UKRAINIAN CATHOLIC UNIVERSITY

MASTER THESIS

Stable and efficient video segmentation via GAN predicting adjacent frame

Author:
Ivan ILNYTSKYI

Supervisor:
Pavel AKAPIAN

A thesis submitted in fulfillment of the requirements for the degree of Master of Science

in the

Department of Computer Sciences
Faculty of Applied Sciences

Lviv 2018

Declaration of Authorship

I, Ivan ILNYTSKYI, declare that this thesis titled, "Stable and efficient video segmentation via GAN predicting adjacent frame" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all the main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

UKRAINIAN CATHOLIC UNIVERSITY

Abstract

Faculty of Applied Sciences

Master of Science

Stable and efficient video segmentation via GAN predicting adjacent frame

by Ivan ILNYTSKYI

Analyzing video streams represents a huge problem not only in terms of accuracy and speed, but also consistency of analysis between adjacent frames as videos are consistent due to real-world nature. Jittering effect of predictions is easily noticed by human vision in video semantic segmentation tasks. But it is not usually taken into account by design of algorithms as being suited for single image recognition and lack of easy solution via classical filters. This jittering leads to quite negative human assessment of algorithms while being good at accuracy. In addition it may lead to unstable or conflicting behavior of control systems that use computer vision. We propose the methods of efficient video semantic segmentation that take into account video consistency and can be implemented without annotated video dataset. Some methods require annotated photo only dataset, other methods additionally use generative adversarial network trained on relevant video dataset with no supervision. The solution is relevant for cases when the domain does not contain large annotated video datasets, but there are available annotated photo datasets and significantly large unlabeled videos.

We show that using semantic segmentation mask of previous frame as a feature for current frame segmentation improves accuracy and consistency. We achieve best results using the network trained with features obtained from GAN and baseline segmentation network.

Acknowledgements

I would like to thank my supervisor Pavel Akapian for the tremendous efforts, proficiency and interesting ideas.
Special thanks to Orest Kupyn for the cooperation and hints.
Also, I am greatful to Intellias Ltd for providing the scholarship for master program.

Contents

List of Figures

List of Tables

List of Abbreviations

NN	**N**eural **N**etwork
CNN	**C**onvolutional **N**eural **N**etwork
ANN	**A**rtificial **N**eural **N**etwork
FCN	**F**ully **C**onvolutional **N**etwork
GAN	**G**enerative **A**dversarial **N**etwork
cGAN	**c**onditional **G**enerative **A**dversarial **N**etwork
ReLU	**Re**ctified **L**inear **U**nit
IoU	**I**ntersection **o**ver **U**nion
mIoU	**m**ean **I**ntersection **o**ver **U**nion

Dedicated to my family.

Chapter 1

Introduction

1.1 Motivation

In recent days there is more and more need for video analysis for different real-world applications like self-driving cars, augmented reality consumer products try-on on mobile phones, sports game analysis applications and etc. Semantic segmentation task is often encountered in this applications. Current state-of-the-art approaches consider using deep convolutional neural networks for segmentation. One of goals while solving this task is not only providing fast and accurate algorithm, but providing 'no jittering' effect for human perception, i.e. outputs for adjacent frames must be consistent. This contrasts to unstable results of neural networks trained in traditional way. Inconsistent in time results may irritate users or even lead to unexpected and unsafe behaviour when applied in control systems like self-driving cars. When solving computer vision tasks like object detection and keypoints estimation some easy solutions like classical signal processing filters may be applied to solve jittering problem. But this approach becomes inapplicable for semantic segmentation as it is output is spatial semantic mask with input image size, not set of point-like objects. Some previous methods tackling described problem for semantic segmentation may require frame-by-frame annotations of datasets, but this can not be achieved for some real-world applications. So there is another question arising on using unlabeled data for this task.

1.2 Goals of the master thesis

- To explore existing methods of video segmentation.
- To develop the training pipeline that produces video segmentation models with accurate and consistent predictions.
- To provide metric for video stability measurement.

1.3 Thesis structure

This work is structured the following way: Chapter 2 contains review of related publications and general overview of relevant fields; Chapter 3 provides the description of proposed methods and network architectures, datasets overview, training details and pipelines. Metrics for evaluation and results are presented in Chapter 4, conclusions - in Chapter 5 respectively.

Chapter 2

Related work and background

2.1 Artificial neural networks

Artificial neural networks is a family of computational models based on the functions and structure inspired by human brain. Main application of the NN technology is focused on understanding and solving high-dimensional and nonlinear patterns or dependencies. There are numerous examples of commercial application, for example, speech to text, face recognition and crop yield prediction.

Neural networks are usually organized in layers. Layers consist of nodes which have an activation function. The nodes between the layers are connected via links that are weighted by influence that one unit has on another. Each layer receives the output from the previous level. During the process of training NN, the weights are modified according to the patterns of input data.

Usually, the neural network includes a large number of variables and requires a big amount of training data respectively. Typically, the data consists of pairs of input and ground truth output. During training the output of model is compared to provided ground truth values. The machine uses backpropagation to update the values of the NN weights.

2.2 Convolutional neural networks

Convolutional neural networks were firstly offered by Lecun et al., 1998 as a way to modify existing fully-connected neural networks for solving the problem of image recognition. Locality and pattern repetition are important properties of images and image recognition tasks. As a reason it was taken as a base.

Locality means that distance between pixels is important information. Closer pixels usually give more information and bring more context to fragment of image. Pattern repetition is a principle based on which some concrete patterns are being found during the image recognition (lines, circles, curves, rectangles).

Pattern repetition does not depend on location on image. For example, to solve face recognition problem, it might be beneficial to search circles (eyes), ellipsis, and some other curves which correspond to nose and mouth. Previously described curves should be searched in any subsection of image, as position of face and its features might differ.

Main types of layers of CNN are Convolutional Layer, Pooling Layer and Fully-Connected Layer. Convolutional layers apply convolution operation on input tensor (multi-channel image) with kernels of fixed size. The coefficients of given kernels are free parameters in such neural network (similarly to weights in ANN). The output of the convolutional layer is a new tensor which contains the results of convolutions. They represent the similarity between the concrete part of the image and the template (kernel). This feature implements locality and pattern repetition: the same template is applied for every part of the image, locality is taken into account at the same time.

2.3 Semantic segmentation problem

Semantic segmentation divides an image into several semantically logical parts and classifies into one of given classes. Nowadays semantic segmentation is one of the main problems in computer vision. It is a high-level task which is key to scene understanding. Importance of scene understanding can be explained by the increasing number of applications. Some of those applications include virtual reality, self-driving cars (identifying lane markings, traffic signs, cars, pedestrians), facial segmentation and medical image diagnostics. Also SS problem can be considered as one of the most general problems because solution may be applied for more narrow tasks - classification or localization.

There are two main types of semantic segmentation: standard semantic segmentation (or full pixel semantic segmentation) and instance-aware semantic segmentation. Standard semantic segmentation classifies each pixel to an object class. Instance-aware semantic segmentation is an extension to the first one by identifying different objects of the same class and assigning them different entity IDs. In the next chapters we are going to mention full pixel semantic segmentation (figure 2.1) as it is the main topic.

