SqueezeDet Report

Zhao Zhixu

<2017-09-24 Sun>

Outline

Background

- Recent CNN research focused on improving accuracy
- Actual deployment: some other issues are equally critical

Motivation

object detection for autonomous driving requires

- realtime speed
- small model size
- energy efficiency to enable embedded system deployment
- high accuracy to ensure safety

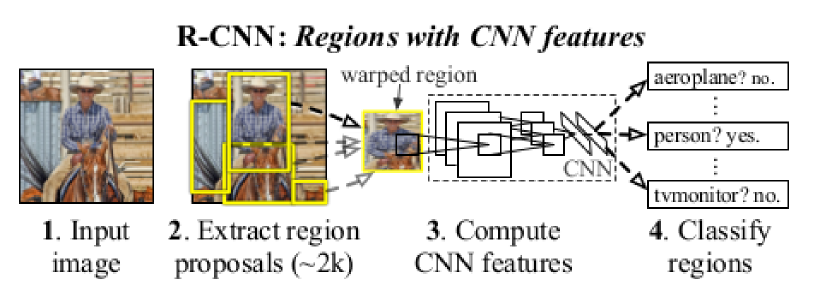
CNN framework for object detection

CNNs for object detection

- R-CNN (Region-based Convolutional Neural Networks)
- Faster R-CNN
- YOLO (You Look Only Once)
- SqueezeDet

R-CNN

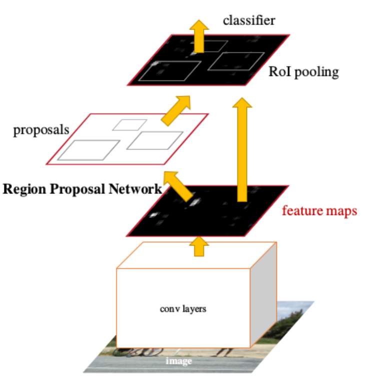
- Selective search for region proposals
- CNN to compute features for each region proposal independently
- Basic module for CNN object detection
- Complex pipline, highly overfit, 40 seconds per image



CNN framework for object detection

Fast/Faster R-CNN

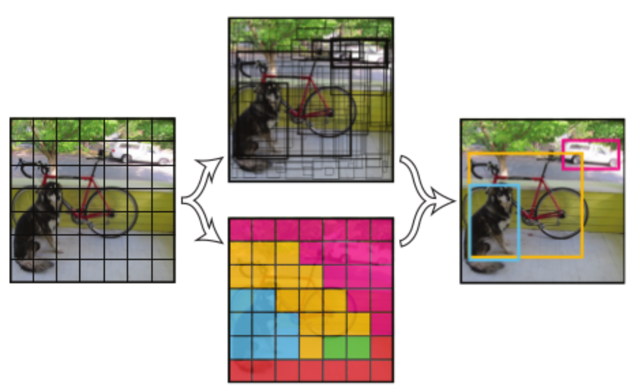
- Speed up R-CNN by sharing CNN computation
- Fast R-CNN still relies on selective search, takes 2 seconds per image
- Faster R-CNN proposed RPN (Region Proposal Network)



CNN framework for object detection

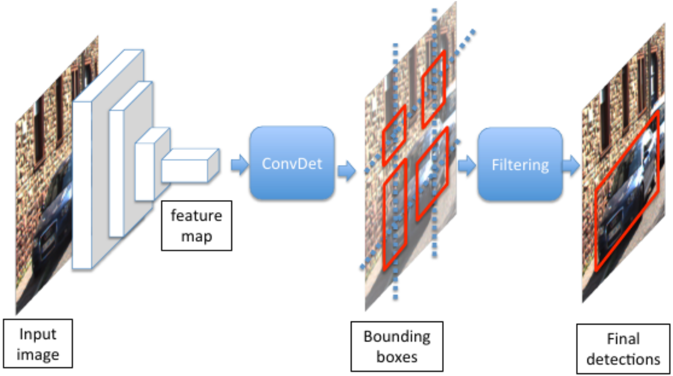
YOLO

- Region proposal and classification are integrated into one single stage
- First CNN object detection model that achieved real-time speed



Inspired by YOLO, SqueezeDet also adopts a single-stage detection pipeline: region proposition and classification is performed by one single network simultaneously.

