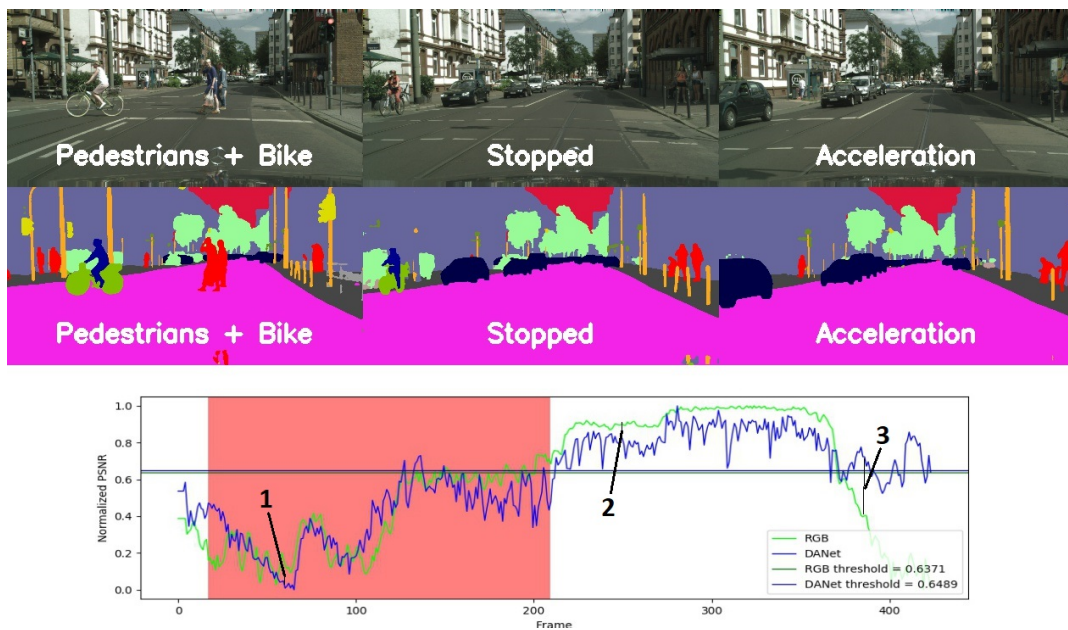

Semantic Segmentation in Anomaly Detection

Researching the feasibility of utilizing semantic segmentation
in anomaly detection



Master Thesis
Michael Bidstrup

Aalborg University
Electronics and IT



Electronics and IT
Aalborg University
<http://www.aau.dk>

AALBORG UNIVERSITY

STUDENT REPORT

Title:

Semantic Segmentation in Anomaly Detection

Theme:

Computer Vision

Project Period:

1. February 2021 - 29. June 2021

Project Group:

Group 1048

Participant(s):

Michael Bidstrup

Supervisor(s):

Kamal Nasrollahi

Jacob Velling Dueholm

Copies: 1**Page Numbers:** 56**Date of Completion:**

June 18, 2021

Abstract:

This project sets out to investigate if semantic segmentation can be used to improve anomaly detection. The problem analysis is centered around finding the best practices for using a dataset on each system without ground truth for both systems available. To achieve this, an analysis is made on segmentation results from models on the anomaly detection dataset UCHK Avenue, and the semantic segmentation dataset Cityscapes is manually annotated with frame specific anomalies. The system used for the anomaly detection is a General adversarial network predicting future frames from spatial and temporal constraints. Testing on the datasets showed that segmented frames can achieve accuracy with the system competitive with RGB frames given a high enough segmentation accuracy.

The content of this report is freely available, but publication (with reference) may only be pursued due to agreement with the author.

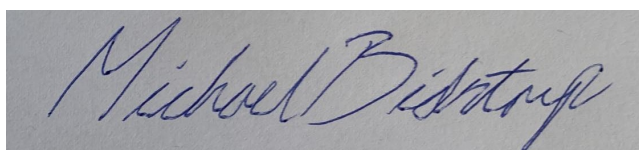
Contents

Preface	vii
1 Introduction	1
1.1 Introduction	1
1.2 Initial Problem Statement	3
2 State of the Art	5
2.1 Learning Temporal Regularity in Video Sequences	5
2.2 Future Frame Prediction for Anomaly Detection	7
2.3 Object-centric Autoencoders and Dummy Anomalies for Abnormal Event Detection in Video	8
3 Problem Analysis	11
3.1 Problem Analysis Overview	11
3.2 Choice of Anomaly Detection System	11
3.3 Anomaly Detection Datasets	12
3.4 Semantic Segmentation Datasets	13
3.5 Dataset Problem	15
3.6 Analysis of Future Frame Prediction System	17
3.7 Future Frame Performance Analysis	23
3.8 Semantic Segmentation	24
3.9 Segmentation Model Test	25
3.10 Experiments Analysis	28
4 Requirement Specification	31
4.1 Requirements	31
4.2 Final Problem Statement	32
5 Problem Solution	33
5.1 Data Processing	33
5.2 Segmentation Network	34
5.3 Anomaly Detection Modifications	35

5.4	Model training	37
5.5	Annotating Cityscapes	37
6	Testing	41
6.1	Testing Overview	41
6.2	Performance Tests	41
7	Discussion	51
7.1	Avenue	51
7.2	Cityscapes	51
8	Conclusion	53
	Bibliography	55

Preface

Aalborg University, June 18, 2021

A handwritten signature in blue ink on a light gray background. The signature is written in a cursive style and reads "Michael Bidstrup".

Michael Bidstrup
<mbidst15@student.aau.dk>

Chapter 1

Introduction

1.1 Introduction

The mapping and understanding of the worlds complexities are at the forefront of technological advancements in current times. With the improvement of deep learning since the 1990's computers has become increasingly able to perform complex pattern recognition within a restrained subject and solve complex tasks. At the center of this development is computer vision. As a signal, images are able to retain vast amounts of information only constraint by the size and resolution off the sensor capturing it. This makes computer vision a perfect input for neural networks, designed to derive high dimensional patterns from large sets of data. Looking at the metrics on google scholar for Engineering and Computer Science[5], the topic with the highest h-index over the past 5 years is the Conference on Computer Vision and Pattern Recognition with an index of 299. The third highest is the International Conference on Learning Representations with an index of 203. This interest also shows in the financial sector, with the global computer vision market being valued at 10.6 billion USD in 2019 and expected to grow at a compound annual growth rate (CAGR) of 7.6% from 2020 to 2027[12].

Computer vision enables computers to obtain spatial and temporal information about given environments. This information allows for robots such as industrial manipulators, developed to perform precise and repetitive manipulations of products continuously, to do complex decision making. It allows for drones to be utilized for automated tasks, cars to approach a self-driving state and surveillance to intelligently identify people and situations of interest.

1.1.1 Anomaly Detection

Anomaly detection, also referred to as abnormal event detection or outlier detection, is the identification of rare events in data. The first anomaly detection systems were developed for data analysis and employed to identify problems such as errors in medical treatments or bank fraud[17]. In computer vision, anomaly detection is researched in the field of surveillance on tasks such as crowd analysis, traffic surveillance and elder care.

With an estimated 300,000 public security cameras in the country of Denmark alone, every minute 5,000 hours of footage is recorded. These cameras are situated at highways, public transportation stations, official buildings and general public areas such as parks[3]. This amount of cameras make manual inspection of surveillance economically unfeasible. Furthermore, storing the footage in case an important moment is captured, require an increasing amount of storage, which force a deletion of footage at least every 30 days. With the ability to automatically determine if footage is relevant or irrelevant through anomaly detection, the amount of stored footage could be greatly reduced and potentially allow for live investigation of the surveillance. This could result in emergency personal receiving notice of a traffic accident before it is called in by bystanders, or police to be aware of an escalating situation requiring their intercession.

1.1.2 Semantic Segmentation

Humans are able to derive context in a scene at an instance of looking. We do this with internal object detection, recognition and permanence. A direct benefit of semantic segmentation is providing the same context we instinctively derive to computers by transforming singular pixel values to semantic groups. This context include:

- What objects are in the scene?
- What are their positions?
- How do the objects move frame-by-frame?
- What are the relationship between objects?

As anomaly detection systems usually has to derive global patterns from singular pixels in RGB space. Training a network with global context classifications could prove to be useful. When training on data were people only walk in certain directions or on a subset of surfaces or objects are found to move at certain velocities between frames, a change in this behaviour should be easier for an anomaly

detection system to detect.

Another benefit of utilizing semantic segmentation for anomaly detection is the anonymity it provides to surveillance systems. By reducing frames to behaviour specific annotations, surveillance could be utilized for security in situations where privacy concern exists such as hospitals or nursery homes[6].

The hypothesis of this project is to test the efficacy of implementing semantic segmentation as a part of anomaly detection to provide further context to a scene on a pixel level. This will either be done by reducing the frame data provided to the system to a lower dimensional semantic space or enhancing the data with the semantic classifications.

1.2 Initial Problem Statement

How can semantic segmentation be used in anomaly detection?

Chapter 2

State of the Art

Anomaly detection can be performed with different architectures of networks. In this section, three types of anomaly detection systems and their individual results in *area under the curve* (AUC) on specific datasets will be presented. For comparison of the systems, three datasets will be used: UCHK Avenue, UCSD Pedestrian 1, UCSD Pedestrian 2. The specification of each dataset will be analyzed further in the problem analysis in Section 3.3.

2.1 Learning Temporal Regularity in Video Sequences

This paper presented by Davis et al. propose two methods of learning regular motion patterns (termed *regularity*) in video sequences[7]. Both methods employ the use of an autoencoder to encode and reconstruct motion signatures from frames. The idea being that the model will produce motion signatures with low error in regular frames and high error in irregular frames. The first approach, denoted IT-AE for *Improved Trajectory Auto-Encoder*, is to reduce the input frames to a set of handcrafted features. For this, Histogram of Oriented Gradients (HOG) and Histogram of Optical Flows (HOF) is used.

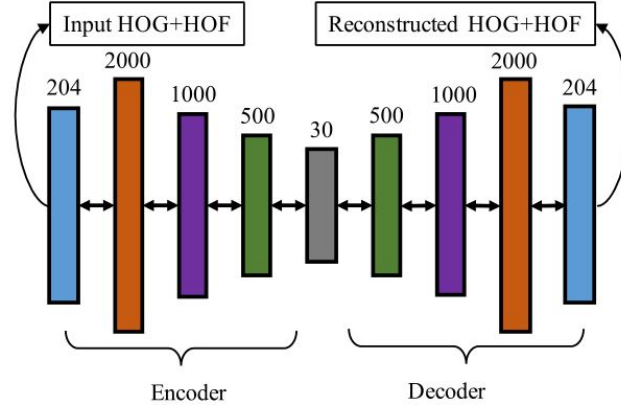


Figure 2.1: HOG+HOF autoencoder architecture[7]

To compute the improved trajectory the input image is divided into a dense grid of 5×5 pixels where the gradients and optical flow are computed at specific interest points. The interest points located in homogeneous texture areas are then excluded following the results of auto-correlation. Finally, the remaining interest points are tracked over a series of frames using optical flow fields and the resulting feature vector is used as input to a fully-convolutional autoencoder.

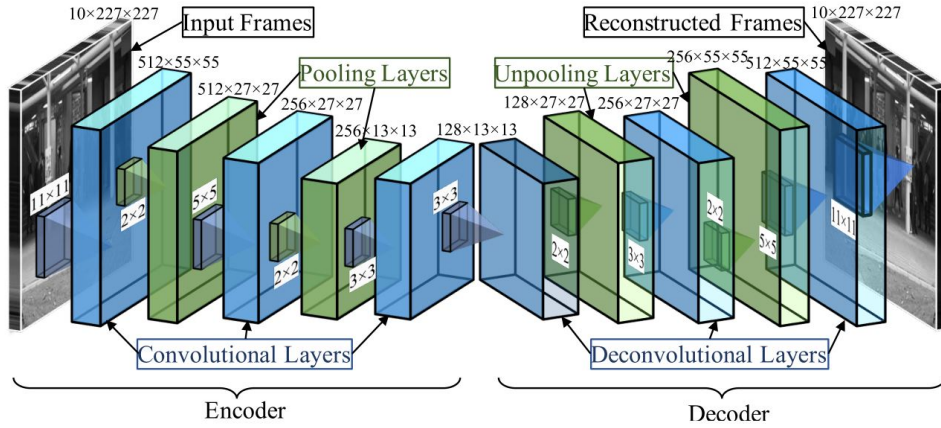


Figure 2.2: Fully convolutional autoencoder[7]

The second approach, denoted Conv-AE for *Convolutional Auto-Encoder*, use a fully convolutional autoencoder with raw video frames as input instead. The autoencoder encodes the motion signatures of the frames and then reconstruct the input from the signatures. The encoder consists of 3 convolutional layers with maxpooling between each layer, reducing the input to 128 feature maps of size 13×13 . The decoder then use deconvolving and unpooling layers in reverse order and size of the encoder for the reconstruction.

