Duboko učenje

Konvolucijski modeli za računalni vid: stanje tehnike

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:

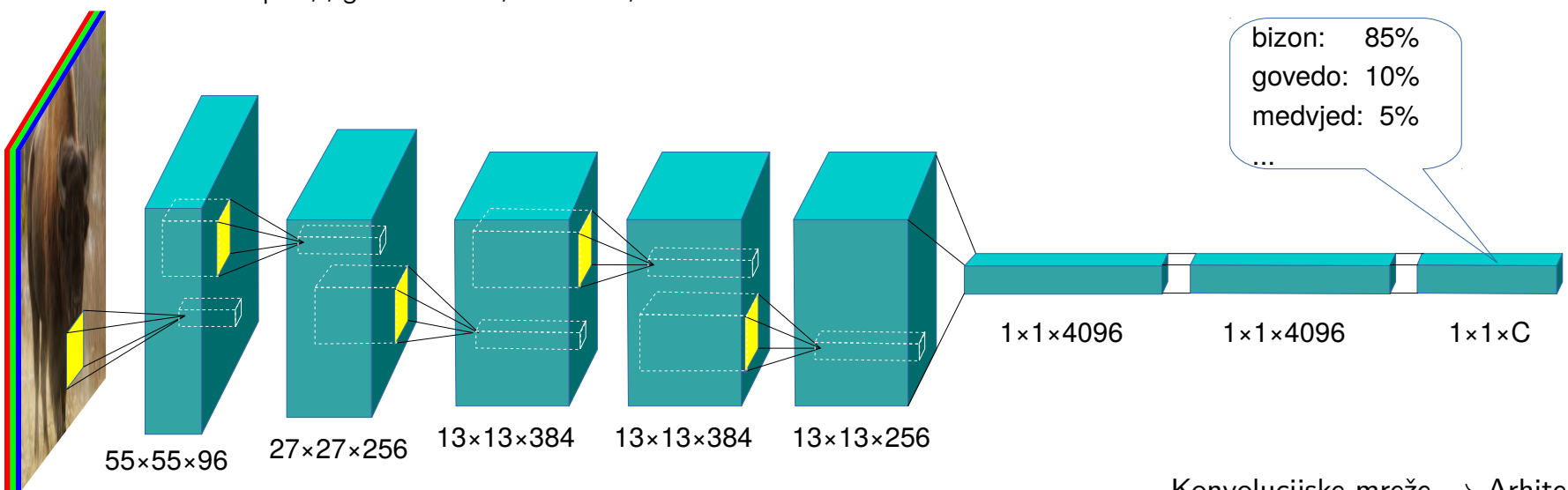
[image-net.org]

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

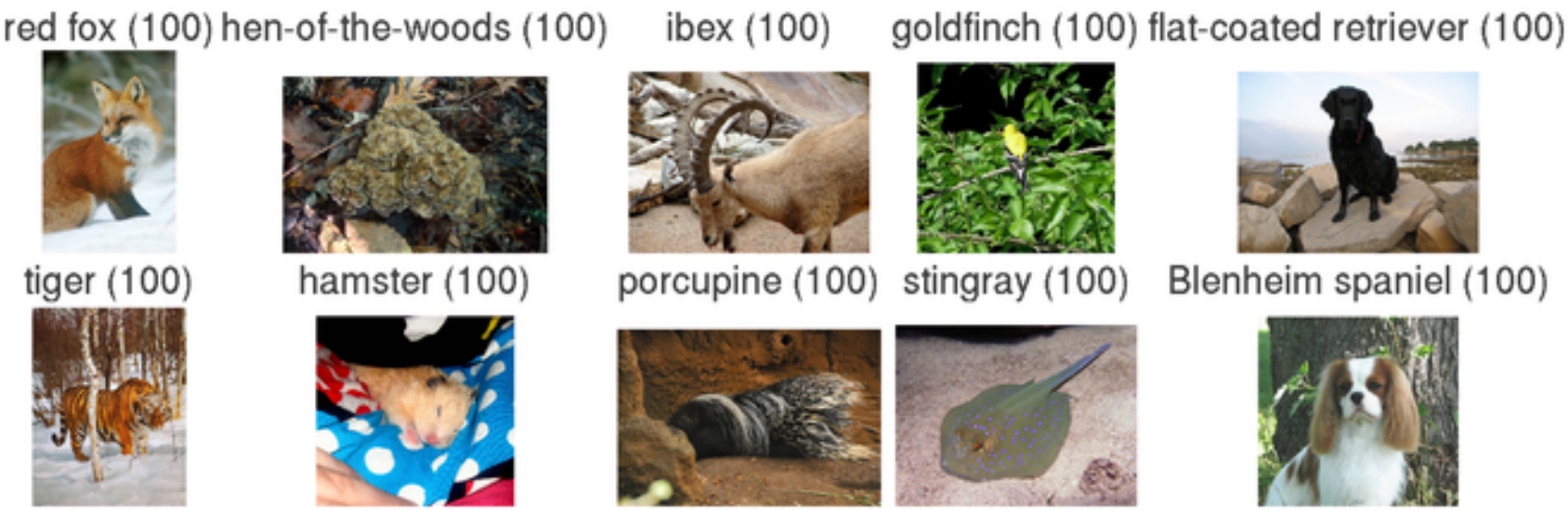
`https://github.com/albanie/convnet-burden`



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%)



Arhitekture: ImageNet teški primjeri

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

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

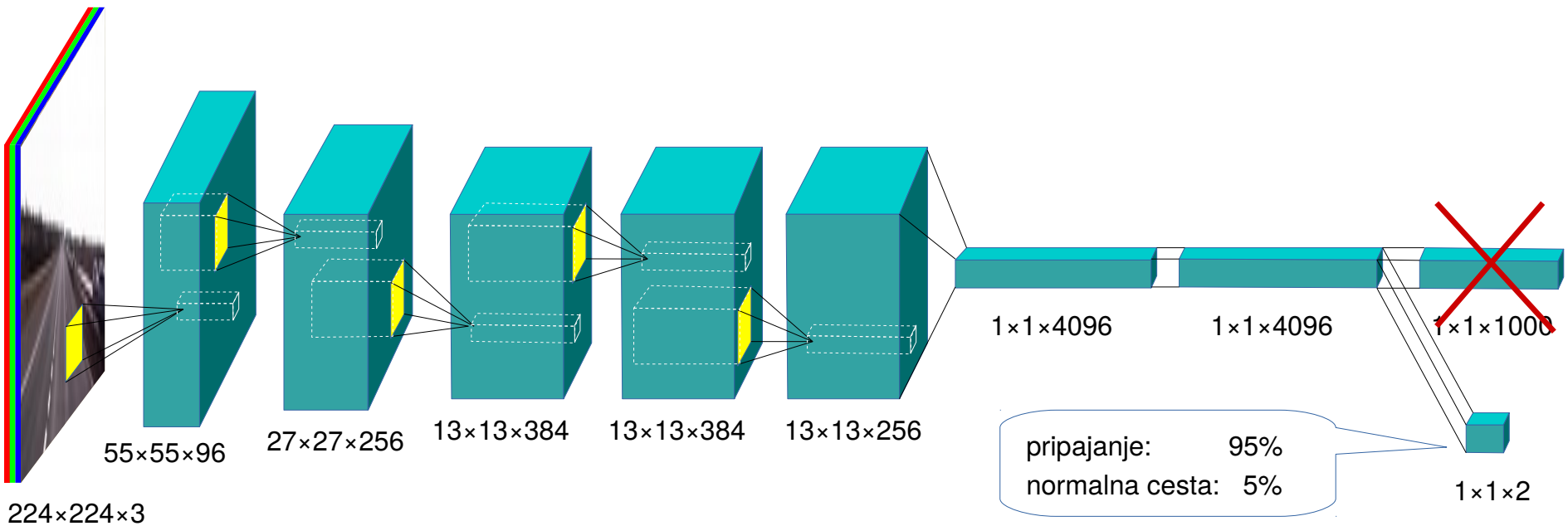
Razredi teški za **GoogleNet** (2014, 6.7%):

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

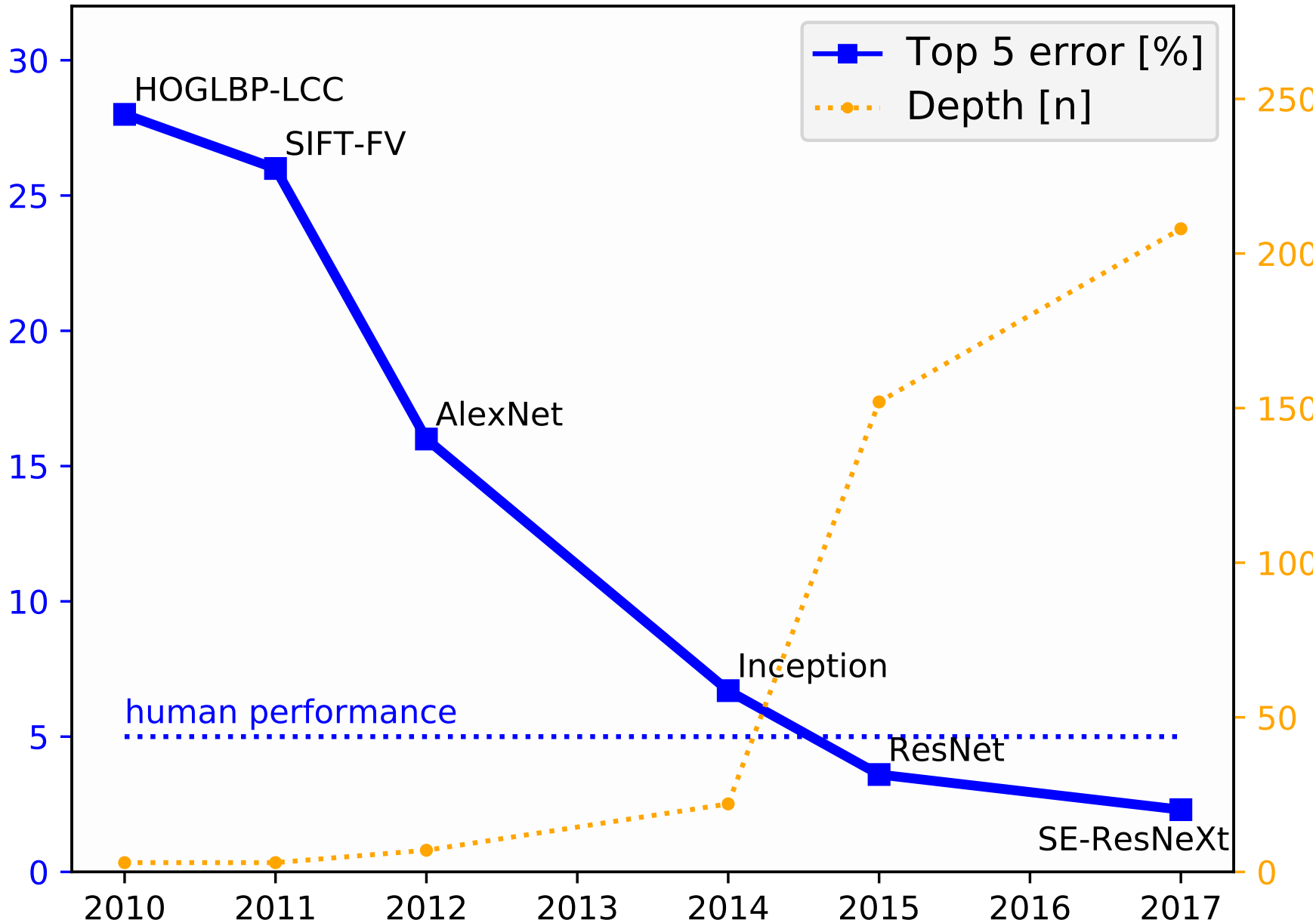
Arhitekture: prijenos znanja

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	22×4	$6{\cdot}10^6$	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%

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

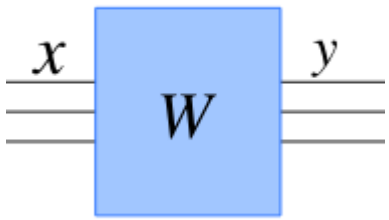
Arhitekture: VGG

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 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

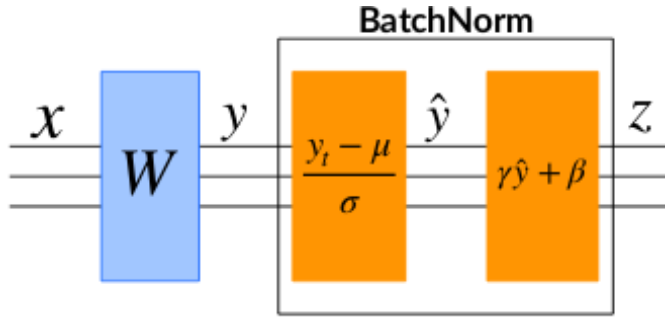
[simonyan15iclr]

Arhitekture: BatchNorm

Important idea: normalize activations produced by each layer



vanilla layer



layer with batchnorm

During training:

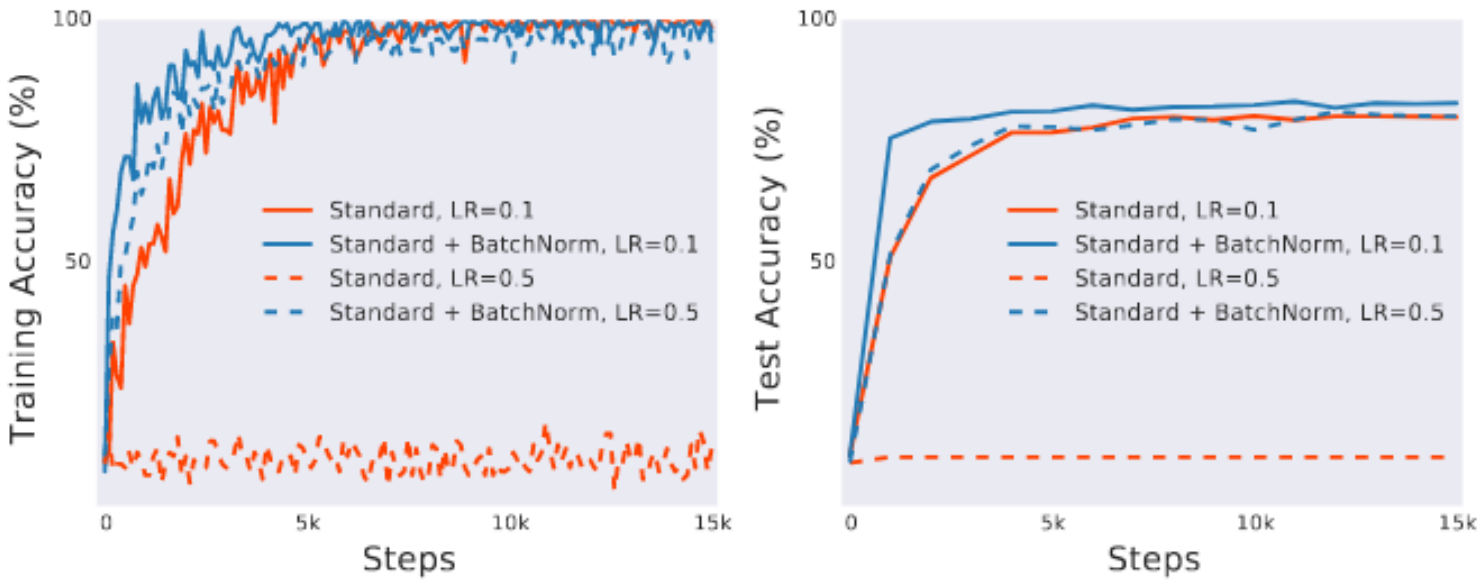
- the activations in the batch are $\mathcal{N}(0, 1)$
  - ideal conditions for ReLU
- all activations are equally important
  - stable learning, tolerance for high learning rates
- each particular image is jittered in an epoch-specific manner
  - this makes it hard for the model to overfit (=regularization)

Arhitekture: 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

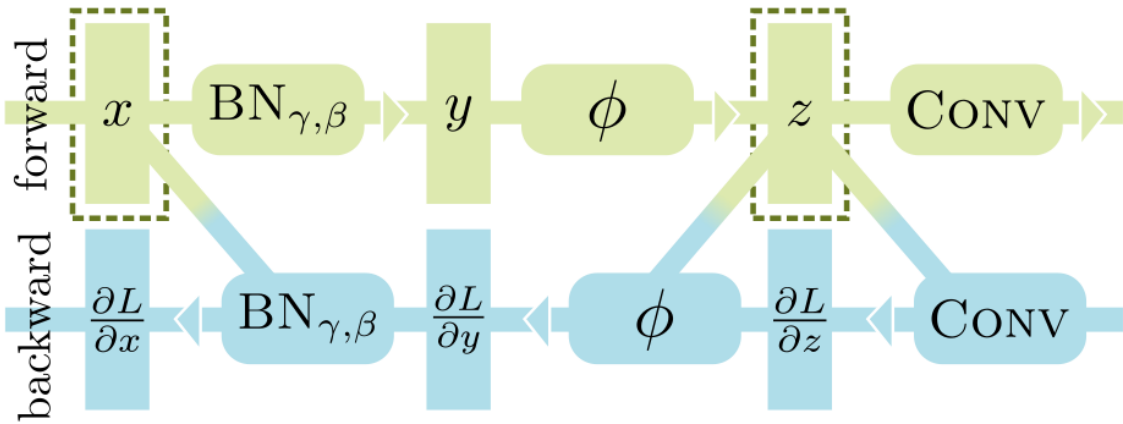


[santurkar18nips]

Arhitekture: BatchNorm (3)

Downsides:

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

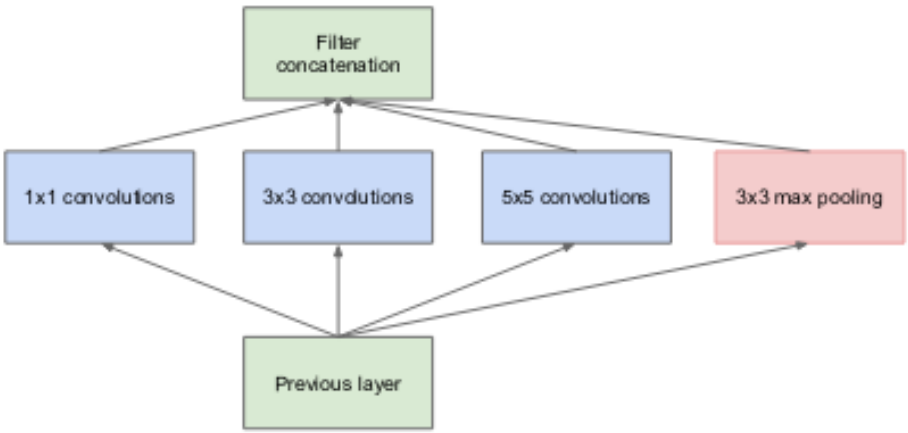


(a) Standard building block (memory-inefficient)

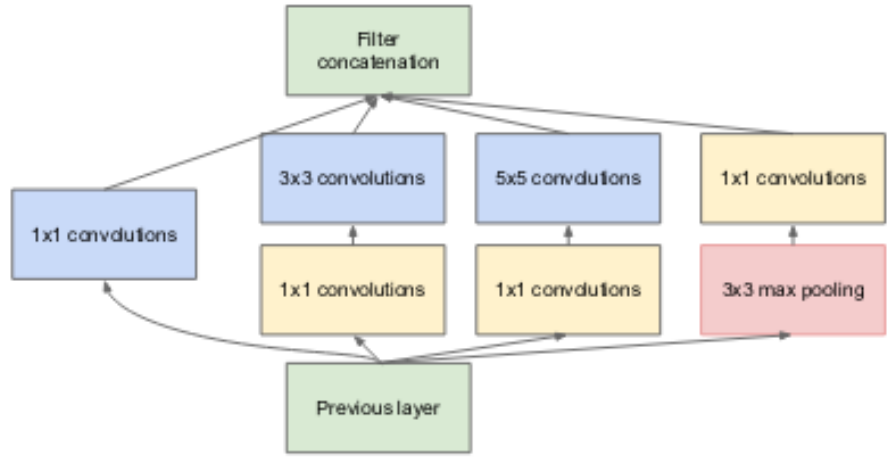
[bulo18cvpr]

Recent research suggests that batchnorm can be replaced with weight standardization and gradient clipping [brock21arxiv].

Arhitekture: inception



(a) Inception module, naïve version



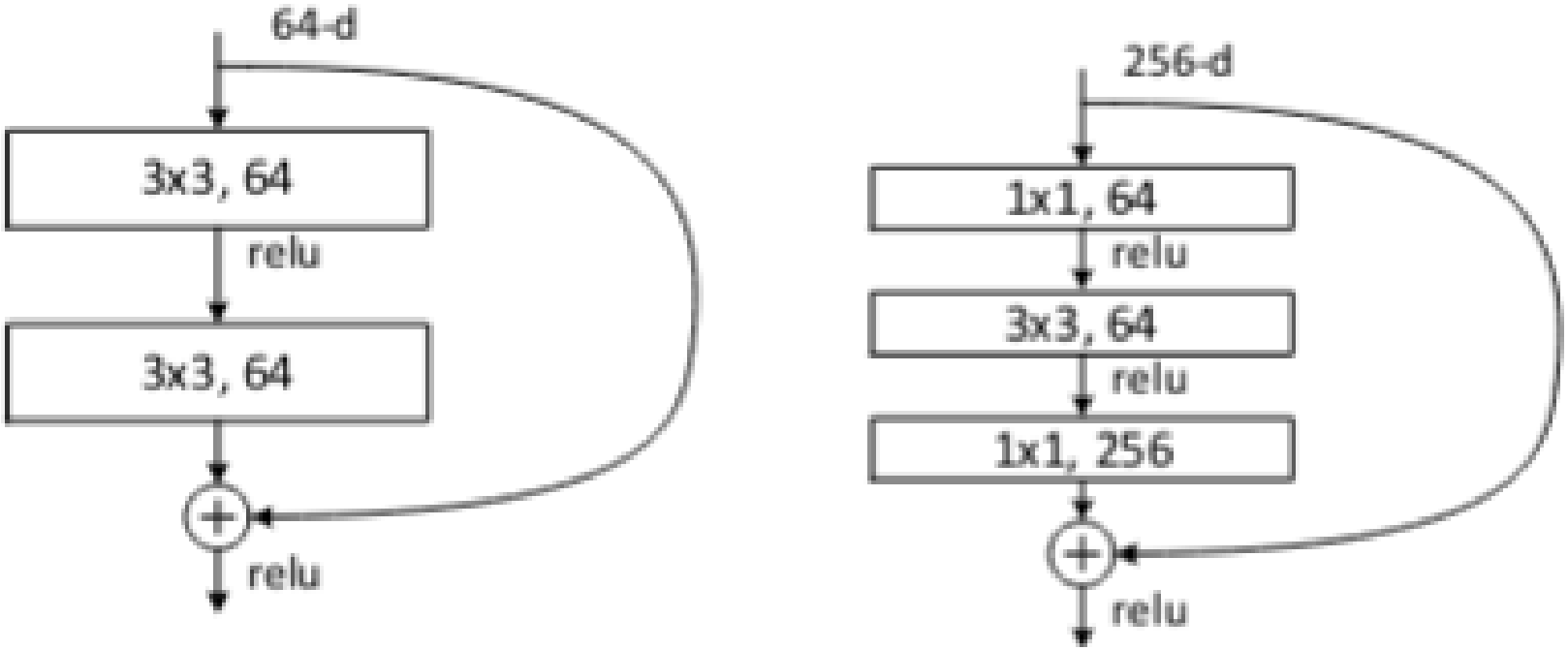
(b) Inception module with dimension reductions

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]

Arhitekture: ResNet - ideja

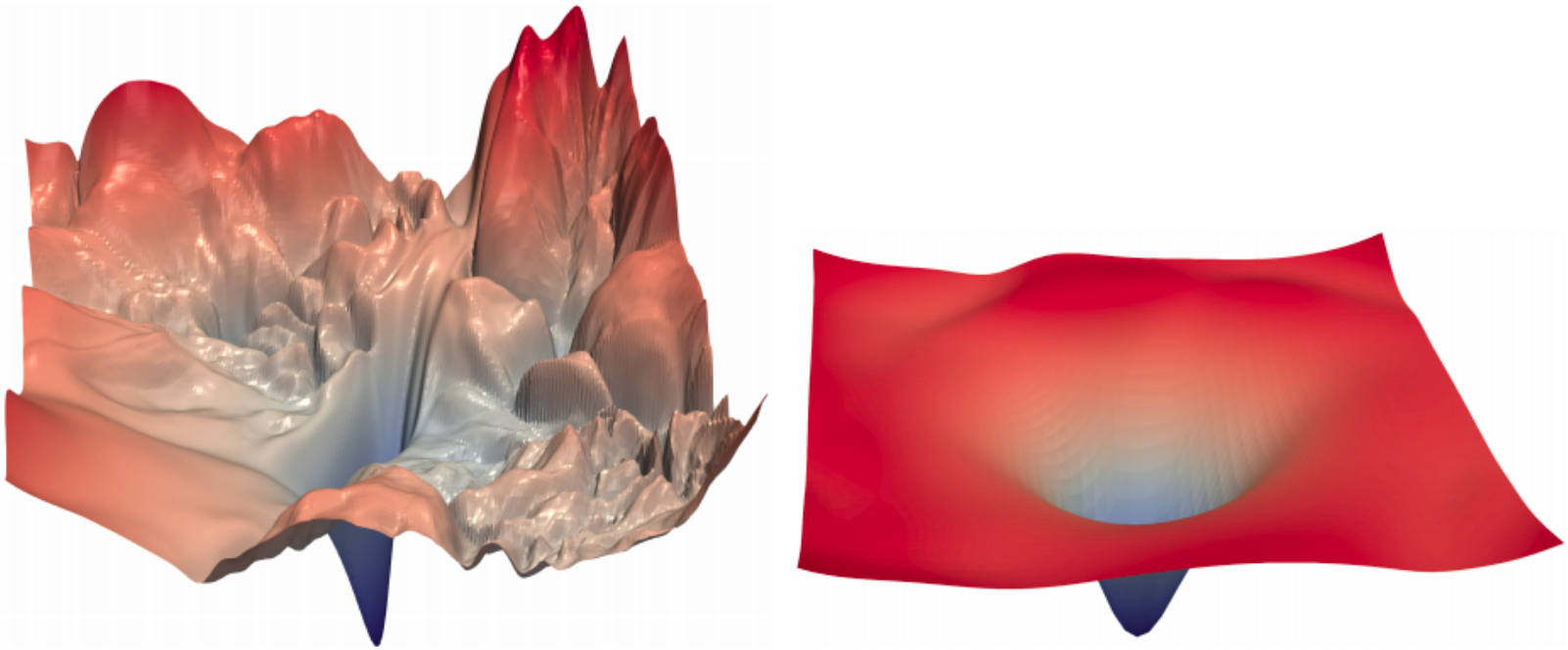
Duboki slojevi aditivno popravljaju približnu predikciju prethodnika:



Duboki sloj modelira aditivnu pogrešku (=rezidual)

