Recurrent Neural Networks for object detection

Ahmad Bin Qasim, Arnd Pettirsch (03708414)

Technical University of Munich

Department of Informatics

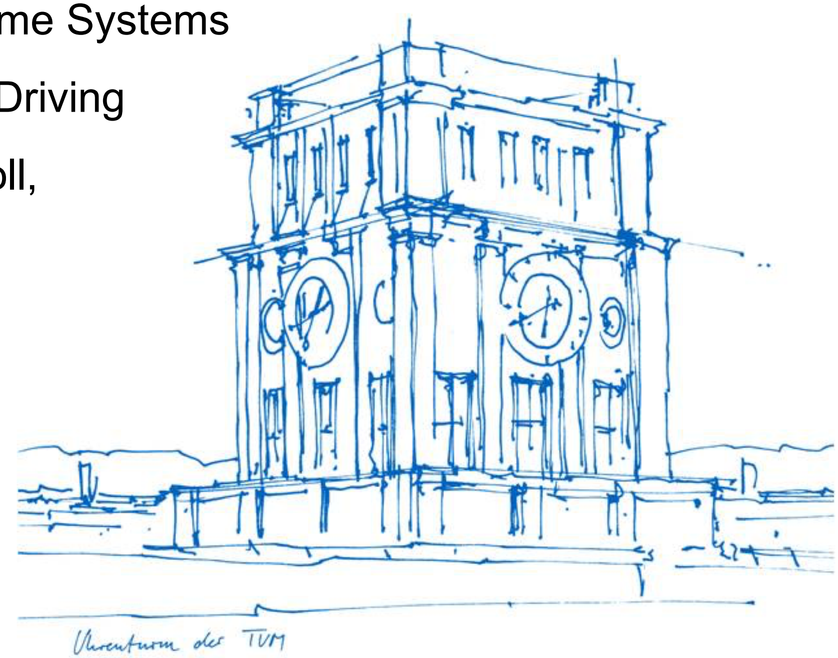
Chair of Robotics, Artificial Intelligence and Real-time Systems

Seminar: Visual Feature Learning in Autonomous Driving

Supervisor: Prof. Dr.-Ing. habil. Alois Christian Knoll,

M.Eng. Emec Ercelik

Garching, June 28th 2019



Agenda

1. Intro
2. Feature-based Video Object Detection
3. Box-Level-based Video Object Detection
4. Flow-based Video Object Detection
5. Comparison of different approaches
6. Outro

Agenda

1. Intro
    1. Image and Video Object Detection in general
    2. Recurrent Neural Networks in general
2. Feature-based Video Object Detection
3. Box-Level-based Video Object Detection
4. Flow-based Video Object Detection
5. Comparison of different approaches
6. Outro

1.1 Image and Video Object Detection

Image object detection history.

- Bayesian methods before deep learning
- ImageNet challenge and VID
- Deep Learning and AlexNet

Single stage and 2-stage image object detectors.

- A two-stage pipeline firstly generates region proposals, which are then classified and refined. (R-CNN, Fast R-CNN, Faster R-CNN). Region Proposal Network.
- A single-stage method is often more efficient but less accurate. Directly regress on bounding boxes and classes. (YOLOv3 and SSD)

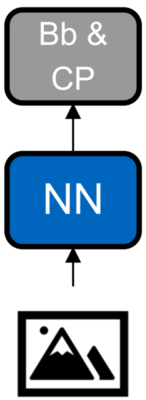
Why is video object detection harder?

- Large size
- Motion blur
- Quality of the dataset
- Partial occlusion
- Unconventional Poses

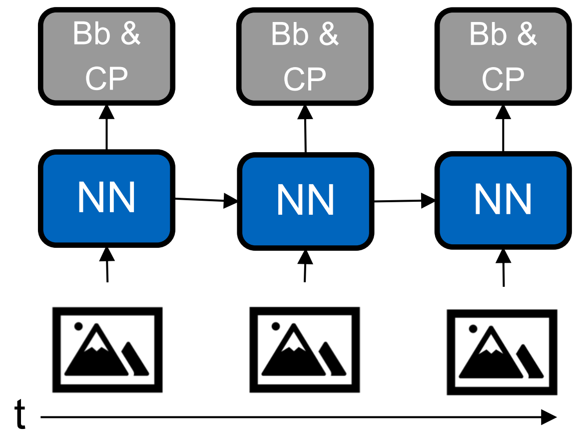
Agenda

1.2 Recurrent Neural Networks

Neural Network:



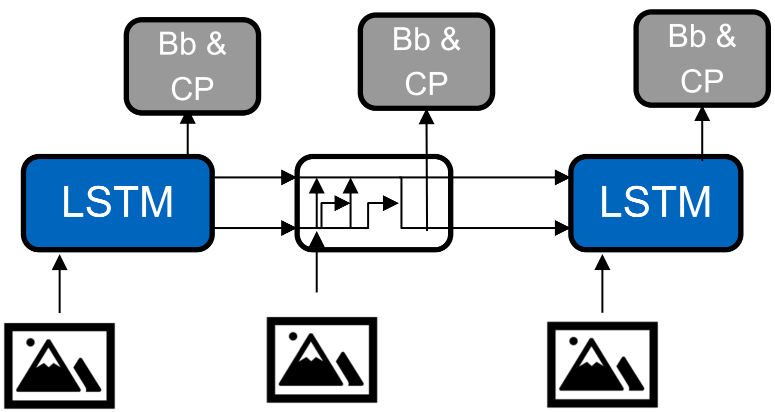
Recurrent Neural Network (RNN):



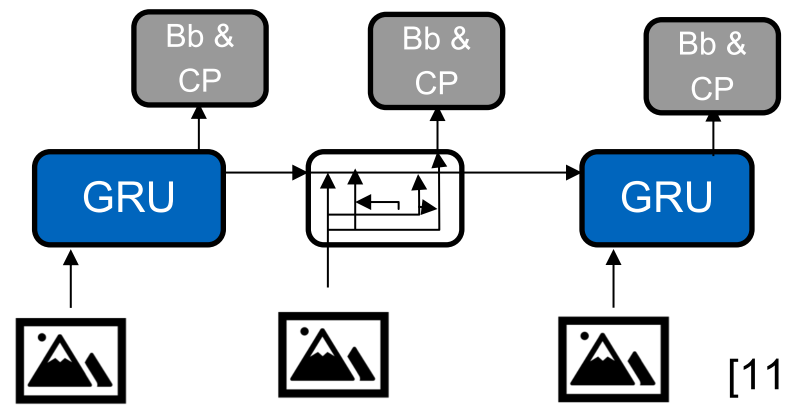
=> Processing sequences of data

Types of Recurrent Neural Networks:

Long-Short Term Memory Units (LSTM)



Gated Recurrent Unit (GRU)

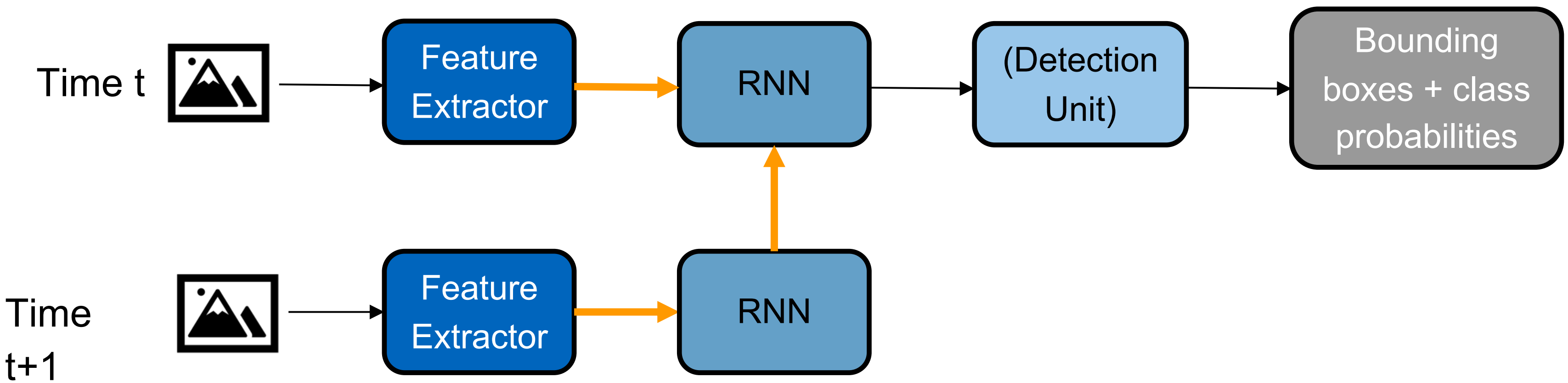


[11] [12] [13]

