Introduction

Motivation:
- Vision Transformer-based trackers are computationally expensive due to a large number of model parameters.
- They require specialized hardware for real-time inference.

<table>
<thead>
<tr>
<th>Tracker</th>
<th>GOT10k OR ↑</th>
<th>SB150 OR ↑</th>
<th>TrackingNet AUC ↑</th>
<th>PSNR ↑</th>
<th>#params (m)</th>
<th>fps CPU ↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>DiMP-50</td>
<td>0.611</td>
<td>0.717</td>
<td>74.0</td>
<td>80.1</td>
<td>26.1</td>
<td>15.0</td>
</tr>
<tr>
<td>TransT</td>
<td>0.671</td>
<td>0.768</td>
<td>81.2</td>
<td>85.4</td>
<td>23.0</td>
<td>2.3</td>
</tr>
<tr>
<td>STARK-ST101</td>
<td>0.688</td>
<td>0.781</td>
<td>82.0</td>
<td>86.9</td>
<td>47.2</td>
<td>7.8</td>
</tr>
<tr>
<td>OTrack-384</td>
<td>0.740</td>
<td>0.835</td>
<td>83.9</td>
<td>88.5</td>
<td>92.1</td>
<td>4.4</td>
</tr>
<tr>
<td>MixFormer-L</td>
<td>0.756</td>
<td>0.857</td>
<td>83.9</td>
<td>88.9</td>
<td>183.9</td>
<td>&lt;5</td>
</tr>
</tbody>
</table>

- **Our Solution**: Mobile Vision Transformer for fast tracking.

### Key Contributions

**Mobile Vision Transformer-based backbone:**
- Cascade of Convolutional and Transformer blocks for feature extraction.
- Convolutional blocks model the spatially local information.
- Transformer blocks capture the long-range feature dependencies.

**Feature fusion in tracker backbone:**
- Self-attention on the concatenated template and search region features.
- Exchange of information **within** and **between** the two regions.

**High inference speed:**
- Joint feature extraction and fusion requires fewer attention operations.
- 175 fps on GPU and 29 fps on CPU (Pytorch).
- 300 fps on GPU (TensorRT) and 70 fps on CPU (ONNX Runtime).

### Proposed Mobile Vision Transformer-based Tracker (MVT)

**Proposed Backbone:**
- Cascaded MobileNetV2 (or **MV2**) and Siam-MoViT blocks for feature extraction.
- **Siam-MoViT** block fuses features from the two branches.

**Neck Module:**
- Cross-correlation between template and search region features.

**Head Module:**
- Two fully-convolutional branches for classification and bounding box regression.

**Ablation Study on Feature Fusion:**
- Classification \((L_{cls})\) and regression \((L_1 \text{ and } L_{giou})\) losses.
- Overall training loss, \(L_{total} = L_{cls} + \lambda_1 \cdot L_1 + \lambda_2 \cdot L_{giou}\).

### Results

**Implementation Details:**
- The template and search region dimensions are 128 \(\times\) 128 and 256 \(\times\) 256.
- GOT10k-train dataset for training the model.
- Training for 100 epochs with a batch size of 128.
- The learning rate is set to \(4 \times 10^{-4}\) with cosine annealing as the scheduler.
- Initialization of our tracker backbone using pretrained MobileViT weights.
- During inference, we apply Hanning window on classification score map.

**Comparison to Related Lightweight Trackers:**

**MVT** has the best performance on server-based GOT10k and TrackingNet.
- **Overall**, MVT outperforms the related trackers in 7 out of 10 metrics.

**Comparison to State-of-the-art:**

- **State-of-the-art**: Deployment of transformers has improved the performance, but at the cost of lowered tracking speed.
- In contrast, our MVT surpasses DiMP-50 with 4.7x fewer parameters while running at 2.8x and 2x its speed on GPU and CPU, respectively.

**Analysis**

**Ablation Study on Feature Fusion:**
- We retrain our model without concatenating the template and search region features inside the proposed Siam-MoViT block.

**Proposed feature fusion improves **AUC** and reduces **FR** on average.

**Robustness Analysis:**
- We compare the **FR** of our MVT on attributes from the LaSOT dataset.
- **MVT** is robust to target deformation and appearance changes.
- **MVT** has a higher **FR** while tracking small, fast-moving target objects, e.g., volleyball.

### Conclusion

- We proposed a tracker that uses Mobile Vision Transformer, for the first time.
- Our tracker performed better than the related lightweight trackers, especially on server-based GOT10k and TrackingNet datasets.
- **MVT** runs at 70 fps on GPU, faster than second-best Stark-Lightning (50 fps).
- **Future work**: Deployment on embedded devices (e.g., smartphones).

**Project Webpage:** Tracker Code & Model