SUDDEN CHANGE DETECTION ON ROADS USING MULTIPLE IMAGE PROCESSING ON AUTOENCODER

by

Pongsaton Mondee

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Engineering in Mechatronics

Examination Committee:	Dr. Mongkol Ekpanyapong (Chairperson) Prof. Manukid Parnichkun Prof. Matthew N. Dailey
Nationality:	Thai
Previous Degree:	Bachelor of Engineering in Mechanics Sirindhorn International Institute of Technology Pathum Thani, Thailand
Scholarship Donor:	Royal Thai Government Fellowship

Asian Institute of Technology
School of Engineering and Technology
Thailand
July 2021

AUTHOR'S DECLARATION

I, Pongsaton Mondee, declare that the research work carried out for this thesis was in accordance with the regulations of the Asian Institute of Technology. The work presented in it are my own and has been generated by me as the result of my own original research, and if external sources were used, such sources have been cited. It is original and has not been submitted to any other institution to obtain another degree or qualification. This is a true copy of the thesis, including final revisions.

Date: July 2021

Name (in printed letters): PONGSATON MONDEE

Signature: Pongsaton mondee

ACKNOWLEDGEMENTS

I am grateful to everyone who has supported and encourages me in the completion of this Master thesis.

Firstly, I would thank my advisor Dr Mongkol Ekpanyapong for his guidance and equipment support for this thesis study. I also would like to thank my committee members Prof. Manukid Parnichkun and Prof. Matthew N. Dailey for helping me clarify the scope of the research and insightful comments.

Secondly, I also would like to thank my friend Mr Siraphop Prasertprasasna for all other technical help and I especially thank Mr Akekalak Tungrattanavalee for his participation and help me collecting the data.

Finally, I express my gratitude to my family for their unfailing support and constantly encourage me throughout this Masters. I also express my profound gratitude to the Royal Thai Government (RTG) fellowships for the scholarship. My work would not have been complete without them.

ABSTRACT

Nowadays, most road accidents are caused by drivers more than the vehicles and environmental conditions combined. One solution is to reduce the number of human decisions in driving by using the autopilot system or driving assistant to help driving more secure. In order to make the decision, autonomous driving vehicles need to be aware of the surrounding moment for calculating and determine to drive precisely and safely. This thesis focuses on computer vision fields with the ability to interpret images of the environment around the car. Instead of using a LiDAR scanner, this using the front car's camera and autoencoder neuron network technologies combined can also be used to identify any anomaly moment that occurs.

In this study, the score is used to indicate anomaly events in the footage. An anomaly score is created by the fact that an abnormal moment occurs infrequently when the point of the system is to learn to reconstruct the image sequences by using an autoencoder network implement with Conv-LSTMs. The anomaly moment will get a greater error from reconstructing sequences. The system is also equipped with 2 subs networks for transform raw images into semantic segmentation images and dense optical flow images. This preprocess is for reducing the complexity of the images before sending them to the Conv-LSTMs autoencoder.

To make a system that can interpret anomaly moments on the road, many parameters are trained and tested by the system such as input image size, sequences, number of color channels, and other parameters. The system is also unsupervised trained between images from the city and the rural scenarios. From the test results, the best system of road anomaly detection is quite well detecting the anomaly events on the road in the three different tests set, city dataset, rural dataset, and Extest dataset with the best F-score of 64.12%, 63.53%, and 66.67 % correspondingly and the overall score is 62.12 %. The best system is made by using HD resolution as input images in the preprocess network and then reducing the sequenced image's dimension down to 256x256 pixels with 1 color channel in segmentation while has 3 color channels in optical flow for the Conv-LSTMs network. In the calculation of the anomaly event process, the anomaly is computed using the mahalanobis distance on a reconstruction score from the 2 different models, dense optical flow, and segmentation with thousand steps and thousand initial points. The network uses around 7-10 minutes in order to process a 1-minute-long video and can detect sudden changes around every 15-20 seconds.

CONTENTS

LIST OF FIGURES

Figures **Page**

LIST OF TABLES

Tables **Page**

LIST OF ABBREVIATIONS

AE	= Autoencoder
CNNs	= Convolutional Neural Networks
Conv-LSTMs	= Convolutional Long Short-Term Memory
ERM	= Error Refinement Module
FCN	= Fully Connected Network
GANs	= Generative Adversarial Networks
LR-ASPP	= Lite Reduced Atrous Spatial Pyramid Pooling
MemAE	= Memory-Augmented Autoencoder
NAS	= Network Architecture Search
PCM	= Predictive Coding Module

CHAPTER 1

INTRODUCTION

1.1 Background of the Study

In the 21st century, passenger cars are becoming an important mode of transportation of residential and commercial because cars are flexible and convenient for communication. Moreover, cars can access most of the areas and do not need expensive infrastructure to be installed. On the other hand, rail transportation such as trains or subway must be on track or build the tunnel. According to statistical data from Statista, there are 92 million worldwide automobiles were produced in 2019. While in 2010, 77 million automobiles were produced, and this number keeps increasing every year.

Road accidents are one of the problems with the increase in automobiles on the road inevitable. According to the study from the U.S. Department of Transportation, most road accidents are caused by drivers more than the vehicles themself or even by the environment. The solution to reducing the number of road accidents is minimal people involved in driving or no people involved in driving. One of the solutions is using autopilot or pilot assistant to help drivers make decisions or drive more safely. Artificial intelligence is one of the watchful technologies of this decade. To develop the autopilot or pilot assistant, it would be equipped with a central computer system and many sensors around the vehicles to improve obstacles detection and create automobiles that can understand an environment before making any driving decisions on behalf of humans.

1.2 Statement of the Problem

Autonomous driving vehicles need to be aware of the surrounding environment of the car in order to calculate and determine to drive precisely and safely. One of the solutions that Autonomous driving cars commonly used in environmental awareness and understanding is the installation of radar or laser scanners to detect the movement of other vehicles surrounding the cars and obstacles while driving. Which is highly accurate and can be used in all weather and times condition, but it comes with the expensive and complicated equipment setting.

Computer Vision is one of the most popular artificial intelligence research fields in recent years. Which it can manage how the computer understands information from digital images, video and understanding visual systems in the same way that humans understand and interpret images. With the ability of computer vision technology, this technology can be used to interpret images and videos of the environment around the car or the front of the car to enable the car to understand the environment from images and videos. The system can be used by the camera only to understand the environment or an event, and it is less complex and cheaper than using radar technology to help the cars make decisions. By using camera and autoencoder neuron network technologies combined can also be used to detect any anomaly events that occur.

In the modern-day, autonomous vehicles can use some algorithm for detecting risky events through computer vision based. This crucial role not only gives a better understanding of the front-end environment but can also improve driving performance and reduces the burden for drivers on long-distance driving.

1.3 Objectives of the Study

The main purpose of this research is to design the system for detection of any events that are considered to be the anomaly that happens on the road from moving front camera images sequence or the event that needs the driver's attention before the accidents occur.

- Make a system that can mimic human driving behavior through awareness of the surrounding especially the moment that needs a driver or car attention base on images processing network by using an autoencoder, Conv-LSTMs
- Make a system for road anomaly detection that can detect things that obviously abnormal such as the car went off the road, the fast-moving nearby objects, or the obstacles with an overall score higher than 70%

1.4 Scope and Limitations

The scope of this research is to make a neuron network base on an autoencoder network, especially Conv-LSTMs that can detect anomaly events on the road such as detection the fast-moving cars toward the camera, pedestrians, or dogs suddenly across the road, the car went off the road, obstacles or any other anomaly that can lead to accidents. The data for the detection and training process comes from the car's front camera mount on the window point outward and receives the same point of view as the driver. The output from the network is the anomaly score interpreted from the events in the video.

CHAPTER 2

LITERATURE REVIEW

2.1 Anomaly Detection

Anomalies events in videos are usually described as events or sudden motions of the objects in the scenes that occur rare, unnatural, or can be some events that need attention. The purpose of anomaly detection is to localize both specific temporal sequences and spatial pixels in video sequences. Temporal localization is starting from the first frames of the anomaly event until the end. It is referred to as frame-level detection. Spatial localization is referred to recognize every corresponding pixel contained in that frame and can refer to the anomaly event. This is referred to as pixel-level detection. Anomaly detection in videos is an achievement for many researchers to study for a decade, while this problem is still hard to figure out due to the complexity of modeling that can capture rare events and the scarcity of data itself. Recognizing anomaly events from other regulars not only requires an understanding of complex spatial patterns but also requires an understanding dynamic of temporal relationships.

Real-world anomaly events, especially on the road that occur rarely and happen suddenly. It is tough for any system that can detect all types of road anomalies. Despite these facts, cars crash footage is still easy to obtain from the internet or on social media and also comes from public security cameras. There are still not cover all possibilities of an anomaly event on the road. Furthermore, most of the footage from the internet has low resolution and is difficult to control their consistency so the Unsupervised training method is introduced. Unsupervised methods are one of the methods that are capable of detecting anomaly events using only regular footage in the training set. Although the unsupervised methods are still unable to accomplish satisfying performance on real-world scenarios, they are considered to have more flexibility to capture rare events patterns.

In recent years, deep learning methods have been developed until they are able to take advantage of big data and powerful computation devices combined with unsupervised anomaly detection techniques. This makes deep learning methods conquer the anomaly detection scope. One of the successful networks based on deep learning techniques is a deep autoencoder, there are several proposed based on this network.

