



EÖTVÖS LORÁND UNIVERSITY

FACULTY OF INFORMATICS

DEPT. OF ARTIFICIAL INTELLIGENCE

Exploring and Addressing Potential Constraints in Image Oversegmentation Models

Supervisor:

Viktor Varga

Research Assistant

Author:

Milán Szász

Computer Science MSc

Budapest, 2024

Abstract

Image oversegmentation is a popular preprocessing choice in computer vision, as it creates a lower-dimensional, more compact image representation. Recent models are fully differentiable, which enables them to create learned superpixel feature vectors for specific downstream tasks. When working with image sequences, creating a series of image oversegmentation is a common starting point for solving tasks related to videos. However, existing methods do not utilize the inherited temporal information of an image sequence. In our proposed method, we present a model that utilizes optical flow information extracted from consecutive images. This enables the generation of superpixels with more accurate boundaries for objects with motion. This is advantageous for solving image oversegmentation and increasing the precision of downstream tasks related to videos. We have chosen a popular superpixel segmentation model, which was trained on purely RGB information. Then, we extended the model with an additional optical flow feature extraction module and trained it. Furthermore, we present our experiments by incorporating estimates for image depth as additional input features. Finally, we review the results of the evaluation and comparison of our model on the DAVIS video object segmentation dataset, demonstrating the increased segmentation accuracy of our proposed model.

Contents

1	Introduction	2
2	Related Work	5
2.1	Supapixel estimation	5
2.2	Optical flow estimation	6
2.3	Monocular depth estimation (MDE)	7
2.4	Supapixel-based Video Object Segmentation (VOS)	8
3	Proposed Method	9
3.1	Motivation to the Proposed Method	9
3.2	Problem Description	10
3.3	Differentiable SLIC	10
3.4	Modular Framework	12
3.5	Training Objective	14
3.6	Implementation	16
4	Experiment	17
4.1	Used metrics and Datasets	17
4.2	Results	19
5	Discussion	22
6	Conclusion	25
	Acknowledgements	26
	Bibliography	26
	List of Figures	31
	List of Tables	32

Chapter 1

Introduction

Image oversegmentation is a computer vision technique that groups pixels based on common low-level features, such as texture. These pixels will subsequently create a superpixel. This process will result in a segmentation mask and superpixel feature vectors. The mask defines which pixels belong to the same group, while the feature vector represents the low-level features of a group. Oversegmentation markedly reduces the number of primitives and redundancy in the image, while maintaining the significance of pixel regions. The use of superpixels as an input is a popular choice for models when explainability is an important factor. In such cases, the superpixel representation of image data is often processed by a Graph Neural Network (GNN). The reduction in the number of input elements resulting from oversegmentation renders the GNN architecture much simpler than that of many Deep Neural Networks (DNN). Furthermore, the GNN models the relation between superpixels, which is not the case with DNNs. In the context of safety-critical applications, the importance of explainability cannot be overstated. A significant number of proposed methods utilize image oversegmentation as a preliminary step in the field of object detection and tracking, image segmentation, image parsing, and so forth.

One of the most popular methods for creating image oversegmentation is the Superpixel Sampling Networks (SSN) [1] approach proposed by Jampani et al. Their model is based on deep neural networks and they were the first to propose an end-to-end solution for superpixel segmentation. In their solution, they first use a feature extraction model to create pixel features based on the RGB data of the image. Finally, the pixel embeddings are utilized to generate the final oversegmentation image using a proposed differentiable SLIC algorithm. The end-to-end approach

allows for the learning of unique feature extractions and oversegmentations for a range of downstream tasks.

Some problems in computer vision, such as Video Object Segmentation (VOS), necessitate the manipulation of image sequences. This type of data contains information about the motion of objects. By understanding how pixels belonging to different moving objects translate between neighboring timesteps, it becomes easier to differentiate object borders. However, current superpixel segmentation models are only capable of working with single image data. They do not make any assumptions about other frames, which means that they cannot utilize this feature of image sequences. Consequently, they are only relying on RGB data, such as that used by SSN. The use of optical flow estimation allows for the calculation of the motion of objects in an image sequence. These algorithms are designed to determine the displacement of pixels from one frame to another by analyzing the intensity patterns in the image sequence. The underlying assumption is that neighboring pixels in consecutive frames move similarly. Optical flow estimation can be carried out using either classical differential methods or modern deep learning models.

The incorporation of image depth into traditional RGB features offers a more comprehensive understanding of the visual information present in an image. This additional information can facilitate more accurate differentiation of object boundaries, leading to more precise image segmentation. There are several methods for obtaining image depth, including the use of specific sensors and neural networks. In the event that only one RGB image is available, the optimal approach is to utilize monocular depth estimation to obtain image depth.

In our proposed method, we have extended the end-to-end trainable SSN model to utilize motion vectors and image depth to produce a more accurate superpixel representation. To obtain the optical flow estimation, we have chosen the Global Motion Aggregation (GMA) model [2] proposed by Shihao et al. This method handles occlusion very well, which is a difficulty in optical flow estimation. Image depth is predicted with the depth-anything model [3]. The initial starting point was a pre-trained SSN model, which was extended with additional feature extraction modules. These created pixel embeddings from the motion vectors and image depth. By separating the feature extractor of the additional image features into different components and concatenating their results, rather than extending the original RGB extractor, training time was saved. This was because the visual knowledge was al-

ready learned; therefore, it was unnecessary to modify or finetune it. The feature extractors are based on Convolutional Neural Network (CNN) architecture, and they create feature vectors in three different resolutions. This way the extractors will achieve the kind of pixel embeddings we would like to have because describing a pixel from its neighborhood will carry much more information about it than if we would only embed each pixel one by one. The lower the resolution of the feature space, the greater a pixel’s vision will be in its local environment. The final embeddings are created by upscaling the lower-resolution representations to the original image size, concatenating them, and applying a convolutional layer to generate the embeddings. The final concatenation is comprised of two residual connections in the neural network, which enhance the gradient flow. These additional embeddings are then concatenated with the original RGB feature vectors, from which the differentiable SLIC algorithm (proposed in [1]) generates the image oversegmentation. During the training phase, the SSN feature extraction layers were frozen, while the optical flow and image depth feature extraction models were trained separately. The DAVIS training set was employed for training purposes.

Our findings indicate that the proposed method exhibits enhanced accuracy for single-image oversegmentation when accompanied by image sequences. This is attributed to the enhanced precision in identifying the boundaries between different moving objects or a moving object and the background. Results show that utilizing image depth can also further increase accuracy.

The main contributions of this paper are summarized as follows:

- Our findings indicate that image sequences contain inherited extra information that can be exploited by image oversegmentation models, which results in increased accuracy.
- Including image depth as an additional input feature can also increase the accuracy in certain cases.
- We have extended the SSN architecture with a motion and image depth feature extraction module and trained it on the DAVIS dataset. These extraction modules can be combined with the original RGB extractor in a modular way.

