Object Detection and Uncertainty

Shreyasha Paudel

NAAMII AI Winter School

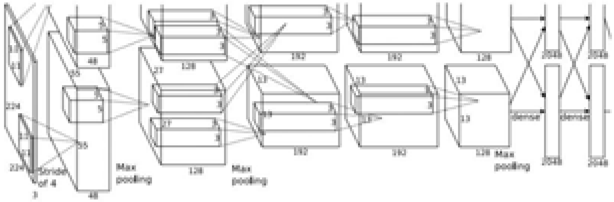
December, 2019

Outline

- Introduction to Object Detection
- Deep Learning Techniques for Object Detection
  - Two stage detectors (Region Proposal Networks)
  - One stage detectors
  - 3D Object Detection
- Applications
- Limitations -> Need for uncertainty measure
- Quantifying uncertainty in neural networks
  - Aleatoric and Epistemic Uncertainty
  - Sampling based techniques
  - Evidential method

Object Detection: Single Object

(Classification + Localization)



Vector: 4096

Fully Connected: 4096 to 1000

Class Scores
Cat: 0.9
Dog: 0.05
Car: 0.01
...

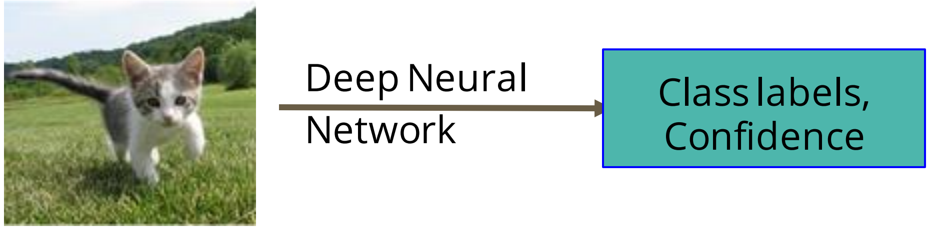
Fully Connected: 4096 to 4

Box Coordinates
(x, y, w, h)

Treat localization as a regression problem!

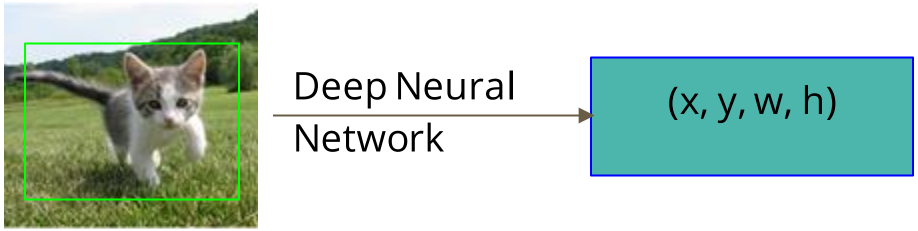
Classification

- Classifying an image into many different categories
- DNNs now perform better than humans
- How convolution works:
  - Break images into small tiles
  - Hidden layers learn to recognize interesting features from these tiles
  - Form a hierarchical layer of features from all tiles
  - These layers will be mapped to a class



Localization

- Finding the location of a single object within an image
- Intuition: replace softmax layer in CNNs with a L2 loss
  - L2 loss = Distance based loss between ground truth boxes and detected box
- In practice:
  - It works better to obtain 4D vectors for each class
  - Only backpropagate the loss for correct class
  - Apply this at multiple location and scales

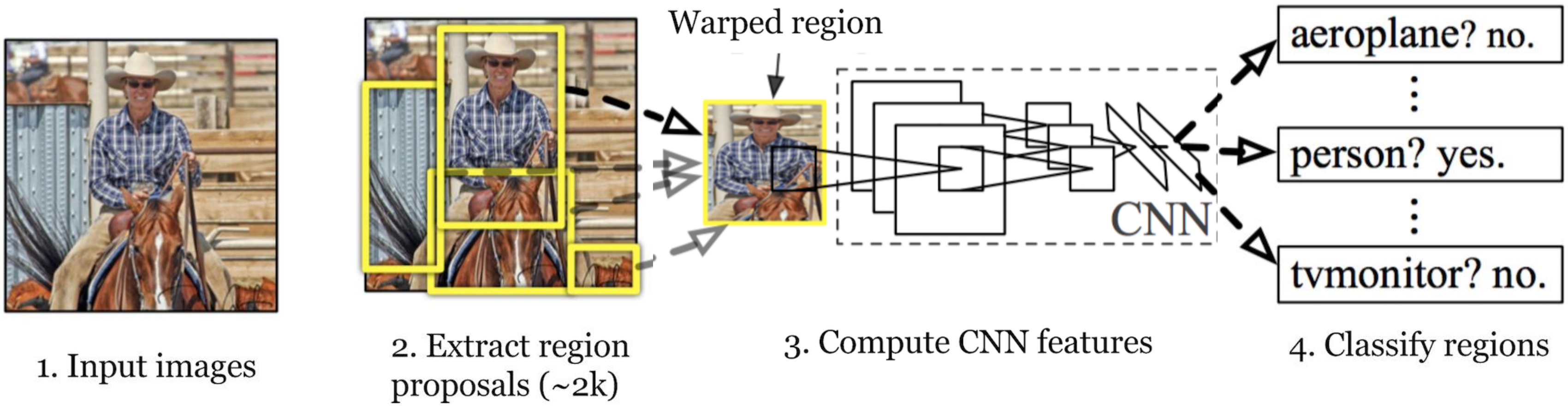


Techniques for Object Detection/Classification

1. Two Stage Detectors (Region Proposal Networks)
   - RCNN (Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014)
   - Fast RCNN (Girshick, "Fast R-CNN", ICCV 2015)
   - Faster RCNN (Ren et al, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", NIPS)

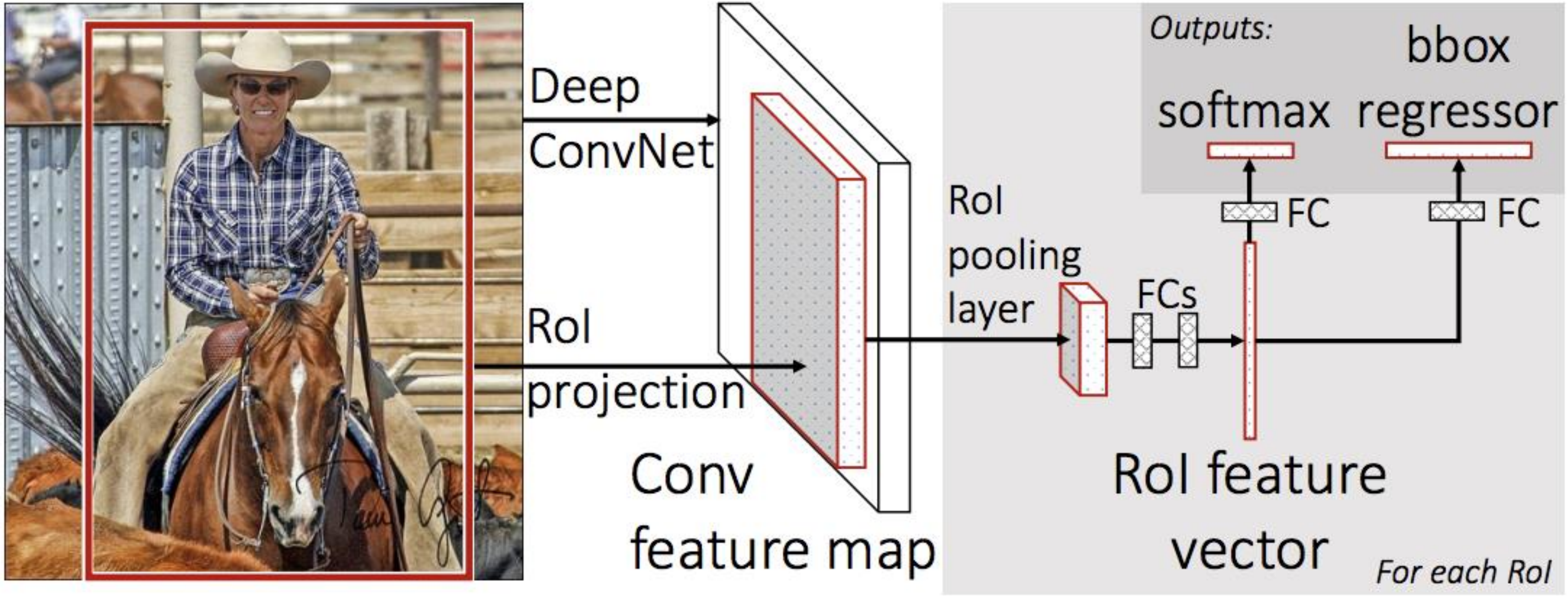
2. One Stage Detectors
   - YOLO (Redmon et al, You Only Look Once: Unified, Real-Time Object Detection, CVPR2016)
   - SSD (Liu et al, SSD: Single Shot MultiBox Detector, ECCV2016)
   - Retinanet (Lin et al, Focal Loss for Dense Object Detection)

