Anomaly detection from video streams in reconnaissance missions

TDDE19 Advanced Project Course - AI and Machine Learning

Group 4
Ludvig Widén, ludwi159
Felix Lindgren, felli675
Peter Wickenberg, petwi187
Hannes Westander, hanwe828

LINKÖPINGS UNIVERSITET

January 5, 2023

1 Introduction

This section introduces the project, describes the background, and outlines the questions to be investigated.

1.1 Background

The company, Combitech, has tasked the project group to create a model for recognizing anomalies from video streams. The videos are supposed to be obtained from drones that patrol the same area every day in an archipelago-like environment. The recordings are to be compared to determine if the difference constitutes an anomaly. The overlying purpose of the model is that it should be able to be included in rescue missions and surveillance. The goal of this project is to implement a model that can identify if a video contains an anomaly or not. Classifying the anomalies and understanding what kind of anomalies are present, is not part of the assignment, the interest lies in highlighting and locating the anomalies.

This report will investigate and compare two different models: an unsupervised and a weakly-supervised model. The two models will be evaluated according to their accuracy and ability to identify anomalies. The unsupervised approach will use *Autoencoders* (AE) and the weakly-supervised approach will use *Multiple Instance Learning* (MIL). Due to limited access to data, the model training data will consist of drone videos in a desert-like environment for the unsupervised model, and surveillance footage from the UCF-Crimes dataset for the weakly-supervised model.

1.2 Anomalies

An anomaly in this project is something that deviates from normality and what is expected. In the newly created datasets, this equates to an environment without people, cars, boats, and drones. In the UFC-Crimes dataset, this equates to an environment where no crime is accruing.

1.3 Question statement

The questions to be answered in this project are the following:

- What is the difference between the two solutions?
- What are the pros and cons of each model?
- What have the project members learned during this project?

1.4 Related Work

Anomaly detection is a classical and widely researched area in computer vision. Techniques are typically divided into three categories: supervised, unsupervised, and weakly-supervised. The correct method depends on the context and available data.

In a study by Waqas et al. [1] they proposed labeling entire videos instead of individual frames for training a Multiple instance learning model, making the model weakly-supervised. The model identifies when something anomalous happens in a surveillance video where an anomalous act is an illegal activity. In this study, the videos are divided into bags of equal size, the bags are placed into a positive bag and a negative bag. The positive bag has a video section from a positively labeled video and the negative bag from a negatively labeled video. A neural network can then train on these bags, one negative and one positive at a time, and tries to determine which of the clips that are positively labeled. The study also introduces a dataset with 128 hours worth of videos from surveillance videos. The proposed method was found to achieve much higher true positive rates than other methods used in the study.

In unsupervised learning, Jie Yang et al. [2] divided image-level anomaly detection into four groups: density estimation, One-class classification, image reconstruction, and self-supervised classification. The density estimation techniques estimate the distribution of normal images and detect if a newly observed image is abnormal by testing against the established distribution. Typical methods are Gaussian mixture, Nearest Neighbor, and kernel density methods. One-class classification methods attempt to construct a boundary between normal and abnormal data in the feature space, for example, Support Vector Machines. Self-supervised methods attempt to learn unique and more significant characteristics of normal samples through self-supervised training. By learning the features of normal samples, the model can then distinguish abnormal images without the learned characteristics. Image reconstruction methods compress images to a low dimensional latent representation and then attempt to reconstruct the image and minimize the reconstruction error. Here typical methods are different variations of AEs which is also one of the proposed solutions in this report.

1.5 Hypothesis

The expected outcome of the project is two models that can discern if anomalies are in any given video. The chosen weakly-supervised method is expected to be better than the chosen unsupervised method when it comes to accuracy and false positives, because of the labels on the videos.

2 Method

This section describes the methods used to obtain the results presented in the report.

2.1 Preprocessing

A training point is a sequence of images taken from videos. These videos are downscaled from their native 2048p to 124p, then two frames per second are saved to the image sequence. The videos are manually labeled as anomalies or without anomalies, this information is also saved for the image sequence.

2.2 Weakly-Supervised

Labeling data can be very time-consuming, especially for video anomaly detection since deciding what an anomaly is and when it occurs in a video can be hard to decide. Weak supervision is done by assigning video-level labels, so instead of certain frames being labeled the entire video is labeled either normal or anomaly. This makes the labeling easier but instead means that the model will have to learn how to find the anomalies.

2.2.1 Multiple instance learning

In order to learn from weakly labeled examples, Multiple instance learning (MIL) is used. For each video, a bag of video segments is created. If a bag contains a video with an anomaly the entire bag will be labeled as anomalous, otherwise, the bag will be counted as normal. The goal here is for the model to find anomalies in videos without being explicitly trained on where these anomalies occur or what they look like.

2.2.2 MIL for video anomalies

The model is fed batches of bag pairs, one bag with an anomaly, and one normal bag. Each sample, one normal and one anomalous bag, will be fed through the model so that each segment gets scored between $0.0 - 1.0$, where 0 is normal and 1 is an anomaly. This creates a series of scores for each video. The highest score from the normal and anomaly video are used for a ranking loss, in order to push anomalies closer to 1 and normal videos closer to 0. Using some prior knowledge about anomalies, they tend to be rare and should also be continuous in time, therefore, sparsity and temporal smoothness terms are added to the loss function given by equation 1.