Agenda

1. Intro
2. Feature-based Video Object Detection
    1. Definition
    2. Recurrent Multi-frame Single Shot Detector for Video Object Detection
    3. Mobile Video Object Detection with Temporally aware Feature Maps
    4. Feature Selective Small Object Detection via Knowledge-based recurrent attentive neural networks
    5. Looking fast and slow: memory-guided mobile video object detection
    6. Delving Deeper into Convolutional Networks for Learning Video Representations
    7. Detect to track and track to detect
3. Box-Level-based Video Object Detection
4. Flow-based Video Object Detection
5. Comparison of different approaches
6. Outro

2.1 Definition

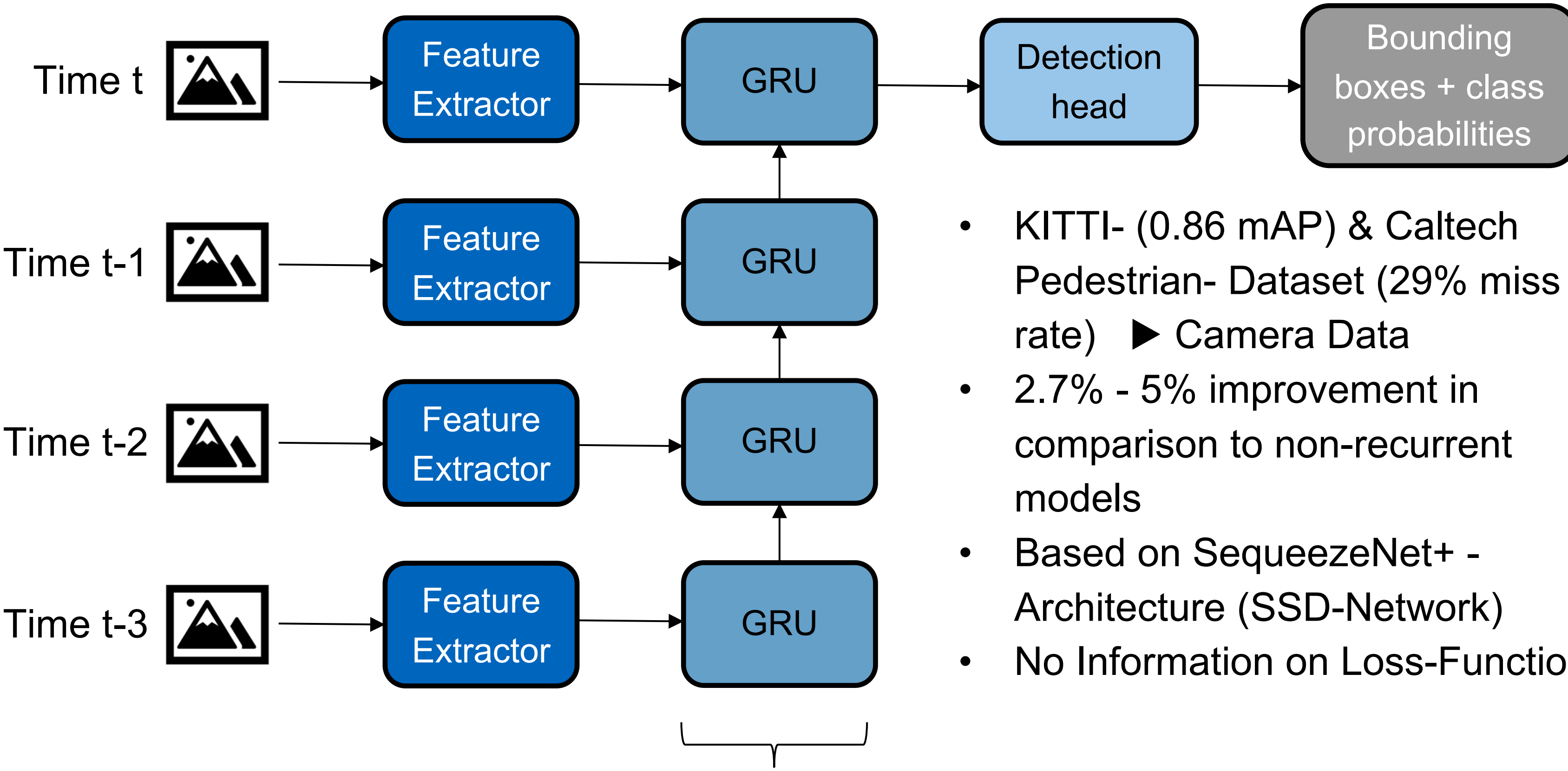


Features feeded into the RNN

Agenda

1. Intro
2. Feature-based Video Object Detection
    1. Definition
    2. Recurrent Multi-frame Single Shot Detector for Video Object Detection
    3. Mobile Video Object Detection with Temporally aware Feature Maps
    4. Feature Selective Small Object Detection via Knowledge-based recurrent attentive neural networks
    5. Looking fast and slow: memory-guided mobile video object detection
    6. Delving Deeper into Convolutional Networks for Learning Video Representations
    7. Detect to track and track to detect
3. Box-Level-based Video Object Detection
4. Flow-based Video Object Detection
5. Comparison of different approaches
6. Outro

2.2 Recurrent Multi-frame Single Shot Detector for Video Object Detection



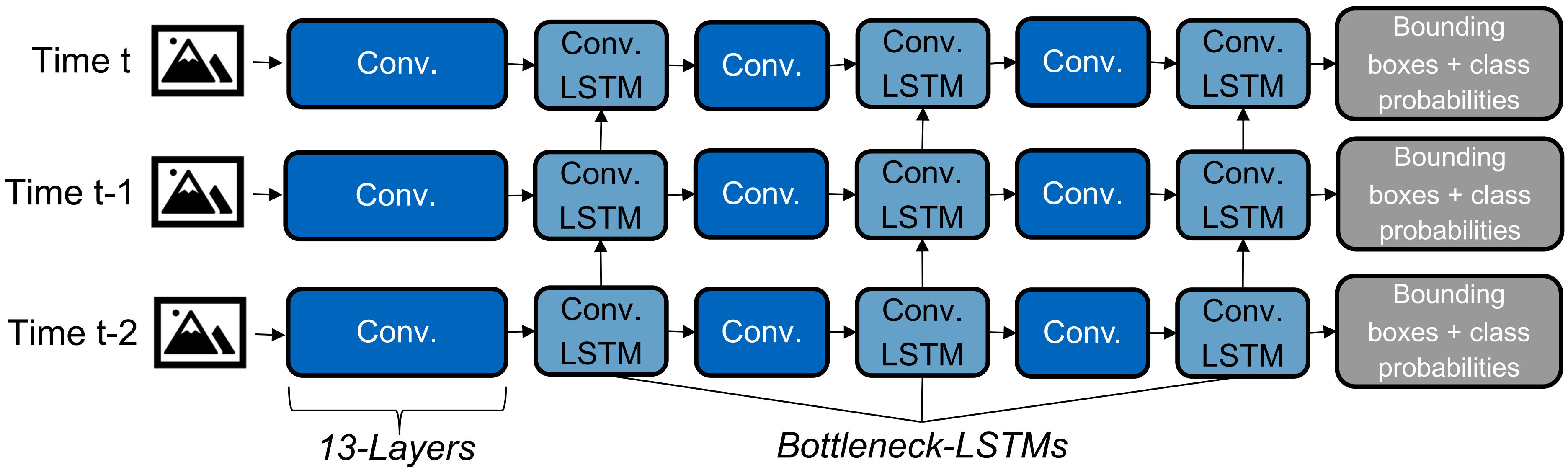
- KITTI- (0.86 mAP) & Caltech Pedestrian- Dataset (29% miss rate) ▶ Camera Data
- 2.7% - 5% improvement in comparison to non-recurrent models
- Based on SequeezeNet+ - Architecture (SSD-Network)
- No Information on Loss-Function