The system is able to achieve an AUC of 70.2% on Avenue, 81% on Peds1 and 90% on Peds2. The full data concerning anomalies and positivity rate can be seen in Table 2.1 below.

	Dataset			Regularity		Anomaly detection						
	Frames			Conv-AE		Anomalous events	Conv-AE		IT-AE		Conv-AE	
	All	Regular	Irregular	Correct	FA		Correct	FA	Correct	FA	AUC	EER
UCHK Avenue	15,324	11,504	3,820	11,419	355	47	45	4	43	8	70.2	25.1
UCSD Peds1	7,200	3,195	4,005	3,135	310	40	38	6	36	11	81.0	27.9
UCSD Peds2	2,010	374	1,636	374	50	12	12	1	12	3	90.0	21.7

Table 2.1: The results of the two methods[7]

2.2 Future Frame Prediction for Anomaly Detection

This system, presented by Liu et al. at CVPR in 2018[13], builds on the Conv-AE method from [7] by applying the concept of regularity to a GAN architecture and adding a temporal motion constraint to the training. The generators task is to produce images that the discriminator is incapable of distinguishing from actual images. Reversely, the discriminator is training to become better at identifying the generated images from actual images. In this system specifically, the generator is trained with intensity and gradient as spatial constraint and optical flow as temporal constraint.

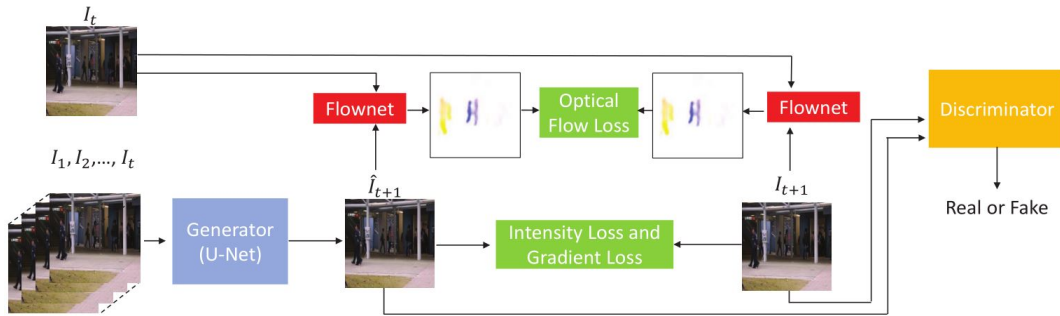


Figure 2.3: Pipeline of system presented in[13]

For the generator a modified version of the autoencoder U-Net is used, which concatenate feature maps from the encoder to the decoder in order to keep spatial relations intact. The optical flow is predicted using FlowNet, which takes two neighboring frames as input and derives the pixel specific motion. FlowNet is also an autoencoder with a CNN as encoder deriving the optical flow and utilizing layers from the encoder for reconstruction.

The intensity and gradient loss is computed as the difference between the generated frame and the actual frame at time t . The optical flow loss is the difference

between the optical flow predicted from the generated frame and frame $t-1$ with the original frame and the frame $t-1$.

The system produced state of the art results at the time on UCHK Avenue and UCSD Pedestrian 1 and 2 with an AUC of 85.1%, 83.1% and 95.4%, respectively. The results for the four dataset is presented in Table 2.2.

	UCHK Avenue	UCSD Peds1	UCSD Peds2	ShanghaiTech
AUC	85.1%	83.1%	95.4%	72.8%

Table 2.2: Results of Future Frame Prediction system presented in [13]

2.3 Object-centric Autoencoders and Dummy Anomalies for Abnormal Event Detection in Video

This paper by Ionescu et al. achieved state of the art results at CVPR in 2019, changing the state of the art from frame reconstruction to object-centric methods.

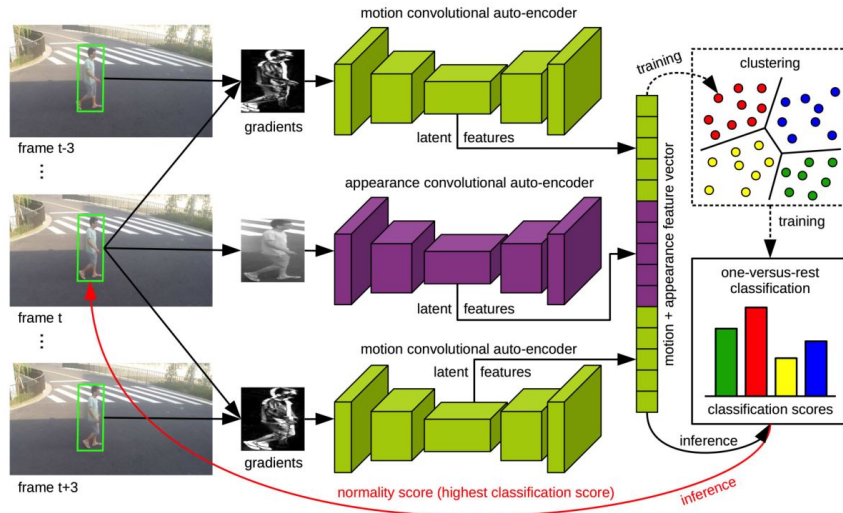


Figure 2.4: The pipeline of the system as presented in [11]

The system first deploy an object recognition algorithm, such as Single Shot Detector (SSD) or You Only Look Once (YOLO), to identify every object of interest in the scene. As anomaly detection in most regards is not concerned with the background but instead the actions of the objects, this method helps reduce the amount of redundant data to process. What sets this paper apart is the treatment of anomaly detection as a clustering problem. By using the encoder from a convolutional autoencoder as to extract features, and instead of feeding the data to a decoder for reconstruction, creates a feature vector, the system can utilize a

support vector machine (SVM) to learn boundaries of clusters[11]. To encode the features the system use three different encoders: An appearance encoder for the frame at time t , and two motion encoders fed the gradients between frame $t-3$ and $t+3$. Doing this for every object detected in the scene the system is able to detect anomalies based on the motion and appearance of the objects.

The system achieves state of the art results on two of three benchmark datasets. UCHK Avenue, UCSD Pedestrian 2. It does not present results on UCSD Pedestrian 1. The respective scores can be seen in Table 2.3.

	UCHK Avenue	UCSD Peds1	UCSD Peds2	ShanghaiTech
AUC	90.4%	N/A	97.8%	84.9%

Table 2.3: AUC results from [11]

Chapter 3

Problem Analysis

3.1 Problem Analysis Overview

The problem analysis in this project tries to derive the best suitable anomaly detection system and semantic segmentation model for the task of testing the problem statement. To do this, an analysis of specific datasets for each system is conducted to address the task of using the systems with data outside their intended framework. It furthermore analyze how the chosen anomaly detection system functions and on what metrics it is evaluated in order to gain an understanding of how to develop the problem solution and benchmark the results.

3.2 Choice of Anomaly Detection System

To select the best anomaly detection system to experiment with semantic segmentation on, the results of the systems from the discussed datasets will presented. As seen in Table 3.1, the future frame prediction system and object-centric SVM outperform the proposed method from the first paper.

Method	CUHK Avenue	UCSD Peds1	UCSD Peds2	Shanghai Tech
Conv-AE	80%	75%	85%	60.9%
Future frame prediction	85.1%	83.1%	95.4%	72.8%
Object centric SVM	90.4%	N/A	97.8%	84.9%

Table 3.1: Comparison of results from anomaly detection system

As mentioned in the introduction, semantic segmentation perform a pixel specific labeling of semantic classes to images. This labeling provide global context such as position, count and relationship of objects in the scene. From this, it is believed that a system which work with entire frames as input would have the greatest benefit of the semantic labeling, as it retains the global relationship of

objects in the scene. Even though state of the art results are achieved with the object-centric SVM, the systems works by segmenting the region of interest for each object, classifying based on the local clusters and performing decision-level fusion of the results. The future frame prediction GAN works by reducing full frames to targeted information and should theoretically benefit greater from the global context. For this reason, the future frame prediction system is chosen as the platform for testing the problem statement.

3.3 Anomaly Detection Datasets

Datasets are an essential part of creating any kind of classifier. A dataset creates the boundaries wherein a network aims to understand the world and in the best cases sets the network up to decode the patterns required to do its task.

For anomaly detection, five main datasets are used for benchmarking systems: Pedestrian 1, pedestrian 2, Avenue, ShanghaiTech and UMN. For this project, the three datasets used in common by the state of the art systems will be presented.

UCSD Peds1 and Peds2[1]

UCSD Pedestrian 1 and 2, abbreviated Peds1 and Peds2, contains 8-bit grayscale videos of pedestrians at walkways at University of California. The anomalies consists of unintended vehicles on the walkways such as bikes, skateboards and carts. The videos are at resolution 238×158 recorded at 10 fps. The dataset is among others used for motion segmentation, crowd counting and anomaly detection.



Figure 3.1: Examples of anomalies by ROI in UCSD Pedestrian 1 dataset

For anomaly detection the dataset includes two types of annotations, frame annotations with a binary flag indicating frames with abnormal events and region annotations with ROI of the area with the anomaly. The specific amount of frames for training and testing can be seen in table 3.2 below:

	Training		Testing	
	Videos	Frames	Videos	Frames
Peds1	34	6800	36	7200
Peds2	16	2550	12	2010

Table 3.2: The number of videos and frames for testing and training in Peds1 and Peds2

UCHK Avenue[14]

Avenue is a dataset developed by the University of China, Hong Kong. It contains RGB videos roughly 2 minutes long of people commuting in a public avenue. The set includes 47 abnormal events such as people running, objects being thrown and loitering. The training set consists of 16 videos with 15,328 frames and the test set of 21 videos of 15,324 frames totalling 30,652 frames. As with UCSD pedestrian avenue is annotated with both frame and region specific anomalies.



Figure 3.2: Examples of anomalies by ROI in UCHK Avenue dataset

From the 15,324 frames in the test set, 11,504 are normal and 3820 are abnormal frames. The specific number of anomalies for each video are presented in Table 3.3.

	Training		Testing	
	Videos	Frames	Videos	Frames
UCHK Avenue	16	15,328	21	15,324

Table 3.3: Number of videos and frames for testing and training in UCHK Avenue.

3.4 Semantic Segmentation Datasets

For semantic segmentation four datasets are used to benchmark the result of systems. COCO, Pascal, ADE20K and Cityscapes.

3.4.1 ADE20K

ADE20K is created by the MIT Computer Vision team and is the largest open source dataset for semantic segmentation and scene parsing[16]. It is therefore also

one of the most used datasets for benchmarking semantic segmentation systems. The focus of ADE20k is to provide a large dataset with a diverse set of scenes and dense annotations of all the objects present in the scene. There are 20,210 images in the training set, 2,000 images in the validation set, and 3,000 images in the testing set. The images are from the SUN, LabelMe and Places datasets and is selected to cover the 900 scene categories defined in the SUN database. For annotations, each image is manually annotated by a single person to ensure consistency using the LabelMe interface. Examples of images and their annotations can be seen in Fig. 3.3 below.



Figure 3.3: Examples of images and annotations from ADE20K dataset

3.4.2 Cityscapes

Cityscapes is developed by the Technical University of Dresden and the Technical University of Darmstadt in cooperation with MPI informatics and Daimler AG[2]. The dataset contains video sequences of urban scenes from the point of view of a driver. It was created by mounting two cameras on the helm of a car and driving through 50 different cities in Germany. The dual camera setup is used to obtain depth information as well as RGB video. The dataset in its basic form contains select images from each drive with fine pixel-level annotations. In total there are 5,000 images with ground truth divided into 3,475 for training and validation, and 1,525 for testing. Additionally, the authors has created 20,000 images with coarse annotations for methods that leverage large volumes of weakly-labeled data. Examples of fine annotated is presented in Fig. 3.4.



