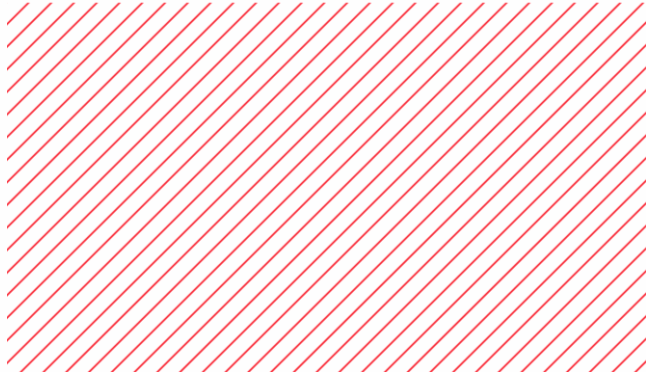
академия больших данных

mail.ru group

Глобальные дескрипторы

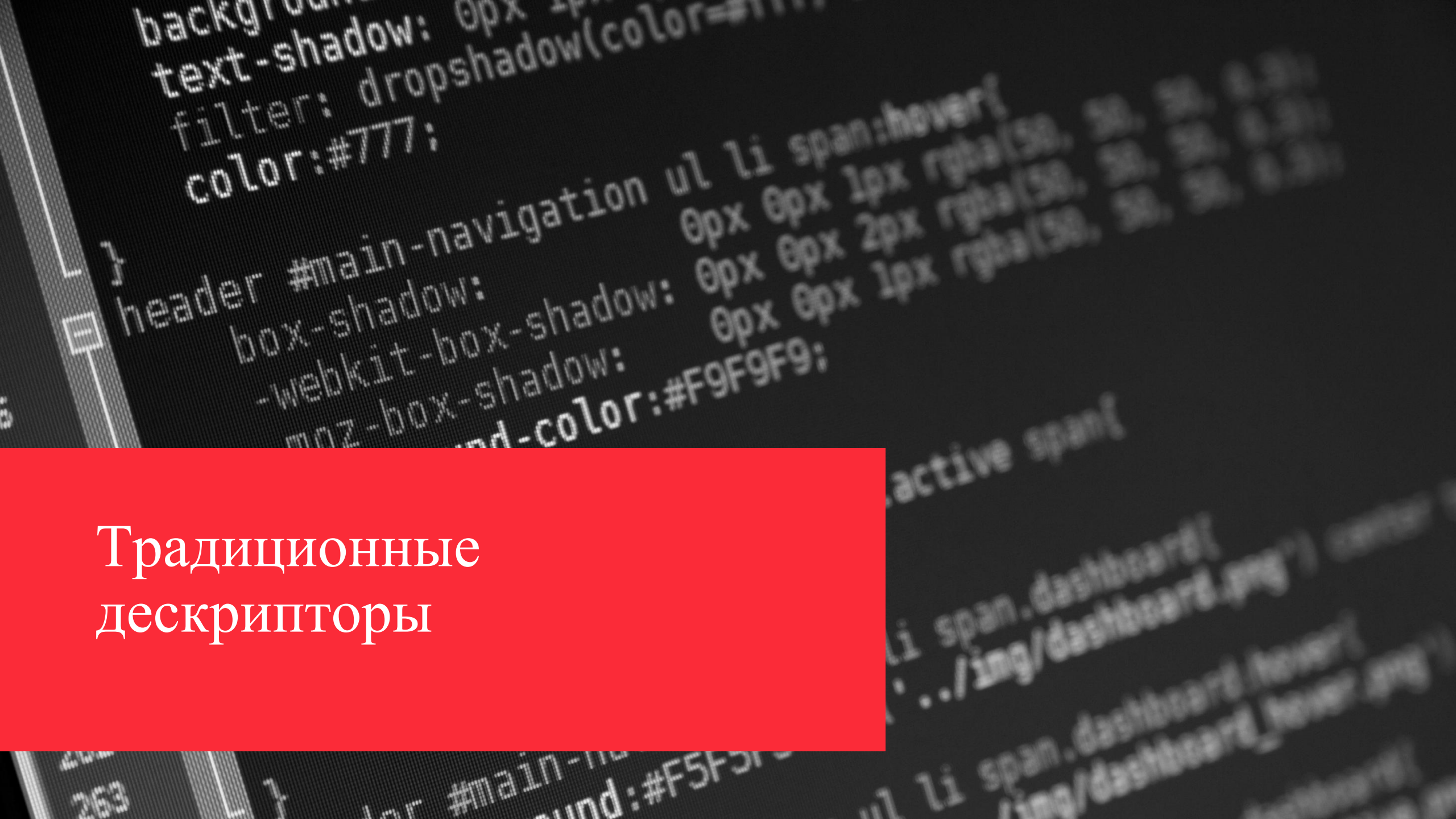
Андрей Савченко

Профессор НИУ ВШЭ-Нижний Новгород



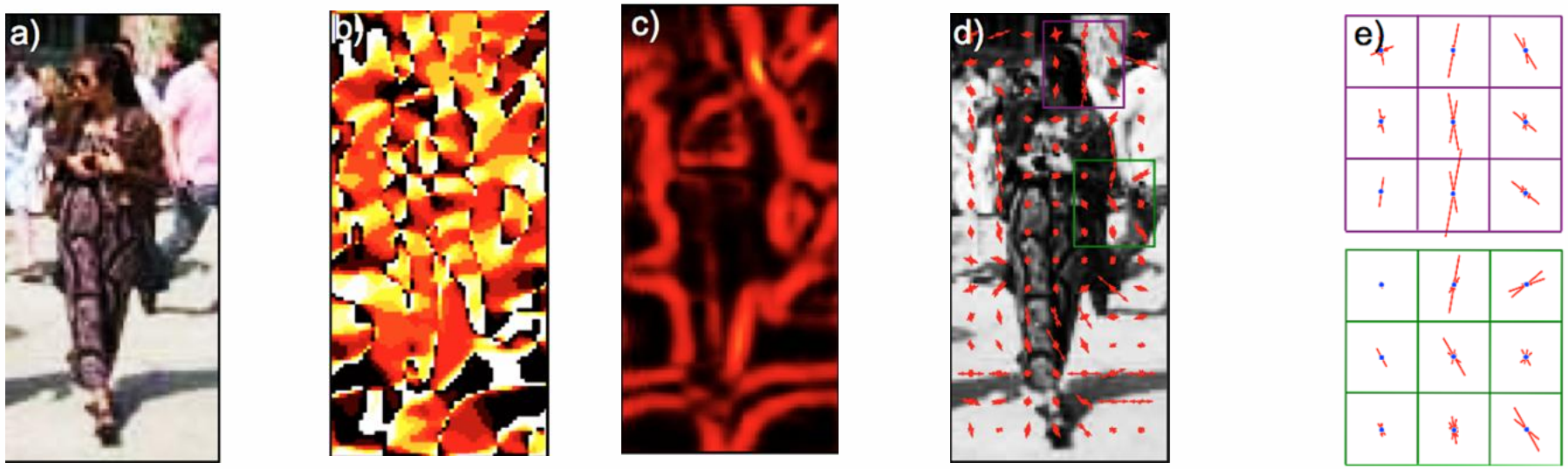
Опрос

https://forms.gle/bfE8QZkMCfC6f4h9A



Традиционные дескрипторы

Глобальный дескриптор HOG (Histogram of Oriented Gradients)



b) Ориентация градиента (квантованная на 9 блоков для 180 градусов)

c) Амплитуда градиента

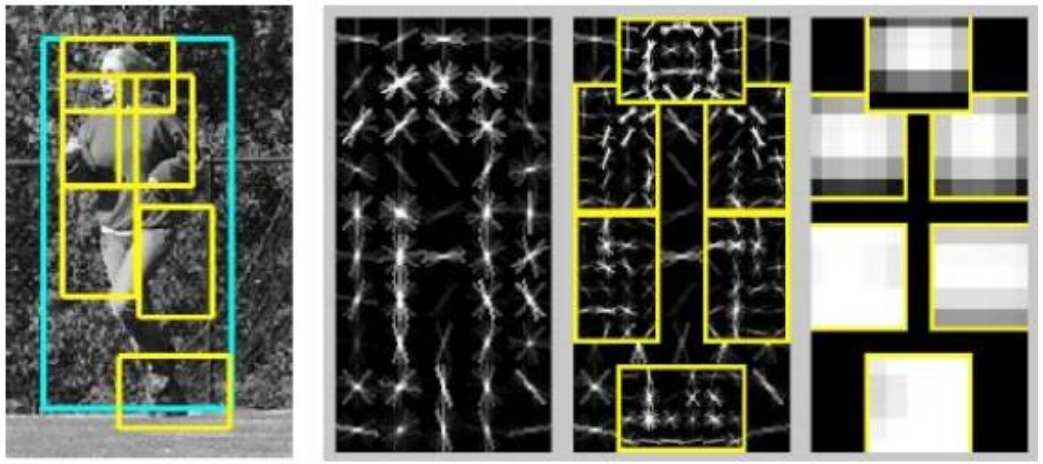
d) Дескрипторы ячеек 6x6 пикселей – взвешенные гистограммы ориентаций

e) Дескриптор блока – нормированный объединенный вектор дескрипторов смежных 3x3 ячеек

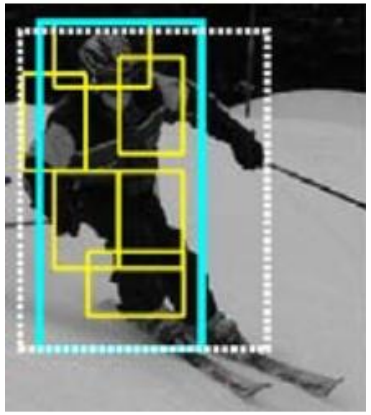
Итоговый дескриптор HOG для изображения 64x128 – объединение дескрипторов блоков

Part-based object detection

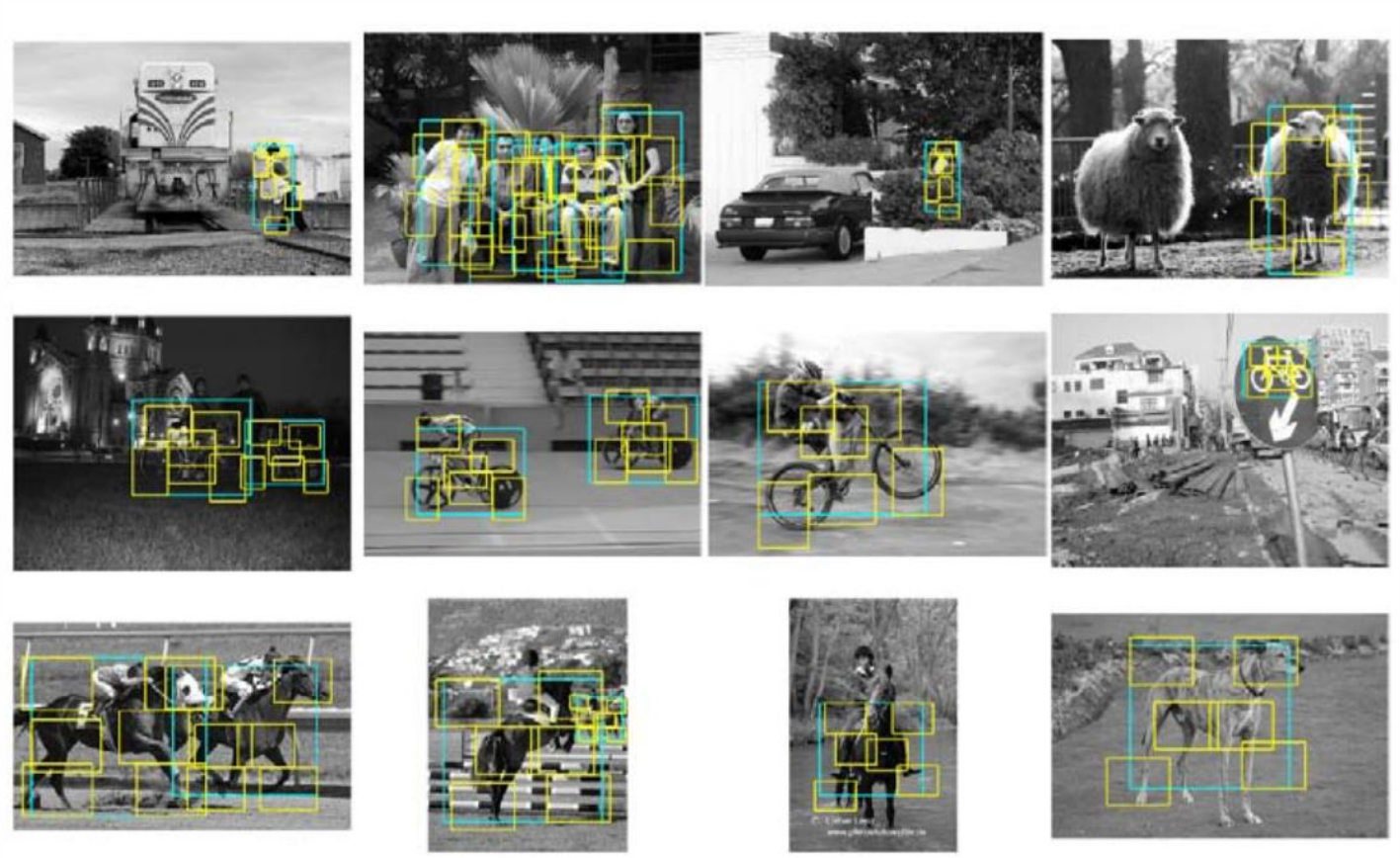
Обучаемые расположения частей относительно всего объекта



True positive



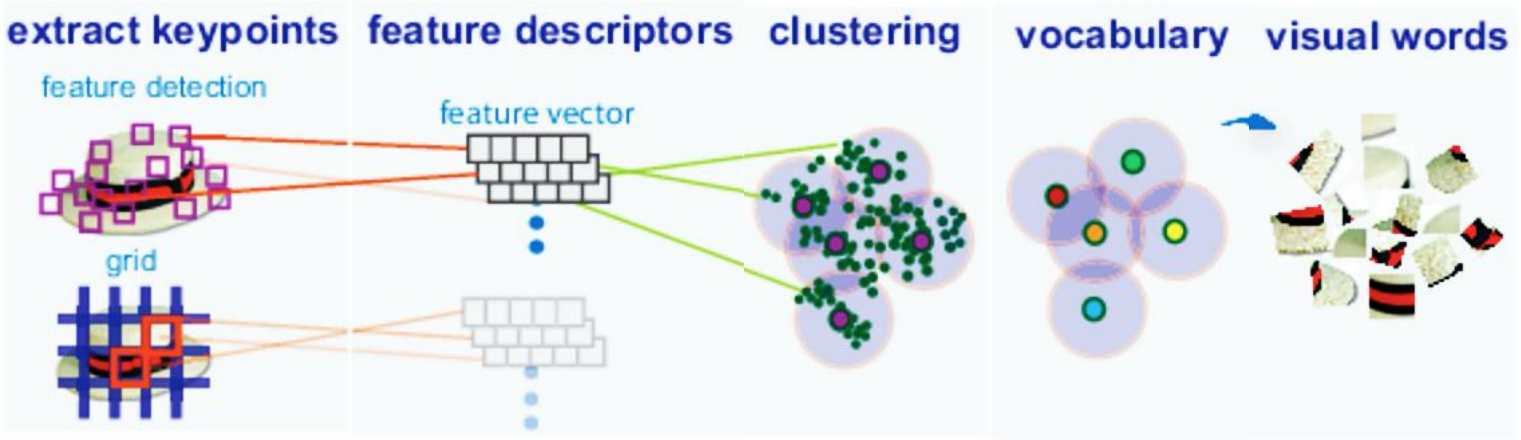
False positive



Felzenszwalb, McAllester, and Ramanan 2008

Visual words/Bag-of-words (BoW)/Bag-of-features (BoF)

Формирование «словаря»

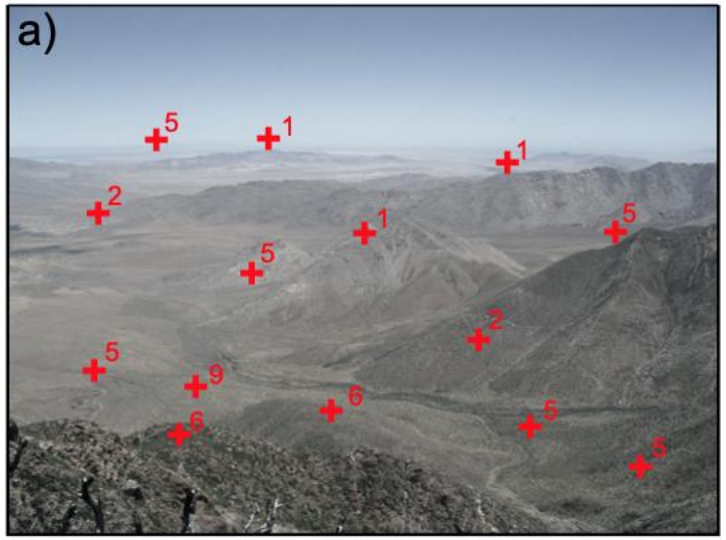


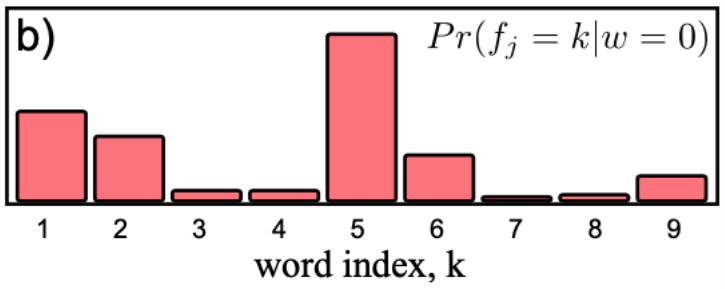
Кодирование изображения

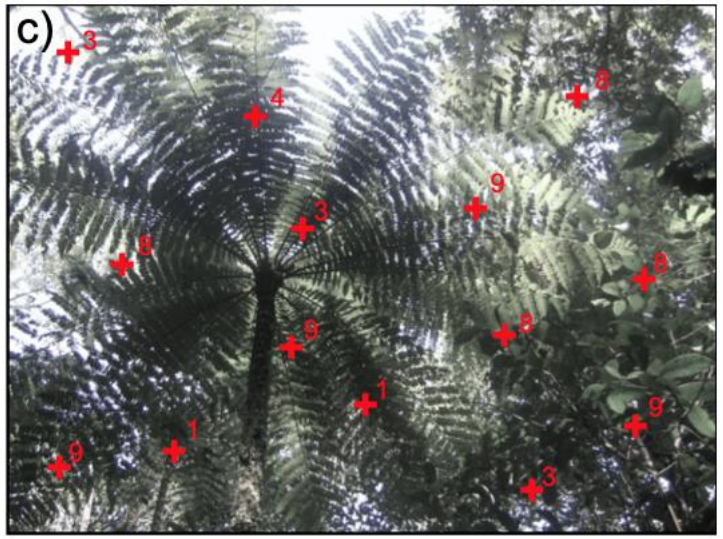
OpenCV: cv::BOWImgDescriptorExtractor

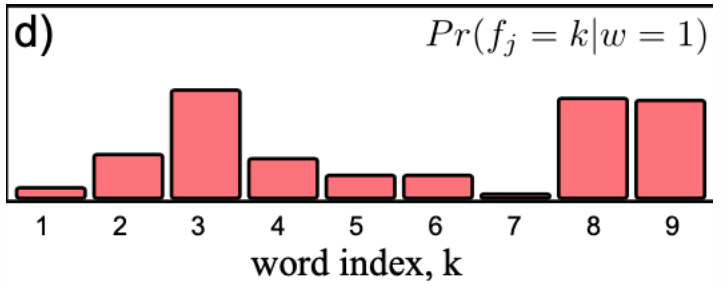
https://www.mathworks.com/help/vision/ug/image-classification-with-bag-of-visual-words.html

