KNOWLEDGE DISTILLATION USING GANS FOR FAST OBJECT DETECTION

E. Finogeev[1,2], V.Gorbatsevich[1]*, A.Moiseenko[1], Yu. Vizilter[1], O. Vygolov[1]

[1] FEDERAL STATE UNITARY ENTERPRISE «STATE RESEARCH INSTITUTE OF AVIATION SYSTEMS», Russian Federation (gvs, moiseenkoas, viz)@gosniias.ru
[2] National Research Nuclear University "MEPhI", Russian Federation (fel001)@campus.mephi.ru

Commission II, WG II/5

KEY WORDS: CNN, Object detection, Knowledge distillation, Real-time, GAN, Single shot detector

ABSTRACT:

In this paper, we propose a new method for knowledge distilling based on generative adversarial networks. Discriminator CNNs is used as an adaptive knowledge distilling loss. In experiments, single shot multibox detector SSD based on MobileNet v2 and ShuffleNet v1 are used as student networks. Our tests showed AP and mAP improvement of more than 3% on PascalVOC and 1% on MS Coco datasets compared with the baseline algorithm without any architecture or dataset changes. The proposed approach is general and can be used not only with SSD but also with any type of object detection algorithms.

1. INTRODUCTION

Nowadays, there are many practical computer vision tasks that can be solved with the previously unattainable quality by using convolutional neural networks (CNN). However, the well-known drawback of CNNs is high computational cost that makes them quite difficult to implement on embedded systems, especially for real-time image sequence analysis. This is the case even despite of the latest advance in embedded hardware capabilities for neural network processing (e.g. Google TPU or NVIDIA Xavier).

From algorithmic side, special "mobile" CNN architectures have been developed (e.g. MobileNet(Howard 2017), ShuffleNet(Zhang 2017), which are very computational efficient in inference (in terms of floating point operations) compared with the regular CNNs. Though, their practical use is restrained by a difficult hyperparameters fine tuning and the further performance improvement for such "mobile" architectures is still an acute task.

Knowledge distillation can be one of the possible solutions in this area since it is a method for knowledge transfer from one neural network to another, usually, from deep and slow to small and fast one. This approach is becoming increasingly popular in the practical application of artificial neural networks.

In this work, we consider the problem of fast object detection as a test task. In contrast to classical distilling technique , we use generative adversarial networks (GAN) as an adaptive loss function for deep feature mimic. As basic object detection algorithm we use single shot multibox detector SSD. The proposed approach allows us to get mAP gain on COCO and Pascal VOC Datasets without any architecture or dataset changes.

2. RELATED WORKS

Object detection. Currently, there are two basic concepts that implemented in object detection algorithms: region proposal object detection and single shot object detection.

Historically, the region proposal detectors have appeared first and implemented the idea to split a detection problem on two stages: to create hypotheses about the possible location of objects on the image without their classification and then on the second stage to verify hypotheses and refine objects location. Examples of such algorithms are R-CNN(Girshick, 2014), Fast RCNN(Girshick, 2015), Faster RCNN(Ren, 2015), R-FCN (Dai, 2016), FPN (Lin, 2017). In RCNN, which can be considered as the basic work for this class of algorithms, the first part that generates hypotheses is based on selective search procedure, whereas on the second stage a neural network is used for classification. In Faster RCNN, that is the further development of RCNN, two stages are combined in one network architecture that delivers significant increase in processing speed. R-FCN improves speed and accuracy by removing fully connected layers for final detection. In the case of R-FPN, a pyramidal architecture with lateral connections was developed for building multi-scale high-level feature maps, in which object detection is performed independently at each level. In general, the region proposal detectors provide high flexibility and accuracy by dividing a processing flow on two stages, but at the same time, it is still extremely difficult to implement this concept in real time.

