CONVOLUTIONAL NEURAL NETWORK MODELS IN HUMAN AND OBJECT DETECTION

GROUP A: KRITI GUPTA, SANMESH SUHAS BHOSALE, ANKET SAH, AMALA CHIRAYIL, KSHEERAJ SAI VEPURI

CS 256

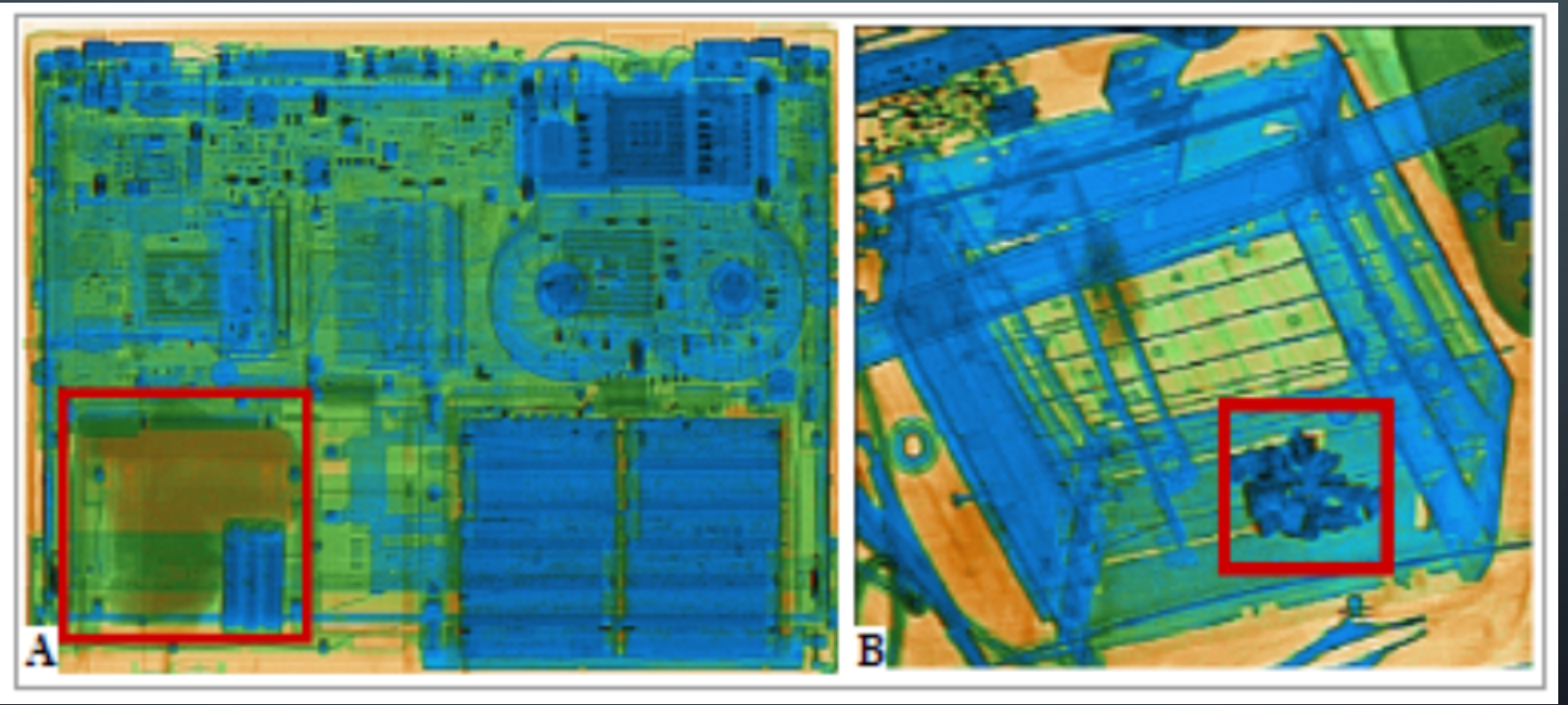
09/16/19

INTRODUCTION

- Dual Convolutional Neural Network (CNN)
- Pooling Pyramid Network (PPN)
- You Look Only Once (YOLO)
- SqueezeDet
- CenterNet

EVALUATION OF A DUAL CONVOLUTIONAL NEURAL NETWORK ARCHITECTURE FOR OBJECT-WISE ANOMALY DETECTION IN CLUTTERED X-RAY SECURITY IMAGERY

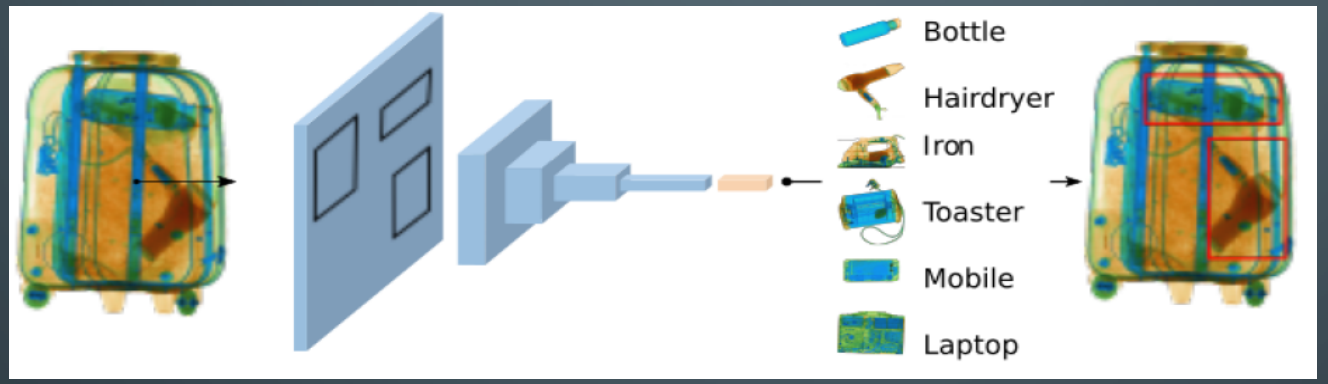
X-RAY SECURITY IMAGERY OF ELECTRONIC ITEMS



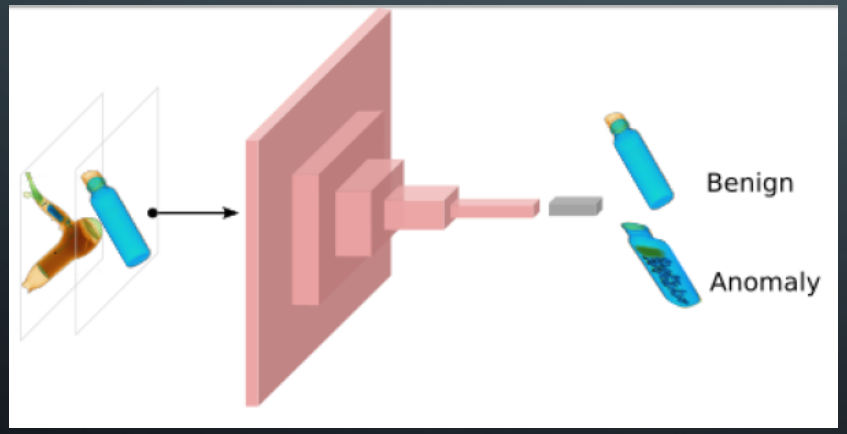
The highlighted box shows concealed anomalous region in (A) laptop and (B) toaster