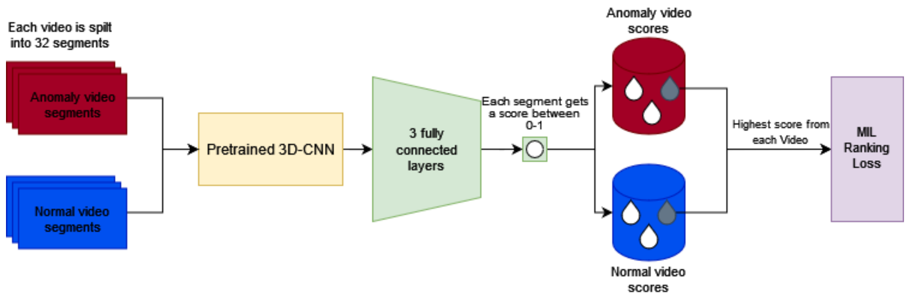


Figure 1: Network topography for MIL video anomaly detector.

$$
\begin{aligned}
l(B_a, B_n) = \max(0, 1 - \max_{i \in B_a} f(V_a^i) \max_{i \in B_n} f(V_n^i)) \\
+\lambda_1 \sum_{i}^{(n-1)} (f(V_a^i) - f(V_a^{i+1}))^2 + \lambda_2 \sum_{i}^{n} f(V_a^i)
\end{aligned}
\tag{1}
$$

B_a & B_n in Equation 1 are the anomalies and the normal bag with video segments, V^i is a segment from a bag, f is the scoring model, and $\lambda_1 \& \lambda_2$ are constants for the sparsity and temporal smoothness. The MIL ranking loss was first used by Waqas et al. [1] whose model was used as the base. Figure 1 shows a diagram of how the model is trained.

2.3 Unsupervised

Aside from the weakly-supervised solution, an unsupervised solution is used. Unsupervised learning refers to machine learning algorithms that are trained using unlabeled data. Unsupervised algorithms discover underlying structures and patterns without human intervention. The ability to discover similarities and differences in data makes unsupervised learning ideal for applications such as anomaly detection, because, it can discover unusual data points [3]. In supervised learning, a data set D consists of M observations $\mathbf{x}_i$ that have one label or expected value $\mathbf{y}_i$, with $i = \{1,..., \text{M}\}$. In unsupervised learning, a data set can instead be described as $D = \{\mathbf{x}_i | i = 1, ..., M\}$ [4], where each observation $\mathbf{x}_i$ is unlabeled.

2.3.1 Autoencoder

An autoencoder (AE) is a special type of neural network, that is primarily designed to learn a compressed and informative representation of some input. This is by compressing the input and then reconstructing it back as similarly as possible. An AE consists of three main components which are: an encoder A, a decoder B, and a *latent feature representation*. The problem is to learn

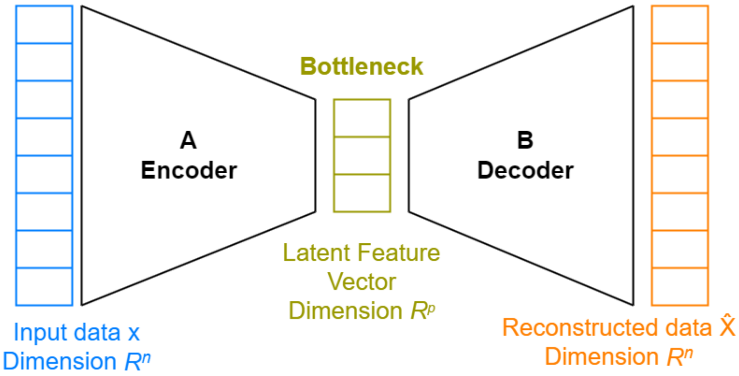

Bottleneck
A
Encoder
B
Decoder
Latent Feature Vector Dimension R^p
Input data x Dimension R^n
Reconstructed data $\hat{X}$ Dimension R^n

Figure 2: Autoencoder

the functions $A : \mathbb{R}^n \to \mathbb{R}^p$ and $B : \mathbb{R}^p \to \mathbb{R}^n$ to minimize the reconstruction loss ΔE, where n is the size of the input, and p is the size of the latent feature representation [5]. The described AE is depicted in Figure 2.

Training the AE to minimize the reconstruction error means learning the functions A and B to minimize the difference between the input $\mathbf{x}_i$ and the reconstruction $\hat{\mathbf{x}}_\mathbf{i}$. The reconstruction can be formulated as $\hat{\mathbf{x}} = B(A(\mathbf{x})) = f(x)$, and the perfect reconstruction would be to learn the identity function which would not be very informative for this project. Therefore, a *bottleneck* is implemented, which is achieved by making the latent feature dimension much lower than the input dimension [4].

