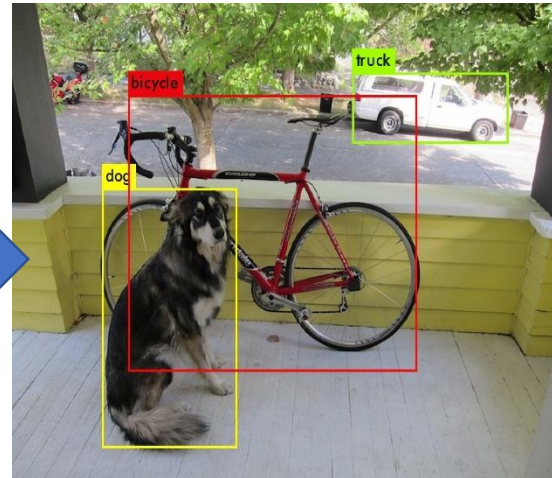
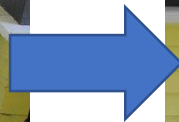
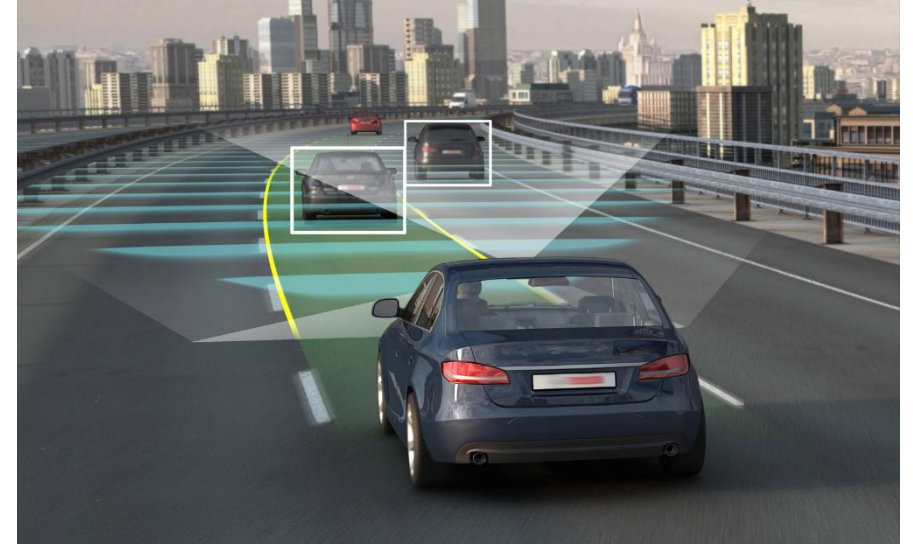


Алгоритм мимикрии с использованием
генеративных состязательных сетей для
задач обнаружения объектов

ФГУП ГОСНИИАС

**Визильтер Ю.В.
Горбацевич В.С.
Финогеев Е.Л.
Моисеенко А.С.**

Обнаружение объектов



Обнаружение

- выделение объемлющих прямоугольников
- классификация объектов

Обнаружение объектов – одна из наиболее практически востребованных задач в техническом зрении

Обнаружение объектов



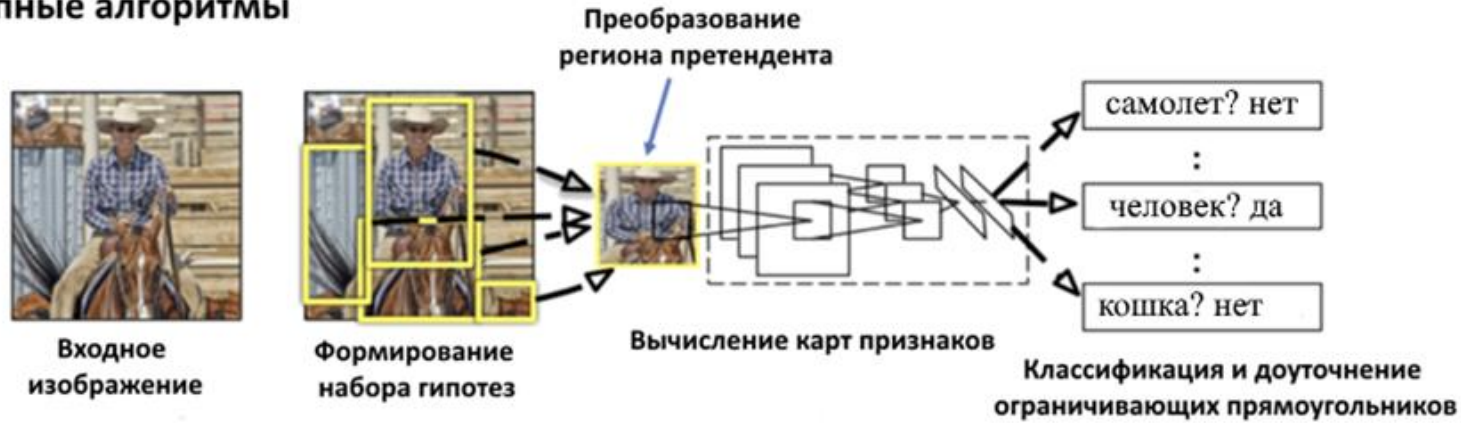
Особенности встраиваемых систем

- ✓ Низкая вычислительная мощность
- ✓ Работа в реальном времени
- ✓ Высокое качество работы



