Efficient Object Tracking For Autonomous Driving

Dheeraj Varma Chittari

Master’s Thesis in Informatics: Artificial Intelligence

Supervisor: Prof.Dr.H.C. Andreas Dengel

Advisor: Abdul Hannan Khan, M.Sc.

Rheinland Pfälzische Technische Universität - Kaiserslautern Landau

Contents

- Overview of Autonomous Driving
  - Planning Module
  - Object Tracking
- CenterTrack Tracker
  - Objective
  - CenterTrack Characteristics
  - CenterTrack Architecture
  - CenterTrack Baseline Setup
- Experiments
  - Impact of Feature Extractors on Object Tracking
  - Introduction of Attention through Mixers
  - Leveraging Unmatched Detections and Unused Tracklets
- Conclusion

High Level Overview of Autonomous Driving

- Autonomous Driving Systems: Perception, Planning, and Control
  - *Perception module:* contains sensors and perceives information about the surrounding environment.
  - *Planning module :* Processes the sensor information and produces a collision-free trajectory.
  - *Control module :* Executes the trajectory produced by the planner module.

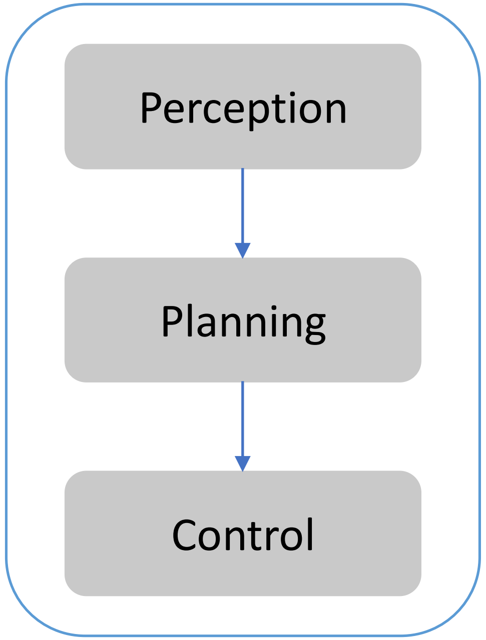

Fig 1: Autonomous Driving Modules

Autonomous Driving - Planning Module

Planning module has two components:

- *Occupancy Maps:* Responsible for finding the optimal route from point A to B.

- *Motion Estimation:* Predicts the next state that the ego vehicle should follow. Employs computer vision based **Object Tracking** techniques.

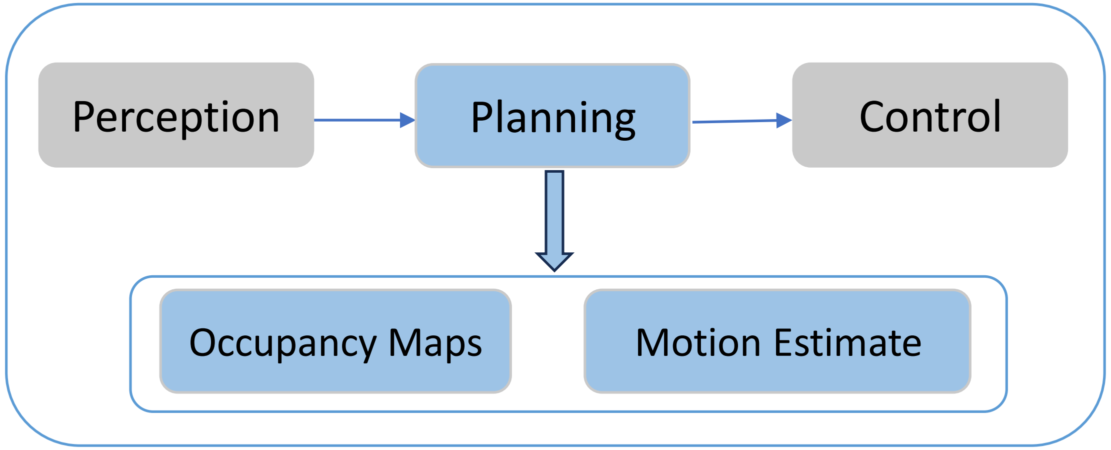

Fig 2: Autonomous Driving Modules with expanded planning.

Ego Vehicle:

- Autonomous vehicle equipped with sensors.

Fig 3:Ego vehicle [1].

Object Tracking

- Identifying the objects of interest and track them over time.

Traditional Computer Vision based Object Tracking

- *Feature Extraction:* SIFT or Canny Edge techniques and masking the background.
- *Feature Tracking:* Kalman or Particle filter methods.