TWO STAGE APPROACH:

1. Primary object detection within the X-ray image.



2. Categorization of each object in two-classes: {*anomaly, benign*}.

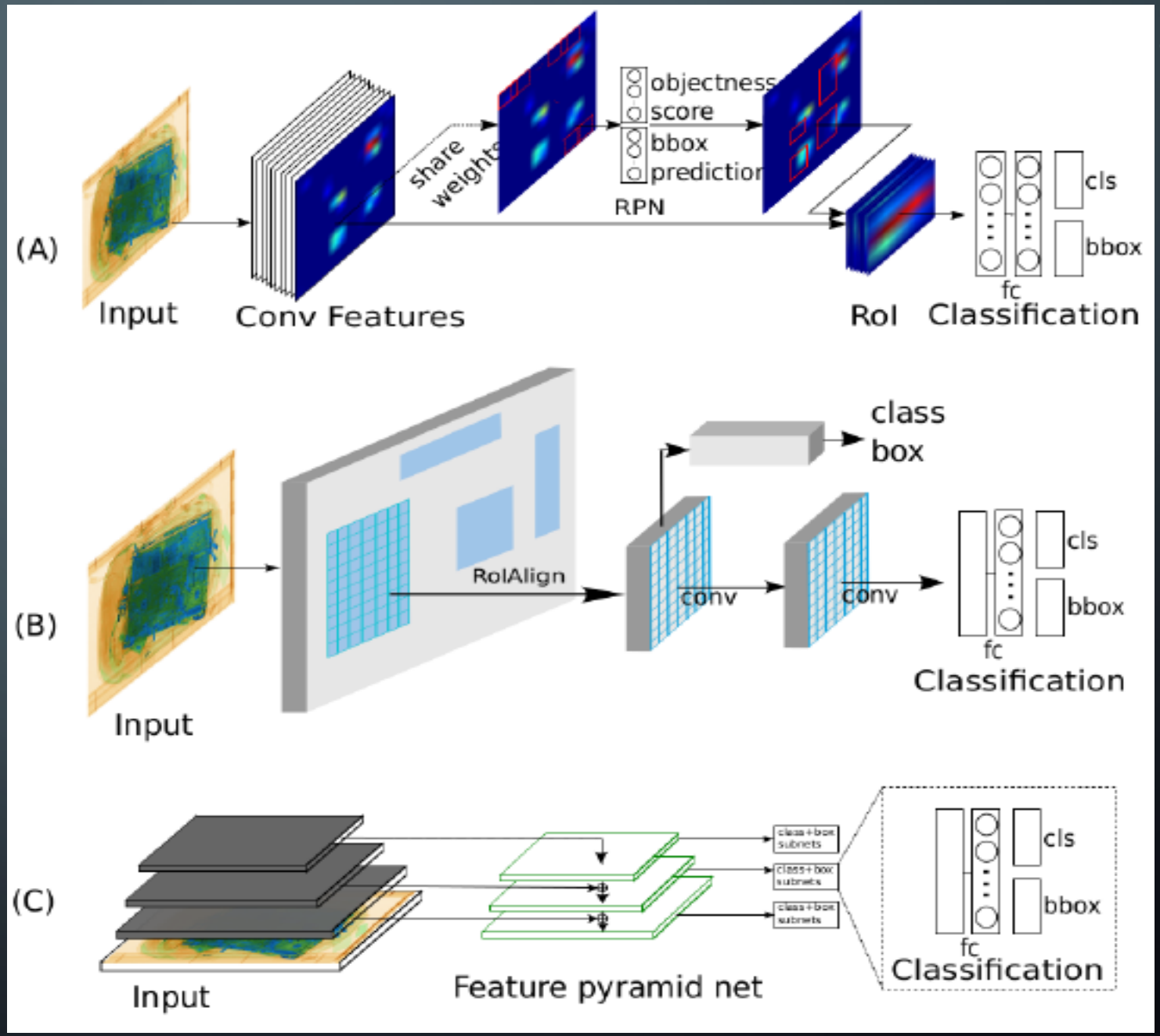


PROPOSED METHODOLOGY

A. Detection strategy
    1. Faster R-CNN
    2. Mask R-CNN
    3. RetinaNet
B. Classification Strategy
    1. SqeeneNet
    2. VGG-16
    3. ResNet

ARCHITECTURE OF CNN BASED APPROACHES

A. Faster R-CNN
B. Mask R-CNN
C. RetinaNet



EXPERIMENTAL SETUP

- Calculate area of intersection

$$\Psi(B_{gt_i}, B_{dt_i}) = \frac{Area(B_{gt_i} \cap B_{dt_i})}{Area(B_{gt_i} \cup B_{dt_i})}$$

- Calculate threshold ground truth

$$b_i = \begin{cases} 1, & \theta_{min} < \Psi(B_{gt_i}, B_{dt_i}) < \theta_{max} \\ 0, & a_i < \theta_{min}. \end{cases}$$

- Calculate true positive and false positive

$$t_i = t_{i-1} + b_i$$
$$f_i = t_{i-1} + (1 - b_i)$$

- Calculate precision and recall

$$p_i = \frac{t_i}{t_i + f_i}$$
$$r_i = \frac{t_i}{n_p}$$

- Calculate Average Precision (AP)

$$AP = \sum_{i}^{n_d} p_i \triangle r$$

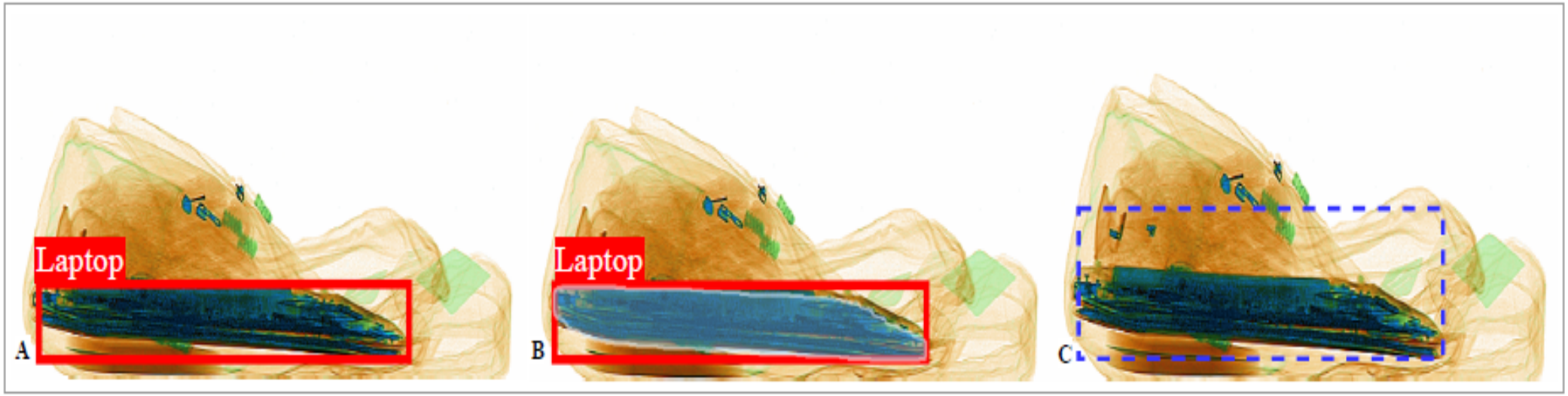
- Calculate Mean Average Precision (mAP)