FIGURE 2.1: Example of semantic segmentation (Silva, 2018)

Generally semantic segmentation consists of three main steps: object detection, object recognition and classification. Each of them has a big impact on efficiency and

is being researched separately. But in semantic segmentation problem they cannot be separated. For example, if a system has a low performance in terms of object detection efficiency, it probably will have an error in shape recognition and classification subsequently. Thus we need a procedure that can solve all three steps together.

A lot of parameters have an impact on performance such as extracted features, quality of dataset etc. The problem of complete datasets always existed in machine learning, but datasets with pixel-level labeling are more time-consuming comparing to other types of datasets. Also, there is still a probability of human error during labeling. As the field was growing and getting more attention from the research community, there were produced bigger and more accurate datasets.

Before deep neural networks were proposed, the most important topics were features and classification methods. The feature is considered to be additional information which is helpful to solve a specific task. Many different features were used for semantic segmentation, e.g. Scale-invariant feature transform (Lowe, 2004), Histogram of oriented gradients (Dalal and Triggs, 2005), Bag-of-visual-words (Csurka and Perronnin, 2011), SURF (Bay and Van Gool, 2006), AGAST (Mair et al., 2010), Textons (Zhu et al., 2002), FAST-ER detector (Rosten, Porter, and Drummond, 2008) and others.

The main types of methods for image segmentation were clustering (based on color, texture, intensity, and location of the pixel), histogram-based (require only one time to pass through the pixels of the image), split-and-merge methods (tree methods, which recursively split the image into sub-images and then merges the neighbors by homogeneous criterion) and others.

2.4 Convolutional neural networks for semantic segmentation

Convolutional neural networks have seen enormous improvement in performance in such fields as object detection, image classification, action recognition, etc. According to researchers and their results, efficiency of the networks was improving by increasing the network depth.

Conventionally in classification problem an input image is downsized when it goes through convolution layers and fully connected layers. The result is predicted label for input image. If fully connected layers are substituted with convolutional layers and the output is upsampled, calculation of pixel-wise output is possible. A fully convolutional network (Long, Shelhamer, and Darrell, 2014) is based on a previously described method. Classification networks such as AlexNet and VGG are redesigned into fully convolutional networks fashion by transforming fully connected layers into convolutional layers to be able to output heatmap and upsampling layers (or transposed convolution instead) to achieve heatmap size correspondent to initial segmentation image size. As FCN does not contain any fully connected layers the model is suitable for images of different sizes. Such kind of models could achieve state-of-the-art results in semantic segmentation on multiple datasets. Schematically FCN architecture is shown in the figure 2.2.

As a solution for upsampling Dumoulin and Visin, 2016 proposed to use trans-

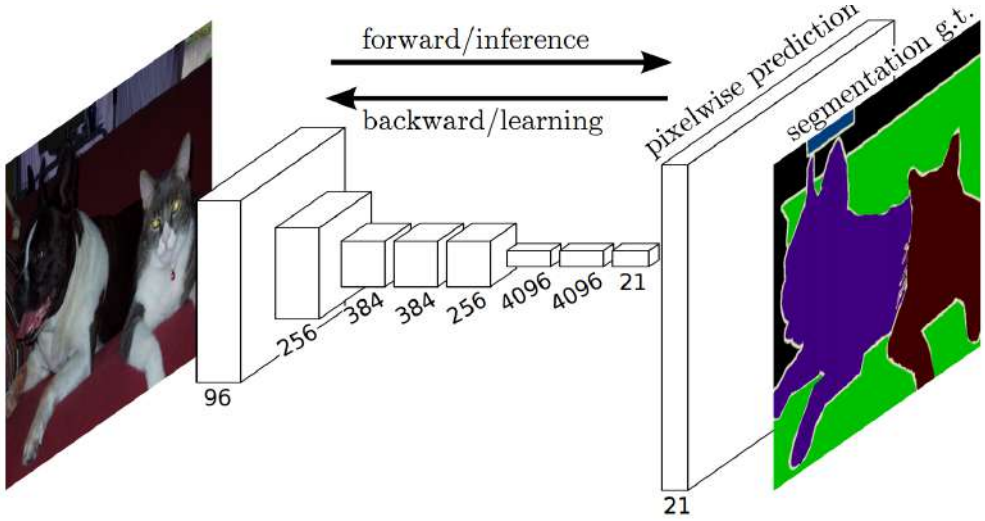

FIGURE 2.2: Dense backpropagation for fully convolutional networks (Long, Shelhamer, and Darrell, 2014)

posed convolution which works by swapping forward and backward passes of convolution (figure 2.3).

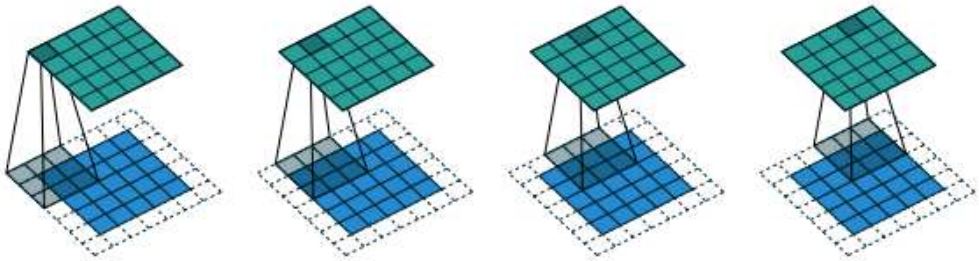

FIGURE 2.3: The transpose of convolving a 3×3 kernel over a 5×5 input using half padding and unit strides. It is equivalent to convolving a 3×3 kernel over a 5×5 input using half padding and unit strides (example provided by Dumoulin and Visin, 2016)

2.5 Generative adversarial networks

2.5.1 General overview

Generative adversarial networks are deep NN architectures offered by Goodfellow et al., 2014 which consist of two networks - generator and discriminator competing one against another. Generator network G receives latent vector (noise) from Gaussian distribution and generates an image which must be hard to distinguish from given set of real images. Discriminator D receives generated or real image and tries to distinguish the type of image. If the quality of generated images is higher then task is harder for the discriminator. It is important to mention that the generator has no access to real images data. As of that the only way to learn is to interact with the discriminator. GAN generator usually is an encoder-decoder network, where encoder maps input into low-dimensional latent space and decoder does the opposite.

