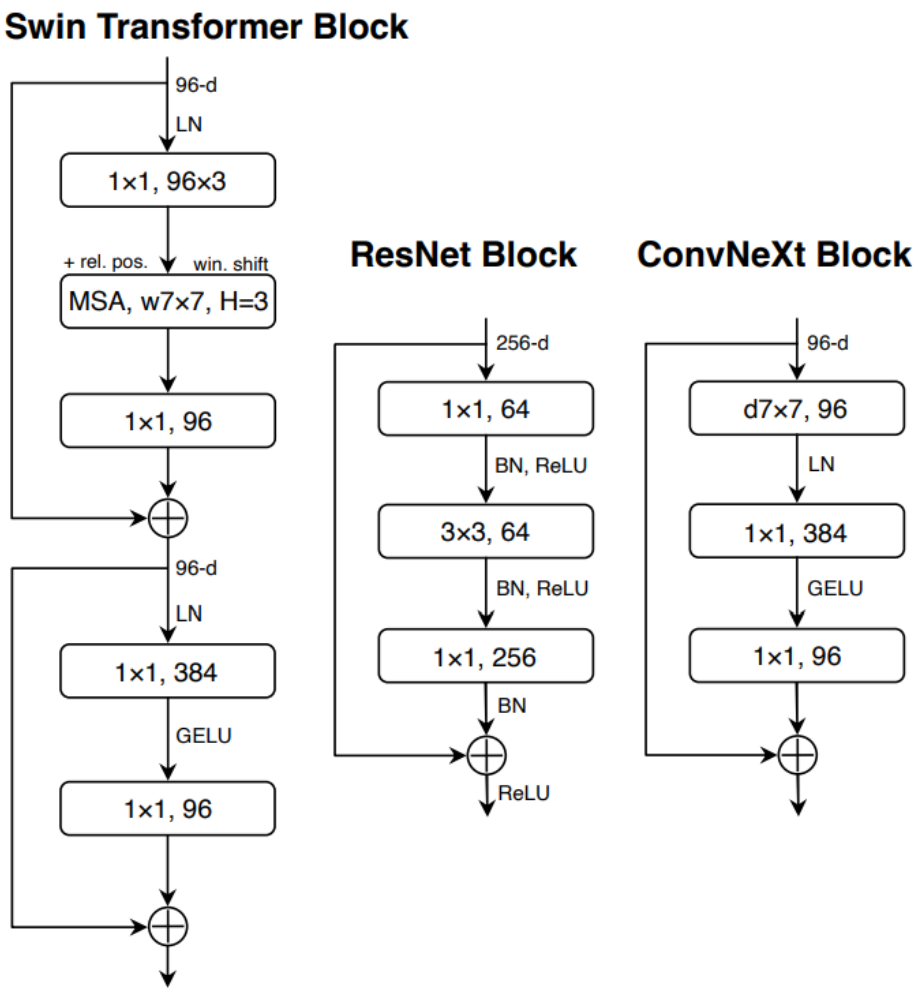


# Tehnike: more validation

ConvNeXt: ConvNet for the 2020:

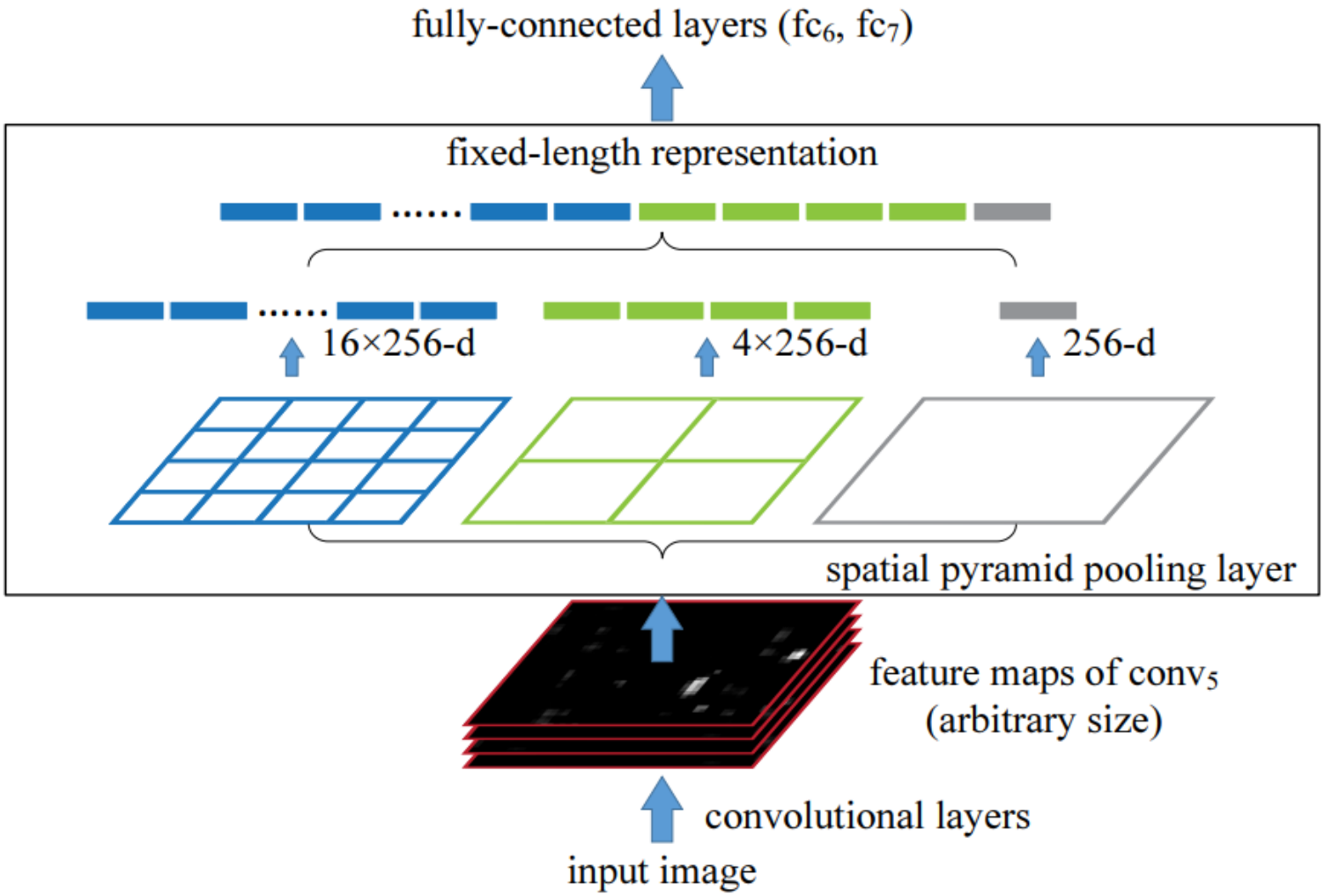



[liu22cvpr]