In the year 2016, Hasan et al. propose an autoencoder-based detection that can capture and identify the anomaly from multiple scenarios. A fully convolutional feed-forward autoencoder can learn the latent features and classifies regular patterns. The trained model predicted regularity score based on the reconstruction error from reconstructed sequences. The reconstruction error is low in regular motion sequences and but will get a high reconstruction cost in unusual moments. (Mahmudul Hasan, Jonghyun Choi, Jan Neumann, Amit K. Roy-Chowdhury, Larry S. Davis, 2016)

Gong et al. propose the developed version of the deep autoencoder by adding a memory module called the memory-augmented autoencoder or MemAE. This module working as memory storage for the reconstruction process after receives the encoded data from the encoder. In the training, the memory content inside the module will get an updated approach to represent the regular data. In testing, the memory content inside the module will freeze and use for reconstruction data from memory records that now represent the regular patterns. (Dong Gong, Lingqiao Liu, Vuong Le, Budhaditya Saha, Moussa Reda Mansour, Svetha Venkatesh, 2019)

Ionescu et al. propose an anomaly detection network based on training convolutional autoencoders on top of an object detection network. The results are concatenated between motion and appearance latent information. They are clustered in the training process by applying k-means clustering and are classified in the inference phase. The sample will be considered irregular when the classification score is negative, or the result is not associated with any class. (Radu Tudor Ionescu, Fahad Shahbaz Khan, Mariana-Iuliana Georgescu, Ling Shao, 2019)

To better understand the temporal relationship within a video, a combination of FCN and LSTM (long short-term memory) is proposed. Yong Shean Chong et al. propose a spatiotemporal network for detecting anomalies in videos. Based on convolutional LSTMs autoencoder (ConvLSTM-AE), this model is equipped with two main components, the first is for spatial features, and another one for learning the temporal progression or the spatial features that make it capable detects normal appearance and motion patterns simultaneously. Luo et al. also propose a combination network between FCN and LSTM as a ConvLSTM-AE to better model the temporal correlation and further improve the performance of the autoencoder framework. ConvLSTM-AE can detect the regularity of appearance and motion for regular moments. (Yong Shean Chong, Yong Haur Tay, 2017) (Weixin Luo, Wen Liu, Shenghua Gao, 2017)

Shi et al. propose the use of autoencoder-based architecture for a forecast rainfall intensity in the region over a short period by implementing the power of stacked convolutional LSTM layers (Conv-LSTMs). This model can beat the state-of-the-art, ROVER algorithm and FC-LSTM network for precipitation nowcasting. (Xingjian, Shi Zhourong, Chen Hao Wang, Dit-Yan Yeung, 2016)

Medel et al. propose the other use of autoencoder-based architecture for anomaly detection by using a special regularity evaluation algorithm at the model's top and the Convolutional Long Short-Term Memory (Conv-LSTMs) module. This model can understand the regular temporal patterns in videos and the progression or movement of the objects in the images sequence. (Jefferson Ryan, Medel Andreas Savakis, 2017)

Instead of directly computing regularity scores based on the reconstruction of current frames, another trend is to reconstruct the future frames based on the current frames. The reconstructed error is then computed based on the difference between predicted future frames and the real future frames. The model that is capable to do this task is GANs or Generative Adversarial Networks. It consists of 2 main components; the discriminator module is trying to distinguish between the predicted future frames and the actual ones while a generator module is trying to render predicted future frames close to the actual future frames as much as possible.

Liu W et al. proposes Generative Adversarial Network (GAN) based network for anomaly detection. In this network, U-net is used as a generator to generate a predicted future frame.The spatial constraints on gradient and intensity is a commonly used method however, they also propose the use of motion or temporal constraint in sequence prediction by implementing the optical flow between predicted frames and actual frames. Pretrained Flownet is responsible for this task and the optical flow between those two should be consistent. (Wen Liu, Weixin Luo, Dongze Lian, Shenghua Gao, 2018)

Ye et al. propose a brand-new network based on both reconstruction and prediction methods in an end-to-end framework that shares the related architecture with U-Net architecture except for the last layer that implements the ERM module, AnoPCN. This network introduces 2 main modules, a predictive coding module (PCM) and an error refinement module (ERM). PCM is a convolutional recurrent neural network equip with feedback connections for sending predicted frames and feedforward connections for sending the errors. ERM is used to reconstruct the prediction and also sharpening it. (Ye Muchao, Peng Xiaojiang, Gan Weihao, Wu Wei and Qiao Yu, 2019)

M. Sabokrou proposes other more efficient techniques a deep anomaly that is based on the pure power of convolutional neural using trained fully modified convolutional neural networks (FCNs). The pre-trained supervised FCN is turned into an unsupervised FCN for detecting anomalies in frames. FCN-based on 2 main responsibilities, feature representation, and outlier detection. After extracting the regional features, In the later stage 2 Gaussian classifier is used for labeling them. (M. Sabokrou, M. Fayyaz, M. Fathy, Z. Moayed, and R.Klette, 2017)

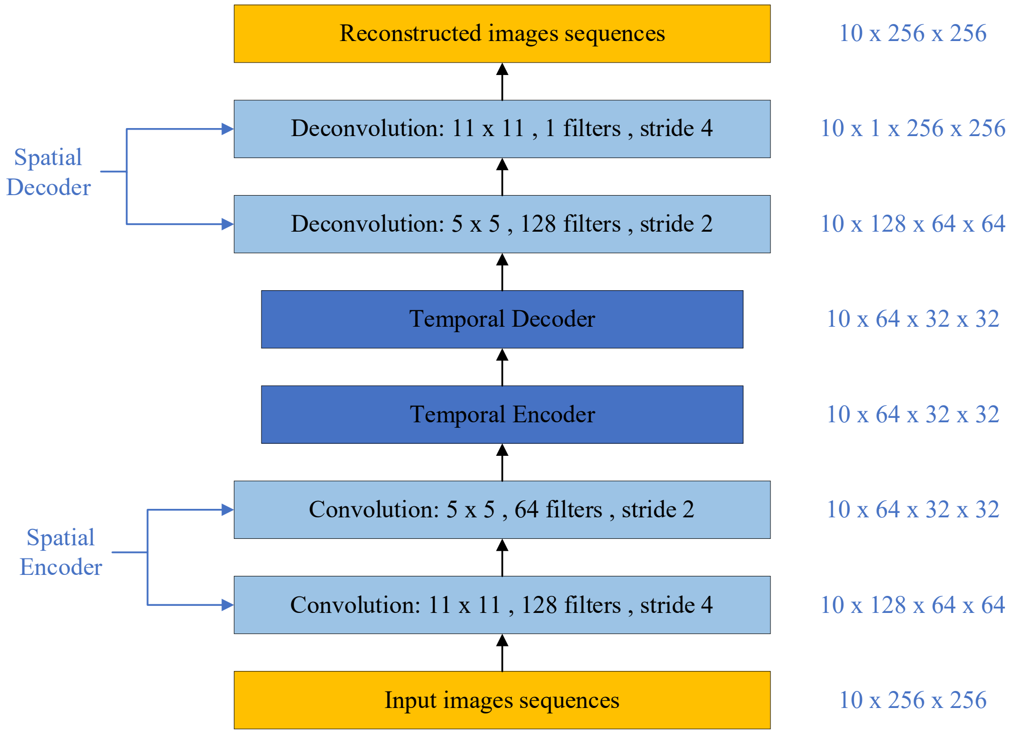
Ryota et al. also propose a network based on the power of CNN that can define the events in the human-understandable style by combining joint detection and recounting of anomaly moments in videos. Recounting is used for describing why these events are supposed to be abnormal to help human observers quickly determine and understand if they are false alarms or not. In this paper, a general CNN model and environment-dependent anomaly detectors have been integrated for learning multiple visual tasks that are useful for recognizing and recounting anomaly events. (Ryota Hinami, Tao Mei, and Shin'ichi Satoh, 2017)

2.1.1 Conv-LSTMs Encoder-Decoder

Conv-LSTMs Autoencoder is an autoencoder that combines convolutional layers with recurrent LSTM layers together. Convolutional layers are a popular network for their superior performance in image recognition, while LSTM is a recurrent layer that is generally used for sequence learning and detect long-term temporal relationships. They have proved performance in many applications such as text-to-speech conversion and handwriting recognition. It can be separated into 2 main categories, the spatial and the temporal extractor module. (Yong Shean Chong, Yong Haur Tay, 2017)

Figure 2.1

Shows an Overall Conv-LSTMs Autoencoder Network



Note. Adapted from "Abnormal Event Detection in Videos using Spatiotemporal Autoencoder," by Yong Shean Chong, Yong Haur Tay, *arXiv preprint arxiv:1701.01546,* 2017.

The spatial extractor modules consist of stacked convolutional neural networks. It accepts video input in a sequence of reshaped frames in chronological order. The extractor reshapes and downsamples the video input into a stack of features vectors. The series of frames will lose some information in this process in order to extract characteristic features from the input images. A convolutional network can capture and store the latent patterns of these features depending on the number of filters layers during the training process. More filters are involved in more image features getting extracted and the better for the network for recognizing hidden patterns in images. However, more filters would result in increased computation time, so the number of filters needs to be adjusted.

The temporal extractor modules consist of a stacked Convolutional Long Short-term Memory (Conv-LSTMs) model that was introduced by Shi et al. Conv-LSTMs is one variant of the LSTMs architectures. This is a modified version from fully connected LSTM (FC-LSTM). It uses the convolution network instead of matrix operations. This allows Conv-LSTMs to work by propagating each spatial characteristic temporally through each Conv-LSTMs layer and result in better work with image sequences. Moreover, this modified convolutional also attaches with an optional shortcut connection to allow the network to acquire former information better.

These make Conv-LSTMs lighter and yield a better result for extracting spatial feature maps from images. The difference between using the Conv-LSTMs layer and traditional convolutional neural networks is the output of each Conv-LSTMs layer can directly feed into the next layer, instead of using max-pooling layers. The Conv-LSTMs model related formula can be summarized :

$$
\begin{gathered}
f_t = \sigma(W_f * [h_{t-1}, x_t, C_{t-1}] + b_f) \\
i_t = \sigma(W_i * [h_{t-1}, x_t, C_{t-1}] + b_i) \\
\hat{C}_t = tanh(W_C * [h_{t-1}, x_t] + b_C) \\
C_t = f_t \otimes C_{t-1} + i_t \otimes \hat{C}_t \\
o_t = \sigma(W_o * [h_{t-1}, x_t, C_{t-1}] + b_o) \\
h_t = o_t \otimes tanh(C_t)
\end{gathered}
$$

Note. These equations share the similarity to equations that use for LSTMs. Instead of using weights for every connection, Conv-LSTMs uses convolutional filters.
(The symbol $*$ denotes a convolution operation)

2.2 Optical Flow using Deep Learning.

To analyze real-time video, most implementations of these techniques only use relationships between the objects within the same frame or only in spatial information, not including temporal information. In short, each frame is processed independently. However, in real-world circumstances, video is recorded the images sequences in a specific temporal resolution in frames per second. This indicates that information in a video is not only encoded independently in the same frame but also sequentially relate to the other frames in a specific order. We need to consider the relationships between sequential frames to give us a better understanding of motion or to recognize and classify actions in the events.