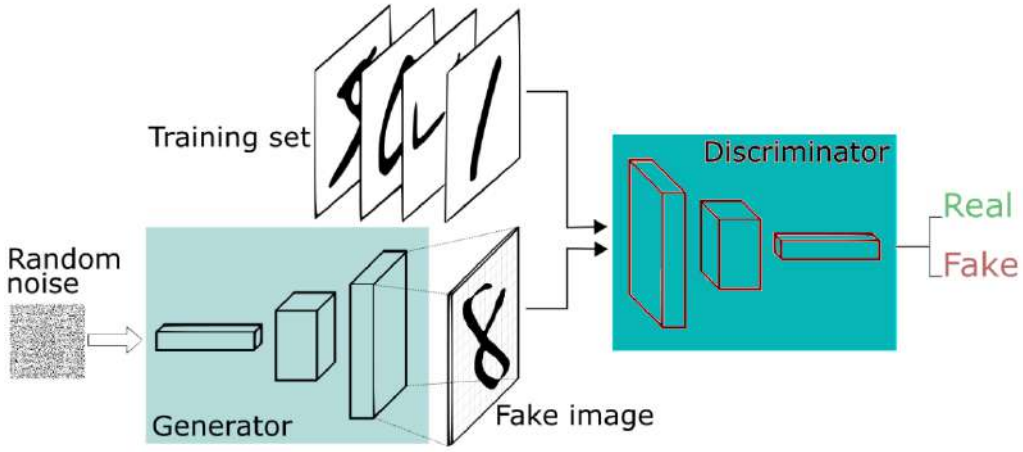

FIGURE 2.4: Pipeline of GAN training (Silva, 2017)

Game between generator G and discriminator D can be described by minimax objective:

$$\min_G \max_D \mathop{\mathbb{E}}_{x \backsim \mathbb{P}_r} [\log(D(x))] + \mathop{\mathbb{E}}_{\tilde{x} \backsim \mathbb{P}_g} [\log(1 - D(\tilde{x}))] \tag{2.1}$$

where $\mathbb{P}_r$ is the data distribution and $\mathbb{P}_g$ is the model distribution, defined by $\tilde{x} = G(z), z \backsim P(z)$, the input z is a sample from simple noise distribution.

Generator and discriminator are trained simultaneously, the parameters of generator minimize $log(1 - D(G(z))$, discriminator maximizes $log(D(X))$ respectively. Also in practice given the objective above gradients for generator training can be relatively flat because first samples are likely classified as fake during early stage of training. The solution was to change the minimization of $log(1 - D(G(z))$ to the maximization of $log(D(G(z))$. This objective function results in the same fixed point of the dynamics of G and D, but provides much stronger gradients early in learning.

In addition, if discriminator is predicting successfully generator is not learning and making progress in samples producing. Based on this discovery least squares loss was chosen to replace cross entropy loss, which penalizes fake samples by distance to real images. Comparing to traditional loss, least-square GAN (Mao et al., 2016) not only classifies samples, but also moves generated images closer to the real ones.

If training is successful then generator understands true distribution of the data and the discriminator can no longer classify images correctly. The main problems that might appear during training are non-convergence, mode collapse (generator produces limited types of samples), diminished gradient (generator gradient vanishes because of too successful discriminator), and etc.

2.5.2 Conditional generative adversarial nets

When using GAN described above there is no control over modes of data. It means GAN produces very similar images to real ones, but very versatile at the same time. Conditional generative adversarial net (Mirza and Osindero, 2014) is an extension over GAN by conditioning generator and discriminator on some extra information y which can be any kind of data. For example, it can be discrete labels, text or images. The conditioning is performed by feeding y into discriminator and generator

simultaneously. The objective function of the game between the generator and discriminator would be as following equation:

$$\min_G \max_D \mathop{\mathbb{E}}_{x \backsim \mathbb{P}_r} [\log(D(x,y))] + \mathop{\mathbb{E}}_{\tilde{x} \backsim \mathbb{P}_g} [\log(1 - D(\tilde{x},y))] \tag{2.2}$$

where $\mathbb{P}_r$ is the data distribution and $\mathbb{P}_g$ is the model distribution, defined by $\tilde{x} = G(z,y), z \backsim P(z)$, the input z is a sample from simple noise distribution.

2.5.3 Overview of pix2pix framework

Pix2pix is implementation of an image-to-image translation using conditional adversarial networks. The idea was introduced by Isola et al., 2016. The discriminator D tries to classify between pairs of the real and generated images. It is mentioned that not only generator but discriminator should be conditioned. Previously it was found beneficial to combine traditional GAN objective and traditional losses. L_1 and L_2 losses are effective, but L_1 encourages less blurring.

In this case the final objective is:

$$\mathcal{L} = \mathcal{L}_{GAN} \quad + \quad \lambda \cdot \mathcal{L}_{L_1} \tag{2.3}$$

where $\mathcal{L}_{GAN}$ is traditional GAN loss and $\mathcal{L}_{L_1}$ is L_1 loss. The proposed mix of losses does not affect discriminator, but generator has additional task to be close with ground truth in L_1 or L_2 sense.

Additional option to increase the quality of generated image was proposed by Johnson, Alahi, and Li, 2016. The authors combine benefits from per-pixels and perceptual loss functions based on the features of pretrained networks.

Given set of pairs of related images, the model understands how to convert one image into another. For example, it can be a pair of image and grey-scaled image and it is needed to represent the colors given only grey-scaled images or vice-versa.

2.6 Video segmentation

Video segmentation is a widely used technique and an extension over photo segmentation. Additional problems which appear in VS are big amount of data and inconsistency. As of this there are two main areas of improvement for procedures of video segmentation: speed and stability.

Nowadays the quality of images and videos has seen tremendous increase. The better quality media we have - the higher memory requirements become. Because of this it is needed to keep in mind amount of data that is processed. At the modern mobile phones size of photo is in range from two to five megabytes. 24-30 FPS is normal level for good video quality. It follows that the size of high-quality video with duration of one second is roughly at least 50 megabytes. Thus, real-time models should process high amount of data each second without significant loss of accuracy. Usually deep neural networks are slow during inference because of big amount of

parameters and large architecture. Paszke et al., 2016 proposed efficient neural network (ENet) which has nearly 80 times less parameters and has nearly similar or even better performance comparing to existing models.

On the other hand, video is a sequence of images which have a lot of similar information, so, it is possible to select and extract features from videos to achieve better performance. There are two main types of features to use. The first category contains features being extracted from the whole set of images, e.g. histogram features, color or features from pre-trained deep neural nets (AlexNet by Krizhevsky, Sutskever, and Hinton, 2012, VGG by Simonyan and Zisserman, 2014, etc). The second one is generally about region-based feature extraction in which features of image regions are placed into feature vector.

Previously, the main approach to solve sequence modeling was usage of recurrent neural networks. It was shown by Bai, Kolter, and Koltun, 2018 that modern convolutional neural nets outperform recurrent ones.

Frame-to-frame consistency is very important human visual perception characteristic. At the same time, there are temporal discontinuities, for example, appearance of new object in the camera view and vice versa. As we know, good quality video has at least 24-30 frames per second, which means that adjacent frames are relatively similar. Once previous frame is processed, we assume that it is beneficial to pass some features from previous frames to model. Some classical methods for video segmentation rely on frame difference to understand and identify moving objects (Leng and Dai, 2007). Thus, the main aim is to use adjacent frames features for more efficient video segmentation.

There is a lack of wide-scale annotated data for video segmentation. However there are many large-scale image datasets. Khoreva et al., 2016 suggest to use large image datasets for training to perform accurate video segmentation. They used roughly estimated previous mask by deforming ground truth masks using affine transformations as well as removing the object contour to mimic object motion in the previous frame (figure 2.5).