Chapter 2

Related Work

This chapter serves to review the literature related to the concepts presented in this paper. The first three sections will concentrate on the techniques employed, while the final section will help us understand the importance of superpixel segmentation by reviewing the topic of video object segmentation.

2.1 Superpixel estimation

The majority of classical solutions in image oversegmentation are based on graph partitioning or clustering methods. Graph partitioning is a method used to divide a graph into smaller subgraphs while minimizing the number of edges cut between partitions. In graph-based approaches, each superpixel is formulated as a subgraph, with graph nodes represented by pixels and edges denoting the similarity between adjacent pixels. The Normalized Cuts algorithm [4] treats the image as a graph in the aforementioned manner. It solves the perceptual grouping of image data by extracting the global impression of an image. During grouping, it maximizes the dissimilarity between groups and minimizes the similarity within a given group. Based on the previously proposed method, Xiaofeng et al. [5] proposed an image oversegmentation technique. Other graph-based methods include the entropy rate superpixels (ERS) [6] and Felzenszwalb and Huttenlocher (FH) [7]. In ERS [6], a new objective function based on the entropy rate of a random walk on a graph and a cluster size balancing term was introduced to solve superpixel segmentation. Clustering-based solutions rely on the traditional k-means algorithm to solve image oversegmentation. SLIC [8] represents each pixel as a five-dimensional feature vector,

comprising the coordinate and Lab color space. Furthermore, they limit the number of k-means comparisons based on local image constraints. Other k-means-based methods, such as LSC and Manifold-SLIC [9], differ only in the representation of the pixel feature vector. LSC [10] projects the SLIC features onto a 10-dimensional space, while Manifold-SLIC utilizes a two-dimensional manifold feature space. Zhihua et al. [11] proposed a method that employs a Gaussian mixture model (GMM) to segment images into superpixels. They represent a superpixel as a Gaussian function, which describes the density of each pixel. In contrast to other GMMs, they utilize constants as mixing weights, which resulted in similar segment sizes and a linear model complexity.

Tu et al. [12] introduced the first deep-learning approach for superpixel segmentation, which was based on handcrafted features. They used an affinity extractor model to predict similarities of neighboring pixels, from which they constructed a graph and applied a traditional graph-based superpixel segmentation algorithm. The Superpixel Sampling Networks (SSN) [1] created by Jampani et al. has improved deep learning-based image oversegmentation. In contrast to traditional methods that rely on non-differentiable algorithms, SSN is fully end-to-end trainable, leveraging deep neural networks to learn superpixel segmentation directly from the data. They have modified the SLIC algorithm to make it probabilistic, allowing for gradient-based optimization. This differentiable SLIC will create the final segmentation based on the learned pixel features. Yang et al. proposed an approach integrating a fully convolutional network (FCN) [13] that predicts superpixels directly on a regular image grid. This method addresses the inefficiency of standard convolution operations when applied to superpixels. Peng et al. [14] combined Deep Affinity Learning (DAL) and Hierarchical Entropy Rate Segmentation (HERS) in their proposed method. HERS utilizes the learned affinities to generate adaptable superpixels through a hierarchical tree structure dynamically. This will result in an adaptive superpixel segmentation, with homogeneous surfaces requiring fewer superpixels.

2.2 Optical flow estimation

The traditional approach to optical flow estimation involved solving an energy minimization problem. Ju et al. [15] proposed a multilayer locally affine motion model and regularisation with transparency based on the aforementioned robust

estimation framework, to improve the estimation of multiple motions in optical flow. The resulting algorithm employed the Expectation Maximisation (EM) method to minimize the energy function. Other methods [16], [17] have experimented with gradient-based solutions to estimate the parameters of the energy function, with the objective of predicting motion vectors.

RAFT [18] is a deep neural network-based method that employs convolutional networks to extract per-pixel features and build multi-scale 4D correlation volumes for all pairs of pixels. These pixel embeddings are then iteratively processed by a recurrent neural network to update the flow field. The issue of occlusions represents a significant challenge for optical flow algorithms that rely on local evidence. Jiang et al. proposed the Global Motion Aggregation module (GMA) [2], which is a transformer-based approach to identifying long-range dependencies between pixels in the first image and performing global aggregation on the corresponding motion features. FlowFormer [19] is an encoder-decoder transformer-based approach for learning optical flow. It employs a tokenization process to transform the 4D feature-pair volume, derived from an image pair, into a cost memory. This is achieved through the use of alternative group transformer layers and a recurrent transformer decoder, which incorporates dynamic positional cost queries. The final displacement vectors are then generated through this process.

2.3 Monocular depth estimation (MDE)

The initial models of MDE were heavily reliant on handcrafted features and traditional computer vision techniques. Saxena et al. [20] employed image oversegmentation as a preliminary processing step, wherein a Markov Random Field was utilized to capture both the three-dimensional location and orientation of each patch.

Eigen et al. [21] proposed a model comprising two deep network stacks. The first is responsible for coarse global prediction based on the entire image, while the second refines this prediction locally. A scale-invariant error was used during training to overcome the scale ambiguity of MDE. This measures the relationships between points in the scene, irrespective of the absolute global scale. Due to the unique characteristics of different datasets, a trained model on one of them will lack generalization ability. MiDaS addressed this issue by employing multi-objective learning. This approach enables the utilization of multiple mixed datasets, despite their in-

compatibility in terms of annotations. MiDaS [22] employed zero-shot cross-dataset transfer to assess its performance in real-world datasets and demonstrated robust generalization capabilities. Building upon the success of MiDaS, depth-anything [3] sought to further enhance its generalization abilities. The training set was expanded by up to 62M images through the design of a data engine for the collection and automatic annotation of large-scale unlabeled data.

2.4 Superpixel-based Video Object Segmentation (VOS)

The segmentation of a video is a high-dimensional and complex task. Consequently, some solutions employ image oversegmentation to reduce the number of primitives. Tan et al. [23] used the attention mechanism to compare the superpixel representation of the reference and target frame. Their model was trained in a self-supervised manner to reconstruct each consecutive frame. Lee et al. [24] proposed a prototype memory network architecture to overcome the inaccuracy resulting from comparing only adjacent frames. The model extracts RGB and motion information, and from these, superpixel-based component prototypes are formed. The most important prototypes are selected and stored in the memory network, which is accessed during the decoding of a segmentation mask for each frame. Varga et al. [25] created a graph structure from the video using superpixels as nodes, and edges were determined from spatial adjacency and temporal motion. A label propagation graph neural network was employed to disseminate segmentation labels across each segment.