The idea of Optical flow has been proposed since the 1980s in the form of handwriting approaches based on brightness constancy. They assume pixel brightness is approximately constant without regard to their movement and try to determine how the brightness of the pixels moves across the screen over time. On the other hand, they try to estimate optical flow displacement vectors. In short, if the original pixel position applies with the displacement vector, the next pixel position appears. Optical flow is the motion extraction of the objects that relate to the movement between the objects in the different frames. Novel researchers are now focusing on applying deep learning to the Optical flow and they have shown satisfying results. In general, Optical flow approaches take two video frames as input and output the optical flow data in the color-coded image. Processing optical flow approaches with deep learning is now a popular topic with variant networks such as FlowNet, SPyNet, PWC-Net, and some outperforming one another on several benchmarks. (Lin, 2019) (Gituma, 2019)

FlowNet is proposed by Smagt et al. based on the general U-Net architecture and convolutional neural networks (CNNs) for solving supervised flow estimation problems. FlowNet consists of 2 modules, FlowNetSimple (FlowNetS) and FlowNetCorr (FlowNetC). FlowNetSimple is used to extract the motion information from feeding images pair through a stacked convolutional neural network (CNN). FlowNetCorr is used for determining the correlates feature vectors at different image locations by creating two identical images stream and recombine them at the refinement stage. (Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazırbas, Vladimir Golkov, 2015)

FlowNet2 is a developed version of the original FlowNet. It was proposed by Ilg et al. The network is based on stacked many flow-related modules, FlowNet, FlowNetC, FlowNetS, and FlowNet-SD combined into a larger model which can outperform state-of-the-art methods and performs much faster. The model has a specific module that focuses only on the large displacement flow and a module FlowNet-SD that particularly focused on small displacement flow. FlowNet2 performance is slightly slower than the original FlowNet but can further reduce the estimation error by more than half. (Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, Thomas Brox, 2016)

Ranjan A. and Black M. propose to use the Spatial Pyramid technique in SPyNet. This network is using a standard spatial pyramid with deep learning and uses a coarse-to-fine strategy to determines motion flows. The flow images are computed and updated through each pyramid level. This approach makes Spatial Pyramid Network (SPyNet) plainer and more efficient for embedded purposes. (Anurag Ranjan, Michael J. Black, 2016)

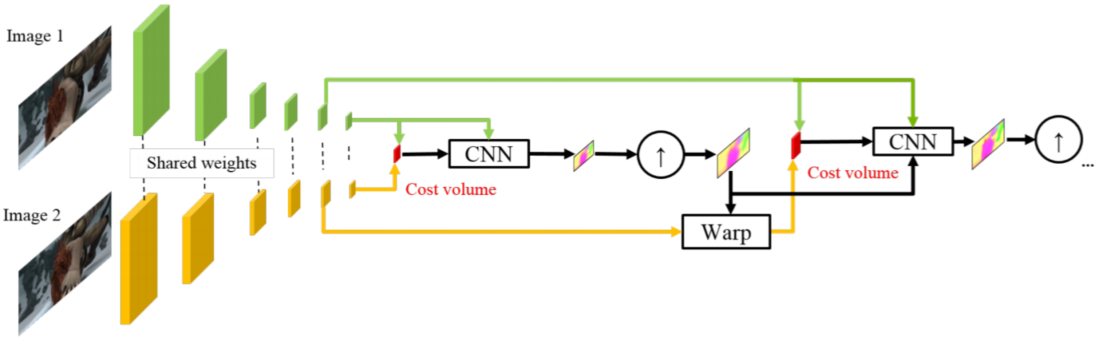
Sun D. et al. propose a compact and effective deep learning flow estimation network, PWC-Net. The model consists of many optical flow techniques such as image pyramid, warping layer, and cost volume layer. A stack of the feature pyramid layer for extraction features. Warping layer for warping the second image features approaching the first image using the upsampled flow and the cost volume layer is for constructing cost volume from their feature. (Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz, 2018)

2.2.1 Optical Flow using Deep Learning PWC-Net

PWC-Net has been designed based on standard methods of optical flow, and then be modified to have better performance from the traditional coarse-to-fine approaches. The key components of PWC-Net are modified and different from the traditional coarse-to-fine approaches. Firstly, PWC-Net uses the warping operation from the conventional approach in order to estimate large changes. Secondly, PWC-Net uses learnable feature pyramids instead of fixed feature pyramids, and third, PWC-Net has a specific layer to construct the cost volume which is a more particular representative of the optical flow than original images. The final optical flow image is constructed by CNN layers from the cost volume. Moreover, after training cost volume layers and the warping layers are freezing which can shrink the model size. Finally, PWC-Net uses a context network to utilize spatial information based on CNN which reduces computation, and it is more energy-efficient to refine the optical flow than the traditional methods such as median filtering and bilateral filtering. An overview of the PWC-Net architecture is shown in figure 2-2. (Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz, 2018)

Figure 2.2

Shows an Overview Architecture of PWC-Net

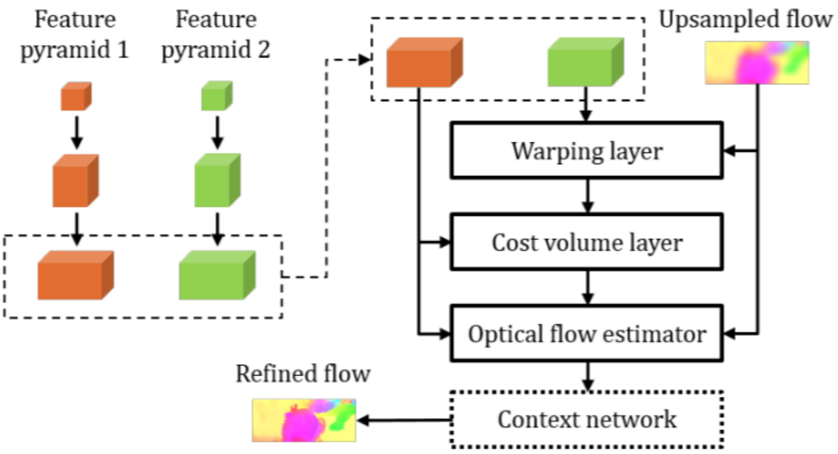


Note. Adapted from "Models Matter, So Does Training: An Empirical Study of CNNs for Optical Flow Estimation," by Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz, *arXiv preprint arxiv:1809.05571v1,* 2018.

The abstract concepts for all components, feature pyramid extractor, warping layer, optical flow estimator, and context networks are explained in figure 2-3.

Figure 2.3

Shows Each Layer and Component of PWC-Net



Note. Adapted from "Models Matter, So Does Training: An Empirical Study of CNNs for Optical Flow Estimation," by Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz, *arXiv preprint arxiv:1809.05571v1,* 2018.

Feature pyramid extractor is used to generate feature representation from input images pair I1 and I2. The pyramid has L-levels for the feature extractor while the images pair is feeding through the bottom (zeroth) level. To generate feature representation, layers of convolutional filters are used to downsample and extract the features at each pyramid level by a factor of 2. It has a number of feature channels 16, 32, 64, 96, 128,192 sequentially from the first to the sixth levels.

The warping layer is used to warp latent features from the I1 image approaching the I2 image after the drafted flow be upsampled and rescaled by a factor of two.

$$c^{l}_{w}(x) = c^{l}_{2}\, (x + 2 \times up_{2}\, (w^{l+1})\, (x))$$

Note. Where x is the pixel index and up2 denote the ×2 upsampling operator.

Bilinear interpolation is used to execute the warping process and calculate the gradients to the input CNN features and backpropagation flow. Warping can also compensate for the geometric deformities and put the image back to the right scale.

The cost volume layer is used to calculate a cost volume from the features which store the matching costs between individual pixels and their correlating in the next frame. In short, the cost volume layer is to find the degree of relation between features of the first image from warped features from the second image in terms of cost.

$$cv^{l}\, (x_{1}, x_{2}) = 1/N\, (c^{l}_{1}\, (x_{1}))^{T}\, c^{l}_{w}(x_{2})$$

Note. Where N is the length of the column vector c^{l}_{1} (x_{1}) and T is the transpose operator.

An optical flow estimator is used to construct fine flow images wl from matching cost and features and previous upsampled optical flow by stacked CNN. The optical flow estimators at different levels have unique characteristics instead of sharing the same things.

DenseNet connections can be used to improve the performance of the estimator by crate more direct connections between every convolutional layer and its early layer. This makes DenseNet has more direct joints than standard layers and also gives a significant improvement in the image classification field.

The context network is used to refine the flow after passing through the optical flow estimator by taking the estimated flow and features of the second last layer through feed-forward CNN and then outputs as a refined flow, $\hat{w}^{l0}_{\Theta}(x)$.

The context network uses a 3×3 spatial kernel and different dilation constants. Convolutional layers that have large dilation constants can extend the receptive field unit at each output without demanding extensive computation. The dilation constants at each convolutional layer are 1, 2, 4, 8, 16, 1, 1 sequentially.

2.3 Semantic Image Segmentation

Image segmentation is one of the computer visions tasks which can label segment regions of an image corresponding to the objects being shown. More precisely, image semantic segmentation is used to identify each pixel of an image with a related class of what objects are being represented. This method is generally referred to as dense prediction.

One thing different for semantic image segmentation is it does not separate instances that share the same class, it only considers which category of each pixel refers to. On the other hand, if the image has multiple objects in the same class in the input image, the segmentation map does not intrinsically identify these as two objects. To separate objects in the same class, there is an instance segmentation model, that can identify two separate objects in the same class.

Generally, the Image segmentation process can take either RGB color images or grayscale images and outputs a segmentation map that each pixel has a corresponding class label. In-depth, it generates a prediction map that has an output channel for each corresponding class. A prediction map can be flattening into a segmentation map by taking the Argmax of each depth-wise pixel vector.

A neural network architecture for semantic image segmentation task is simply stacking convolutional layers or deep convolutional networks with the same padding to preserve the spatial information before output a final segmentation map. This directly determines how to map from the input image to its corresponding classes through the continuous filter or transformation of feature mappings. In earlier layers, it attends to learn low-level concepts while higher layers extend more high-level and specific pattern feature mappings. In order to maintain the latent information, it typically requires a growing number of feature maps or channels as going further into the network. Despite this, this method is considered to be computationally extensive for preserve the information of the image throughout the network.

Another conventional method for image segmentation tasks is using an autoencoder-based structure. The network extracts the latent information by downsamples the input images and forms lower-resolution features which are discovered to be an effective way to distinguishing classes, and for upsampling that lower-resolution features back into a full-resolution segmentation map. (JORDAN, 2018)

In late 2014, Long et al. proposed the other method of using the pure power of fully convolutional network (FCN) trained, pixels-to-pixels for the image segmentation task. The network is implemented by modifying existing and well-studied image classification networks, eg. AlexNet, the VGG net, and GoogLeNet, to work as the encoder module for the network. A decoder module is implemented with transpose convolutional layers for slowly upsampling the encoded information, connecting skip connections from earlier layers, and combining these two feature maps into a high detailed full-resolution segmentation map. (Jonathan Long, Evan Shelhamer, Trevor Darrell, 2015)

Ronneberger et al. also propose the use of fully convolutional networks for biomedical image segmentation based on U-Net architecture. The U-Net architecture uses symmetric expanding in the contracting path that allows precise localization to capture latent information and the upsampling part that has a large number of feature channels to propagate context to higher resolution layers.

The general U-Net architecture compost stacked convolution operations for each block in the structure. Recently many researchers developed an original U-Net model and propose more superior modules that can be replaced instead of sequenced convolutional layers. (Olaf Ronneberger, Philipp Fischer, and Thomas Brox, 2015)

Drozdzal et al. introduce skip connections into the U-Net structure, one favor of residual blocks for building very deep FCNs. The short skip connection is used in the same block while allowing for faster convergence and allow for deeper models to be trained. The long skip connection is used for connection between encoder and decoder modules.