2.3.2 Convolutional Autoencoder

To be able to capture 2D structures in images and videos, the AE can be adjusted by introducing convolutional layers. The adjusted AE is called the Convolutional Autoencoder (CAE) and in addition to convolutional layers, it also has shared weights that preserve spatial locality. The reconstruction is obtained by using $\hat{\mathbf{x}}_i = Act(\sum h * W + b)$, where * is a convolution and h represents a feature map given by $h = Act(\mathbf{x} * W + b)$ [6].

The training of the CAE is just as for other neural network models done by using a backpropagation algorithm. A paper by Jonathan et al. [6] proposes a solution that minimizes the mean square error, and in another paper, Usha and Vamsidhar [7] propose a CAE that uses Binary Cross Entropy as its loss function instead. To obtain translation-invariant representations, a max-pooling layer is often introduced. Max-pooling calculates the maximum value in each patch of each feature map, see Figure 4. The max-pooling down-samples the data by a constant factor and has been shown to improve the filter selectivity. In other words, highlighting the most present features in a patch [6]. Another layer that can be applied to improve the performance of deep neural networks is Batch Normalization (BN). BN is a technique to

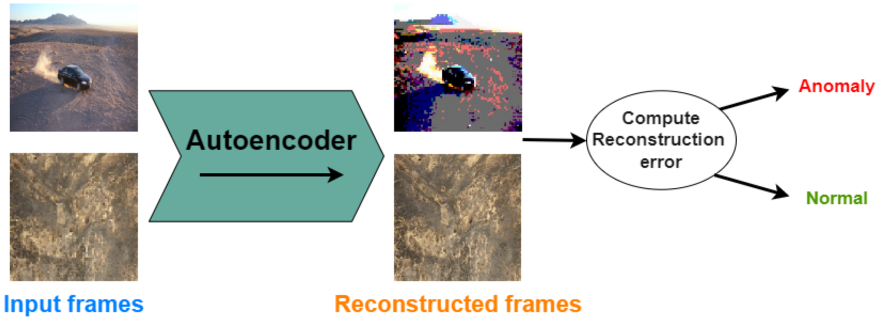


Figure 3: Anomaly detection with an autoencoder

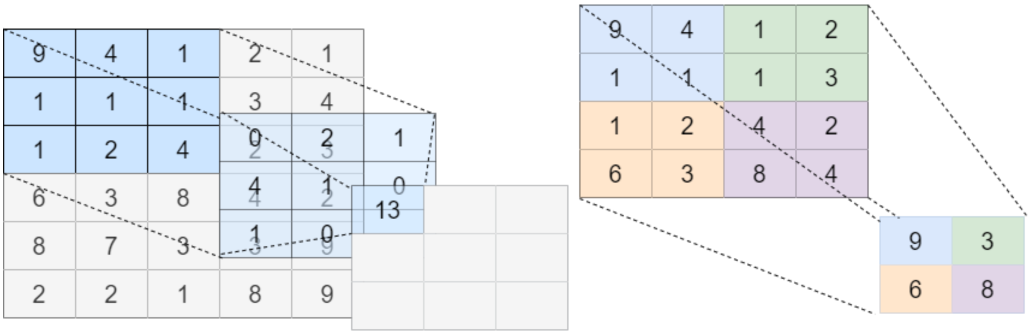


Figure 4: Convolutional layers and Map Pooling

standardize the inputs to layers and has the effect of stabilizing the learning process. Wang et al [8] proposed an AE solution that applied a batch normalization layer before the activation functions. This is to generate a stable distribution of activation values during training and to avoid overfitting.

To classify a video or image as an anomaly, a threshold value is selected. The value decides how much the output reconstructions are allowed to deviate from the input, see Figure 3. A large threshold means that reconstructed frames are required to differ significantly from the input to be classified as an anomaly. A too-large threshold value may lead to anomalies being wrongly classified as normal. On the other hand, a small threshold means that the reconstructed frames should be similar to the normal frames, and can result in classifying normal frames as anomalies. The threshold is selected based on the average error on a validation set and tuned in such a way that the model separates the anomaly and the normal frames with high accuracy.

2.4 Evaluation

The problem of identifying anomalies in images or video frames can be seen as a classification problem. The models will therefore be evaluated based on their accuracy in classifying anomalies and normal frames. As anomalies are generally understood to be rare, the data set will consist of mostly normal data points, which means that the models are primarily evaluated based on their ability to separate the classes. With that said, the models will also be evaluated based on f1-score, precision, and recall.

3 Results

In this section, the results given of each method will be presented.

3.1 Weakly-supervised

The following section presents the used datasets, the model, the model architecture, and performance metrics for the Weakly-supervised method

3.1.1 Dataset

At first, the model was trained with the custom drone footage, however, the model did not learn well at all, so instead, the UCF-crimes dataset was used to train and evaluate the weakly-supervised MIL model. The dataset was created by Waqas et al. [1] and uses stationary footage labeled into eight different classes, one normal class, and seven anomalous classes: Burglary, fighting, arrest, abuse, arson, and explosion. The full dataset contains around 1300 hours of footage, due to computational and memory limitations, only a subset of all normal and anomalous videos was used.

