SAPIENZA
UNIVERSITÀ DI ROMA

DIAG
Dipartimento di Ingegneria
informatica, automatica e gestionale
Antonio Ruberti

PEDESTRIAN CROSSING PREDICTION FOR ENHANCED AUTONOMOUS DRIVING SAFETY

● PROJECT PRESENTATION

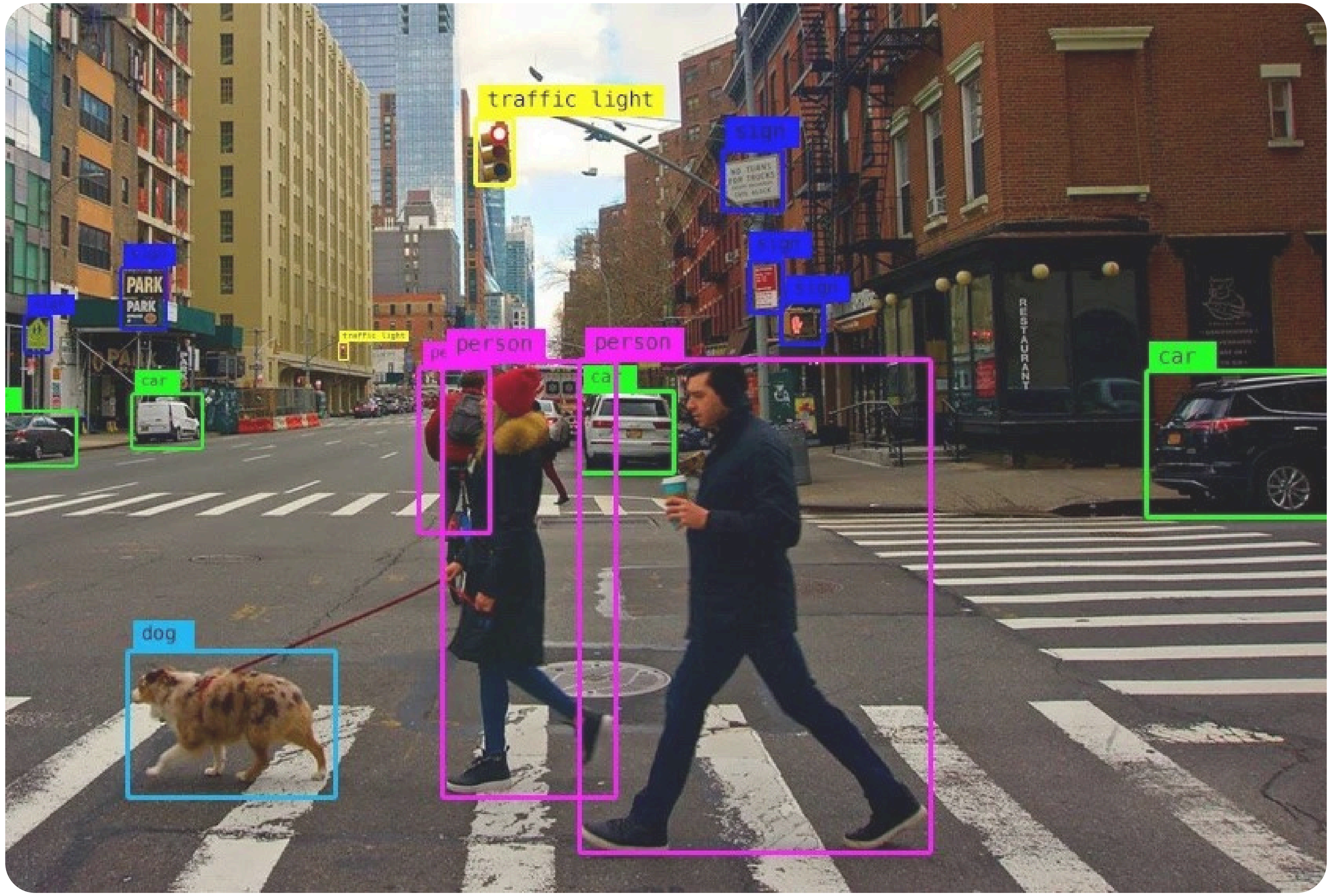
Gianmarco Donnesi
Matr. n. 2152311

Michael Corelli
Matr. n. 1938627

A.Y. 23/24

PROBLEM AND MOTIVATION

- **Pedestrian intention estimation** in urban environments is crucial for autonomous driving systems.
- **Predicting** whether a pedestrian will cross the street in real-time is complex due to the unpredictable nature of human actions.
- Enhancing **safety** and **reliability** in autonomous driving by accurately predicting pedestrian behavior.