# Tehnike: increased receptive field

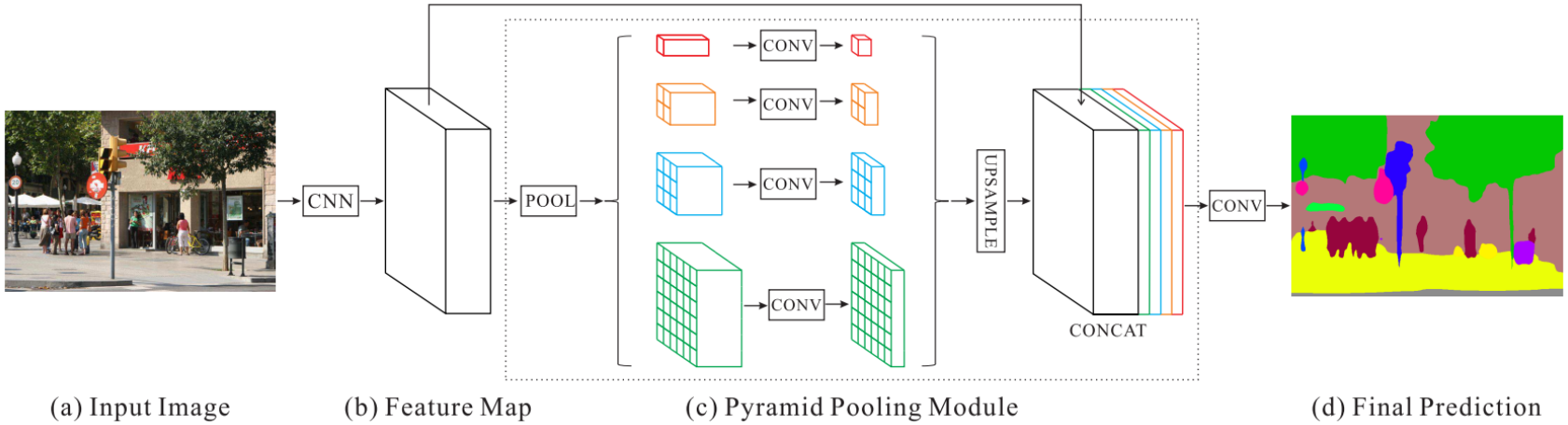
Spatial pyramid pooling (SPP): provide wide contextual information to subsequent convolutions




[he15tpami]

# Tehnike: increased receptive field (2)

Convolutional adaptation for dense prediction:




[zhao17cvpr]

Helps to recognize large objects with a model pre-trained on small images

# Implementacija: tensorflow

```
class TFLogreg:
  def __init__(self, d, m, param_delta=0.5, param_lambda=1e-3):
    self.X    = tf.placeholder(tf.float32, [None, d])
    self.Yoh_ = tf.placeholder(tf.float32, [None, m])
    self.W    = tf.Variable(tf.zeros([d, m], tf.float32))
    self.b    = tf.Variable(tf.zeros([m], tf.float32))
    self.probs = tf.nn.softmax(tf.matmul(self.X, self.W) + self.b)

    self.truelogprobs = tf.reduce_sum(
        self.Yoh_ * tf.log(self.probs), reduction_indices=1)
    self.cross_entropy = tf.reduce_mean(- self.truelogprobs)

    self.loss = self.cross_entropy
    self.trainer = tf.train.GradientDescentOptimizer(param_delta)
    self.train_step = self.trainer.minimize(self.loss)
    self.sess = tf.Session()

  def forward(self, X):
    probs = self.sess.run(self.probs, feed_dict={self.X: X})
    return probs

  def train(self, X,Yoh_, param_niter=100):
    self.sess.run(tf.global_variables_initializer())
    for i in range(param_niter):
      loss,_ = self.sess.run([self.loss, self.train_step],
        feed_dict={self.X: X, self.Yoh_: Yoh_})
      if i % 10 == 0:
        print(i, loss)
```

# Implementacija: pytorch

```
class LogisticRegression(nn.Module):
  def __init__(self, n_input, n_classes):
    super(LogisticRegression, self).__init__()
    self.W = nn.Parameter(torch.rand((n_input, n_classes)))
    self.b = nn.Parameter(torch.rand((1, n_classes)))

  def forward(self, x):
    return torch.matmul(x, self.W) + self.b

  def train(self, X,Yoh_, param_niters=100):
    data_x = Variable(torch.from_numpy(np.float32(X)).cuda())
    data_y = Variable(torch.from_numpy(np.int64(Y)).cuda())
    optimizer = optim.SGD(self.parameters(), lr=learning_rate)
    self.cuda()

    for e in range(param_niters):
      output = self.forward(data_x)
      loss = nn.CrossEntropyLoss().forward(output, data_y)
      loss.backward()
      optimizer.step()
      optimizer.zero_grad()
```

# Razumijevanje: vizualizacija

Motivacija: steći uvid u funkcioniranje dubokog modela

Pristup:

- odabrati mapu značajki nekog sloja
- pronaći istaknutu aktivaciju u toj mapi
- pokazati prozor ulazne slike koji rezultira danom aktivacijom

članak

# Razumijevanje: vizualizacija (2)


[zeiler14eccv]

# Razumijevanje: vizualizacija (3)

Rezultat:

- prikazane vizualizacije pokazuju da značajke postaju sve apstraktnije kako slojevi postaju dublji
- također raste i invarijantnost na promjene oblika i razlikovanje razreda
  - to se lijepo vidi na primjeru značajke koja reagira na pse
- možemo li zaključiti su latentni slojevi mreže naučili semantiku dijelova promatranih razreda?

Nažalost, takav zaključak bio bi preoptimističan:

- standardno učenje dubokih modela za klasifikaciju slike pronalazi efikasne značajke koje nisu robusne

Na to ukazuje postojanje neprijateljskih primjera

# Razumijevanje: Optimiranje slike

Ideja: maksimizirati ulaz u softmaks za promatrani razred izmjenom ulazne slike

Na ulaz možemo postaviti:

- praznu sliku
- sliku nekog drugog razreda

Rezultati:

- ovim putem možemo generirati sliku koja predstavlja "srž" razreda i tako dokazati da mreža ipak jest nešto naučila
- promatranjem apsolutnog gradijenta gubitka (ili dominantnog logika) s obzirom na sliku možemo lokalizirati piksele koji su odgovorni za odluku modela
  - ovo može poboljšati razumijevanje načina rada naučenog modela (posebno korisno pri debuggiranju!)

članak

# Razumijevanje: Optimiranje slike (2)


Figure 1: **Numerically computed images, illustrating the class appearance models, learnt by a ConvNet, trained on ILSVRC-2013.** Note how different aspects of class appearance are captured in a single image. Better viewed in colour.

