Joint-Detection-and-Tracking (JDT) Methods for Multi-Object Tracking (MOT)

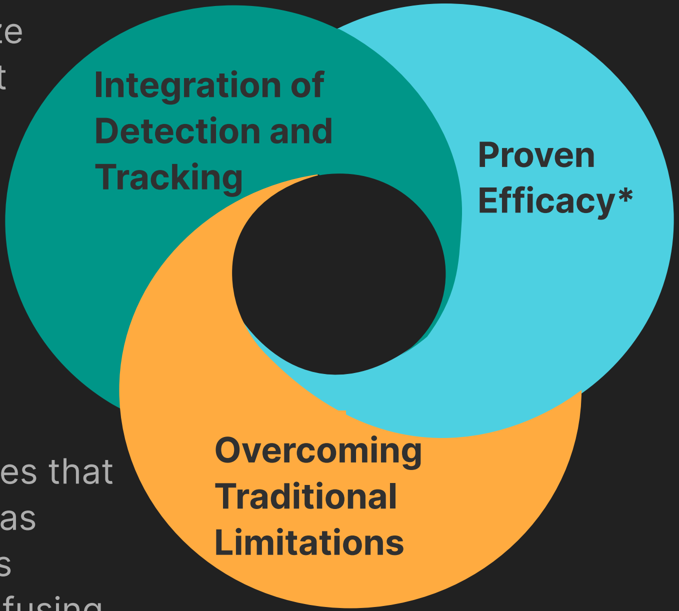
Enhancing MOT Performance through Seamless Integration

- Traditional MOT methods often involve disjointed models for detection and tracking, leading to the need of multiple trainings.
- JDT methods simplify architectures by unifying detection and tracking, offering potential advantages in efficiency and performance.

Multi-object tracking (MOT) (image from [1])

Joint-detection-and-tracking (JDT) methods

JDT methods revolutionize MOT by combining object detection and tracking processes.



JDT methods exhibit notable advantages, particularly in challenging scenarios such as crowded scenes and occlusions

Conventional MOT approaches that treat detection and tracking as separate tasks, JDT methods address these limitations by fusing information from both stages.

*almost every presented method beats or come close it’s disjointed state-of-the-art version.

Notable JDT Methods

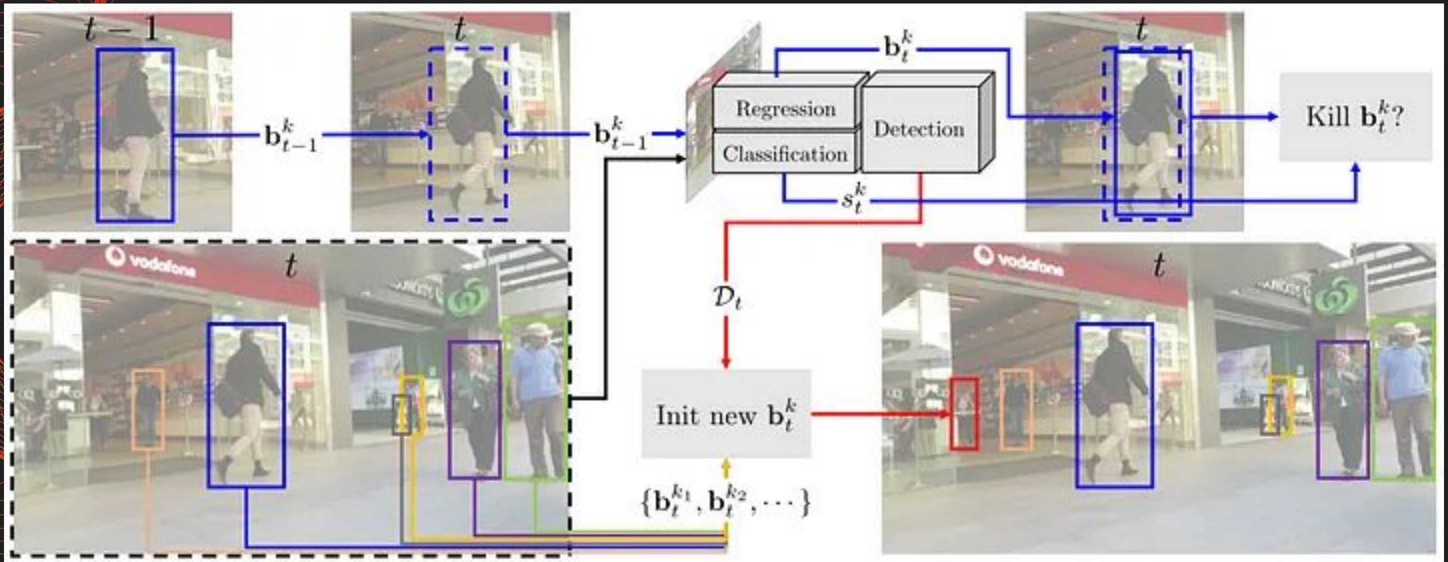
- **Tracktor** [2]: Regression Based tracking using object detector's regression head.
- **JDE** [3]: Shared model for detection and embedding generation.
- **CenterTrack** [4]: Based on CenterNet object detector for efficient tracking.
- **PermaTrack** [5]: Object permanence based tracking of invisible objects.
- **TransTrack** [6]: Transformer Based tracking for enhanced feature representation.
- **TrackFormer** [7]: Single Transformer decoder for joint detection and tracking.