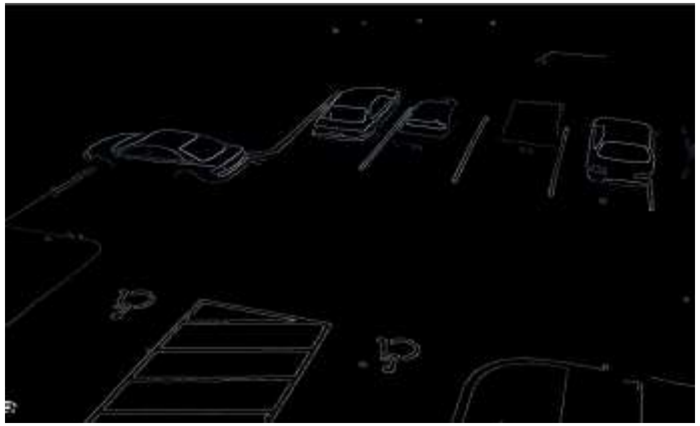

Fig 4: Canny edge detection on cars and road [2].

Advantages

- Does not require extensive training data not training time, better interpretability.

Downsides

- Sensitive to the changes in brightness, partial occlusion, abrupt motion and direction changes.

Object Tracking Continue

- Deep Learning based object tracking has two stages
  - *Object Detection:* Pretrained object detectors for feature extraction.
  - *Motion and Appearance Modelling:* Estimating the next state of an object.

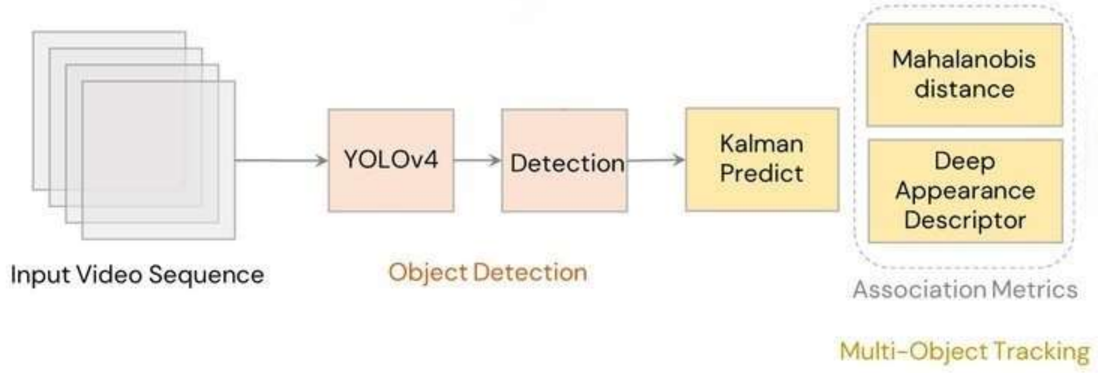

Fig 5: General overview of Deep Learning based Object Tracker Algorithm [3].

- Better handle changes in brightness, partial occlusion and motion blur that lead to a higher tracking accuracy.
- As these advantages **outweighs** downsides of traditional methods, we considered using deep learning approaches for our experiments.

Contents

- Overview of Autonomous Driving
  - Planning Module
  - Object Tracking
- CenterTrack Tracker
  - Objective
  - CenterTrack Characteristics
  - CenterTrack Architecture
  - CenterTrack Baseline Setup
- Experiments
  - Impact of Feature Extractors on Object Tracking
  - Introduction of Attention through Mixers
  - Leveraging Unmatched Detections and Unused Tracklets
- Conclusion

Objective

- Analyse the impact of various components of Deep Learning based Object Tracker.

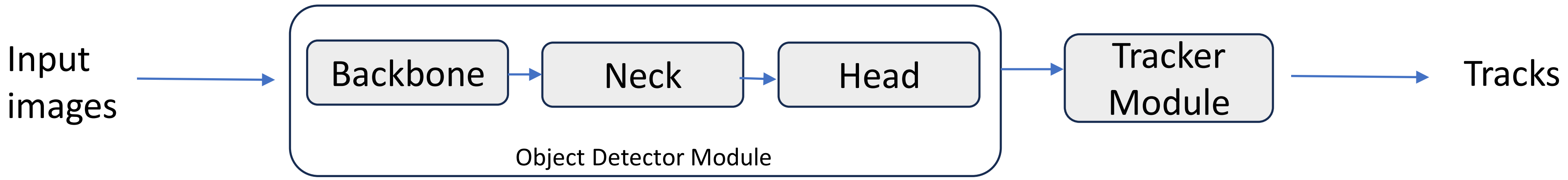

Fig 6: Various modules of deep learning based object tracker.