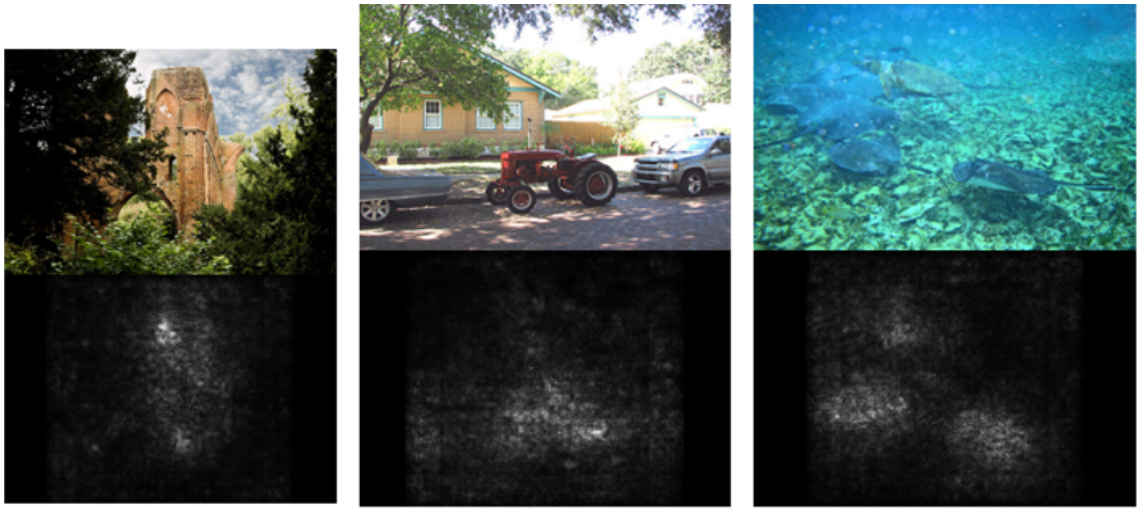
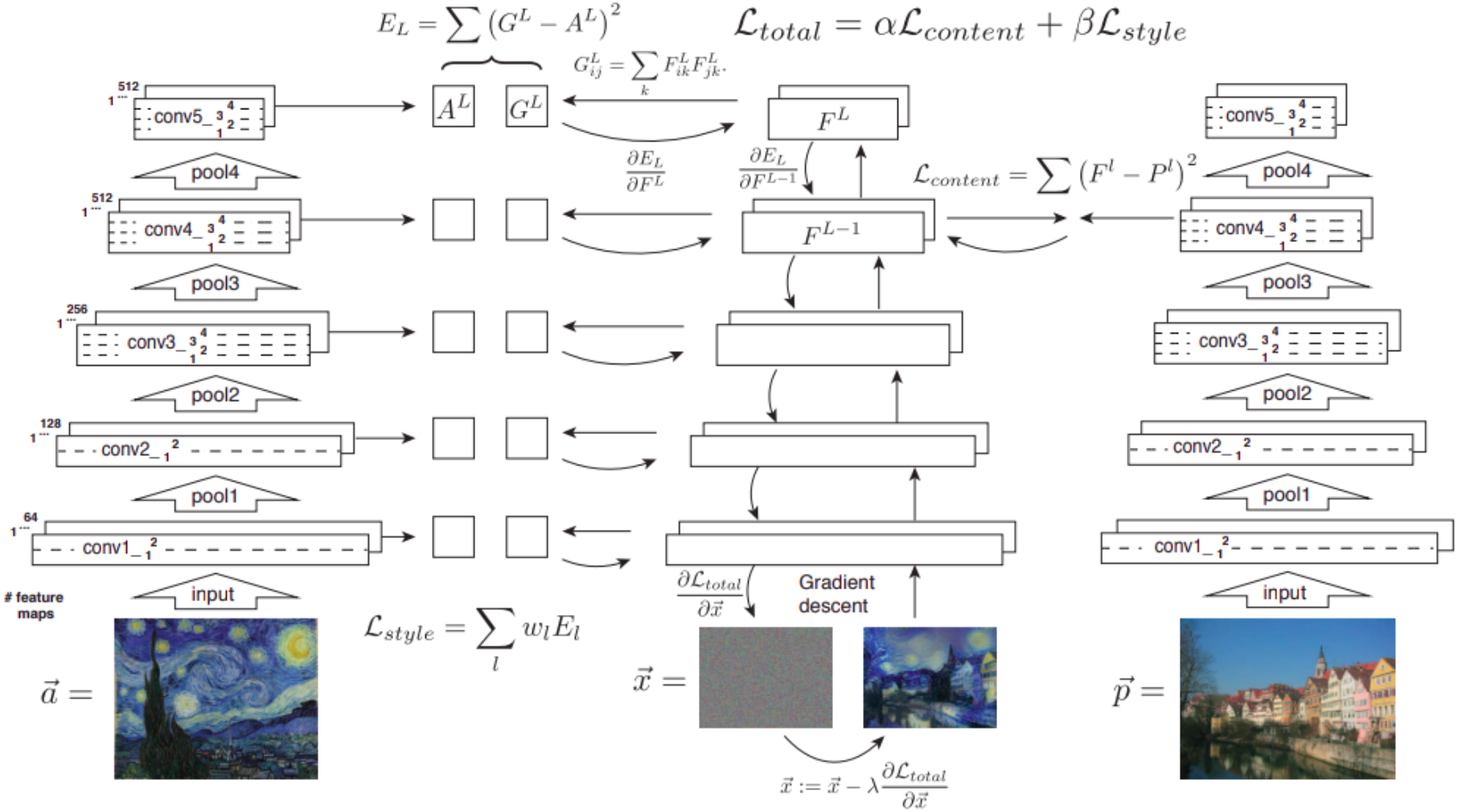


Figure 2: **Image-specific class saliency maps for the top-1 predicted class in ILSVRC-2013 test images.** The maps were extracted using a single back-propagation pass through a classification ConvNet. No additional annotation (except for the image labels) was used in training.

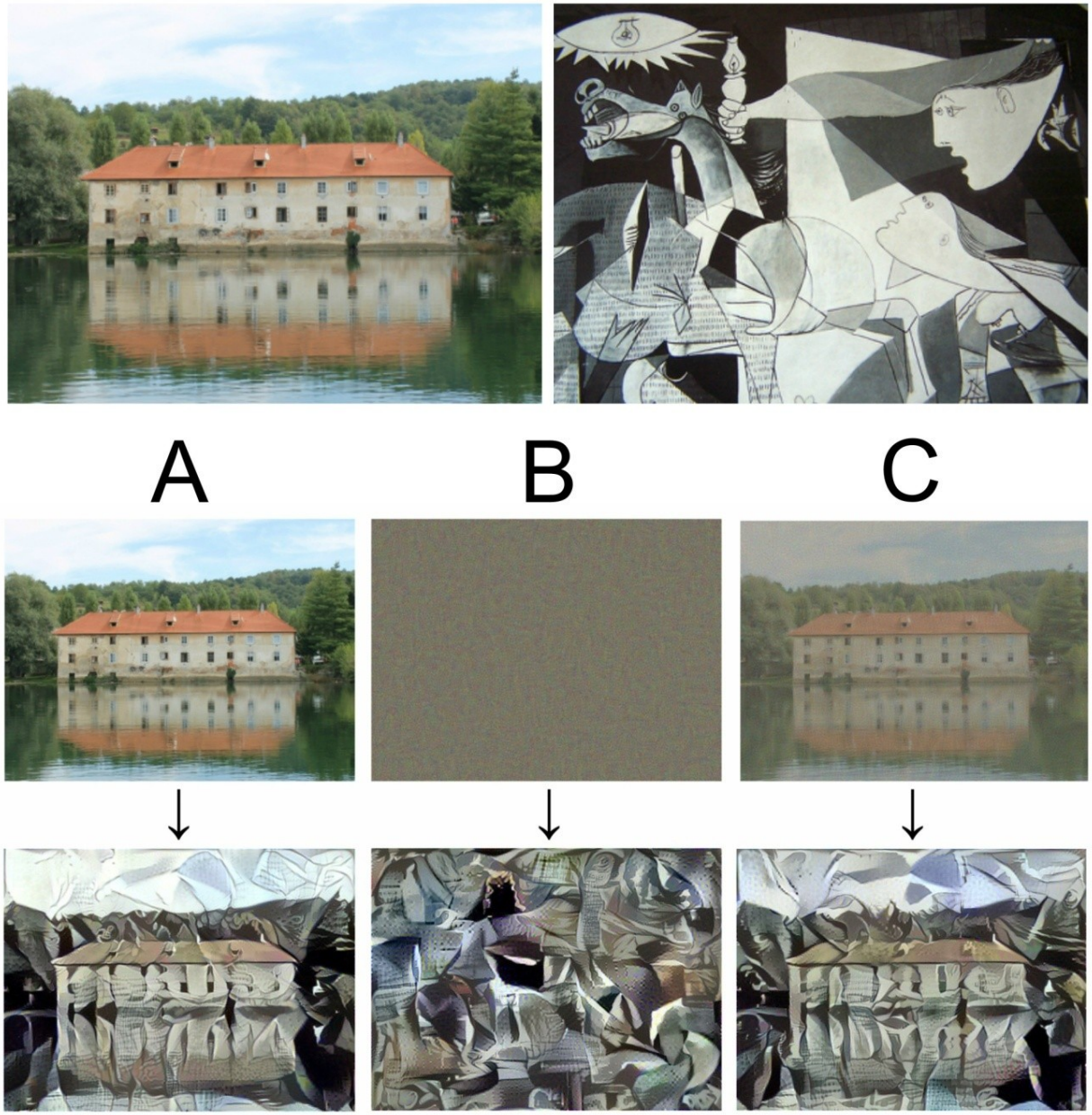
[simonyan14iclr]

