Curriculum Learning for Recurrent Video Object Segmentation

Author:
Maria González Calabuig

Co-Directors:
Xavier Giró Nieto
Carles Ventura Royo

A thesis submitted in partial fulfillment of the requirements
for the Master's degree in Telecommunications Engineering

BARCELONA, JULY 2020

Author
Maria González Calabuig

Co-Directors
Xavier Giró Nieto
Carles Ventura Royo

Project Details

Start date:	February 2020
End date:	July 2020
Defense date:	July 2020
Credits:	30 ECTS

Image Processing Group
Signal Theory and Communications Department
Universitat Politècnica de Catalunya. BARCELONATECH

Campus Nord UPC 1-3
Carrer de Jordi Girona
08034 Barcelona

Abstract

Video object segmentation (VOS) is a computer vision task that aims at determining the pixels of an object of interest along a video sequence. This thesis explores different curriculum learning strategies for a deep neural network trained to solve this task.

Curriculum learning defines a methodology where the training data are not randomly presented to the model, instead, they are organized in a meaningful way. Simple concepts are first presented and gradually become more complex. Four different curriculum strategies are explored: schedule sampling, frame skipping, the effect of temporal and spatial recurrence variations and loss penalization by the object's area.

This work focuses on the RVOS neural architecture, a recurrent architecture originally tested on the DAVIS and YouTube-VOS datasets for one-shot video object segmentation, over the cars class of the KITTI-MOTS dataset. Even though this architecture is a fast solution for the VOS task, the model struggles with the KITTI-MOTS dataset, whose videos are more crowded and challenging.

For the schedule sampling curriculum, both the classic and inverse implementations are evaluated. Results show how inverse schedule sampling strategies improve the model's performance instead of the classic approach, the forward one. The different frame skipping schemes are also beneficial, but only when training with the ground truth mask instead of the predicted ones. Lastly, both the curriculums that vary the temporal and spatial recurrence or penalize the loss by the object's area have shown poor model's performance.

These results show how curriculum learning strategies affect greatly the performance of recurrent neural networks. Moreover, the results on the inverse schedule sampling and frame skipping strategies invite to further explore these schemes to exploit their benefits.

Acknowledgements

I would like to offer my special thanks to my co-directors, Xavier Giró and Carles Ventura for all the given advice. Their willingness to give their time, their help and guidance is very much appreciated.

I would also like to thank my family, partner and friends for all their support during all these months. Thank you for always being here.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Motivation

During the past few years, computer vision has been gaining popularity for its potential and its applications. This field of computer science focuses on the study of how computers "see" and understand images and videos. It tries to replicate the complexity of the human visual system so computers can identify and process objects in images and videos in the same way that humans do.

Until recently, computer vision performance was limited and required a large amount of manual effort by developers. It is not until machine learning and, more precisely, deep learning algorithms were introduced that this field was boosted. Deep learning uses logical structures that mimic the human brain and its neurons, creating a network of artificial neurons in order to detect and determine the characteristics of the perceived objects. This entailed an efficient, fast and easy approach for development and deployment of task related to this field.

Under this context appears Video Object Segmentation (VOS), a task which is still very challenging in the research community. Video Object Segmentation's objective is to determine the pixels which correspond to the objects of interest in the consecutive frames of a video sequence. Figure 1.1 shows two examples of this task. Among the multiple challenges that it faces; occlusions, deformations or scale variations are some examples. The rapid development of intelligent mobile terminals and the Internet has resulted in an increment in video data. This fact has put Video Object Segmentation in the spotlight, in order to be able to analyze and use this data in an efficient way.

Figure 1.1: Example of video object segmentation on five consecutive frames of two video sequences of the YouTube-VOS dataset [1].

Video Object Segmentation is a versatile task, which can be applied to a wide range of practical applications: autonomous vehicles, high definition video compression, large video collections analysis and activity recognition among many others.

The optimization process of deep neural networks is greatly influenced by how training data is used. This thesis focuses on applying curriculum learning strategies on a recurrent neural network model which addresses the one-shot VOS task in order to improve its performance. One-shot VOS fits under semi-supervised VOS and it consists of giving the manual annotation of the first time that an instance appears to the model in order to estimate the instance segmentation for the remaining frames until the instance disappears [2].

Curriculum learning describes a methodology of training where the examples are not randomly presented to the model, instead, they are organized in a meaningful way. Simple concepts are first presented and gradually become more complex [3]. These training techniques are inspired by the learning processes of humans and animals. Humans' training is highly organized and structured to ease learning and increase the speed of the learning process. This raises the question of whether artificial intelligence based models can benefit from this methodology.

Four training curriculums will be presented, followed by its implementation and results, both quantitative and qualitative. These four strategies are performed on the *cars* class of the KITTI-MOTS [4] benchmarks and the most promising ones on the YouTube-VOS [5]. The model that has been used is the End-to-End Recurrent Network for Video Object Segmentation (RVOS) which outperforms state-of-art results on YouTube-VOS and DAVIS-2017 benchmarks [6] for models that do not use online learning.

1.2 Thesis Outline

The thesis is organized in five chapters.

Chapter 2 exposes the state of the art of curriculum learning techniques on video object segmentation and presents the different techniques that have been tested.

Chapter 3 explains the implementation of the four training curriculums and its variations, as well as its settings and specific parameters.

In Chapter 4, the results of the multiple experiments applying the curriculum techniques are presented.

Finally, in Chapter 5, the conclusions of the thesis are presented.

2 State of the art

Curriculum learning is referred to the concept of training a model starting with simple concepts and, gradually, increase the complexity of these. Similar to what humans do, it tries to enhance the learning by structuring the concepts that are to be fed to the model by creating a "curriculum". It first was presented by Bengio et al. [3] hypothesising that curriculum learning was able to boost the convergence speed of the training process as well as find a better local minimum than the existing solvers for non-convex problems.

On the literature, some examples of curriculum learning can be found applied to a diverse number of tasks. One example is proposed by Gong et al. [7], where a curriculum learning approach is used on image classification. The authors generate a multi-modal curriculum learning in a simple-to-difficult order to optimize the quality of semi-supervised image classification. Hacohen and Weinshall [8] use curriculum learning on image recognition. Two problems are addressed: (i) sort the training examples by difficulty; (ii) compute a series of mini-batches that exhibit an increasing level of difficulty. They define a curriculum learning algorithm by a scoring function and a pacing function. The scoring function evaluates the difficulty and the pacing function evaluates the rhythm in which the difficulty is increased. The authors end defining the concept of an ideal curriculum.

The tasks where recurrent neural networks (RNN) have had the most impact are the ones which involve text analysis. This type of artificial neural networks is commonly used in speech recognition and natural language processing. RNNs are designed to recognize sequential characteristics of data and use patterns to predict the next likely scenario which makes them ideal for tasks which involve time-series predictions. On the area of speech recognition, Shi et al. [9] propose three curriculum learning strategies on language modelling. On the other hand, Platanios et al. [10] are an example of recurrent neural networks focusing on neural machine translation by using curriculum learning which reduces training time, reduces the need for specialized heuristics or large batch sizes, and results in overall better performance. To do so, the authors decide which training samples are shown to the model at different times during training, based on the estimated difficulty of a sample and the current competence of the model. On the area of natural language processing, Rao et al. [11] and Sido and Konopík [12] focus on sentiment analysis using similar strategies to the ones exposed previously. Both works focus their strategy on the sentence length, in

which the longer the sentence, the more difficulty it supposes to the model. Moreover, Sido and Konopík [12] also experiment with the frequency of the words.

As explained above, curriculum learning has been widely used for many tasks, providing interesting results on these. Focusing on video object segmentation, a large number of works have relied on curriculum learning strategies to improve the performance of their models [1, 13–21]. The concept behind curriculum learning can be implemented in multiple ways. This thesis focus its analysis on four strategies: scheduled sampling, frame skipping, from only temporal to spatio-temporal and creating a curriculum by the object's area.

2.1 Scheduled Sampling

Introduced by Bengio et al. [14], scheduled sampling is a curriculum learning approach which objective is to slowly eliminate the gap between training and inference for sequence prediction tasks using recurrent neural networks. It was proposed for sequence prediction with recurrent neural networks, and successfully applied in the wining bid in the MSCOCO image caption challenge 2015.

When training recurrent neural networks, an interesting technique widely used in the literature is teacher forcing. Teacher forcing is a strategy which replaces the generated output of a unit by the ground-truth or actual output in subsequent computation, using this latest as an input in the next training step [22]. This technique provides a fast and effective way to train a recurrent neural network and boosts the convergence speed of the model. Nevertheless, this training methodology leads to exposure bias: discrepancy between training and inference. During training, the model is trained on the ground-truth data distribution while in inference, the predictions are conditioned to what the model generates itself. This difference results in instability and poor model performance. Schedule sampling takes benefit from teacher forcing while avoiding exposure bias by gradually replacing the ground-truth tokens by the model's predictions. With this strategy, the model is forced to gradually learn to deal with its own mistakes as it would during inference.

On the original work [14], three different decays were proposed: exponential, inverse sigmoid and linear. On the literature, works on instance segmentation apply the linear schedule sampling. Ren and Zemel [16] define a stochastic switch (θt) in the input of the external memory regulates whether to use either the maximally overlapping ground-truth instance segmentation or the output of the network from the previous time step. By the end of the training, the model completely relies on its own output from the previous step, which matches the test-time inference procedure.

Lai and Xie [15], in order to tackle the challenges from tracker drifting, due to complex object deformations, illumination changes and occlusions, propose training with this schedule sampling strategy. Their strategy does not go from all ground truth labels to total model predictions in their linear schedule, but consider of having a higher probability value (0.9) of using ground-truth frames in early training stages

and uniformly anneal to a probability of 0.6. This forces the model to recover from error states and improves the robustness to drifting.

Finally, Xu et al. [1] and RVOS [6] define a more drastic scheme, using ground truth labels in the first half of the training, and predicted masks in the second half. In this Master thesis, this approach has been named as *step* schedule, as in the well-known Heaviside step function. Xu et at. [1] go even further and once the training losses become stable, the ground-truth annotations are replaced by the model's predictions.

Motivated by the affirmations of Huszár [23], when studying schedule sampling, the question whether an inverse strategy makes sense arises. The author affirms that, for generative models, schedule sampling is an inconsistent training strategy. Even though his work does not take on video object segmentation, inverse strategies have been tested in this thesis when studying the schedule sampling approach to provide a full understanding of the technique.

2.2 Frame Skipping

Frame skipping is a training curriculum in which video sequences are progressively sub-sampled in time, so that the model is exposed to sequences with faster changes, even if synthetically generate. It appears partially motivated by the limitations of the number of frames that the model can see per mini-batch due to memory constraints. These constraints force the model to train with short sequences of frames which may cause redundancy in the case of training with consecutive frames.

On the literature, Oh et al. [17–19], rely on this strategy on their different works that address video object segmentation. On [17], they introduce this concept for their model, the Space-Time Memory Networks (STM). STMs achieved the state of the art on one-shot video object segmentation by randomly skipping frames during sampling in order to learn the appearance change over a long time. They limited the maximum number of frames to be skipped to 25 and applied a curriculum learning strategy which gradually increases the skipped number of frames from 0 to 25. On their previous works [18] and [19], they use a similar strategy of randomly skip frames to simulate fast motion, with the difference that in [18] they also gradually increased the length of a training video clip from 4 to 8.

Alabed et al. [20] propose three implementations of this technique for video object segmentation: downsampling by 67% (skip two frames every three frames for slow-moving objects), 50% (skip a frame every two frames for moderate speed moving object), and 0% (no frame skipping for fast-moving objects).

Finally, Wu et al. [24] have achieved relevant gains when addressing the action recognition task with a neural network that processes the video streams at a fast and a slow frame rates in two different pathways that merge at the deepest layer. In this thesis, a single pathway is kept but considering different frame rates during the training curriculum.

2.3 From temporal only to spatio-temporal

The work on [21] is an example in which spatio-temporal CNN is trained for video object segmentation where the temporal branch is pre-trained separately. Specifically, it consists of first training with only temporal information and later adding spatial information, providing both temporal and spatial information to the model in the latest phases of the training process.

2.4 Loss penalization by object area

The loss penalization by object area technique discriminates which instances are used for training on each stage of the process. It first starts training with instances considered easy for the model, omitting the rest. Latter, the difficult ones are added. Even though no related works have been found for VOS, works on a closely connected task, object detection, use this technique. Wang et al. [25] and Siyang et al. [26] present in their works curriculums which discriminate the instances by difficulty when training. The authors determine the difficulty of the instances by how their model performs on them. Then, they start training with this type of "easy" data and latter add the remaining one.

3 Methodology

This chapter presents the methodology followed to perform the different experiments, starting with the environment set-up and narrowing down to the implementation of each strategy. The set-up includes the details of the datasets used on the experiments. After that, context of the RVOS model is given, in order to understand its performance. Finally, the implementation and intuition behind each strategy are explained.

3.1 Datasets

This thesis focuses mainly on the KITTI-MOTS benchmark [4], even so, some of the most promising strategies have been tested on the YouTube-VOS benchmark [5] in order to obtain a robust insight on their effect.

3.1.1 KITTI-MOTS Benchmark

The KITTI benchmark was introduced in 2012 to address the autonomous driving challenge [4] but it was not until 2018 when a semantic segmentation and semantic instance segmentation approach to the benchmark was presented. This dataset consists of a set of video sequences captured by driving around a mid-size city, in rural areas and on highways. Figure 3.1 is an example of the type of images that can be found in this dataset.

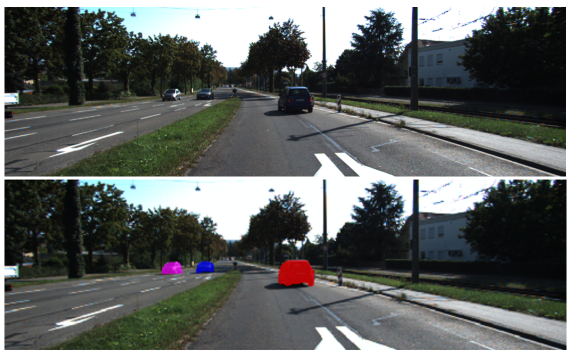


Figure 3.1: Example of images from the KITTI-MOTS dataset. On top, the original image; on the bottom, the original image with the ground-truth annotations superimposed.

The annotations are provided in both text or image format. The background is defined by an id value of 0. Ignore regions are defined as 10.000. The class id for cars is defined as 2 and the class id for pedestrians is 1. An object will have an object id of the order of 200X for a car and 100X for a pedestrian. The object id is maintained during all the video sequence.

The sequences are split into train, validation and test sets. The training set is composed of 12 sequences while the validation set is composed of 9. Figure 3.2 shows the distribution and the number of cars and pedestrians of each set. It can be seen how many instances (denominated "# Tracks") the two sets contain as well as the total number of masks annotated to track these instances (denominated "# Masks"). For either set, the number of cars and the masks that they generate at least double the number of pedestrians and their masks.

The test set has not been used for the experiments. The KITTI-MOTS competition addresses a zero-shot challenge while this thesis has been focused on addressing a one-shot challenge. On one-shot learning, the manual annotation of the first time that an instance appears is given to estimate the instance segmentation for the remaining frames until the instance disappears [2]. With zero-shot learning, there is no initialization to perform the frames segmentation. While a one-shot approach fits under semi-supervised video object segmentation, zero-shot fits under unsupervised video object segmentation. RVOS, the model which has been used for all the experiments, has demonstrated better performance and results with one-shot learning, which has been the motivation of choosing this approach. This choice has resulted in a modification on the official splits, as the test slip does not contain its annotations and, therefore, cannot be used on a one-shot approach.

	KITTI MOTS		MOTSChallenge
	train	val	
# Sequences	12	9	4
# Frames	5,027	2,981	2,862
# Tracks Pedestrian	99	68	228
# Masks Pedestrian			
Total	8,073	3,347	26,894
Manually annotated	1,312	647	3,930
# Tracks Car	431	151	-
# Masks Car			
Total	18,831	8,068	-
Manually annotated	1,509	593	-

Figure 3.2: Information about the sequence splits on the KITTI-MOTS dataset [27].

For the reasons exposed above, the training set has been split into two sets: a train-train set, with 9 sequences, and a train-val set, with 3 sequences. The official validation set has not been modified and has been used as the test set.

The challenges that offers this dataset are varied. The sequences are conformed by a large number of frames, being the shortest one compounded by 78 frames and

the longest one by 1059 frames. Motion changes are observed. There are right and left turns which are fast and happen in few frames while the majority of the other scenes are slow paced, meaning that it takes a higher number of frames for a change to be noticed. Multiple occlusions, both partial and total, take place on the sequences. There are also changes in illumination and resolution on the objects. Due to all of that, this dataset presents many challenges.

