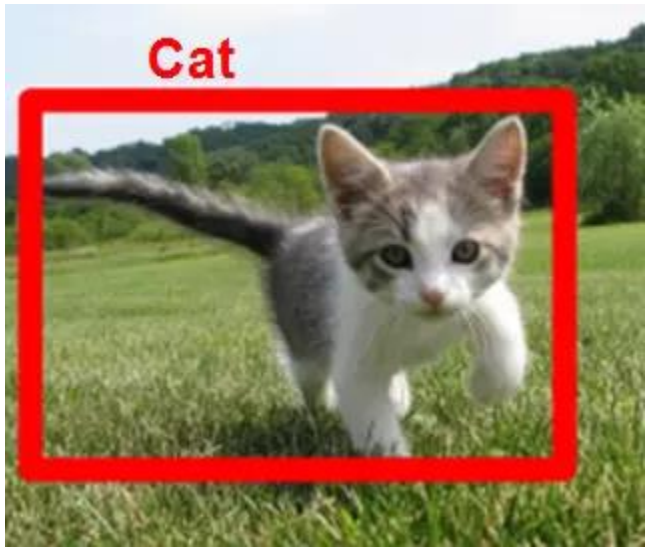
物体検出技術（SSD）

2020/02/25

馬　力

物体検出のイメージ



物体検出(Object Detection)

➢類別：

- マルチカテゴリ検出multi-categories detection
- エッジ検出edge detection
- 注目物体抽出salient object detection
- ポーズ検出pose detection
- シーンテキスト位置scene text detection
- 顔検出face detection
- 歩行者検出pedestrian detection

など

➢ 運用領域

セキュリティ
軍事
輸送
医療
ライフフィールド

➢ データ

画像
vedio

物体検出の発展

- DL前時代、 HOG-like特徴抽出
- R-CNN
- Fast R-CNN
- Faster R-CNN
- Mask R-CNN
- SSD (Single-shot Detection)
- YOLO
- YOLO_v2
- YOLO_v3
- YOLO_v4

概念

ボックス

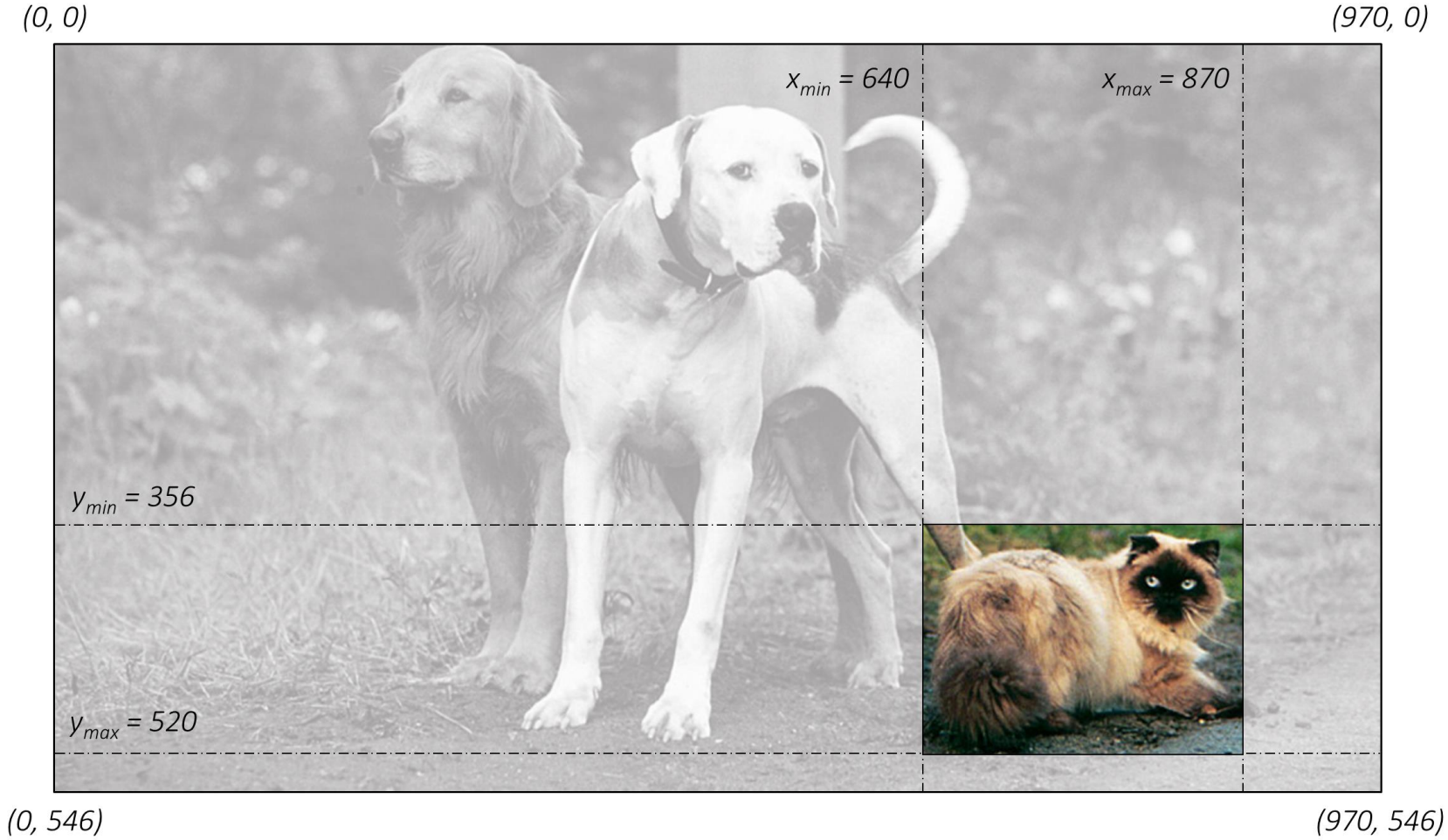
種類

確率数字

ボックス

- ボックス
- バウンディング

  Bounding box

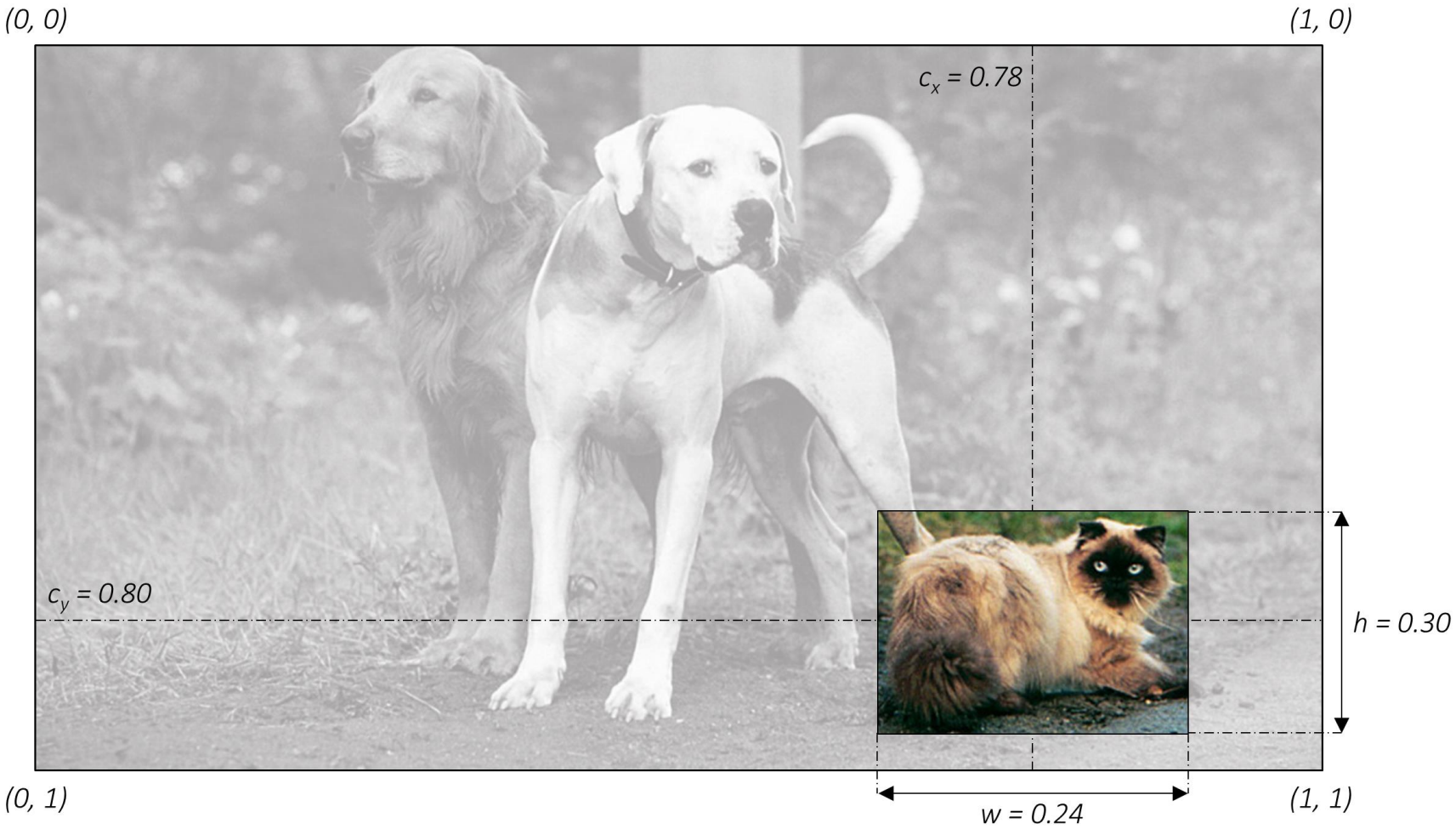


バウンディング座標　$(x_{min}, y_{min}, x_{max}, y_{max}) = (640, 356, 870, 520)$

ボックスの（left, top）, (right, bottom)