The very deep FCNs can beat near-to-state-of-the-art. (Michal Drozdzal, Eugene Vorontsov, Gabriel Chartrand, Samuel Kadoury, and Chris Pal, 2016)

Jegou et al. proposed the network that each layer is directly connected to every other layer in a feed-forward method as known as Densely Connected Convolutional Networks (DenseNets). This direct connection makes the network more precise and easier to train. In short, these connections make network reuse features more efficient by giving an opportunity to carry more low-level features from earlier layers along with higher-level features from newer layers. (Simon Jegou, Michal Drozdzal, David Vazquez, Adriana Romero Yoshua Bengio, 2017)

Chollet from Google Inc. proposes a network inspired by the Inception network. Instead of Inception modules, this deep convolutional neural network uses a depthwise separable convolutions layer. The background for this idea comes from the possibility of totally decoupled between spatial relationships and cross-channel mapping. The stacked depthwise separable convolution module makes this architecture easy to define, modify, and more efficient use parameters. (Franc¸ois Chollet, 2017)

A small and efficient model is proposed by Howard et al. from Google Inc. called MobileNets. This model has used a depth-wise separable convolution layer that implements a single filter to each input channel and a streamlined architecture base for mobile and embedded applications. (Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, Hartwig Adam, 2017)

In early 2019, MobileNetsV2 is a developed version of MobileNets was introduced by Sandler et al. from Google Inc. MobileNetsV2 is built based on DeepLabv3 but on a smaller version called Mobile DeepLabv3. The model uses an inverted residual structure to create a shortcut between the thin bottleneck layers and also uses lightweight depthwise convolutions to filter features from its central expansion layer. (Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, Liang- Chieh Chen, 2019)

MobileNetV3 is also introduced later in the same year as MobileNetsV2 by Howard et al. from Google Inc. This network implements a combination of hardware with network architecture search (NAS) from the NetAdapt algorithm and Lite Reduced Atrous Spatial Pyramid Pooling (LR-ASPP) to achieve a new state-of-the-art and also compatible with the mobile phone CPU. (Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, Quoc V.Le, Hartwig Adam, 2019)

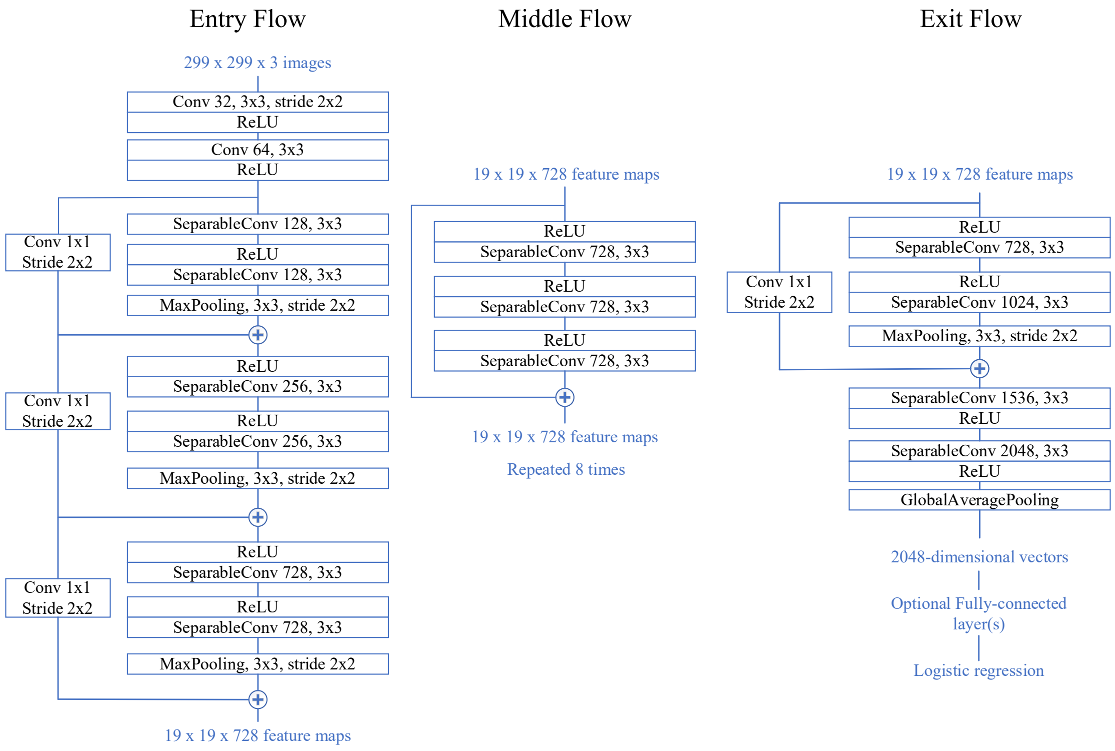
2.3.1 Semantic Image Segmentation Using Xception Network

Chollet et al. from Google Inc. propose a segmentation network name Xception, which is short for “Extreme Inception”. The network is based on fully depthwise separable convolution layers in convolutional neural network architecture. The background concept comes from the idea that “the mapping of cross-channel relationships and spatial relationships in the feature maps of convolutional neural networks can be completely decoupled”. This hypothesis is a more intense version of the Inception architecture. In the Inception model, this concept is just independently looking across spatial and cross-channel correlations by performing a set of 1x1 convolutions before applying regular 3x3 or 5x5 convolutions.

The Xception network can be separated into three-unit: entry flow, middle flow and exit flow. The detail of network architecture is shown in figure 2-4.

Figure 2.4

Shows The Detail of Xception Network Architecture



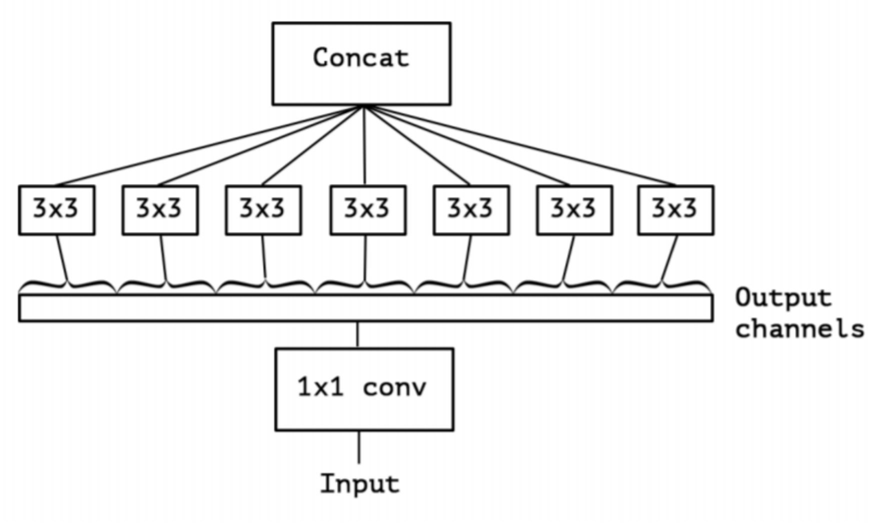
Note. Adapted from "Xception: Deep Learning with Depthwise Separable Convolutions," by Franc¸ois Chollet, *arXiv preprint arxiv:1610.02357,* 2017.

The depthwise separable convolution layer is used to independently perform spatial convolutions on every non-overlapping segment of the output channels after being processed by a large 1x1 convolution and a pointwise 1x1 convolution layer while creating the new channel space from the channel's output.

The depthwise separable convolution layer shares the same similarity with the extreme version of an Inception. However, there are some minor differences. In the extreme Inception module, the 1x1 convolution is performed first and then followed by channel-wise spatial convolution. Another difference is ReLU non-linearity operations are implemented in extreme Inception modules instead of linearities operations which are usually implemented in depthwise separable convolutions.

Figure 2.5

Shows an Extreme Version of The Inception Module



Note. This figure demonstrated an extreme version of the inception module, which performs one spatial convolu-tion per output channel of the 1x1 convolution. Adapted from "Xception: Deep Learning with Depthwise Separable Convolutions," by Franc¸ois Chollet, *arXiv preprint arxiv:1610.02357,* 2017.

CHAPTER 3
METHODOLOGY

The method used for detecting a change in the road environment or road anomaly events is based on reconstructed sequences that represent the regular patterns. when an abnormal event happens, that recent image sequence will contain an anomaly and be significantly different from the other frames. Inspired by deep learning techniques, a deep autoencoder is suitable for reconstructed these sequences. It is an end-to-end model that can extract and understand spatial and temporal features together.

Unsupervised training methods are preferred to train the system for detecting anomaly events by using regular road footage in the training set. The objective of the model is to minimize the reconstruction error between the actual video frames and the reconstructed video frames from the trained network. After the model is well trained, normal scenarios are supposed to have low reconstruction error due to the high probability of that frames being represented in the footage videos, whereas irregular scenes are supposed to be rarely represented and have high reconstruction error.

Based on this hypothesis, the AE model would be able to detect and distinguish when and where an irregular event occurs. The optical flow and semantic segmentation techniques are introduced to ensure the discrimination between normal events and abnormal events by the network. The input frames are preprocessed or be reduced to the spatial resolution by the optical flow network, PWC-Net, or by the semantic image segmentation, DeepLab Xception network before further sending to the deep autoencoder for detection of anomaly events in the input frames.

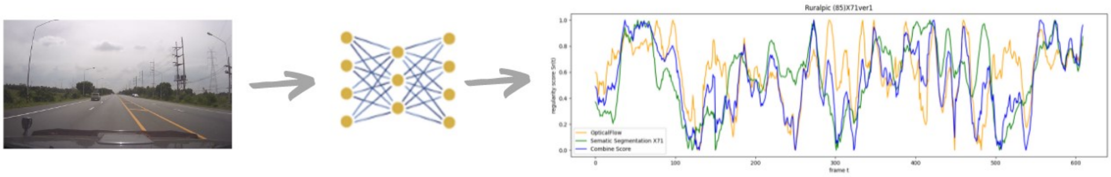
3.1 Conv-LSTMs Encoder-Decoder

A Conv-LSTMs Autoencoder is the main component for road anomaly detection. It is supposed to learn and understand the regular patterns in the training videos by learning spatial and temporal relationships in the events. To interpret spatial and temporal relationships, it requires to consist of spatial and temporal extractor modules. The spatial extractor is used to extract the latent spatial patterns of each frame in the videos.

For more details, the spatial extractor structure has stacked convolutional and deconvolutional layers and has the temporal extractor layer in between. The temporal layers have 3 convolutional long short-term memory (Conv-LSTMs) layers used for learning temporal patterns of the encoded spatial structures.

Figure 3.1

Shows Vanilla Conv-LSTMs Autoencoder



Note. This figure demonstrated the overall vanilla Conv-LSTMs pathway. The color image sequences are fed directly into the Conv-LSTMs network to reconstruct sequences and predict anomaly score

3.2 Optical Flow using Deep Learning PWC-Net

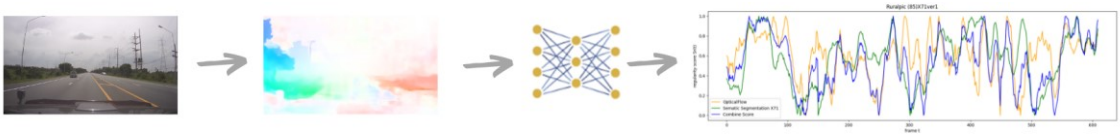
PWC-Net is introduced for estimating the flow vector from the series of the input. The input frames are transformed into dense optical flow images before being interpreted by the Conv-LSTMs autoencoder.