Подходы к обнаружению объектов

Двухэтапные алгоритмы



Алгоритмы: *Faster R-CNN, R-FPN, Light-Head R-CNN*

- ✓ Медленные
- ✓ Высокое качество

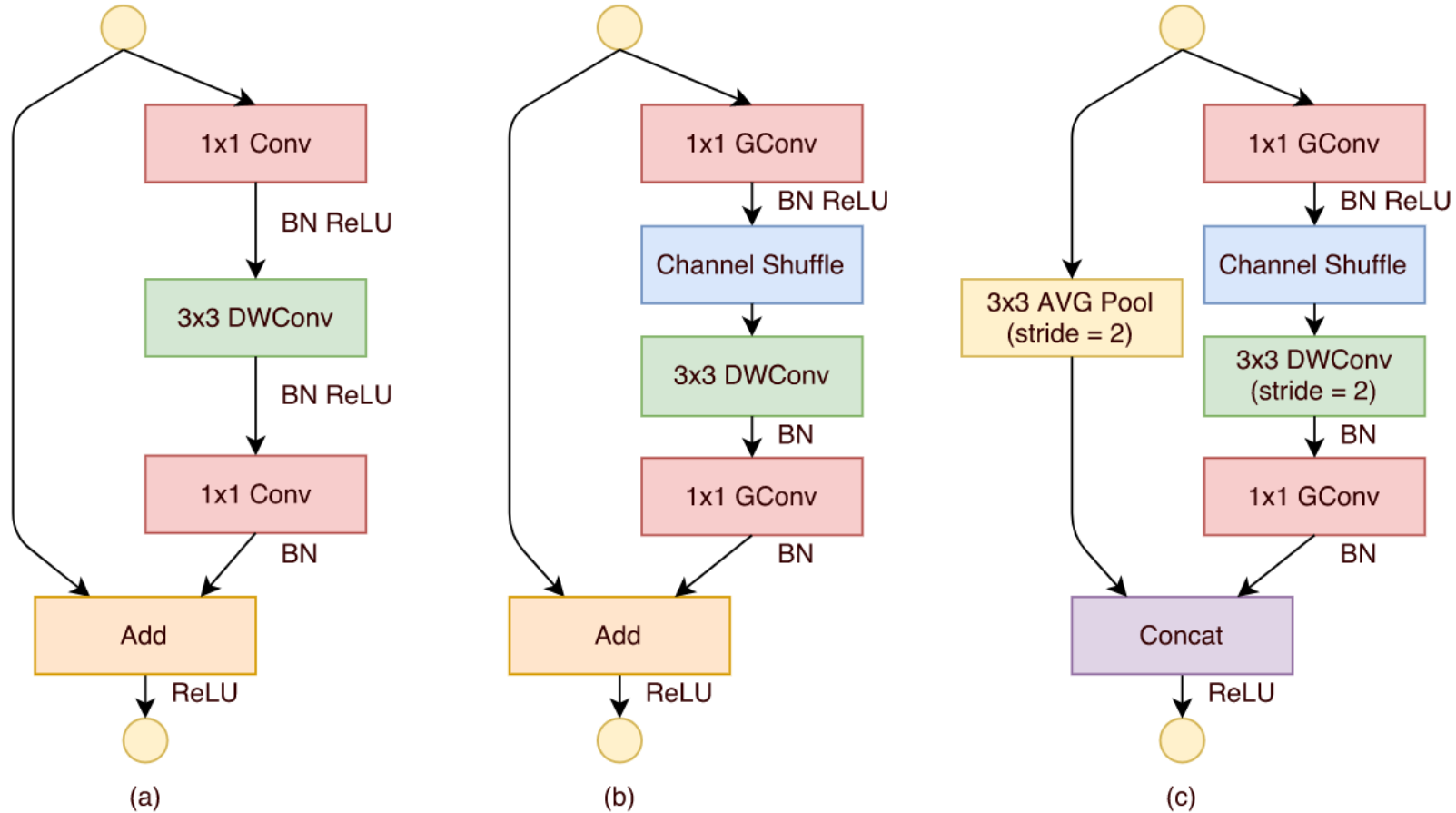
Одноэтапные алгоритмы



Алгоритмы: *SSD, DSOD, DetectNet, YOLO*

- ✓ Реальное время

Мобильные архитектуры ГКНС



Архитектуры:

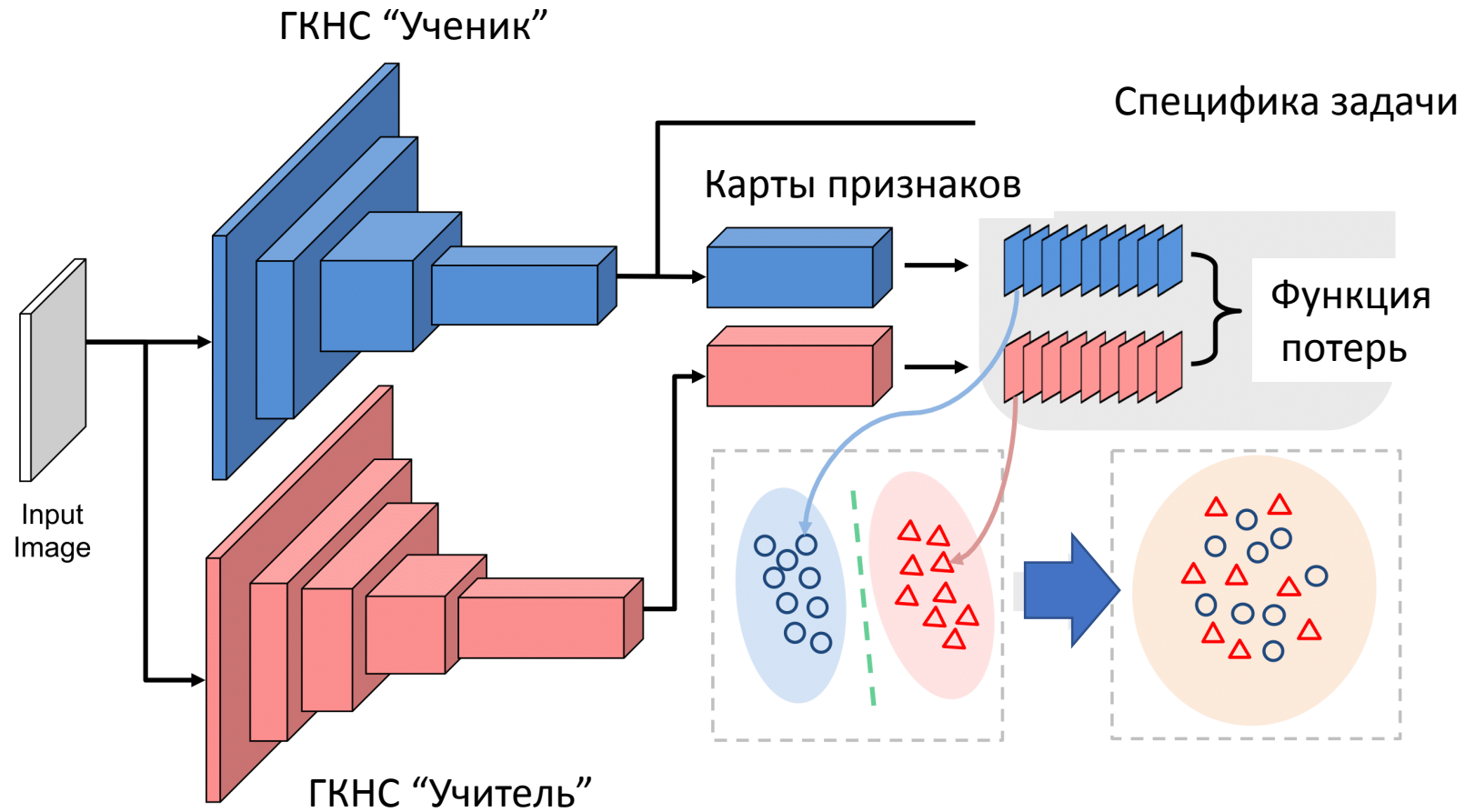
MobileNet v1,v2

ShuffleNet v1,v2

Особенности

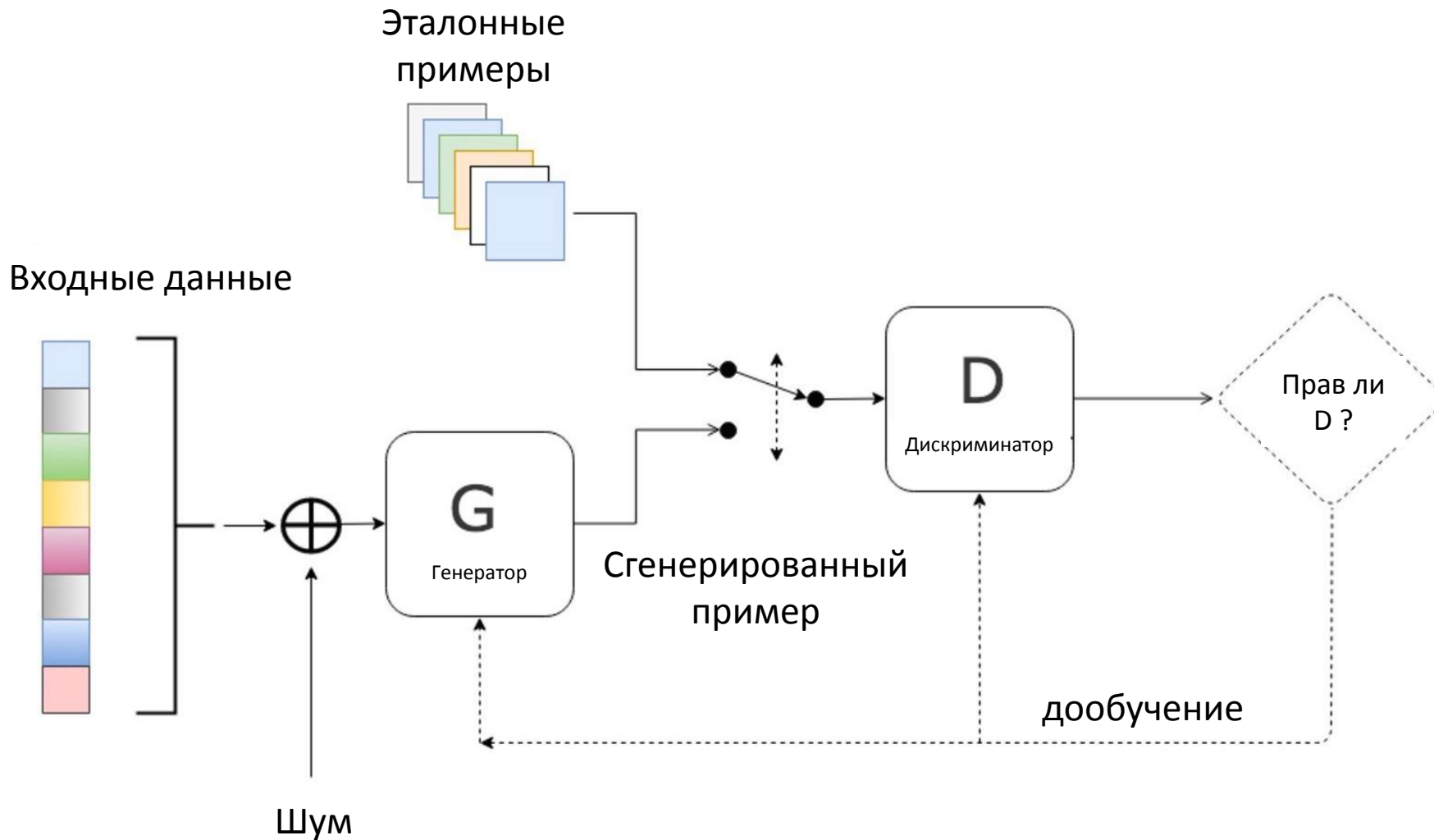
- ✓ Низкая вычислительная сложность
- ✓ Эффективные слои

Мимикрия – передача знаний



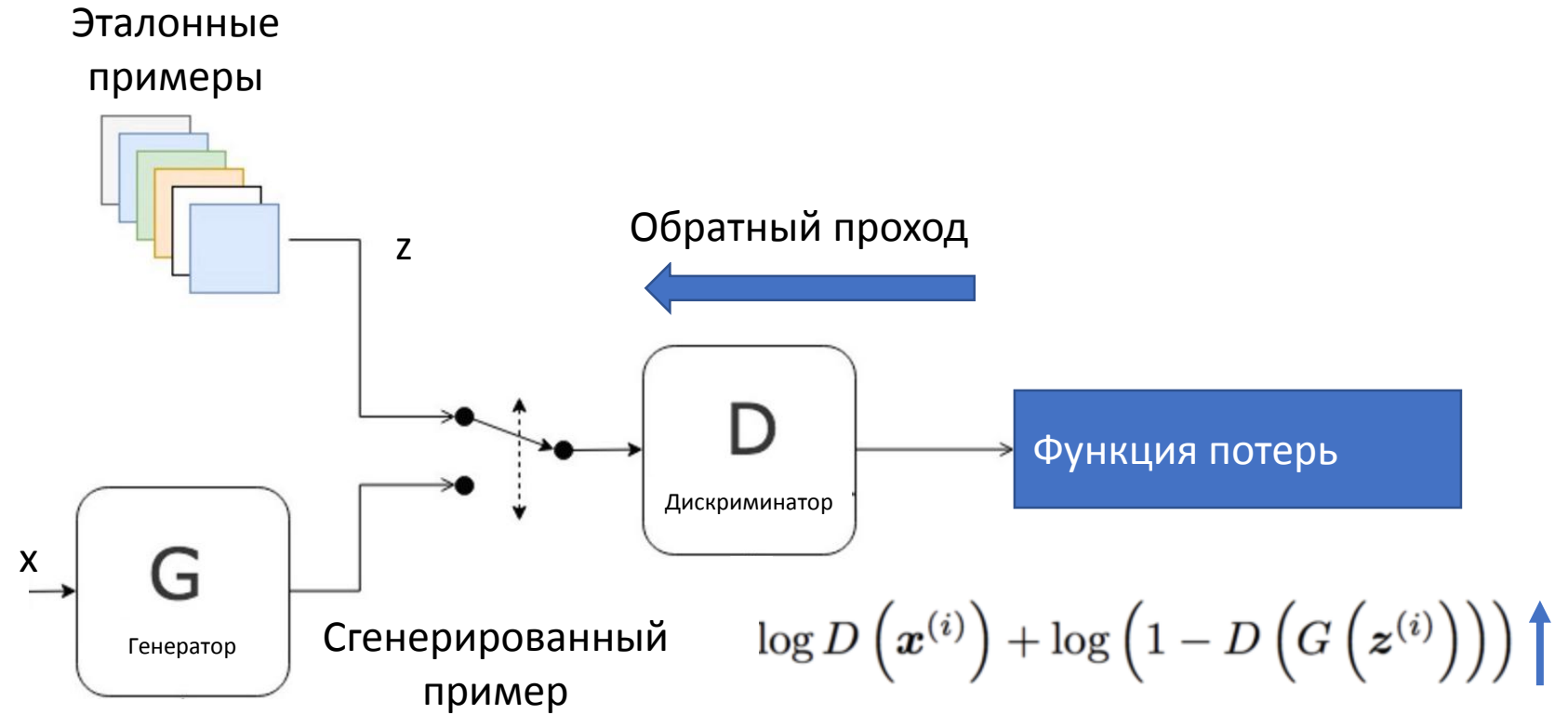
Мимикрия – это процесс, при котором менее нейронная сеть ученик обучается путем повторения глубоких признаков либо результатов сети учителя

