COMPARATIVE STUDY OF LATEST PAPERS

ON OBJECT DETECTION

Before this, what is Two-stage vs One-stage Detectors?

Models in the R-CNN family are all region-based. The detection happens in two stages: (1) First, the model proposes a set of regions of interests by select search or regional proposal network. The proposed regions are sparse as the potential bounding box candidates can be infinite. (2) Then a classifier only processes the region candidates. They are **slow** compared to one stage Detectors.

The other different approach skips the region proposal stage and runs detection directly over a dense sampling of possible locations. This is how a one-stage object detection algorithm works. This **is faster** and simpler, but might potentially drag down the performance (accuracy) a bit.

Source: https://lilianweng.github.io/lil-log/2018/12/27/object-detection-part-4.html#two-stage-vs-one-stage-detectors

Let's now review some of the common model architectures used for object detection:

- R-CNN Family
- SSD (Single Shot MultiBox Defender)
- YOLO (You Only Look Once)
- Objects as Points
- Data Augmentation Strategies for Object Detection

Here, we will get the detailed explanation about two stage detectors and one stage detectors (RCNN family vs YOLO/SSD/RetinaNet). **We already saw this, but let's see quickly:**

==========

DL? allows smarter way of filtering the pixels in an image with smarter computation cost performing tasks like recognition, classification, classification with localization, object detection, object tracking etc. by giving high scope for fine tuning (that is for accuracy and precision). Transfer learning is an extra-ordinary feature of DNNs. Compromise in Data/Potential hand design components or net adjustments

CNN? Classification with Object Localization we can get object recognized, not only that but also we get the bounding box for the object in an image. In this way you get the ground truth box. Sequential sliding search through the defined windows and passing them through the CNN is the very basic approach.
But again it still has a problem of not quite outputting the most accurate bounding boxes.

YOLO? The advantage of this algorithm is that the neural network outputs precise bounding boxes. Limitation is long as you don't have more than one object in each grid cell, this algorithm should work okay.
So, decided to go with YOLO for Detection! CNN on Grid cells with bounding boxes IoU! (Simple Algorithm!)

State of CV: Data (vs) Hand Engineering : DL are prone to high Data feeding or else given less data we should be able to manually handle the features of the network

YOLOv3 (8 Apr 2018) → FEEDBACK: THERE ARE MUCH NEWER ALGORITHMS PUBLISHED! …

Here we compare few recent papers:

#	Model	Date	Type	Model/Method	Exp. Setup/Tools	Results	Pros	Cons	Applications
1	YOLOv4	23rd Apr 2020	Single Stage _ Heads:: _ Dense Prediction (one-stage): , FCOS (anchor free) _ Sparse Prediction (two-stage):	We use new features: WRC, CSP, CmBN, SAT, Mish activation, Mosaic data augmentation, CmBN, DropBlock regularization, and CIoU loss, and combine some of them for CNN. Input: Image, Patches, Image Pyramid YOLOv4 consists of: _ Backbone: CSPDarknet53 _ Neck: SPP, PAN _ Head: YOLOv3	Use a 1080 Ti or 2080 Ti GPU. .Bag-of-Freebies and Bag-of-Specials. We modify state-of-the-art methods for single GPU training - For GPU - CLayers: CSPResNeXt50 / CSPDarknet53 _ For VPU - Squeeze-and-excitement (SE) blocks: EfficientNet-lite / MixNet / GhostNet/ MobileNetV3 CPU Config: Intel Core i7 7700HQ, RAM:16GB DDR4, NVIDIA GeForce GTX1050 (Notebook), HardDisk: Samsung SSD 960 EVD 1TB	43.5% AP (65.7% AP50) for the MS COCO dataset at a real time speed of _65 FPS on Tesla V100. Tested on: MS COCO and ImageNet Comparison on GPUs: Maxwell /Pascal/ Volta FPS (~65 to 120)	Selection of BoF and BoS. Introduced: method of data augmentation Mosaic and Self-Adversarial Training (SAT), CBN/PAN		Fast and Accurate Detectors than ever The original concept of one-stage anchor-based detectors has proven its viability.
2	ResNeST	19th Apr 2020		Stacking these Split-Attention blocks ResNet-style, we obtain a new ResNet variant which we call ResNeSt. Radix-major Split-Attention Block	ImageNet, ResNet-50, even trained on GPUs, MS-COCO, FPN, Auto Augmentation, train GluonCV with MXNet	ResNeSt-50 achieves 81.13% top-1 accuracy on using a single crop-size of 224 224, outperforming previous best ResNet variant by more than 1% accuracy. we improve the mAP of Faster-RCNN on MS-COCO from 39.3% to 42.3% and the mIoU for DeeplabV3 on ADE20K from 42.1% to 45.1%.	mAP by around 3% on both Faster-RCNNs and Cascade-RCNNs. See Performance tables		Downstream tasks including object detection, instance segmentation and semantic segmentation.