Bag-of-words. Примеры



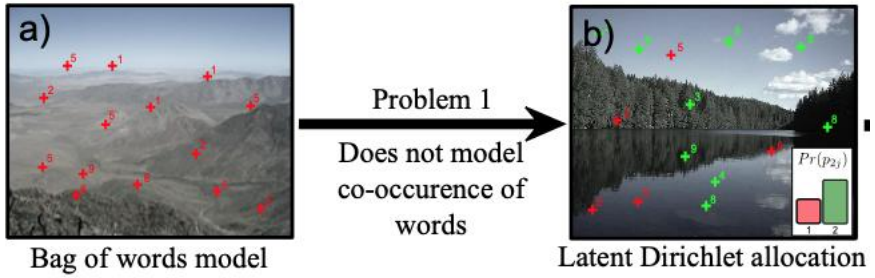




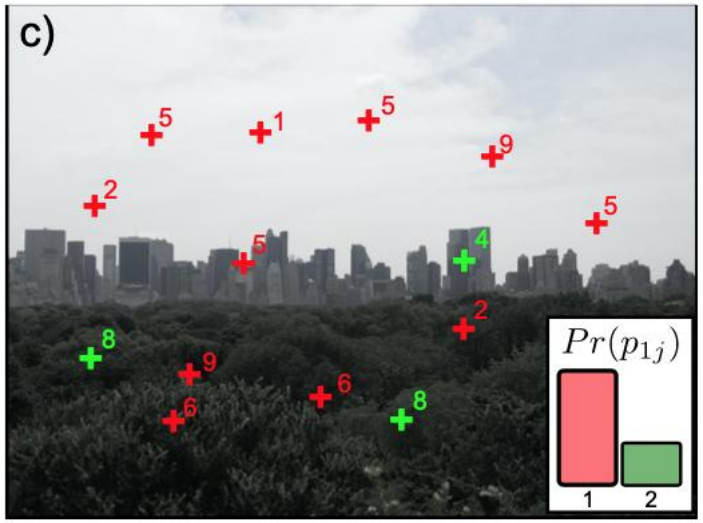


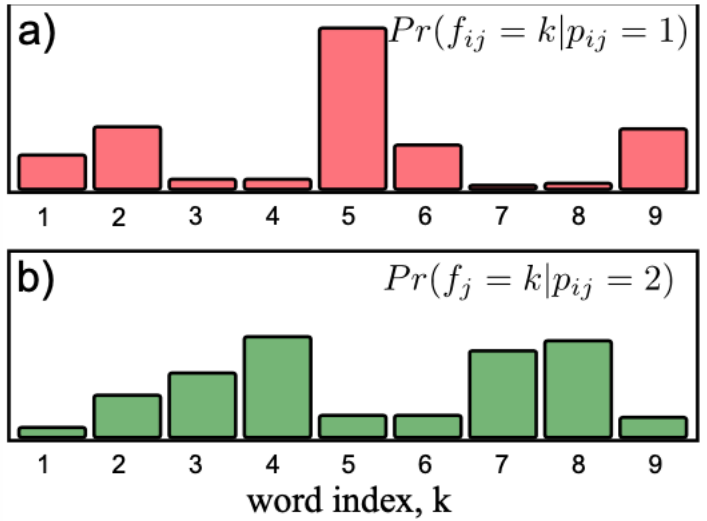
Prince «Computer vision: models, learning and inference»

Latent Dirichlet allocation (LDA)



В LDA каждое «слово» может принадлежать к одной из *M* частей, для каждой – свое распределение

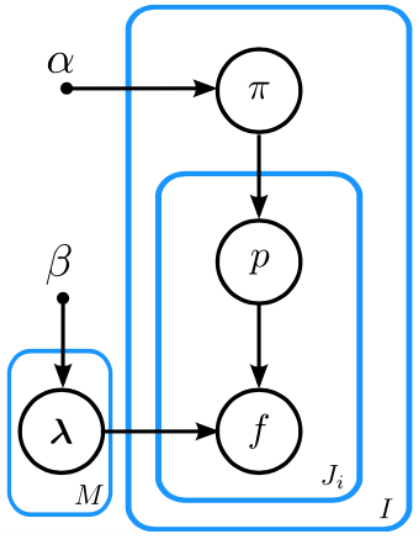




$$Pr(f_{ij}) = \sum_{m=1}^{M} Pr(f_{ij}|p_{ij} = m)Pr(p_{ij} = m).$$

Probability of observing each of K possible words under part label m

Обучение LDA – с помощью EM-алгоритма



Probability of selecting each of M parts in i^{th} image

Part label for j^{th} word in i^{th} image

Word label for j^{th} feature in i^{th} image

p_{ij} – скрытые переменные (**слова не независимы, как в BoW**)

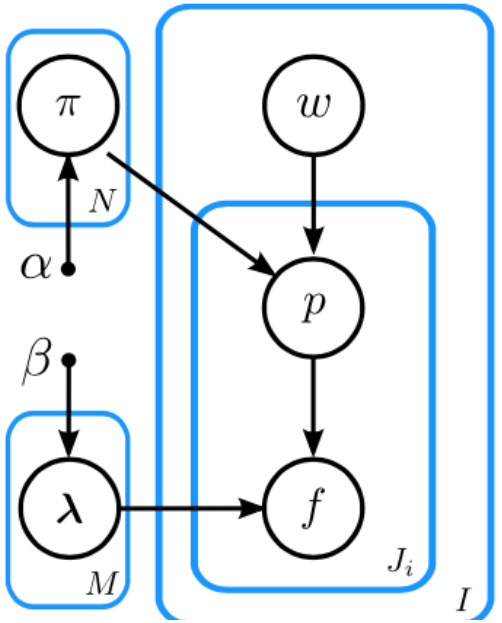
Prince «Computer vision: models, learning and inference»

Single author-topic

Предполагается, что на изображении находится ровно один объект из N возможных. Его метка (неизвестная) w_i

Probability of selecting m^{th} part when n^{th} object is present

Probability of observing each of K possible words under part label m



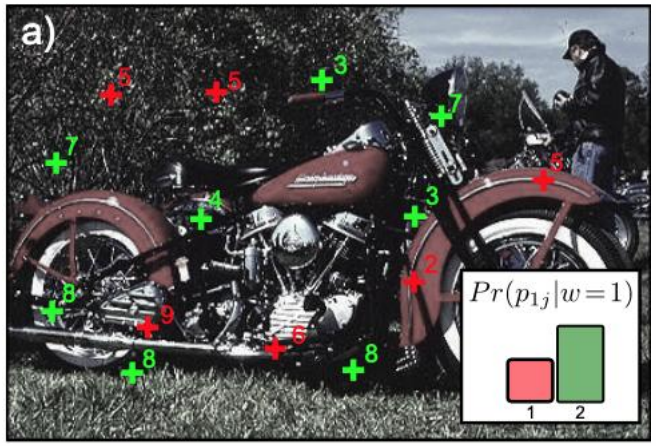
Object label for i^{th} image

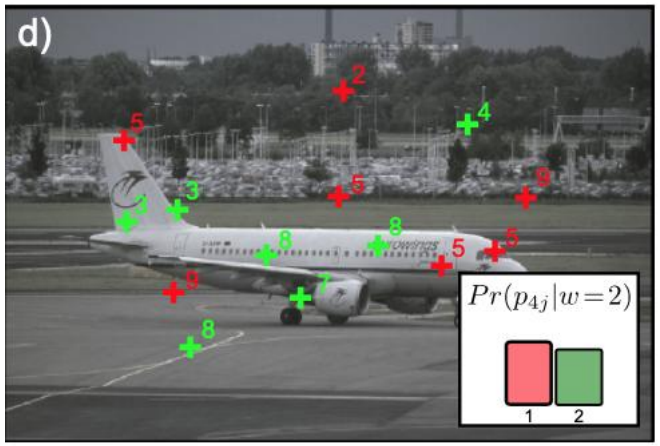
Part label for j^{th} word in i^{th} image

Word label for j^{th} feature in i^{th} image

$$Pr(\mathbf{f}|w=n) = \prod_{j=1}^{J}\sum_{p_j=1}^{M} Pr(p_j|w=n)Pr(f_j|p_j)$$

$$= \prod_{j=1}^{J}\sum_{p_j=1}^{M} \mathrm{Cat}_{p_j}[\boldsymbol{\pi}_n]\mathrm{Cat}_{f_j}[\boldsymbol{\lambda}_{p_j}]$$

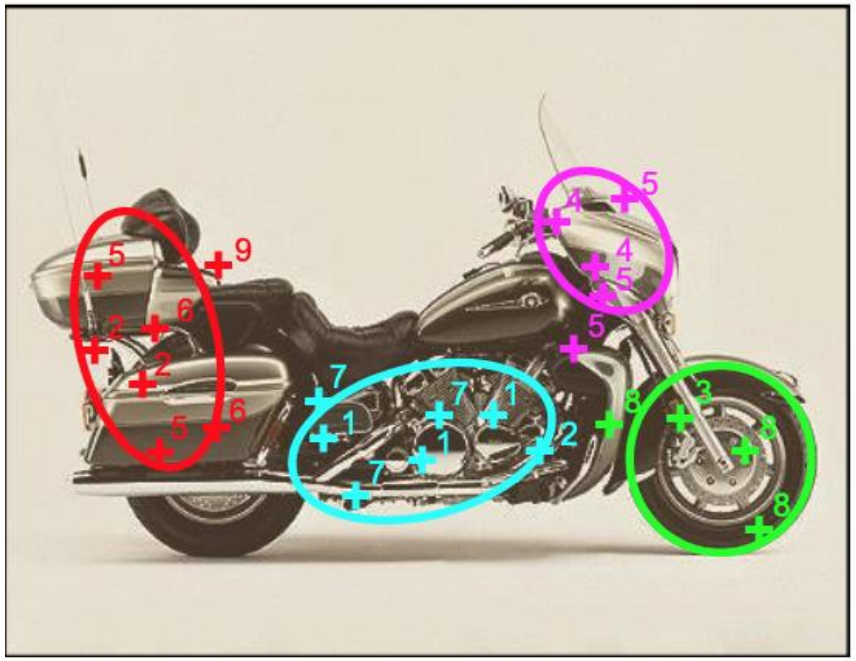


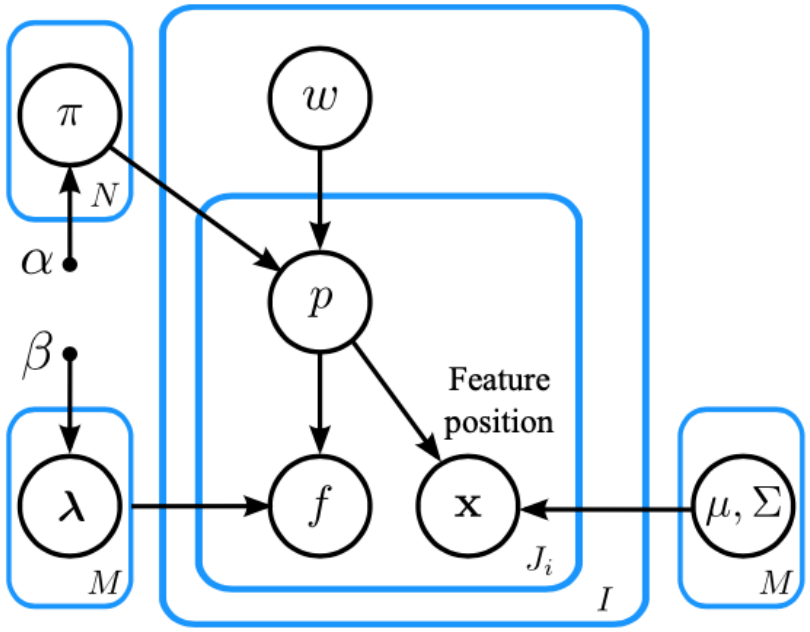


Prince «Computer vision: models, learning and inference»

Part-based recognition: constellation («Созвездие»)

Особые точки частей объекта расположены рядом



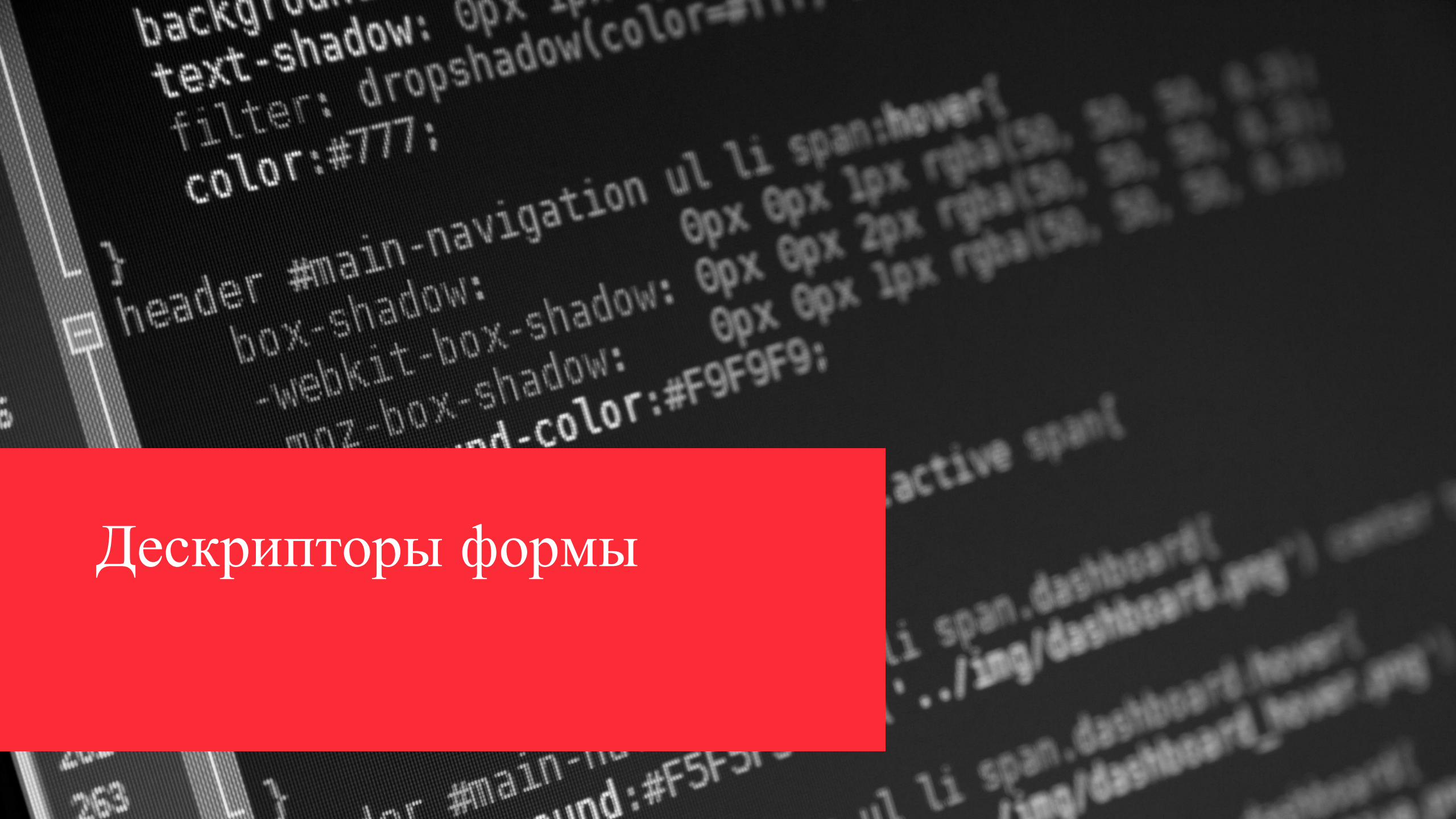


$$Pr(\mathbf{f}, \mathbf{X}|w=n) = \prod_{j=1}^{J}\sum_{m=1}^{M} Pr(p_j=m|w=n)Pr(f_j|p_j=m)Pr(\mathbf{x}_j|p_j=m)$$

$$= \prod_{j=1}^{J}\sum_{p_j=1}^{M} \mathrm{Cat}_{p_j}[\boldsymbol{\pi}_n]\mathrm{Cat}_{f_j}[\boldsymbol{\lambda}_{p_j}]\mathrm{Norm}_{\mathbf{x}_{ij}}[\boldsymbol{\mu}_{p_j}, \boldsymbol{\Sigma}_{p_j}]. \qquad (20.29)$$

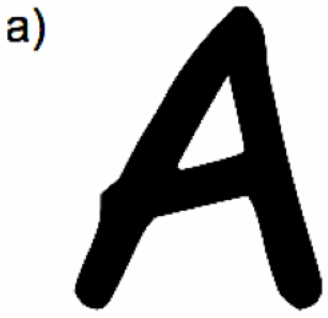
Prince «Computer vision: models, learning and inference»

(Fergus, Perona, and Zisserman 2007)

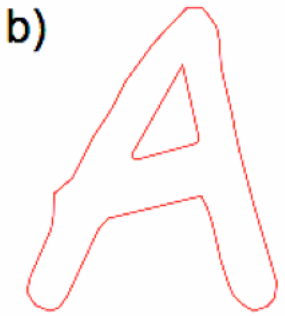


Дескрипторы формы

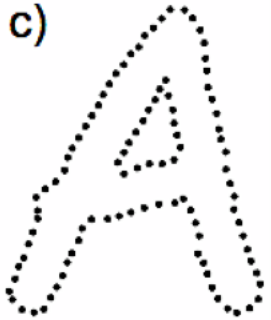
Дескриптор формы (shape context descriptor)



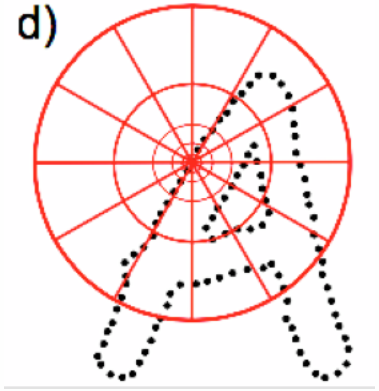
Исходный силуэт



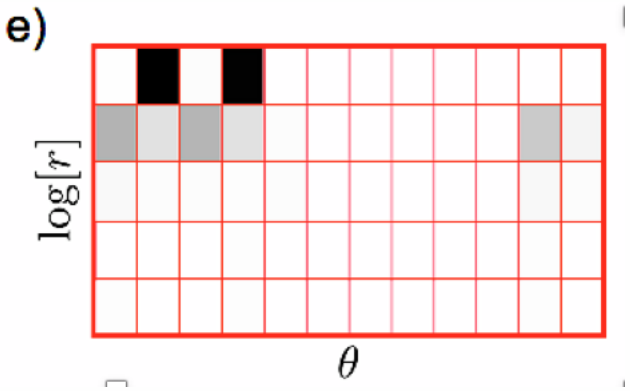
Контур



Точки, расположенные на одинаковых расстояниях на контуре



Логарифм расстояний до каждой точки (в полярных координатах)

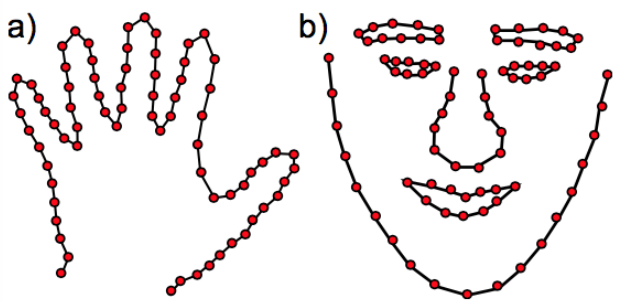


Вычисляется гистограмма числа точек, попавших в ячейку (зависит от логарифма расстояний и угла)
Дескриптор – объединение гистограмм для всех точек

Prince «Computer vision: models, learning and inference»

Параметрические контурные модели формы (1)

Представление формы.
Landmark points W=[w_1,...,w_N]