3.1.2 Model

The MIL model uses a pre-trained 3D-CNN called *Seperable 3D CNN* for feature extraction as proposed by Xie et al. [9]. The features are fed into three fully connected layers to compute the anomaly scores, as seen in Figure 1. Each batch consisted of 30 bag pairs, where each bag was a video that had been split into 32 clips, these clips were the segments that the model evaluated and scored.

The score the model gives to five videos of each anomaly type and 35 videos labeled normal can be seen in Figure 5.

3.1.3 Evaluation

Below is the evaluation of the model.

Table 1: Results for weakly supervised

Dataset	Accuracy	Precision	Recall	F1-score
UCF-Crimes	78%			
Anomaly		87%	57%	69 %
Normal		68 %	91 %	78%

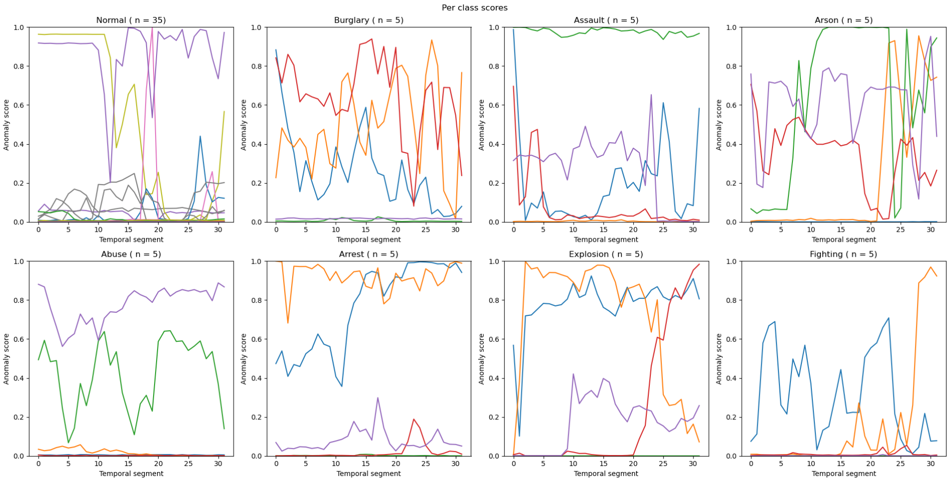


Figure 5: Shows the temporal scores for the 35 normal and 35 anomaly test videos.

Table 2: Confusion matrix for UCF-Crimes

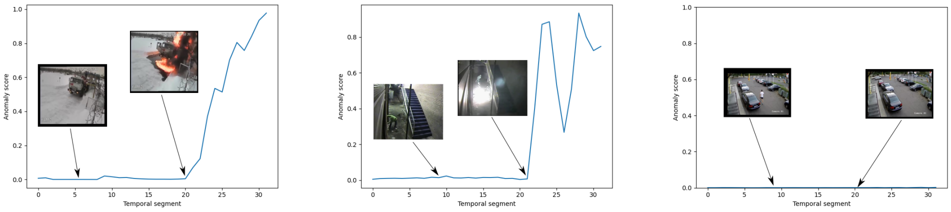
	Predicted Anomaly	**Predicted Normal**
True Anomaly	20	15
True Normal	3	32

3.2 Unsupervised

The following section presents the used datasets, the model, the model architecture, reconstructions, and performance metrics for the unsupervised method.

3.2.1 Dataset

The evaluation was done on two different created datasets. Dataset 1 was the original dataset where the anomalies could be in different colors or images from different environments, see Figure 8. Dataset 2 consisted of 8 videos filmed from the same area from different angles and the training was performed on all the videos. The anomalies consisted of shorter clips from the videos where self-produced anomalies were edited in, see Figure 9. The two datasets consisted of 200-500 images of resolution (124x124). 80% of the images were assigned to a training set, 10% for a validation set to determine the threshold, and 10% for final unbiased testing.

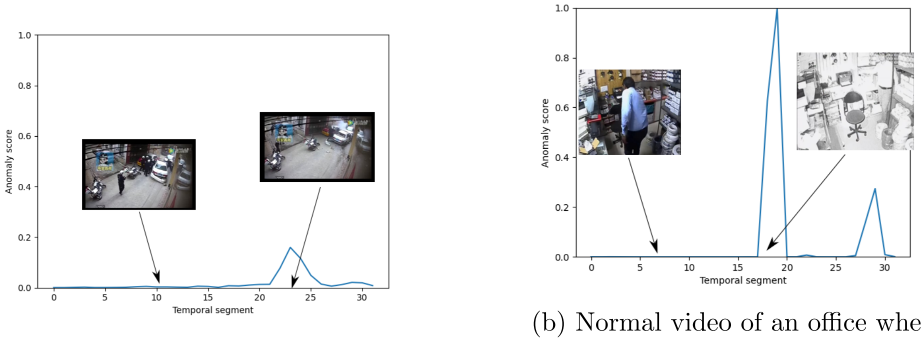


(a) Video of explosion anomaly.

(b) Video of arson anomaly.

(c) Normal video of a parking lot.

Figure 6: Correct prediction



(a) Video of arrest anomaly.

(b) Normal video of an office where the light is turned off.

Figure 7: Incorrect predictions.

