BRNO FACULTY
UNIVERSITY OF INFORMATION
OF TECHNOLOGY TECHNOLOGY

http://fit.vut.cz

KNN Project report
2023/2024

# Segmentation tracking in video

Eva Mičánková, Lenka Šoková, David Kedra

**Abstract**
This project explores the application of single object tracking by predicting subsequent object masks. The method involves processing the current image and the previous mask together with the prior image and its associated mask, utilizing a Siamese UNet architecture. The method was trained and validated on a specially adapted MOSE dataset and evaluated using the DAVIS dataset. Results indicate that the proposed method can tracks objects, demonstrating its potential utility in real-time video analysis applications.

**Keywords:** U-Net — Fully-convolutional Siamese network — Video object segmentation

**Supplementary Material:** Baseline solution — Final model solution — Pretrained model

## 1. Task definition

The task of *Segmentation Tracking in Video* focuses on the visual tracking of a selected object in a video. The process is initiated by the user marking the target object for segmentation in the first frame of the video—this pertains to a single object in *Single Object Segmentation Tracking*, while *Multi Object Segmentation Tracking* involves multiple objects. The object may be marked by a bounding box (the object within the box is segmented) or possibly by directly creating a segmentation mask. Based on this initialization, the model predicts the segmentation mask for the selected object in subsequent video frames. In our project, we have focused on *Single Object Segmentation Tracking*.

## 2. Review of existing solutions

**SiamMask** [1] utilizes a fully convolutional Siamese network for real-time tracking and segmentation of video objects, initialized with just a bounding box for both single and multi-object tasks. It offers dual and triple branch architectures; however, we focus on the dual-branch for single object tracking and segmentation. Each image area (search image $x$ and target image $z$) is processed by the same CNN to extract features. These features drive a deep cross-correlation computation resulting in a response map $g_\theta$, indicating likely target positions. In the dual-branch version, one branch handles classification ($p_\omega$), distinguishing target from background, while the segmentation branch ($h_\phi$) outputs a segmentation mask for each Region of Window (RoW):

$$m^n = h_\phi(g_\theta^n(x,z)) \tag{1}$$

This setup uses both the search and reference images for mask prediction and initialization.

**Lucid Tracker** [2] employs a streamlined methodology for object segmentation in video streams, leveraging a minimal dataset while emphasizing the critical role of optical flow. The architecture of Lucid Tracker is designed to efficiently transfer a segmentation mask from one frame to the next. This process uses the current image $I_t$ and optical flow information $||F_t||$ to update the mask from $M_{t-1}$ to $M_t$, as expressed by the equation:

$$M\_t = f(I\_t, F\_t, M\_t - 1) \tag{2}$$

This model capitalizes on the smooth and predictable motion of objects across frames, utilizing $M_{t-1}$ as a foundational estimate for $M_t$. It enhances this estimate by incorporating additional data from the current image and the optical flow. The optical flow is accurately estimated using FlowNet2.0, calculated by:

$$F\_t = h(I\_t - 1, I\_t) \tag{3}$$

**AGUnet** [3] is designed for object tracking in video sequences, offering an efficient balance between speed and accuracy. Unlike Lucid Tracker, which requires complex computation of optical flow for each frame, AGUnet operates with reduced temporal demands.

AGUnet incorporates a dual-component architecture: *1. Fully Convolutional Siamese Network:* Utilizes identical transformations, $\phi$, on both reference ($IR$) and new ($I$) images to assess similarity through the function $g(\cdot)$, resulting in the expression:

$$f(IR, I) = g(\phi(IR), \phi(I))$$

This network generates a base score map, $S_{base}$, reflecting the similarity between the reference image and various areas in the new frame, across multiple scales.

*2. U-net with Skip Connections:* U-net layers utilize skip connections to merge existing feature maps with score maps, enhancing the accuracy of segmentation. A new feature map, defined as the annotation map, combines the feature map $F$ and score map $S$ through:

$$F^{AG} = F \odot S, \quad F^{AG} \in R^{H_F \times W_F \times C_F}$$

# 3. Datasets

**DAVIS2016** dataset [4] is a benchmark in computer vision for training and evaluating single object segmentation systems. This dataset includes 50 video sequences in two resolutions (480p and 1080p), specifically annotated for *Single object segmentation tracking* task.

