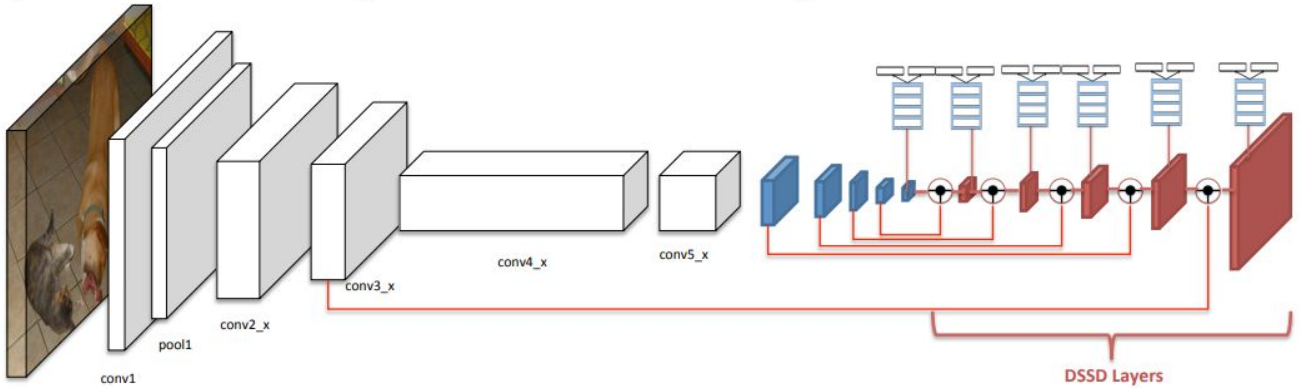
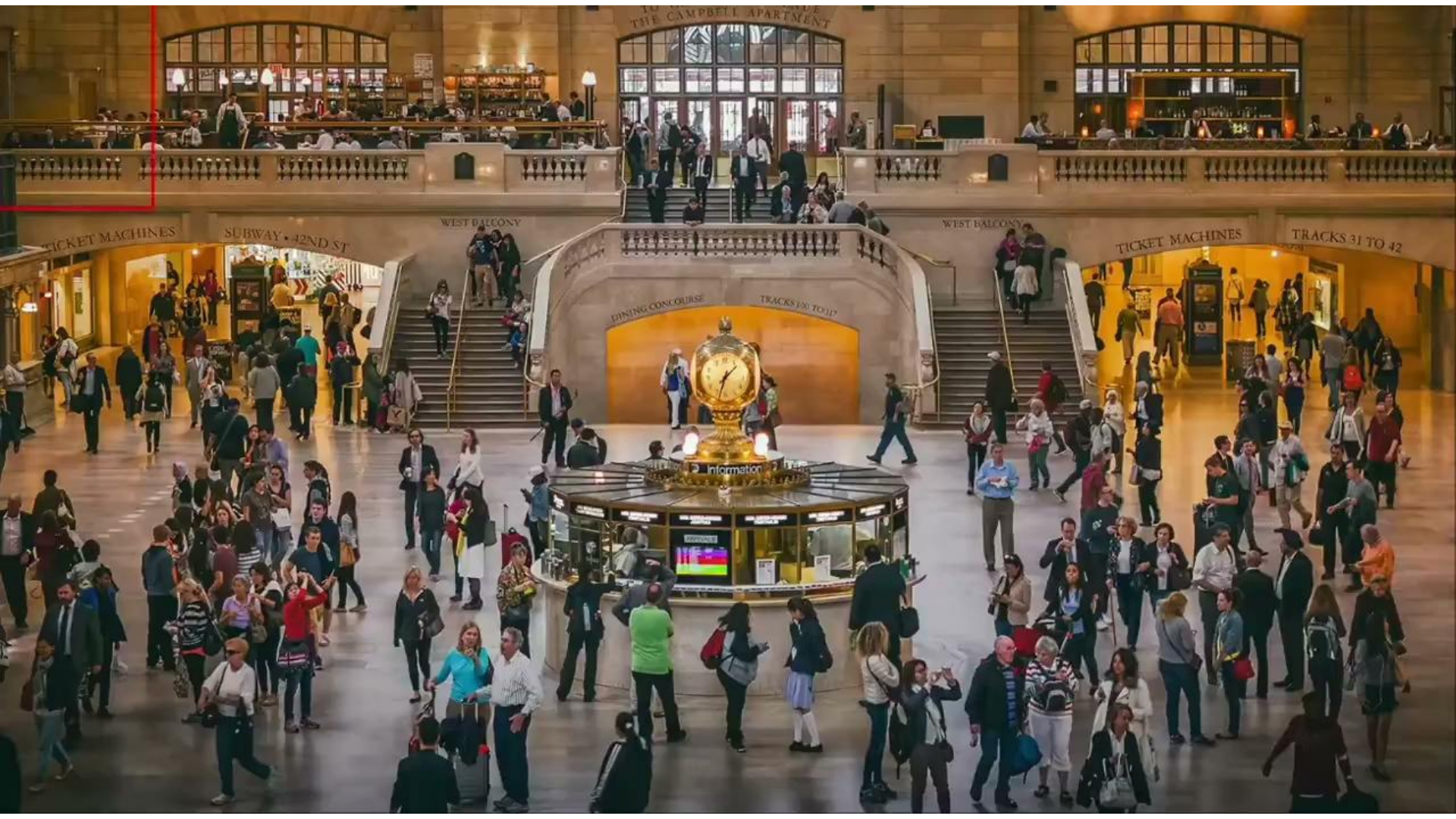
DSSD : Deconvolutional Single Shot Detector

Cheng-Yang Fu[1]*, Wei Liu[1]*, Ananth Ranga[2], Ambrish Tyagi[2], Alexander C. Berg[1]
[1]UNC Chapel Hill, [2]Amazon Inc.
{cyfu, wliu}@cs.unc.edu, {ananthr, ambrisht}@amazon.com, aberg@cs.unc.edu




THE CAMPBELL APARTMENT
TICKET MACHINES
SUBWAY • 42ND ST
WEST BALCONY
WEST BALCONY
TICKET MACHINES
TRACKS 31 TO 42
DINING CONCOURSE
TRACKS 100 TO 117
Information

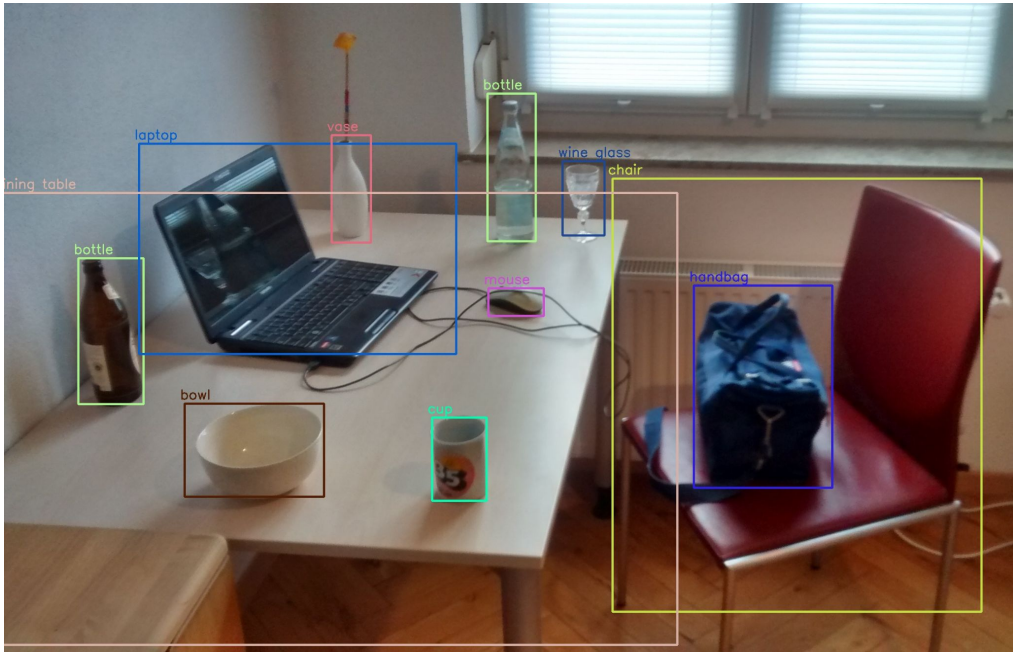
OBJECT DETECTION

Majority of object detection models including:

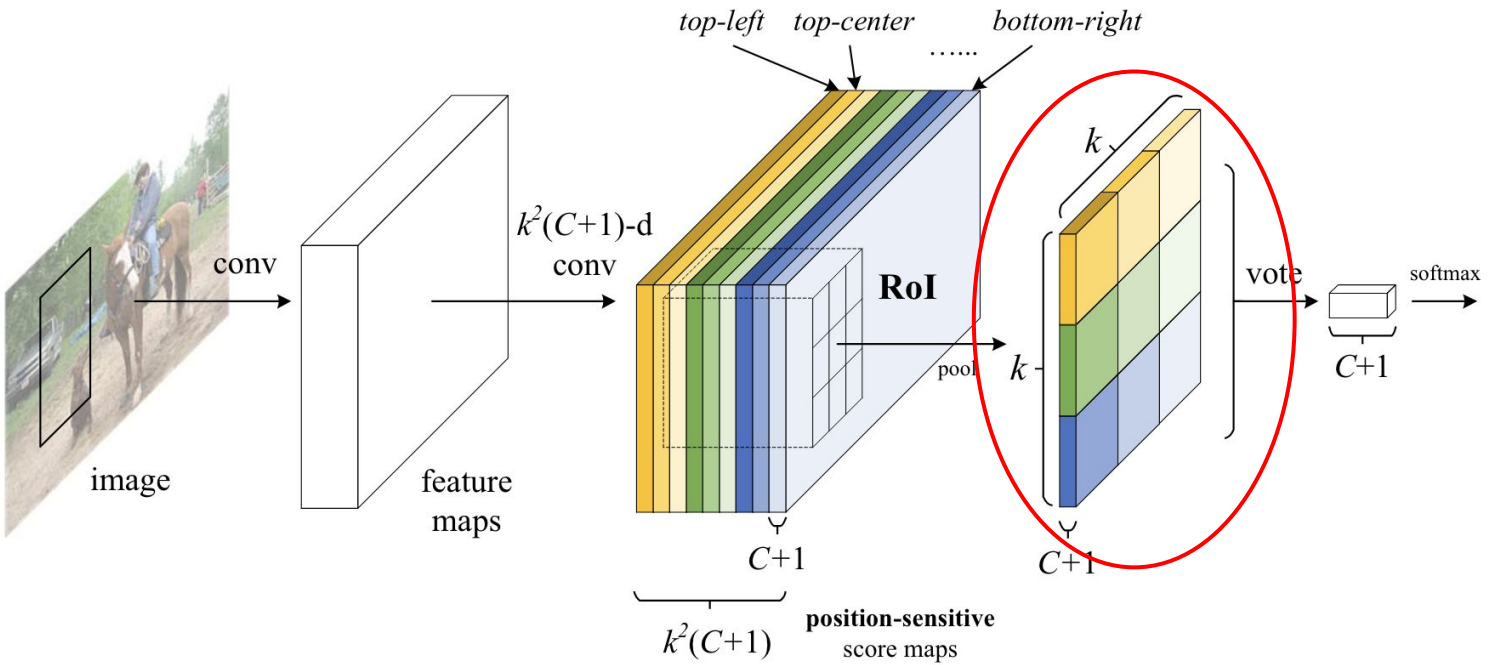
- RFCN (Region-based Fully Convolutional Network)
- Faster R-CNN