Figure 3.4: Examples of annotated images from Cityscapes dataset

3.5 Dataset Problem

A problem which needs to be addressed for this project is the requirement of using datasets across their intended systems. When testing the efficacy of utilizing semantic segmentation in anomaly detection, it would either be required to perform semantic segmentation on an anomaly detection dataset, for which there exists no ground truth in respect to semantic groups, or reversely, obtain a semantic segmentation dataset with anomaly annotations. This problem can be solved in two ways:

1. Find a semantic segmentation model which is trained on data similar to an anomaly detection dataset
2. Find a semantic segmentation dataset which contains both normal and abnormal events and annotate manually.

As this project strives to improve anomaly detection, using a dataset used for benchmarking by other systems is necessary for result comparison. However, in order to research what influence the semantic segmentation quality has on the anomaly detection, having access to different qualities of segmented data up to state of the art results is important for research purposes. For this reason both solutions will be pursued.

3.5.1 Choice of anomaly detection dataset

Finding an anomaly detection dataset which resembles a semantic segmentation dataset can be evaluated on three requirements: View level, objects and color. Looking at ADE20k and Cityscapes both datasets contain images in eye level view and RGB color. From this, both Peds1 and Peds2 can be removed from consideration as these are grayscale and has an elevated view. The dataset UMN and ShanghaiTech can also be occluded for the same reason. This leaves UHCK Avenue, which satisfies the first two requirements.

For objects in Avenue, there are two categories: Moving objects and static background environment. The moving objects are what defines the abnormal behaviour and includes people, bags, papers and bicycles as illustrated in 3.12 in the next section. The background is the same across the dataset and contain grass, sidewalk, poles, stairs and walls. It should be noted that a consistent classification of the objects in Avenue is more important than a correct classification as the specific class predictions can be neglected for the context they provide to the scene. Because of this, the most important objects to segment are the moving objects, as the anomalies in Avenue comes from the behaviour of these. The background should

theoretically help provide global context to the scene, but could also result in unwanted noise, which will be tested further in 6.

ADE20K and Cityscapes both have classes to adequately describe the background in Avenue, but differs a little on the moving objects. As described in 3.4 ADE20K is created to be as versatile as possible with 150 different classes. These include all moving objects in Avenue except papers. Cityscapes is a little more specific to the urban scene with only 30 classes and therefore have no class for the bag nor the papers.

3.5.2 Choice of Semantic Segmentation Dataset

In order to test the problem statement in the best conditions, with minimal noise in the semantic segmentation, a dataset with a trained segmentation model has to be utilized. As manually annotating Avenue with semantic ground truths is too extensive a task given the time constraints of the project, acquiring a semantic segmentation dataset to be fed into the anomaly detection system is a more suitable solution. The task here is to find a dataset which upholds two main requirements: The dataset has to contain videos and it has to contain a form of normal and abnormal events. Datasets such as COCO, Pascal and ADE20K are based on singular images and can therefore be excluded from consideration. A lesser used dataset which is formatted as videos is the Densely Annotated Video Segmentation dataset (DAVIS). This dataset seems obvious to use as it is specifically made for semantic segmentation of videos, however, the length of the videos are not very long and each video contain a different type of singular event such as a guy breakdancing or rabbits on a field.

The standard format of Cityscapes cannot be used either, even though this dataset is made from video footage. This is because the dataset consists of a subset of frames from the original videos and the originals are not available for download. What is available is all of the frames from one out the 50 videos the dataset consists of. This is the drive in Frankfurt, which can be made available upon request to the Cityscapes team. As the majority of this dataset contain scenes similar to what is in the Cityscapes dataset, the segmentation should be almost as good. The dataset is therefore suitable for the application in the first requirement.

The Cityscapes dataset is not created to contain any normal or abnormal events. So for the second requirement, abnormal events will have to be defined based on the context of the data.

3.6 Analysis of Future Frame Prediction System

To gain an in depth understanding of the anomaly detection system selected for the project, a deeper analysis of the components and constraints of the system will be conducted. As described in Section 2.2, the design of the system is to predict the upcoming frame at time $t+1$ from the intensity, gradients and motion of the scene. In the best scenario, general behaviour will be taught by the generator and predicted successfully during inference while abnormal events will not and therefore produce a larger prediction error. The pipeline of the system can be seen in Fig. 3.5

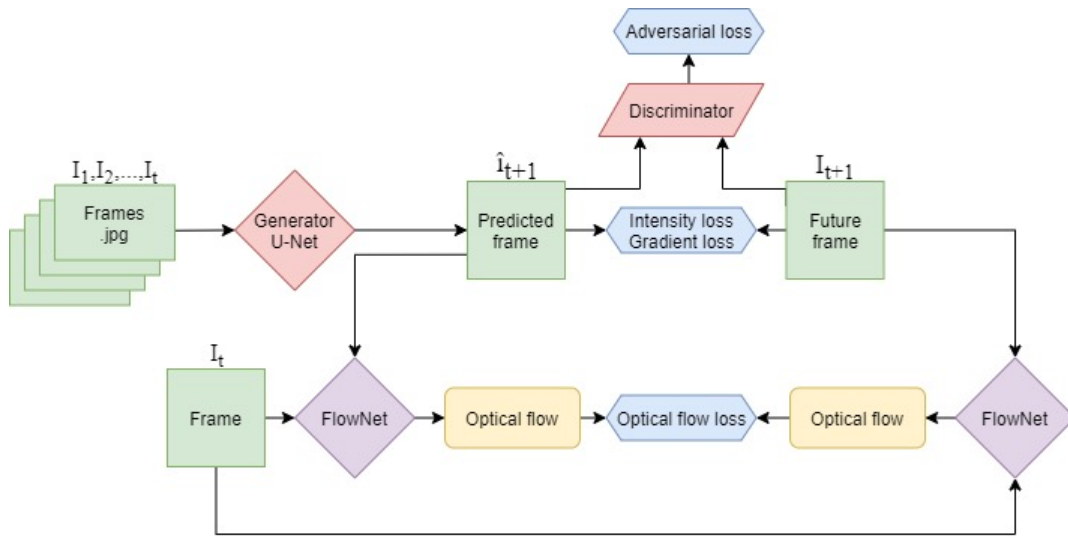


Figure 3.5: The pipeline of the Future Frame Prediction in Anomaly Detection system

As illustrated, the system works by loading a series of sequential frames to the generator to construct the prediction of the future frame denoted $t+1$. To quantify the prediction difference, peak signal-to-noise ratio (PSNR) is used as error metric and input to a receiver operating characteristics (ROC) curve for finding the best cut off threshold for the binary classification.

3.6.1 U-Net

Networks used for image generation usually contains two modules: an encoder which extracts features by gradually reducing spatial resolution and a decoder which gradually recovers the frame by increasing the spatial resolution. A problem with auto-encoders is they consists of many hidden layers and therefore suffer from the vanishing gradient problem. Vanishing gradient problem occur when the gradient of the loss function becomes too small, effectively preventing the weights

in the network from changing values during back-propagation. In other words, when errors are back-propagated to the first few layers they become too insignificant, and the network will almost always learn to reconstruct the average of all the training data. In the worst case, this stops the neural network from training any further. To avoid this, U-Net is modified in the future frame prediction system by adding a shortcut between encoder and decoder layers, as shown in 3.6 by the green horizontal lines.

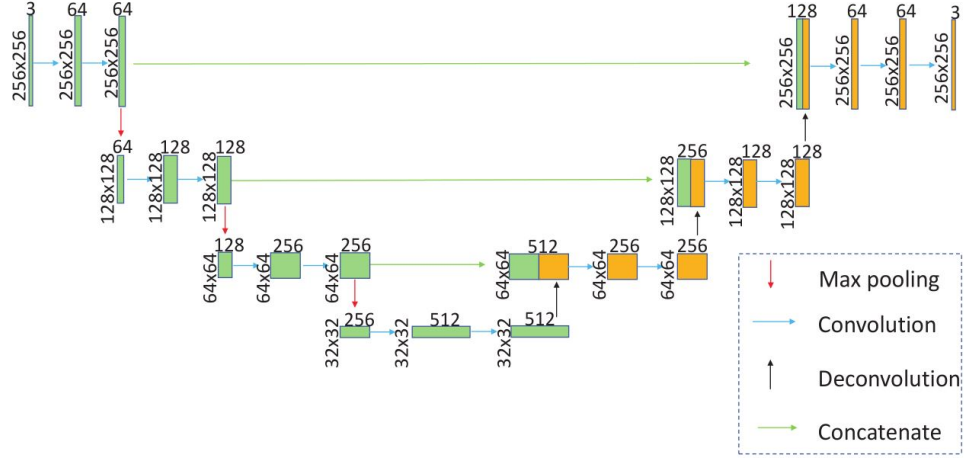


Figure 3.6: Modified U-Net from [13] with shortcuts from encoder to decoder

The shortcut, used in Residual networks, concatenates the data to the decoder, expanding the dimensions to fit the encoder. This suppress the gradient vanishing problem as it adds data which has not been through the activation functions that suppress the gradient. Another benefit of this methods is that it maintains spatial resolution, resulting in information symmetry and removes the need for cropping and resizing. The kernel size of all convolution and deconvolution layers are set to 3x3 and pooling layers are set to 2x2[15]. An example of a frame and a predicted frame can be seen in Fig. 3.7.

Figure 3.7: A frame and predicted frame from U-Net

3.6.2 FlowNet

Optical flow describes the pixel specific motion in a scene caused by movements of the camera or objects. To obtain this in the system a pre-trained network, FlowNet, is used. FlowNet is also an autoencoder, with a CNN trained to derive optical flow between images as encoder and a decoder to reconstruct the frames to a larger resolution for the prediction[4]. At its core the FlowNet encoder is a generic

network of convolutional layers processing how best to extract motion information itself. It use convolutional and max pooling layers to derive motion features in a lower dimensional space as seen in Fig. 3.8. The FlowNet model used in the project is FlowNetSD, developed to improve the original model on small displacement by reducing the kernel size and strides in the input layers[10].

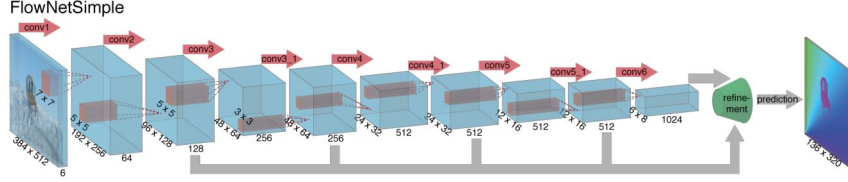


Figure 3.8: The correlation encoder as presented in [4]

Similar to U-net the decoder has been modified to account for the resolution reduction of pooling. The benefits of pooling in a network is that it makes training computationally feasible and allows for aggregation of information over large areas of input images. A problem with pooling though, is a reduction in resolution unwanted for per-pixel predictions. To solve this, a refinement of the coarse pooled representations using upconvolutional layers is used. Here, similar to U-Net, the output from the input layers are concatenated to each layer of the decoder. Furthermore, the output from the previous decoder layers is upsampled and added as well as illustrated in Fig. 3.9.