Recurrent-Layer for Data fusion

[1] [14]

Agenda

2.3 Mobile Video Object Detection with Temporally-Aware Feature Maps

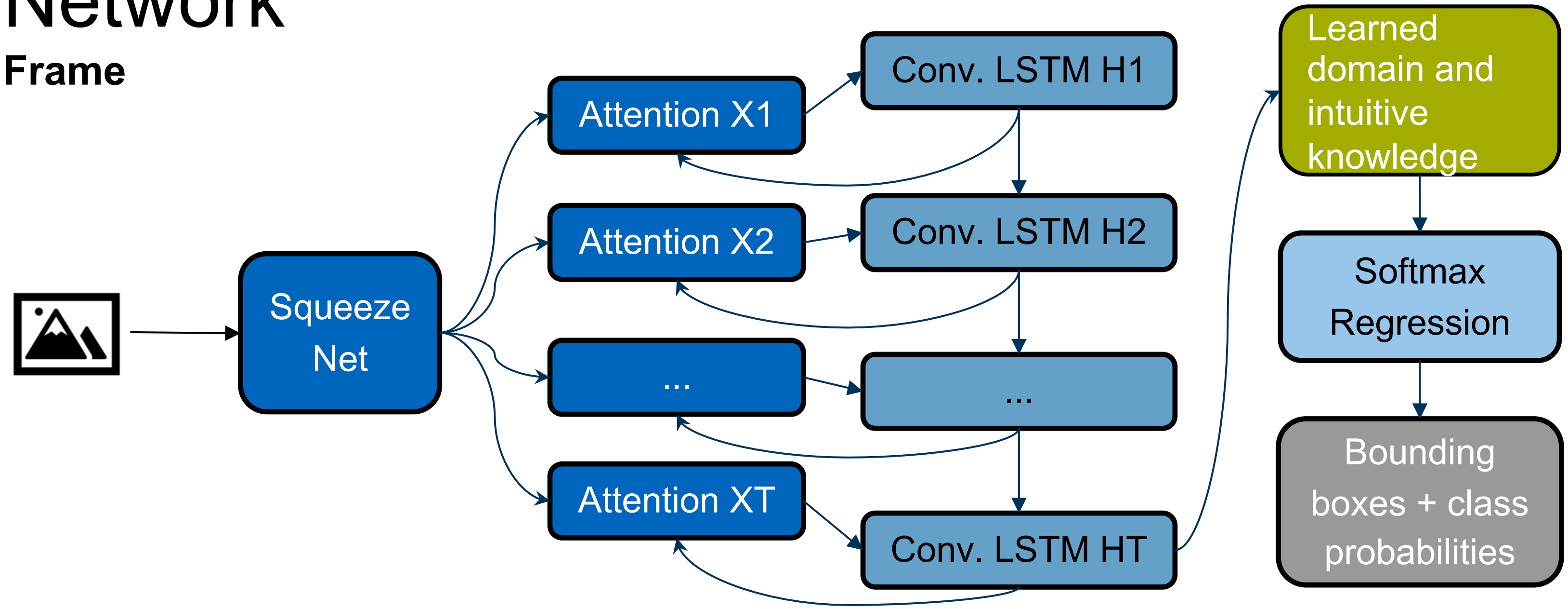


- Bottleneck-LSTMs to increase the computation speed
- Maximal 54.4 mAP on Imagent VID (15 fps, mobile device - Pixel Phone 2)
- Based on SSD-Network, with different numbers of LSTMs after Conv. Layers
- No Information on Loss-Function

[2]

Agenda

2.4 Feature Selective Small Object Detection via Knowledge-based Recurrent Attentive Neural Network



- Compute feature maps using a modified SqueezeNet architecture.
- Propagate the features through a Recurrent Attentive Neural Network, comprised of:
    - Attention Mechanism to detect key areas within the feature maps.
    - Convolutional LSTM for temporal feature propagation.
- Reverse gaussian feature maps are combined with the maps obtained from Conv. LSTM.
    - These feature maps are based on learnable mean and covariance terms.
    - This prior knowledge is derived from the assumption that traffic signs are always located at the bias of the center.

2.4 Feature Selective Small Object Detection via Knowledge-based Recurrent Attentive Neural Network

Loss Function:

$$\frac{\lambda_{bbox}}{N_{obj}} \sum_{i=1}^{W} \sum_{j=1}^{H} \sum_{k=1}^{K} I_{ijk}(Q_x + Q_y + Q_w + Q_h)$$
$$+ \sum_{i=1}^{W} \sum_{j=1}^{H} \sum_{k=1}^{K} \frac{\lambda_c^+ onf}{N_{obj}} I_{ijk} Q_\gamma + \frac{\lambda_{conf}^-}{WHK - N_{obj}} \tilde{I}_{ijk} \gamma_{ijk}^2$$
$$+ \frac{1}{N_{obj}} \sum_{i=1}^{W} \sum_{j=1}^{H} \sum_{k=1}^{K} \sum_{c=1}^{C} I_{ijk} l_c^G \log(p_c).$$

The Loss function consists of three parts:

1. Bounding box regression
2. Confidence score regression
3. Cross entropy loss of classification

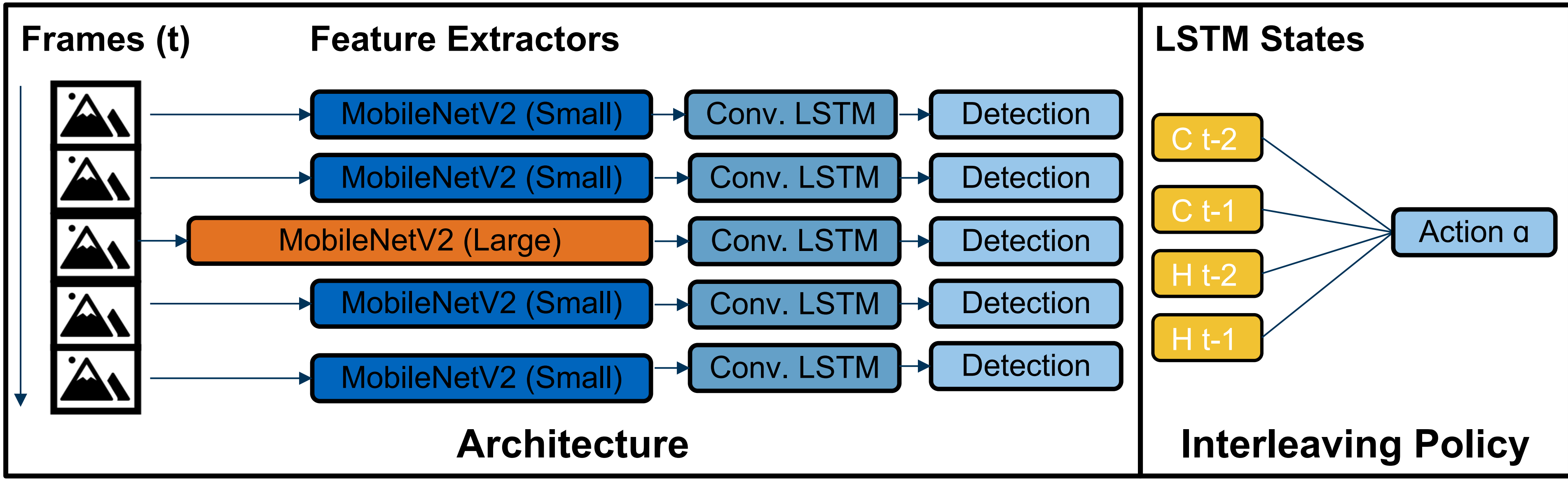
Results:

mAP of 81.3% at 30.8 FPS on KITTI dataset using Nvidia Titan X.
mAP of 57.8% at 37.5 FPS on COCO dataset (only the pedestrian class tested) using Nvidia Titan X.