Chapter 3

Proposed Method

3.1 Motivation to the Proposed Method

The primary objective of our methodology is to enhance the accuracy of image oversegmentation when a sequence of images is provided. Existing models do not assume the existence of past or future frames during superpixel sampling. By not incorporating motion information that can be extracted from neighboring frames, these solutions are constrained to base their prediction on RGB information alone. When objects with similar textures appear nearby, it becomes considerably more challenging to separate them. Incorporating the motion of these objects and detecting patterns within them makes this task easier. As evidenced by related works, superpixel segmentation is a prevalent approach in video object segmentation (VOS). Consequently, enhancing the accuracy of superpixel segmentation will directly improve the precision of downstream computer vision tasks.

Explainability is a crucial aspect of deep learning. Humans often perceive models as black boxes, which makes decision-making for humans much more challenging. This is a more serious problem in safety-critical applications such as healthcare, finance, and autonomous driving. A superpixel segmented image is often combined with a Graph Neural Network, which is a more straightforward approach to explainability than CNN or Transformer architectures. By improving the performance of image oversegmentation, we also aim to enhance explainability in this field.

Therefore, we initiated an exploration aimed at addressing the performance loss by not utilizing the aforementioned additional information. The SSN image oversegmentation model has revealed the potential of differentiable SLIC, a method that

enables the neural network to be trained in an end-to-end manner, thereby allowing the learning of unique representations for different downstream tasks. The simplicity of the SSN architecture has led to the creation of a modular network that can utilize RGB, optical flow, and even depth features of images.

The motivation for this study was the article in question and the importance of accurate superpixel segmentation for solving video-based tasks. To improve segmentation accuracy, it was necessary to take advantage of additional image features.

3.2 Problem Description

Superpixel segmentation is defined as a partitioning of an image with extra constraints for the segments. Let P be the set of all pixels in a given image and K the required number of segments. $S = \{S_i\}_{i=1}^K$ is a partitioning of the image if:

- The union of the segments are equal to the set of pixels $\bigcup_{S_i \in S} S_i = P$
- The intersection of any two segments is the empty set $(\forall S_i, S_j \in S) i \neq j \implies S_i \cap S_j = \emptyset$

It is pertinent to note that in mathematics, no partition can be an empty set. However, during superpixel segmentation, it is possible that some segments may contain zero pixels.

Stutz et al. [26] have proposed a revised set of criteria for defining superpixels:

- Connectivity. Superpixels are expected to represent connected sets of pixels.
- Boundary Adherence. Superpixels should preserve image boundaries.
- Compactness, Regularity, and Smoothness. In the absence of image boundaries, superpixels should be compact, placed regularly, and exhibit smooth boundaries.
- Efficiency. Superpixels should be generated efficiently.
- Controllable. The number of generated superpixels should be controllable.

3.3 Differentiable SLIC

The Differentiable SLIC is a modification of the famous SLIC algorithm, which enables gradient flow through the clustering procedure.

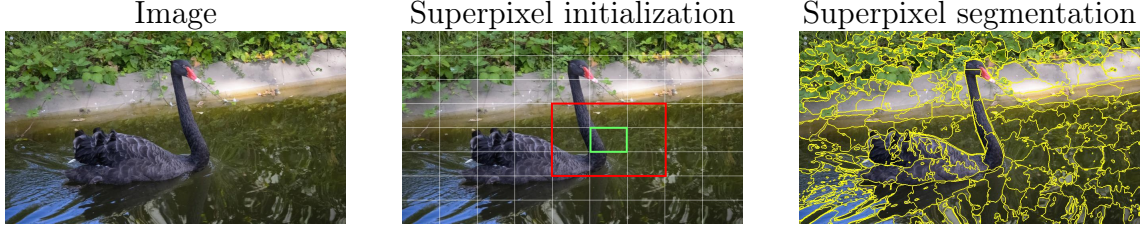


Figure 3.1: **Superpixel initialization and final result.** This example presents the different stages of the SLIC algorithm. In the middle, we can see the initial superpixel grids, the red boundary shows the locality constraint applied during the algorithm. The right picture shows the boundary of the final superpixel segmentation.

To understand it, it is first necessary to review SLIC. The method is a modification of the K-means clustering algorithm with some image-specific constraints. It has a fast runtime and produces compact and uniform superpixels. The input of the algorithm is the pixels of the whole image, which are represented by their color and position. The color is represented in the CIELAB (Lab) colorspace, which comprises of three numbers L defines the lightness, a is the position between the red and green components, and b is the position between the yellow and blue component. The position is represented by the XY coordinate of the pixel. Formally, given an image $I \in \mathbb{R}^{n \times 5}$ with 5-dimensional (XY Lab features) pixel vectors the task of superpixel computation is to assign each of them to one of the m superpixels (centroids). This will result in an association map $H \in \{0, 1, \dots, m-1\}^{n \times 1}$. The SLIC algorithm has four steps:

0. Initialization: Pixels are sampled and aggregated to create the initial five-dimensional feature vectors for each superpixel. This sampling is typically conducted uniformly across the pixel grid. $S^0 \in \mathbb{R}^{m \times 5}$
1. Pixel-Superpixel association: each pixel p is assigned to the nearest superpixel in the five-dimensional space

$$H_p^t = \arg \min_{i \in \{0, \dots, m-1\}} D(I_p, S_i^{t-1}) \quad (3.1)$$

where D denotes the euclidean distance on the power of two $D(a, b) = \|a - b\|^2$

2. Superpixel center update: new superpixel cluster centers S^t are computed by aggregating associated pixel features (XY Lab). For each superpixel i , the

centroid of that cluster is computed,

$$S_i^t = \frac{1}{Z_i^t} \sum_{p|H_p^t=i} I_p \quad (3.2)$$

, where Z_i^t denotes the number of pixels in the superpixel cluster i .

The algorithm is executed iteratively from step 1 until the defined number of iterations is reached. The computational cost of determining the association between all the pixels and superpixels in Eq. 3.1 is usually constrained to a fixed neighborhood around each superpixel center.

The primary concern with SLIC is that it is non-differentiable. This is due to the use of a non-differentiable nearest neighbor search in Eq. 3.1, which results in the formation of rigid associations between pixels and superpixels. The differentiable SLIC approach addresses this issue by transforming the hard associations between $H \in \{0, 1, \dots, m-1\}^{n \times 1}$ into soft associations $Q \in \mathbb{R}^{n \times m}$. This transformation is achieved by modifying Eq. 3.1 as follows:

$$Q_{pi}^t = e^{-D(I_p, S_i^{t-1})} = e^{-\|I_p - S_i^{t-1}\|^2} \quad (3.3)$$

The computation of new superpixel cluster centers, as described in Eq. 3.2, is also modified to utilize soft associations:

$$S_i^t = \frac{1}{Z_i^t} \sum_{p=1}^n Q_{pi}^t I_p, \quad (3.4)$$

where $Z_i^t = \sum_p Q_{pi}^t$ is the normalization constant. The application of these modifications enables the gradient to flow through each step, thereby facilitating the learning of task-specific feature extraction by the model.