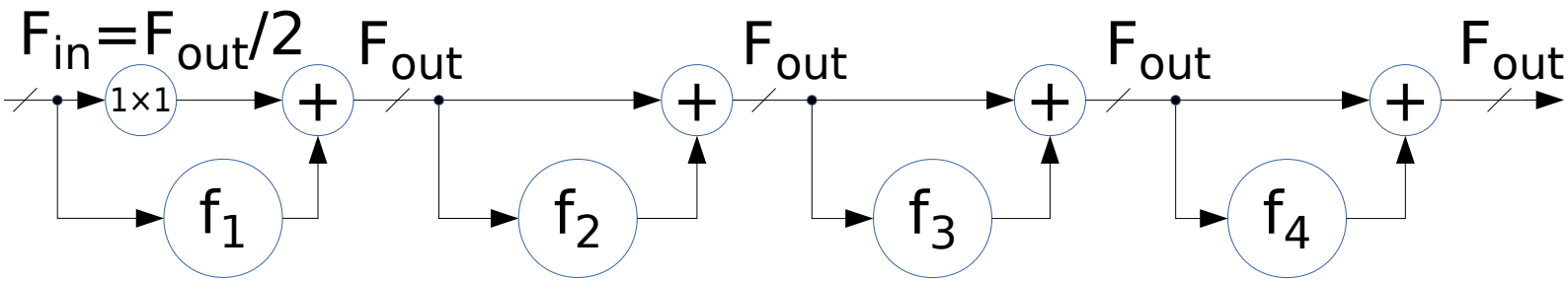
Arhitekture: ResNet - učinak

Rezidualne veze glade funkciju gubitka i poboljšavaju tok gradijenta prema ranim slojevima



(a) without skip connections

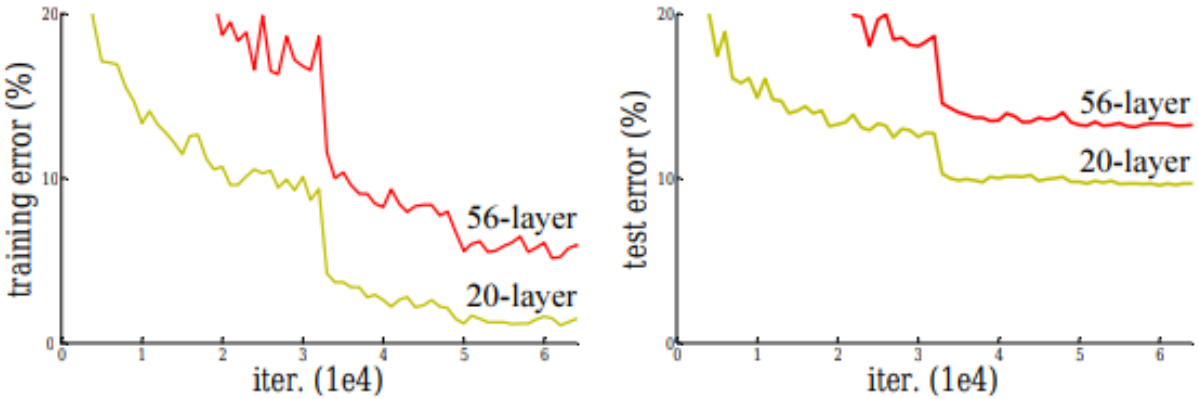
(b) with skip connections



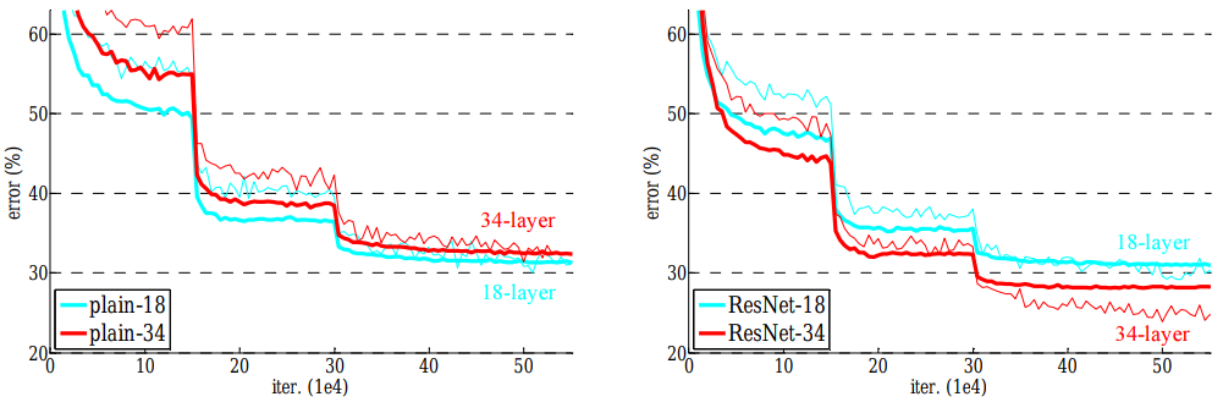
Arhitekture: ResNet - dubina

Slijedne modele možemo učiti do dubine cca 20

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



Rezidualni modeli nemaju problema s dubinom:



[he15iclr,he16eccv]

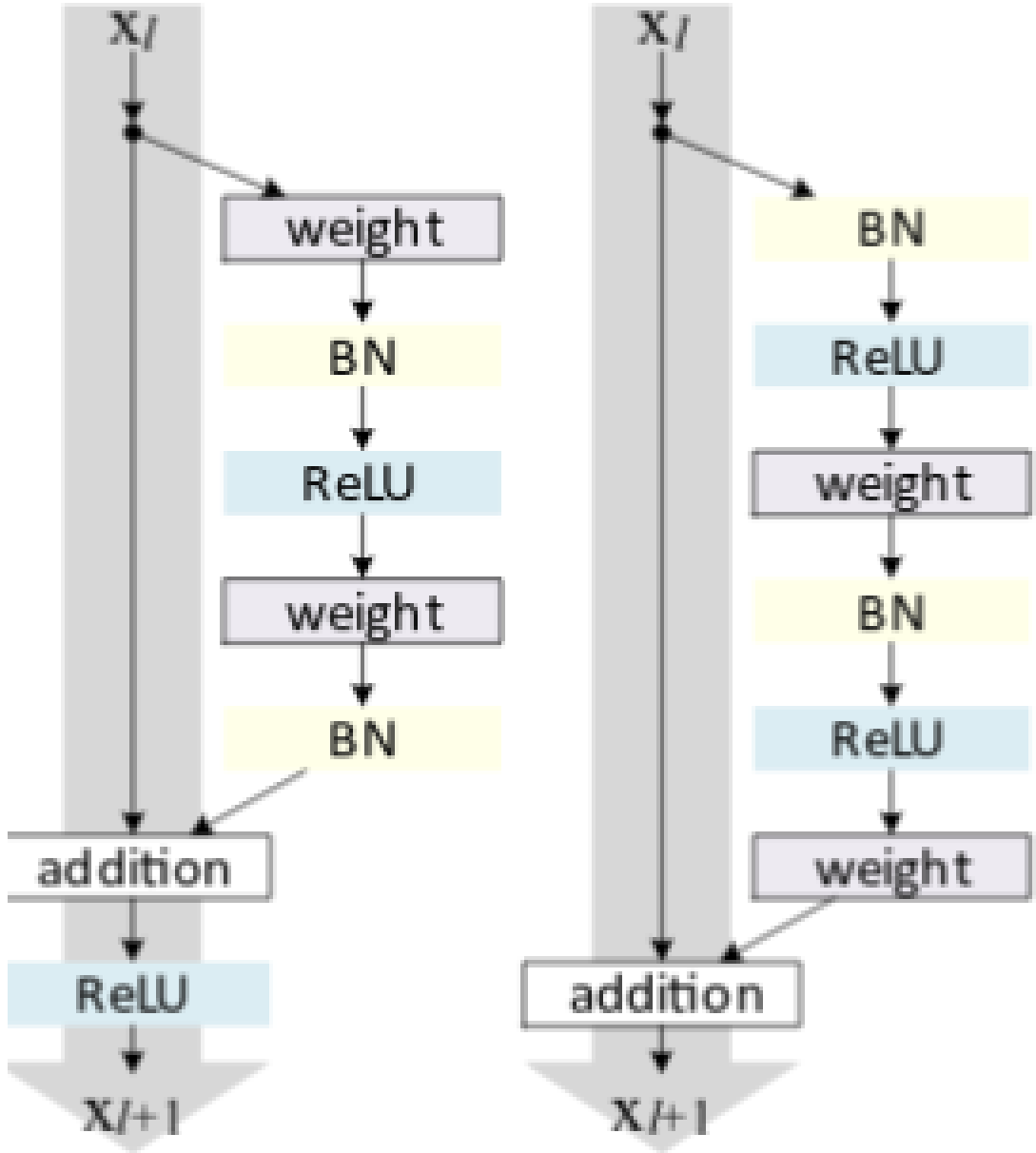
Arhitekture: ResNet - struktura

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)



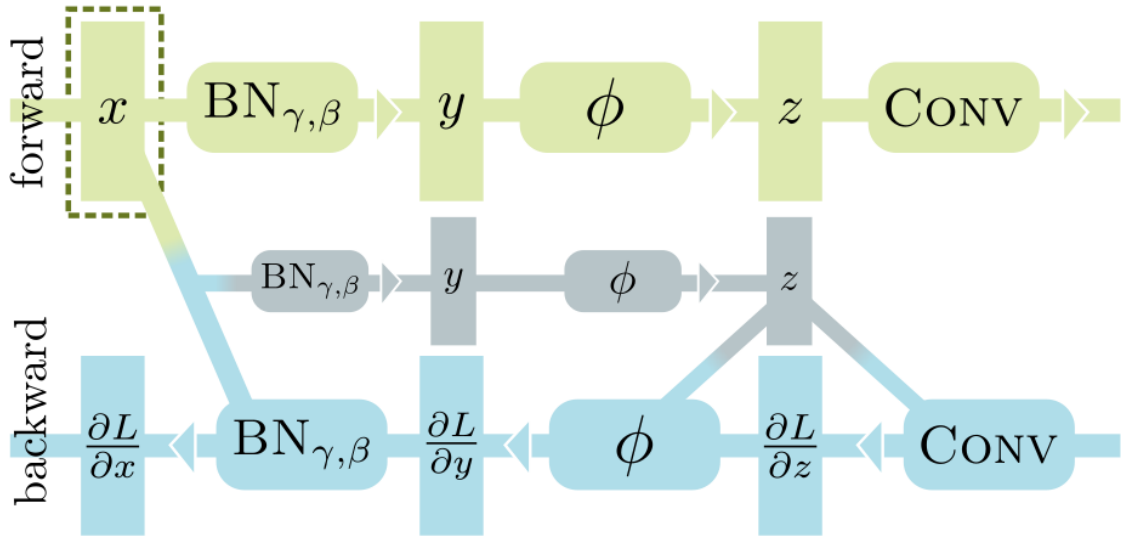
[he15cvpr,he16eccv]

Arhitekture: ResNet - modeli

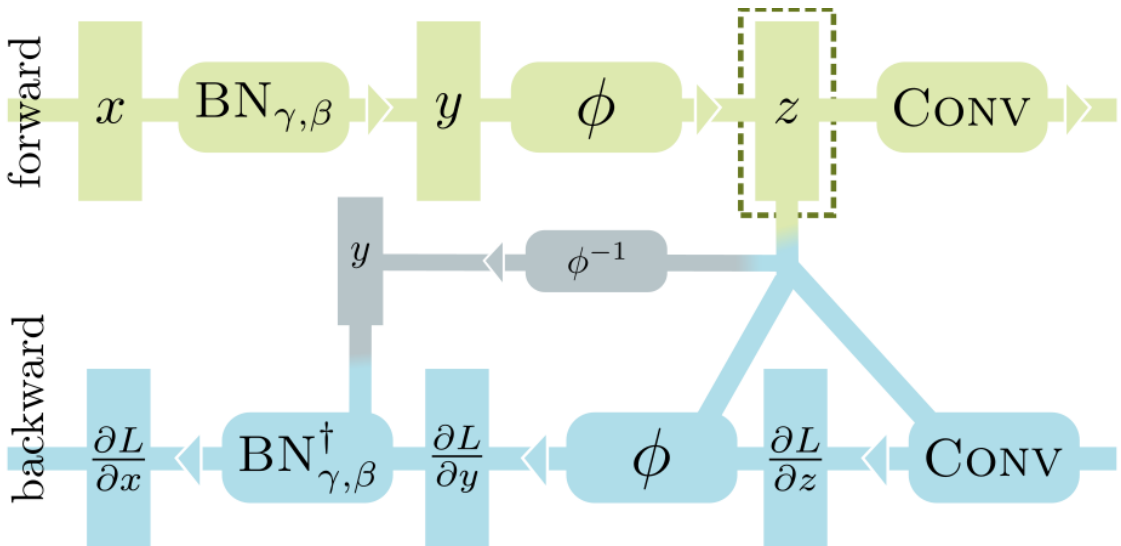
layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

[he15iclr,he16eccv]

Arhitekture: ResNet - BatchNorm



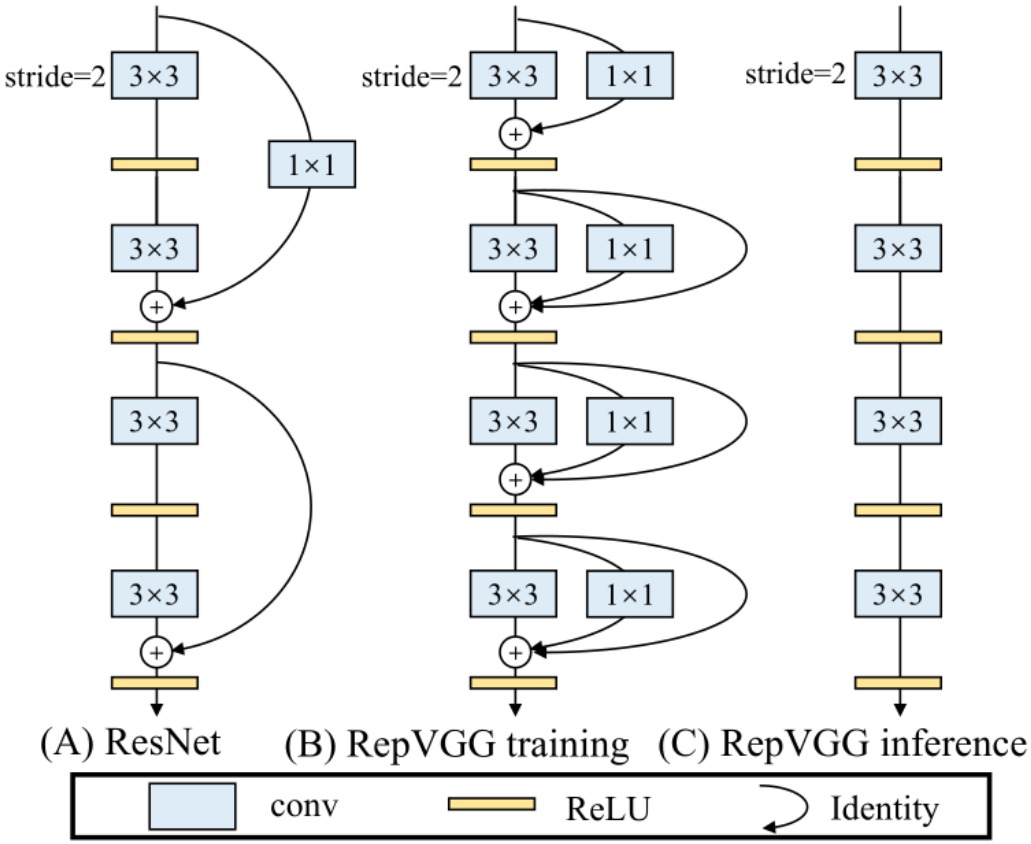
(b) Checkpointing [4, 19]



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

Arhitekture: ResNet - RepVGG

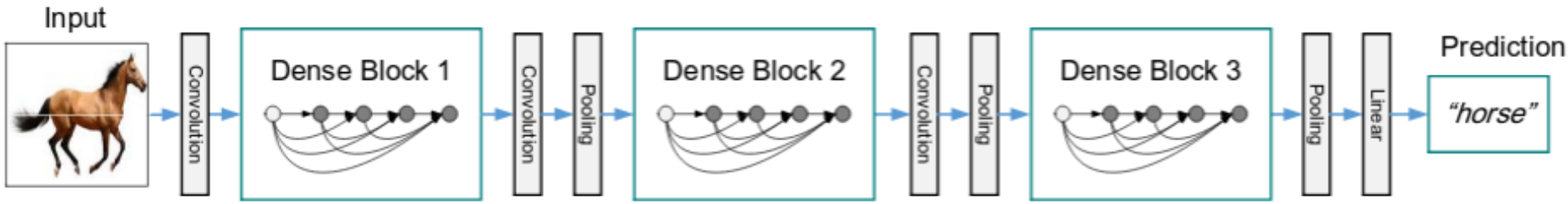
Rezidualne veze mogu se simulirati strukturnom reparametrizacijom slijednog modela:

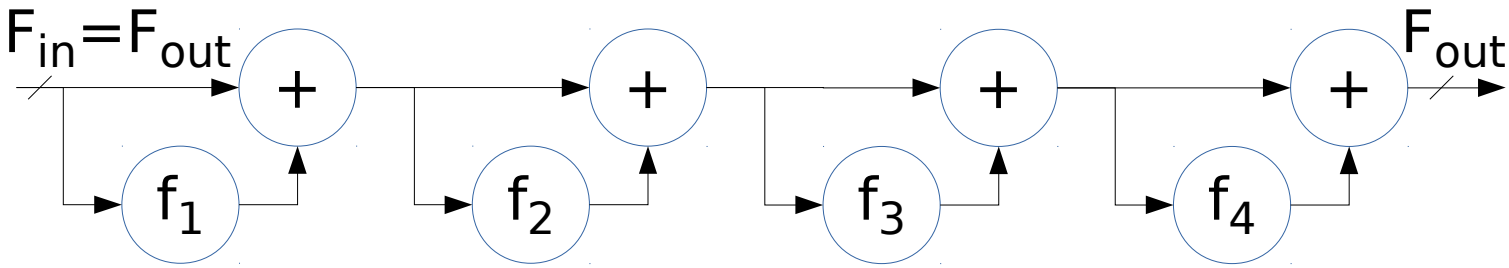


[ding21cvpr]

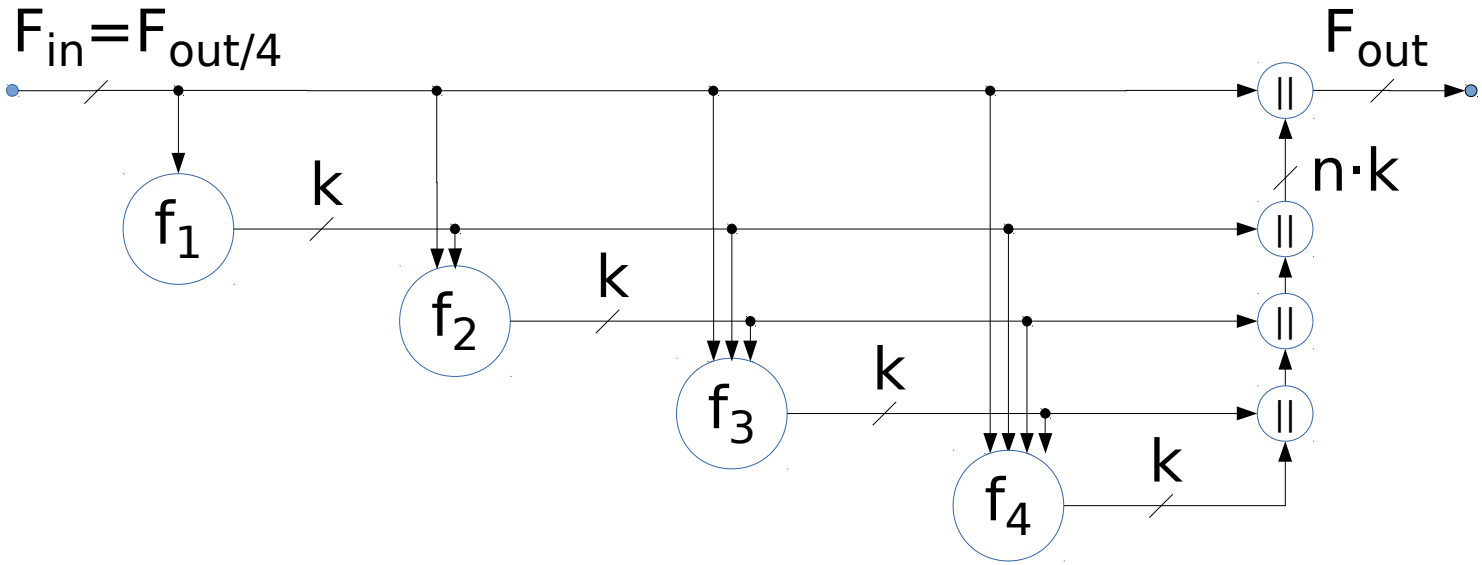
Tijekom zaključivanja, tri paralelne grane mogu se izvesti jednom 3x3 konvolucijom.

Arhitekture: DenseNet - ideja





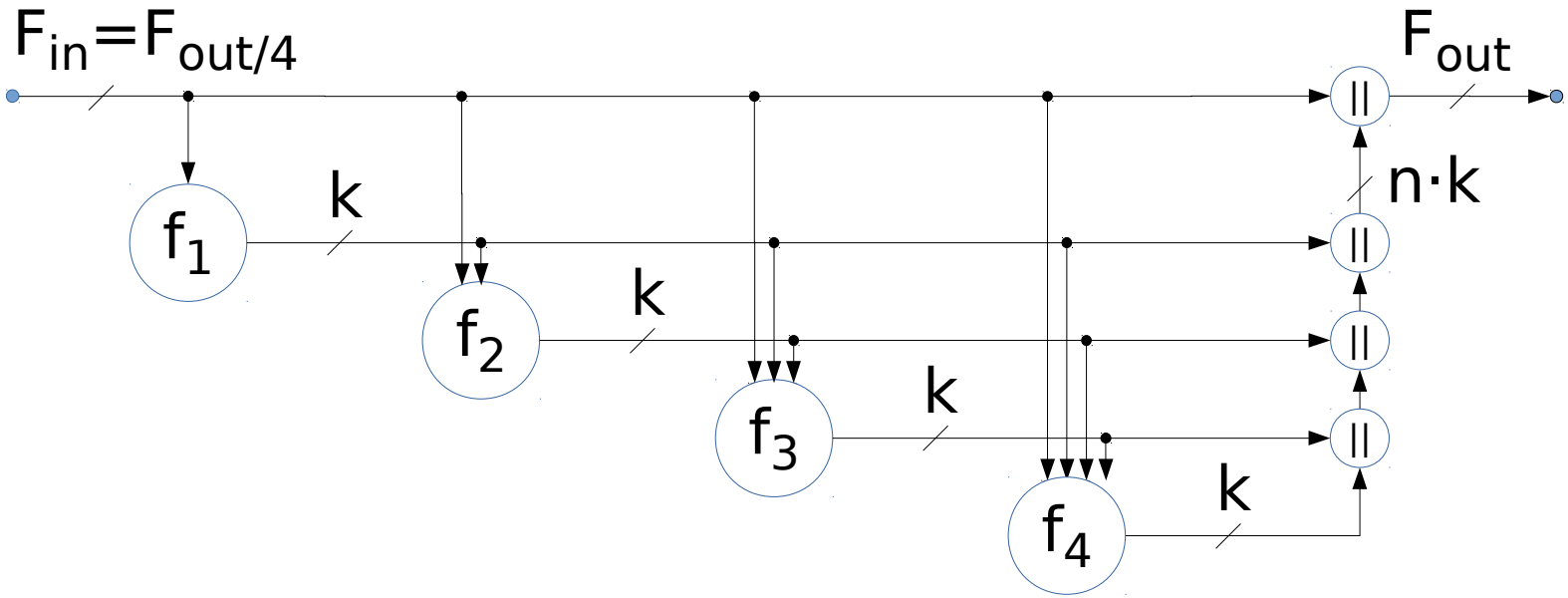
ResNet

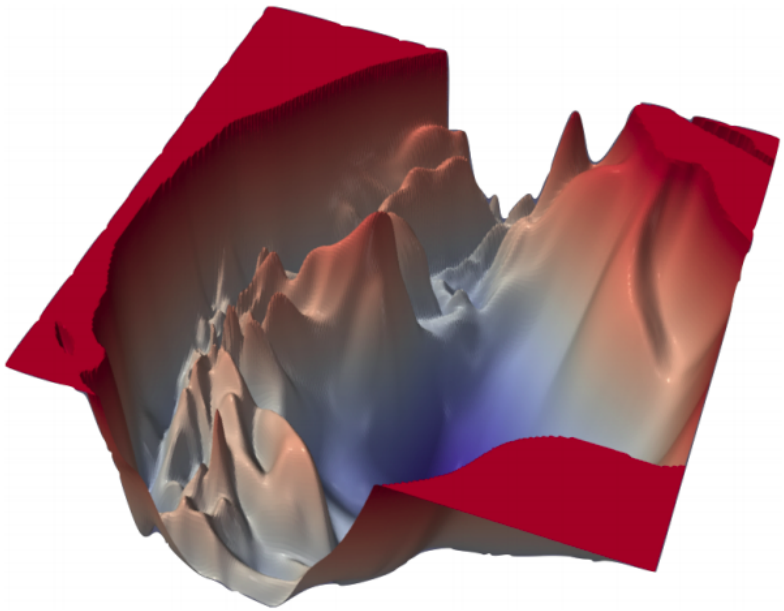


[huang17cvpr]

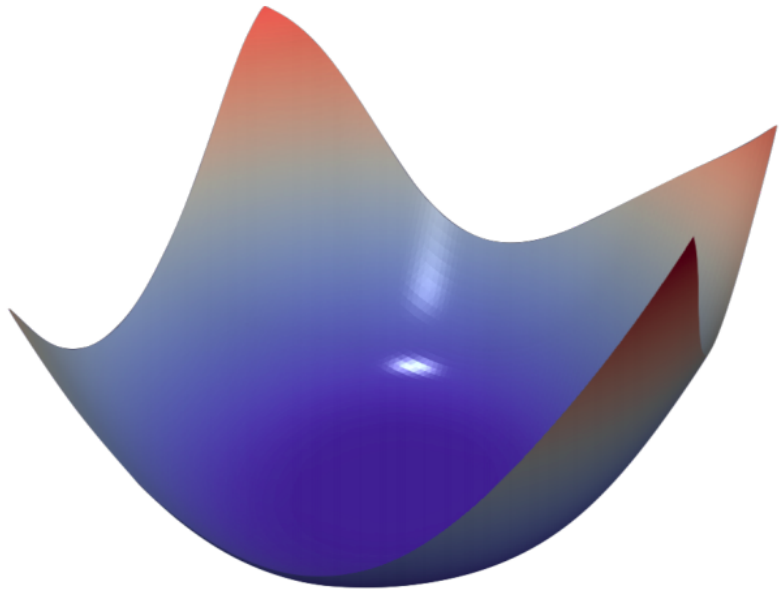
DenseNet

Arhitekture: DenseNet - učinak





(a) ResNet-110, no skip connections



(b) DenseNet, 121 layers

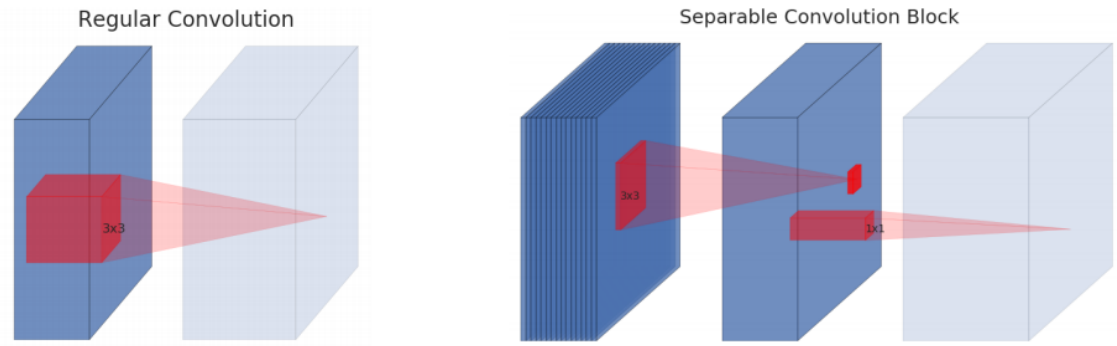
[li18nips]