$$mAP = \frac{1}{C} \sum_{c=1}^{C} AP_c$$

- For anomaly detection:
  - Accuracy (A)
  - Precision (P)
  - Recall (R), F-score (F1%)
  - True Positive (TP%)
  - False Positive (FP%)

RESULTS OF DETECTION STRATEGY

Model	Network configuration	Average precision						mAP
		Bottle	Hairdryer	Iron	Toaster	Mobile	Laptop	
Faster R-CNN [32]	$ResNet_{101}$	96.7	97.5	98.0	98.2	96.4	97.3	97.4
	$ResNet_{50}$	95.5	96.4	97.6	94.0	94.3	96.8	95.8
Mask R-CNN [14]	$ResNet_{101}$	**99.4**	92.2	**100.0**	**100.0**	96.5	**99.6**	**97.9**
	$ResNet_{50}$	97.8	90.8	99.9	99.6	95.5	98.3	96.9
RetinaNet [33]	$ResNet_{101}$	95.2	**99.2**	98.6	98.5	**97.7**	86.7	95.9
	$ResNet_{50}$	98.3	98.6	98.9	96.7	87.5	88.5	94.8



A. Faster R-CNN B. Mask R-CNN C. RetinaNet

RESULTS FOR CLASSIFICATION STRATEGY

Object Detection	Model	Network configuration	A	P	R	F1	TP(%)	FP(%)
Dual CNN (pre-localization)	Classification via CNN	$ResNet_{18}$	**0.66**	0.67	0.58	0.30	58.11	26.56
		$ResNet_{50}$	0.66	0.67	0.59	0.63	59.25	27.67
		SqueezeNet	0.59	0.57	**0.77**	0.57	**76.86**	57.16
		VGG-16	0.59	**0.74**	0.75	0.56	74.51	55.31
	Classification via Fine-Grained	VGG-16	0.64	0.62	0.70	**0.66**	70.00	58.00
Full Image (no localization)	Classification via CNN	$ResNet_{18}$	0.57	0.57	0.58	0.50	58.19	43.42
		$ResNet_{50}$	0.59	0.58	0.61	0.58	61.24	42.81
		SqueezeNet	0.58	0.72	0.27	0.53	26.76	**10.36**
		VGG-16	0.52	0.53	0.23	0.43	22.86	19.08

CONCLUSION

- This paper evaluates the effectiveness of dual CNN architecture for anomaly detection in the multiple-class item, {bottle, hairdryer, iron, toaster, mobile, laptop} in cluttered X-ray security imagery and classifying then as anomaly or benign.
- Experimentation demonstrates that fine-tuning of Mask R-CNN with ResNet101 for X-ray imagery yields 97.9% mAP for the first stage of object detection.
- However, experimental results on secondary anomaly detection, shows the benefits of a dual CNN architecture (TP: 76.86% Accuracy: 66%) false positive detection remains a significant issue (FP >=10%).

POOLING PYRAMID NETWORK (PPN) FOR OBJECT DETECTION

SINGLE SHOT MULTIBOX DETECTOR (SSD)

- Faster than RCNN.
- Uses a single DNN.
- Fast, simple, portable.

ARCHITECTURE

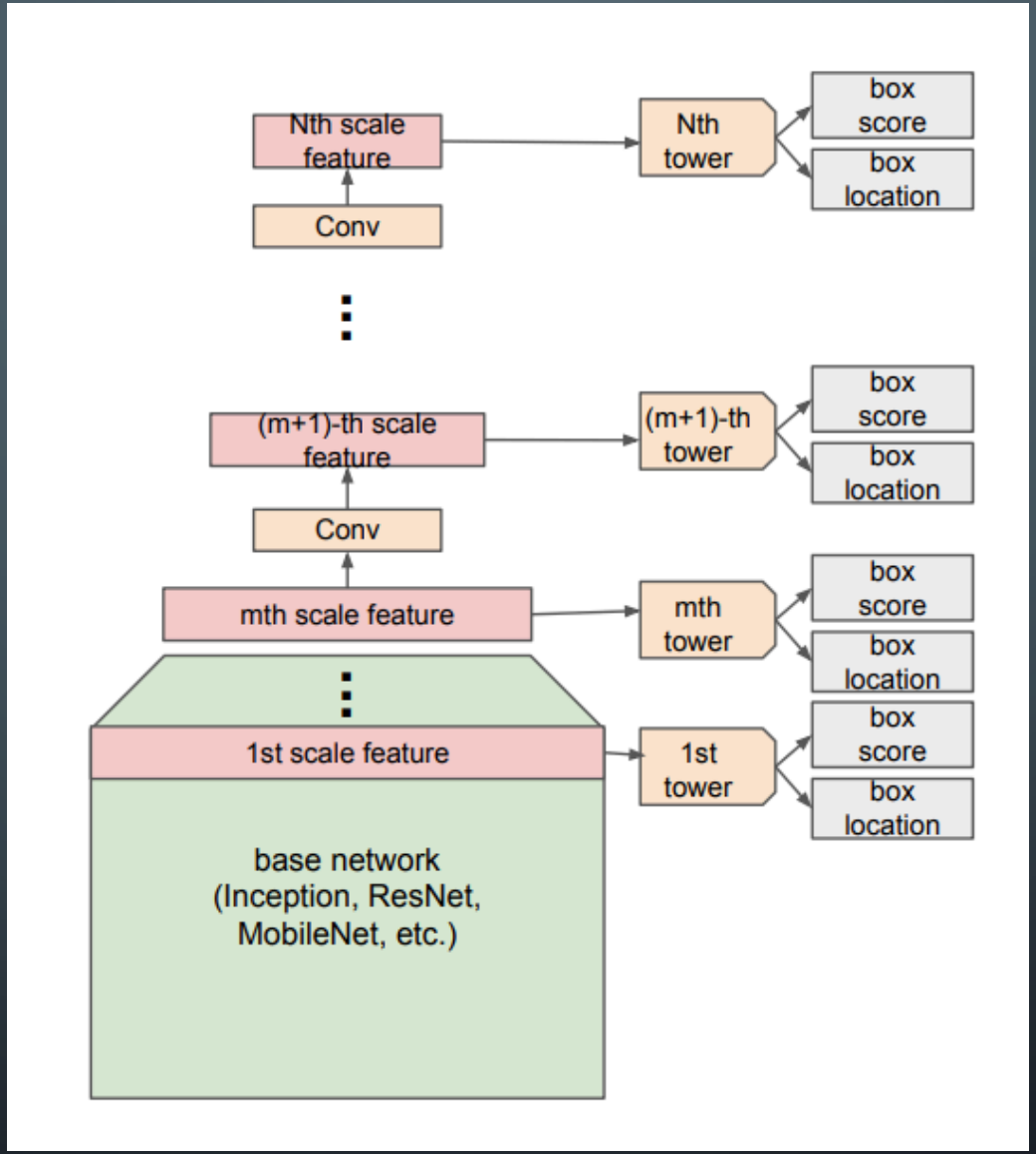


Figure 1: SSD Architecture

DRAWBACKS OF SSD

- Each box predictor trains independently.
- Miscalibration of prediction scores.
- Scores fall in vastly different ranges.