3.1.2 YouTube-VOS Benchmark

Presented on [5] in 2018, YouTube-VOS is the largest video object segmentation dataset. It contains 4,453 YouTube video clips from 94 selected object categories. The categories include animals, vehicles, common objects and humans in various activities in order to provide a complete and comprehensive dataset. The YouTube-VOS dataset is used for unsupervised, semi-supervised, interactive, and weakly supervised VOS approaches. An example of the type of images that this dataset contains can be seen on Figure 3.3.

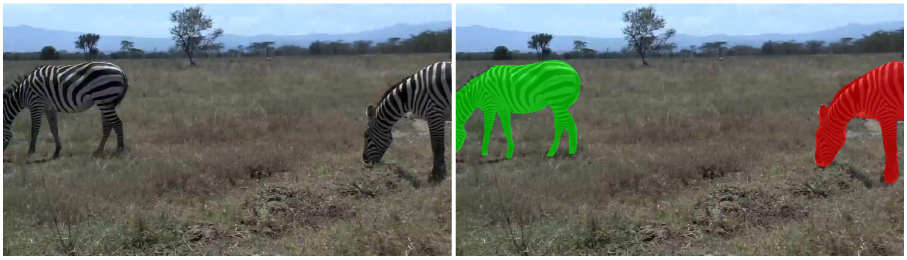


Figure 3.3: Example of images from the YouTube-VOS dataset. On the left, the original image; on the right, the original image with the ground-truth segmentation masks superimposed.

The whole dataset which consists of 4,453 videos is split into training (3,471), validation (474) and test (508) sets. Since the dataset is used for competitions, the test set is only available during the competition period. This is the reason why the validation set is used for evaluation instead, as this set will be always publicly available. To train the model, the same 80%-20% split of the training set as in [6] is used.

3.2 The model

The model that has been used is the End-to-End Recurrent Network for Video Object Segmentation (RVOS) [6]. It has been developed by researchers of three different entities: Universitat Oberta de Catalunya, Barcelona Supercomputing Center and Universitat Politècnica de Catalunya.

RVOS is an end-to-end trainable model which incorporates recurrence on two different domains, spatial and temporal. We understand as an end-to-end model those models where all modules are differentiable and, when training, gradient-based learning can be applied to the system as a whole [28]. Meanwhile, spatial recurrence allows

to discover the different object instances within a frame and the temporal recurrence allows to keep the coherence of the segmented objects along time.

This model is used for video object segmentation that tackles multi-object segmentation. It is able to perform either in one-shot or zero-shot learning. This thesis has been focused on working with the one-shot scenario, as mentioned before.

The architecture of the model can be seen in Figure 3.4. It is based on an encoder-decoder architecture for both scenarios mentioned above. For the encoder, a pre-trained model of ResNet-101 [29] is used. For the decoder, it is designed as a hierarchical recurrent architecture of ConvLSTMs [30]. The optimal assignment between predicted and groundtruth masks is found with the Hungarian algorithm using the soft Intersection over Union score as cost function. The input of the model will consist of a set of RGB image frames of a video sequence. As we are in a one-shot scenario, the mask of the objects at the frame where each object appears for the first time will be also part of the input of the model. A set of object segmentation predictions for each frame will be output at the end of the decoder.

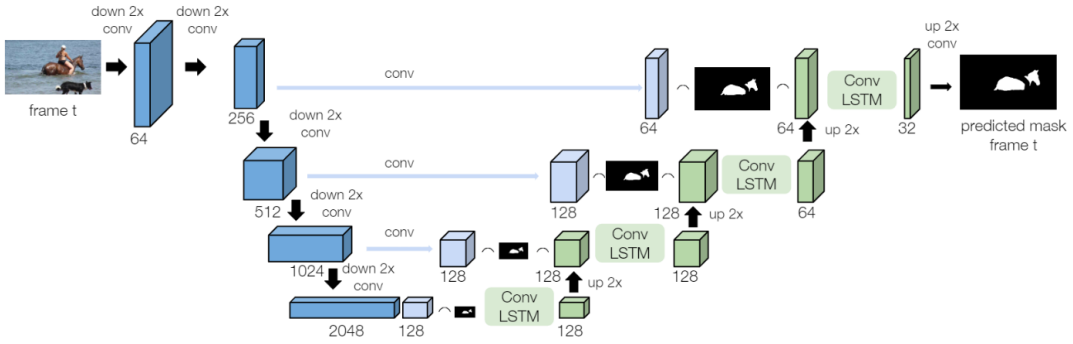


Figure 3.4: Model architecture for a single frame at time step *t*. A single forward is of the decoder is depicted, predicting only the first mask of the image [6].

The concept behind the implementation of the spatio-temporal recurrence is shown in Figure 3.5. Each ConvLSTM layer, on one hand, depends on the preceding ConvLSTM layer, the features obtained from the encoder from the same temporal frame and the object segmentation prediction mask of the object at the previous frame. On the other hand, it will also depend on two hidden states; the temporal, which is the representation from the same object at the previous frame and the spatial hidden state, which is the representation from the previous object at the same frame.

The RVOS model has been tested on YouTube-VOS [5] and DAVIS-2017 benchmarks [31]. It has been proved that adding spatio-temporal recurrence outperform models which only consider the spatial and temporal domains. At the same time, the model outperforms sate-of-the-art techniques that do not make use of online learning for one-shot scenarios.

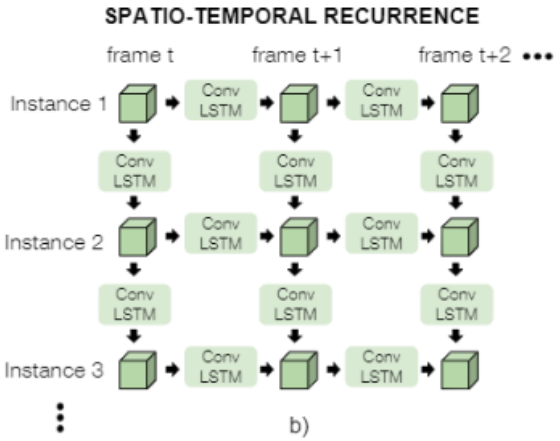


Figure 3.5: Spatio-temporal recurrence scheme [6].

3.3 Implementation

On this section, the implementation of the different strategies which apply curriculum learning will be explained as well as the motivation behind them.

3.3.1 Scheduled sampling

As explained on Chapter 2, Section 2.1, schedule sampling is a curriculum learning strategy where, gradually, the use of the ground-truth annotations as input on the next step is replaced by the model's output. The motivation behind schedule sampling is to obtain a robust model able to learn from its own mistakes. The gradual change from ground-truth annotation to the model's output can be implemented in multiple ways. On [14], the authors which presented this training strategy present some examples of schedule decays (Figure 3.6).

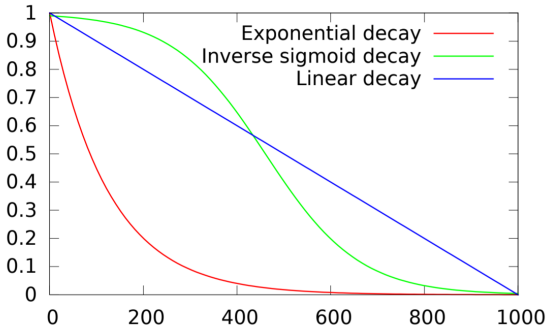


Figure 3.6: Examples of decay schedules [14].

RVOS [6] was already implemented using a hard schedule sampling strategy. The authors were training half time with the ground-truth annotations and, on the last half of training, they switched to train with the model's outputs of the previous step as inputs. This implementation can be seen as a step decay, depicted in Fig. 3.7a. This

methodology was already proved on [6] to improve considerably the performance of the model. On this thesis, a more gradual schedule decay has been studied to find if these results can be improved furthermore: the linear decay.

Three schemes have been implemented and analysed: no schedule sampling, forward schedule sampling and inverse schedule sampling.

No Schedule Sampling

This implementation refers to not using a schedule sampling scheme. There is no change from ground-truth annotations to the model's outputs. It consists of applying a teacher forcing approach, using all the time ground-truth annotations.

Forward Schedule Sampling

The forward scheme implements a schedule sampling strategy as defined on [14]. To implement a linear decay, the model chooses according to a threshold if it uses the ground-truth annotations or the model's output as inputs. A random number $\in (0, 1)$ is generated, if it is higher than the actual threshold, the model chooses to use the ground-truth annotations. If the random number generated is smaller than the threshold, the model chooses to use its output as inputs on the following step. The threshold starts with a value of 1 ($threshold_{i=0} = 1$) and linearly decreases during each epoch (n) following the next equation, for $n > 0$:

$$threshold_{i=n} = threshold_{i=n-1} - \frac{1}{\text{maximum epoch}} \tag{3.1}$$

At the start of the training process, ground-truth annotations are always chosen. In the same way, at the end of the training, the model's output is always chosen. During the rest of the epochs the probability of choosing ground-truth over outputs decreases. The forward step can be seen as following the same reasoning as the linear decay but implementing the following equation:

$$threshold = \begin{cases} 1, & n < 20 \\ 0, & n \geq 20 \end{cases} \tag{3.2}$$

Inverse Schedule Sampling

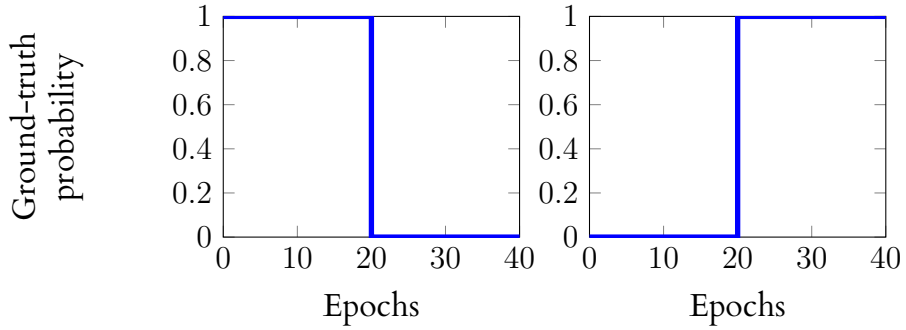
The inverse scheme implements a contrary schedule sampling methodology. The threshold varies following Eq. 3.3. It starts with a value of 0 ($threshold_{i=0} = 0$) and linearly increments during each epoch (n) until reaching a value of 1, for $n > 0$. In this case, at the start of the training process, the model's outputs will always be chosen and, at the end of the training, the ground-truth annotations will always be chosen.

$$threshold_{i=n} = threshold_{i=n-1} + \frac{1}{\text{maximum epoch}} \tag{3.3}$$

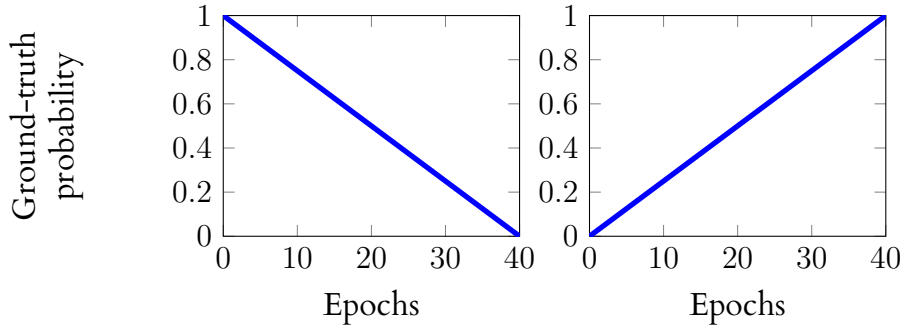
As one-shot learning is being applied, the model is trained using the first ground-truth annotation for each object. The model does not start blind. The hypothesis from inverse schedule sampling is that, if the model starts training with its model's outputs, it will force the model to focus on the areas of the image where there are objects, as the error, and therefore the loss penalization, will be higher than when using perfect outputs. This way, the objective is to study if the training process can speed up by using this scheme.

Apart from the linear approach, it has also been implemented as an inverse step schedule sampling scheme. As in the forward step approach, the model trains half time using ground-truth annotations and the other half using the model's outputs. In the inverse approach, it starts training with the model's outputs and changes to ground-truth annotations on the second half of training. The inverse step is implemented following Eq. 3.4.

$$threshold = \begin{cases} 0, & n < 20 \\ 1, & n \geq 20 \end{cases} \tag{3.4}$$



(a) Forward Step (b) Inverse Step



(c) Forward Linear (d) Inverse Linear

Figure 3.7: Decay schemes implementing schedule sampling.

3.3.2 Frame Skipping

The second curriculum learning technique that has been studied is frame skipping. As mentioned in previous sections, this strategy is motivated by the appearance changes

of the objects on video sequences. Limited by the number of consecutive frames that the model sees on a training iteration, parameter named length clip, if the changes of a sequence are slow, the model may not be taking the most advantage of the information of the sequences during training.

The KITTI-MOTS dataset is compounded by slow motion sequences, the appearance changes are slow. A sequence may have a large number of frames with almost no variation. This fact has been illustrated in Figure 3.8. The three images on the top are three consecutive frames, skipping step of 0. On the other hand, the three images on the bottom have a skipping step of 9 frames between each one of them. While on the top row there is almost no change, on the bottom row it can be seen how the camera is making a turn to the right. The model sees more changes, making it more robust to these variations.

Figure 3.8: Comparison on the training sequences of the model with and without frame skipping.

By applying curriculum learning, the experiments under frame skipping have been performed by gradually increasing the number of frames skipped between the frames shown to the model. Two schemes are considered:

- Skipping from 0 to 9 frames

  For this implementation, at the start, no frames are skipped. After that, every number of epochs, the number of skipped frames increases until 9 consecutive frames are skipped. The total number of epochs is divided equally so each skipping step is trained during the same number of epochs. A skipping step is defined as a finite number of skipped frames. In this scheme, there are 10 skipping steps. From one skipping step to the next, the increment unit is the addition of one more frame to skip.

  KITTI-MOTS sequences are long enough to skip 9 consecutive frames and still be able to train correctly the model. This maximum number of skipped frames has been chosen taking into account the number of epochs for each training step and the speed of the changes in the sequences. A higher maximum number of skipped frames implies less training epochs for each skipping step. This could lead to the model not having enough time to learn the changes in motion. Moreover, after analysing the motion of different sequences, it has been concluded that when skipping 9 frames, the model is able to see significant changes.

Increasing more this skipping step would not contribute to the information provided to the model. Further details of this study can be seen in Appendix A.

- Skipping from 1 to 5 frames

  In this case, the model starts training with one skipped frame and increases the skipping step until 5. In the same way as in the previous implementation, the total number of epochs is equally divided to provide the same training time for each skipping step. Compared to the previous scheme, this implementation reduces the skipping steps from 10 to 5, providing the double amount of training time for each skipping step.

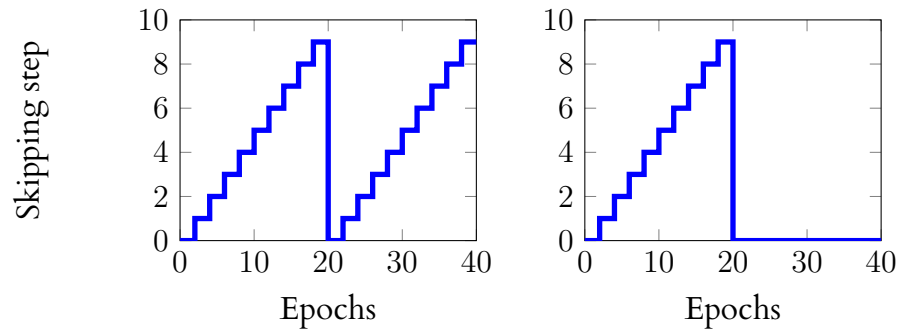
  The motivation behind this scheme arises from questioning the training time available for each skipping step. This approach tries to find the optimum balance between the changes that the model sees during training and the training time for each one of this changes, as the changes that the model sees are different on each step. The model starts training with a skipping step of 1 instead of 0 due to the similarity between two consecutive frames. It has been observed that the sequences shown to the model without skipping any frame offer almost no change. The evolution of the sequences is too slow for consecutive frames to caption any variation. Due to this similarity, the model ends seeing several consecutive frames as if it was only one frame due to the information that it provides. The length clip is not used to its most. It has been considered that starting to train without skipping frames and, later, increase to skip one frame was not necessary. Furthermore, increasing the step until 5 provided enough changes to the sequences and allowed to duplicate the training time per step. The whole study can be found in Appendix A.