PWC-Net is used to reduce the spatial features that the autoencoder requires to learn. PWC-Net can be separated into two main levels. In the first level, a feature pyramid extract features from the images pair. At the end of the first level, a cost volume is created by comparing the difference of features from the first image with corresponding features from the second image. This level has a miniature spatial resolution, the cost volume is then constructed using a small search range. The CNN is then used to draft the predicted flow from this cost volume and features of the first image. This drafted flow then being upsampled and rescales before sending through the second level.

In the second level, the features of the second image are transformed toward the first image for reconstructs a new cost volume from these warped features and the first image's features. PWC-Net still uses a small search range to create the second level cost volume because the warping technique already compensates for the large motion. The CNN is again used to draft the new predicted flow from the second level cost volume, the upsampled flow, and features from the first image. This second drafted flow then being upsampled and rescales before sending through further to the next level. This process is then repeated until reaching a satisfying level.

Figure 3.1

Shows Overall PWC-Net-Conv-LSTMs Autoencoder



Note. This figure demonstrated the overall PWC-Net-Conv-LSTMs Autoencoder pathway. The input frames sequences are transformed into dense optical flow images before being interpreted by the Conv-LSTMs autoencoder

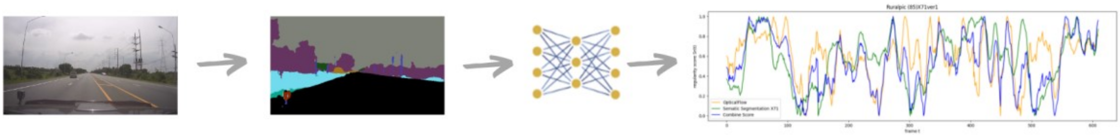
3.3 Semantic Image Segmentation

The DeepLab Xception network is introduced for construction segmentation images. It is used to reduce the spatial features by reducing the resolution of the input images to the colour-related segmentation images before being interpreted by Conv-LSTMs Autoencoder. In order to build segmentation images, the input images need to pass through many stacked depthwise separable convolution layers in the network.

The segmentation model is based on the Xception architecture. It consists of 36 stacks of depthwise separable convolution layers with residual connections in every module except for the first and last modules. These convolutional layers are constructed into 14 modules.

Figure 3.2

Shows Overall Xception-Conv-LSTMs Autoencoder



Note. This figure demonstrated the overall Xception-Conv-LSTMs Autoencoder pathway. The input frames sequences are reduced the spatial features before being interpreted by the Conv-LSTMs autoencoder

3.4 Anomaly Score

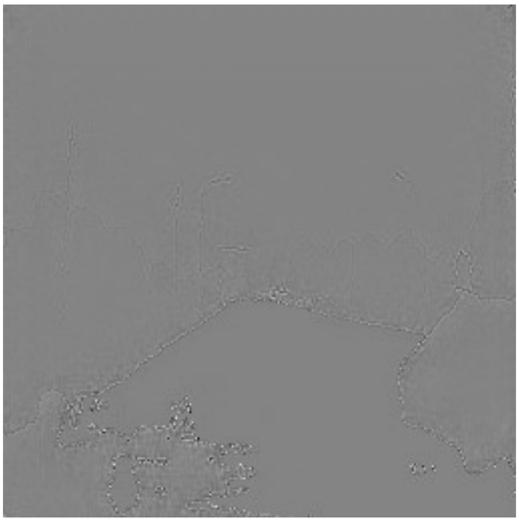
The anomaly score can be calculated from the reconstruction error between the actual video frames and the output reconstructed video that has been constructed from the autoencoder. The reconstruction error is an absolute anomaly score. The huge reconstruction error indicates the anomaly events. In practice, this absolute score cannot be directly usable. It already interferes with the environment in the long term because it already has a low resolution to recognize any irregular events. In order to accurately calculate the anomaly score, the time interval base and GMM base are introduced to calculate the relative anomaly score in one period of time.

The reconstruction error = matrix norm (The actual video frames - The output reconstructed video frames)

The anomaly score = The reconstruction error / (Max of the reconstruction error of that time interval – Min of the reconstruction error of that time interval)

Figure 3.4

Shows The Reconstruction Error Mapping



In time interval base, the time interval needs to be short enough to prevent long term environmental changes to affect and needs to be long enough to encase irregular events. For simple interpretation, the anomaly score is normalized in the 0-1 range format. Zero is completely abnormal and one is normal. The formula to calculate the anomaly score is the reconstruction error divided by the max-min score range of that time interval. If the anomaly score is less than 0.2, that moments will be considered as anomaly events.

GMM base or Gaussian Mixture Modelling is used for clustering the reconstruction error from the 2 models. The algorithm will calculate the best number of clusters from thousand steps and thousand initial random starting points or call it ‘max’. if starting points fix is used, it calls ‘rt’. It can further be divided into the mahalanobis distance method and the Mean method.

Mahalanobis distance is used to calculate the distance from any point in the 2-model pair score to the centroid in the mahalanobis distance method and also used to calculate the distance from the combined score point to the mean of that cluster in the mean method. If the distance is higher than the threshold, it will be considered as anomaly points.

Figure 3.3

Mahalanobis Distance Method on Pair

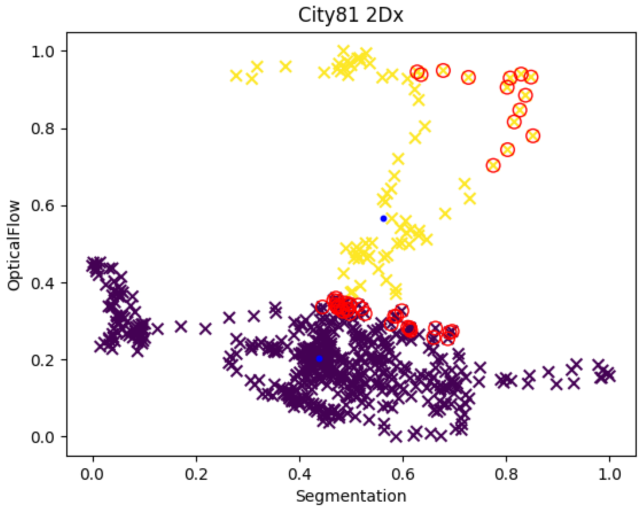
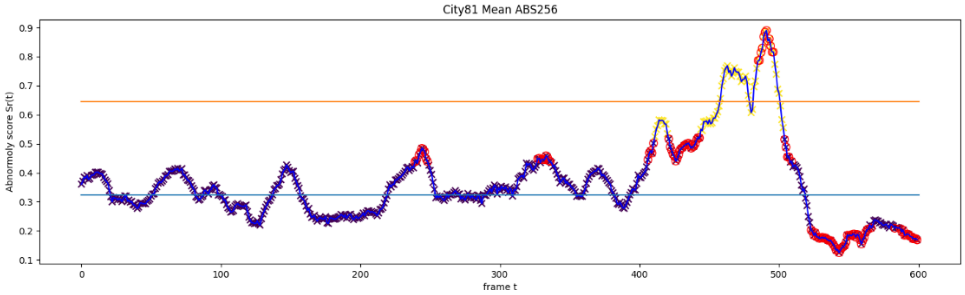


Figure 3.6

Mahalanobis Distance Method on Mean



3.5 Training

In the training process, the Conv-LSTMs model is trained with 160 videos footage or 160 minutes duration of the road scenes. The video footage consists of a city dataset and a rural dataset. The model is shuffling training with sequenced images from these 2 datasets. To prevent overfitting in a specific scenario, the model is trained with an epoch equal to one. To construct sequenced images, the sliding technique is also used to triple the training images at a different order of sequenced images.

Figure 3.4

Example of Training Image from Cityset

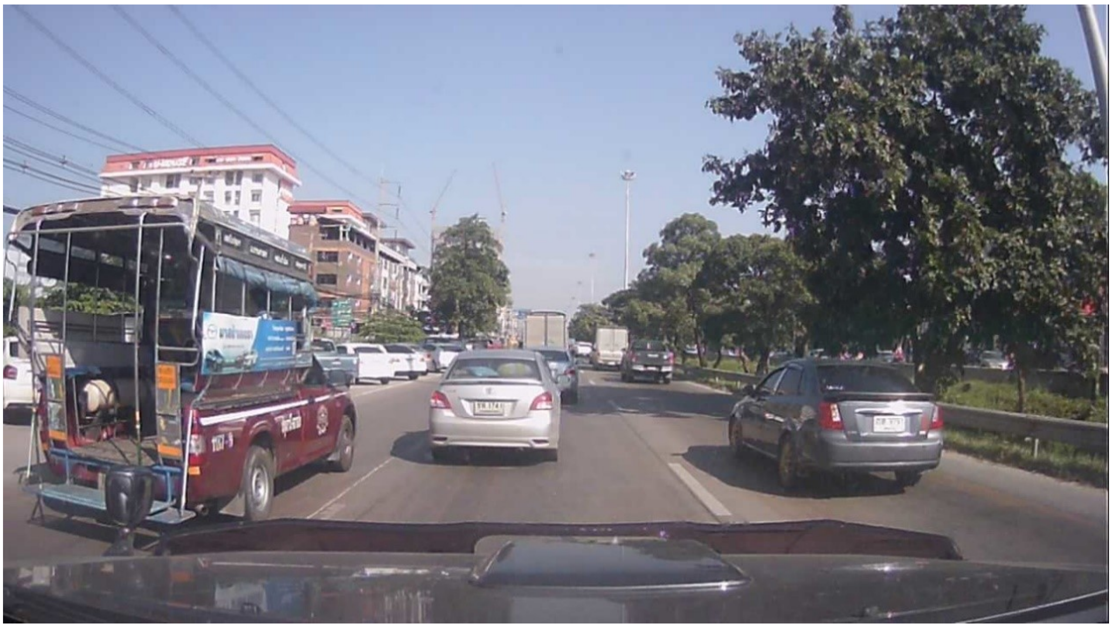
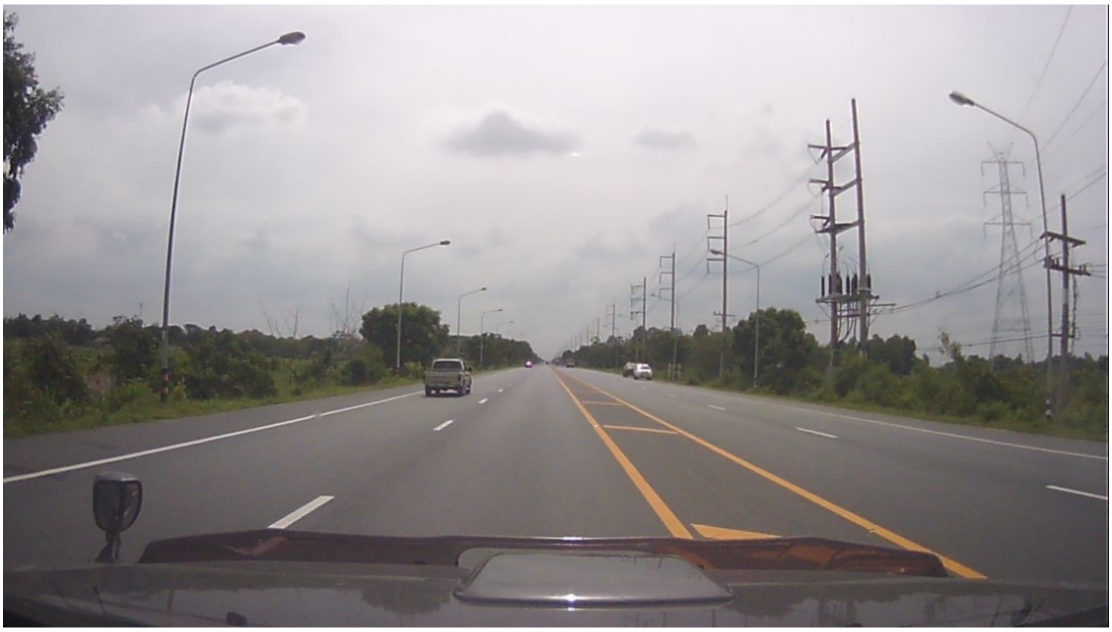