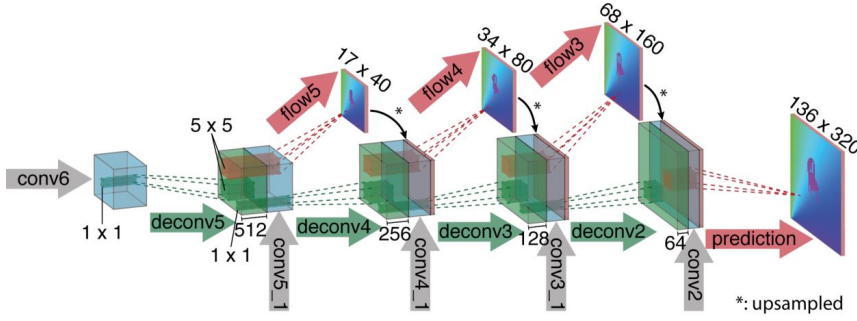


Figure 3.9: The decoder of FlowNet as presented in [4]

The training loss for the network is computed as the endpoint errors (EPE). This is the standard error measure for optical flow estimation, computed as the average of the Euclidean distance between the predicted flow vector and the ground truth flow vector over all pixels. Defined mathematically as:

$$||V_{est} - V_{gt}|| \quad (3.1)$$

3.6.3 Constraints

The motion constraint for the flow loss is computed as the norm of the vector difference from the optical flows:

$$L_{op}(\hat{I}_{t+1}, I_{t+1}, I_t) = \|f(\hat{I}_{t+1}, I_t) - f(I_{t+1}, I_t)\|_1 \quad (3.2)$$

The intensity loss is used to train direct similarity of the pixels in RGB space. To compute the intensity loss the 2 norm is computed between the difference of the predicted frame and its ground truth I as:

$$L_{int}(\hat{I}, I) = \|\hat{I} - I\|_2^2 \quad (3.3)$$

The gradient loss is used to train sharpness in the generated images. The gradients are computed as the absolute difference between neighbouring pixels for both the predicted and original image. The 1 norm is then computed for each direction before being added. Mathematically, the gradient loss is defined as:

$$L_{gd}(\hat{I}, I) = \sum_{i,j} |||\hat{I}_{i,j} - \hat{I}_{i-1,j}| - |I_{i,j} - I_{i-1,j}|||_1 + |||\hat{I}_{i,j} - \hat{I}_{i,j-1}| - |I_{i,j} - I_{i,j-1}|||_1 \quad (3.4)$$

where i, j denotes the pixels of a frame.

Finally, the adversarial loss function for the generator is defined as:

$$L_{adv}^G(\hat{I}) = \sum_{i,j} 1/2 L_{MSE}(D(\hat{I})_{i,j}, 1) \quad (3.5)$$

Each individual loss function presented above is able to scale to n-dimensions without mathematical problems. This is important as a possible solution to the problem statement might include a combination of RGB and segmented data. For this to work, the system must be able to train on images which contain more channels than RGB.

Combined objective function

The global loss g_loss for training the generator is defined in the combined objective function. It is computed as a weighted combination of the four constraints:

$$L_G = \lambda_{int} L_{int}(\hat{I}_{t+1}, I_{t+1}) + \lambda_{gd} L_{gd}(\hat{I}_{t+1}, I_{t+1}) + \lambda_{op} L_{op}(\hat{I}_{t+1}, I_{t+1}) + \lambda_{adv}^G L_{adv}^G(\hat{I}_{t+1}) \quad (3.6)$$

The weights controlling the influence of each constraint when training is set in `hyper_params.ini`. The weights used to train the pre-trained model on Avenue use weight values: intensity = 0.001, gradients = 1, adversarial = 0.05 and flow = 2.

3.6.4 Inference

At inference the system computes the prediction error of the future frames and quantify the signal using Peak signal-to-noise ratio.

Peak signal-to-noise ratio

Peak signal-to-noise ratio (PSNR), measured in decibels (dB), is as the name suggest a measure of ratio between a signal and noise introduced to the signal. Specifically, it is the relation between the maximum power of a signal and the noise. It is commonly used to quantify reconstruction quality of images subject to compression. The signal being the original image and the noise the error introduced after compression. This makes it a good measuring tool for comparing generated images from a GAN to a target image. PSNR is computed as a logarithmic function on the relationship between the maximum signal power squared divided by the Mean Square Error (MSE). Defined in Equation 3.7 below[9].

$$PSNR(f, g) = 10 * \log_{10}(255^2 / MSE(f, g)) \quad (3.7)$$

Where MSE is computed following:

$$MSE(f, g) = 1/MN \sum_{i=1}^M \sum_{j=1}^N (f_{ij} - g_{ij})^2 \quad (3.8)$$

As the MSE approaches zero the PSNR approaches infinity, meaning a higher PSNR represents a better generated image. At inference PSNR scores for each frame prediction is computed and normalized to 0,1 across the frames of each video following:

$$S_t = PSNR_t - \min(PSNR) / \max(PSNR) - \min(PSNR) \quad (3.9)$$

This allows for the scores to be treated as probabilities when computing the cut off threshold for the binary classification.

3.6.5 Evaluation

Based on the evaluation parameter given to the inference scripts, the system can compute different evaluation metrics. These include average PSNR, EER and AUC. For this project, the metric used to compare results when testing is the *area under the curve* from *receiver operating characteristics* curve.

ROC curve

Receiver operating characteristics (ROC) curve is a two-dimensional measure of classification performance in a binary classifier. It maps the true positive rate (TPR), also denoted as the *sensitivity* of the system on the y-axis against the false positive rate (FPR), denoted (*1-specificity*) on the x-axis. TPR and FPR are computed following:

$$TPR = TP / (TP + TN) \text{ and } FPR = (FP / FP + FN) \quad (3.10)$$

An ROC curve following the diagonal line $y = x$, called the reference line, produce true positive results at the same rate as false positive results. It follows that the goal of a system is to produce as many true positives as possible, resulting in an ROC curve in the upper left triangle[8]. Here, a threshold can be decided based on the importance of capturing every true positive at the cost of more false positives.

Figure 3.10: The correlation encoder as presented in [4]

Finding the optimal cut off threshold for classification is done by computing the TPR and FPR of different threshold values. In the system, this is done by inputting the scores and labels to sklearn's metric function `roc_curve()` which returns the TPR's and FPR's off different cut off values. The resolution of the thresholds tested for the curve is dependent on the number of unique prediction in the data. The optimal threshold value for the pre-trained model on Avenue results in a threshold of 0.69.

Area under the curve

To obtain a global measure of a systems classification performance, the area under the curve (AUC) is used. AUC represents the percentage of true positives in relation to the number of samples in ROC. An AUC of 1.0 represents perfect discrimination in the test with every positive being true positive and every negative being true negative. An AUC of 0.5 represents no discriminating ability with classifications being no better than chance[8]. As illustrated in the graph 3.11 below, the pre-trained model achieves an AUC of 0.85.

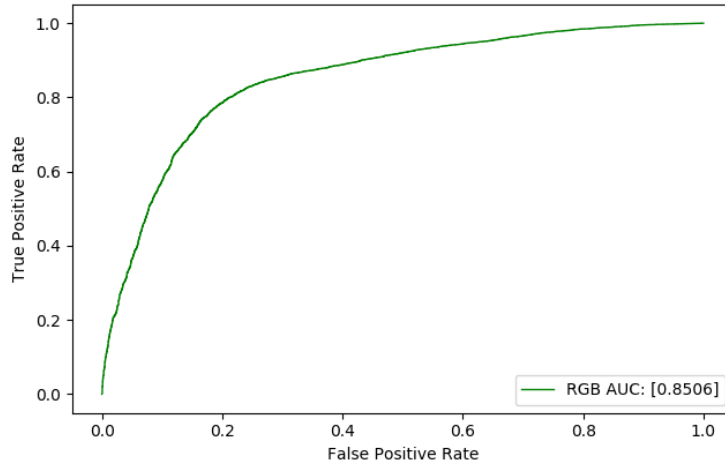


Figure 3.11: ROC and AUC of Future frame prediction model

3.7 Future Frame Performance Analysis

In order to quantify the target results of the proposed solution, an analysis of the future frame prediction systems performance on Avenue is conducted. As stated in 3.6 above, the system utilize PSNR scores to perform binary classification. To better understand which anomalies the system is able and unable to detect, each anomaly has been annotated and a comparison of the TPR, FPR, TNR and FNR for each anomaly is presented in Table 3.4 below.

	Running	Direction	Papers	Bag	Close	Kids playing	Camera shake	Total
Videos	1-4	9-11,13-16, 20	13,14,20	5,6,9-12	1,6,19	7-9,17,18,21	2	All
Abnormal events	9	10	3	12	4	8	1	47
Abnormal frames	377	456	187	1154	743	854	49	3820
Detected events	9/9	10/10	3/3	12/12	4/4	8/8	1/1	47/47
True positive	258	281	184	1045	632	600	25	3025
False negative	119	175	3	109	111	254	24	795
TPR	0.6844	0.6162	0.984	0.9055	0.8506	0.7026	0.5102	0.7918

Table 3.4: Performance of system for each anomaly

As listed in the table, the system is able to detect all anomalies. The most consistent detections are the objects being thrown with 98% and 91%. The weakest are the more distant anomalies with the person running and people walking outside the regular walking area with 68% and 62%, respectively.

3.8 Semantic Segmentation

For the research in this project, the semantic segmentation system is used to transform the data to the anomaly detection system. For this reason, the development of a system is not of importance. Instead, a series of different models will be used to segment UCHK Avenue in order to determine quantitatively which model is best suited for the task. To test the models, the system *Dual Attenuation Network for Scene Segmentation* (DANet) is used. The system achieved state of the art results on Cityscapes in 2018 with a mean IoU of 81.5%. For benchmarking their results, the authors setup a server with pre-trained models from previous state of the art systems for comparison.

Before analyzing different model segmentations on Avenue an analysis of how semantic segmentation is evaluated will be conducted.

3.8.1 Semantic segmentation ground truth

The ground truth for semantic segmentation datasets consists of grayscale images with pixel values representative of the semantic classes. When a system produce a prediction, the accuracy can be evaluated by comparing the prediction for each pixel directly to the values in the ground truth image. Using 8-bit grayscale images for ground truth does limit the amount of classes to 256 in range [0,255], but as described in 3.4, the dataset with the most classes to date is ADE20K with 150, which leaves room for further development of datasets. For a dataset such as Cityscapes with 19 classes, the pixel values range from [0,19] with non-classified pixels denoted in white with value 255.

3.8.2 Segmentation evaluation metrics

Semantic segmentation systems are evaluated on two metrics: Pixel accuracy (Pix-Acc) and mean intersection over union (mIoU). Pixel accuracy is a comparison of each pixel with the ground truth classification computed for each class as:

$$PixAcc = (TP + TN) / (TP + TN + FP + FN) \quad (3.11)$$

Where:

- TP (true positive) = Class pixels correctly classified.
- TN (true negative) = Non-class pixels correctly classified as not in the class.
- FP (False positive) = Class pixels incorrectly classified
- FN (False negative) = Non-class pixels incorrectly classified.

When a ratio of correctness for each class has been computed they are averaged over the set of classes. A problem with PixAcc is that classes with a small amount of pixels achieve high pixel accuracy from the true negative being high. In other words, as true negative approaches infinity pixAcc approaches 1.

To avoid this problem, mIoU computes the accuracy of each class as the relationship between the intersection with other classes over the union of the classes (IoU):

$$IoU = \text{area of overlap} / \text{area of union} \quad (3.12)$$

Or in other terms:

$$IoU = TP / (TP + FP + FN) \quad (3.13)$$

Before averaging over the amount of classes. This removes the true negatives from the equation and solves the problem.

3.9 Segmentation Model Test

After segmenting Avenue with 5 selected models, a quantitative analysis of the segmentations are performed to compare the results of the different models. To ensure the comparison covers as many scenarios as possible, six scenes were selected based on their difference in objects and motions. A frame from each scene is illustrated in Fig. 3.12.



Figure 3.12: Six frames selected for segmentation test

For comparison and to qualitatively test the best performing models, the six frames have also been manually annotated using the LabelMe annotation interface as illustrated in Fig. 3.13.

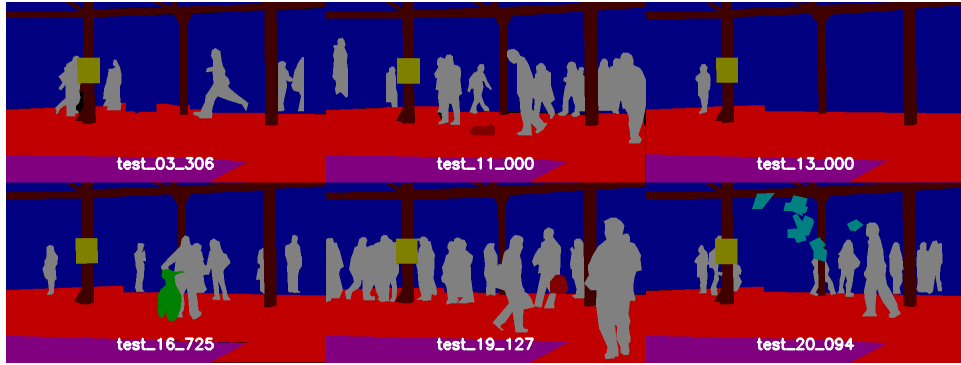


Figure 3.13: Annotated ground truth of the six frames

The pre-trained segmentation models vary on four parameters: network, backbone, dataset and number of classes. The pre-trained models tested is listed in Table 3.5.