3.4 Modular Framework

The base model comprises an RGB (with additional 2d coordinates) extraction module, which generates a feature vector for each pixel. The differentiable SLIC algorithm employs these learned pixel embeddings to create a final estimation. The algorithm utilises Euclidean distance to compare each pixel, thereby negating the assumption of fixed-length vectors. This property renders it possible to create a

modular framework. Optical flow and image depth-based features may be concatenated with the original features in any combination prior to the application of the differentiable SLIC.

In order to utilize optical flow and image depth as an input feature, it is necessary to pre-process the images using an appropriate model. An optical flow estimation can be performed on images captured from sequences. In this instance, we only consider the past and future neighboring frames, and for estimation, we utilize the methodology outlined in [2]. The forward and backward motion is computed using the target image and its two neighbors. To obtain depth images, we employ the monocular approach, which entails the prediction being based solely on a single image. This enables the depth extraction module to be utilized in a wide range of scenarios. To obtain image depth, we utilize the method described in [3].

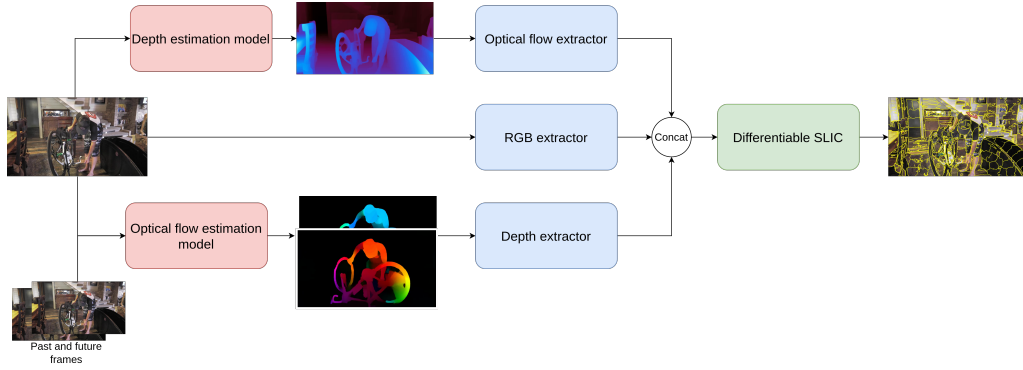


Figure 3.2: **Pipeline of the proposed method:** First the image is processed by an optical flow/image depth estimation model. Then the input features are extracted by their respective neural network, these can be used in any combination. The output embedding vectors will be concatenated before the differentiable SLIC will create the superpixel segmentation

A convolutional neural network is employed for feature extraction, with each convolutional layer followed by a batch normalization and ReLU activation function. Max-pooling is utilized to halve the resolution of the input, after the second and fourth convolutional layers, thereby expanding the receptive field of the neural network. The downsampling process results in three feature maps with decreasing resolution. Higher-resolution feature maps preserve low-level visual information, while lower-resolution ones recognize high-level patterns. To exploit these properties, lower-scaled feature maps are bilinearly upsampled to the original resolution. The combination of outputs from different layers in the sequence also serves as residual connections, which further enhances stable gradient flow. After concatenating the

three different levels of features, a final convolutional layer will create the output. The convolutional layers employ a 3×3 kernel size and a bias parameter that is turned off. The number of output channels is set to 64 in each layer, with the exception of the final convolutional layer, which outputs are parameterized. Each module utilizes the same feature extractor architecture with minor variations. The RGB module has an input size of 5, the optical flow module has 4, and the depth module has 1. Following the acquisition of the feature vectors, they are concatenated with each other and the original XY LAB values.

The differentiable SLIC algorithm outputs a segmentation map and a list of superpixel feature vectors. The segmentation map contains, for each pixel, the id of the assigned superpixel. The superpixel feature vectors encode the visual features of the entire segment.

3.5 Training Objective

The differentiable pixel clustering algorithm enables end-to-end training of the model, allowing it to be used for task-specific superpixel segmentation. The CNN-based feature extraction modules facilitate the use of any computer vision-related task-specific loss function for learning optimized superpixel representations. In our case, semantic labels are employed to train the model to separate semantic boundaries, even if the semantic labels are not particularly detailed. This training goal helps to separate low-level textures without semantic meaning. The task-specific reconstruction loss is combined with a compactness loss to achieve the best performance.

Mapping between pixel and superpixel representations. The generated segmentation map is not sufficient for computing semantic segmentation loss. To bridge this gap, it is necessary to review the mapping between pixel and superpixel representations. This transformation can be performed using either the hard or soft pixel-superpixel associations. During training, the soft association is employed to allow gradient flow, which is achieved by using the soft association. The mapping from a pixel to a superpixel representation is a simple matrix multiplication operation. The transpose of the column-normalized Q matrix (denoted as $\hat{Q}$) is used as the mapping function: $S = \hat{Q}F$, where F and S denote pixel and superpixel representations, respectively. The inverse mapping from superpixel to pixel representation is

accomplished by multiplying the row-normalized Q , denoted as $\tilde{Q}$, with the superpixel representations, $F = \tilde{Q}S$. Consequently, the pixel-superpixel feature mappings are represented by simple matrix multiplications with the association matrix and are differentiable.

Semantic reconstruction loss. The segmentation mask that we wish to represent efficiently with superpixels is denoted by $R \in \mathbb{R}^{n \times l}$. This is only used during the training phase. As previously discussed, pixel properties can be mapped onto superpixels using the column-normalized association matrix $\hat{Q}$, $\check{R} = \hat{Q}^T R$, where $\check{R} \in \mathbb{R}^{m \times l}$. Subsequently, the row-normalized association matrix $\tilde{Q}$ is employed to map $\check{R}$ to the pixel representation R^* , $R^* = \tilde{Q}S$, where $R^* \in \mathbb{R}^{n \times l}$. The reconstruction loss can then be computed in the following manner:

$$L_{recon} = \mathcal{L}(R, R^*) = \mathcal{L}(R, \tilde{Q}\hat{Q}^T R), \quad (3.5)$$

where $\mathcal{L}(.,.)$ denotes the cross-entropy loss-function.

Compactness loss. The model is capable of generating spatially compact superpixels, which reduces the spatial variance in superpixels. This is achieved by compensating for the loss of compactness that occurs during the process of superpixel creation. Let I^{xy} denote positional pixel features. These positional features are then mapped into our superpixel representation, $S^{xy} = \hat{Q}^T I^{xy}$. Subsequently, the inverse mapping is performed on the pixel representation utilizing the hard associations, rather than the soft associations, by assigning the same superpixel positional feature to all pixels belonging to that superpixel. This is expressed as $I_i^{xy} p = S_i^{xy} | H_p = i$. The inverse mapping with hard associations is differentiable, in contrast to the mapping from pixel to superpixel. The compactness loss is defined as the following L_2 norm:

$$L_{compact} = \left\| I^{xy} - \bar{I}^{xy} \right\|_2, \quad (3.6)$$

The utilization of these loss functions enables the model to be trained in accordance with the specifications of an optimal image oversegmentation model. The comprehensive loss utilized in this study is a synthesis of these two loss functions.

$$L = L_{recon} + \lambda L_{compact}, \quad (3.7)$$

where we set λ to 10^{-5} in all our experiments.