Точки близки к краям (edge)

$$Pr(\mathbf{x}|\mathbf{W}) \propto \prod_{n=1}^{N} \exp\left[-(\text{dist}\left[\mathbf{x}, \mathbf{w}_n\right])^2\right]$$

$$\hat{\mathbf{W}} = \underset{\mathbf{W}}{\text{argmax}}\left[Pr(\mathbf{W}|\mathbf{x})\right] = \underset{\mathbf{W}}{\text{argmax}}\left[Pr(\mathbf{x}|\mathbf{W})Pr(\mathbf{W})\right]$$

$$Pr(\mathbf{W}) \propto \prod^{N} \exp\left[\alpha\,\text{space}[\mathbf{w}, n] + \beta\,\text{curve}[\mathbf{w}, n]\right]$$

$$\text{space}[\mathbf{w}, n] = -\left(\frac{\sum_{n=1}^{N}\sqrt{(\mathbf{w}_n - \mathbf{w}_{n-1})^T(\mathbf{w}_n - \mathbf{w}_{n-1})}}{N} - \sqrt{(\mathbf{w}_n - \mathbf{w}_{n-1})^T(\mathbf{w}_n - \mathbf{w}_{n-1})}\right)^2, \quad (17.5)$$

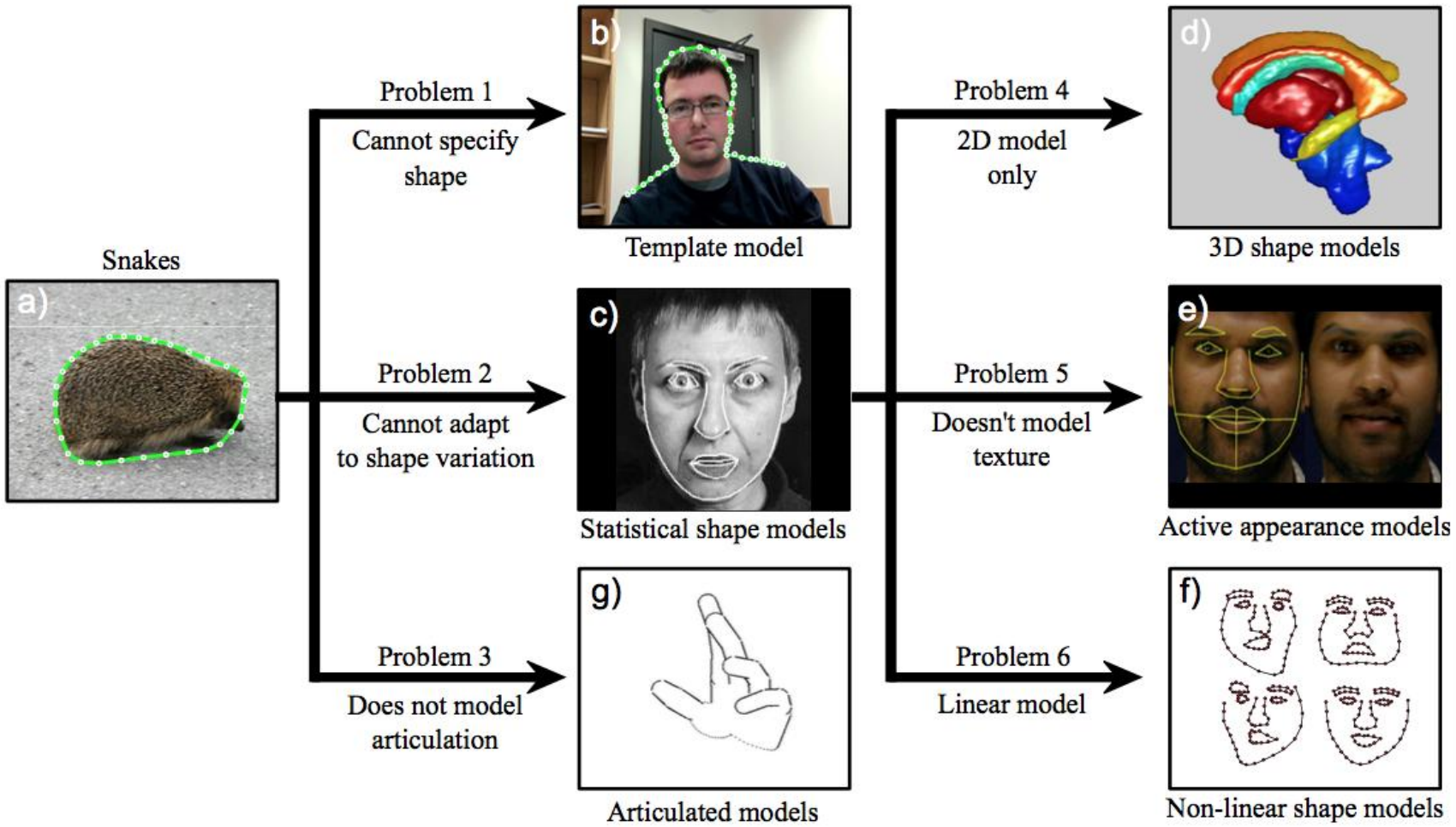
– одинаковое расстояние между точками

$$\text{curve}[\mathbf{w}, n] = -(\mathbf{w}_{n-1} - 2\mathbf{w}_n + \mathbf{w}_{n+1})^T(\mathbf{w}_{n-1} - 2\mathbf{w}_n + \mathbf{w}_{n+1})$$

– контур должен быть гладким

Prince «Computer vision: models, learning and inference»

Параметрические контурные модели формы (2)

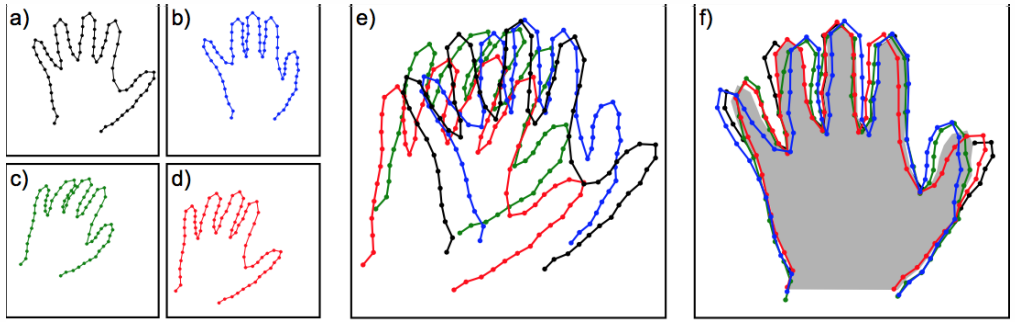


Prince «Computer vision: models, learning and inference»

Active shape model

$$\mathbf{w}_i = \boldsymbol{\mu} + \boldsymbol{\Phi}\mathbf{h}_i + \epsilon_i \qquad Pr(\mathbf{w}_i) = \int Pr(\mathbf{w}_i|\mathbf{h}_i)Pr(\mathbf{h}_i)d\mathbf{h}_i = \text{Norm}_{\mathbf{w}_i}[\boldsymbol{\mu}, \boldsymbol{\Phi}\boldsymbol{\Phi}^T + \sigma^2\mathbf{I}].$$

Выравнивание контуров в обучающем множестве (Procrustes analysis)



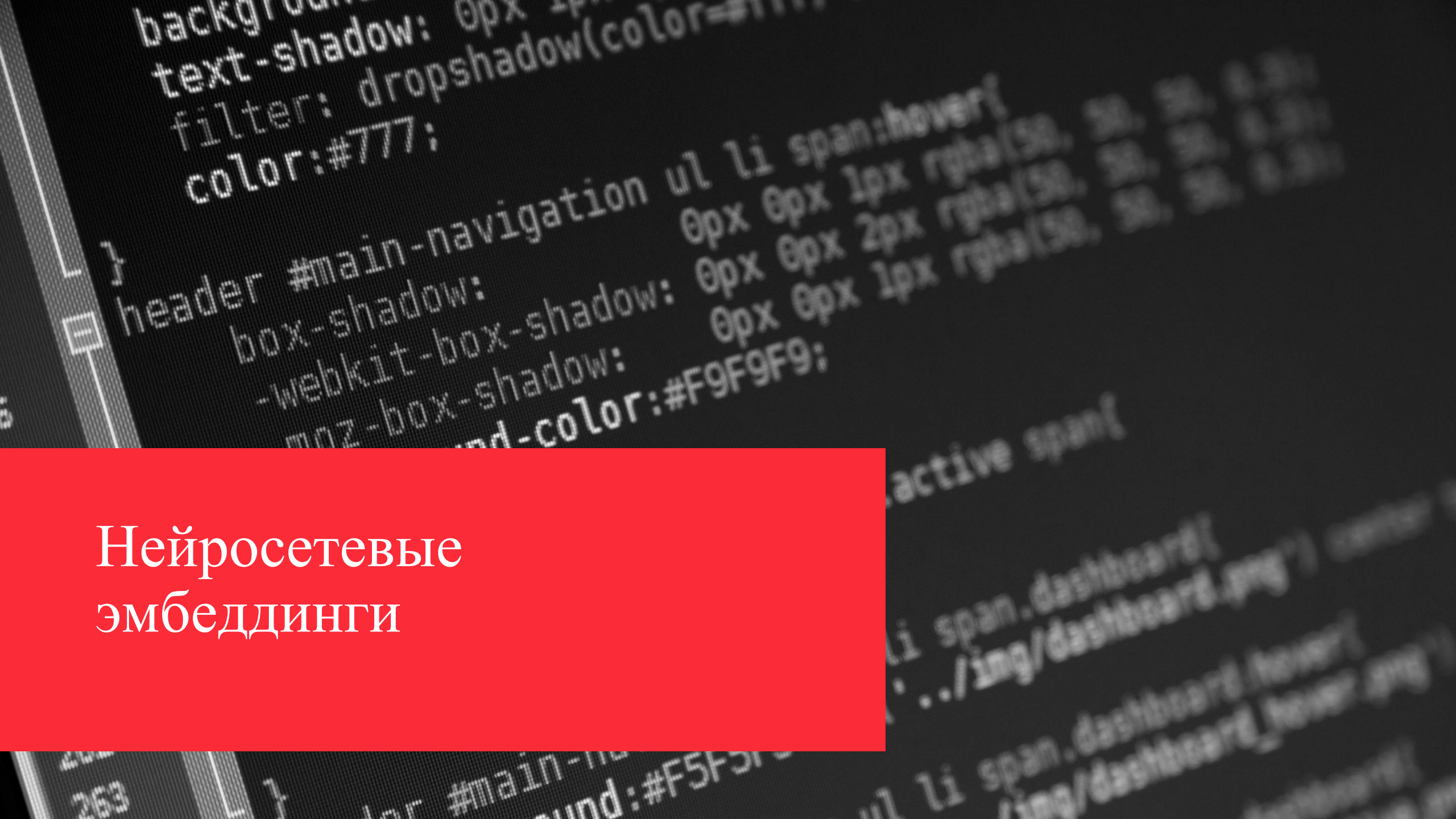
Iterative closest point (ICP) – итеративный поиск ближайших точек контура $\mathbf{y}_n$

$$\hat{\mathbf{h}} = \underset{\mathbf{h}}{\text{argmax}}\left[\max_{\boldsymbol{\Psi}}\left[\sum_{n=1}^{N}\left(-\frac{(\text{dist}\left[\mathbf{x}_i, \mathbf{trans}[\boldsymbol{\mu}_n + \boldsymbol{\Phi}_n\mathbf{h}, \boldsymbol{\Psi}]\right])^2}{\sigma^2}\right) + \log[\text{Norm}_{\mathbf{h}}[\mathbf{0}, \mathbf{I}]]\right]\right]$$

$$\hat{\mathbf{h}} = \underset{\mathbf{h}}{\text{argmax}}\left[\sum_{n=1}^{N}\log[Pr(\mathbf{y}_n|\mathbf{h}), \boldsymbol{\Psi}] + \log[Pr(\mathbf{h})]\right]$$

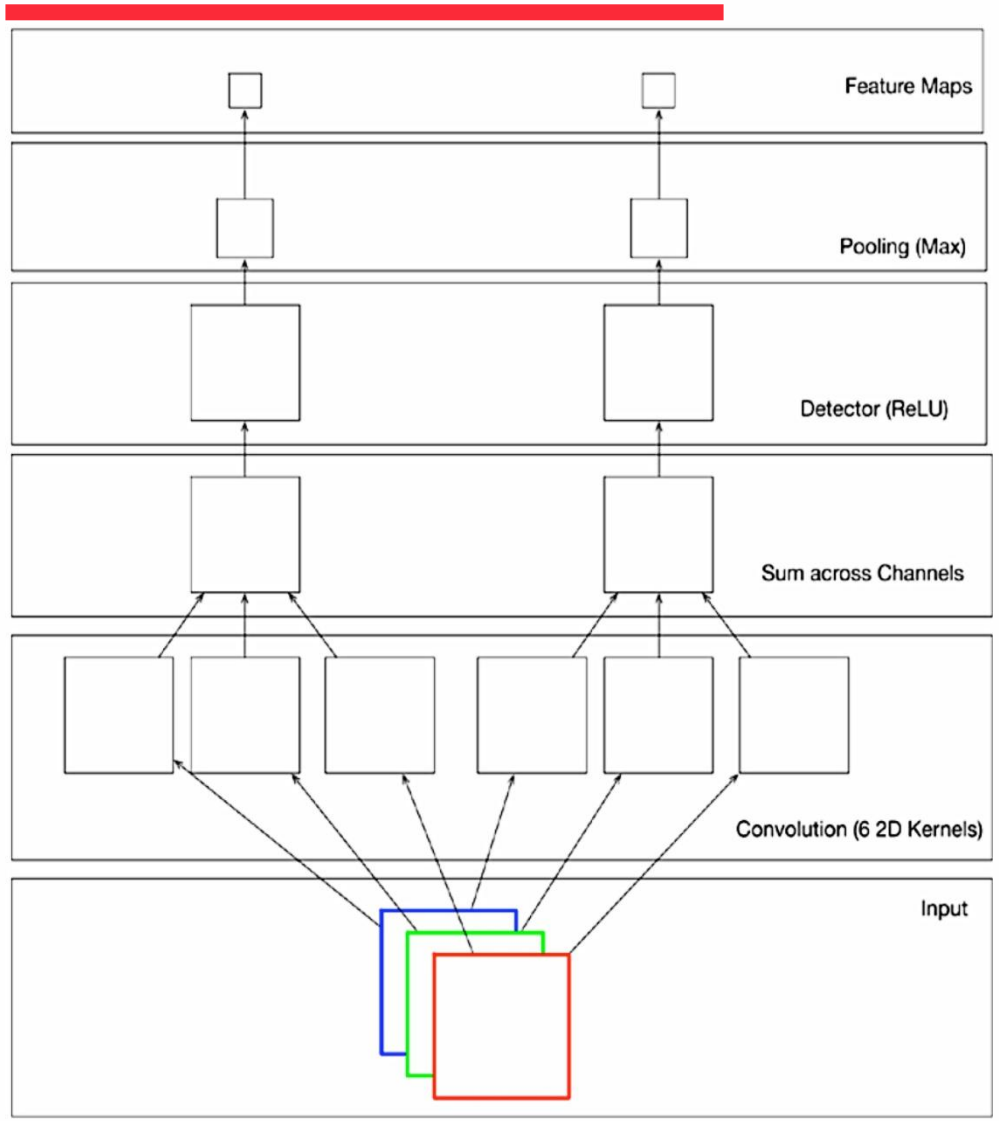
$$= \underset{\mathbf{h}}{\text{argmax}}\left[\sum_{n=1}^{N} -\left(\mathbf{y}_n - \text{trans}[\boldsymbol{\mu}_n + \boldsymbol{\Phi}_n\mathbf{h}, \boldsymbol{\Psi}]\right)^2/\sigma^2 - \log[\mathbf{h}^T\mathbf{h}]\right]$$

Prince «Computer vision: models, learning and inference»

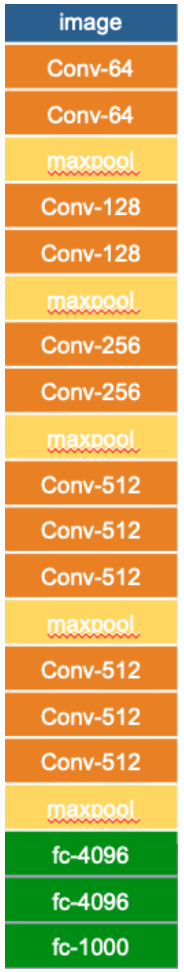


Нейросетевые эмбеддинги

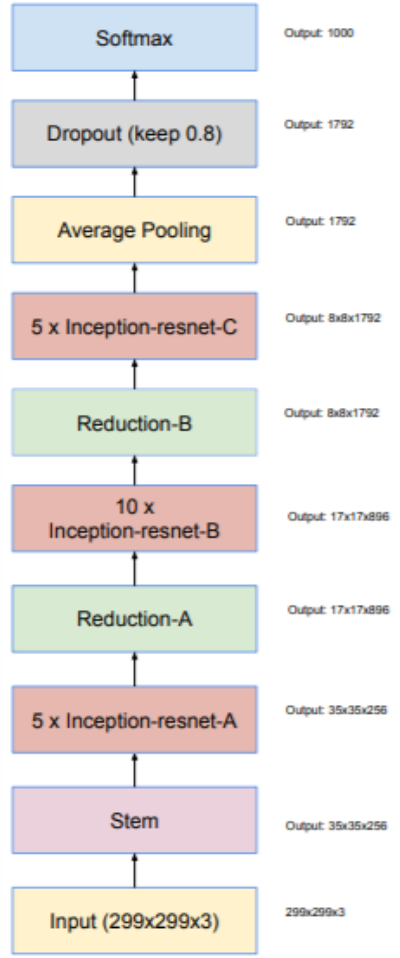
Convolutional Neural Networks (CNN)



VGGNet
(16-19 layers, ~500 Мб)



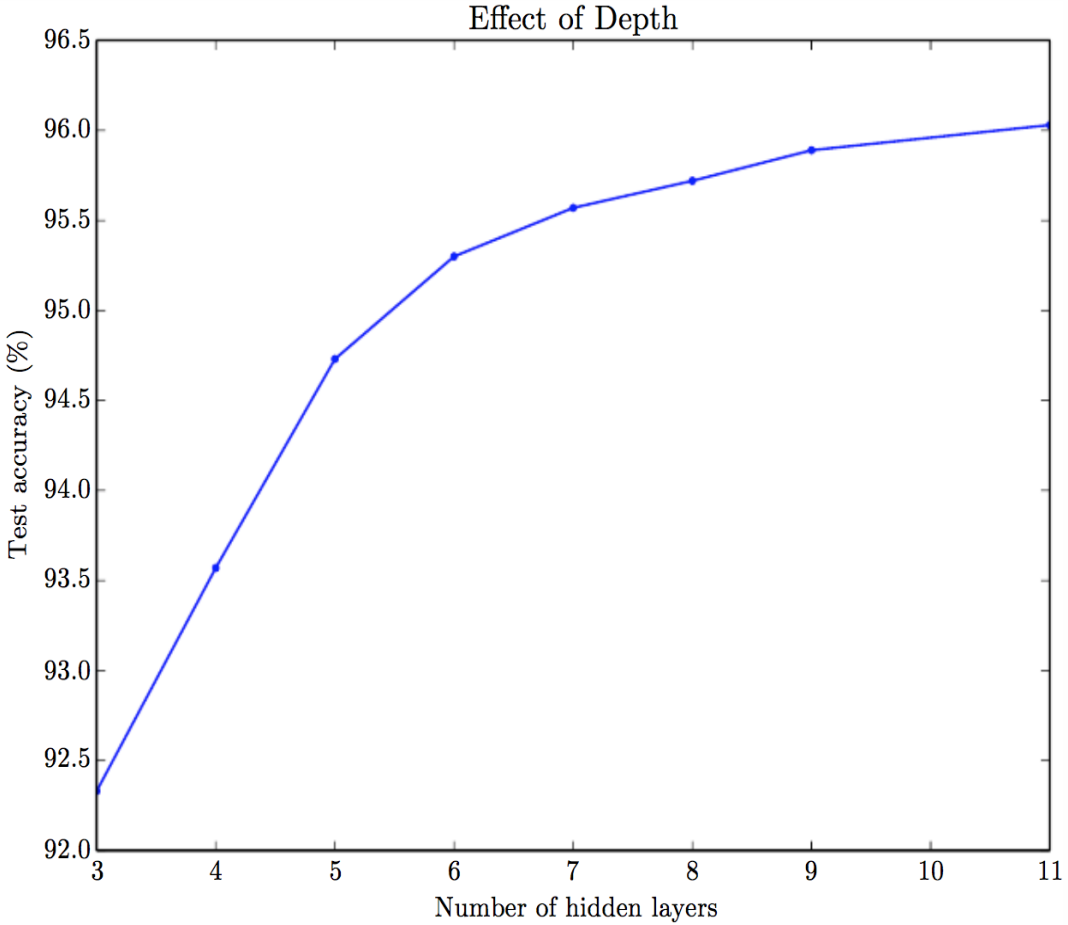
InceptionResNet
(550 layers)

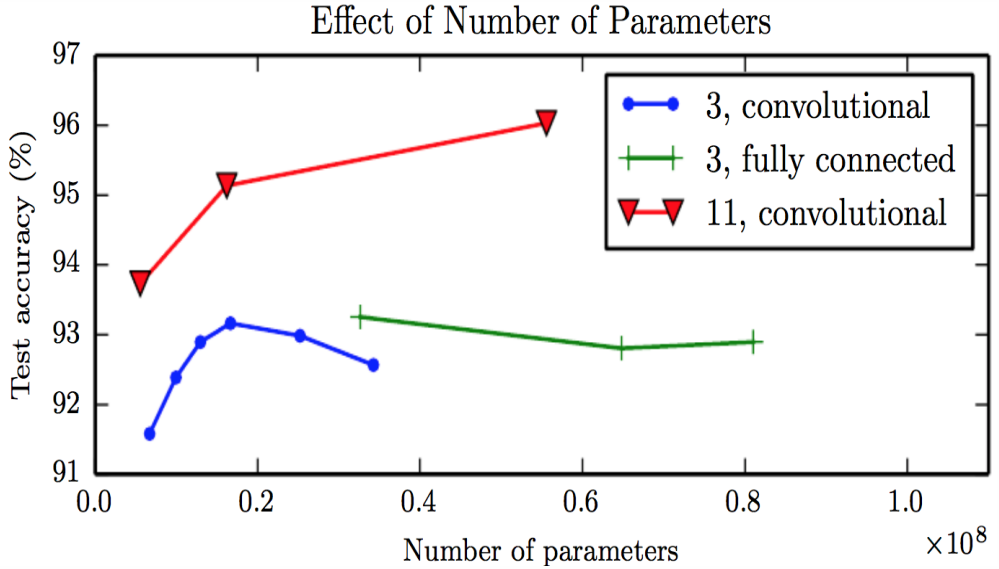


The more the better (1)

Для теоремы универсальной аппроксимации может потребоваться экспоненциально большое число нейронов!