- With the main motivation of improving the effectiveness of the object tracking accuracy and latency.

CenterTrack Object Tracker

- Point-based Object Tracker
  - Representing objects as points rather than bounding boxes.
- Optical Flow Modelling
  - Incorporates optical flow estimation network.

Fig 7: CenterTrack Object Tracker [4].

- Single Network Architecture
  - All the tasks, object center estimation, size regression, and optical flow estimation, are achieved within a single network architecture.
- Low Latency and Competitive Performance
  - Achieves competitive results leveraging object features extracted from both previous & current frames.

Reasons For Choosing CenterTrack

- Leading Object Tracker:
  - CenterTrack excels in multi-object tracking, especially for autonomous driving.
- Low Latency:
  - Real-time adoption.
- Unique Methodology:
  - Considers object center points for tracking.
  - Estimates the Motion model through the DL network.
  - Considers the previous image features.
- Open-Source Project:
  - As an active open-source project on GitHub.
  - Allows customization and integration.

Rank	Model	MOTA↑	HOTA	Paper	Code	Result	Year	Tags
1	UCMCTrack	90.4	77.1	UCMCTrack: Multi-Object Tracking with Uniform Camera Motion Compensation			2023	
2	OC-SORT	90.3	76.5	Observation-Centric SORT: Rethinking SORT for Robust Multi-Object Tracking			2022	
3	SRK ODESA	90.03		Learning Local Feature Descriptors for Multiple Object Tracking			2020	
4	CenterTrack	89.44		Tracking Objects as Points			2020	
5	DEFT	88.95		DEFT: Detection Embeddings for Tracking			2021	
6	EagerMOT	87.82	74.39	EagerMOT: 3D Multi-Object Tracking via Sensor Fusion			2021	

Fig 8: Performance leaderboard on Multi-Object KITTI Tracking dataset [5].

CenterTrack Architecture

- Input:
  - Images or videos as input.
- Backbone Network:
  - Utilizes Deep Layer Aggregation (DLA34) for feature extraction
- Neck Network:
  - Deep Layer Aggregation (DLAUp) refines features through upsampling and downsampling layers.
- Head Network:
  - Consists of conv blocks to predict the final outputs.
- Track Association:
  - Connects the tracks among object in two frames using the learned offsets.
- Feature Augmentation:
  - Incorporates features from the previous image and object heatmap to provide rich information.

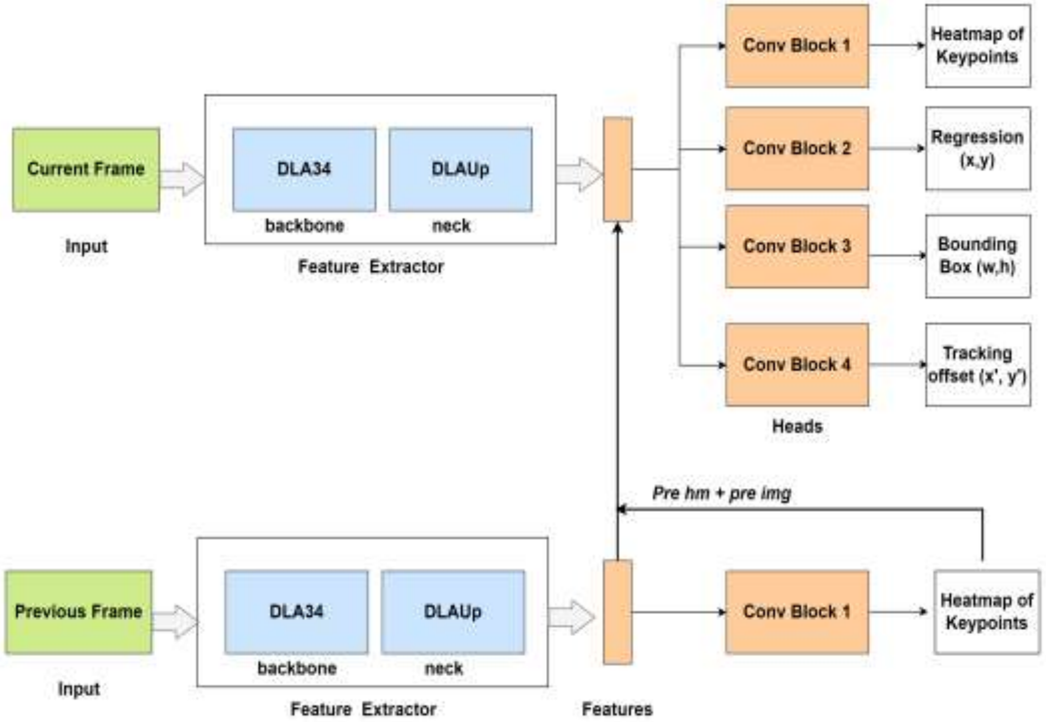

Fig 9: CenterTrack Architecture.