# Razumijevanje: Prijenos stila



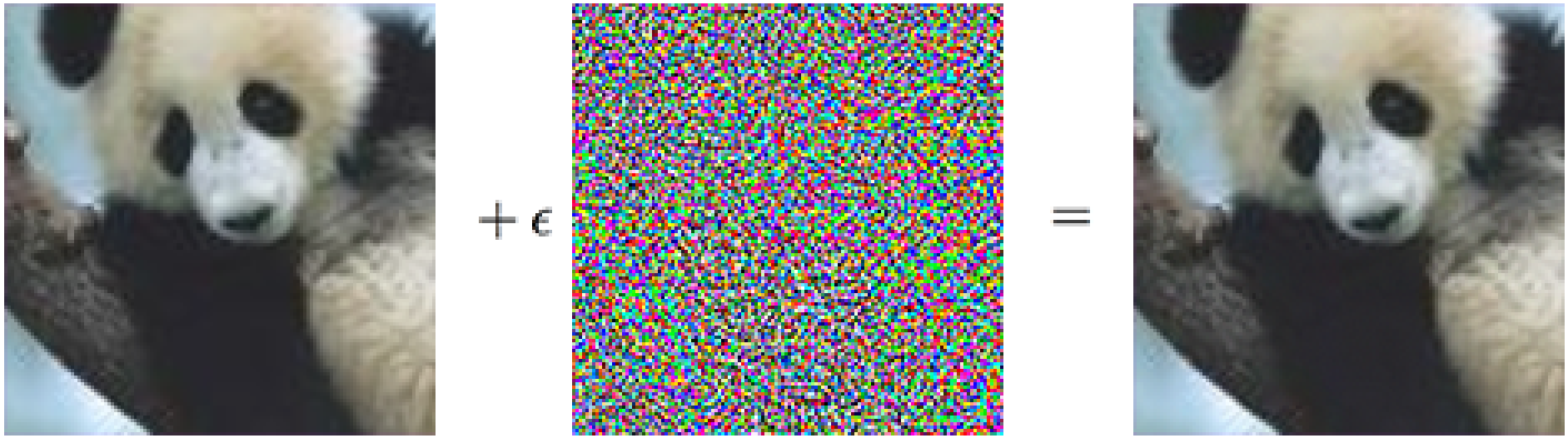

[gatys16cvpr]

# Razumijevanje: Prijenos stila (2)




[livkovic17bs]

# Razumijevanje: neprijateljski primjeri




"panda"
57.7% confidence

"gibbon"
99.3% confidence

[szegedy14iclr]

```
# neprijateljski primjeri u PyTorchu:
output = model(x)
loss = criterion(output, realLabel)
loss.backward()
x_grad = torch.sign(x.grad.data)
x_adv = torch.clamp(x.data + epsilon * x_grad, 0, 1)
```

# Razumijevanje: robusnost

Classic training: maximize log-likelihood of the data

$$\hat{\theta} = \arg\min_{\theta} \mathbb{E}_{\mathcal{D}} - \log P(y_i|\mathbf{x}_i, \theta)$$

Adversarial training: maximize log-likelihood of worst-case data

$$\hat{\theta} = \arg\min_{\theta} \mathbb{E}_{\mathcal{D}} \max_{\delta} - \log P(y_i|\mathbf{x}_i + \delta, \theta)$$

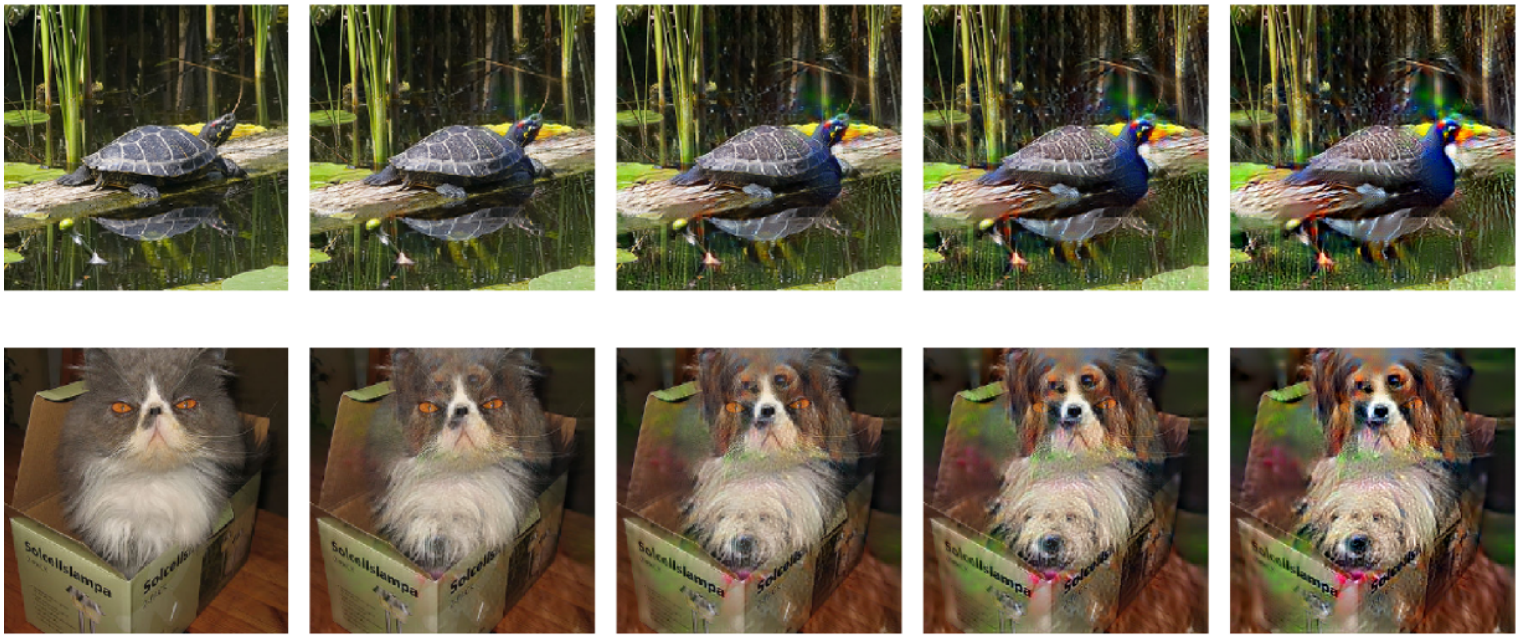
Adversarial training has been introduced years ago as a regularization technique [szegedy14iclr]

Recent results show additional benefits may be achieved by more aggressive search for the worst-case perturbation [madry18iclr]

# Razumijevanje: robusnost

Result 1: robustness to adversarial examples [madry18iclr]

Result 2: interpretable gradients [tsipras18arxiv]



[ilyas19neurips]

Downside 1: robust features currently do not lead to better performance

Downside 2: much more computationally intensive than classic training

# Razumijevanje: sigurnost



[gu17arxiv]

Subtle changes to training images install a backdoor into the model.