Model	Backbone	Classes	Dataset
FCN	ResNet	50	ADE50k
DeepLab	ResNest	50	ADE50k
EncNet	ResNet	101	ADE50k
DANet	ResNet	101	CityScapes
DRANet	ResNet	101	Cityscapes

Table 3.5: The pre-trained models tested on Avenue

The first test is made to see how well the background environment gets segmented. Here, an empty scene is picked to focus on the segmentation of the grass, walkway, building and pillars. As can be seen in 3.14, every model segmenting with 50 classes is not able to segment the pillars from the background.



Figure 3.14: Segmentation of empty frame by selected models

The second test includes three different frames selected for their different objects present in the scene: a backpack, bicycle and papers. As these objects represent abnormal behaviour in the dataset, they are instrumental to recognize for the segmented data to function in the anomaly detection system. The comparison for the bicycle frame can be seen in Fig. 3.15.



Figure 3.15: Segmentation of bicycle from selected models

As can be seen in the above images the bicycle is only recognized by the DANet and DRANet models trained on Cityscapes. The final highlighted test is the models sensitivity to blurred objects. The most extreme case for this is the man running past the camera in test video 16. The comparison of the models can be seen in Fig. 3.16



Figure 3.16: Segmentation of running man from selected models

From the above comparison it is clear that DANet and DRANet outperforms the other networks. The noise in the scene from the models trained on ADE20K,

The best performing model between DANet and DRANet comes down to individual factors. DRANet segments the people a little more precisely and is also able to detect the third pillar in the background more consistently. However DANet is able to segment the grass from the walkway. To find the best model of the two,

a qualitative analysis of the models is performed by mapping the ground truth annotations to fit the Cityscapes class labels. It does not matter what the segmentation system classifies each individual object as, as long as it is consistent in the classification. So for this analysis, the predictions from each system were output as a label image and analysed in order to create the label mapping for the ground truth. An example of a predicted label image and the ground truth label image can be seen in 3.17.



Figure 3.17: Predicted image (left) and label image (right)

The pixAcc and mIoU for the two models can be seen in 3.6

Model	Classes	PixAcc	mIoU
DANet101	19	71.69%	19.94%
DRANet101	19	59.34%	14.91%

Table 3.6: Pixel accuracy and mean IoU of best performing models on Avenue

3.10 Experiments Analysis

The proposed solution to the problem statement for this project will a series of experiments. Before describing the requirements for the solution an analysis of what experiments will be conducted and how the data will be formatted is presented. The solution overview will draw from the conclusions of the problem analysis.

The first thing to address is the dataset problem explained in Section 3.5. In order to research the possible impacts of the segmentation quality, both the anomaly detection dataset Avenue and the segmentation dataset Cityscapes should be experimented on.

3.10.1 Avenue

When experimenting on Avenue the most feasible solution is to experiment with the model found to best segment the data in 3.9 and conduct different experiments

with data formatting. Here, four manipulations of the data is found to be of interest:

- Segmented frames
- Segmented frames without background
- RGB frames with predictions included
- RGB frames with predictions included without background

The first thing to test is how well the segmented data is able to function in the anomaly detection system by itself. The hypothesis here is that a reduction from individual pixels to semantic groups could be beneficial as long as the semantic classification is precise enough to capture the background and the moving objects precisely.

As the background doesn't serve much of a purpose in Avenue, testing if the anomaly detection improves by only training on the moving objects should be experimented with. The reasoning behind this experiment is to see if removing potential noise in the segmentation improves the results from the segmented frames.

As the pre-trained Avenue model is already able to achieve a good result on the RGB frames with an AUC of 85%, testing if the semantic predictions could help improve this score should be tested. This could be done by adding the semantic predictions to the RGB data as a fourth channel and adapting the GAN to account for the new dimensions.

3.10.2 Cityscapes

To experiment on Cityscapes it is necessary to annotate the dataset with ground truth anomalies for computing an ROC curve and AUC. For this, specific anomalies and normal behaviour will have to be defined and separated to training and testing data. To annotate the data an annotation interface will have to be developed in order to assure consistency in the annotations across the entire dataset.

3.10.3 Comparison of results

For comparing the results when training on RGB and segmented frames, four different evaluation methods can be applied:

- Visual comparison
- TPR and TNR comparison

- ROC and AUC
- Standard deviation of PSNR

In order to identify which scenarios the system works and fails in when trained on the data, visualizing the PSNR scores with their respective frames should help create an understanding of the systems capabilities on a general level. Here, each test video with PSNR graphs and ground truth annotations should be rendered to review the strengths and weaknesses of each system.

To conduct the same analysis on a more quantifiable level, comparing the number of anomalies, true positives and true negatives from the RGB and segmented data systems for each video should also be done.

For the overall comparison of results, the ROC curve and AUC should be applied.

Mathematically, the two systems can also be compared on their behaviour in PSNR. Specifically, the standard deviation of the scores without fluctuations from anomalies. As it is expected that systems with stable PSNR graphs for normal frames will perform the best, comparing the overall result of each system with the standard deviation of the PSNR scores could yield insight into the relationship between the segmentation quality and the AUC.

Chapter 4

Requirement Specification

4.1 Requirements

The requirements for the project solution is divided into performance requirements on Avenue and Cityscapes. The performance requirements are centered around comparing the results of the applied method with existing metrics. The performance requirements has to be met for the problem statement to be validated.

UCHK Avenue

The requirements for the performance on Avenue is to produce results better than the original system. For Avenue the performance must show:

- Stable PSNR scores for normal events
- A true negative rate higher than 79.33%
- A true positive rate higher than 79.19%
- An AUC of 85% or above

Cityscapes

The requirements for the performance on Cityscapes is to produce better results on segmented data than RGB. For Cityscapes the performance must show:

- Learned general behaviour in the moving scene
- Detection of anomalies defined in the annotation for both data types
- Stable PSNR scores for normal events
- A higher true positive rate for segmented data than RGB data

- A higher true negative rate for segmented data than RGB data
- A higher AUC with segmented data than with RGB data

4.2 Final Problem Statement

Can semantic segmentation be used to improve future frame prediction for anomaly detection?

Chapter 5

Problem Solution

5.1 Data Processing

Working with the anomaly detection and semantic segmentation system require some amount of data processing. In order to standardize the layout of the data each dataset is setup following:

`<root>/<dataset name>/<data type>/<frames>/<video number>/<images>`

Where data type refers to either: training, testing, segmented or processed segmented.

To perform the experiments, evaluation and visualization of results in the project six python scripts were created. A list of the scripts and a description of their function can be seen in Table 5.1.

Script	Functionality	Input	Output
evaluate_models.py	Finds the model with the best AUC from inference on different checkpoints and outputs metrics	PSNR scores and evaluation metric	AUC, EER, precision recall AUC, avg. PSNR, avg. score
make_graphs.py	Loads PSNR scores and creates animated graph with input scores and ground truth annotations.	One or multiple PSNR data files	Graph with scores and ground truth for each test video
combined_video.py	Combines folders with video frames and graphs into video for visual inspection	One or multiple frame folders and animated graph folders	Combined video of input
segment.py	Performs semantic segmentation on a dataset with desired model, validates output files and writes semantic prediction to text file	Folders of RGB images and segmentation model	Prediction file and segmented image for each input image
format_dataset.py	Moves data from input path to desired folder structure	Folder of images	Formatted dataset
annotate.py	Annotation interface displaying frames in window with overlay while listening for key inputs for annotating anomalies	Folders of video sequences as images	Array of binary annotations

Table 5.1: Description of scripts created to process data for experiments

5.1.1 Visualizing results

Besides computing the AUC from the models to obtain the overall accuracy, a visual representation of the normalized PSNR scores on a frame-by-frame basis was created. This was done to gain a deeper understanding of which scenarios the model is able and unable to predict. To do this analysis an animated graph

containing the scores was rendered alongside the frames from the respective videos as seen in figure 5.1 below. The red areas indicate ground truth anomalies, the horizontal lines are the thresholds computed from the ROC curve as classification boundaries for each model and the current frame is marked by the red dots on the graph.

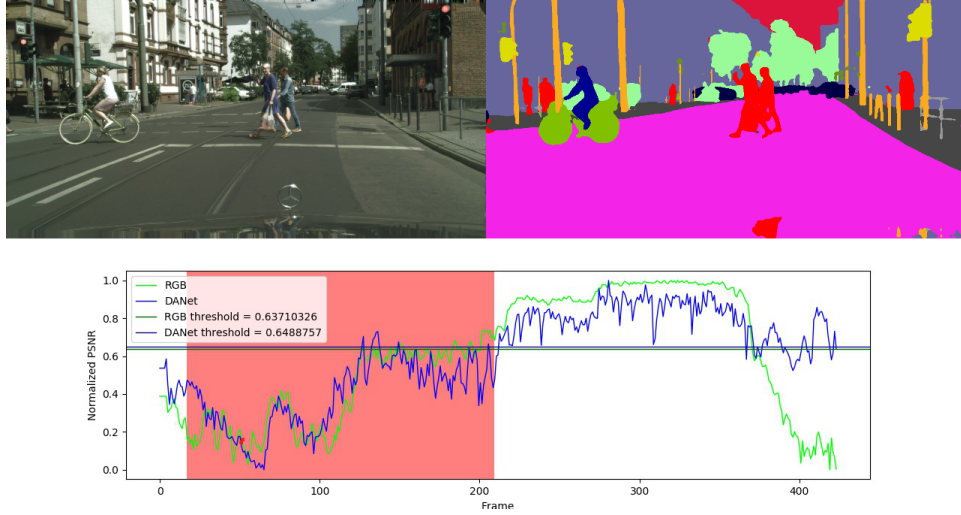


Figure 5.1: Visualization of results on test video 16

5.2 Segmentation Network

DANet was setup on an Ubuntu 16.04 platform running Python 3.6 in an Anaconda environment. It uses PyTorch 1.4 with CUDA 10.1 and cudnn 8 on Nvidia driver 431. As the system runs its own modified version of Pytorch Encoding, these specific versions of each supporting library is necessary for them to cooperate successfully.

5.2.1 Evaluation of pre-trained DANet model

In order to confirm the accuracy of the pre-trained model DANet101 provided by the authors, the supplied test script were used. As found out in the problem analysis, the dataset of most interest for this task is Cityscapes, which the model is able to achieve a pixACC of 95.97% and mIoU of 81.5% on. To test this, the dataset `leftImg8bit_trainvaltest` and ground truth `leftImg8bit_gtFine` were downloaded from the Cityscapes website and placed within the appropriate project directories. As the system use 19 classes for Cityscapes instead of 30, a project named `cityscapesscripts` were used to format the labeling. This system loads the json files with class annotations included in the dataset, translates the classes

through a customized header and then creates the label images described in 3.9. After the ground truth images are created a data formatting script is used to create a text file linking the path of each image with the path of its corresponding ground truth. Running the script with matching input parameters as the paper confirmed the accuracy of the model.

5.2.2 Segmentation script

To use the DANet system to segment the datasets a segmentation script was created. The script loads in sub-directories with RGB frames, creates the folder structure for the segmented images at the designated output path and then segments one sub-directory at a time. When defining the model in the segmentation script two functions are available depending on the choice of model: `get_file()` and `get_segmentation_file()`. `get_file()` is used to load pre-trained models from other systems stored on a webserver. `get_segmentation_file()` is used for custom models using the DANet system. When using the DANet101 model, the settings recommended by the authors for achieving the best results on Cityscapes were used. The predictions for each pixel is output by the `model.evaluate()` function which returns a probability in range 0,1 of each class for every pixel. The final class prediction is found by picking the class with the highest probability.

The segmentation script output two different files: 1) an image with a color palette representing the semantic classification and 2) a text file with the prediction of each pixel for each frame. When trying to input the images to the anomaly detection system, it was found that some images were not written properly to the disc, from the corrupted files not being able to be read by OpenCV. To solve this, a corruption check function were added to the segmentation script running after the segmentation of each directory. In case an image is not able to be read by OpenCV, the path of the image is written to a list, deleted on the disc and then passed through the system again. The check function then runs on the directory again and iterate back and forth in a loop before either all files are written successfully or it reaches a maximum repetition of 10 tries. In case the system fails at segmenting an image completely, the path to the image is written to a text file for manual inspection afterwards.