POOLING PYRAMID NETWORK (PPN)

- Single box predictor is used.
- Convolutions replaced with max pooling operations.

COMPARING THE ARCHITECTURES

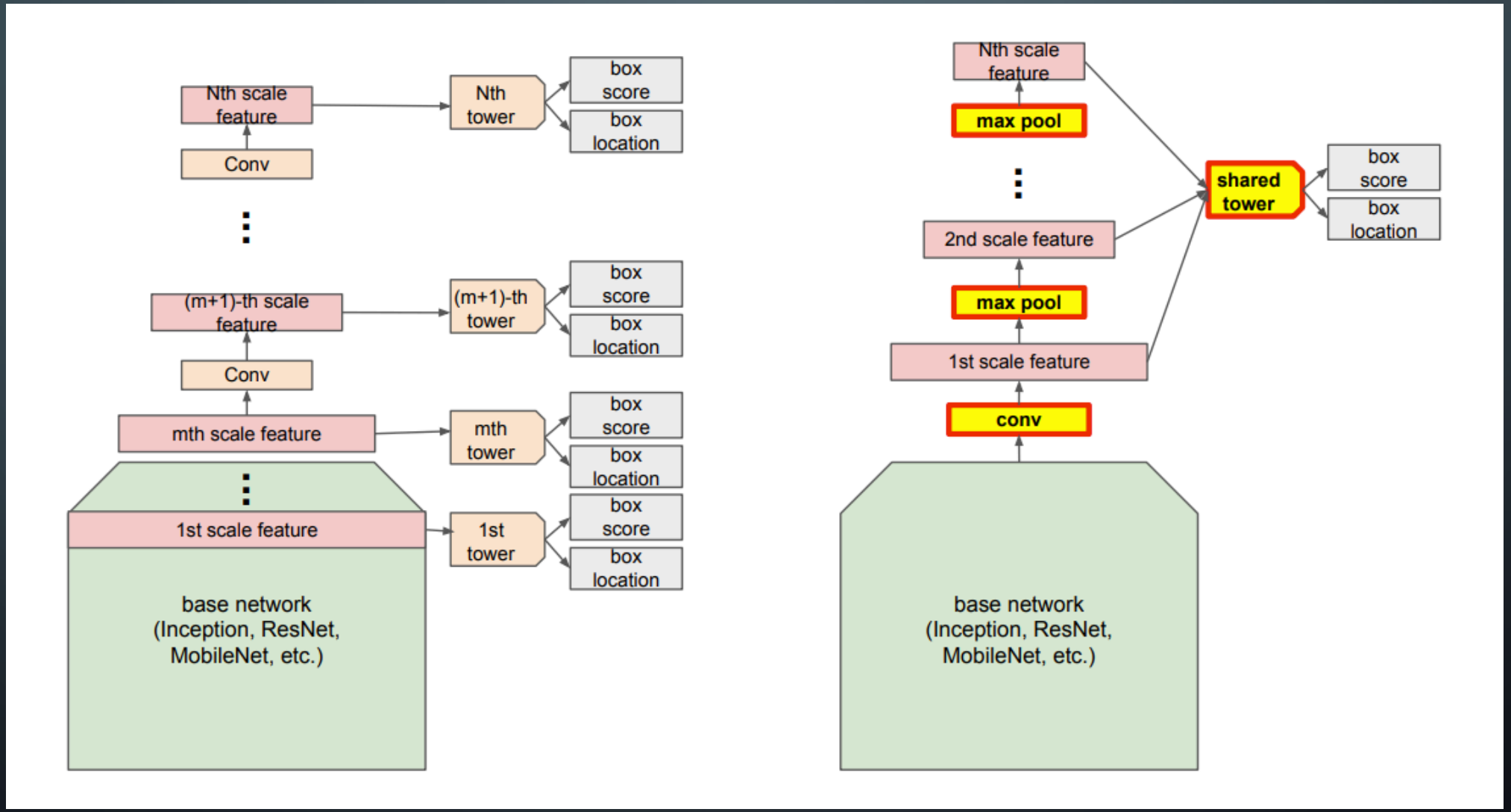


Figure 2: Left – SSD Architecture, Right – PPN Architecture

ADVANTAGES OF PPN

- Single box predictor trains on all the data.
- Miscalibration and unstable prediction scores are reduced.
- Very fast to compute during inference.

