Convolutional Neural Network Models in Human and Object Detection

I. Introduction

Computer vision, a field of artificial intelligence (AI), has been gaining momentum in the recent years due to advancements being made in hardware, rise of deep learning, and presence of large datasets. It can be found and applied to various industries, ranging from automotive to agricultural and industrial. Due to the enormity of computer vision, this AI subfield can be divided into different categories, such as face detection and recognition and pose estimation. For our project, we will be working on an application within the human and object detection categorization of computer vision.

In today's world, human and object detection applications can be found everywhere. For example, video surveillance systems on the streets or homes are able to detect humans and other objects within the vicinity. In addition, due to the popularity of smart cars, another popular example is a pedestrian detection system. Before we begin working on an application, we researched and studied state-of-the-art approaches that researchers have developed in the realm of human and object detection.

In this report, we will be examining different state-of-the-art convolutional neural networks (CNNs) by providing a concise description of each. More specifically, we will first take a look at the evaluation of a dual CNN architecture for object-wise anomaly detection in cluttered X-ray security imagery to get a sense of how a CNN is applied. Next, we will dive into Pooling Pyramid Network (PPN), followed by You Only Look Once (YOLO). Subsequently, we will look at SqueezeDet and CenterNet. Lastly, we will end our discussion of the state-of-the-art approaches by choosing one of the discussed models for our project.

II. Dual CNN Object Detection in Security Cameras

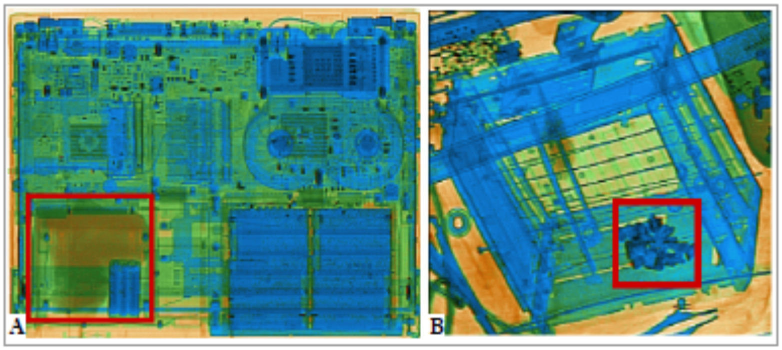
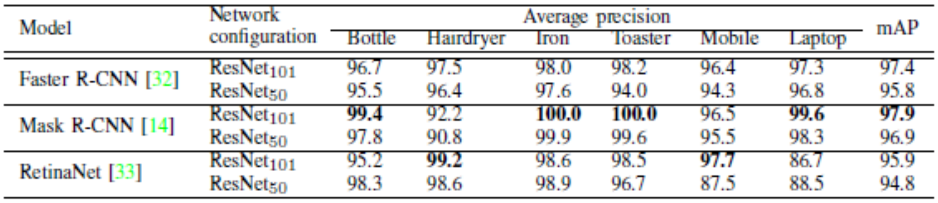


Fig. 1. X-ray security imagery of exemplar electronics items with a highlighted (red box) concealed anomalous region in (A) laptop and (B) toaster.

This paper presents a dual convolutional neural network (CNN) architecture for automatic anomaly detection within complex security X-ray imagery. This leverages the recent advancements in region-based CNN (R-CNN), mask-based CNN (Mask R-CNN) and detection architectures such as RetinaNet to provide object localization variants for specific object classes of interest. The best performing object localization method is able to perform with 97.9% mean average precision (mAP) over a six-class X-ray object detection problem, subsequent two-class anomaly/benign classification is able to achieve 66% performance for within object anomaly detection. A coloured-map of X-Ray image is first produced that corresponds to material properties which is detected via the dual-energy X-ray scanning process. Then, the threat detection performance is characterized by high detection and low false alarm rates for operational viability.

The author has proposed two stage analysis: (a) is primary object detection in the X-ray image and (b) is classification of the objects as {anomaly, benign}. For detection strategy the paper compares the results of Faster R-CNN, Mask R-CNN and RetinaNet over a range of following objects: {bottle, hairdryer, iron, toaster, mobile, laptop}.