**MOSE** dataset [5], originally designed for multiple object segmentation tracking in complex environments, includes 2,149 video clips with annotations for 5,200 objects. For this project, focused on single object segmentation tracking, we adapted the dataset by converting multi-object annotated frames into binary frames, each highlighting only one object. This modification process isolates individual objects into separate video sequences, resulting in the same video clip containing multiple sequences, each dedicated to tracking a different object.

# 4. Our solutions

## 4.1 Baseline

Our baseline model utilizes the U-Net architecture [6] typically used for image segmentation. We slightly adjusted the classic U-Net architecture by adding an additional double convolution block to the first layer to enhance its performance for video object segmentation tracking, as shown in Figure 1. Among various architectures tested, this modified version of U-Net achieved the best evaluation scores, marking it as our superior choice for baseline model.

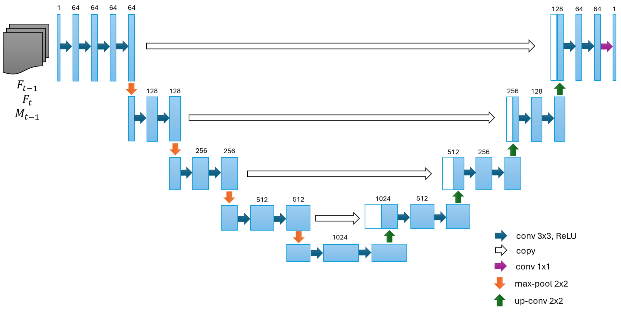


**Figure 1.** Architecture of our baseline model, based on U-Net.

## 4.2 Final model

In our final model shown in Figure 2, we integrated a U-Net architecture with a Siamese VGG11 encoder to enhance precision in feature extraction. This setup exploited the Siamese network's capability for detailed comparative analysis, alongside the robust feature extraction strengths of the VGG11 model. We opted to average the outputs from the dual encoders, a technique that proved effective in stabilizing feature representation and reducing noise. Additionally, we explored other methods of combining encoder outputs, such as Max pooling and Min pooling, but these approaches yielded inferior results. Experiments with a VGG16 encoder were also conducted; however, the Siamese VGG11 configuration demonstrated superior performance in our final evaluations.

# 5. Training

For training we used Dice loss function [7], which is commonly used for image segmentation tasks to measure overlap between the predicted segmentation mask and the ground truth. It ranges from 0 to 1, where 0 indicates no overlap and 1 indicates perfect overlap. It is defined as:

$$L_{dice} = 1 - \frac{2 * |p_{true} \cap p_{pred}| + \varepsilon}{\sum p_{true}^2 + \sum p_{pred}^2 + \varepsilon}, \tag{4}$$

where $p_{true}$ is the ground truth mask, $p_{pred}$ is the predicted mask and $\varepsilon$ is the smoothing constant (e.g., $1e-5$).

Training was performed on powerful machines with GPUs provided by MetaCentrum. The focus was on maximizing the utilization of available resources. By analyzing the processes, we discovered a bottleneck in the loading of images into memory, which was causing low GPU utilization. Therefore, image