Figure 3.8

Example of Training Image from Ruralset



CHAPTER 4

EXPERIMENTAL RESULTS

The implementation of 2 subnetworks is time-consuming and computation consuming result in the prediction is far beyond real-time implementation especially in the Conv-LSTMs network. The video 1 minute long takes at least 400 seconds or over 6 times the length of the input video duration before presenting the resulting graph.

In the evaluation process, the test data will be divided into 3 main categories. First, the city dataset collects the footage in the urban environment with high traffic. it contains high rise building scenes, a sky train rail track, the overbridged, construction site scenes and the real congestion scenes on the roads. Second, Rural and AIT campus dataset that contain traffic scenes in the suburban areas with fewer cars on the road, narrowed roads and the trees along the side. The third dataset is an extended test, which contains the scenes in specific conditions such as off-road scenes, dense areas road scenes, late afternoon light conditions and crash detection test.

There is no ground truth for categorization anomaly events from the footage, therefore the human sense is approximately used to classify the events that are obviously abnormal and sudden change for analyzing the performance of the model. The system also detects sudden changes around every 15-20 seconds depending on the scenarios.

Overview:

There are two variations of the results, both are tested with the input frames size at 256x256. First is the time interval base with 2 versions of weight ratio between dense optical flow to semantic segmentation of 1:1 and 2:1. The second is the GMM base score, the mahalanobis and mean.

In the city test set, from 13 footage videos. The best method is the time Interval base 20s and model ratio 2:1. It can predict 69 anomaly events that can be divided into true-positive 42 times and false-positive 27 times. Comparison with human sense at 62 anomaly events, the system has the precision at 60.87 percentage, the recall at 67.74 percentage and the F1 score at 64.12 percentage.

In the rural test set, from 8 footage videos. The best method is the mahalanobis 70 max in the GMM base model. It can predict 48 anomaly events that can be divided into true-positive 27 times and false-positive 21 times. In comparison with human sense at 37 anomaly events, the system has the precision at 56.25 percentage, the recall at 72.97 percentage and the F1 score at 63.53 percentage. The performance of the time Interval base slightly drops possibly come from the fixed time interval that is not suitable for rural scenarios.

In the Extest test set, from 13 footage videos. There are 2 methods that have the exact same score, the time interval base at the 20s and model ratio of 1:1 and the GMM base is the mahalanobis 70 at a score of 66.67 percentage. The time Interval base can predict 69 anomaly events that can be divided into true-positive 45 times and false-positive 24 times. In comparison with human sense at 66 anomaly events, the system has the precision at 65.22 percentage, the recall at 68.18 percentage and the F1 score at 66.67 percentage.

The GMM base, the mahalanobis 70 can predict 75 anomaly events that can be divided into true-positive 47 times and false-positive 28 times. In comparison with human sense at 66 anomaly events, the system has the precision at 62.67 percentage, the recall at 71.21 percentage and the F1 score at 66.67 percentage.

In the other test set such as AIT footage, the system can detect similar to the rural dataset. It can recognize the tree, parking roof, building but it fails when the car hits a bumper. For the night test set, the system predicts barely accurately due to poor camera night vision, high noise level and lack of training for night scenes. There is some environment that can lead to false detection such as the tree’s shadow. the sunlight directly to the camera, dawn time and other low light conditions.

To summarize, the best system for all datasets comes from the GMM base method, the mahalanobis 70 max of overall weight F1 score at 62.12 percentage. In comparison with the best time interval base method, the time Interval at the 20s and model ratio 2:1 at 60.87 percentage of the F1 score. The overall performance of the system is no significant difference between the model weight ratios.

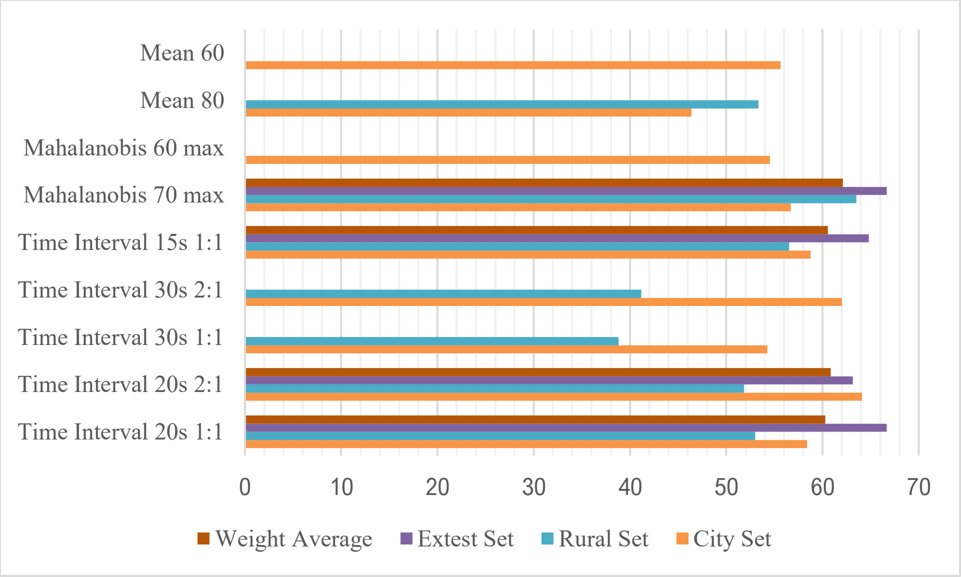
Table 4.1

Score Comparison on Different Methods

	City Set	Rural Set	Extest Set	Average	Weight Average
Time Interval 20s 1:1	58.39416	53.01205	66.66667	59.35763	60.29079823
Time Interval 20s 2:1	64.12214	51.85185	63.15789	59.71063	60.86633037
Time Interval 30s 1:1	54.26357	38.80597			
Time Interval 30s 2:1	62.0155	41.17647			
Time Interval 15s 1:1	58.75	56.52174	64.82759	60.03311	60.54948629
Mahalanobis 70 max	56.71642	63.52941	66.66667	62.30417	62.12398217
Mahalanobis 60 max	54.54545				
Mean 80	46.4	53.33333			
Mean 60	55.62914				

Table 4.2

Score Comparison on Different Methods on Bar Chart



The anomaly events can be evaluated in terms of category, the Extest dataset is used for these specific events. The below table shows the GMM base method, the mahalanobis 70 max test on Extest dataset with category on events.

Table 4.3

Score Comparison on Different Category

	Ground truth	Mahalanobis 70 max	Percentage
Off road	6	6	100%
Intersection	4	3	75%
Dog	2	0	0%
Obstacle	5	3	60%
Car pass front	1	1	100%
Car stop	2	2	100%

The results show that the system is good at detecting cars driving off-road. car passing in front, intersection, and car stop however it cannot detect dog sitting on the road due to its small area compared to the whole frame. The system fails to detect the front car obstacles. It may consider them as cars parking in the traffic.

In this chapter, the effects of changing some parameters on the results will be further discussed below.

The specification of the computer that used to train and evaluate is:

Device name:	HP Z440 Workstation
Processor:	E5-2678V3 2.50-3.40GHz 12C 24T
RAM:	Memory 64 GB
SSD:	2.5" 512GB
OS:	Windows 10 Pro 64Bit
Graphic card:	GTX 1070 8GB

Device name:	DESKTOP-95MU2RL
Processor:	Intel(R) Core (TM) i7-9700F CPU @ 3.00GHz 3.00 GHz
RAM:	Memory 16.0 GB
HDD:	1TB
OS:	Windows 10 Education
Graphic card:	RTX 2070 8GB

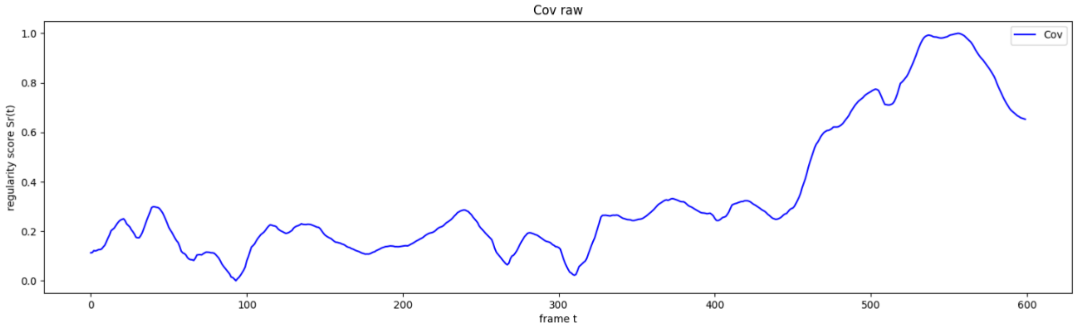
4.1 Conv-LSTMs Encoder-Decoder

The vanilla Conv-LSTMs Encoder-Decoder as the anomaly detection network is quite good. It has the ability to detect the change in the surrounding environment such as a huge building, walk bridge, sky train station but it also has nonsense false detection, and it can be affected by some vibration. The overall graph is in low-resolution detection compared with other networks.

In the city dataset, it can detect motorbikes, the front car ahead breaking, and the building near the road. In the rural dataset, it detects the passing cars, trees, houses.