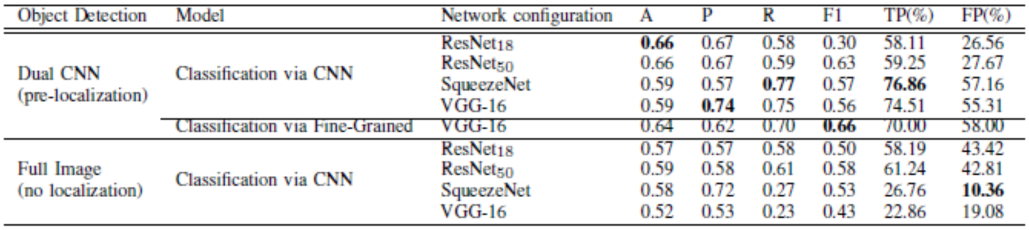
TABLE I
OBJECT DETECTION RESULTS FOR FASTER R-CNN, MASK R-CNN AND RETINANET FOR DUAL CNN ARCHITECTURE. CLASS NAMES INDICATES CORRESPONDING AVERAGE PRECISION (AP) OF EACH CLASS, AND MAP INDICATES MEAN AVERAGE PRECISION OF THE CLASSES.



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

For classification strategy the paper uses architectures like SqueezeNet, VGG-16, ResNet pre-trained on ImageNet and focuses on fine-grained classification that aims at distinguishing subordinate visual categories such as species of dogs, birds and plants. This method learns a set of convolution filters such that each is initialized and discriminatively trained in order to capture highly discriminative sub-image patches. The model uses a dataset of 3534 images and calculates Accuracy(A), Precision(P), Recall(R), F-score(F1%), True Positive(TP%), and False Positive(FP%).

TABLE II
ANOMALY CLASSIFICATION VIA VARYING CNN ARCHITECTURES (SQUEEZENET, VGG, AND RESNET) WITH AND WITHOUT THE PRE-LOCALIZATION OFFERED BY THE PROPOSED DUAL CNN ARCHITECTURE.



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

In conclusion, 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, while experimental results on secondary anomaly detection via a two-class classification problem, anomaly, benign show the benefits of a dual CNN architecture (TP: 76.86% Accuracy: 66%) false positive detection remains a significant issue (FP >10%).

III. Pooling Pyramid Network (PPN) for Object Detection

Single Shot Multibox Detector (SSD)

It is a popular algorithm in object detection which is generally faster than RCNN. It uses a single deep neural network for detecting objects in images. SSD detectors have been popular as they run fast with simple implementation and have high portability to different types of hardware.

This paper shares a simple tweak of the Single Shot Multibox Detector (SSD) which reduces the model size while maintaining the same quality of the algorithm. This has two advantages over vanilla SSD: (1) it avoids score miscalibration across scales; (2) a shared predictor sees the training data over all scales.

Pooling Pyramid Network (PPN)

The proposed model is very similar to vanilla SSD with simple changes:

1. A single box predictor is shared across feature maps with different scales.
2. Convolutions between feature maps are replaced with max pooling operations.

Since the vanilla SSD uses independent box predictors for feature maps at different scales one problem is miscalibration of the prediction scores across different scales as each box predictor is trained independently which produces incomparable scores in vastly different ranges. This is solved by designing PPN with a shared box predictor across feature maps of different scales which reduces the effect of miscalibration and unstable prediction scores.

Max pooling operations ensures feature maps with different scales live in the same embedding space increasing efficiency of shared box predictor. Additionally, max pooling does not require any additions and multiplications making it very fast to compute during inference.

