Nizhny Novgorod State University
Institute of Information Technologies, Mathematics and Mechanics
Department of Computer software and supercomputer technologies

Educational course
«Modern methods and technologies
of deep learning in computer vision»

Instance segmentation of images using deep learning

Supported by Intel

Svetlana Nosova

Content

Goals

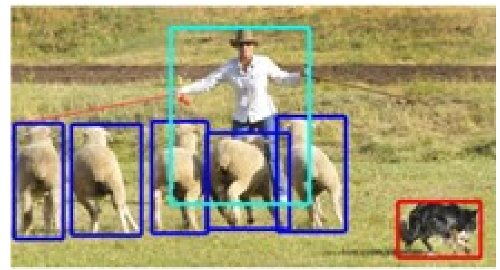
- ***The goal*** is to study deep models for solving problem of instance segmentation

INSTANCE SEGMENTATION PROBLEM STATEMENT

Problem statement (1)

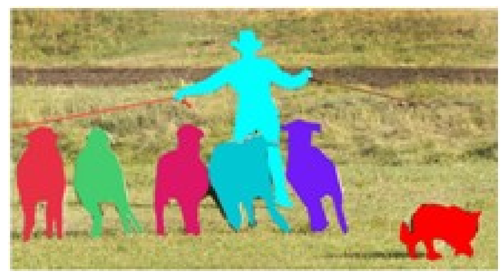
- The problem of instance segmentation is to match each image pixel with the class of objects and with image object number to which this pixel belongs
- Object detection and semantic segmentation results should be combined

Classification



Object detection

Semantic segmentation



Instance segmentation

Lin T.Y., et al. Microsoft COCO: Common objects in context-– 2014. – [https://arxiv.org/pdf/1405.0312]

Problem statement (2)

- Comparison with semantic segmentation:
  - It’s also pixel classification task, but the mark of every pixel responds object class and object number
  - Comparison with object detection:
- More accurate object borders detection in comparison with bounding boxes
  - non-maximum suppression is more accurate

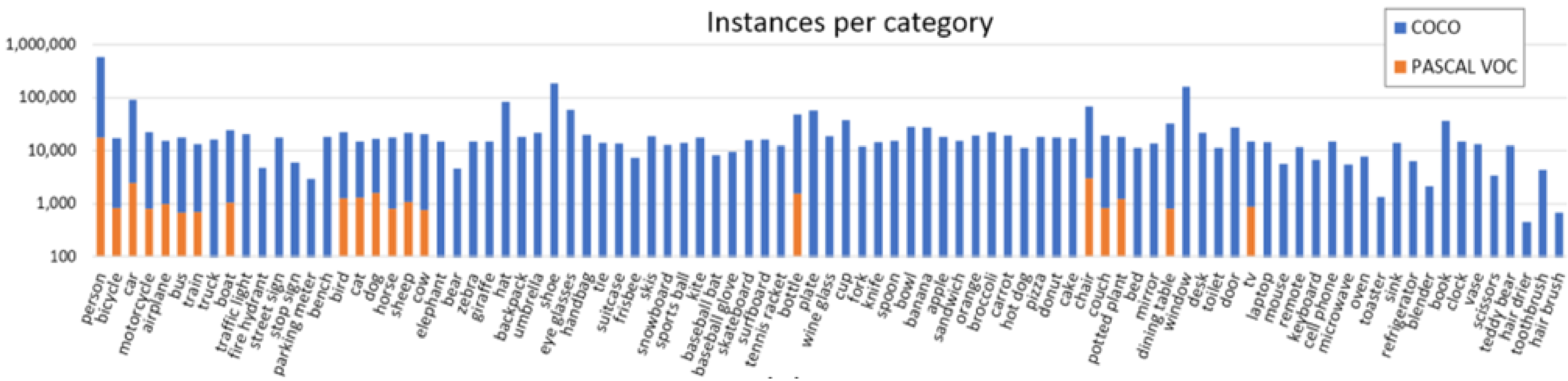
PUBLIC DATASETS

Public datasets (1)

Dataset	Number of images in train dataset	Number of images in test dataset	Number of classes
PASCAL VOC 2012 [http://host.robots.ox.ac.uk/pascal/VOC/voc2012]	9 963	1 447	20
MS COCO’15 [http://mscoco.org]	80 000	40 000	80
Sun-RGBD [http://rgbd.cs.princeton.edu]	10 355	2 860	37
Cityscapes [https://www.cityscapes-dataset.com]	2 975	500	19

MS COCO’15 (1)

- MS COCO’15 is the largest public dataset of real-life images (similar to PASCAL VOC) by the number of object classes (80 categories) and the number of images; each category contains a significant number of images (approximately equal number of objects for each class)



* Lin T.Y., et al. Microsoft COCO: Common objects in context // Lecture Notes in Computer Science. – 2014. — [https://arxiv.org/pdf/1405.0312].

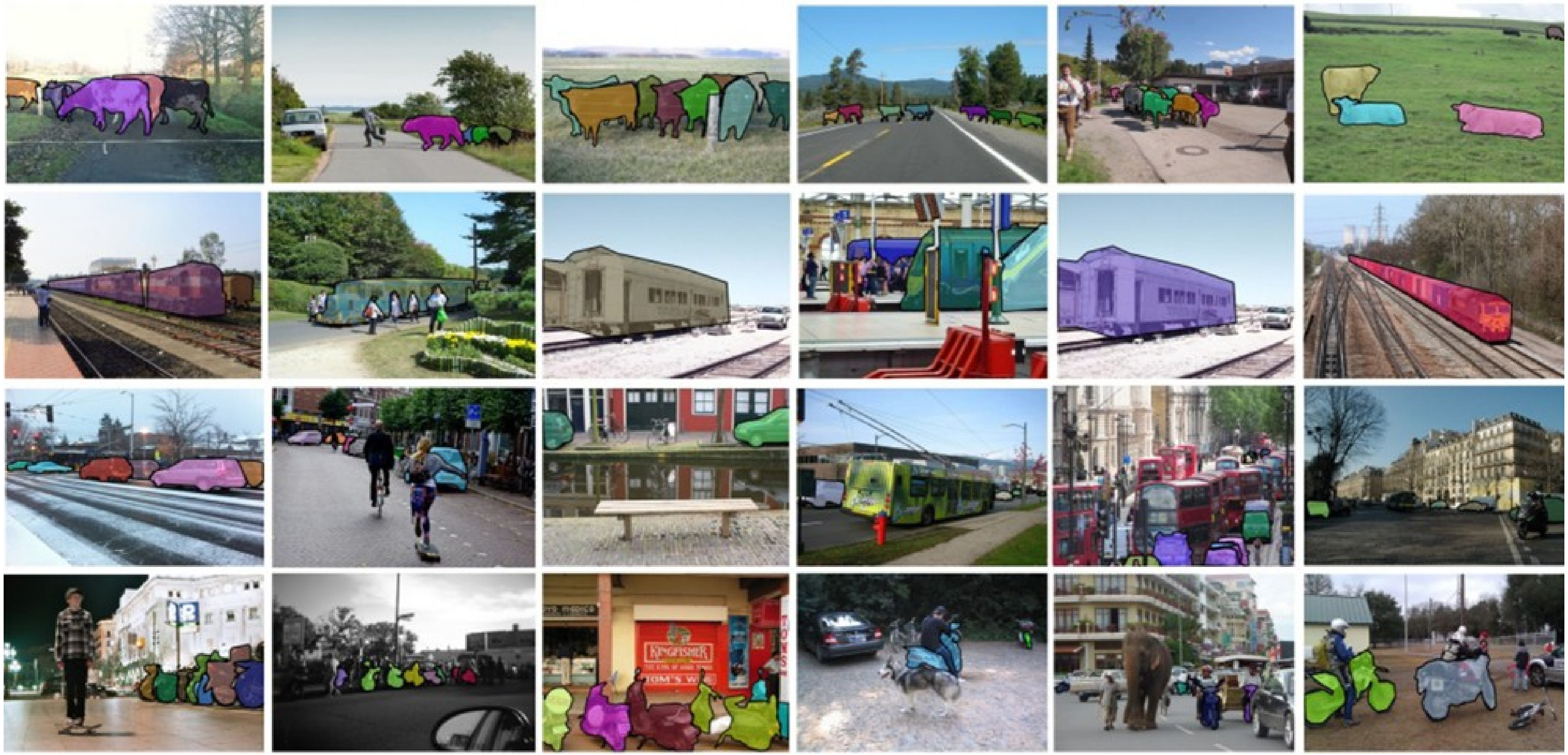
MS COCO’15 (2)

Cow

Train

Car

Motobike



* Lin T.Y., et al. Microsoft COCO: Common objects in context. – 2014. – [https://arxiv.org/pdf/1405.0312].

SUN RGB-D

- Object classes are relatively few

bedroom

conference room

classroom

home office

Song S., Lichtenberg S. P., Xiao J. Sun rgb-d: A rgb-d scene understanding benchmark suite.– 2015. – [https://rgbd.cs.princeton.edu/]

Applications

- medical diagnostics
- object parameters research
- scene understanding and scene reconstruction:
  - aircraft autodriving
  - car autodriving
  - scene reconstruction
  - scene modelling
  - placing virtual objects on the scene

QUALITY METRICS

Average Precision (1)

- **IoU (Intersection Over Union**) - is a ratio of overlapping the segmented and labeled (groundtruth) masks (Intersection over Union)

- **TP** - is a number of segmented objects for which intersection over union is not less a certain threshold t (we think of the object is segmented correctly, it is a true positive)

- **FP** - is a number of segmented objects for which intersection over union is less than t (the object was segmented incorrectly), or the object was segmented more than once (false positives)

- **FN** - is a number of unsegmented objects (false negatives)

Average Precision (2)

- The threshold value usually is chosen equal to 0.5

- ***Precision*** is a ratio of true positives by the overall number of detections