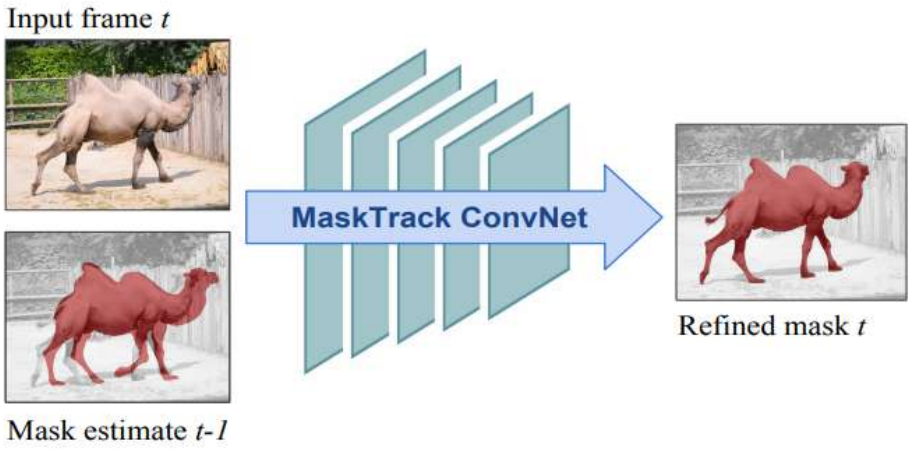

FIGURE 2.5: Training pipeline using mask estimate of previous frame as a feature (Khoreva et al., 2016)

Chapter 3

Proposed methods

3.1 Segmentation network architecture

For experiments we chose to use U-net architecture proposed by Ronneberger, Fischer, and Brox, 2015. It is very similar to encoder-decoder architecture and based on FCNs (Long, Shelhamer, and Darrell, 2014). The network is illustrated in figure 3.1. The architecture is separated in two main parts:

1. Contracting or downsampling path (left side).
2. Expansive or upsampling path (right side).

The downsampling path is similar to classical convolutional neural networks. It is composed of four blocks, each of them consists of two 3×3 convolutions followed by ReLU activation (Nair and Hinton, 2010), batch normalization (Ioffe and Szegedy, 2015) and 2×2 max pooling with stride 2.

The expansive path has four blocks in accordance to the contracting one. Each block is composed of transposed convolution layer with stride 2, concatenation with respective feature map from downsampling path (skip-connection) and two subblocks with 3×3 convolution layer, ReLU and batch normalization. And finally 1×1 convolution layer is used to map feature vector to the required number of classes.

The main advantages of U-net are:

- Computational efficiency.
- Combines location information from downsampling (encoder) and contextual information from upsampling path (decoder).

Baseline receives single input image and predicts segmentation mask.

3.2 Datasets

As we mentioned previously, visual scene understanding is main aim of semantic segmentation problem. We chose Cityscapes dataset introduced by Cordts et al., 2016) as a benchmark suite for pixel-level semantic labeling. The dataset is a large and diverse collection of video sequences recorded from cars in fifty different cities during different seasons of year. The authors showed that Cityscapes overperformed recent approaches in terms of complexity, annotation richness, scene versatility, and dataset size. Dataset is divided into train, validation and test sets based on the size of city, geographic location and time of the year in equal shares. Cityscapes contains packages with different sizes and types of data. The original size of the

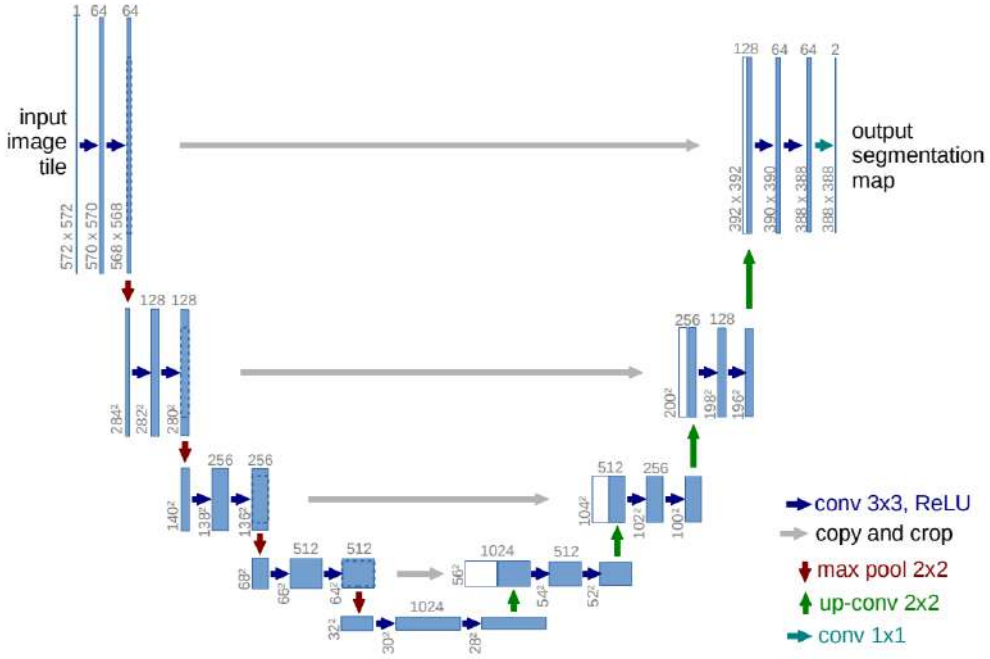

FIGURE 3.1: U-net architecture. Each blue box corresponds to multi-channel feature map. White boxes represent copied feature maps. The arrows denote the different operations. (Ronneberger, Fischer, and Brox, 2015)

images and masks is 2048×1024 which we converted to 256×256 to reduce task complexity. For experiments we required two specific datasets:

1. **Dataset for GAN.** To synthesize the dataset for GAN we used raw data from Cityscapes. It consists of video snippets 30 frames long which result in approximately 150000 images. Each video snippet is separately transformed into set of image pairs as described in the figure 3.3. As adjacent pairs are very similar we decided to shrink the dataset by taking pairs with some step to make the dataset more versatile.

2. **Dataset for segmentation network.** The dataset consists of 5000 images with fine annotations. Each image belongs to one video snippet from the raw data. The segmentation mask consists of 30 different visual classes. As some classes appear very rarely only 19 classes are considered for evaluation.

3.3 Adjacent frame generation using GAN

3.3.1 GAN architecture

To generate the adjacent frame we used pix2pix architecture. There are two ideas for the generator: encoder-decoder and U-net (figure 3.2).
We chose to use the architecture based on U-net. It contains eight encoding blocks which consist of convolution layer, instance normalization (Ulyanov, Vedaldi, and Lempitsky, 2016), LeakyReLU (Xu et al., 2015) and dropout (Srivastava et al., 2014) except first three blocks, and eight decoding blocks with sequential application of transposed convolution, instance normalization, ReLU and dropout (except last three blocks). Respective encoding and decoding blocks have a skip connection.

The discriminator consists of four blocks (convolution layer, instance normalization, and leaky ReLU), zero padding and final convolution layer. The output of discriminator is a small image where each pixel has a value between 0 and 1 and represents how trustworthy each subarea of the picture is. Such architecture was also used by Isola et al., 2016.