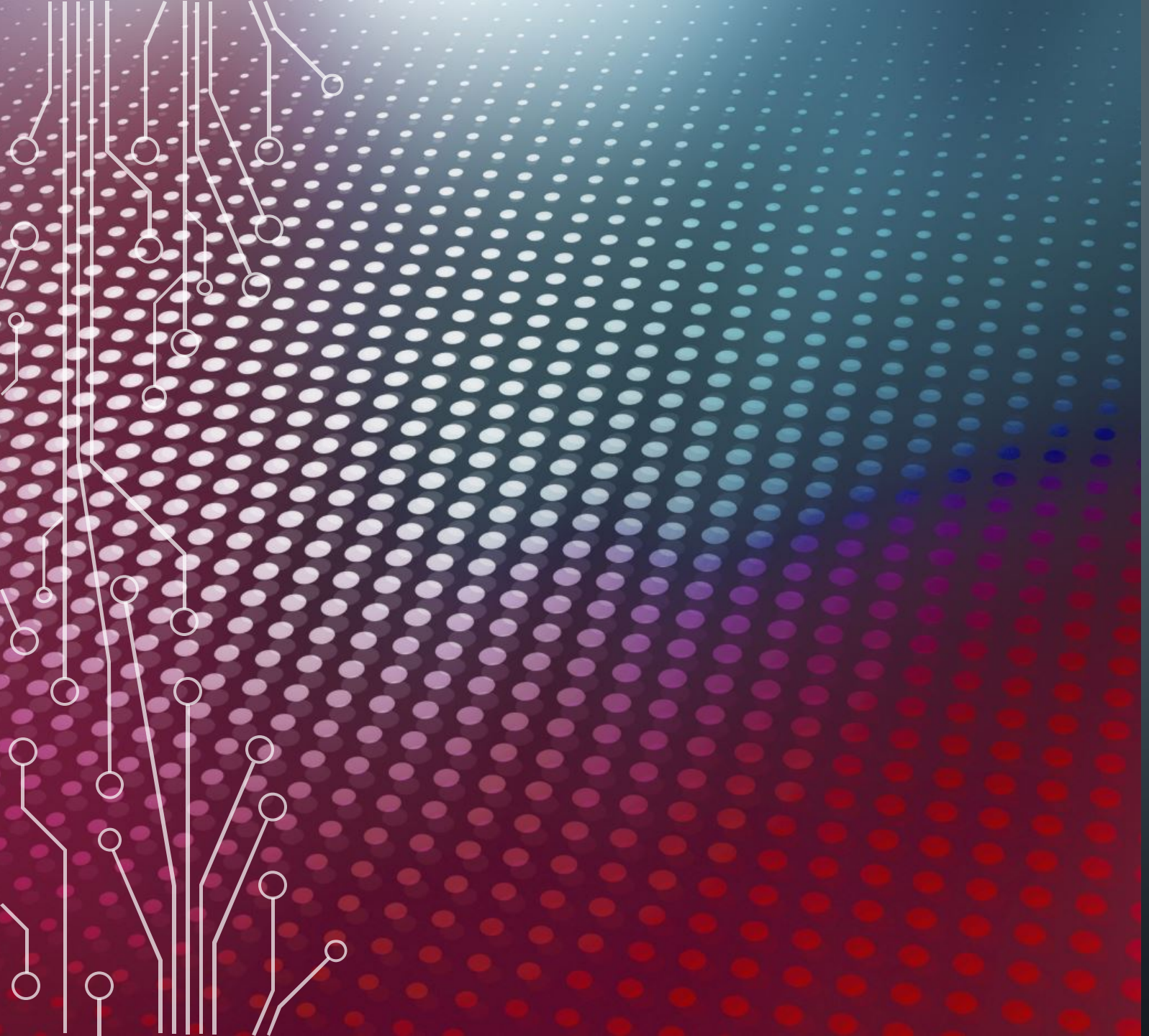
EXPERIMENTAL RESULTS

Model	mAP	inference FLOPs	number of parameters	GPU inference time
MobileNet SSD	20.8	2.48B	6.83M	27ms
MobileNet PPN	20.3	2.35B	2.18M	26ms

Figure 3: COCO detection – SSD vs PPN

CONCLUSION

- PPN achieves similar mAP (20.3 vs 20.8), comparable FLOPs and inference time while being 3x smaller in model size than SSD.

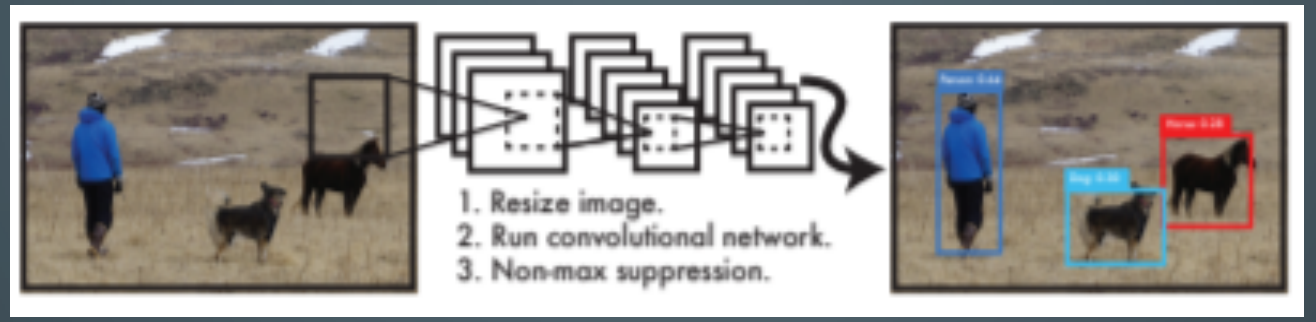


YOU ONLY LOOK ONCE: REAL-TIME OBJECT DETECTION

INTRODUCTION

- YOLO frames object detection as a regression model
- YOLO sees the entire image during training and test time so it implicitly encodes contextual information about classes as well as their appearance
- It runs neural network on a new image at test time to predict detections

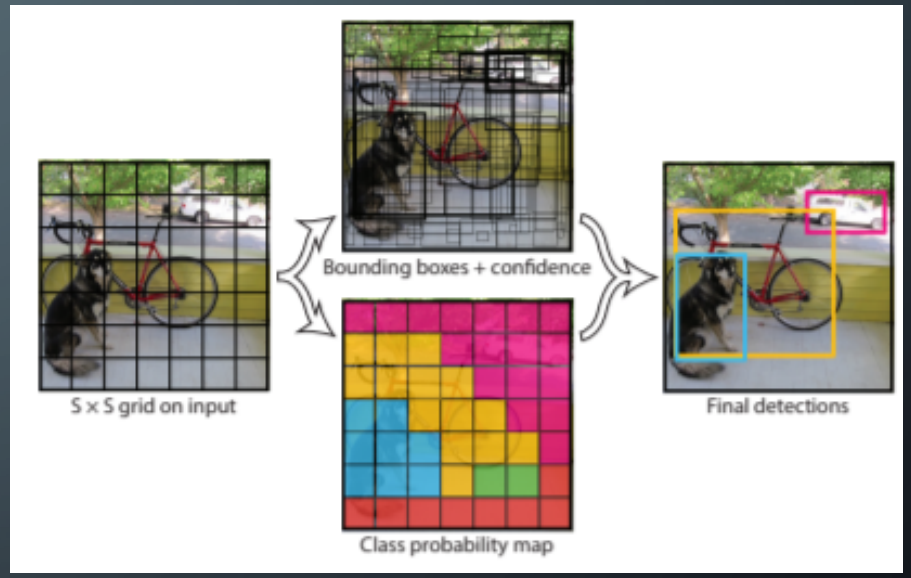
YOLO DETECTION SYSTEM



- Processing images with YOLO is simple and straightforward:
- First, resize the input image to 448 × 448
- Then run a single convolutional network on the image
- threshold the resulting detections by the model's confidence

UNIFIED DETECTION

- The system models detection as a regression problem.
- It divides the image into an S × S grid and for each grid cell predicts B bounding boxes, confidence for those boxes, and C class probabilities.
- These predictions are encoded as an S × S × (B ∗ 5 + C) tensor



COMPARISON WITH OTHER MODELS

- **Deformable parts model (DPM**) uses a sliding window approach for object detection
- **R-CNN** and its variants use region proposals instead of sliding windows to find objects in images
- **OverFeat** trains a convolutional NN to perform localization and adapt that localizer to perform detection
- **MultiGrasp** only predicts a single graspable region for an image containing one object.

