UNIVERSITÀ DEGLI STUDI FIRENZE

Event Object Detection and Classification

Temporal Binary Representation and object detection for event-based videos

Visual Multimedia Recognition
Prof. Alberto Del Bimbo

Francesco Areoluci

VMR project 2020/2021

Project Overview
Project aim

The aim of the project is to address and investigate about the following problems:

1. Develop an **object detection framework** to perform object detection and recognition for **Event-Based videos**;

2. Use a novel technique for event encoding: **Temporal Binary Representation** [1] and compare it against other baseline methods: **Polarity** and **Surface Active Events** (SAE) encodings.

Project Overview

Event-Based Cameras

- Event cameras are based on sensors that capture illumination changes of the scene (events).
- These cameras can produce an asynchronous stream of events indipendently for each pixel.

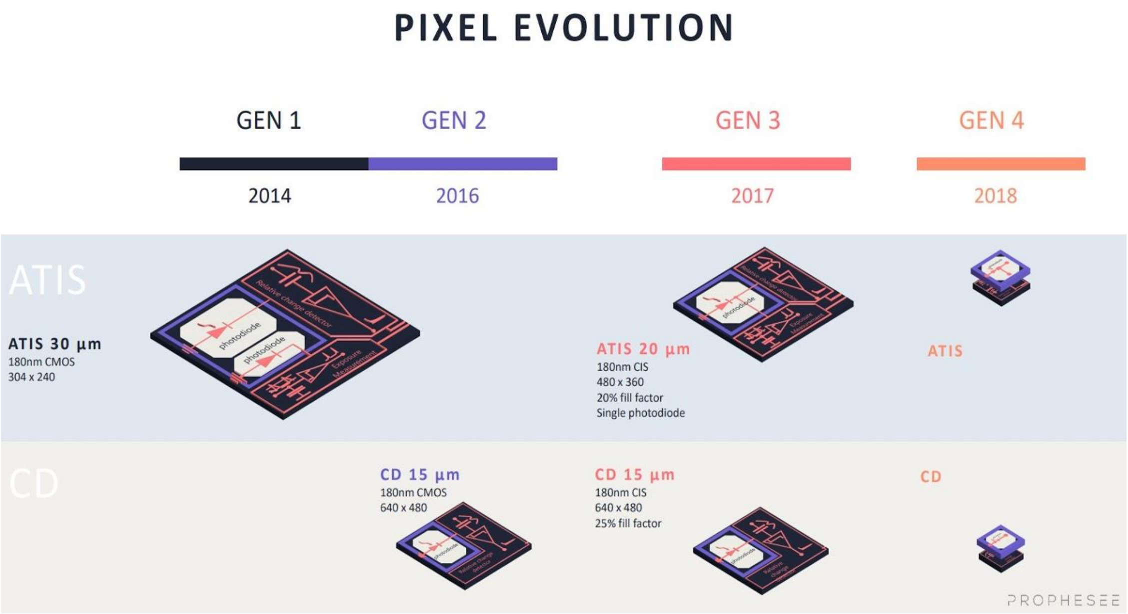

***Figure 1**: Prophesee's event-based sensors*

Project Overview

Computer Vision algorithms for events

Traditional computer vision algorithm (such as Deep Learning techniques) are incompatible with event streams.
In order to feed events to a Deep Learning Model, these must be **encoded to produce frames**, which can be later used as an input for the model.

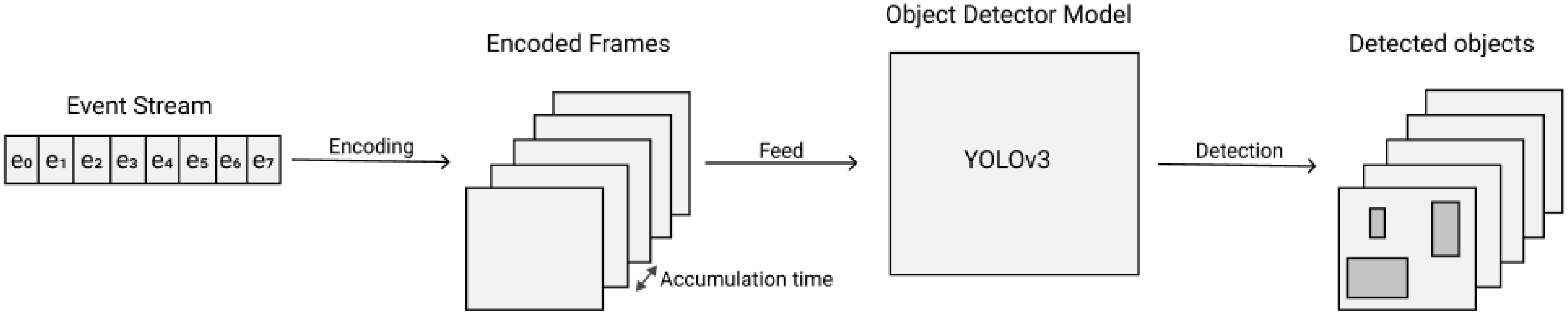

***Figure 2**: event object detection pipeline*

Project Overview

Events encoding concepts

- Each event is characterized by a **polarity:** changes of illumination in a certain position of the scene are represented as a positive or negative change of polarity;
- Encoded frames aggregate the information (events) acquired for a certain amount of time, named **accumulation time**, over the event stream;
- As a consequence, finer accumulation times allows to represent less events in a single frame while grainer accumulation times allows to represent more events.