概念

- ボックス



中心サイズ座標 $(c_x, c_y, w, h) = (0.78, 0.8, 0.24, 0.30)$

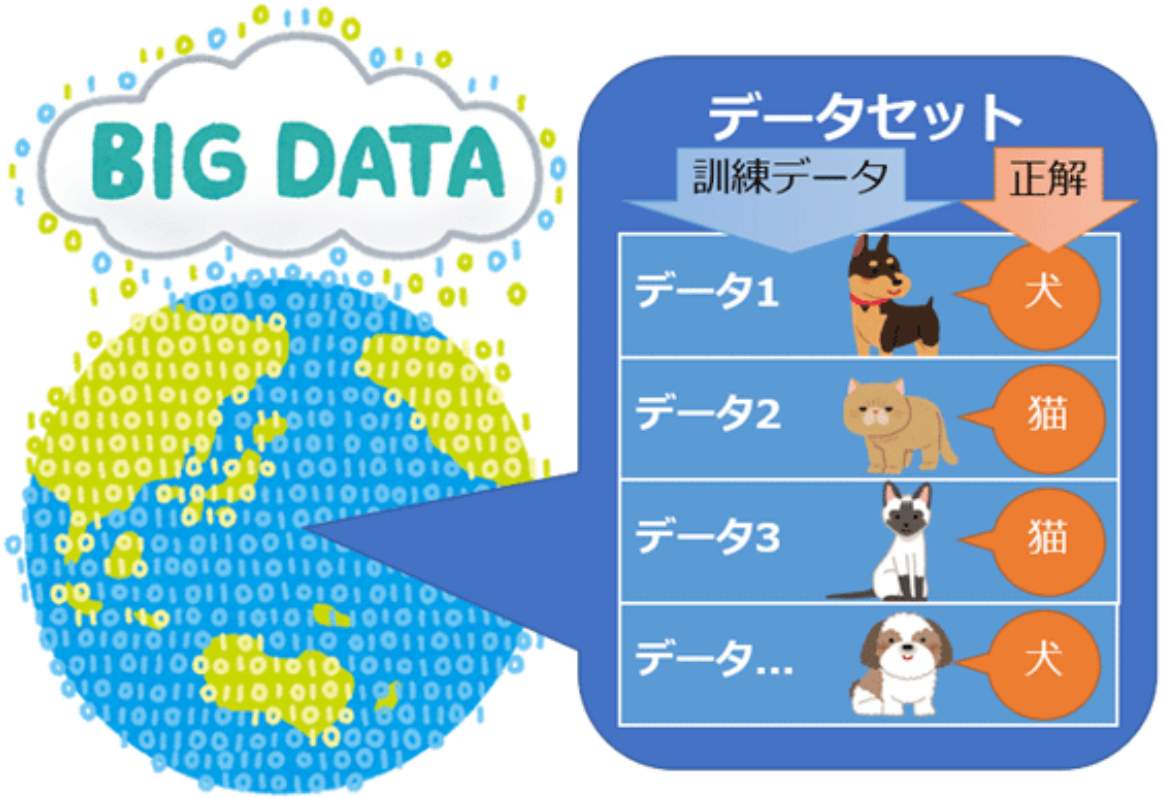
ボックスの中心と長さ、幅

Ground Truth

- 正解、教師データ
- 教師あり学習

物体検出の場合

教師データに
1、種類情報
2、位置情報

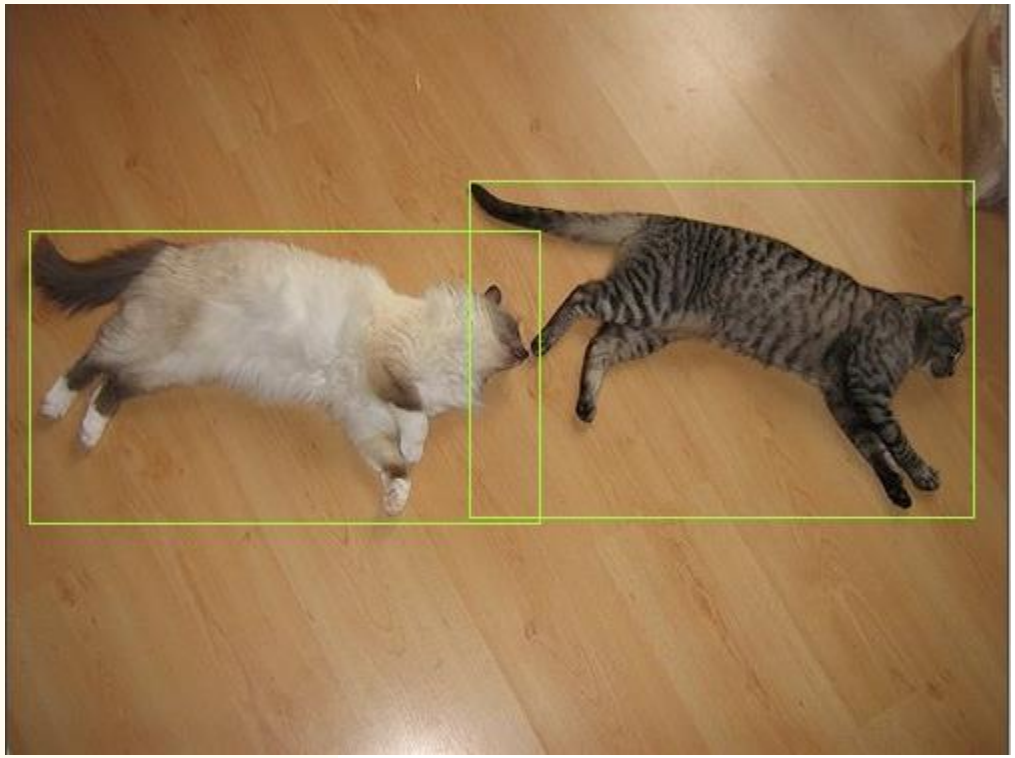


Ground Truth

アノテーション

```
<annotation>
    <folder>VOC2007</folder>
    <filename>000019.jpg</filename>
    <source>
        <database>The VOC2007 Database</database>
        <annotation>PASCAL VOC2007</annotation>
        <image>flickr</image>
        <flickrid>330638158</flickrid>
    </source>
    <owner>
        <flickrid>Rosenberg1/ Simmo</flickrid>
        <name>?</name>
    </owner>
    <size>
        <width>500</width>
        <height>375</height>
        <depth>3</depth>
    </size>
    <segmented>0</segmented>
    <object>
        <name>cat</name>
        <pose>Right</pose>
        <truncated>0</truncated>
        <difficult>0</difficult>
        <bndbox>
            <xmin>231</xmin>
            <ymin>88</ymin>
            <xmax>483</xmax>
            <ymax>256</ymax>
        </bndbox>
    </object>
    <object>
        <name>cat</name>
        <pose>Right</pose>
        <truncated>0</truncated>
        <difficult>0</difficult>
        <bndbox>
            <xmin>11</xmin>
            <ymin>113</ymin>
            <xmax>266</xmax>
            <ymax>259</ymax>
        </bndbox>
    </object>
</annotation>
```