Пример. Классификация изображений номеров домов





Goodfellow I. Bengio Y., Courville A. Deep learning

The more the better (2)

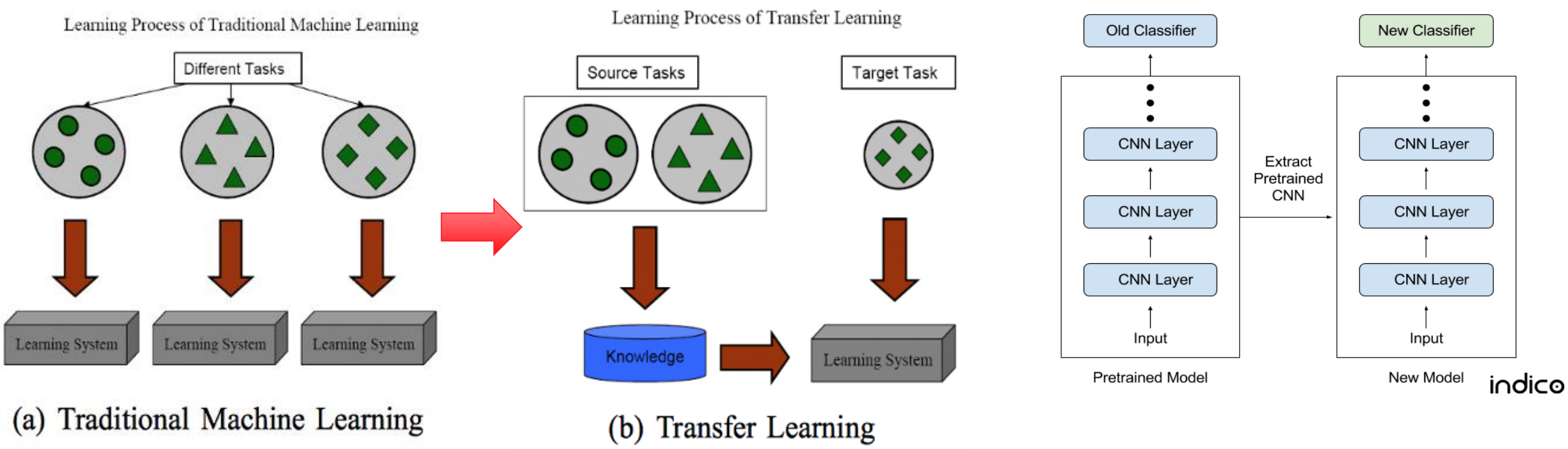
Много параметров:

- Сложно обучать, нужны большие наборы данных
- Большой размер и долгое время предсказания (inference)

https://keras.io/applications/

Model	Size	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth
Xception	88 MB	0.790	0.945	22,910,480	126
VGG16	528 MB	0.713	0.901	138,357,544	23
VGG19	549 MB	0.713	0.900	143,667,240	26
ResNet50	98 MB	0.749	0.921	25,636,712	-
ResNet101	171 MB	0.764	0.928	44,707,176	-
ResNet152	232 MB	0.766	0.931	60,419,944	-
ResNet50V2	98 MB	0.760	0.930	25,613,800	-
ResNet101V2	171 MB	0.772	0.938	44,675,560	-
ResNet152V2	232 MB	0.780	0.942	60,380,648	-
InceptionV3	92 MB	0.779	0.937	23,851,784	159
InceptionResNetV2	215 MB	0.803	0.953	55,873,736	572
MobileNet	16 MB	0.704	0.895	4,253,864	88
MobileNetV2	14 MB	0.713	0.901	3,538,984	88
DenseNet121	33 MB	0.750	0.923	8,062,504	121
DenseNet169	57 MB	0.762	0.932	14,307,880	169
DenseNet201	80 MB	0.773	0.936	20,242,984	201
NASNetMobile	23 MB	0.744	0.919	5,326,716	-
NASNetLarge	343 MB	0.825	0.960	88,949,818	-
EfficientNetB0	29 MB	-	-	5,330,571	-
EfficientNetB1	31 MB	-	-	7,856,239	-
EfficientNetB2	36 MB	-	-	9,177,569	-
EfficientNetB3	48 MB	-	-	12,320,535	-
EfficientNetB4	75 MB	-	-	19,466,823	-
EfficientNetB5	118 MB	-	-	30,562,527	-
EfficientNetB6	166 MB	-	-	43,265,143	-
EfficientNetB7	256 MB	-	-	66,658,687	-

Transfer learning. Дообучение сети (fine-tuning)



BigTransfer (BiT)

Pre-training:

- BiT-S: ILSVRC-2012 (1.3M изображений)
- BiT-M: ImageNet-21k (14M изображений)
- BiT-S: JFT (300M изображений)

```
def get_mixup(dataset_size):
  return 0.0 if dataset_size < 20_000 else 0.1

def get_schedule(dataset_size):
  if dataset_size < 20_000:
    return [100, 200, 300, 400, 500]
  elif dataset_size < 500_000:
    return [500, 3000, 6000, 9000, 10_000]
  else:
    return [500, 6000, 12_000, 18_000, 20_000]
```

https://arxiv.org/pdf/1912.11370.pdf

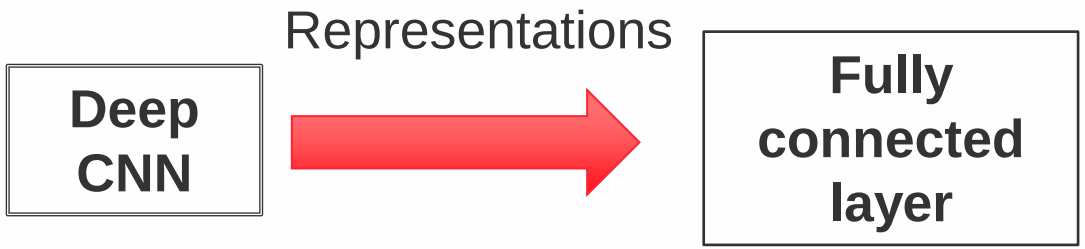
https://github.com/google-research/big_transfer/blob/master/bit_hyperrule.py

Эвристика для подбора гиперпараметров HyperRule:

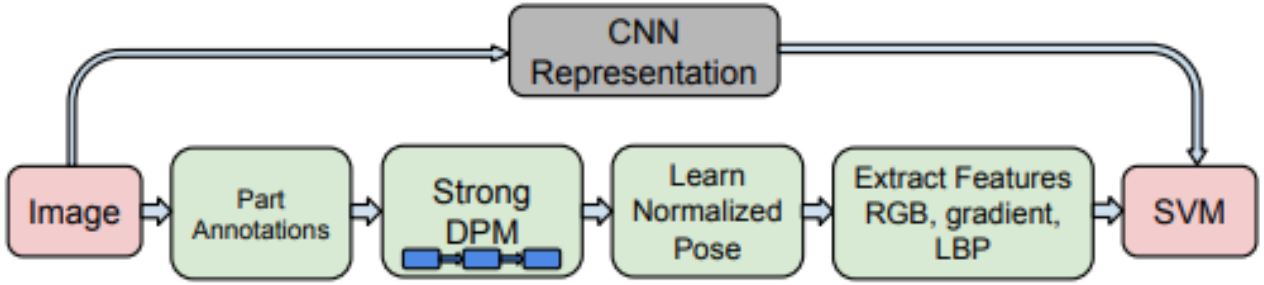
- Resolution
- training schedule length
- whether to use MixUp regularization

```
def get_lr(step, dataset_size, base_lr=0.003):
supports = get_schedule(dataset_size)
  # Linear warmup
  if step < supports[0]:
    return base_lr * step / supports[0]
  # End of training
  elif step >= supports[-1]:
    return None
  # Staircase decays by factor of 10
else:
    for s in supports[1:]:
      if s < step:
        base_lr /= 10
    return base_lr
```

CNN representations (embeddings)

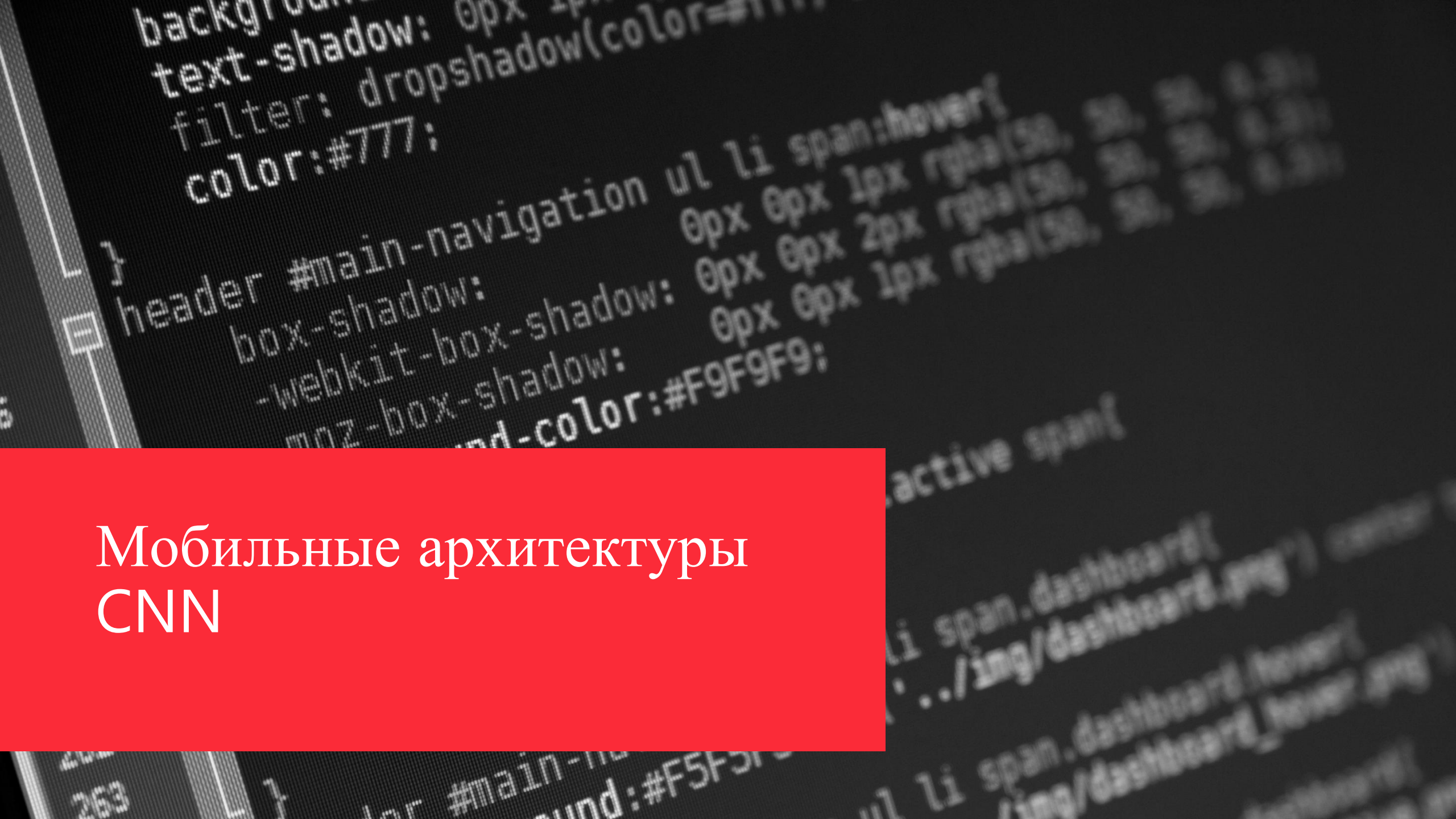


OverFeat features → Trained classifier on other data sets



	VOC07c	VOC12c	VOC12a	MIT67	SUN397	VOC07d	VOC10d	VOC11s	200Birds
best non-CNN results	70.5	82.2	69.6	64.0	47.2	34.3	40.4	47.6	56.8
off-the-shelf ImageNet Model	80.1[13] 80.1[10] 77.2[1]	82.7[10] 79.0[6]	-	69.0[1]	40.9[4]	46.2[2] 46.1[11] 44.9[13]	44.1[11]	-	61.8[1] 58.8[4]
off-the-shelf ImageNet Model + rep learning	-	-	-	68.9[3]	52.0[3]	-	-	-	65.0[4]
fine-tuned ImageNet Model	82.42[10] 77.7[5]	83.2[10] 82.8[5]	70.2[5]	-	-	60.9[13] 58.5[2]	53.7[2]	47.9[2]	75.7[12]

Razavian, Azizpour, Sullivan, Carlsson "CNN features off-the-shelf: An astounding baseline for recogniton", CVPR 2014



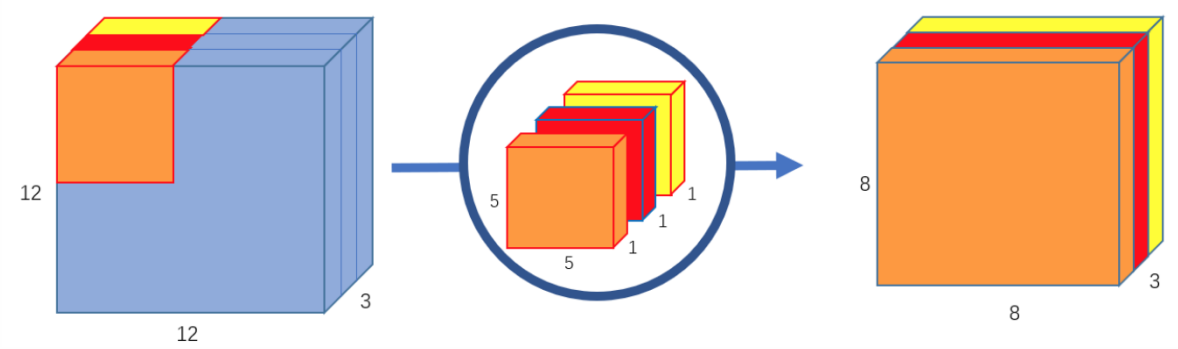
Мобильные архитектуры CNN

Depthwise separable convolution

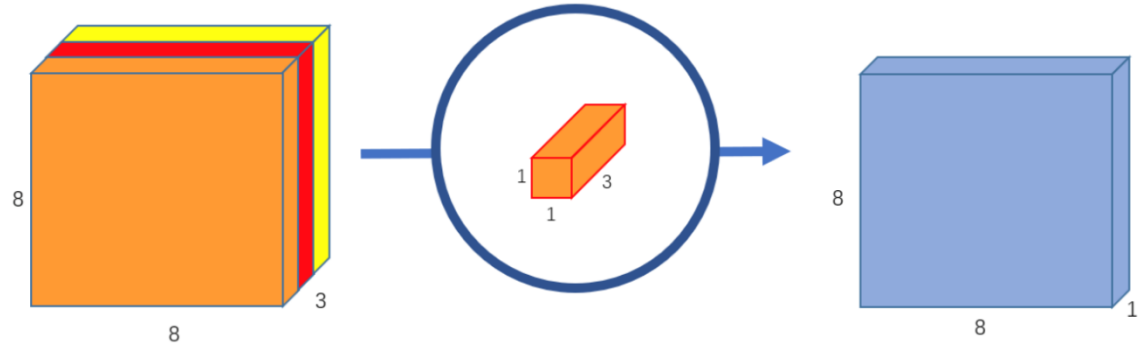
Пример: фильтр Собеля

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \times \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$$

Depthwise convolution



Pointwise convolution



Обычная свертка

256 ядер 5x5x3, проходящих 8x8 раз. Всего 256x3x5x5x8x8 = 1,228,800 операций умножения

«Разделяемая» свертка

1) **depthwise**: 3 ядра 5x5x1 проходят 8x8 раз, 3x5x5x8x8 = 4,800 операций

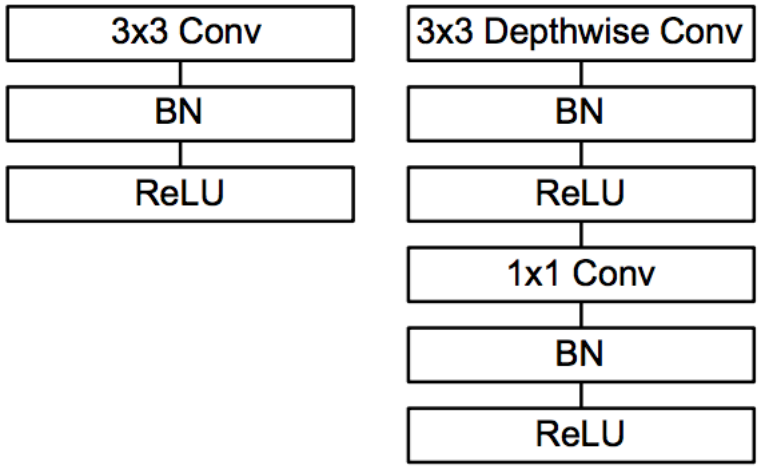
2) **pointwise**: 256 ядер 1x1x3 проходят 8x8 раз. 256x1x1x3x8x8=49,152 операций

Итого: 53,952 умножений

https://towardsdatascience.com/a-basic-introduction-to-separable-convolutions-b99ec3102728