For the YouTube-VOS benchmark, the first scheme, which skips from 0 to 9, has been adapted. As the sequences on this benchmark are shorter than in KITTI-MOTS, the skipping scheme is from 0 to 3 consecutive frames skipped. This is the maximum number of skips that these sequences allow.

As will be explained on Chapter 4, the baseline model which has been used on most experiments implements a Forward Step Schedule Sampling (FSSS) strategy, as it was used on the original paper of RVOS [6]. This means that there are two training stages, the first where the ground-truth annotation is used as input of the next step, and the second stage where the outputs of the model are used instead.

While on the first half of training frame skipping is always implemented, during the second half both with and without frame skipping has been studied. This results in two training methodologies: on one hand training during all time with a frame skipping strategy and, on the other hand, only training with frame skipping when using ground-truth annotations. When using the first strategy, the skipping step is restarted to the original value (skipping step of 0 or skipping step of 1) when the second phase starts. On the second strategy, when training with the model's outputs, no frame is skipped. As different strategies are combined, FSSS and frame skipping,

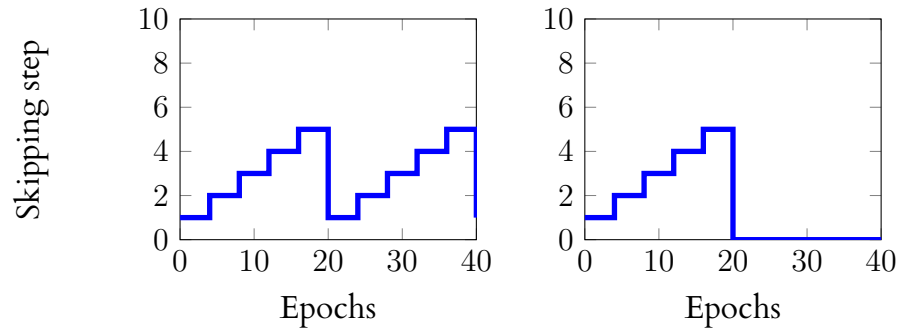
an excessive increment on the difficulty wants to be avoided. The four combinations are illustrated in Figures 3.9 and 3.10.



(a) Scheme applied during all the training process

(b) Scheme applied on the first half of the training process

Figure 3.9: Training strategies for the skipping scheme from 0 to 9.



(a) Scheme applied during all the training process

(b) Scheme applied on the first half of training

Figure 3.10: Training strategies for the skipping scheme from 1 to 5.

3.3.3 From temporal only to spatio-temporal

As explained on Section 3.2, RVOS [6] implements spatio-temporal recurrence. This model gives the option to train by using spatio-temporal recurrence or only using the temporal recurrence. Even though the authors of the paper define that using spatio-temporal recurrence during all training obtains the best performance, the effect of the two recurrences on KITTI-MOTS has been questioned. For this reason, the set of experiments on this section explore the effect of this information on the model's performance.

Different cases can be defined when exploiting and combining the two options. The first one would be training with spatio-temporal recurrence during all the training time while the second case would be training only with temporal recurrence. Nevertheless, under the curriculum learning context, more emphasise is made into the case where the model is first trained with only temporal recurrence and, later, spatial

recurrence is added, training the model with spatio-temporal recurrence during the last half of the training time. Using a reverse curriculum learning approach, the opposite procedure has also been implemented. In this case, the model starts training with both spatio-temporal recurrence and, in the second half of training, only temporal recurrence is used.

3.3.4 Loss penalization by object area

This approach tries to take benefit on first learning easier objects, which are bigger, with more resolution and more defined. This technique motivation is to address the challenge that entails the changes in resolution of the objects during a video sequence. To do so, the model is forced to focus only on penalizing the big and medium objects at first. Once a certain number of epochs has passed, the small objects are also taken into account.

To define which objects are considered small, medium or big, the official definition on COCO has been used [32]. For a resolution of 480x640, an interval of values is defined for each category. These intervals have been adjusted to the resolutions used on the experiments. The used intervals for a resolution of 256x448 are the following:

- Big objects: [59^2,$1e5$]
- Medium objects: [20^2, 59^2]
- Small objects: [0, 20^2]

Meanwhile, the used intervals for a resolution of 287x950 are the following:

- Big objects: [90^2,$1e5$]
- Medium objects: [30^2, 90^2]
- Small objects: [0, 30^2]

To penalize only big and medium objects, the loss has been masked. For each object, the number of pixels is computed using the ground-truth. To focus on big and medium objects, if the area does not surpass a value of 20^2 for an image resolution of 256x448 or 30^2 for an image resolution of 287x950, the object is not taken into account.

3.4 Results Preparation

RVOS outputs the predictions generating a white and black image for each detected instance. For a frame with 3 cars, it will generate three images, one per car. Each instance will be depicted in white on a black background. To evaluate the results, post-processing of these images has been done where the images which correspond to the same frame have been combined. The importance of this post-processing falls

on how is made the decision when two instances overlap.

In the KITTI-MOTS dataset, as instances are close to each other, there is a great number of overlapping instances. For these cases, the decision has been made by observing the behaviours of the area of the instance on previous frames. A common behaviour that has been observed when an instance overlaps completely another one is the fast change of the object's area. For example, the growth in the area when an instance which was first perceived as one single car changes to being perceived as two cars is fast in time (in one or two frames) and significant. For this reason, in the case of overlapping, the area of each instance is calculated on the two previous frames. These two values are added and compared between instances. The instance with less area change over the two previous frames is the one which is considered to be correct in case of overlapping.

4 Experiments

In this chapter, the results obtained with each of the experiments that have been performed are exposed. First of all, the used metrics for evaluating and comparing the models' results are explained. After that, the results of each experiment are presented.

4.1 Evaluation metrics

For the experiments performed with the KITTI-MOTS benchmark, the metrics used to evaluate the obtained results follow the official procedure of the KITTI-MOTS challenge [33]. The organizers of the challenge provide a GitHub repository with scripts to evaluate the results [34]. These scripts offer the score performance on a wide range of metrics and they also compute values of interest such as the number of true positives or false positives, for example.

Four metrics of the available range have been selected. These are sMOTSA, MOTSP, Recall and Precision. These four metrics provide a full and global vision of the models' performance. All these metrics are separately computed for the cars class and the pedestrians class. On this thesis, the focus is made on the cars class due to the imbalance of classes, which can be seen in Figure 3.2.

- **sMOTSA: soft Multi-object Tracking and Segmentation Accuracy**

  This metric has been taken as the reference in order to compare the performance between models. The reason why this is the reference metrics can be found on [35], where the different multi-object tracking segmentation challenges are defined. On the official site, it is specified that the sMOTSA score is the one used to evaluate and compare the performance of the models participating. sMOTSA is defined as follows [27]:

  $$sMOTSA = \frac{\widetilde{TP} - |FP| - |IDS|}{|M|} \tag{4.1}$$

  This metric measures segmentation as well as detection and tracking quality. $\widetilde{TP}$ refers to the soft number of true positives. Instead of counting the number of true positives TP by counting how many masks reach an IoU of more than 0.5, for $\widetilde{TP}$ all the true positives are accumulate. $|FP|$ refers to false positives; $|IDS|$ refers as ids switches, which means the set of ground truth masks whose

predecessor was tracked with a different id; and $|M|$ is the non-empty ground-truth pixel mask.

- **MOTSP: Multi-object Tracking and Segmentation Precision**

  This metric implements a mask IoU based version of MOTP defined on [36]. It represents the total error in the estimated position for matched object-hypothesis pairs over all frames, averaged by the total number of matches made. It shows the ability of the tracker to estimate precise object positions, independent of its skill at recognizing object configurations, keeping consistent trajectories, and so forth.

$$MOTSP = \frac{\widetilde{TP}}{|TP|} \tag{4.2}$$

- **Recall**

  The recall metric computes the ratio of actual positives that are captured by the model's prediction.

$$Recall = \frac{TP}{TP + FN} \tag{4.3}$$

- **Precision**

  The precision metric measures the proportion of actual positive predictions out of the total positive predictions.

$$Precision = \frac{TP}{TP + FP} \tag{4.4}$$

On the experiments, the overall metrics have been computed following two procedures. The first one follows the official evaluation of the KITTI-MOTS challenge. It computes the metrics on the total amount of predicted masks of all sequences equally, averaging the metrics per pixel. In this work, a second procedure to obtain the overall metrics is presented, due to the unique characteristics of each sequence and the unbalance of the number of frames, depicted on Figure 4.1. The overall metrics are obtained averaging per sequence to avoid the domination of the results over very long sequences with specific challenges.

The metrics used to evaluate the experiments performed on YouTube-VOS are different from KITTI-MOTS. The evaluation metrics used are the ones used on the workshop of YouTube-VOS which are defined on [37].

The first metric is the region similarity $\mathcal{J}$. It measures the number of mislabeled pixels. It is defined as Eq. 4.5 where M is a given output segmentation and G is its corresponding ground-truth mask.

$$\mathcal{J} = \frac{M \cap G}{M \cup G} \tag{4.5}$$

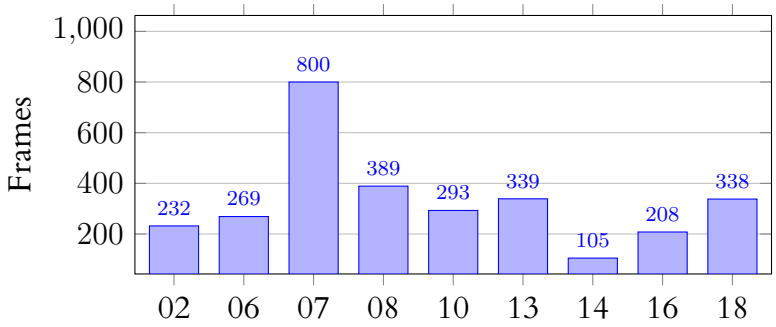


Figure 4.1: Frames per sequence of the validation set on the KITTI-MOTS benchmark.

The second used metric is the contour accuracy $\mathcal{F}$, computed as defined on Eq. 4.6. P_c and R_c are the contour-based precision and recall, computed between the contour points of $c(M)$ and $c(G)$, via a bipartite graph matching in order to be robust to small inaccuracies. $c(M)$ is the set of closed contours delimiting the spatial extent of the mask. The same definition extends to $c(G)$, applied to the ground-truth mask.

$$\mathcal{F} = \frac{2P_cR_c}{P_c + R_c} \tag{4.6}$$

For both metrics, two categories are differentiated: seen and unseen categories. On the YouTube-VOS dataset, in the training set, there are 65 object categories which are regarded as seen categories. In the validation set, there are 91 unique object categories which include all the seen categories and 26 unseen categories. The unseen categories are used to evaluate the generalization ability of different algorithms. The evaluation is performed on these two categories, providing the performance score of region similarity $\mathcal{J}$ and contour accuracy $\mathcal{F}$ for both seen and unseen categories.

Lastly, an average of the four metrics is computed and defined as "Overall". Due to computational and time limitations, not all training methodologies have been performed on the YouTube-VOS dataset. Only the most promising strategies have been tested on this benchmark. The promising strategies have been chosen based on the best results obtained on KITTI-MOTS.

4.2 Experiment Sets

On the following sections, the results of the different curriculum learning strategies are exposed. For each technique, two sets of experiments are presented. The first set of experiments are performed using an image resolution of 256x448, a batch size of 4 and a length clip of 5 consecutive frames. These are the original parameters in which RVOS was trained for YouTube-VOS. The second set of experiments uses an image resolution of 287x950, a batch size of 2 and a length clip of 3. The image resolution of these models is 287x950, which maintains the aspect-ratio of the original images of the KITTI-MOTS benchmark, which is approximately 3,3; value obtained by dividing the width of the image by its height. At the same time, compared with the

resolution of the first set of examples, in this case, the image resolution is bigger. Its the highest value that the models allow in terms of memory usage when using a batch size of 2 and length clip of 3, the minimum acceptable values for these parameters. A length clip value lower than 3 when tackling the video object segmentation task is considered unsuitable. These two sets of parameters have been found to provide the best performances compared to other combinations of image resolutions, batch size and length clip. Table 4.1 and Table 4.2 show the results of the four combinations performed. These results are performed using a forward step schedule sampling, as it was originally implemented RVOS [6], and obtained on the KITTI-MOTS benchmark. It can be seen how the best sMOTSA scores are obtained with the two chosen configurations.

Table 4.1: Image resolution variations, averaged per pixels.

Image resolution	Aspect ratio	Batch size	Length clip	sMOTSA	MOTSP	Recall	Precision
256x448	YouTube-VOS	4	5	-2,80	76,30	42,20	55,80
412x723	YouTube-VOS	2	3	-56,70	76,90	50,20	35,10
178x590	KITTI-MOTS	4	5	-31,90	74,30	48,90	42,60
287x950	KITTI-MOTS	2	3	-18,10	71,70	39,00	46,10

Table 4.2: Image resolution variations, averaged per sequence.

Image resolution	Aspect ratio	Batch size	Length clip	sMOTSA	MOTSP	Recall	Precision
256x448	YouTube-VOS	4	5	-6,83	68,12	37,38	49,70
412x723	YouTube-VOS	2	3	-46,24	76,00	44,75	37,84
178x590	KITTI-MOTS	4	5	-27,30	73,53	47,97	41,23
287x950	KITTI-MOTS	2	3	-11,70	75,68	46,47	47,63

On the tables presenting the results for each technique on the following sections, the baseline model defined for each training set can be found highlighted in grey. The results which surpass the performance of the baseline are highlighted in bold. For all the cases, the higher the value of the metric, the better. All experiments are trained during a total number of 40 epochs.

4.3 Schedule Sampling

On this section, the results of the experiments with the schedule sampling technique are presented, implemented as explained on Section 3.3.1.

4.3.1 KITTI-MOTS Benchmark

The two sets of experiments, performed with their correspondent training parameters on the KITTI-MOTS benchmark and evaluated computing the metrics with the two previous explained strategies, are shown on the following tables.

Table 4.4 and Table 4.5, show the results of the first set of experiments. This set has been trained with the parameters specified on Table 4.3. Evaluating the results, it can be observed how the results differ from one table to the other. On the pixel level evaluation, it can be seen that none of the other models outperform the baseline model. Instead, when averaging over the sequences, better performance is obtained for all the proposed models, excluding the teacher forcing approach.

Focusing on Table 4.5, the model with the best performance is the inverse step approach, followed by the forward linear schedule sampling. With a forward schedule sampling strategy, the linear approach outperforms the step approach. Instead, when using an inverse strategy, the step approach increases the overall performance of the sequences more than the linear approach.

Table 4.3: Training parameters for the experiments on Table 4.4 and Table 4.5.

Epochs	Resolution	Batch size	Length clip
40	256x448	4	5

Table 4.4: Quantitative results on the schedule sampling strategies averaged per pixels.

	sMOTSA	MOTSP	Recall	Precision
Teacher Forcing	-21,50	73,80	33,90	43,00
Forward Step	-2,80	76,30	42,20	55,80
Inverse Step	-5,30	75,20	**42,90**	54,20
Forward Linear	-4,80	75,50	41,20	54,40
Inverse Linear	-11,40	74,30	**49,80**	51,60

Table 4.5: Quantitative results on the schedule sampling strategies averaged per sequence.

	sMOTSA	MOTSP	Recall	Precision
Teacher Forcing	-16,57	**73,98**	32,81	43,62
Forward Step	-6,83	68,12	37,38	49,70
Inverse Step	**-1,57**	**73,17**	**42,79**	**55,00**
Forward Linear	**-2,29**	**72,97**	**41,00**	**53,64**
Inverse Linear	**-4,77**	**73,35**	**48,60**	**53,06**

To understand more these results, a qualitative analysis has been performed. First, the focus will be made on the forward strategies where it has been seen in the quantitative analysis that, with the image resolution of 256x448,the linear decay outperforms

the step decay.

Though the two schemes have negative scores, after analysing the images it has been observed that the linear decay adapts less to the changes of appearance of the objects, contrary to what one would think when observing the quantitative analysis. Using this technique, instances disappear faster than when using the step decay. Instead, with the step decay, as instances do not disappear, they offer more errors and thus more false positives appear which greatly influence on the computation of the score metrics. For this reason, quantitatively, this model performs worse than the linear decay. This phenomenon can be observed in Figure 4.2. Focusing on the green mask, it can be seen how it generates much more error on the baseline model. Using a forward linear strategy, this mask disappears quicker. For this same scheme, the blue and pink masks are already lost meanwhile, when using the forward step, the blue mask is still detected.