						Speed: (~2ms to 28ms)			
3	LapNet	17th Mar 2020	Fast Single Stage	An anchor based detector, trained end-to-end without any sampling strategy. PONO Anchor assignments. Additional weights at inference	Fully convolutional network with encoder-decoder and skip connections PASCAL VOC, MSCOCO	PASCAL VOC (automatic balancing method), MSCOCO (mAP/fps) Speed: ~50ms to 75ms	Good compromise in performance and computational speed, simple and generic. Fast than RetinaNet		Most accurate detector among real-time methods.
4	NAS-FPN	16th Apr 2019	Multistage	New feature pyramid architecture in a novel scalable search space covering all cross-scale connections (Neural Architecture Search). top-down and bottom-up connections to fuse features across scales	Backbone RetinaNet. Search to learn a RNN controller to discover the NAS-FPN architecture. Trained on TPUs with 64 images in a batch. COCO train2017, AmoebaNet. Our controller is a recurrent neural network (RNN) and it is trained using the Proximal Policy Optimization (PPO) algorithm.	NAS-FPN improves mobile detection accuracy by 2 AP compared to state-of-the-art SSDLite with MobileNetV2 model in and achieves 48.3 AP which surpasses Mask R-CNN detection accuracy with less computation time. Speed: 50ms to 300ms	Better accuracy and latency tradeoff. Scalable Feature Pyramid Architecture. Performance see the tables below.		Architectures for high detection accuracy. Architectures for fast inference
5	RetinaNet	7th Feb 2018	Single Stage	propose to address this class imbalance by reshaping the standard cross entropy loss, Focal Loss focuses training on a sparse set of hard examples	Single, unified network composed of a backbone network and two task-specific subnetworks. The backbone (FPN) is responsible for computing a convolutional feature map over an entire input image and is an off-the-self convolutional network. SGD).We use synchronized SGD over 8 GPUs with a total of 16 images per	AP: ~33 to 40 Speed: 70ms to 200ms	able to match the speed of previous one-stage detectors while surpassing the accuracy of all existing state-of-the-		Fully Convolutional one stage detector