- SqueezeNet
- ConvDet
- NMS Filter



SqueezeNet

SqueezeNet is the backbone CNN of SqueeDet. It extracts feature maps to ConvDet for region proposal and classification. The focus is mainly on model size and energy efficiency. Smaller model size leads to less DRAM access, leads to less energy consume.

Why SqueezeNet?

Model	Size	Top-5 accuracy
AlexNet	240MB	80.3%
VGG-19	575MB	87%
SqueezeNet	4.8MB	80.3%
GoogleLeNet	53MB	93.3%

SqueezeNet Design Strategies

1. Replace 3x3 filters with 1x1 filters
2. Decrease the number of input channels to 3x3 filters
3. Downsample late in the network

SqueezeNet

Implementation

1. Proposed Fire Module
   comprised of a squeeze layer of 1x1 filters and a expand layer of 1x1 and 3x3 filters
2. Limit the number of input channels to the 3x3 filters
3. Strides greater than 1 are concentrated toward the end of the network

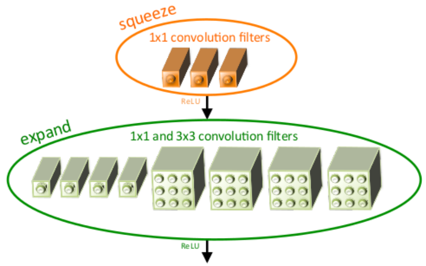
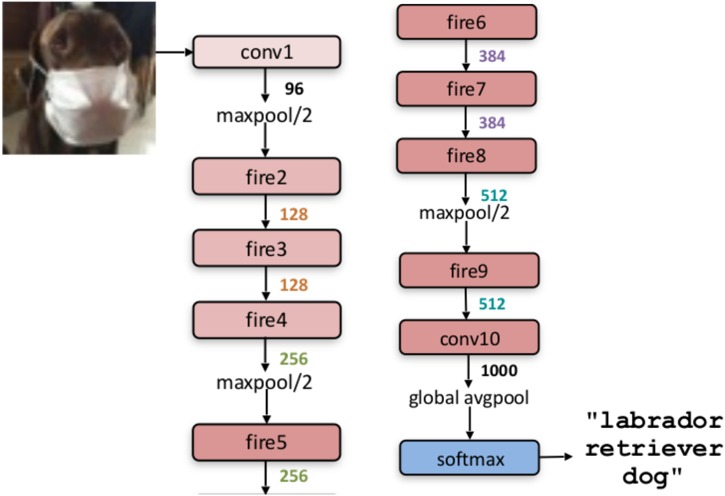


Figure: Fire Module in SqueezeNet

SqueezeNet Architecture



ConvDet

ConvDet is a convolutional layer that is trained to output bounding box coordinates and class probabilities, added to SqueezeNet.
ConvDet is similar to the last layer of RPN in Faster R-CNN.

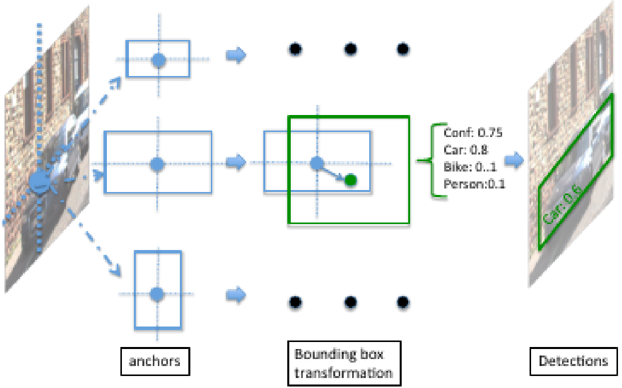


Figure: ConvDet

ConvDet Output

Notation

- $W \times H$: feature map size
- K: the number of pre-selected bounding boxes (achors) at each position
- $(\hat{x}_i, \hat{y}_j, \hat{w}_k, \hat{h}_k), i \in [1, W], j \in [1, H], k \in [1, K]$: pre-selected anchor coordinate
- $(x_i^P, h_j^P, w_k^P, h_k^P)$: proposed anchor coordinates
- $(\delta x_{i,j,k}, \delta y_{i,j,k}, \delta w_{i,j,k}, \delta h_{i,j,k},)$: relative coordinates, trainable parameters to compute proposals