Single short detectors solve object detection problem using one processing stage based on one neural network. Such neural network receives images as an input and outputs bounding boxes relative to positions of detected objects along with their class labels. This group of detectors are represented by YOLO(Redmon, 2016), SSD(Liu, 2016), DSOD(Shen, 2017), RetinaNet(Lin, 2017).One of the first single shot algorithms is YOLO, which is based on the original CNN architecture and provides processing speed of 244 FPS in TinyYolo modification. The further development of single short detectors is SSD. In contrast to YOLO, the SSD architecture uses deep features from various layers of the neural network, depending on the size of the object. In addition, to improve the quality of object detection of various shapes, anchor boxes similar to those proposed in the R-CNN algorithm are used. Currently, there are a lot of various SSD detectors (e.g. Yolo v2, RetinaNet, DSOD, DSSD), which have modified network architecture and loss function, but employing the same ideology.

Single short detectors are currently used in strict real-time applications such as real-time face detectors SSH, S3FD and FaceBoxes.

* Corresponding author

Knowledge distillation. Knowledge distillation task involves the transfer of knowledge from a "teacher" network to a "student" network in order to improve the quality of the latter. One of the first works in this area was (Romero, 2014), in which the transfer of knowledge in relation to the classification problem was considered. The proposed method minimizes L2 difference in deep features of teacher and student networks. This method is widely used in practice because of its simple implementation. For example, in (Quanquan, 2017) the method was used to transfer the knowledge for region proposal two-stages Faster R-CNN detector.Another approach is to "implicitly" learn deep attributes (Hinton, 2015). In (Hinton, 2015) the so-called "soft labels" were proposed. These labels are created from the teacher answers and are further used in the student loss function. In this case, direct minimization of differences in deep features does not occur. (Chen, 2017) employs relationship between different samples to improve quality. In (Huang, 2017) knowledge distilling problem was transformed to distribution matching problem. In (Wang, 2018) an algorithm for transferring knowledge by using generative adversarial networks was proposed. The discriminator is used as a loss function to minimize differences in deep features. In the original article, knowledge transfer problem was considered relatively to the classification task. In our work, we propose a similar approach, but adapted for the task of object detection.

3. GAN FOR KNOWLEDGE DISTILLING

Classical GANs generate some signal or image from random vector. Conditional GANs transforms some input data and maybe random vector to output image or vector. Typically, GAN presumes two neural networks: G – generator and D – discriminator. Generator network G generates some signal in target domain from input data. Discriminator network D is trained to distinguish "real" signals from the target domain from the "fakes" produced by Generator. Generator and Discriminator are trained simultaneously. Discriminator provides the adversarial loss that enforces Generator to produce "fakes" that cannot be distinguished from "real" signal. Condition generative adversarial networks are widely used for domain-to-domain translation.

Generator and Discriminator are trained simultaneously using the following loss:

$$L = L_{GAN}(\mathrm{G},\mathrm{D}) + \lambda L_{REC}(\mathrm{G}) \qquad (1)$$

where

L_{GAN} – "adversarial" loss,
L_{REC} – reconstruction loss.

Binary cross entropy loss is widely used as an adversarial loss:

$$L_{GAN} = \log(D(x,y)) + \log(1 - D(x,G(x)) \qquad (2)$$

where

y - real sample
x – input data,
D – Discriminator CNN,
G – Generator CNN.

L1 or L2 distances are usually used as a reconstruction loss function(L_{REC}). Reconstruction loss leads to better convergence and prevents constant output.

Knowledge distilling problem can be easily represented as a domain-to-domain transform. In that case, the source domain is the student feature space and the target domain is the teacher feature space (see figure 1).

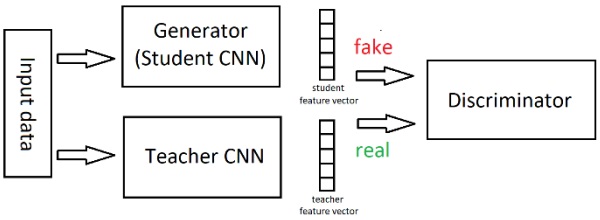

Figure 1. GAN for knowledge distilling.