Architecture Comparisons

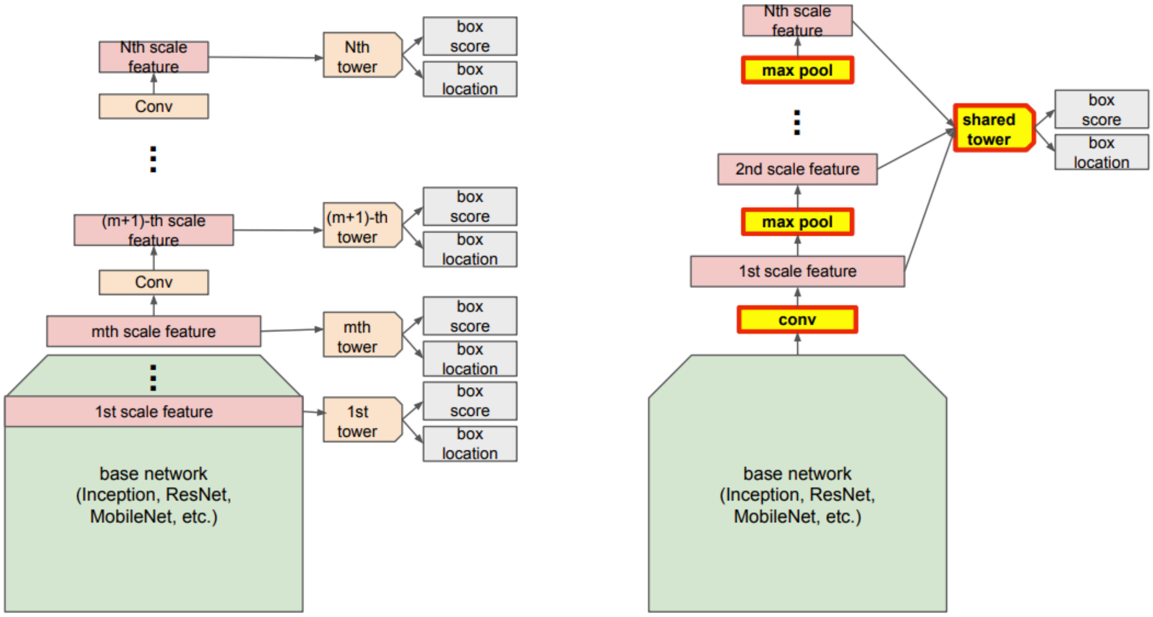
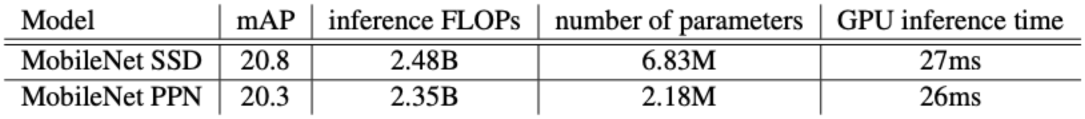


Figure) Left: Vanilla SSD; Right: PPN

Changes in PPN like max pool for building feature pyramid and shared convolutional predictors for classification and regression are highlighted for reference.

Experiments

The experiments are run on the COCO detection dataset and performance of PPN with vanilla SSD is compared on standard implementations of MobileNet-v1.



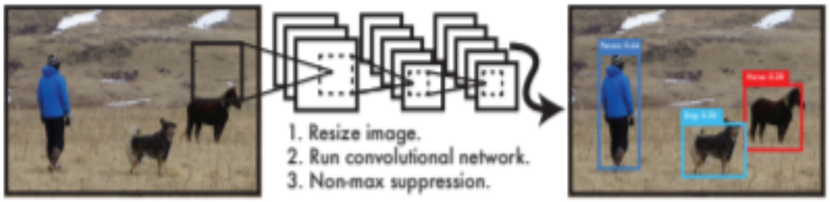
Model	mAP	inference FLOPs	number of parameters	GPU inference time
MobileNet SSD	20.8	2.48B	6.83M	27ms
MobileNet PPN	20.3	2.35B	2.18M	26ms

Table 1. COCO detection: MobileNet SSD vs MobileNet PPN

PPN achieves similar mAP (20.3 vs. 20.8) with comparable FLOPS and inference time but is 3x smaller in model size than SSD.

IV. YOLO: Unified, Real-Time Object Detection

Real-time object detection system detects the object in 3 steps



1. The input image is resized into 448 x 448 pixels
2. A single convolutional network is run on the image
3. Threshold the resulting detection by model's confidence

What are the datasets available for training and evaluations?

ImageNet 1000, Pascal Visual Object classes (VOC) 2007 and Pascal VOC 2012 datasets provide a standard dataset of images, annotation, and standard evaluation procedures for YOLO detection system.

How was the problem of human and object detection solved using classical computer vision?

Prior work on **object detection repurposes classifiers to perform detection**. Approaches like R-CNN use region proposal methods to generate potential bounding boxes in an image and then run a classifier on these boxes. After classification, post-processing is used to refine bounding boxes, eliminate duplicate detections, and rescore the boxes based on other objects in the scene.

Summaries of the state-of-the-art approaches

- **Deformable Parts Model (DPM)** uses a sliding window approach for object detection. DPM uses a disjoint pipeline to extract static features, classify regions, and predict bounding boxes for high scoring regions.

- **R-CNN** and its variants use region proposals instead of sliding windows to find objects in images.