(a) Ground-truth annotation

(b) Forward Step (baseline model)

(c) Forward Linear

Figure 4.2: Qualitative results on non-consecutive frames for the schedule sampling strategies with an image resolution = 256x448, batch size = 4 and length clip = 5. Special focus is made on the error that generated the green instance for the baseline model. The effect of instances that disappear quickly can also be observed for the blue and pink instances.

For the inverse strategies, each of them has been compared to the baseline (forward step). Figure 4.3 shows the improvement of the inverse step versus the forward step scheme. The inverse step has learned better the appearance of the cars compared to the baseline. Horizontal changes of appearance are the ones in which the camera turns to the right/left and the position of the cars changes relative to the camera on the

horizontal axis. Vertical changes are the ones that appear when the car goes through a straight road and it gets closer to the cars on each side of the road or to the cars in front, with a change in depth. While horizontal changes are a common problem on all the models, independently from the schedule sampling scheme, the inverse step strategy shows improvements with vertical changes of appearance. In Figure 4.3c, it can be seen how the inverse step model has not lost the green and turquoise masks on the first and second frames, compared with Figure 4.3b.

(a) Ground-truth annotation

(b) Forward Step (baseline model)

(c) Inverse Step

Figure 4.3: Qualitative results on non-consecutive frames for the step strategies with an image resolution = 256x448. The baseline model losses faster instances like the green of the turquoise segmented instances.

When comparing the inverse linear strategy with the baseline model, the improvement of performance is subtle. In this case, it is observed that some instances are defined much better with the inverse strategy meanwhile some other instances get greatly confused with vertical close by objects. Figure 4.4 shows the performance of the inverse linear strategy on the same piece of the sequence evaluated in Figure 4.2. In this case, neither of the blue nor pink masks disappears and the green mask is more defined, producing less error. The model seems to adapt more to the changes in the objects.

(a) Forward Step (baseline model)

(b) Inverse Linear

Figure 4.4: Qualitative results on non-consecutive frames comparing the inverse linear strategy with the baseline. Improvement on the segmentation of the green masked car as well as the non disappearance of the pink mask demonstrate a better performance.

Even though this segmentation shows great improvement compared with the baseline model, the gain in adaptation of the model also produces important errors. Figure 4.5 shows how the blue instance, due to perspective occludes a vertical pole during some consecutive frames (left image). When this instance finally separates from the pole (right image), the model has assimilated the two of them as the same instance and false positives are generated. Due to the balance between the improvement on assimilating the changes of appearance and the mistakes due to this adaptation, this model improves the performance over models that also generate false positives but still perform worse than models such as the forward linear strategy where the instances disappear quickly.

Figure 4.5: Error on the inverse linear scheme due to the adaptation of changes in appearance of objects.

Table 4.7 and Table 4.8 show the results obtained for the second set of experiments with an image resolution that maintains the KITTI-MOTS aspect-ratio, with training parameters specified in Table 4.6.

In this set of experiments, the same behaviour as in the results obtained for the first set of experiments is observed. Focusing on the forward strategies, the linear decay outperforms the step decay. Instead, when focusing on the inverse strategies, the step scheme outperforms the linear. However, compared to the previous set of experiments, in this set, the results obtained with the inverse strategies provide greater

improvement of the performance. A score of **8,90** for the sMOTSA can be observed for the model with the best performance, the inverse step approach.

Table 4.6: Training parameters for the experiments on Table 4.7 and Table 4.8.

Epochs	Resolution	Batch size	Length clip
40	287x950	2	3

Table 4.7: Quantitative results on the schedule sampling strategies averaged per pixel.

	sMOTSA	MOTSP	Recall	Precision
Teacher Forcing	**-0,10**	**77,70**	**46,10**	**56,40**
Forward Step	-18,10	71,70	39,00	46,10
Inverse Step	**8,60**	**78,90**	**47,20**	**63,00**
Forward Linear	**-8,90**	**77,20**	**47,20**	**51,50**
Inverse Linear	**2,10**	**77,80**	**50,30**	**58,20**

Table 4.8: Quantitative results on the schedule sampling strategies averaged per sequence.

	sMOTSA	MOTSP	Recall	Precision
Teacher Forcing	**4,24**	**77,00**	45,84	**57,87**
Forward Step	-11,70	75,68	46,47	47,63
Inverse Step	**8,90**	**77,90**	42,86	**60,33**
Forward Linear	**-5,58**	**76,76**	**46,72**	**51,53**
Inverse Linear	**2,48**	**77,87**	**47,12**	**57,07**

As said before, comparing the forward strategies, the linear decay outperforms the step decay. On most of the sequences, their performance is similar. Even so, in the sequences where the linear decay outperforms the step scheme, the improvement is notable. Figure 4.6 shows the sequence which improves the most. It can be seen how the step scheme drags some error from a previous instance (mask coloured in green on top of the blue segmented car). This error produces false positives and reduces the true positives when coming across another instance, which is going to be segmented erroneously (see the third frame). Also, the orange car is perceived as if it was part of the car segmented in red, error which appears on the first frame and is dragged when the instance gets more defined (second frame). The forward linear approach does not make these errors.

(a) Ground-truth annotations

(b) Forward Step (baseline model)

(c) Forward Linear

Figure 4.6: Qualitative results on non-consecutive frames for the forward strategies with an image resolution = 287x950. On the baseline model, instances get mixed and error from previous instances (mask in green) is dragged through the sequence.

When focusing on the inverse strategies, the inverse step shows impressive performance results. This strategy, with an image resolution of 287x950, batch size of 2 and length clip of 3 outperforms all the models and shows improvements in almost all of the sequences (see Figure 4.7).

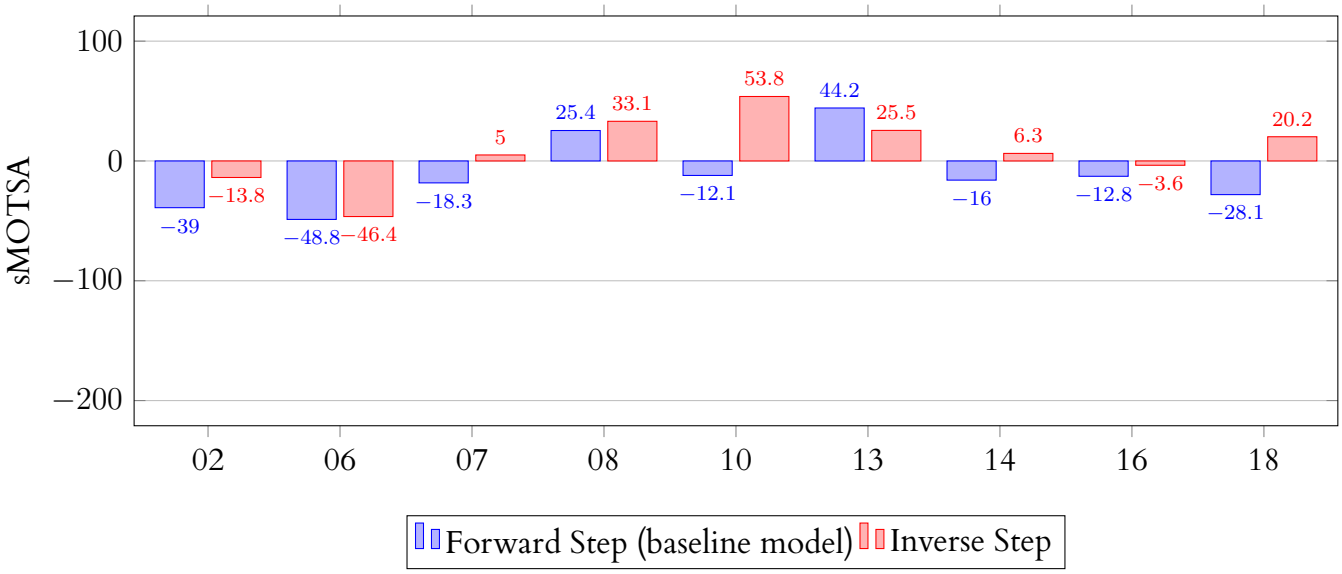


Figure 4.7: sMOTSA per sequence for the forward step (baseline) and the inverse step strategies.

The improvements of the inverse step over the forward strategy (baseline) are shown in Figure 4.8. The errors on the baseline model can be seen on the pink instance. On the second frame, it is not well defined, producing false positives between the separation of the cars. Moreover, in the third frame, the pink instance gets confused for the blue instance. When using the inverse step, the pink mask and the other masks are well defined and do not spread error.

(a) Ground-truth annotations

(b) Forward Step (baseline model)

(c) Inverse Step

Figure 4.8: Qualitative results on non-consecutive frames for the step strategies with an image resolution = 287x950. The pink instance, which latter is assimilated by the blue mask, produces significant errors on the baseline model.

The teacher forcing approach, in this set of experiments has also outperformed the baseline model. Still, this technique does not surpass the score obtained with the inverse step approach. It seems that the configuration of this set of experiments benefits greatly from fine-tuning at the end of training with the ground-truth annotations.

Finally, the results obtained with the inverse linear strategy also offer improvements compared with the forward schemes. This can be seen in fragments of scenes such as the one depicted in Figure 4.9. Both the forward step and forward linear approaches have negative performance scores. These two strategies produce a large number of false positives. For the baseline model (Figure 4.9b), it can be seen how the first frame is correctly segmented but during the second frame, the error between the green and red instance starts spreading. On the last frame, both instances are identified as the same one. The forward linear (Figure 4.9c) in this case, due to the reflection of the road and the size of the instance, it gets confused and identifies the road as if it is the green instance. Contrary to all of that, the inverse linear scheme (Figure 4.9d) segments almost perfectly all masks.

(a) Ground-truth annotations

(b) Forward Step (baseline model)

(c) Forward Linear

(d) Inverse Linear

Figure 4.9: Qualitative results on non-consecutive frames for the inverse linear strategy compared with the forward strategies with an image resolution = 287x950. False positives are generated with either of the forward strategies.

4.3.2 YouTube-VOS Benchmark

The four training variations have also been tested on the YouTube-VOS benchmark, in order to evaluate if the different strategies depend on the dataset or if a more general conclusion can be extracted. The training parameters that have been used are the same defined originally for RVOS [6].

On Table 4.9, it is seen that the best performance is obtained with the baseline model. Neither the inverse strategies nor the forward linear strategy improve the performance of the model. Even so, it can be seen that the second best performance is obtained with the inverse step, obtaining the same pondering for the category of J *unseen*.

Table 4.9: Quantitative results on the schedule sampling strategies for the YouTube-VOS dataset.

Epochs	Resolution	Batch size	Length clip
40	256x448	4	5

	Overall	J seen	J unseen	F seen	F unseen
Forward Step	0,566	0,629	0,451	0,673	0,514
Inverse Step	0,557	0,622	0,451	0,655	0,499
Forward Linear	0,545	0,601	0,442	0,639	0,499
Inverse Linear	0,551	0,625	0,430	0,660	0,490

4.4 Frame Skipping

This section introduces the quantitative and qualitative results for the models trained by using frame skipping strategies.

4.4.1 KITTI-MOTS Benchmark

Below, the results of the experiments implementing frame skipping strategies performed on the KITTI-MOTS dataset are shown.

The baseline model used for evaluating the performance implements a forward step schedule sampling, as the original RVOS. In a forward step strategy, two parts of the training process can be differentiated. On all the result's tables of this section, it is specified in which part of the training process a skipping scheme is applied and which scheme it is. The study of two skipping schemes on two training implementations, previously explained in Section 3.3.2, has been conducted.

Table 4.11 and Table 4.12 show the results for the first set of experiments, with the training parameters specified in Table 4.10. It can be seen that, when implementing a skipping strategy in each one of the training phases, it does not improve the performance of the model. In the second phase of training, the difficulty is being increased by using the model's outputs instead of the ground-truth annotations as the input of the next step and also by applying the frame skipping strategy. By combining the two strategies on the last part, the model benefits of neither of them.

Instead, taking focus on the second training implementation where the frame skipping scheme is only applied when using the ground-truth annotations, the model benefits from using this technique. It improves its performance considerably with either of the frame skipping schemes. The second phase of training then fine-tunes the model.

These results agree on the two evaluation methodologies, the official evaluation and the averaged per sequence evaluation.

In this set of experiments, skipping from 1 frame to 5 frames outperforms the other skipping scheme. This demonstrates that the model benefits from seeing more changes as the sequences on the KITTI-MOTS dataset have a slow motion and, at the same time, the training time per skipping step is an important factor in order to avoid instability. If the changes occur too fast, the model does not have time to adjust to them.

Table 4.10: Training parameters for the experiments on Table 4.11 and Table 4.12.

Epochs	Resolution	Batch size	Length clip	Sampling
40	256x448	4	5	FSSS

Table 4.11: Quantitative results on the frame skipping strategies averaged per pixel.

	Skip @ GT	Skip @ Pred.	sMOTSA	MOTSP	Recall	Precision
No skip	No	No	-2,80	76,30	42,20	55,80
From 0 to 9	Yes	Yes	-31,90	58,30	1,00	3,10
From 1 to 5	Yes	Yes	-51,90	70,60	27,40	28,70
From 0 to 9	Yes	No	**2,30**	75,50	**51,90**	**59,40**
From 1 to 5	Yes	No	**4,90**	**79,10**	41,00	**60,50**

Table 4.12: Quantitative results on the frame skipping strategies averaged per sequence.

	Skip @ GT	Skip @ Pred.	sMOTSA	MOTSP	Recall	Precision
No skip	No	No	-6,83	68,12	37,38	49,70
From 0 to 9	Yes	Yes	-39,39	58,30	1,57	3,33
From 1 to 5	Yes	Yes	-43,44	**70,43**	27,16	32,06
From 0 to 9	Yes	No	**-0,87**	**74,73**	**49,43**	**55,49**
From 1 to 5	Yes	No	**0,51**	**79,10**	**39,26**	**53,57**

Qualitative results for the best schemes in Table 4.12 are shown in Figure 4.10. In this case, the three images are non-consecutive frames that belong to a part of a turn scene of a sequence. It can be seen that when applying any of the two frame skipping schemes only when training with ground-truth annotations improve the quality of the segmentation. On both Figure 4.10c and 4.10d, the false positives of the green instance disappear.

(a) Ground-truth annotation

(b) Baseline model

(c) Frame skipping from 0 to 9

(d) Frame skipping from 1 to 5

Figure 4.10: Qualitative results of a turning scene with frame skipping schemes for an image resolution = 256x448, batch size = 4 and length clip = 5. The baseline model does not adapt well to the changes in position of the green segmented car, spreading the mask. The frame skipping schemes solve this problem.

The second set of experiments, which maitains the KITTI-MOTS aspect-ratio, has been trained with the parameters specified on Table 4.13. Observing the results on Tables 4.14 and 4.15, the model also benefits from using either of the frame skipping frames during the first half of training. The same behaviour as on the previous set of experiments is observed. On the per sequence evaluation, the frame skipping scheme from 1 to 5 still outperforms the other strategies.

Table 4.13: Training parameters for the experiments on Table 4.14 and Table 4.15.

Epochs	Resolution	Batch size	Length clip	Sampling
40	287x950	2	3	FSSS

Table 4.14: Quantitative results on the frame skipping strategies averaged per pixels.

	Skip @ GT	Skip @ Pred.	sMOTSA	MOTSP	Recall	Precision
No skip	No	No	-18,10	71,70	39,00	46,10
From 0 to 9	Yes	Yes	-26,70	**77,00**	**47,00**	43,30
From 1 to 5	Yes	Yes	-28,00	76,90	44,30	43,00
From 0 to 9	Yes	No	**-5,60**	**78,90**	**52,40**	**53,30**
From 1 to 5	Yes	No	**-12,90**	**76,90**	**52,30**	**50,80**

Table 4.15: Quantitative results on the frame skipping strategies averaged per sequence.

	Skip @ GT	Skip @ Pred.	sMOTSA	MOTSP	Recall	Precision
No skip	No	No	-11,70	75,68	46,47	47,63
From 0 to 9	Yes	Yes	-17,66	74,99	**46,70**	**50,00**
From 1 to 5	Yes	Yes	-22,87	75,20	41,77	45,99
From 0 to 9	Yes	No	**-8,18**	**76,92**	**44,67**	**48,21**
From 1 to 5	Yes	No	**-7,05**	**75,86**	**53,00**	**54,49**

Notice that, in this set, the difference between the performance of the two skipping schemes, from 0 to 9 and from 1 to 5 when applying the skipping scheme only when using the ground-truth annotations, is not very significant. This is due to the length clip parameter. Both skipping schemes were thought for a length clip value of 5. When seeing only three frames per training iteration, using a frame skipping scheme benefits but it is limited to seeing fewer appearance changes. Both schemes become more similar, with fewer changes of appearance. The model does not benefit to the fullest of the variation between the two schemes.