Tracktor - Converting Object Detection to JDT with Zero Training

Tracktor employs a regression head within an object detector for tracking, eliminating the need for a separate tracking model.

Key features : proved effectiveness in high frame rate sequences.

Limitations : reliance on the assumption that target objects move slightly between frames, potentially unsuitable for low frame rate sequences or cases with moving cameras.



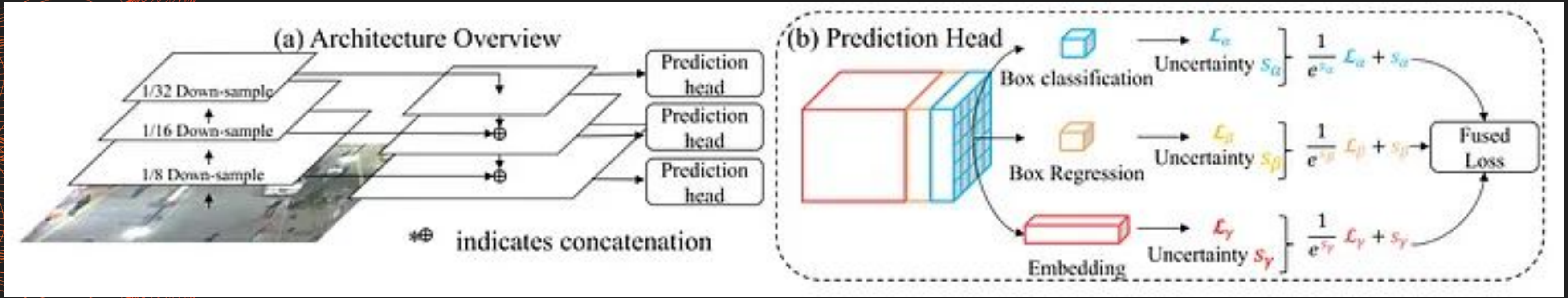
Tracktor (image from [2])
Code available in github.

JDE - Combining Detector and Visual Appearance Model

JDE combines the detector and visual appearance model into a shared framework, facilitating efficient information flow between detection and tracking stages.

Key features : a shared model, efficient information flow, and a single stage object detector for real-time performance.

Limitations : may include the need for more computational resources compared to simpler JDT methods.



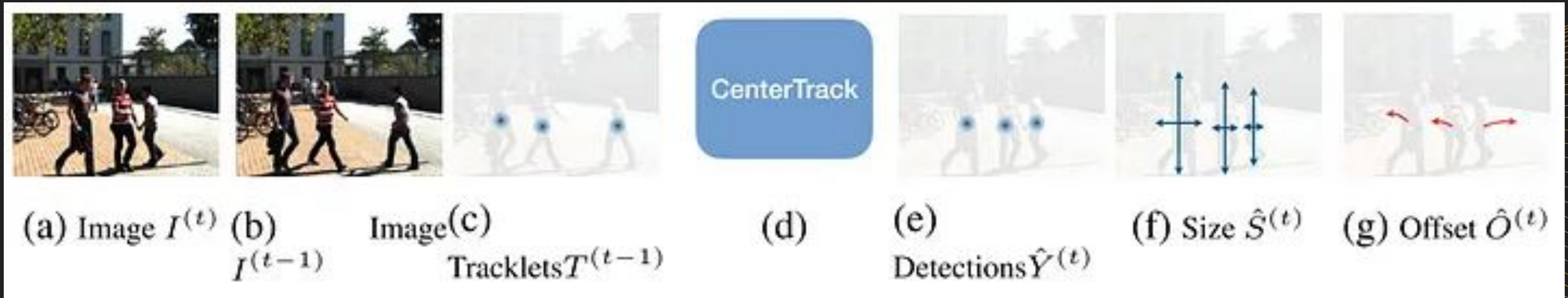
The JDE’s architecture and loss functions (image from [3])