Use the **top-most layer** to model

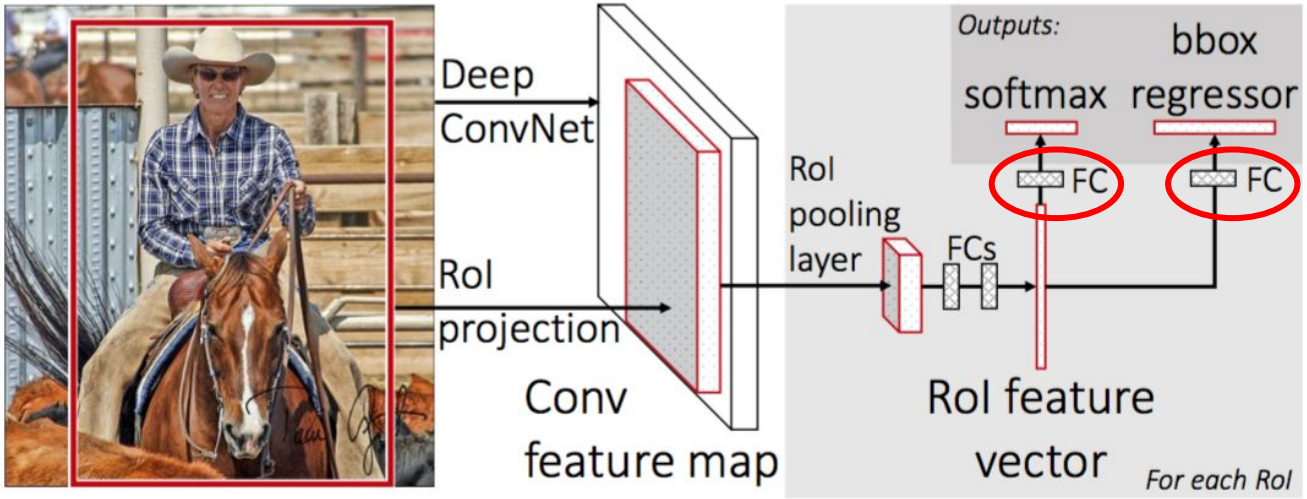
all possible object scales and shapes



RFCN


top-left
top-center
bottom-right
.....
conv
image
feature maps
$k^2(C+1)$-d conv
RoI
pool
k
k
vote
softmax
$C+1$
$C+1$
$C+1$
$k^2(C+1)$
position-sensitive score maps

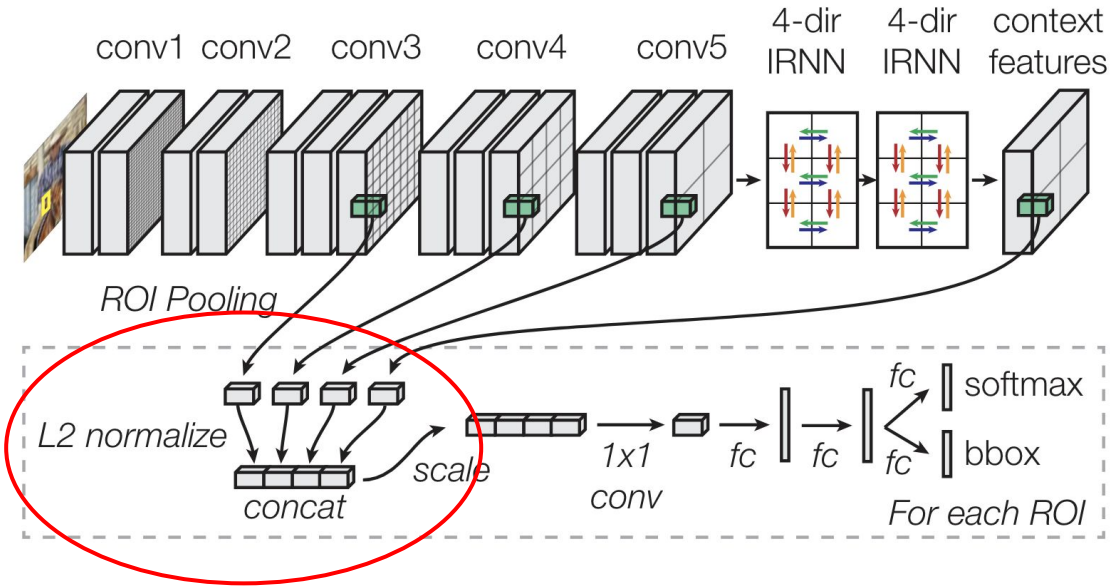
R-CNN


Deep
ConvNet
RoI
projection
Conv
feature map
RoI
pooling
layer
FCs
Outputs:
softmax
bbox
regressor
FC
FC
RoI feature
vector
For each RoI

FIRST APPROACH

Combines feature maps from different layers of a ConvNet to be used for prediction.

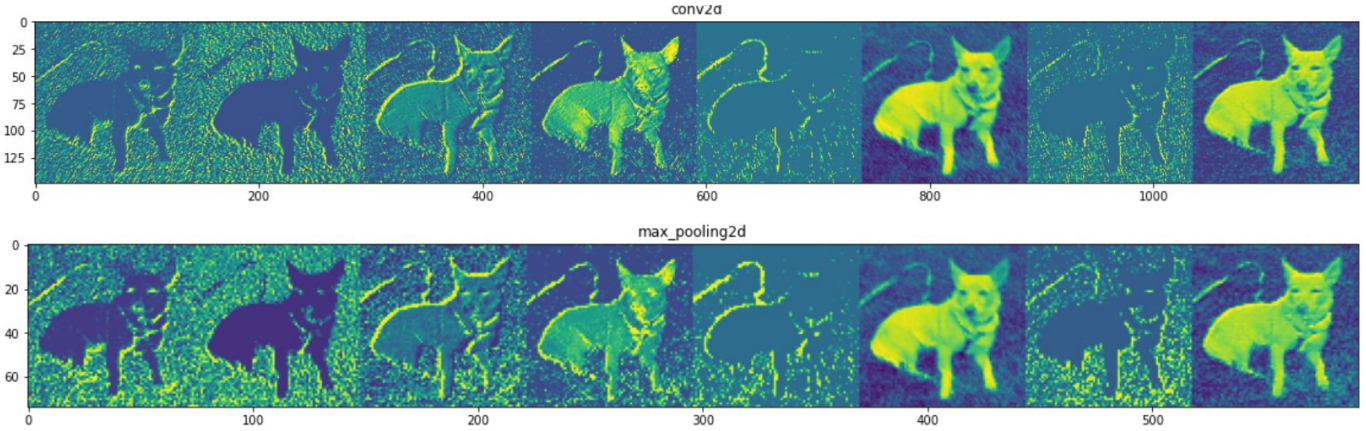
ION
(Inside-Outside- Net)



WHY DOES THIS WORK?

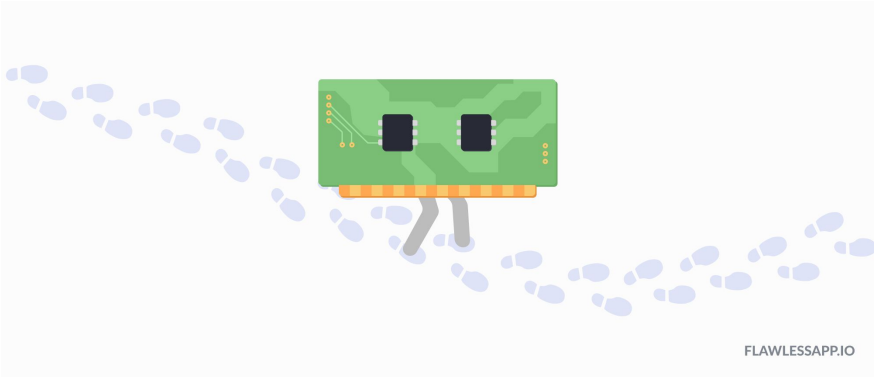
Because the combined feature maps have features from **different levels** of abstraction of the input image,

the pooled feature is **more descriptive** and is **better suitable for localization** and **classification**.



DOWNSIDES ?

Significant increase of memory footprint



Decreases the speed of the model

SECOND APPROACH

In natural images, objects can appear at very **different scales**, as illustrated by the yellow bounding boxes.