元データ

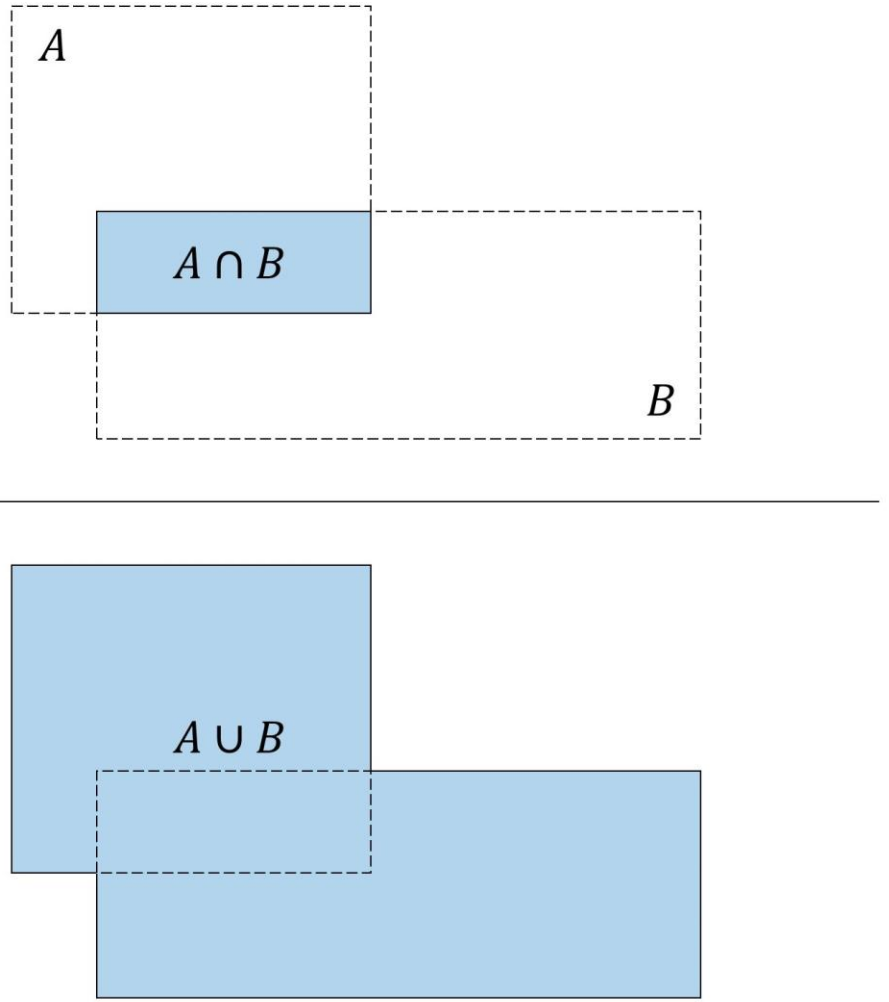


Ground Truth可視化

判断標準

- IoU

$$Jaccard\ Index = \frac{A \cap B}{A \cup B} =$$



データセット

- PASCAL VOC
- MS COCO
- ImageNet
- Caltech
- KITTI
- Open Images V5
- Vision Meets Drones

など

backboneモデル選択

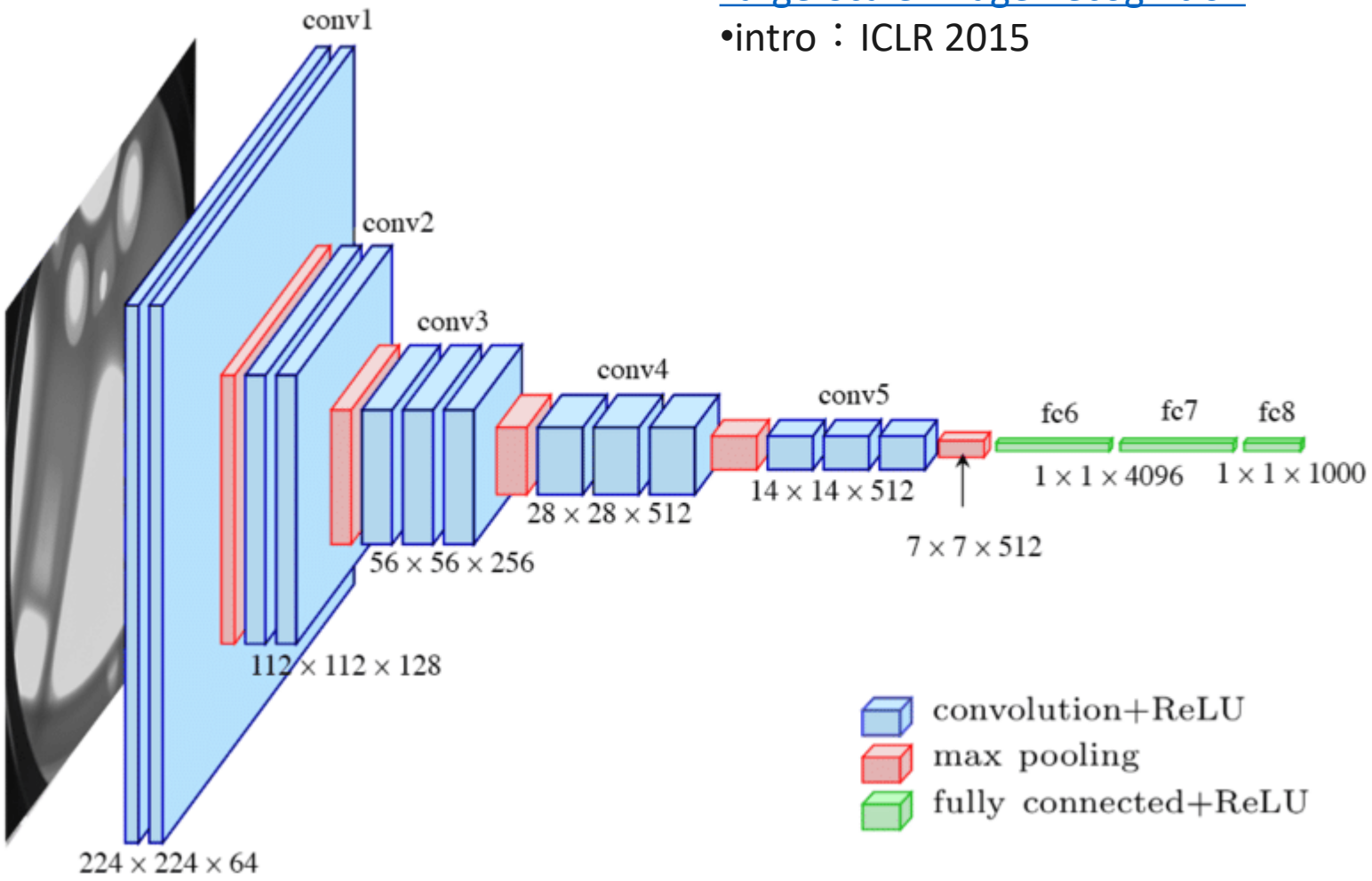
- 原則：精度と効率

精度：
ResNet
ResNeXt
AmoebaNet

効率 ：
MobileNe
ShuffleNet
SqueezeNet
Xception
MobileNetV2

VGG

•arXiv：[1409.1556] Very Deep Convolutional Networks for Large-Scale Image Recognition
•intro：ICLR 2015



損失関数：smooth-l1-loss

- https://stats.stackexchange.com/questions/351874/how-to-interpret-smooth-l1-loss/369380

Smooth L1-loss can be interpreted as a combination of L1-loss and L2-loss. It behaves as L1-loss when the absolute value of the argument is high, and it behaves like L2-loss when the absolute value of the argument is close to zero. The equation is:

$$L_{1;smooth} = \begin{cases} |x| & \text{if } |x| > \alpha; \\ \frac{1}{|\alpha|}x^2 & \text{if } |x| \leq \alpha \end{cases}$$

α is a hyper-parameter here and is usually taken as 1. $\frac{1}{\alpha}$ appears near x^2 term to make it continuous.

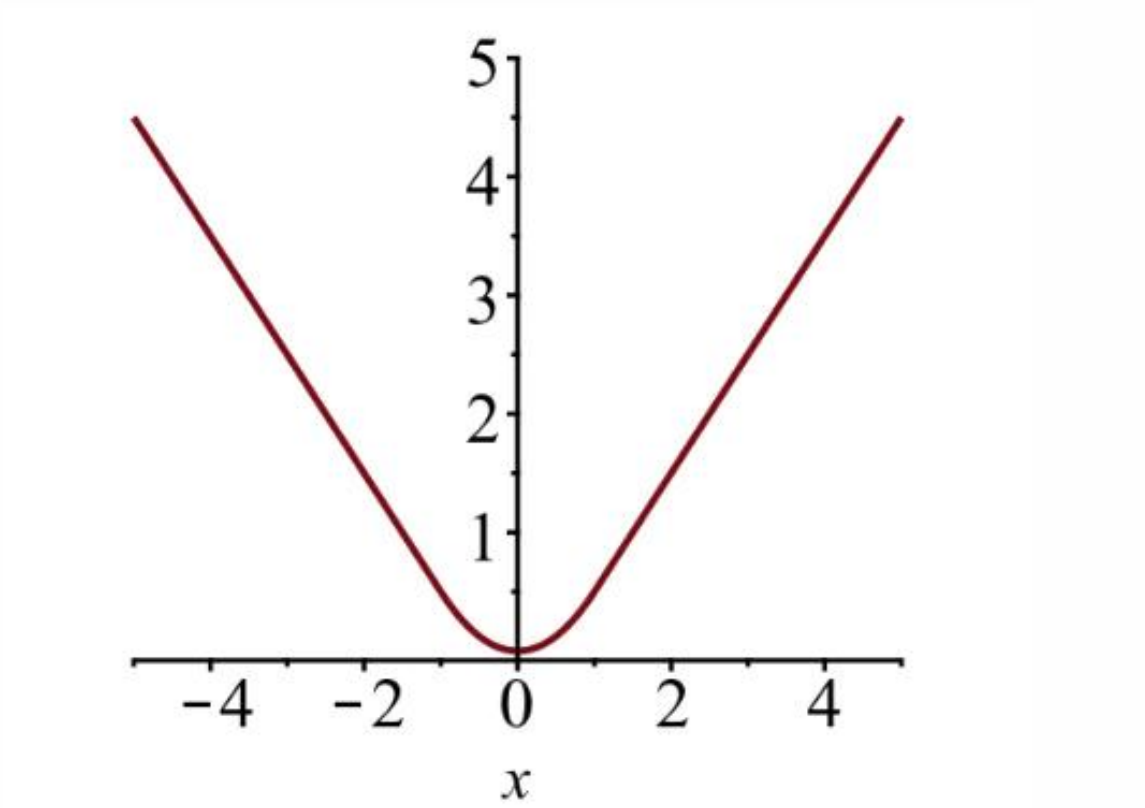
Smooth L1-loss combines the advantages of L1-loss (steady gradients for large values of x) and L2-loss (less oscillations during updates when x is small).