OVERVIEW OF OUR APPROACH

JAAD Dataset

Our model was trained on JAAD dataset, which includes annotated video sequences of pedestrian behaviors, providing a comprehensive source for training and evaluating our model.

Bboxes & Pose Keypoints

Used bounding box annotations to track pedestrian movements across video frames and extracted pose keypoints to enhance the model's understanding of pedestrian behavior.

VGG19 + LSTM

Implemented a modified version of the pre-trained VGG19 Convolutional Neural Network (CNN) followed by a Long Short-Term Memory (LSTM) network to analyze spatial and temporal patterns in pedestrian movements.

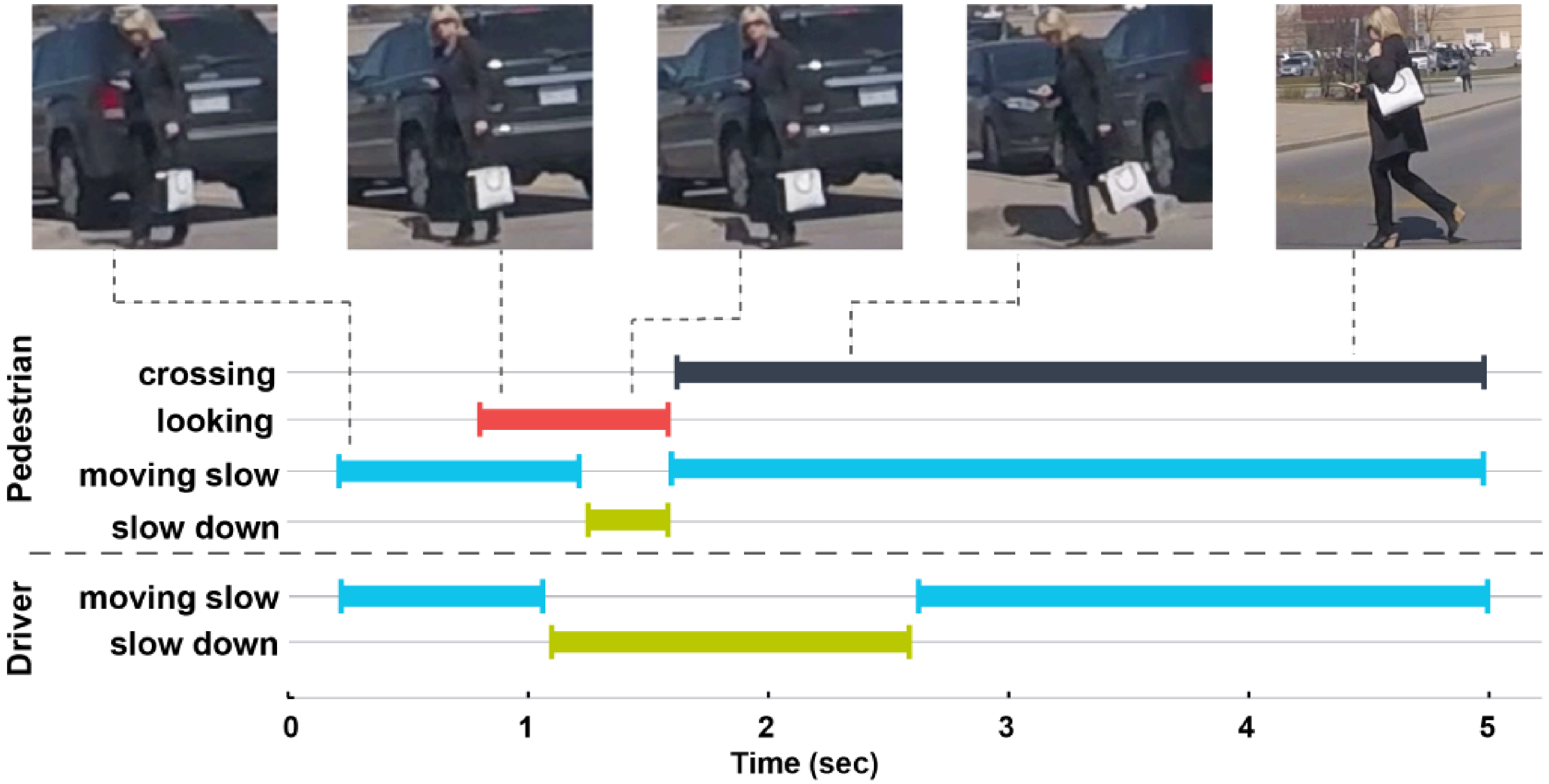
Performance Evaluation

Evaluated model performance using standard metrics such as accuracy, recall, and F1-score, ensuring robust and reliable predictions of pedestrian crossing behavior.

JAAD 2.0 DATASET

- The **JAAD** (*Joint Attention for Autonomous Driving*) 2.0 dataset is designed for research in autonomous driving, focusing on **pedestrian behavior** and **intention prediction**.

- It includes annotated video sequences of pedestrian behaviors captured in various urban environments.



- Provides precise localization of pedestrians in video frames using **Bounding Boxes** coordinates and indicates the level of visibility of each pedestrian including **occlusion informations**.

[1] Rasouli, Amir, Iuliia Kotseruba, and John K. Tsotsos. "*Are they going to cross? A benchmark dataset and baseline for pedestrian crosswalk behavior*." In Proceedings of the IEEE International Conference on Computer Vision Workshops, pp. 206-213. 2017.

[2] Rasouli, Amir, Iuliia Kotseruba, and John K. Tsotsos. "*Agreeing to cross: How drivers and pedestrians communicate.*" In IEEE Intelligent Vehicles Symposium (IV), pp. 264-269. 2017.