CenterTrack Baseline Setup

- Baseline Model:
  - CenterTrack original model is **pretrained on Nuscenes dataset and finetuned on KITTI Tracking dataset**.
  - Our **baseline model is using only KITTI training dataset** as training on Nuscenes is taking approximately **7-10 days**.
  - CenterTrack original model has MOTA of 89.4 but our **baseline model** has an accuracy of **83.02 MOTA**.
  - All our experiments are compared to our baseline results.

- Accuracy Metrics:
  - False Positives: Tracking the wrong object.
  - False Negative: Failing to track the correct object.
  - IDSW: Tracking an object with wrong identity, identity belongs to a different object.
  - MOTA = 1 – [(FP + FN + IDSW) / GT]

- Model Efficiency Metrics:
  - FLOPS: Floating point operations required to process one image (input).
  - Model Size: Determined by the total number of parameters.
  - Inference Time : Time taken to infer results for a single image (input).

Contents

- Overview of Autonomous Driving
  - Planning Module
  - Object Tracking
- CenterTrack Tracker
  - Objective
  - CenterTrack Characteristics
  - CenterTrack Architecture
  - CenterTrack Baseline Setup
- Experiments
  - Impact of Feature Extractors on Object Tracking
  - Introduction of Attention through Mixers
  - Leveraging Unmatched Detections and Unused Tracklets
- Conclusion

Analysing the Impact of Feature Extractors on Tracking

- CenterTrack uses *Deep Layer Aggression* (DLA34) as a feature extractor.
- *Resnet101 Backbone*: More Deeper Network.

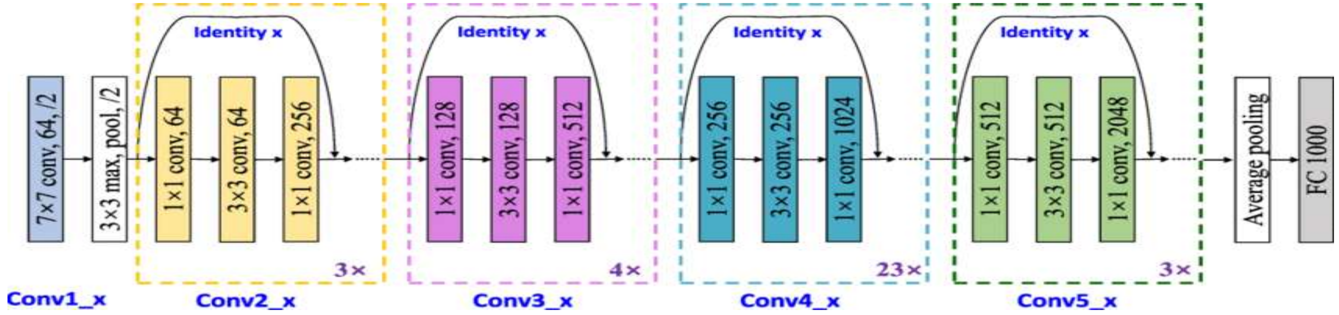

Fig 11: Resnet101 block diagram [7].

- *High Resolution Network (HRNet):* Aggregates image features from different scales. Pose Estimation.

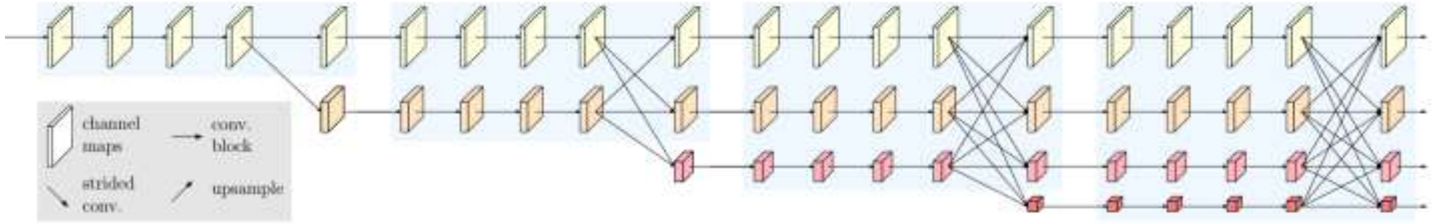

Fig 12: HRNet backbone block diagram [8].

Analysing the Impact of Feature Extractors on Tracking

- Replaced the DLA34 backbone with Resnet-101.