In order to make the attack **hard to notice** the attackers aim to:

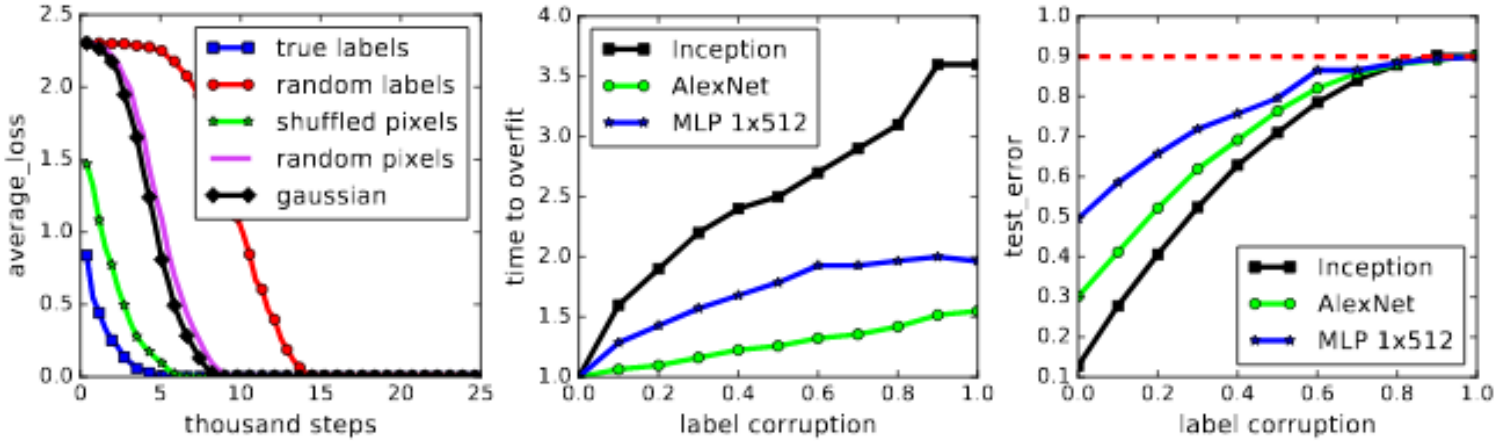
- poison a small fraction of training images
- choose an unobtrusive trigger
- keep the generalization performance on the test dataset.

**Backdoor defenses** strive to detect the triggers and attenuate the backdoor while keeping generalization through robust training

# Razumijevanje: prenaučenost?

Effective capacity of convolutional models is large enough to shatter popular image classification datasets:

- the model is able to memorize the entire training dataset!




(a) learning curves (b) convergence slowdown (c) generalization error growth

[zhang17iclr]

Yet, our models generalize well when trained on correct labels:

- some argue that this happens since the effective model capacity is determined by early stopping
- we always stop training our models before convergence

# Razumijevanje: tekstura (1)

Sliku možemo prepoznati preko teksture (lijevo) ili preko oblika (sredina).

(a) Texture image

| | |
|---|---|
| 81.4% | **Indian elephant** |
| 10.3% | indri |
| 8.2% | black swan |

(b) Content image

| | |
|---|---|
| 71.1% | **tabby cat** |
| 17.3% | grey fox |
| 3.3% | Siamese cat |

(c) Texture-shape cue conflict

| | |
|---|---|
| 63.9% | **Indian elephant** |
| 26.4% | indri |
| 9.6% | black swan |

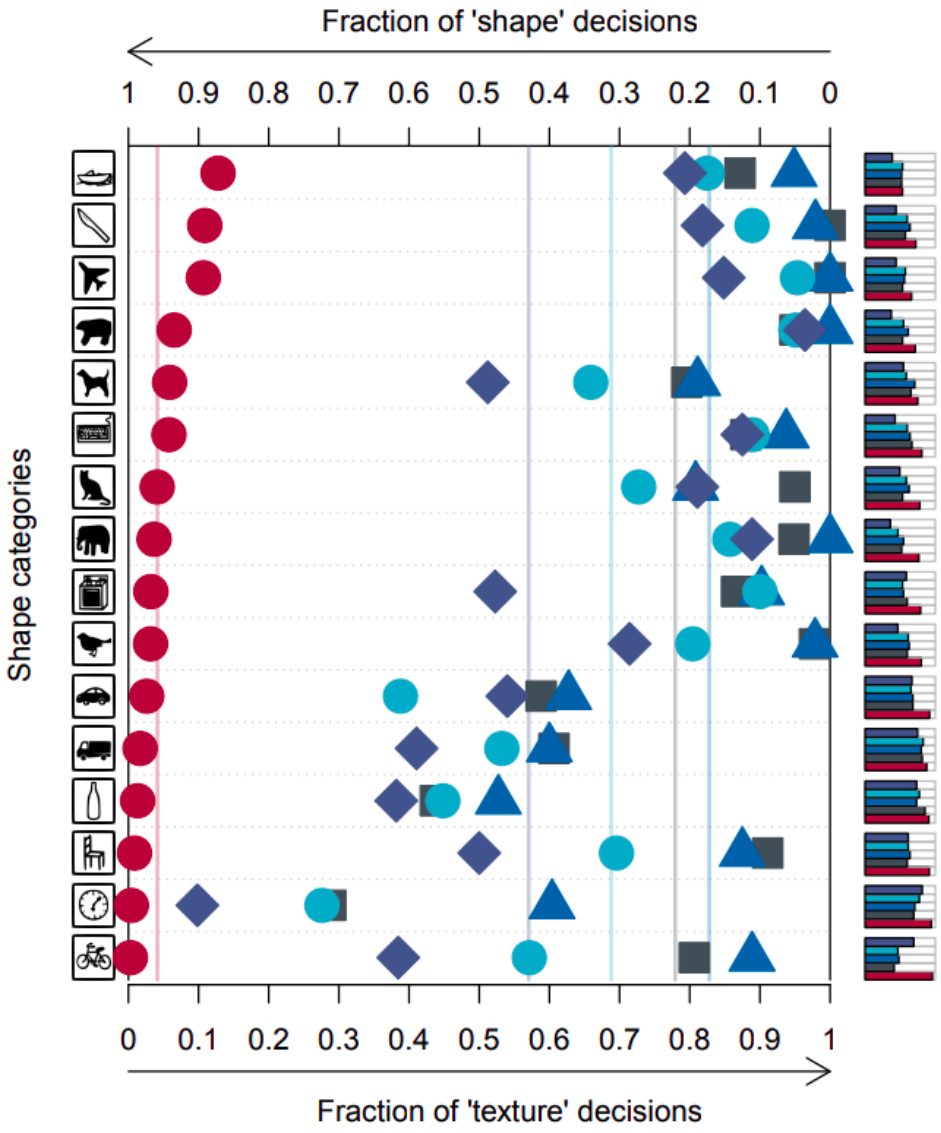
[geirhos19iclr]

Ako je su tekstura (lijevo) i oblik (sredina) u konfliktu, konvolucijski model će se prikloniti teksturi (desno)

# Razumijevanje: tekstura (2)

Kod slika s konfliktnim indicijama ljudi preferiraju oblik a konvolucijski modeli teksturu

- crveni krug - osobe
- romb - AlexNet
- trokut - VGG-16
- plavi krug - GoogLeNet
- kvadrat - ResNet-50




[geirhos19iclr]

# Razumijevanje: tekstura (3)

Postavlja se pitanje: možemo li konvolucijske modele nagovoriti da se ponašaju više kao ljudi?