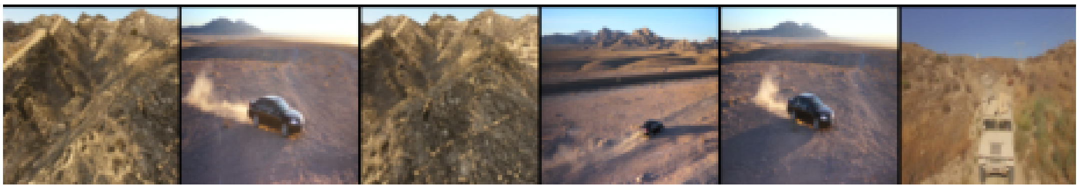


Figure 8: Example images from dataset 1

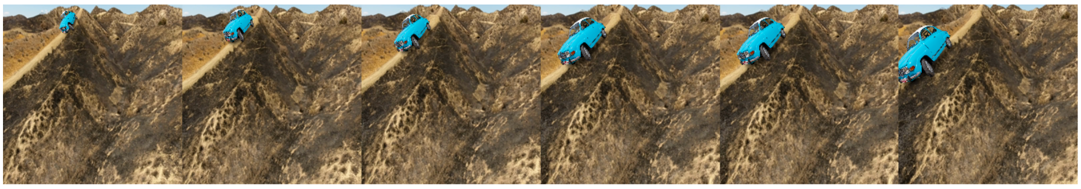


Figure 9: Example images from dataset 2

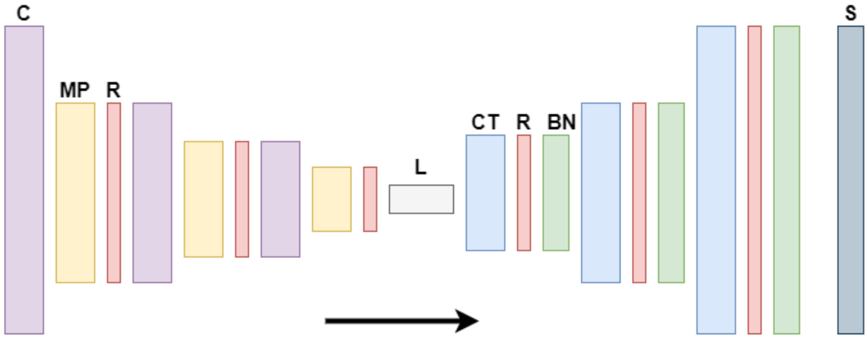


Figure 10: Network Architecture, C=convolution layer, MP=max-pooling, R=ReLU, L=Linear, CT=Transposed Convolution, BN=Batch Normalization, S=Sigmoid

3.2.2 Model

The final CAE model consisted of three convolutional layers, three max-pooling layers, and three transposed convolutional layers to be able to reconstruct the data. The network also contained six ReLU activation functions in both the encoder and the decoder. Since input images were normalized between (0,1), the network ended with a sigmoid activation function to be able to calculate the loss. The loss functions tested were MSE and BCE and it turned out that MSE gave the best results in the experiments. To see the complete network with the corresponding parameters, see Appendix 1. Settings that were used were learning rate=0.01, the number of epochs=200, and batch size=20. As an optimization technique and to train the networks, the Adam optimizer was chosen which is as a replacement optimizer for gradient descent.

3.2.3 Evaluation

The calculated metrics when running the model on dataset 1 and dataset 2 are shown in Table 3. The accuracy represents how close the anomaly or normal classifications were to their true value. Anomaly is seen as the true positive.

Table 3: Results for unsupervised

Dataset	Accuracy	Precision	Recall	F1-score
Dataset 1	78.7%	100%	62.8%	77.2%
Dataset 2	50.9%	66.7%	16.7%	26.7%

The classification of anomalies and normal images on the test data from both datasets is shown in Figure 4 and Figure 5.

Table 4: Confusion matrix for dataset 1

	Predicted Anomaly	**Predicted Normal**
True Anomaly	22	13
True Normal	0	26

Table 5: Confusion matrix for dataset 2

	Predicted Anomaly	**Predicted Normal**
True Anomaly	10	50
True Normal	5	47

3.2.4 Reconstructions

In Figure 11 and Figure 12 the original input are compared with reconstructions for number of epochs=(0, 10, 20, 200). As the reconstructions are of the same dimension as the input images, a pixel difference can easily be calculated. The difference can be used to show where the images differ the most and to identify anomalies, see Figure 13.

4 Discussion

In this section, the research questions, the results, and future work are discussed.

4.1 Research questions

The most significant difference between the two models is the weakly-supervised and unsupervised approaches. One of the models requires labeled data to some extent and the other does not. If the data needs to be labeled, it requires someone to do it, which can be time-consuming and tedious. Since most machine learning models require large amounts of data, labeling can be a very huge process and can become a large part of a project. In the context of anomaly detection, the normal is often by far the largest part, and getting through large data sets to find unusual events can be difficult.