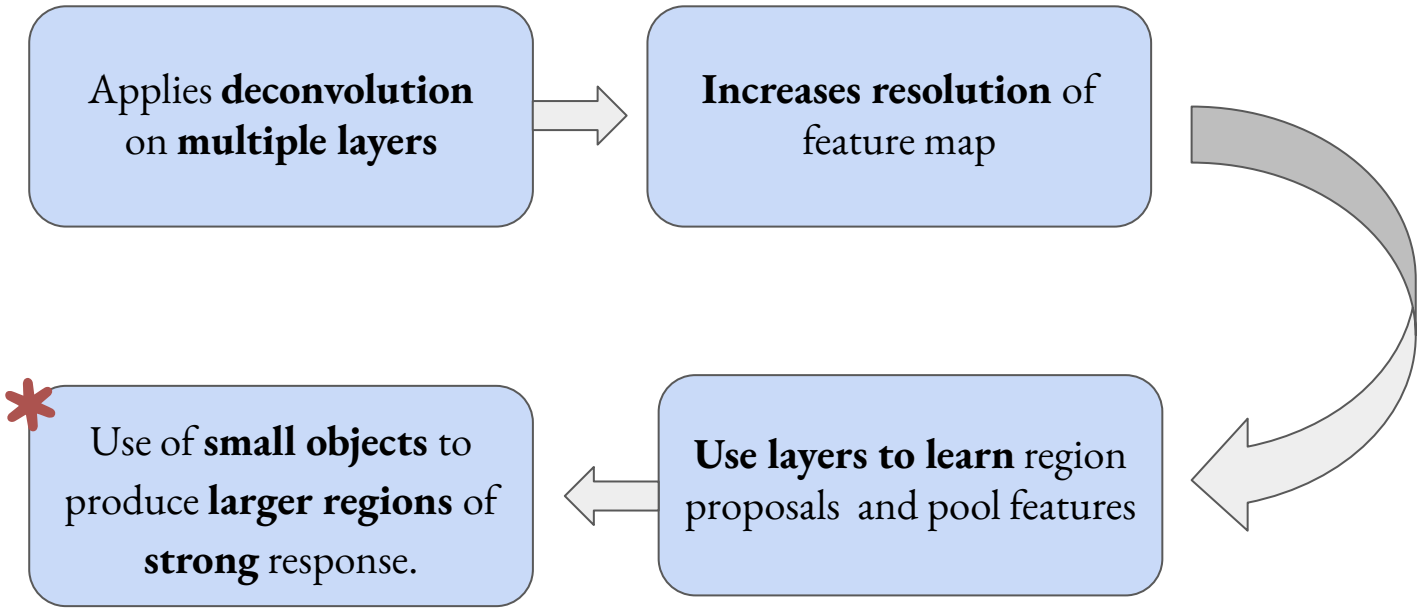
SECOND APPROACH

Use different layers within a ConvNet to predict objects of different scales.

SSD
(Single Shot Detector)

MS-CNN
(Multi-Scale CNN)

MULTI-SCALE CNN



S S D

The Single Shot Multibox Detector is compose of (2) parts:

Extract feature maps using VGG

Apply convolution filters to detect objects

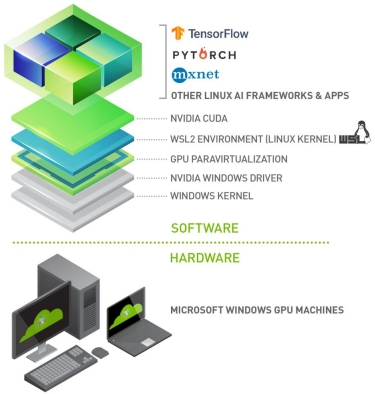
IMPLEMENTATION

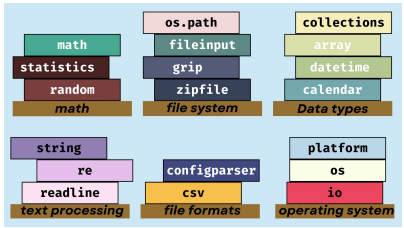
DSSD is a sophisticated algorithm framework

- Better object detection than SSD
- Technological Improvements
- High end processing power
- Requirements

IMPLEMENTATION

- Hardware and Software Requirements
  - **GPU requirements:** High-performance GPUs (ex: NVIDIA CUDA-enabled GPUs for training).
  - **Software requirements:** Python, PyTorch or TensorFlow, CUDA and cuDNN libraries.
  - **Operating system compatibility** (ex: Linux, Windows, and macOS).





Libraries

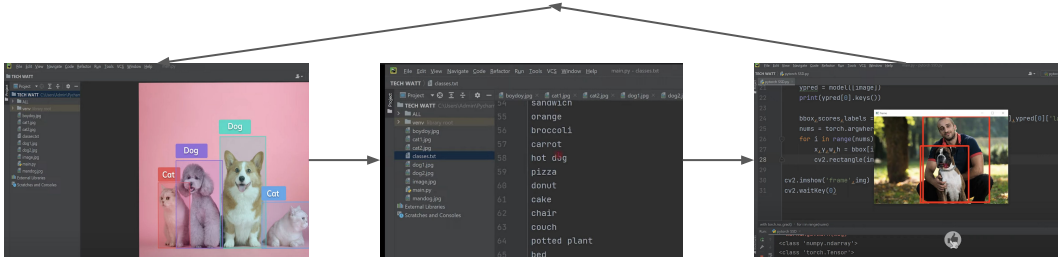
Tools

Hardware

IMPLEMENTATION

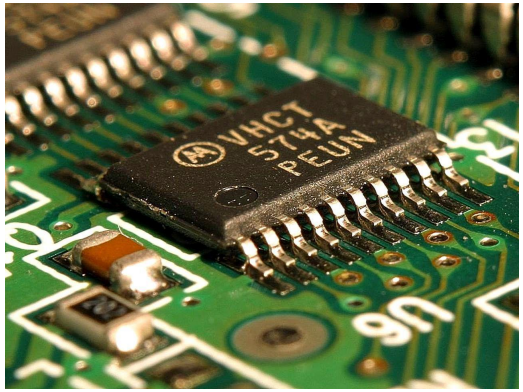
Training

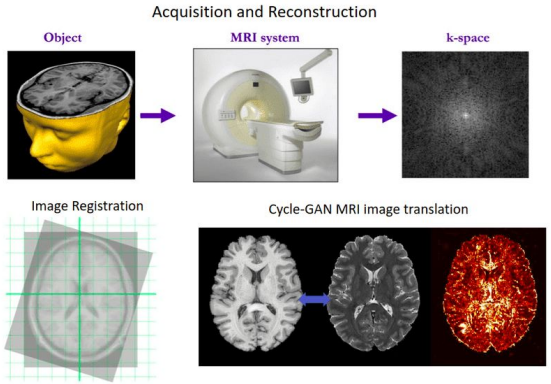
- Gather pictures/data
- Start recognizing basic objects
- Acknowledge Mistakes
- Make corrections
- Showing model more pictures/data
- Repeat so model gets better at recognizing objects



IMPLEMENTATION

The use of DSSD in scenarios where detecting small or densely packed things is supported, like in medical imaging (MRIs and X-rays) or in quality control (computer chips and parts).





IMPLEMENTATION

Use it in real life

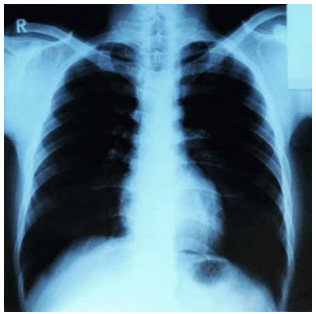
We have to **identifying areas** where we can apply DSSD our advanced object detection methods into real-world applications like ...