Project Overview

Temporal Binary Representation

Several methods exist in literature to encode events into frames. Our aim is to study how **Temporal Binary Representation** (TBR) performs in Object Detection and recognition.

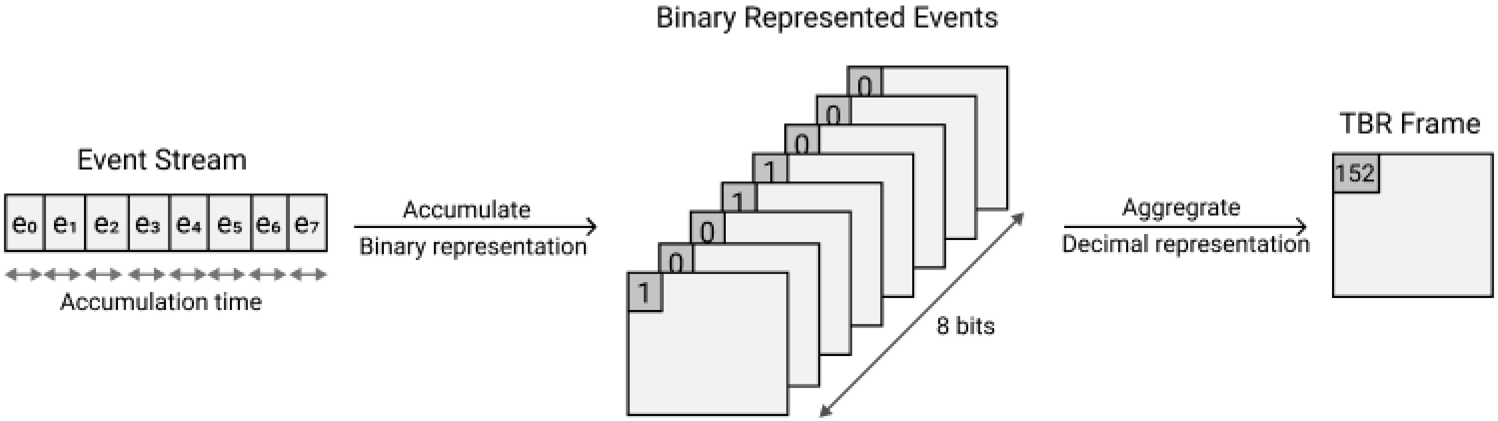

***Figure 3**: Temporal Binary Representation technique*

Project Overview

Other event encoding methods

Other encoding methods, in particular **Polarity** and **SAE**, have been implemented to show how TBR performs w.r.t baseline methods in the object detection context.

(x, y): Event pixel coordinates
Δt: Accumulation time
t_p: Time of last observed event
t_0: Beginning of accumulation time

$$I_p(x,y) = \begin{cases} 0, & \text{if event polarity is negative} \\ 0.5, & \text{if no events happen in } \Delta t \\ 1, & \text{if event polarity is positive} \end{cases}$$

***Figure 4**: Polarity encoding*

$$I_{SAE}(x,y) = 255 \times \left(\frac{t_p - t_0}{\Delta t} \right)$$

***Figure 5**: Surface Active Events encoding*

Context of application

Vehicle detection

- The chosen context application for the analysis of this technique is the **vehicle detection**: high framerates of event cameras can be particularly useful in context where low response time is expected.

- In order to address this aim, the **Prophesee's GEN1 dataset** has been used.

Context of application

Prophesee GEN1 dataset [2] [3]

This dataset has been built using a **Prophesee's GEN1 sensor** (304x240 sensor) mounted on a car dashboard. It features:

- 39 hours of videos
- 228123 cars
- 27658 pedestrians

Bounding boxes (for cars and pedestrians) are annotated with a frequency between 1 and 4Hz.

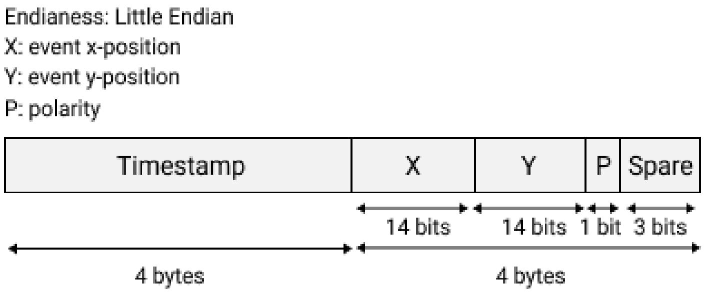

***Figure 6**: Prophesee's GEN1 event encoding*

Context of application

YOLOv3 object detector [4]

YOLOv3 and its tiny version have been chosen as models to perform object detections over the encoded frames