Agenda

1. Intro
2. Feature-based Video Object Detection
    1. Definition
    2. Recurrent Multi-frame Single Shot Detector for Video Object Detection
    3. Mobile Video Object Detection with Temporally aware Feature Maps
    4. Feature Selective Small Object Detection via Knowledge-based recurrent attentive neural networks
    5. Looking fast and slow: memory-guided mobile video object detection
    6. Delving Deeper into Convolutional Networks for Learning Video Representations
    7. Detect to track and track to detect
3. Box-Level-based Video Object Detection
4. Flow-based Video Object Detection
5. Comparison of different approaches
6. Outro

2.5 Looking Fast and Slow: Memory-Guided Mobile Video Object Detection



- Run multiple feature extractors sequentially or concurrently to obtain feature maps.
    - The idea is to use small and large feature extractors to optimize performance
- Aggregate and refine these feature maps using convolutional LSTM based memory network.
    - To improve speed of LSTM network, add skip connections and LSTM state groups.
- Apply SSD-style detection on refined features to obtain classification and bounding boxes.
- Use a reinforcement learning based policy for selection of which feature extractor to run.
- Large and small frame extractors can run in parallel using asynchronous mode.

2.5 Looking Fast and Slow: Memory-Guided Mobile Video Object Detection

Loss Function:

1. Reinforcement Learning Policy

$$R(a) = \begin{cases} \min_i L(D^i) - L(D^0) & a = 0 \\ \gamma + \min_i L(D^i) - L(D^1) & a = 1 \end{cases}$$

The Loss function consists of two parts:

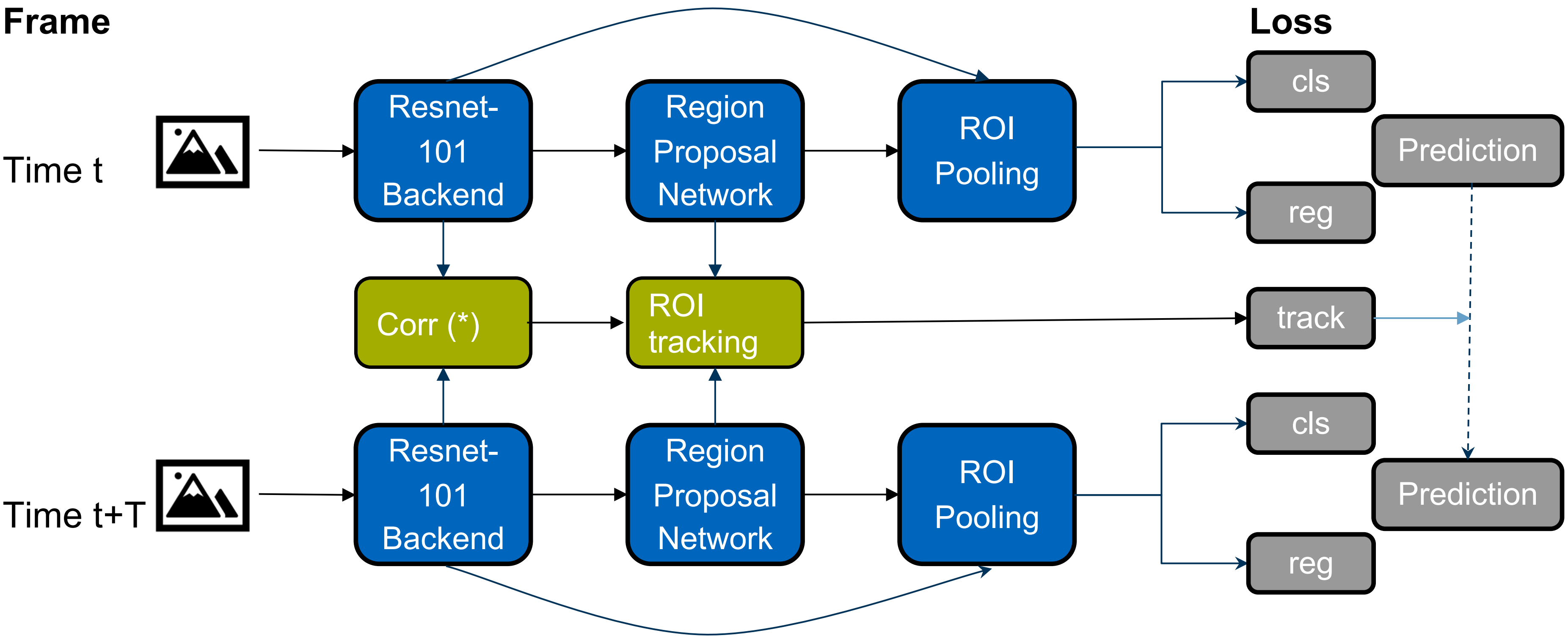
1. Bounding box regression
2. Cross entropy loss of classification

Results:

mAP of 60.7% at 48.8 fps on ImageNet VID dataset using a Pixel 3 phone

Agenda

2.7 Detect to track and track to detect



- Compute Convolutional feature maps using a Resnet-101 architecture.
- Use a RPN (region proposal network) to find candidate regions in the frame.
- ROI Pooling layer, to classify boxes and refine their coordinates (regression).
- Find correlation features between two frames’ feature maps and do ROI tracking.
- Due to memory constraints, use tracklets, which are class-based optimal paths in video.

2.7 Detect to track and track to detect

Loss Function:

$$L(\{p_i\},\{b_i\},\{\Delta_i\}) = \frac{1}{N}\sum_{i=1}^{N} L_{cls}(p_{i,c^*})$$
$$+\lambda \frac{1}{N_{fg}} \sum_{i=1}^{N} [c_i^* > 0] L_{reg}(b_i, b_i^*)$$
$$+\lambda \frac{1}{N_{tra}} \sum_{i=1}^{N_{tra}} L_{tra}(\Delta_i^{t+\tau}, \Delta_i^{*,t+\tau}).$$

The Loss function consists of three parts:

1. The cross entropy classification loss.
2. The bounding box regression loss
3. The tracking regression loss.

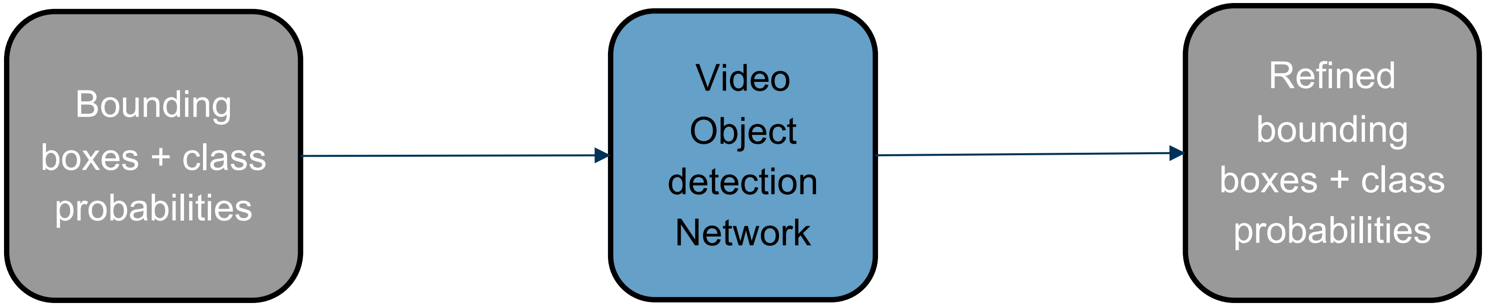
Results:

mAP of 82.0% at 7 fps on ImageNet VID dataset using a Nvidia TITAN X or
mAP of 78.5% at 55 fps on ImageNet VID dataset using a Nvidia TITAN X

Agenda

1. Intro
2. Feature-based Video Object Detection
3. Box-Level-based Video Object Detection
    1. Definition
    2. Context Matters: Refining Object Detection in Video with Recurrent Neural Networks
    3. Object Detection from Video Tubelets with Convolutional Neural Networks
    4. Optimizing Video Object Detection via Scale-Time Lattice
    5. Spatially Supervised Recurrent Convolutional Neural Networks for Visual Object Tracking
4. Flow-based Video Object Detection
5. Comparison of different approaches
6. Outro

3.1 Definition

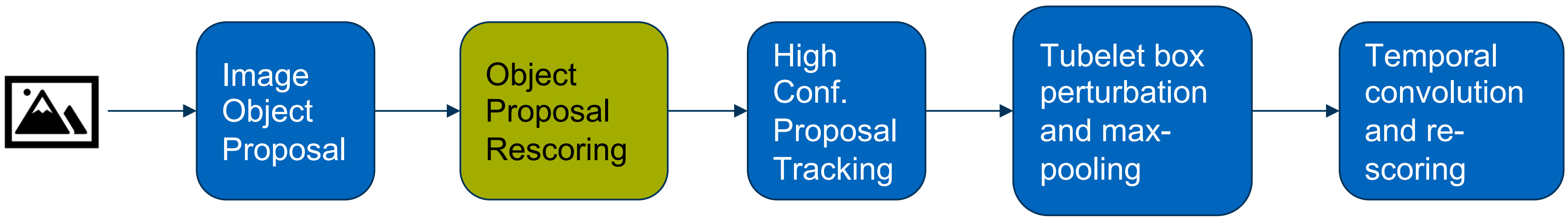


➡ Bounding Boxes and Class probabilities are fed into the network and are refined temporally and/or spatially

Agenda

1. Intro
2. Feature-based Video Object Detection
3. Box-Level-based Video Object Detection
    1. Definition
    2. Object Detection from Video Tubelets with Convolutional Neural Networks
    3. Optimizing Video Object Detection via Scale-Time Lattice
    4. Context Matters: Refining Object Detection in Video with Recurrent Neural Networks
    5. Spatially Supervised Recurrent Convolutional Neural Networks for Visual Object Tracking
4. Flow-based Video Object Detection
5. Comparison of different approaches
6. Outro

3.2 Object Detection from Video Tubelets with Convolutional Neural Networks



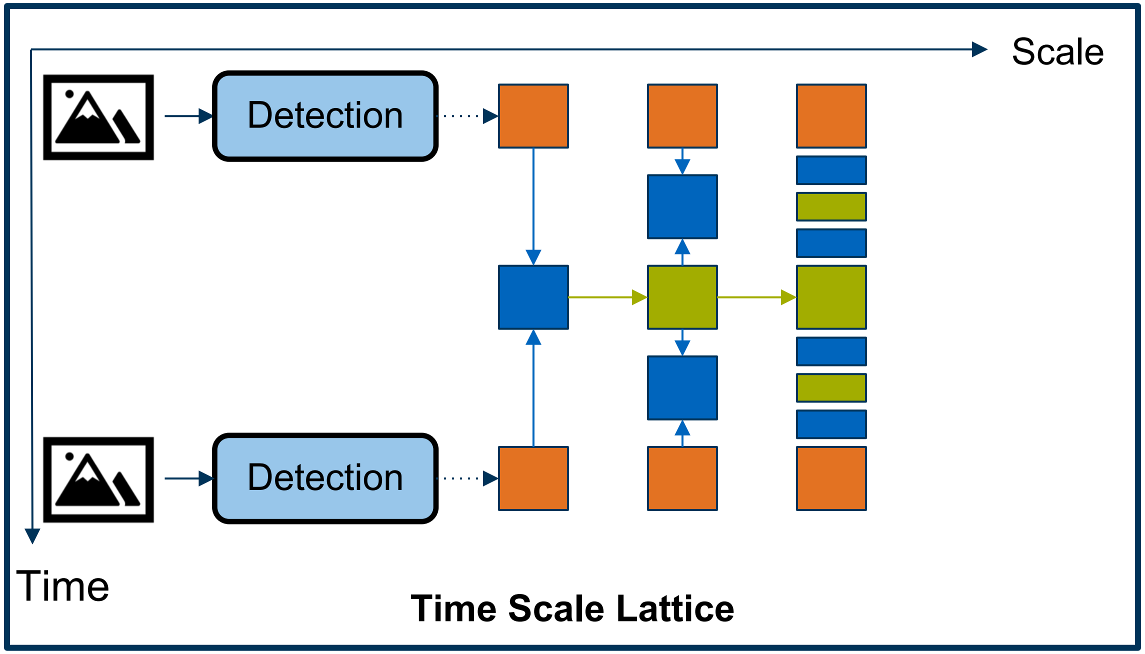
- Use selective search algorithm to generate around 2000 object proposals on each frame.
- Use GoogleNet for feature extraction and then 30 SVMs for 30 VID classes to generate object proposal scores for each object proposal.
- Track high confidence targets bi-directionally.
- Two kinds of Perturbations:
  - The first method is to generate new boxes around each tubelet box on each frame by randomly perturbing the boundaries of the tubelet box.
  - The second perturbation method is to replace each tubelet box with original object detections that have overlaps with the tubelet box beyond a threshold.
- Train a class-specific TCN using the tubelet features as input. The inputs are time series including detection scores, tracking scores and anchor offsets. The output values are probabilities whether each tubelet box contains objects of the class.

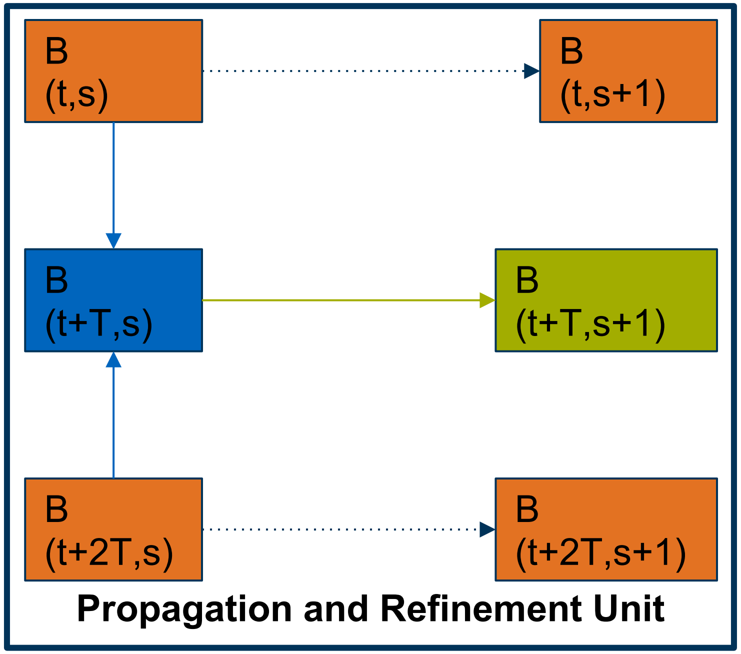
.