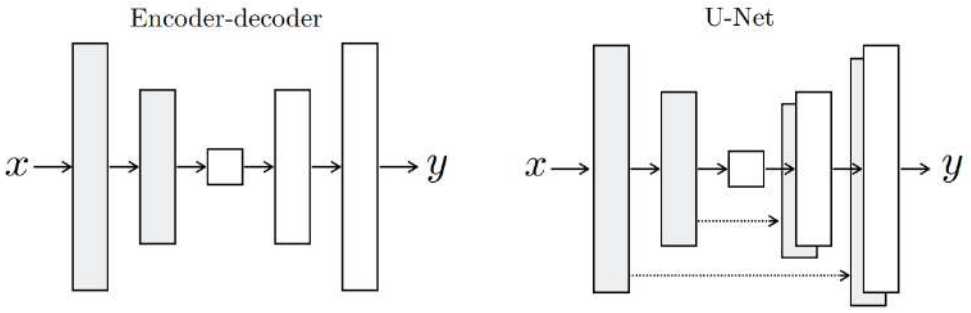

FIGURE 3.2: Choices for the generator architecture. (Isola et al., 2016)

3.3.2 Synthesizing dataset for GAN

To train GAN which was described previously, we need a set of videos or sequences of photos. Our aim is to convert the raw data into a set of pairs of images with some given frame (time) difference. The main hyperparameter of resulting dataset is sequential or time difference between the frames. The bigger hyperparameter is - the more different images in each pair are. Also, the parameter is dependent on type of original dataset (how fast the scene is changing in video). Given the sequence of photos and hyperparameter of sequential difference, we can combine photos as it is shown in the figure 3.3.

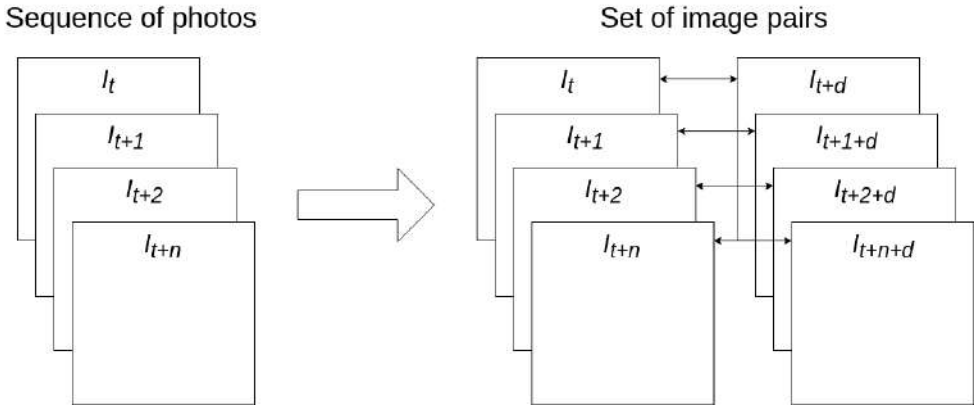

FIGURE 3.3: Transformation of the image sequence I, given the sequential difference d, which results in GAN dataset.

3.3.3 Training and results

During the training of pix2pix network we faced two different problems:

- **Best value of frame difference of the dataset.** Resulting generative adversarial network is expected to produce different image from the input. Performance

of the network highly depends on the dataset and its parameters. Frame difference d is the hyperparameter of dataset which represents relation between the frames in each pair (figure 3.3). Firstly, we trained the network using the datasets with $d = 1, 2, 3$ but the pairs contained very similar photos. As a result, the generator was not learning dynamics and semantic of the dataset. After multiple attempts, we came up with value $d = 6$ which corresponds to the dataset with pairs of images that are separated by 5 sequential images or have approximately 0.36s time difference (each video snippet contains 30 photos and has 1.8s duration).

- **Quality of generated image.** Firstly, generative adversarial network was trained using loss as the combination of classic CGAN objective and L_1 distance. Another approach was to combine traditional loss with perceptual loss from the feature maps of VGG19 pretrained on ImageNet. After multiple experiments we came up with following loss formula:

$$\mathcal{L} = \mathcal{L}_{GAN} \quad + \quad \lambda \cdot \mathcal{L}_{L_1} \tag{3.1}$$

  where λ equals to 100.

Having checked the results of validation set, we found two patterns of generated images. To keep in mind, the dataset is collected from camera attached to the moving car.

Firstly, objects which are very close to car and are on the edges of the photo will likely be passed and will not be found on the following frames. Respectively, the image, predicted by GAN, could not predict position of close objects on the right and left side of picture. In the figure 3.4, predicted images have strongly blurred or distorted objects (cars) on the edges of photo.

Secondly, we consider situation when photo contains objects which are moving across the road, orthogonally to car trajectory, for example, pedestrians crossing the road. Objects are expected to stay in the frame as camera is not moving, but the position of dynamic objects will change significantly. In this case, to produce trustworthy image, the generator substitutes objects with the background as position of objects is hard to predict. The following case is represented in the figure 3.5.

3.4 Experiments

The main difference between photo and video segmentation is connection between the adjacent frames in a video, as they are sequential. Information from the segmentation of previous frame is insightful for the following segmentation process. We propose to use segmentation models which require mix of current image and features from the previous image segmentation as an input. We came up with three feature types:

- **Previous image.** The main idea is that having information about current and previous frame, segmentation model understands the dynamics of objects by the frame difference.

- **Previous estimated segmentation mask.** As adjacent images are expected to be relatively similar, the same hypothesis is applied to segmentation masks.

FIGURE 3.4: Examples of images being produced by the generator. Objects (cars) which are close to camera and situated from the sides are expected to partially disappear from the following frames, respectively blurred or distorted by generator.

The prediction of previous image brings context and semantic understanding already gained from the past.

- **Previous estimated segmentation mask converted to RGB.** Considering the cityscapes dataset, it contains 19 major classes. Segmentation mask, produced by the network, has 19 channels respectively. To visualize the data as RGB image, each class is attributed some specific color (human - red, motorcycle - blue, nature - green, etc.). We assume that following dimensionality reduction is as efficient as original segmentation mask and the network should be faster due to smaller input size (3 channels).

As we do not have annotated video dataset, we need to synthesize the adjacent frame data. We assume that the following information can be obtained by two approaches:

- Transformation of ground truth segmentation mask.
- Generating previous or adjacent frame using generative adversarial network.

Let us consider the following methods separately.

3.4.1 Generated adjacent frame as a feature

The following proposed methods consider that we have annotated photo dataset and video dataset from same source or from same domain. Video dataset is used to

FIGURE 3.5: Cases of "lazy" work of generator. The generator substitutes dynamic objects with the background as their position changes significantly. Examples: pedestrians or cyclists crossing the road, cars on the crossroads.

train the GAN. After the network is trained only the generator network is needed. Annotated photo dataset is required for pipeline to train the segmentation network. Having generated adjacent image using GAN we implemented different approaches how to use the information for training pipeline:

- **Pure generated image.** Having observed the pattern of generated images, we decided to use generated image as additional feature. Segmentation network receives concatenation of original image and generated adjacent image as an input. The pipeline is described in the figure 3.6.

  The main assumption is that generated image is similar to previous frame if we had a video sequence.