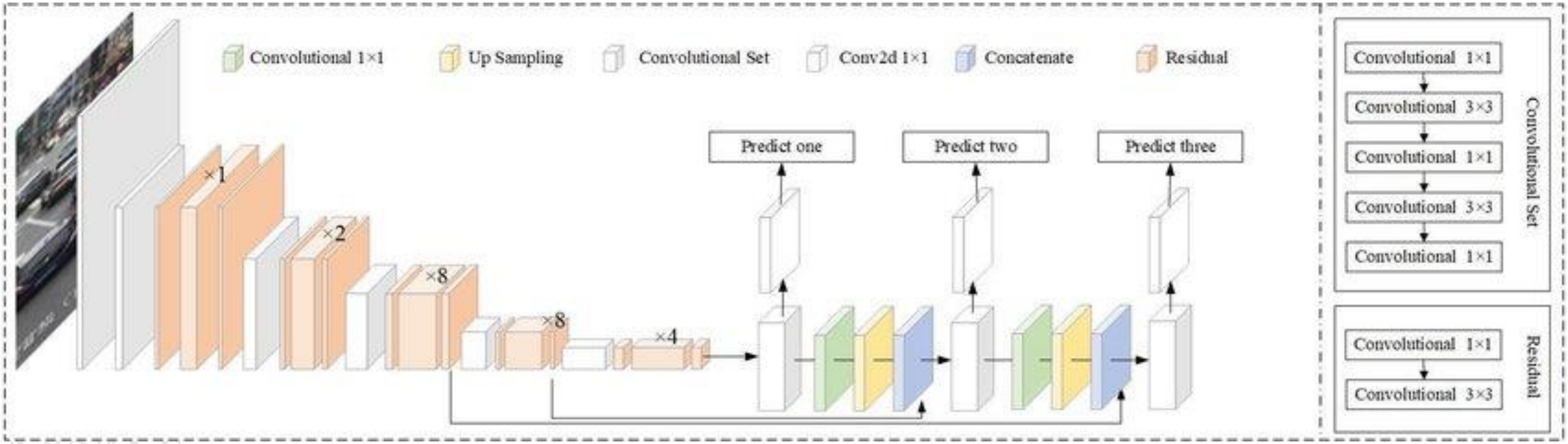

***Figure 6**: YOLOv3 architecture*

Workflow

Steps

1. Develop an encoder application to build datasets of frames starting from the Prophesee's event dataset;

2. Train YOLOv3 and YOLOv3-tiny with the created encoded datasets;

3. Analyze the precision of the trained models with different encoding parameters (accumulation times, bits);

4. Compare the results against other baseline encoding methods (polarity, SAE).

Workflow

Encoder application [4]

The developed encoder application allows to encode event data into frames in a structure compliant with the one requested from the YOLOv3 implementation. Frames can be reconstructed from events using TBR, Polarity and SAE methods.

Figure 7: *TBR encoding – 5ms, 8bits*

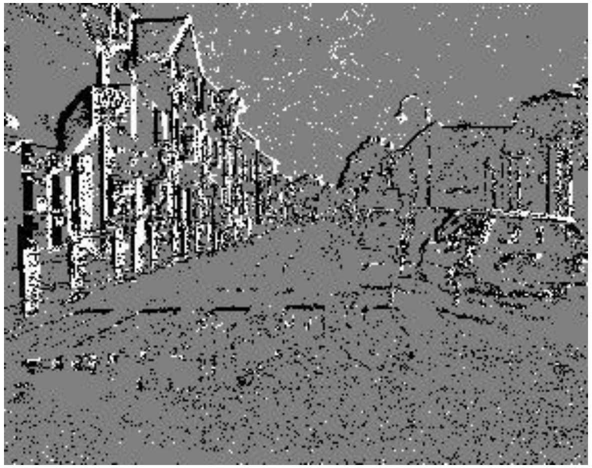

Figure 8: *Polarity encoding – 40ms*

Figure 9: *SAE encoding – 40ms*

Workflow

Encoder application [5]

- Using a stream of events from a file of the dataset, frames are encoded using the requested parameters (accumulation times, bits).

- Then, using the dataset's annotated bounding boxes, only the frames that contains at least a bounding box are saved (along with YOLOv3 compliant bboxes), to be later feed to the detector.

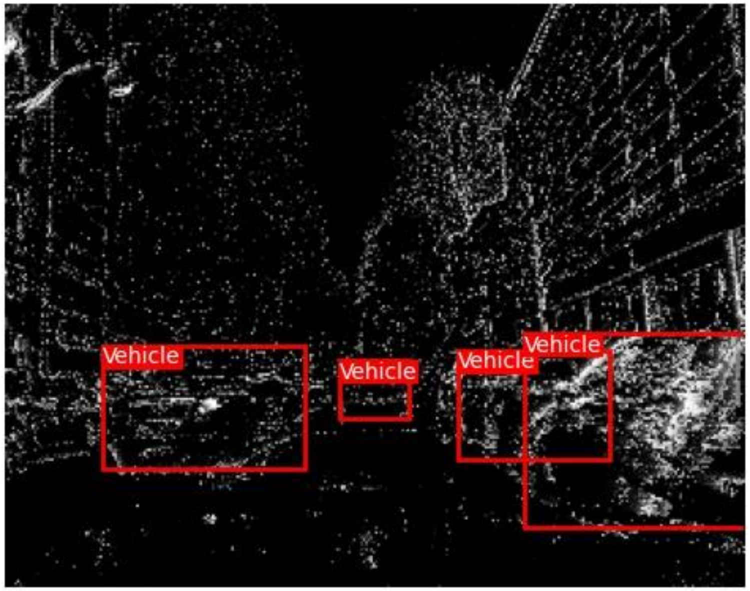

***Figure 10**: TBR encoded frame with annotated bounding boxes represented*

Workflow

Datasets creation

A subset of videos from Prophesee's dataset have been selected in order to build the datasets of encoded frames:

- **257 training videos** - 12965 images, 24128 car bboxes, 3755 pedestrian bboxes
- **75 validation videos** - 3191 images, 6644 car bboxes, 586 pedestrian bboxes
- **75 test videos** - 3856 images, 7311 car bboxes, 1034 pedestrian bboxes