Arhitekture: DenseNet - modeli

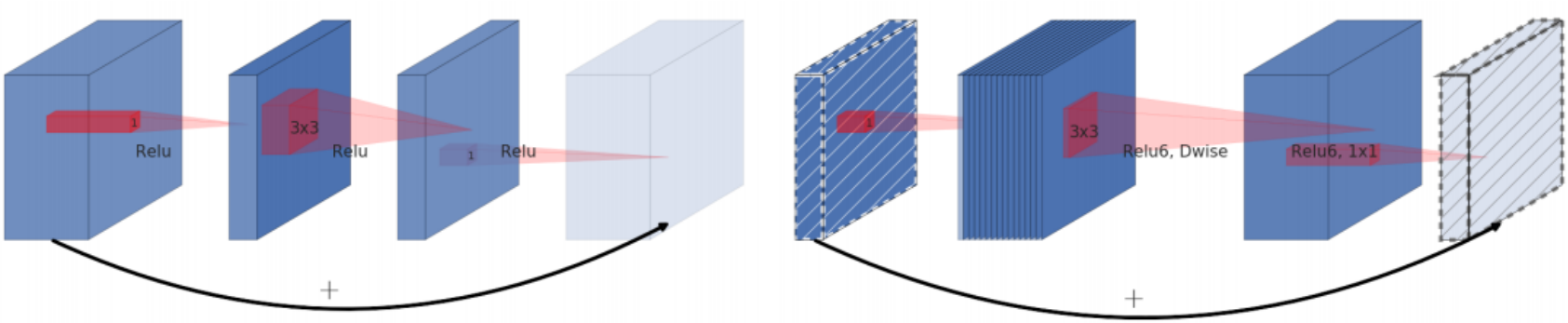
Layers	Output Size	DenseNet-121	DenseNet-169	DenseNet-201	DenseNet-264
Convolution	112 × 112	7 × 7 conv, stride 2			
Pooling	56 × 56	3 × 3 max pool, stride 2			
Dense Block (1)	56 × 56	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times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

[huang17cvpr]

Arhitekture: MobileNet v2



ideja 1: dubinski separirana konvolucija



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

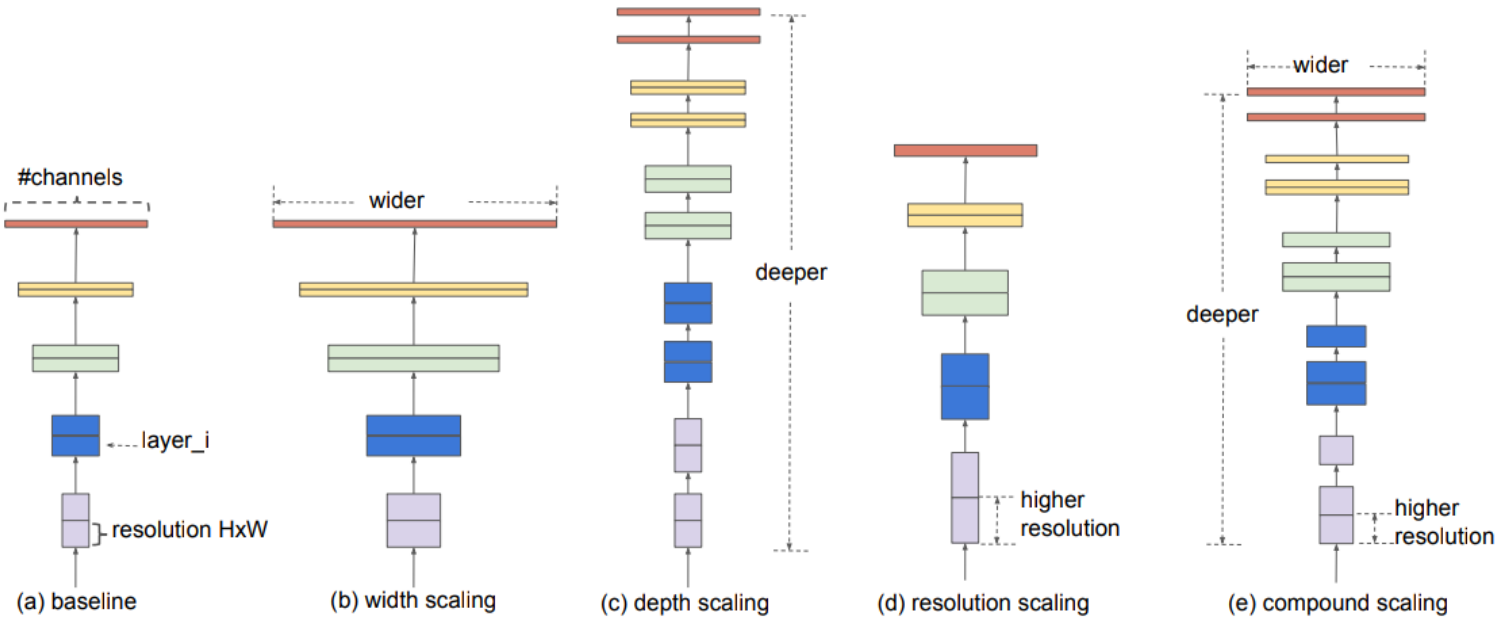
[sandler18cvpr]

Arhitekture: MobileNet v2

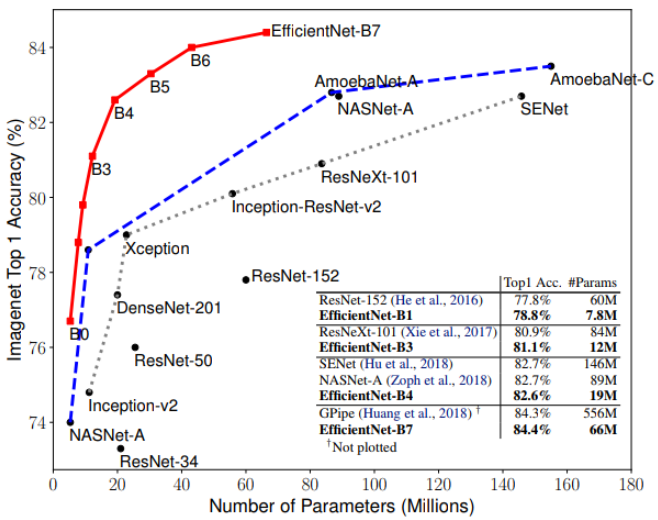
Input	Operator	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck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

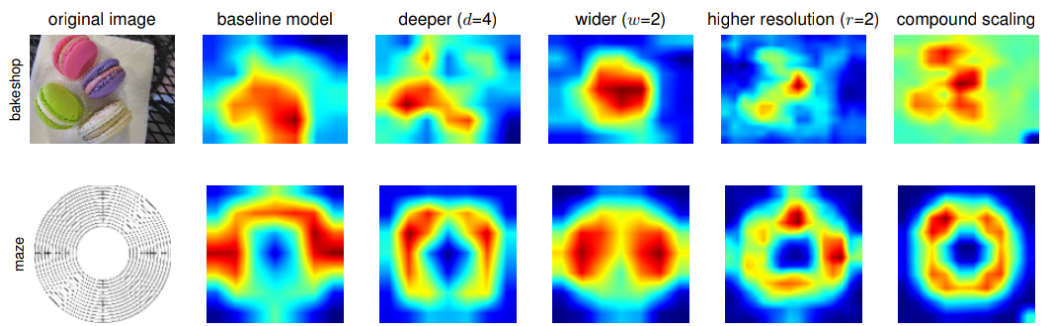
[sandler18cvpr]

Arhitekture: EfficientNet

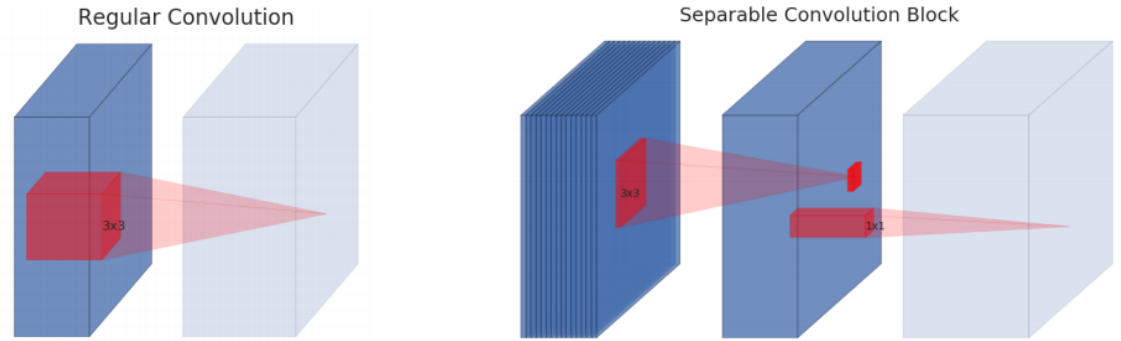


ideja: dosljedno skaliranje po tri osi

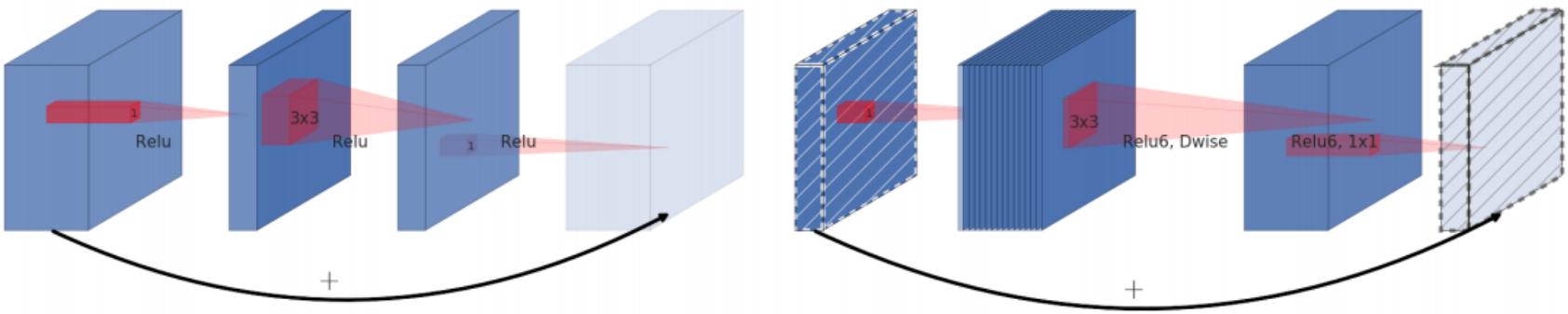




Tehnike: less parameters



idea 1: depthwise separable convolution



regular residual unit

dws separable residual unit

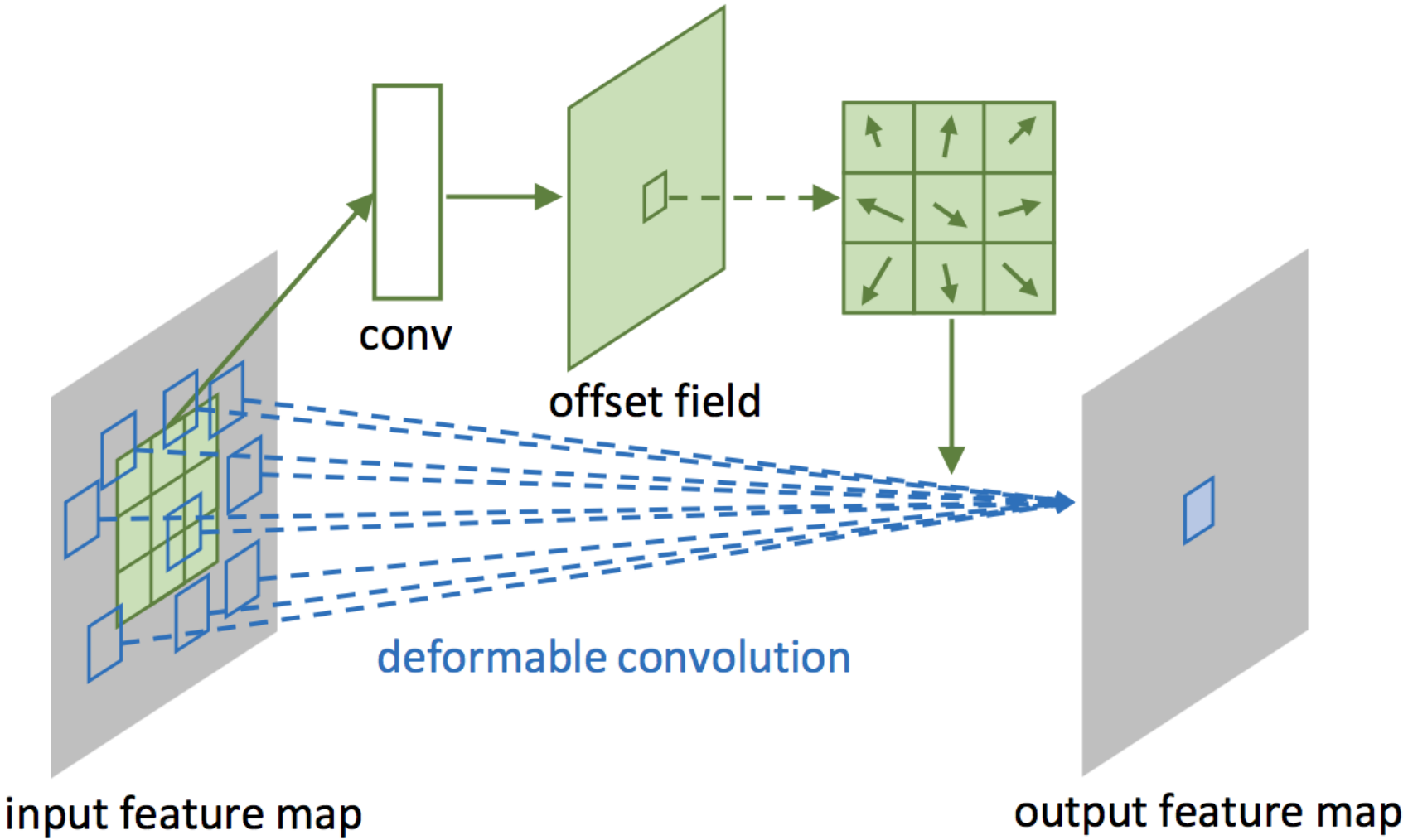
idea 2: depthwise separable convolution on "inflated" representation

"inverted residual": zbrajaju se tenzori sa smanjenim brojem kanala

[sandler18cvpr]

Tehnike: more flexibility

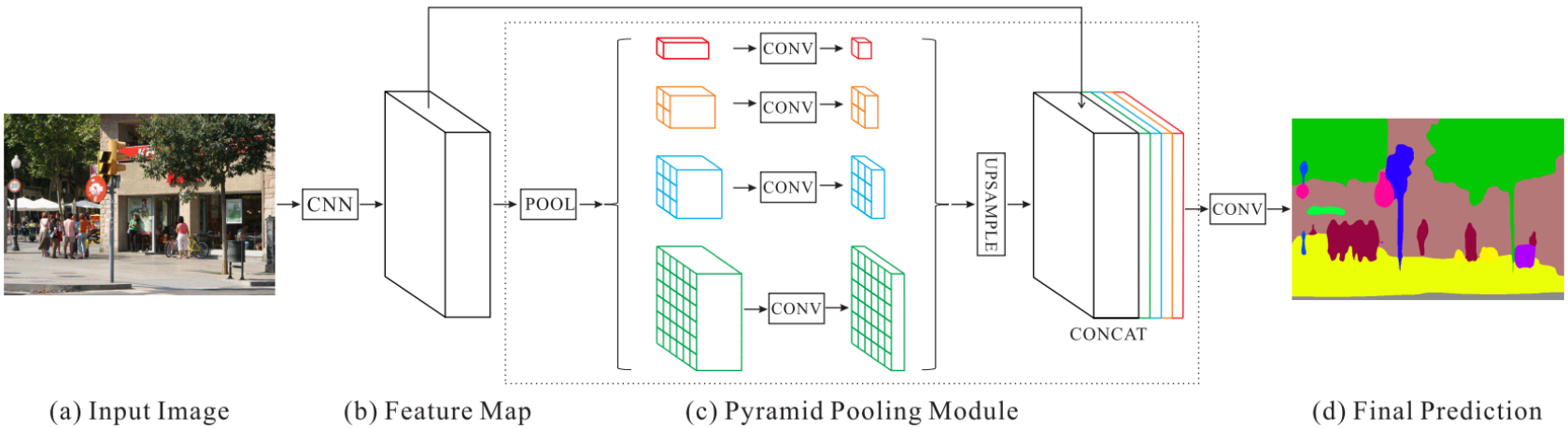
Deformable convolutions:



[dai17iccv]

Tehnike: increased receptive field

Spatial pyramid pooling (SPP)



[zhao17cvpr]

Idea: provide wide contextual information to subsequent convolutions

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 untrašnji 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 gradijenta gubitka obzirom na sliku možemo lokalizirati objekte (iako se sada čini da to nije najbolja metoda)
- ovaj formalizam generalizira dekonvolucijski prikaz Zeilera i Fergusa.

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.

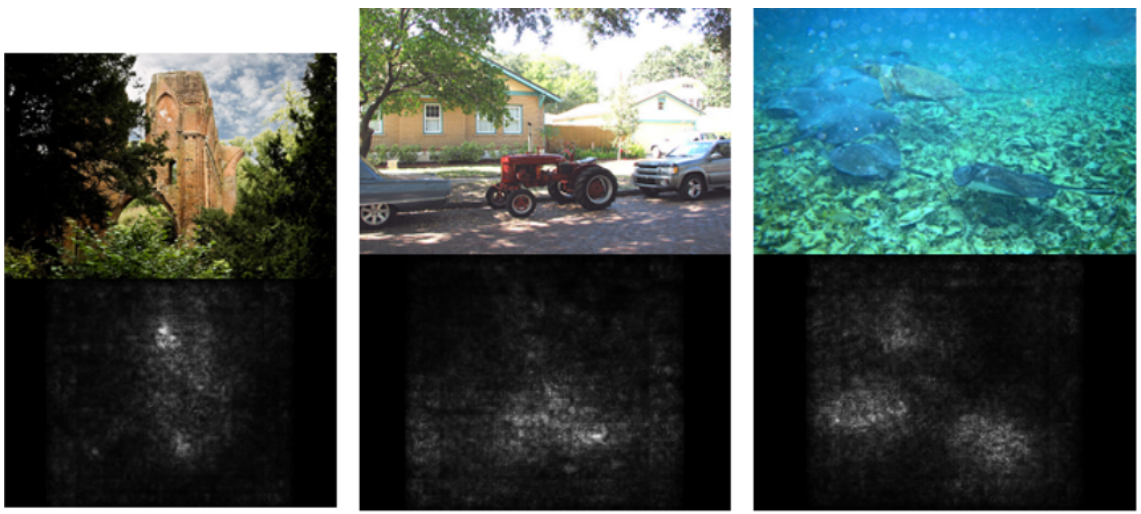
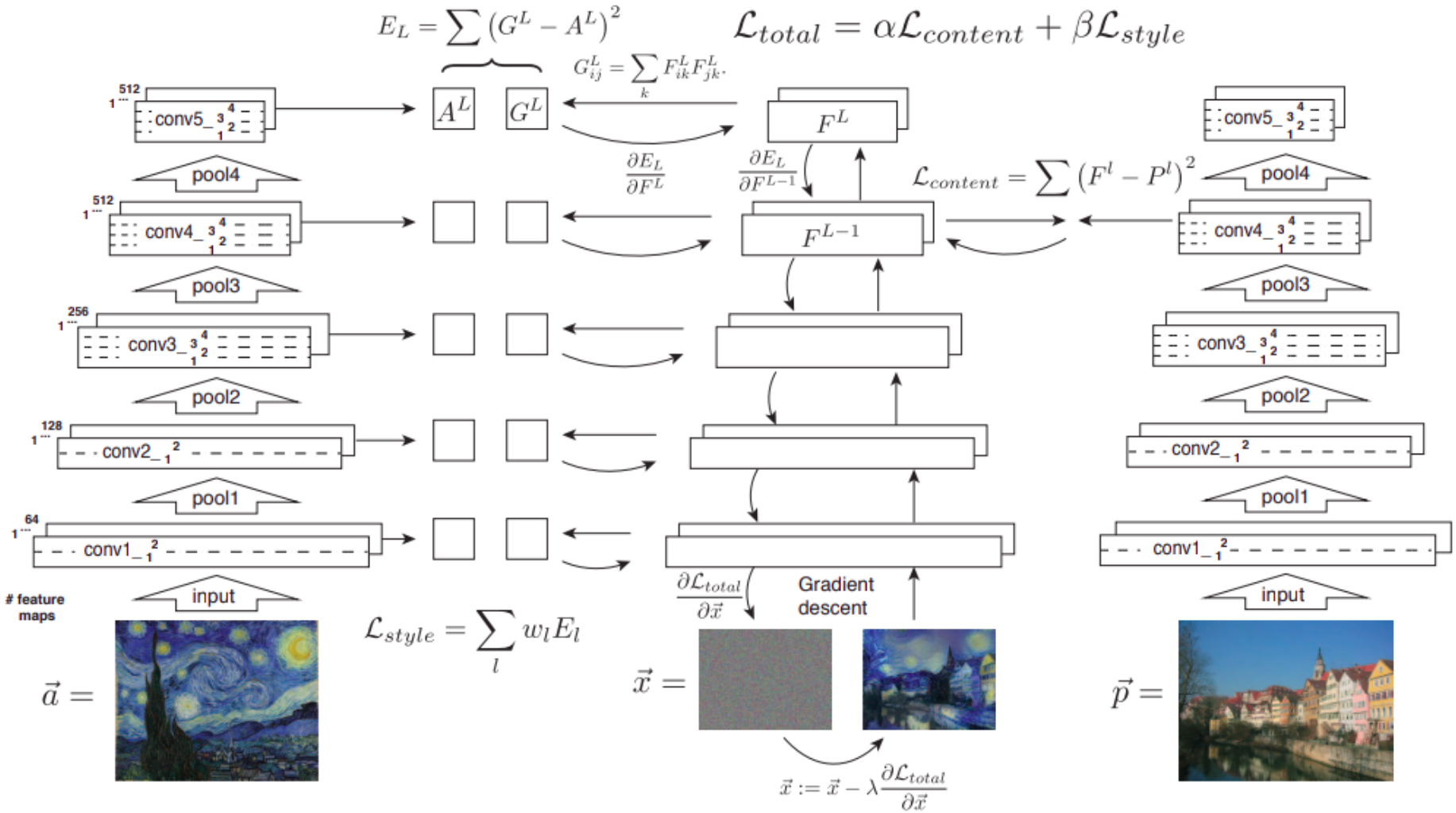


Figure 2: **Image-specific class saliency maps for the top-1 predicted class in ILSVRC-2013 test images.** The maps were extracted using a single back-propagation pass through a classification ConvNet. No additional annotation (except for the image labels) was used in training.

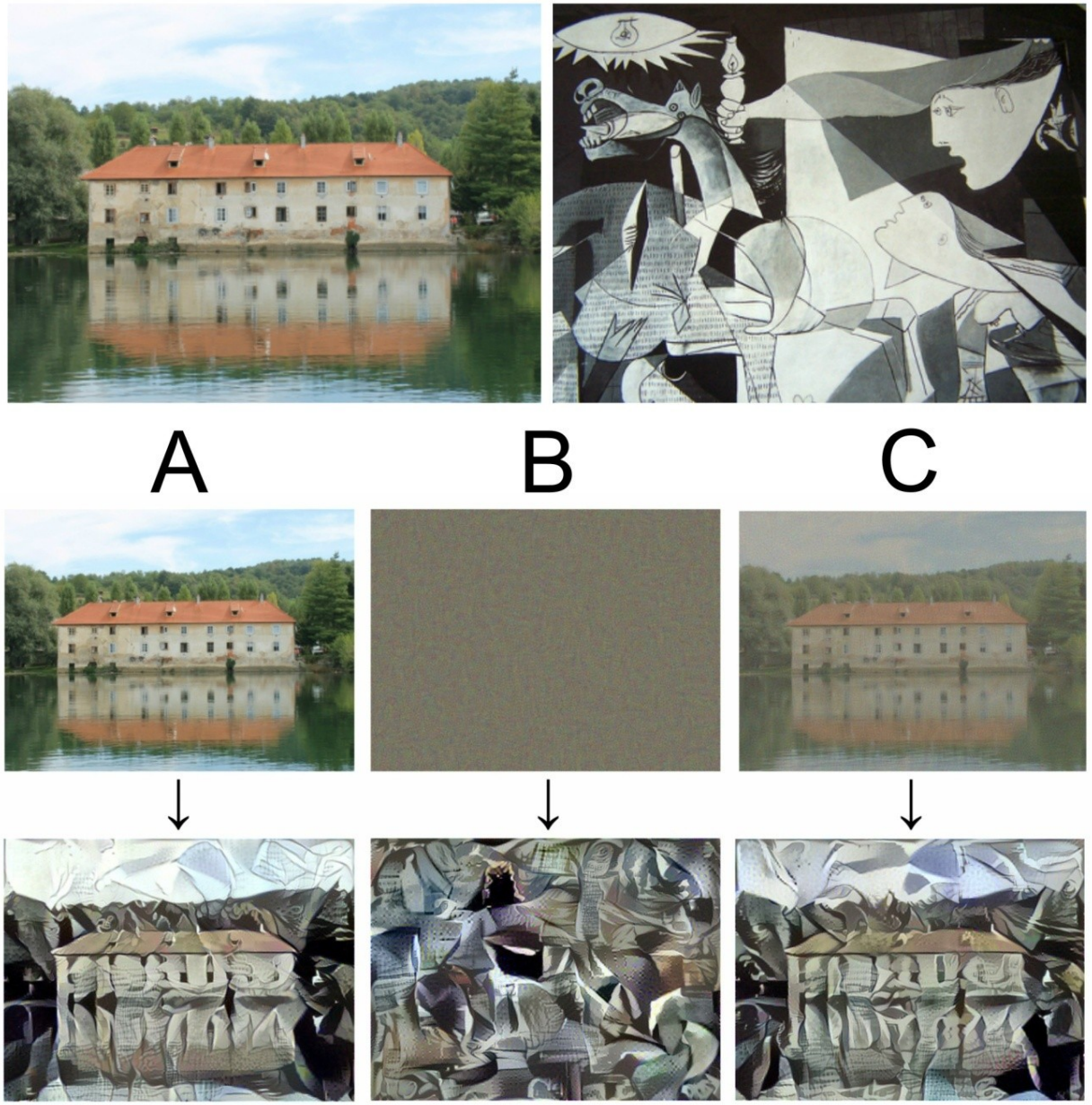
[simonyan14iclr]

Razumijevanje: Prijenos stila



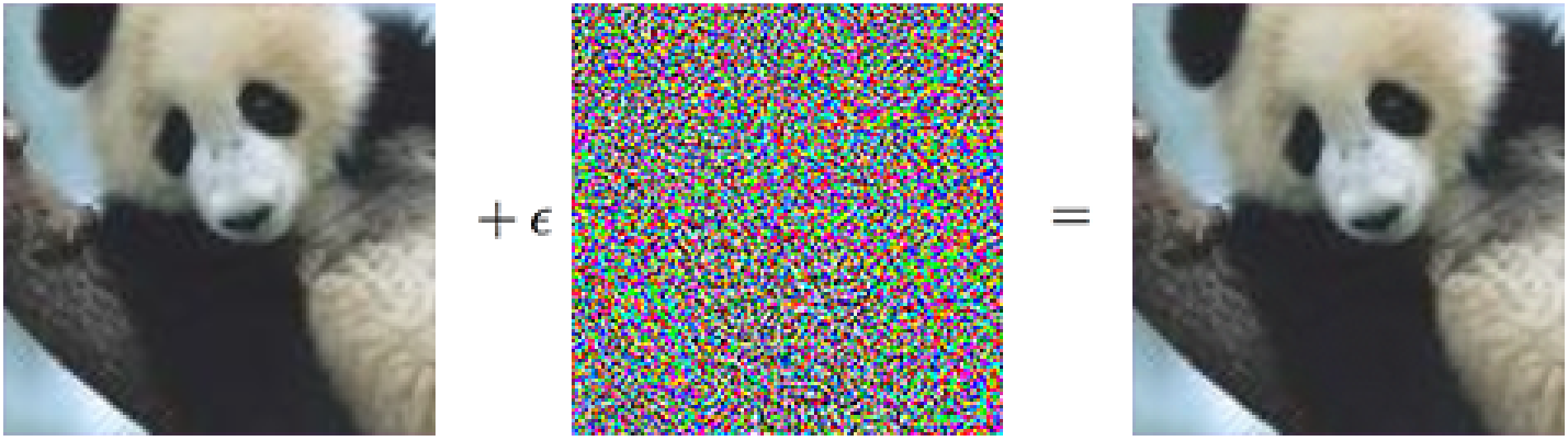
[gatys16cvpr]