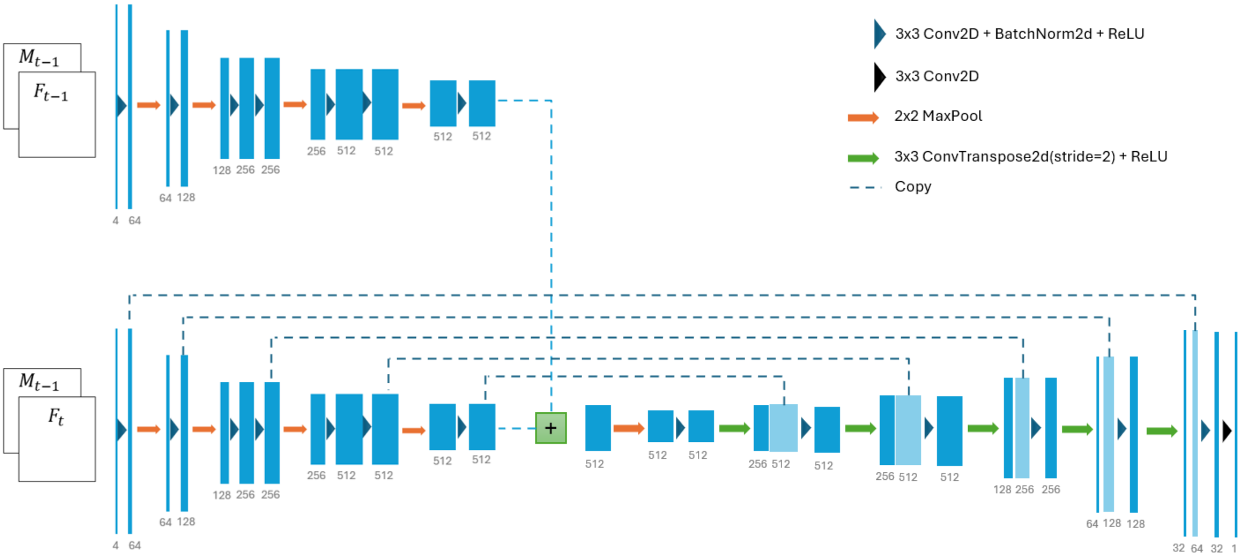
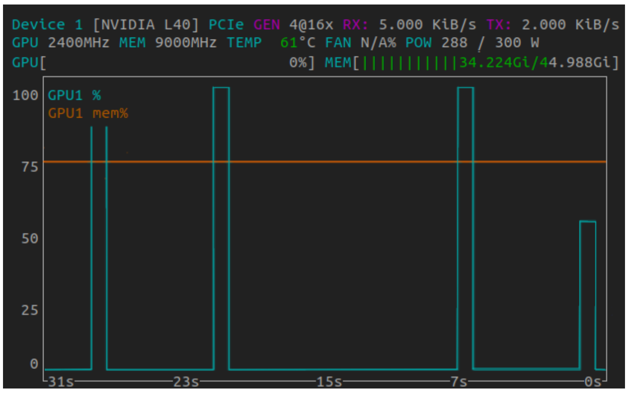


**Figure 2.** Architecture of our final model, which is based on U-Net utilizing Siamese VGG11 encoder.

pre-processing was performed in advance and multiple CPU workers were used for loading, achieving maximum utilization of graphical memory and its continuous load.

Numerous variants were trained with different parameters such as learning rate, batch size, weight decay, input image sizes, and so on. The best model was trained for 48 hours with a learning rate of 1E-5 and 1E-6, a batch size of 64, and images sized 256x256 pixels, which took around 175 epochs. Models with images sized 128x128, a batch size 256, BCE loss, and various numbers of epochs were also experimented with, but the evaluation results came closest to our best trained weights.

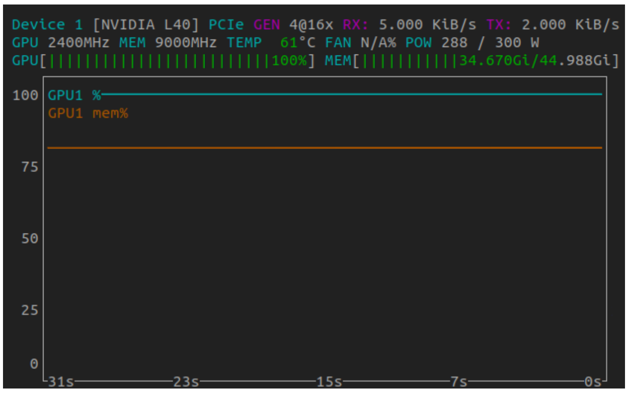


```
Device 1 [NVIDIA L40] PCIe GEN 4@16x RX: 5.000 KiB/s TX: 2.000 KiB/s
GPU 2400MHz MEM 9000MHz TEMP  61°C FAN N/A% POW 288 / 300 W
GPU[                               0%] MEM[|||||||||||34.224Gi/44.988Gi]
```

**Figure 3.** Infrequent GPU usage peaks before the improvement.

## 6. Evaluation & Results

We evaluated our models on the DAVIS2016 dataset [4]. The metrics we used for evaluating models include the Jaccard Index ($\mathscr{J}$) and the F-measure ($\mathscr{F}$) [4]. These metrics are essential for assessing the accuracy and robustness of segmentation tracking performance.



```
Device 1 [NVIDIA L40] PCIe GEN 4@16x RX: 5.000 KiB/s TX: 2.000 KiB/s
GPU 2400MHz MEM 9000MHz TEMP  61°C FAN N/A% POW 288 / 300 W
GPU[|||||||||||||||||||||||||100%] MEM[|||||||||||34.670Gi/44.988Gi]
```