Workflow

Datasets creation

Then, datasets with different encoding methods and parameters have been built:

- TBR datasets - 1-20ms/8bits, 4-8-16bits/2.5ms

Figure 11: *Temporal Binary Representation encoded with the following accumulation times are represented (left to right): 1ms, 2.5ms, 10ms, 20ms.*
Each frame is encoded using 8 bits

Workflow

Datasets creation

Then, datasets with different encoding methods and parameters have been built:

- Polarity datasets - 10/20/40ms

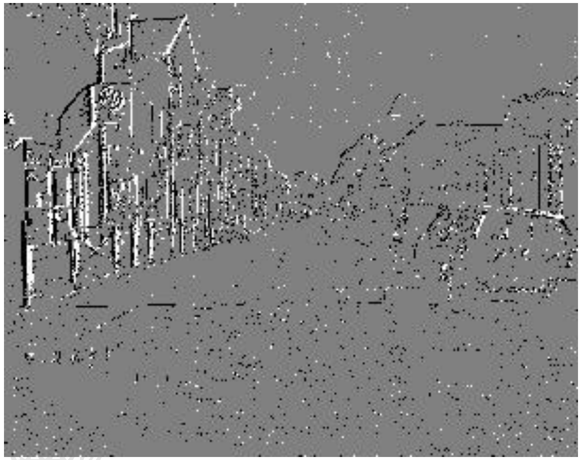

Figure 12: *Polarity frame, 10ms*

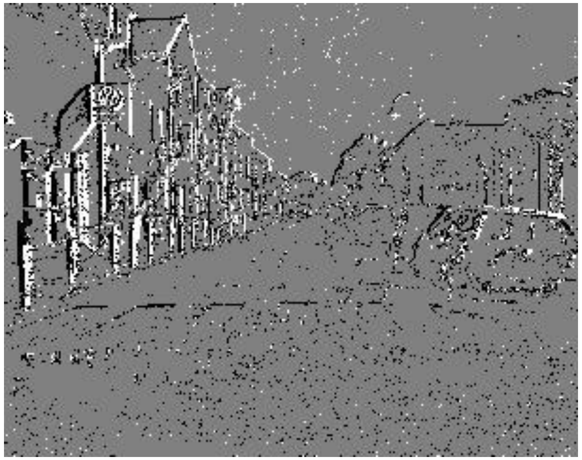

Figure 13: *Polarity frame, 20ms*

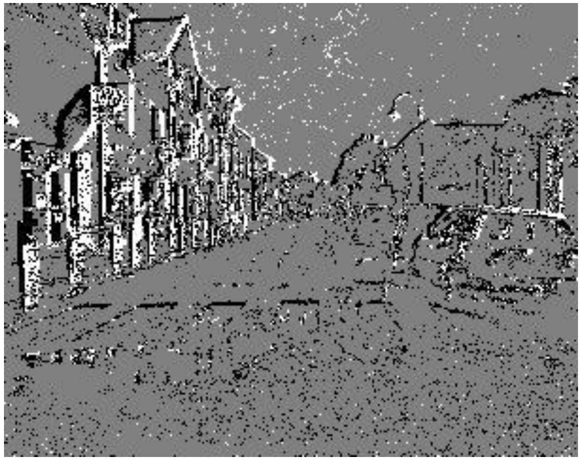

Figure 14: *Polarity frame, 40ms*

Workflow

Datasets creation

Then, datasets with different encoding methods and parameters have been built:

- Surface Active Event datasets - 10/20/40ms

***Figure 15**: SAE frame, 10ms*

***Figure 16**: SAE frame, 20ms*

***Figure 17**: SAE frame, 40ms*

Workflow

Detector training

YOLOv3 and YOLOv3-tiny models have been trained using the previously built datasets, with the following parameters:

- 100 epochs
- 0.001 learning step
- batch size = 8

Workflow

Evaluation protocol

Once YOLOv3 and YOLOv3-tiny models have been trained on all datasets, the following values have been evaluated against the test set:

- TBR trained-models precision/recall
- TBR trained-models mAP
- TBR trained-models inference times
- Polarity trained-models mAP
- SAE trained-models mAP

Mean average precision have been evaluated both with YOLOv3 implementation's evaluator and Prophesee's evaluator (COCO ap50 [6])