5.3 Anomaly Detection Modifications

The Future Frame Prediction for Anomaly Detection system were setup on with Tensorflow 1.4.1, CUDA 8.1 and cudnn 6.1 running on a Windows 10 system with an i7 10700k Intel processor and a Nvidia GeForce 1080ti 11GB graphics card.

For testing purposes the anomaly detection system were modified to work with the different data modifications listed in Section 3.10. The most notable changes were made when including the semantic predictions as a fourth channel to the original frames. To do this the following modifications seen in Fig. 5.2 were made to the system:

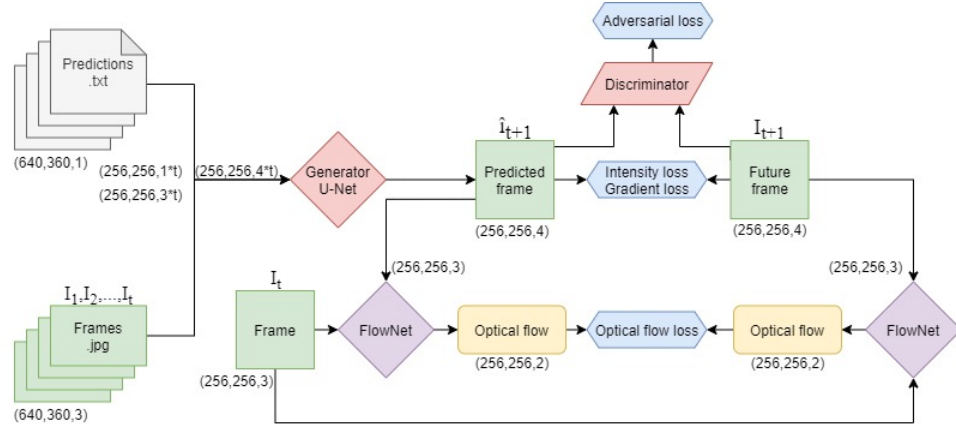


Figure 5.2: The proposed pipeline for including semantic segmentation into anomaly detection system

Before the images are loaded to tensorflow, they are resized to 256x256 pixels using OpenCV's `cv2.resize()` function. For the predictions to maintain the pixel accuracy with the frames they are resized using the same function. To do this, the arrays are reshaped to three axes to conform with OpenCv's data format for grayscale images. After the resizing the predictions are scaled to the range -1,1 following the example of the image scaling.

The input tensor size is controlled by the time step selected when the training and testing. For most of the tests in this project this was set to the default value of four following the recommendations of the paper. As every data point gets loaded as a tensor, the input tensor therefore has the size $(256, 256, 3 \times \text{step})$. When adding the predictions as a fourth channel the size were modified to fit the new channel for three out of the four constraints: Intensity, gradient and adversarial. As analyzed in ?? Each loss function for these constraints are computed in a way which allows for scaling to multiple dimensions without mathematical errors. The optical flow loss however is computed from the difference in optical flow predicted by FlowNet through a pre-trained model FlowNetSD. Since this part of the system use a pre-trained model it is unable to take inputs of a different size. The optical flow loss were therefore only computed from the optical flow of RGB data. As the system weighs the optical flow heavily during training an experiment with changing blue color channel with the semantic predictions was made. The result of which will be

presented in Section ??

5.4 Model training

As described in the paper[13], Avenue does not contain any validation set. To account for this, the system saves a checkpoint of the model at every n cycles of training the generator and the discriminator. In this project a model were set to save after each 1,000st iteration. For this reason, each model were trained and tested sequentially to assure that time were not wasted with overfitting. This was done by adding the inference function from the evaluation script `inference_func()` at the end of the tensorflow session loop in the training script.

The hyper-parameters used for the training were kept the same except for the intensity which were changed from 0.01 to 1 to give more weight to the semantic classes.

Parameter	Description	Value
L_NUM	L1 or L2 loss for intensity	L1
ALPHA_NUM	Power of the each gradient term in gradient loss	1
LAMBDA_ADV	Weight of the adversarial loss	0.05
LAMBDA_LP	Weight of the intensity loss	1
LAMBDA_GDL	Weight of the gradient loss	1
LAM_FLOW	Weight of the flow loss	2
LRATE_G	Range of learning rates for the generator	0.0002-0.00002
LRATE_G_BOUNDARY	Boundary for the generator learning rate	100,000
LRATE_D	Range of learning rater for the discriminator	0.00002-0.000002
LRATE_D_BOUNDARY	Boundary for the discriminator learning rate	100,000

Table 5.2: Main configuration of hyper-parameters used for experiments in the system

5.5 Annotating Cityscapes

Given that the output of the future frame prediction network is a difference score, training and testing on Cityscapes could be done without any ground truth for anomalies as reference. Here, the PSNR scores from the RGB and segmented data could be compared using MSE to test the systems ability to learn general behaviour on the different data. However, testing this theory by splitting the data into a 80/20 split for training and testing yielded nothing of value. At closer inspection of UCHK Avenue it becomes apparent that the training set consists of videos without anomalies, meaning the model is only subject to anomalies during inference. To solve this, the Cityscapes has to be annotated and sorted.

The drive in Frankfurt totals 106,971 frames, which is a lot to annotate manually.

Luckily, the ground truth for the anomaly detection system is binary. It consists of a two-dimensional array with the beginning and end frames of anomalies on each axis. To create this annotation the dataset was loaded into a script frame-by-frame, displayed using OpenCV's `cv2.imshow()` function and annotated with keyboard inputs. To do this, a simple graphical interface, illustrated in Fig. 5.3, was created to keep track of the frame progress and how the current frame is being annotated. The interface allows for single frame annotations on key input for precision when needed, but can also be set to continues mode where the frames play as a video.



Figure 5.3: The interface used for the annotation of Cityscapes

The output of the annotation script is an array with the same length as the dataset with either a zero for normal frames or a one for abnormal frames. The array is then formatted upon completion to fit the ground truth loader in the future frame prediction system which requires a `.mat` file.

As Cityscapes were created as a segmentation dataset, it does not contain any specific normal or abnormal events. Because of this, the anomalies has to be specified. The first annotation was created in regards to the vehicles state of motion. As abnormal events almost exclusively occur as the car is not moving, annotating the frames from this parameter allowed for a coarse definition for sorting the frames to training and testing data. After this, a subset of each dataset were hand picked for the normal and abnormal events. The training data resulted in 49 videos with 30,150 frames selected to be as similar as possible while still exposing the network to scenarios which should not be considered abnormal. These include turns, cars in the opposite lane and parking behind other parks at intersections. Six frames from the training set is presented in Fig. 5.4 below.

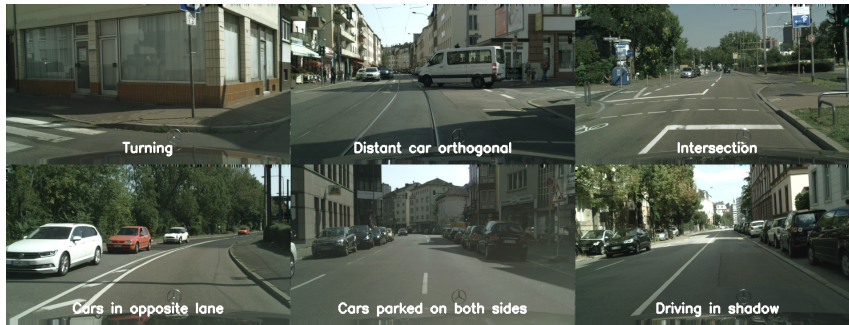


Figure 5.4: Frames annotated as normal

The test set consists of 19 videos of 11,296 frames with defined abnormal events and some normal data for analysis on the model. As seen in Fig. 5.5 The abnormal events are defined as pedestrians crossing the road, cars moving in an orthogonal direction to the camera at close range, bicycles in front of the car, roadwork and shuttle busses.

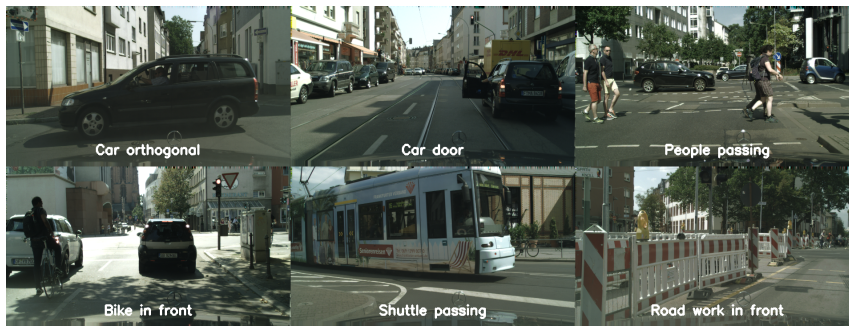


Figure 5.5: Frames annotated as abnormal

Chapter 6

Testing

6.1 Testing Overview

The testing for this project is formulated following the requirements set in the Requirement Specification⁴ about the performance on the two datasets. After each test is described the results of the test will be presented.

6.2 Performance Tests

6.2.1 Tests on UCHK Avenue

Stable PSNR scores for normal events

To test the systems ability to produce stable PSNR scores for normal frames on Avenue the anomalies are removed from the data and the standard deviation of the scores are computed. To test the systems ability to correctly classify the normal frames, the number of true negatives and false positive scores are counted. For visual comparison video nr. 2 were selected as a sample video. The PSNR scores with the anomalies removed for can be seen in Fig. 6.1 below. In this test, the combined data performed slightly worse than the segmented data with a standard deviation of 0.1352 against the 0.1307. The RGB data obtained a standard deviation of 0.09.

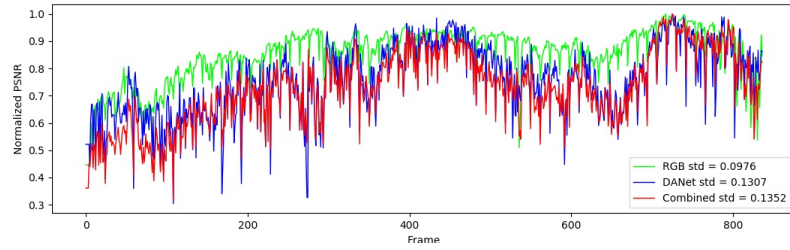


Figure 6.1: PSNR score of normal frames from Avenue test video 2

The scores for the entire dataset can be seen in Fig. 6.2. Here, the RGB data obtained a standard deviation of 0.165, followed by 0.1705 and 0.1736 from the segmented and combined data, respectively.

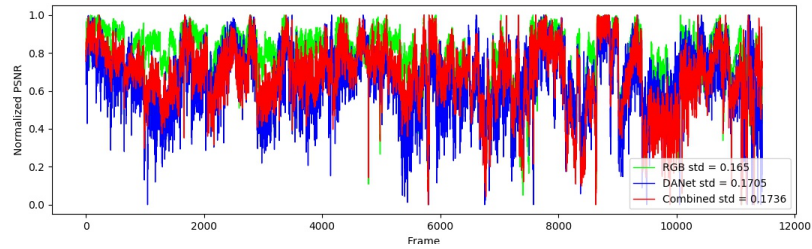


Figure 6.2: PSNR score of normal frames from Avenue

True negative rate higher than 79.33%

The test set contains 11,504 frames annotated as normal. From these frames the benchmark system is able to correctly classify 9,126. The combined model achieves 7,182 and the segmented model achieves 6,211. This gives the RGB, combined and segmented model a true negative ratio of 79.33%, 62.43% and 53.99%, respectively

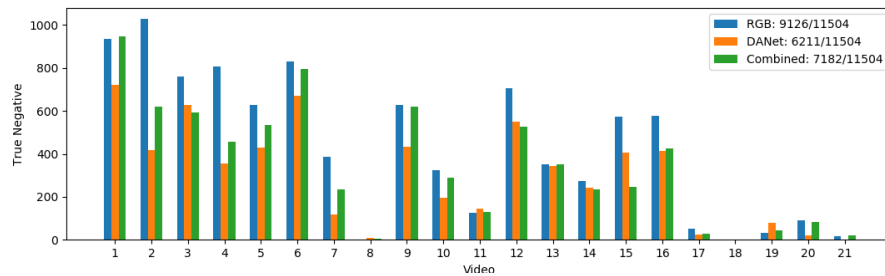


Figure 6.3: The number of true negatives achieved by the three models

True positive rate higher than 79.19%

From 3.7 it was found that the pre-trained model achieved a true positive rate of 79.19% on Avenue. To test this requirements, the segmented and combined model are tested following the same notation as in the analysis.