- **Segmentation mask of generated adjacent image.** We would like to reproduce the flow when during video segmentation the input of current frame contains the current image and previous predicted mask. To produce segmentation mask, given the adjacent frame only, we applied segmentation baseline, which requires single image as input and outputs segmentation mask. Finally, predicted mask (by the baseline) is concatenated with original image. Resulting tensor, which has 22 channels, is used as input to train the network (figure 3.6).

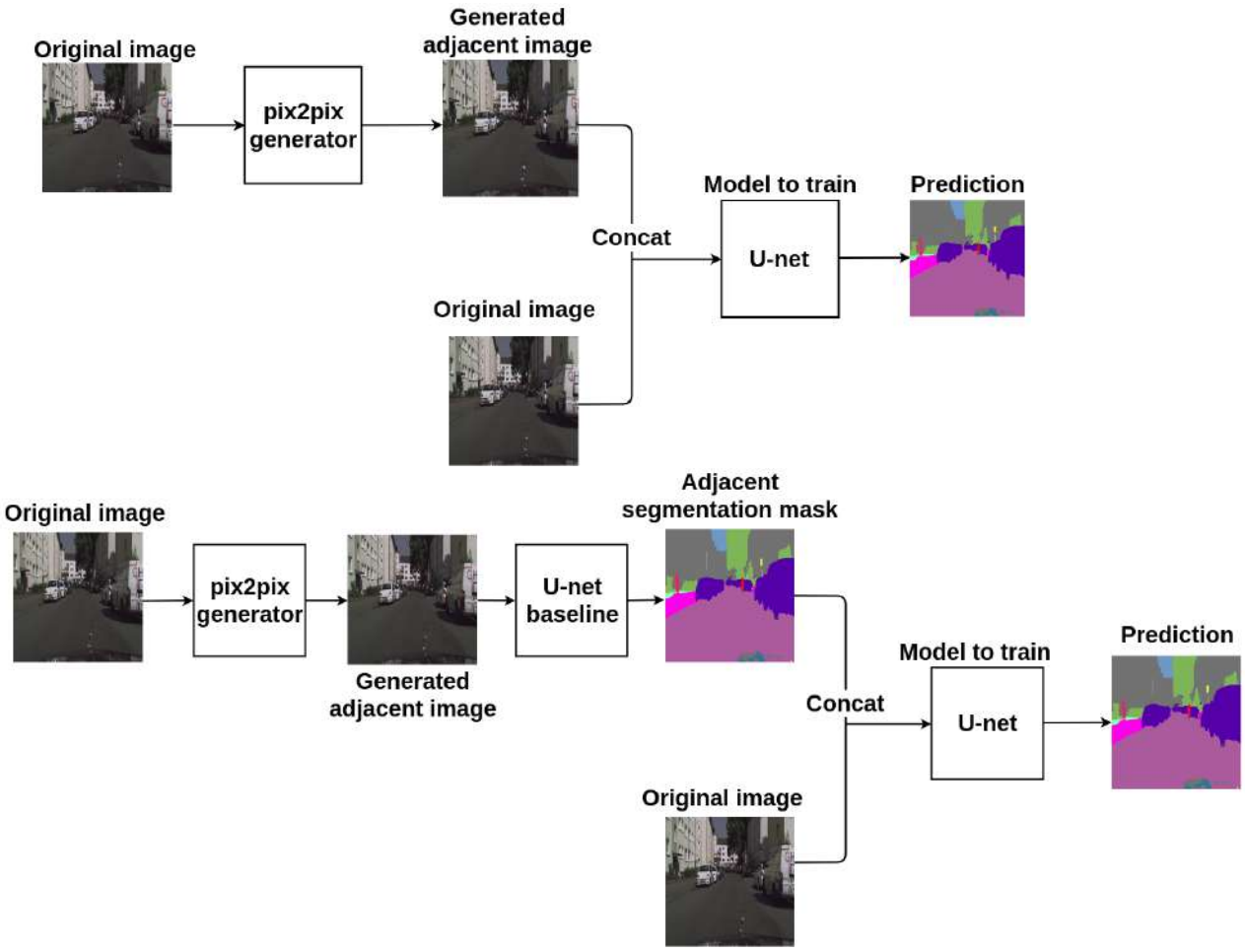

FIGURE 3.6: Training pipelines of the segmentation network using adjacent frame generator. Usage of pure generated adjacent image (top) and segmentation mask of generated image (bottom)

- **Segmentation mask of the generated adjacent image converted to RGB.** The pipeline is nearly same comparing to previously proposed approach, but before predicted mask is concatenated with original image, we convert the mask into RGB image. The number of channels of input is 6, which is significantly smaller than previous method.

3.4.2 Augmented ground truth segmentation mask as a feature

Let us consider the case when we have annotated photo dataset only. Given the segmentation mask of current frame, we assume that previous mask is very similar to current one. Thus, to estimate the segmentation mask of previous frame, we augment ground truth segmentation mask.

The segmentation mask can be represented in a classic way or we can transform it into RGB photo. Given two segmentation mask representations, we came up with two training pipelines described in the figure 3.7:

- Augmented GT segmentation mask as a feature
- Augmented GT segmentation mask converted to RGB as a feature