Agenda

1. Intro
2. Feature-based Video Object Detection
3. Box-Level-based Video Object Detection
    1. Definition
    2. Object Detection from Video Tubelets with Convolutional Neural Networks
    3. Optimizing Video Object Detection via Scale-Time Lattice
    4. Context Matters: Refining Object Detection in Video with Recurrent Neural Networks
    5. Spatially Supervised Recurrent Convolutional Neural Networks for Visual Object Tracking
4. Flow-based Video Object Detection
5. Comparison of different approaches
6. Outro

3.3 Optimizing Video Object Detection via a Scale-Time Lattice





- Apply object detection on keyframes extracted adaptively.
  - The extraction policy is based on number of objects and amount of movement in frames.
  - If higher number/movement of objects in frames then higher extraction rate.
- Propagation and refinement unit, propagates the frames temporally and refines spatially.
- For temporal propagation, use a small network such as resnet-18 to extract box features and a regressor to predict object movement from t to t + T.
- For spatial refinement, use a regressor to refine the bounding boxes over increasing scale.

3.3 Optimizing Video Object Detection via a Scale-Time Lattice

Loss Function:

$$L(\Delta_{\mathcal{F}_T}, \Delta_{\mathcal{F}_S}, \Delta^*_{\mathcal{F}_T}, \Delta^*_{\mathcal{F}_S}) =$$
$$\frac{1}{N}\sum_{j=1}^{N} L_{\mathcal{F}_T}(\Delta^j_{\mathcal{F}_T}, \Delta^{*j}_{\mathcal{F}_T}) + \lambda \frac{1}{N}\sum_{j=1}^{N} L_{\mathcal{F}_S}(\Delta^j_{\mathcal{F}_S}, \Delta^{*j}_{\mathcal{F}_S})$$

The Loss function consists of two parts:

1. Smooth L1 loss of temporal propagation.
2. Smooth L1 loss of spatial refinement.

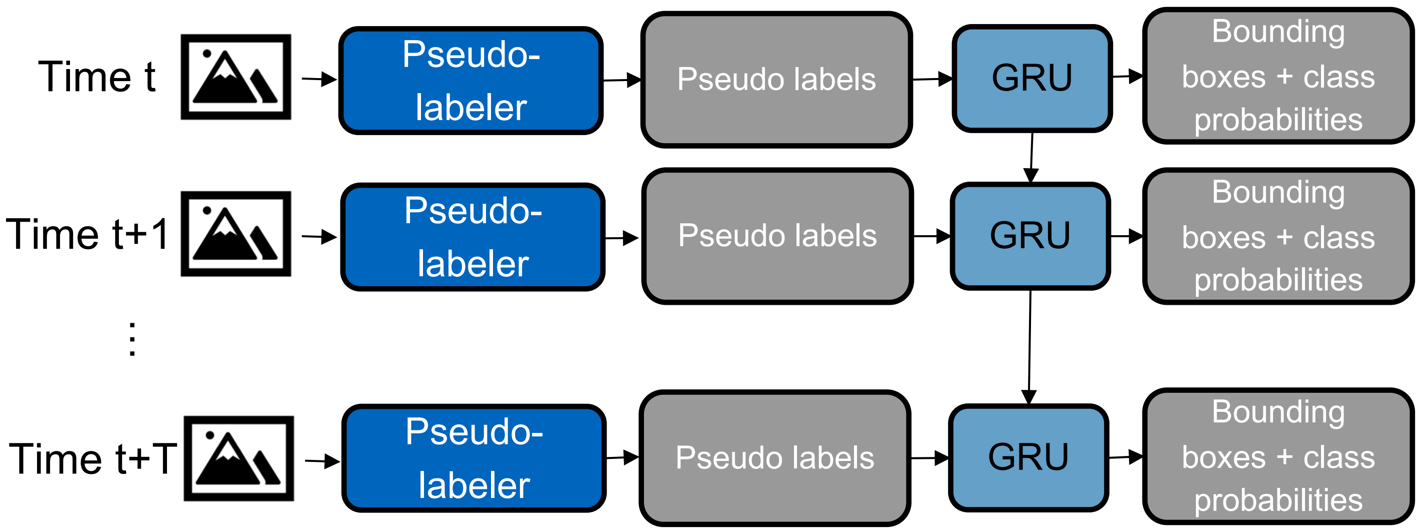
Result:

mAP of 79.6% at 20 fps on ImageNet VID dataset using Nvidia Titan X or
mAP of 79% at 62 fps on ImageNet VID using Nvidia Titan X

Agenda

1. Intro
2. Feature-based Video Object Detection
3. Box-Level-based Video Object Detection
    1. Definition
    2. Object Detection from Video Tubelets with Convolutional Neural Networks
    3. Optimizing Video Object Detection via Scale-Time Lattice
    4. Context Matters: Refining Object Detection in Video with Recurrent Neural Networks
    5. Spatially Supervised Recurrent Convolutional Neural Networks for Visual Object Tracking
4. Flow-based Video Object Detection
5. Comparison of different approaches
6. Outro

3.4 Context Matters: Refining Object Detection in Video with Recurrent Neural Networks



Pseudo-Labeler:

- CNN
- YOLO and RNN

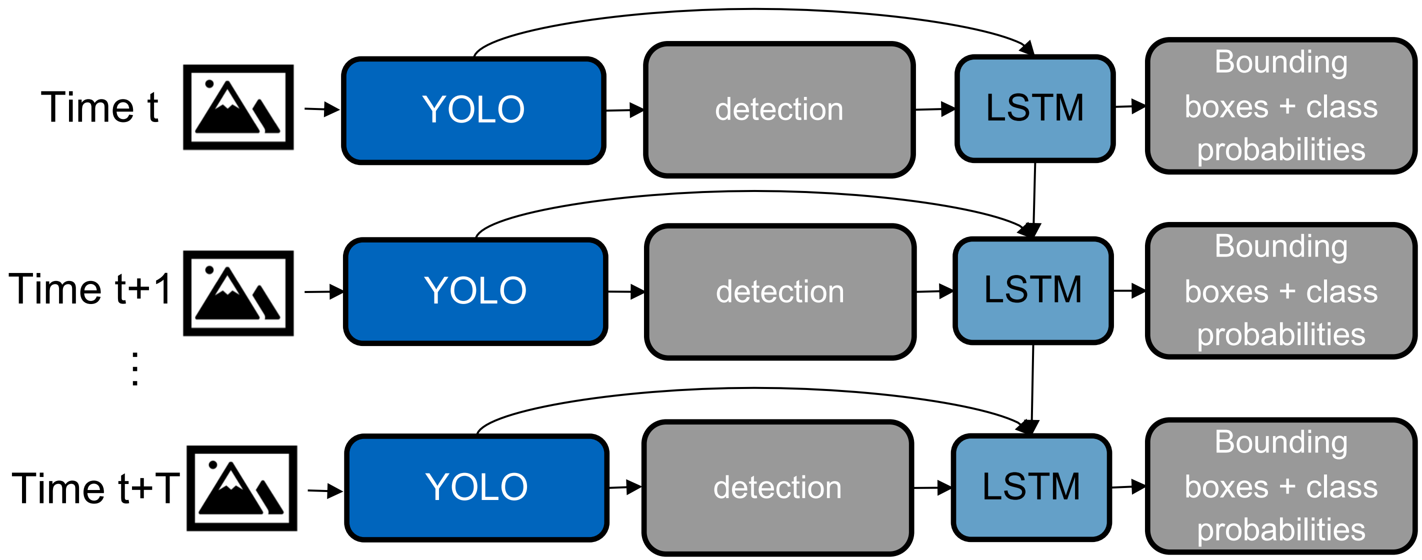
Pseudo labels feed into GRU

- First Train Pseudo-Labeler then whole network
- $loss = d_{loss} + \alpha \cdot s_{loss} + \beta \cdot c_{loss} + \gamma \cdot pc_{loss}$
  - Detection loss; similarity loss; Category loss and prediction consistent loss
  - Each Square losses
- Maximal 68.73 mAP (no info on fps) on Youtube-Objects dataset