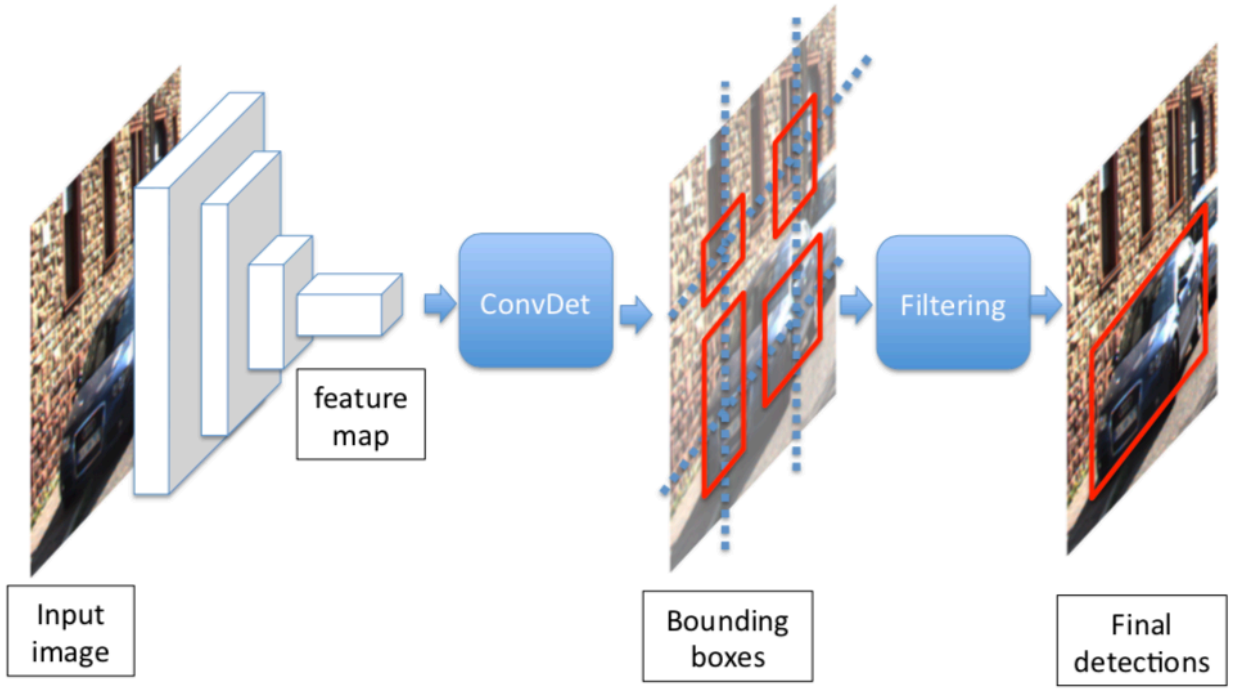
SQUEEZEDET: CNN FOR AUTONOMOUS DRIVING

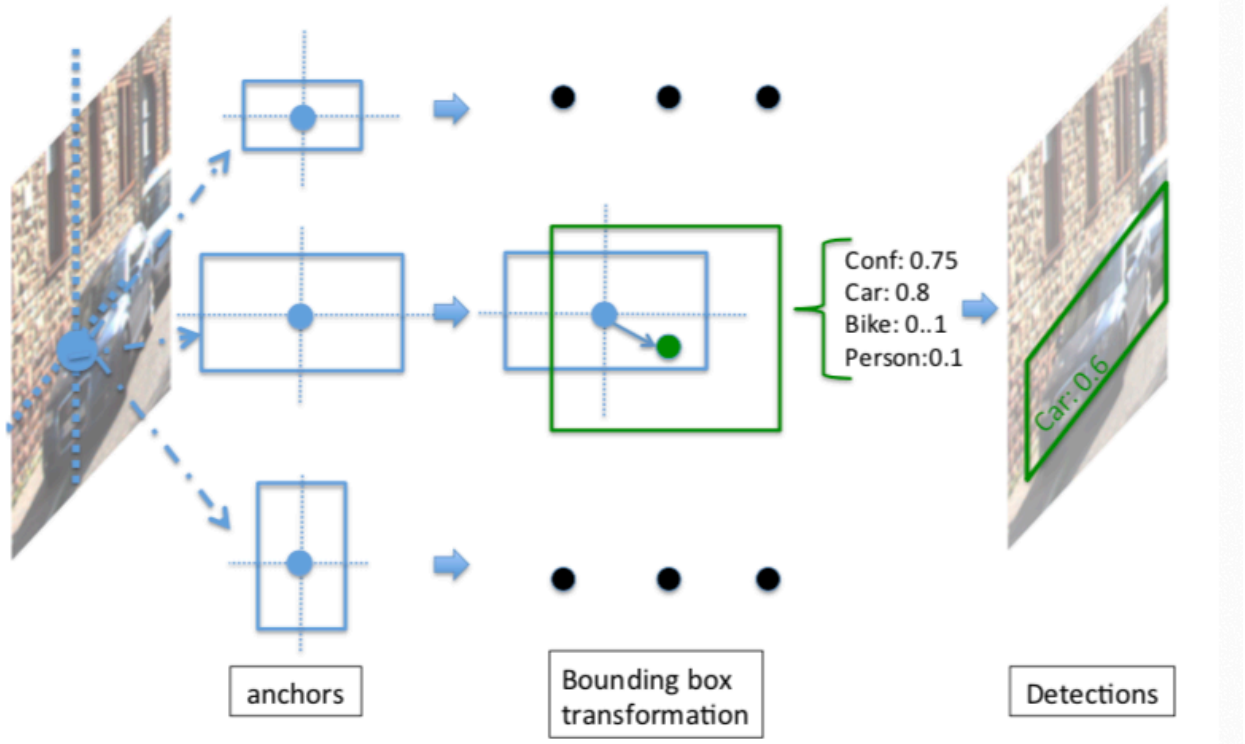
BASIC REQUIREMENTS

- Accuracy
- Speed
- Small model size
- Energy Efficiency

SQUEEZEDET DETECTION PIPELINE

- Single-stage detection pipeline





CONVDET LAYER

- Convolutional layer that is trained to output bounding box coordinates and class probabilities
- Sliding window
- K x (4 + 1 + C)

NEURAL NETWORK DESIGN

- SqueezeDet
  - 4.72 MB of model size and > 80.3% ImageNet accuracy
- SqueezeDet+
  - 19 MB of model size and 86.0% of ImageNet accuracy

EXPERIMENTAL SETUP

- All input images were scaled to 1242x375
- Randomly split 7381 training images into a training set and a validation set
- Stochastic Gradient Descent with momentum was used to optimize loss function
- Trained the model until mAP on training set converged and the model was evaluated on validation set

Method	Car mAP	Cyclist mAP	Pedestrian mAP	All mAP	Model size (MB)	Speed (FPS)
FRCN + VGG16[2]	86.0	-	-	-	485	1.7
FRCN + AlexNet[2]	82.6	-	-	-	240	2.9
SqueezeDet (ours)	82.9	76.8	70.4	76.7	**7.9**	**57.2**
SqueezeDet+ (ours)	85.5	**82.0**	**73.7**	**80.4**	26.8	32.1
VGG16-Det (ours)	**86.9**	79.6	70.7	79.1	57.4	16.6
ResNet50-Det (ours)	86.7	80.0	61.5	76.1	35.1	22.5