CenterTrack - Efficient JDT using CenterNet

CenterTrack employs the CenterNet object detector as its foundation. CenterNet's efficient design enables CenterTrack to achieve real-time tracking performance.

Key features : archives real-time tracking performance and does not need video annotation for training.

Limitations : May not be as accurate as more complex JDT methods.



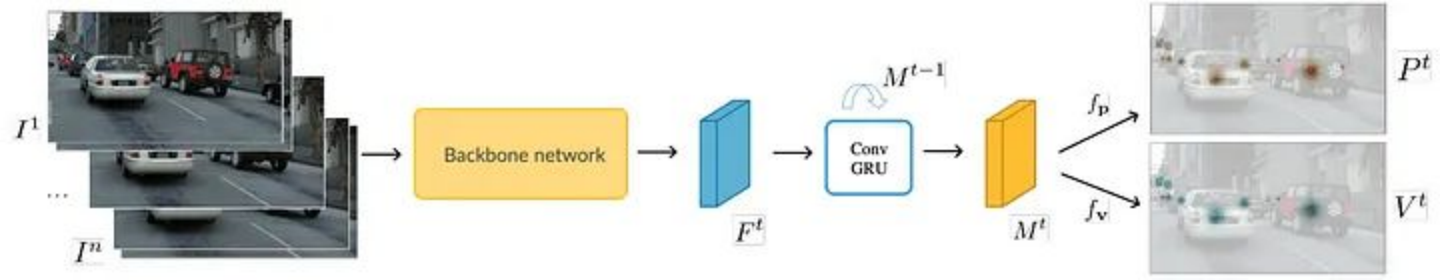
The inputs and outputs of CenterTrack (image from [4])

PermaTrack - Object Tracking with Object Permanence

PermaTrack is a JDT method that tracks invisible objects using object permanence. It employs Convolutional Gated Recurrent Unit (ConvGRU) and a visibility map to identify visible and invisible objects.

Key features : provides a visibility map to identify both visible and invisible objects.

Limitations : requires synthetic data due to the lack of invisible object annotations in real world datasets.



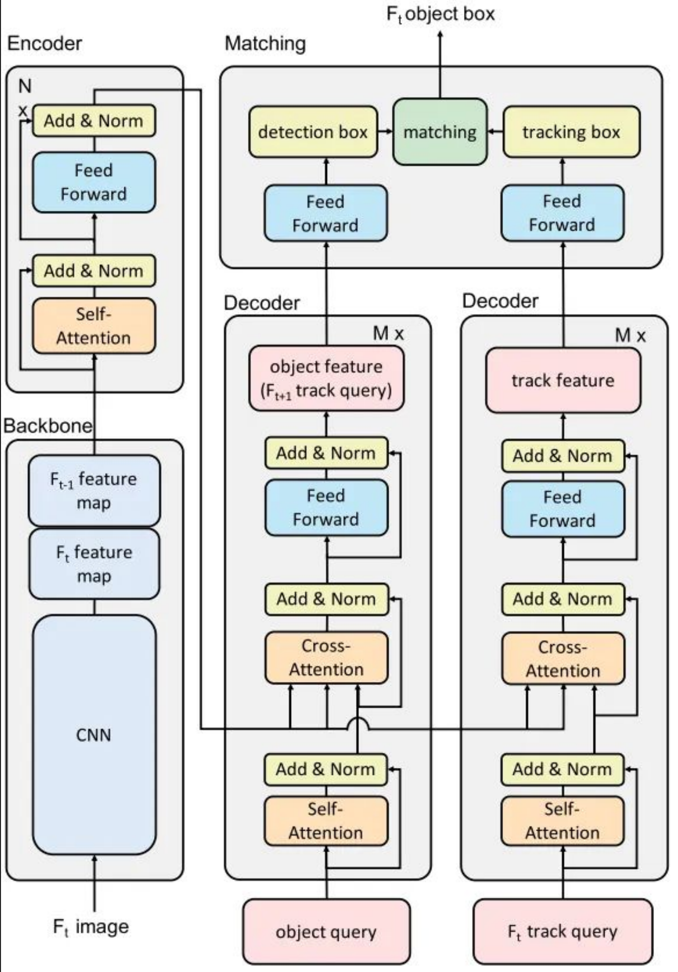
PermaTrack (image from [5])