RCNN



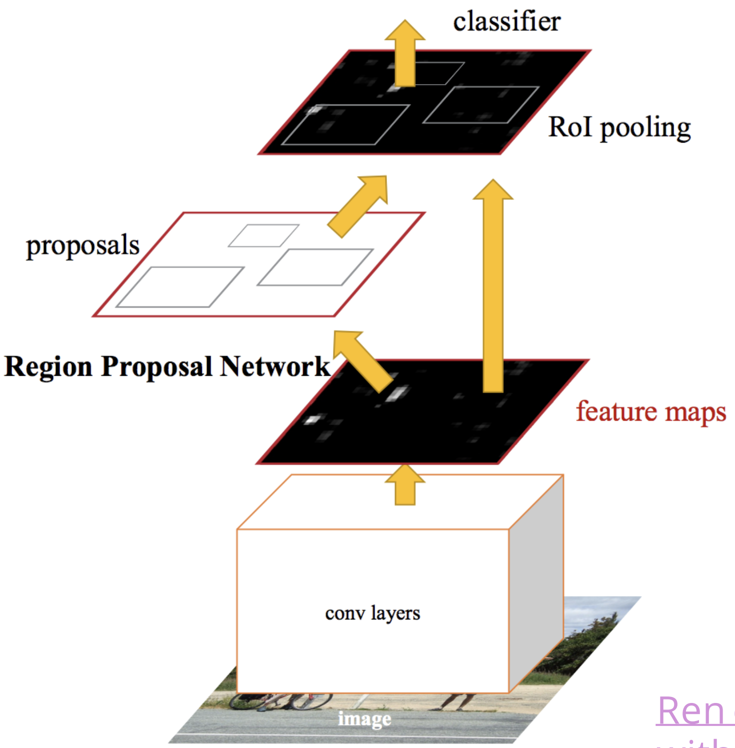
[Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014]()

Fast RCNN



Girshick, "Fast R-CNN", ICCV 2015

Faster RCNN



Two modules:

1. **Region Proposal Network :** CNN for proposing regions and the type of object to consider in the region.
2. **Fast R-CNN**. CNN for extracting features from the proposed regions and outputting the bounding box and class labels.

Ren et al, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", NIPS

Region Proposal Networks

1. Feature Extraction Network

- Input images are passed through a CNN to get a high-dimensional representation
- Usually, networks pretrained on Imagenet are used (eg: VGG, Resnet, Mobilenet)

Region Proposal Networks

1. Feature Extraction Network

2. Region Proposal Network

- Using the features, find a predefined number of regions where an object is located
- For optimization, anchor boxes are used

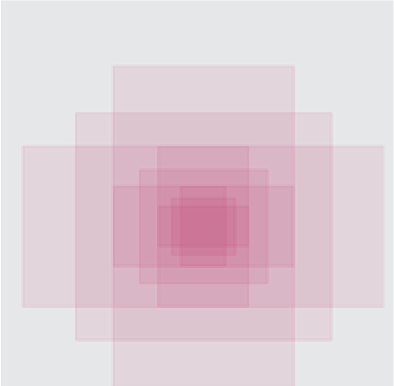
Optimization: Anchor boxes

- If we just search over the whole image, too many options (different sizes, scales, etc). Need long time and extremely deep networks to optimize.
- Idea: Provide reference boxes and find offsets from them insted

Anchor Boxes

Anchor box centers are distributed throughout the image in a grid

Each center has a number of boxes at different scales/aspect ratios



https://tryolabs.com/blog/2018/01/18/faster-r-cnn-down-the-rabbit-hole-of-modern-object-detection/

Region Proposal Networks

1. Feature Extraction Network

2. Region Proposal Network

- Using the features, find a predefined number of regions where an object is located
- For optimization, anchor boxes are used

Region Proposal Networks

1. Feature Extraction Network

2. Region Proposal Network

3. RoI Pooling

- Extract only the features from regions of interest detected in previous step

Region Proposal Networks

1. Feature Extraction Network

2. Region Proposal Network

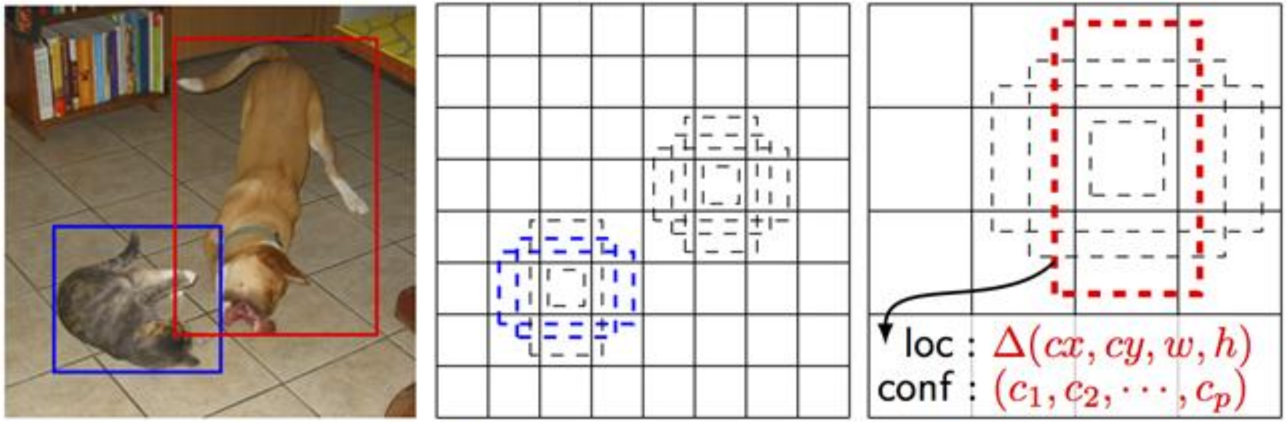
4. R-CNN

- Classify the content from the features or discard it (if background class)
- Fine tune the bounding box co-ordinates

3. ROI Pooling

One Stage Detectors

Combine region proposal and classification into a single stage



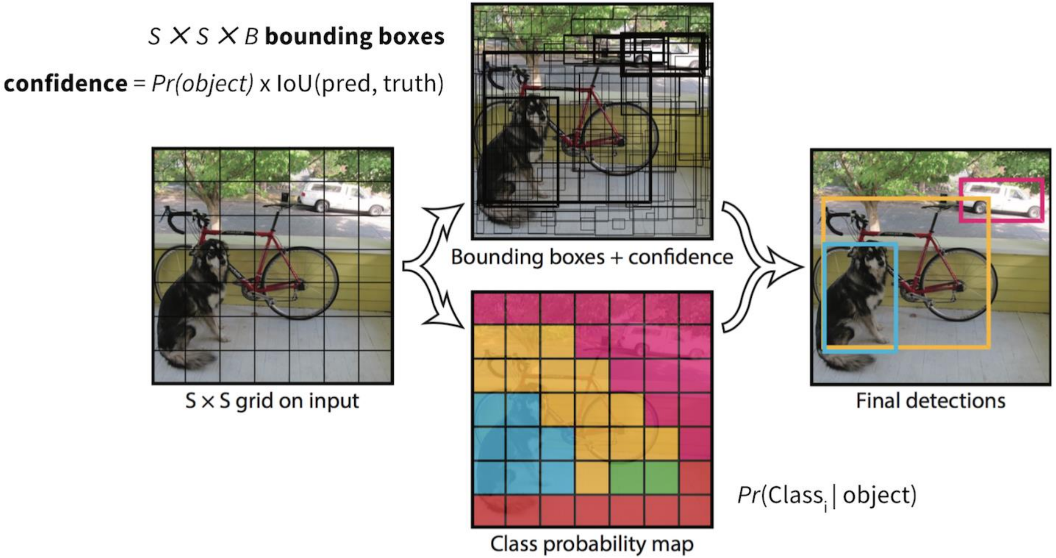
(a) Image with GT boxes (b) 8×8 feature map (c) 4×4 feature map

https://towardsdatascience.com/understanding-ssd-multibox-real-time-object-detection-in-deep-learning-495ef744fab