Quantitative Results

YOLOv3 vs YOLOv3-tiny precision/recall comparison

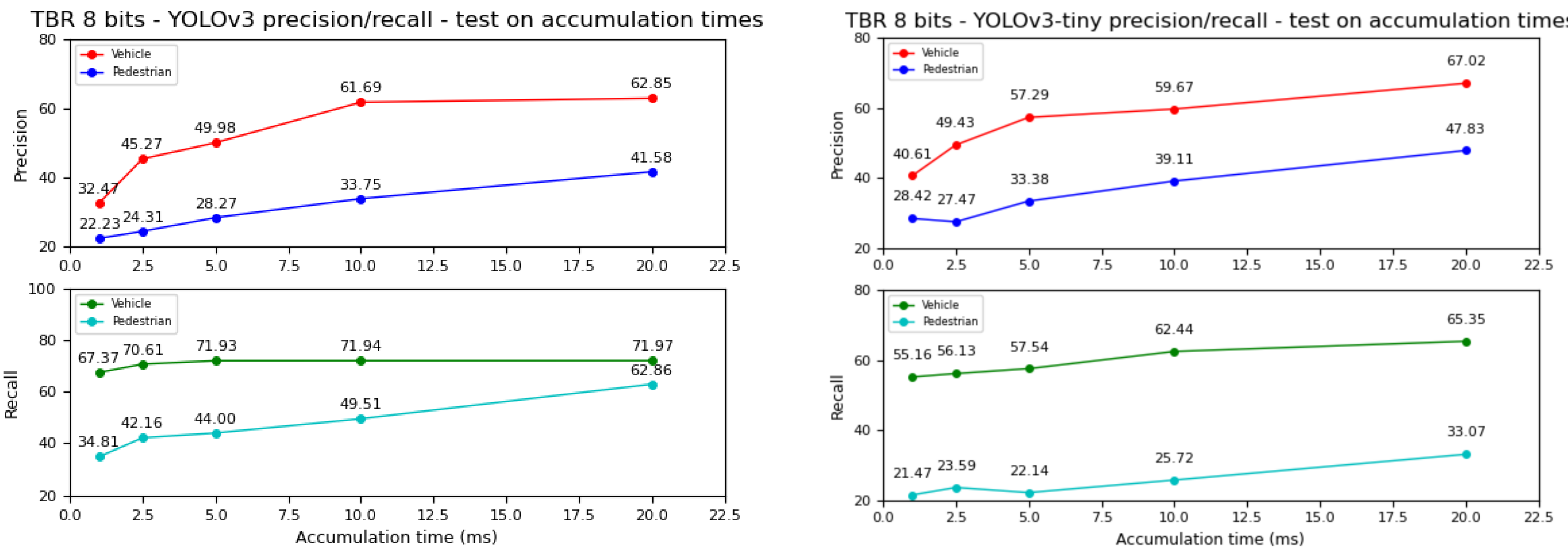

***Figure 18**: precision/recall comparison between YOLOv3 and YOLOv3-tiny models trained on TBR frames with a fixed number of bits (8)*

Quantitative Results

YOLOv3 vs YOLOv3-tiny mAP comparison

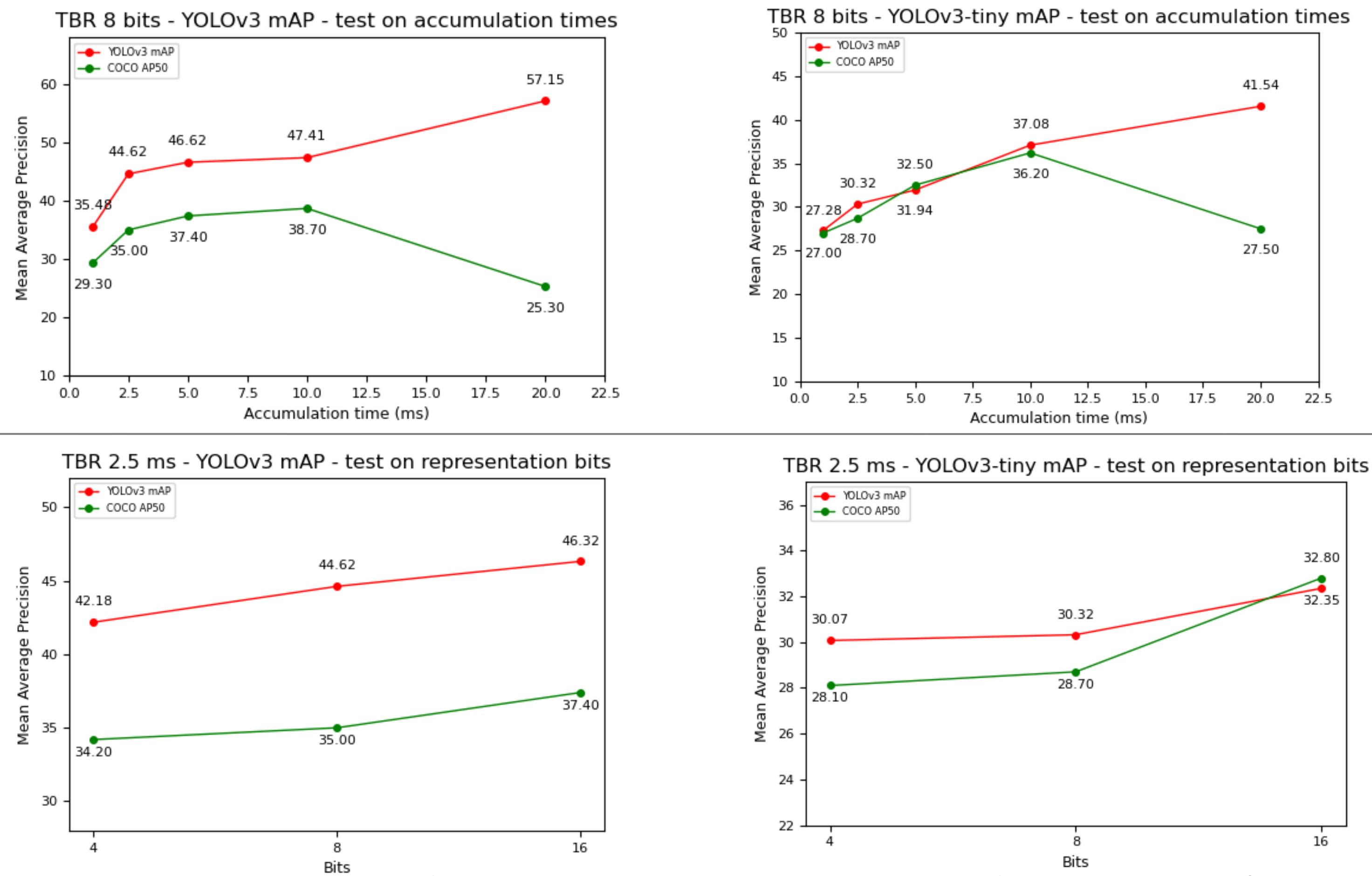

Figure 19*: mAP comparison between YOLOv3 and YOLOv3-tiny models trained on TBR frames*