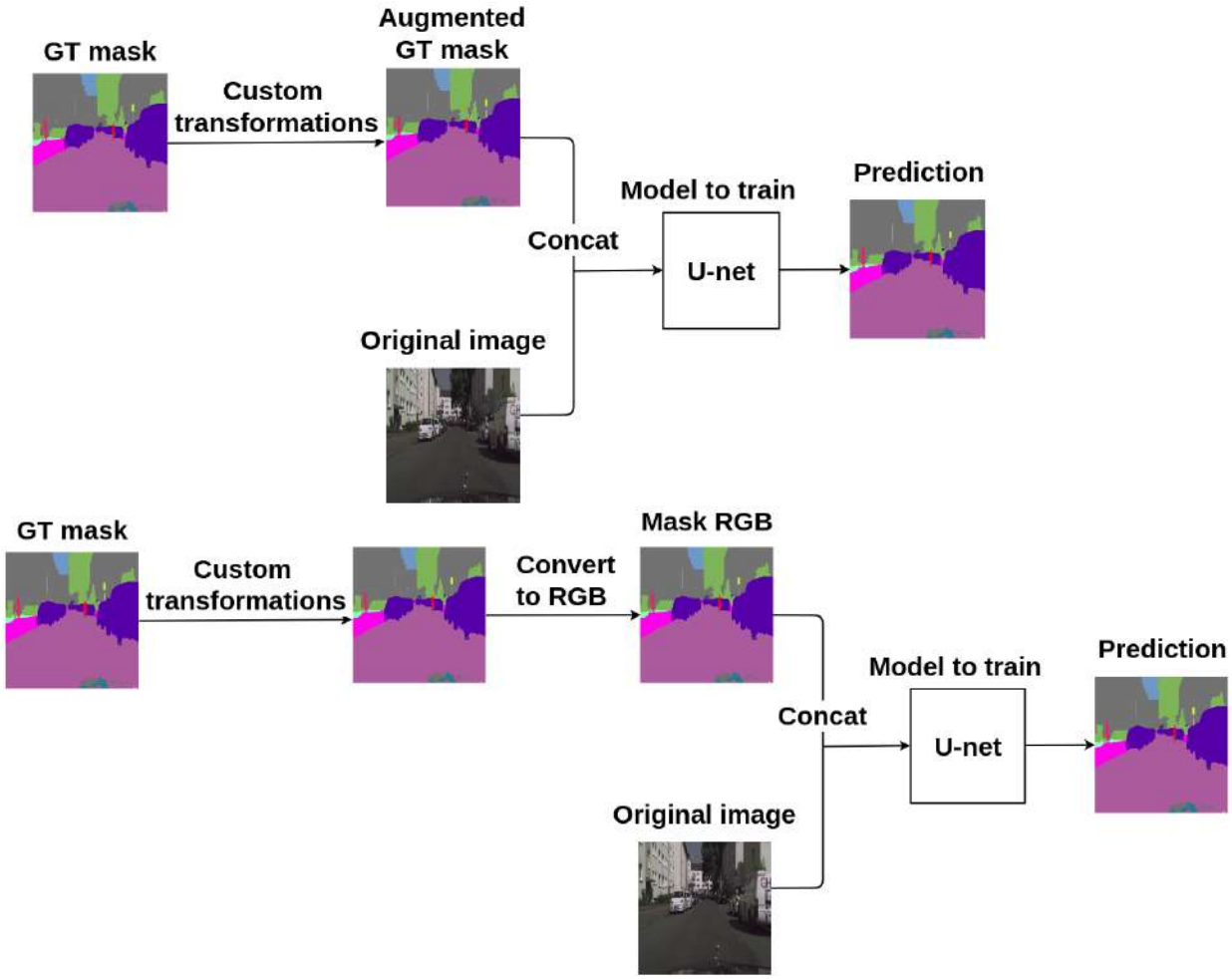

FIGURE 3.7: Training pipelines of segmentation network using transformed ground truth mask: augmented mask as a feature (top) and augmented mask converted to RGB (bottom)

The main concern of this method is an algorithm of ground truth mask transformation. To transform the ground truth segmentation mask, we used a mix of affine transformation and grid distortion. As a result of custom ground truth mask transformations, segmentation network can receive the following features as input:

- Slightly augmented GT mask ($p = 0.2$)
- Moderately augmented GT mask ($p = 0.2$)
- Strongly augmented GT mask ($p = 0.1$)
- Empty mask ($p = 0.5$)

where p is the probability of transformation and words "slightly", "moderately" and "strongly" mean the relation between hyperparameters values of affine transformation and grid distortion.

When using augmented GT mask as input, there is a probability to overfit the network as input feature is very similar to expected output. To avoid overfitting, we decided to nullify the mask randomly with probability 0.5. When input mask is nullified, segmentation network has only one source to learn: the original photo. Also,

while segmentation network processes the first frame of video, there is no information about previous mask as it doesn't exist. Thus, video segmentation network should be useful in cases when the previously predicted segmentation mask is not available.

3.4.3 Training details

All the models we trained were implemented using *PyTorch* framework. The training was performed on GeForce GTX 1080 GPU using the Cityscapes(Cordts et al., 2016) dataset. The images and segmentation masks were downsized to 256×256 to reduce complexity of the task. Optimization was performed using Adam (Kingma and Ba, 2014) as a solver by the method of error back-propagation. The models had been trained for 150 epochs with learning rate 2×10^{-5} and a batch size = 4. All of our models have U-net architecture but with different number of input channels (3,6 or 22).

Chapter 4

Metrics and evaluation

4.1 Metrics

To evaluate the performance of segmentation network we compared models using two criteria: segmentation accuracy and consistency. Ideally, video segmentation network should provide accurate and stable results, but stability is not helpful in combination with poor segmentation accuracy. Thus, we consider the network to be "better" if two criteria overperform simultaneously:

Segmentation accuracy. To measure the accuracy of segmentation network we used annotated photo dataset (validation subset). Given predicted mask and ground truth mask, we calculated mean intersection over union metric. The formula of intersection over union for a class looks the following way:

$$IoU_i = \frac{TP_i}{FP_i + FN_i + TP_i} \tag{4.1}$$

where i is a class id, TP is number of true positives FN - false negatives and FP - false positives respectively. Let us denote N as number of segmentation classes. The values of TP, FN and FP are calculated from $N \times N$ confusion matrix C where c_{ij} is amount of pixels from ground truth class i predicted as class j.

Mean intersection over union is an average of all IoUs:

$$\overline{IoU} = \frac{1}{N} \sum_{i=1}^{N} IoU_i \tag{4.2}$$

Video segmentation consistency (stability). Assuming that the adjacent frames of video sequence are similar, we expect that ideally segmentation masks of these frames are similar too. If segmentation network is unstable then, given two adjacent frames, network can recognize the same object differently as scene is changing dynamically.

To measure the consistency of predictions, we propose the metric to measure stability. Given the video sequence $\{I_1, I_2, \dots L_N\}$, segmentation network predicts the mask sequence $\{M_1, M_2, \dots M_N\}$. The main idea is that previous mask M_{k-1} is considered to be ground truth mask for the M_k. Confusion matrix C is calculated from the mask pairs $\{M_{k-1}, M_k, \}$ ($k \in \overline{2, N}$). Having obtained the confusion matrix, we calculate mean intersection over union using previously described formula.

Important to mention that the single value of stability metric does not represent the consistency itself. Metric strongly depends on the dataset type, whether scene is changing dramatically or slightly. We use stability metric for comparison only.

4.2 Results

For the evaluation, we calculated accuracy and consistency metrics of models with following additional features:

- No features (**Baseline**)
- Generated adjacent frame (**GANImg**)
- Segmentation mask of generated adjacent image (**GANMask**)
- Segmentation mask of generated adjacent image converted to RGB (**GANMaskRGB**)
- Augmented GT segmentation mask (**AugmGT**)
- Augmented GT segmentation mask converted to RGB (**AugmGTRGB**)

Results of segmentation accuracy are presented in the table 4.1. The values marked in red correspond to experiments which used ground truth mask as input. Thus, segmentation accuracy of AugmGT and AugmGTRGB is not comparable to baseline. We observe that average segmentation accuracy of experiments has seen increase comparing to baseline.

Method		Generated adjacent frame			Transformed GT mask	
Experiment	Baseline	GANImg	GANMask	GANMaskRGB	AugmGT	AugmGTRGB
Best value	50.47	**52.24**	47.64	51.68	52.36	57.04
Average value (10 last epochs)	39.11	**44.62**	40.97	44.12	42.57	43.77

TABLE 4.1: Evaluation results. Segmentation accuracy (mIoU)

Results of video segmentation consistency are presented in the table 4.2. Metric was calculated using multiple video snippets from validation subset.
We have observed the increase of consistency in all experiments comparing to baseline. The highest metric value corresponds to GANMaskRGB.