One Stage Detectors : YOLO

Basic Idea:

- Split image into arbitrary cells.
- A cell is responsible for detecting object at its center.
- Each cell predicts:
  - *i)* B bounding boxes,
  - *ii)* confidence score,
  - *iii)* probability of object class

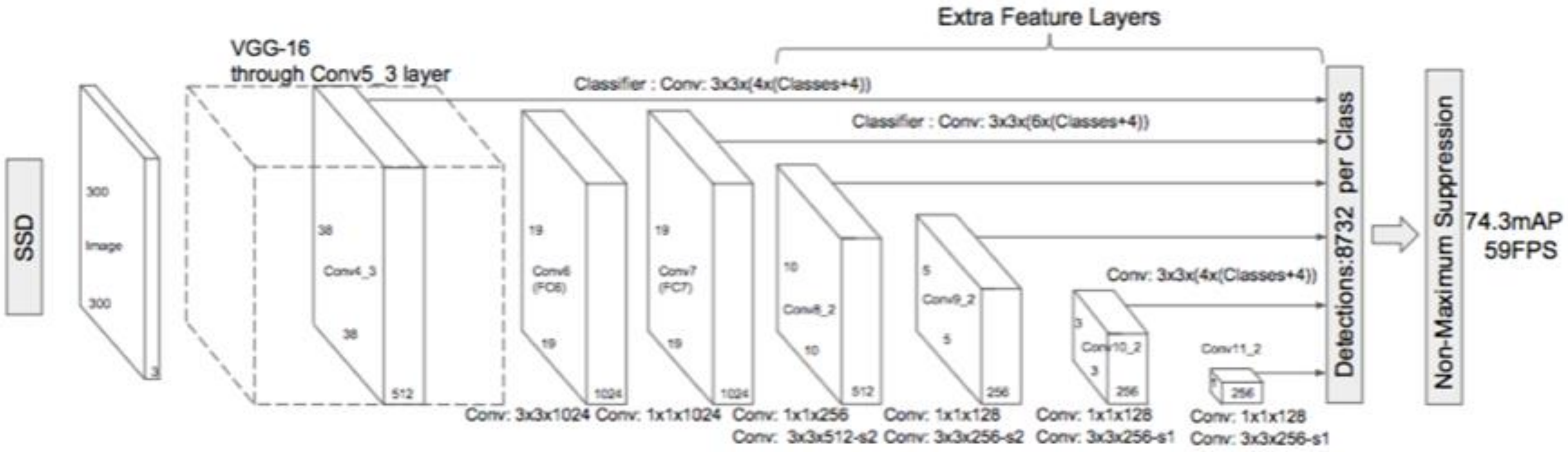


Improving versions:

- YOLO
- YOLO v2
- YOLO v3 – State of the art

Redmon et al, You Only Look Once: Unified, Real-Time Object Detection, CVPR2016

One Stage Detectors : SSD

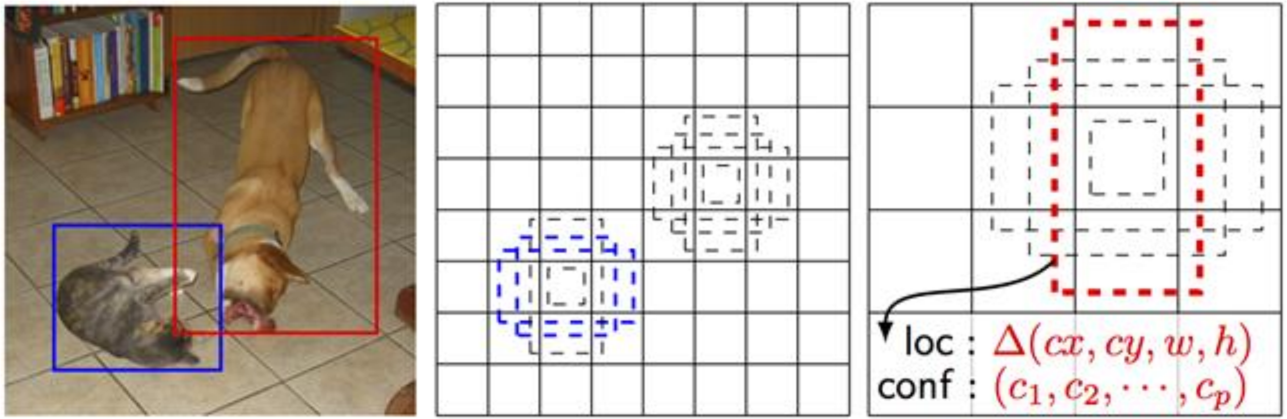


Unlike YOLO, SSD predicts offsets from predefined anchor boxes.

[Liu et al, SSD: Single Shot MultiBox Detector, ECCV2016]

Optimizations: Anchor Boxes

- In SSD like architectures, every feature map cell is associated with a set of default bounding boxes of different dimensions and aspect ratios.
- These priors are manually (but carefully) chosen



(a) Image with GT boxes (b) 8×8 feature map (c) 4×4 feature map

https://towardsdatascience.com/understanding-ssd-multibox-real-time-object-detection-in-deep-learning-495ef744fab

One Stage Detectors: Retinanet

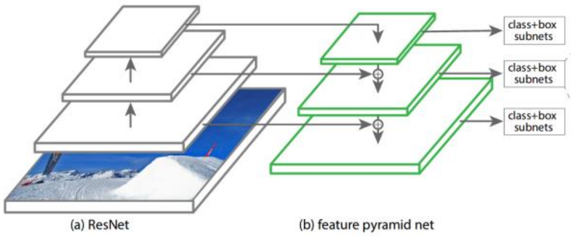
- An extension to SSD
- Improvements:

Lin et al, Focal Loss for Dense Object Detection

One Stage Detectors: Retinanet

- An extension to SSD
- Improvements:

    1. Feature Pyramid Network for Object Detection

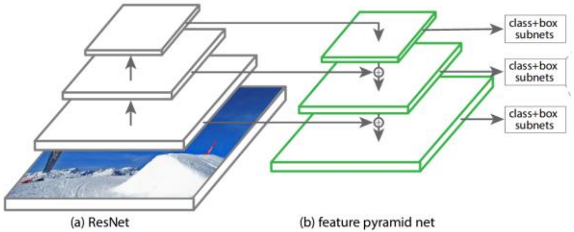


[Lin et al, Focal Loss for Dense Object Detection]

One Stage Detectors: Retinanet

- An extension to SSD
- Improvements:

1. Feature Pyramid Network for Object Detection



2. Focal Loss for Dense Object Detection : Modification of categorical cross-entropy to help solve SSD's class imbalance problem

[Lin et al, Focal Loss for Dense Object Detection]

NMS : Non-max suppression

- Boxes with confidence threshold < c_t and IOU greater than i_t are ignored
- Only the top N prediction for each class are kept

Figure 1: We propose a non-maximum suppression convnet that will re-score all raw detections (top). Our network is trained end-to-end to learn to generate exactly one high scoring detection per object (bottom, example result).

https://towardsdatascience.com/understanding-ssd-multibox-real-time-object-detection-in-deep-learning-495ef744fab

Different choices for 2D object detection

Base Network:

- VGG-16
- Resnet-101
- Inception
- Resnext
- Mobilenet

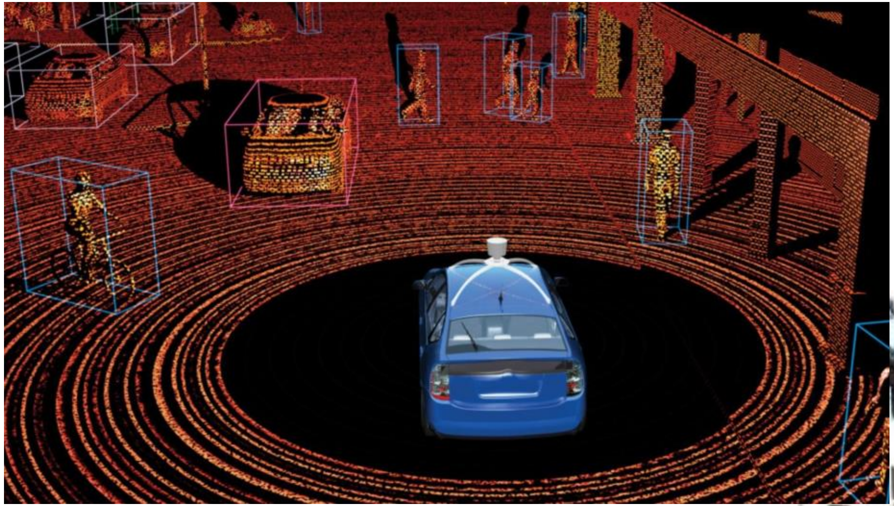
Architecture

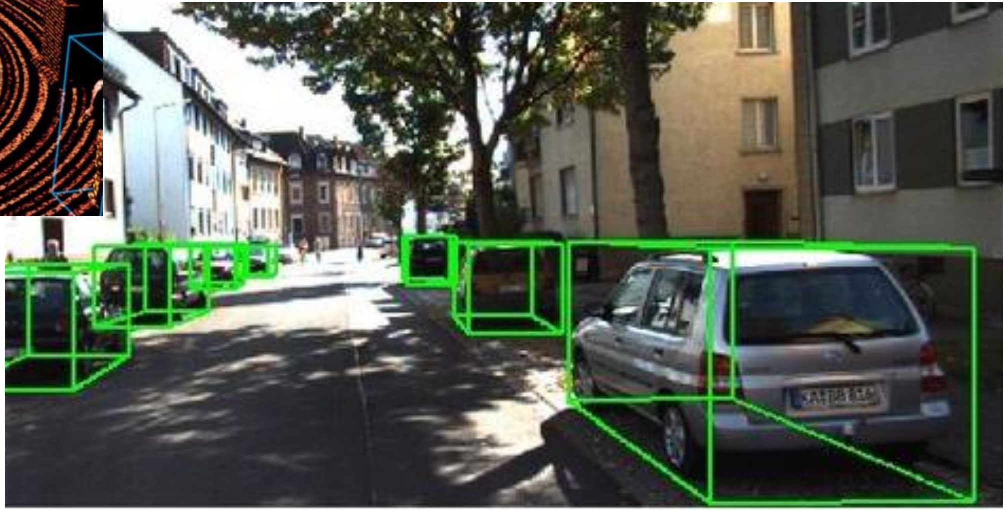
- Faster R-CNN
- Retinanet
- Yolo v3

Hyper-parameters:

- **Image size**
- **Number of proposal**
- **Anchor boxes**

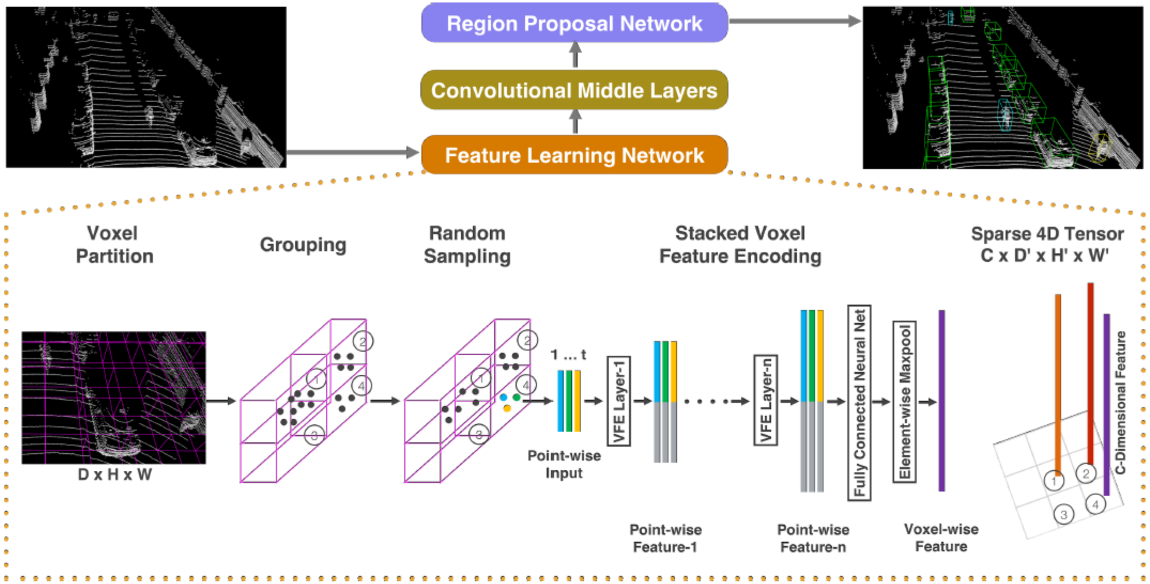
3D Object Detection





3D Object Detection

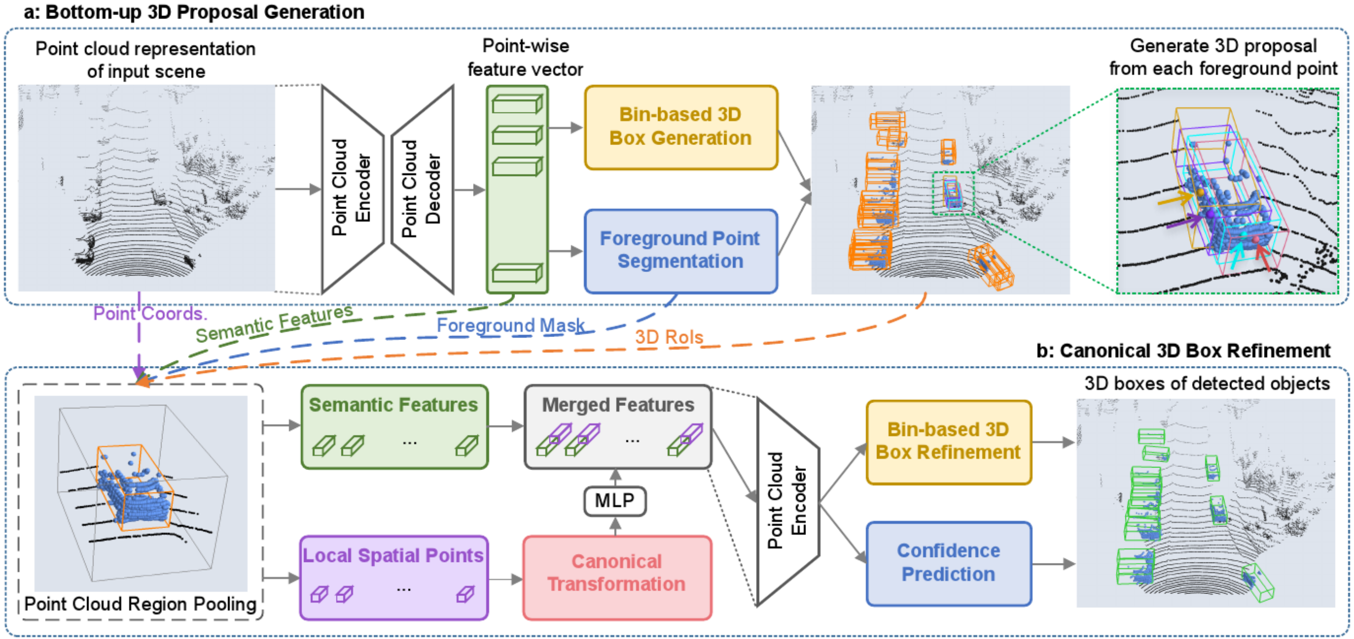
Voxel based approaches: Voxelnet



[VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection, CVPR 2018]

3D Object Detection

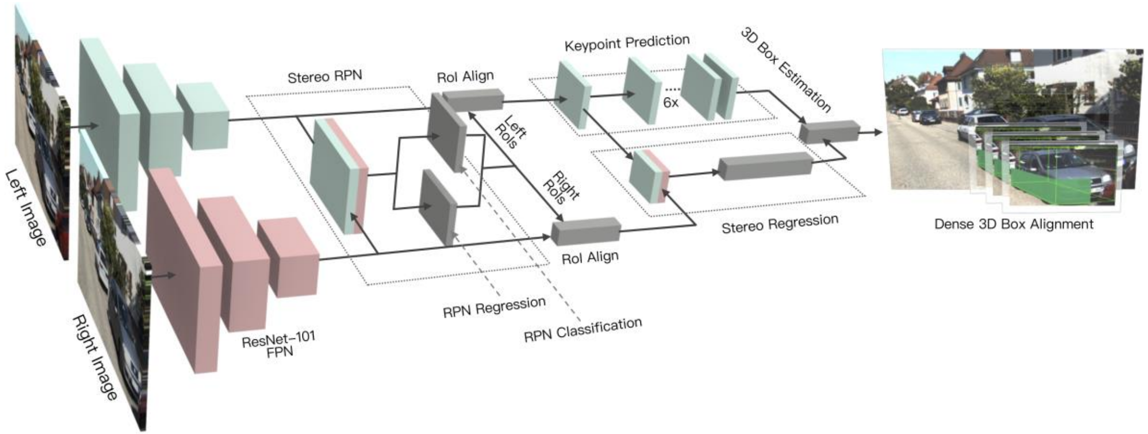
Point based approach: Point RCNN



[PointRCNN:3D Object Proposal Generation and Detection from Point Cloud, CVPR 2019]