For each anchor (i, j, k), it computes 4 proposed coordinates as follows:

$$x_i^P = \hat{x}_i + \hat{w}_k \delta x_{ijk} \quad y_j^P = \hat{y}_j + \hat{h}_k \delta y_{ijk}$$
$$w_k^P = \hat{w}_k exp(\delta w_{ijk}) \quad h_K^P = \hat{h}_k exp(\delta h_{ijk})$$

ConvDet Output

Besides 4 proposed coordinates above, it also computes conditional class probabilities and confidence score indicating how likely the bounding box actually contain an object.

Notation

- C: number of class to distinguish
- $Pr(Object)$: probability that an object of interest does exist
- $Pr(class_c|Object), c \in [1, C]$: conditional class probability
- IOU^{pred}_{truth}: intersection rate between ground-truth and predicted bounding box

It assigns the label with the highest conditional class probability $Pr(class_c|Object)$ to the bounding box and the confidence score is computed as

$$\max_c Pr(class_c|Object) \times Pr(Object) \times IOU^{pred}_{truth}$$

Non-Maximum Supression Filter

The *Non-Maximum Supression Filter (NMS)* is the last step to give final results.
It keeps the top N bounding boxes with the highest confidence scores and drops other redundant bounding boxes to obtain the final detections.

Training Protocol

Since SqueezeDet only has one single neural network, it can be trained end-to-end.

To train the ConvDet layer to learn detection, localization and classification, it has a multi-task loss function:

$$\frac{\lambda_{bbox}}{N_{obj}} \sum_{i=1}^{W} \sum_{j=1}^{H} \sum_{k=1}^{K} I_{ijk} [(\delta x_{ijk} - \delta x_{ijk}^{G})^2 + (\delta y_{ijk} - \delta y_{ijk}^{G})^2 + (\delta w_{ijk} - \delta w_{ijk}^{G})^2$$

$$+ (\delta h_{ijk} - \delta h_{ijk}^{G})^2] + \sum_{i=1}^{W} \sum_{j=1}^{H} \sum_{k=1}^{K} \frac{\lambda_{conf}^{+}}{N_{obj}} I_{ijk} (\gamma_{ijk} - \gamma_{ijk}^{G})^2 + \frac{\lambda_{conf}^{-}}{WHK - N_{obj}} \bar{I}_{ijk} \gamma_{ijk}^{2}$$

$$+ \frac{1}{N_{obj}} \sum_{i=1}^{W} \sum_{j=1}^{H} \sum_{k=1}^{K} \sum_{c=1}^{C} I_{ijk} l_{c}^{G} log(p_c)$$

Training Protocol

Part 1 - Bounding box regression

$$\frac{\lambda_{bbox}}{N_{obj}} \sum_{i=1}^{W} \sum_{j=1}^{H} \sum_{k=1}^{K} I_{ijk}[(\delta x_{ijk} - \delta x_{ijk}^G)^2 + (\delta y_{ijk} - \delta y_{ijk}^G)^2 + (\delta w_{ijk} - \delta w_{ijk}^G)^2 + (\delta h_{ijk} - \delta h_{ijk}^G)^2]$$

- λ_{bbox}: hyper-parameter, set to 5 empirically
- I_{ijk}: equals 1 if anchor at (i, j, k) has the largest overlap (IOU) with a ground truth box, else 0
- $(\delta x_{ijk}^G, \delta y_{ijk}^G, \delta w_{ijk}^G, \delta h_{ijk}^G)$: ground truth bounding box coordinates, computed as follows (inverse of previous equations):

$$\delta x_{ijk}^G = (x^G - \hat{x}_i)/\hat{w}_k \quad \delta y_{ijk}^G = (y^G - \hat{y}_j)/\hat{h}_k$$
$$\delta w_{ijk}^G = log(w^G/\hat{w}_k) \quad \delta h_{ijk}^G = log(h^G/\hat{h}_k)$$

Training Protocol

Part2 - Confidence score regression

$$\sum_{i=1}^{W}\sum_{j=1}^{H}\sum_{k=1}^{K}\frac{\lambda_{conf}^{+}}{N_{obj}}I_{ijk}(\gamma_{ijk}-\gamma_{ijk}^{G})^2+\frac{\lambda_{conf}^{-}}{WHK-N_{obj}}\bar{I}_{ijk}\gamma_{ijk}^{2}$$