Razumijevanje: Prijenos stila (2)



[ivkovic17bs]

Razumijevanje: neprijateljski primjeri



[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

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

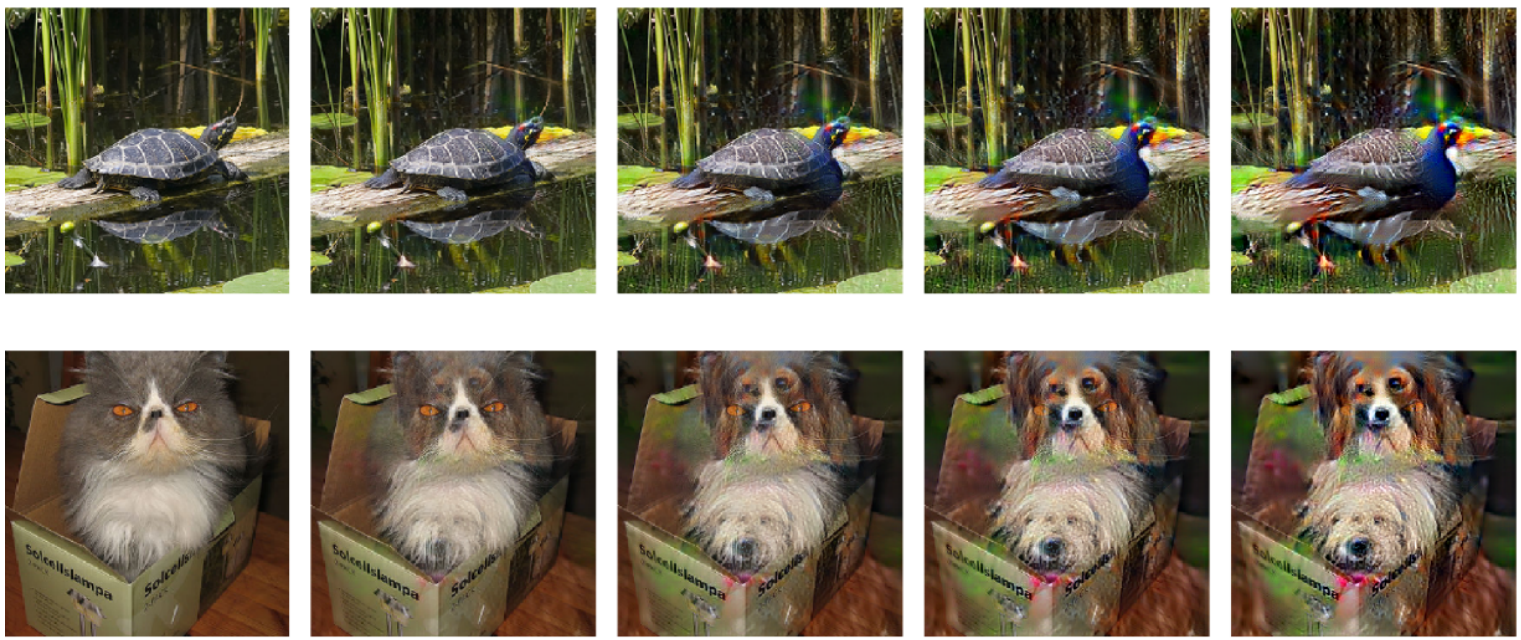
Adversarial training has been introduced years ago as a regularization technique [szegedy14iclr]

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]



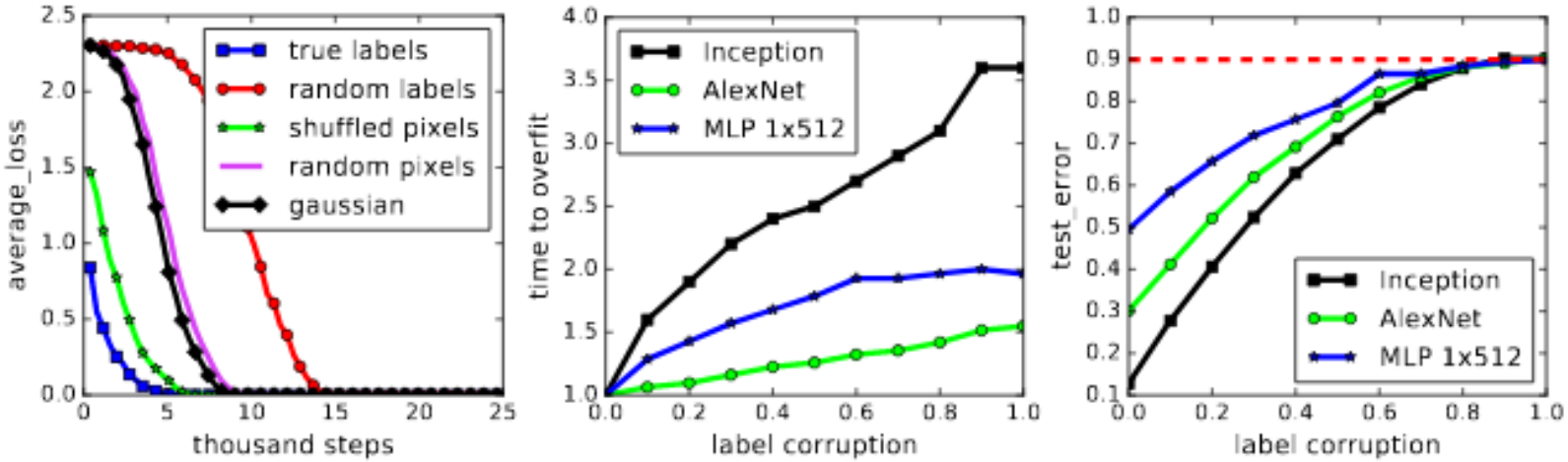
[tsipras18arxiv]

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

Downside 2: much more computationally intensive than classic training

Razumijevanje: prenaučenost?

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



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

[zhang17iclr]

In simple words, the model is able to memorize the entire training data

Yet, deep models generalize well when trained on correct labels

A theory to explain this behaviour is missing.

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

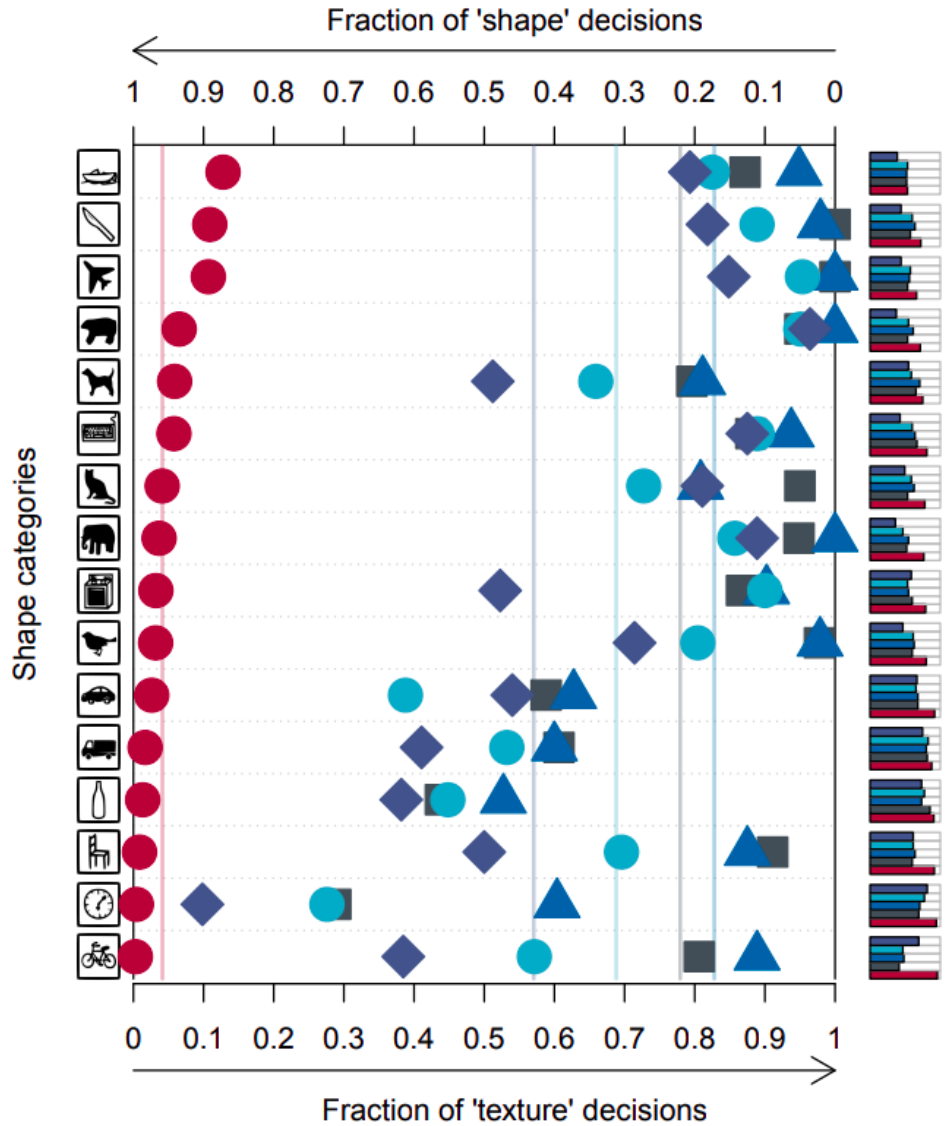
[geirhos19iclr]

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

Razumijevanje: tekstura (2)

Kod slika s konfliktnim indicijama ljudi preferiraju oblik a duboki 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 duboke 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]

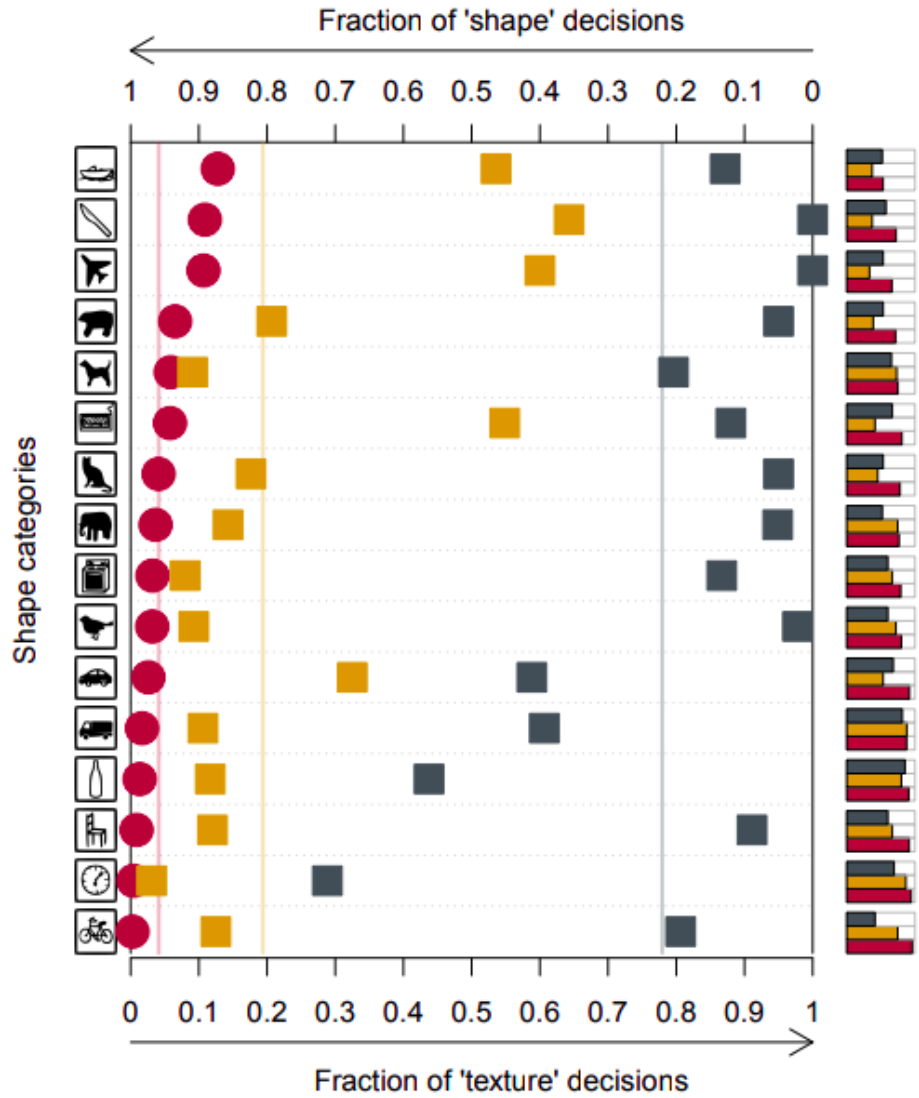
Razumijevanje: tekstura (4)

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

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

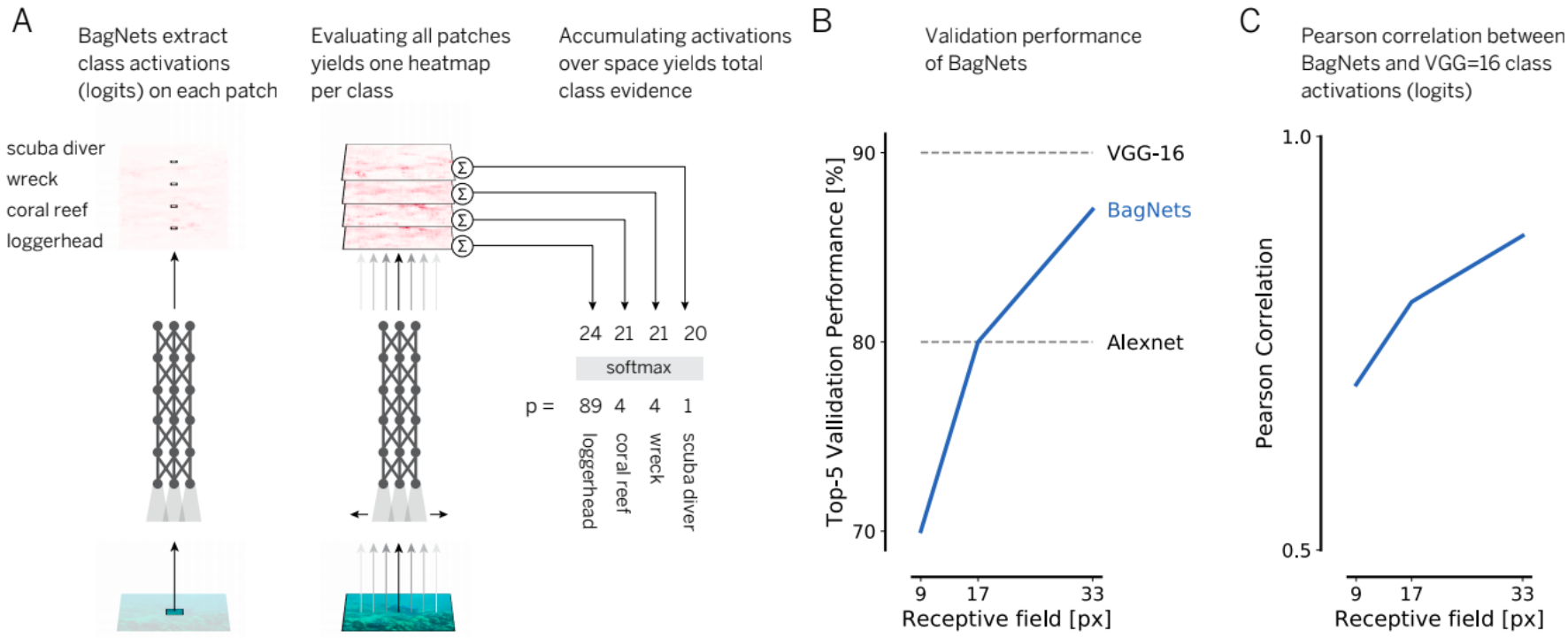
Zaključak: duboki modeli skloni su oslanjati se na teksturu.

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



[geirhos19iclr]

Razumijevanje: tekstura (5)



[brendel19iclr]

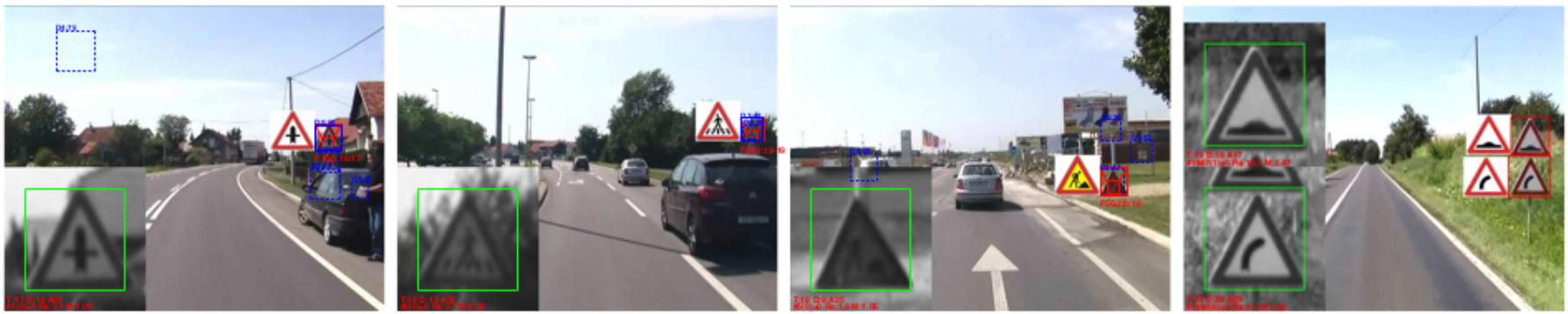
- Ovakvo učenje sprječava model da uzme u obzir širi kontekst
- međutim, validacija je slična kao i u standardnom slučaju
- ovo ponovo sugerira da se duboki modeli pretjerano oslanjaju na lokalnu teksturu

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
- neke demonstracije:
  - semantička segmentacija

Objekti: pomično okno

Klasična detekcija u pomičnom oknu zahtijeva relativno jednostavne značajke i klasifikatore



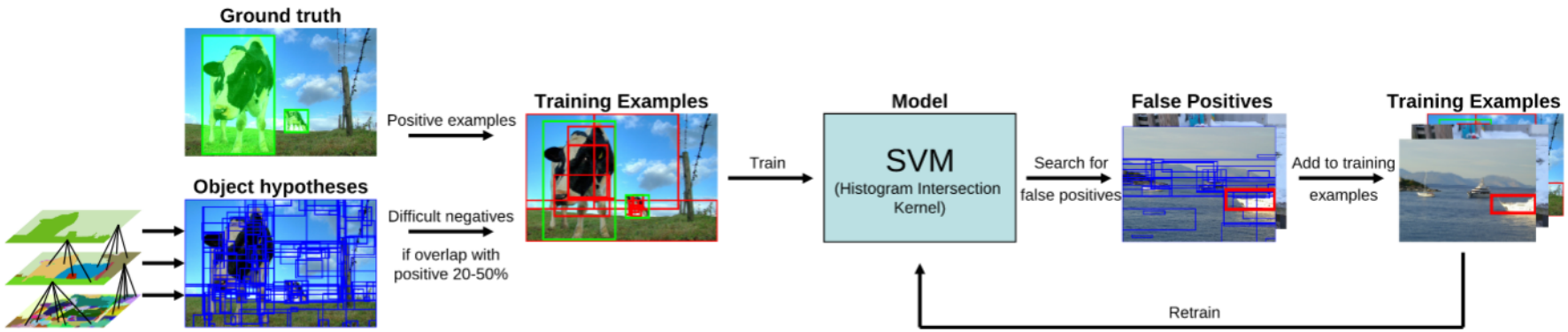
Takav pristup nije prikladan za istovremenu detekciju više razreda

- npr: stol, tanjuri, pribor, mačke, kameleon, kotači, automobili

Objekti: dvoprolazni pristupi

Zato su početkom 2010-ih godina popularizirani dvoprolazni pristupi

1. prvo pronaći kandidate ($\sim$1000) općenitim brzim postupkom
2. testirati i klasificirati kandidate teškom artiljerijom (BoW)



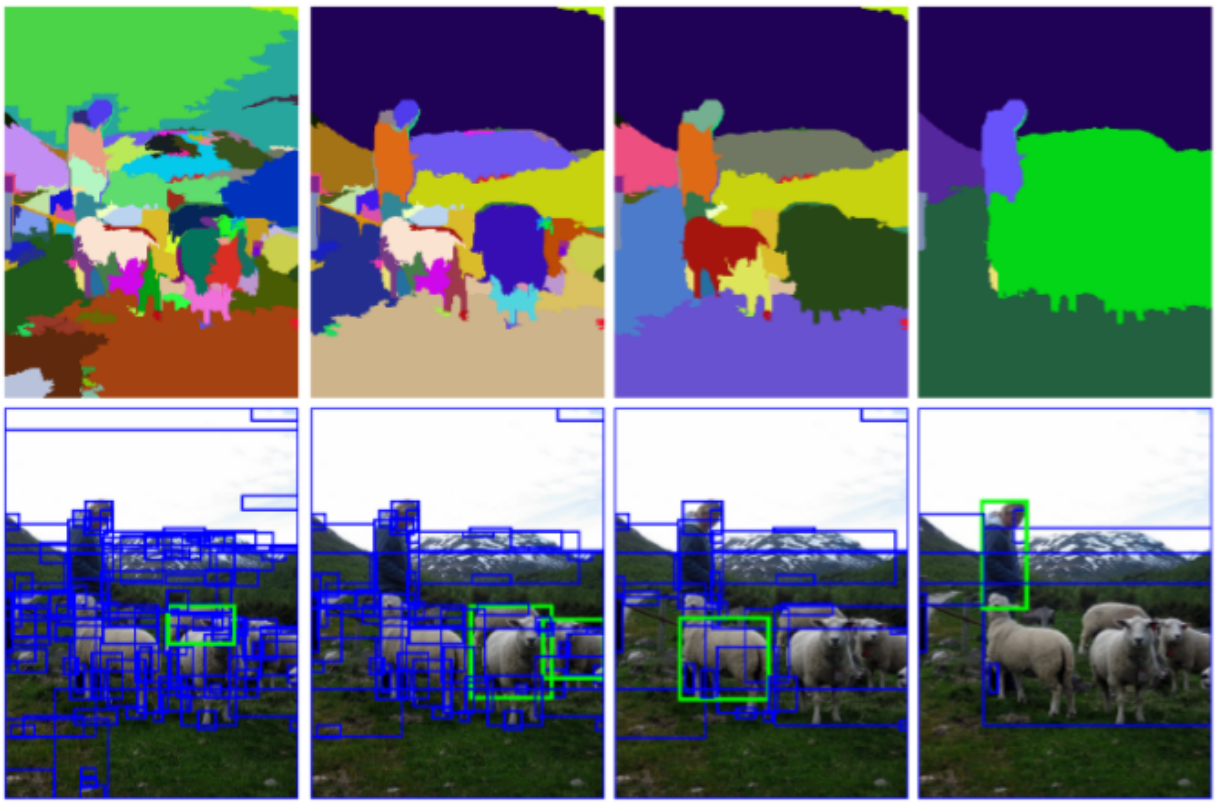
[uijlings13ijcv]

Ovo možemo promatrati i kao najnapredniji klasični pristup i kao prijelazni oblik prema dubokim modelima

Objekti: selektivno pretraživanje

Ideja [uijlings13ijcv]: detektirati kandidate primjenom i) segmentacije u superpiksele te ii) ručno dizajniranih strategija grupiranja

- kriteriji grupiranja: boja, tekstura, veličina, nadopunjavanje

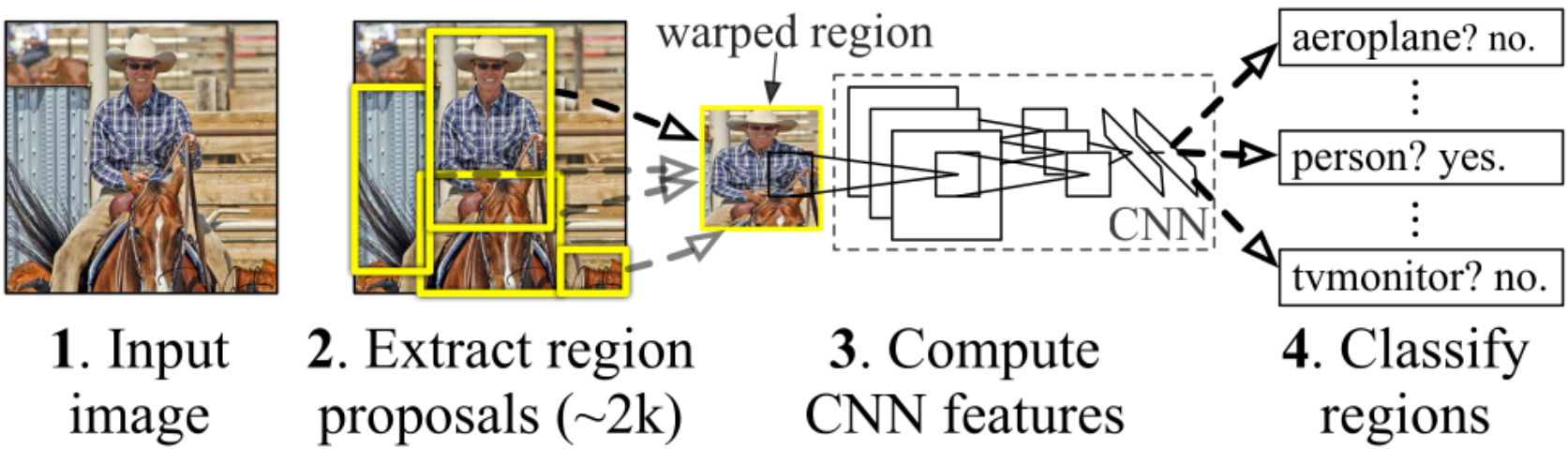


- točnost na VOC'12 test: 35.0% mAP@0.5

Objekti: R-CNN

Glavna ideja: features matter

- zamijeniti colourSIFT-BoW predtreniranim dubokim modelom
- ugoditi duboki model za klasifikaciju u C+1 razred na grupama od 32 pozitivnih i 96 negativnih kandidata
- konačnu klasifikaciju provesti SVM-om (validacija: +4pp mAP@0.5)
- VOC'12 test: 53.3% mAP@0.5