Using Discriminator CNN instead of L1 or L2 distances provides adaptive loss function that leads to better accuracy.

In case of knowledge distilling reconstruction loss from (1) can be replaced by task specific loss (like object detection, classification or segmentation loss). A similar approach was used in (Wang, 2018) for the classification problem, which increased accuracy from 68.43 to 74.1 for MobileNet.

4. PROPOSED APPROACH

In our work, we consider SSD based object detection algorithms (SSD, DSOD, RetinaNet and others). In contrast to region proposal approaches, SSD detects objects using only one forward pass. These algorithms discretizing the output space of objects bounding boxes into a set of default boxes over different aspect ratios and scales per feature map location. During prediction, CNN generates scores separately for each default box and for each object class type. Instead of direct bounding box prediction, CNN generates adjustments to default boxes. In addition, the network combines predictions from multiple feature maps with different resolutions to naturally handle objects of various sizes.

SSD is one of the most popular and fast object detectors. For example, SSD with the base CNN MobileNet v2, which is discussed in our article, provides 5 FPS processing speed on a Google Pixel 1 phone with a Qualcomm Snapdragon 821 processor (Sandler 2018). SSD-MobileNet v2 reaches processing speed of more than 250 FPS on NVIDIA GTX 2080 Ti GPU with the usage of Tensor RT library.

Basic Single Shot MultiBox Detector (Liu, 2016) can be divided into two parts(see Figure 2) - basic network and extra layers.

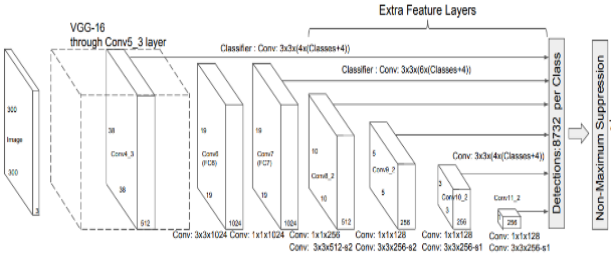

Figure 2. Single Shot MultiBox Detector.

In original paper, VGG-16 network is used as the basic network. By replacing VGG on mobile networks like

MobileNets(Howard ,2017) or ShuffleNets(Zhang ,2017), opens possibility to provide high computational speed on embedded platforms and for real-time solutions.

Extra layers are a series of progressively smaller convolutional layers (see Figure 2). Layers from "extra layers", along with some of the earlier base network layers, are used to predict scores and bounding boxes. We name these layers as "feature layers".These predictions are performed by 3x3 convolutions, one filter for each category score and one for each dimension of the bounding box that is regressed. At the end, a non-maximum suppression (NMS) is used for post-processing of the predictions to get final detection results.

Therefore, in SSD case, we have several feature layers for mimicking. The Student network plays the role of Generator that generates features, and we need unique discriminator for each feature map (see Figure 3).Using this approach, we can transfer knowledge between two SSDs with different basic networks and the same extra layers architecture.

The loss function according to (1) will be the following:

$$L = \sum_i L_{GANi}(G, D) + \lambda L_{MBOX}(G) \qquad (3)$$

where

L_{GANi} – "adversarial" loss for i-th discriminator,

L_{MBOX} – multibox loss.

In our work, we use the original multibox loss function from (Liu 2016). L_{GANi} is similar to (2).