Automobiles

Surveillance

Manufacturing

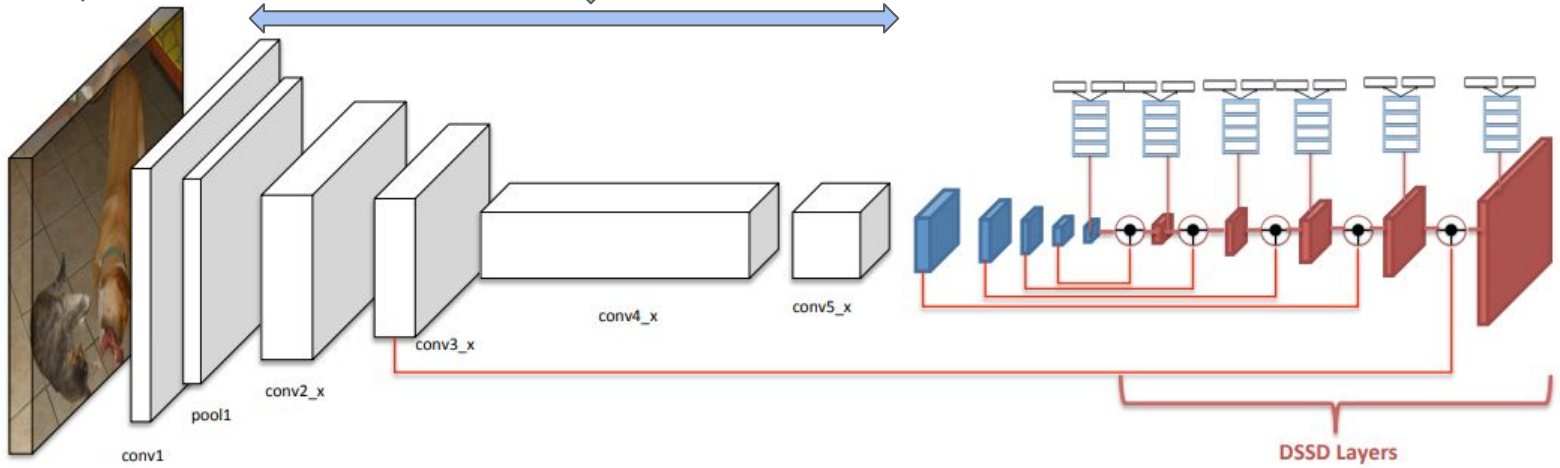
Healthcare



ALGORITHM

1. The input image is fed into the network

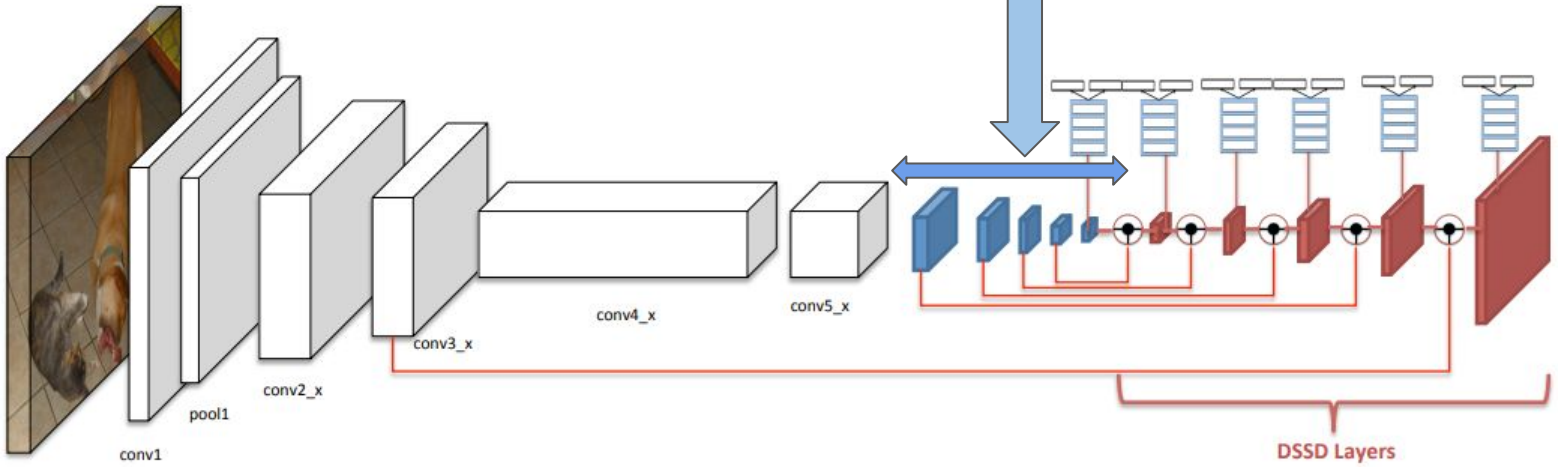
2. The input goes through a series of convolutional layers.



ALGORITHM

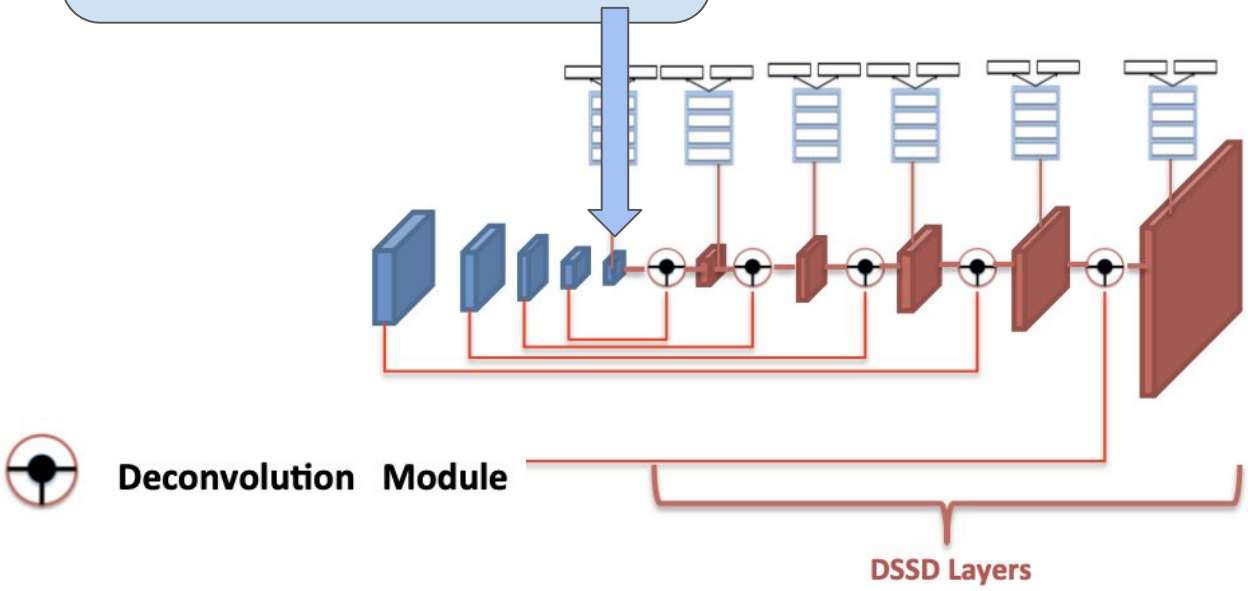
* Note that each SSD block is a feature used to produce detection outputs.

3. Then through the SSD layers.

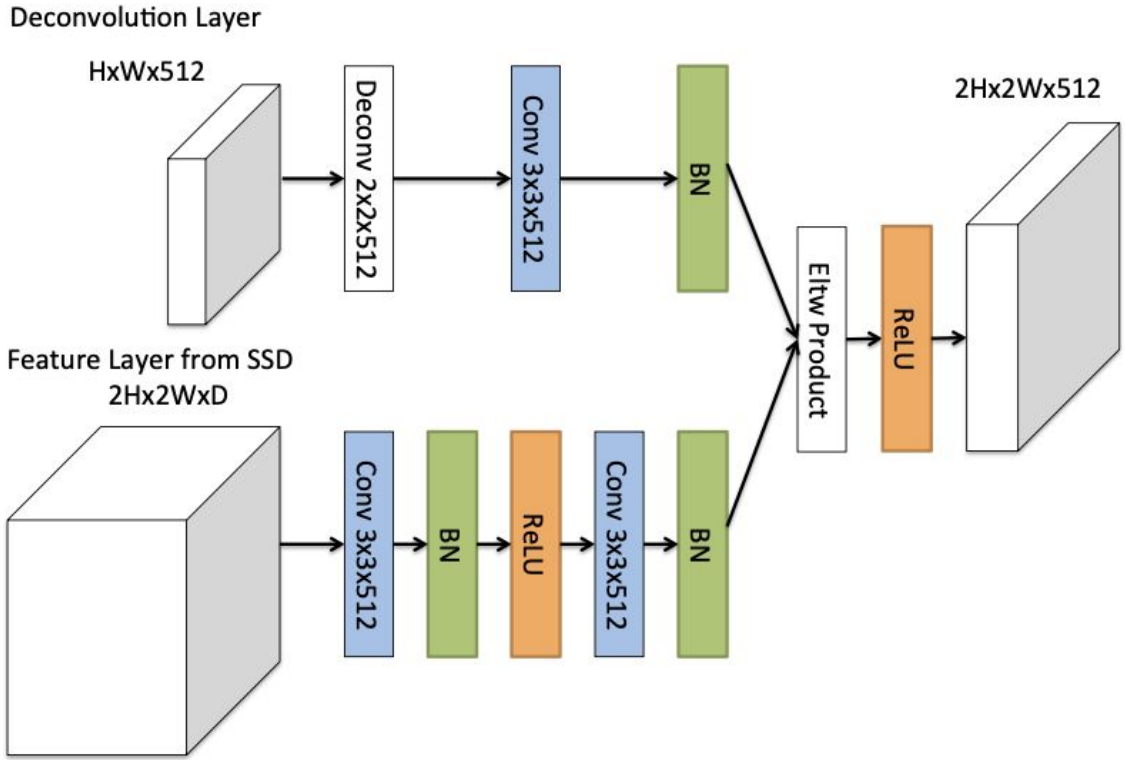


ALGORITHM

4. **Deconvolution** is applied to the first SSD feature..

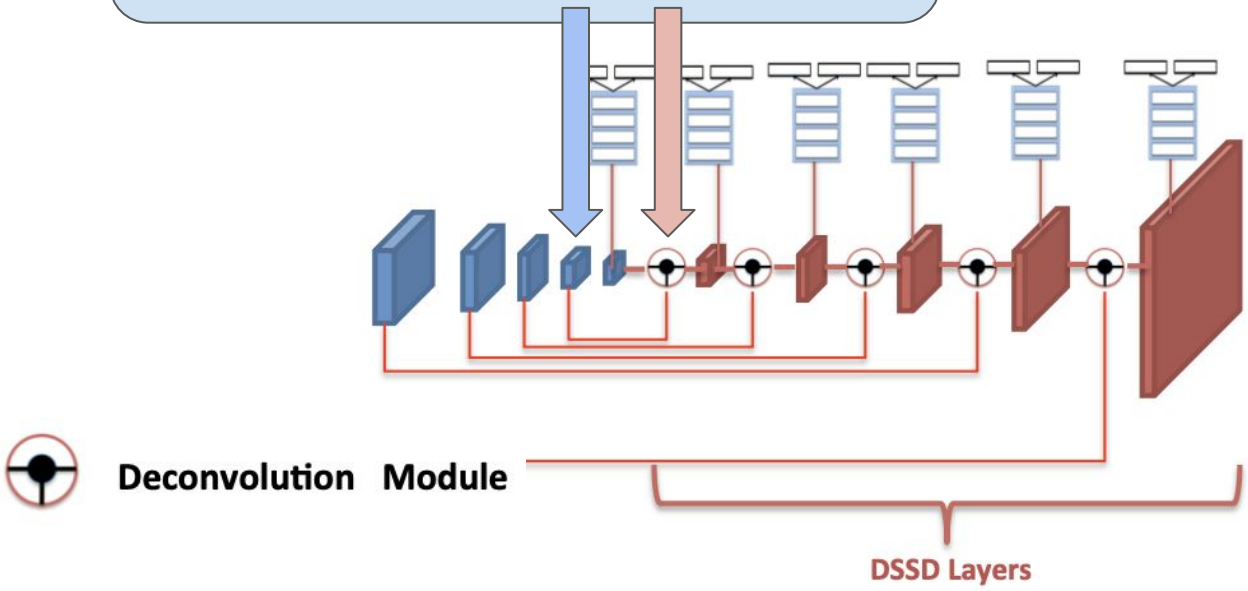


ALGORITHM


Deconvolution Layer
HxWx512
Deconv 2x2x512
Conv 3x3x512
BN
2Hx2Wx512
Feature Layer from SSD
2Hx2WxD
Conv 3x3x512
BN
ReLU
Conv 3x3x512
BN
Eltw Product
ReLU