3.6 Implementation

The code was implemented in its entirety in Python, with the deep learning framework Pytorch employed. The input features XY, Lab, optical flow, and image depth were scaled, with the position, color, motion vector, and depth feature scales represented as γ_{pos} , γ_{color} , $\gamma_{optflow}$, γ_{depth} respectively. The value of γ_{color} is independent of the number of superpixels and is set to 0.26 with color values ranging between 0 and 255. The value of γ_{pos} is dependent on the number of superpixels. It can be expressed as follows: $\gamma_{pos} = \eta \max(m_w/n_w, m_h/n_h)$, where m_w, n_w, m_h, n_h represent the number of superpixels and pixels along the image width and height, respectively. In practice, we have observed that $\eta = 2.5$ performs well. For the depth estimation task, we employed a value of $\gamma_{depth} = 2.4$. For the optical flow estimation task, we used $\gamma_{optflow} = 5$. These scales were determined with a hyperparameter search. The feature extraction modules were trained separately. The original RGB module was trained from scratch, while the other modules were trained in conjunction with the previously trained RGB module. For training, we utilized randomly cropped image patches of size 200×200 . Additionally, we applied data augmentation techniques, including random horizontal flips and random scaling. For stochastic optimization, we employed the Adam optimizer with a batch size of 6 and a learning rate of 0.0001. Table 3.1 describes the different training parameters of the different modules. The RGB module was trained for 500,000 iterations with 100 superpixels, as described in [1]. The optical flow and image depth modules were trained for 6,000 iterations with 600 superpixels. Each module was trained on a fixed number of superpixels, with varying numbers of superpixels being used for estimation. During training, five iterations of differentiable SLIC were used, with ten iterations being used during testing. This resulted in only marginal performance gains during evaluation.

Module	N. of iterations	N. of superpixels	Dataset	RGB included
RGB	500k	100	BSDS500	✓
optflow	6k	600	DAVIS	✓(frozen)
depth	6k	600	BSDS500/DAVIS	✓(frozen)

Table 3.1: The varying training parameters of the different modules. The last column represents whether the RGB module was included during training.

Chapter 4

Experiment

4.1 Used metrics and Datasets

BSDS500 - comprises 200 training images, 100 validation images, and 200 test images. Each image has been assigned multiple different ground-truth segmentation labels by several different annotators. The labels were generated based on low-level image features, such as texture, and are not semantic. In total, 1,633 image label pairs may be used for training and validation, while 1,063 may be used for testing. This dataset has been utilized for the training of the original SSN model employed in our method.

DAVIS2017 - is a competition designed for the task of video object segmentation. It is based on a public dataset and benchmark, and it is intended to facilitate the development of unsupervised, semi-supervised, and interactive video object segmentation algorithms. The dataset comprises a total of 150 videos, with 60 designated for training, 30 for validation, and 60 for testing. However, only the training and validation sets are publicly available. This dataset was employed to train the optical flow feature extraction module and to evaluate both the original and new methods. During the evaluation, the results were tested on every tenth frame.

Metrics - during the evaluation process, three distinct metrics were employed. The first is **Achievable Segmentation Accuracy (ASA)** [27], which quantifies the achievable accuracy for segmentation using superpixels as a pre-processing step, as detailed in the cited source. Let $S = \{S_j\}_{j=1}^K$ and $G = \{G_i\}$ be partitions of the same image $I : x_n \rightarrow I(x_n), 1 \leq n \leq N$, where S represents a superpixel segmentation and G a ground truth segmentation. Furthermore, let K be the number

of superpixels and N be the number of segmentation masks.

$$\text{ASA}(G, S) = \frac{1}{N} \sum_{S_j} \max_{G_i} \{|S_j \cap G_i|\} \quad (4.1)$$

The other metric is employed in traditional segmentation and is referred to as the **Intersection Over Union (IOU)** [28]. This metric describes the extent of overlap between two areas. The greater the region of overlap, the greater the IOU. The segmentation S_{major} is created from the superpixels in the following manner: for each superpixel, the number of ground truth labels falling within the given superpixel is calculated, and then the superpixel is relabeled with the id of the most frequent ground truth label. The union of the relabeled segments is taken to obtain S_{major} . The ground truth segmentation mask of the image is denoted by G_{mask} .

$$\text{IOU} = \frac{S_{major} \cap G_{mask}}{S_{major} \cup G_{mask}} \quad (4.2)$$

Finally, the concept of **Compactness**, as defined in [29], is employed not only as a training objective but also to quantify the spatial variance of superpixels during evaluation. The incorporation of optical flow and image depth modules is intended to prevent the deformation of superpixels. This metric is utilized to assess the change in superpixel compactness. The value of Eq. 3.6 is used to compute the compactness of the superpixels. The lower the value, the more compact the superpixels are.

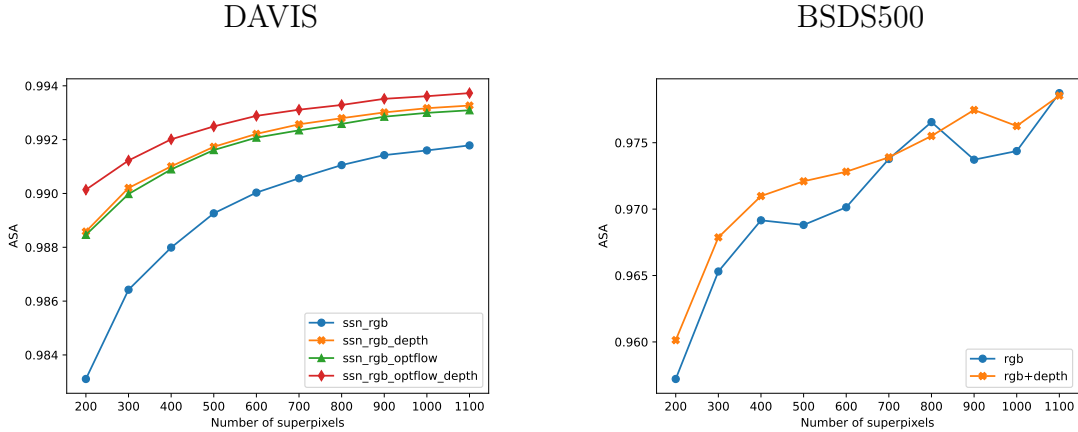


Figure 4.1: **ASA results.** We can see that SSN performs better with all the additional module configurations on the DAVIS dataset. Extending with image depth shows also an increased ASA score on the BSDS dataset.