Quantitative Results

TBR vs Polarity and SAE techniques

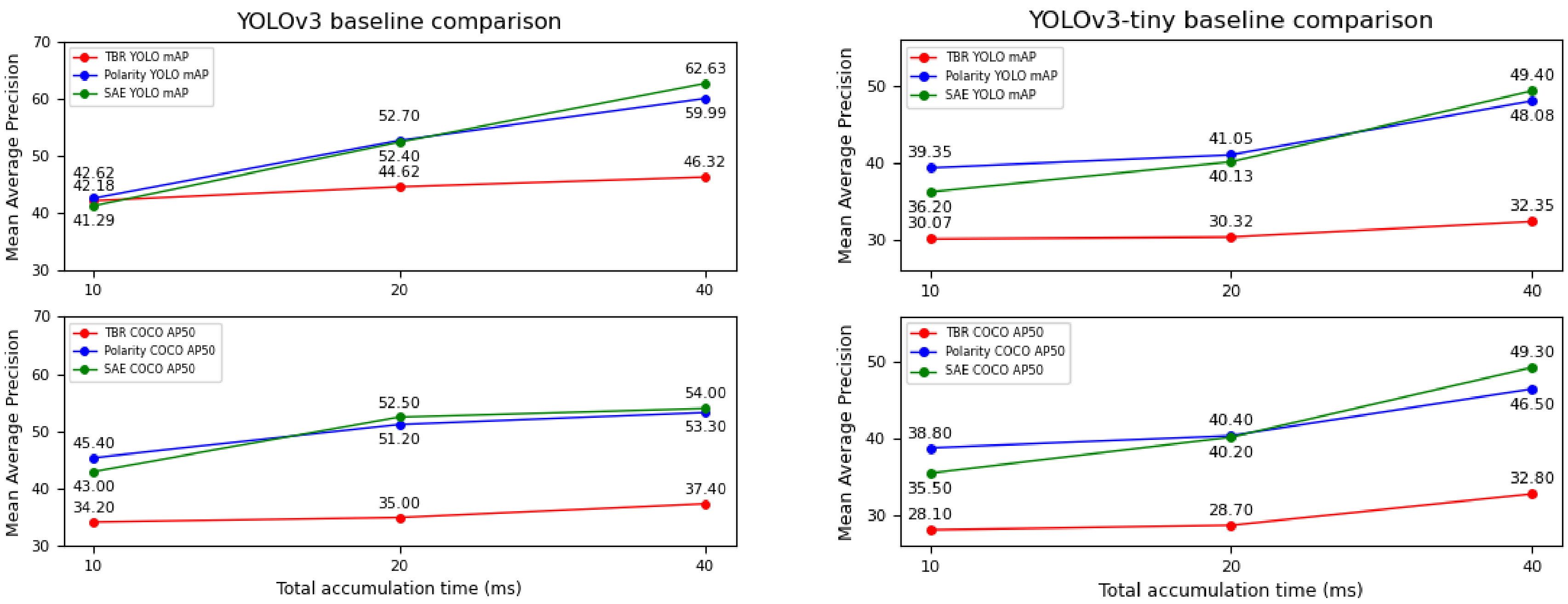

***Figure 20**: comparison between YOLOv3 and YOLOv3-tiny models trained on TBR, Polarity and SAE frames*

Quantitative Results

YOLOv3 vs YOLOv3-tiny inference performances

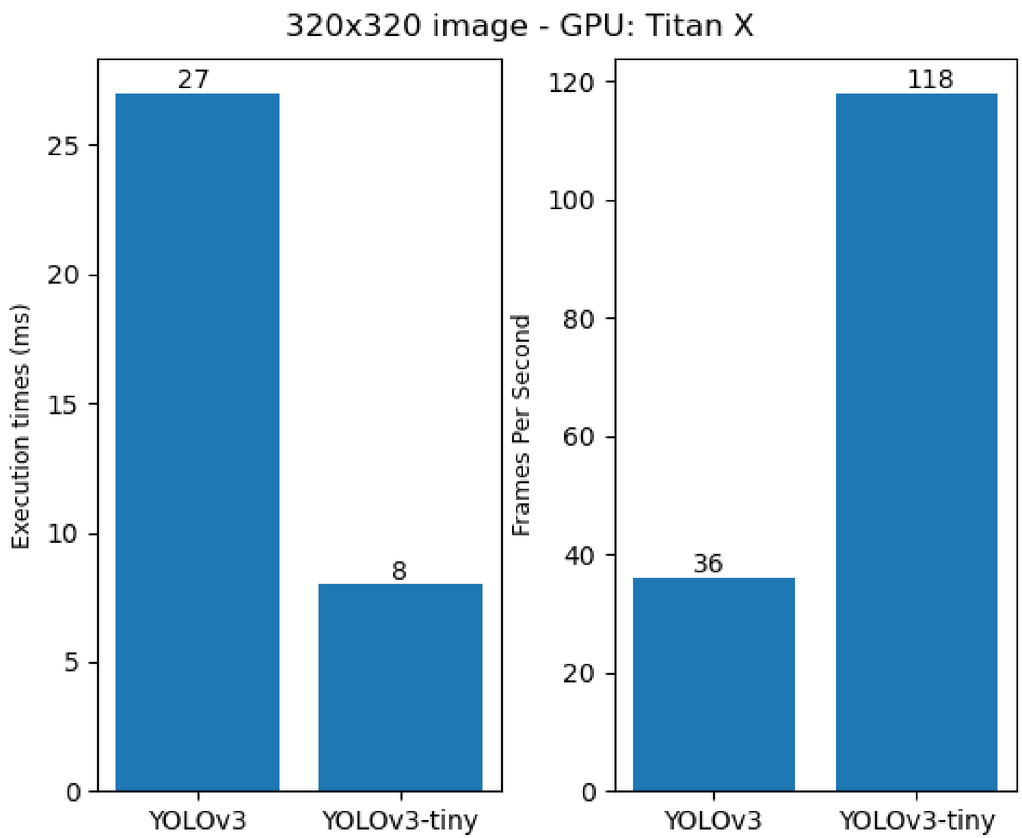

***Figure 21**: comparison between YOLOv3 and YOLOv3-tiny models on inference times*

Qualitative Results

YOLOv3 vs YOLOv3-tiny comparison

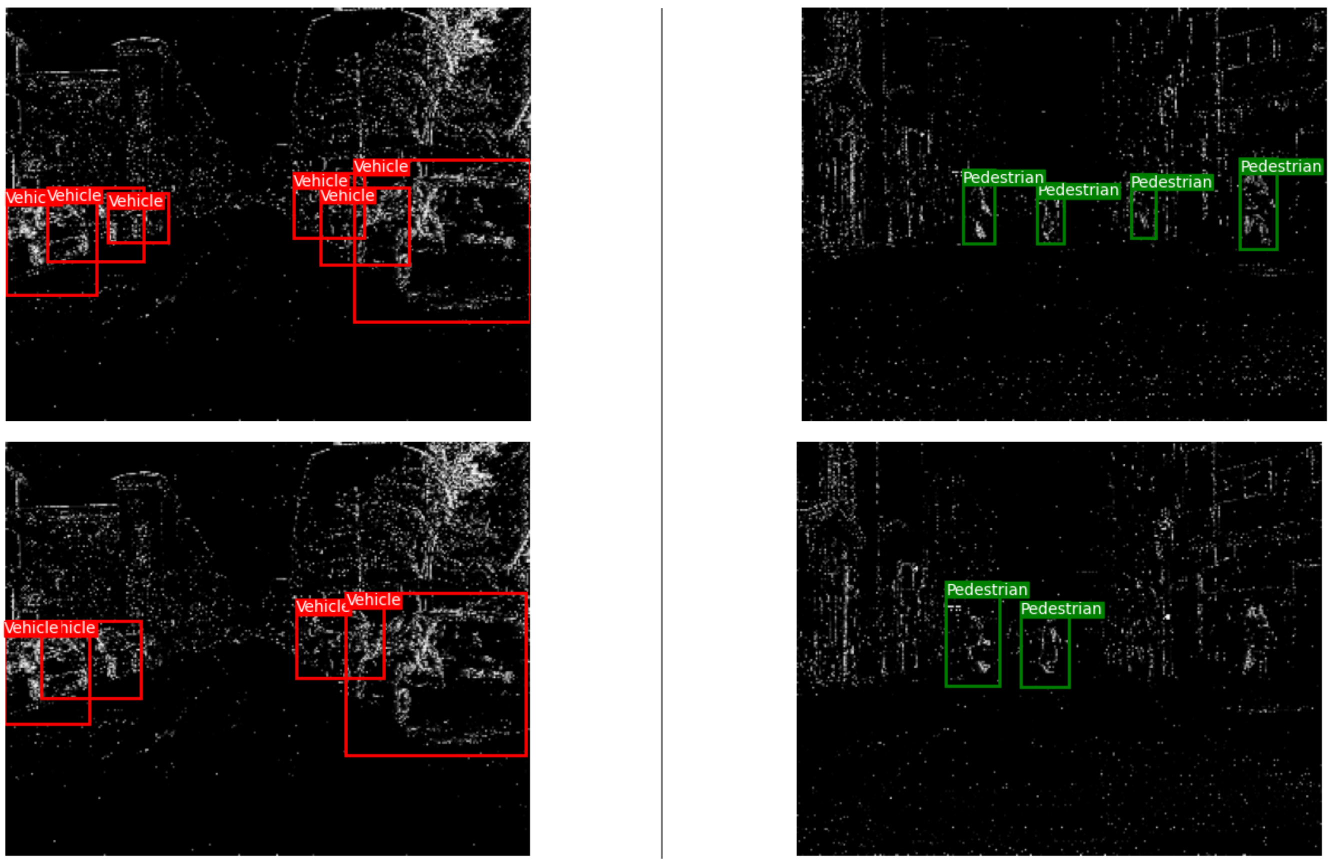

***Figure 22**: detections on test events with TBR YOLOv3 (up) and YOLOv3-tiny (down) models*

Qualitative Results

TBR vs Polarity and SAE encoding techniques

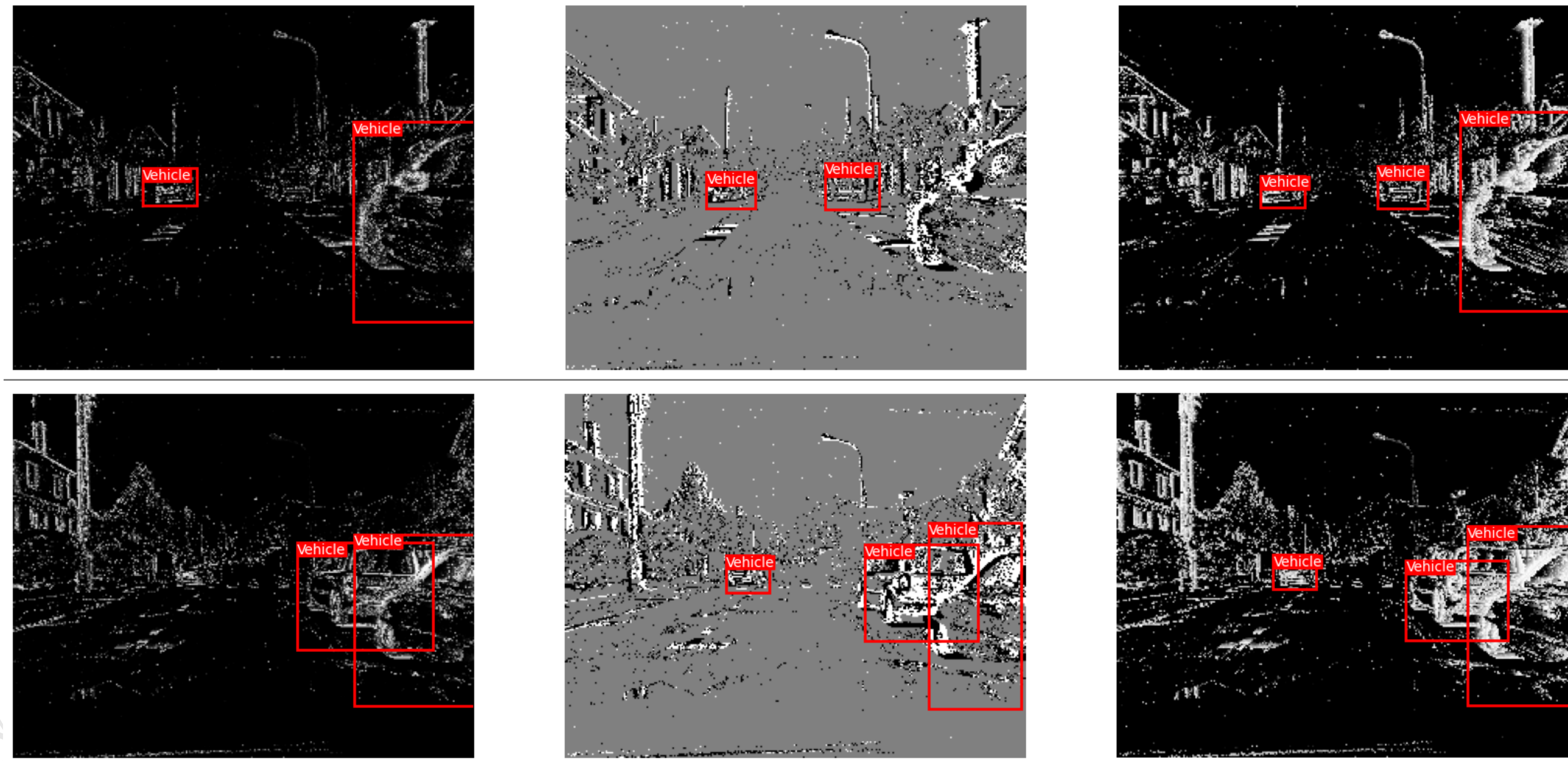