$$Precision = p = \frac{TP}{TP + FP}$$

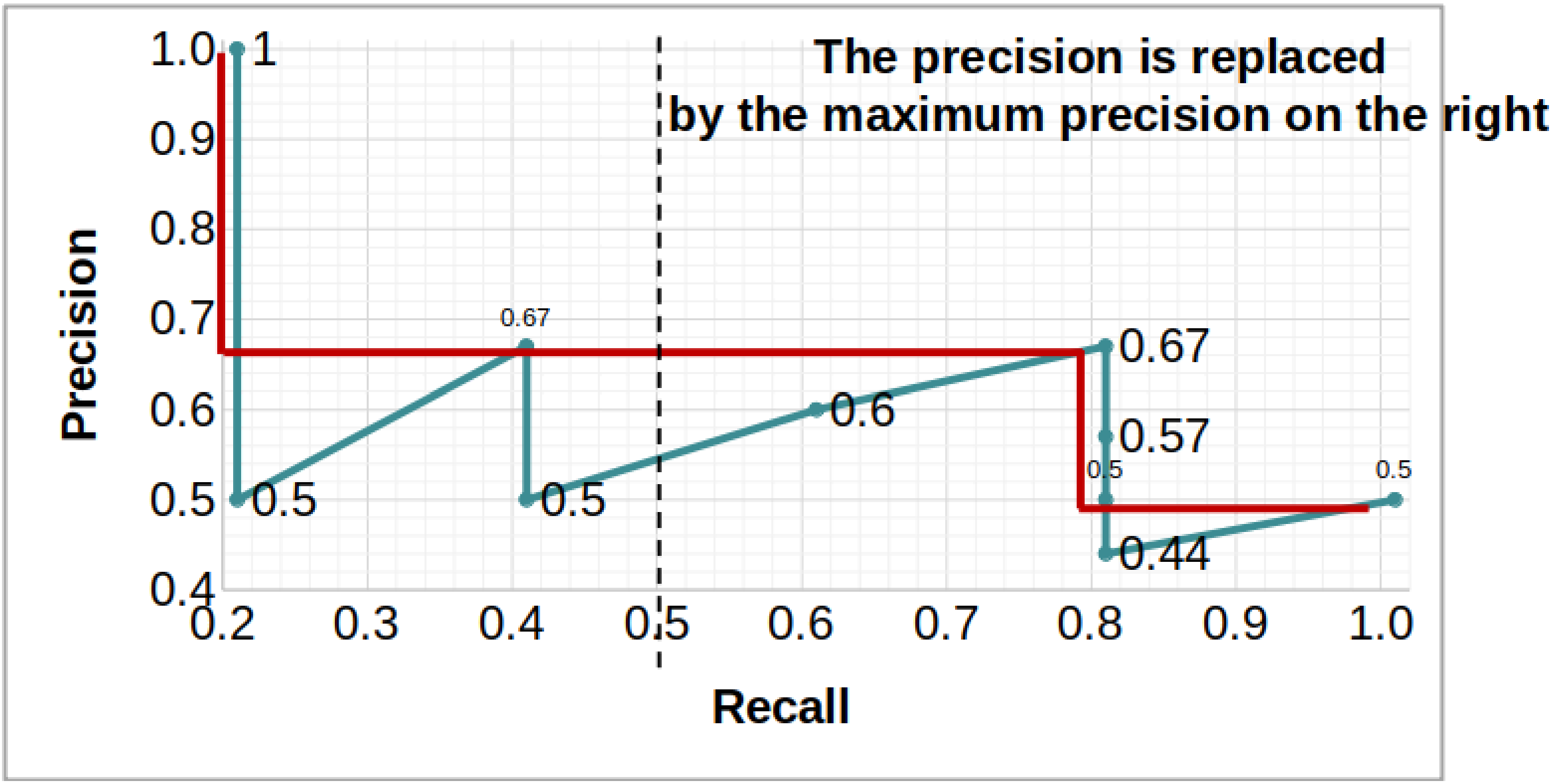
- ***Recall*** is a ratio of true positives by the overall number of objects

$$Recall = r = \frac{TP}{TP + FN}$$

- ***meanAP*** – mean AP from all object classes

Average precision (3)

- Example of calculating average precision:
  - Calculating the area under the zigzag curve, i.e. interpolating and calculating the area under the stepped curve



DEEP MODELS FOR INSTANCE SEGMENTATION

Deep models (1)

Sliding window

- ***DeepMask (2015), Instance FCN (2016)***
  - Pinheiro P. O., Collobert R., Dollár P. Learning to segment object candidates //Advances in neural information processing systems. – 2015. – [https://arxiv.org/pdf/1506.06204.pdf]
- ***Instance FCN (2016)***
  - Dai J. et al. Instance-sensitive fully convolutional networks //European Conference on Computer Vision. – 2016. – [https://arxiv.org/pdf/1603.08678.pdf]

Two-stage models

- ***MNC (2016)***
  - Dai J., He K., Sun J. Instance-aware semantic segmentation via multi-task network cascades //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. – 2016. – [https://arxiv.org/pdf/1512.04412.pdf]

Deep models (2)

Two-stage models

- ***Mask R-CNN (2017)***
  - He K. et al. Mask r-cnn. – 2017. – [https://openaccess.thecvf.com/content_ICCV_2017/papers/He_Mask_R-CNN_ICCV_2017_paper.pdf]
- ***Mask Scoring R-CNN (2019)***
  - Huang Z. et al. Mask scoring r-cnn. – 2019. – [https://arxiv.org/pdf/1903.00241.pdf]
- ***PANet (2018)***
  - Liu S. et al. Path aggregation network for instance segmentation. – 2018. – [https://arxiv.org/pdf/1803.01534.pdf]

Deep models (3)

One-stage models

- ***YOLACT (2019)***
  – Bolya D. et al. Yolact: Real-time instance segmentation //Proceedings of the IEEE international conference on computer vision. – 2019. – [https://arxiv.org/pdf/1904.02689.pdf]
- ***YOLACT++ (2019) -***
  – Bolya D. et al. Yolact++: Better real-time instance segmentation. – 2019.– [https://arxiv.org/pdf/1904.02689.pdf]
- ***CenterMask (2020)***
  – Lee Y., Park J. CenterMask: Real-time anchor-free instance segmentation. – 2020. – [https://arxiv.org/pdf/1911.06667.pdf]
- ***SOLO (2020)***
  – Wang X. et al. Solo: Segmenting objects by locations //arXiv preprint arXiv:1912.04488. – 2020. – [https://arxiv.org/pdf/1912.04488.pdf]

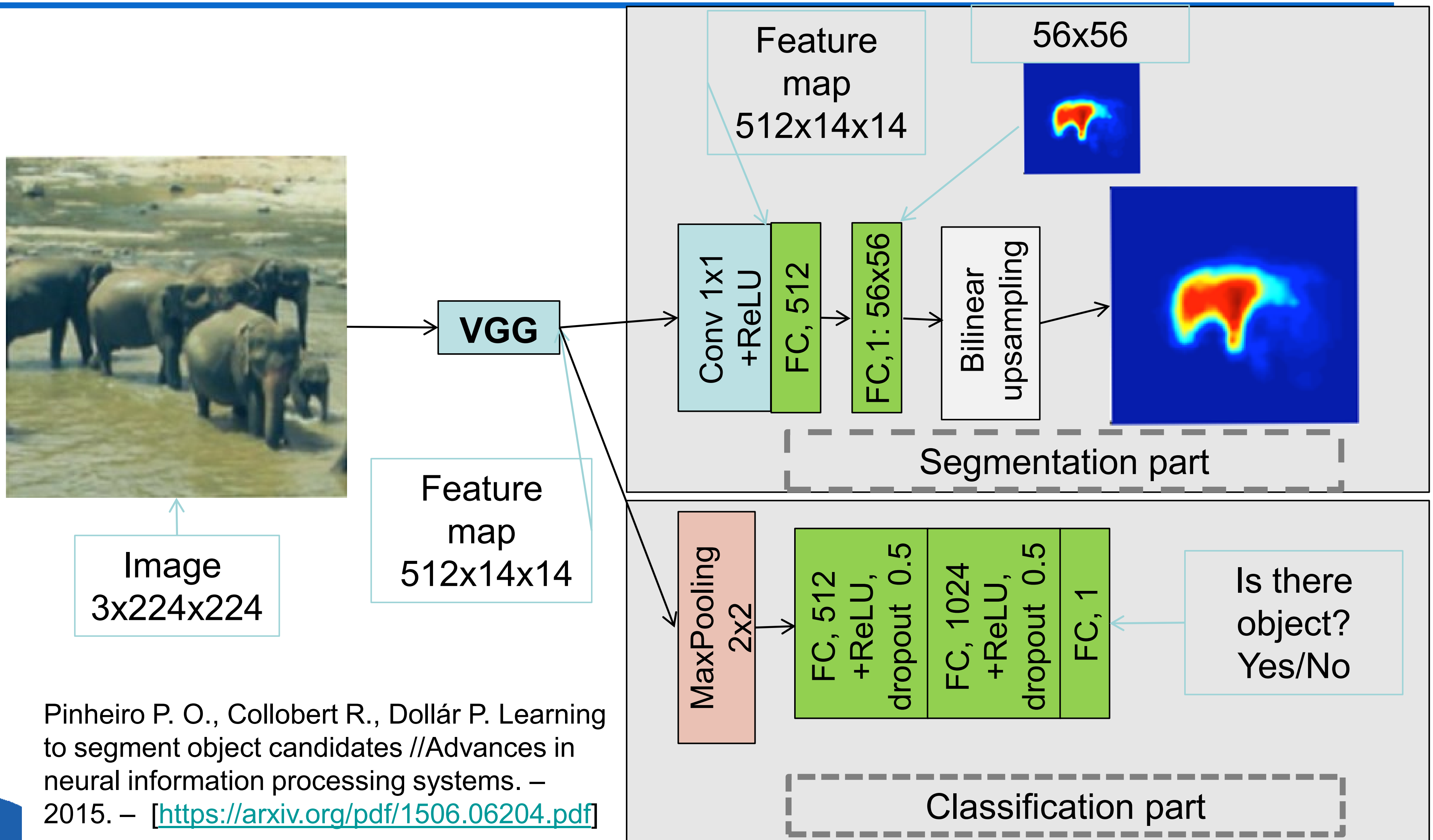
DeepMask (1)

- DeepMask was developed in 2015
- It’s one of the first decision for instance segmentation task using deep learning
- For the feature map construction ImageNet pretrained VGG-A is used
  - 5-pooling layer and fc-layers are deleted
- Input image size is 224x224
- There are two stages: ***classification*** and ***segmentation***

DeepMask (2)

- This deep learning model was developed to segment object on image patch
- Object placed in the center in fully included on image patch
- Segmentation is applied for the following variants:
  - different image resolution ratio ([1/4, 2])
  - different position of sliding windows (stride = 16)
- In training process batch size is 32
- Number of parameters - 75M

DeepMask (3)



Pinheiro P. O., Collobert R., Dollár P. Learning to segment object candidates //Advances in neural information processing systems. – 2015. – [https://arxiv.org/pdf/1506.06204.pdf]

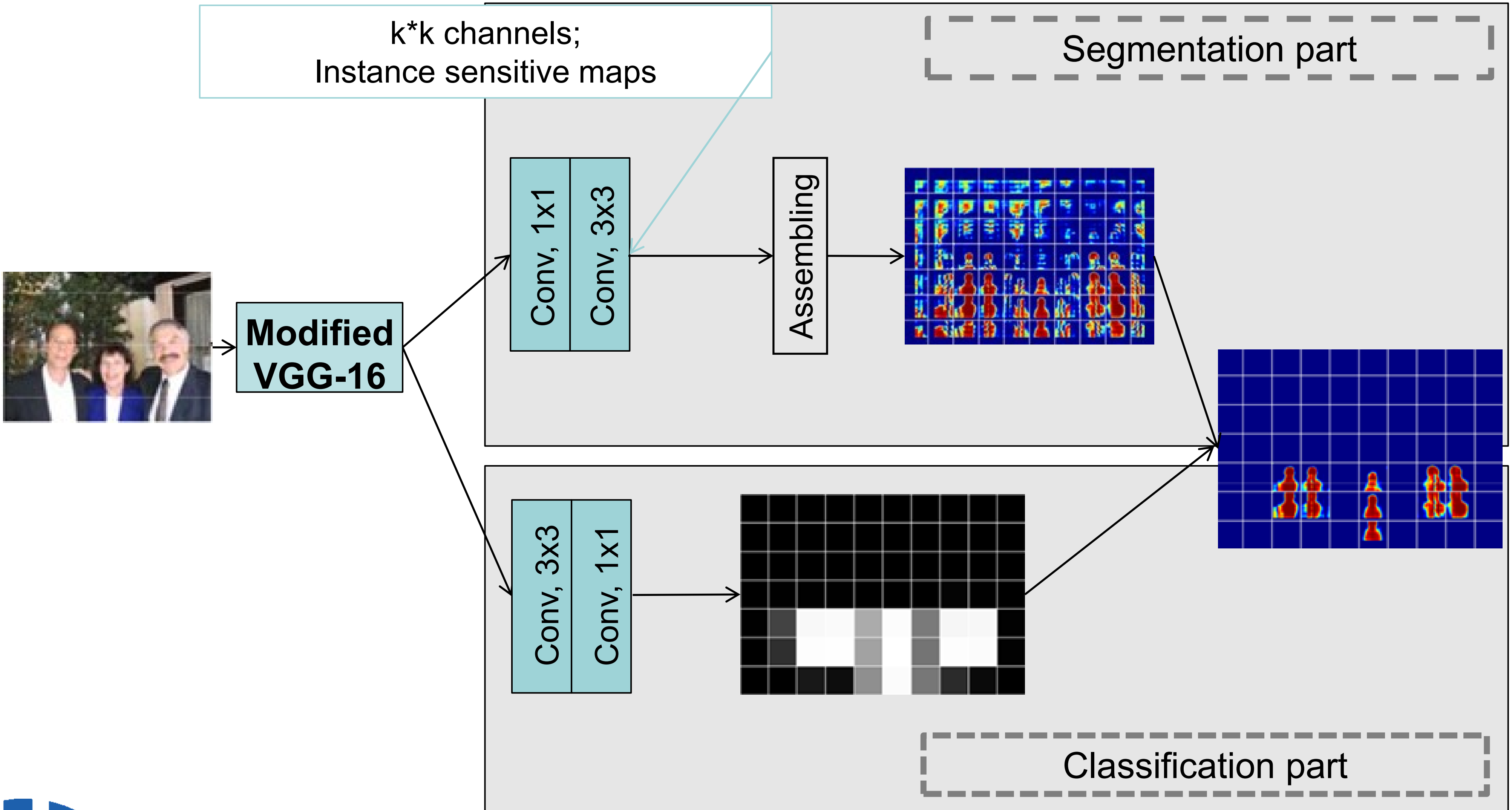
DeepMask (4)