ALGORITHM

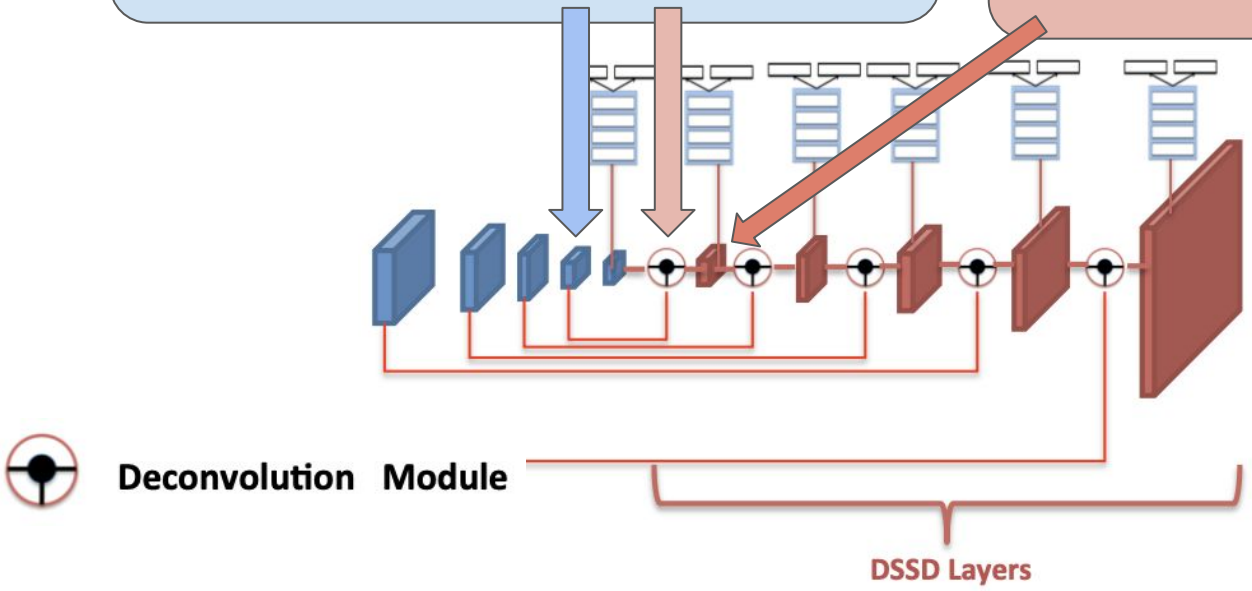
5a. **Combine** the resulting DSSD feature with the SSD corresponding in dimension



ALGORITHM

5a. **Combine** the resulting DSSD feature with the SSD corresponding in dimension

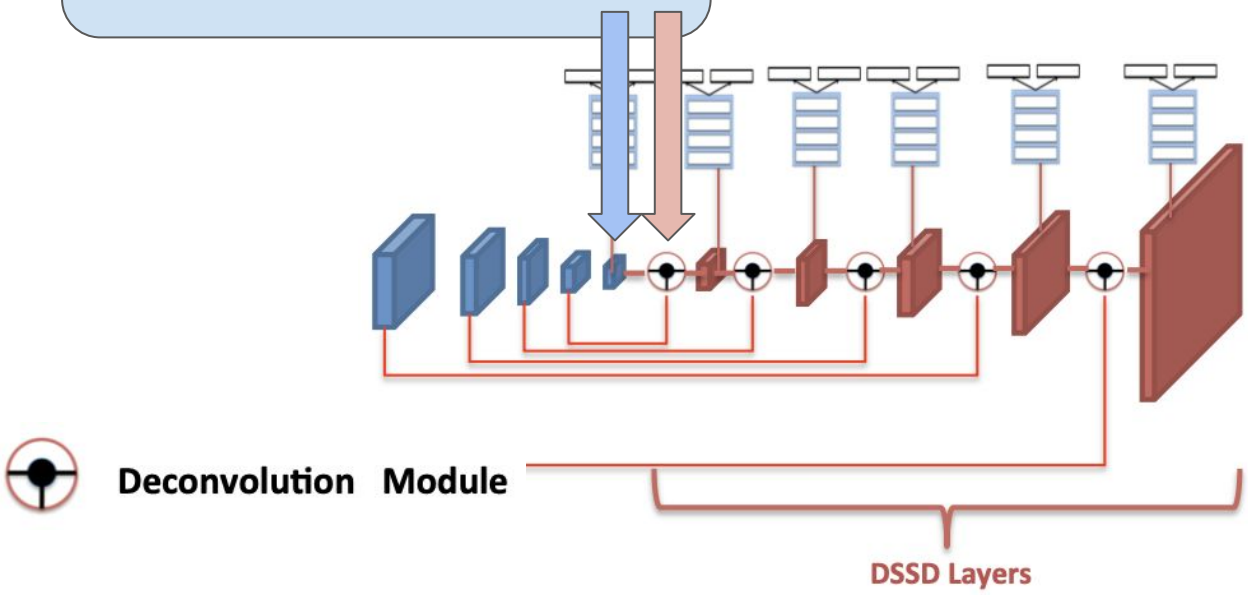
5b. This then produces our first DSSD block.



ALGORITHM

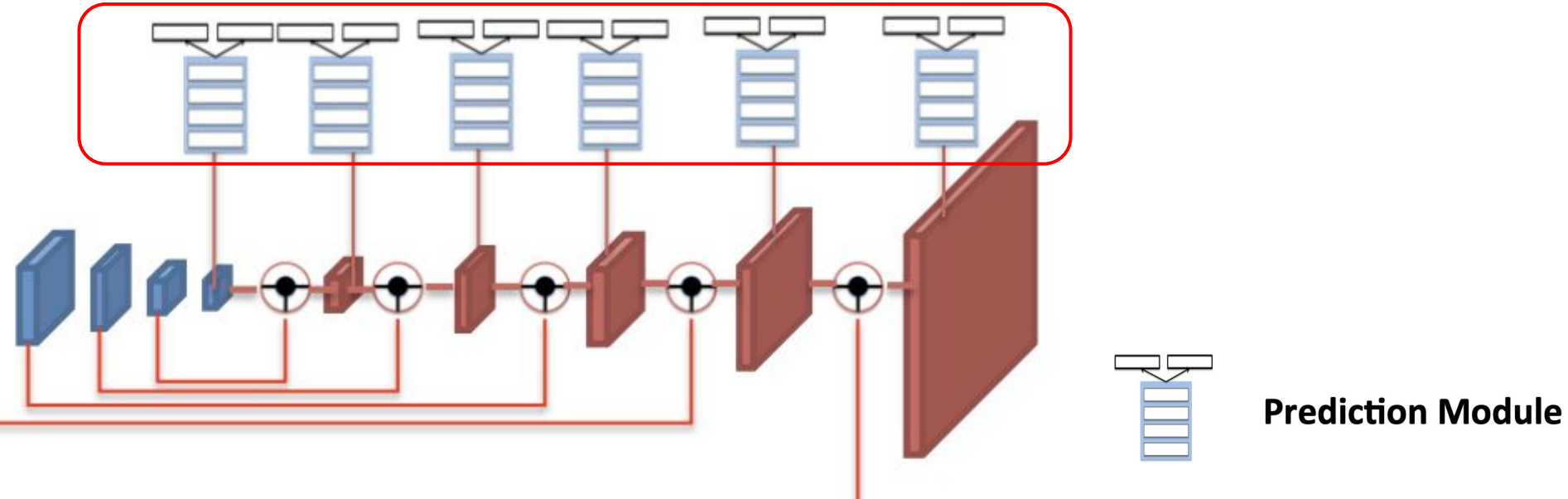
6. **Deconvolution** is applied to the last DSSD feature..

* This process is repeated, resulting in a group of DSSD features.



ALGORITHM

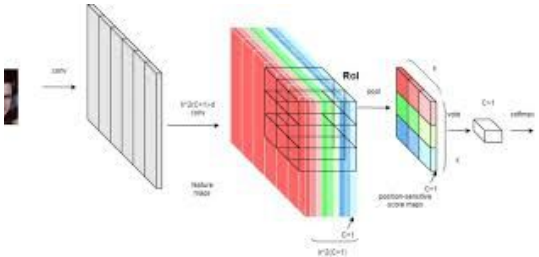
For each extracted DSSD model, a prediction model is applied to produce a bounding box and class output of an object to better improve accuracy.

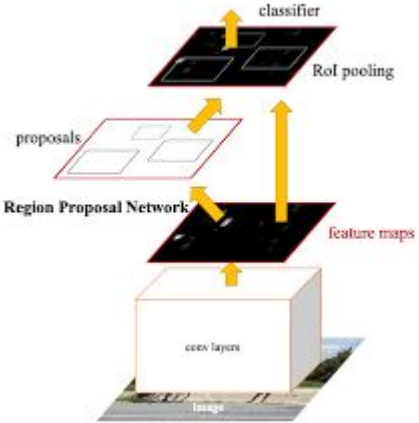


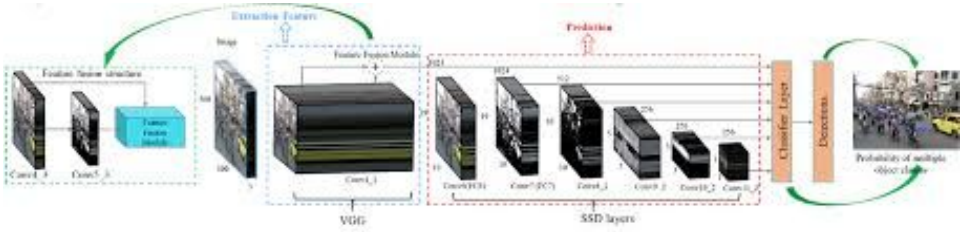
RELATIONSHIPS TO OTHER METHODS

DSSD is related to other methods such as...