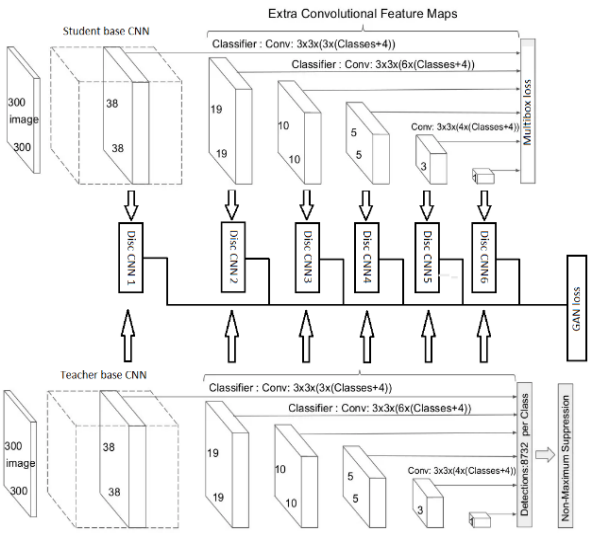

Figure 3. Proposed knowledge distilling framework.

The proposed approach is general and can be used for any type of object detection algorithms or any convolutional neural network. In this case, for each feature layer we need individual discriminator. For example, for simple classification we need only one discriminator.

5. EXPERIMENTAL RESULTS

Architecture.

Teacher Network. We used SSD based on DarkNet-53 as the teacher network due o the following reasons:

1. In comparison to well-known ResNet-152 CNN Darknet-53 provides similar performance on ImageNet dataset but twice faster

2. It provides much higher accuracy on test datasets with SSD: Pascal VOC0712 ~ 0.77 mAP than MobileNet or ShuffleNet based SSD;

3. It has feature layer shapes similar to MobileNet and ShuffleNet based SSD.

This neural network was proposed in (Redmon, 2018) for training the Yolo v3 object detector and has the architecture presented in figure 4.

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Figure 4. DarkNet53 CNN architecture.

Student Network. As we have mentioned before, in this work, we consider the class of algorithms for fast object detection. In our experiments we used SSD-Lite modification of SSD. MobileNet v2(Sandler 2018) and ShuffleNet v1 was used as the student networks. That CNNs are specially developed for fast and embedded applications. Due to the features architecture of the basic networks, original SSD was slightly modified following (Sandler 2018).

Discriminator Networks. In SSD case we have 6 discriminator networks. We tried different architectures and discriminator types. The best results were obtained with medium size discriminator CNNs shown in Tables 2 and 3. Deeper discriminator architectures can lead to overfitting and worse results.

Our discriminators are built from blocks that contain convolutional layers, instance normalization layers (Ulyanov, 2016) and leaky relu activation functions. CNN architecture for

the first 5 discriminators(DNet1-DNet5) is based on patch GAN ideology (Isola, 2017). This approach implies that instead of one answer, the output of the discriminator is a semantic map for the two classes - "real" and "fake". These types of discriminators are often used in image processing and can improve the quality of the generator compared to the classical ones. In our case, SSD feature maps also are spatial-aware except the last feature layer so we use path-gan architecture for DNet1-DNet5 with the following output sizes: 3x3 for DNet1-DNet4 and 2x2 for DNet5.

The additional tests were performed to study the influence of the effect of mimicry on the quality of object detection for various scales. Objects were divided into three groups: small, medium and large, in accordance with the rules for mapping anchors of the original SSD.

Object size	*AP gain % to baseline*
Small(D1+D2)	*+8%*
Medium(D3+D4)	*+14%*
Large (D5+D6)	*+12%*

Table 1. AP gain for different object sizes.

As can be seen from the table above(see Table 1), the use of mimicry improves the quality of detection for all considered objects sizes.