- ***Segmentation*** branch is for 1-object binary segmentation. It includes the following sequence of layers: conv 1x1-, 1 fc-layer; fc-layer for classification pixels of 56x56 map. There is no ReLU after fc-layers. To get 224x224 segmentation result bilinear interpolation is used
- ***Classification*** branch solves binary classification task of object presence (is there object? or not). There are 2 maxpooling 2x2 layers, 2x dropout fc-layers (with 512 и 1024 neurons of inner layer). ReLU activation is applied after fc-layers. The output is 1 value. It’s reliability of object presence

Instance FCN (1)

- ***Instance-sensitive Fully Convolutional Networks***
- It’s developted to solve instance segmentations task for various size images
- Model architecture contains two part: segmentation branch and classification branch
- Classification and segmentation branches are FCN-models
- To take feature map 13 convolution layers of ImageNet pretrained VGG-16 are used
  - modification: stride = 1 for maxpooling-4 (not 2). As a results the size of feature map is larger
- In 5-th conv-layer dilated-convolutions with stride = 8 are used

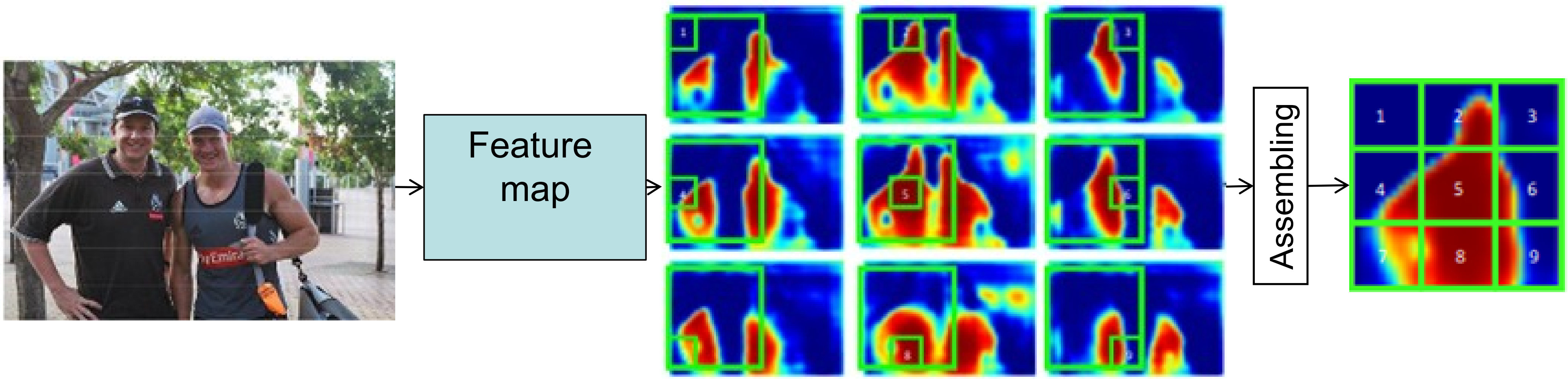
Instance FCN (2)



Dai J. et al. Instance-sensitive fully convolutional networks. – 2016. – [https://arxiv.org/pdf/1603.08678.pdf]

Instance FCN (3)

Instance sensitive maps generation



Instance FCN (4)

- Segmentation branch. The stage of calculation instance sensitive feature map:
  - conv1x1 with ReLU-activation is used to transfrom feauture map
  - conv3x3 is used to generate instance sensitive map;as a result, k^2 channels are generated; it corresponds k^2 different locations of sliding window center
- Assembling module is applied on instance sensitive feature map using mxm sliding window(21x21). Every element of result map is copied from corresponding layer of input feature map

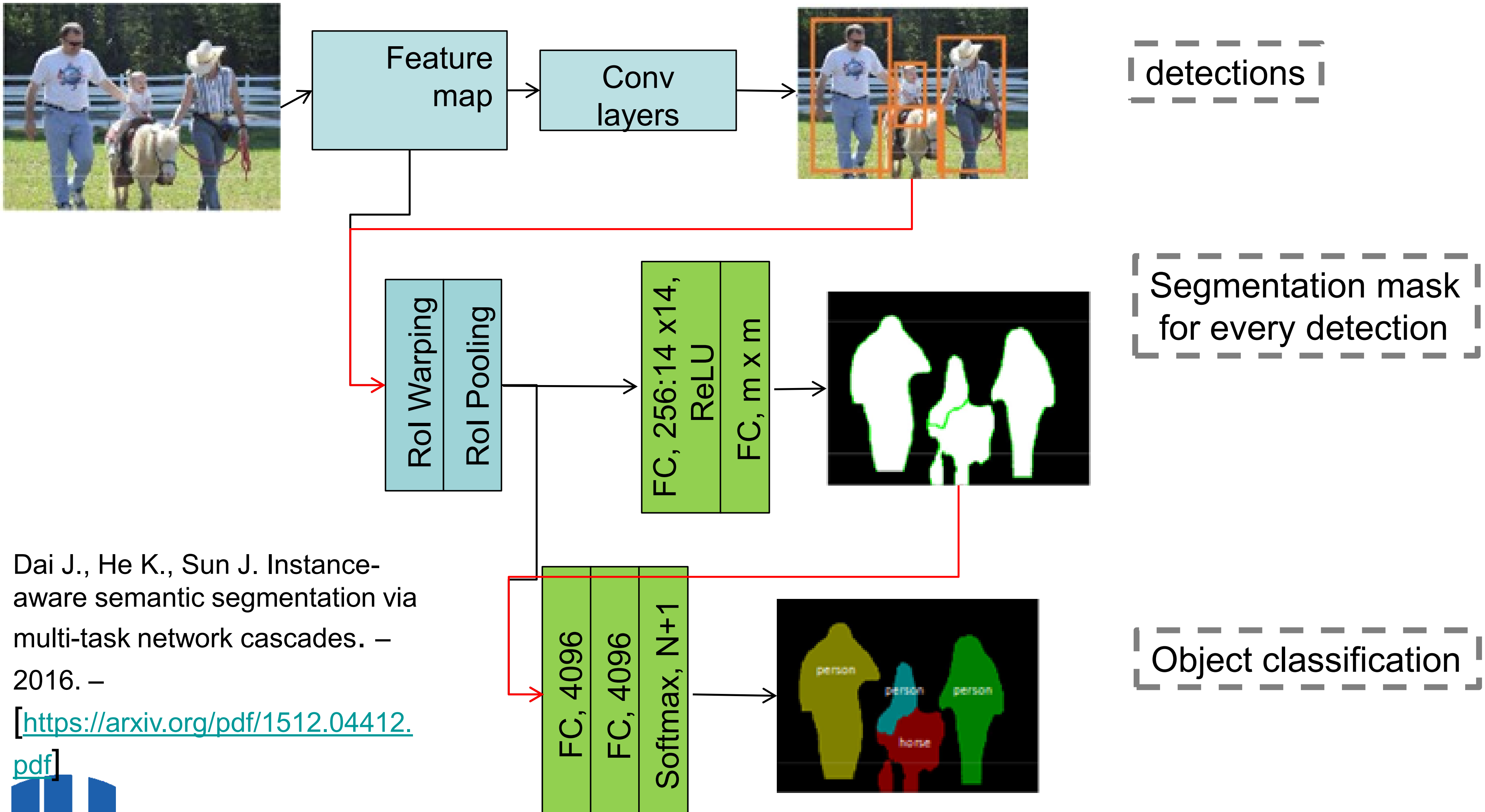
Instance FCN (5)

- Classification branch contains the following layers:
  - conv3x3 with ReLU-activation
  - conv1x1
  - Sliding mxm window for generation reliability of object presence
- Reliability of object presence is calculated

MNC (1)

- ***Multi-task Network Cascades***
- The winner if the 2015 COCO competition
- Cascade includes the following branches::
  - ***detection branch***
  - ***segmentation branch***
  - ***classification of instance branch***
- Current model backbone is VGG-16. It’s shared part for every branch
- For initialization of backbone weights and two fc-layers (4096 elements) ImageNet pretrained VGG model is used
- For another layers random initialization is used

MNC (2)



Dai J., He K., Sun J. Instance-aware semantic segmentation via multi-task network cascades. – 2016. – [https://arxiv.org/pdf/1512.04412.pdf]

MNC (3)

- Object detection branch(region-of-interest subtraction) is realized using RPN-model. Non-maximum suppression in used
- ***Segmentation branch*** used feature map and detected RoI-set. Segmentation is realized for every RoI
  – RoI warping stage extracts RoI-corresponding features from image feature map
  – RoI pooling is used to get RoI-feature map of fixed size(14*14)
  – 2 fc-layers
- The output is 28*28 feature map

MNC (4)

- ***Classification branch*** used image feature map,RoI candidats, segmentation mask
  - ROI warping is used to get RoI feature map
  - ROI pooling is used to get feature map of fixed size (14*14).
  - Using segmentation mask non-object feature map elements are setted in 0
  - 2 fc-layes (4096 neurons in each layer) are used for classification

Mask R-CNN (1)