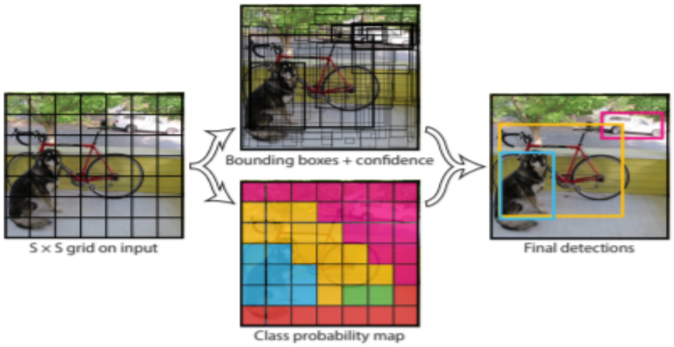
- **OverFeat** trains a CNN to perform localization and adapt that localizer to perform detection.

- **MultiGrasp** only predicts a single graspable region for an image containing one object. It doesn't estimate the size, location, or boundaries of the object or predict its class, only finds a region suitable for grasping. YOLO predicts both bounding boxes and class probabilities for multiple objects of multiple classes in an image

Evaluation of YOLO Model Systems

Conventional R-CNN models have complex pipelines that are slow and hard to optimize because each individual component must be trained separately. **YOLO 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 $S \times S \times (B * 5 + C)$ tensor.



E.g.: For evaluating YOLO on PASCAL VOC, we use S = 7, B = 2. PASCAL VOC has 20 labeled classes so C = 20. Our final prediction is a 7 × 7 × 30 tensor.

1. **Network Design:**

   Features of image are extracted by initial convolutional network and the fully connected layers predict the coordinates and output probabilities. YOLO network has 24 convolutional layers followed by 2 fully connected layers.

2. **Alternating 1 × 1 convolutional layers reduce the features space from preceding layers:**

   We pre-train the convolutional layers on the ImageNet classification task at half the resolution (224 × 224 input image) and then double the resolution for detection.

3. **Training:**

   Pre-train model on the ImageNet 1000-class dataset using 20 convolutional layers, an average pooling layer, and a fully connected layer for a week.

   Then convert the model to perform detection. Adding both convolutional and connected layers to pre-trained networks can improve performance. Further, add four convolutional layers and two fully connected layers with randomly initialized weights.

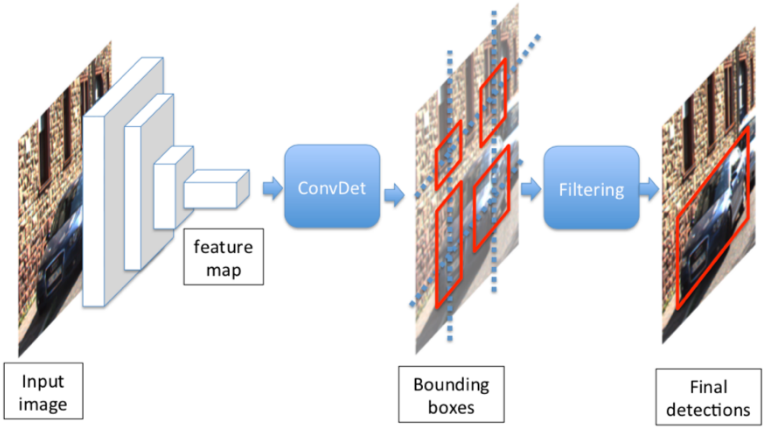
Maximize loss from bounding box coordinate predictions λ_{coord} and minimize loss from confidence predictions for boxes with no objects λ_{noobj}.

4. **Inference:**
   Predicting detections for a test image only requires one network evaluation. On PASCAL VOC the network predicts 98 bounding boxes per image and class probabilities for each box. **YOLO is extremely fast at test time since it only requires a single network evaluation**, unlike classifier-based methods.

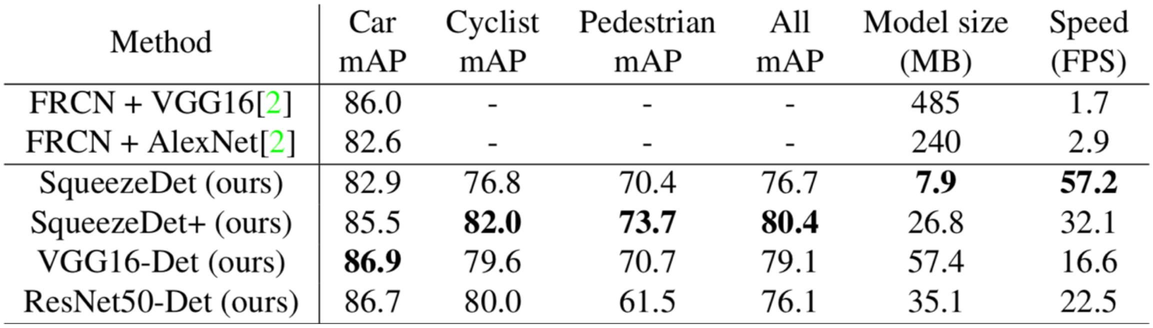
V. SqueezeDet: Fully CNN for Real-Time Object Detection for Autonomous Driving