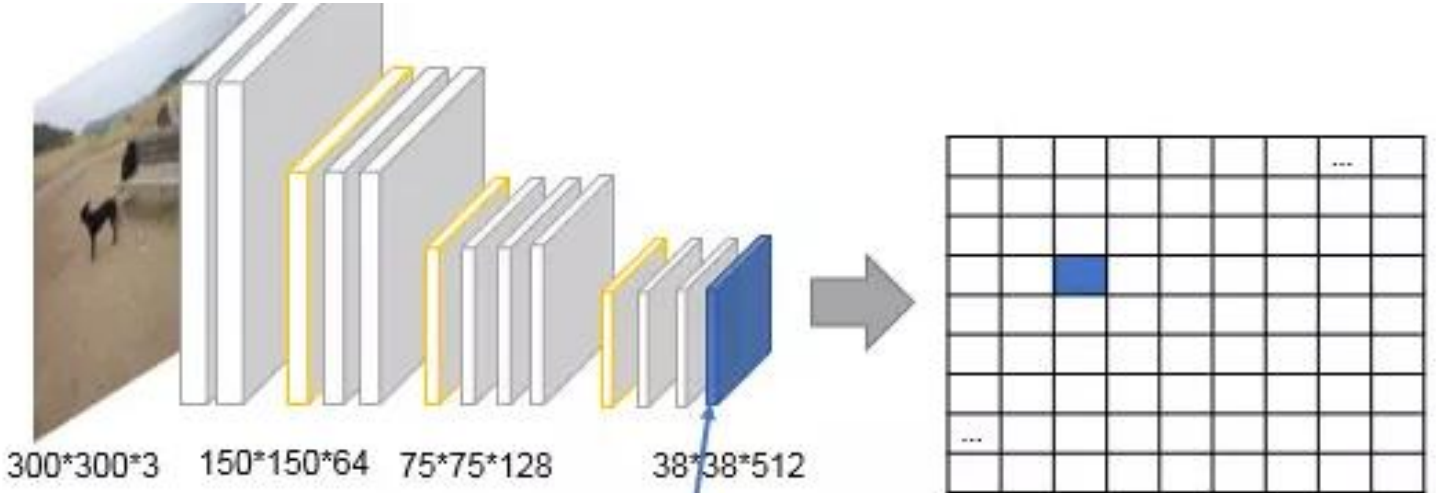
Another form of smooth L1-loss is Huber loss. They achieve the same thing. Taken from Wikipedia, Huber loss is

$$L_\delta(a) = \begin{cases} \frac{1}{2}a^2 & \text{for } |a| \leq \delta, \\ \delta(|a| - \frac{1}{2}\delta), & \text{otherwise.} \end{cases}$$

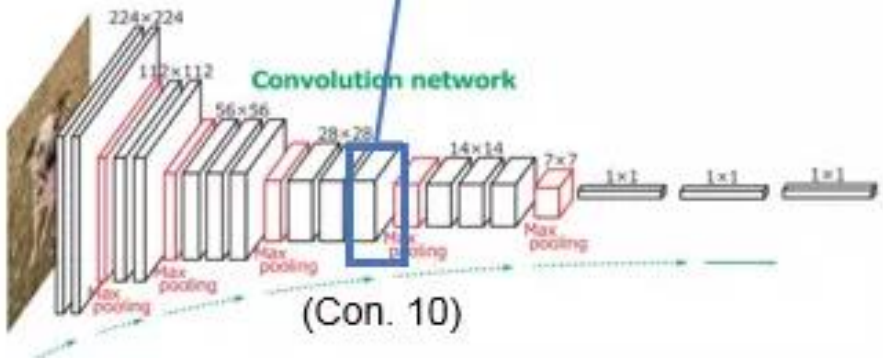
損失関数：smooth-l1-loss

$$L_1^{\text{smooth}}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases}$$



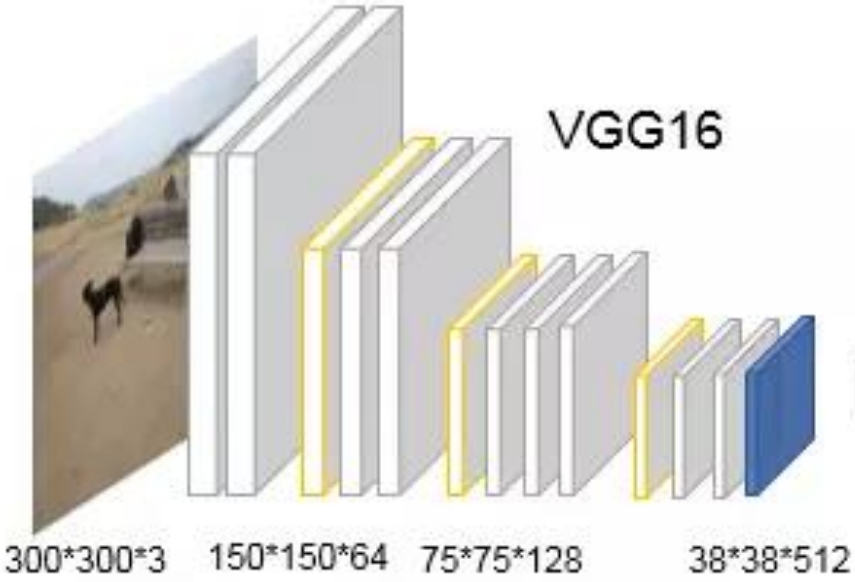


Middle layer of VGG16

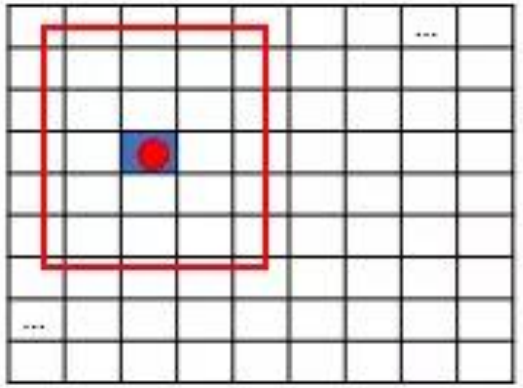


特徴マップ

分類



回帰タスク
One box per position



38*38*512

20種類＋背景

目標

25-d vector:

バウンディングボックス回帰

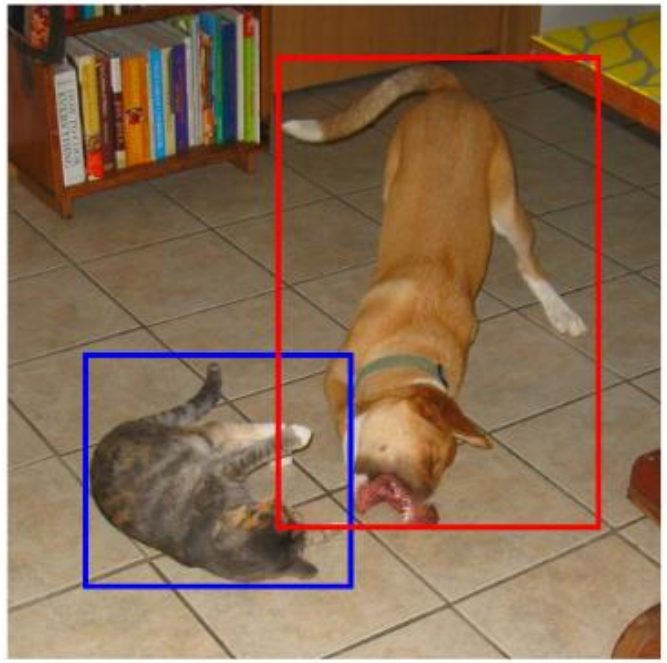
分類

cx, cy: **セル自体中心座標から距離**

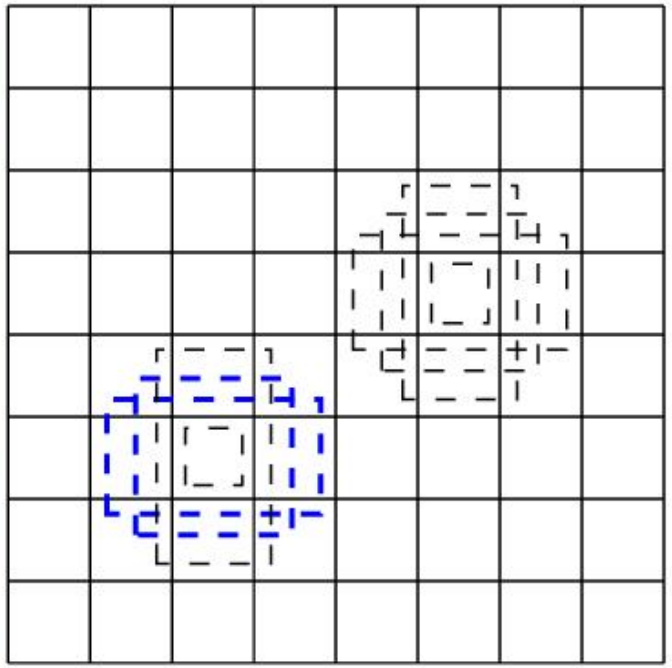
w, h: **画像サイズから比率**

回帰(Regression)