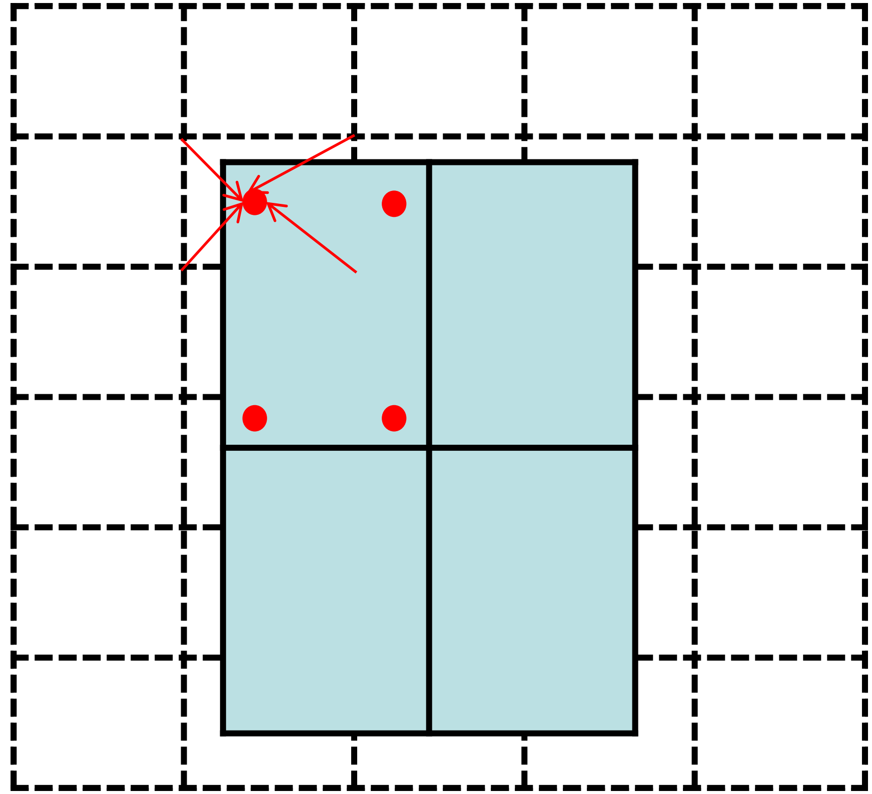
- It’s extension of Fast R-CNN
- Instance segmentation decomposed on two stages
  - ***Object detection*** is to get bounding box for every object
  - ***Binary segmentation***. It’s applied for every RoI and its parallel to classification branch and bounding box regression. The output is k m×m binary masks. k - number of classes on every region of interest
- Mask R-CNN is flexible structure for object-level vision task. It’s used for different tasks, such as key-point detection and solving pose-estimation problem

Mask R-CNN (2)

- To take ***object candidates*** RPN model is used
- ***Segmentation mask*** and object ***class*** are predicted using independent from each other way
- *RoIAlign* layer is used. It saves accurate values of RoI feature map elements without quatonization
  - Bilinear interpolation is used to calculate accurate values of RoI mapin 4-point regular grid for every RoI-element
  - Maximum or average value is calculated for every 4-point set
  - Accuracy of segmentation result up to 10–50%

Mask R-CNN (3)

- ***RoIAlign:***
  - dotted lines is image feature map
  - solid lines - RoI
  - RoIAlign calculates every checked point value using bilinear interpolation from nearest points on image feature map
  - result value of RoI feature map element is average or maximum value from checked points

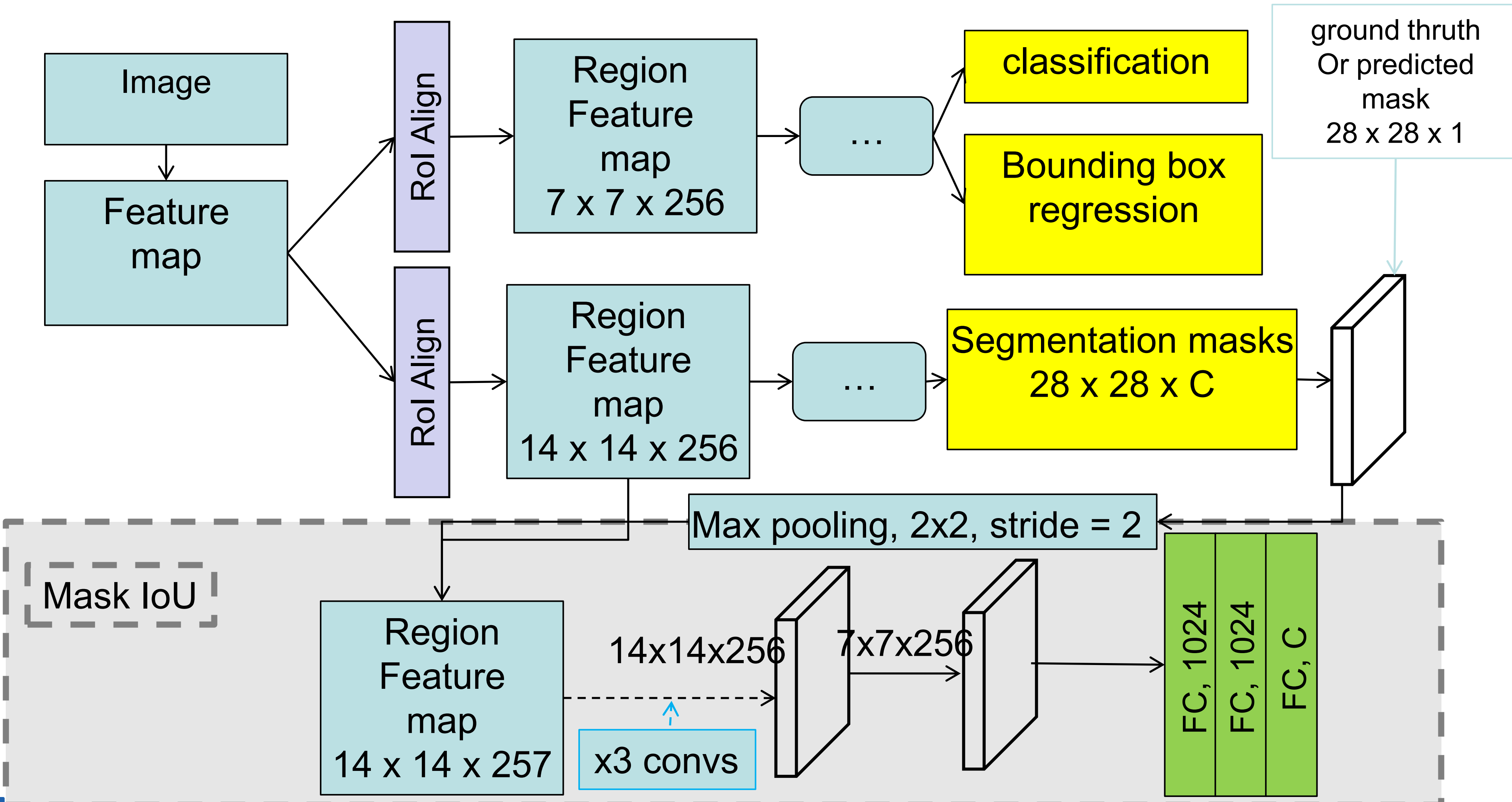


He K. et al. Mask r-cnn. – 2017. –. https://openaccess.thecvf.com/content_ICCV_2017/papers/He_Mask_R-CNN_ICCV_2017_paper.pdf

Mask scoring R-CNN (1)

- Based on ***Mask R-CNN***
- It's developted to solve the following problem:
  - the most accurate segmentation mask has the highest probability value
- The ideal segmentation mask:
  - is on 100% same with labeled mask
  - has correct classification result
- Additional branch Mask IoU was presented to approximate mask segmentation result to ideal

Mask scoring R-CNN (2)



Huang Z. et al. Mask scoring r-cnn. – 2019. – [https://arxiv.org/pdf/1903.00241.pdf]

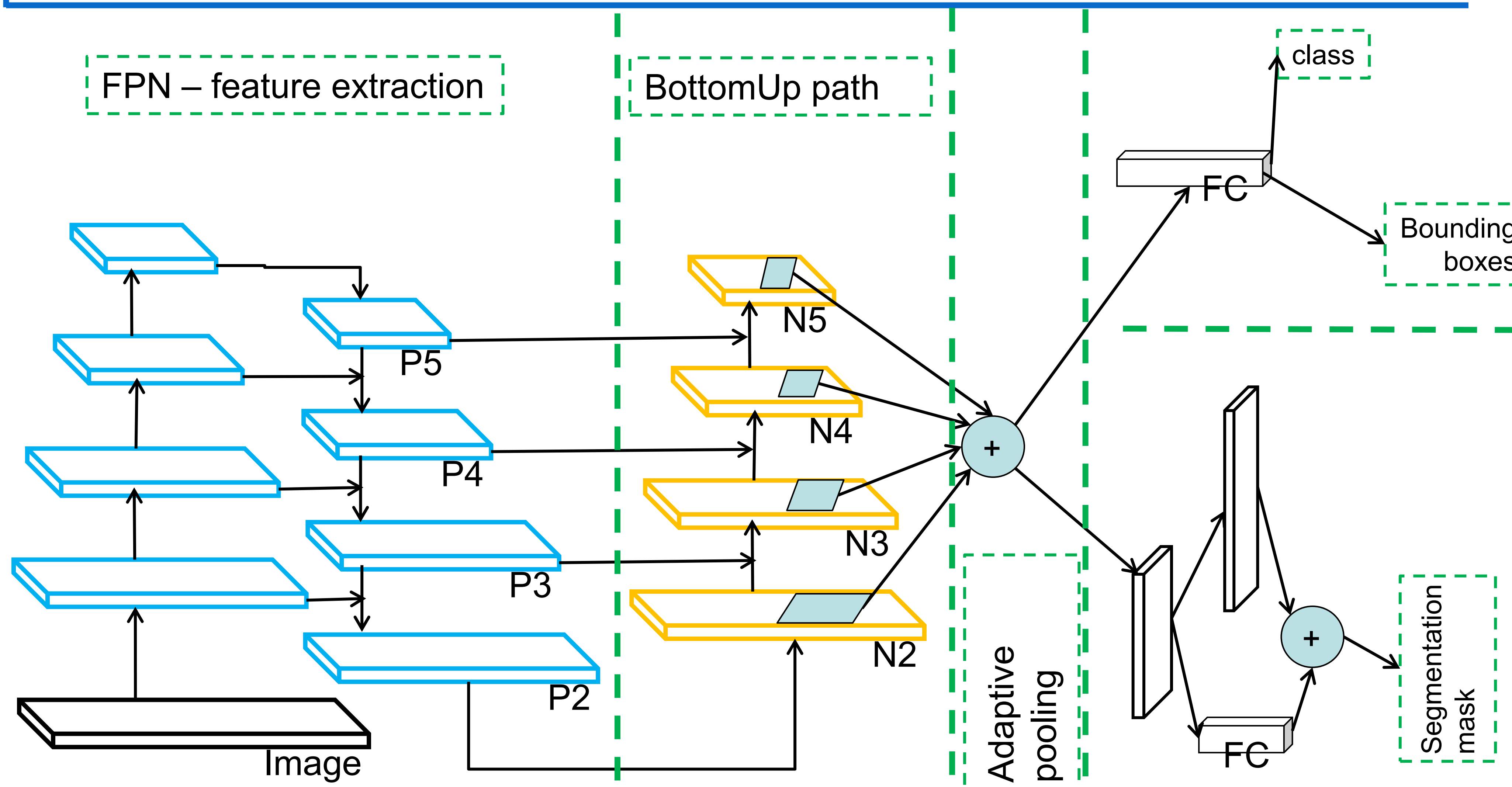
Mask scoring R-CNN (3)

- ***Mask IoU***:
  - RoI Align align feature map is concatenated with predicted mask
  - max-pooling 2x2 is applied
  - model branch consists from 4 convolution layers (3x3) and 3 fc-layers
- In the training stage the input of MaskIoU branch is intersection over union between predicted and labeled mask
  - In the inference time the MaskIoU is used only to calculate correct object class
  - New reliability values are calculated by element-wise multiplication of classification-stage results and Mask IoU-stage results

PANet (1)

- ***Path Aggregation Network*** is extension of Mask R-CNN to up segmentation quality
- The winner of ***COCO 2017 Challenge (Instance Segmentation)***
- To take feature map FPN model is used
- To improve localization and segmentation accuracy the bottom-up augmentation path is improved
  - High-level features are expanded by low-level features
  - The direct path between high-level features and low-level features are constructed

PANet (2)