3D Object Detection

Pixel based approach: Stereo RCNN



[Stereo R-CNN based 3D Object Detection for Autonomous Driving, CVPR 2019]()

3D Object Detection

3D Object Detection with Monocular Camera: M3D-RPN

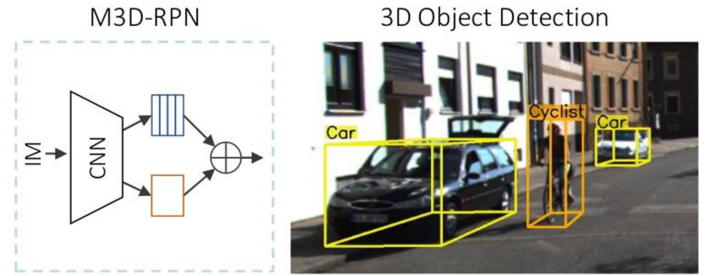


Figure 1. M3D-RPN uses a *single* monocular 3D region proposal network with global convolution (orange) and local depth-aware convolution (blue) to predict multi-class 3D bounding boxes.

[M3D-RPN: Monocular 3D Region Proposal Network for Object Detection]

Applications of Object Detection

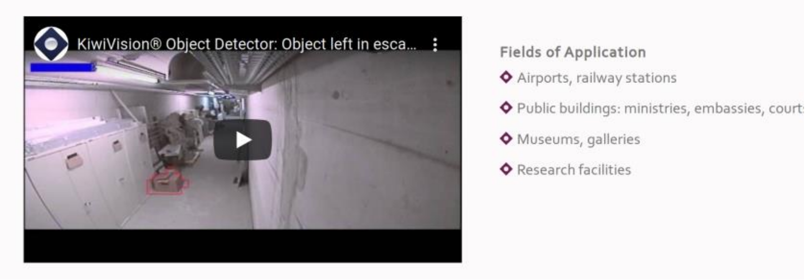
- **Robotics**

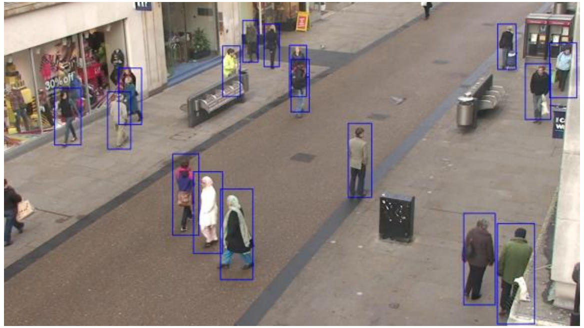
Self driving cars

Warehouse robots

Applications of Object Detection

- **Robotics**
  - Self driving cars
  - Warehouse robots
- **Security**
  - Surveillance





Applications of Object Detection

- **Robotics**
  - Self driving cars
  - Warehouse robots
- **Security**
  - Surveillance
- **Agriculture**

Article

Detection of Cattle Using Drones and Convolutional Neural Networks

Alberto Rivas [1], Pablo Chamoso [1,*], Alfonso González-Briones [1] and Juan Manuel Corchado [1,2,3]

[1] BISITE Digital Innovation Hub, University of Salamanca, Edificio Multiusos I+D+i, Calle Espejo 2, 37007 Salamanca, Spain; rivis@usal.es (A.R.); alfonsogb@usal.es (A.G.-B.); corchado@usal.es (J.M.C.)

[2] Department of Electronics, Information and Communication, Faculty of Engineering, Osaka Institute of Technology, Osaka 535-8585, Japan

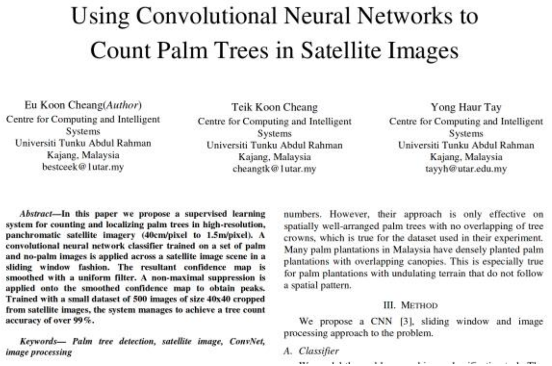
[3] Pusat Komputeran dan Informatik, Universiti Malaysia Kelantan, Karung Berkunci 36, Pengkaan Chepa, Kota Bharu 16100, Malaysia

* Correspondence: chamoso@usal.es; Tel.: +34-677-522-675

Received: 29 May 2018; Accepted: 25 June 2018; Published: 27 June 2018

check for updates

Abstract: Multirotor drones have been one of the most important technological advances of the last decade. Their mechanics are simple compared to other types of drones and their possibilities in



Using Convolutional Neural Networks to Count Palm Trees in Satellite Images

Eu Koon Cheang(*Author*)
Centre for Computing and Intelligent Systems
Universiti Tunku Abdul Rahman
Kajang, Malaysia
bestceek@1utar.my

Teik Koon Cheang
Centre for Computing and Intelligent Systems
Universiti Tunku Abdul Rahman
Kajang, Malaysia
cheangtk@1utar.my

Yong Haur Tay
Centre for Computing and Intelligent Systems
Universiti Tunku Abdul Rahman
Kajang, Malaysia
tayyh@utar.edu.my

***Abstract*—In this paper we propose a supervised learning system for counting and localizing palm trees in high-resolution, panchromatic satellite imagery (40cm/pixel to 1.5m/pixel). A convolutional neural network classifier trained on a set of palm and no-palm images is applied across a satellite image scene in a sliding window fashion. The resultant confidence map is smoothed with a uniform filter. A non-maximal suppression is applied onto the smoothed confidence map to obtain peaks. Trained with a small dataset of 500 images of size 40x40 cropped from satellite images, the system manages to achieve a tree count accuracy of over 99%.**

***Keywords*— Palm tree detection, satellite image, ConvNet, image processing**

numbers. However, their approach is only effective on spatially well-arranged palm trees with no overlapping of tree crowns, which is true for the dataset used in their experiment. Many palm plantations in Malaysia have densely planted palm plantations with overlapping canopies. This is especially true for palm plantations with undulating terrain that do not follow a spatial pattern.

III. METHOD

We propose a CNN [3], sliding window and image processing approach to the problem.

A. Classifier

Applications of Object Detection

- Robotics
  - Self driving cars
  - Warehouse robots
- Security
  - Surveillance
- Agriculture
  - Crop monitoring
  - Cattle counting

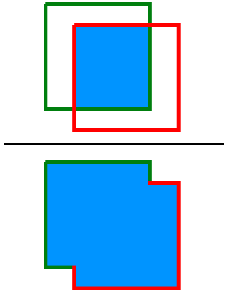
And many more ...

Are state of art object detectors enough?

How to evaluate object detectors - <u>Detection</u>:

IOU (Intersection Over Union) :

$$\frac{Area\ of\ overlap}{Area\ of\ union}$$



Are state of art object detectors enough?

How to evaluate object detectors - <u>Classification</u>

- **Accuracy:** $\dfrac{TP+TN}{TP+TN+FP+FN} = \dfrac{\text{Correctly classified objects}}{\textit{Total number of objects}}$

- **Precision:** $\dfrac{TP}{TP+FP} = \dfrac{\text{True Positives}}{\textit{All Detections}}$

- **Recall:** $\dfrac{TP}{TP+FN} = \dfrac{\textit{True Positives}}{\textit{All ground truth objects}}$

TP : True Positives
FP : False Positives
TN : True Negatives
FN : False Negatives

Are state of art object detectors enough?