MobileNet v1

Depthwise convolution



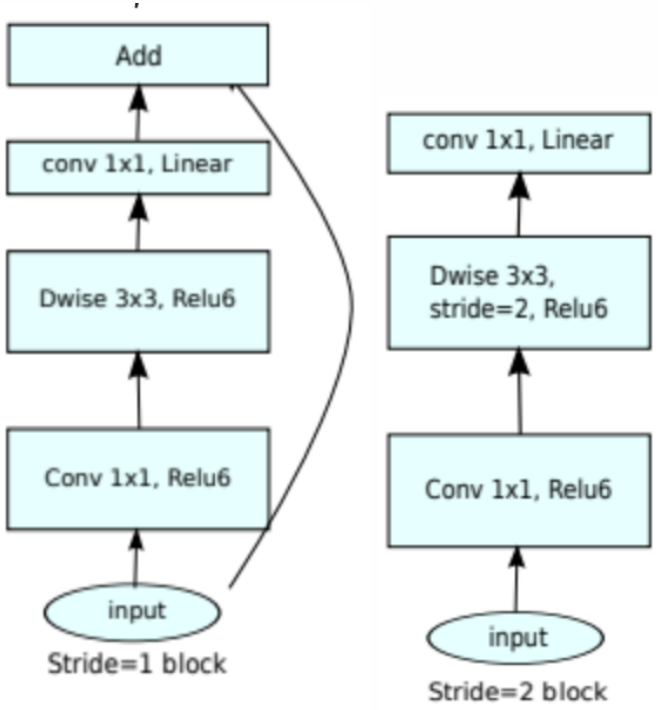
- 28 слоев (вместо max-pooling – сверточный слой с параметром stride=2)
- ImageNet Top-5 Error rate: 12.81%
- 4.2M весов

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
$5\times$ Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
$5\times$ Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

https://arxiv.org/abs/1704.04861

MobileNet v2

Bottleneck residual block



Bottleneck layer (снижение числа каналов)

Depthwise convolution

Expansion layer (повышение размерности в t=6 раз)

Input	Operator	Output
$h \times w \times k$	1x1 conv2d , ReLU6	$h \times w \times (tk)$
$h \times w \times tk$	3x3 dwise s=s, ReLU6	$\frac{h}{s} \times \frac{w}{s} \times (tk)$
$\frac{h}{s} \times \frac{w}{s} \times tk$	linear 1x1 conv2d	$\frac{h}{s} \times \frac{w}{s} \times k'$

Residual block

```
def residual_block(x, squeeze=16, expand=64):
  m = Conv2D(squeeze, (1,1), activation='relu')(x)
  m = Conv2D(squeeze, (3,3), activation='relu')(m)
  m = Conv2D(expand, (1,1), activation='relu')(m)
  return Add()([m, x])
```

Inverted residual block

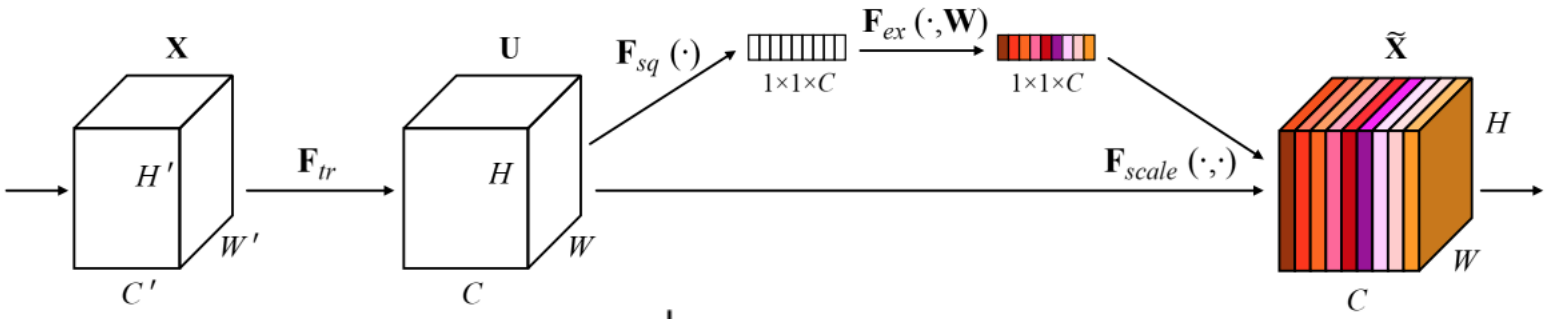
```
def inverted_residual_block(x, expand=64,
squeeze=16):
  m = Conv2D(expand, (1,1), activation='relu')(x)
  m = DepthwiseConv2D((3,3),
activation='relu')(m)
  m = Conv2D(squeeze, (1,1), activation='relu')(m)
  return Add()([m, x])
```

- ImageNet Top-1 Accuracy: 74.7% vs 70.6% for MobileNet v1
- 6…9M весов

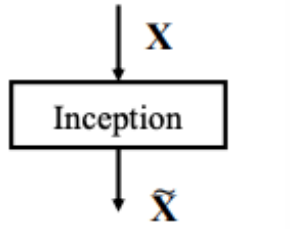
http://arxiv.org/abs/1801.04381

Squeeze-and-Excitation Networks (SENet)

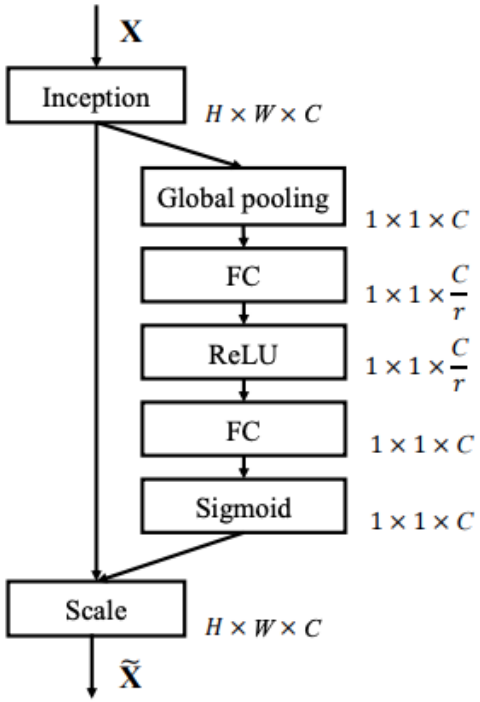
Squeeze-and-Excitation block

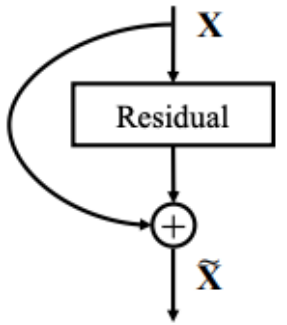



```
def se_block(in_block, ch, ratio=16):
 x = GlobalAveragePooling2D()(in_block)
 x = Dense(ch//ratio, activation='relu')(x)
 x = Dense(ch, activation='sigmoid')(x)
 return multiply()([in_block, x])
```

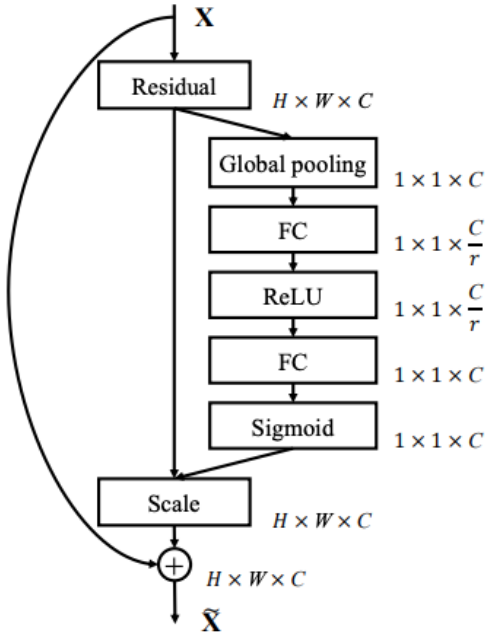


Inception Module





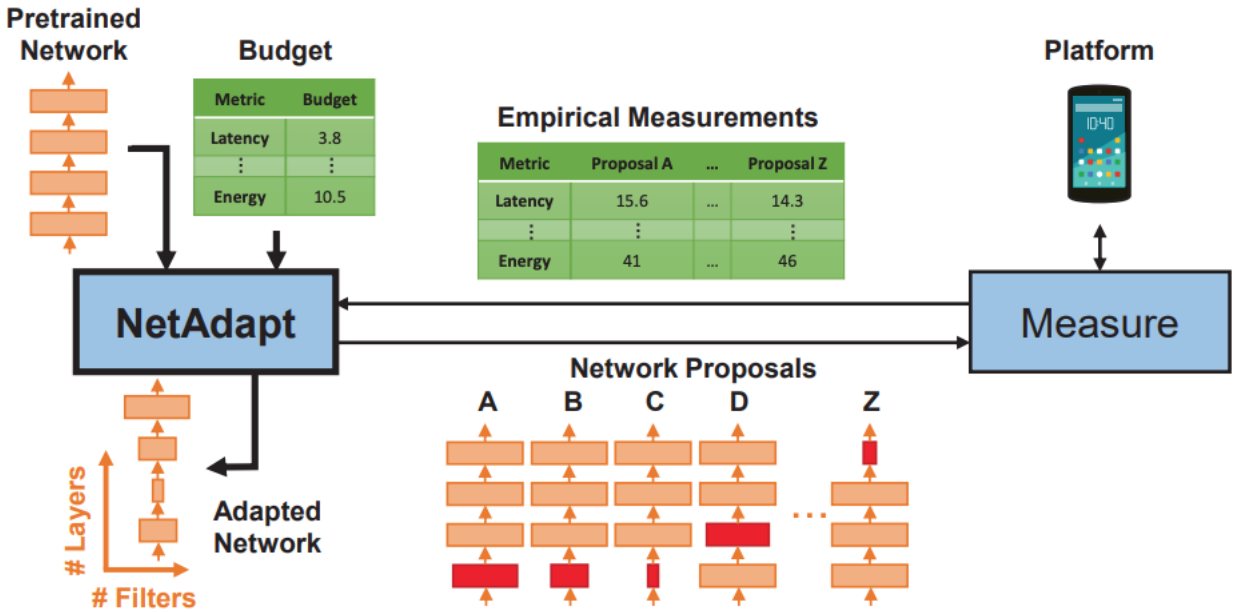
ResNet Module



SE-ResNet Module

Hu J. et al, CVPR 2018

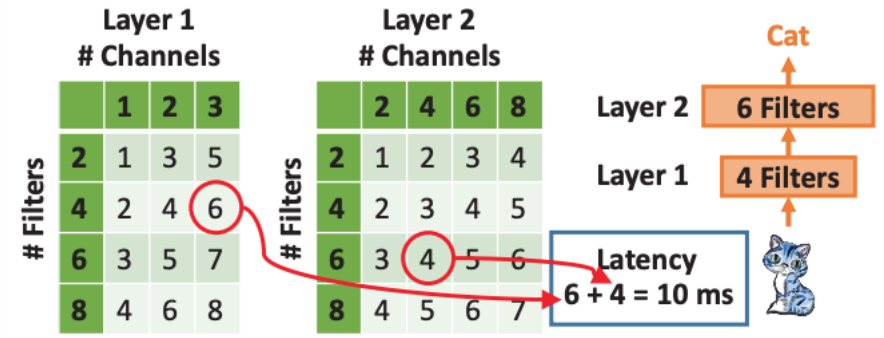
Neural Architecture Search (NAS). NetAdapt

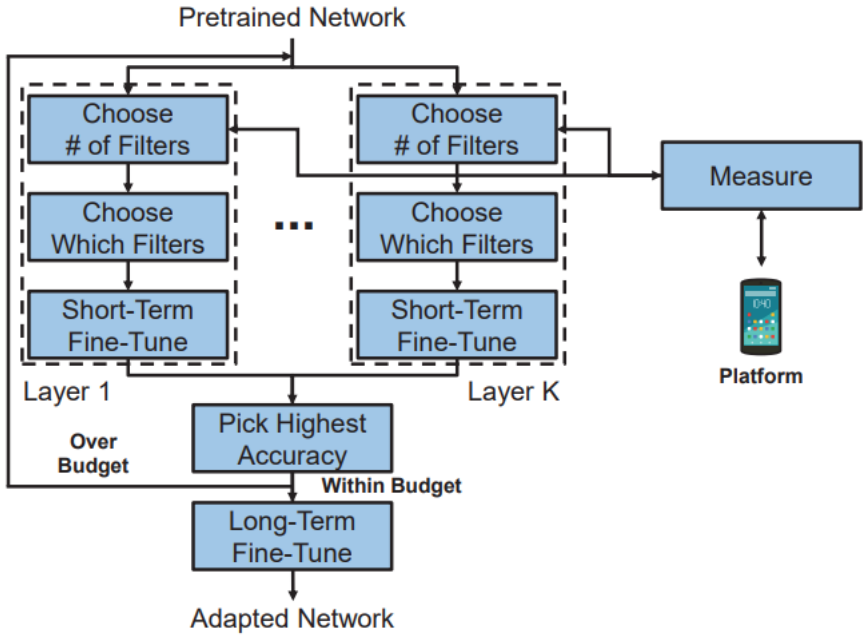


$$\underset{Net}{\text{maximize}} \quad Acc(Net)$$
$$\text{subject to} \quad Res_j(Net) \leq Bud_j,\ j = 1, \ldots, m,$$

$$\underset{Net_i}{\text{maximize}} \quad Acc(Net_i)$$
$$\text{subject to} \quad Res_j(Net_i) \leq Res_j(Net_{i-1}) - \Delta R_{i,j},\ j = 1, \ldots, m,$$

fast resource consumption estimation

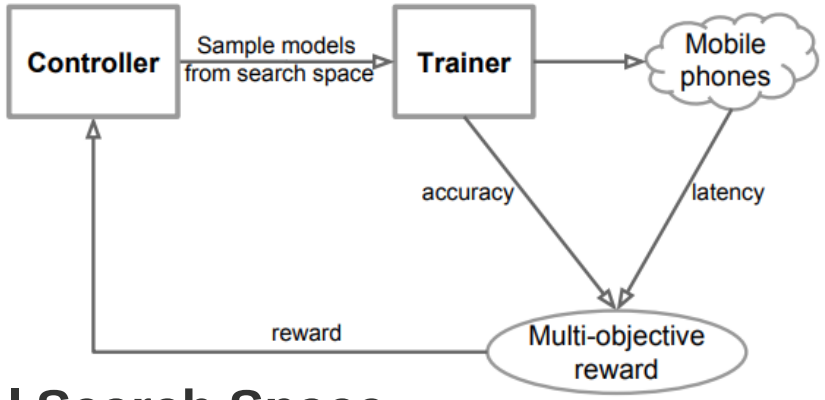




Tien-Ju Yang et al, ECCV 2018

MnasNet

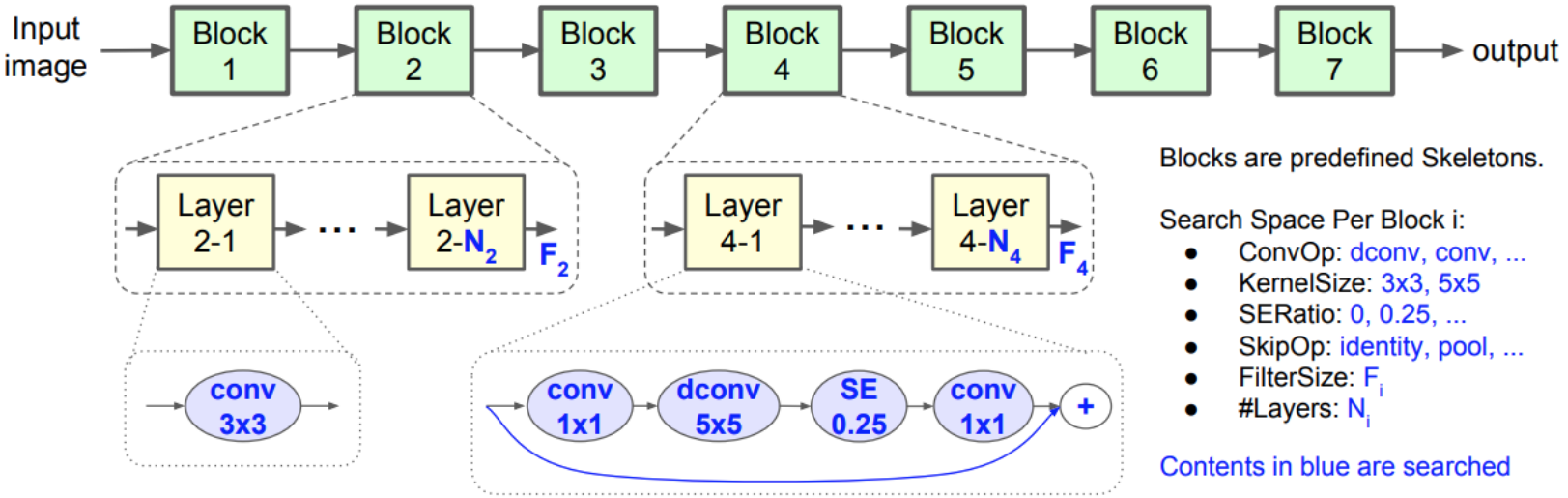
Platform-Aware NAS



$$\underset{m}{\text{maximize}} \quad ACC(m)$$
$$\text{subject to} \quad LAT(m) \leq T$$

$$\underset{m}{\text{maximize}} \quad ACC(m) \times \left[\frac{LAT(m)}{T}\right]^{w}$$

Factorized Hierarchical Search Space

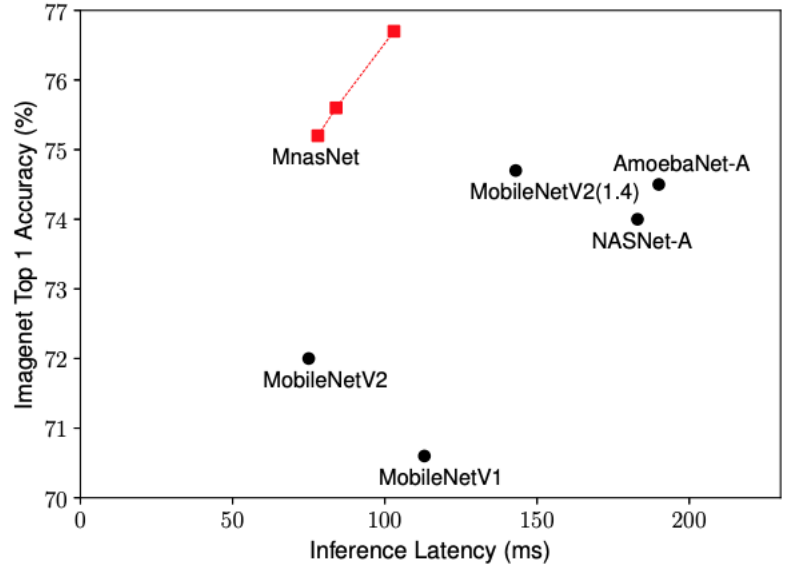


Blocks are predefined Skeletons.