On Figure 4.11, the same sequence as in the previous set of experiments has been evaluated. In this case, it can also be observed improvement when using frame skipping techniques in the first phase of training. When turning, the baseline model does not segment correctly the green instance, it gets confused with the background. On the frame skipping models, this error is smaller, there are still false positives but the amount of them is considerably smaller than on the baseline model.

(a) Ground-truth annotation

(b) Baseline model

(c) Frame skipping from 0 to 9

(d) Frame skipping from 1 to 5

Figure 4.11: Qualitative results of a turning scene with frame skipping schemes for an image resolution = 287x950, batch size = 2 and length clip = 3. The baseline model does not adapt well to the changes in position of the green segmented car, spreading the mask. The frame skipping schemes solve this problem.

4.4.2 YouTube-VOS

The strategy that has obtained the best gains with the KITTI-MOTS dataset has been tested on YouTube-VOS. This strategy implemented a forward step schedule sampling and used a frame skipping scheme only when using ground-truth annotations.

For the YouTube-VOS, a different skipping scheme from the KITTI-MOTS benchmark has been used, as the video sequences are shorter. In this case, the used scheme starts without skipping any frame and, gradually, increases until 3 consecutive frames are skipped.

Table 4.16 shows the results obtained. It can be seen that the model performance does not improve. YouTube-VOS does not benefit from this strategy as most of its video sequences have faster motion compared to KITTI-MOTS. It has to be taken into account that this dataset already applies a skip-frame annotation strategy. To creat the dataset, annotations are generated every five frames. The authors believe

that the temporal correlation between five consecutive frames is sufficiently strong that annotations can be omitted for intermediate frames to reduce the annotation efforts [5].

Table 4.16: Quantitative results on the frame skipping strategies for the YouTube-VOS dataset.

Epochs	Resolution	Batch size	Length clip	Sampling
40	256x448	4	5	FSSS

	Skip @ GT	Skip @ Pred.	Overall	J seen	J unseen	F seen	F unseen
No skip	No	No	0,566	0,629	0,451	0,673	0,514
From 0 to 3	Yes	No	0,553	0,619	0,434	0,661	0,497

4.5 From temporal only to spatio-temporal

This set of experiments has explored the impact of spatial and temporal recurrence on the model's performance. The baseline model implements the forward step schedule sampling with spatio-temporal recurrence during all the training. The other experiments also implement this scheme, differentiating the two training stages where the model trains first using the ground-truth annotations and, on the second half of training, it uses its outputs. On each stage, it has been chosen which combination of temporal and spatial recurrence has been used. Under the curriculum learning context, the experiment of interest is the one which starts by training with only temporal recurrence and latter, on the second half of training, spatial recurrence is added. This way, the model starts with less information and it is increased after 20 epochs. Even though this is the relevant experiment in the context of the thesis, two more experiments, excluding the baseline, have been performed to provide a global view. The first of them trains the model only with temporal recurrence. The second one mirrors the curriculum technique, providing a reverse approach. It starts training with spatio-temporal recurrence and, after 20 epochs, the model only trains with temporal recurrence for the rest of the training process. These experiments have only been tested for the KITTI-MOTS datset.

Results on the first set of experiments have been obtained training with a compressed image resolution of 256x448, a batch size of 4 and a length clip of 5 (Table 4.17). The results are presented in Table 4.18 and Table 4.19. On both evaluations, by pixel level and by sequence level, it can be seen that the second best performance is obtained with the curriculum learning strategy where the model starts training with only temporal recurrence with scores of **-1,10** and **1.71**. Even so, the best performance is obtained when only using temporal recurrence during all training improving the performance around 4 points over the curriculum learning strategy and improving even more when compared to the baseline. When using only temporal recurrence, the improvement compared with the baseline, on the pixel level, is around 7 points and, on the sequence level, is around 14 point. The reverse strategy offers poor performance.

Table 4.17: Training parameters for the experiments on Table 4.18 and Table 4.19.

Epochs	Resolution	Batch size	Length clip	Sampling
40	256x448	4	5	FSSS

Table 4.18: Quantitative results on strategies with temporal and spatial recurrence averaged per pixels.

Ground-truth		Prediction		Metrics			
Temporal	Spatial	Temporal	Spatial	sMOTSA	MOTSP	Recall	Precision
Yes	Yes	Yes	Yes	-2.80	76,30	42,20	55,80
Yes	No	Yes	No	**5,00**	**77,70**	**46,70**	**62,00**
Yes	No	Yes	Yes	**-1,10**	**76,40**	**49,50**	**57,40**
Yes	Yes	Yes	No	-7,00	**77,50**	**43,60**	53,10

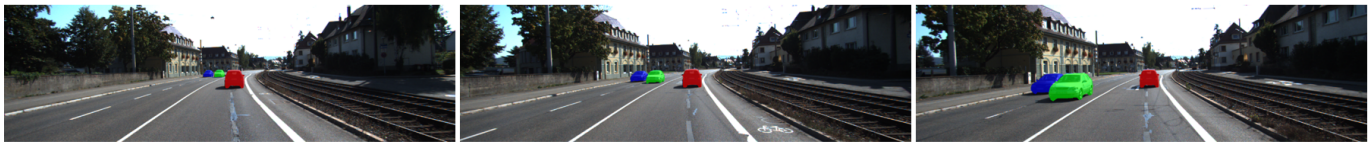
Table 4.19: Quantitative results on strategies with temporal and spatial recurrence averaged per sequence.

Ground-truth		Prediction		Metrics			
Temporal	Spatial	Temporal	Spatial	sMOTSA	MOTSP	Recall	Precision
Yes	Yes	Yes	Yes	-6.83	68,12	37,38	49,70
Yes	No	Yes	No	**6,95**	**75,72**	**46,93**	**60,16**
Yes	No	Yes	Yes	**1,71**	**74,66**	**49,10**	**59,49**
Yes	Yes	Yes	No	-10,23	**75,25**	**38,43**	48,35

This set of results demonstrates that, when working with temporal and spatial recurrence, a curriculum learning strategy does not provide the best results. The improvement over the baseline is due to the effect that spatial recurrence has over the performance instead of the idea of the planned strategy in which more information is gradually added to the model. In this set of experiments, the spatial recurrence makes the model confused. This is due to the fact that the KITTI-MOTS dataset contains a lot of small instances very close to each other. When removing this information, the model improves in all cases except for the reverse strategy. On the reverse strategy, as the model has already started training with the spatial component, the error is already present in the experiment from the start.

Figure 4.12 shows the segmentation mask predicted by the four models on a sequence with two nearby cars. It can be observed that on the models where spatial recurrence is used even once (b,d and e), the blue car gets merged with the green car. Instead, when using only temporal recurrence, even though the green mask also disappears as on the baseline model, the car which should have been segmented in green does not merge with the car next to it. For the model with a curriculum learning strategy, it can be seen how it is the model that preserves the most the segmentation of the green car. As this model uses spatial recurrence on the second half of training, the blue car ends assimilating part of the green car as in the other examples with

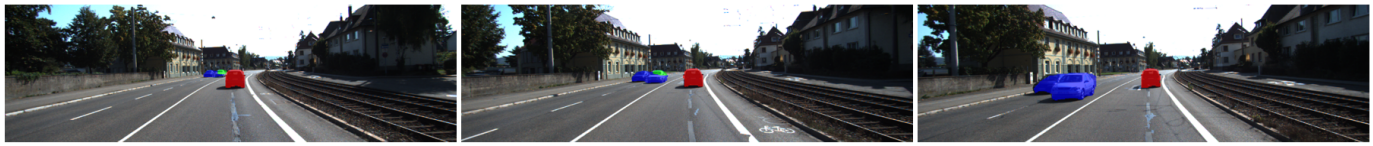
spatio-temporal recurrence. Models that use spatial recurrence have a larger number of false positives.



(a) Ground-truth annotations

(b) Baseline model

(c) Only temporal model



(d) Only temporal on the first half of training

(e) Only temporal on the second half of training

Figure 4.12: Qualitative results of nearby instances with different combinations of temporal and spatial recurrence for an image resolution = 256x448, batch size = 4 and length clip = 5. For models which use spatial recurrence, the instances close to each other (blue and green masks) merge into one.

Tables 4.21 and 4.22 show the results of the second set of experiments, performed with smaller values of batch size and length clip and with a higher image resolution which maintains the aspect-ratio of the KITTI-MOTS dataset. All the training parameters are specified on Table 4.20

For the model which uses spatio-temporal recurrence during all training (the baseline model) and for the model which uses spatio-temporal recurrence only during the

first half of training and latter changes to only using temporal recurrence, the quantitative results agree on both Table 4.21 and 4.22. While on Table 4.21 it seems that the curriculum learning strategy improves, looking at Table 4.22, the opposite can be seen. The same happens with the model which uses only temporal recurrence during all training, where on Table 4.21 the performance is far worse than the baseline and on Table 4.22 the performance is similar to the baseline.

Table 4.20: Training parameters for the experiments on Table 4.21 and Table 4.22.

Epochs	Resolution	Batch size	Length clip	Sampling
40	287x950	2	3	FSSS

Table 4.21: Quantitative results on strategies with temporal and spatial recurrence averaged per pixel.

Ground-truth		Prediction		Metrics			
Temporal	Spatial	Temporal	Spatial	sMOTSA	MOTSP	Recall	Precision
Yes	Yes	Yes	Yes	-18,10	71,70	39,00	46,10
Yes	No	Yes	No	-49,50	**77,30**	**46,10**	35,70
Yes	No	Yes	Yes	**-14,60**	**79,20**	**54,20**	**49,00**
Yes	Yes	Yes	No	-100,10	**73,30**	33,30	21,20

Table 4.22: Quantitative results on strategies with temporal and spatial recurrence averaged per sequence.

Ground-truth		Prediction		Metrics			
Temporal	Spatial	Temporal	Spatial	sMOTSA	MOTSP	Recall	Precision
Yes	Yes	Yes	Yes	-11,70	75,68	46,42	47,63
Yes	No	Yes	No	-14,62	75,28	39,34	**51,96**
Yes	No	Yes	Yes	-81,54	**77,15**	45,81	39,09
Yes	Yes	Yes	No	-84,18	74,17	31,59	24,68

To understand what is happening in this set of experiments, an analysis on the sequence level has been done. The following bar charts show the sMOTSA of the two models which differ from one table to the other.

On Figure 4.13, the sMOTSA per sequence of the model which uses only temporal recurrence during all training is represented in red. The baseline model is represented in blue. While all the results fit inside an interval of [-55,70], there is a sequence which performs very poorly respect to the others: sequence 7. This sequence is one of the sequences with more weight when computing the sMOTSA on a pixel level. It has 800 frames, being the longest of all the evaluated sequences and through all the frames, there is a large number of cars. This sequence produces the difference that can be observed on the tables. If the other sequences are observed, most of them perform similarly or outperform the baseline, providing a performance that aligns with what has been seen on the previous set of experiments with an image size of 256x448.

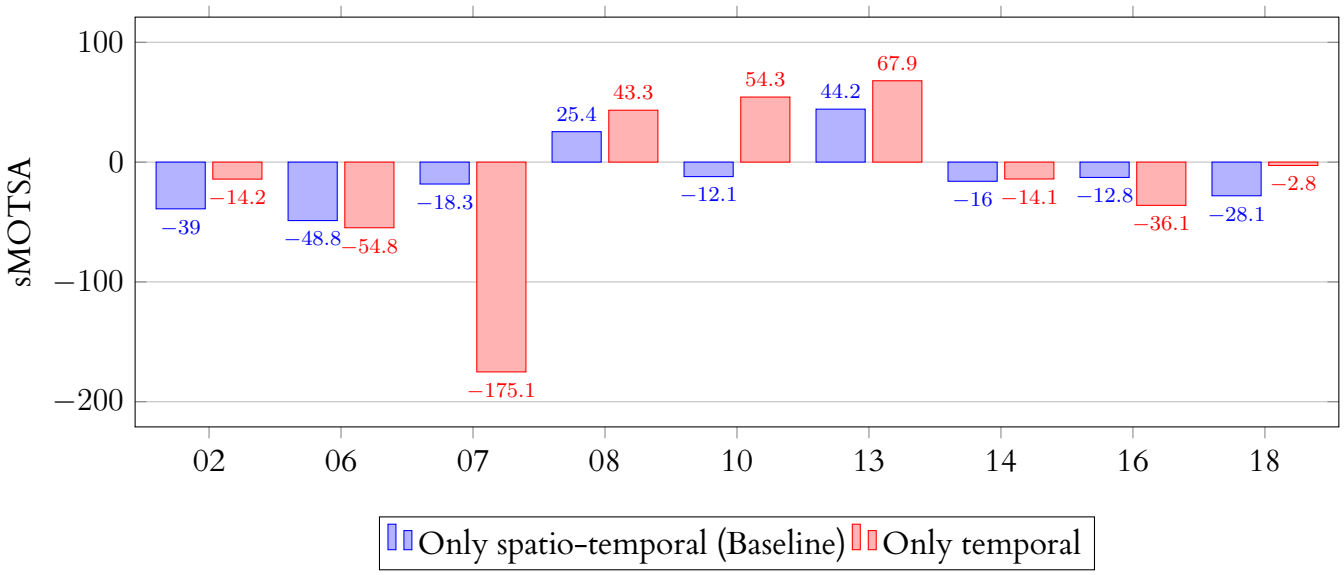


Figure 4.13: sMOTSA per sequence for the baseline model and the model that uses only temporal recurrence.

The sMOTSA per sequence for the model which trains with temporal recurrence during the first half of training and adds latter the spatial recurrence is shown in Figure 4.14. In this case, there is also a sequence that shadows the other sequences. For this model, the sequence with poor performance is sequence number 13. This sequence contains 340 frames and, even though it is not one of the shortest, it is one of the sequences which contains fewer cars. This means that it contributes less when computing the sMOTSA on a pixel level. Instead, when computing the overall averaged over sequences, it lowers considerably the score. Focusing on the other sequences, 5 out of 9 outperform the baseline.

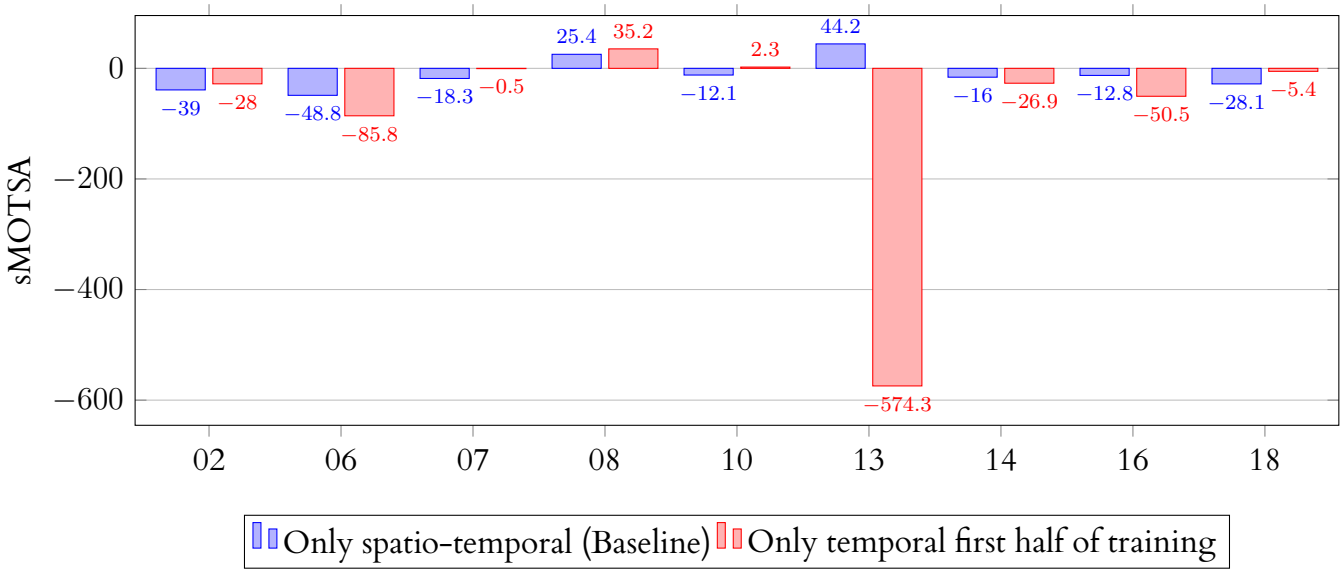


Figure 4.14: sMOTSA per sequence for the baseline model and the model that uses only temporal recurrence during the first half of training.