Odgovor je: da, ako ih tijekom učenja spriječimo da profitiraju na teksturi

To pokazuje učenje na stiliziranom ImageNetu

[geirhos19iclr]

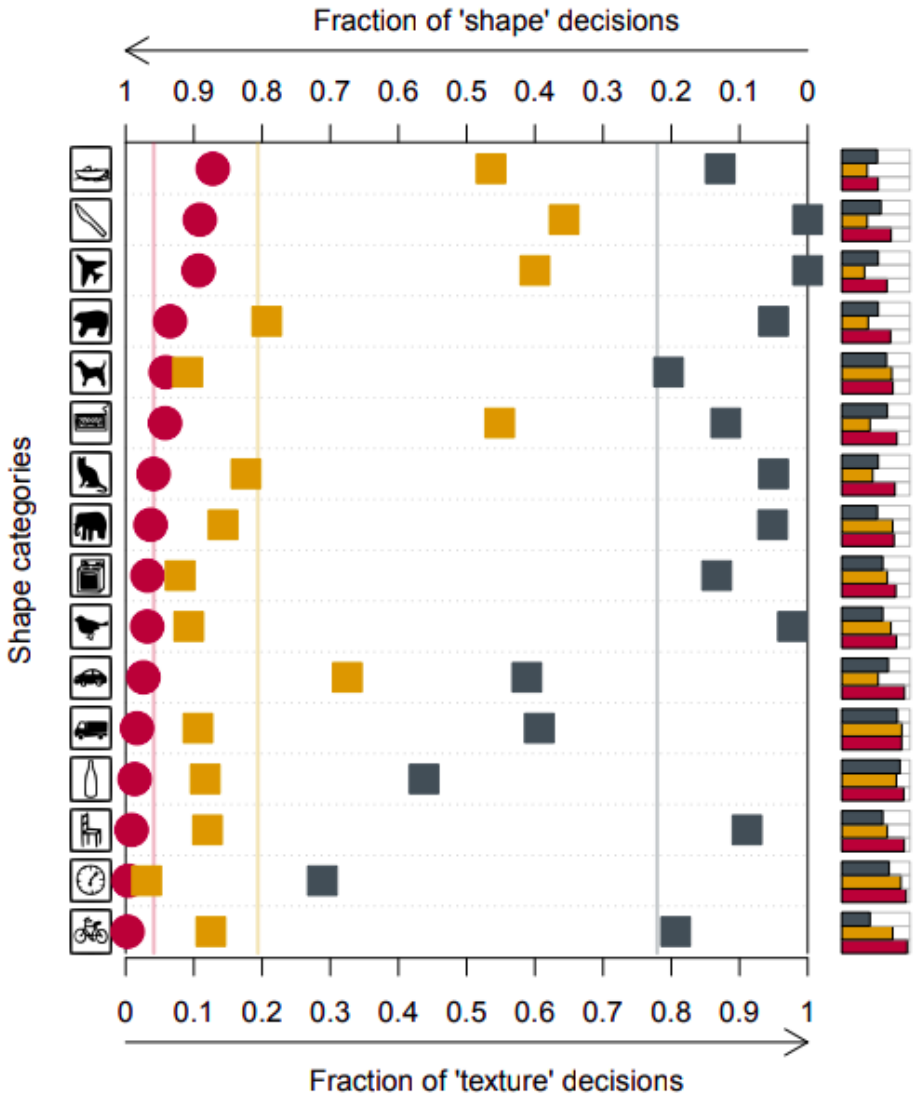
# Razumijevanje: tekstura (4)

Model učen na stiliziranom ImageNetu (SIN) češće preferira oblik nauštrb teksture

- crveni krug - osobe
- žuti kvadrat - ResNet-50 SIN
- plavi kvadrat - ResNet-50 IN

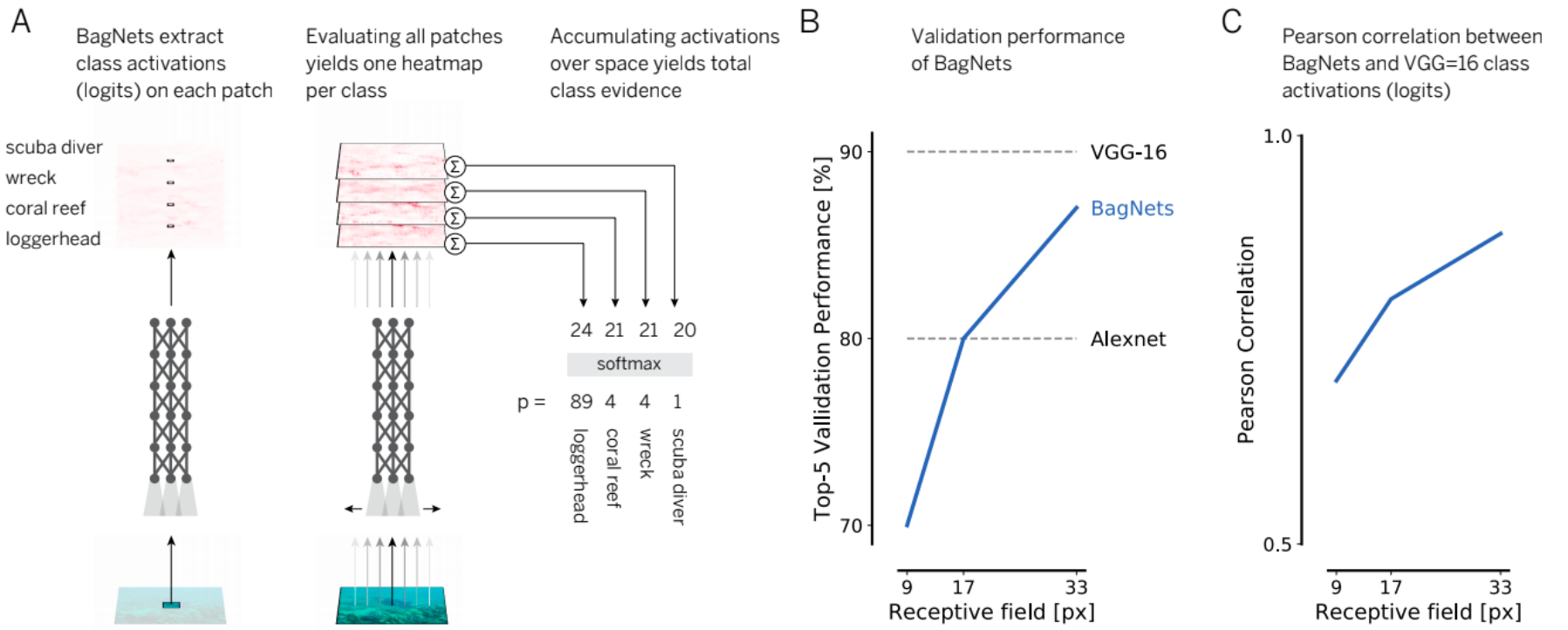
Zaključak: konvolucijski modeli skloni su oslanjati se na teksturu.