The pros of using AE are especially that they can work in an unsupervised manner and there is no need for labeling data. Autoencoders can have

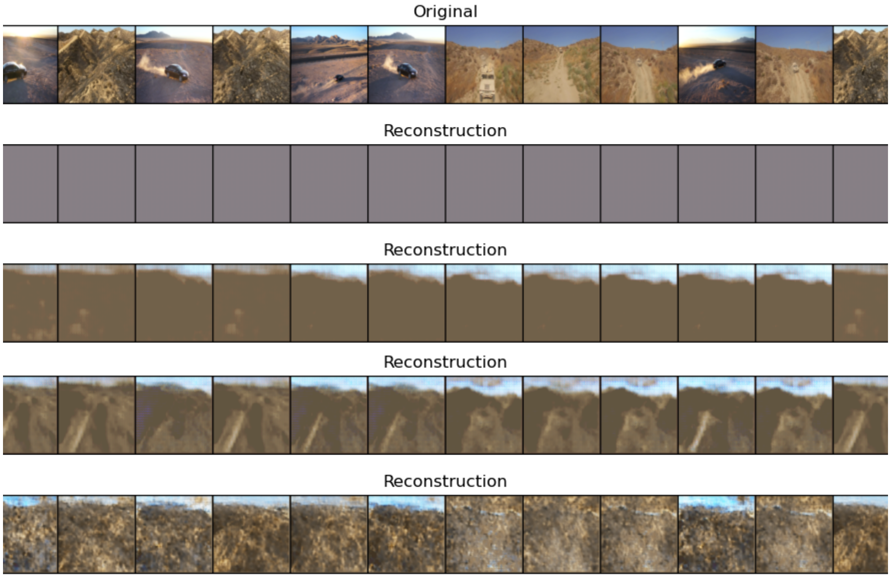


Figure 11: Batch of input images from dataset 1 and reconstructions after 0, 10, 20, 200 epochs

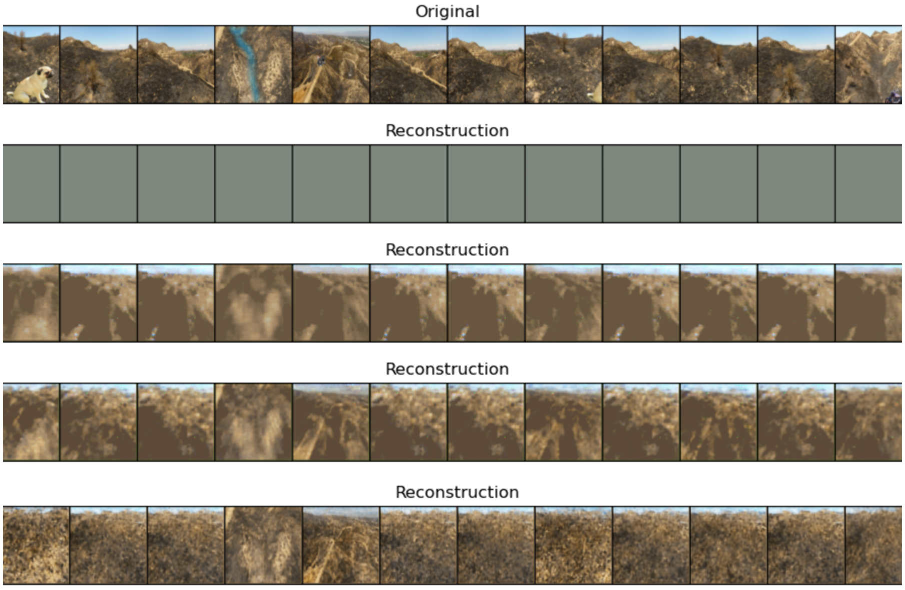


Figure 12: Batch of input images from dataset 2 and reconstructions after 0, 10, 20, 200 epochs

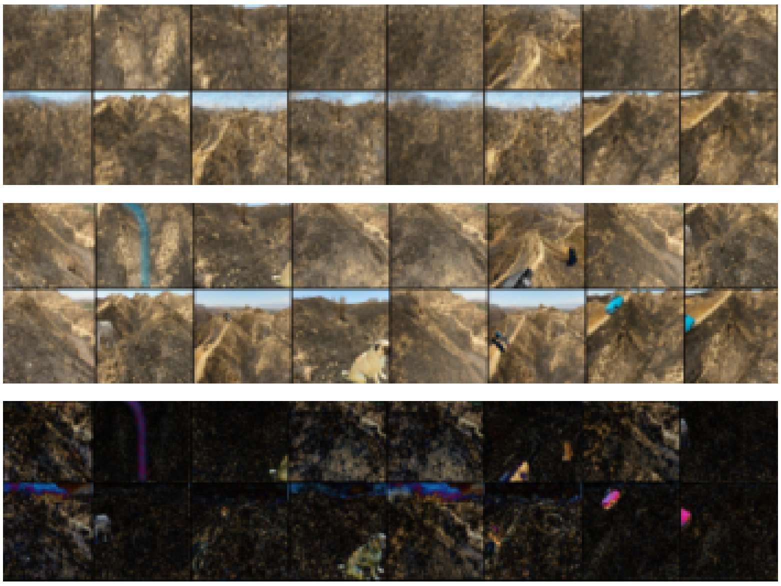


Figure 13: Input images, reconstructed images, and the pixel difference of an image batch from dataset 2.