Figure 4.1

The Vanilla Conv-LSTMs from Cityset81



4.1.1 The Effect of Input Images Size on Conv-LSTMs

The video dataset used to train the model is recorded in HD resolution or 1280 x 720 pixels. In order to reduce the dimension for capable of training and conserve the ratio of the sequenced images. The images have reduced the resolution to 640 x 340 pixels. For comparisons of the effect of input images size, another training model further reduces the dimension down to 256 x 256 pixels.

The result shows that the low-resolution input has a higher score than the high resolution. This suggests at the lower resolution, the system is better in capturing latent data than the high-resolution images, although the high-resolution training images can make the system a little bit more sensitive to the change in the environment.

Figure 4.2

The Result of Input Image Size 256x256 from Cityset81

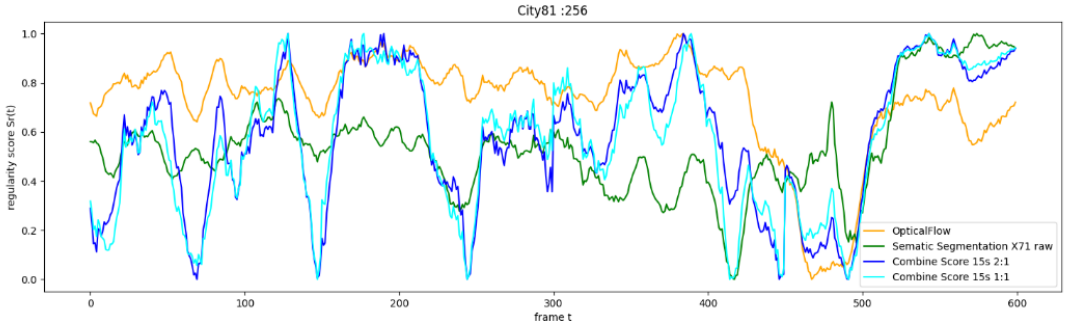
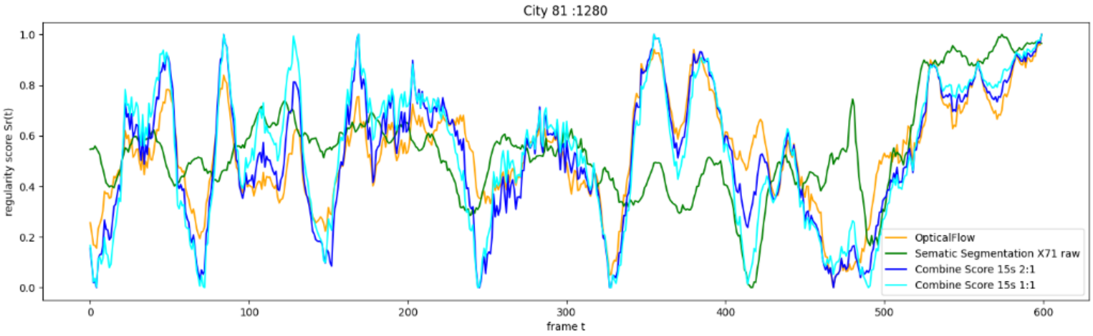


Figure 4.3

The Result of Input Image Size 640x360 from Cityset81



4.1.2 The Effect of Input Images Sequence on Conv-LSTMs

The Conv-LSTMs network requires the input sequenced images to learn the latent content in the sequenced images. The sequenced images must be large enough to capture and distinguish the normality in the events. The sequenced images relate to the frame rate of the videos in the dataset. The framerate in the video is 30 fps. It is too large to train 30 frames or 1-second duration at once, so the model is trained with 10 frames per sec instead and for comparison 1 frame per sec is also trained.

The result shows that at 1 fps, the time gap between images is too high for the system to capture the latent content in the events and provide a false prediction.

4.1.3 The Effect of Color Channels of Input Images Sequence on Conv-LSTMs

The output of the sequenced images from semantic image segmentation appears in 1 color channel result from a hot-code method, the number in the pixels represent a corresponding class of that pixel. To make an array be an image, it can translate to an RGB image with color in that pixel corresponding to class it be, or it can be translated to a grayscale image that the corresponding class is separated between 0-255 values.

The differences between an RGB image and the grayscale image is insignificant and can be negligible hence, a grayscale image is chosen to reduce model loads and make prediction faster. The Conv-LSTMs network that uses greyscale as input can reduce the computation time by around 5 minutes per 1-minute footage.

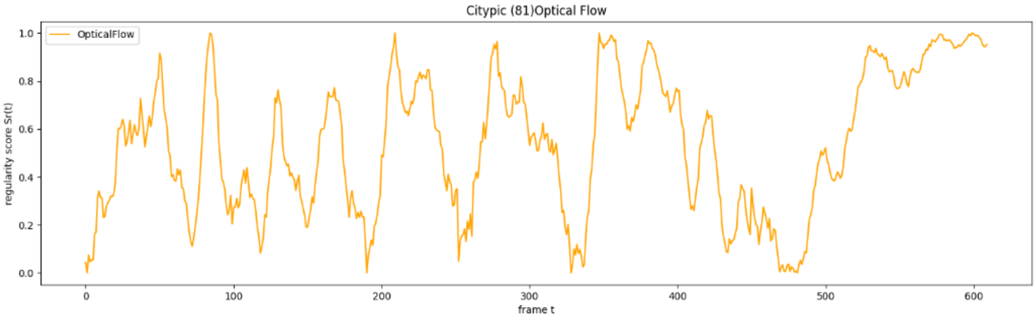
4.2 Optical Flow as Preprocess Network

The optical flow using PWC-Net combined with Conv-LSTMs is very good at detecting moving objects around the car including the structure that builds nearby the road. The limitation is the low range of field detection. The objects must be near or huge enough before the network can detect them and the objects must be moved between two frames therefore, it is harder to recognize obstacles in front of the car because the pixels are barely moving, however it is excellent at detecting sharp edges when passing the other car or a wall's corner. It can also notice the acceleration, deceleration and some kind of car's vibration.

In the city dataset, it can detect motorbikes, the passing car, the building, or the car that park near the road. In the rural dataset, it detects the nearby passing cars, trees, wall.

Figure 4.4

The Optical Flow from Cityset81

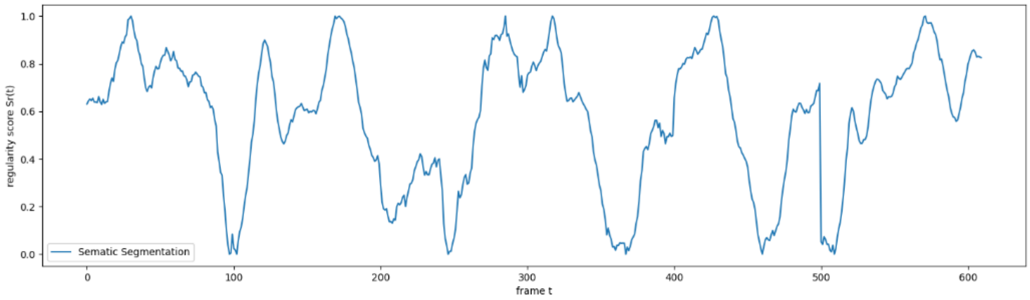


4.3 Semantic Image Segmentation as Preprocess Network

The semantic image segmentation using Deeplab Xception combined with Conv-LSTMs is good at detecting the change in the surrounding environments such as motorbikes, people, side buildings, walk bridges, sky train stations and huge trees at higher resolution or sensitivity than vanilla Conv-LSTMs. Further, it can also distinguish the intersection and off-road moments. The disadvantage is the segmentation process can predict false detection easily and the network is delicate to vibration.

Figure 4.5

Example of Segmentation as Preprocess from Cityset81



4.3.1 The Effect of Input Images Size on Semantic Segmentation

In the image semantic segmentation process, the input image could be at a high resolution as possible because at a high resolution, the system will predict the classes with higher accuracy than low resolution and the Conv-LSTMs can furthermore achieve the fine detail of the environments.

In the city dataset, it can detect motorbikes, people, the passing, and lane changing car, the building and in the rural dataset, it detects trees, walls and off-road scenes.

4.4 Anomaly Score Interpretation

4.4.1 The Combined Score Between Optical Flow and Semantic Image Segmentation

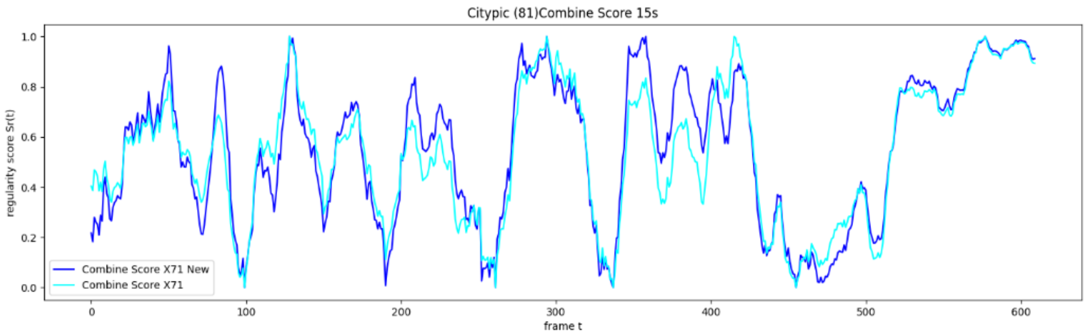
To overcome the limitation of both models, a combined score is applied. A combined score is a combination of anomaly score from the *Optical Flow* model and the *semantic image segmentation* that is calculated from the average score between 2 models. It is sensitive to either change in the surrounding environment or nearby moving objects. Further, the combined score can reduce the effect of false prediction from any model by equalizing it. The result from the combined score reveals the best accuracy compared to the individual model.

4.4.2 The Effect of Model Weight Ratio for Time Interval-Based Prediction.

Normally, the combined score is calculated from the average score from the *Optical Flow* model and the *semantic image segmentation* or at the ratio of 1:1 however, this ratio can be changed to any number. In the experiment, 2:1 is chosen because the optical flow is better in detecting the surrounding environment and less sensitive to the vibration than the semantic image segmentation. This ratio can slightly reduce the number of false predictions caused by false segmentation and vibration, except it also reduces the number of predicted abnormal events. This downside is overcome its advantages, thus the model ratio at 1:1 is preferred.

Figure 4.6

Result of Model Weight Ratio at 2:1 (New) and 1:1 from Cityset81



4.4.3 The Effect of a Time Interval for Time Interval-Based Prediction.

In order to distinguish between normal scenes and anomaly scenes, the score is used. If the score is less than 0.2, indicate this moment is an anomaly event and to calculate the relative score, the length of a time interval is applied.

The best time interval for calculating anomaly moments is 15s for capturing all possible anomaly events including passing the trees, passing side road building, the car passing that happen in a short period. If the time interval is too high, the system will predict a long-term anomaly score and the graph will be smoother with small fluctuation instead of rapid change between normal and abnormal moments. However, this time interval setting makes the system more delicate to false prediction and vibration in calm conditions and suburban areas.