DNet1	*DNet2*	*DNet3*
Size = 19x19 Conv (512-> 512, k=3, s=1, p=1, bias=True) InstanseNorm LeakyRely	**Size = 10x10 Conv (1024-> 1024, k=3, s=1, p=1, bias=True) InstanseNorm LeakyRely**	**Size = 5x5 Conv (512-> 512, k=3, s=1, p=1, bias=True) InstanseNorm LeakyRely**
Size = 19x19 Conv (512-> 512, k=3, s=2, p=1, bias=True) InstanseNorm LeakyRely	**Size = 10x10 Conv (1024-> 1024, k=3, s=2, p=1, bias=True) InstanseNorm LeakyRely**	**Size = 5x5 Conv (512-> 512, k=3, s=1, p=0, bias=True) InstanseNorm LeakyRely**
Size = 10x10 Conv (512-> 512, k=3, s=2, p=1, bias=True) InstanseNorm LeakyRely	**Size = 5x5 Conv (1024-> 1024, k=3, s=1, p=0, bias=True) InstanseNorm LeakyRely**	**Size = 3x3 Conv (512-> 1, k=3, s=1, p=0, bias=False)**
Size = 5x5 Conv (512-> 512, k=3, s=1, p=0, bias=True) InstanseNorm LeakyRely	**Size = 3x3 Conv (1024-> 1, k=3, s=1, p=0, bias=False)**	
Размер = 3x3 Conv (512-> 1, k=3, s=1, p=0, bias=False)		
Sigmoid		

Table 2. Discriminator CNN architecture.

DNet4	*DNet5*	*DNet6*
Size = 3x3 Conv (256-> 256, k=3, s=1, p=1, bias=True) InstanseNorm LeakyRely	**Size = 2x2 Conv (256-> 256, k=2, s=2, p=1, bias=True) InstanseNorm LeakyRely**	**Size = 1x1 Conv (256-> 256, k=1, s=1, p=0, bias=True) LeakyRely**
Size = 3x3 Conv (256-> 256, k=3, s=1, p=1, bias=True) InstanseNorm LeakyRely	**Size = 2x2 Conv (256-> 256, k=2, s=2, p=1, bias=True) InstanseNorm LeakyRely**	**Size = 1x1 Conv (256-> 256, k=1, s=1, p=0, bias=True) LeakyRely**
Size = 3x3 Conv (256-> 1, k=3, s=1, p=0, bias=False)	**Size = 2x2 Conv (256-> 1, k=2, s=1, p=0, bias=False)**	**Size = 1x1 Conv (256-> 1, k=1, s=1, p=0, bias=False)**
Sigmoid		

Table 3. Discriminator CNN architecture.

Training.
Training process was divided in four stages:

1. Discriminator pre-learning;
2. Student-Net learning;
3. Discriminator fine tuning;
4. Student-Net post-learning.

Discriminator pre-learning – Discriminator CNN pre-learning stage. On this stage we used only 75 iterations to pretrain discriminator and only adversarial loss (2) (Student network was frozen).

Student-Net learning – student CNN training stage (Generator training). On this stage, the discriminator network was frozen (weights were not changing), only the student network was learnt using loss (3). According to GAN ideology, a "real" label is passed to the Discriminator as an answer. This stage was running until the mean adversarial loss per epoch became less than a given threshold.

Discriminator fine tuning – discriminator CNN training stage. On this stage only adversarial loss (2) were used (the student network was frozen) and number of iteration were also equal to 75.

Student-Net post-learning – on this stage the classical SSD training method (Liu, 2016) and SGD solver was used according to the original paper (for other stages we used Adam optimizer with fixed learning rate).

After pre-learning stage we applied the Student network and Discriminator learning stages during 180 epochs. Then we applied 80 epochs of post learning stage.

Following the original paper, we used backbones pretrained on the ImageNet dataset and the same augmentation set.

Datasets. For our experiments, we used Pascal VOC (Everingham, 2014) and MS Coco ((Lin, 2014)) datasets to balance between large scaled and small sized objects.

PASCAL VOC 0712 dataset contains 21493 images of annotated predominantly large objects from 20 classes. For testing we used metrics and test subset from PASCAL VOC benchmark described in (Everingham, 2014). For training we used the united dataset from PASCAL VOC and MS COCO training subsets.