Autonomous driving systems are at the forefront of today's fast-paced and technology-driven world. Before meeting the increased demands for autonomous driving technology, object detection must be used "to require high accuracy to ensure safety, real-time inference speed to guarantee prompt vehicle control, and a small model size and energy efficiency to enable embedded system deployment." In order to meet these constraints, researchers at UC Berkeley and DeepScale developed a fully convolutional neural network (CNN) for object detection called SqueezeDet. These researchers strongly believe that by "addressing the problems of accuracy, speed, small model size, and energy efficiency" the capabilities of deep neural networks can be maximized for autonomous driving, and hence, SqueezeDet was developed.

SqueezeDet uses SqueezeNet as the "backbone" CNN architecture and was inspired by the popular YOLO object detection system. Similar to YOLO, SqueezeDet performs "region proposition and classification in one single network simultaneously." After feeding an input image into a CNN, a low-resolution, high dimensional feature map is extracted. Subsequently, the extracted feature map is fed into the *ConvDet* layer. The trained *ConvDet* convolutional layer works as a "sliding window that moves through each spatial position on the feature map." It is trained by a multi-task loss function to learn detection, localization, and classification in order to output bounding box coordinates and class probabilities. Each of the computed bounding boxes contains C + 1 values where C is the number of distinguishable classes and 1 is for the confidence score. The confidence score indicates the likelihood of an object's existence within the bounding box and it is computed as $Pr(Object) * IOU^{pred}_{truth}$. Based on this equation, a high confidence score indicates a high probability that an object exists and an overlap between the predicted bounding box and ground truth is high. The C values represent conditional class probabilities and is denoted as $Pr(class_c \mid Object),\ c \in [1,C]$. The label is then assigned to the bounding box with the highest conditional probability and the metric $max_c\ Pr(class_c \mid Object) * Pr(Object) * IOU^{pred}_{truth}$ is used to get an estimate of the confidence of the bounding box prediction. Lastly, top N bounding boxes are kept and Non-Maximum Suppression (NMS) is used to remove redundant bounding boxes to retrieve the final detected objects.



Before examining experimental results, it's important to note that the researchers adopted two versions of the SqueezeNet architecture. The first model is known as SqueezeDet and uses the SqueezeNet v1.1 model with 4.72MB of model size and > 80.3% ImageNet top-5 accuracy. The second model is known as SqueezeDet+ and uses 86.0% of ImageNet accuracy and 19MB of model size. These two models were tested on the KITTI object detection dataset on a NVIDIA TITAN X GPU. The models' accuracy was measured by average precision (AP), recall, speed, and model size and later compared with a faster-RCNN based object detector trained on the KITTI dataset under the same experimental settings (shown in the table below). As a result, the total model size is less than 8MB, inference speed can reach 57.2 FPS with input image resolution of 1242x375. In addition, it requires fewer DRAM accesses resulting in a consumption of only 1.4J of energy per image. In the end, this model is 30.4x smaller, 19.7x faster, and consumes 35.2x lower energy.

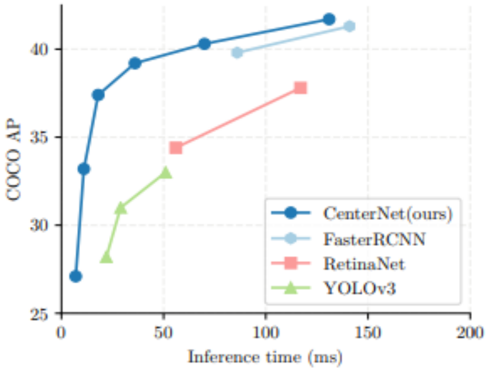


Method	Car mAP	Cyclist mAP	Pedestrian mAP	All mAP	Model size (MB)	Speed (FPS)
FRCN + VGG16[2]	86.0	-	-	-	485	1.7
FRCN + AlexNet[2]	82.6	-	-	-	240	2.9
SqueezeDet (ours)	82.9	76.8	70.4	76.7	**7.9**	**57.2**
SqueezeDet+ (ours)	85.5	**82.0**	**73.7**	**80.4**	26.8	32.1
VGG16-Det (ours)	**86.9**	79.6	70.7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