How to evaluate object detectors – Multi-object Detection and Classification

```
Average Precision (AP):
   AP            % AP at IoU=.50:.05:.95 (primary challenge metric)
   AP^IoU=.50    % AP at IoU=.50 (PASCAL VOC metric)
   AP^IoU=.75    % AP at IoU=.75 (strict metric)
AP Across Scales:
   AP^small      % AP for small objects: area < 32²
   AP^medium     % AP for medium objects: 32² < area < 96²
   AP^large      % AP for large objects: area > 96²
Average Recall (AR):
   AR^max=1      % AR given 1 detection per image
   AR^max=10     % AR given 10 detections per image
   AR^max=100    % AR given 100 detections per image
AR Across Scales:
   AR^small      % AR for small objects: area < 32²
   AR^medium     % AR for medium objects: 32² < area < 96²
   AR^large      % AR for large objects: area > 96²
```

Are state of art object detectors enough?

How to evaluate object detectors –
Multi-object Detection and Classification

AP (Average Precision) :

Area under the precision-recall curve.
AP is calculated by interpolating points in the Precision-Recall curve.

Are state of art object detectors enough?

How to evaluate object detectors –
Multi-object Detection and Classification

AP (Average Precision) : Area under the precision-recall curve. AP is calculated by interpolating points in the Precision-Recall curve.

mAP (Mean Average Precision) :
Average of AP across all classes. For multi-class detector metrics (eg: Coco Detection metrics), mAP and AP are used interchangeably.

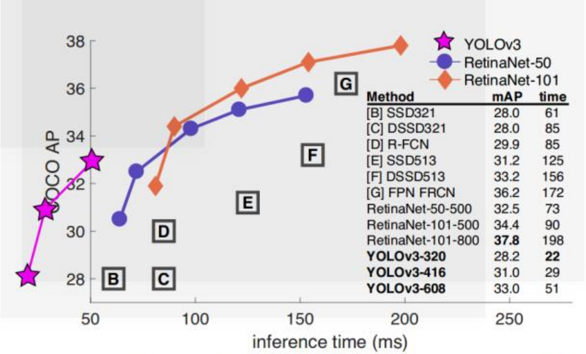
Are state of art object detectors enough?

How to evaluate object detectors – Multi-object Detection and Classification

Coco Detection Metrics

```
Average Precision (AP):
   AP            % AP at IoU=.50:.05:.95 (primary challenge metric)
   AP^IoU=.50    % AP at IoU=.50 (PASCAL VOC metric)
   AP^IoU=.75    % AP at IoU=.75 (strict metric)
AP Across Scales:
   AP^small      % AP for small objects: area < 32²
   AP^medium     % AP for medium objects: 32² < area < 96²
   AP^large      % AP for large objects: area > 96²
Average Recall (AR):
   AR^max=1      % AR given 1 detection per image
   AR^max=10     % AR given 10 detections per image
   AR^max=100    % AR given 100 detections per image
AR Across Scales:
   AR^small      % AR for small objects: area < 32²
   AR^medium     % AR for medium objects: 32² < area < 96²
   AR^large      % AR for large objects: area > 96²
```

Are state of art object detectors enough?



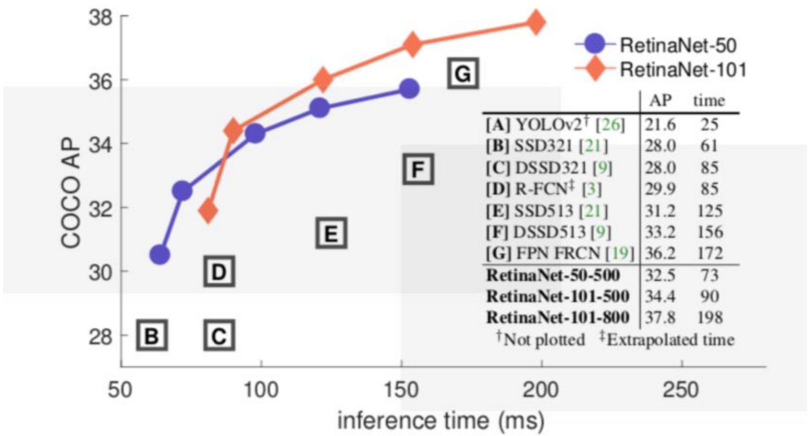
Method	mAP	time
[B] SSD321	28.0	61
[C] DSSD321	28.0	85
[D] R-FCN	29.9	85
[E] SSD513	31.2	125
[F] DSSD513	33.2	156
[G] FPN FRCN	36.2	172
RetinaNet-50-500	32.5	73
RetinaNet-101-500	34.4	90
RetinaNet-101-800	**37.8**	198
YOLOv3-320	28.2	**22**
YOLOv3-416	31.0	29
YOLOv3-608	33.0	51

Figure 1. We adapt this figure from the Focal Loss paper [9]. YOLOv3 runs significantly faster than other detection methods with comparable performance. Times from either an M40 or Titan X, they are basically the same GPU.

2D Object Detection AP = 38

[YOLOv3: An Incremental Improvement]

Are state of art object detectors enough?



Method	AP(IoU=0.7) Easy	Moderate	Hard
MV3D [4]	71.29	62.68	56.56
VoxelNet [43]	81.98	65.46	62.85
SECOND [40]	87.43	76.48	69.10
AVOD-FPN [14]	84.41	74.44	68.65
F-PointNet [25]	83.76	70.92	63.65
Ours (no GT-AUG)	88.45	77.67	76.30
Ours	**88.88**	**78.63**	**77.38**

Table 2. Performance comparison of 3D object detection with previous methods on the car class of KITTI *val* split set.

2D Object Detection AP (all classes) = 38

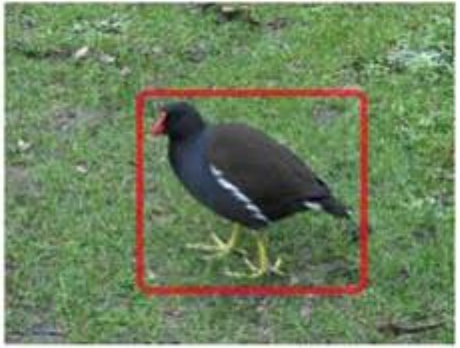
3D Object Detection AP (car only) = 77.38

Lin et al, Focal Loss for Dense Object Detection

PointRCNN:3D Object Proposal Generation and Detection from Point Cloud, CVPR 2019

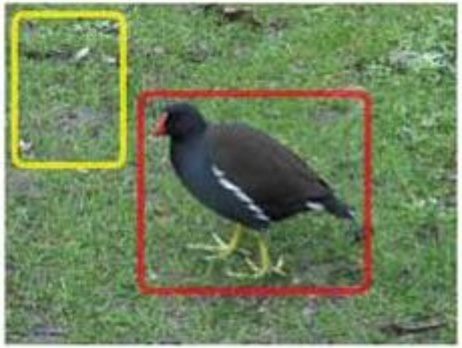
Failure Modes of Object Detectors

- Missed detection



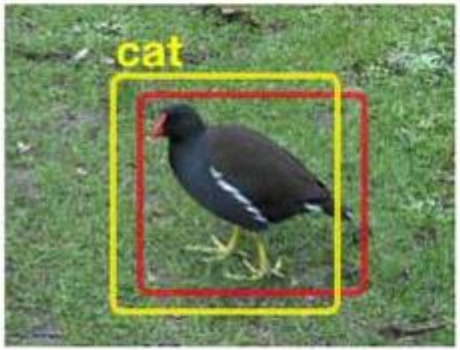
Failure Modes of Object Detectors

- Missed detection
- False detection



Failure Modes of Object Detectors

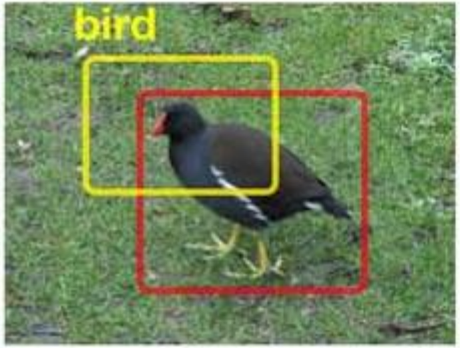
- Missed detection
- False detection
- False classification



wrong class

Failure Modes of Object Detectors

- Missed detection
- False detection
- False classification
- Partial detection



IOU < 0.5

Are state of art object detectors enough?