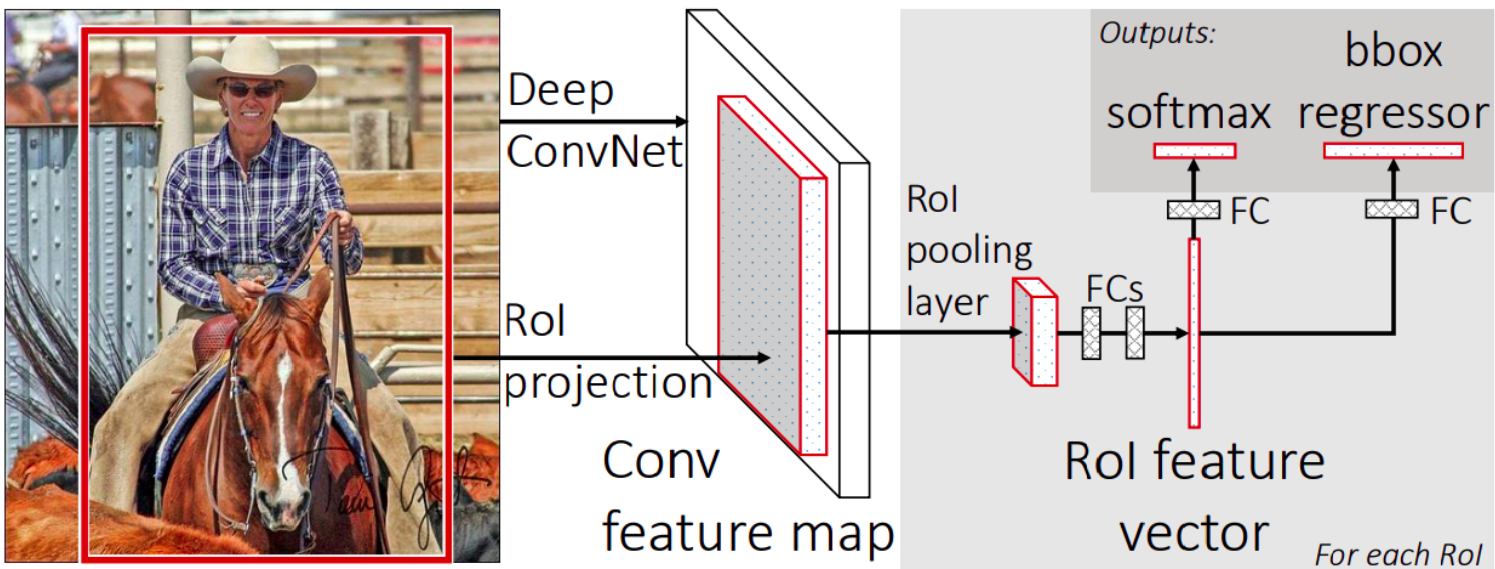


[girshick14cvpr]

Objekti: Fast R-CNN

Glavna ideja: učenje s kraja na kraj, kandidate još uvijek generira spori klasični postupak [uijlings13ijcv]

- provući sliku kroz duboki model, izlučiti značajke h/32×w/32×D
  - izraziti regiju svakog kandidata deskriptorom 7×7×D (ROI pool)
  - svaki element ROI poola je kopija odgovarajuće značajke
  - nema interpolacije, backprop je jednostavan jer je okno fiksno
- klasifikaciju i popravljanje okvira provodi zasebno naučeni model
  - VOC'12 test: 66% mAP@0.5; COCO test-dev: 19.7% mAP COCO

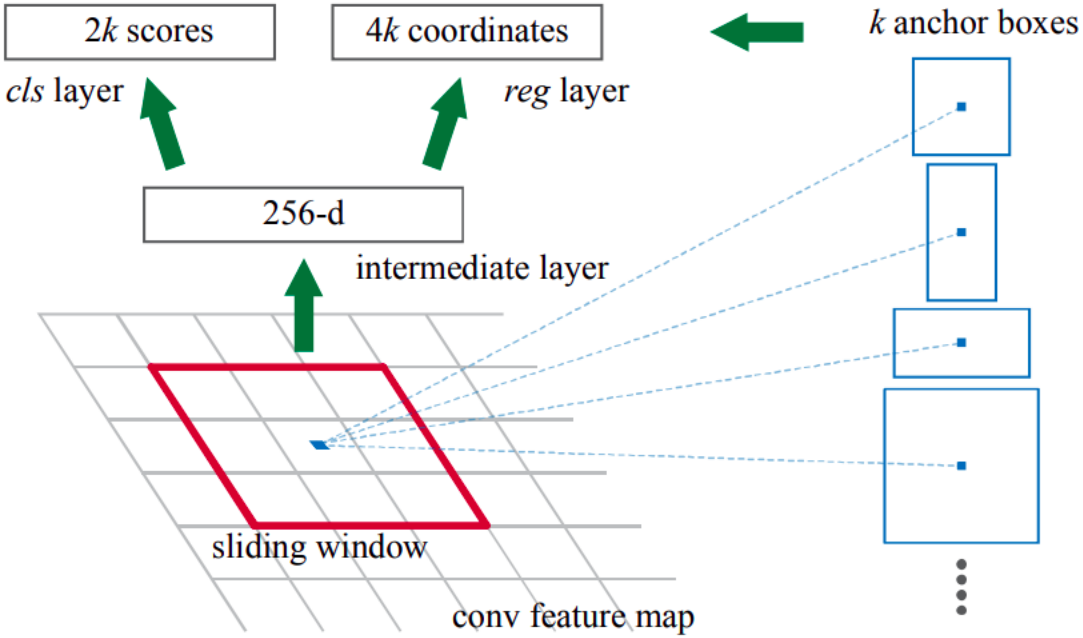


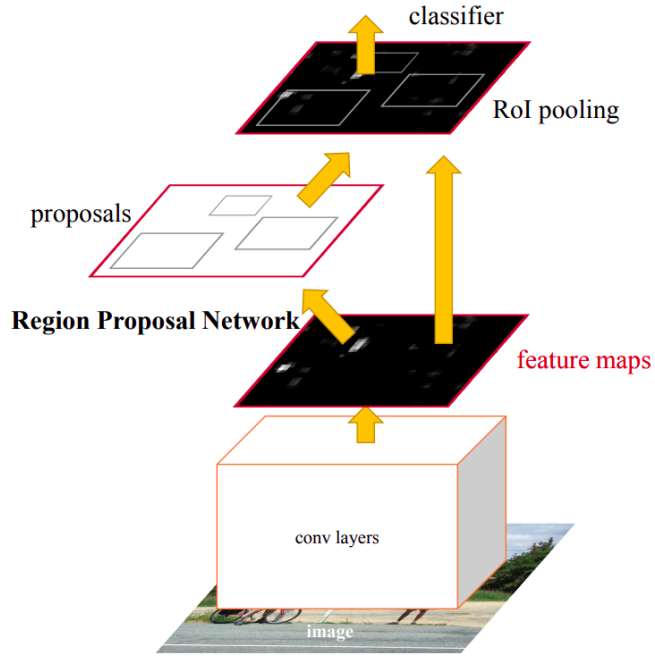
[girshick15iccv]

Objekti: Faster R-CNN

Glavna ideja: integrirano generiranje kandidata i detekcija objekata

- provući sliku kroz duboki model, izlučiti značajke h/32×w/32×D
- iz tih značajki, gusto detektirati kandidate i pomake za k sidara
  - modul za predlaganje kandidata (RPN - region proposal network)
- izlučiti deskriptor svakog pozitivnog kandidata (7×7×D, ROI pool) i proslijediti ga klasifikacijskom modulu (slično Fast R-CNN)





[ren15nips]

Objekti: Faster R-CNN (2)

Backprop kroz ROI pool sada je kompliciran jer okno nije fiksno

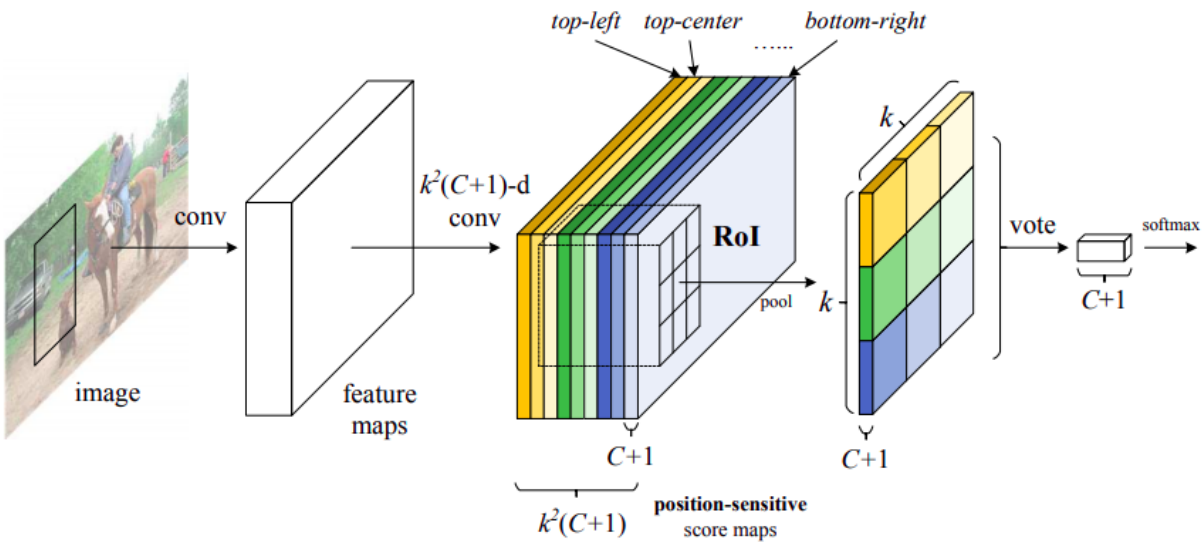
- zbog toga učenje alternira između optimiranja RPN i klasifikacijskog modula
- ovaj problem rješava bilinearno uzorkovanje deskriptora regije (ROI align)

VOC'12 test: 70.4% mAP@0.5; COCO test-dev: 21.9% mAP COCO

Objekti: R-FCN

Ideja: cjelokupan posao provesti jednim modulom

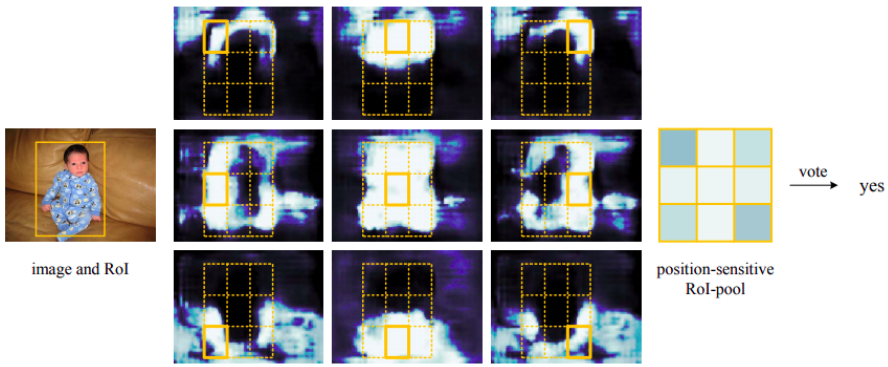
- glavni trik: pozicijski osjetljivo sažimanje
  - okvir kandidata se podijeli na kxk zona;
  - svaka mapa agregira jednu od tih zona za svaki razred
  - puno veći domet od konvolucija
- jednako točno kao Faster R-CNN, ali značajno brže

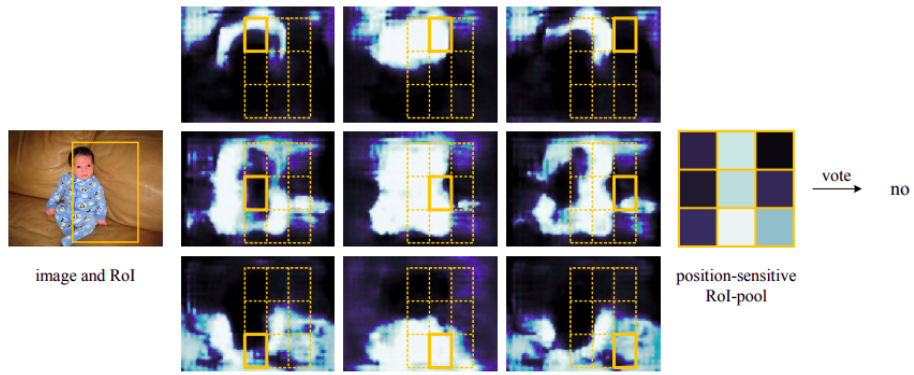


[dai16nips]

Objekti: R-FCN (2)

Pozicijski osjetljive mape aktiviraju se na odgovarajućim relativnim položajima u odnosu na objekt:

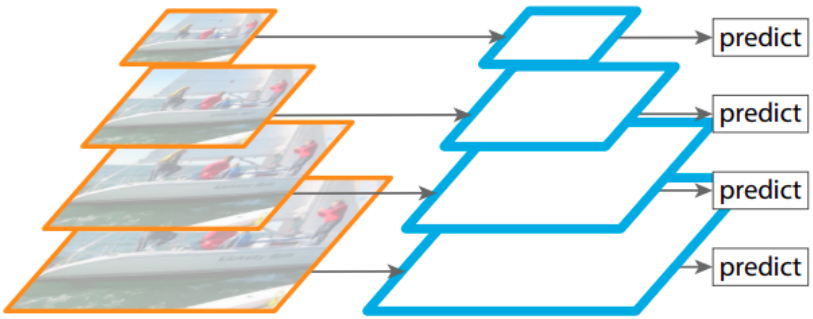




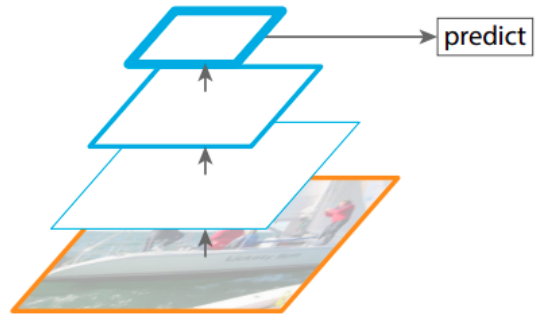
[dai16nips]

VOC'12 test: 77.6% mAP@0.5; COCO test-dev: 31.5% mAP COCO

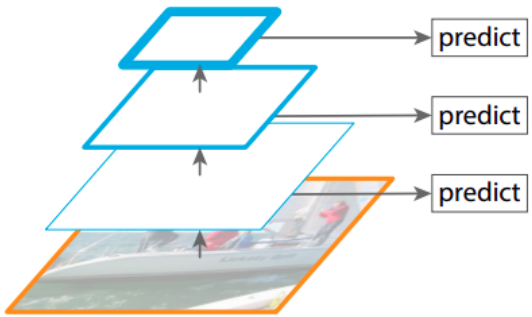
Objekti: FPN



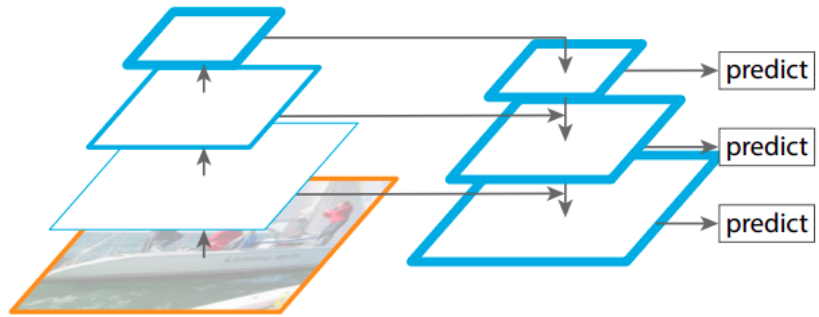
(a) Featurized image pyramid



(b) Single feature map



(c) Pyramidal feature hierarchy



(d) Feature Pyramid Network

[lin17cvpr]

Faster R-CNN + FPN, COCO test-dev: 36.2% mAP COCO

Objekti: SoftNMS

Input : $\mathcal{B} = \{b_1, .., b_N\}, \mathcal{S} = \{s_1, .., s_N\}, N_t$
$\mathcal{B}$ is the list of initial detection boxes
$\mathcal{S}$ contains corresponding detection scores
N_t is the NMS threshold

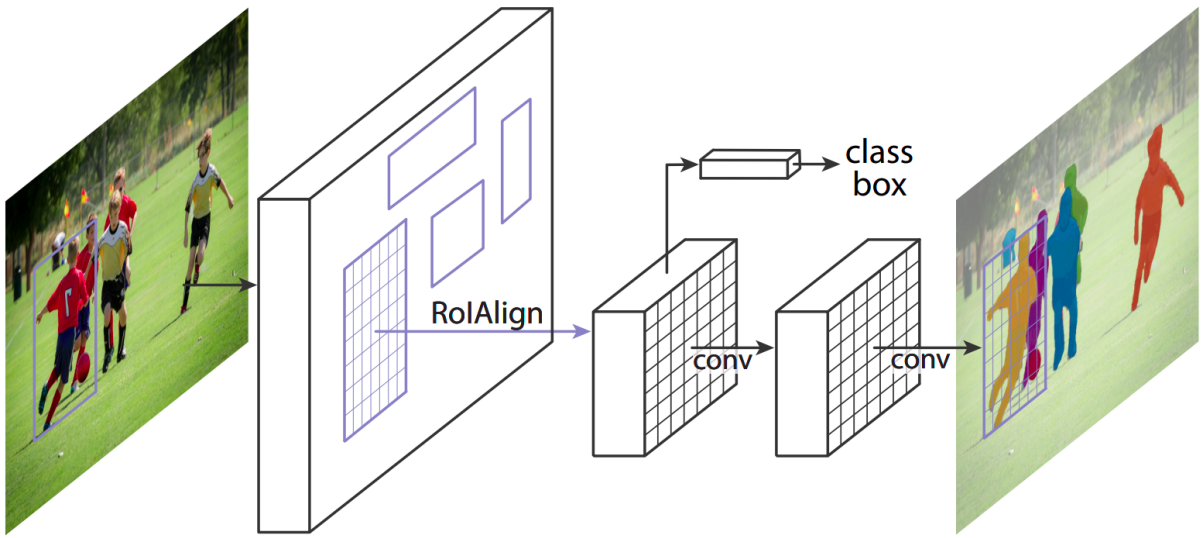
begin
  $\mathcal{D} \leftarrow \{\}$
  while $\mathcal{B} \neq$ *empty* **do**
    $m \leftarrow$ argmax $\mathcal{S}$
    $\mathcal{M} \leftarrow b_m$
    $\mathcal{D} \leftarrow \mathcal{D} \bigcup \mathcal{M}; \mathcal{B} \leftarrow \mathcal{B} - \mathcal{M}$
    for b_i *in* $\mathcal{B}$ **do**
      if $iou(\mathcal{M}, b_i) \geq N_t$ **then**
        $\mathcal{B} \leftarrow \mathcal{B} - b_i; \mathcal{S} \leftarrow \mathcal{S} - s_i$
      end
      NMS
      $s_i \leftarrow s_i f(iou(\mathcal{M}, b_i))$
      Soft-NMS
    end
  end
  return $\mathcal{D}, \mathcal{S}$
end

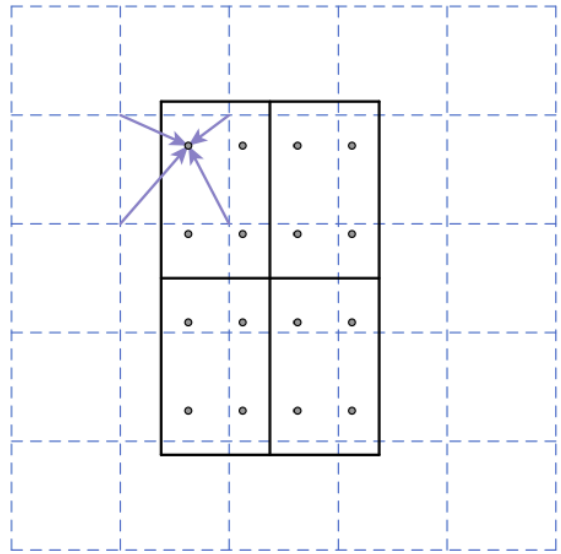
[bodla17iccv]

D-RFCN + MST + SoftNMS G, COCO test-dev: 40.9% mAP COCO

Objekti: Mask R-CNN

Ideja: proširiti Faster glavom koja prediktira segmentaciju primjerka, dodati ROI align.



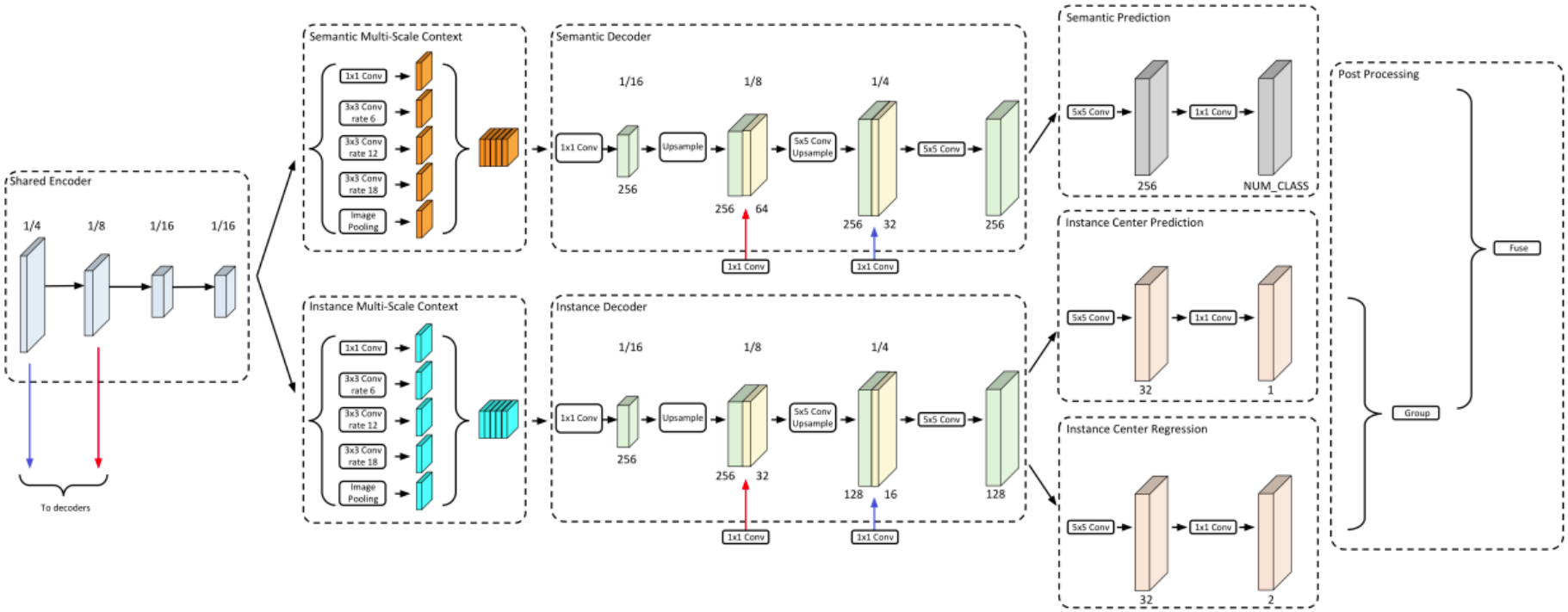


[he17iccv]

Mask R-CNN + FPN, COCO test-dev: 39.8% mAP COCO

Objekti: PDL

Ideja: gusto predviđati i) semantički razred, ii) središta objekata te iii) vektore do središta. Instance se formiraju tijekom postprocesiranja



[cheng20cvpr]

COCO test-dev: 41.4% PQ COCO

Gusta predikcija: segmentacija

Razumijevanje slike na razini piksela (semantička segmentacija):

- svrstati svaki slikovni element u odgovarajući razred
- razredi imaju značenje koje je važno za misiju agenta

  sudionici: osoba, ciklist, auto, bicikl, kamion, autobus, vlak, motor

  signalizacija: stup, znak, semafor

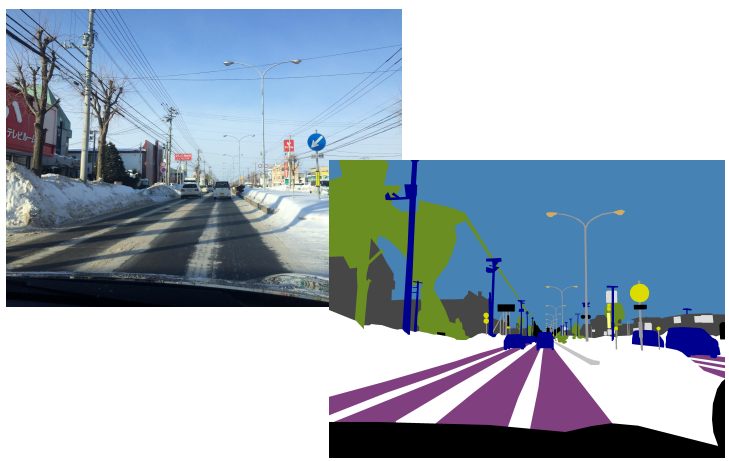
  okoliš: cesta, nogostup, zgrada, bilje, teren, ograda, zid, nebo




Gusta predikcija: datasets

- Cityscapes [cordts16cvpr]:
  - perspektiva vozača, 19 razreda
  - 5000 stereo slika, 2MPixela
  - dobar odabir razreda i kategorija
  - 50 gradova, proljeće do jeseni

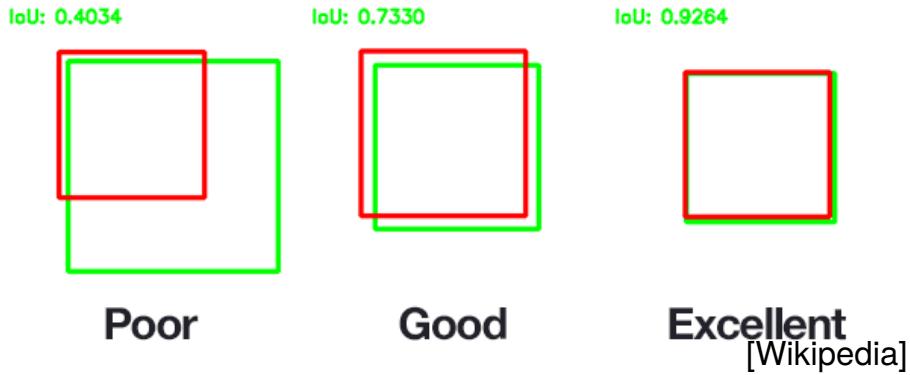
- Vistas [neuhold17iccv]:
  - perspektiva vozača, 100 razreda
  - 25000 slika, 2-8 MPixel
  - instance level annotations
  - širom svijeta, snijeg, magla, noć



Gusta predikcija: točnost

Široko korištena metrika: omjer presjeka i unije (IoU)

- skup A: označeni pikseli razreda c
- skup B: pikseli klasificirani u c
- $\text{IoU}_X = |A \cap B| \,/\, |A \cup B|$



[Wikipedia]

Ukupnu uspješnost izražavamo kao srednji IoU preko svih razreda

- $\text{mIoU} = \frac{\sum_c \text{IoU}_c}{C}$
- ovo povećava utjecaj piksela rijetkih razreda
- primjeri: zid, ograda, stup, boca, sobna biljka