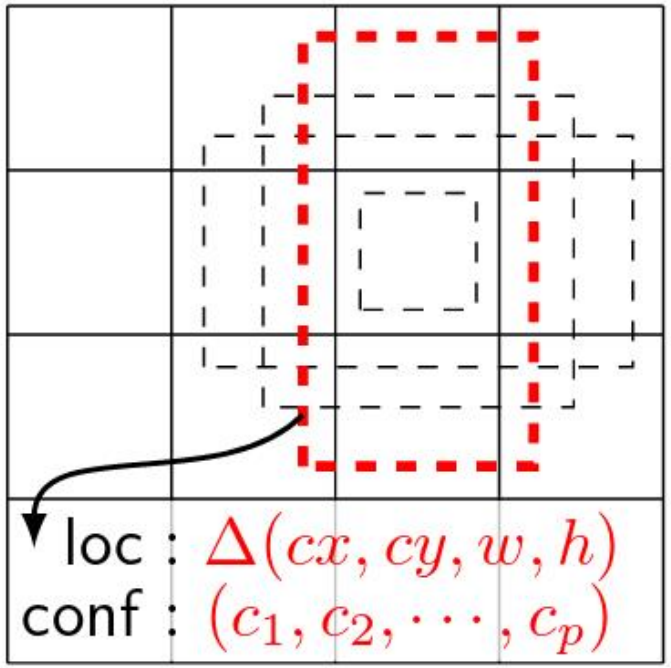
元のデフォルトボックスの オフセットの値4つ（x座標、y座標、幅、高さ）



画像とground truth

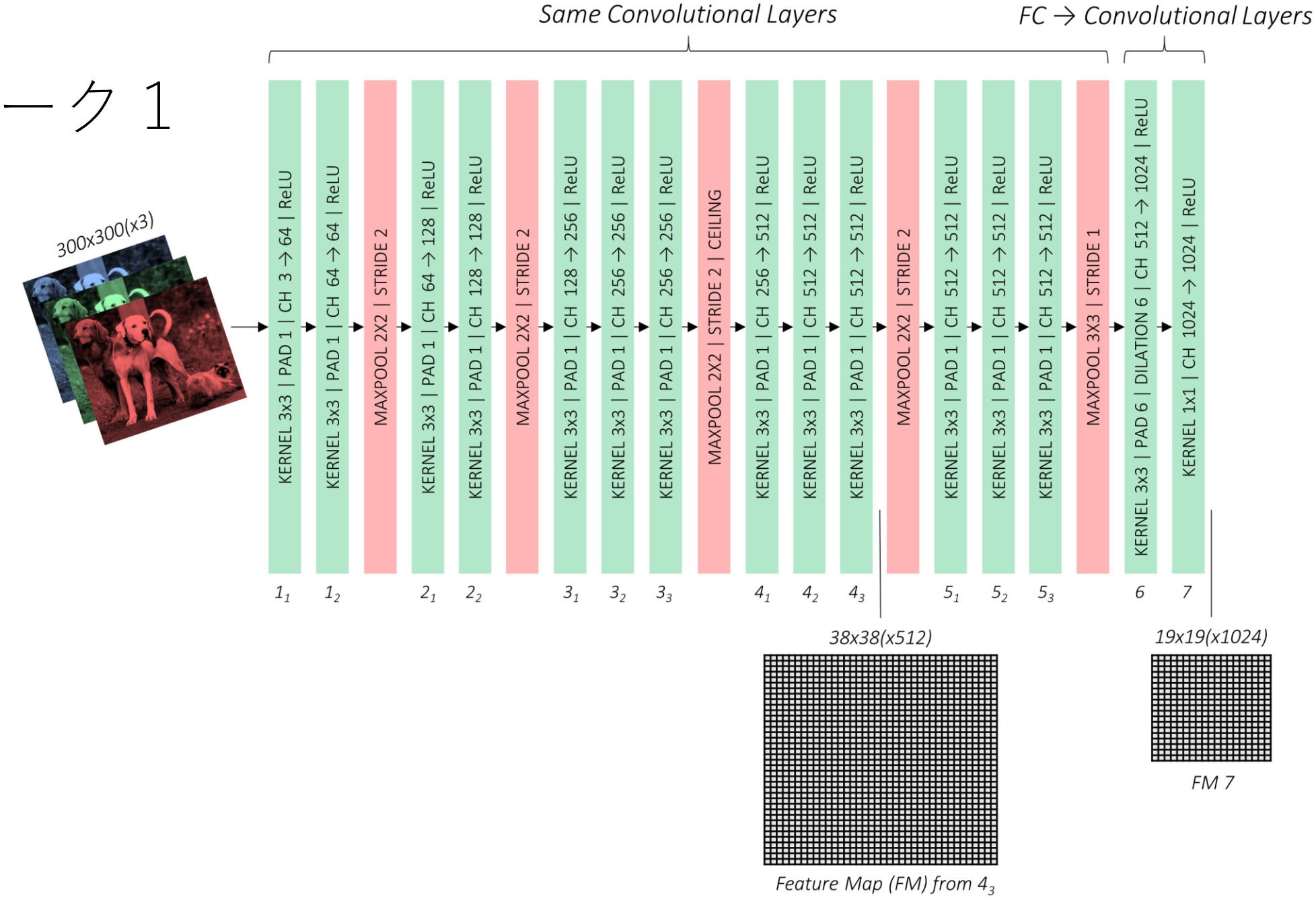


8×8 特徴マップ

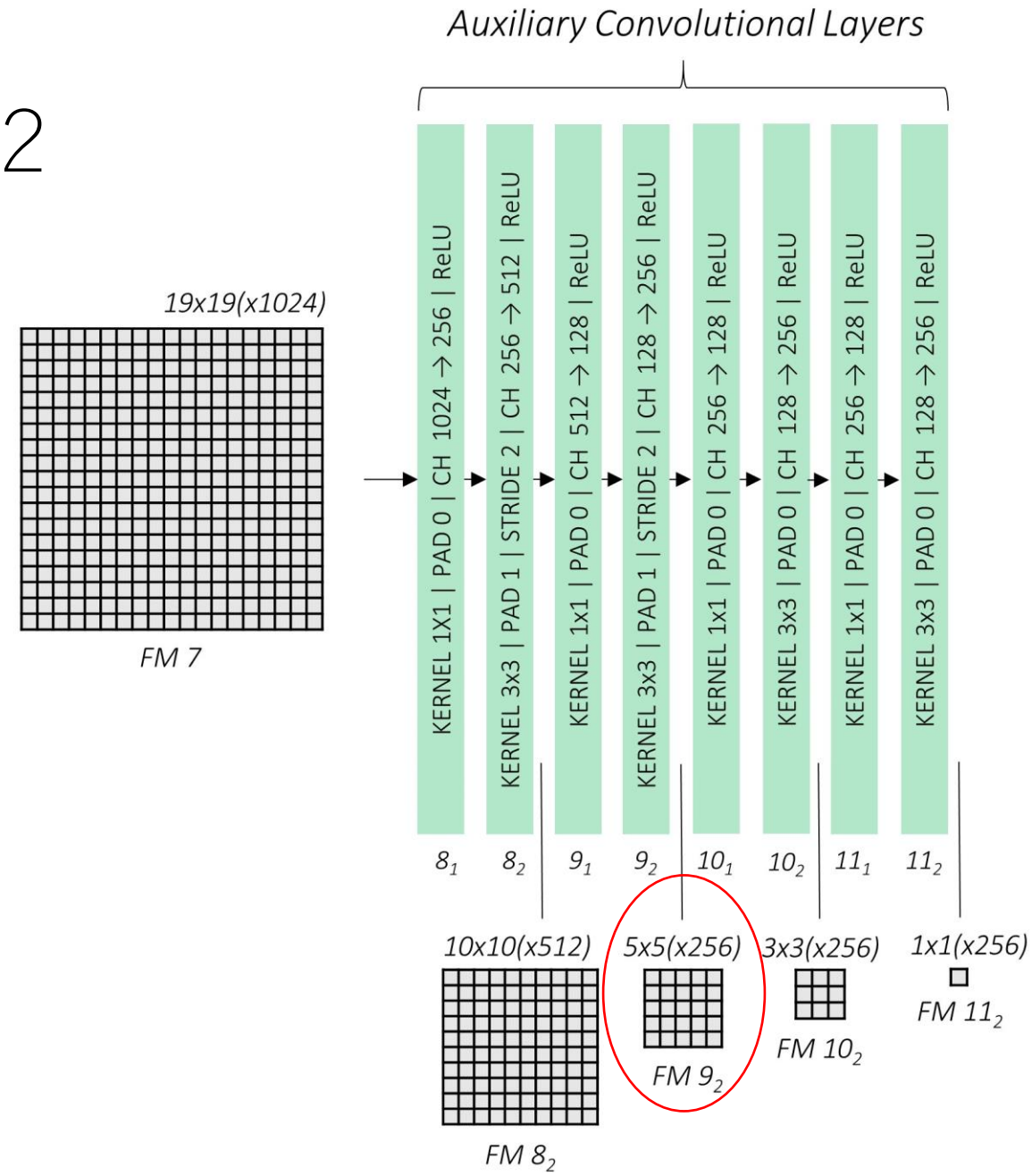


4×4 特徴マップ

SSD
ネットワーク１


Same Convolutional Layers
FC → Convolutional Layers
300x300(x3)
KERNEL 3x3 | PAD 1 | CH 3 → 64 | ReLU
KERNEL 3x3 | PAD 1 | CH 64 → 64 | ReLU
MAXPOOL 2X2 | STRIDE 2
KERNEL 3x3 | PAD 1 | CH 64 → 128 | ReLU
KERNEL 3x3 | PAD 1 | CH 128 → 128 | ReLU
MAXPOOL 2X2 | STRIDE 2
KERNEL 3x3 | PAD 1 | CH 128 → 256 | ReLU
KERNEL 3x3 | PAD 1 | CH 256 → 256 | ReLU
KERNEL 3x3 | PAD 1 | CH 256 → 256 | ReLU
MAXPOOL 2X2 | STRIDE 2 | CEILING
KERNEL 3x3 | PAD 1 | CH 256 → 512 | ReLU
KERNEL 3x3 | PAD 1 | CH 512 → 512 | ReLU
KERNEL 3x3 | PAD 1 | CH 512 → 512 | ReLU
MAXPOOL 2X2 | STRIDE 2
KERNEL 3x3 | PAD 1 | CH 512 → 512 | ReLU
KERNEL 3x3 | PAD 1 | CH 512 → 512 | ReLU
KERNEL 3x3 | PAD 1 | CH 512 → 512 | ReLU
MAXPOOL 3X3 | STRIDE 1
KERNEL 3x3 | PAD 6 | DILATION 6 | CH 512 → 1024 | ReLU
KERNEL 1x1 | CH 1024 → 1024 | ReLU
1_1 1_2 2_1 2_2 3_1 3_2 3_3 4_1 4_2 4_3 5_1 5_2 5_3 6 7
38x38(x512)
19x19(x1024)
FM 7
Feature Map (FM) from 4_3

SSD
ネットワーク 2


Auxiliary Convolutional Layers
19x19(x1024)
FM 7
KERNEL 1X1 | PAD 0 | CH 1024 → 256 | ReLU
KERNEL 3x3 | PAD 1 | STRIDE 2 | CH 256 → 512 | ReLU
KERNEL 1x1 | PAD 0 | CH 512 → 128 | ReLU
KERNEL 3x3 | PAD 1 | STRIDE 2 | CH 128 → 256 | ReLU
KERNEL 1x1 | PAD 0 | CH 256 → 128 | ReLU
KERNEL 3x3 | PAD 0 | CH 128 → 256 | ReLU
KERNEL 1x1 | PAD 0 | CH 256 → 128 | ReLU
KERNEL 3x3 | PAD 0 | CH 128 → 256 | ReLU
8_1 8_2 9_1 9_2 10_1 10_2 11_1 11_2
10x10(x512)
5x5(x256)
3x3(x256)
1x1(x256)
FM 8_2
FM 9_2
FM 10_2
FM 11_2

物体の多様性

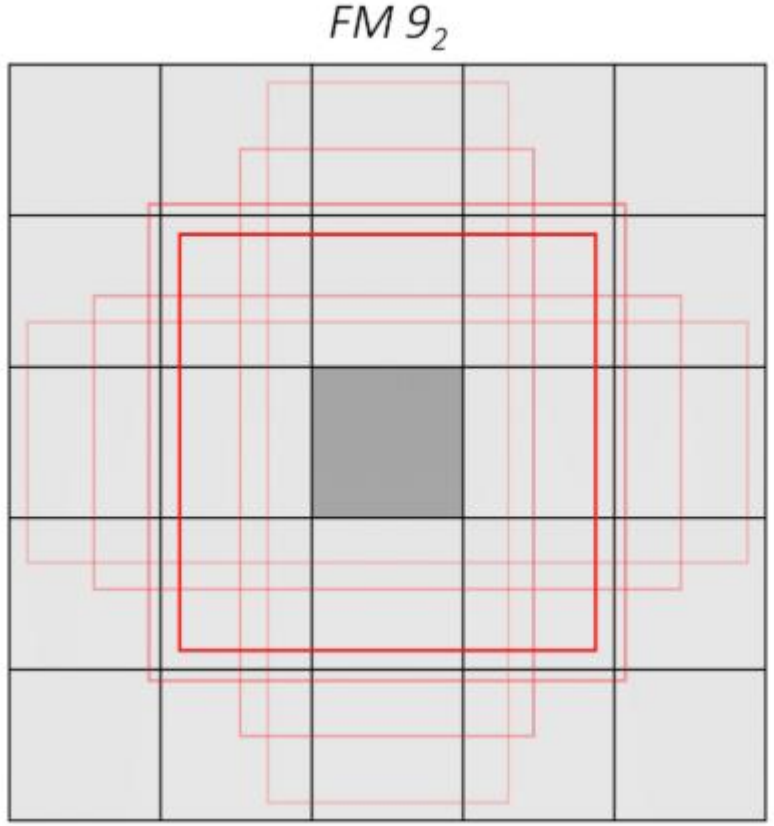
物体の幅、高さはそれぞれ違う

アスペクト比
2:1

アスペクト比
1:2

アスペクト比
1:1

物体の多様性対策



例：Conv9_2からの特徴マップ

- 各セルに対して、先験的なアスペクト比
- 1:1　　0.55, 0.65
- 2:1
- 3:1
- 1:2
- 1:3

Anchor Scale

- mはレイヤ数に対応
  - 大きいほど、小さいオブジェクトを扱える
- s_kは、$k \in [1, m]$でオブジェクトサイズ。

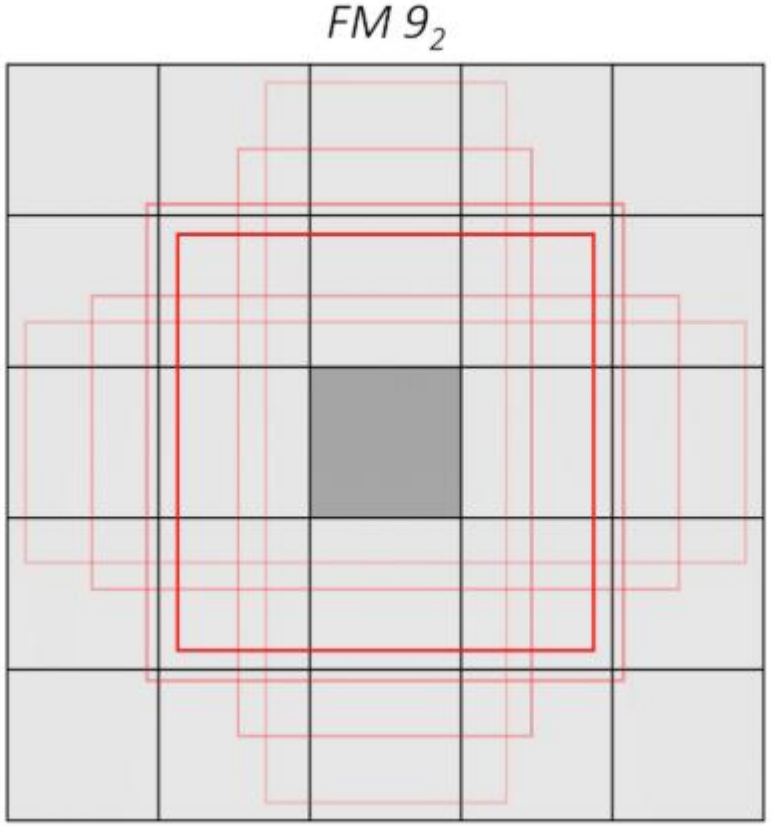
$$S_{k=s_{min}+\frac{s_{max}-s_{min}}{m-1}(k-1)}$$