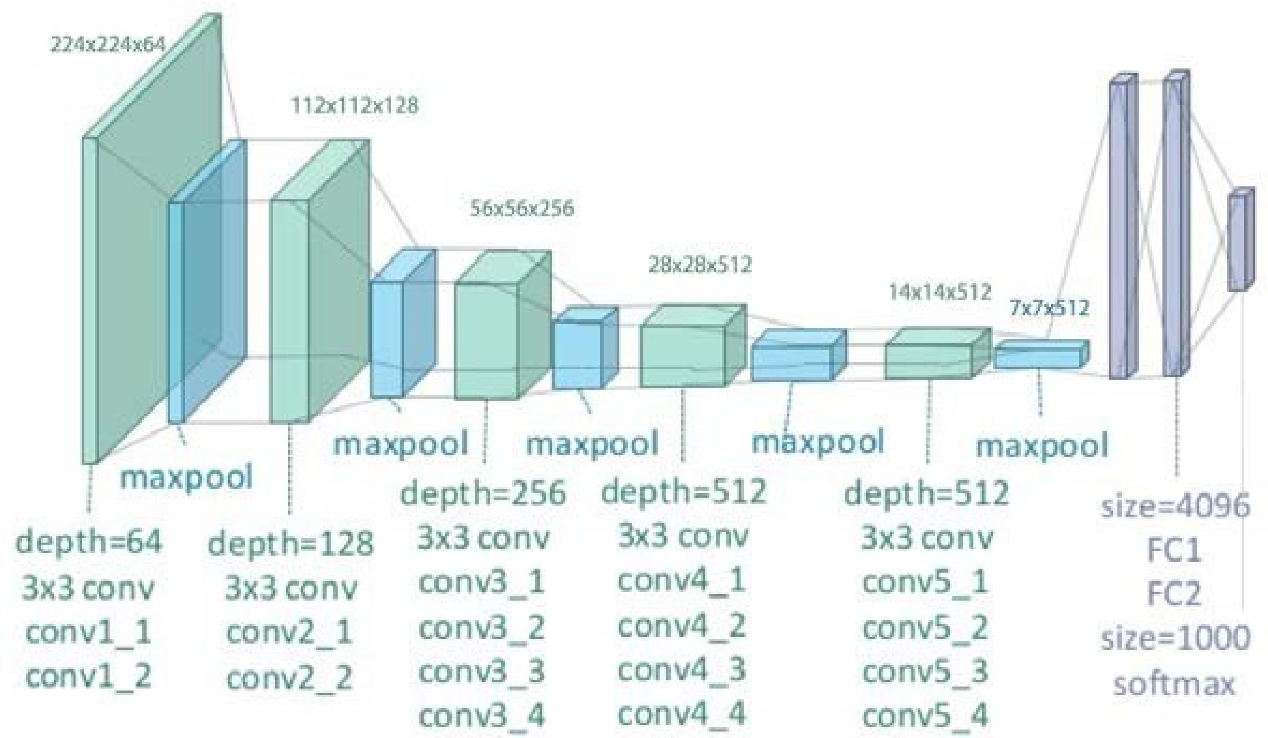
JAAD 2.0 DATASET - ANNOTATIONS

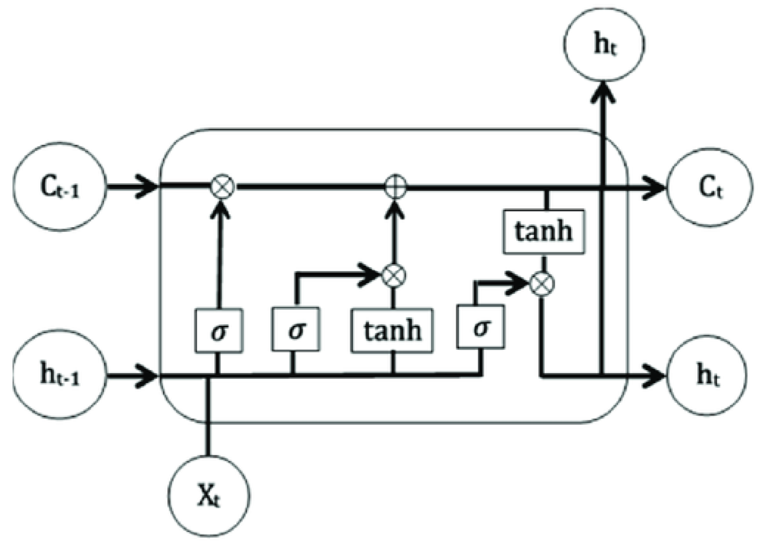
Annotation Types:

- **Generic Annotations**: Video attributes (time of day, weather, location), pedestrian bounding box coordinates, occlusion information, and activities (e.g., walking, looking).
- **Attributes**: Information regarding pedestrian demographics, crossing points, crossing characteristics.
- **Appearance**: Pedestrian appearance details such as pose, clothing, objects carried (high visibility videos).
- **Traffic**: Information about traffic signs and traffic lights for each frame.
- **Vehicle**: Vehicle actions per frame (e.g., moving fast, speeding up).

MODEL ARCHITECTURE: VGG19 + LSTM

Our model combines the **VGG19 Convolutional Neural Network** (CNN) for spatial feature extraction and **Long Short-Term Memory (LSTM)** networks for temporal pattern analysis.





VGG19 Feature Extraction:

- Pre-trained VGG19 model is used to extract features from the input images.
- Layers up to the 36th are frozen to utilize pre-trained weights and reduce training time.
- The extracted features are pooled and passed through the VGG19 classifier (excluding the final layer).

LSTM for Temporal Analysis:

- The output from the VGG19 is reshaped and fed into an LSTM network.
- LSTM captures the temporal dependencies in the sequence of frames, crucial for understanding pedestrian behavior over time.