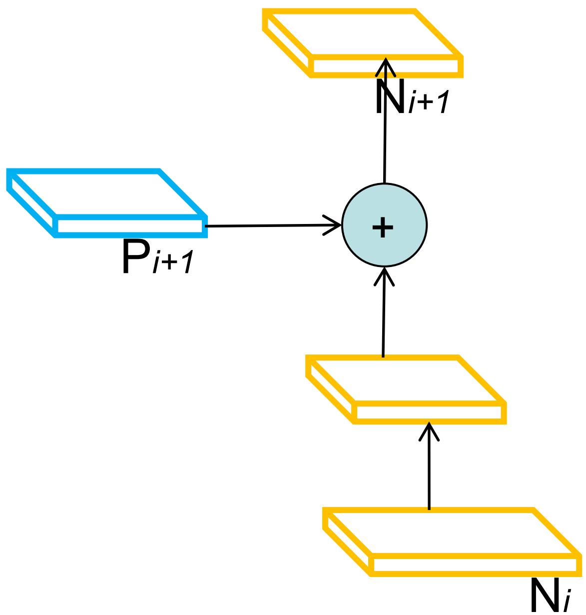
Liu S. et al. Path aggregation network for instance segmentation. – 2018. – [https://arxiv.org/pdf/1803.01534.pdf]

PANet (3)

- High-level features describes general, universe data features
- Low-level features describes local data features

PANet (4)

- ***Structure block of bottom-up augmentation path***
- The output of this stage is pyramid of feature maps N2, N3, N4, N5 (N2=P2)



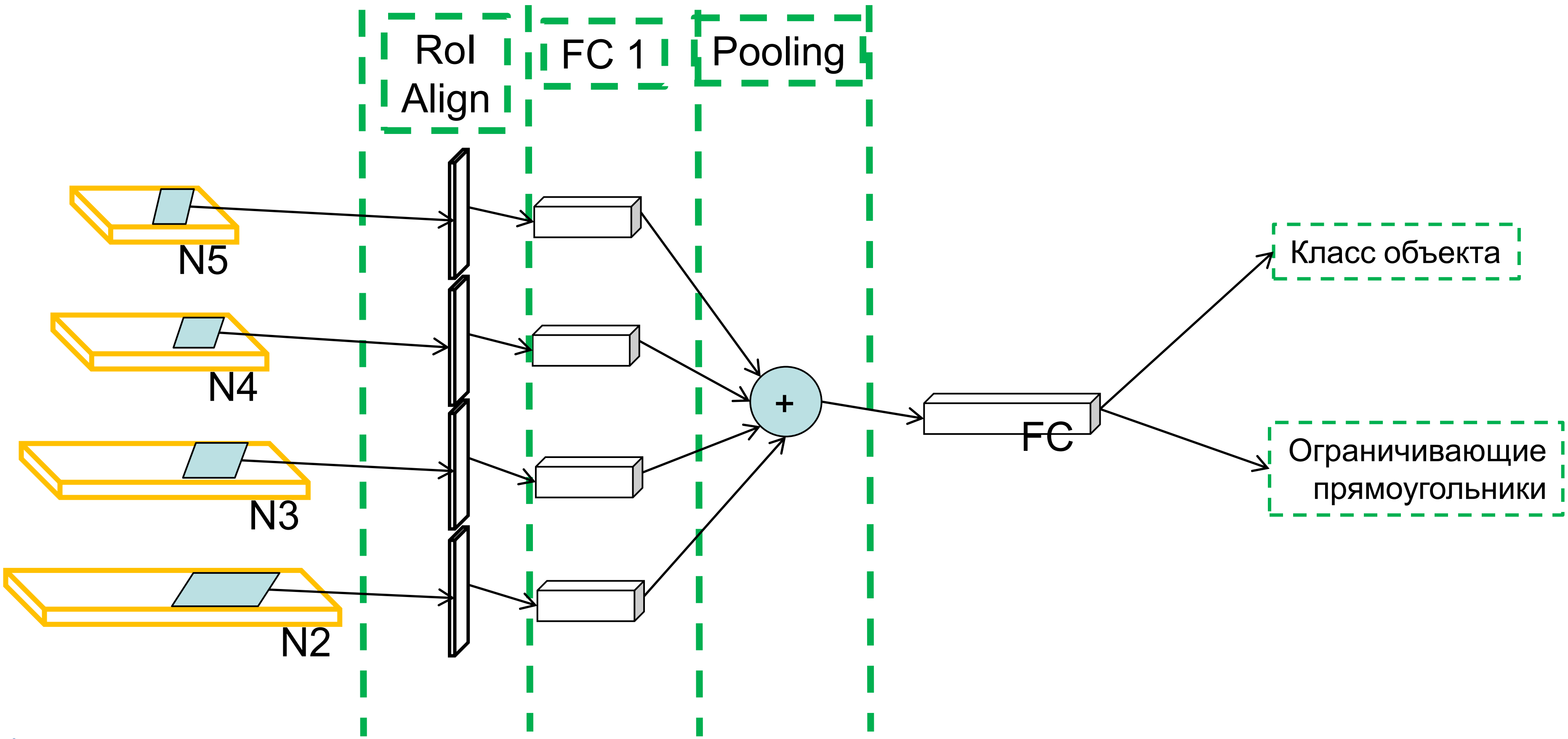
Liu S. et al. Path aggregation network for instance segmentation. – 2018. – [https://arxiv.org/pdf/1803.01534.pdf]

PANet (5)

- Adaptive pooling structure:
  - region feature map on every level of pyramid features are extracted
  - RoIAlign is applied to take accurate valued of feature elements
  - feature maps from different pyramid level are concatenated
  - final map is calculated using maximum or average calculation between pixels on every pyramid level
- Final feature map is used on classification, detection, segmentation steps

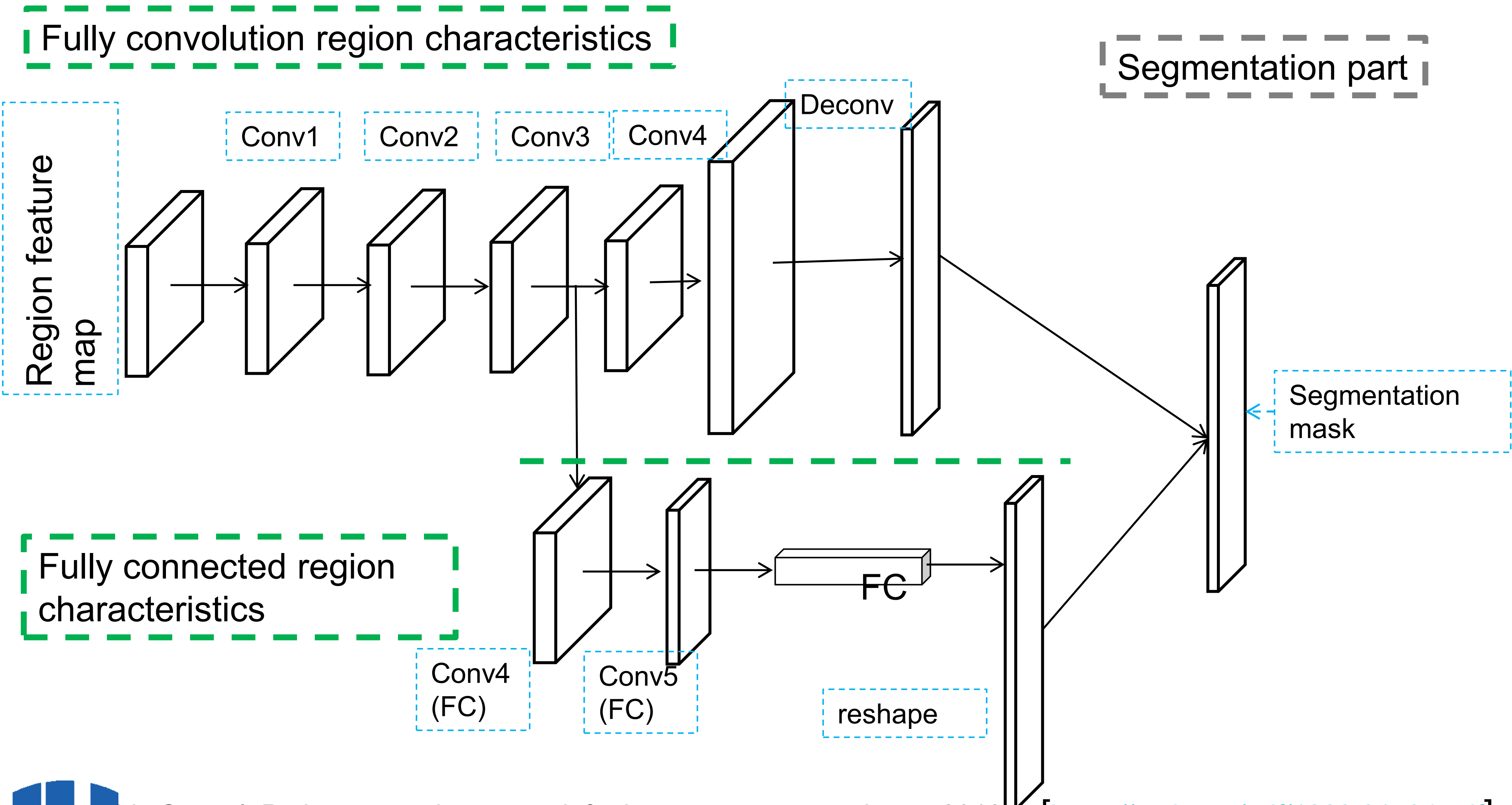
PANet (6)

- *Adaptive pooling structure*



Liu S. et al. Path aggregation network for instance segmentation. – 2018. – [https://arxiv.org/pdf/1803.01534.pdf]

PANet (7)

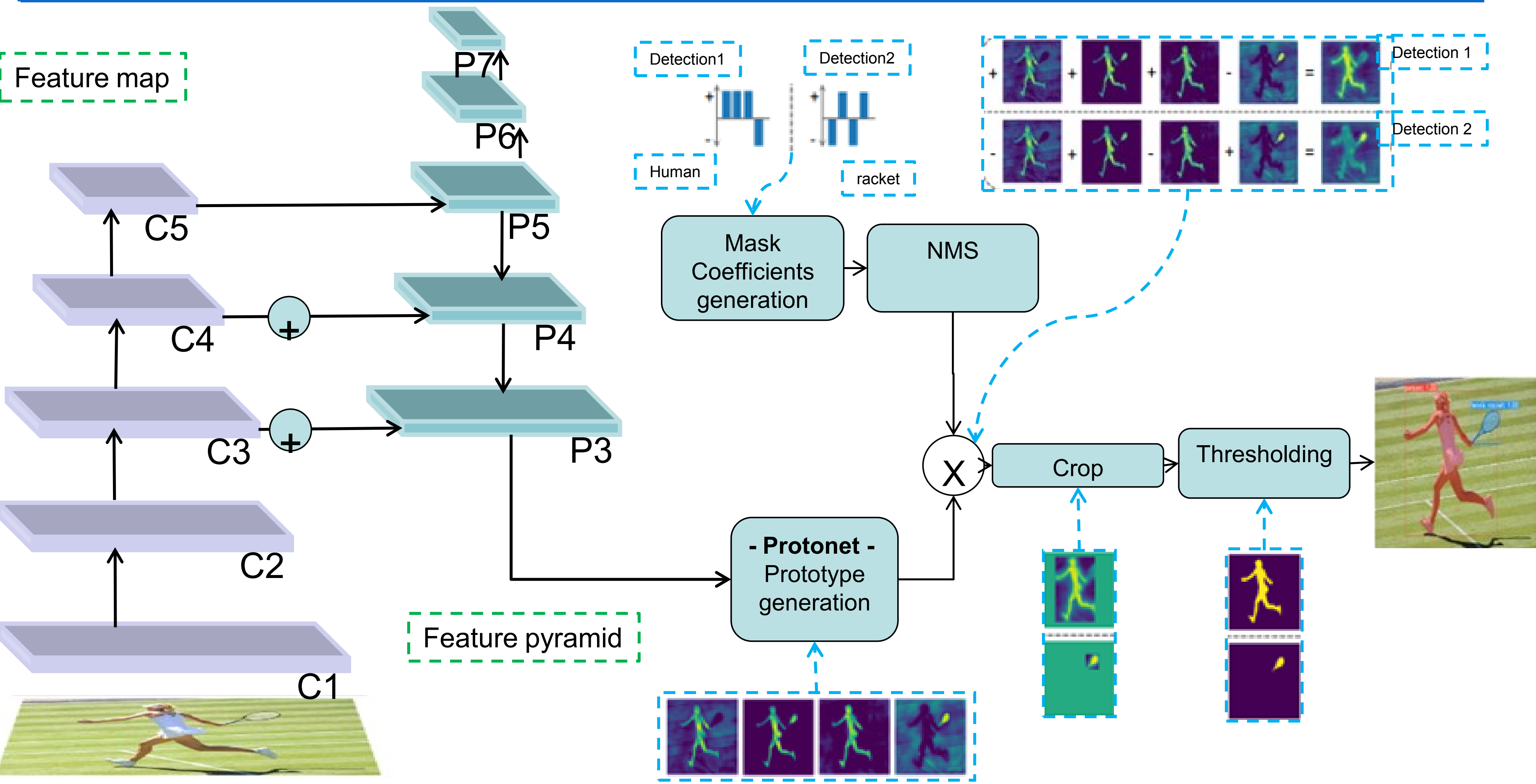


Liu S. et al. Path aggregation network for instance segmentation. – 2018. – [https://arxiv.org/pdf/1803.01534.pdf]