[4]

Agenda

3.4 Spatially Supervised Recurrent Convolutional Neural Networks for Visual Object Tracking

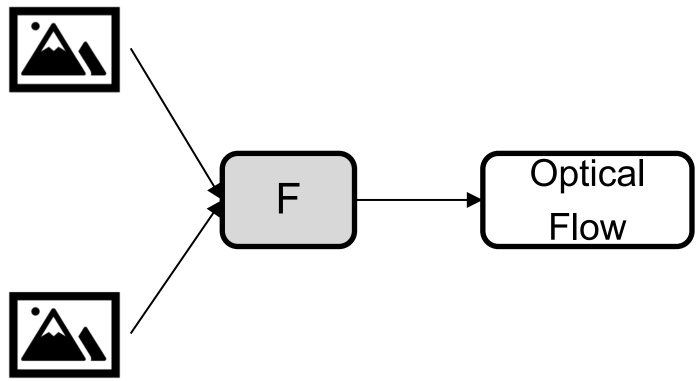


- YOLO creates high-level visual features and preliminary location inferences
- LSTM for sequence processing
- Alternative: Convert Yolo detection into heatmap -> better to visualize
- Mean Squared Error: $L_{MSE} = \frac{1}{n}\sum_{i=1}^{n} ||B_{target}\ \_\ Bp_{red}||_2^2$
- 0.455 IoU (20 / 60 fps; NVIDA TITAN X GPU and octa core processor; better then 9 other trackers on overlapping objects (no map given) – OTB challenge [5]

Agenda

1. Intro
2. Feature-based Video Object Detection
3. Box-Level-based Video Object Detection
4. Flow-based Video Object Detection
    1. Definition
    2. Deep Feature Flow for Video Recognition
5. Comparison of different approaches
6. Outro

4.1 Definition



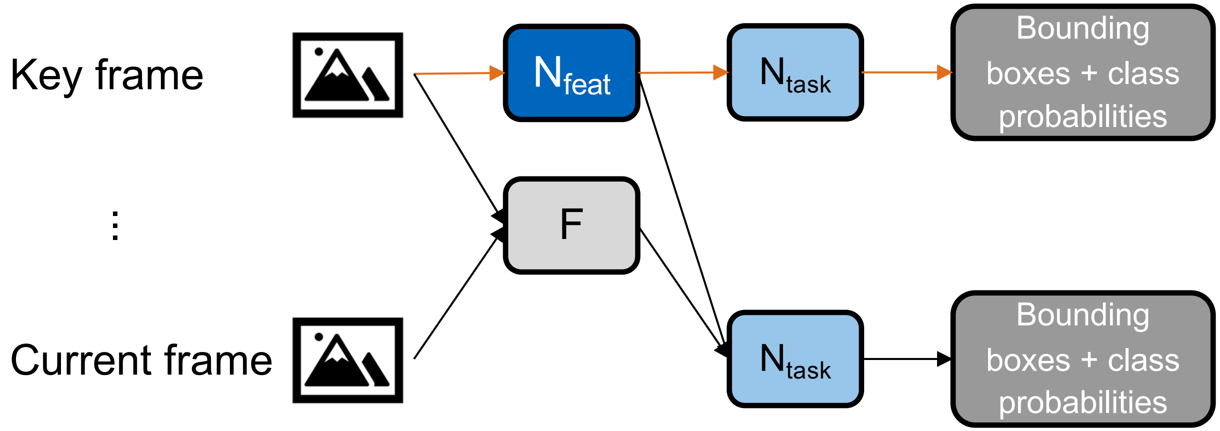
Uses Flow Network

- Estimates the optical flow
- Projects back location in current frame to location in an earlier frame

[3]

Agenda

4.2 Deep Feature Flow for Video Recognition



- 3 different Networks:
  - N_{feat} -> Feature Network (ResNet): Provides FeatureMaps for Key frames
  - F -> Flow Network (FlowNet): Propagation of optical flow (feature maps)
  - N_{task} -> Classifies based on the feature maps (R-FCN)
- Max. 73.9 % mAP on ImageNET VID Dataset (4.05 fps; 73.1% mAP – 20.25 fps); Max. 71.1% mIoU on Cityscapes Dataset (1.52 fps; 69.2& mIoU – 5.6 fps)
- No information on loss function
- Fixed Key frame scheduling => possible imporvement

[3]

Agenda

1. Intro
2. Feature-based Video Object Detection
3. Box-Level-based Video Object Detection
4. Flow-based Video Object Detection
5. Comparison of different approaches
    1. General
    2. Conclusion Computational power
    3. Conclusion prediction quality
6. Outro

Results - KITTI Dataset

Model	MAP	FPS	Machine	Architecture
Recurrent Multi-frame Single Shot Detector for Video Object Detection	**86.0%**	**50**	**Nvidia Titan X**	**Feature - Level**
Feature Selective Small Object Detection via Knowledge-based Recurrent Attentive Neural Network	81.3%	30.8	Nvidia Titan X	Feature - Level

Results - ImageNet VID Dataset

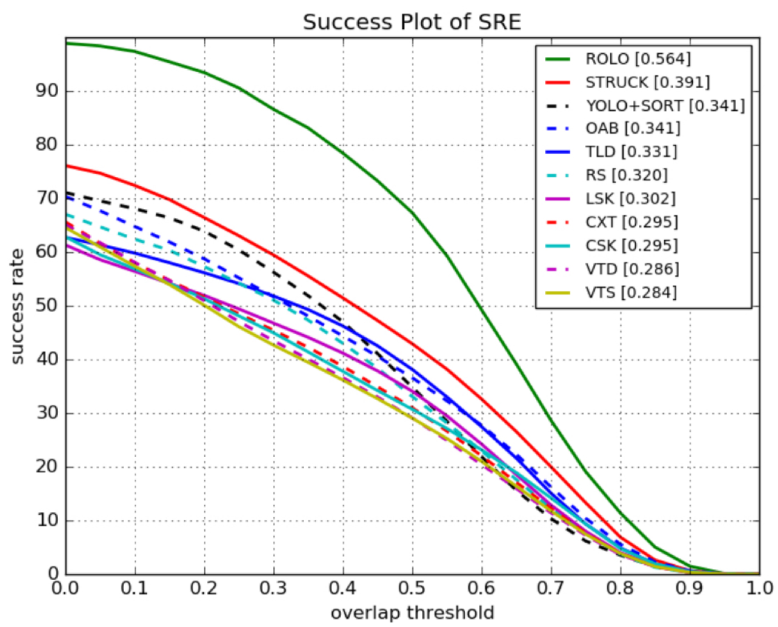
Model	MAP	FPS	Machine	Architecture
Detect to track and track to detect	**82.0%**	**7**	**Nvidia TITAN X**	**Feature-Level**
Optimizing Video Object Detection via a Scale-Time Lattice	79.6%	20	Nvidia TITAN X	Box-Level
Optimizing Video Object Detection via a Scale-Time Lattice - Lightweight	**79%**	**62**	**Nvidia TITAN X**	**Box-Level**
Detect to track and track to detect - Lightweight	78.5%	55	Nvidia TITAN X	Feature-Level
DeepFeature Flow for Video Recognition	73.9%	3		Flow-based
DeepFeature Flow for Video Recognition	73.1%	20.25		Flow-based
Looking Fast and Slow: Memory-Guided Mobile Video Object Detection	60.7%	48.8	Pixel 3 Phone	Feature-level
Mobile Video Object Detection with Temporally-Aware Feature Maps	54.4%	15	Pixel 2 Phone	Feature-level