MODEL ARCHITECTURE: SOFT ATTENTION MECHANISM

→ **Attention Weights:**

- The LSTM outputs a sequence of hidden states.
- Each hidden state is passed through a series of linear layers and ReLU activation to compute **attention scores**.
- The attention scores are normalized using a softmax function to produce **attention weights.**

→ **Context Vector:**

- The attention weights are applied to the LSTM hidden states.
- A weighted sum of the hidden states is computed, resulting in a **context vector**. This vector captures the most relevant temporal features from the sequence.

```python
# Attention mechanism
class SoftAttention(nn.Module):
    def __init__(self, hidden_dim):
        super(SoftAttention, self).__init__()
        self.hidden_dim = hidden_dim
        # Attention network: 2 linear layers + ReLU activation
        self.attention = nn.Sequential(
            nn.Linear(hidden_dim, hidden_dim),
            nn.ReLU(inplace=True),
            nn.Linear(hidden_dim, 1)
        )

    def forward(self, lstm_output):
        # lstm_output: [batch_size, seq_len, hidden_dim]
        # Calculate attention weights
        attn_weights = self.attention(lstm_output)
        # Normalize the attention weights using softmax
        attn_weights = torch.softmax(attn_weights, dim=1)
        # Compute the context vector as a weighted sum of LSTM outputs
        context = torch.sum(attn_weights * lstm_output, dim=1)
        return context, attn_weights
```

BOUNDING BOXES & KEYPOINT EXTRACTION WITH MEDIAPIPE

- **Pose Estimation:** the frame is resized and converted to RGB before being processed by MediaPipe to detect pose keypoints.
- **Bounding Boxes**: Annotated on each frame to capture pedestrian locations.
- **Drawing and Saving Keypoints:** if pose landmarks are detected, the keypoints are extracted, drawn on the frame, and saved as '*.npy*' file for each image.

- **Media Pipe** is an open-source framework from Google that provides tools for working with media data or for media processing.
  - Provides high-accuracy pose estimation.
  - Efficient and works in real-time, suitable for processing video frames.

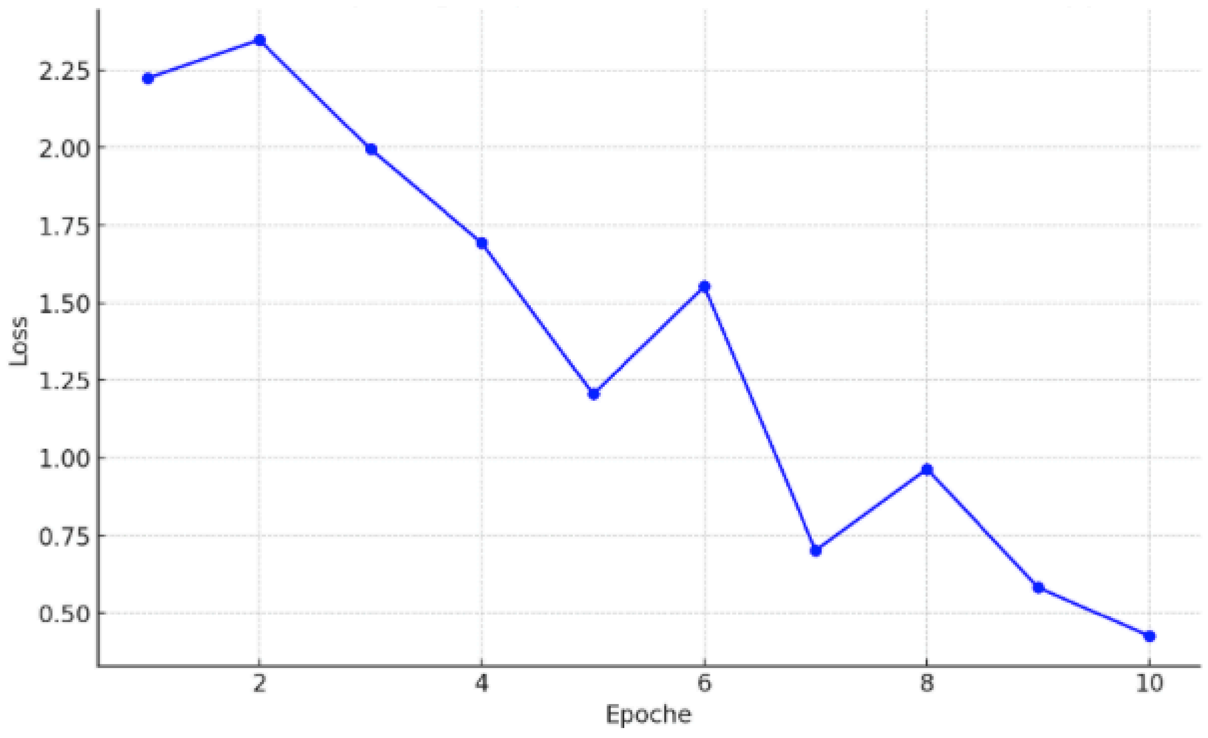
Google

MediaPipe

MODEL TRAINING AND VALIDATION

The JAAD dataset was split into **training** and **validation** sets with:

- *.pkl* files contain preprocessed video data
- *.pt* files contain the processed frames, keypoints, and additional information: traffic, vehicle, appearance and attributes.