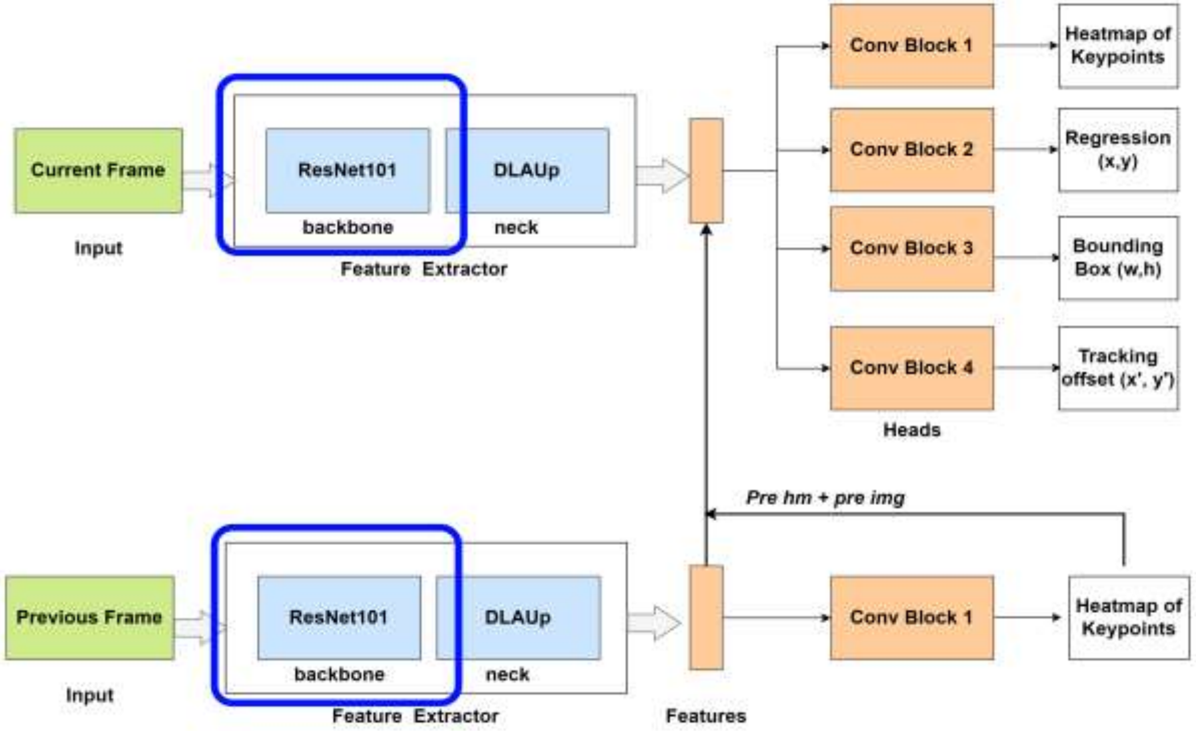

Fig 13: CenterTrack Architecture with Resnet101 backbone.

Analysing the Impact of Feature Extractors on Tracking

- Replaced the DLA34 backbone with HRNet backbone.