Ako ih želimo natjerati da se ponašaju sličnije ljudima, trebamo ih odučiti od toga :-)




[geirhos19iclr]

# Razumijevanje: tekstura (5)




[brendel19iclr]

- Ovakvo učenje sprječava model da uzme u obzir širi kontekst
- međutim, validacija je slična kao i u standardnom slučaju
- ovo ponovo sugerira da se konvolucijski modeli pretjerano oslanjaju na lokalnu teksturu

# Primjene

- detekcija pojavljivanja objekata, svojstava scene ili događaja
  - višerazredna (multi-class) ili višeoznačna (multi-label) klasifikacija
- određivanje položaja (lokaliziranje) objekata
- razvrstavanje piksela slike u semantičke razrede
  - semantička i panoptička segmentacija, segmentacija instanci
- rekonstrukcijski zadatci
  - stereoskopska dubina, optički tok, monokularna dubina
- anticipiranje događaja, prognoziranje budućnosti
- asocijativno pretraživanje slikovnih biblioteka
- korespondencijske metrike
- raspoznavanje akcija u videu

# Challenges: false positives due to context

Current models tend to produce false positive detections due to context

- good performance likely due to recognition of **easy context**

Recognition on Pascal VOC 2012 (confident TP, high FP, low FN):

airplane



bicycle

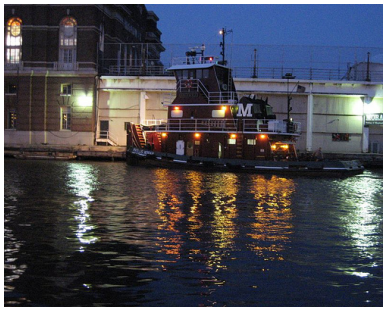


bird



# Challenges: false positives due to context (2)

boat



bottle

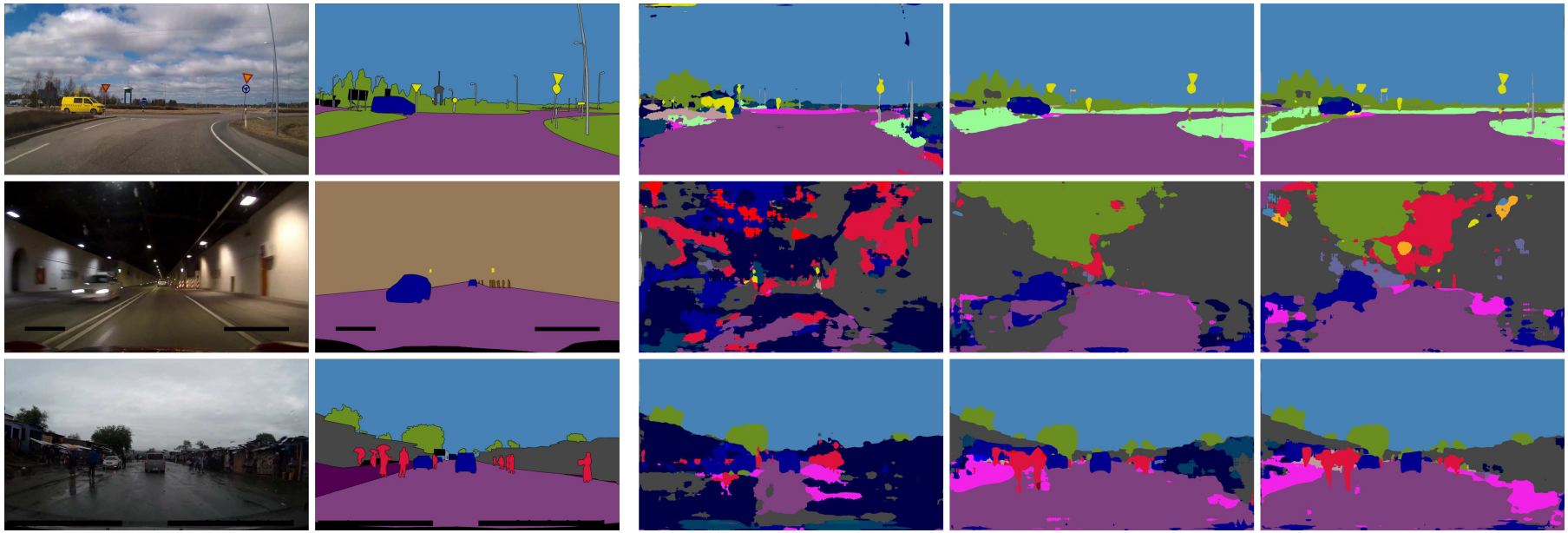

bus

car

# Challenges: cross-dataset generalization

Often our models generalize well only within dataset

For instance, images from the novel WildDash dataset (left) fool most models trained on popular datasets such as Cityscapes or Vistas (right)



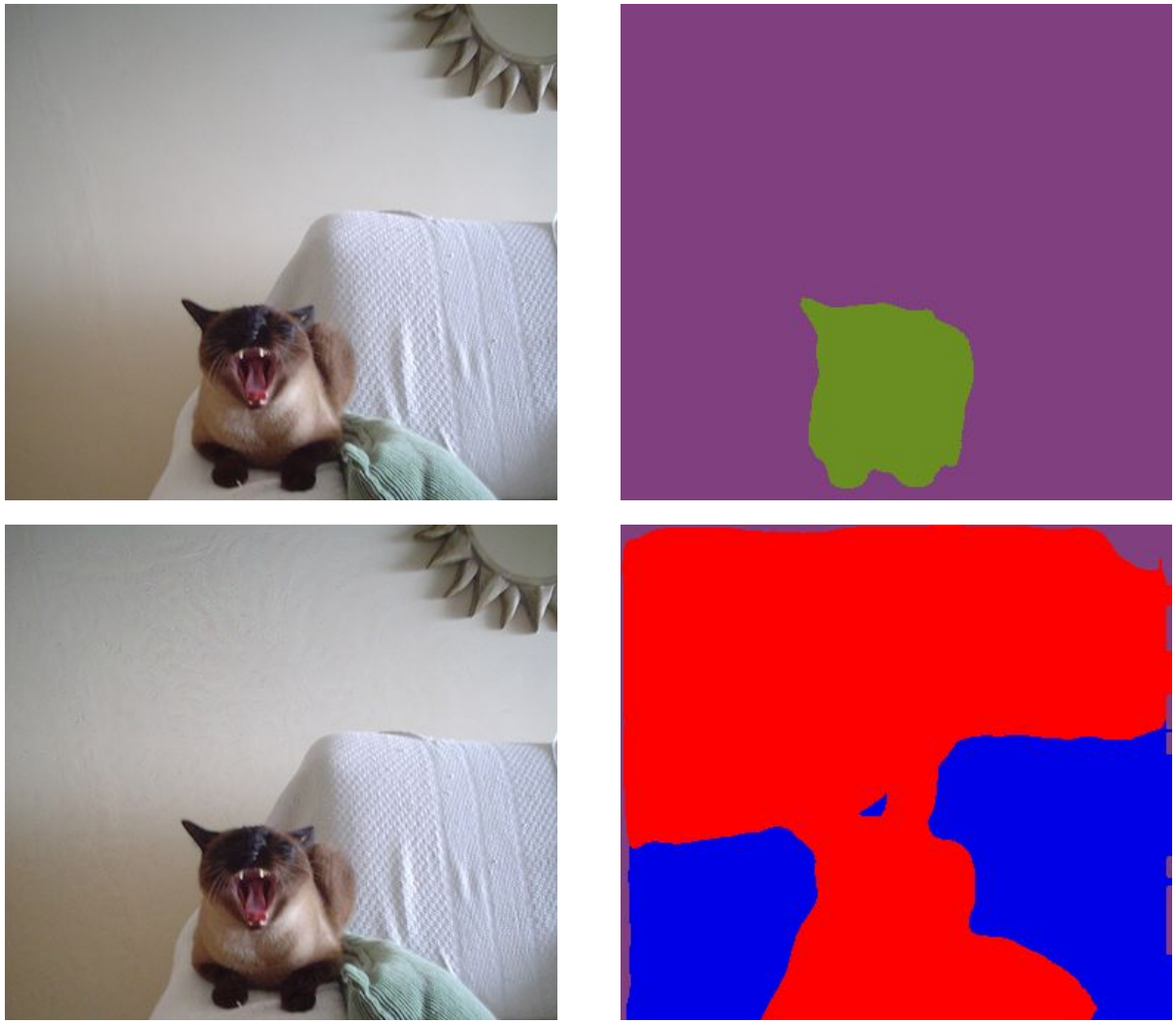
[zendel18eccv]