4.2 Results

The primary objective of the proposed method is to enhance the accuracy of superpixel segmentation in images captured from image sequences by leveraging optical flow information. Furthermore, we have explored the potential of incorporating image depth as an additional feature to enhance accuracy. The combinations of the aforementioned image features were employed to assess the efficacy of the proposed modular network on the DAVIS dataset. Furthermore, the addition of image depth was tested on the BSDS500 dataset. In both cases, the results were evaluated using varying numbers of superpixels, ranging from 200 to 1100, and the aforementioned metrics.

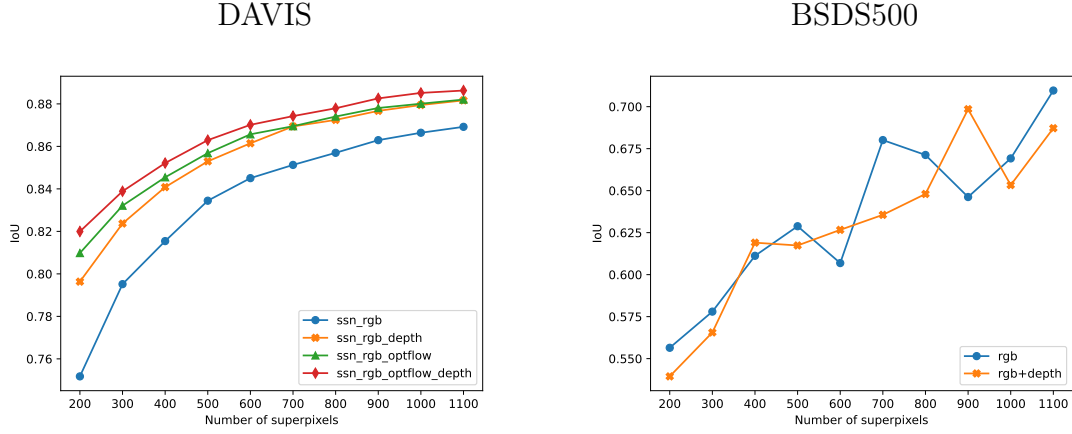


Figure 4.2: **IoU results.** The extended SSN versions produce a higher IoU score on the DAVIS dataset. On the BSDS dataset the image depth extended model cannot outperform the original in IoU.

DAVIS results - the extension of the original SSN model with any combination of additional input features demonstrates a significant improvement in both the ASA and IoU scores (see Figures 4.1 and 4.2). Additionally, the figures illustrate that the combination of both additional modules yields the most favorable performance. The greatest improvement is observed when only 200 superpixels are utilized for segmentation. The addition of image depth results in a 0.05 IoU improvement, while the combination of optical flow and image depth yields an improvement of 0.07. The compactness value of the superpixels remains unaltered (see Figure 4.3), indicating that the incorporation of additional image features did not affect the shape of the superpixels. Both the optical flow and image depth extraction modules were trained on the DAVIS training set. Additionally, some qualitative results are presented in Figure 4.4. By examining the superpixel boundaries around the object with the seg-

mentation label, it can be observed that the extended models have superior contours around the objects.

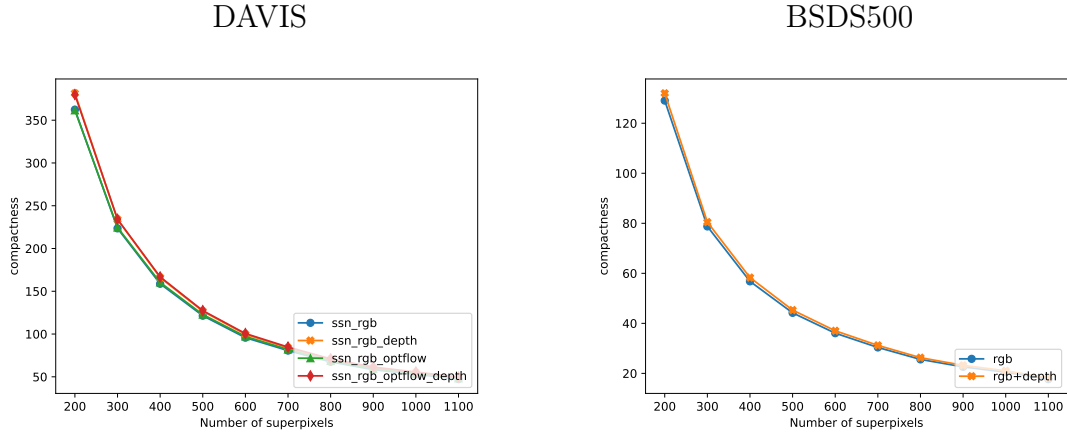


Figure 4.3: **Compactness results.** The extended versions of SSN retain the same compactness as the original model.

BSDS500 results - the addition of the image depth extractor was also tested on the BSDS500 dataset. In this case, the depth module was only trained on the train set belonging to this dataset. The results show improvement in ASA score, however, the IoU values show oscillation between the two configurations (see Figure 4.1 and Figure 4.2). The compactness values are almost unchanged (see Figure 4.3).