Method		Generated adjacent frame			Transformed GT mask	
Experiment	Baseline	GANImg	GANMask	GANMaskRGB	AugmGT	AugmGTRGB
mIoU	40.06	45.80	53.40	**55.49**	47.19	51.76

TABLE 4.2: Evaluation results. Video segmentation consistency

Chapter 5

Conclusion

In this work, we proposed the methods of efficient video semantic segmentation that take into account video consistency and can be implemented without annotated video dataset. We show that using semantic segmentation mask of previous frame as a feature for current frame segmentation improves accuracy and consistency.
We described two methods of adjacent mask estimation for training pipeline:

- Augmenting ground truth mask via affine transformation and non-linear grid distortion
- Predicting mask via baseline segmentation network on image produced by GAN that models adjacent in time images

Described methods overperform baseline in terms of segmentation accuracy and consistency. Also we showed that reduction of previous mask dimensionality and using compressed mask as a feature yields higher performance.
We achieved best results using the network trained with features obtained from GAN and baseline segmentation network (GANMaskRGB).

Bibliography

Bai, Shaojie, J. Zico Kolter, and Vladlen Koltun (2018). "An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling". In: *CoRR* abs/1803.01271. arXiv: 1803.01271. URL: http://arxiv.org/abs/1803.01271.

Bay Herbertand Tuytelaars, Tinne and Luc Van Gool (2006). "SURF: Speeded Up Robust Features". In: *Computer Vision – ECCV 2006*. Ed. by Aleš Leonardis, Horst Bischof, and Axel Pinz. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 404–417. ISBN: 978-3-540-33833-8.

Cordts, Marius et al. (2016). "The Cityscapes Dataset for Semantic Urban Scene Understanding". In: *CoRR* abs/1604.01685. arXiv: 1604.01685. URL: http://arxiv.org/abs/1604.01685.

Csurka, Gabriela and Florent Perronnin (2011). "Fisher Vectors: Beyond Bag-of-Visual-Words Image Representations". In: vol. 229, pp. 28–42. DOI: 10.1007/978-3-642-25382-9_2.

Dalal, Navneet and Bill Triggs (2005). "Histograms of Oriented Gradients for Human Detection". In: *In CVPR*, pp. 886–893.

Dumoulin, Vincent and Francesco Visin (2016). *A guide to convolution arithmetic for deep learning*. cite arxiv:1603.07285. URL: http://arxiv.org/abs/1603.07285.

Goodfellow, Ian J. et al. (2014). "Generative Adversarial Networks". In: URL: https://arxiv.org/abs/1406.2661 (visited on 01/08/2017).

Ioffe, Sergey and Christian Szegedy (2015). "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". In: pp. 448–456. URL: http://jmlr.org/proceedings/papers/v37/ioffe15.pdf.

Isola, Phillip et al. (2016). "Image-to-Image Translation with Conditional Adversarial Networks". In: *arxiv*.

Johnson, Justin, Alexandre Alahi, and Fei-Fei Li (2016). "Perceptual Losses for Real-Time Style Transfer and Super-Resolution". In: *CoRR* abs/1603.08155. arXiv: 1603.08155. URL: http://arxiv.org/abs/1603.08155.

Khoreva, Anna et al. (2016). "Learning Video Object Segmentation from Static Images". In: *CoRR* abs/1612.02646. arXiv: 1612.02646. URL: http://arxiv.org/abs/1612.02646.

Kingma, Diederik P. and Jimmy Ba (2014). "Adam: A Method for Stochastic Optimization." In: *CoRR* abs/1412.6980. URL: http://dblp.uni-trier.de/db/journals/corr/corr1412.html#KingmaB14.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton (2012). "ImageNet Classification with Deep Convolutional Neural Networks". In: *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*. NIPS'12. Lake Tahoe, Nevada: Curran Associates Inc., pp. 1097–1105. URL: http://dl.acm.org/citation.cfm?id=2999134.2999257.

Lecun, Yann et al. (1998). "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE*, pp. 2278–2324.

Leng, Bing and Qionghai Dai (2007). "Video Object Segmentation based on Accumulative Frame Difference". In: URL: https://pdfs.semanticscholar.org/f82c/bfc85a90d46bda72142dd78e1516823d8c39.pdf.

Long, Jonathan, Evan Shelhamer, and Trevor Darrell (2014). "Fully Convolutional Networks for Semantic Segmentation". In: *CoRR* abs/1411.4038. arXiv: 1411.4038. URL: http://arxiv.org/abs/1411.4038.

Lowe, David G. (2004). "Distinctive Image Features from Scale-Invariant Keypoints". In: *Int. J. Comput. Vision* 60.2, pp. 91–110. ISSN: 0920-5691. DOI: 10.1023/B:VISI.0000029664.99615.94. URL: https://doi.org/10.1023/B:VISI.0000029664.99615.94.

Mair, Elmar et al. (2010). "Adaptive and generic corner detection based on the accelerated segment test". English (US). In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 6312 LNCS. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) PART 2, pp. 183–196. ISBN: 3642155510. DOI: 10.1007/978-3-642-15552-9_14.

Mao, Xudong et al. (2016). *Least Squares Generative Adversarial Networks*. cite arxiv:1611.04076. URL: http://arxiv.org/abs/1611.04076.

Mirza, Mehdi and Simon Osindero (2014). "Conditional Generative Adversarial Nets". In: *CoRR* abs/1411.1784. arXiv: 1411.1784. URL: http://arxiv.org/abs/1411.1784.

Nair, Vinod and Geoffrey E. Hinton (2010). "Rectified linear units improve restricted boltzmann machines". In: *International Conference on Machine Learning (ICML)*, pp. 807–814.

Paszke, Adam et al. (2016). "ENet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation". In: *CoRR* abs/1606.02147. arXiv: 1606.02147. URL: http://arxiv.org/abs/1606.02147.

PyTorch. http://pytorch.org.

Ronneberger, Olaf, Philipp Fischer, and Thomas Brox (2015). "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *CoRR* abs/1505.04597. arXiv: 1505.04597. URL: http://arxiv.org/abs/1505.04597.

Rosten, Edward, Reid Porter, and Tom Drummond (2008). "Faster and better: a machine learning approach to corner detection". In: *CoRR* abs/0810.2434. arXiv: 0810.2434. URL: http://arxiv.org/abs/0810.2434.

Silva, Thalles (2017). *A Short Introduction to Generative Adversarial Networks*. Ed. by sthalles.github.io. [Online; posted 7-June-2017]. URL: https://sthalles.github.io/intro-to-gans/.

— (2018). *Deeplab Image Semantic Segmentation Network*. Ed. by sthalles.github.io. [Online; posted 29-January-2018]. URL: https://sthalles.github.io/deep_segmentation_network/.

Simonyan, K. and A. Zisserman (2014). "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *ArXiv e-prints*. arXiv: 1409.1556 [cs.CV].

Srivastava, Nitish et al. (2014). "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". In: *J. Mach. Learn. Res.* 15.1, pp. 1929–1958. ISSN: 1532-4435. URL: http://dl.acm.org/citation.cfm?id=2627435.2670313.

Ulyanov, Dmitry, Andrea Vedaldi, and Victor S. Lempitsky (2016). "Instance Normalization: The Missing Ingredient for Fast Stylization". In: *CoRR* abs/1607.08022. arXiv: 1607.08022. URL: http://arxiv.org/abs/1607.08022.

Xu, Bing et al. (2015). "Empirical evaluation of rectified activations in convolutional network". In: *arXiv preprint arXiv:1505.00853*.

Zhu, Song et al. (2002). "What Are Textons?" In: vol. 4, pp. 793–807. DOI: 10.1007/3-540-47979-1_53.