The results show that the segmented model achieves a true positive rate of 67.30% and the combined model 77.67%. The anomaly specific results for the segmented model can be seen in Table 6.1 and the results for the combined models in Table. 6.2.

Segmented	Running	Direction	Papers	Bag	Close	Kids playing	Camera shake	Total
Abnormal events	9	10	3	12	4	8	1	47
Abnormal frames	377	456	187	1154	743	854	49	3820
Detected events	9/9	10/10	3/3	12/12	4/4	8/8	1	47/47
True positive	293	252	137	823	416	622	28	2571
TPR	0.7771	0.5526	0.7326	0.7132	0.5599	0.7283	0.5714	0.6730

Table 6.1: Anomaly specific results for segmented model on Avenue

Combined	Running	Direction	Papers	Bag	Close	Kids playing	Camera shake	Total
Abnormal events	9	10	3	12	4	8	1	47
Abnormal frames	377	456	187	1154	743	854	49	3820
Detected events	7/9	10/10	3/3	12/12	4/4	8/8	1/1	45/47
True positive	266	295	164	966	587	661	28	2967
TPR	0.7056	0.6469	0.877	0.8371	0.79	0.774	0.5714	0.7767

Table 6.2: Anomaly specific results for combined model on Avenue

A comparison of the true positive rates for the three models are presented in 6.3.

	Running	Direction	Papers	Bag	Close	Kids playing	Camera shake	Total
RGB TPR	0.6844	0.6162	0.984	0.9055	0.8506	0.7026	0.5102	0.7918
Segmented TPR	0.7771	0.5526	0.7326	0.7132	0.5599	0.7283	0.5714	0.6730
Combined TPR	0.7056	0.6469	0.877	0.8371	0.79	0.774	0.5714	0.7767

Table 6.3: Comparison of true positive rates each anomaly in Avenue for the three models

Achieve an AUC of 85% or above on Avenue

To test the requirement for the segmented data to improve the results on Avenue in regards to AUC, a comparison of all the different experiments on the dataset is made. These experiments are:

- Segmentation with DeepLab
- Segmentation with DeepLab with background removed
- Segmentation with DANet

- RGB with DANet predictions
- RGB with DANet predictions with background removed
- RG with DANet predictions

Every system were trained for at least 50,000 iterations and the best model for each system were found by testing every generated checkpoint.

As seen in Fig. 6.4 none of the data manipulations produced better results than the original RGB data. The best performing model from the test were found to be the RGB with predictions added as a fourth channel. This system produced an AUC of 77.46%. The model trained with red and green as the first and second channel and predictions as the third channel is almost the same with an AUC of 77.36%.

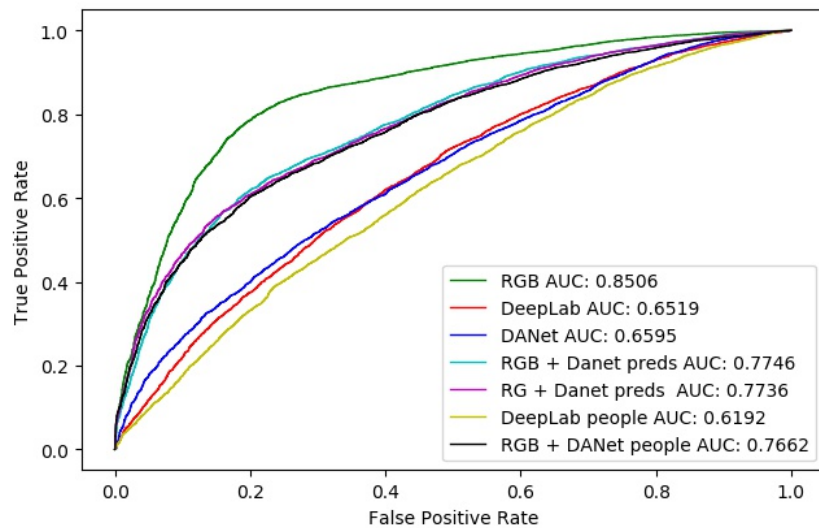


Figure 6.4: ROC curve of all experiments on Avenue

Even though the mixed data performed better than the fully segmented data, adding the semantic predictions to the RGB data dropped the AUC of the system from 85% to 77%. The ROC for DANet and the combined data is presented in Fig. 6.5.

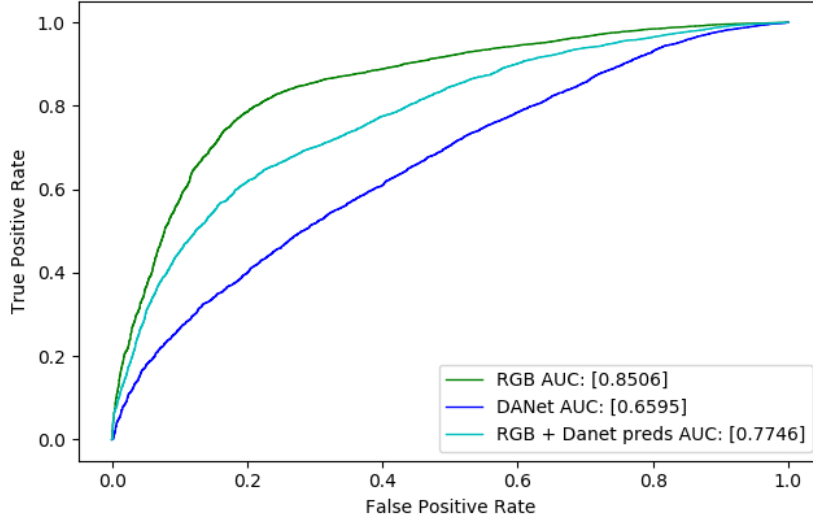


Figure 6.5: ROC curve of RGB, DANet and combined data

6.2.2 Tests on Cityscapes

Learn general behaviour of a moving scene

To test if the model is able to learn the movement of the car, both the RGB data and segmented data is evaluated. This is done by analysing the behaviour of the PSNR score at specific interest points of two videos. For the ability to learn the general behaviour of the scene in motion video 3 in the test set is used. This video is included in the set because it does not have any annotated anomalies and it contains some important scenarios such as driving in an open area, turning, driving past a queue of cars in the opposite lane and driving in shadow. The results on the video can be seen in Fig. 6.6 below. As seen, the PSNR scores fluctuate throughout the video, but stays above the threshold the majority of the drive showing that the model can correctly classify the frames. The significant drop in the segmented model is due to a black frame in the dataset.

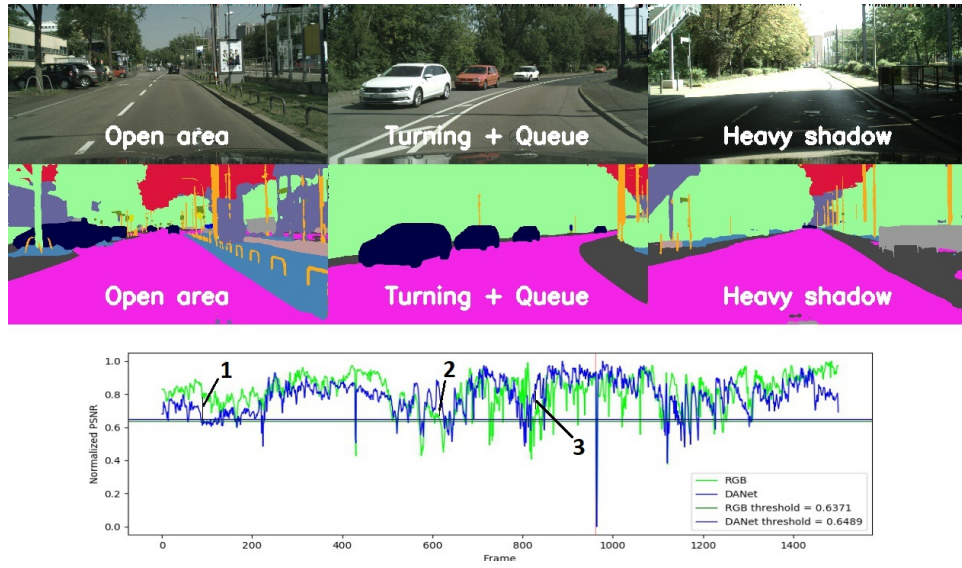


Figure 6.6: PSNR score and normal frames from video 3 in Cityscapes test set

The first image is highlighted as it contains a lot of classes in the segmentation. Because of this, the RGB data performs a little better than the segmented PSNR scores, which are on the border of the cut off. The second image is an important case in the dataset with both a light turn and cars in the opposite lane. Here, both models fluctuated as a result but the segmented data performed better overall. The third image is included to show how the segmented data performs better in changing light conditions. As the segmentation model takes care of the intensity changes, the segmented data is unaffected by the heavy shadow.

Detect anomalies defined in the annotation for both data types

To show the models ability to detect anomalies, video 14 is used. This video contains both movement, acceleration, deceleration and significant anomalies from pedestrians and bikes. The results for the video can be seen in Fig. 6.7. The first image illustrates how the segmented data is more sensitive to objects such as bikes, with the lower score from the frame. The trade-off to the sensitivity is how the RGB data is more stable when the car is not in motion and the camera is stationary, illustrated in the second image. The third image is the most important as it shows the segmented models low sensitivity to acceleration and deceleration compared to the RGB data.

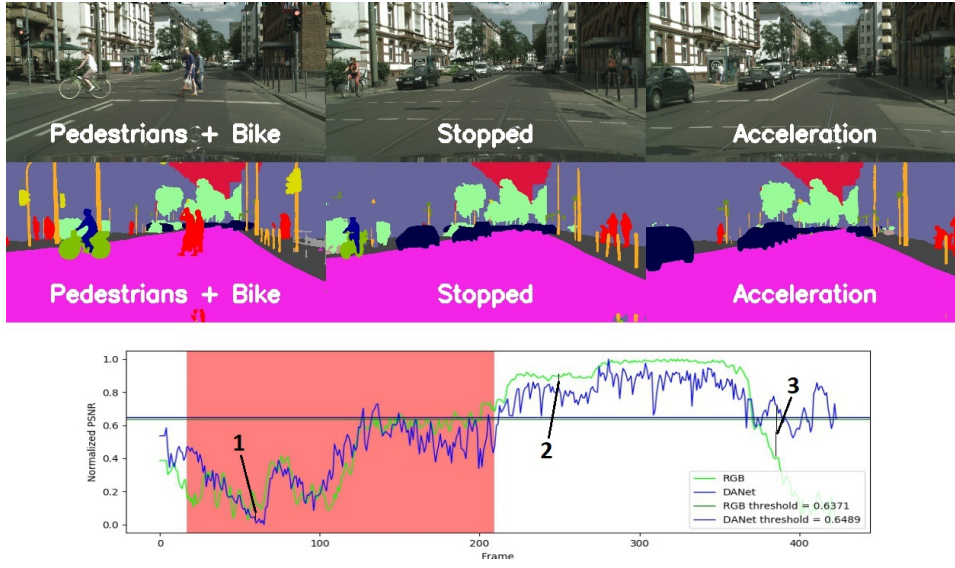


Figure 6.7: PSNR score and normal frames from video 14 in Cityscapes test set

Stable PSNR scores for normal events

To test the stability of the PSNR scores for the normal frames, the same method applied to Avenue is used. The Fig. 6.8 below illustrates the scores of the normal frames across the test set.