Conclusion: models tend to *overfit* to **dataset specifics**

- camera, weather, environment, climate, ...

# Challenges: Adversarial examples

Imperceptible perturbations may invalidate prediction [szegedy14iclr]:



[kreso18ep]

# Challenges: solutions

Presented challenges suggest that state-of-the-art systems:

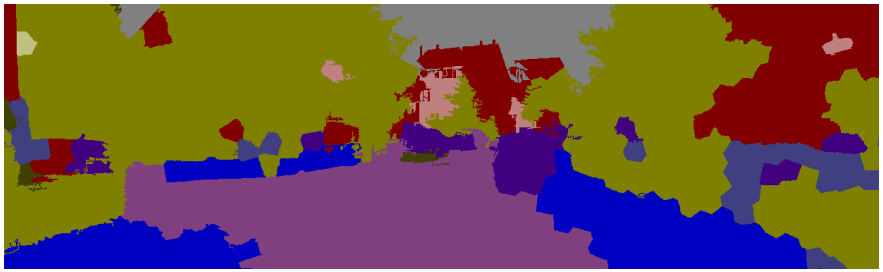
- are unable to comprehend limits of their expertise
- under-achieve by relying on **easy** but **non-robust** features
  - humans also jump to conclusions: e.g. a person without a wedding ring is considered single
  - such inferrence is not appropriate for mission critical systems

Prominent ways for getting closer to truly intelligent artificial vision:

- improve training data [rob18cvpr]
- detect out-of-distribution (OOD) input [bevandic18arxiv]
- improve the training process [tsipras18arxiv]

# Zaključak: state of the art

Dramatično poboljšanje predikcije u posljednjih nekoliko godina:

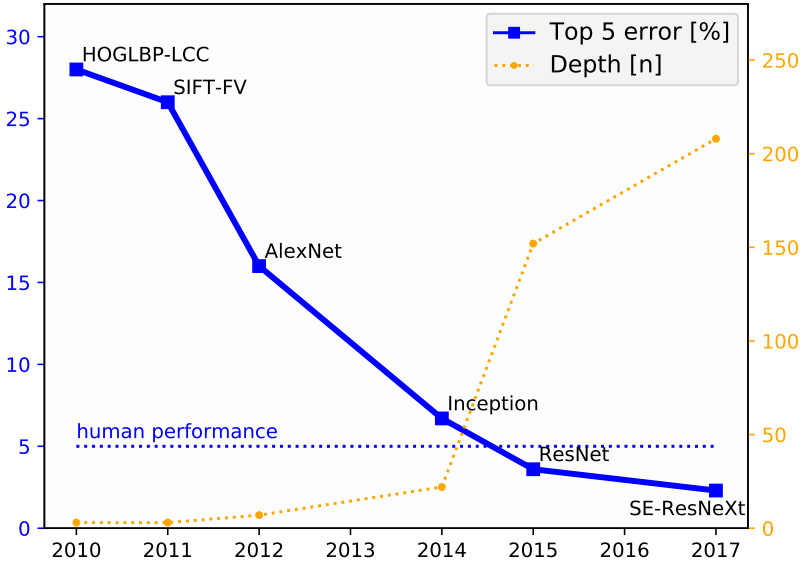


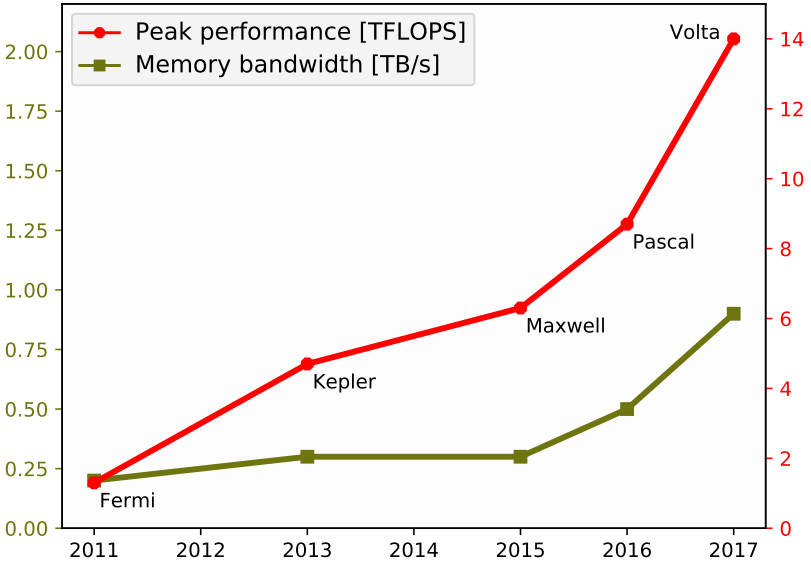
2015 [ros15wacv]

2017 [kreso17iccvw]

# Zaključak: izgledi

Točnost sustava računalnog vida i dalje će rasti:



# Zaključak: izgledi (2)

Važni smjerovi istraživanja:

- smanjenje ovisnosti o označenim podatcima (nenadzirano, polunadzirano, samonadzirano i slabo nadzirano učenje)
- procjena nesigurnosti predikcija (izvandistribucijski i neprijateljski primjeri)
- zaključivanje na sklopovlju s malom snagom (kvantizacija, destilacija)
- novi zadatci (npr. prognoziranje) i arhitekture (transformeri)

Glavni izazovi:

- pretjerano oslanjanje na teksturu i kontekst
- pomak domene
- neprijateljske perturbacije

# ZAHVALA

Ova predavanja proizišla su iz istraživanja koje je financirala Hrvatska zaklada za znanost projektom IP-2020-02-5851 Adept.

http://multiclod.zemris.fer.hr