MS COCO dataset contains 328 k images of 91 object types (more than 2.5 million labeled predominantly small sized objects). For testing we used COCO 17 val. subset and metrics according to the object detection protocol described in original paper (Lin, 2014). It is necessary to mention that mAP values for MSCOCO and PASCAL VOC datasets differ significantly.

Implementation details. Our code was developed in PyTorch framework. For all stages, except post-training, we used Adam optimizer with fixed 5*e-4 learning rate. The adversarial threshold was equal to 0.0005. For post training stage, SGD with initial learning rate 5*e-4 with 10 times decrease every 20 epochs was used. On single Nvidia GeForce 2080Ti it took approximately 14 hours to train MobileNet v2 SSD on PascalVOC dataset and two days - on MS Coco dataset.

We trained MobileNet v2 and ShuffleNet v1 based SSD-lite object detectors by using SSD based on darknet 53 as the teacher. Test results are shown in Table 4.

Student network	Dataset	Teacher network	Test VOC07 mAP
SSD_Lite_MobileNetV2	VOC0712		69.3
SSD_Lite_MobileNetV2	VOC0712	SSD_DarkNet53	**74.0**
SSD_Lite_ShuffleNetV1	VOC0712		62.4
SSD_Lite_ShuffleNetV1	VOC0712	SSD_DarkNet53	**65.6**
Student network	Dataset	Teacher network	Test COCO17 val mAP IoU=0.5:0.95
SSD_Lite_MobileNetV2	COCO14		16.6
SSD_Lite_MobileNetV2	COCO14	SSD_DarkNet53	**18.1**
SSD_Lite_ShuffleNetV1	COCO14		13.1
SSD_Lite_ShuffleNetV1	COCO14	SSD_DarkNet53	**14.6**

Table 4. Results on Pascal VOC and MS Coco Datasets.

CONCLUSIONS

This paper introduces a new approach for knowledge distilling based on generative neural networks. Instead of classical approaches, which employ loss function based on distance, we propose to use adversarial loss function. In this case, knowledge distilling problem is transformed to the domain-to-domain translation. Generative adversarial networks successfully applied to such type of problems in many practical applications. In this case, the feature space of Teacher network is a target domain, an input image is a source domain, and Student network is a generator network.

We used SSD Single shot multibox detector as the basic algorithm and object detection as the test task. We targeted the problem of improving quality of fast SSD, which is based on MobileNet, through knowledge distilling from SSD, which is based on deep and slow network.
According to SSD architecture, we used 6 discriminators networks with original architecture based on patch GAN ideology.
Training process presumes 4 main stages:

1. Discriminator pre-learning;
2. Student-Net learning;
3. Discriminator fine tuning;
4. Student-Net post-learning.

Unlike the classical generative adversarial network, we train discriminator and generator sequentially. Generator CNN is trained with frozen Discriminators until the adversarial loss reaches some level and then Discriminator is fine tuned with the frozen generator network.

Our approach was tested on well-known PASCAL VOC and MS COCO datasets. SSD based on DarkNet-53 network was employed as the teacher network, and MobileNet v2 and ShuffleNet v1 – as two options for the student network. The proposed approach allows us to get about 3% mAP gain (depends on the selected basic CNN architecture) on Pascal VOC Dataset and approximately 1.5% mAP gain on COCO Dataset without any architecture or dataset changes. In addition, it is worth to mention that the approach is general and can be used not only with SSD but also with any type of object detection algorithms.

ACKNOWLEDGEMENTS

The reported study was funded by Russian Science Foundation (Project No. 16-11-00082).

REFERENCES

Howard A., Zhu M., Chen B., Kalenichenko D., Wang W., Weyand T., Andreetto M., Adam H,. 2017: Mobilenets: Efficient convolutional neural networks for mobile vision applications. CoRR, abs/1704.04861, 2017. 2

Liu W., Anguelov D., Erhan D., Christian S., Reed S., Fu C.-Y., and A. C. Berg.,2016: SSD: single shot multibox detector. In ECCV, 2016.