Results - COCO Dataset

Model	MAP	FPS	Machine
Feature Selective Small Object Detection via Knowledge-based Recurrent Attentive Neural Network	57.8%	37.5	Nvidia Titan X

Results - OTB Challenge Dataset

Model	Success rate	IoU	FPS	Machine
Spatially Supervised Recurrent Convolutional Neural Networks for Visual Object Tracking	0.564	0.455	20 / 60 fps	Nvidia Titan X



Succes plot of Spatial Robustness Evaluation on OTB-30 [5]

Agenda

5.2 Conclusion Performance

Observation	Hypothesis
Networks with more convolutional aspects perform better and provide better results then recurrent [6], [10]	RNNs by definition have a recurrent nature and this can be a bottleneck for Video Object detection in real time
Networks employing an intelligent keyframe extraction policies gain performance benefits [7], [8]	Processing each and every frame of the video is not an efficient way of Video object detection
A lot of performance can be gained by compromising a little on the results [8], [10]	It is important to have a flexible network so that different aspects e.g. depth, keyframe extraction policy can be modified depending upon the application easily
Propagating multiple frames at the same time through the network results in better performance [8], [10]	Networks processing multiple frames at the same time can provide better flexibility on how to propagate them through the network

Agenda

1. Intro
2. Feature-based Video Object Detection
3. Box-Level-based Video Object Detection
4. Flow-based Video Object Detection
5. Comparison of different approaches
    1. General
    2. Conclusion Computational power
    3. Conclusion prediction quality
6. Outro

5.3 Conclusion Prediction Quality

Observation	Hypothesis
[8] and [10] on ImageNet Vid and both processing on multiple frames	**Very beneficial to use previous and future frames**
[1] better prediction quality then [6] on KITTI	**RNN processing on multiple frames better then RNNs processing on regions within a single frame**
[10] good results	**Beneficial to use RNNs to look also on different scales not only at different time steps**
[2] with multiple LSTMs comparatively bad	**one RNN module is enough**
[3] comparatively low map on ImageNet Vid	**better to use Box-level or Feature-level approaches instead of FlowNets**
[5] leading results on OTB challenge dataset	**Combination of Box-Level and Feature-Level approaches leads to promising results**

Agenda

6.1 Conclusion

RNN

- Beneficial to use RNNs in comparisons to similar non-recurrent networks
- Comparatively good results can also be reached with non-recurrent networks, <u>but temporal context matters</u>

General

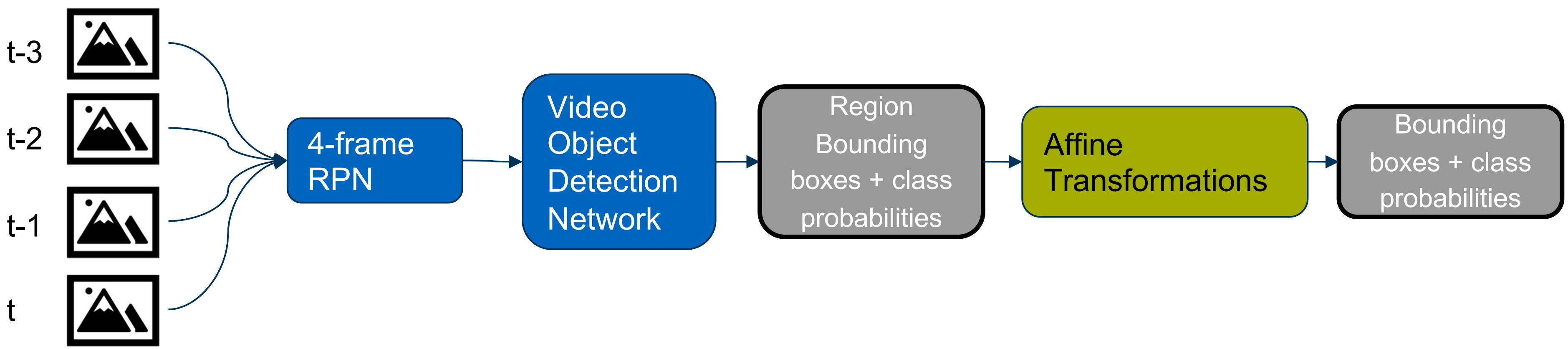
- Good to operate on multiple frames at the same time
- Recurrent layers should not be to deep
- Beneficial to operate on different scales
- Import to use a intelligent key frame policy

[5]

Agenda

1. Intro
2. Feature-based Video Object Detection
3. Box-Level-based Video Object Detection
4. Flow-based Video Object Detection
5. Comparison of different approaches
6. Outro
    1. Conclusion
    2. Further work

6.2 Further Work



Overview

- A region proposal network that is based on the N-Gram concept in Natural Language Processing.
- Given a window of N previous frames propose the regions where the object bounding boxes could be detected from within the next frame.
- The RPN (region proposal network) should be recurrent in nature for detecting the temporal dependencies and can be very lightweight.
- Use only those region proposals and feed them to the video object detection network.
    - So rather than feeding the whole image, feed only the region proposals made by RPN.
- Perform affine transformations to the output bounding boxes to overlay them over the image.

Potential drawbacks

- Region proposals can be of different resolutions depending upon the objects.
- The RPN and the detection network have to lightweight otherwise RPN is just a overhead.

Sources

[1] Alexander Broad, Michael Jones, Teng-Yok Lee. Recurrent Multi-frame Single Shot Detector for Video Object Detection. 2018.

[2] Mason Liu, Menglong Zhu. Mobile Video Object Detection with Temporally-Aware Feature Maps. 2018.

[3] Xizhou Zhu, Yuwen Xiong, Jifeng Dai, Lu Yuan, Yichen Wei. Deep Feature Flow for Video Recognition. 2017.

[4] Subarna Tripathi, Zachary C. Lipton, Serge Belongie, Truong Nguyen. Context Matters: Refining Object Detection in Video with Recurrent Neural Networks.

[5] Guanghan Ning, Zhi Zhang, Chen Huang, Zhihai He, Xiaobo Ren, Haohong Wang. Spatially Supervised Recurrent Convolutional Neural Networks for Visual Object Tracking. 2016.

Sources

[6] Kai Yi, Zhiqiang Jian, Shitao Chen, Nanning Zheng. Feature Selective Small Object Detection via Knowledge-based Recurrent Attentive Neural Network. 2019.

[7] Mason Liu, Menglong Zhu, Marie White, Yinxiao Li, Dmitry Kalenichenko. Looking Fast and Slow: Memory-Guided Mobile Video Object Detection. 2019.

[8] Christoph Feichtenhofer, Axel Pinz, Andrew Zisserman. Detect to Track and Track to Detect. 2017.

[9] Kai Kang, Wanli Ouyang, Hongsheng Li, Xiaogang Wang. Object Detection from Video Tubelets with Convolutional Neural Networks. 2016.

[10] Kai Chen, Jiaqi Wang, Shuo Yang, Xingcheng Zhang, Yuanjun Xiong, Chen Change Loy, Dahua Lin. Optimizing Video Object Detection via a Scale-Time Lattice. 2018.

Sources

[11] Ian Goodfellow, Yoshua Bengio, Aaron Courville Deep Learning (Adaptive Computation and Machine Learning) 2017

[12] https://colah.github.io/posts/2015-08-Understanding-LSTMs/ visited on 26.06.2019

[13] https://medium.com/mlrecipies/deep-learning-basics-gated-recurrent-unit-gru-1d8e9fae7280 visited on 27.06.2019

[14] Alexander Broad, Michael Jones, Teng-Yok Lee, Supplementary Material for Recurrent Multi-frame Single Shot Detector for Video Object Detection

Questions?