**Figure 4.** High GPU usage after reducing the bottlenecking image loading to the memory.

**Jaccard Index** $\mathscr{J}$ expresses the similarity of segmented regions, i.e., the number of incorrectly labeled pixels. A Jaccard Index close to the value of 1 indicates a high similarity between the mask and the *groundtruth*. Let G be the *groundtruth* and M be the predicted mask, then $\mathscr{J}$ is defined as:

$$\mathscr{J} = \frac{|M \cap G|}{|M \cup G|} \tag{5}$$

**F-measure** $\mathscr{F}$ measures the accuracy of contours. Since the mask M can be interpreted as a set of closed contours $c(M)$, we can calculate the precision and recall of contours $P_c$ and $R_c$ between the contour points $c(M)$ and $c(G)$, then F-measure $\mathscr{F}$ is defined as:

$$\mathscr{F} = \frac{2P_cR_c}{P_c + R_c} \tag{6}$$

The overall quality score of the segmentation is given by their average:

$$\mathscr{J}\&\mathscr{F} = \frac{\mathscr{J} + \mathscr{F}}{2} \tag{7}$$

**Table 1.** Evaluation results

| | $\mathscr{J}_{mean}$ | $\mathscr{F}_{mean}$ | $\mathscr{J}\&\mathscr{F}$ |
|---|---|---|---|
| baseline | 37.16% | 41.91% | 39.53% |
| final model | 56.5% | 62.42% | 59.46% |

The evaluation shows that our final model significantly outperformed the baseline across all metrics, demonstrating improvements of 19.34 percentage points for the Jaccard Index, 20.51 percentage points for the Boundary F-score, and 19.93 percentage points in the combined $\mathscr{J}\&\mathscr{F}$ mean score. These results highlight the effectiveness of the enhancements incorporated into the final model.

## 7. Conclusions

In this project, we introduced a method for tracking a single object in a video through a straightforward approach. Our technique involves predicting subsequent masks by convolving both current and previous images and masks, utilizing custom-developed SiamMask-UNet architecture. This model was trained and evaluated on the MOSE dataset, which has been specifically adjusted for single object tracking. For the training process, we employed the Dice loss function and experimented with various training parameters to optimize performance. The model was evaluated on the DAVIS dataset, using metrics such as the Jaccard index and F-measure. The evaluation shows that our final model significantly outperformed the baseline model across all metrics.

This methodology allowed us to develop a model for single-object tracking in video sequences. However, there are opportunities for further improvement. Future enhancements could include refining the model architecture by integrating transformers, enhancing the model's ability to remember masks over longer sequences, and exploring multi-object segmentation. These modifications could extend the applicability of our methodology to more complex scenarios.

## 8. Task distribution

**Eva Mičánková (xmican10)** - Model designs, architectural experimentation, evaluation of trained models, documentation.

**Lenka Šoková (xsokov01)** - Dataset selection and preprocessing, evaluation scripts, documentation.

**David Kedra (xkedra00)** - Model training at the MetaCentrum, optimized GPU training processes, documentation.

## References

[1] Hu, W., Wang, Q., Zhang, L., Bertinetto, L. and Torr, P. H. SiamMask: A Framework for Fast Online Object Tracking and Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2023, vol. 45, no. 3, p. 3072–3089.

[2] Khoreva, A., Benenson, R., Ilg, E., Brox, T. and Schiele, B. Lucid Data Dreaming for Video Object Segmentation. 2019.

[3] Yin, Y., Xu, D., Wang, X. and Zhang, L. AGUnet: Annotation-guided U-net for fast one-shot video object segmentation. *Pattern Recognition*. 2021, vol. 110, p. 107580. Available at: https://www.sciencedirect.com/science/article/pii/S0031320320303836. ISSN 0031-3203.

[4] Perazzi, F., Pont Tuset, J., McWilliams, B., Van Gool, L., Gross, M. et al. A Benchmark Dataset and Evaluation Methodology for Video Object Segmentation. In: *Computer Vision and Pattern Recognition*. 2016.

[5] Ding, H., Liu, C., He, S., Jiang, X., Torr, P. H. S. et al. *MOSE: A New Dataset for Video Object Segmentation in Complex Scenes*. 2023.

[6] Ronneberger, O., Fischer, P. and Brox, T. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. 2015.

[7] Sarkar, A. *Implementation of All Loss Functions Deep Learning in Numpy, Tensorflow, and Pytorch* [https://arjun-sarkar786.medium.com/implementation-of-all-loss-functions-deep-learning-in-numpy-tensorflow-and-pytorch-e20e72626ebd]. 2021. Accessed: 2024-05-05.