GAN



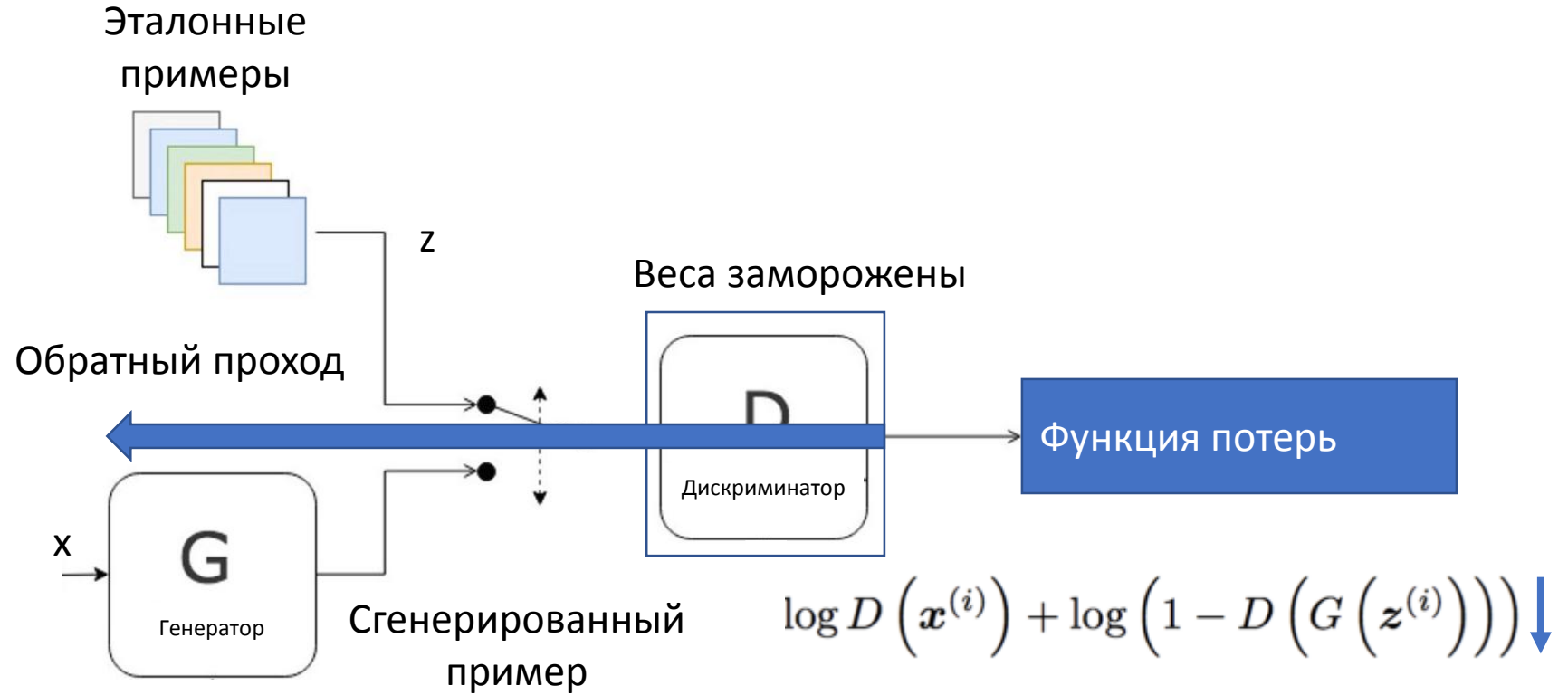
Идея состязательных сетей в одновременном обучении двух сетей – Дискриминатора и Генератора. При этом генератор должен сгенерировать данные неотличимые от некоторого набора эталонных данных, а дискриминатор должен отличать сгенерированные данные от эталонных

GAN



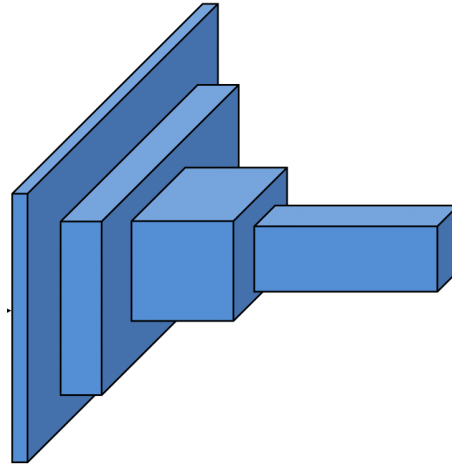
Этап 1. Обучение дискриминатора

GAN



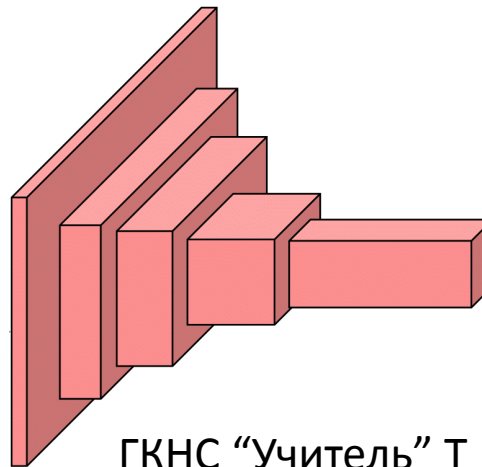
Этап 2. Обучение генератора

GAN



Сеть –ученик выступает в роли “Генератора”