VI. CenterNet: Objects as Points

Intro to CenterNet

"Objects as points" is a unique way of detecting objects through key point estimation as opposed to the conventional object detectors which use sliding window to enumerate an exhaustive list of potential object locations and classify each. This enables CenterNet, a center point-based approach, to outperform a range of state-of-the-art algorithms such as YOLOv3, Faster-RCNN and RetinaNet.



What is Key Point Estimation?

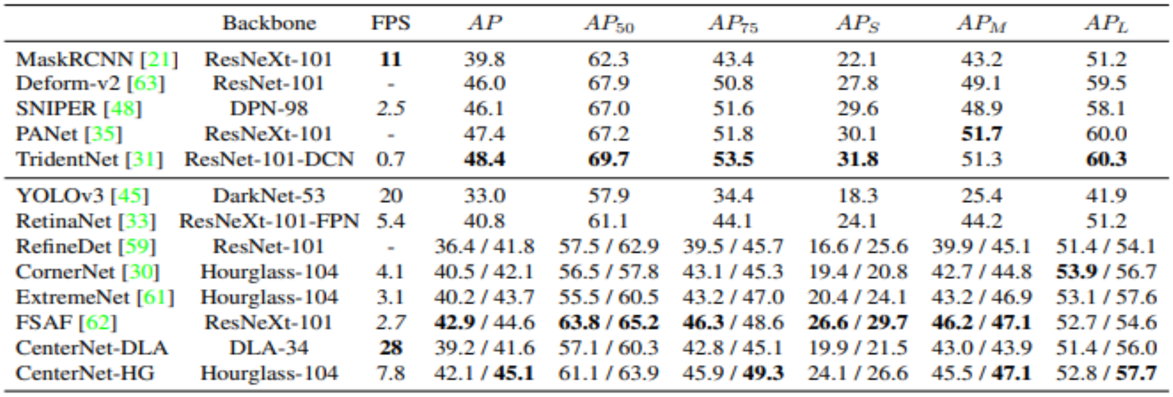
Representing objects by a single point at their bounding box center is basically key point estimation. Other properties such as object size, dimension, 3D extent, orientation, and pose are then regressed directly from image features at the center location. The input image is fed to a fully convolutional network which generates a heatmap corresponding to object centers. Peaks in the heatmap correspond to object centers. Image features at each peak predict the objects bounding box height and weight. The model is trained using standard dense supervised learning.

How is it different from classical object detectors?

Classical object detectors use an axis-aligned bounding box to surround the object. They then reduce object detection to image classification of an extensive number of potential object bounding boxes. For each bounding box, the classifier determines if the image content is a specific object or background. We have one-stage detectors which classify the image directly and two-stage detectors which recompute image features and classify them. This is inefficient & requires additional processing.

How CenterNet is built and its performance on COCO dataset.

CenterNet runs at a very high speed with a simple Resnet-18 and up-convolutional layers. It achieves the best speed-accuracy trade-off on the MS COCO dataset, with 28.1% AP at 142 FPS. With a carefully designed keypoint detection network, DLA - 34, it achieves 37.4% AP at 52 FPS. Equipped with the state-of-the-art keypoint estimation network, Hourglass - 104, and multi scale testing, it achieves 45.1% AP with multi-scale testing at 1.4 FPS.



	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**

VII. Conclusion

Through this research on different state-of-the-art approaches for human and object detection in computer vision, we were able to gain an understanding of different methodologies and examine a speed/accuracy tradeoff to determine the best fit model for object detection. Aside from the dual CNN Object Detection in Security Cameras research, the rest of the research papers fall within the real-time object detection subtask of human and object detection. Furthermore, since the focus of our group is to perform real-time object detection such as retail shelf-monitoring and highway cleanup automation, we prioritize speed over accuracy while also maintaining a healthy balance between the two. Therefore, we believe that CenterNet would be the best model for our project.

VII. 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.
- Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi, University of Washington, Allen Institute for AI, Facebook AI Research, “You Only Look Once: Unified, Real-Time Object Detection,” May 9, 2016.
- 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.