different areas of use which make them flexible, but they are not necessarily the best in any area. Autoencoders work well for extracting features as seen in Figure 11 and Figure 12 where reconstructions are reasonably similar even though they are reconstructed with significantly fewer dimensions. The drawbacks are that they require a lot of data, and there are many parameters to tune where changes may have little or great significance. The relation between the training data and the test data is strong and if the different data sets are not representative of each other, strange results can be obtained. The results of the models can be hard to interpret and can give you an average of the input, reconstruct the images exactly, or something in between. Since the dimensions of the input and output are the same, comparisons can easily be made, which can be beneficial when, for example, locating deviating pixels. This is shown in Figure 13 where we can see that e.g. the blue line, the turquoise car, and the dog are clearly captured.

As the MIL model is a weakly-supervised method, it has similar drawbacks as other supervised methods when it comes to classification. The model has a difficult time classifying anomalies that are not distinct and do not have enough training data. Therefore, this method needs a lot of data in order to be accurate. On the other hand, the model can be trained to reliably find anomalies, given that the anomalies are well represented in the dataset. This gives the method a very high rate of true positives and a low rate of false positives. Besides this, the score given to classify each bag of pictures is directly a certainty the model has of the classification, however, it does not tell where in the picture the anomaly lies.

During the course of the project, the group members have learned and experienced working with a real AI and machine learning project. The project group was divided into two parts and from that, the members learned and tested different techniques. We have realized and learned that access to data can be complicated and that you may need to create your own datasets. We have learned how to create an image dataset using movies, that can then be used to train and test different models. We have also learned that you cannot always expect your customer to provide the data that is needed for your solutions.

The group that worked with AE has learned about its use cases which are not only anomaly detection. AE can also be used for data denoising, dimensionality reduction, data generation, data compression, and feature extraction.

The group that worked with Multiple instance learning learned that while a method usually is used for traditional supervised classification of data, it may have more uses, for example in this project, where MIL is used to classify videos through weakly supervised data.

4.2 Results

In this section, the results for both of the models are discussed. The difference between results from papers, why there is a difference, and what can be done next to improve the results are also discussed.

4.2.1 Weakly-supervised

As seen in figure 5 the model was especially good at classifying arson and explosion, while fighting and abuse were only classified correctly two out of five times. This is likely because arson and explosion are quite distinct compared to other anomalies and the normal video, as they are the only ones that contain fire or smoke. Fighting and abuse on the other hand are both less distinct, involving shorter disputes such as a single punch toward a person or pet. This makes it much more difficult for the model to differentiate these anomalies from the normally labeled video. In order for the model to better classify these anomalies there would likely need to be even more data of this type.

Overall, the model performs well when classifying the UCF-dataset, as seen in Table 1 and Table 2. Potential improvements would be more to use more data, trying out different backbones for feature extraction, and trying different model configurations for the linear classifier. Due to time constraints, there was almost no hyperparameter tuning. Different lengths of the clips fed to the 3D-CNN, different learning rates, and trying different

λ values for the smoothness and sparsity could possibly give better performance. Since the 3D-CNN weights were frozen, all the features were computed beforehand for faster training, however, testing an unfrozen network could possibly allow for better results.

From doing some basic analysis of the results, it seems that the model performed best when there were distinct and colorful anomalies, like fires or smoke as seen in figure 6a or 6b. The model also seemed sensitive to abrupt scene changes like what occurs in 7b where someone turns off the light. The transition from light to dark triggers the anomaly score to spike which it should not do.

4.2.2 Unsupervised

The results in Table 3 are not terrible but they are not good either. It turned out that the result differed between the two datasets where the f1-score obtained for Dataset 2 was really low in comparison to Dataset 1. One problem with the results is that they varied a lot from experiment to experiment. This is probably because the results are dependent on how the data is distributed between training, validation, and testing. A reason for this behavior is probably because the data set is too small. A more reliable result could have been given if a larger and more qualitative dataset was available. Available movies found were limited and the movies could only be sampled to a certain limit to not overfit. It was also difficult to store too many videos and images due to the availability of data memory.

The results in Table 4 and 5 shows that for true normal frames, it was unlikely that the model would predict wrong. This could mean that the model is very adapted to the training data, which is the normal environment. It is not necessarily bad because the most important thing for the model is to be able to separate normal and anomaly frames and not reconstruct frames perfectly. It was more difficult to predict anomalies as the model could reconstruct most parts of images except the anomaly in a good way, which gave a small reconstruction loss.

The objective of the project was to recognize anomalies in video streams, but for the unsupervised approach, the project has rather become for images. First, movies are framed to be able to create image sets, then the models can be trained on these images. The models do not accept a continuous flow of images, first, videos must be preprocessed. For the AE, support was first implemented for images where the training data consisted of randomly selected images from the normal environment. This was done to first see if it was possible to extract characteristics from a certain environment. Then, support was implemented to take 10 images in a sequence to resemble a movie. The first approach gave better results and the second approach was significantly slower.