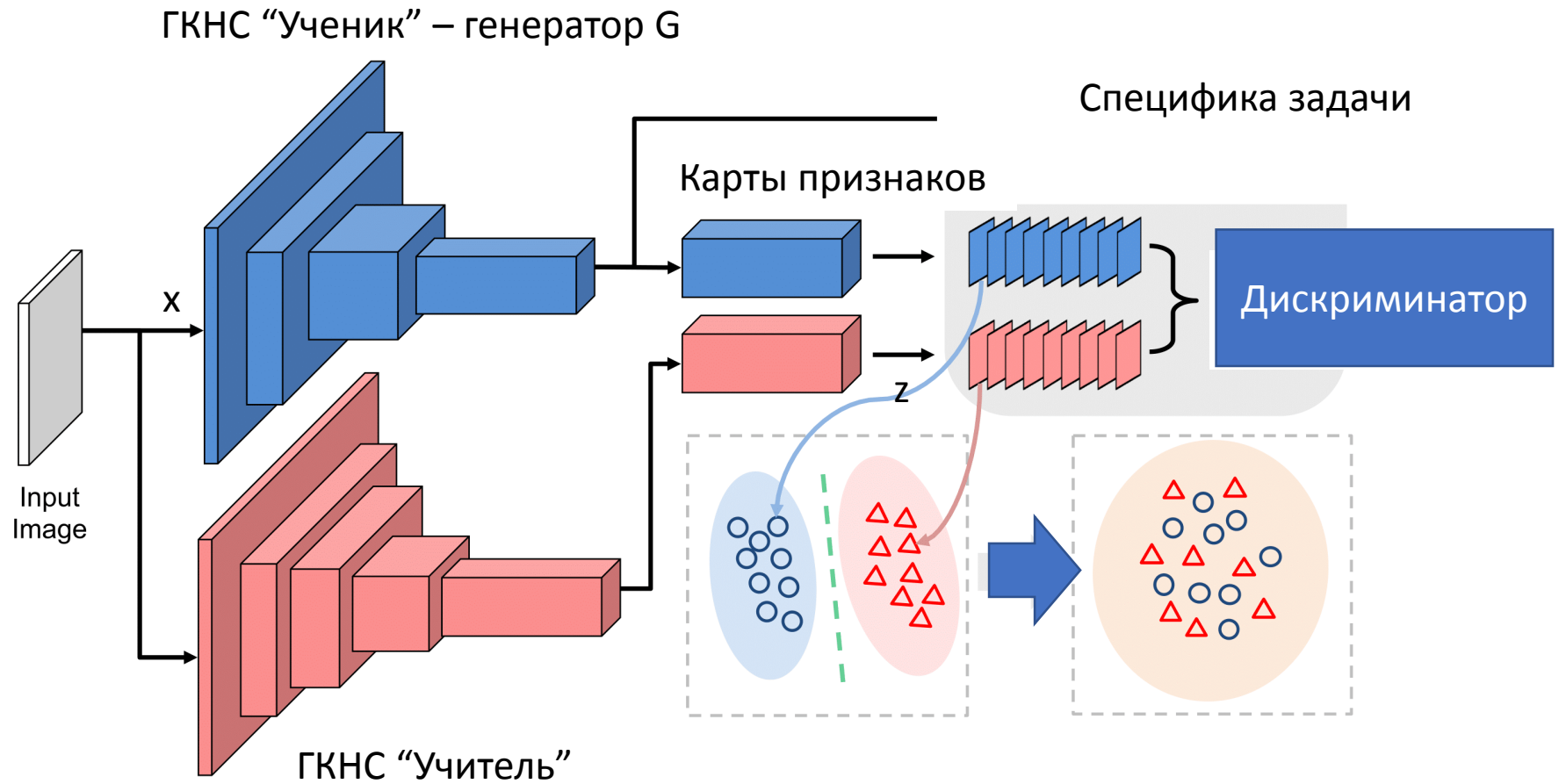
ГКНС “Ученик” G



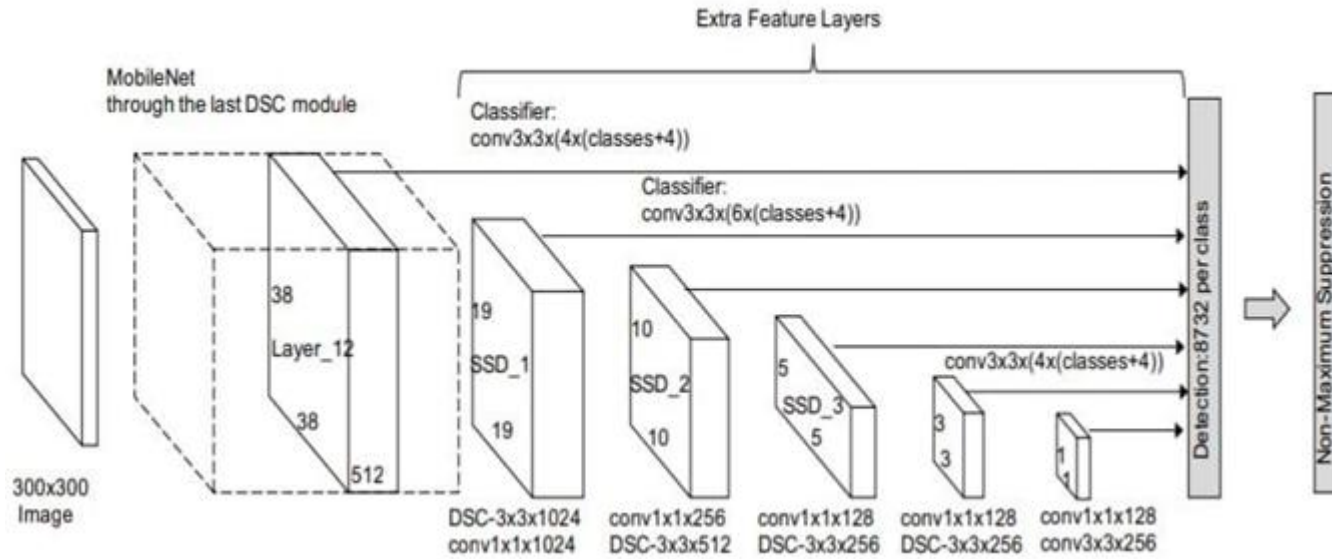
Сеть –учитель выступает в роли источника референтных сигналов

ГКНС “Учитель” T

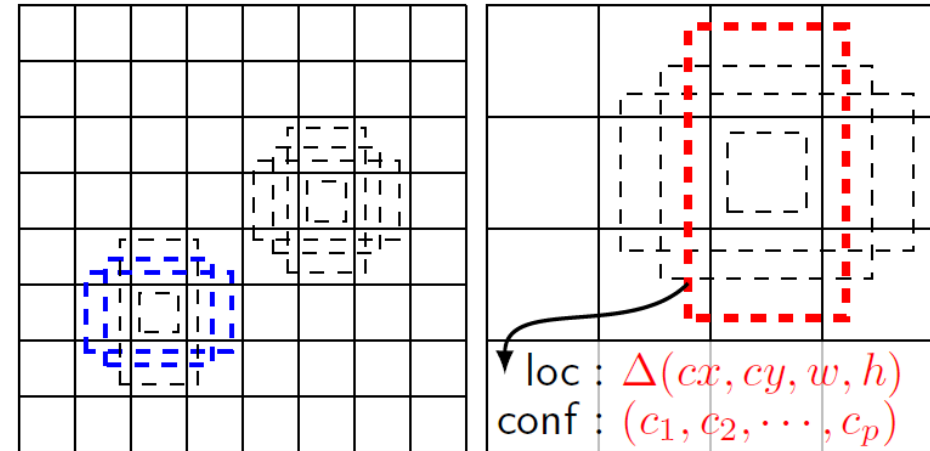
GAN



Алгоритм SSD



Архитектура SSD



(b) 8 × 8 feature map (c) 4 × 4 feature map

Особенности

- ✓ Можно использовать любую архитектуру базовой ГНС
- ✓ Признаки с разных слоев используются обнаружения объектов разных размеров
- ✓ Одна из наиболее популярных на практике