					minibatch (2 images per GPU). Training time ranges between 10 and 35 hours. COCO benchmark		art two-stage detectors		
6	DENet	17th Apr 2019	Mask R-CNN	Detection network and encoder-decoder estimation network.	ShanghaiTech Part A, UCF and WorldExpo'10 datasets. STech: Part A has 300 images for training and 182 images for testing, and Part B has 400 images for training and 316 images for testing. The UCF CC 50 dataset includes 50 annotated images with 63974 persons in total. WorldExpo'10 dataset includes 3980 annotated images from 1132 video sequences that taken from 108 different surveillance cameras. UCSD dataset has 2000 images taken from surveillance cameras.	See the tables Mask R-CNN + MCNN/CSRNEt These performance is given as improvement in MAE and MSE over diffterent datasets	Lower Mean Absolute Error (MAE). high-quality density maps without losing resolution		Estimate the density map for the areas where individuals cannot be segmented due to high crowded- ness.
7	DetNet	19th Apr 2018	Mukti-stage	Novel backbone network from classification task, extra stages (like FPN) for high spatial resolution in deeper layers for detection.	Stage 1,2,3,4 the same as original ResNet-50, spatial resolution FPN used. 8 Pascal TITAN XP GPUs, optimized by synchronized SGD with a weight decay of 0.0001 and momentum of 0.9. image flipping Data augmentation. RoI-Align technique	MSCOCO, ImageNet classification DetNet-59 has 23.5% top-1 error at the cost of 4.8G FLOPs, AP50/AR50 is impressive. mAP is 40.2 FPS not mentioned	Better than RetinaNet (ResNet-101) and Mask RCNN (ResNet-101). DetNet has just (4.8G FLOPs)		object detection and instance segmentation based on the COCO benchmark
8	CenterNet	25th Apr 2019	Objects as Points	We experiment with 4 architectures: ResNet-18, ResNet101, DLA-34 [58], and Hourglass-104. We modify both ResNets and DLA-34 using deformable convolution layers and use the	On MS COCO dataset, which contains 118k training images, 5k validation images (val2017) and 20k hold-out testing images (test-dev). Run Time Test with Intel Corei7-8086K CPU, Titan Xp GPU, Pytorch 0.4.1, CUDA 9.0, and CUDNN 7.1.	MS COCO dataset, with 28.1% AP at **142 FPS**, 37.4% AP **at 52 FPS**, and 45.1% AP with multi-scale testing at **1.4 FPS.** –	CenterNet, is end-to-end differentiable, simpler, faster, and more accurate than corresponding bounding box		Our detector uses keypoint estimation to find center points and regresses to all other object properties, such as size, 3D location, orientation, and

				Hourglass network as is.		Speed: 5-120ms	based detectors.		even pose. (Real Time app)
9	PoolNet	21st Apr 2019		U-shape architecture, we first build a global guidance module (GGM), feature aggregation module (FAM), then Joint Training	a single NVIDIA Titan Xp GPU. Less than 6 hours on a training set of 5,000 images. PyTorch. We only use the simple random horizontal flipping for data augmentation. Backbone VGG-16 and ResNet-50 are initialized with the corresponding models pre-trained on the ImageNet dataset [	> **30 FPS** when processing a **300X400 image**. joint training our network with the standard edge detection task in an end-to-end learning manner can greatly enhance the details of the detected salient objects (MAE metrics given)	High-level semantic features to be progressively refined. First paper that aims at pooling-based modules. flexibly applied to any pyramid-based models		More accurately locate the salient objects. Impact of Joint Training boosts MAE metric
10	SimCLR	13th Feb 2020	Data Augmentation	Contrastive Learning - data augmentations, from larger batch sizes and more training, contrastive loss substantially	Single GPU/ TPUs. But no Multi-GPUs. TensorFlow v1 and v2 can be used.	76.5% top-1 accuracy, which is a 7% relative improvement over previous supervised ResNet-50. fine-tuned on only 1% of labels, 85.8% top-5 accuracy, outperforming AlexNet with 100X fewer labels	Out-perform previous methods for self-supervised and semi-supervised learning on ImageNet. (handles more parameters)		No specialized architectures or a memory bank. Useful for accurate detections
11	Gaussian Mixture Probability MOT + Occlusion Group Mngt.	31st Jul 2019		(GMPHD) filter and occlusion. Group management scheme, Tack State Machine (D2T and T2T) “track merging, “occlusion group energy minimization” occlusion ratio between visual objects, sum-of-intersection-over-area (SIOA) instead of IoU. MOT15 and MOT17 Datasets.	GMPHD-OGM tracker is implemented by Visual C++ with OpenCV3.4.1 and boost1.61.0 libraries, and without any GPU-accelerated libraries such as CUDA. All experiments are conducted on Windows 10 with Intel i7-7700K CPU @ 4.20GHz and DDR4 32.0GB RAM.	See the tables for performance and CLEAR-MOT metrics FPS: 30-170	SIOA works better than IOU. GMPHD-OGM shows the competitive value in “MOTA versus speed”		efficient online multiobject tracking framework
12	Joint Monocular 3D Vehicle Detection	12th Sep 2019		We propose a novel online framework for 3D vehicle detection and tracking from monocular videos.	KITTI, and Argoverse Datasets, detection of interest (DOI), Occlusion-aware Data Association. Deep Motion Estimation and	MOTA: 50-85% Processing Time, FPS: 12	framework can not only associate detections of		On Argoverse, our image based method is significantly better for tracking 3D