- SSD (Single Shot MultiBox Detector)
- Faster R-CNN
- R-FCN





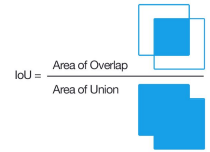


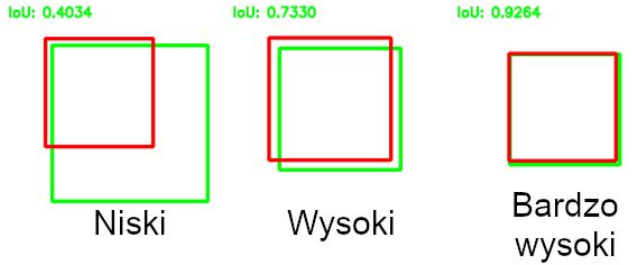
DATASETS

Method	data	network	Avg. Precision, IoU: 0.5:0.95	0.5	0.75	Avg. Precision, Area: S	M	L	Avg. Recall, #Dets: 1	10	100	Avg. Recall, Area: S	M	L
Faster [24]	trainval	VGG	21.9	42.7	-	-	-	-	-	-	-	-	-	-
ION [1]	train	VGG	23.6	43.2	23.6	6.4	24.1	38.3	23.2	32.7	33.5	10.1	37.7	53.6
Faster+++ [14]	trainval	Residual-101	34.9	55.7	-	-	-	-	-	-	-	-	-	
R-FCN [3]	trainval	Residual-101	29.9	51.9	-	10.8	32.8	45.0	-	-	-	-	-	-
SSD300* [18]	trainval35k	VGG	25.1	43.1	25.8	6.6	25.9	41.4	23.7	35.1	37.2	11.2	40.4	58.4
SSD321	trainval35k	Residual-101	28.0	45.4	29.3	6.2	28.3	49.3	25.9	37.8	39.9	11.5	43.3	64.9
DSSD321	trainval35k	Residual-101	28.0	46.1	29.2	7.4	28.1	47.6	25.5	37.1	39.4	12.7	42.0	62.6
SSD512* [18]	trainval35k	VGG	28.8	48.5	30.3	10.9	31.8	43.5	26.1	39.5	42.0	16.5	46.6	60.8
SSD513	trainval35k	Residual-101	31.2	50.4	33.3	10.2	34.5	49.8	28.3	42.1	44.4	17.6	49.2	65.8
DSSD513	trainval35k	Residual-101	33.2	53.3	35.2	13.0	35.4	51.1	28.9	43.5	46.2	21.8	49.1	66.4

Table 6: **COCO `test-dev2015` detection results.**

COCO test-dev2015 detection results - shows the performance of DSSD compared to other models on more challenging data like handling data of small objects





DATASETS

Method	network	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
Faster [24]	VGG	73.2	76.5	79.0	70.9	65.5	52.1	83.1	84.7	86.4	52.0	81.9	65.7	84.8	84.6	77.5	76.7	38.8	73.6	73.9	83.0	72.6
ION [1]	VGG	75.6	79.2	83.1	77.6	65.6	54.9	85.4	85.1	87.0	54.4	80.6	73.8	85.3	82.2	82.2	74.4	47.1	75.8	72.7	84.2	80.4
Faster [14]	Residual-101	76.4	79.8	80.7	76.2	68.3	55.9	85.1	85.3	**89.8**	56.7	87.8	69.4	88.3	88.9	80.9	78.4	41.7	78.6	79.8	85.3	72.0
MR-CNN [10]	VGG	78.2	80.3	84.1	78.5	70.8	68.5	88.0	85.9	87.8	60.3	85.2	73.7	87.2	86.5	85.0	76.4	48.5	76.3	75.5	85.0	81.0
R-FCN [3]	Residual-101	80.5	79.9	**87.2**	81.5	72.0	**69.8**	86.8	88.5	89.8	**67.0**	**88.1**	74.5	**89.8**	**90.6**	79.9	81.2	**53.7**	81.8	81.5	85.9	79.9
SSD300*[18]	VGG	77.5	79.5	83.9	76.0	69.6	50.5	87.0	85.7	88.1	60.3	81.5	77.0	86.1	87.5	83.97	79.4	52.3	77.9	79.5	87.6	76.8
SSD 321	Residual-101	77.1	76.3	84.6	79.3	64.6	47.2	85.4	84.0	88.8	60.1	82.6	76.9	86.7	87.2	85.4	79.1	50.8	77.2	**82.6**	**87.3**	76.6
DSSD 321	Residual-101	78.6	81.9	84.9	80.5	68.4	53.9	85.6	86.2	88.9	61.1	83.5	78.7	86.7	88.7	86.7	79.7	51.7	78.0	80.9	87.2	79.4
SSD512*[18]	VGG	79.5	84.8	85.1	81.5	73.0	57.8	87.8	88.3	87.4	63.5	85.4	73.2	86.2	86.7	83.9	82.5	55.6	81.7	79.0	86.6	80.0
SSD 513	Residual-101	80.6	84.3	87.6	**82.6**	71.6	59.0	88.2	88.1	89.3	64.4	85.6	76.2	88.5	88.9	87.5	83.0	53.6	83.9	82.2	87.2	81.3
DSSD 513	Residual-101	**81.5**	**86.6**	86.2	**82.6**	**74.9**	62.5	**89.0**	**88.7**	88.8	65.2	87.0	**78.7**	88.2	89.0	**87.5**	**83.7**	51.1	**86.3**	81.6	85.7	**83.7**

PASCAL VOC2007 test detection results - compares mAP scores from different models and highlights DSSD's performance with higher input resolutions.

DATASETS

Method	network	mAP	With BN layers		BN layers removed		# Proposals	GPU	Input resolution
			FPS	batch size	FPS	batch size			
Faster R-CNN [24]	VGG16	73.2	7	1	-	-	6000	Titan X	$\sim 1000 \times 600$
Faster R-CNN [14]	Residual-101	76.4	2.4	1	-	-	300	K40	$\sim 1000 \times 600$
R-FCN [3]	Residual-101	80.5	9	1	-	-	300	Titan X	$\sim 1000 \times 600$
SSD300*[18]	VGG16	77.5	46	1	-	-	8732	Titan X	300×300
SSD512*[18]	VGG16	79.5	19	1	-	-	24564	Titan X	512×512
SSD321	Residual-101	77.1	11.2	1	16.4	1	17080	Titan X	321×321
SSD321	Residual-101	77.1	18.9	15	22.1	44	17080	Titan X	321×321
DSSD321	Residual-101	78.6	9.5	1	11.8	1	17080	Titan X	321×321
DSSD321	Residual-101	78.6	13.6	12	15.3	36	17080	Titan X	321×321
SSD513	Residual-101	80.6	6.8	1	8.0	1	43688	Titan X	513×513
SSD513	Residual-101	80.6	8.7	5	11.0	16	43688	Titan X	513×513
DSSD513	Residual-101	81.5	5.5	1	6.4	1	43688	Titan X	513×513
DSSD513	Residual-101	81.5	6.6	4	6.3	12	43688	Titan X	513×513

Table 7: **Comparison of Speed & Accuracy on PASCAL VOC2007 `test`.**

PASCAL VOC2007 detection results - shows the efficiency of DSSD in practical applications comparing it with other models

DATASETS

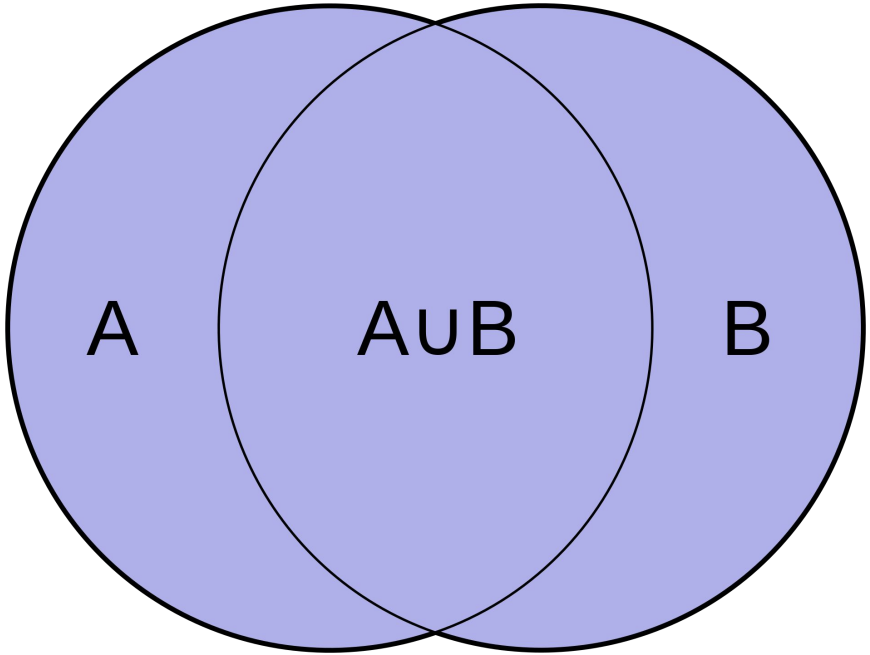
Model	mAP (VOC2007)	mAP (COCO)	FPS (PASCAL VOC)	Note on Small Objects
Faster R-CNN	73.20%	34.90%	2.4	Poor on small objects
R-FCN	80.50%	29.90%	9	Better on large objects
SSD300* (VGG)	77.50%	25.10%	46	Decent overall performance
SSD512* (VGG)	79.50%	28.80%	19	Improves on small objects
SSD321 (Residual-101)	77.10%	28.00%	11.2	Mixed results on small objects
DSSD321 (Residual-101)	78.60%	28.00%	9.5	Significant improvement on small objects
DSSD513 (Residual-101)	81.50%	33.20%	5.5	Best performance on all objects sizes

Although DSSD models have a slightly lower FPS than some older models, they offer a good balance of speed and much better accuracy.

mAP (VOC2007 and COCO) shows the accuracy of each model

FPS (PASCAL VOC) shows how many frames per second each model can process.