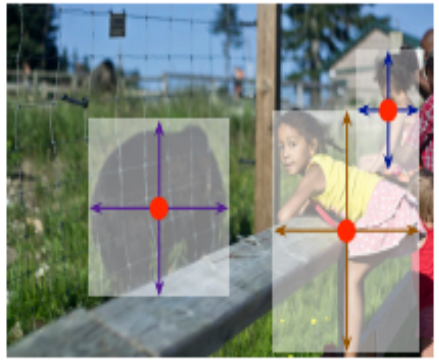
RESULTS

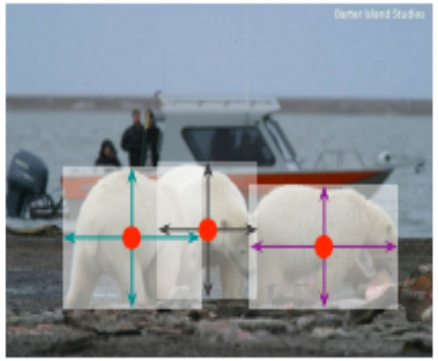
- Average Precison: SqueezeDet+
- Recall: SqueezeDet (91%) and SqueezeDet+ (92%)
- Speed: SqueezeDet (57.2 FPS)
- Model size: SqueezeDet 61x smaller than FRCCN + VGG16 and 30x smaller than FRCNN + AlexNet

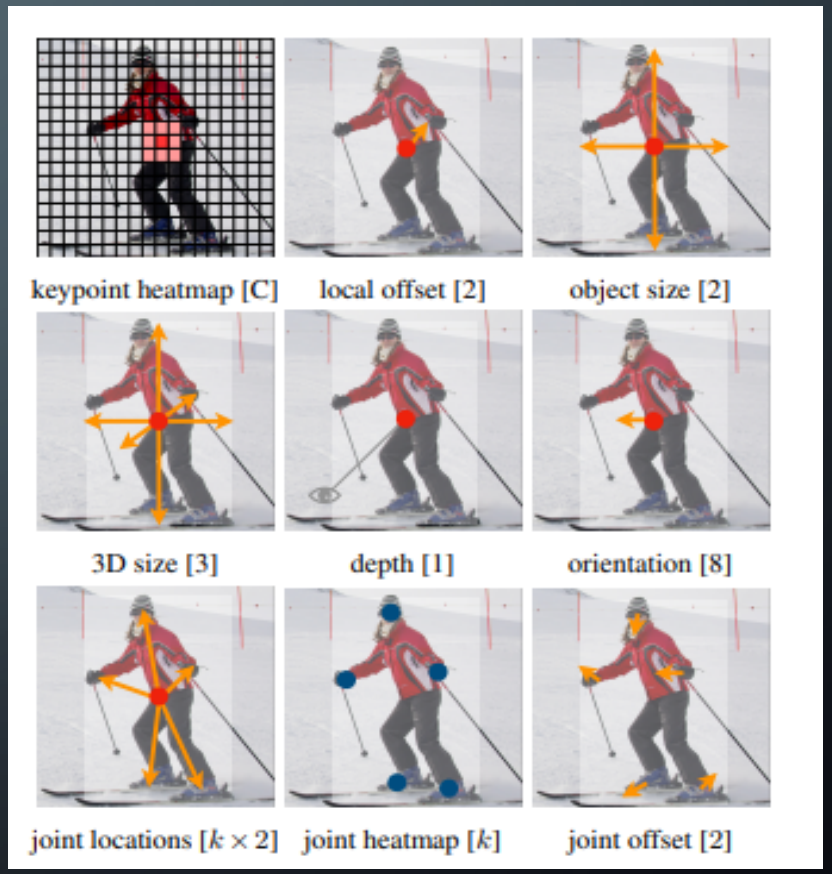
SUMMARY

- SqueezeDet was able to achieve 30.4x smaller model size, 19.7x faster inference speed, and 35.2x lower energy in comparison with its baselines