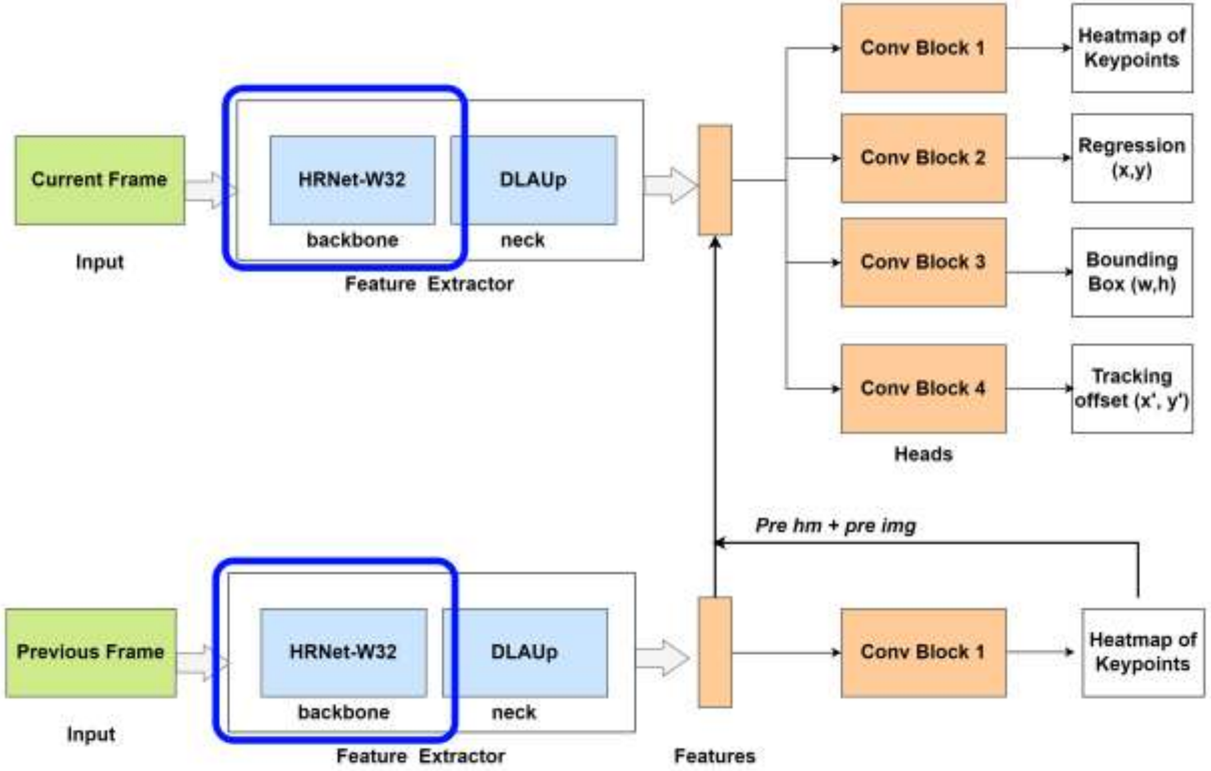

Fig:14 CenterTrack Architecture with HRNet backbone.

Analysing the Impact of Feature Extractors on Tracking

Feature Extractors	Class	MOTA (↑)	FP (↓)	FN (↓)	IDSW(↓)
DLA Backbone	Car Pedestrian	83.02 61.49	933 397	906 1317	23 13
Resnet101	Car Pedestrian	66.38 40.93	686 248	2805 2354	197 47
HRNet	Car Pedestrian	74.8 53.4	931 187	1660 1845	166 58

Table 2: Tracking accuracy comparison of CenterTrack with various Backbone.

Feature Extractors	FLOPS (↓)	Model Size (↓)	Infer Time (↓)
DLA Backbone	105.56 G	19.91 M	28.4 ms
Resnet101	224.63 G	98.22 M	52.7 ms
HRNet	164.15 G	30.53 M	44.6 ms

Table 3: Efficiency evaluation of CenterTrack with various Backbone.

Conclusion

Reason for under performance:

- Backbone and Neck networks are complement to each other.
- DLA34 itself has multiple iterative and hierarchical connections.

Introducing Attention through Mixers in the Head Network

Approximate the Attention with MLPs

- Token and channel mixers as presented by MetaFormer.

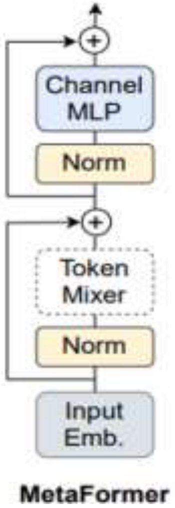

Fig:15 MetaFormer Architecture [9].

Reparametrize Mixers (RepMixer)

- Token Mixer : Depthwise Convolution
  - Applies a single convolutional filter per input channel
- Channel Mixer: ConvFFN
  - Combination of pointwise convolutions and Feedforward Network.

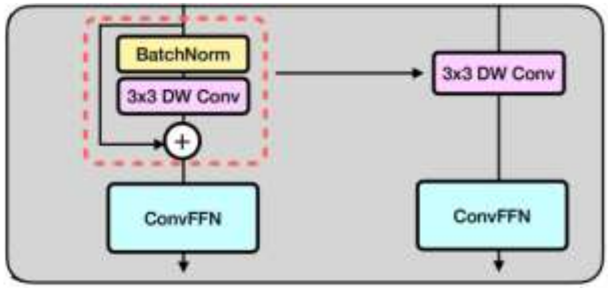

Fig:16 RepMixer Architecture [10].

Introducing Attention through Mixers in the Head Network

- Replaced the several conv blocks in the head network with one RepMixer blocks followed by one conv block.