When analysing the same sequence as in the previous set of experiments, shown in Figure 4.17, spatial information has the same effect in both sets. In the baseline model (Figure 4.17b), the instances close to each other are grouped. Moreover, on the last frames, error from a previous object segmentation (orange) is added. Instead, the model which uses only temporal information (Figure 4.17c) does not get confused and even though the green mask is lost, it does not group together the two cars. Evaluating the results obtained with training only with temporal information during the first half of the training and adding afterwards the spatial information (Figure 4.17d), it seems as if the segmentation is perfect for the first and second frame. As in the previous set of experiments, this model is the one that preserves the most the green segmentation. Taking a closer look to the third frame, error from a previous segmented instance (pink) gets mixed with the green and blue masks. The error due to spatial information also affects this model. Lastly, the model which starts with spatio-temporal recurrence and, in the second half of training, trains with only temporal recurrence (Figure 4.17e) is the model with the worst performance. Its instability can be seen as the predictions for the three frames are confusing spots. In the middle of the erroneous segmentation and the error that drags from previous instances, it can be seen how this model started segmenting correctly the blue car but got the red and green mixed, considering both cars as the red one.

For the baseline model and the model which used only temporal recurrence on the first half of training, it has been seen how error from previous masks is dragged. Figure 4.15 shows the moment in which the red mask and the orange masks come close enough for them to be mixed in the case of the baseline model. From this point onwards, a part of the orange mask is attached to the red mask, causing errors on instances that appear latter. Moreover, in this case, the red car is already dragging error from a previously seen instance segmented in pink. The same thing happens with the pink mask for the model with only temporal recurrence on the first half of training. The moment in which the two masks come closer can be seen in Figure 4.16.

(a) First frame where the orange instance crosses paths with the red mask.

(b) Frames later, when the orange instance is already attached to the red mask.

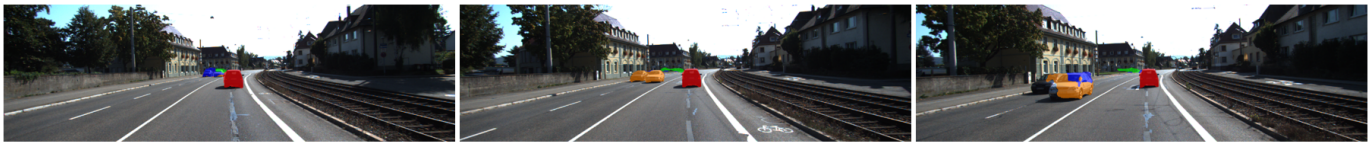
Figure 4.15: Origin of the error due to spatial recurrence shown on Figure 4.17b. Images on the right are the zoomed in version of the images in the left.

(a) First frame where the pink instance crosses paths with the red mask.

(b) Frames later, when the pink instance is already attached to the red mask.

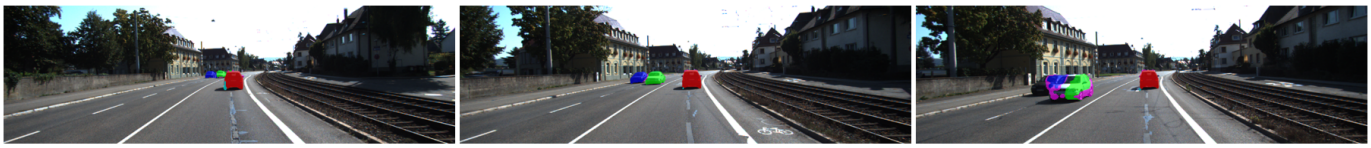
Figure 4.16: Origin of the error due to spatial recurrence shown on Figure 4.17d. Images on the right are the zoomed in version of the images in the left.

(a) Ground-truth annotations

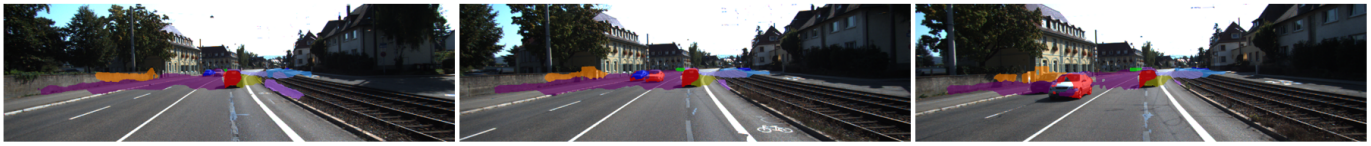


(b) Baseline model

(c) Only temporal model



(d) Only temporal on the first half of training



(e) Only temporal on the second half of training

Figure 4.17: Different combinations of temporal and spatial recurrence for an image resolution = 287x950, batch size = 2 and length clip = 3. For models which use spatial recurrence, the instances close to each other (blue and green masks) merge into one.

4.6 Loss penalization by object area

Penalizing the cost function has not improved the model's performance on either of the sets of experiments. The idea behind this implementation was to start first focusing the training on objects with a considered big and medium area and latter introduce smaller objects. This technique has only been performed on the KITTI-MOTS benchmark. The baseline model implements the forward step schedule sampling technique. Table 4.24 and Table 4.25 show the results of the first set of experiments, with an image resolution of 256x448, batch size of 4 and length clip of 5 (see Table 4.23). Table 4.27 and 4.28 correspond to the second set of experiments with an image resolution of 287x950, batch size o 2 and length clip of 3 (see Table 4.26). On both sets, the results show a great deterioration.

Table 4.23: Training parameters for the experiments on Table 4.24 and Table 4.25.

Epochs	Resolution	Batch size	Length clip	Sampling
40	256x448	4	5	FSSS

Table 4.24: Quantitative results the loss penalization strategy averaged per pixels.

sMOTSA	MOTSP	Recall	Precision
-2,80	76,30	42,20	55,80
-31,30	72,90	31,80	38,10

Table 4.25: Quantitative results the loss penalization strategy averaged per sequence.

sMOTSA	MOTSP	Recall	Precision
-6.83	68,12	37,38	49,70
-23,11	**73,24**	28,13	41,00

Table 4.26: Training parameters for the experiments on Table 4.27 and Table 4.28.

Epochs	Resolution	Batch size	Length clip	Sampling
40	287x950	2	3	FSSS

Table 4.27: Quantitative results the loss penalization strategy averaged per pixels.

sMOTSA	MOTSP	Recall	Precision
-18,10	71,70	39,00	46,10
-23,70	**75,60**	34,00	41,70

Table 4.28: Quantitative results the loss penalization strategy averaged per sequence.

sMOTSA	MOTSP	Recall	Precision
-11,70	75,68	46,42	47,63
-17,02	75,47	31,82	45,47

This approach ends worsening the performance due to the error propagation on the smallest instances. The KITTI-MOTS dataset is formed by video sequences recorded by a car driving around a mid-city. In these sequences, there are a lot of cases in which the instances first appear far away and slowly come closer to the camera. In these cases, as this approach started focusing on objects with larger areas, the smallest instances were neglected. This produces more errors on small area cars. Small area instances have more errors, some instances disappear on their following frames and these errors propagate through time. Even if the instance's area grows bigger as it comes closer to the camera, the model has already an erroneous segmentation as a base and it can not be segmented correctly. On the following figures, the error spreading can be seen for both models trained with the loss penalization approach.

Two images are presented. The first one is a shot taken from far away, where the cars are seen in the distances. The second one is a closer shot of the same cars when the camera has come closer to them. The first two figures in Fig. 4.18 are the ground-truth annotations to allow the reader to see the true segmentation of the instances.

Figure 4.18: Ground-truth annotations for a far away shot and close by shot of the same video sequence.

Figure 4.19 shows the predictions of the models with a compressed image resolution (256x448). On the figure on top, it can be seen that the model spreads the segmentation of the closest car to the cars around. On the closer shot, figure on the bottom, even though the segmented car in red is recovered, the spreading error lingers around the other three cars. Also, the pink car is confused by the blue car which appeared on the left side of the picture.

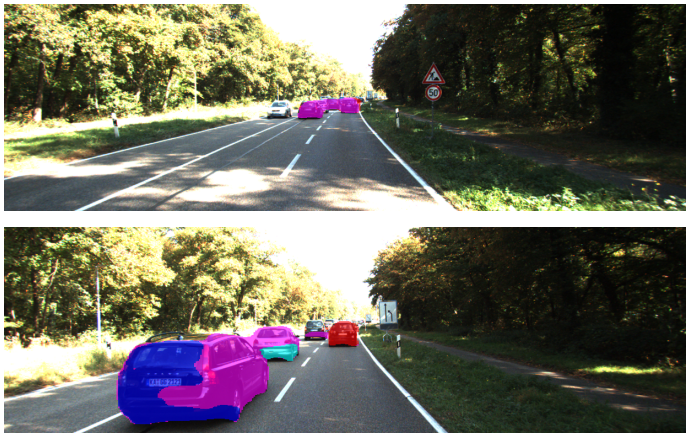


Figure 4.19: Qualitative results with an image resolution=256x448, batch size=4 and length clip=5 for a far away shot and close by shot of the same video sequence. Errors with the pink mask of the far away shot maintain when the instances get closer to the camera.

On figure 4.20, the predictions of the model with a larger image resolution (287x950) can be seen. In this case, when seen from far away, the model sees three cars as one unique instance, segmented in red. This mismatch propagates until the close by shot, where, even though the model perceives the three cars as another instance compared with the far away shot (green instance this time), it still groups the three of them. At the same time, the red and pink segmentation have mixed as if they were the same object.

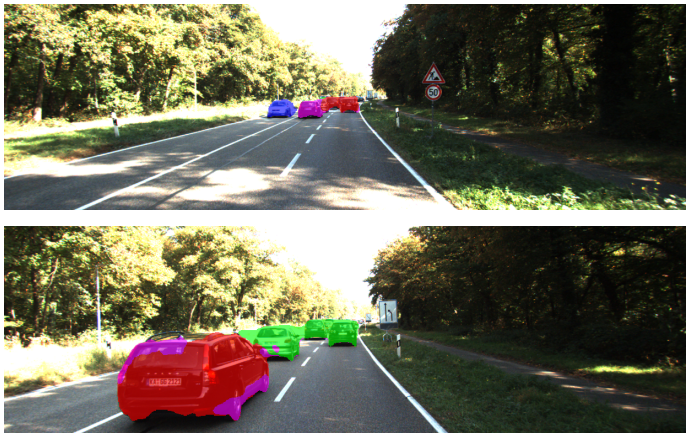


Figure 4.20: Qualitative results with an image resolution=287x950, batch size=2 and length clip=3 for a far away shot and close by shot of the same video sequence. Error with the red mask of the far away shot maintain when the instances get closer to the camera and get more defined, identifying three cars as a sole instance.

5 Conclusions

In this thesis, curriculum learning techniques have been implemented on RVOS to take on the challenge of one-shot video object segmentation for the cars' class of the KITTI-MOTS challenge. It has been demonstrated how curriculum learning affects greatly the performance of a recurrent neural network.

Focusing on the schedule sampling curriculum, surprising results have been obtained. While the forward strategies increase the performance, the inverse strategies, have improved even further the model's performance. Contrary to the reasoning behind curriculum learning, it has been seen how the model benefits greatly from the inverse step, where the difficulty of the starting point is higher than the difficulty of the ending.

For the frame skipping curriculum, significant gains have been obtained for either of the two proposed schemes, skipping from 0 to 9 frames and skipping from 1 to 5 frames. Even so, the model only benefits of this technique when using the ground-truth annotations as input of the next step. This may be due to the increment in difficulty that the model is exposed to when combining the forward step, defined as the baseline model, and the frame skipping curriculum.

The other two curriculums, from only temporal to spatio-temporal recurrence and loss penalization by object area, have demonstrated not to provide gains. For both strategies, the characteristics of the dataset were not favourable. For the curriculum which worked with temporal and spatial recurrences, it has been seen how the distance between instances and their similarity produced errors when adding the spatial information. For the loss penalization curriculum, as the first appearance of the instances is from far away, their area is small. Due to this fact, as this curriculum focused on learning better bigger instances, the error from the small instances is dragged when the instance comes closer and changes its area size.

These results demonstrate the importance of knowing well the challenges and characteristics of the dataset that is being dealt with. The KITTI-MOTS dataset is a dataset for autonomous driving. It contains a large number of partial and complete occlusions as well as many left/right turns. It has been observed how RVOS does not perform well in these cases, independently of the curriculum used. This has entailed low quantitative results compared to the state of the art of the KITTI-MOTS chal-

lenge. This fact invites to explore these curriculum learning with better performing architectures that may produce more stable and confident results.

The techniques that have improved the performance for the KITTI-MOTS dataset have been implemented on the YouTube-VOS dataset. Even so, for the YouTube-VOS dataset none of the techniques has surpassed the baseline model. The characteristics of the two datasets are very different. YouTube-VOS contains a wider variety of objects, with larger resolution and with the camera focused on them. KITTI-MOTS is a more crowded dataset, where the model has to differentiate similar and nearby instances that can have a small resolution when they first appear. Also, in this dataset new instances can appear in the video sequence at any time. These differences are the reason that the methods that work with the KITTI-MOTS benchmark do not offer the same gains with the YouTube-VOS dataset.

To sum up, interesting results have been obtained with curriculum learning strategies. This demonstrates how these techniques affect the performance of a recurrent neural network and how gains can be obtained without modifying its architecture. Even so, further research needs to be done to provide a complete understanding and characterization of the techniques.

Future work

The results obtained in this thesis invite to further explore strategies like the inverse schedule sampling and frame skipping as well as other curriculums.

For the schedule sampling curriculum, the surprising finding of the improvement in the performance of the inverse step encourage to work with this strategy for a better understanding. Other schemes combining the forward and inverse strategies such as quadratic or triangular pulses would be interesting to test.

The results for the inverse schedule sampling question whether an inverse strategy with the frame skipping curriculum would improve the results. Further investigations on this area are left open due to a lack of time. At the same time, more frame skipping schemes tested on different datasets would provide the full characterization of this technique.

For the loss penalization by the object area technique, the poor results make wonder whether the initial reasoning of the experiment is correct. It was assumed that larger instances are easier to learn for the model but this may not be the case as this dataset has a wide variety of scenes with great complexity (turns, occlusions, etc.). With the loss penalization approach, it has been observed that the error due to the small instances dragged when they come closer to the camera is important as it degrades considerably the performance. Instead, observing the model's performance, instances that start close to the camera and, as time goes by, get distanced getting smaller, are usually well segmented. This suggests that it may be interesting to change the initial approach and create a curriculum where the easy cases are nearby instances

that move away and the difficult cases are far away instances that come close to the camera. This way the difficulty is determined by the area of the first apparition of the instance instead of the area of the mask of the instances at any frame. Further research can be done on this line by letting the model determine the difficulty of the examples that it sees. By using methods such as the one introduced by Bellver et al. [38], where the model predicts the quality score of the mask for each instance, it may be interesting to let the model create its own curriculum.

Another interesting curriculum worth exploring and that has been omitted due to a lack of time is the multigrid approach, introduced by Wu et al. [24]. This curriculum is of special interest as, in this work, two sets of experiments with different image resolution and batch size have been defined. This approach would unify the obtained results, combining these two sets when training first with an image resolutions-batch size pair and, after some time, changing these parameters.

Finally, it is also left for future work the combination of the best curriculums that have been explored.

Bibliography

[1] Ning Xu et al. "Youtube-VOS: Sequence-to-sequence video object segmentation". In: *ECCV*. 2018.

[2] Sergi Caelles et al. "One-Shot Video Object Segmentation". In: *CoRR* abs/1611.05198 (2016). arXiv: 1611.05198. URL: http://arxiv.org/abs/1611.05198.

[3] Yoshua Bengio et al. "Curriculum Learning". In: *ICML*. 2019.

[4] Andreas Geiger, Philip Lenz, and Raquel Urtasun. "Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2012.

[5] Ning Xu et al. "YouTube-VOS: A Large-Scale Video Object Segmentation Benchmark". In: *CoRR* abs/1809.03327 (2018). arXiv: 1809.03327. URL: http://arxiv.org/abs/1809.03327.

[6] Carles Ventura et al. "RVOS: End-to-End Recurrent Network for Video Object Segmentation". In: *CoRR* abs/1903.05612 (2019). arXiv: 1903.05612. URL: http://arxiv.org/abs/1903.05612.