Train:

- **Mixed Precision Training**: GradScaler and autocast are used for mixed precision training to speed up computation and reduce memory usage.
- **Optimizer and Scheduler**: the Adam optimizer and StepLR learning rate scheduler are set up.
- **Loss Criterion**: the loss function is defined as BCEWithLogitsLoss.

Validation:

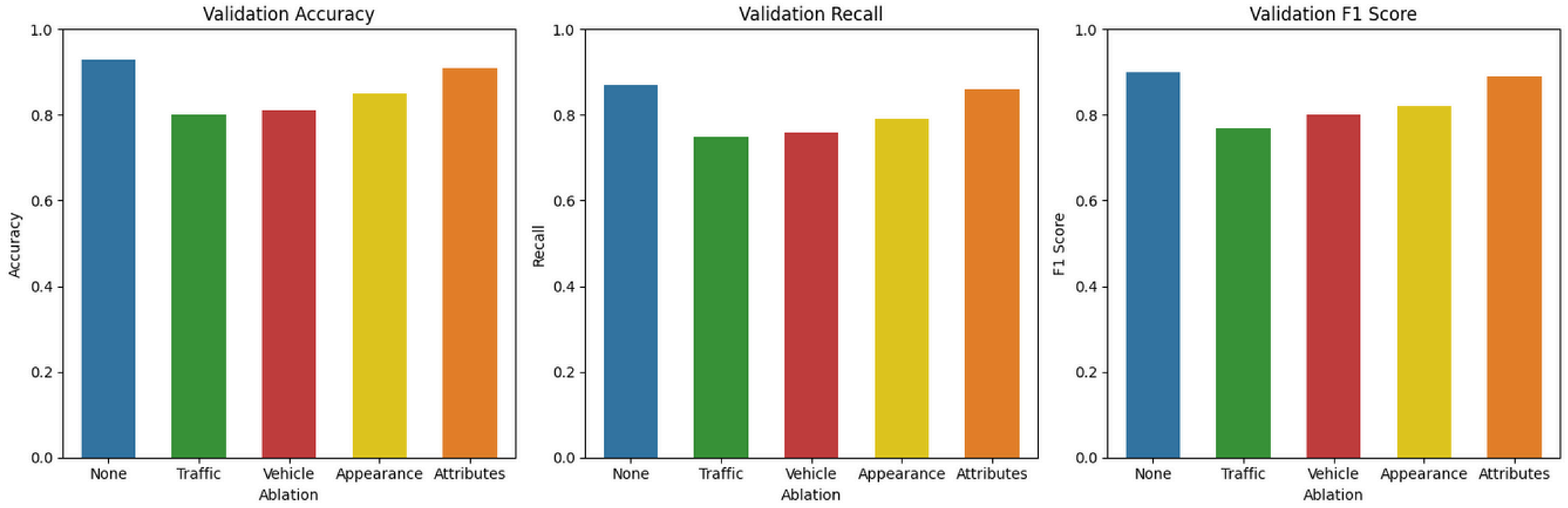
- average loss
- accuracy
- recall
- F1 score

Depending on the specified ablation type, the corresponding feature is zeroed out to study its impact on model performance.

PERFORMANCE EVALUATION: ABLATION STUDIES

Ablation Studies

- Evaluates the model with and without specific features to understand their impact on performance.
- The function iterates over different ablation types (or no ablation)



Results

- **Ablation "traffic" and “vehicle“:**
  - The removal of traffic information drastically reduces model performance, underscoring its importance for accurate predictions.
  - In contrast, removing vehicle information has a less severe impact, suggesting the model can still perform relatively well without it.