- γ_{ijk}: ConvDet-predicted confidence score for anchor-k at position (i, j)
- γ_{ijk}^{G}: obtained by computing IOU_{truth}^{pred}
- $\bar{I}_{ijk}\gamma_{ijk}^{2}$: $\bar{I}_{ijk} = 1 - I_{ijk}$, penalize the confidence score with this item for anchors that are not "responsible" for the detection
- $\lambda_{conf}^{+}, \lambda_{conf}^{-}$: empirical hyper-parameters to balance the influence of lots of anchors that are not assigned to any object, set to 75 and 100
- Feed ConvDet output into a *sigmoid* function to normalize γ_{ijk} in range $[0, 1]$

Training Protocol

Part3 - Classification Cross-entropy Loss

$$\frac{1}{N_{obj}} \sum_{i=1}^{W} \sum_{j=1}^{H} \sum_{k=1}^{K} \sum_{c=1}^{C} I_{ijk} l_c^G log(p_c)$$

- $l_c^G \in \{0, 1\}$: ground truth label
- $p_c \in [0, 1], c \in [1, C]$: probability distribution predicted by the neural net
- Feed ConvDet output into *softmax* to make sure p_c is in range $[0, 1]$

Comparison on the KITTI dataset

Speed vs accuracy

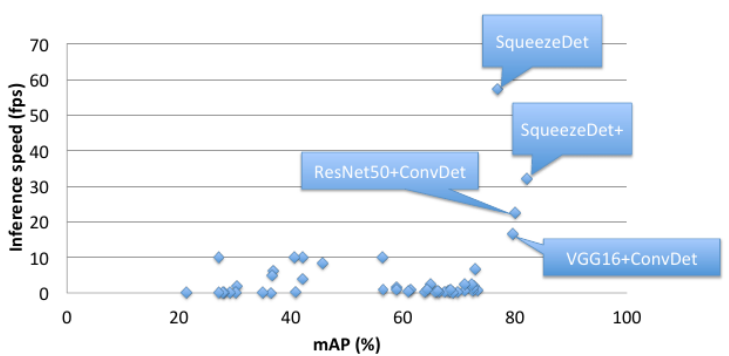


Figure: Inference speed vs mean average precision for cyclist detection

Comparison on the KITTI dataset

Model size vs mean average

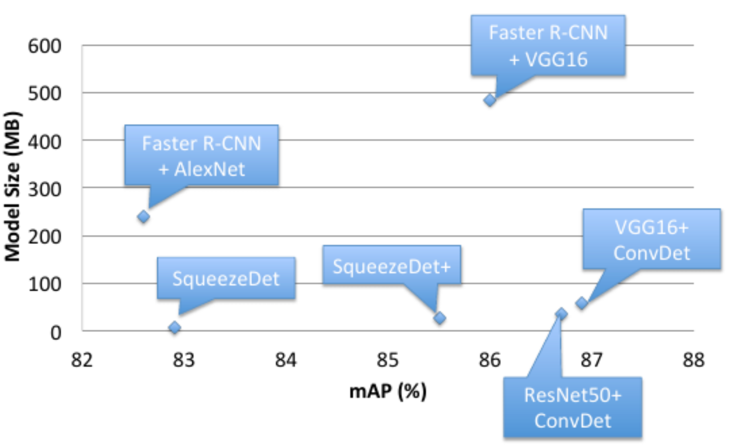


Figure: Model size vs. mean average precision for car detection

References

1. R. B. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In CVPR, 2014.
2. F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer. SqueezeNet: Alexnet-level accu- racy with 50x fewer parameters and <0.5mb model size. arXiv:1602.07360, 2016.
3. S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In NIPS, 2015.
4. J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In CVPR, 2016.
5. B. Wu, F. Iandola, P. H. Jin, K. Keutzer. SqueezeDet: Unified, Small, Low Power Fully Convolutional Neural Networks for Real-Time Object Detection for Autonomous Driving. arXiv:1612.01051, 2016

Question Time

Thank you for listening.

Any questions are welcome.