TransTrack - An Introduction

TransTrack presents a Transformer-based JDT method for enhanced feature representation. It incorporates a Transformer module for tracking, demonstrating improved performance, especially in challenging scenarios.

Key features : Transformer-based tracking for enhanced feature representation Improved tracking performance.

Limitations : may include increased computational resource requirements compared to simpler JDT methods.



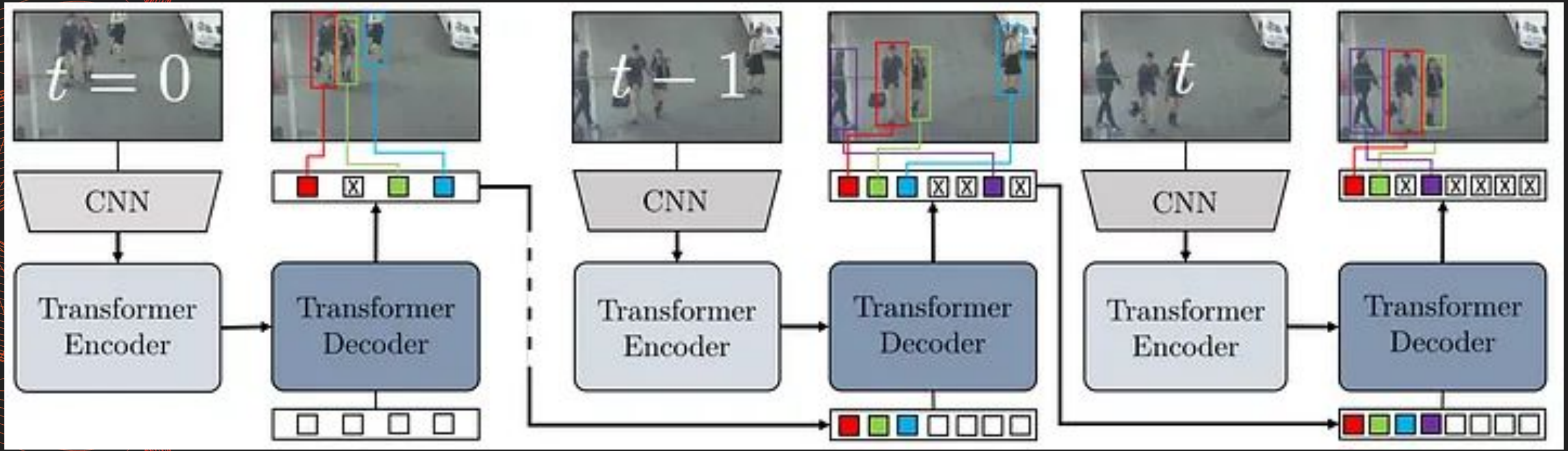
TransTrack (image from [6])

TrackFormer - Introduction

TrackFormer is an efficient Transformer-based JDT method using a single Transformer decoder for joint detection and tracking. It aims for a balance between performance and efficiency.

Key features : Single Transformer decoder Improved tracking performance compared to previous methods, maintaining a reasonable computational complexity.

Limitations : include potential accuracy tradeoffs compared to more complex Transformer-based JDT methods.



TrackFormer (image from [7])

Conclusions

- Reviewed architecture in our practical context.
- Already available resources.
- Computational viability.

Our practical context:

- **Cows** do not move very fast.
- **Detection training data** is already present and fairly accurate.
- The real application is not particularly **error sensible**.

Already available resources:

- Some **similar researches** are already available online [8].
- We **may** be able to **find** an **accurate** enough **tracker** specific for cows [9].
- **Context adaptation** examples are **available** for some of the proposed papers [10].

Computational viability:

- A **single frame can take a while** to process since we do not need many of them.
- **From-scratch training** will not be necessary since context adaptation has been proven valid[11].
- **Real-time** tracking is **NOT required**.

Questions? (bi-directional)