Tesla that crashed into police car was in 'autopilot' mode, California official says

If confirmed, it would be the third time a Tesla in autopilot has crashed into a stationary emergency vehicle this year

▲ Photo provided by Laguna Beach police shows a Tesla sedan that crashed into a parked police cruiser on Tuesday. Photograph: AP

A Tesla car operating in "autopilot" mode crashed into a stationary police car in Laguna Beach, California, leaving the driver injured and the patrol vehicle "totalled", according to an official.

NTSB Report implies serious fault for Uber in fatality

Submitted by brad on Thu, 2018-05-24 11:19

Topic:
Robocars

The NTSB has released its preliminary report on the fatality involving the Uber prototype self driving car. The NTSB does not attempt to assign blame, but there are some damning facts in the report.

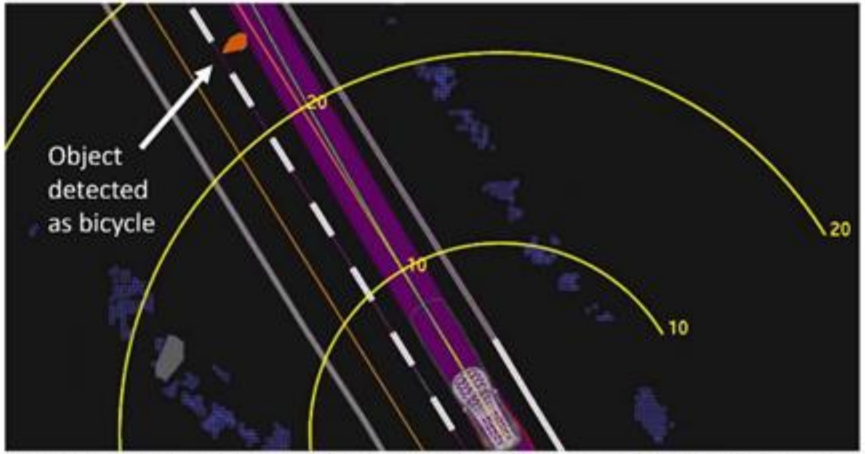


Figure 2. View of the self-driving system data playback at about 1.3 seconds before impact, when the system determined an emergency braking maneuver would be needed to mitigate a collision. Yellow bands are shown in meters ahead. Orange lines show the center of mapped travel lanes. The purple shaded area shows the path the vehicle traveled, with the green line showing the center of that path.

Are state of art object detectors enough?

In safety critical system, it's better if downstream system knows that the network is unsure.

If system had identified its own uncertainty:

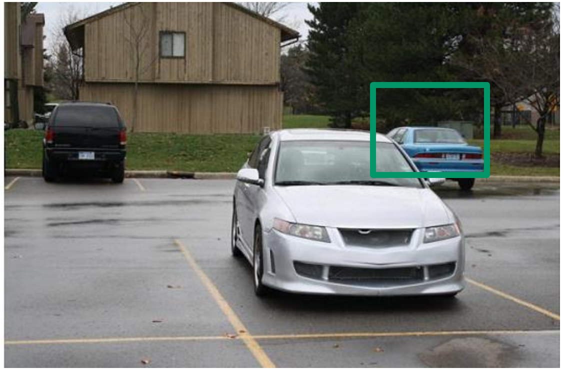
- alert user to take control over steering
- propagate uncertainty for more conservative decision making

http://bdl101.ml/MLSS_2019_BDL_1.pdf

Sources of Uncertainty

Errors in training data - **Aleatoric Uncertainty**

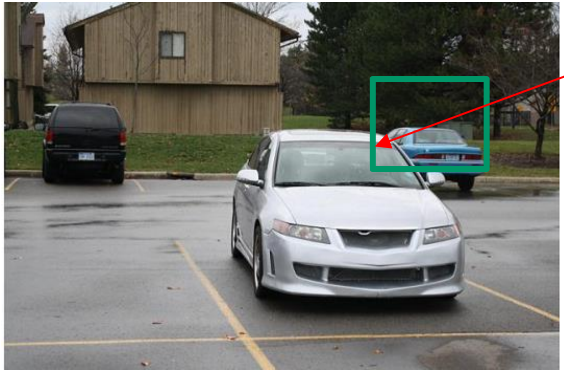
- Ambiguous data
  - Eg: occlusions



Sources of Uncertainty

Errors in training data - **Aleatoric Uncertainty**

- Ambiguous data
  - Eg: occlusions



Difficult to identify where this edge should be

Sources of Uncertainty

Errors in training data - **Aleatoric Uncertainty**

- Ambiguous data (Eg: occlusions)
- Inaccurate annotations

Sources of Uncertainty

Errors in training data - **Aleatoric Uncertainty**

- Ambiguous data (Eg: occlusions)
- Inaccurate annotations

Person, table labeled as aeroplane

Sources of Uncertainty

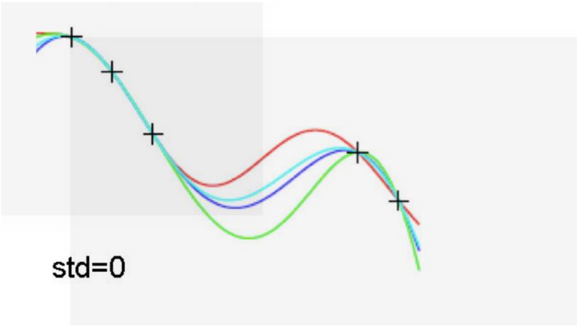
Errors in training data - **Aleatoric Uncertainty**

- Ambiguous data (Eg: occlusions)
- Inaccurate annotations
- Real world has sensor noise, environmental noise

Sources of Uncertainty

Uncertainty from training model - **Epistemic Uncertainty**

- Data can be explained by multiple models



http://bdl101.ml/MLSS_2019_BDL_1.pdf

Sources of Uncertainty

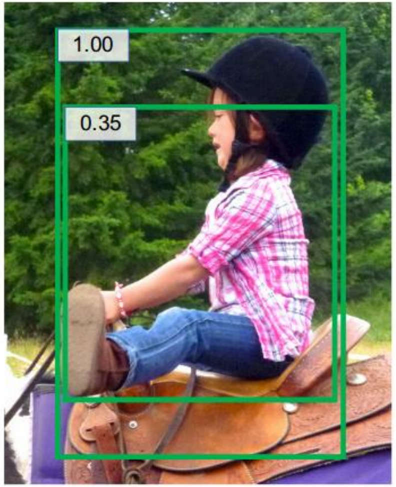
Uncertainty from training model - **Epistemic Uncertainty**

- Data can be explained by multiple models

- Training sample will not have all possible scene

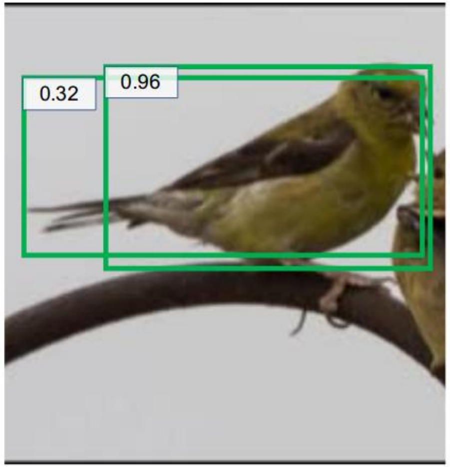
Uncertainty in Object Detection

Regression layers -> NMS step reduces bounding boxes to a few



(a)

Both bounding boxes do not perfectly fit the detected object

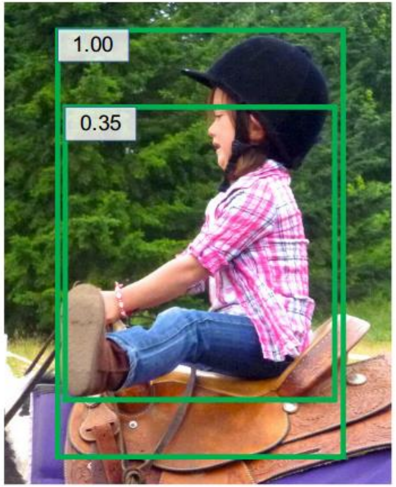


(b)

[He et al, Bounding Box Regression with Uncertainty for Accurate Object Detection]()

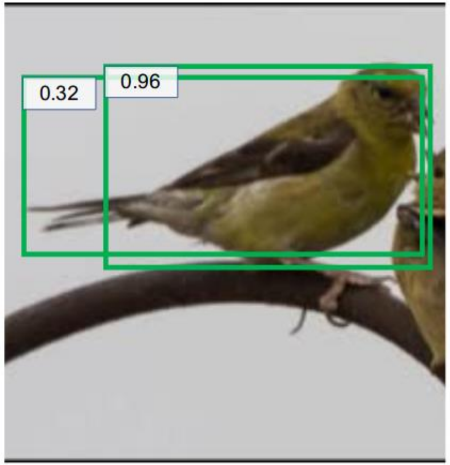
Uncertainty in Object Detection

Regression layers -> NMS step reduces bounding boxes to a few



(a)

Both bounding boxes do not perfectly fit the detected object



(b)

Uncertainty Information is lost.