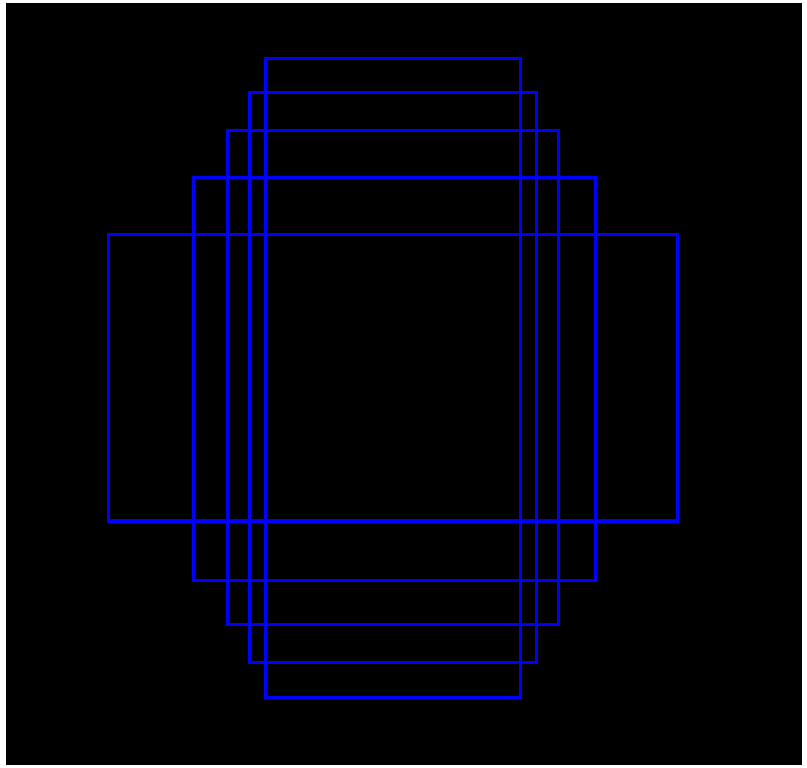
- 例えば、s_{min}=0.2, s_{max}=0.9,一番小さいスケールが0.2 で、大きいのが0.9。その間は等間隔となる。
- 上記のケース結果結果：

  m=6

  s=[0.2, 0.34, 0.48, 0.62, 0.76, 0.9]

物体の多様性対策





$h_k^a = s_k/\sqrt{a_r}$

$w_k^a = s_k\sqrt{a_r}$

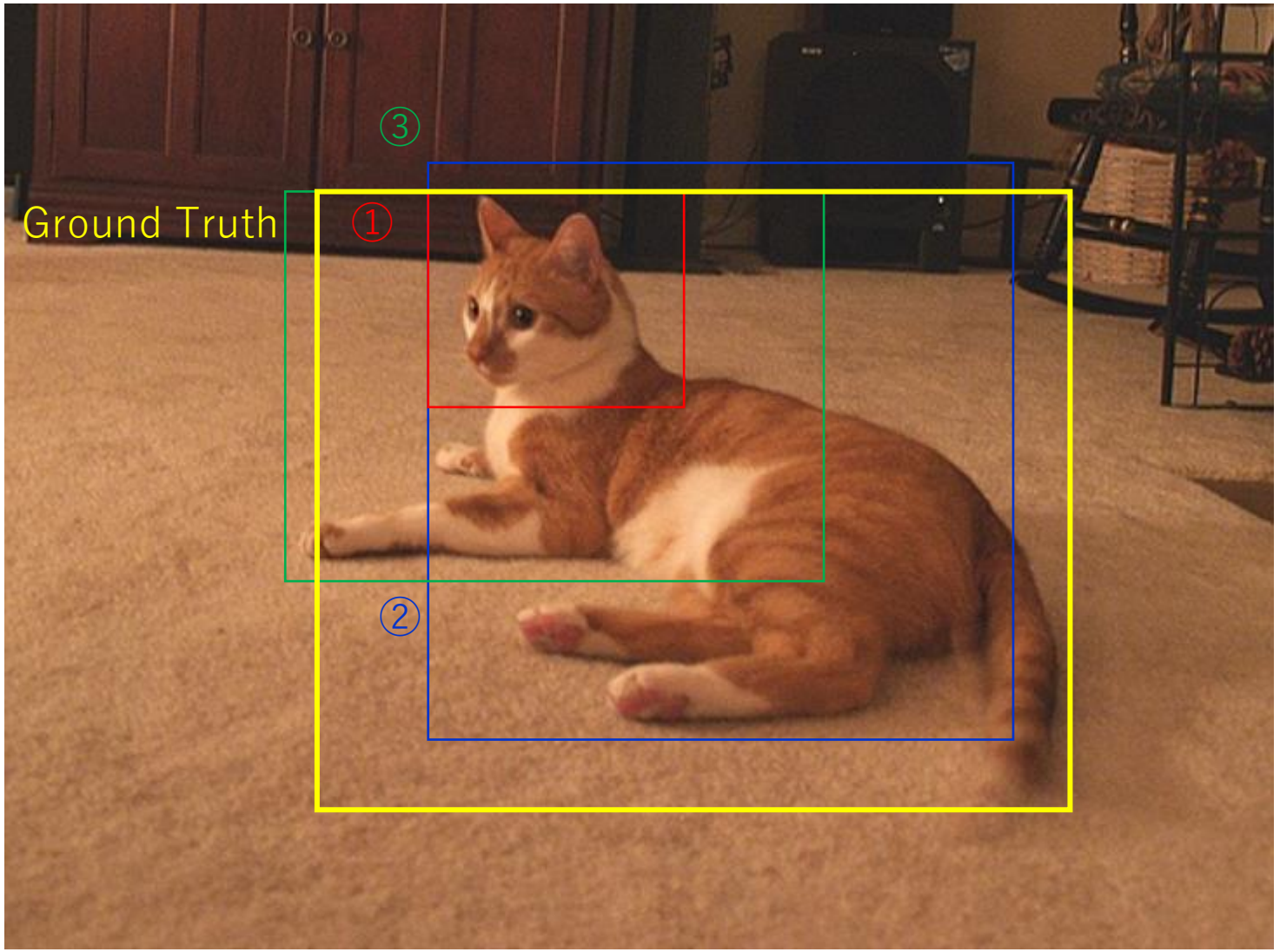
$a \in \left[1,2,3,\frac{1}{2},\frac{1}{3}\right]$