Search Space Per Block i:

- ConvOp: dconv, conv, ...
- KernelSize: 3x3, 5x5
- SERatio: 0, 0.25, ...
- SkipOp: identity, pool, ...
- FilterSize: F_i
- #Layers: N_i

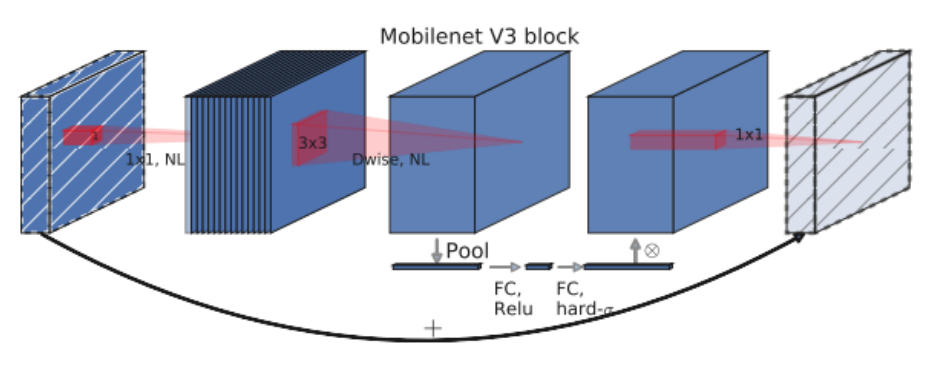
Contents in blue are searched



Mingxing Tan et al, CVPR 2019

MobileNet v3

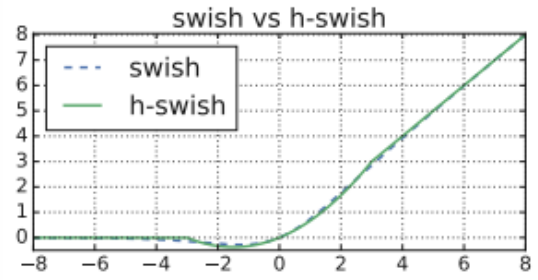
MobileNetV2 + Squeeze-and-Excite



Функции активации

$$\text{swish}\, x = x \cdot \sigma(x)$$

$$\text{h-swish}[x] = x\frac{\text{ReLU6}(x+3)}{6}$$



Platform-aware NAS+NetAdapt

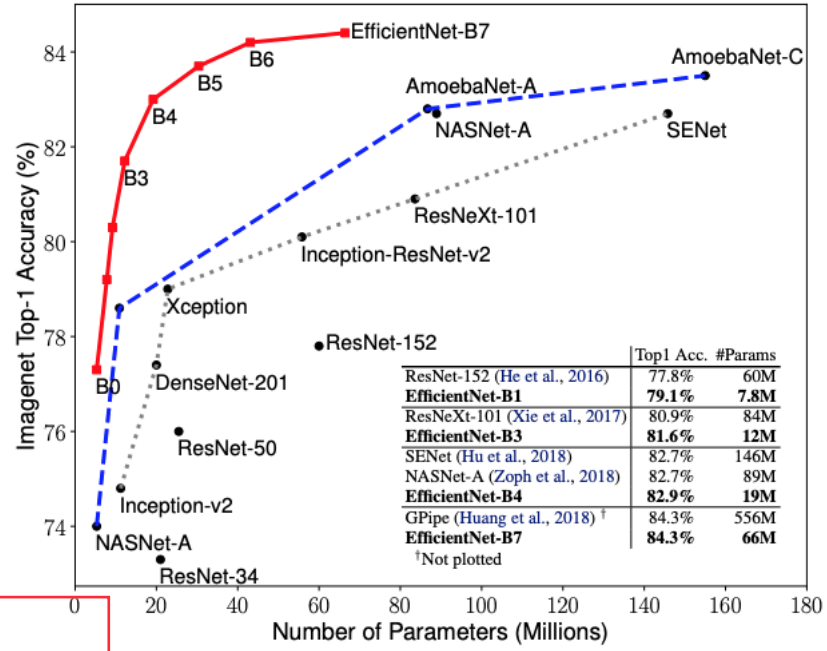
MobileNetV3-Large

Input	Operator	exp size	#out	SE	NL	s
$224^2 \times 3$	conv2d	-	16	-	HS	2
$112^2 \times 16$	bneck, 3x3	16	16	-	RE	1
$112^2 \times 16$	bneck, 3x3	64	24	-	RE	2
$56^2 \times 24$	bneck, 3x3	72	24	-	RE	1
$56^2 \times 24$	bneck, 5x5	72	40	✓	RE	2
$28^2 \times 40$	bneck, 5x5	120	40	✓	RE	1
$28^2 \times 40$	bneck, 5x5	120	40	✓	RE	1
$28^2 \times 40$	bneck, 3x3	240	80	-	HS	2
$14^2 \times 80$	bneck, 3x3	200	80	-	HS	1
$14^2 \times 80$	bneck, 3x3	184	80	-	HS	1
$14^2 \times 80$	bneck, 3x3	184	80	-	HS	1
$14^2 \times 80$	bneck, 3x3	480	112	✓	HS	1
$14^2 \times 112$	bneck, 3x3	672	112	✓	HS	1
$14^2 \times 112$	bneck, 5x5	672	160	✓	HS	2
$7^2 \times 160$	bneck, 5x5	960	160	✓	HS	1
$7^2 \times 160$	bneck, 5x5	960	160	✓	HS	1
$7^2 \times 160$	conv2d, 1x1	-	960	-	HS	1
$7^2 \times 960$	pool, 7x7	-	-	-	-	1
$1^2 \times 960$	conv2d 1x1, NBN	-	1280	-	HS	1
$1^2 \times 1280$	conv2d 1x1, NBN	-	k	-	-	1

http://arxiv.org/abs/1801.04381

EfficientNet

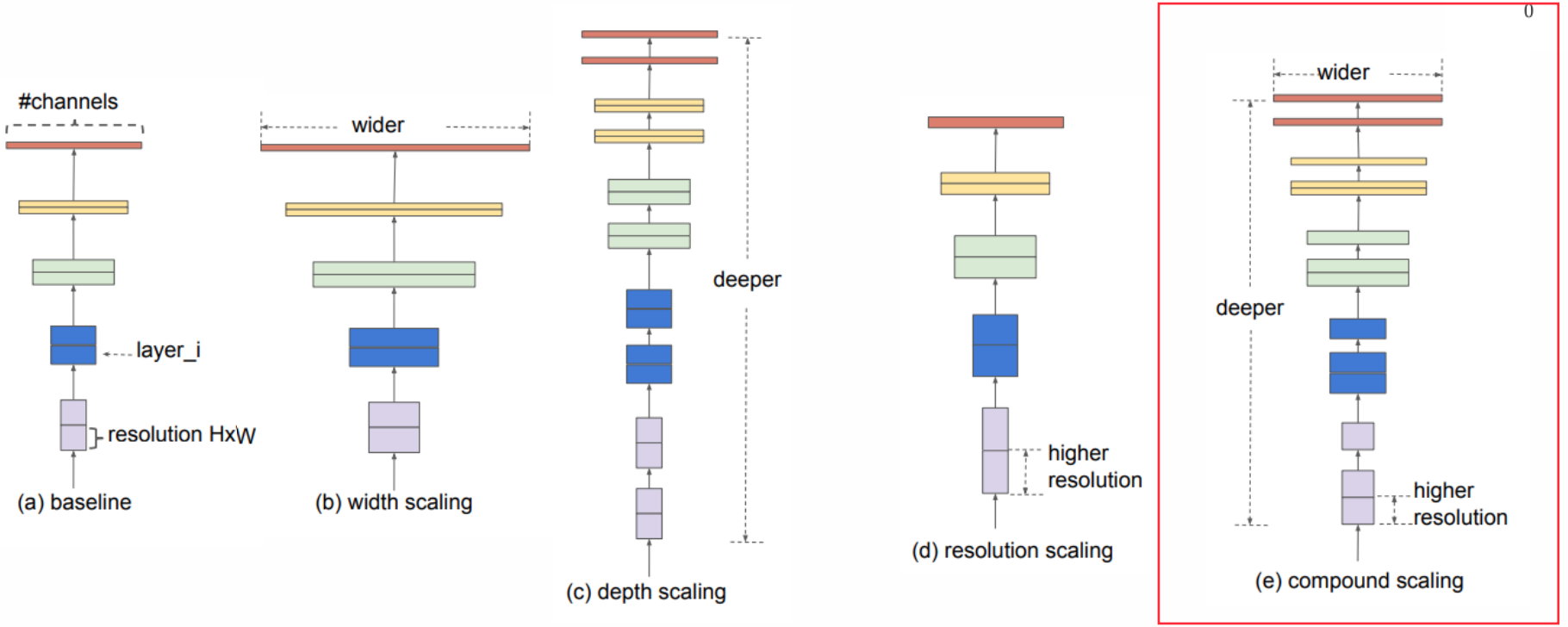
Автоматический поиск архитектуры (MnasNet)



	Top1 Acc.	#Params
ResNet-152 (He et al., 2016)	77.8%	60M
EfficientNet-B1	**79.1%**	**7.8M**
ResNeXt-101 (Xie et al., 2017)	80.9%	84M
EfficientNet-B3	**81.6%**	**12M**
SENet (Hu et al., 2018)	82.7%	146M
NASNet-A (Zoph et al., 2018)	82.7%	89M
EfficientNet-B4	**82.9%**	**19M**
GPipe (Huang et al., 2018) †	84.3%	556M
EfficientNet-B7	**84.3%**	**66M**

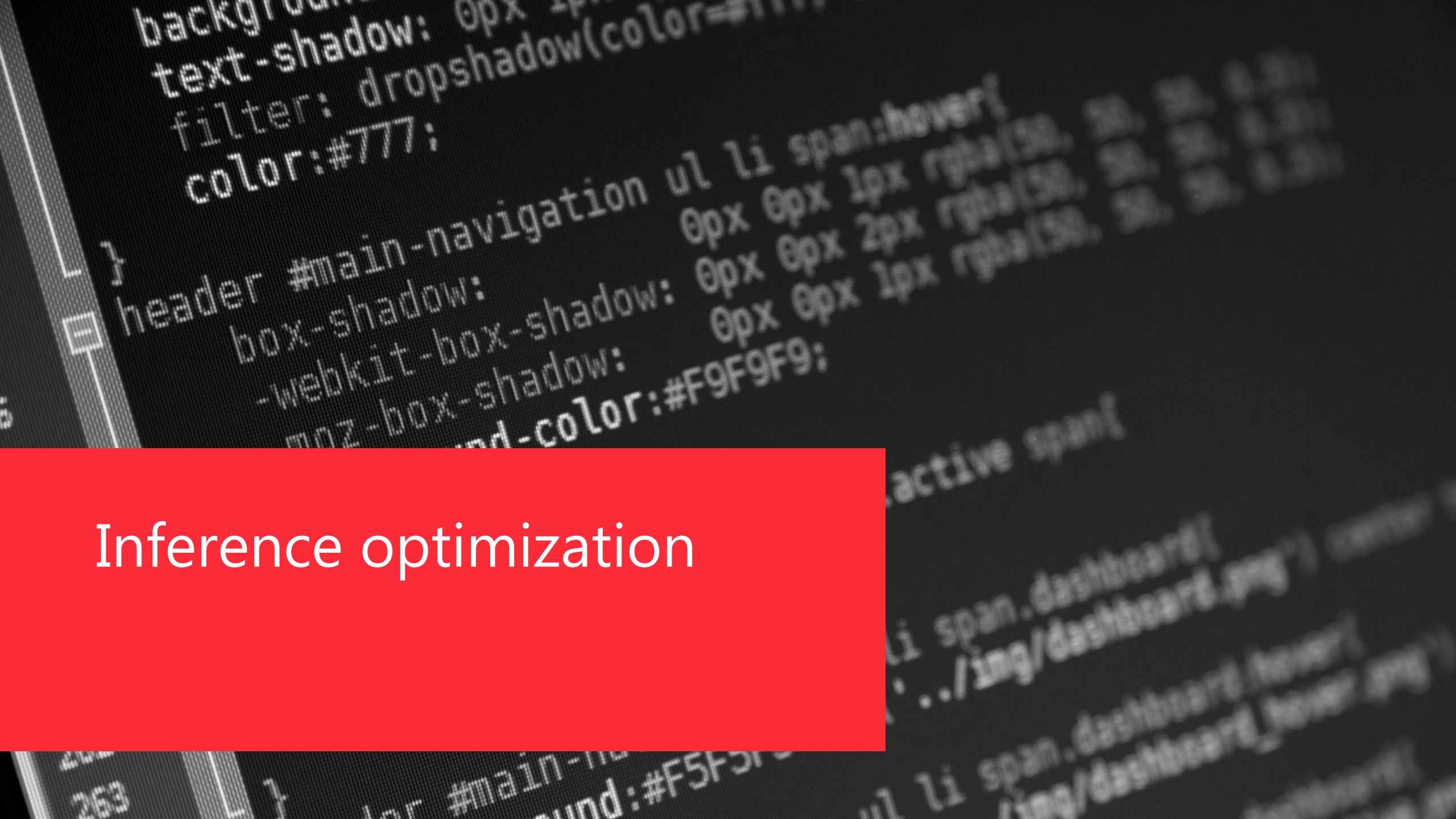
†Not plotted

Масштабирование (scaling) моделей



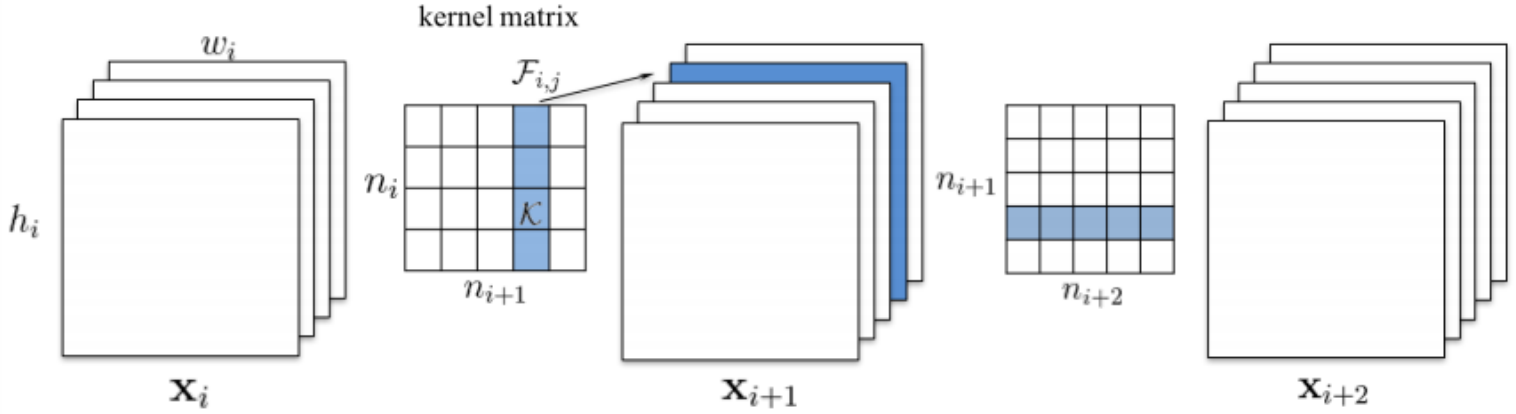
$$\text{depth: } d = \alpha^\phi$$
$$\text{width: } w = \beta^\phi$$
$$\text{resolution: } r = \gamma^\phi$$
$$\text{s.t. } \alpha \cdot \beta^2 \cdot \gamma^2 \approx 2$$
$$\alpha \geq 1, \beta \geq 1, \gamma \geq 1$$

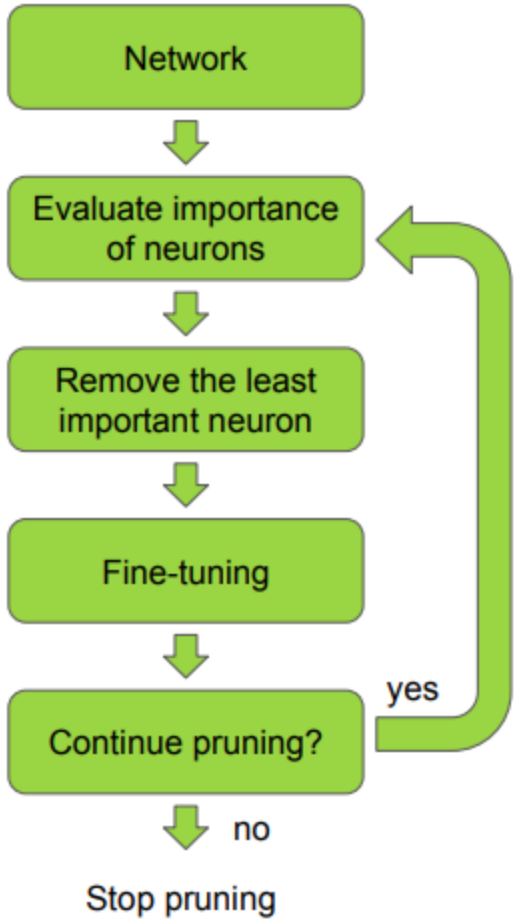
https://arxiv.org/pdf/1905.11946.pdf


Inference optimization

Structural pruning

Удаление фильтров сверточного слоя





Как выбрать каналы для удаления?

$$\min_{\mathcal{W}'} \left| \mathcal{C}(\mathcal{D}|\mathcal{W}') - \mathcal{C}(\mathcal{D}|\mathcal{W}) \right| \quad \text{s.t.} \quad ||\mathcal{W}'||_0 \leq B,$$

Average Percentage of Zeros (APoZ)
(https://arxiv.org/pdf/1607.03250.pdf)

$$APoZ_c^{(i)} = APoZ(O_c^{(i)}) = \frac{\sum_k^N \sum_j^M f(O_{c,j}^{(i)}(k) = 0)}{N \times M}$$

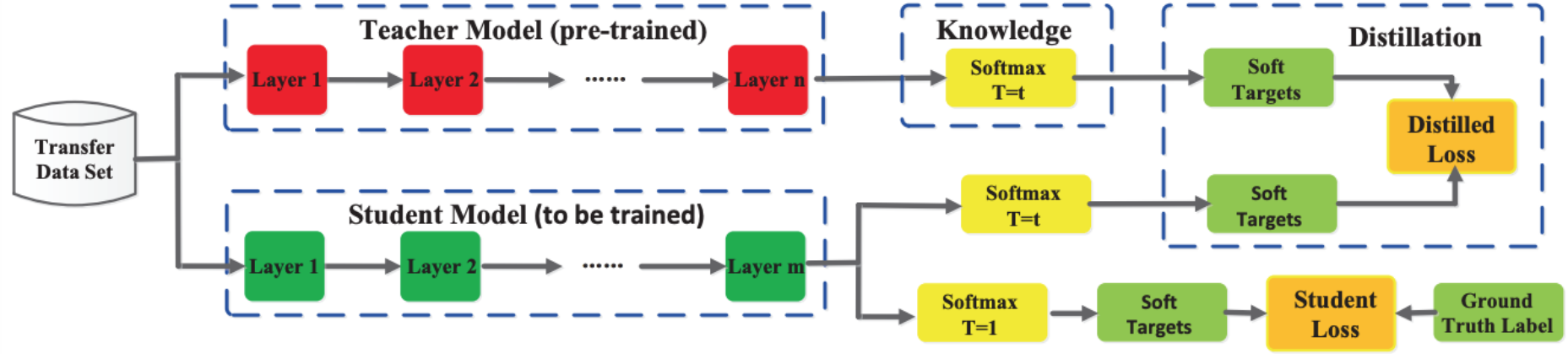
Taylor expansion

$$\Theta_{TE}(z_l^{(k)}) = \left| \frac{1}{M} \sum_m \frac{\delta C}{\delta z_{l,m}^{(k)}} z_{l,m}^{(k)} \right|,$$