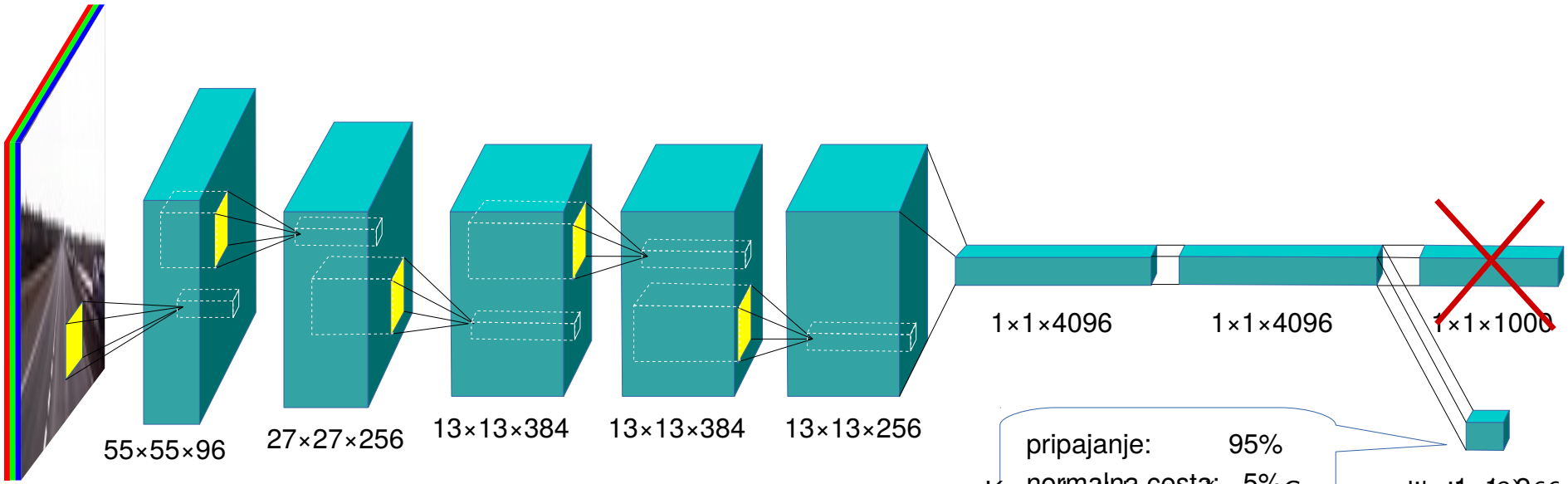
Oznake ispitnih slika nisu javno dostupne:

- ispitnu točnost određujemo podnošenjem na evaluacijski server
- naš mIoU po kategorijama: 89.7 (najbolji rezultat: 91.6)

Gusta predikcija: prijenos znanja

Duboki klasifikacijski model može biti prilagođen za novi (lakši) zadatak:

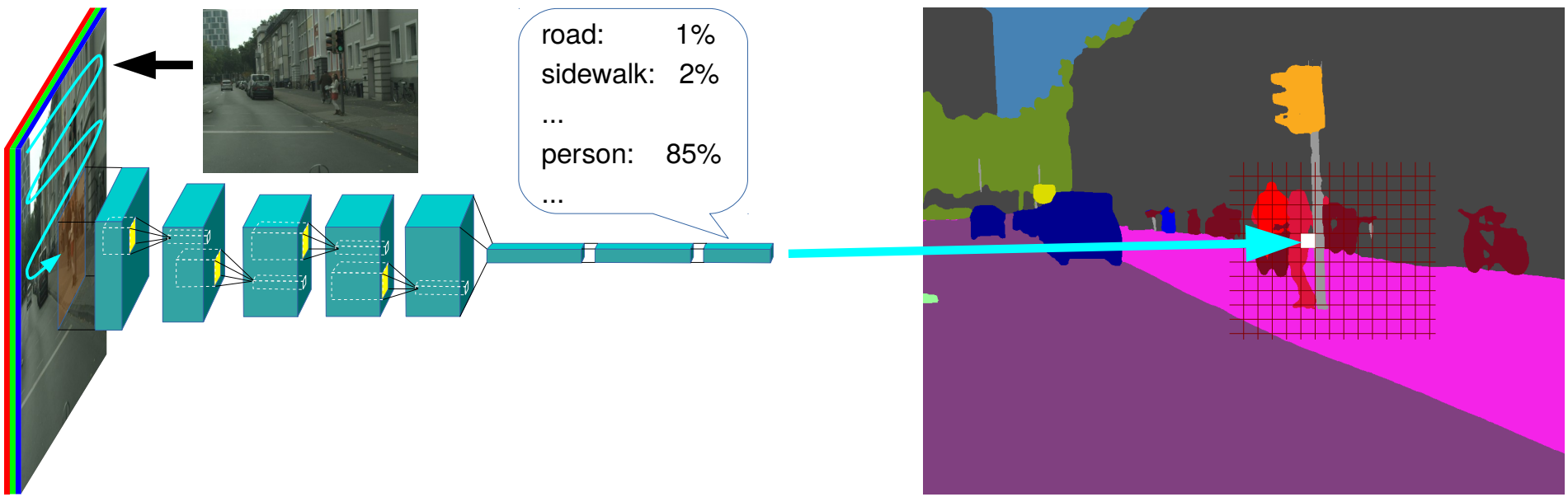
- odrezati posljednjih nekoliko slojeva
- spojiti preostale slojeve s prednjim krajem za novi zadatak
- trenirati dobiveni model za novi zadatak
- naslijeđeni slojevi su već naučeni pa sada možemo učiti s manje podataka (nekoliko tisuća slika)



Gusta predikcija: povratak pomičnog okna

Ideja: primijeniti klasifikacijski model u **pomičnom oknu**

- svako okno producira semantički razred piksela (ili okvir objekta)
- gusto **označene** slike omogućavaju učenje s kraja na kraj



U praksi potrebna optimizacija: 10^6 piksela $\times$ 10^9 množenja?

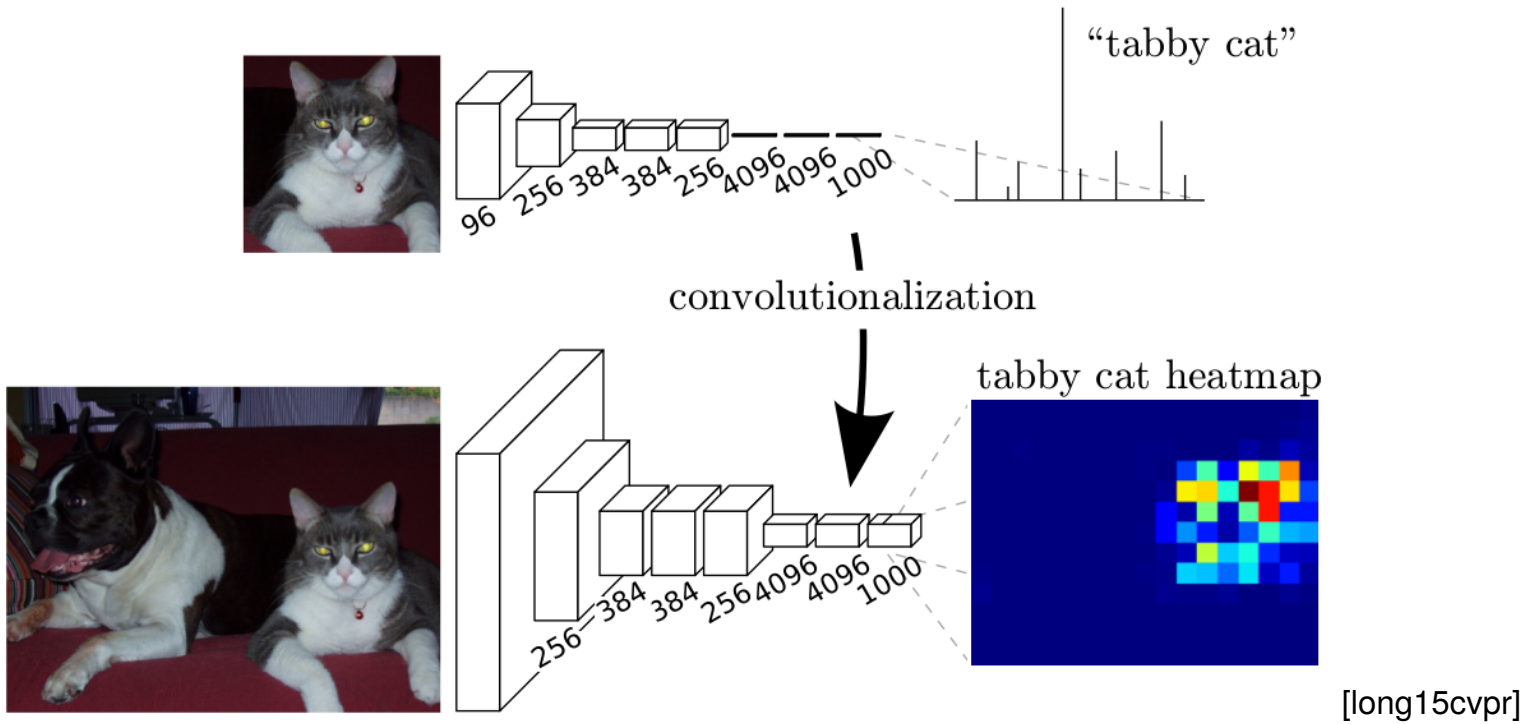
Izazovi: veliki objekti, mali objekti, računska složenost.

Gusta predikcija: segmentacija u praksi

Obrada susjednih okana zahtijeva računanje istih latentnih aktivacija

Optimizacija: evaluirati pomično okno sloj po sloj [long15cvpr]:

- izlazni tenzor je poduzorkovan zbog sažimanja

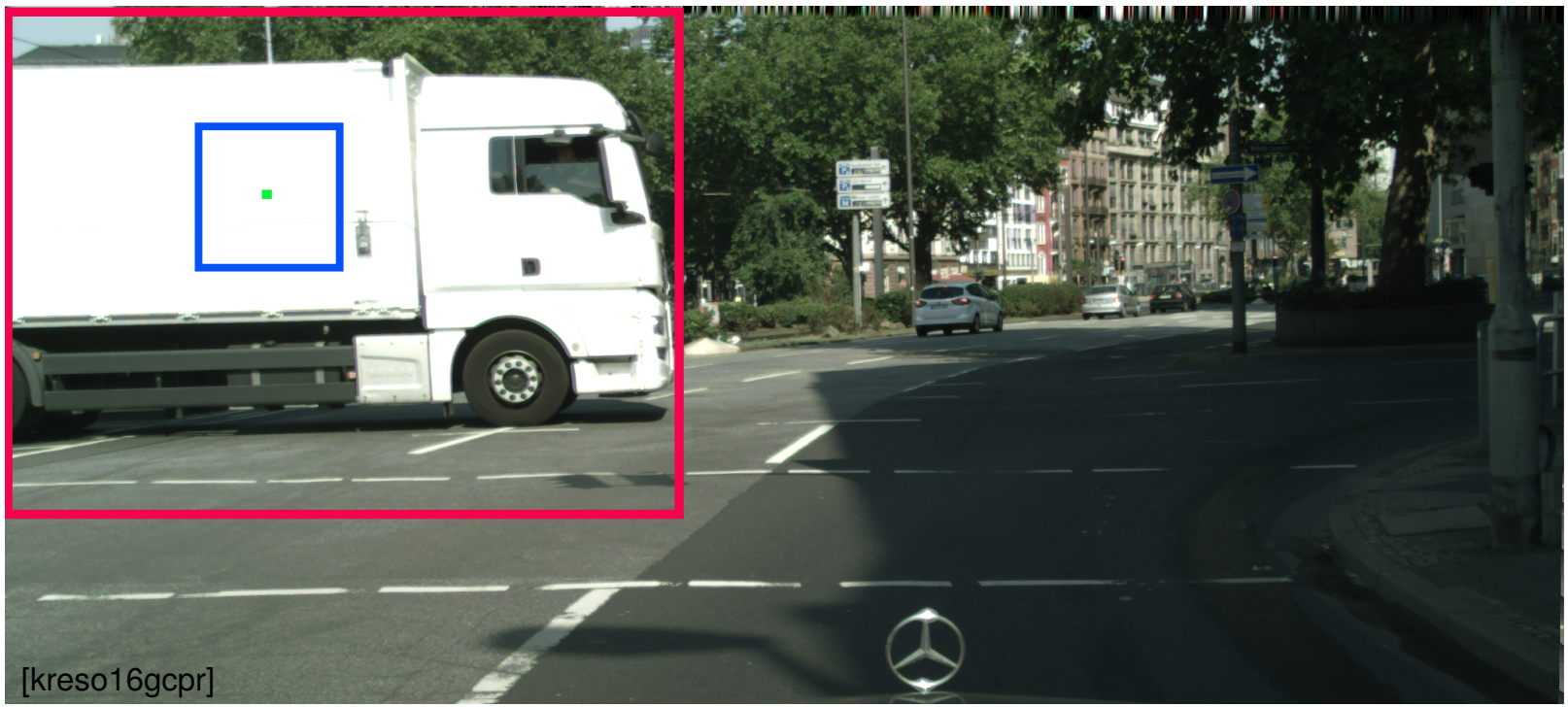


[long15cvpr]

Gusta predikcija: veliki objekti

Prepoznavanje velikih objekata zahtijeva gigantsko receptivno polje

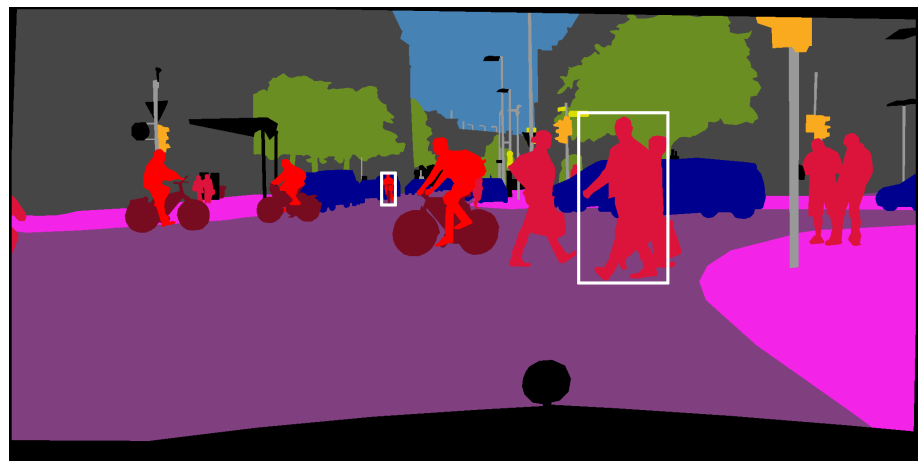
- velik broj lokalnih susjedstava nije dovoljno diskriminativan
- takva susjedstva mogu biti prepoznata samo u većem kontekstu
- problemi nastaju kada je kontekst veći od receptivnog polja



Gusta predikcija: mali objekti

Prepoznavanje **malih** objekata moćnim modelom s **velikim** receptivnim poljem rasipa resurse:

- mali objekti mogu biti prepoznati s malim brojem slojeva
- kasniji slojevi moraju prosljeđivati aktivacije bez doprinosa kvaliteti obrade
- to vodi do gubitka reprezentacijske moći i prenaučenosti modela

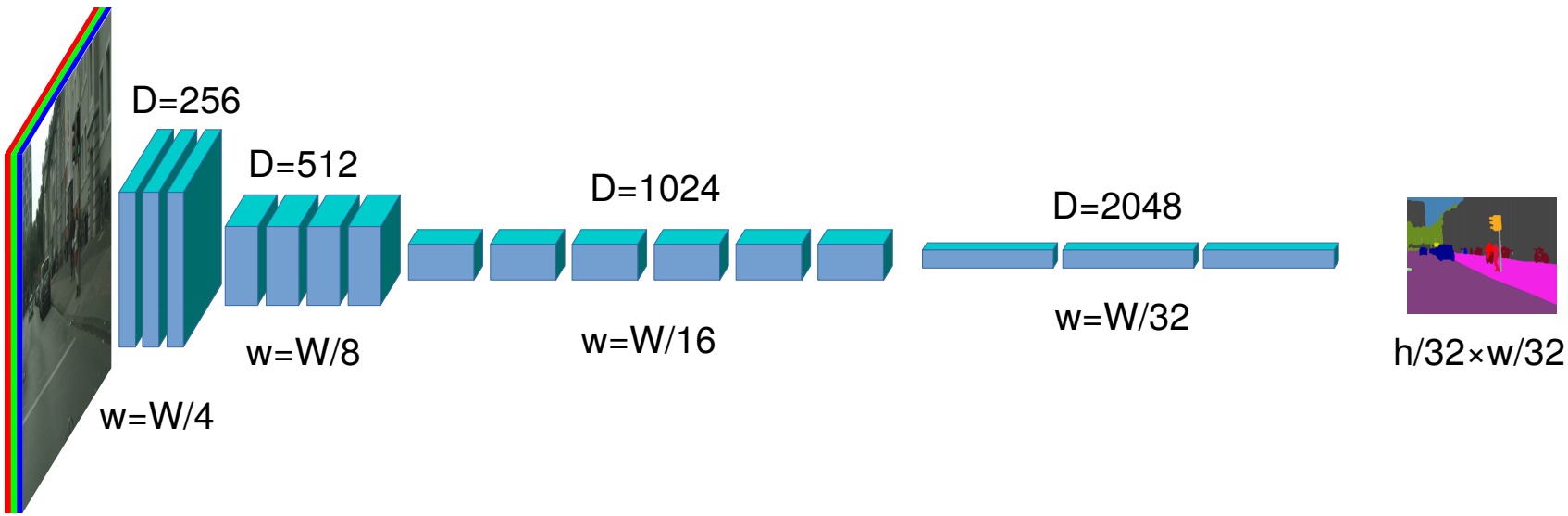


[kreso16gcpr]

Gusta predikcija: složenost

Uspješne segmentacijske arhitekture temelje se na prednaučenim klasifikacijskim modelima: mala ulazna i još manja izlazna rezolucija

- u segmentaciji trebamo veliku rezoluciju i na ulazu i izlazu
- to postavlja ogromne zahtjeve na GPU memoriju
  - prilikom učenja moramo pamtiti svih 100 latentnih tenzora
- to otežava evaluaciju modela na jednostavnim računalima
  - pola milijarde množenja po slici za najjednostavniji model

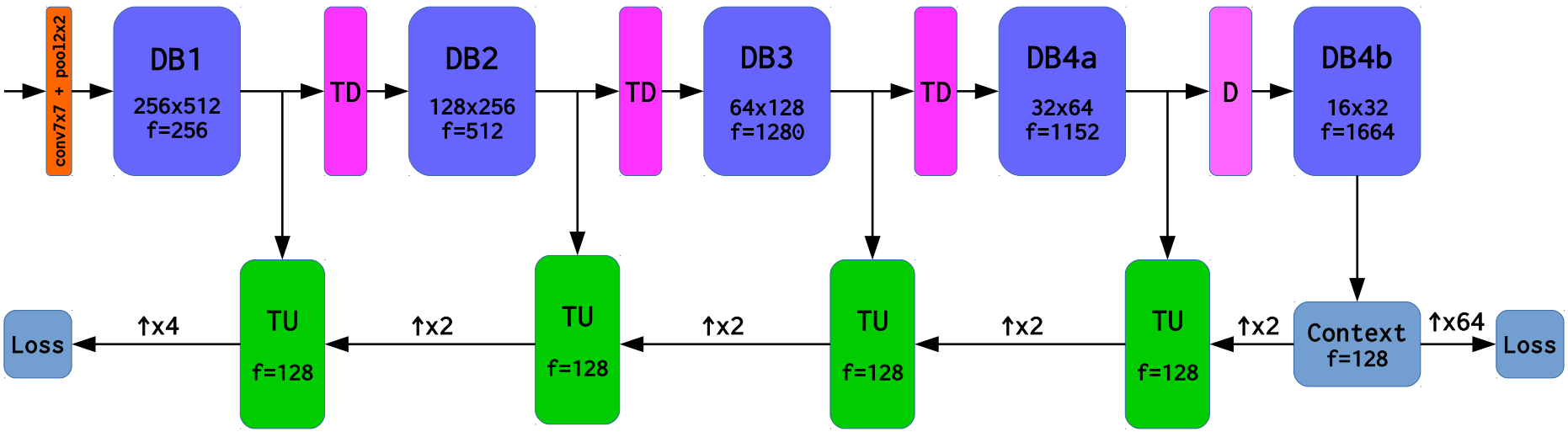


Gusta predikcija: pametno naduzorkovanje

Ideja: nadoknaditi poduzorkovanje **miješanjem** slojeva na različitim dubinama [valpola14arxiv,ronneberger15arxiv,lin17cvpr]:

Prepoznavanje razreda je teže od finog podešavanja granica:

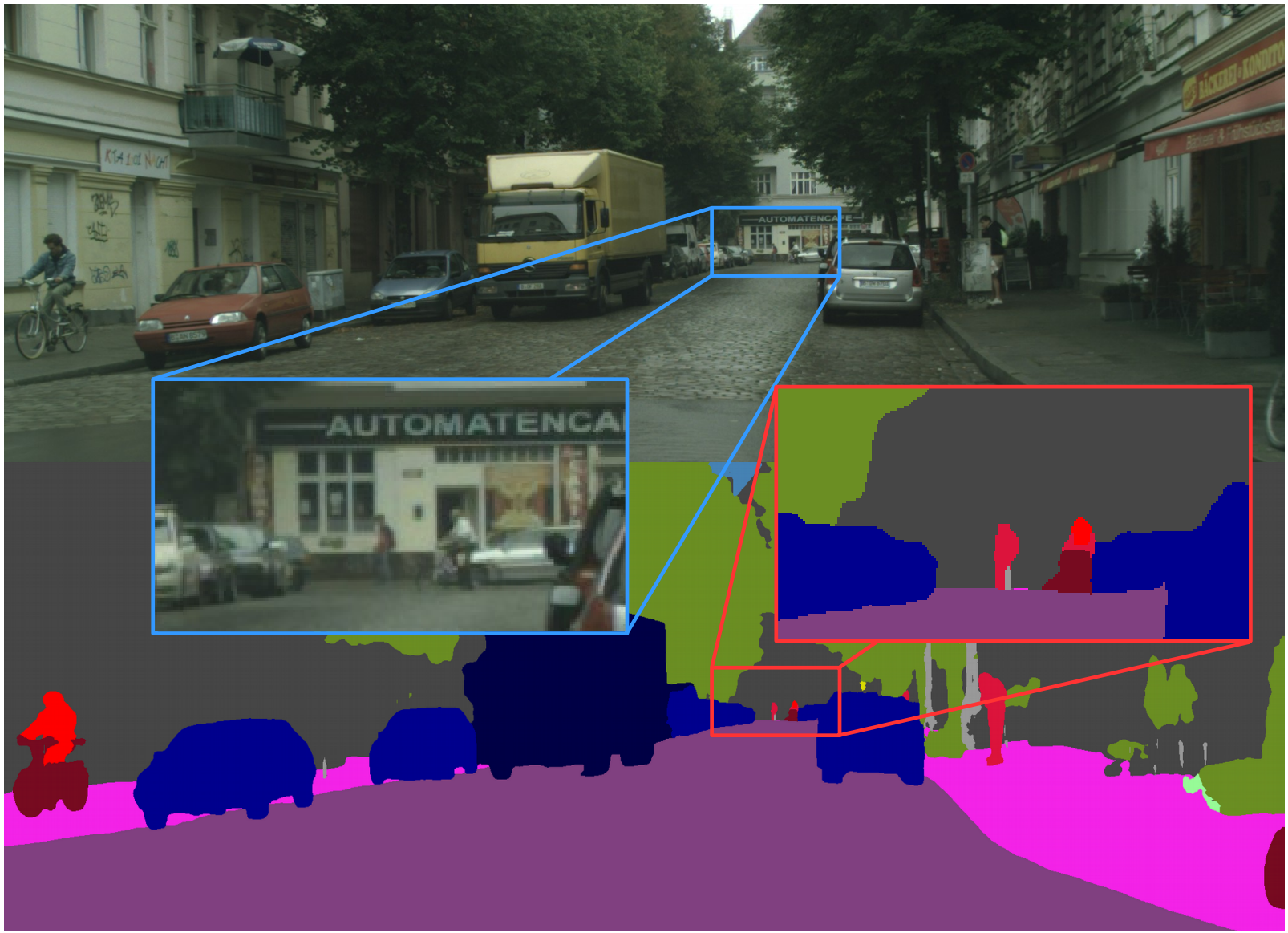
- na visokoj rezoluciji može se koristiti **malena** reprezentacija



[kreso17iccvw]

Gusta predikcija: studija slučaja

Pješak i biciklist na 200m primjerno prepoznati sa 6 od 7 modela:



Real-time prediction: approach

Recent work provides solid empirical evidence that convolutional models are extremely resistant to overfitting [zhang17iclr]

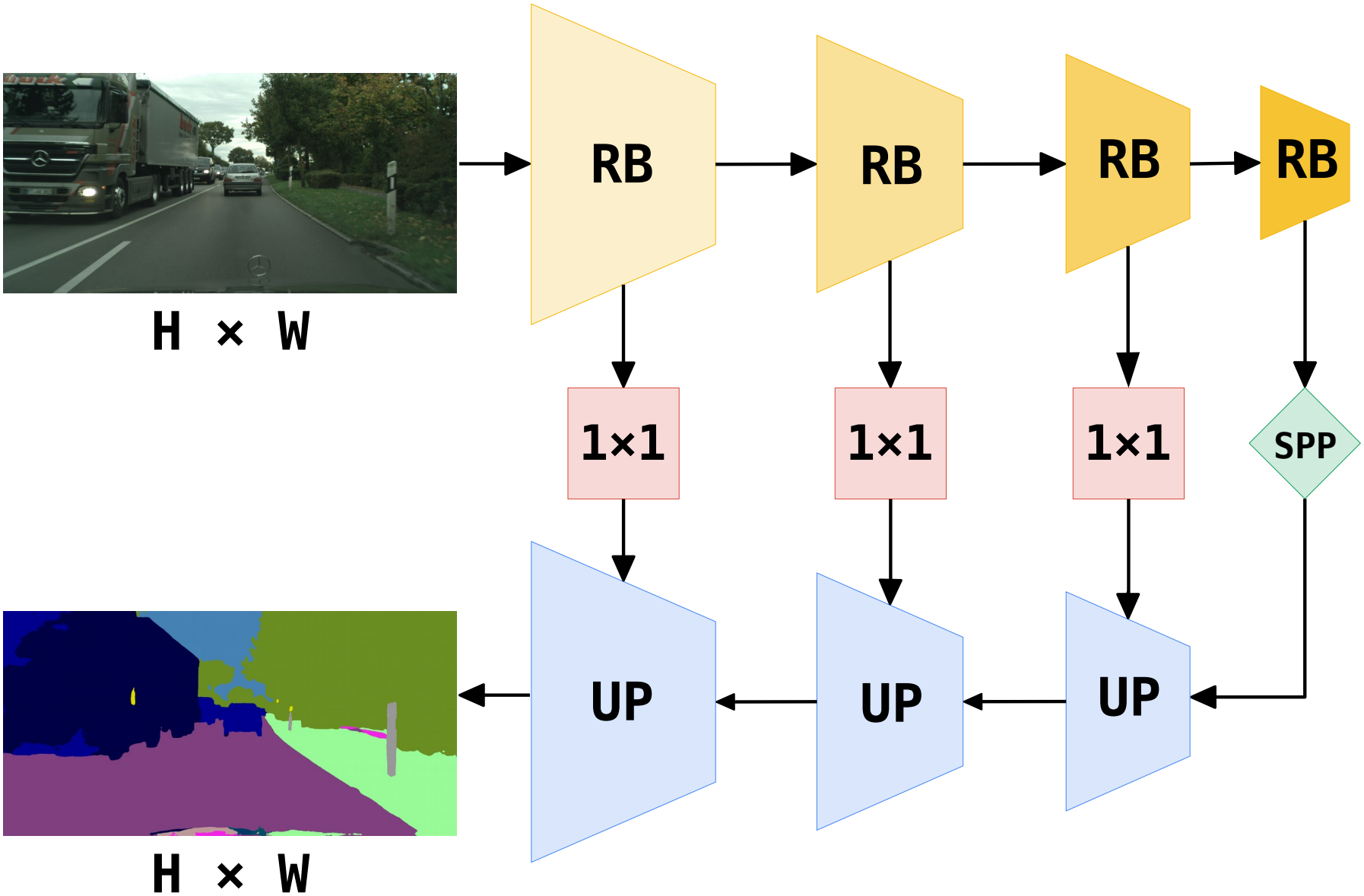
Hence, practitioners tend to overshoot the model capacity

Oversupply of modelling power leads to diminished returns.