YOLACT (1)

- It was developed for real-time object segmentation
- The main idea is to add branch to calculate binary segmentation mask (sa in Mask R-CNN) but without localization step
- Model architecture is allowed to solve 2 different task. Results of two different tasks are combined to form final segmentation results
- It's extension anchor-based models for object detection task
- To produce feature map FPN model is used
- Используется ResNet-101, размер входного изображения 550×550
- Mask:
  - Linear combination of masks with mask coefficients is calculated
- Final mask is cropped by predicted bounding boxThe threshold binarization is applied to form final segmentation mask

YOLACT (2)

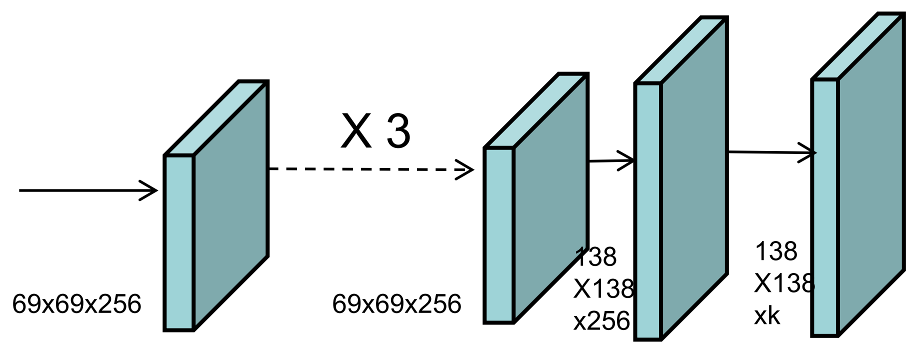


Bolya D. et al. Yolact: Real-time instance segmentation. – 2019. – [https://arxiv.org/pdf/1904.02689.pdf]

YOLACT (3)

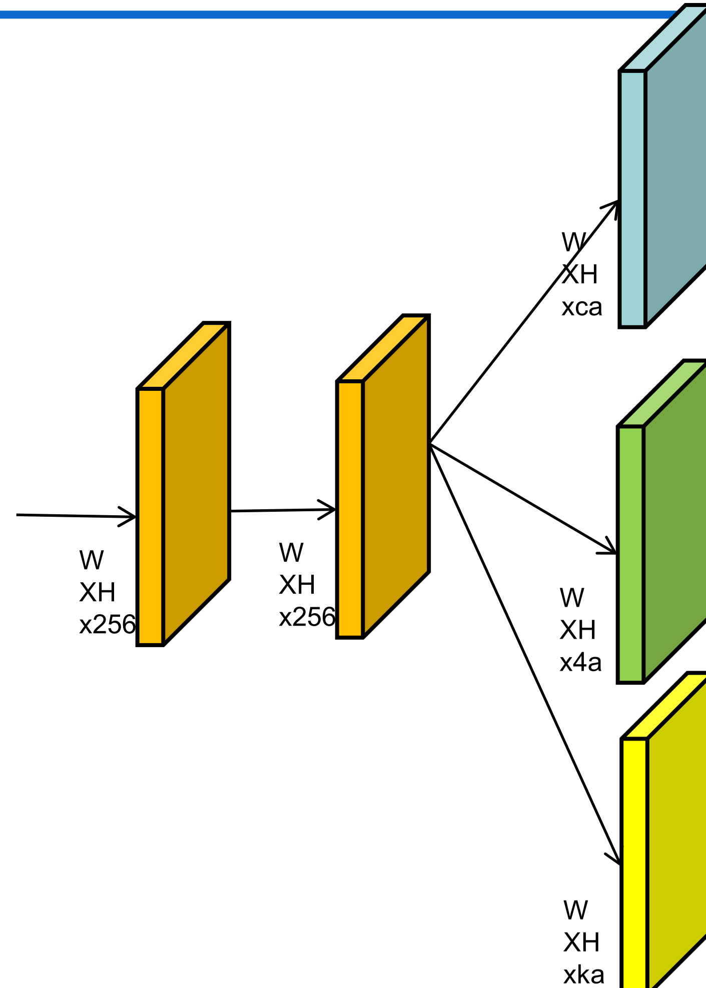
- Branch of ***mask prototype generation***:
  - Generate k predicted mask with size of input image
  - It’s FCN with k-channel last layer
  - It’s also contained 3 convolution layer 3x3
  - Last convolution layer has 1x1 convolutions
  - ReLU-activation is used
  - Upsampling is used to up feature size

Bolya D. et al. Yolact: Real-time instance segmentation. – 2019. – [https://arxiv.org/pdf/1904.02689.pdf]



YOLACT (4)

- ❑ ***Mask coefficients map***:
  - – New branch is added
  - – k mask coefficient are predicted, one for every prototype. For every anchor 4+c+k vector is calculated (offset - 4, c - class prediction, k - mask prediction)
  - – tanh-activation is used
- ❑ Non-maximum suppression procedure and thresholding are applied for anchors



Bolya D. et al. Yolact: Real-time instance segmentation. – 2019. – [https://arxiv.org/pdf/1904.02689.pdf]

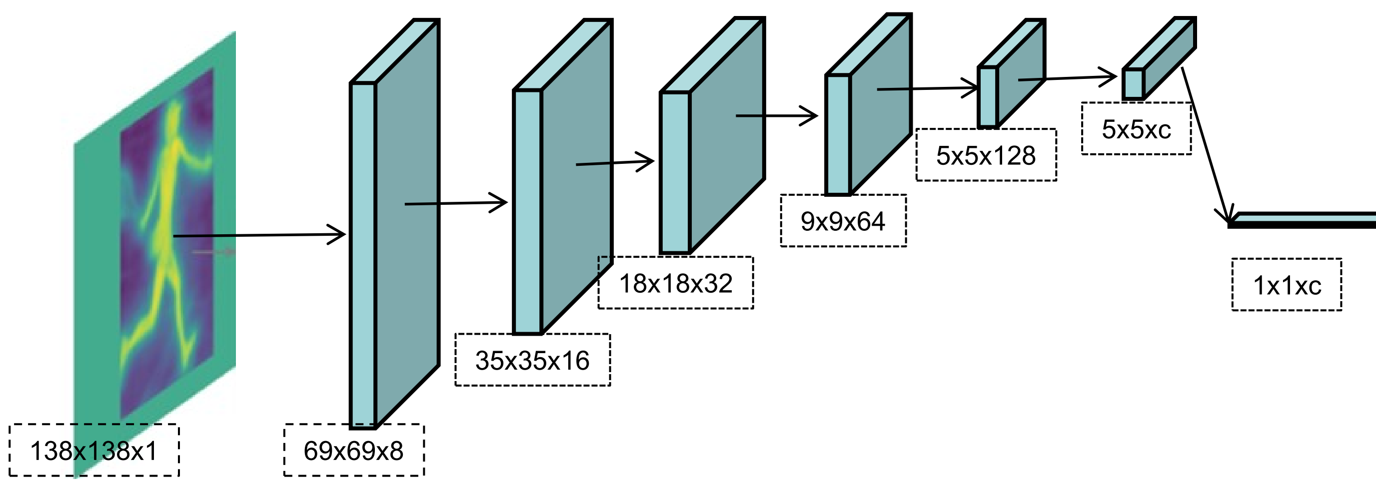
YOLACT++ (1)

- It's improvement of YOLACT model
- The problem of inconsistency between predicted mask and mask koefficient is solved by special branch of mask re-scoring
- Convolutions 3x3 of convolutional layers 3-5 are changed on deformable convolutions 3x3
- There is optimization for anchor choosing for every level of FPN

YOLACT++ (2)

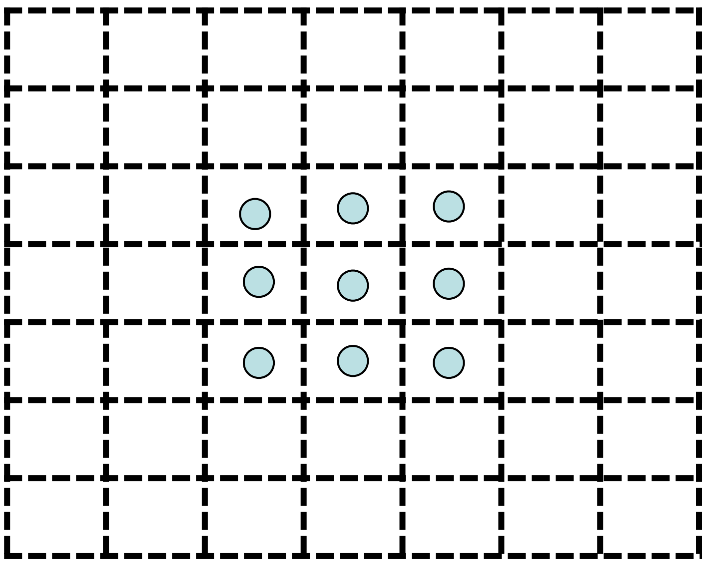
- Branch of mask re-scoring containes:
  - 6 convolutional layers with ReLU-activation function
  - 1 layer of global pooling
  - The input is predicted mask of image size (with zero-values out of the anchor-box)
- FCN based model

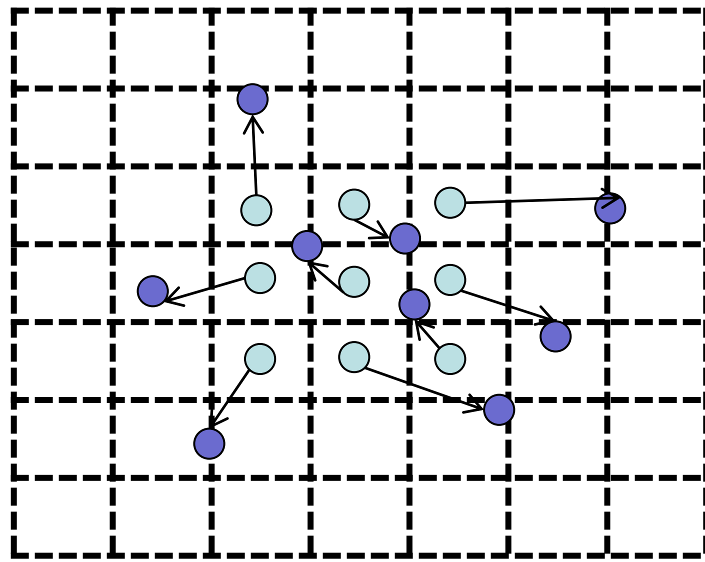
Bolya D. et al. Yolact++: Better real-time instance segmentation. – 2019. – [https://arxiv.org/pdf/1904.02689.pdf]