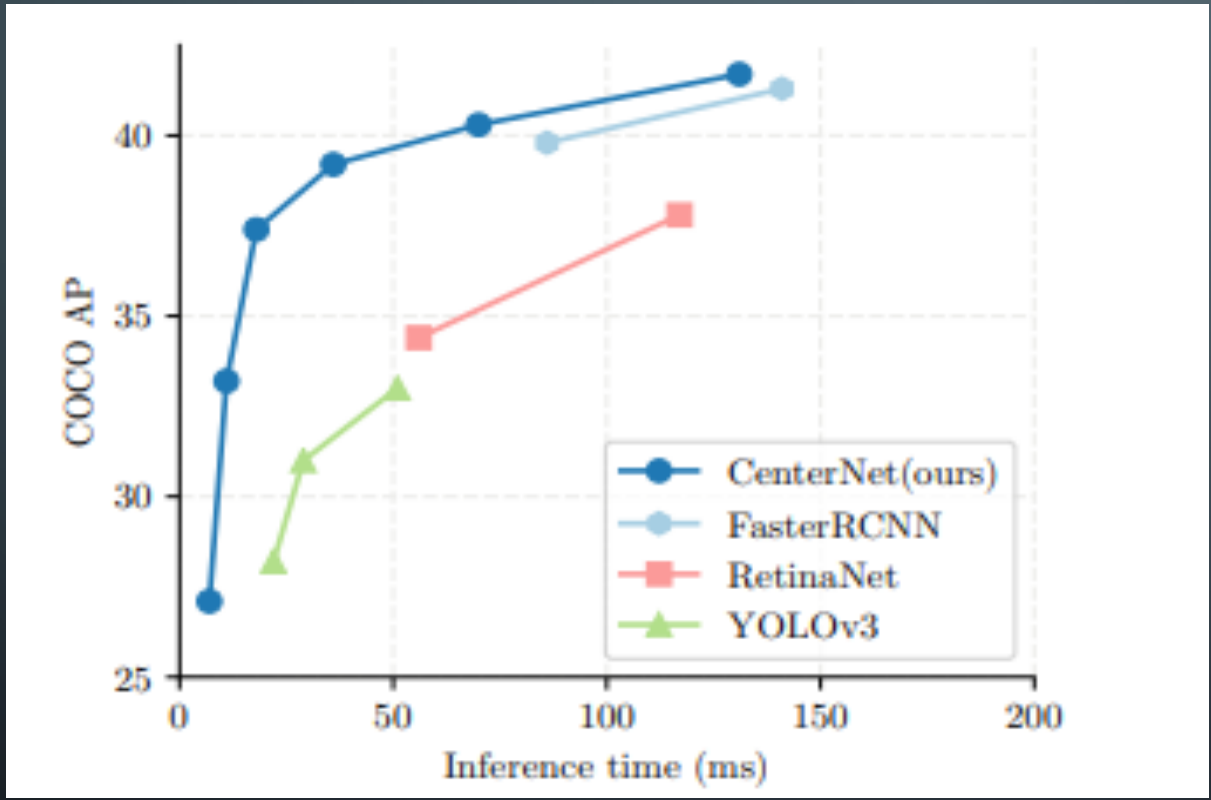
CENTERNET: OBJECTS AS POINTS

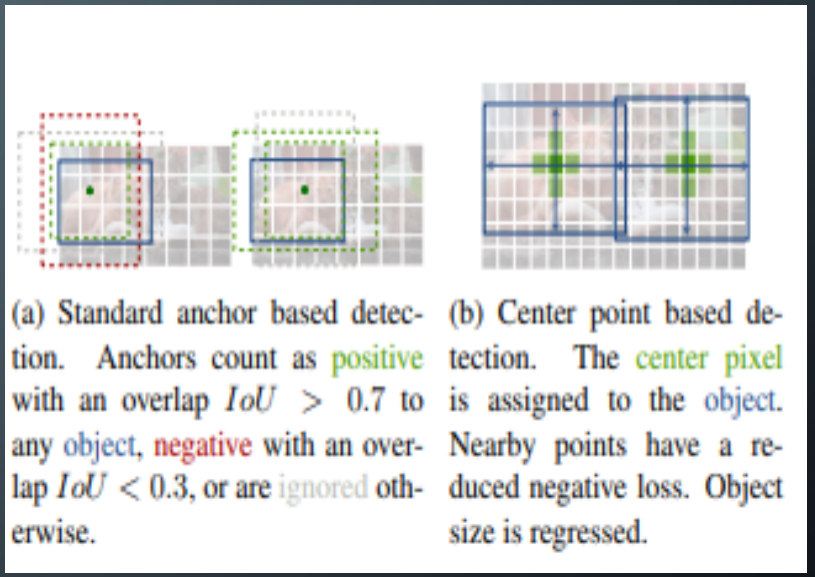






SPEED/ACCURACY TRADE OFF ON COCO VALIDATION FOR REAL-TIME DETECTORS





(a) Standard anchor based detection. Anchors count as positive with an overlap $IoU > 0.7$ to any object, negative with an overlap $IoU < 0.3$, or are ignored otherwise.

(b) Center point based detection. The center pixel is assigned to the object. Nearby points have a reduced negative loss. Object size is regressed.

PERFORMANCE

	Backbone	FPS	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
MaskRCNN [21]	ResNeXt-101	**11**	39.8	62.3	43.4	22.1	43.2	51.2
Deform-v2 [63]	ResNet-101	-	46.0	67.9	50.8	27.8	49.1	59.5
SNIPER [48]	DPN-98	2.5	46.1	67.0	51.6	29.6	48.9	58.1
PANet [35]	ResNeXt-101	-	47.4	67.2	51.8	30.1	**51.7**	60.0
TridentNet [31]	ResNet-101-DCN	0.7	**48.4**	**69.7**	**53.5**	**31.8**	51.3	**60.3**
YOLOv3 [45]	DarkNet-53	20	33.0	57.9	34.4	18.3	25.4	41.9
RetinaNet [33]	ResNeXt-101-FPN	5.4	40.8	61.1	44.1	24.1	44.2	51.2
RefineDet [59]	ResNet-101	-	36.4 / 41.8	57.5 / 62.9	39.5 / 45.7	16.6 / 25.6	39.9 / 45.1	51.4 / 54.1
CornerNet [30]	Hourglass-104	4.1	40.5 / 42.1	56.5 / 57.8	43.1 / 45.3	19.4 / 20.8	42.7 / 44.8	**53.9** / 56.7
ExtremeNet [61]	Hourglass-104	3.1	40.2 / 43.7	55.5 / 60.5	43.2 / 47.0	20.4 / 24.1	43.2 / 46.9	53.1 / 57.6
FSAF [62]	ResNeXt-101	2.7	**42.9** / 44.6	**63.8 / 65.2**	**46.3** / 48.6	**26.6 / 29.7**	**46.2 / 47.1**	52.7 / 54.6
CenterNet-DLA	DLA-34	**28**	39.2 / 41.6	57.1 / 60.3	42.8 / 45.1	19.9 / 21.5	43.0 / 43.9	51.4 / 56.0
CenterNet-HG	Hourglass-104	7.8	42.1 / **45.1**	61.1 / 63.9	45.9 / **49.3**	24.1 / 26.6	45.5 / **47.1**	52.8 / **57.7**

CENTERNET WITH 4 DIFFERENT ARCHITECTURES

	AP			AP_{50}			AP_{75}			Time (ms)			FPS		
	N.A.	F	MS	N.A.	F	MS	N.A.	F	MS	N.A.	F	MS	N.A.	F	MS
Hourglass-104	**40.3**	**42.2**	**45.1**	**59.1**	**61.1**	**63.5**	**44.0**	**46.0**	**49.3**	71	129	672	14	7.8	1.4
DLA-34	37.4	39.2	41.7	55.1	57.0	60.1	40.8	42.7	44.9	19	36	248	52	28	4
ResNet-101	34.6	36.2	39.3	53.0	54.8	58.5	36.9	38.7	42.0	22	40	259	45	25	4
ResNet-18	28.1	30.0	33.2	44.9	47.5	51.5	29.6	31.6	35.1	**7**	**14**	**81**	**142**	**71**	**12**

CONCLUSION

- We plan to implement CenterNet to perform real-time object detection on shelf-monitoring, highway clean-up automation as it provides the best speed/accuracy trade off.

THANK YOU!

REFERENCES

- Yona Falinie, A. Gaus, Neelanjan Bhowmik, Samet Akcay, Paolo M. Guillen-Garcia, Jck W. Barker, Toby P. Breckon, "Evaluation of a Dual Convolutional Neural Network Architecture for Object-wise Anomaly Detection in Cluttered X-Ray Security Imagery," April 10, 2019.
- Pengchong Jin, Vivek Rathod, Xiangxin Zhu, Google, "Pooling Pyramid Network for Object Detection", July 9, 2018.
- "*You Only Look Once: Unified, Real-Time Object Detection*", Joseph Redmon, Santosh Divvalay, Ross Girshick, Ali Farhadi, University of Washington, Allen Institute for AI, Facebook AI Research
- "*Body part detectors trained using 3d human pose annotations*", L. Bourdev and J. Malik. Poselets, In International Conference on Computer Vision (ICCV), 2009
- D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov, "*Scalable object detection using deep neural networks*". In Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, pages 2155–2162. IEEE, 2014
- Bichen Wu, Alivin Wan, Forrest Iandola, Peter H. Jin, Kurt Keutzer, UC Berkeley, DeepScale, "SqueezeDet: Unified, Small, Low Power Fully Convolutional Neural Networks for Real-Time Object Detection for Autonomous Driving," Jun 11, 2019.
- Xingyi Zhou, Dequan Wang, Philipp Krahenbuhl, UT Auston, UC Berkeley, "Objects as Points," Apr 25, 2019.