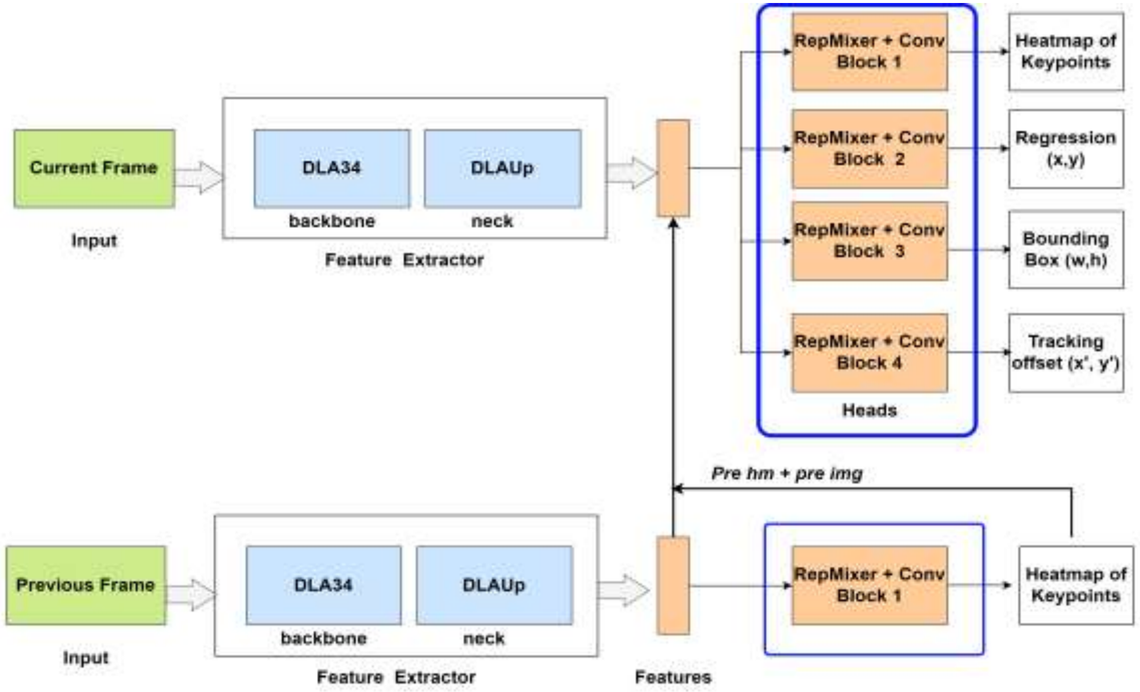

Fig:17 CenterTrack Architecture with RepMixer hybrid head.

Introducing Attention through Mixers

Head Networks	Class	MOTA (↑)	FP (↓)	FN (↓)	IDSW(↓)
Conv Blocks	Car Pedestrian	83.02 61.49	933 397	906 1317	23 13
RepMixer + Conv Block	Car Pedestrian	80.48 40.93	1165 661	959 1350	17 15

Table 4: Tracking accuracy comparison of CenterTrack with Hybrid Head Network.

Head Network	FLOPS (↓)	Model Size (↓)	Infer Time (↓)
Conv Blocks	105.56 G	19.91 M	28.4 ms
RepMixer + Conv Blocks	70.33 G	19.2 M	27.5 ms

Table 5: Efficiency evaluation of CenterTrack with Hybrid Head Network.

Conclusion from the results

- Slight decrease in performance, as we replaced several conv blocks from head with a single RepMixer head.
- Improvement in the overall model efficiency.

Analysing the Unused Tracks and Detections in the Tracker Module

- Leveraging unused tracklets and unmatched detections in the tracker module.
  - Formulated by ByteTrack Tracker.

- Tracker Module
  - Considers only those objects with confidence score > T.
  - Low score objects (T< 0.5) are ignored as false positives.

- Modified Tracker Module:
  - Step 1: Associate high score objects with tracklets.
  - Step 2: Match the low score objects (0.25-0.5) with unmatched tracklets in the current frame.

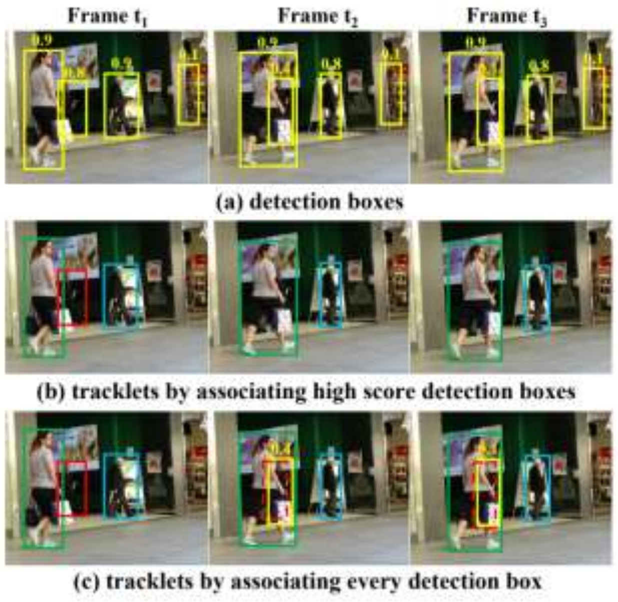

Fig:18 ByteTrack inference on partially occluded object [11].