	and Tracking			Our method leverages 3D box depth-ordering matching for robust instance association and utilizes 3D trajectory prediction for re-identification of occluded vehicles. We also design a motion learning module based on an LSTM for more accurate long-term motion extrapolation.	Update. Prediction LSTM (P-LSTM). Our simulation is based on Grand Theft Auto V, a modern game that simulates a functioning city and its surroundings in a photo-realistic three-dimensional world. Depth-Ordering Matching. GPUs (effective minibatch size is 20). GT camera and Mono-VO		vehicles in motion over time, but also estimate their complete 3D bounding box information fssrom a sequence of 2D images captured on a moving platform.		vehicles within 30 meters than the LiDAR-centric baseline methods, advantage of 3D estimation leveraging our collection of dynamic 3D trajectories
13	Instance-Aware, Context-Focused, & Memory-Efficient Weakly Supervised OD	9th April 2020		Instance-aware and context-focused unified framework. It employs an instance-aware self-training algorithm and a learnable Concrete DropBlock while devising a memory efficient sequential batch back-propagation. Multiple instance selftraining (MIST)	COCO 2014 train/val/test split and VOC 2007 and 2012. use 8 GPUs during training with one input image per device. SGD is used for optimization. ImageNet- VID	COCO (12.1% AP, 24.8% AP50), VOC 2007 (54.9% AP), and VOC 2012 (52.1% AP), improving baselines by great margin. The default p and IoU in our proposed MIST technique are set to 0:15 and 0:2. Process Time: >=3s	First to benchmark ResNet based models and weakly supervised video object detection. novel online self-training algorithm with ROI regression to reduce instance ambiguity and better leverage the self-training supervision.	Missing Instances' issue	Part-domination for classes with large intra-class variance via a novel end-to-end learnable 'Concrete DropBlock'. short-term motion patterns
14	PV-RCNN	31st Dec 2019		PV-RCNN for Point Cloud Object Detection, Feature Encoding and Proposal Generation, cene Encoding viaVoxel Set	Waymo Open Dataset, KITTI Dataset, only point clouds, ADAM optimizer, KITTI: 8 GTX 1080 Ti GPUs, WOD: 32 GTX 1080 Ti GPUs	RoI-grid Pooling IoU-guided Scoring E92.57 M84.83H 82.69	efficient learning and high-quality proposals of the 3D voxel CNN and the flexible receptive fields of the		Conventional pooling operations, the RoI-grid feature points encode much richer context information for accurately

				Abstraction, Keypointtogrid RoI Feature Abstraction for Proposal Refinement			PointNet-based networks		estimating object confidences and locations.

HERE! Every New Algorithm/Paper has the unique type of architecture and potentially used for various applications of CV (DNN) on Aerial Robotics in Object Detection and Tracking.

Git Codes :

YOLOv4 : https://github.com/AlexeyAB/darknet

Tutorial for YOLOv4: https://robocademy.com/2020/05/01/a-gentle-introduction-to-yolo-v4-for-object-detection-in-ubuntu-20-04/

CenteNet: https://github.com/xingyizhou/CenterNet

SimCLR: https://github.com/vamshikodipaka/simclr

RetinaNet: https://github.com/facebookresearch/Detectron

ResNeST: https://github.com/zhanghang1989/ResNeSt

NAS-FPN : https://github.com/tensorflow/tpu/tree/master/models/official/detection

PV-RCNN: https://github.com/jhultman/vision3d

PoolNet : https://github.com/backseason/PoolNet

Instance-Aware, CF, ME,WS-OD: https://github.com/NVlabs/wetectron

GMPHD Filter and Occlusion Group Management: https://github.com/SonginCV/GMPHD-OGM_Tracker

Joint Monocular 3D Vehicle Detection and Tracking: https://github.com/ucbdrive/3d-vehicle-tracking

References:

1. A 2019 Guide to Object Detection:

https://www.kdnuggets.com/2019/08/2019-guide-object-detection.html

2. Paperswithcode.com