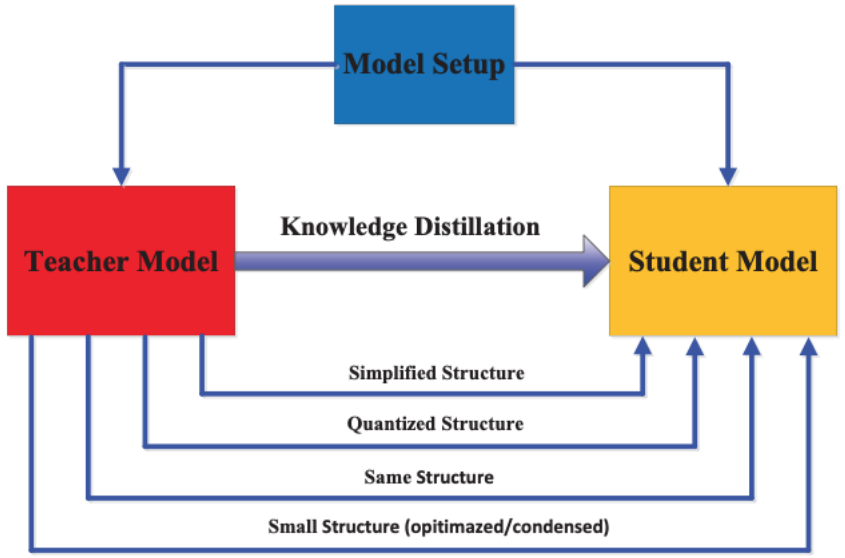
https://arxiv.org/pdf/1608.08710.pdf

https://arxiv.org/pdf/1611.06440.pdf

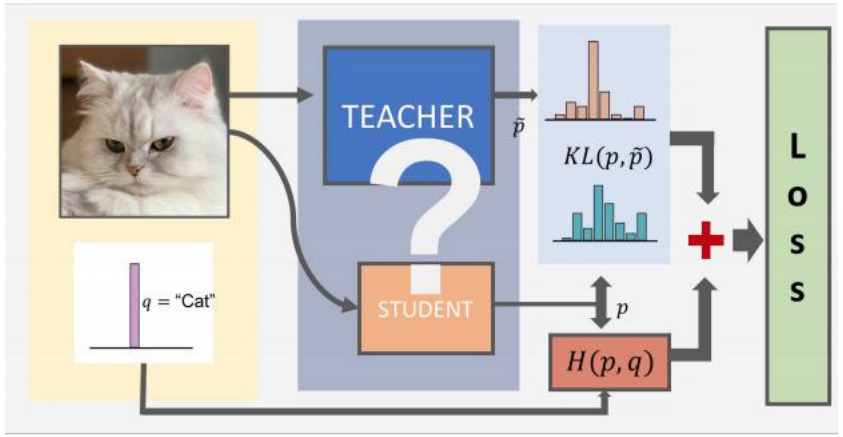
Knowledge distillation



Варианты дистилляции



Функция потерь



$$\mathcal{L} = \alpha \mathcal{L}_{cls} + (1-\alpha)\mathcal{L}_{KD}$$

$$\mathcal{L}_{KD} = -\tau^2 \sum_k \tilde{p}_k^t(x) \log \tilde{p}_k^s(x)$$

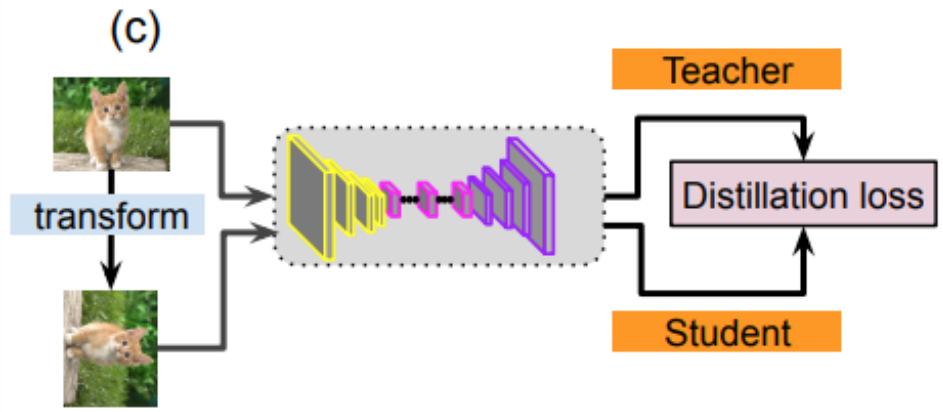
$$\tilde{p}_k^t(x) = \frac{e^{s_k^t(x)/\tau}}{\sum_j e^{s_j^t(x)/\tau}}$$

https://arxiv.org/pdf/2006.05525.pdf

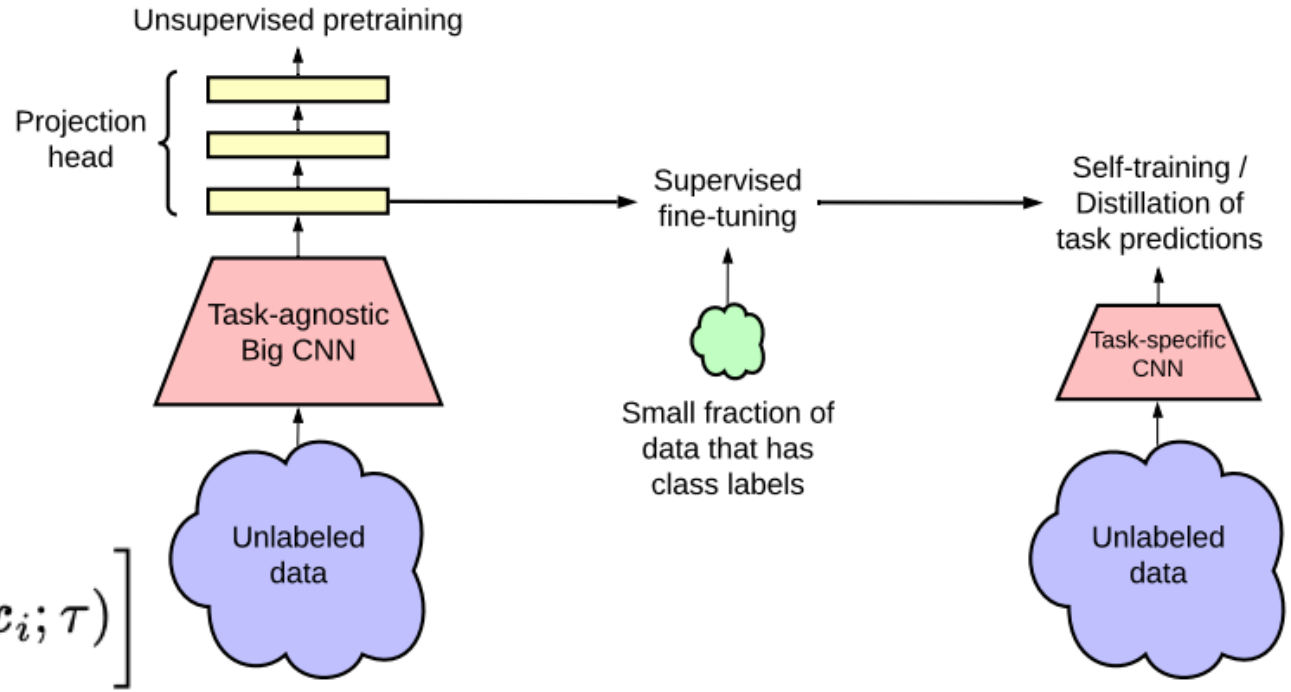
Jang Hyun Cho & B. Hariharan, ICCV 2019

Пример

Обучение без учителя



SimCLR v2 (Simple Framework for Contrastive Learning)



$$\mathcal{L}^{\text{distill}} = -\sum_{\boldsymbol{x}_i \in \mathcal{D}} \left[\sum_{y} P^T(y|\boldsymbol{x}_i;\tau) \log P^S(y|\boldsymbol{x}_i;\tau) \right]$$

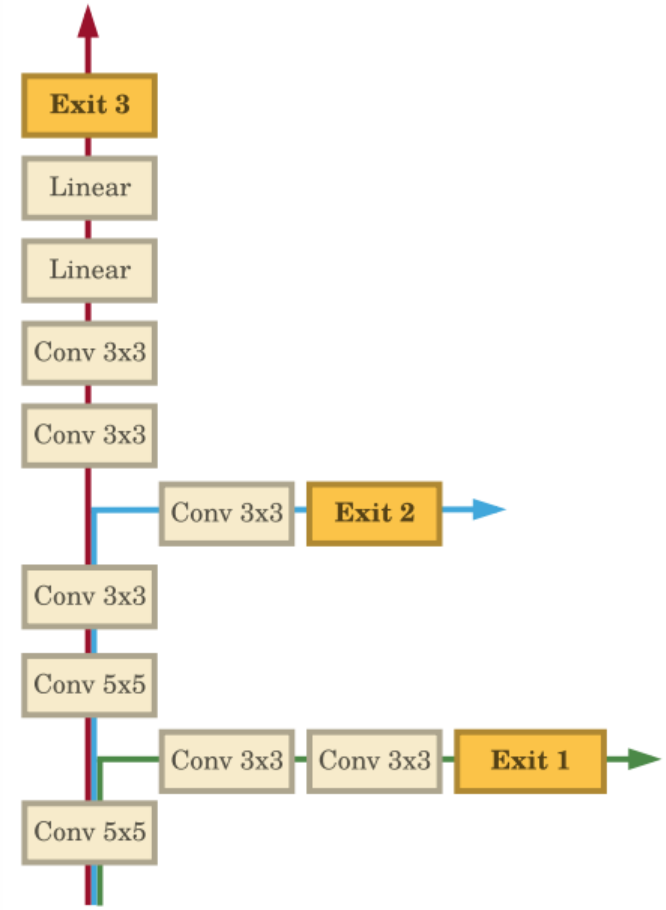
$$P(y|\boldsymbol{x}_i) = \exp(f^{\text{task}}(\boldsymbol{x}_i)[y]/\tau) / \sum_{y'} \exp(f^{\text{task}}(\boldsymbol{x}_i)[y']/\tau)$$

https://arxiv.org/pdf/2006.10029.pdf

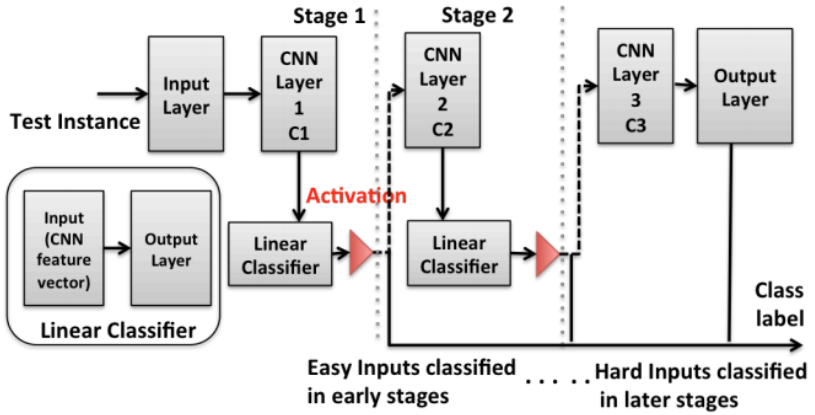
https://arxiv.org/pdf/2004.05937.pdf

Адаптивные нейронные сети

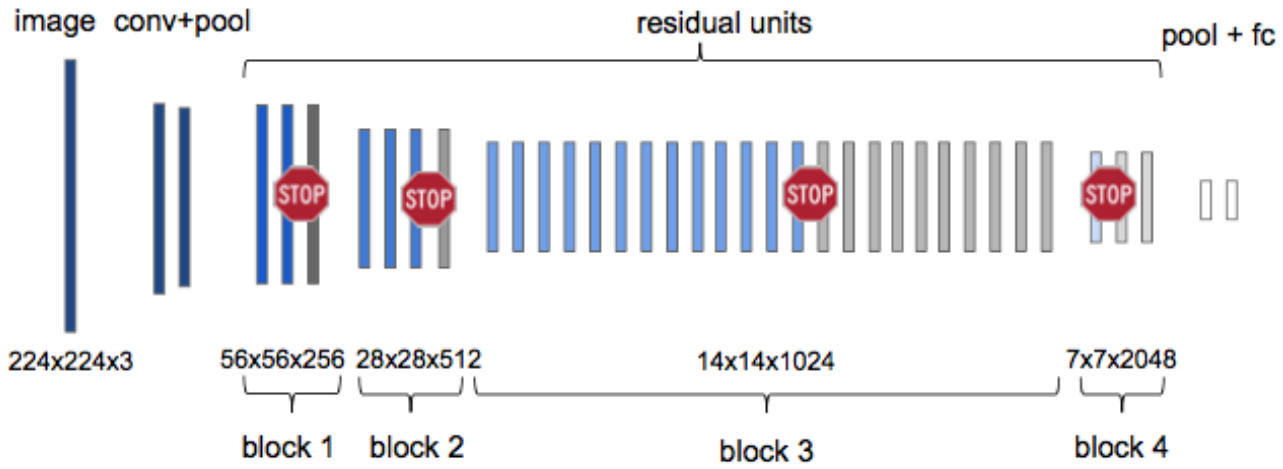
BranchyNet
(https://arxiv.org/pdf/1709.01686.pdf)

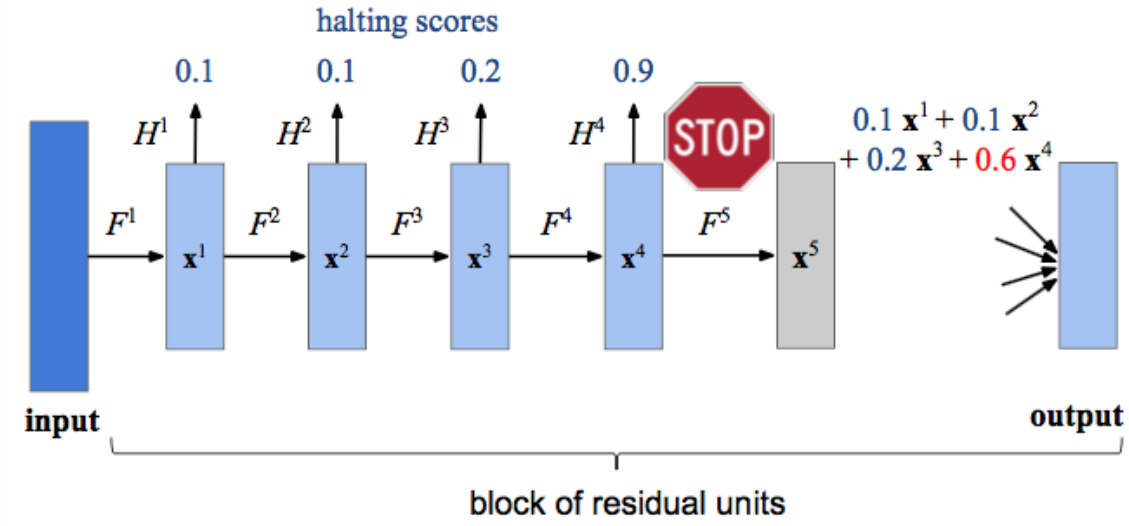


Conditional deep learning (CDL) network
(https://arxiv.org/pdf/1509.08971.pdf)



Adaptive computation time (ACT)





$$\mathbf{x}^0 = \mathbf{input},$$
$$\mathbf{x}^l = F^l(\mathbf{x}^{l-1}) = \mathbf{x}^{l-1} + f^l(\mathbf{x}^{l-1})$$
$$\mathbf{output} = \mathbf{x}^L.$$

$$h^l = H^l(\mathbf{x}^l) = \sigma(W^l\,\mathrm{pool}(\mathbf{x}^l) + b^l).$$

No. of residual units to evaluate:

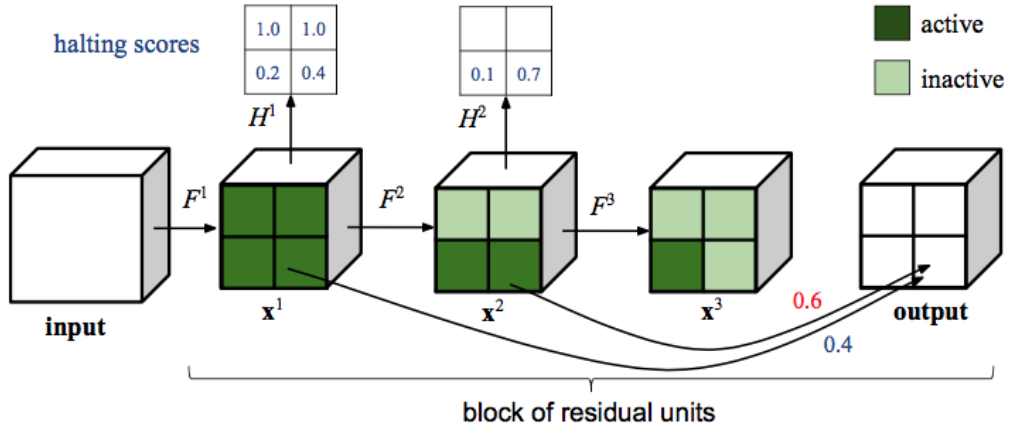
$$N = \min\left\{n \in \{1 \dots L\} : \sum_{l=1}^{n} h^l \geq 1 - \varepsilon\right\}$$

$$\text{output} = \sum_{l=1}^{L} p^l \mathbf{x}^l = \sum_{l=1}^{N} p^l \mathbf{x}^l.$$

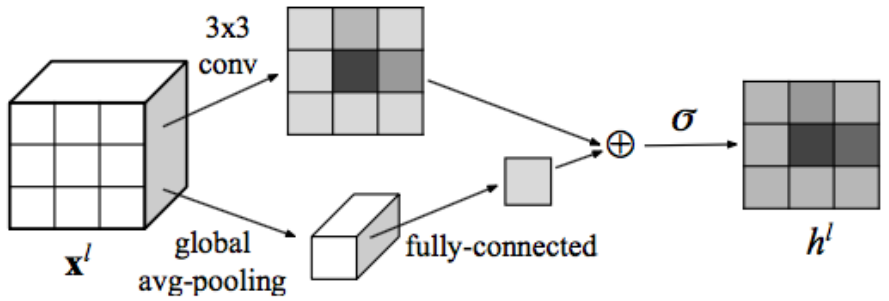
https://arxiv.org/abs/1612.02297

Spatially adaptive computation time

Apply ACT to each spatial position of the block



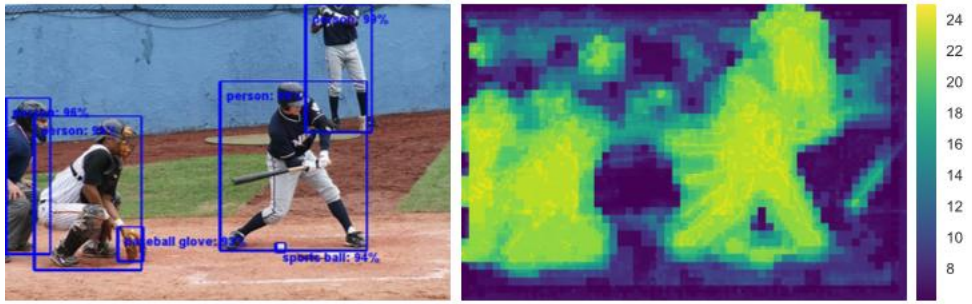
Halting score



COCO val set. Faster R-CNN with SACT

Feature extractor	FLOPs (%)	mAP @ [.5, .95] (%)
ResNet-101 [15]	100	27.2
ResNet-50 (our impl.)	46.6	25.56
SACT $\tau = 0.005$	**56.0** $\pm$ 8.5	**27.61**
SACT $\tau = 0.001$	72.4 $\pm$ 8.4	29.04
ResNet-101 (our impl.)	100	29.24

