Literature Review

Philipp Duernay

January 31, 2018

1 Summaries

1.1 Object Detection

1.1.1 Scalable Object Detection using Deep Neural Networks[4]

- Generates number of bounding boxes as object candidates (class agnostic) and confidences for each box
- For each Bounding Box a classifier is run e.g. DNN
- Training: If the number of boxes k is larger than the number of objects b, only b boxes are matched while the confidence of the others is minimized
- Assignment problem

$$F_{match}(x,l) = \frac{1}{2}\sum_{i,j} x_{ij}||l_i - g_j||_2^2$$

where x_ij is one if the ith prediction is assigned to the jth ground truth object

- Confidence:

$$F_{conf}(x,c) = -\sum_{i,j} x_{ij} * \log(c_i) - \sum_i (1 - \sum_j x_{ij}) \log 1 - c_j$$

- Speed up training by clustering (kmeans) of ground truth and using it as prior (prior matching)
- Can be defined to output boxes only for a particular class by training the bounding boxes on that class
- Number of parameters grows linearly with number of classes
- Authors argue two step process (region proposal + classification) is better
- Architecture based on AlexNet
- Predicted boxes are merged using non-maxima surpression
- One shot(50%), +2scales (75%)
- OverFeat/ Selective Search are faster but much more expensive

1.2 Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks[7]

- Introduced Region Proposal Networks, Fully Convolutional Networks that generate region proposals. Feature maps are shared with detection networks and can be trained end-to-end
- Introduces Anchor Boxes:
  - A fully convolutional network predicts a feature map, the feature map is fed into two fully connected sibling networks for box regression and classification(Object/No object)
  - The fully connected layers work in sliding window fashion. For each window location is predicted: 4 coordinates with respect to the anchor box. An anchor box is located at a the center of the window and associated with an aspect ratio
  - (General Remark) Anchor boxes are used to spatially constrain regression heads. Otherwise they would always interfere with each other. What if certain objects only appear at certain locations in the image? Aren't the regression heads then only trained on those objects?
- TODO

1.3 Yolo

TODO

1.4 Yolo v2

- Better
  + Batch Normalization (gets rid of dropout, improves mAP)
  + Higher resolution 416x416, improves mAP
  + Anchor Boxes are predicted by convolutional layer, that predicts offsets. Offsets are easier to learn. Improve in Recall although decrease in accuracy.
  + Center of image should be single box

1.4.1 Single Shot Multibox Detector [6]

- Evaluates feature maps with different scales for all (a few) boxes in the image
- Scale of feature map decreases each layer (feature map with different scales is a key difference to yolo and overfeat)
- Based on VGG-16
- Uses convolutional layers for classification instead of fully connected layers (yolo)
- Similar to other box predictors, the ground truth box has to be chosen for training. Here this is done with the jaccard overlap (¿0.5). Boxes can overlap.
- LOSS FCN

- Uses lower level feature maps in later state for prediction
- Data augmentation significantly increases performance (9%)
- More default boxes is better
- Uses atrous algorithm to cover up holes when changing top layer
- Replaces pooling with feature map at different scales
- Non-maxima surpression at the end to get rid of the big amount of boxes (1.7ms)
- Most time spent in base network and nms
- Default boxes can have different aspect ratios

1.5 (Re-) Localization

1.5.1 PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization [5]

- Relocalizes, is trained on images from the scenes where it is applied
- Accuracy 2m and 3° in 50 km^2 outdoors, 0.5m and 5° indoors, 5ms per frame
- ConvNet 23 layers, Image resolution 224x224
- transfer learning from recognition/classification datasets (ConvNet is trained on classification tasks)
- based on GoogleNet, affine regressors instead of softmax
- automatic training data generation (structure from motion)
- learns p from arbitrary global reference frame
- $loss(I) = ||\hat{x} - x||_2 + \beta * ||\hat{q} - \frac{q}{||q||}||_2$
- separating position/orientation led to drop in performance
- PoseNet evaluation at single center crop + Dense PoseNet 128 uniformly spaced crops (time increase 95ms, only slight accuracy increase)
- Training data generated using structure from motion (Cambridge Scene) and 7 Scenes (Microsoft) for indoor

1.5.2 A Deep Learning Based 6 Degree-of-Freedom Localization Method for Endoscopic Capsule Robots [9]

- not published yet?
- Uses 6-DOF camera pose directly
- based on GoogleNet (9 Inception modules) trained on ImageNet
- $loss(I) = ||\hat{x} - x||_2 + ||\hat{q} - q||_2$
- Dataset of 10 000 frames taken from LM103 - EDG (EsophagoGastroDuodenoscopy) Simulator
- 0.18cm RMSE on a trajectory of 18cm
- Although 3 different cameras are used and the frames are separated for training and testing, its still the same "stomach". With 10 000 frames on a trajectory of 18 cm, won't the system just recognize the position?
- Ground truth determined by seperate cameras