Unused Tracks and Detections in the Tracker Module

- No change in the CenterTrack model architecture.
  - Requires an additional function to associate the low score detections with unmatched tracklets.
  - Model size, Params, FLOPS remains the same.

Tracker	Class	MOTA (↑)	FP (↓)	FN (↓)	IDSW(↓)
Original	Car Pedestrian	83.02 61.49	933 397	906 1317	23 13
Improved	Car Pedestrian	83.99 40.93	916 388	852 1242	17 11

Table 6: Tracking accuracy comparison of CenterTrack with improved tracker module.

Conclusion

- False Negatives have reduced
  - Able to recover true objects having low scores.
- False Positives have reduced
  - Able to continue the tracks for longer,
    reducing the chance of detecting unwanted objects present near true positives.

Conclusion

- Networks with **iterative and hierarchical connections** in the backbone tend to produce better features than deeper networks alone.
- A good combination of **backbone and neck networks** is crucial for extracting high-quality feature representations.
- **MLP Mixers** can be considered as a alternative for convolutional blocks, offering comparable performance with improved efficiency.
- **Detection boxes with low scores** can be useful for identifying and recovering objects belonging to the target class.

References

1. https://medium.com/intro-to-artificial-intelligence/motion-planning-module-in-autonomous-vehicle-mission-planner-671b2155dec1
2. https://stackoverflow.com/questions/18802498/how-to-identify-a-car-after-performing-canny-edge-in-a-still-image-opencv
3. https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.researchgate.net%2Ffigure%2FArchitecture-of-Deep-SORT-Simple-online-and-real-time-tracking-with-deep-association_fig2_353256407
4. https://arxiv.org/abs/2004.01177
5. https://paperswithcode.com/sota/multiple-object-tracking-on-kitti-tracking
6. https://openaccess.thecvf.com/content_cvpr_2018/papers/Yu_Deep_Layer_Aggregation_CVPR_2018_paper.pdf
7. https://images.app.goo.gl/YabGd7ctuGH16Qoe8
8. https://arxiv.org/abs/1908.07919
9. https://arxiv.org/abs/2111.11418
10. https://arxiv.org/pdf/2303.14189.pdf
11. https://arxiv.org/abs/2110.06864

Thank you!

Questions?

Additional slides for my reference

Deep Layer Aggregation (DLA) Backbone

- Aggregates features from multiple layers for better representation.
- Implements hierarchical learning across stages for multi-scale feature extraction.

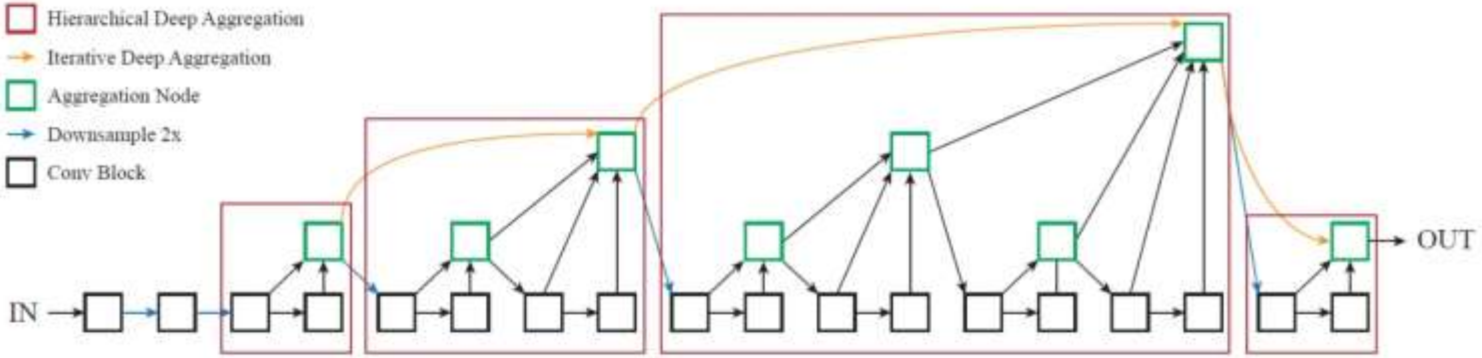

Fig 10: Deep Layer Aggregation [6].

- Uses pooling for downsampling and convtranspose2d for upsampling.
- Captures both spatial and semantic features effectively.

Additional slides for my reference