COMMONALITIES and DIFFERENCES
OF DSSD TO OTHER MODELS

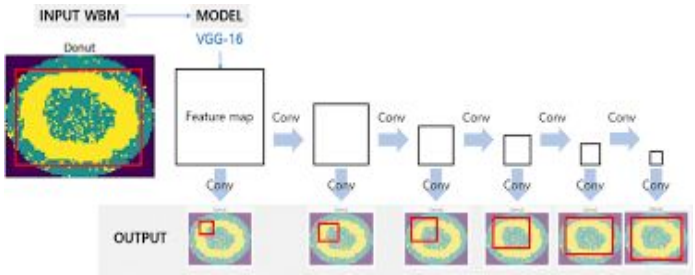


COMMONALITIES OF DSSD & SSD

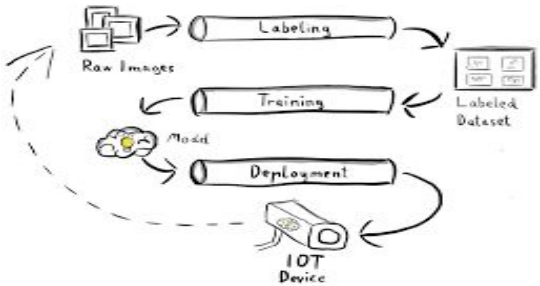
Single Shot Detectors

Efficient and Speed

Multiscale Feature Maps

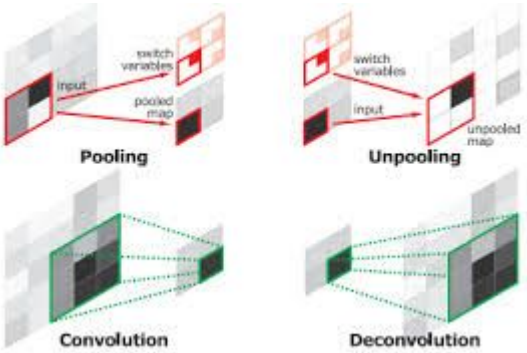


End-to-End Training



DIFFERENCES BETWEEN DSSD & SSD

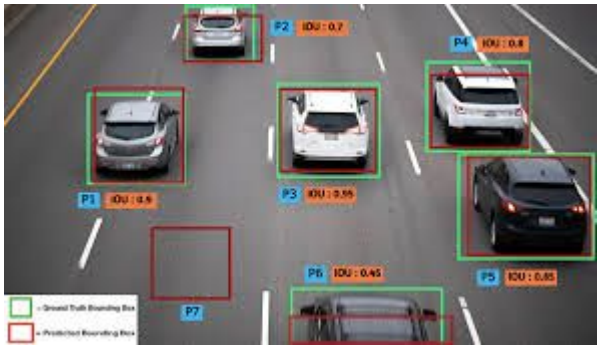
Layers



SSD: Convolutional layers

DSSD: Deconvolutional Layers

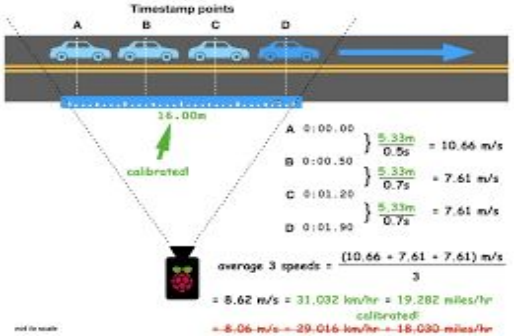
Accuracy



SSD: Very Fast but resolution loss occurs

DSSD: Addresses those weaknesses

Complexity

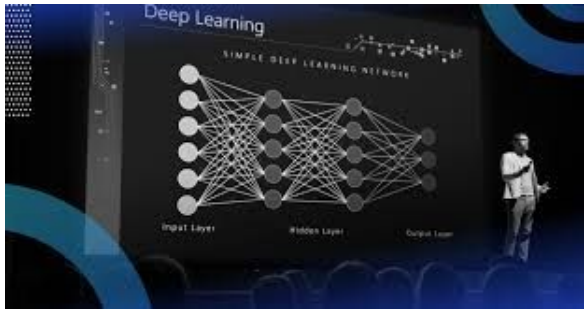


SSD: Simpler and Faster

DSSD: More Complex and slightly slower

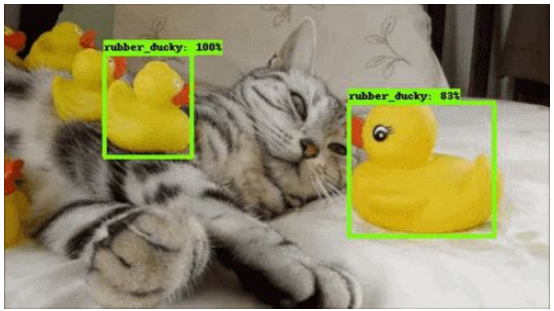
COMMONALITIES OF DSSD & FASTER R-CNN

Deep Learning



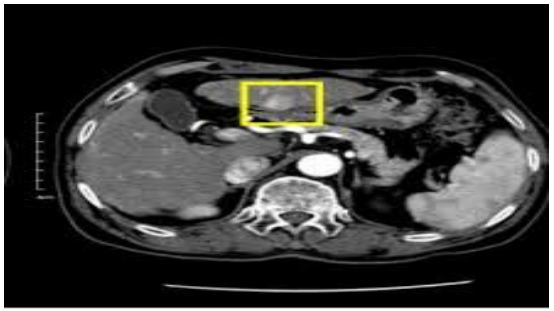
- Uses deep learning to understand and analyze images

Detection



- Recognizes objects in images in different ways

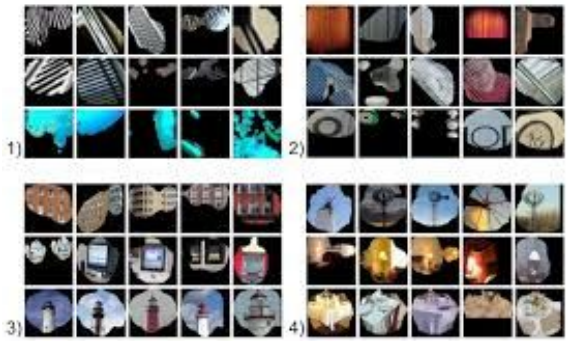
Training



- Figures out object locations and recognizing what they are

DIFFERENCES BETWEEN DSSD & FASTER R-CNN

Images



DSSD: More useful for seeing details

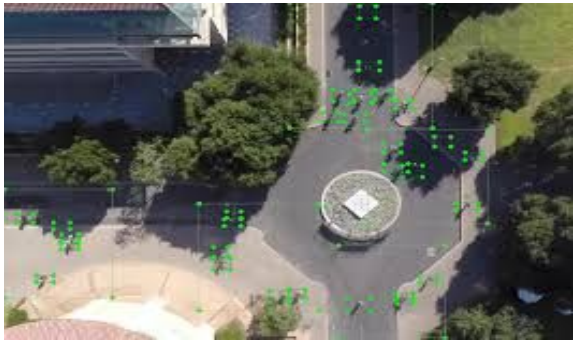
R-CNN: Focuses on accuracy, double checks guesses

Speed/Accuracy

DSSD: Better if speed is more important

R-CNN: Slower but more accurate

Strengths OD



DSSD: Better at spotting smaller objects with those extra layers

R-CNN: Better for more complex images

COMMONALITIES OF DSSD & R-FCN

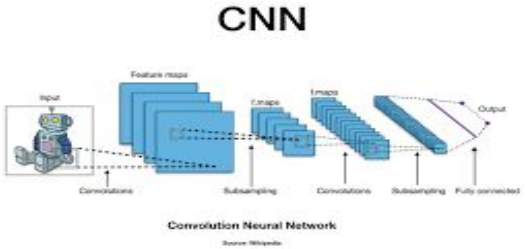
Advanced OD

- Both are high performance models
- Utilizes deep learning for images that are more complex

Improvements

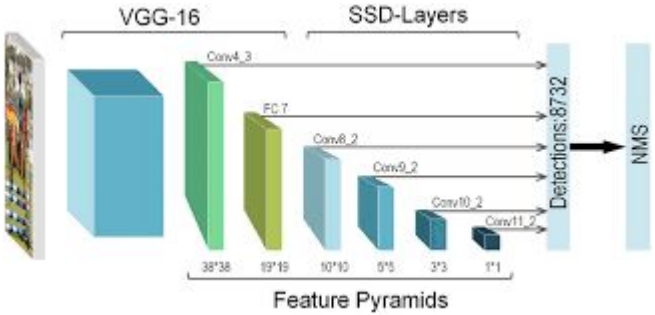
- They are designed to be faster and more efficient than earlier models like R-CNN.

Based on CNNs



- Both use (convolutional neural networks) to extract details from pictures.