Параметры используемых ГКНС

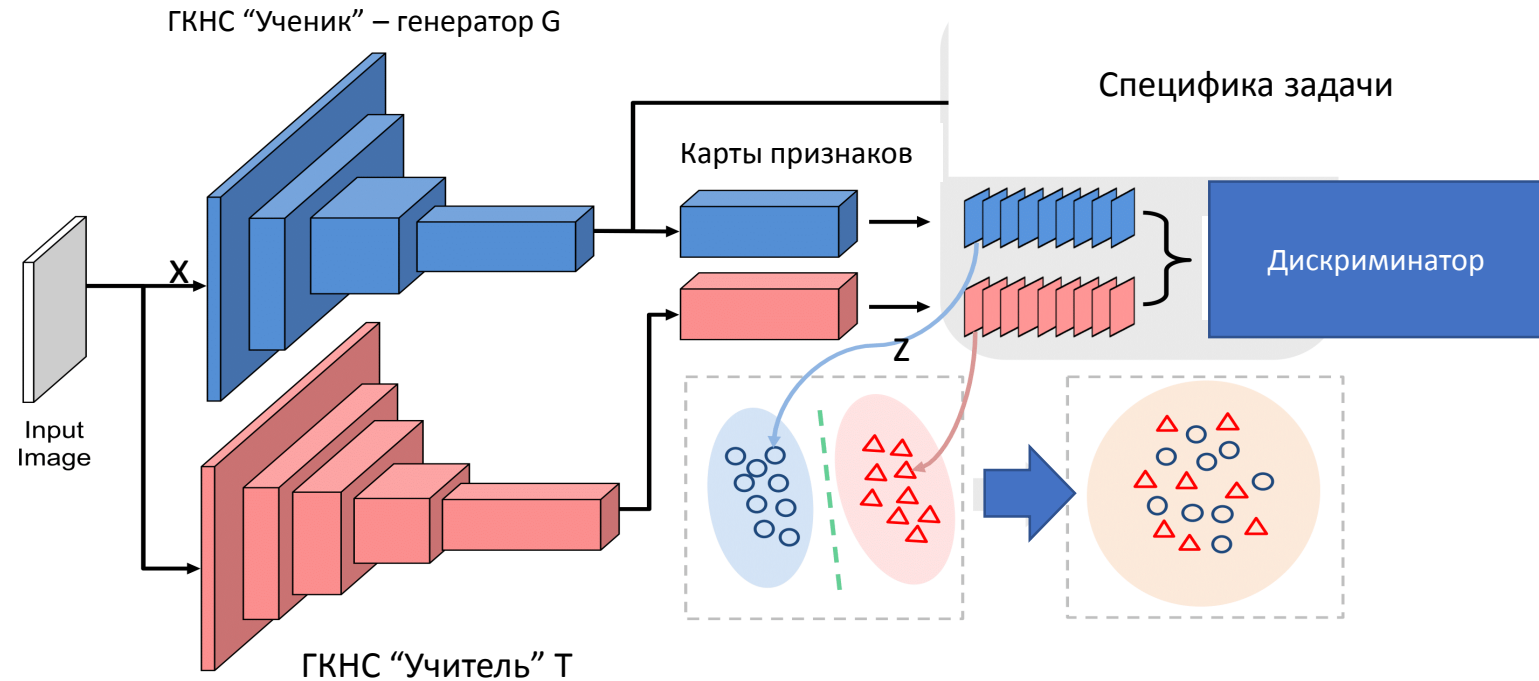
Используемые ГКНС:

Сеть учитель: SSD – DarkNet 53 (предобученная)

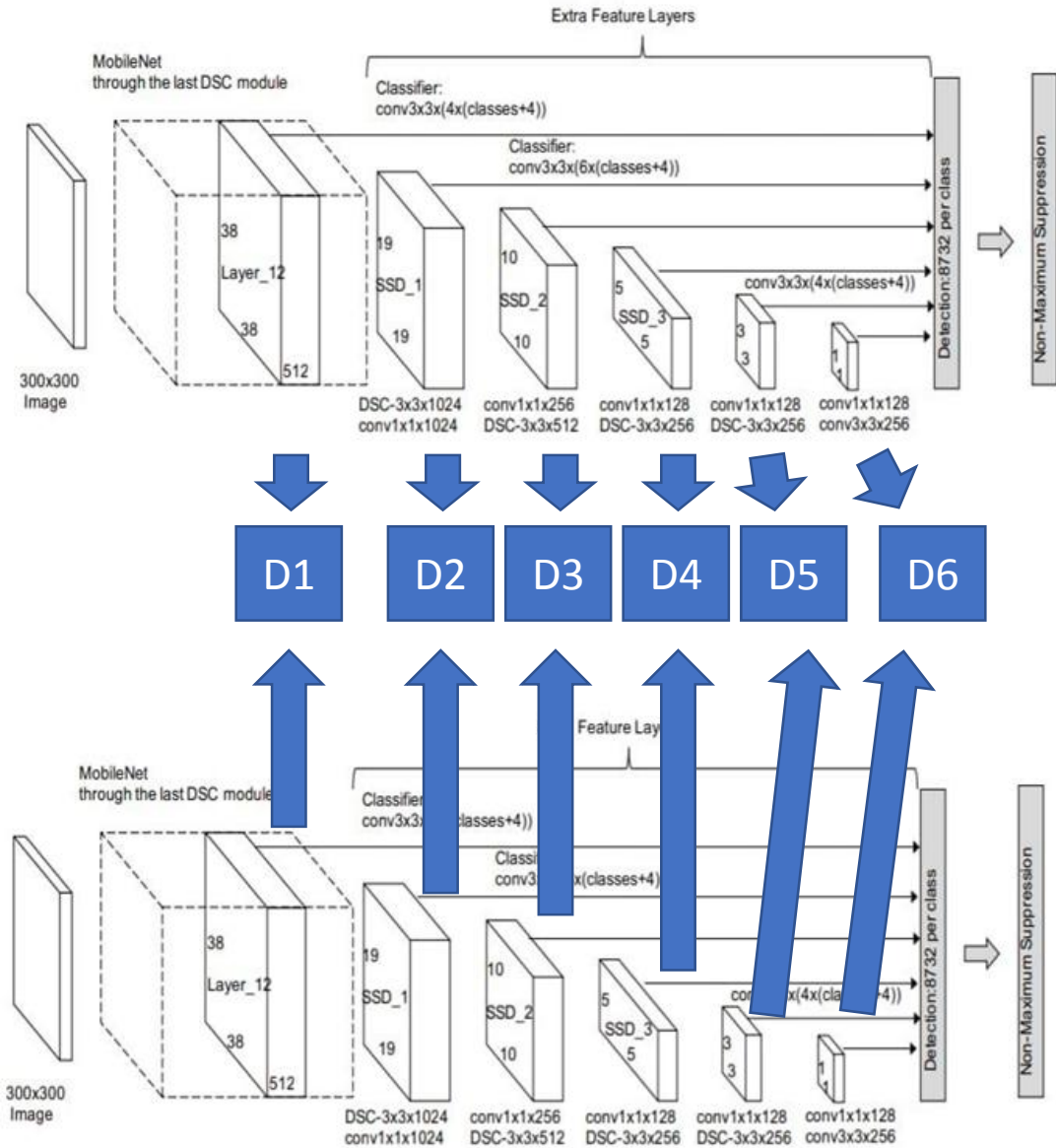
Сеть ученик: MobileNetV2, ShuffleNetV1, MNAS Net

Датасет:

PascalVOC 2007/2012



Описание архитектур



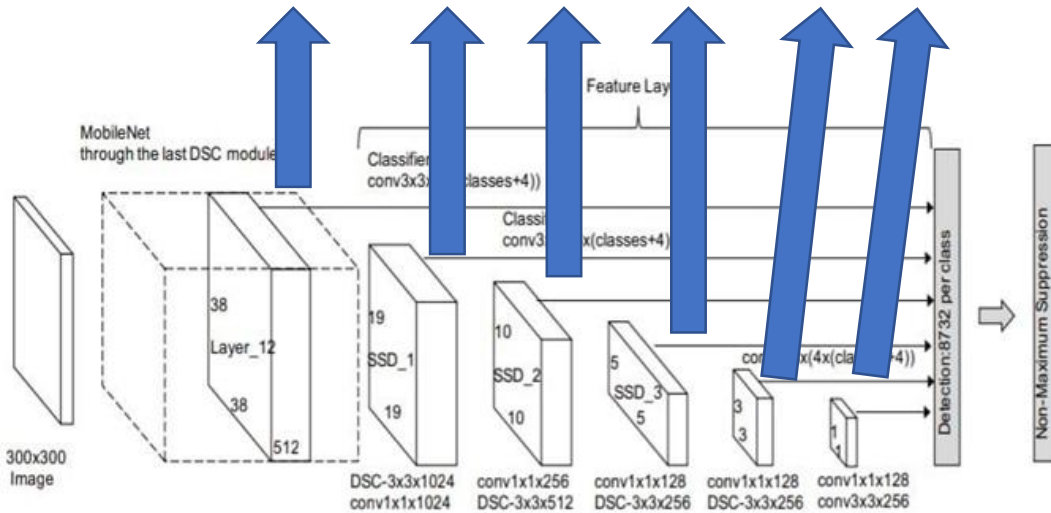
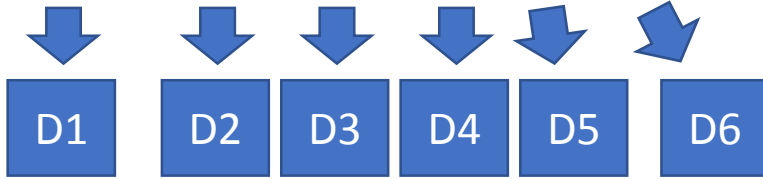
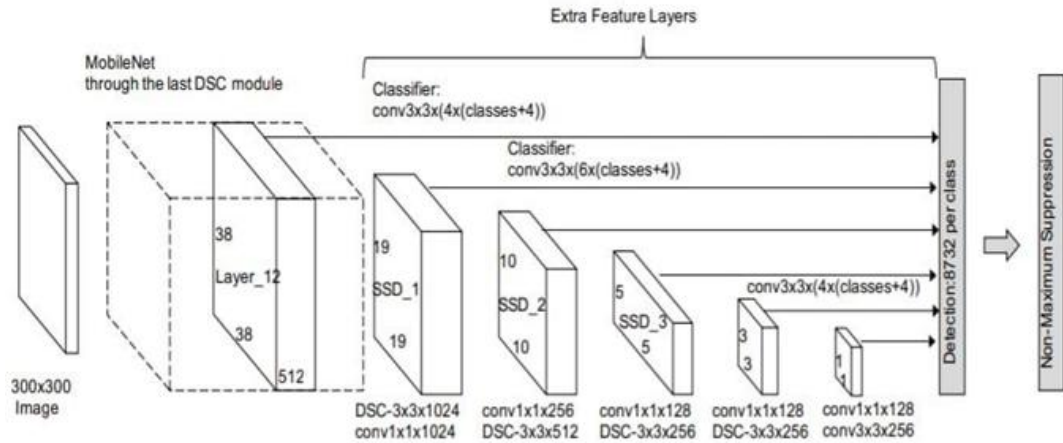
Архитектура Дискриминаторов:

<i>DNet1</i>	<i>DNet2</i>	<i>DNet3</i>	<i>DNet4</i>	<i>DNet5</i>	<i>DNet6</i>
Размер = <u>19x19</u> Conv (512->512, k=3, s=1, p=1, bias= True) InstanceNorm LeakyRelu	Размер = <u>10x10</u> Conv (1024->1024, k=3, s=1, p=1, bias= True) InstanceNorm LeakyRelu	Размер = <u>5x5</u> Conv (512->512, k=3, s=1, p=1, bias= True) InstanceNorm LeakyRelu	Размер = <u>3x3</u> Conv (256->256, k=3, s=1, p=1, bias= True) InstanceNorm LeakyRelu	Размер = <u>2x2</u> Conv (256->256, k=2, s=2, p=1, bias= True) InstanceNorm LeakyRelu	Размер = <u>1x1</u> Conv (256->256, k=1, s=1, p=0, bias= True) LeakyRelu
Размер = <u>19x19</u> Conv (512->512, k=3, s=2, p=1, bias= True) InstanceNorm LeakyRelu	Размер = <u>10x10</u> Conv (1024->1024, k=3, s=2, p=1, bias= True) InstanceNorm LeakyRelu	Размер = <u>5x5</u> Conv (512->512, k=3, s=1, p=0, bias= True) InstanceNorm LeakyRelu	Размер = <u>3x3</u> Conv (256->256, k=3, s=1, p=1, bias= True) InstanceNorm LeakyRelu	Размер = <u>2x2</u> Conv (256->256, k=2, s=2, p=1, bias= True) InstanceNorm LeakyRelu	Размер = <u>1x1</u> Conv (256->256, k=1, s=1, p=0, bias= True) LeakyRelu
Размер = <u>10x10</u> Conv (512->512, k=3, s=2, p=1, bias= True) InstanceNorm LeakyRelu	Размер = <u>5x5</u> Conv (1024->1024, k=3, s=1, p=0, bias= True) InstanceNorm LeakyRelu	Размер = <u>3x3</u> Conv (512->1, k=3, s=1, p=0, bias= False)	Размер = <u>3x3</u> Conv (256->1, k=3, s=1, p=0, bias= False)	Размер = <u>2x2</u> Conv (256->1, k=2, s=1, p=0, bias= False)	Размер = <u>1x1</u> Conv (256->1, k=1, s=1, p=0, bias= False)
Размер = <u>5x5</u> Conv (512->512, k=3, s=1, p=0, bias= True) InstanceNorm LeakyRelu	Размер = <u>3x3</u> Conv (1024->1, k=3, s=1, p=0, bias= False)				
Размер = <u>3x3</u> Conv (512->1, k=3, s=1, p=0, bias= False)					
Sigmoid					

Функция потерь дискриминатора:

$$I_{GAN} = a L_1(x,z) - (\log D(x) + \log(1-D(G(z))))$$

Алгоритм SSD



Алгоритм обучения:

Этап 1. Обучение дискриминаторов

$$loss = - \sum I_{GAN}$$

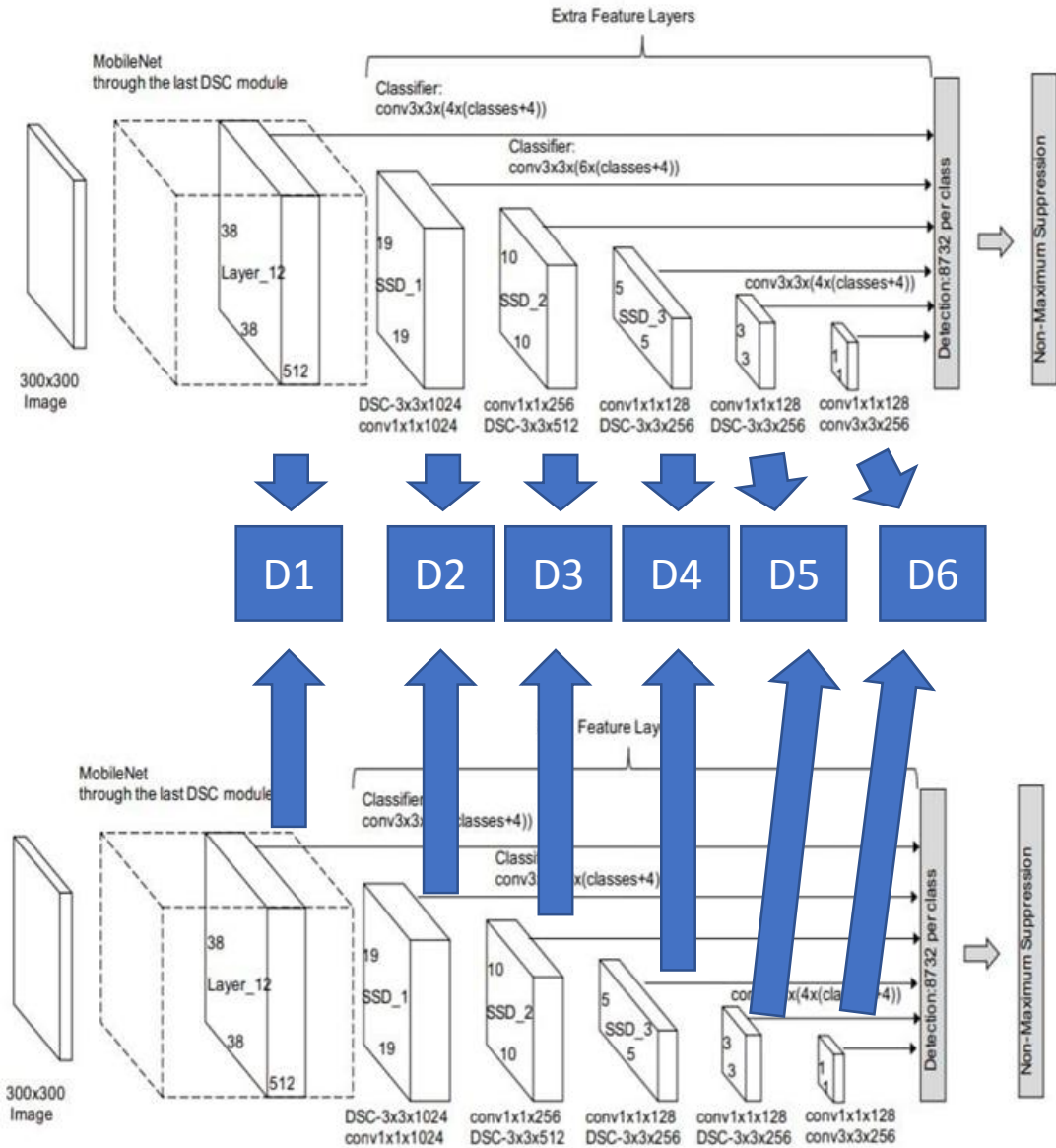
Этап 2. Мимикрия

$$loss = I_{Multibox} + \sum I_{GAN}$$

Этап 3. Обучение “ученика”

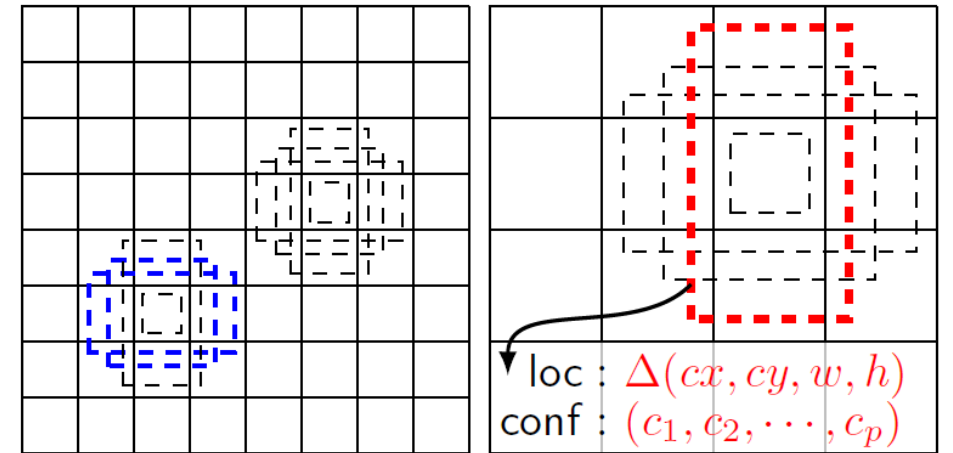
$$loss = I_{Multibox}$$

Алгоритм SSD



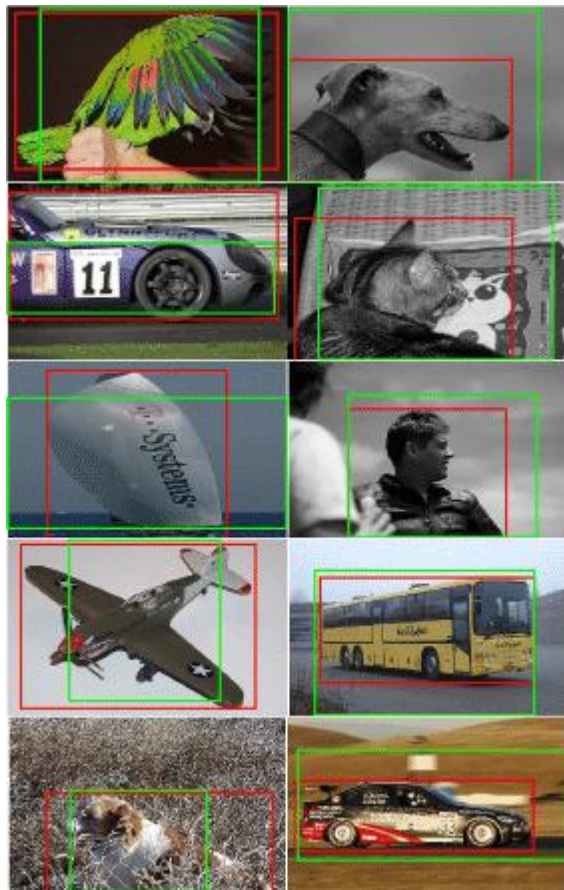
Вклад D соотв. Различным размерам :

Размер объектов	Прирост AP
Малые(D1,D2)	+ 8%
Средние(D3,D4)	+ 14%
Большие(D5,D6)	+ 12%



(b) 8 × 8 feature map (c) 4 × 4 feature map

Результаты



<u>ГКНС “ученик”</u>	<u>ГКНС “учитель”</u>	<u>Test VOC07 mAP, %</u>
SSD_Lite_MobileNetV2	-	69.3
SSD_Lite_MobileNetV2	SSD_DarkNet53	71.2
SSD_Lite_ShuffleNetV1	-	62.4
SSD_Lite_ShuffleNetV1	SSD_DarkNet53	65.6
SSD_Lite_MNASnet1	-	65.8
SSD_Lite_MNASnet1	SSD_DarkNet53	70.2
Pascal + MS Coco		
SSD_Lite_MobileNetV2	-	71.9
SSD_Lite_MobileNetV2	SSD_DarkNet53	73.4

Выводы:

- ✓ Предлагаемый алгоритм позволяет существенно повысить качество обнаружения
- ✓ Не требует изменения архитектуры
- Очень чувствителен к выбору параметров обучения