***Figure 23**: detections on test events with TBR (left), Polarity (center) and SAE (right) models*

Conclusions

- A framework for event object detection has been developed.
  The framework includes:
  1. **Prophesee's GEN1 dataset**: event-based sensor in automotive context;
  2. **Encoder application**: converts events to frames (TBR, polarity, SAE);
  3. **YOLOv3 implementation**: detects objects on encoded frames.

- YOLOv3 and its tiny version have been trained on various encoded datasets.

- Evaluation of **mAP**, **precision/recall** and **inference times** for the **TBR trained models** have been done. Good results have been obtained in both quantitative and qualitative terms.

- Moreover, evalutation of **mAP** for **Polarity** and **SAE trained models** have been done. This test shows an higher precision of these models than the TBR one.

References

1. Temporal Binary Representation - https://arxiv.org/pdf/2010.08946.pdf
2. Prophesee's GEN1 dataset - https://www.prophesee.ai/2020/01/24/prophesee-gen1-automotive-detection-dataset/
3. Prophesee's GEN1 dataset paper - https://arxiv.org/pdf/2001.08499.pdf
4. Pytorch YOLOv3 implementation - https://github.com/eriklindernoren/PyTorch-YOLOv3
5. Framework repository - https://github.com/francescoareoluci/tbr-event-object-detection
6. COCO evaluation - https://cocodataset.org/#detection-eval