[7] C. Gong et al. "Multi-Modal Curriculum Learning for Semi-Supervised Image Classification". In: *IEEE Transactions on Image Processing* 25.7 (2016), pp. 3249–3260.

[8] Guy Hacohen and Daphna Weinshall. "On The Power of Curriculum Learning in Training Deep Networks". In: *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, 2019, pp. 2535–2544. URL: http://proceedings.mlr.press/v97/hacohen19a.html.

[9] Yangyang Shi, Martha Larson, and Catholijn M. Jonker. "Recurrent neural network language model adaptation with curriculum learning". In: *Computer Speech Language* 33.1 (2015), pp. 136–154. ISSN: 0885-2308. DOI: https://doi.org/10.1016/j.csl.2014.11.004. URL: http://www.sciencedirect.com/science/article/pii/S0885230814001211.

[10] Emmanouil Antonios Platanios et al. "Competence-based Curriculum Learning for Neural Machine Translation". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019.

[11] Vijjini Anvesh Rao, Kaveri Anuranjana, and Radhika Mamidi. "A Sentiwordnet Strategy for Curriculum Learning in Sentiment Analysis". In: *Natural Language Processing and Information Systems*. Ed. by Elisabeth Métais et al. Cham: Springer International Publishing, 2020.

[12] Jakub Sido and Miloslav Konopík. "Curriculum Learning in Sentiment Analysis". In: *Speech and Computer*. Ed. by Albert Ali Salah, Alexey Karpov, and Rodmonga Potapova. Cham: Springer International Publishing, 2019, pp. 444–450.

[13] Vik Goel, Jameson Weng, and Pascal Poupart. "Unsupervised Video Object Segmentation for Deep Reinforcement Learning". In: *CoRR* abs/1805.07780 (2018). arXiv: 1805.07780. URL: http://arxiv.org/abs/1805.07780.

[14] Samy Bengio et al. "Scheduled sampling for sequence prediction with recurrent neural networks". In: *NIPS*. 2015.

[15] Zihang Lai and Weidi Xie. "Self-supervised Learning for Video Correspondence Flow". In: *BMVC*. 2019.

[16] Mengye Ren and Richard S Zemel. "End-to-end instance segmentation with recurrent attention". In: *CVPR*. 2017.

[17] Seoung Wug Oh et al. "Video object segmentation using space-time memory networks". In: *ICCV*. 2019.

[18] Seoung Wug Oh et al. "Fast User-Guided Video Object Segmentation by Interaction-and-Propagation Networks". In: *CoRR* abs/1904.09791 (2019). arXiv: 1904.09791. URL: http://arxiv.org/abs/1904.09791.

[19] S. W. Oh et al. "Fast Video Object Segmentation by Reference-Guided Mask Propagation". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 7376–7385.

[20] A. Alabed, O. A. Nasr, and E. Hemayed. "Speeding up dominant object video segmentation". In: *2017 13th International Computer Engineering Conference (ICENCO)*. 2017, pp. 55–60.

[21] Kai Xu et al. "Spatiotemporal CNN for Video Object Segmentation". In: *CoRR* abs/1904.02363 (2019). arXiv: 1904.02363. URL: http://arxiv.org/abs/1904.02363.

[22] R. J. Williams and D. Zipser. "A Learning Algorithm for Continually Running Fully Recurrent Neural Networks". In: *Neural Computation* 1.2 (1989), pp. 270–280.

[23] Ferenc Huszár. *How (not) to Train your Generative Model: Scheduled Sampling, Likelihood, Adversary?* 2015. arXiv: 1511.05101 [stat.ML].

[24] Chao-Yuan Wu et al. "A Multigrid Method for Efficiently Training Video Models". In: *CVPR*. 2020.

[25] J. Wang, X. Wang, and W. Liu. "Weakly- and Semi-supervised Faster R-CNN with Curriculum Learning". In: *2018 24th International Conference on Pattern Recognition (ICPR)*. 2018, pp. 2416–2421.

[26] Siyang Li et al. "Multiple Instance Curriculum Learning for Weakly Supervised Object Detection". In: *British Machine Vision Conference 2017, BMVC 2017, London, UK, September 4-7, 2017*. BMVA Press, 2017.

[27] Paul Voigtlaender et al. "MOTS: Multi-Object Tracking and Segmentation". In: *CoRR* abs/1902.03604 (2019). arXiv: 1902.03604. URL: http://arxiv.org/abs/1902.03604.

[28] Tobias Glasmachers. "Limits of End-to-End Learning". In: *CoRR* abs/1704.08305 (2017). arXiv: 1704.08305. URL: http://arxiv.org/abs/1704.08305.

[29] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *CoRR* abs/1512.03385 (2015). arXiv: 1512.03385. URL: http://arxiv.org/abs/1512.03385.

[30] Xingjian Shi et al. "Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting". In: *CoRR* abs/1506.04214 (2015). arXiv: 1506.04214. URL: http://arxiv.org/abs/1506.04214.

[31] F. Perazzi et al. "A Benchmark Dataset and Evaluation Methodology for Video Object Segmentation". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 724–732.

[32] *COCO API*. https://github.com/cocodataset/cocoapi.

[33] *The KITTI Vision Benchmark Suite*. http://www.cvlibs.net/datasets/kitti/eval_mots.php.

[34] Visual Computing Institute. *Tools for evaluating and visualizing results for the Multi Object Tracking and Segmentation (MOTS) task*. https://github.com/VisualComputingInstitute/mots_tools.

[35] *MOTSChallenge 2020*. https://motchallenge.net/workshops/bmtt2020/tracking.html.

[36] Keni Bernardin and Rainer Stiefelhagen. "Evaluating multiple object tracking performance: The CLEAR MOT metrics". In: *EURASIP Journal on Image and Video Processing* 2008 (Jan. 2008). DOI: 10.1155/2008/246309.

[37] F. Perazzi et al. "A Benchmark Dataset and Evaluation Methodology for Video Object Segmentation". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 724–732.

[38] Mıriam Bellver et al. "Mask-guided sample selection for Semi-Supervised Instance Segmentation". In: *Multimedia Tools and Applications* (July 2020). DOI: 10.1007/s11042-020-09235-4. URL: http://link.springer.com/article/10.1007/s11042-020-09235-4.

[39] *Women in Computer Vision*. https://sites.google.com/view/wicvworkshop-eccv2020/.

[40] *Perception for Autonomous Driving*. https://sites.google.com/view/pad2020.

A Study of appearance changes

The KITTI-MOTS dataset offers multiple challenges in terms of occlusions, variation of the illumination or variation in the objects resolution. This thesis has made special focus on one of them: the appearance changes on slow motion sequences. This has been addressed with a frame skipping strategy.

This chapter presents the study of how the different schemes which implement frame skipping, presented on Section 3.3.2, have been chosen.

Appearance changes on objects refers to the variation that the objects on video sequences suffer. It could be that objects disappear, get deformed as the camera turns or grow bigger as the object comes closer to the camera. It may happen that these changes take a long time to occur, this is referred as slow motion sequences. It takes a high number of frames to notice the changes on objects. Due to memory constrains and depending on the resolution of the image and the used batch size, the model will be able of training with more or less consecutive frames on one iteration. This means that the model is limited to a certain number of frames to see appearance changes.

The strategy of skipping frames comes into scene in order to speed up these variations. By skipping frames, a fastest motion can be simulated. To take the most benefit of this strategy and the KITTI-MOTS dataset, the "sight" of the model has been studied. To study the "sight" of the model means to try to understand what changes the model sees on different iterations. To simulate what the models sees, the different training batches of frames have been depicted in order to see the effect of skipping a certain number of frames on appearance change. This can bee seen on Fig. A.1 and Fig. A.2 where two video sequences are studied.

The maximum number of frames per training iteration that can be passed to the model is 5. This value is considered the maximum has it affects directly to the image resolution. It is the value used on the original paper of RVOS [6]. Also, increasing it would result on lower image resolution which would affect notably to the model's performance. The study of the adaptation of the model to higher values of length clip has been left out of this project's scope as it is too wide to explore. For this reason, the images that would be passed to the model with different skipping steps have been gathered with a length clip of 5.

On the two figures, each row implements a skipping step, starting without skipping any frame (skip step of 0) and ending with skipping 9 consecutive frames (skipping step of 9). Each consecutive row increments 1 skipped frame from its previous row. It can be seen how a skip of 9 consecutive frames is enough for the model to observe notable changes on the appearance of the objects. On Fig. A.1, it is observed how the camera turns to the right. With 10 skipping steps (from 0 to 9 skipping frames) a sequence change is fully illustrated. Figure A.2 shows a similar sequence change with a left turn of the camera. To increment event more the number of skipping steps has been discarded. If the number of skipped frames is too large, the model will not be able to relate the information on one frame to the next one which will lead to poor performance. At the same time, with a larger number of skipping steps, the model will have less time per step to train which could lead to instability.

The second scheme, which consists of 5 skipping steps from skipping 1 frame until 5 frames are skipped, is motivated by the training time spent on each skipping step. This scheme tries to halve the skipping steps to double the training time per skipping step. Instability wants to be avoided and more robustness to changes wants to be gained. Analysing Fig. A.1 and A.2, on the first row, almost no change is observed between consecutive frames. This is the reason why the skipping step of 0 is omitted. A skipping step of 1 is much richer on information for training. Without skipping any frame, the model perceives five images as one, in terms of information. After starting to train with a skipping step of 1, it is increased until a skipping step of 5. This cuts in half the total number of skipping steps as well as provides enough information about changes in the model. On both figures it can be seen that with a skipping step of 5 the change on the sequence is starting to be perceived.

More schemes could be studied under the frame skipping context but it has been considered that the previous schemes illustrate well this strategy on the KITTI-MOTS dataset.

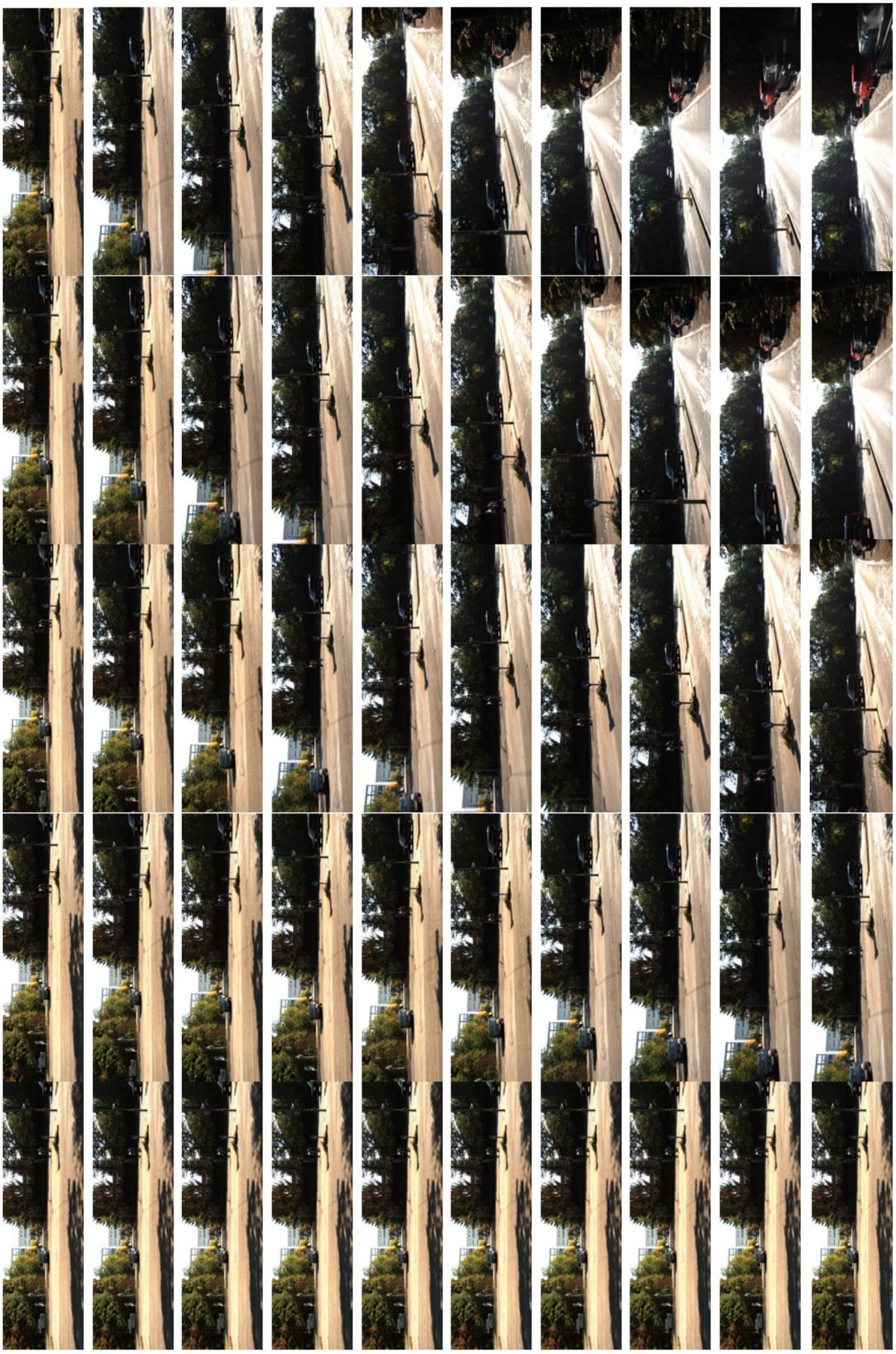


Figure A.1: Study of the variation of the objects' appearance for a right turn scene. Each row depicts a skipping step, starting without skipping any frame and increase by 1 frame the number of skipped frames until the last row is reached, where 9 consecutive frames are skipped.

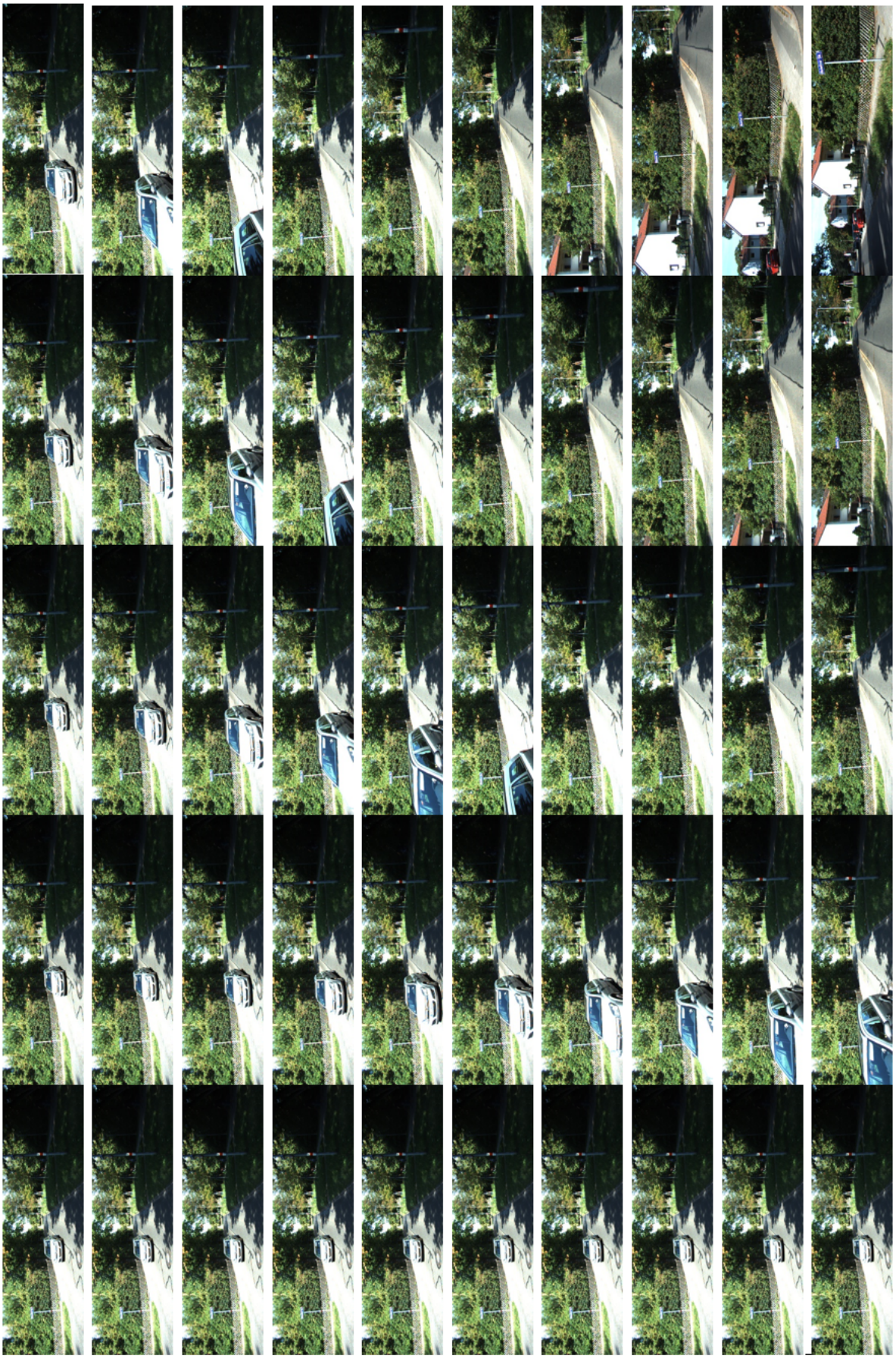


Figure A.2: Study of the variation of the objects' appearance for a left turn scene. Each row depicts a skipping step, starting without skipping any frame and increase by 1 frame the number of skipped frames until the last row is reached, where 9 consecutive frames are skipped.