On the other hand, it is not sensible to renounce on ImageNet pre-training whenever we deal with natural images

Hence, we base all our real-time models on lightweight ImageNet pre-trained models [orsic19cvpr]

Real-time prediction: baseline



[orsic19cvpr]

Real-time prediction: details

Downsampling path (ImageNet backbone): ResNet-18 [he15cvpr] or MobileNetv2 [sandler18cvpr]

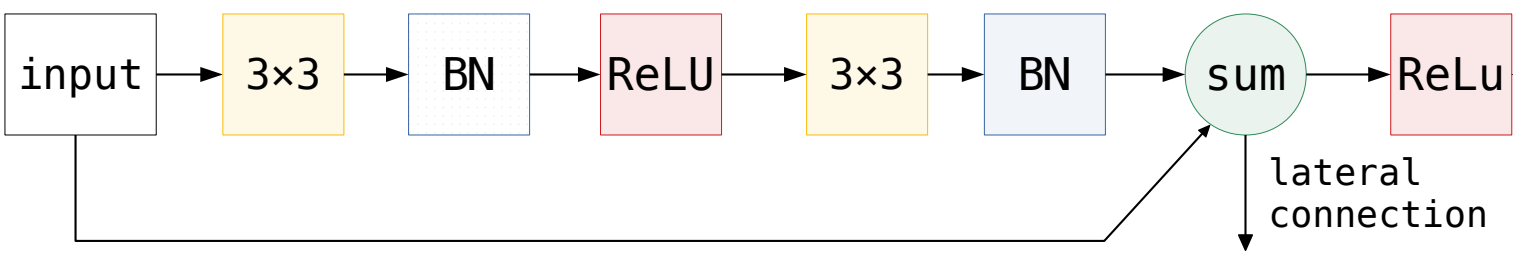
- ensures efficient recognition

Spatial pyramid pooling [zhao17cvpr,kreso19arxiv]

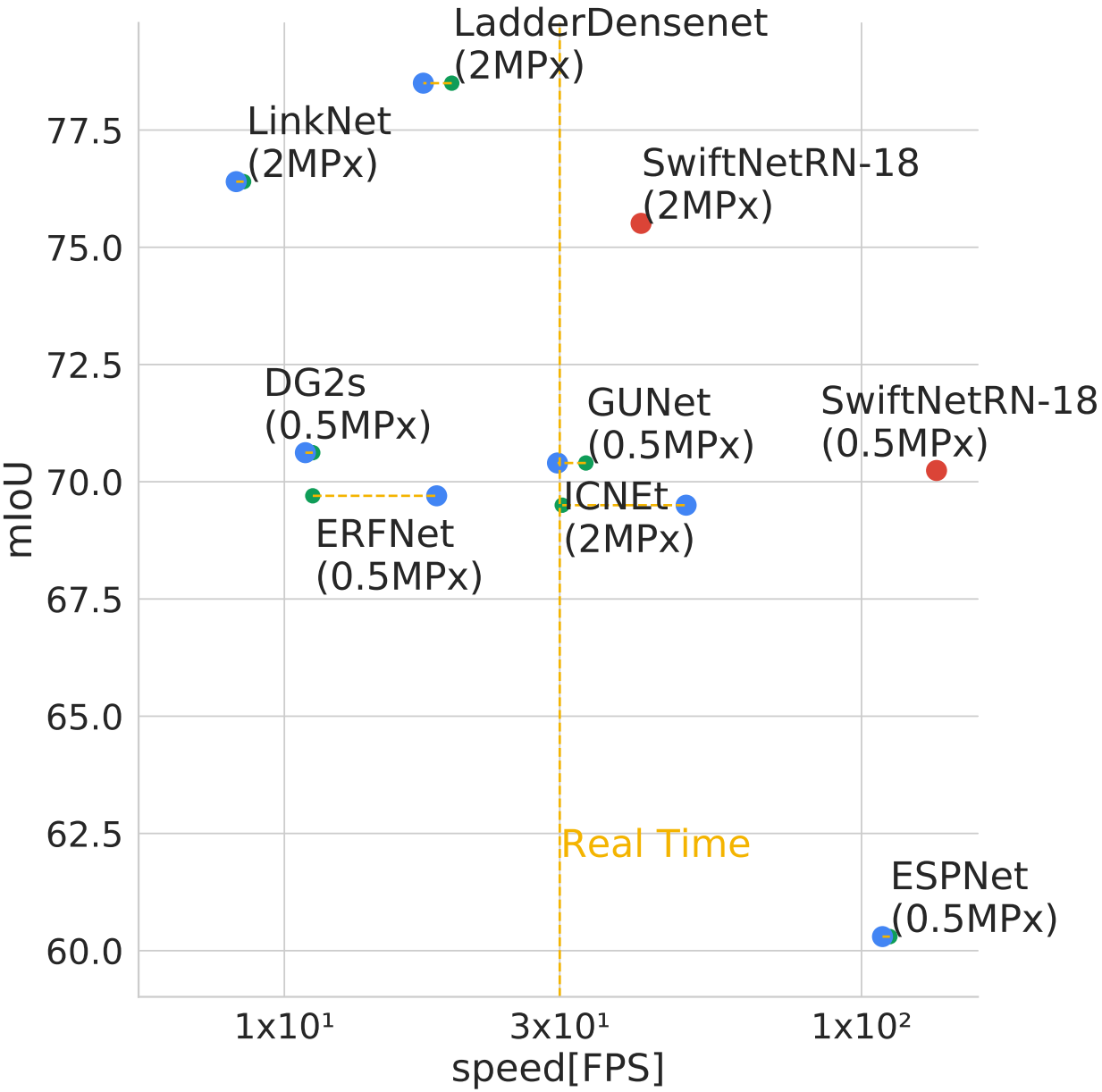
- ensures large receptive field (for large objects)

Ladder-style upsampling [lin17cvpr,kreso17iccvw]

- recovers details (for small objects)
- skip-connection taken before ReLU [orsic19cvpr]



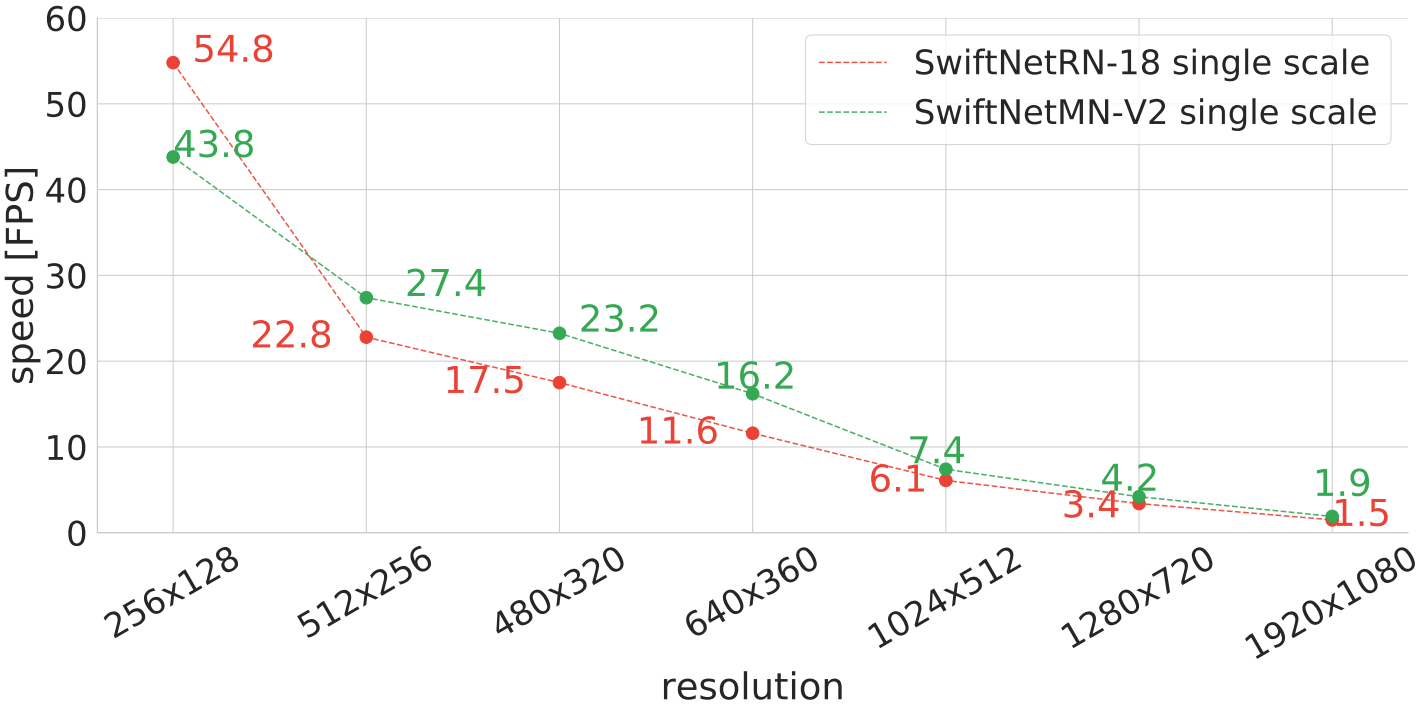
Real-time prediction: results



[orsic19cvpr]

Best accuracy/latency among all previously published models

Real-time prediction: embedded

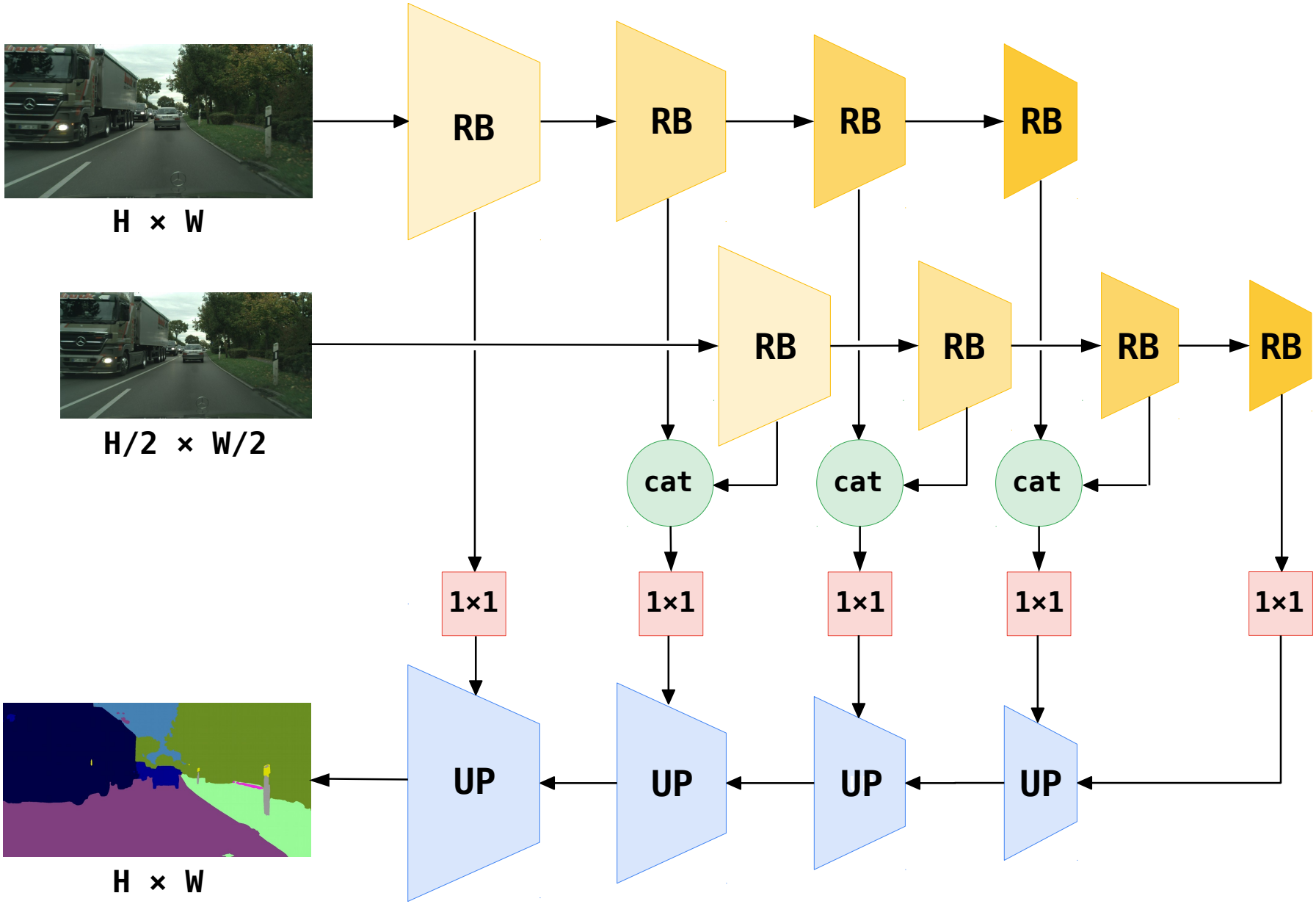


[orsic19cvpr]

Real-time performance on an embeded SoC

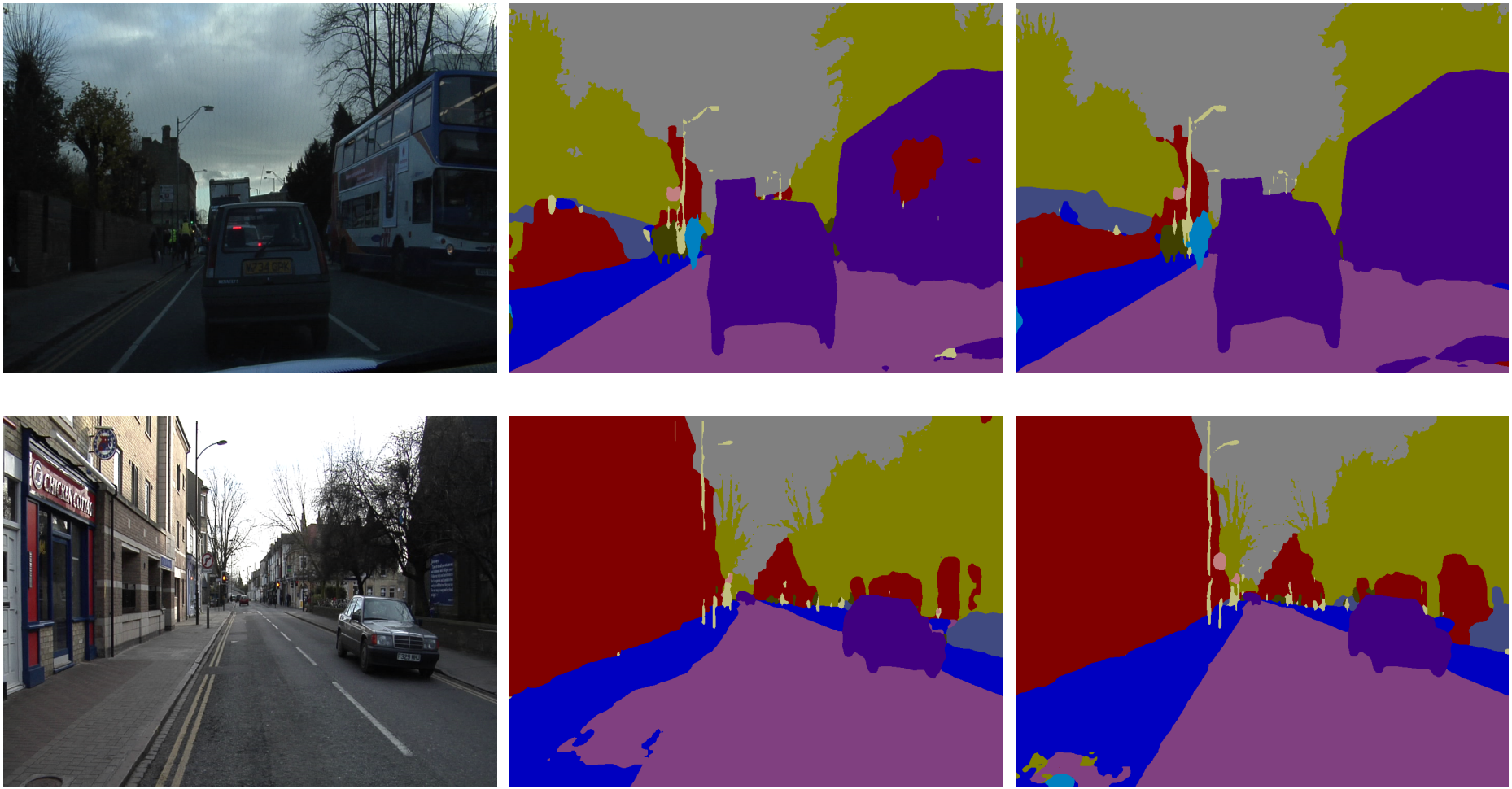
- 256×512 pixels (RGB)
- 27.4 Hz at Jetson TX2 (15W)

Real-time prediction: pyramid fusion



[orsic19cvpr]

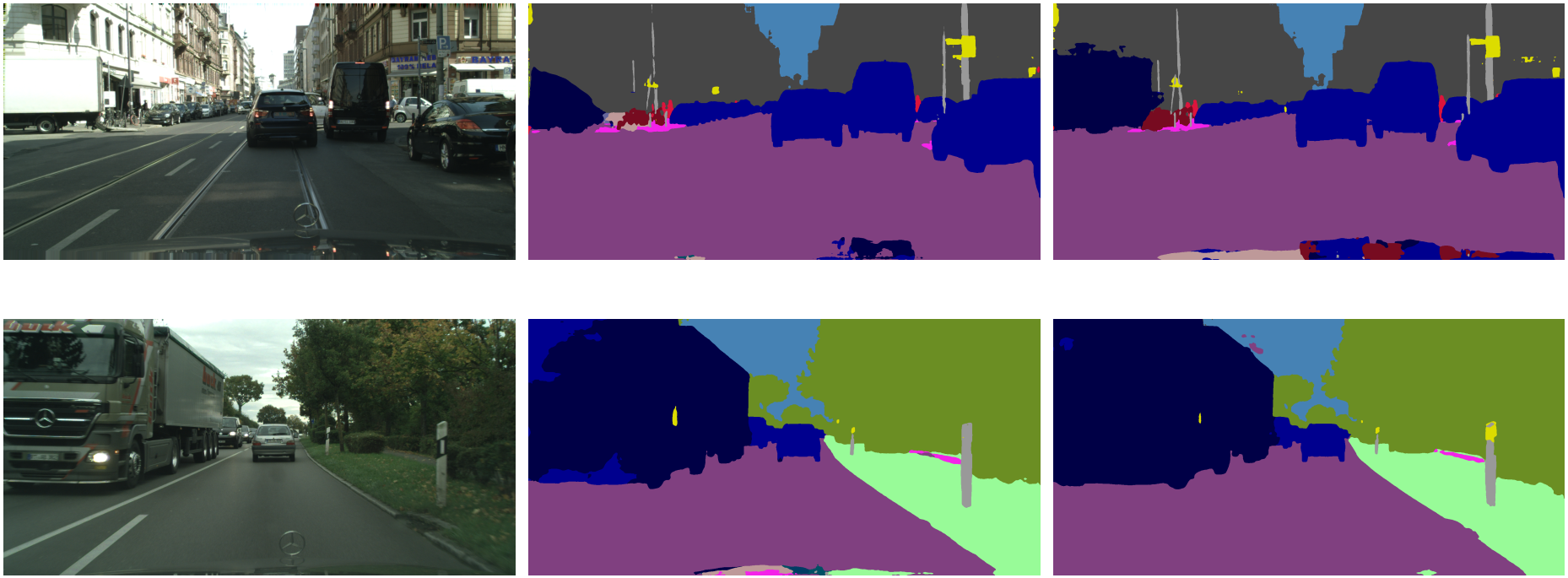
Real-time prediction: results



[orsic19cvpr]

In both cases, the pyramid fusion improves recognition of surfaces at large structures (bus, sidewalk)

Real-time prediction: results



[orsic19cvpr]

Again, the pyramid fusion improves recognition of large objects

Other experiments show that pyramid fusion enlarges the effective receptive field of the predictions.

Real-time prediction: conclusions

ImageNet pre-training leads to 5pp mIoU improvement

Classifier capacity can be compensated by careful design:

- spatial pyramid pooling [zhao17cvpr]
- pyramidal fusion [orsic19cvpr]
- ladder-style upsampling [lin17cvpr,kreso17iccvw]

Pyramidal fusion vs spatial pyramid pooling:

- improved accuracy for 1pp
- improved effective receptive field
- 10% increase in the FLOP count

Semantic forecasting: the task

Anticipate events by forecasting semantic segmentation of an unobserved future frame (Δt = 180 or 540 ms)

current frame (observed)

future frame (unobserved)

groundtruth (used for evaluation)

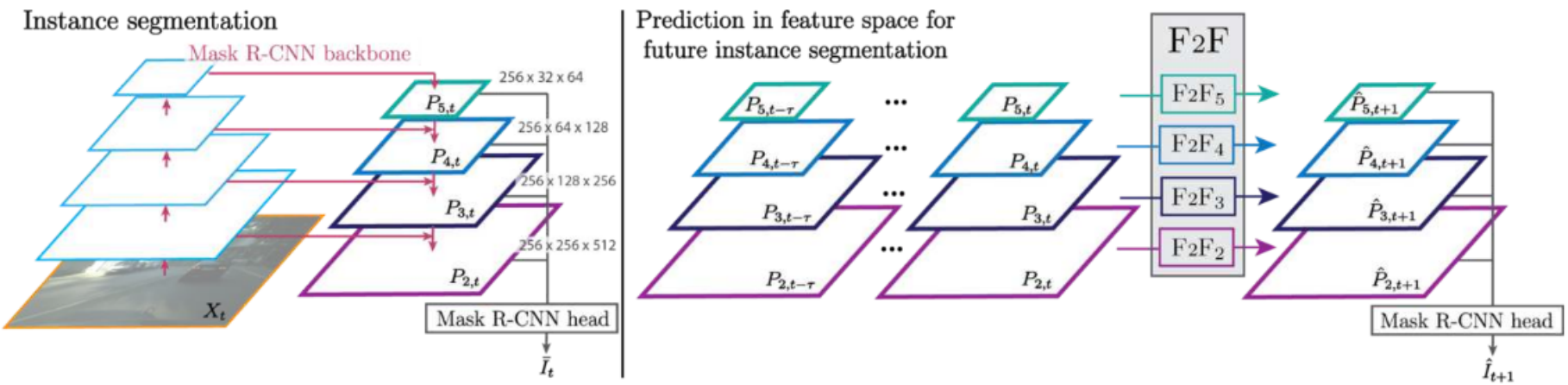
semantic forecast (our result)

Semantic forecasting: previous work

Recent work suggests that forecasting abstract features is easier than forecasting pixels [luc18eccv]

- abstract features are more informative than pixels
- especially interesting since it can be trained with **no labels**

However, they propose a heavyweight model which requires training a separate mapping at different levels of abstraction [luc18eccv]



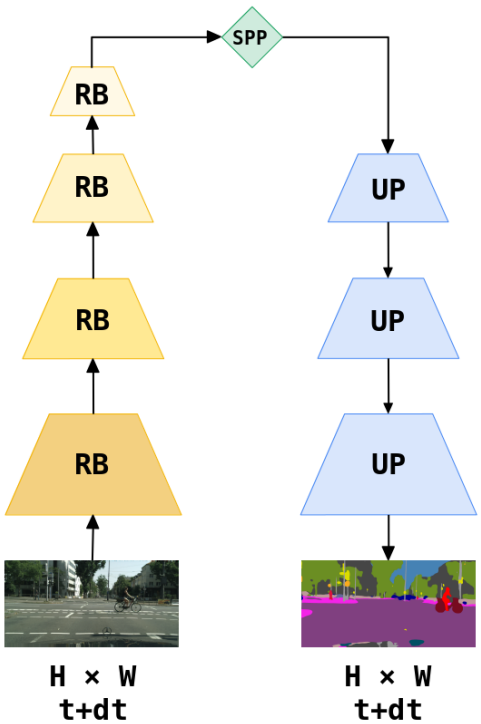
Semantic forecasting: approach

We again hypothesize that model capacity can be compensated by smart design:

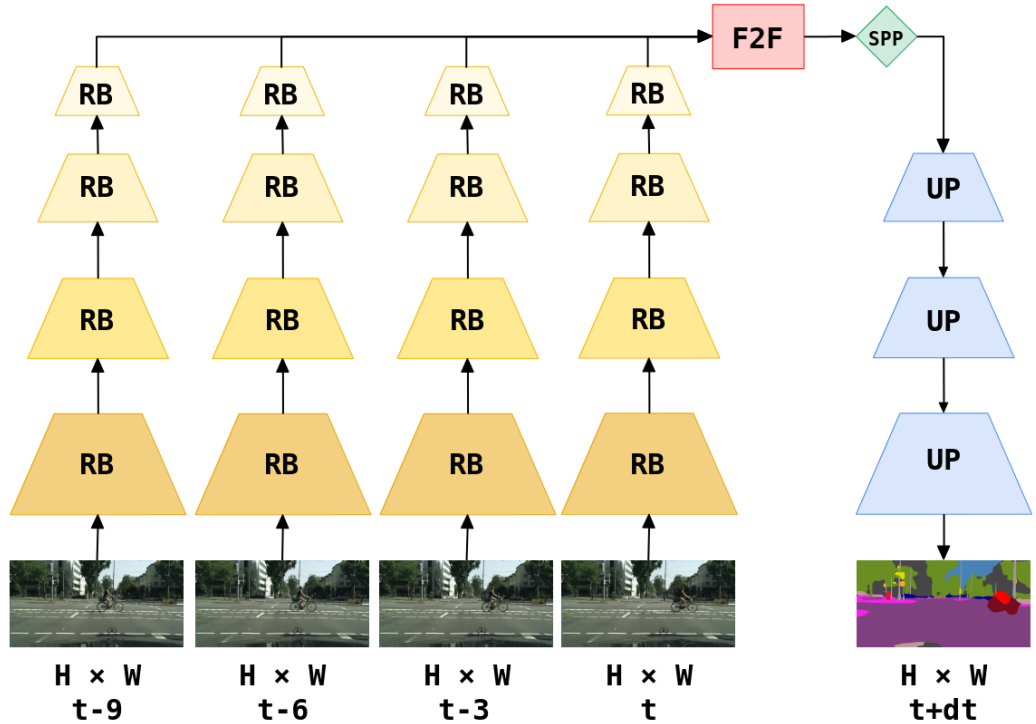
- use a lightweight ImageNet pre-trained recognition backbone (ResNet-18) [orsic19cvpr]
- forecast only the most abstract features
  - use the single-frame model **without** ladder-style upsampling
- use a simple F2F model with **deformable convolutions**
- due to simplicity, we can finetune F2F with supervised loss

A forecasting model with more capacity could not fit into GPU RAM and would require more data to train

Semantic forecasting: the proposed solution



single-frame model



forecasting model

[saric19gcpr]