DIFFERENCES BETWEEN DSSD & R-FCN



Detection Technique

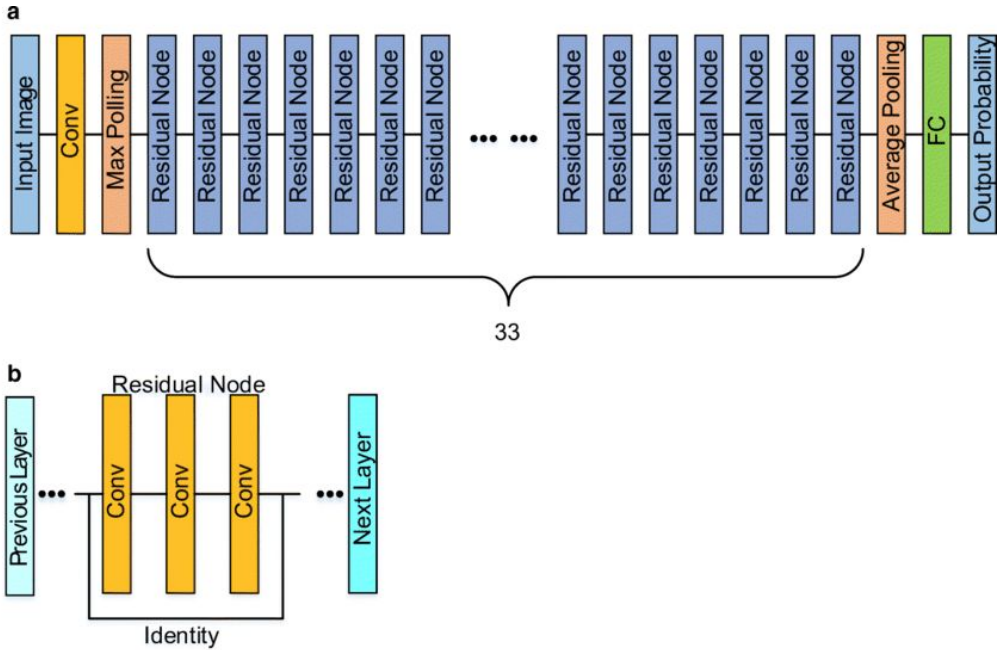
- DSSD adds layers to detect small details
- R-FCN pools parts of objects

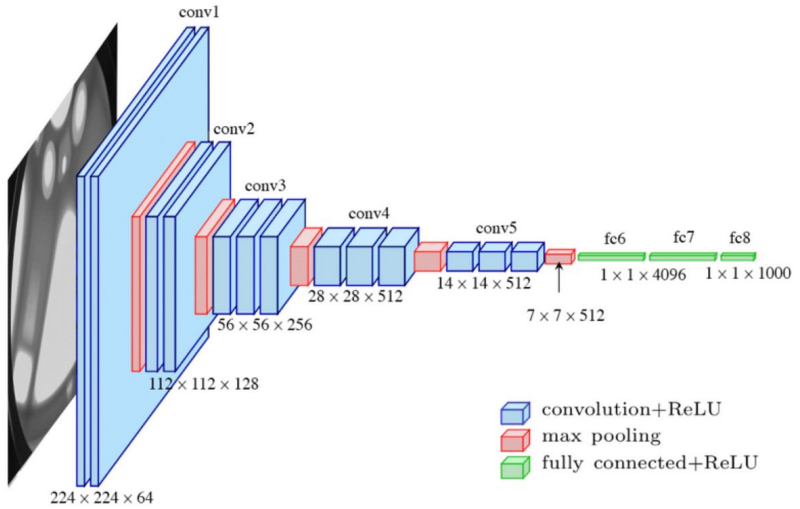
Focus

- DSSD aims to regain lost details
- R-FCN focuses on precise positioning

DSSD ARCHITECTURE

Using **Residual-101** in place of **VGG** with the goal to improve accuracy





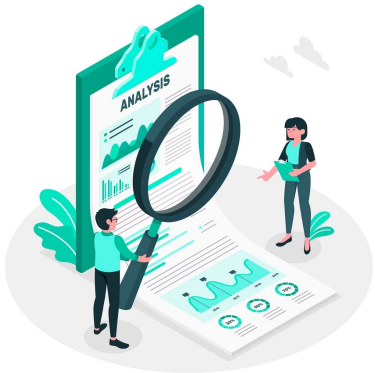
DSSD ARCHITECTURE

Method	network	mAP
Faster [24]	VGG	73.2
ION [1]	VGG	75.6
Faster [14]	Residual-101	76.4
MR-CNN [10]	VGG	78.2
R-FCN [3]	Residual-101	80.5
SSD300*[18]	VGG	77.5
SSD 321	Residual-101	77.1
DSSD 321	Residual-101	78.6
SSD512*[18]	VGG	79.5
SSD 513	Residual-101	80.6

According to the ablation study,

VGG performs better than Residual-101, however

by **adding the prediction model**, results improve



DSSD ARCHITECTURE

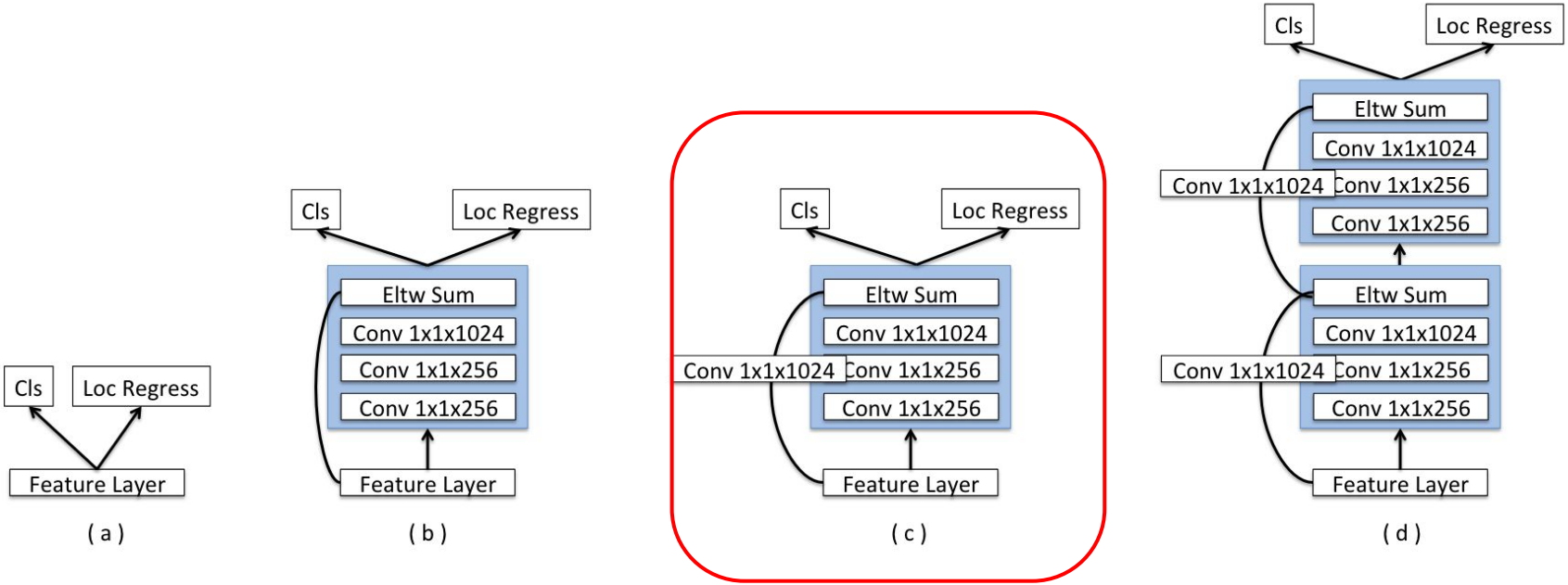
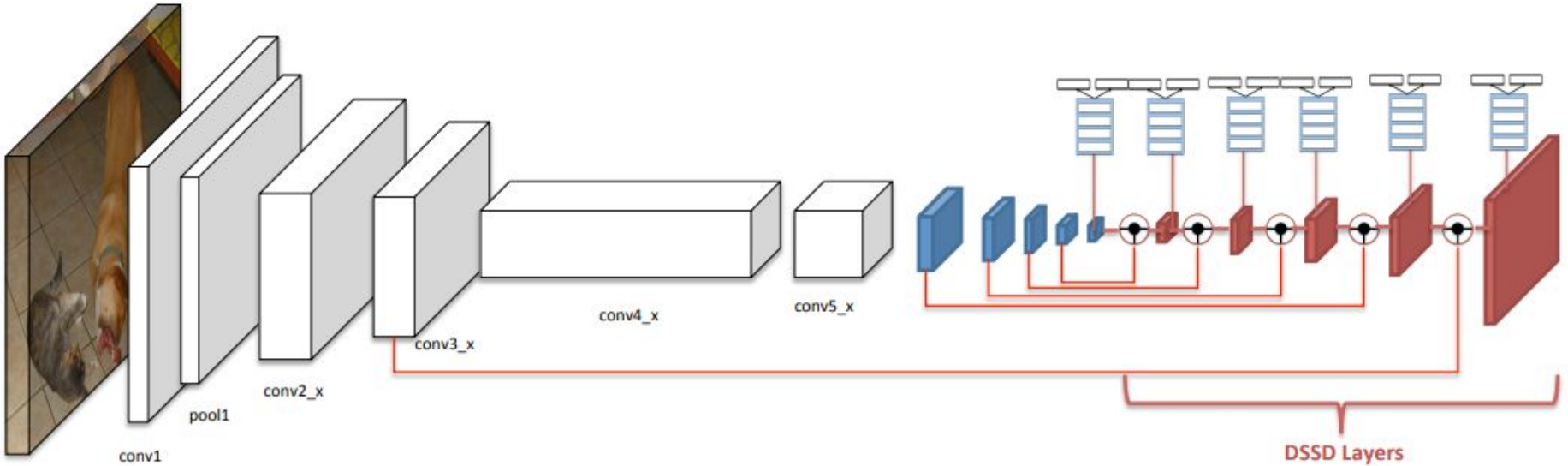


Figure 2: Variants of the prediction module

DSSD ARCHITECTURE

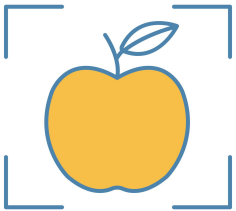
Furthermore, the authors propose to use an encoder-decoder hourglass structure to pass semantic context information before doing prediction.



STRENGTHS & WEAKNESSES

- This paper demonstrates a newer and more effective model
- There is a lot of testing with many conditions
- Provides a lot of data/tables to back up its claims

- Residual-101 requires dense feature layers to predict smaller object. Last feature layer must be used.
- No strong evidence or reason for selecting SSD framework.
- Lacked numerical evidence of improvement with the prediction module



DISCUSSION

RetinaNet model was introduced in 2017, with single-shot detection capabilities.

YOLOV7 is the latest and fastest object detection model, conducting a post processing via non-maximum suppression (NMS) to arrive at its final prediction.