4.2.3 Future work and improvements

We believe that the CAE results we obtained were reasonably consistent with our expectations when it comes to reconstructing frames and extracting important features. Some results were a little disappointing but it is hard to say if that depends on the model or the dataset. To improve the results further, the AE solution could have been made more advanced by adding components, for example, M. Zaighan et al. [10] proposed a solution with an AE as a generator in a generator/discriminator network and not as the complete solution. Another solution could be to use Elvan and Osmans solution [11], to use optical flow along with the CAE, which was proved to outperform CAE by 14.3 percentage points.

5 Contributions

- Ludvig - Implemented the unsupervised model, contributed largely to the unsupervised method, and the unsupervised results, and created all images, figures, and tables used for unsupervised. Was in part responsible for the unsupervised approach.
- Felix - Implemented the weakly-supervised model, contributed to created all images, figures and tables used for the weakly-supervised results. Was in part responsible for the weakly-supervised approach.
- Peter - Contributed in the form of planning, gathering video material for dataset, taking part in creation of the weakly-supervised model, and writing part of report, pertaining to the weakly-supervised model. Was in part responsible for the weakly-supervised approach.
- Hannes - Contributed by creating data-sets, taking part of creating the unsupervised model, testing different loss functions, writing parts the report with a lot of focus on the method chapter. Was in part responsible for the unsupervised approach.

References

[1] Waqas Sultani, Chen Chen, and Mubarak Shah. Real-world anomaly detection in surveillance videos, 2018.

[2] Jie Yang, Ruijie Xu, Zhiquan Qi, and Yong Shi. Visual anomaly detection for images: A survey. https://arxiv.org/pdf/2109.13157.pdf.

[3] IBM Cloud Education. Unsupervised learning. https://www.ibm.com/cloud/learn/unsupervised-learning, 2020-09-21.

[4] Umberto Michelucci. An introduction to autoencoders. https://arxiv.org/pdf/2201.03898.pdf, 2022-01-12.

[5] Raja Giryes Dor Bank, Noam Koenigstein. Autoencoders. https://arxiv.org/pdf/2003.05991.pdf.

[6] Jonathan Masci, Ueli Meier, Dan Cireşan, and Jürgen Schmidhuber. Stacked convolutional auto-encoders for hierarchical feature extraction. In Timo Honkela, Włodzisław Duch, Mark Girolami, and Samuel Kaski, editors, *Artificial Neural Networks and Machine Learning – ICANN 2011*, pages 52–59, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[7] Usha Ruby and Vamsidhar Yendapalli. Binary cross entropy with deep learning technique for image classification. *International Journal of Advanced Trends in Computer Science and Engineering*, 9, 10 2020.

[8] Jinrui Wang, Shunming Li, Baokun Han, Zenghui An, Yu Xin, Weiwei Qian, and Qijun Wu. Construction of a batch-normalized autoencoder network and its application in mechanical intelligent fault diagnosis. *Measurement Science and Technology*, 30(1):015106, dec 2018.

[9] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, pages 318–335, Cham, 2018. Springer International Publishing.

[10] M. Zaigham Zaheer, Arif Mahmood, M. Haris Khan, Fisher Yu Mattia Segu, and Seung-Ik Lee. Generative cooperative learning for unsupervised video anomaly detection. https://openaccess.thecvf.com/content/CVPR2022/papers/Zaheer_Generative_Cooperative_Learning_for_Unsupervised_Video_Anomaly_Detection_CVPR_2022_paper.pdf.

[11] Elvan Duman and Osman Ayhan Erdem. Anomaly detection in videos using optical flow and convolutional autoencoder. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8936359.

6 Appendix 1

Component	In_channel	Out_channel	Kernel	Stride	Padding	Dim
Input	-	-	-	-	-	(124x124)
Conv2d	3	15	(3,3)	(1,1)	(1,1)	(124x124)
MaxPool2d	-	-	(2,2)	-	0	(62x62)
ReLU	-	-	-	-	-	(62x62)
Conv2d	15	30	(3,3)	(1,1)	(1,1)	(62x62)
MaxPool2d	-	-	(3,3)	(1,1)	(1,1)	(31x31)
ReLU	-	-	-	-	-	(31x31)
Conv2d	30	60	(3,3)	(1,1)	(1,1)	(31x31)
MaxPool2d	-	-	(3,3)	(1,1)	(1,1)	(15x15)
ReLU	-	-	-	-	-	(15x15)
Flatten	-	-	-	-	-	15*15*60
Linear	-	-	-	-	-	15*15
Linear	-	-	-	-	-	15*15
Unflatten	-	-	-	-	-	(15x15)
ConvTranspose2d	60	30	(3,3)	(2,2)	0	(31x31)
ReLU	-	-	-	-	-	(31x31)
BatchNorm2d	30	30	-	-	-	(31x31)
ConvTranspose2d	30	15	(3,3)	(2,2)	1	(61x61)
ReLU	-	-	-	-	-	(61x61)
BatchNorm2d	15	15	0	3	8	(61x61)
ConvTranspose2d	15	3	(3,3)	(2,2)	output 1	(124x124)
ReLU	-	-	-	-	-	(124x124)
BatchNorm2d	3	3	0	2	10	(124x124)
Sigmoid	-	-	-	-	-	(124x124)