Heat map of computation time

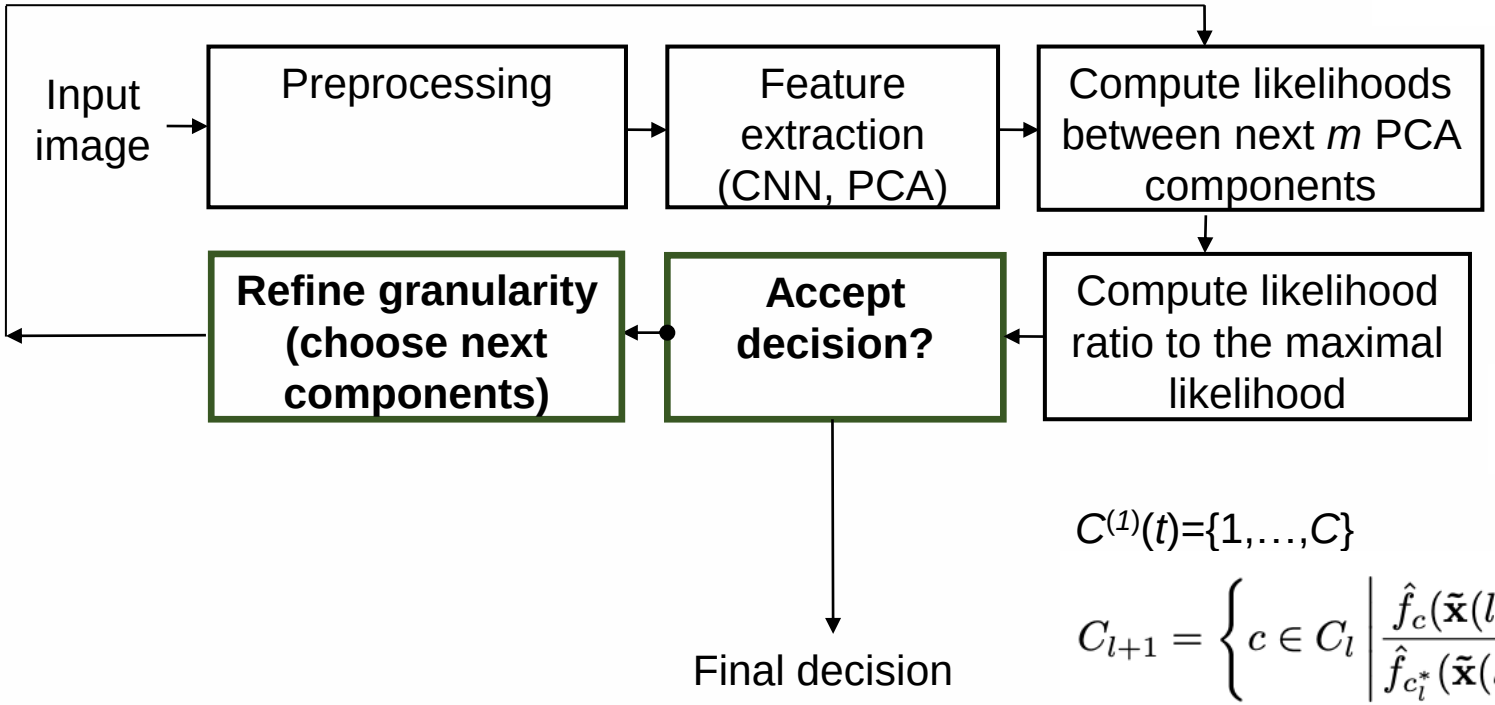


https://arxiv.org/abs/1612.02297

Последовательный анализ главных компонент (1)

Упорядоченные главные компоненты эмбеддингов анализируются последовательно

$$\rho\left(\tilde{\mathbf{x}}^{(l+1)}(t), \tilde{\mathbf{x}}_r^{(l+1)}\right) = \rho\left(\tilde{\mathbf{x}}^{(l)}(t), \tilde{\mathbf{x}}_r^{(l)}\right) + \sum_{d=d^{(l)}+1}^{d^{(l+1)}} \rho(\tilde{x}_d(t), \tilde{x}_{r;d})$$



$$c_l^* = \underset{c \in C_l}{\operatorname{argmax}} \hat{f}_c(\tilde{\mathbf{x}}(l))$$

$C^{(1)}(t)=\{1,\ldots,C\}$

$$C_{l+1} = \left\{ c \in C_l \,\middle|\, \frac{\hat{f}_c(\tilde{\mathbf{x}}(l))}{\hat{f}_{c_l^*}(\tilde{\mathbf{x}}(l))} \geq \delta \right\}$$

Надежность проверяется как в SIFT (2nd NN)

- Savchenko, Information Sciences, 2019
- https://ig.hse.ru/en/news/289417737.html

Последовательный анализ главных компонент (2)

Fig. 2. Images from face recognition example: (a) Probe photo, (b)-(e) Closest gallery photos.

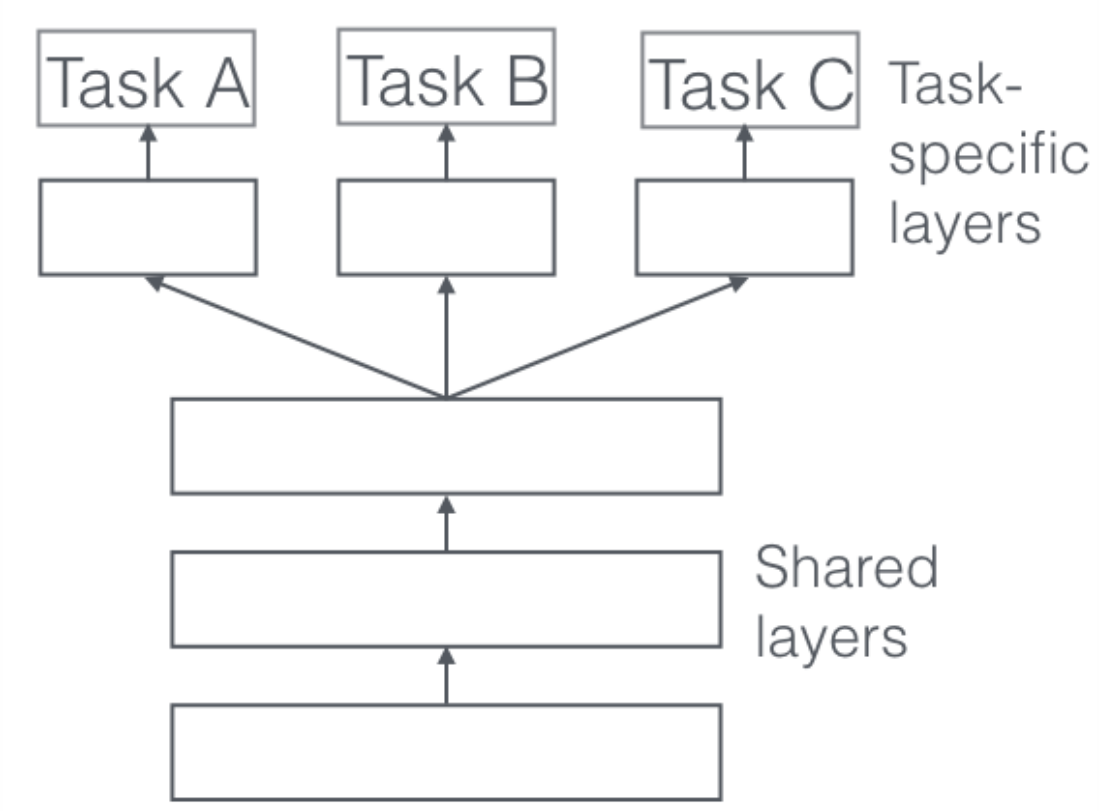
		Armstrong (Fig. 2b)	Auriemma (Fig. 2c)	Williams (Fig. 2d)	Wirayuda (Fig. 2e)	McEwen (Fig. 2f)	LeBron (Fig. 2g)
$l = 1$	$\rho[r]$	0.0086	0.0074	0.0100	0.0103	0.0104	0.0105
	DF: $\rho_{\min}/\rho[r]$	0.87	1.00	0.75	0.73	0.72	0.71
	$\hat{P}(c\|\mathbf{x})$	0.182	0.205	0.158	0.153	0.152	0.150
	BF: $(\rho[r] - \rho_{\min})/l$	0.0012	0	0.0026	0.0029	0.003	0.0031
$l = 2$	$\rho[r]$	0.0122	0.0170	0.0300	0.0217	0.0188	0.0200
	DF: $\rho_{\min}/\rho[r]$	1.00	0.71	0.41	0.56	0.65	0.61
	$\hat{P}(c\|\mathbf{x})$	0.237	0.187	0.097	0.148	0.171	0.161
	BF: $(\rho[r] - \rho_{\min})/l$	0	0.0024	0.0089	0.00475	0.0033	0.0039
$l = 3$	$\rho[r]$	0.0129	0.0195	-	-	-	-
	DF: $\rho_{\min}/\rho[r]$	1.00	0.66	-	-	-	-
	$\hat{P}(c\|\mathbf{x})$	0.555	0.445	-	-	-	-
	BF: $(\rho[r] - \rho_{\min})/l$	0	0.0022	-	-	-	-

Savchenko, Information Sciences, 2019

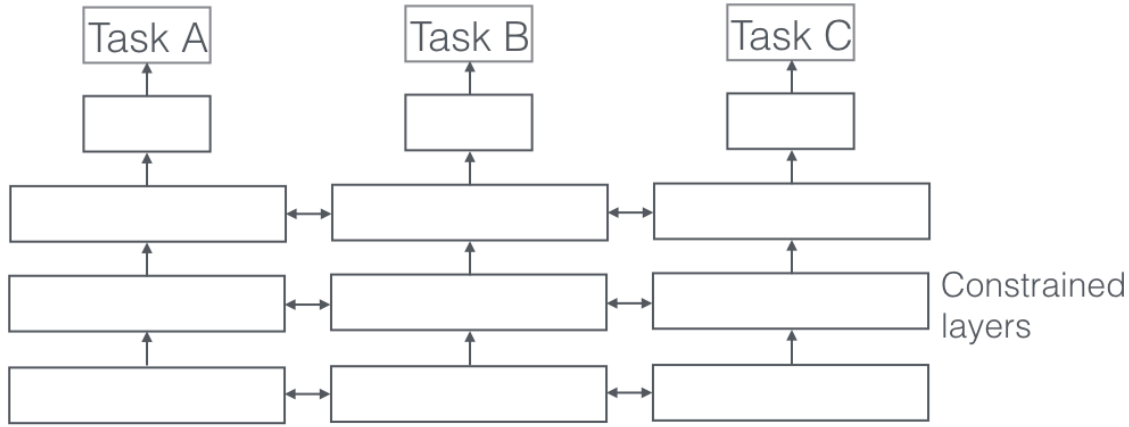

Multitask learning

Разделение весов

Hard parameter sharing

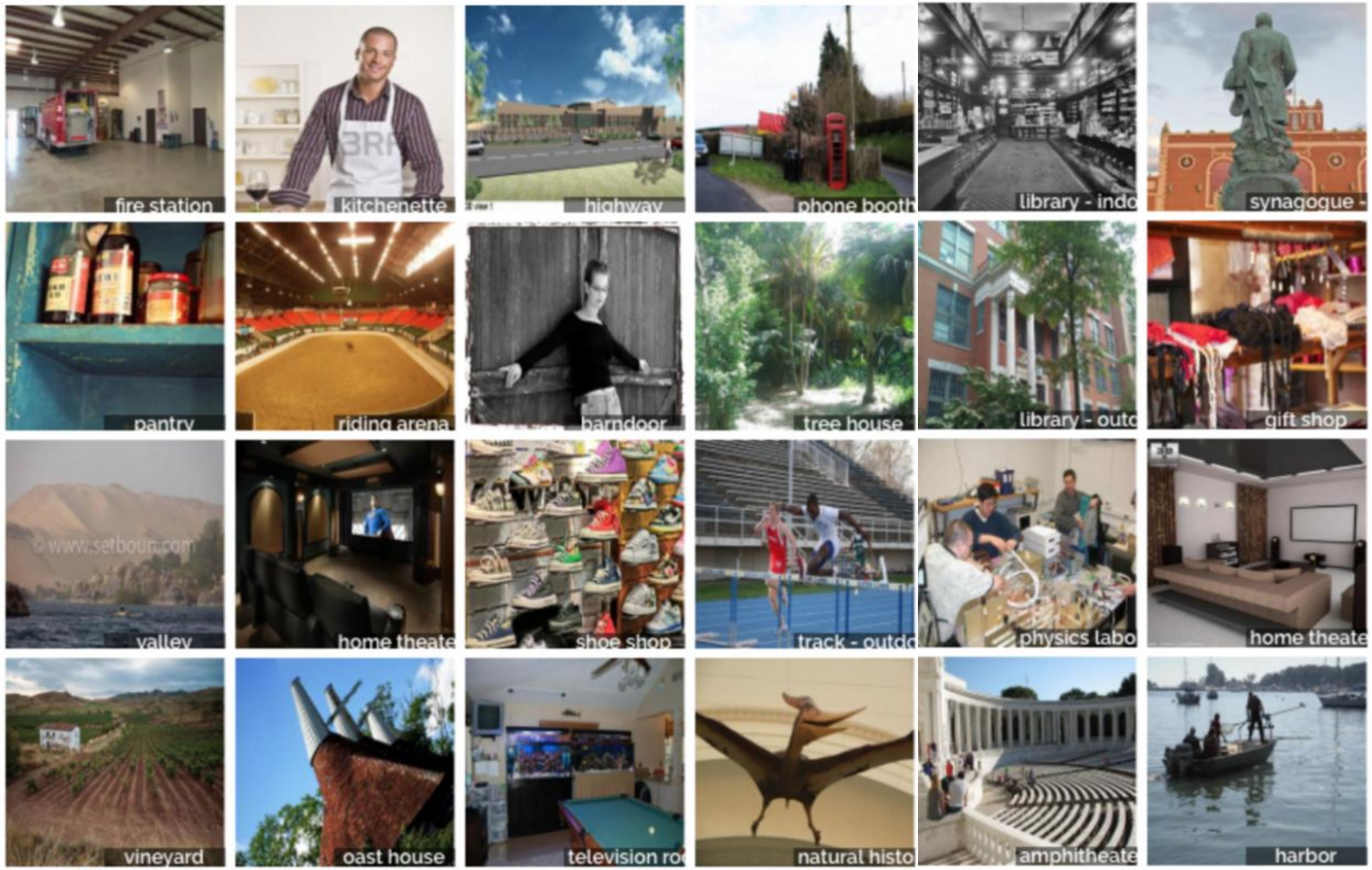


Soft parameter sharing



https://ruder.io/multi-task/

Распознавание сцен



Places2 scenes dataset, http://places2.csail.mit.edu

Распознавания событий на фотографии

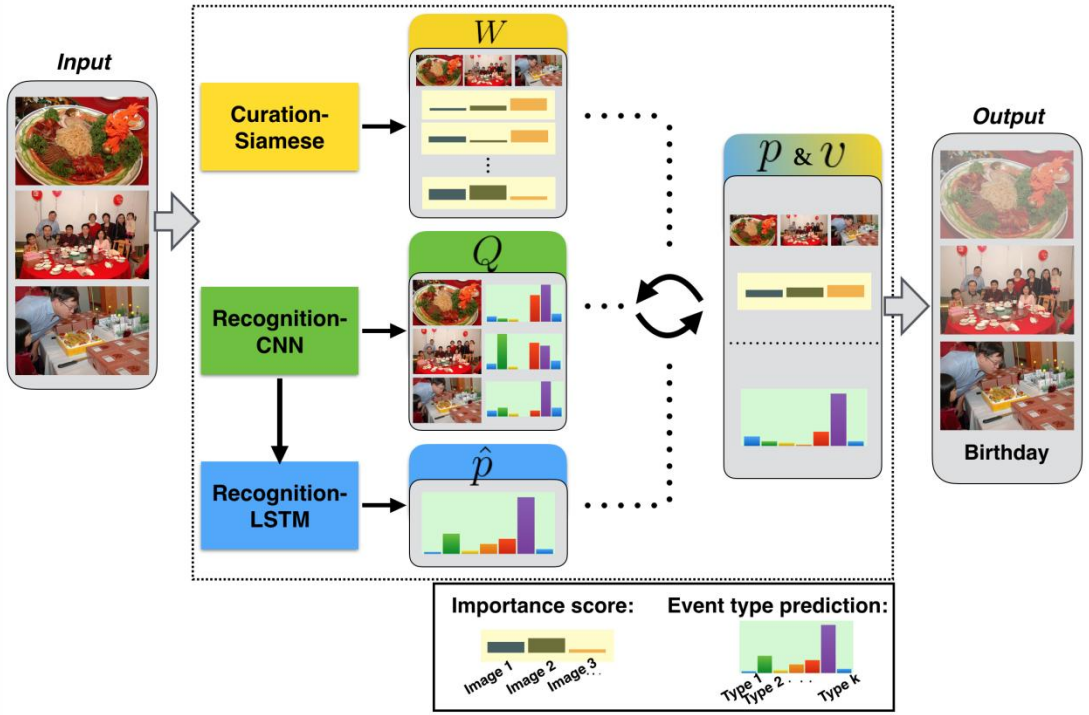
“An event captures the complex behavior of a group of people, interacting with multiple objects, and taking place in a specific environment. Images from the same event category may vary even more in visual appearance and structure” (Wang et al, IJCV 2018)

WIDER (Web Image Dataset for Event Recognition)

PEC (Photo Event Collection)

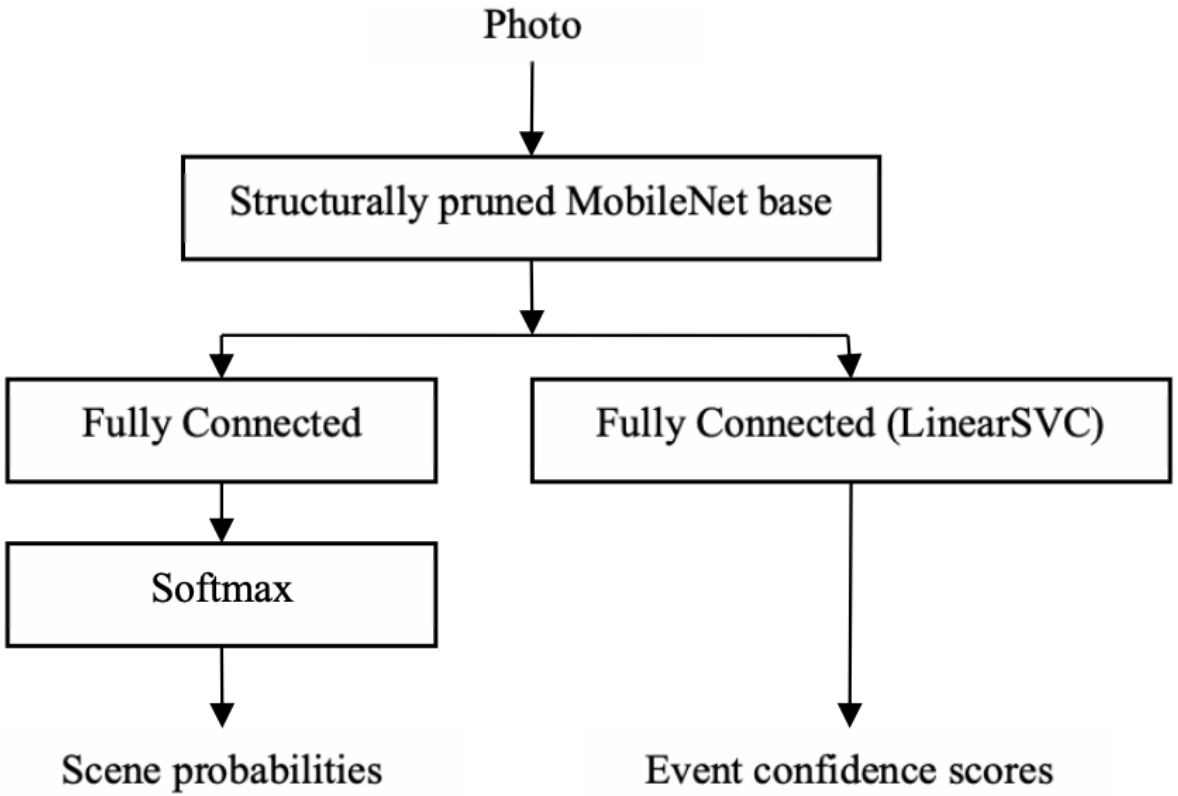
Распознавания событий в фотоальбоме

Image-set recognition: альбом фотографий X_t, $t \in \{1,\dots T\}$ необходимо отнести к одному из C классов событий. Задано обучающее множество из N альбомов: n-й й альбом с заданной меткой класса $c_n \in \{1,\dots C\}$ состоит из множества изображений $\{X_n(1),\dots X_n(L_n)\}$



Wang, Y. et al. Recognizing and Curating Photo Albums via Event-Specific Image Importance, BMVC17

Multi-task scene/event recognition



Перейдем к примерам

https://github.com/HSE-asavchenko/MADE-mobile-image-processing/tree/master/lesson5/src