B Workshop submissions

A summarized version of this master thesis has been submitted to two workshops in the European Conference on Computer Vision (ECCV), to be hold virtually:

M. Gonzalez-i-Calabuig, C. Ventura, X. Giró-i-Nieto, "Curriculum Learning for Recurrent Video Object Segmentation"

They are both under review at the time of writing this master thesis report. The description of these two workshops follow:

- **Women in Computer Vision (WiCV)** *Computer vision has become one of the largest computer science research communities. We have made tremendous progress in recent years over a wide range of areas, including object recognition, image understanding, video analysis, 3D reconstruction, etc. However, despite the expansion of our field, the percentage of female faculty members and researchers both in academia and in industry is still relatively low. As a result, many female researchers working in computer vision may feel isolated and do not have a lot of opportunities to meet with other women* [39].
- **Perception for Autonomous Driving** *Autonomous Driving (AD) has the potential to revolutionize mobility and bring lasting benefits to society. It is thus at the forefront of AI research and has attracted the attention of both academia and industry. As an example, half of the exhibitions at CVPR 2019 are related to AD. From a Computer Vision perspective, the most relevant task in AD is Perception, i.e. understanding the world around the car. After discussions with both academic researchers and industrial practitioners, we feel that the temporal and multi-modal aspects of perception have been overlooked. Robust tracking and more importantly prediction of movement, for both vehicles and pedestrians, are critical for AD. This issue is particularly acute in dense urban environments, which are heterogeneous multi-agent systems consisting of diverse traffic participants with a great variety of shapes, dynamics, behaviors, and intents* [40].

Curriculum Learning for Recurrent Video Object Segmentation

Maria Gonzalez-i-Calabuig[1], Carles Ventura[2], and Xavier Giró-i-Nieto[1]

[1] Universitat Politècnica de Catalunya
{maria.gonzalez.calabuig,xavier.giro}@upc.edu
[2] Universitat Oberta de Catalunya cvernturaroy@uoc.edu

Abstract. Video object segmentation can be understood as a sequence-to-sequence task that can benefit from the curriculum learning strategies for better and faster training of deep neural networks. This work explores different schedule sampling and frame skipping variations to significantly improve the performance of a recurrent architecture. Our results on the car class of the KITTI-MOTS challenge indicate that, surprisingly, an inverse schedule sampling is a better option than a classic forward one. Also, that a progressive skipping of frames during training is beneficial, but only when training with the ground truth masks instead of the predicted ones.

Keywords: Video Object Segmentation, Recurrent Neural Networks, Curriculum Learning

1 Introduction

The optimization process of deep neural networks is greatly influenced by how training data is used. Curriculum learning [3] is a training strategy for machine learning that consists on presenting simple concepts to the model first to, gradually, increasing their complexity.

Our work proposes two training curriculums for a Recurrent Video Object Segmentation engine (RVOS) [9], a neural model for one-shot (or semi-supervised) video object segmentation (VOS). In this task, a binary mask of an object is provided for a single frame and the goal is predicting the mask of the selected object across the rest of the frames in the video sequence. RVOS architecture is based on an end-to-end recurrent Conv-LSTM [14] decoder that tracks objects across frames, with no need of any post-processing. The recurrent architecture makes RVOS a fast solution for the task, capable of processing more than 20 frames per second [1]. RVOS was originally tested on the DAVIS and YouTube-VOS datasets for one-shot video object segmentation. We show how RVOS struggles with the *cars* in the KITTI-MOTS dataset [10], whose videos are more crowded and challenging than DAVIS or YouTube-VOS. We improve the off-the-shelf RVOS baseline by modifying its training curriculum in two ways. First, with a schedule sampling [2] totally contrary to the one original

one in RVOS and, secondly, by gradually increasing the complexity of the task by subsampling video frames at training time.

The developed source code and trained models will be published upon acceptance to facilitate reproducibility.

2 Related Work

Schedule Sampling [2] was proposed for sequence prediction with recurrent neural networks, and successfully applied in the wining bid in the MSCOCO image caption challenge 2015. It offers an alternative to *teacher forcing* [11] where, during training time, the model has access to the ground truth label of the previous time-step in each new prediction. During inference, the model uses its predictions as input in the next training step, which may produce exposure bias because of the discrepancy between training and inference. This difference may result in instability and poor model performance. Schedule sampling takes benefit from teacher forcing while avoiding exposure bias by gradually replacing the ground-truth tokens by the model's predictions. The model is forced to learn to deal with its own mistakes as it would during inference. Three different decay schedules were proposed in the original work [2]: exponential, inverse sigmoid and linear.

Related works on instance segmentation have linear schedule sampling in their trainings. Ren and Zemel [8] used a linear schedule in their recurrent instance segmentation model on still images, and Xu et al. [15] apply it for video object segmentation. Lai and Xie [5] do not go from all ground truth labels to total model predictions in their linear schedule, but consider intermediate probabilities between 0.9 and 0.6. Oh et al. [13] and RVOS [9] adopted a more drastic scheme, using ground truth labels in the first half of the training, and predicted masks in the second half. We have named this approach as a *step* schedule, as in the well-known Heaviside step function.

Frame Skipping is a training curriculum in which video sequences are progressively sub-sampled in time, so that the model is exposed to sequences with faster changes, even if synthetically generated. This technique is partially motivated by the tight constraints in terms of memory resources when training deep neural networks with video sequences. The limited sizes of the mini-batches typically force training with short sequences which, in the case of video, may be highly redundant if considering consecutive frames.

Frame Skipping was introduced in the Space-Time Memory Networks (STM) [7], inspired by a previous work on 3D reconstruction with RNNs [16] and related to their own previous model[6] trained by randomly removing frames in the training sequences. STMs achieved the state of the art on one-shot video object segmentation by training with a gradually increased amount of skipped frames, from 0 to 25. Their attention-based architecture could be trained with clips of only length 3.

Other approaches for building a curriculum by building video mini-batches may be combining different spatial-temporal resolutions that change according

to a schedule, as in *multigrid* by Wu et al. [12]. However, multigrid always considers consecutive frames. On the other hand, the same authors have also achieved relevant gains when addressing the action recognition task with a neural network that processes the video streams at a fast and a *slow* frame rates in two different pathways that merge at the deepest layer. In our work, we keep a single pathway but consider the different frame rates during the training curriculum.

3 Experiments

We have explored different schedule sampling and frame skipping strategies with the RVOS model [9] evaluated on the car class in the validation partition of the KITTI-MOTS benchmark [10]. The task addressed is the one-shot (or semi-supervised) video object segmentation (VOS) task, where a mask of the object is provided to the model to estimate the masks in the rest of the frames in the video sequence. All models are trained during a fixed amount of 40 epochs.

We adopt the official metrics for the MOTS Challenge [10] to obtain quantitative results: sMOTSA, MOTSP, Recall and Precision. In all cases, the higher the metric, the better. However, instead of averaging the metrics per pixel/frame as in the public benchmark, we have averaged them by sequence. Otherwise, the results over one very long sequence with specific challenges would dominate over the rest.

Two different strategies have been considered when allocating memory in the GPUs for training: whether we considered a lower spatial resolution (256x448 pixel) and longer clips of 5 frames, or a higher spatial resolution (287x950) at the cost of a shorter clips of 3 frames. While the 287x950 definition matches the aspect ratio of the KITTI-MOTS dataset [10], the 256x448 one corresponds to the aspect ratio of the YouTube-VOS dataset [15], for which RVOS was originally trained.

3.1 Schedule Sampling

Our experiments on schedule sampling consider the step and linear schedules in addition to the teacher forcing, provided as a baseline to compare with. The study extends to the non-conventional inverse variations for both the step and linear cases, inspired by the finding reported in [4]. The inverse variations actually defy the curriculum learning paradigm, as they start the training with the prediction of the model as references, and progress into a set up that considers only ground truth labels at the end.

The results presented in Table 1 indicate that actually the Forward Step curriculum adopted in the original RVOS baseline is the worst option, and that actually the best option is the inverse step approach. Figure 1 shows a fragment of a sequence in which the inverse step outperforms the baseline model.

Table 1. Schedule sampling variations of one-shot VOS on KITTI-MOTS *cars*. Best values are shown in **bold** and second best values in blue.

	Image resolution	Batch size	Length clip	sMOTSA	MOTSP	Recall	Precision
Teacher Forcing	256x448	4	5	-16.57	73.98	32.81	43.62
	287x950	2	3	4.24	77.00	45.84	57.87
Forward Step	256x448	4	5	-6.83	68.12	37.38	49.70
	287x950	2	3	-11.70	75.68	46.47	47.63
Forward Linear	256x448	4	5	-2.29	72.97	41.00	53.64
	287x950	2	3	-5.58	76.76	46.72	51.53
Inverse Step	256x448	4	5	-1.57	73.17	42.79	55.00
	287x950	2	3	**8.90**	**77.90**	42.86	**60.33**
Inverse Linear	256x448	4	5	-4.77	73.35	**48.60**	53.06
	287x950	2	3	2.48	77.87	47.12	57.07

Fig. 1. Qualitative results on three non-consecutive frames comparing the baseline model (row 1) and the model with best performance: inverse step (row 2). Compared to the inverse step strategy, during all the sequence, on the baseline model a wrong mask in red is observed next to the blue instance. Also, the orange mask is confused by the green mask.

3.2 Frame Skipping

Two frame skipping schemes were explored. In the *0 to 9* scheme, the number of skipped frames, which will be referred to as skipping step, is changed every 2 epochs. The total number of skipping steps is 10. The model starts training without skipping any frame and, gradually, increases the number of skipped frames by 1 until 9 consecutive frames are skipped. The second scheme, the *1 to 5* one, halves the number of skipping steps from 10 to 5. In this case, the number of skipped frames is increased after 4 epochs, doubling the training time per skipping step.

These experiments are run with the RVOS baseline mode, which follows the Forward Step schedule sampling. On the first training phase, when using the ground-truth (GT) annotations, frame skipping is always used. During the second training phase, when the model's predictions (Pred.) are used for training, we consider the two cases of skipping and non-skipping frames. We consider this hybrid approach because the difficulty of having to deal with the noisy predictions of the model, may be overwhelming for our model when adding on

Table 2. Frame skipping variations of one-shot VOS on KITTI-MOTS *cars*. Best values are shown in **bold** and second best values in blue.

	Image resolution	Batch size	Length clip	Skip @ GT	Skip @ Pred.	sMOTSA	MOTSP	Recall	Precision
No skip	256x448	4	5	No	No	-6,83	68,12	37,38	49,70
	287x950	2	3	No	No	-11,70	75,68	46,47	47,63
0 to 9	256x448	4	5	Yes	Yes	-39,39	58,30	1,57	3,33
	287x950	2	3	Yes	Yes	-17,66	74,99	46,70	50,00
	256x448	4	5	Yes	No	-0,87	74,73	49,43	**55,49**
	287x950	2	3	Yes	No	-8,18	76,92	44,67	48,21
1 to 5	256x448	4	5	Yes	Yes	-43,44	70,43	27,16	32,06
	287x950	2	3	Yes	Yes	-22,87	75,20	41,77	45,99
	256x448	4	5	Yes	No	**0,51**	**79,10**	39,26	53,57
	287x950	2	3	Yes	No	-7,05	75,86	**53,00**	54,49

Fig. 2. Qualitative results on three non-consecutive frames depicting a turning scene comparing the baseline model (row 1) and the model with best performance using frame skipping: skipping from 1 to 5 (row 2). The baseline model does not adapt well to the changes in position of the red segmented car, spreading the mask.

top the temporal sub-sampling. During the second phase, when frame skipping is applied, the skipping step begins from 0 and increases to 9 again.

The results in Table 2 actually show that applying a frame skipping strategy during all training does not improve the performance of the model, maybe due to the difficulty of combining the two schemes. Instead, when using frame skipping only during the first training phase, the performance improves considerably for either set of experiments. As the sequences of KITTI-MOTS present a slow motion, the model benefits from training with this scheme. Analysing the results for both configurations, it can be seen how the best results are obtained with a frame skipping scheme of increasing from 1 to 5 skipped frames. The model benefits more when seeing changes but with enough time to process them. Figure 2 shows the improvement of performance over a fragment of a turning scene, comparing the baseline to the frame skipping strategy from 1 to 5.

4 Conclusions

This work has shown how the curriculum learning greatly affects the performance of a deep neural network trained for the task of one-shot video object segmenta-

tion. The two techniques explored, schedule sampling and frame skipping, have brought significant gains to the RVOS model. These results encourage further research for a complete understanding and characterisation of the techniques, especially in the surprising findings that an inverse step set up may result in better results. However, the low values of the quantitative results also invite to explore these curriculum learning with better performing architectures that may produce more stable and confident results.

References

1. Athar, A., Mahadevan, S., Ošep, A., Leal-Taixé, L., Leibe, B.: Stem-seg: Spatio-temporal embeddings for instance segmentation in videos. In: ECCV (2020)
2. Bengio, S., Vinyals, O., Jaitly, N., Shazeer, N.: Scheduled sampling for sequence prediction with recurrent neural networks. In: NIPS (2015)
3. Bengio, Y., Louradour, J., Collobert, R., Weston, J.: Curriculum learning. In: ICML (2019)
4. Huszár, F.: How (not) to train your generative model: Scheduled sampling, likelihood, adversary? (2015)
5. Lai, Z., Xie, W.: Self-supervised learning for video correspondence flow. In: BMVC (2019)
6. Oh, S.W., Lee, J.Y., Xu, N., Kim, S.J.: Fast user-guided video object segmentation by interaction-and-propagation networks. In: CVPR (2019)
7. Oh, S.W., Lee, J.Y., Xu, N., Kim, S.J.: Video object segmentation using space-time memory networks. In: ICCV (2019)
8. Ren, M., Zemel, R.S.: End-to-end instance segmentation with recurrent attention. In: CVPR (2017)
9. Ventura, C., Bellver, M., Girbau, A., Salvador, A., Marques, F., Giro-i Nieto, X.: Rvos: End-to-end recurrent network for video object segmentation. In: CVPR (2019)
10. Voigtlaender, P., Krause, M., Osep, A., Luiten, J., Sekar, B.B.G., Geiger, A., Leibe, B.: Mots: Multi-object tracking and segmentation. In: CVPR (2019)
11. Williams, R.J., Zipser, D.: A learning algorithm for continually running fully recurrent neural networks. Neural Computation **1**(2), 270–280 (1989)
12. Wu, C.Y., Girshick, R., He, K., Feichtenhofer, C., Krahenbuhl, P.: A multigrid method for efficiently training video models. In: CVPR (2020)
13. Wug Oh, S., Lee, J.Y., Sunkavalli, K., Joo Kim, S.: Fast video object segmentation by reference-guided mask propagation. In: CVPR (2018)
14. Xingjian, S., Chen, Z., Wang, H., Yeung, D.Y., Wong, W.K., Woo, W.c.: Convolutional lstm network: A machine learning approach for precipitation nowcasting. In: NIPS (2015)
15. Xu, N., Yang, L., Fan, Y., Yang, J., Yue, D., Liang, Y., Price, B., Cohen, S., Huang, T.: Youtube-vos: Sequence-to-sequence video object segmentation. In: ECCV (2018)
16. Yang, J., Reed, S.E., Yang, M.H., Lee, H.: Weakly-supervised disentangling with recurrent transformations for 3d view synthesis. In: NIPS (2015)