[He et al, Bounding Box Regression with Uncertainty for Accurate Object Detection]

Uncertainty in Object Detection

Classification layer -> outputs class vector with probability scores -> but these scores do not capture real uncertainty from the network. Easy to fool

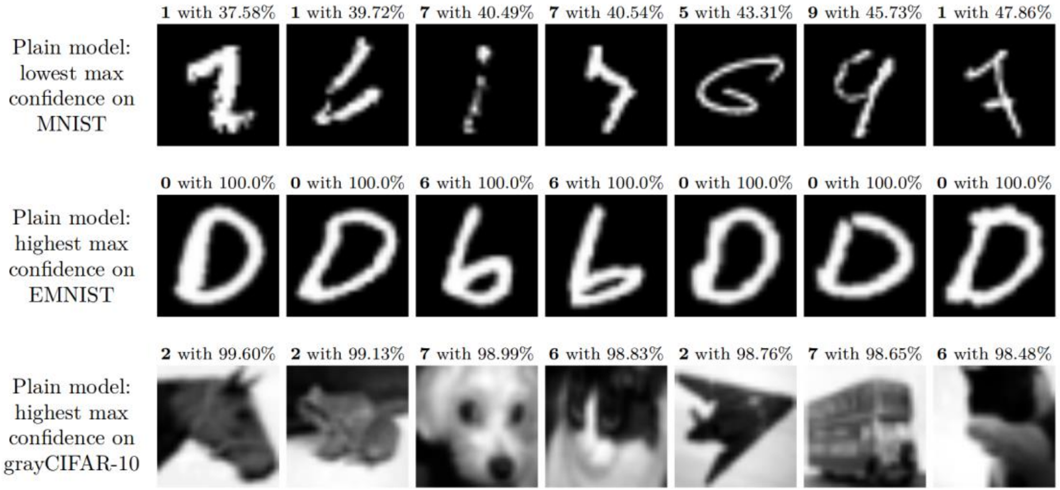


Figure 5: Top Row: predictions of the plain MNIST model with the lowest maximum confidence. Middle Row: predictions of the plain MNIST model on letters 'a', 'b', 'c', 'd' of EMNIST with the highest maximum confidence. Bottom Row: predictions of the plain MNIST model on the grayscale version of CIFAR-10 with the highest maximum confidence. Note that although the predictions on EMNIST are mostly justified, the predictions on CIFAR-10 are overconfident on the images that have no resemblance to digits.

[Hein et al, Why Relu Networks yield high confidence predictions away from training data]()

Learning Uncertainty —Basics

Bayesian Probability Modeling :Explicitly modeling the assumption we made while developing ML algorithms

Learning Uncertainty –Basics

Bayesian Probability Modeling: Explicitly modeling the assumption we made while developing ML algorithms

Example:

- How was the data generated?
- How was the observation/inference generated?

Recap —Bayesian Learning

Bayes' Rule in probabilistic modeling:

$$\textit{Posterior} \propto \textit{Prior x Likelihood}$$

- **Prior :** What I think the data looks like
- **Likelihood :** How I believe the data was generated given my prior
- **Posterior :** Given the data, this is what I currently think the distribution could be, and I need more data to be certain

Recap –Bayesian Learning

Bayes' Rule in probabilistic modeling:

Posterior ∝ Prior x Likelihood

- **Prior :** What I think the data looks like
- **Likelihood :** How I believe the data was generated given my prior
- **Posterior :** Given the data, this is what I currently think the distribution could be, and I need more data to be certain

Uncertainty is the inverse of belief -> can come from the posterior

Learning Uncertainty - NN as probabilistic model

- Neural Network is a probabilistic model $p(\mathbf{y}|\mathbf{x}, \boldsymbol{\omega})$

Maximum Likelihood Estimate

- We learn W using MLE:

$$w^{MLE} = argmax_w logP(D|w)$$
$$= argmax_w \sum_i logP(y_i|x_i, w)$$

Maximum a Posterior Estimate

- Adding regularization using MAP:

$$w^{MAP} = argmax_w logP(w|D)$$
$$= argmax_w logP(D|w) + logP(w)$$

[PyData Tel Aviv Meetup: Uncertainty in Deep Learning - Inbar Naor]

Recap —Bayesian NN

- Instead of one set of weights, let's find probability over the weights
- Start with some prior -> Initialization
- While learning, update posterior based on training data
- Following Bayes' Rule:

$$p(y|x, X, Y) = \int p(y|x, w)p(w|X, Y)dw$$

Recap –Bayesian NN

- Instead of one set of weights, let's find probability over the weights
- Start with some prior -> Initialization
- While learning, update posterior based on training data
- Following Bayes' Rule:

$$p(y|x, X, Y) = \int p(y|x, w)p(w|X, Y)dw$$

Intractable

Learning Uncertainty — Sampling Methods

- Use dropout during inference
- Dropout approximately integrates over model parameters
- Run multiple inferences with varying dropout:
  - **Prediction** = Mean of all the output values
  - **Uncertainty** = Variance of all the output values

[Gal and Gharmani, Dropout as a Bayesian Approximation]

Learning Uncertainty – Sampling Methods

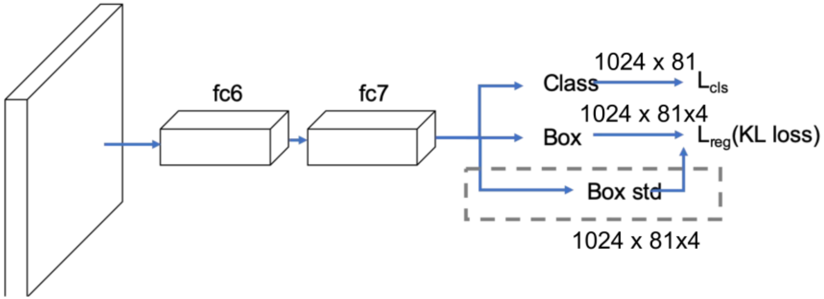
- Use dropout during inference
- Dropout approximately integrates over model parameters
- Run multiple inferences with varying dropout:
  - Prediction = Mean of all the output values
  - Uncertainty = Variance of all the output values

This helps us to learn Model Uncertainty

Gal and Gharmani, Dropout as a Bayesian Approximation

Learning Uncertainty – Data Uncertainty (Regression)

- Learn variance as an additional output node



- Modify regression loss with KL loss

$$L_{reg} \propto \frac{(x_g - x_e)^2}{2\sigma^2} + \frac{1}{2}\log(\sigma^2)$$

[Le et al, Uncertainty Estimation for Deep Neural Object Detectors inSafety-Critical Applications]

Learning Uncertainty – Evidential Theory (Classification)

- Dempster–Shafer Theory of Evidence (DST) is a generalization of the Bayesian theory tosubjective probabilities
- Subjective Logic is a formalization of DST
- For a set of possible class labels k= 1,...,K SL considers a belief mass of $\mathbf{b_k}$ and overall uncertainty $\mathbf{u}$ such that

$$u + \sum_{k=1}^{K} b_k = 1, \qquad b_k >= 0, u >= 0$$

Sensoy et al, Evidential Deep Learning to Quantify ClassificationUncertainty

Related Papers: Uncertainty in Object Detection

- Le et al, Uncertainty Estimation for Deep Neural Object Detectors inSafety-Critical Applications
- Harkeh et al, BayesOD: A Bayesian Approach for Uncertainty Estimation in DeepObject Detectors
- He et al, Bounding Box Regression with Uncertainty for Accurate Object Detection
- Kraus and Dietmayer, Uncertainty Estimation in One-Stage Object Detection
- The Robotic Vision Probabilistic Object Detection Challenge , CVPR 2019 Workshop

Resources on Uncertainty in Deep Learning

- Yarin Gal's Tutorial
- Yarin Gal's Thesis
- Uncertainty in Deep Learning – PyData Tel Aviv Meetup