候補枠まとめ

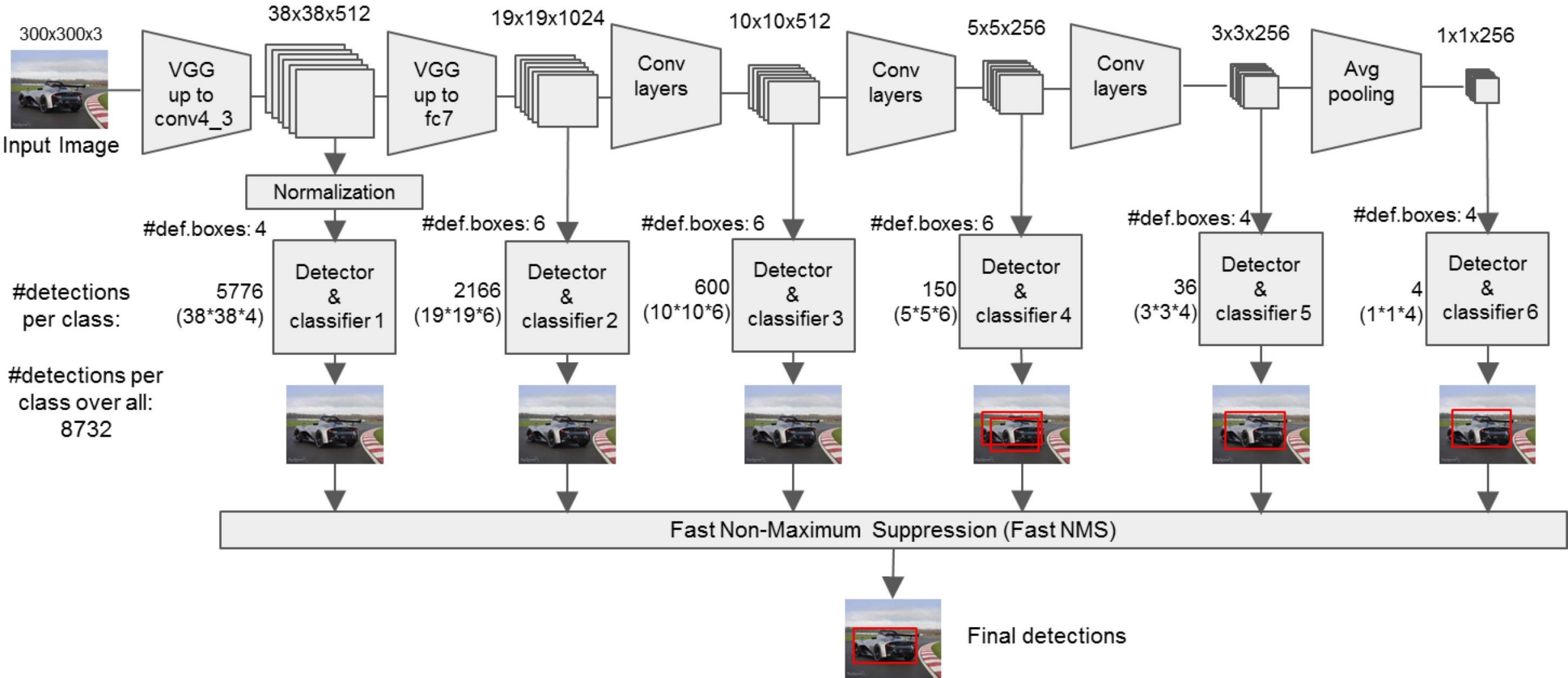
特徴マップ	特徴マップ 次元	先験的 スケール	アスペクト比	先験的 枠数/位置	特徴マップ 先験枠総数
conv4_3	38, 38	0.1	1:1, 2:1, 1:2 + an extra prior	4	5776
conv7	19, 19	0.2	1:1, 2:1, 1:2, 3:1, 1:3 + an extra prior	6	2166
conv8_2	10, 10	0.375	1:1, 2:1, 1:2, 3:1, 1:3 + an extra prior	6	600
conv9_2	5, 5	0.55	1:1, 2:1, 1:2, 3:1, 1:3 + an extra prior	6	150
conv10_2	3, 3	0.725	1:1, 2:1, 1:2 + an extra prior	4	36
conv11_2	1, 1	0.9	1:1, 2:1, 1:2 + an extra prior	4	4
Grand Total	–	–	–	–	8732 priors

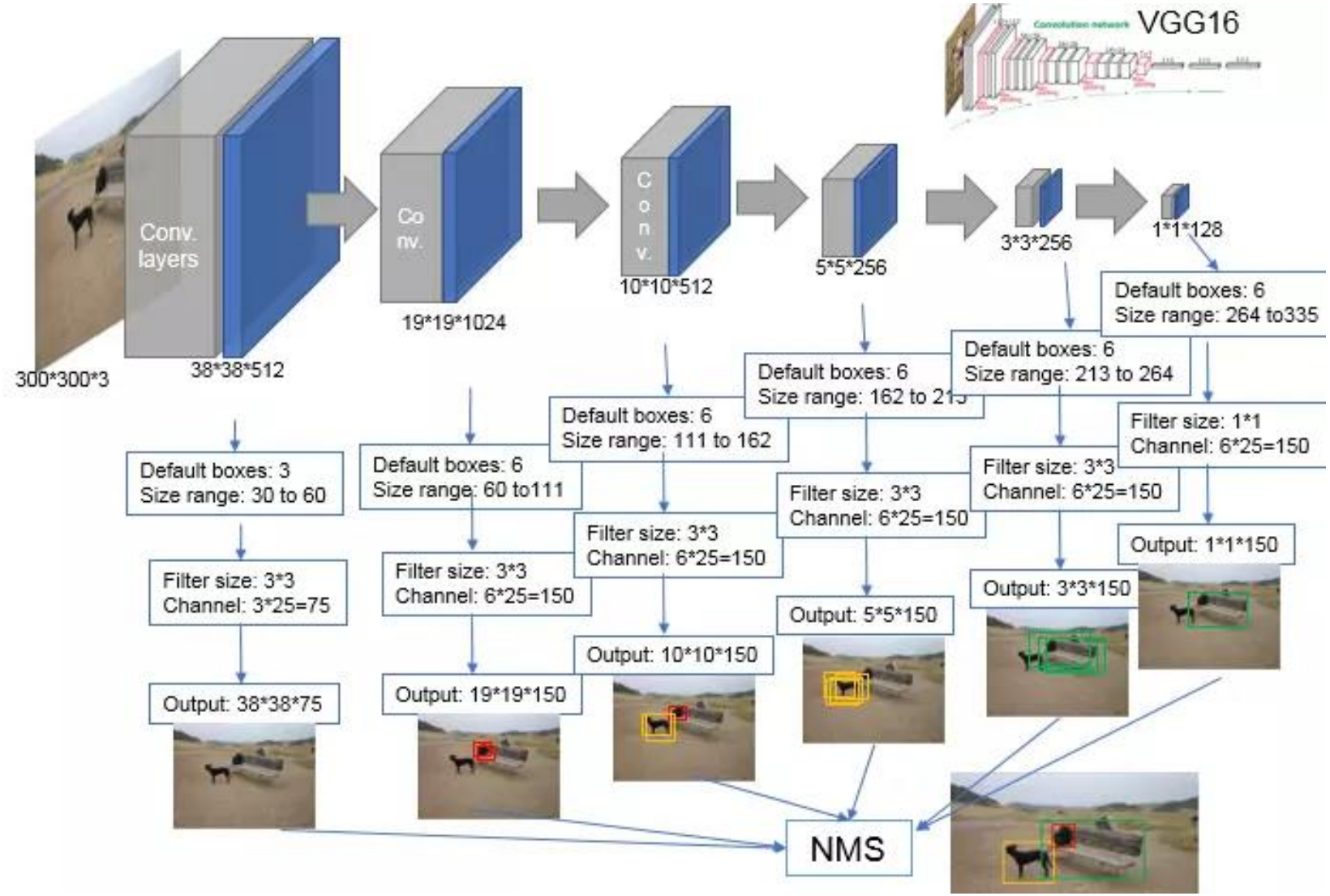
Non-Maximum Suppression

- 画像の全ての候補領域に得点が付けられている場合、ある領域に、より高い得点で選択され、学習された閾値より大きな値を持つ領域とのIoU値が重複した場合、その領域を排除するnon-maximum suppressionを（各々クラスに対して独立して）適用する。



SSDモデル全体像


300x300x3
Input Image
VGG up to conv4_3
38x38x512
VGG up to fc7
19x19x1024
Conv layers
10x10x512
Conv layers
5x5x256
Conv layers
3x3x256
Avg pooling
1x1x256
Normalization
#def.boxes: 4
#def.boxes: 6
#def.boxes: 6
#def.boxes: 6
#def.boxes: 4
#def.boxes: 4
#detections per class:
5776 (38*38*4)
2166 (19*19*6)
600 (10*10*6)
150 (5*5*6)
36 (3*3*4)
4 (1*1*4)
Detector & classifier 1
Detector & classifier 2
Detector & classifier 3
Detector & classifier 4
Detector & classifier 5
Detector & classifier 6
#detections per class over all: 8732
Fast Non-Maximum Suppression (Fast NMS)
Final detections


VGG16
Conv. layers
300*300*3
38*38*512
Co nv.
19*19*1024
C o n v.
10*10*512
5*5*256
3*3*256
1*1*128
Default boxes: 6
Size range: 264 to335
Default boxes: 6
Size range: 213 to 264
Default boxes: 6
Size range: 162 to 213
Default boxes: 6
Size range: 111 to 162
Default boxes: 6
Size range: 60 to111
Default boxes: 3
Size range: 30 to 60
Filter size: 1*1
Channel: 6*25=150
Filter size: 3*3
Channel: 6*25=150
Filter size: 3*3
Channel: 6*25=150
Filter size: 3*3
Channel: 6*25=150
Filter size: 3*3
Channel: 6*25=150
Filter size: 3*3
Channel: 3*25=75
Output: 1*1*150
Output: 3*3*150
Output: 5*5*150
Output: 10*10*150
Output: 19*19*150
Output: 38*38*75
NMS

参照リンク

- https://qiita.com/ikeyasu/items/a95448254dff958a05b5
- https://github.com/YutaroOgawa/pytorch_advanced/tree/master/2_objectdetection
- https://stats.stackexchange.com/questions/351874/how-to-interpret-smooth-l1-loss/369380