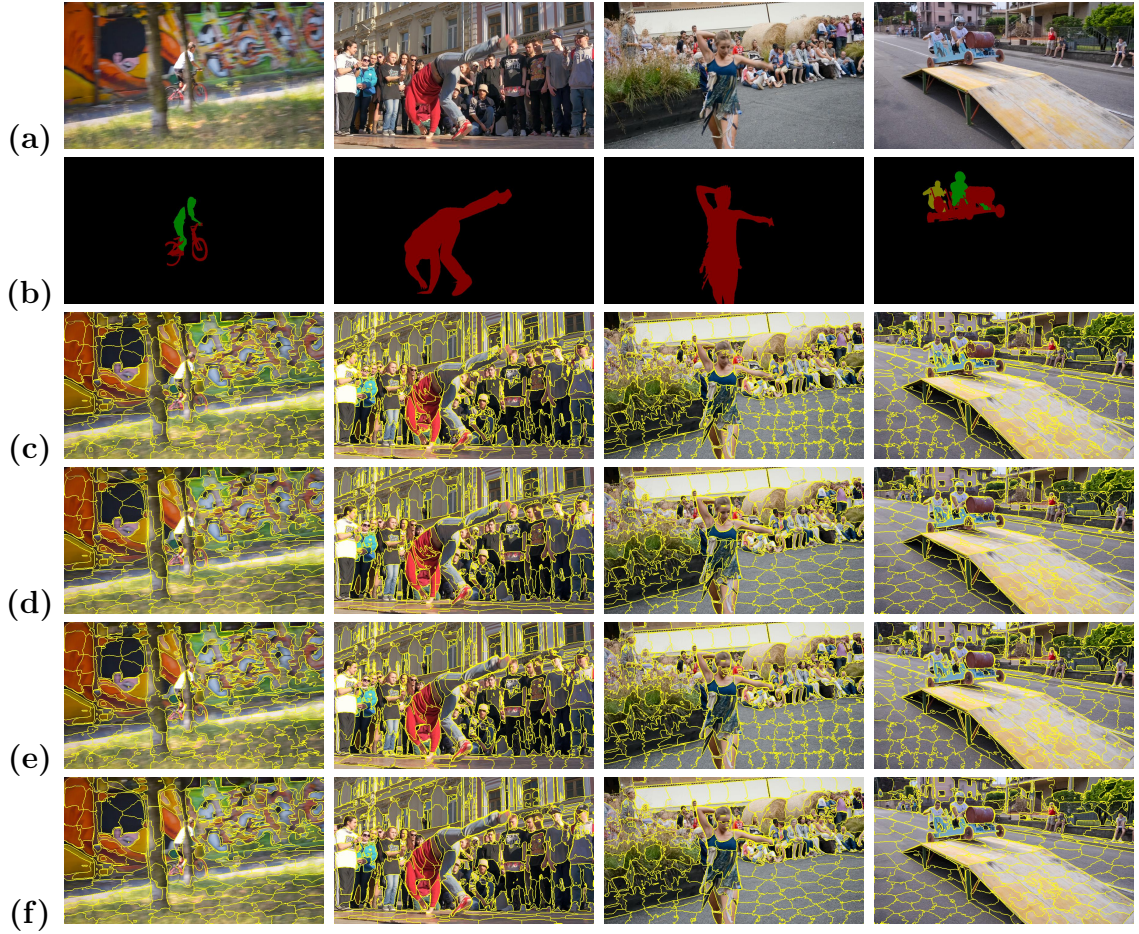


Figure 4.4: **Qualitative comparison on DAVIS.** Row (a) shows the input images, (b) the ground truth label, (c) RGB module's predictions, (d) RGB+depth predictions, (e) RGB+optflow predictions, (f) RGB+depth+optflow predictions.

Chapter 5

Discussion

In this chapter, we will present our concerns with SSN and discuss the experiments we have conducted to address these concerns. While these experiments did not become part of the proposed method, we believe further investigation and experimentation could be beneficial.

Small training set - The SSN utilized the BSDS500 dataset for training, which contains many segmentation masks in various forms for each image. However, the dataset comprises merely 200 training images, the majority of which were captured in a natural setting, thereby limiting the diversity of the scene. This may potentially impede the model’s performance. Therefore, it is recommended to expand the training set with more diverse images to enhance the model’s accuracy. To address this limitation, we employed transfer learning. We utilized MobileNetV3 [30] as the backbone of the RGB extraction module. This backbone was trained on the ImageNet dataset for the classification task. To preserve low-level visual information, we only used some shallow layers from it. This experiment demonstrated a loss of accuracy compared to the original method. It is postulated that this phenomenon is due to the fact that classification models are designed to focus on class-related patterns, even in the shallow layers, which results in the loss of information regarding the background. Further experimentation was conducted with backbones trained on segmentation tasks, but similar issues were encountered. It is therefore proposed that it would be beneficial to investigate the use of a backbone that was trained on unsupervised tasks, such as image denoising and super-resolution. These tasks encourage the model to preserve low-level image features, in contrast to classification and segmentation.

Feature aggregation - The differentiable SLIC utilized by SSN is both efficient and accurate, a consequence of the K-means-like approach. During the superpixel update step of the algorithm, pixel features are averaged, a process that may result in a certain level of performance loss. This is because feature aggregation can cause the loss of some important details, which can result in pixel-superpixel associations. To address this challenge, we devised a solution that involved the integration of a separate label propagation network, which operates in parallel with the differentiable SLIC. This approach entails the generation of an image oversegmentation by both networks, which is subsequently combined with a neural network to yield the final results. The label propagation process can be described as follows:

1. A horizontal and vertical affinity map is generated from the image, with an affinity value describing the degree of similarity between two neighboring pixels.
2. Seed points are selected on the pixel grid and the label map is initialized. In our case, each seed point represents a distinct superpixel and is assigned a unique label. The label map stores the degree of similarity between each pixel and the seed points, with a channel for each label and values ranging from 0 to 1. Seed points are initialized with a value of 1 on their respective channel, while all other pixels are set to 0.
3. Labels are iteratively propagated to adjacent pixels, with the number of iterations determining the area covered by the labels. The affinity values between neighboring pixels define the propagated label value, as a pixel has several neighbors. Therefore, a label aggregation strategy is necessary to determine the final value.
4. The pixel-superpixel soft associations are derived from the label map.

In our approach, we utilize a neural network to determine the propagated value for each pixel. We track the affinity path of the propagated label for each pixel, and when propagating, we extend this affinity path with the actual affinity neighbor value. We then apply a one-dimensional convolution to predict the propagated value. Our aggregation strategy is to choose the maximal neighbor value.

Unfortunately, the training of this method was unstable, which prevented it from improving the results. We hypothesize that further investigation could resolve this issue. Testing with synthetic data could reveal some of the underlying causes.

Chapter 6

Conclusion

A novel superpixel segmentation pipeline has been implemented with the objective of enhancing the precision of image oversegmentation on images derived from video sequences. To achieve this, we have employed optical flow estimation as an additional input feature. Furthermore, we have explored the potential of utilizing image depth to enhance precision. All of these additions can be combined in a modular fashion. It is our contention that these enhancements will enhance the performance of subsequent computer vision operations.

Although our proposed method has demonstrated considerable enhancements, there are numerous avenues for future research that hold considerable promise. The directions for further improvement detailed in the subsequent discussion section represent a promising avenue for future research and could yield further performance gains. Moreover, an alternative superpixel initialization strategy, as opposed to the initial grid structure, could enhance the accuracy of the model. Dynamic superpixel segmentation enables the assignment of a greater number of superpixels to image regions exhibiting high textural variance, while limiting the number assigned to homogeneous areas. This approach has been employed by only a few methods to date. Integrating this approach into the existing pipeline and conducting further research could yield promising results.

Acknowledgements

I would like to express my gratitude to my supervisor Viktor Varga, who not only supported me in the presented research but also during my whole MSc studies and even before that. I am also grateful for Bosch their financial and professional support, as well as to my industrial supervisors Tamás Gönczy and Márton Vargyas. Special thanks to Lu Fengjiao who gave me emotional support, it helped me a lot to finish this research.