Sandler M., Howard A., Zhu M., Zhmoginov A., Chen L.,2018: Mobilenetv2: Inverted residuals and linear bottlenecks. In CVPR, 2018.

Zhang X., Zhou X., Lin M., Sun J.,2018: Shufflenet: An extremely efficient convolutional neural network for mobile devices. CoRR, abs/1707.01083, 2017

Howard A., Zhu M., Chen B., Kalenichenko D., Wang W., Weyand T., Andreetto M., Adam H., 2017: Mobilenets: Efficient convolutional neural networks for mobile vision applications CoRR, abs/1704.04861, 2017.

Lin T., Maire M., Belongie S., Hays J., Perona P., Ramanan D., Dollar P., and Zitnick C.,2014: Microsoft coco: Com- ´
mon objects in context. In ECCV, 2014.

Everingham M., Ali Eslami S.M., Gool L., Williams C., Winn J., Zisserman A.,2014: The pascal visual object classes challenge a retrospective. In IJCV, 2014.

Isola, P., Zhu, J., Zhou, T., Efros A., 2017: Image-to-Image Translation with Conditional Adversarial Networks. In CVPR, 2017.

Romero A., Ballas N., Kahou S., Chassang A., Gatta C., Bengio Y., 2014: Fitnets: Hints for thin deep nets. arXiv:1412.6550, 2014.

Hinton G,. Vinyals O,. Dean J,. 2015: Distilling the knowledge in a neural network. arXiv:1503.02531, 2015.

Chen Y., Wang., Zhang Z., 2017: Darkrank: Accelerating deep metric learning via cross sample similarities transfer. arXiv:1707.01220, 2017.

Huang Z., Wang N., 2017: Like what you like: Knowledge distill via neuron selectivity transfer. arXiv:1707.01219, 2017.

Li Q., Jin S., Yan J., 2017: Mimicking very efficient network for object detection. In CVPR, 2017.

Ulyanov D., Andrea V., Lempitsky V., 2016: Instance Normalization: The Missing Ingredient for Fast Stylization. arXiv: 1607.08022, 2016.

Girshick R., Donahue J., Darrell T. and Malik J., 2014: Rich feature hierarchies for accurate object detection and semantic segmentation. In CVPR, 2017.

Girshick, R., 2015: Fast R-CNN, In ICCV, 2015.

Dai J., Li Y., He K., Sun J., 2016: R-FCN: Object detection via region-based fully convolutional networks. In: NIPS, 2016.

Lin Y., Dollar P., Girshick R., He K., Hariharan B., Belongie S., 2017: Feature Pyramid Networks for Object Detection. In CVPR, 2017.

Uijlings J.R.R., van de Sande K.E.A., Gevers T., Smeulders A.W.M., 2012: Selective Search for Object Recognition, In IJCV, 2012.

Redmon J., Divvala S., Girshick R., Farhadi A., 2016: You Only Look Once: Unified, Real-Time Object Detection. In CVPR, 2016.

Shen Z., Liu Z., Li J., Jiang Y.-G., Chen Y., Xue X., 2017: DSOD: Learning Deeply Supervised Object Detectors from Scratch. In ICCV, 2017.

Lin T.-Y., Goyal P., Girshick R., He K., Dollar P., 2017: Focal loss for dense object detection. In ICCV, 2017

Wang X., Zhang R., Sun Y., Qi J., 2018: KDGAN: Knowledge Distillation with Generative Adversarial Networks. In NIPS, 2018.

Girshick R., Donahue J., Darrell T., Malik J.. 2014: Rich feature hierarchies for accurate object detection and semantic segmentation. In CVPR, 2014

Ren S., He K., Girshick R., Sun J., 2015: Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In CVPR, 2015.

Redmon J., Farhadi A., 2018: YOLOv3: An Incremental Improvement. arXiv: 1804.02767