YOLACT++ (3)

- Deformable convolutions example
- Deformations depends from features of previous layer



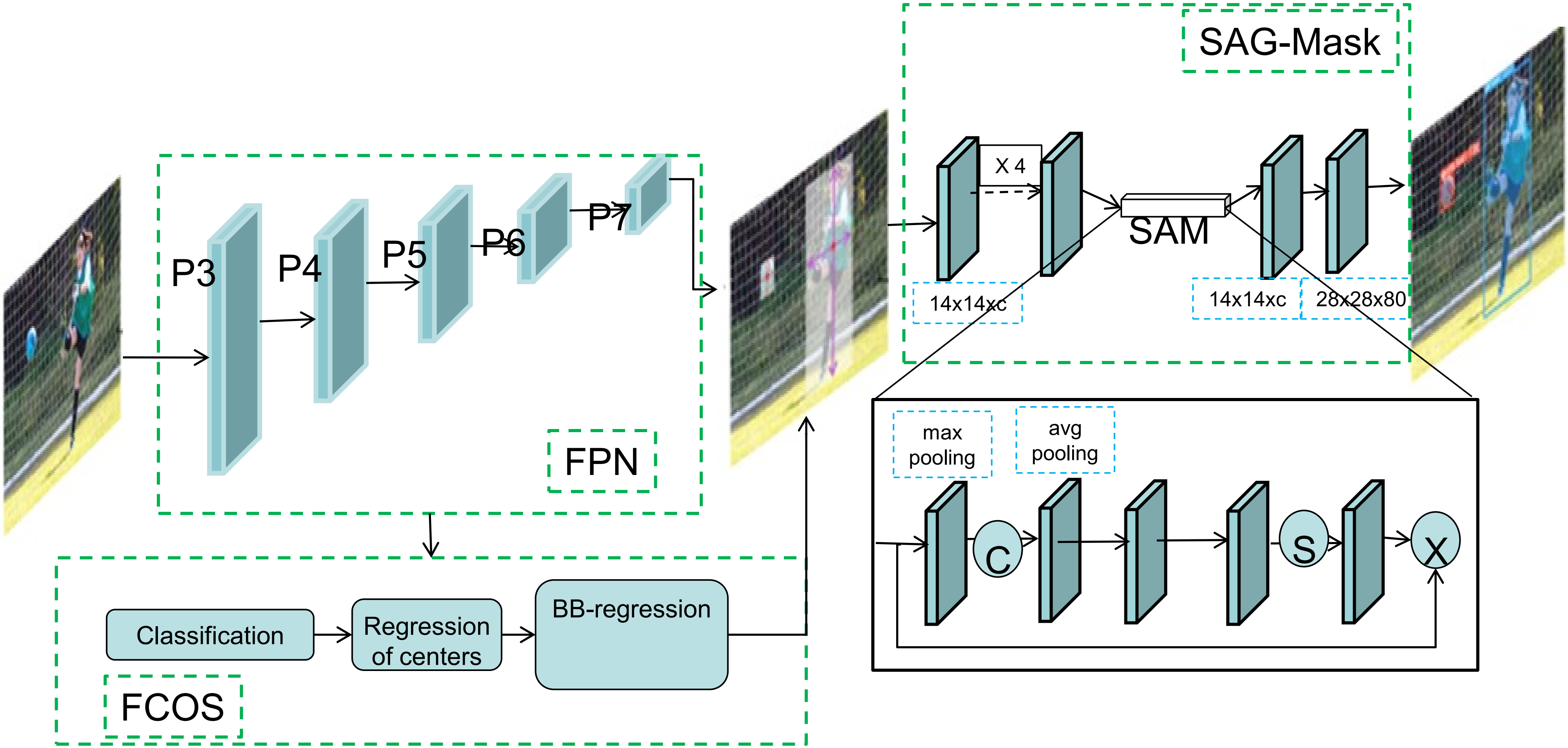


Bolya D. et al. Yolact++: Better real-time instance segmentation.– 2019. [https://arxiv.org/pdf/1904.02689.pdf]

CenterMask (1)

- Simple, effective, one-stage network for object segmentation
- There are three parts:
  - Feature map extraction is realized using ***FPN model***
    - The base model version backbone is ResNet-101
    - The lite-model version backbone is VoVNetV2
  - **Object detection** (without candidates generation)is realized using FCOS model
  - **Object segmentation** is realized using SAG-Mask

CenterMask (2)

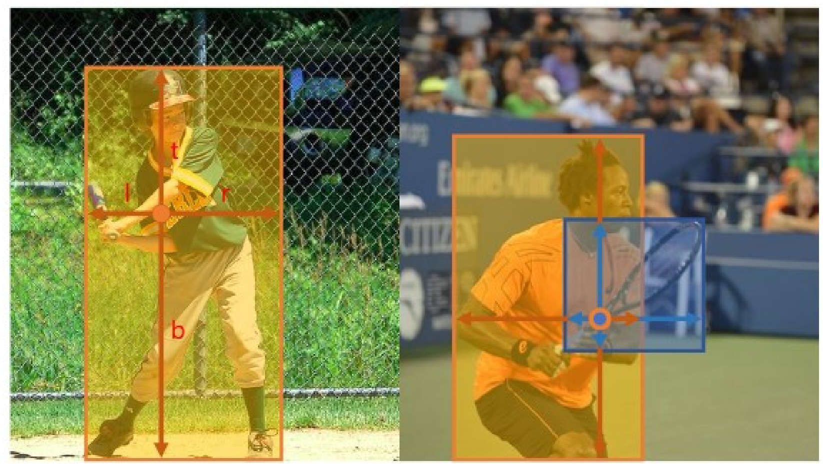


Lee Y., Park J. CenterMask: Real-time anchor-free instance segmentation. – 2020. – [https://arxiv.org/pdf/1911.06667.pdf]

CenterMask (3)

- ***FCOS*** (Fully convolutional one-stage object detector) directly,without candidates choosing predicts the following values:
  - object center
  - 4 offsets from center
  - class reliability

Tian Z. et al. Fcos: Fully convolutional one-stage object detection. – 2019. – [https://arxiv.org/pdf/1904.01355.pdf]



CentrMask (4)

- Feature map extraction is realized according Mask R-CNN description
- Multi-level feature pyramid is presented
- Choose the pyramid level for feature extruction is realized according formula

$$k = \lceil k_{\max} - \log_2 A_{input} / A_{RoI} \rceil$$

- Ainput, ARoI - areas of segmented and labeled region
- Feature map size is 14x14
- RoIAlign is applied to calculate elements of feature map

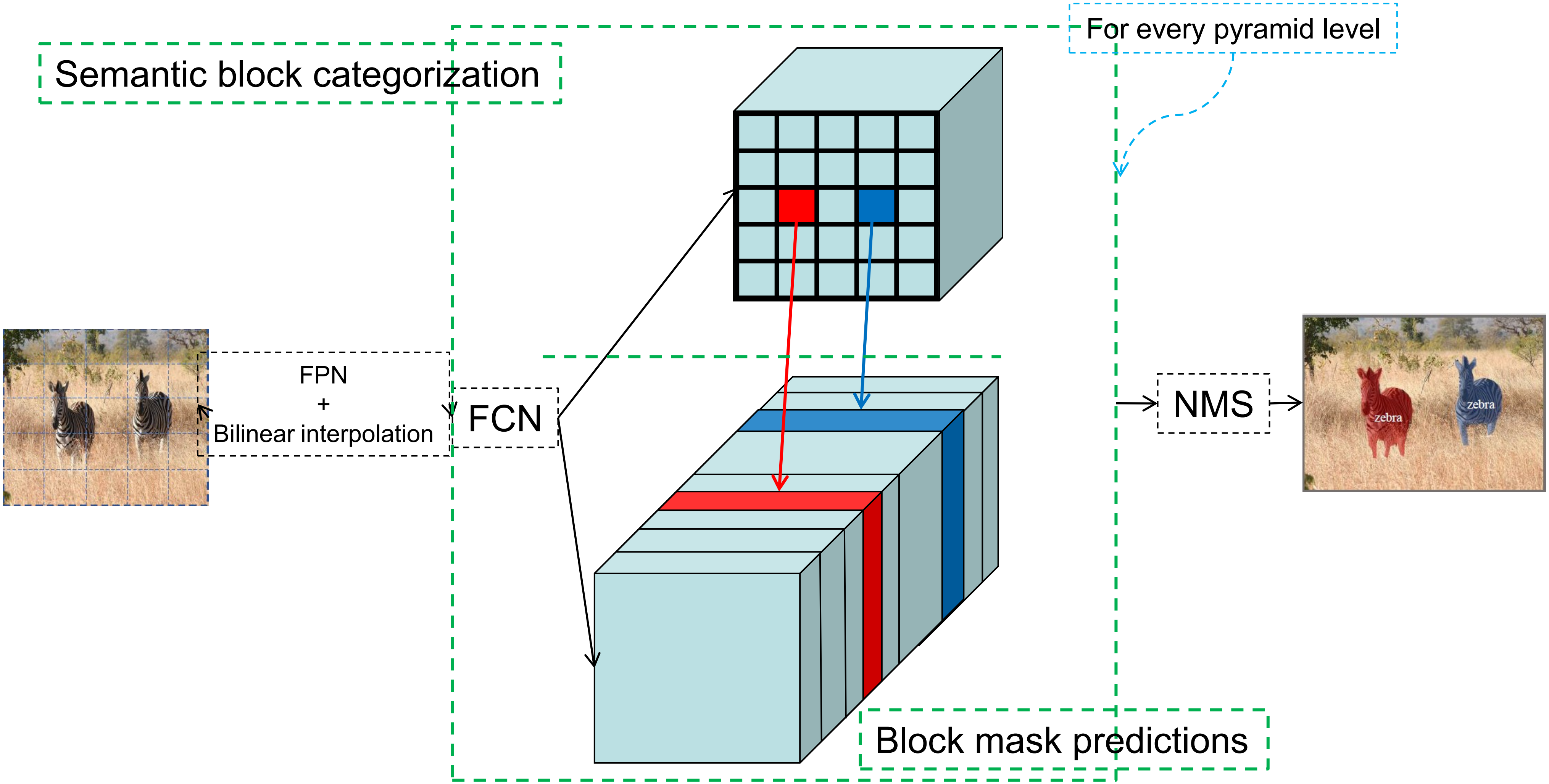
CenterMask (5)

- ***SAG-Mask (Spatial Attention-Guided Mask)***
- The goal is highlight informative features and hide non-informative features
  - Region feature map is the input of 4 convolution layer sequence. The result is feature map A
  - Max-pooling and average pooling are applied, the results are concatenated
  - 3x3 convolution layer with sigmoid activate is presented. Feature map B is the result
  - A and B are concatenated by element-wise multiplication
  - Deconvolution 2x2 is applied, size of feature map is 28x28
  - Convolution layer 1x1 is applied to generate segmentation mask for every class

SOLO (1)

- ***SOLO (Segment objects by locations)***
- In contrast to semantic segmentation, it is proposed to distinguish between instances of objects in the image by introducing the concept of "category of instances“
- Input image is divided on SxS blocks
- To generate feature map with fixed channel count (256) on every level FPN model is used