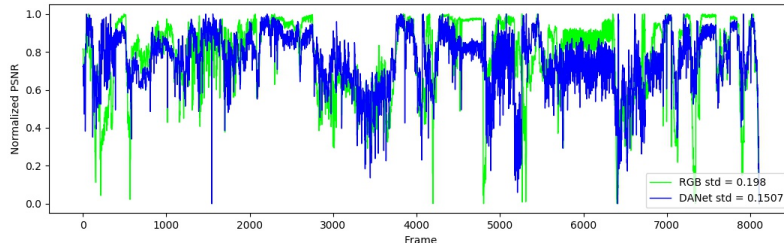


Figure 6.8: PSNR score for all normal frames in Cityscapes test set

The PSNR scores for all the normal frames in the test set resulted in a standard deviation of 0.198 for the RGB data and 0.1507 for the segmented data.

A higher true negative rate for segmented data than with RGB

The Cityscapes test set contains 8,146 normal frames. The classification for the two models classified 6,256 true negatives for the RGB data and 6,473 true negatives for the segmented data. This results in a true negative rate of 76.8% for RGB and

79.5% for segmented. The true negative results for each video can be seen in Fig. 6.9.

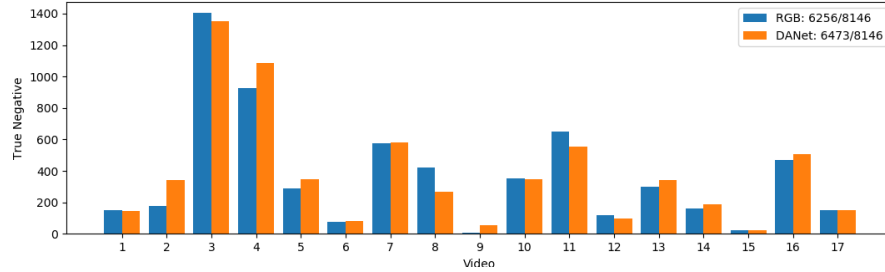


Figure 6.9: The number of true negatives achieved by the RGB and segmented data

A higher true positive rate for segmented data than with RGB data

For true positive classification, there are 3,119 frames in the test set. Here, the RGB model obtained 2,232 and the segmented data 2,383 true positive classifications. This results in true positive rates are 71.56% for RGB and 76.4% for segmented.

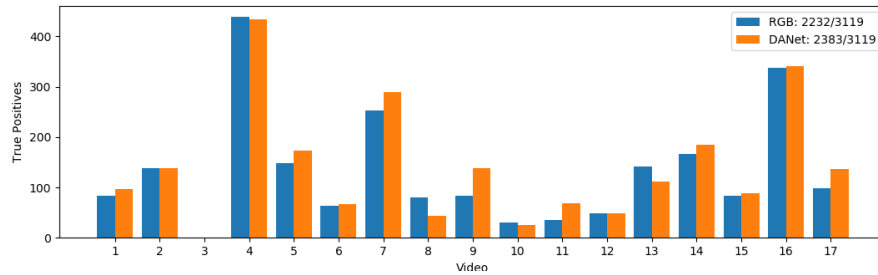


Figure 6.10: The number of true positives achieved by the RGB and segmented data

Produce better AUC for the segmented data than with RGB data

To test the overall performance on Cityscapes the ROC curve and AUC were computed following the ground truth annotations. For this test, both systems were trained for 50,000 iterations with the same hyper-parameters.

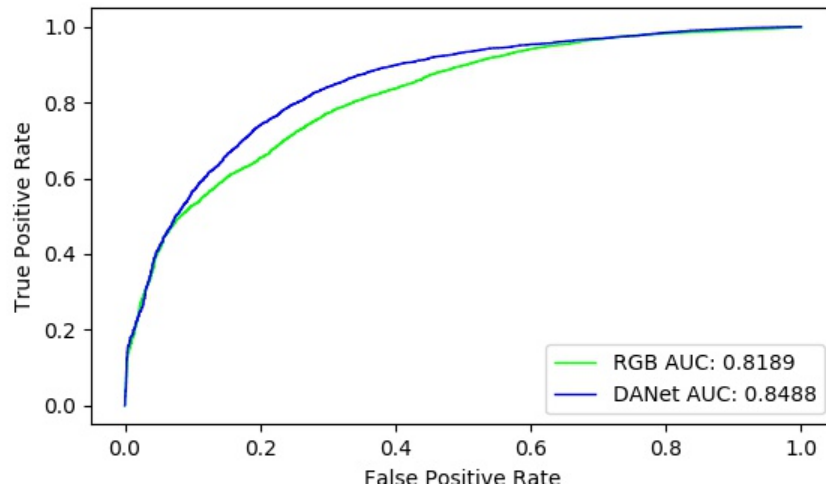


Figure 6.11: ROC curve of RGB and segmented Cityscapes model

As can be seen in Fig. 6.11 the segmented data performs better than the RGB data on Cityscapes with an AUC of 84.88% against 81.89%. The best AUC of the RGB data was achieved at checkpoint 46,000 and the best for the segmented data at 38,000.

Chapter 7

Discussion

7.1 Avenue

The results from the tests on Avenue show that the implementation of semantic segmentation in the future frame prediction system relies heavily on the accuracy of the segmentation. Looking at the anomaly specific detection rates for the three models in Table 6.3, it shows that some anomaly predictions benefited from the added semantic information. Specifically, the anomalies involving people, such as the running man, the kids playing and the people walking in the wrong direction. Here, the segmentation models outperform the RGB in true positive rate. Where the segmented data falls apart is in the normal frames. Because of the unstable prediction accuracy the model trained on DANet only achieved a 53.99% accuracy in true negative rate and the combined model consisting mostly of RGB data achieved 62.43%.

The drop in AUC for the segmentation models also stems from the inability to detect and segment the specific objects in the scene being thrown. The original model achieves great results on these anomalies in comparison to the other anomalies with a TPR of 98%, 91% and 85%, whereas the segmented model is only able to predict 73%, 71% and 56%, respectively.

7.2 Cityscapes

Looking at the results from the tests on Cityscapes, both RGB and segmented data is able to learn the motion behaviour of the car and produce consistent true negatives. When analysing the visualizations of the PSNR graphs, it becomes clear though that the better performance from the segmented data comes from the models reduced sensitivity to the cars change in movement. Specifically, the car turning, accelerating and decelerating, which has been defined as normal behaviour in the

annotation. These scenarios are consistently classified incorrectly by the RGB system. As the environment in the scene have been reduced to single value surfaces in the segmentation, the changes in the cars movements does not produce as big a prediction difference as with the pixel detailed RGB data. As one of the potential problems with applying the future frame prediction system to the Cityscapes dataset is the change from stationary to moving scenes, this is considered a benefit of the segmented data.

Another strength of the segmented data is the sensitivity to the different objects. During the annotation of the dataset a special effort were made to identify and classify every scenario where bikes and pedestrians are directly in front of the car, as these are considered important anomalies for a real life application of the system. Bikes in particular functioned well with the segmented data. As bikes in many cases blend with the background in the RGB frames and take up little space in the scene, the RGB system did not react as strongly to these scenarios as the segmentation system. A disadvantage of the sensitivity is a higher chance false positives when a bus enters the scene outside the perimeter which would define an anomalous event.

It should also be noted that the difference between the cut off threshold between the two models trained on Cityscapes are only different by 0.012 with 0.6489 and 0.6371. Even though local differences exists between the performance of the models, the global cut off defining the classification boundary from the fluctuations of the graphs totals the same across the systems. This becomes more evident when comparing the thresholds with that of Avenue which is 0.6628. This shows there exists some system in the PSNR scores generated by the future frame prediction system which translates universally across anomalies behaviour for determining what is normal and abnormal.

Chapter 8

Conclusion

This project set out to investigate the feasibility of utilizing semantic segmentation in anomaly detection. The task was analyzed and it was concluded in the problem analysis that to account for missing ground truths across the frameworks, the research should be conducted on both an anomaly detection dataset and a semantic segmentation dataset. By analyzing the different datasets through the scope of applying them to the opposite framework, UCHK Avenue and Cityscapes were concluded to be suitable for the project.

For anomaly detection, Future Frame Prediction was chosen as it was concluded that it was the system with the best accuracy which also benefit the most from the global context produced by the segmentation. It was also concluded that the constraints utilized in training the network is scalable to account for experiment on combined data.

From the segmentation model test on the anomaly detection dataset, it was concluded that segmentation models are able to segment datasets with similarity in objects to those in the intended dataset to a reasonable degree. It was also concluded that to achieve segmentation results with enough precision acceptable for the application, the models would have to be trained on the targeted dataset.

Following the analysis specific requirements were formulated and a problem statement was set:

Can semantic segmentation be used to improve future frame prediction for anomaly detection?

To answer the problem statement a solution for the two datasets were created and tested. From the tests on Avenue it was found that the low quality in the segmen-

tation made the future frame prediction system unable to perform with the same global accuracy as with the original RGB data. Looking at the anomaly specific results in the discussion it was concluded that anomalies with correct segmentations benefited from the semantic segmentation while anomalies with incorrect segmentation significantly reduced the accuracy.

From the testing on Cityscapes it was concluded that frame prediction can be utilized for moving scenes as well as stationary. It was also concluded that the annotated anomalies in the dataset are defined enough to produce systematic results in the anomaly detection.

From the results on Cityscapes it was concluded that segmented data with a mIoU of 81.5% or above retain enough information to create low reproduction error for normal events and fluctuations in abnormal events at the same quality of RGB data. It was also concluded that semantic segmentation is able to produce better accuracy than RGB data in an anomaly detection setting given certain criteria,

From the results of the tests the final problem statement can be validated and it can be concluded that semantic segmentation is feasible for utilization in an anomaly detection.

Bibliography

- [1] Antoni Chan and Nuno Vasconcelos. “Ucsd pedestrian dataset”. In: *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)* 30.5 (2008), pp. 909–926.
- [2] Marius Cordts et al. “The cityscapes dataset for semantic urban scene understanding”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 3213–3223.
- [3] Sikkerhedsbranchen DK. *TV Overvågning - Ofte stillede spørgsmål*. FAQ. PCO AG, 2020. URL: <https://www.sikkerhedsbranchen.dk/wp-content/uploads/2018/12/TV0-FAQ-1.pdf> (visited on 2020-05-03).
- [4] Alexey Dosovitskiy et al. “Flownet: Learning optical flow with convolutional networks”. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 2758–2766.
- [5] *Engineering Computer Science - Google Scholar Metrics*. https://scholar.google.com/citations?view_op=top_venues&hl=en&vq=eng. ©2021.
- [6] Weiguo Feng, Rui Liu, and Ming Zhu. “Fall detection for elderly person care in a vision-based home surveillance environment using a monocular camera”. In: *signal, image and video processing* 8.6 (2014), pp. 1129–1138.
- [7] Mahmudul Hasan et al. “Learning temporal regularity in video sequences”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 733–742.
- [8] Zhe Hui Hoo, Jane Candlish, and Dawn Teare. *What is an ROC curve?* 2017.
- [9] Alain Hore and Djemel Ziou. “Image quality metrics: PSNR vs. SSIM”. In: *2010 20th international conference on pattern recognition*. IEEE. 2010, pp. 2366–2369.
- [10] Eddy Ilg et al. “Flownet 2.0: Evolution of optical flow estimation with deep networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 2462–2470.

- [11] Radu Tudor Ionescu et al. "Object-centric auto-encoders and dummy anomalies for abnormal event detection in video". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 7842–7851.
- [12] V. Kumar. *Computer Vision Marketing Analytics*. <https://www.grandviewresearch.com/industry-analysis/computer-vision-market>. ©2020.
- [13] Wen Liu et al. "Future frame prediction for anomaly detection—a new baseline". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 6536–6545.
- [14] Cewu Lu, Jianping Shi, and Jiaya Jia. "Abnormal event detection at 150 fps in matlab". In: *Proceedings of the IEEE international conference on computer vision*. 2013, pp. 2720–2727.
- [15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation". In: *International Conference on Medical image computing and computer-assisted intervention*. Springer. 2015, pp. 234–241.
- [16] Bolei Zhou et al. "Scene parsing through ade20k dataset". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 633–641.
- [17] Arthur Zimek and Erich Schubert. "Outlier Detection". In: *Encyclopedia of Database Systems*. Ed. by Ling Liu and M. Tamer Özsu. New York, NY: Springer New York, 2017, pp. 1–5. ISBN: 978-1-4899-7993-3. DOI: 10.1007/978-1-4899-7993-3_80719-1. URL: https://doi.org/10.1007/978-1-4899-7993-3_80719-1.