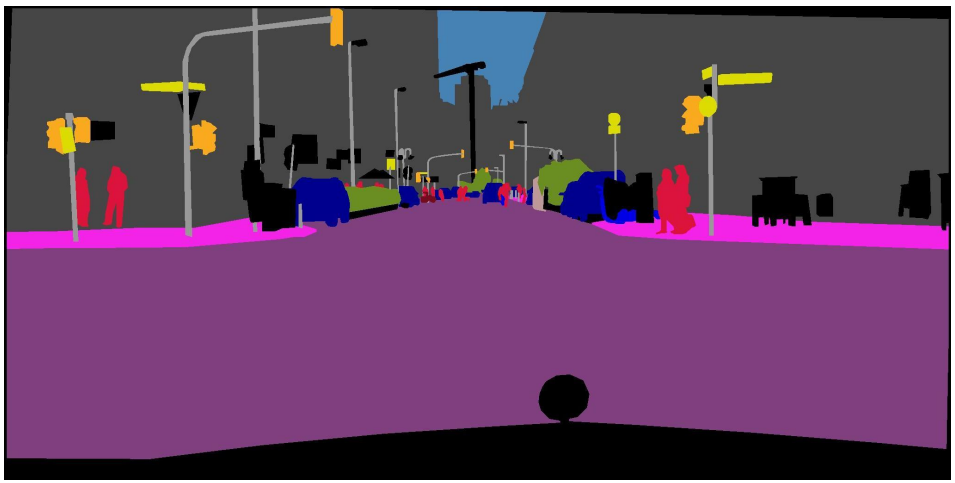
Semantic forecasting: results

	Short-term		Mid-term	
	mIoU	mIoU-MO	mIoU	mIoU-MO
Oracle	72.5	71.5	72.5	71.5
Copy last segmentation	52.2	48.3	38.6	29.6
Luc Dil10-S2S [luc17iccv]	59.4	55.3	47.8	40.8
Luc Mask-S2S [luc18eccv]	/	55.3	/	42.4
Luc Mask-F2F [luc18eccv]	/	61.2	/	41.2
Nabavi [nabavi18bmvc]	60.0	/	/	/
Bhattacharyya [bhattacharyya19iclr]	65.1	/	51.2	/
Terwilliger [terwilliger19wacv]	**67.1**	**65.1**	51.5	46.3
Luc F2F (our implementation)	59.8	56.7	45.6	39.0
DeformF2F-8	64.4	62.2	52.0	48.0
DeformF2F-8-FT	64.8	62.5	**52.4**	**48.3**
DeformF2F-8-FT (2 samples per seq.)	65.5	63.8	**53.6**	**49.9**

Semantic forecasting: mid-term example

most recent input

future frame (unobserved)



ground truth

mid-term forecast

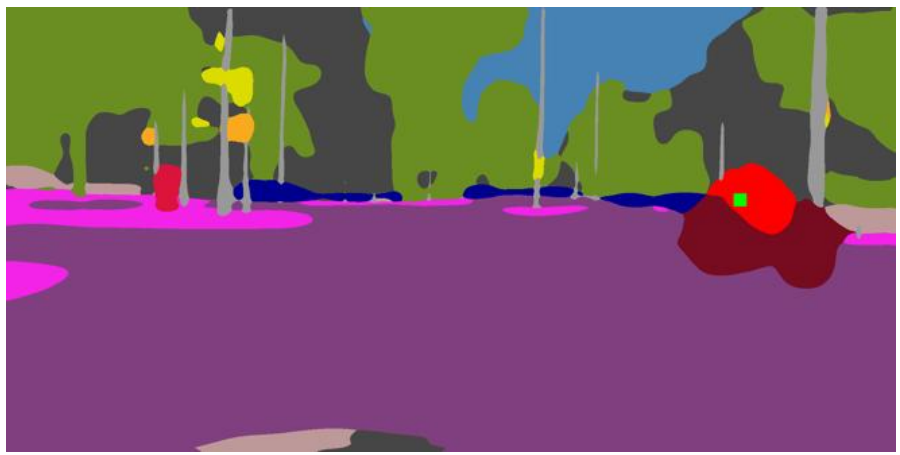
Semantic forecasting: explaining results

We show pixels with the strongest log-max-softmax gradient (red) in a hand-picked pixel (green)

t-3

t

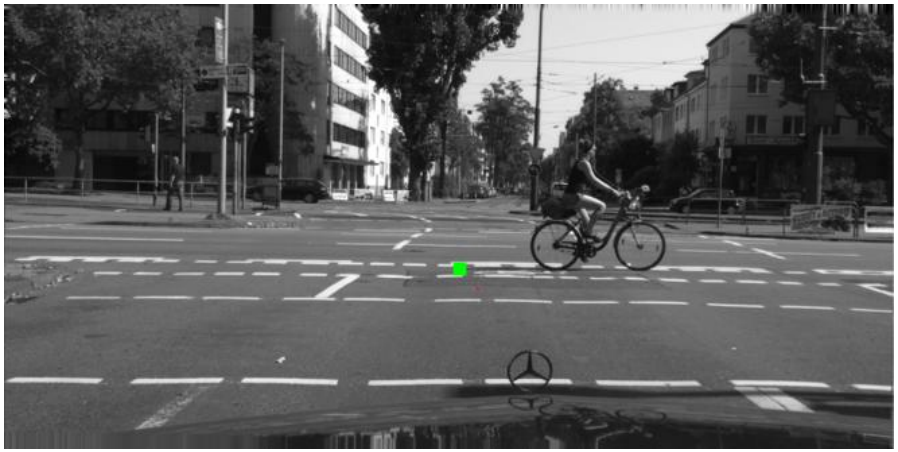
t + 9



forecast

Semantic forecasting: explaining results (2)

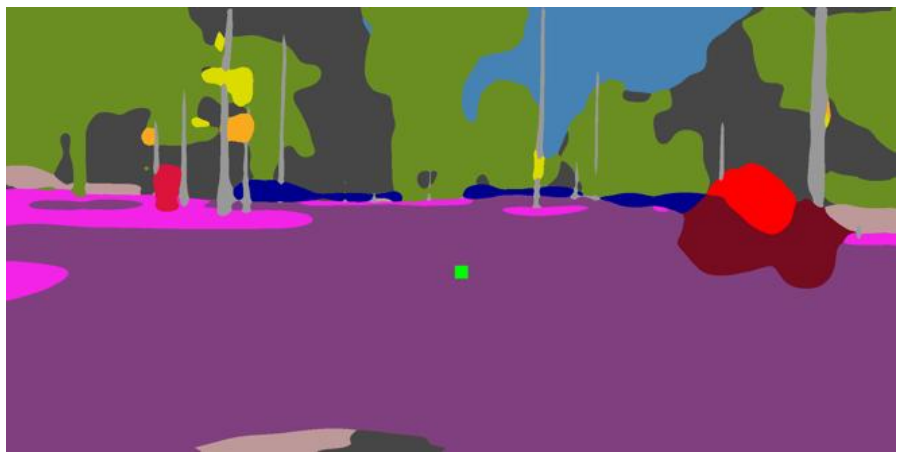
We show pixels with the strongest log-max-softmax gradient (red) in a hand-picked pixel (green)



t-3

t

t + 9



forecast

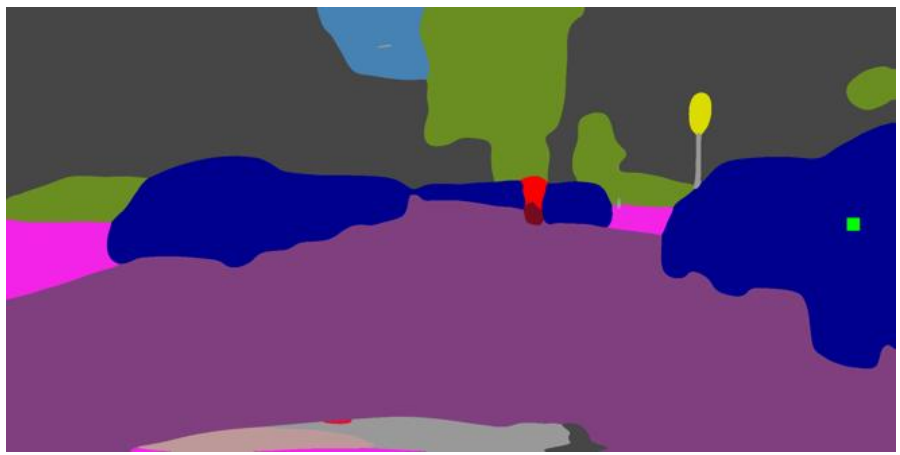
Semantic forecasting: explaining results (3)

We show pixels with the strongest log-max-softmax gradient (red) in a hand-picked pixel (green)

t-3

t

t + 9



forecast

Semantic forecasting: explaining results (4)

We show pixels with the strongest log-max-softmax gradient (red) in a hand-picked pixel (green)

t-3

t

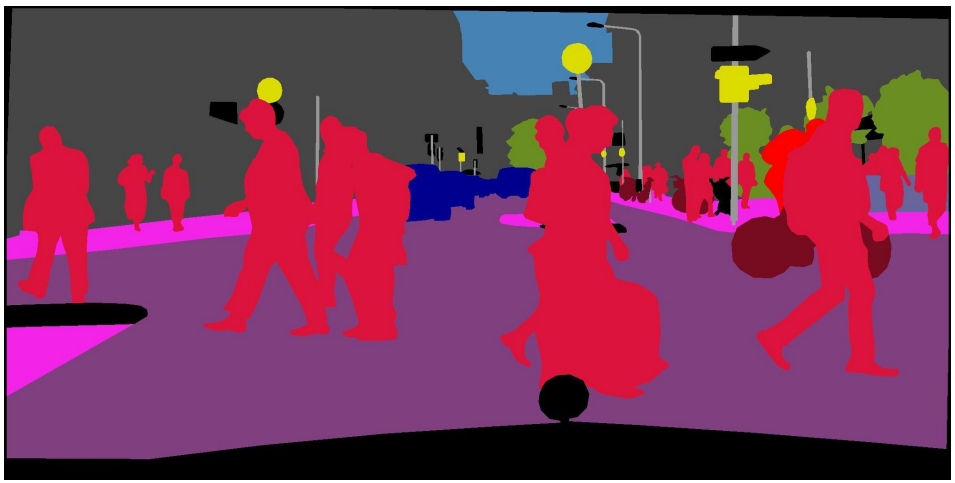
t + 9

forecast

Semantic forecasting: pedestrians (short-term)

most recent input

future frame (unobserved)



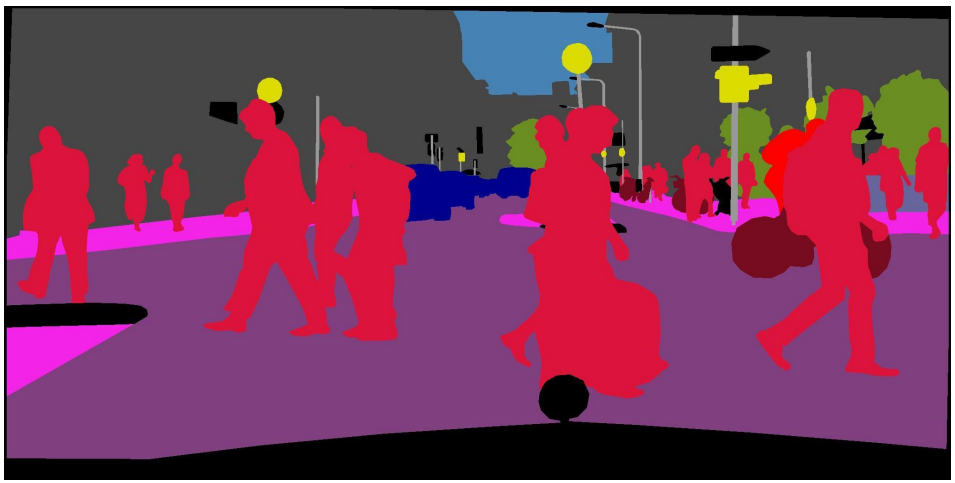
ground truth

short-term forecast

Semantic forecasting: pedestrians (mid-term)

most recent input

future frame (unobserved)



ground truth

mid-term forecast

Semantic forecasting: conclusion

- novel method for anticipating semantic segmentation in driving scenarios based on feature-to-feature forecasting
- we forecast only the most abstract features because of coarse resolution and high semantic content
- we favor deformable convolutions in order to account for geometric nature of F2F forecasting
- due to simplicity our F2F module allows joint fine-tuning with the upsampling path and achieves real-time performance
- state-of-the-art results on Cityscapes mid-term forecast

Conclusion: lightweight models rule

We improve upon the state-of-the-art real-time semantic prediction and forecasting by trading in model capacity

Model capacity can be compensated by careful design:

- residual connections [he15cvpr]
- dws and deformable convolutions [sandler18cvpr,dai17iccv]
- spatial pyramid pooling [zhao17cvpr]]
- pyramidal fusion [orsic19cvpr]
- ladder-style upsampling [kreso17iccvw,lin17cvpr]

Future work:

- custom lightweight architectures for efficient recognition
- efficient architectures for video analysis
- include the remaining ingredients (uncertainty, robustness, ...)

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

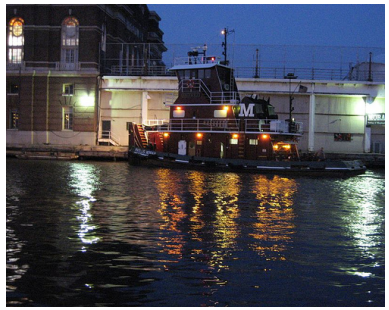


bird



Challenges: false positives due to context (2)

boat



bottle

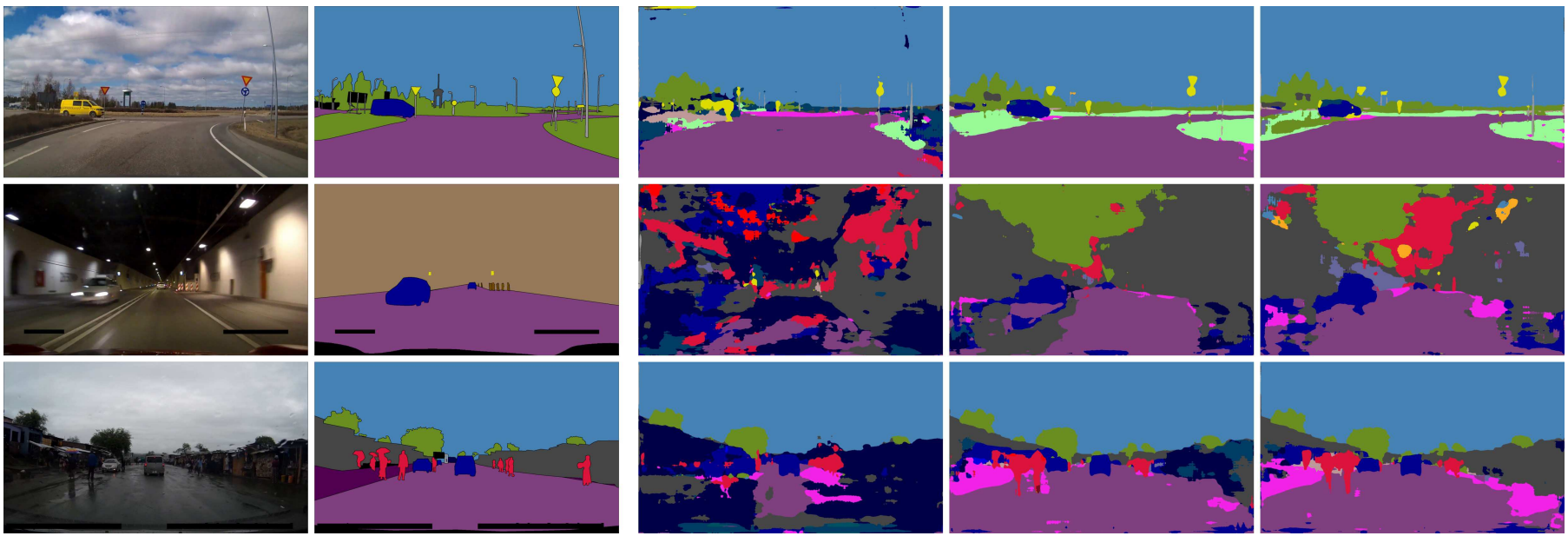
bus

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)



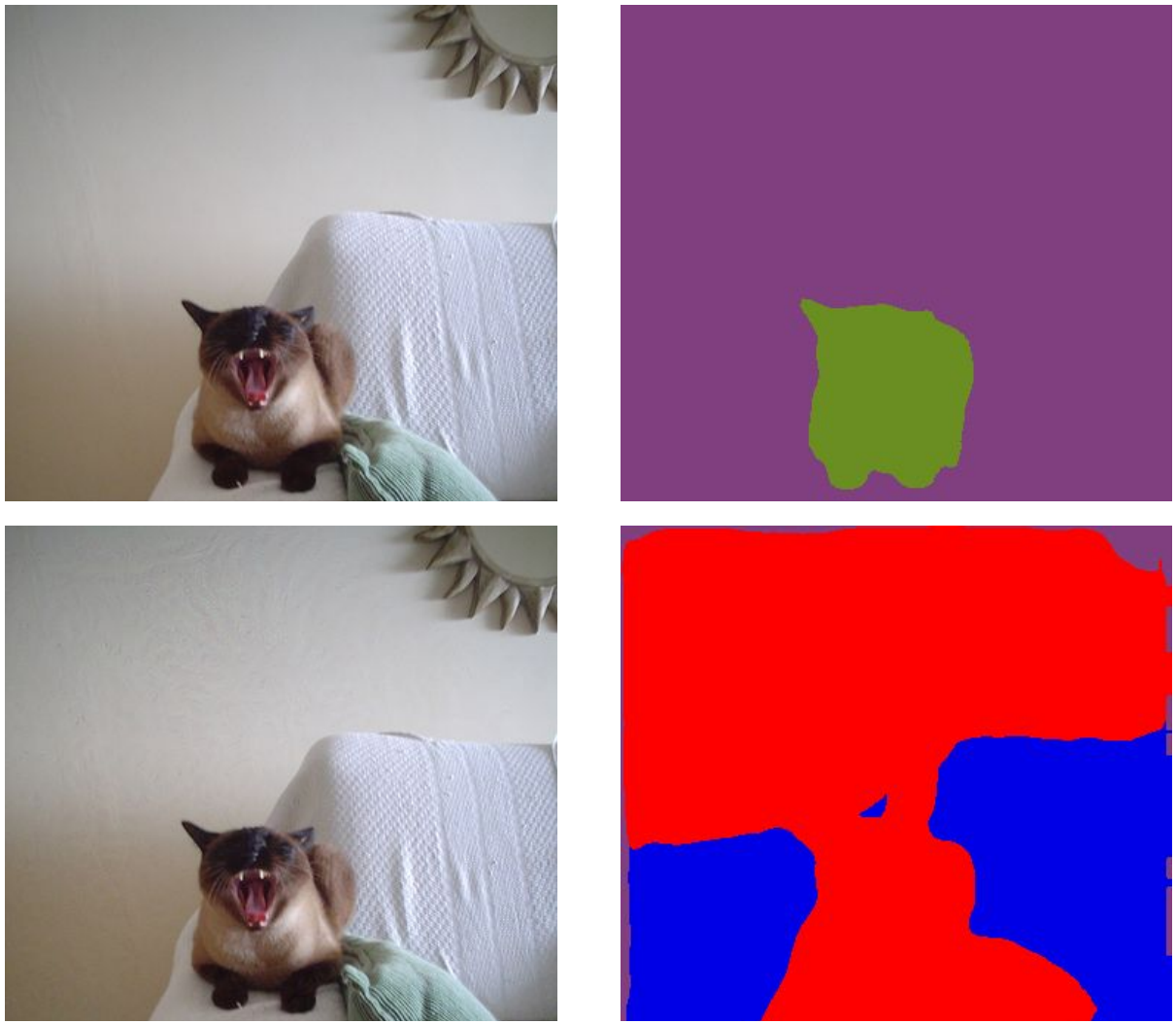
[zendel18eccv]

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

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

Challenges: Adversarial examples

Imperceptible perturbations may invalidate prediction [szegedy14iclr]:



[kreso18ep]

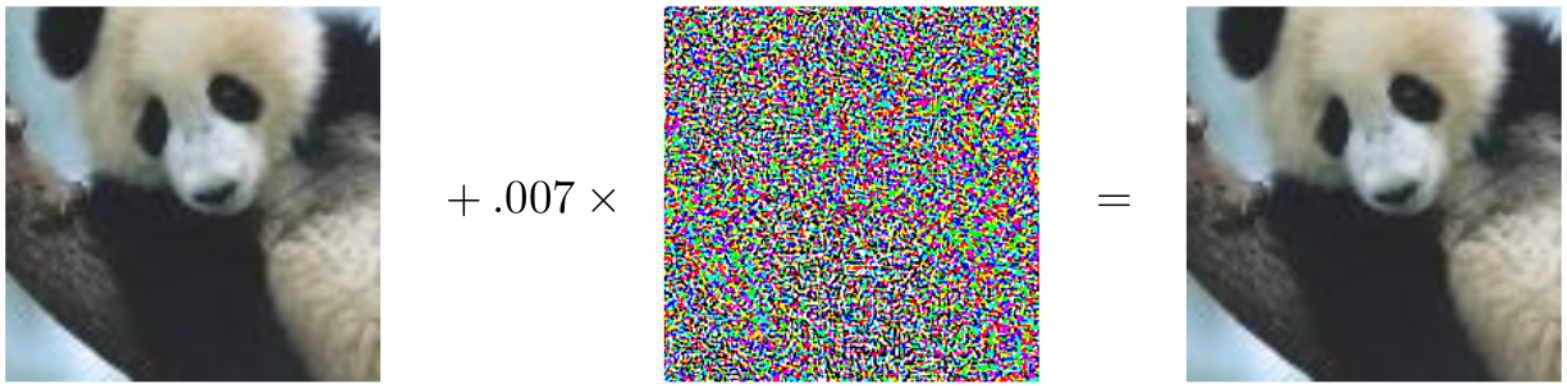
Challenges: Adversarial examples (2)

Adversarial perturbation:

$$\delta = \arg\min_{\delta} p(Y = y_i | \mathbf{x}_i + \delta, \Theta)$$

Adversarial example: $\mathbf{x}_i + \delta$

Existence of adversarial examples suggests that current vision systems are *free-riding* on **easy features** while ignoring the gist of the scene



$\boldsymbol{x}$

"panda"
57.7% confidence

$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$

"nematode"
8.2% confidence

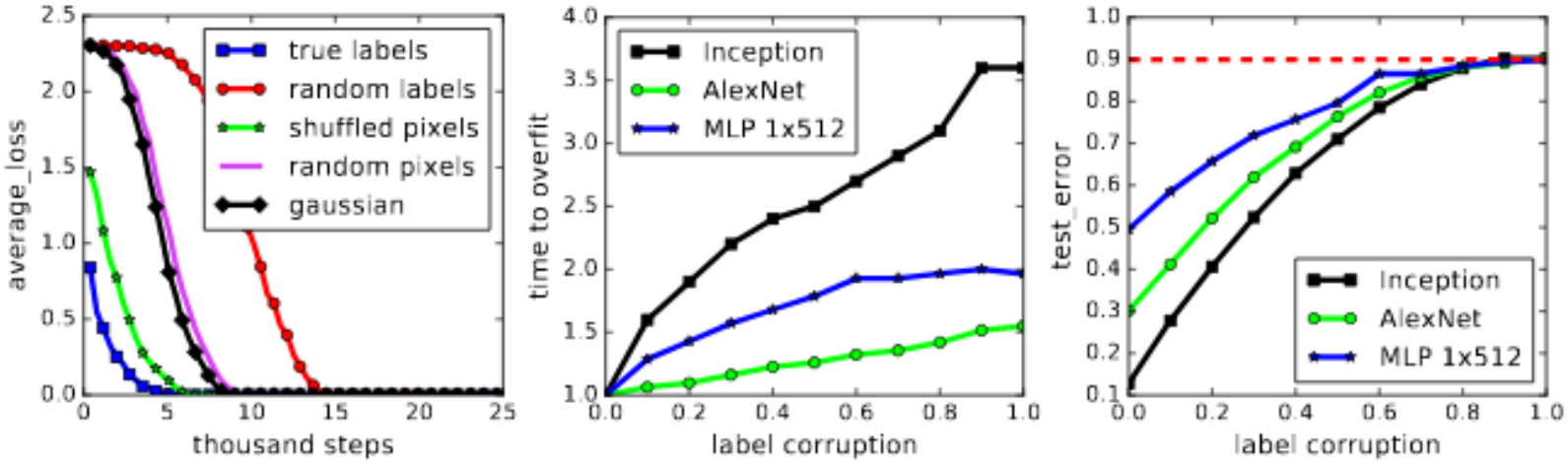
$\boldsymbol{x} + \epsilon \text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$

"gibbon"
99.3 % confidence

[goodfellow15iclr]

Challenges: Theory

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



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

[zhang17iclr]

In simple words, the model is able to memorize the entire training data

Yet, the models generalize well when trained on well sorted data

A theory to explain this behaviour is missing.

Challenges: solutions

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

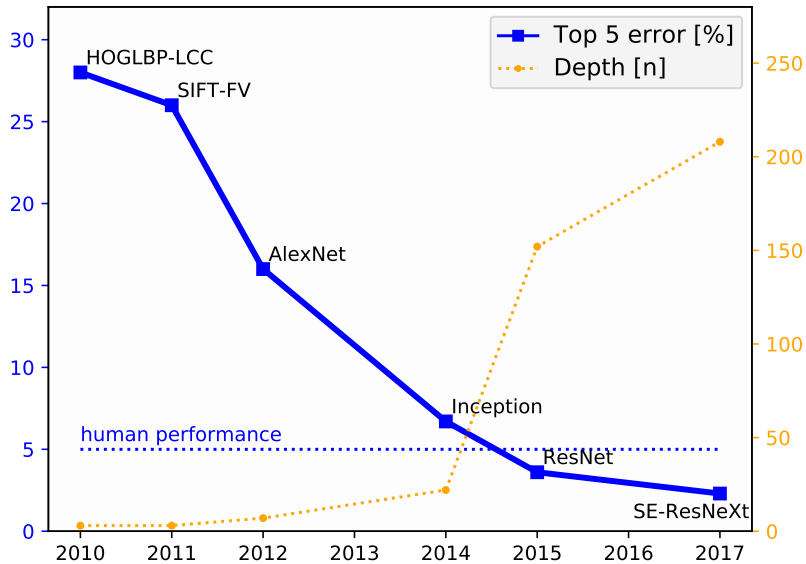
- are unable to comprehend limits of their expertise
- under-achieve by relying on **easy** but **non-robust** features
  - humans also jump to conclusions: e.g. a person without a wedding ring is considered single
  - such inferrence is not appropriate for mission critical systems

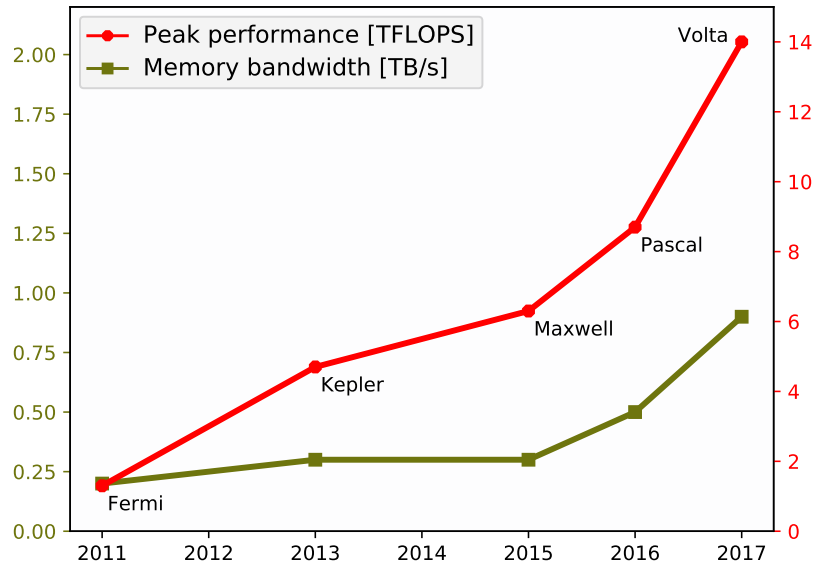
Prominent ways for getting closer to truly intelligent artificial vision:

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

Challenges: izgledi

Točnost sustava računalnog vida i dalje će rasti:





Challenges: 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 I-2433-2014 MultiCLoD.

`http://multiclod.zemris.fer.hr`