SOLO (2)



Wang X. et al. Solo: Segmenting objects by locations.– 2019. – [https://arxiv.org/pdf/1912.04488.pdf]

SOLO (3)

- ***Semantic categorization of block***:
  - object class predicted for every block
  - C-channel vector demonstrated reliability for current class
- ***Block structure***:
  - 7 convolution layer (SxSx256 is output)
  - 1 convolution layer (SxSxC is output)
- The result of classification for (i,j) block is computed in an obvious way. Index k of mask mk that is correspond with current block is:
  - $k = i*S + j$

SOLO (4)

- ***Binary segmentation of the objects***:
  - The input is feature map that is concatenated with normalized ([-1,1]) space coordinates of the image
  - SxS object masks are generated
- ***Structure***:
  - 7 convolution layer (H x W x 256 is output)
  - 1 convolution layer (H x W x S^2 is output)
  - Bilinear interpolation for the inference mode
- Non-maximum suppression is used

COMPARISON OF DEEP MODELS FOR INSTANCE SEGMENTATION

Comparison of deep models for instance segmentation (1)

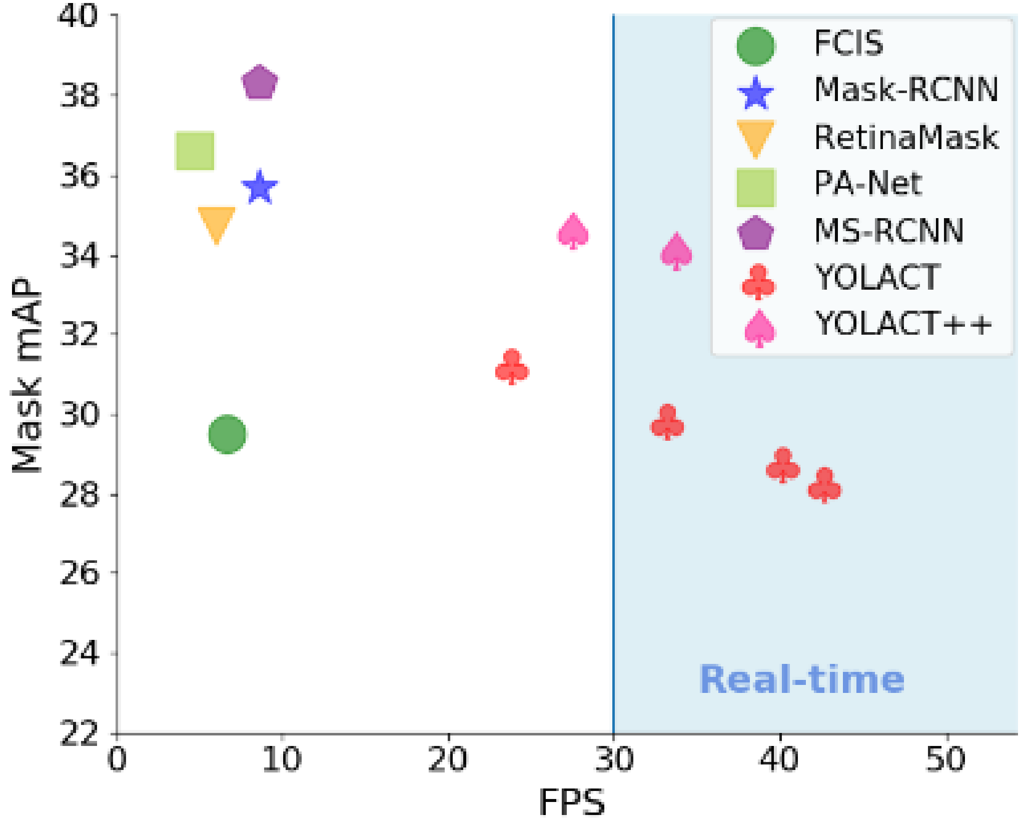
- MS COCO dataset

Model	AP	Year
PANet	**42.0%**	**2018**
SOLOv2	41.7%	**2020**
SOLO	40.4%	2019
Mask Scoring R-CNN	39.6%	2019
CenterMask	38.3%	2020
Mask R-CNN	37.1%	2017
YOLACT	29.8%	2019

Hafiz A. M., Bhat G. M. A survey on instance segmentation: state of the art .– 2020. – [https://link.springer.com/article/10.1007/s13735-020-00195-x]

Comparison of deep models for instance segmentation (2)

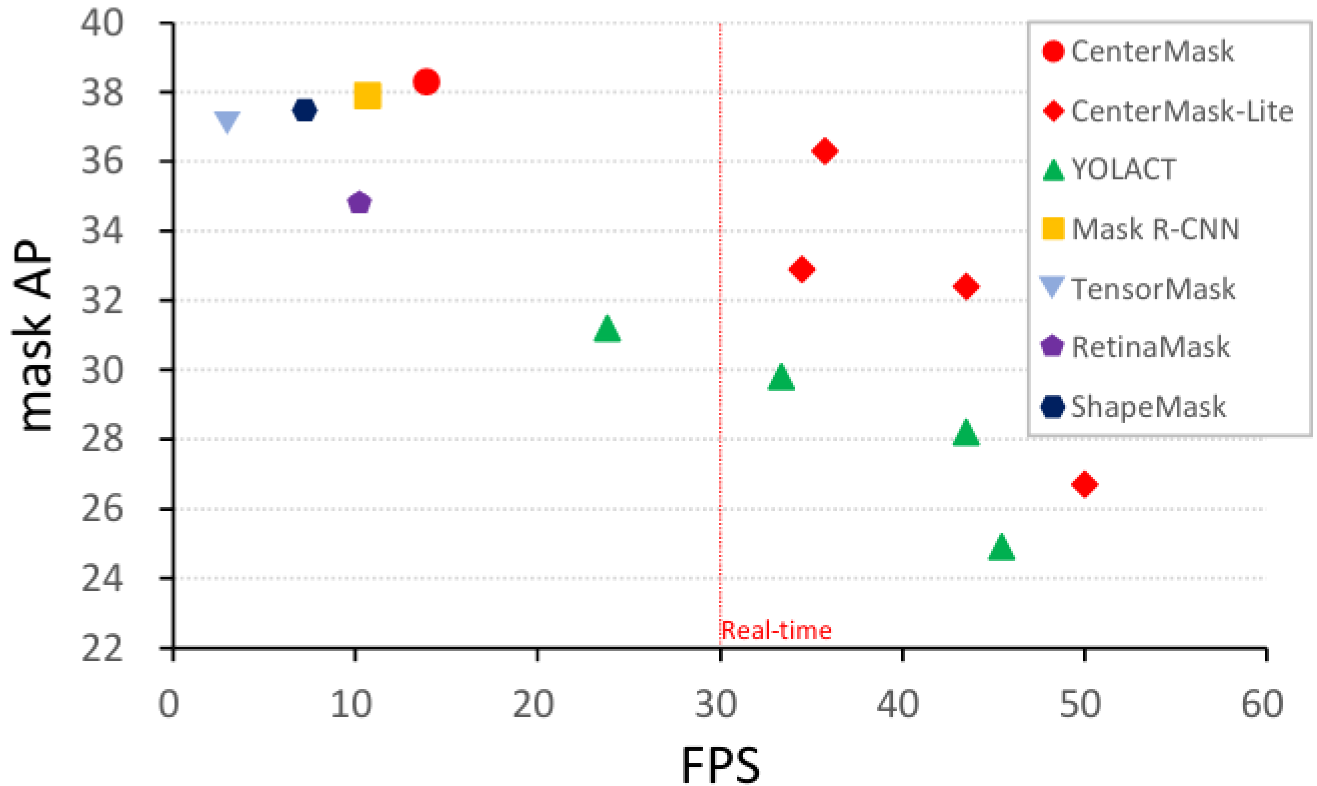
- ***An effective model is a compromise between quality and performance***



Bolya D. et al. Yolact++: Better real-time instance segmentation. – 2019. [https://arxiv.org/pdf/1904.02689.pdf]

Comparison of deep models for instance segmentation (3)

- ***An effective model is a compromise between quality and performance***



Lee Y., Park J. CenterMask: Real-time anchor-free instance segmentation. – 2020. – [https://arxiv.org/pdf/2004.04446.pdf]

Conclusion

- Models for instance segmentation are not limited to those discussed in the lecture
- The main problem constructing instance-segmentation models is to combine results of object detection part and semantic segmentation part
- The considered models solve this problem in different ways. As a rule, the decision greatly affects the performance
- The optimal model is a compromise between quality and complexity
  - Quality is determined by the requirements for solving a practical problem
  - Complexity is determined by the available computational resources and inference time requirements

Literature

- Pinheiro P. O., Collobert R., Dollár P. Learning to segment object candidates. – 2015.[https://arxiv.org/pdf/1506.06204.pdf]
- Dai J. et al. Instance-sensitive fully convolutional networks. – 2016. – [https://arxiv.org/pdf/1603.08678.pdf]
- Dai J., He K., Sun J. Instance-aware semantic segmentation via multi-task network cascades. – 2016. –. [https://arxiv.org/pdf/1512.04412.pdf]
- He K. et al. Mask r-cnn. – 2017. – [https://openaccess.thecvf.com/content_ICCV_2017/papers/He_Mask_R-CNN_ICCV_2017_paper.pdf]
- Huang Z. et al. Mask scoring r-cnn. – 2019. – [https://arxiv.org/pdf/1903.00241.pdf]
- Liu S. et al. Path aggregation network for instance segmentation. – 2018. – [https://arxiv.org/pdf/1803.01534.pdf]

Literature

- Bolya D. et al. Yolact: Real-time instance segmentation. – 2019. – [https://arxiv.org/pdf/1904.02689.pdf]
- Bolya D. et al. Yolact++: Better real-time instance segmentation. – 2019. [https://arxiv.org/pdf/1904.02689.pdf]
- Lee Y., Park J. CenterMask: Real-time anchor-free instance segmentation. – 2020. – [https://arxiv.org/pdf/1911.06667.pdf]
- Wang X. et al. Solo: Segmenting objects by locations //arXiv preprint arXiv:1912.04488. – 2020. – [https://arxiv.org/pdf/1912.04488.pdf
- Hafiz A. M., Bhat G. M. A survey on instance segmentation: state of the art .– 2020. – [https://link.springer.com/article/10.1007/s13735-020-00195-x]
- Tian Z. et al. Fcos: Fully convolutional one-stage object detection. – 2019. – [https://arxiv.org/pdf/1904.01355.pdf]

Authors (1)

- **Turlapov Vadim Evgenievich,** Dr., Prof., department of computer software and supercomputer technologies
  vadim.turlapov@itmm.unn.ru
- **Vasiliev Engeny Pavlovich,** lecturer, department of computer software and supercomputer technologies
  evgeny.vasiliev@itmm.unn.ru
- **Getmanskaya Alexandra Alexandrovna,** lecturer, department of computer software and supercomputer technologies
  alexandra.getmanskaya@itmm.unn.ru
- **Kustikova Valentina Dmitrievna**
  Phd, assistant professor, department of computer software and supercomputer technologies
  valentina.kustikova@itmm.unn.ru

Authors (2)

- **Zolotykh Nikolai Yurievich,** Dr., Prof., department of algebra, geometry and discrete mathematics
  nikolai.zolotykh@gmail.com
- **Nosova Svetlana Alexandrovna,** lecturer, department of computer software and supercomputer technologies
  nosova.sv.a@gmail.com
- **Tuzhilkina Anastasiya Andreevna,** master student, department of computer software and supercomputer technologies
  tan98-52@yandex.ru