Figure 4.7

The Citytest 81 at 15s-Time Interval

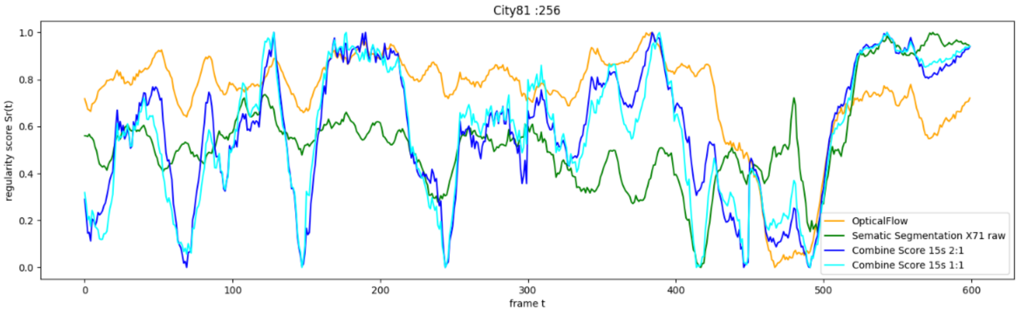
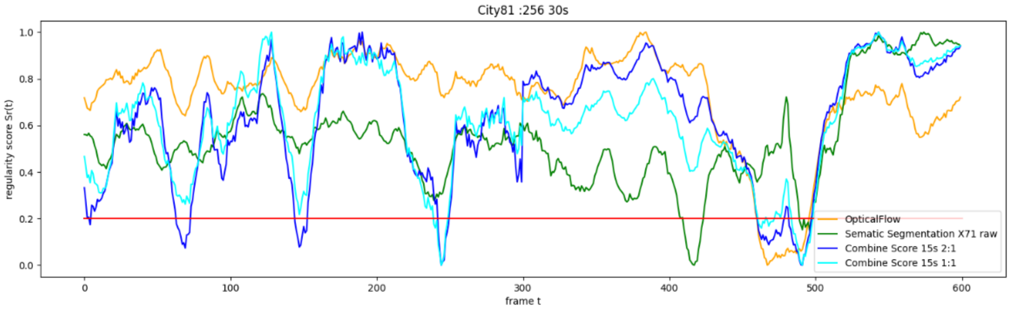


Figure 4.8

The Citytest 81 at 30s-Time Interval

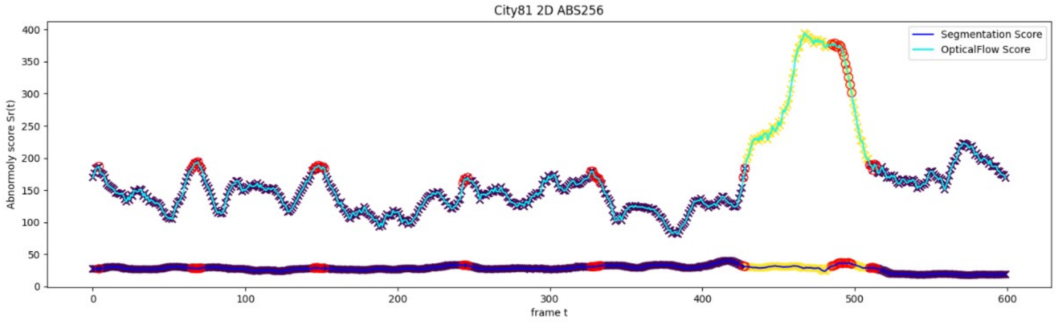


4.4.4 The Effect of Distance Percentile for GMM Based Prediction.

Mahalanobis distance is used to calculate the anomaly points from the points that have a distance higher than the control percentile. The best average percentile that results in the highest score is 70 percentiles without fix the initial point in the mahalanobis distance method and gets around 60 percentiles in the mean method.

Figure 4.9

The Citytest 81 at The Mahalanobis 70 Percentiles



Note. The red dots in this figure indicate moment that be considered as an anomaly.

CHAPTER 5
CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

The final model of road anomaly detection can detect the anomaly events on the road in the three different tests set, city dataset, rural dataset, and Extest dataset quite well but unfortunately, the overall performance is not getting an F-score higher than 70 percent. The best score from the three datasets is 64.12%, 63.53%, and 66.67 %, correspondingly and the overall score is 62.12 %. The best system is made by using HD resolution as input images in the preprocess network and then reducing the sequenced image's dimension down to 256x256 pixels with 1 color channel in segmentation while has 3 color channels in optical flow for the Conv-LSTMs network. In the calculation of the anomaly event process, the anomaly is computed using the mahalanobis distance on a reconstruction score from the 2 different models, dense optical flow, and segmentation with thousand steps and thousand initial points. The network uses around 7-10 minutes in order to process a 1-minute-long video and can detect sudden changes around every 15-20 seconds.

The limitation of the system is it cannot be used in low light conditions. The weather needs to be clear for the highest performance. The system is also delicate to every form of vibration or any rapid acceleration such as irregular road, road bumper, and fast turning or breaking however there is some evaluation that considers these as anomaly events.

5.2 Recommendations

To improve the overall performance of the system, the camera that use to capture the footage should have a good image stabilization system and night vision mode to capture the footage steady and reduce the noise in the low light environment, the camera better installs outside of the car to minimize the shadow and light glare error on the car's window. With better camera equipment, the system can be further trained in low light and night conditions and can be trained in other weather conditions such as the raining and foggy environment.

To reduce the sensitivity to the change in the surrounding environment, the input images can be cropped sky out, reduce the visibility and view angles to mainly focus on the road. Sky and cloud conditions can influence the predicted results.

To increase the precision in the rural dataset and other environments, in the time interval base method adjustable time interval should be used. The time interval can be related to the speed of the car or the magnitude of the flow vector in the optical flow process can be used to estimate the car speed. In the other method, the system might be trained with longer periods and varying situations.

To reduce the computation time of the system, the higher computation graphic card and larger RAM capacity can be used to make the system close to real-time detection. The other higher efficient and newer model network can replace Conv-LSTMs in future work.

REFERENCES

Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazırbas, Vladimir Golkov. (2015). FlowNet: Learning Optical Flow With Convolutional Networks. *In ICCV*.

Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, Hartwig Adam. (2017) . MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv preprint arxiv:1704.04861*.

Andrew Howard, Mark Sandler, Grace Chu, Liang- Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, Quoc V.Le, Hartwig Adam. (2019) . Searching for MobileNetV3. *arXiv preprint arxiv:1905.02244*.

Anurag Ranjan, Michael J. Black. (2016). Optical Flow Estimation using a Spatial Pyramid Network. *arXiv preprint arxiv:1611.00850v2*.

Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. (2018). Models Matter, So Does Training: An Empirical Study of CNNs for Optical Flow Estimation. *arXiv preprint arxiv:1809.05571v1*.

Dong Gong, Lingqiao Liu, Vuong Le, Budhaditya Saha, Moussa Reda Mansour, Svetha Venkatesh. (2019) . Memorizing Normality to Detect Anomaly: Memory-augmented Deep. *arXiv preprint arxiv: 1904. 02639*. Retrieved from https://arxiv.org/pdf/1904.02639.pdf

Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, Thomas Brox. (2016). FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks. *arXiv preprint arxiv:1612.01925*.

Franc¸ois Chollet. (2017) . Xception: Deep Learning with Depthwise Separable Convolutions. *arXiv preprint arxiv:1610.02357*.

Gituma, M. (2019). *medium*. Retrieved from https://medium.com/swlh/what-is-optical-flow-and-why-does-it-matter-in-deep-learning-b3278bb205b5

Jefferson Ryan, Medel Andreas Savakis. (2017). Anomaly Detection in Video Using Predictive Convolutional Long Short-Term Memory Networks. *arXiv preprint arxiv:1612.00390*.

Jonathan Long, Evan Shelhamer, Trevor Darrell. (2015) . Fully Convolutional Networks for Semantic Segmentation. *arXiv preprint arxiv:1411.4038.*

JORDAN, J. (2018). *jeremyjordan*. Retrieved from https://www.jeremyjordan.me/semantic-segmentation/

Lin, C.-e. (2019). *nanonets*. Retrieved from https://nanonets.com/blog/optical-flow/

M. Sabokrou, M. Fayyaz, M. Fathy, Z. Moayed, and R.Klette. (2017). Deep-Anomaly: Fully Convolutional Neural Network for Fast Anomaly Detection in Crowded Scenes. *arXiv preprint arxiv:1609.00866.*

Mahmudul Hasan, Jonghyun Choi, Jan Neumann, Amit K. Roy-Chowdhury, Larry S. Davis. (2016). Learning Temporal Regularity in Video Sequences. *arXiv preprint arxiv:1604.04574.*

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, Liang-Chieh Chen. (2019). MobileNetV2: Inverted Residuals and Linear Bottlenecks. *arXiv preprint arxiv:1801.04381.*

Michal Drozdzal, Eugene Vorontsov, Gabriel Chartrand, Samuel Kadoury, and Chris Pal. (2016) . The Importance of Skip Connections in Biomedical Image Segmentation. *arXiv preprint arxiv:1608.04117.*

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. *arXiv preprint arxiv:1505.04597.*

Radu Tudor Ionescu, Fahad Shahbaz Khan, Mariana-Iuliana Georgescu, Ling Shao. (2019). Object-centric Auto-encoders and Dummy Anomalies. *arXiv preprint arxiv:1812.04960.*

Ryota Hinami, Tao Mei, and Shin'ichi Satoh. (2017). Joint Detection and Recounting of Abnormal Events by Learning Deep Generic Knowledge. *arXiv preprint arxiv:1709.09121.*

Simon Jegou, Michal Drozdzal, David Vazquez, Adriana Romero Yoshua Bengio. (2017). The One Hundred Layers Tiramisu:Fully Convolutional DenseNets for Semantic Segmentation. *arXiv preprint arxiv:1611.09326.*

Weixin Luo, Wen Liu, Shenghua Gao. (2017) . Remembering history with convolutional LSTM for anomaly detection. *2017 IEEE International Conference on Multimedia and Expo (ICME)* , 439- 444. doi:10.1109/ICME.2017.8019325

Wen Liu, Weixin Luo, Dongze Lian, Shenghua Gao. (2018). Future Frame Prediction for Anomaly Detection – A New Baseline. *arXiv preprint arxiv:1712.09867v3*.

Xingjian, Shi Zhourong, Chen Hao Wang, Dit-Yan Yeung. (2016). Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. *In NIPS*.

Ye Muchao, Peng Xiaojiang, Gan Weihao, Wu Wei and Qiao Yu. (2019). AnoPCN: Video Anomaly Detection via Deep Predictive Coding Network. *In Proceedings of the 27th ACM International Conference on Multimedia (MM '19)*, 1805–1813. doi:10.1145/3343031.3350899

Yong Shean Chong, Yong Haur Tay. (2017). Abnormal Event Detection in Videos using Spatiotemporal Autoencoder. *arXiv preprint arxiv:1701.01546*.