Bibliography

- [1] Varun Jampani et al. “Supersixel sampling networks”. In: *ECCV*. 2018, pp. 363–380.
- [2] Shihao Jiang et al. *Learning to Estimate Hidden Motions with Global Motion Aggregation*. 2021. arXiv: 2104.02409 [cs.CV].
- [3] Lihe Yang et al. *Depth Anything: Unleashing the Power of Large-Scale Unlabeled Data*. 2024. arXiv: 2401.10891 [cs.CV].
- [4] Jianbo Shi and Jitendra Malik. “Normalized Cuts and Image Segmentation”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22.8 (2000).
- [5] Xiaofeng Ren and Jitendra Malik. “Learning a classification model for segmentation”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2003.
- [6] Ming-Yu Liu et al. “Entropy rate supersixel segmentation”. In: *CVPR 2011*. 2011, pp. 2097–2104. DOI: 10.1109/CVPR.2011.5995323.
- [7] P.F. Felzenszwalb and D.P. Huttenlocher. “Efficient Graph-Based Image Segmentation”. In: *International Journal of Computer Vision* 59 (2004), pp. 167–181.
- [8] Radhakrishna Achanta; Appu Shaji; Kevin Smith; Aurelien Lucchi; Pascal Fua; Sabine Süsstrunk. “SLIC supersixels compared to state-of-the-art supersixel methods”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34.11 (2012), pp. 2274–2282.
- [9] Yong-Jin Liu; Cheng-Chi Yu; Min-Jing Yu; Ying He. “Manifold slic: A fast method to compute content-sensitive supersixels”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

- [10] Jiansheng Chen Zhengqin Li. “Superpixel segmentation using linear spectral clustering”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.
- [11] Zhihua Ban, Jianguo Liu, and Li Cao. “Superpixel Segmentation Using Gaussian Mixture Model”. In: *IEEE Transactions on Image Processing* 27.8 (2018), pp. 4105–4117. DOI: 10.1109/TIP.2018.2836306.
- [12] Wei-Chih Tu et al. “Learning superpixels with segmentation-aware affinity loss”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.
- [13] Fengting Yang et al. *Superpixel Segmentation with Fully Convolutional Networks*. 2020. arXiv: 2003.12929 [cs.CV].
- [14] Hankui Peng, Angelica I. Aviles-Rivero, and Carola-Bibiane Schönlieb. *HERS Superpixels: Deep Affinity Learning for Hierarchical Entropy Rate Segmentation*. 2021. arXiv: 2106.03755 [cs.CV].
- [15] S. X. Ju, M.J. Black, and A.D. Jepson. “Skin and bones: multi-layer, locally affine, optical flow and regularization with transparency”. In: *Proceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 1996, pp. 307–314. DOI: 10.1109/CVPR.1996.517090.
- [16] Zhou and Chellappa. “Computation of optical flow using a neural network”. In: *IEEE 1988 International Conference on Neural Networks*. 1988, 71–78 vol.2. DOI: 10.1109/ICNN.1988.23914.
- [17] Shang-Hong Lai and Baba Vemuri. “Reliable and Efficient Computation of Optical Flow”. In: *International Journal of Computer Vision* 29 (Aug. 1998), pp. 87–105. DOI: 10.1023/A:1008005509994.
- [18] Zachary Teed and Jia Deng. *RAFT: Recurrent All-Pairs Field Transforms for Optical Flow*. 2020. arXiv: 2003.12039 [cs.CV].
- [19] Zhaoyang Huang et al. *FlowFormer: A Transformer Architecture for Optical Flow*. 2022. arXiv: 2203.16194 [cs.CV].
- [20] Ashutosh Saxena, Min Sun, and Andrew Y. Ng. “Make3D: Learning 3D Scene Structure from a Single Still Image”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31.5 (2009), pp. 824–840. DOI: 10.1109/TPAMI.2008.132.

- [21] David Eigen, Christian Puhrsch, and Rob Fergus. *Depth Map Prediction from a Single Image using a Multi-Scale Deep Network*. 2014. arXiv: 1406.2283 [cs.CV].
- [22] René Ranftl et al. *Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-shot Cross-dataset Transfer*. 2020. arXiv: 1907.01341 [cs.CV].
- [23] Clement Tan et al. “Self-Supervised Superpixel Correspondences For Video Object Segmentation”. In: *Proceedings of the 2022 4th International Conference on Video, Signal and Image Processing*. VSIP ’22. , Shanghai, China, Association for Computing Machinery, 2023, 117–121. ISBN: 9781450397810. DOI: 10.1145/3577164.3577182. URL: <https://doi.org/10.1145/3577164.3577182>.
- [24] Minhyeok Lee et al. *Unsupervised Video Object Segmentation via Prototype Memory Network*. 2022. arXiv: 2209.03712 [cs.CV].
- [25] Viktor Varga and Milán Szász. “Fast interactive video object segmentation with graph neural networks”. In: *2024 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2024, pp. 1–10.
- [26] David Stutz, Alexander Hermans, and Bastian Leibe. “Superpixels: An evaluation of the state-of-the-art”. In: *Computer Vision and Image Understanding* 166 (Jan. 2018), 1–27. ISSN: 1077-3142. DOI: 10.1016/j.cviu.2017.03.007. URL: <http://dx.doi.org/10.1016/j.cviu.2017.03.007>.
- [27] Ming-Yu Liu; Oncel Tuzel; Srikumar Ramalingam; Rama Chellappa. “Entropy rate superpixel segmentation”. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2011, pp. 2097–2104.
- [28] Huaijia Lin, Xiaojuan Qi, and Jiaya Jia. “Agss-vos: Attention guided single-shot video object segmentation”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 3949–3957.
- [29] Alexander Schick, Mika Fischer, and Rainer Stiefelhagen. “Measuring and evaluating the compactness of superpixels”. In: *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*. 2012, pp. 930–934.

- [30] Andrew Howard et al. *Searching for MobileNet V3*. 2019. arXiv: 1905.02244 [cs.CV].

List of Figures

3.1	Supervoxel initialization and final result. This example presents the different stages of the SLIC algorithm. In the middle, we can see the initial supervoxel grids, the red boundary shows the locality constraint applied during the algorithm. The right picture shows the boundary of the final supervoxel segmentation.	11
3.2	Pipeline of the proposed method: First the image is processed by an optical flow/image depth estimation model. Then the input features are extracted by their respective neural network, these can be used in any combination. The output embedding vectors will be concatenated before the differentiable SLIC will create the supervoxel segmentation	13
4.1	ASA results. We can see that SSN performs better with all the additional module configurations on the DAVIS dataset. Extending with image depth shows also an increased ASA score on the BSDS dataset.	18
4.2	IoU results. The extended SSN versions produce a higher IoU score on the DAVIS dataset. On the BSDS dataset the image depth extended model cannot outperform the original in IoU.	19
4.3	Compactness results. The extended versions of SSN retain the same compactness as the original model.	20
4.4	Qualitative comparison on DAVIS. Row (a) shows the input images, (b) the ground truth label, (c) RGB module's predictions, (d) RGB+depth predictions, (e) RGB+optflow predictions, (f) RGB+depth+optflow predictions.	21

List of Tables

- 3.1 The varying training parameters of the different modules. The last column represents whether the RGB module was included during training. 16