1.6 Object Pose Estimation

1.6.1 3D generic object categorization localization and pose estimation [8]

- Other approaches use different class for different poses
- Object model is separated in different parts of the object based on different view points (front view)
- Different parts are connected when another part is visible from the front view via affine transformation
- Generally such models can't handle inter class variations very good or increase in complexity as number of parts is increased. In this paper this is apparently not the case

1.6.2 Uncertainty-Driven 6D Pose Estimation of Objects and Scenes from a Single RGB Image [1]

- Intermediate representation are object coordinates, continious part labeling that are jointly regressed for every pixel in the image
- Based on auto context (Classifiers with several stages)
- (1) (Auto context) Random forest with L1 regularization predicts labels and object coordinates for every pixel (2) Ransac predicts poses from 2d-3d correspondences guided by uncertainty labels (3) Refinement
- Random forest predicts (probability to belong to object + 3d coordinate—given belonging to object)
- Stacked Forests (Auto context) refine output on previous smoothed output (Geodesic Forest). The smoothing is done to enforce coupling of neighbors
- RANSAC formulates hypothesis by drawing 4 correspondences and solving PnP
- Outperforms PoseNet in indoor localization
- 6D within 5cm and 5 degree only 40 % (With RGB-D 82.5%), on other set 50 % with unknown scene average median error 8.5cm 3.3
- Biggest translational error in z direction
- Multi object detection/pose estimation in 1-4 seconds, not optimized, most time spend in searching for object hypothesis

1.6.3 A Comparative Analysis and Study of Multiview CNN Models for Joint Object Categorization and Pose Estimation[3]

- While detection needs pose invariant features, pose estimation needs the pose
- Single instance 3d model
- Discrete pose approaches (pose as classification)
- Trains pose regressor and classifier on output of different levels to measure quality of features
- Later layers "forget" about pose, paper suggests early branching

1.7 Other

1.7.1 Deformable Convolutional Networks [2]

- Addresses problem of modeling geometric transformations
- Introduces *Deformable Convolution* which adds 2D offsets to the regular sampling grid. The offsets are learned from the data.
- Introduces *Deformable RoI pooling* which adds offsets to bins of pooling layers. The offsets are also learned from the data.
- Further alternatives to have more variable feature maps: Spatial Transformer Networks, Active Convolution, Effective Receptive Field, Atrous Convolution, DeepID-Net, Spatial manipulation in RoI pooling (handcrafted), DPM (handcrafted)
- Light-weight version of STN, easier to train and to integrate
- Receptive fields seem to scale with the size of objects
- Model complexity is increased by only 1-2

Table 1: Object Detection

	Traditional			Deep			
	Viola&Jones	HoG	DPM	R-CNN	YOLO	SSD	OverFeat
Feature Detector	Haar	HoG	Multiple Hogs and virtual springs	Learned by CNN	Learned by CNN		
Detection	Sliding Window, high filter responses indicate there is an object			NN in sliding window detects regions for possible objects, For each proposed region a classification is run	Image is split in Grid each Grid spawns Bounding boxes and gives class probabilities		
Accuracy (voc)				73.2 mAP	63.4 mAP	74.3 mAP	
Speed				7 FPS (Faster-RCNN)	45 FPS	59 FPS	
Strengths							
Weaknesses							

References

[1] Brachmann, E., Michel, F., Krull, A., Yang, M. Y., Gumhold, S., and Rother, C. Uncertainty-Driven 6D Pose Estimation of Objects and Scenes from a Single RGB Image.

[2] Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., and Wei, Y. Deformable Convolutional Networks.

[3] Elhoseiny, M., El-Gaaly, T., Bakry, A., and Elgammal, A. A Comparative Analysis and Study of Multiview CNN Models for Joint Object Categorization and Pose Estimation.

[4] Erhan, D., Szegedy, C., Toshev, A., and Anguelov, D. Scalable Object Detection using Deep Neural Networks.

[5] Kendall, A., Grimes, M., and Cipolla, R. PoseNet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE International Conference on Computer Vision* (may 2015), vol. 2015 Inter, pp. 2938–2946.

[6] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C. SSD: Single Shot MultiBox Detector.

[7] Ren, S., He, K., Girshick, R., and Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks.

[8] Savarese, S., and Fei-Fei, L. 3D generic object categorization, localization and pose estimation.

[9] Turan, M., Almalioglu, Y., Konukoglu, E., and Sitti, M. A Deep Learning Based 6 Degree-of-Freedom Localization Method for Endoscopic Capsule Robots.