Anomaly Detection in Egocentric Traffic Videos using Deep Learning

Mostafa Kamal
Department of Computer Science, Graduate
University of Massachusetts Lowell
Lowell, Massachusetts, United States
mostafa_kamal@student.uml.edu

***Abstract*—Spotting anomalies in ego-centric driving videos is important in keeping drivers safe. It plays a significant role in furthering autonomous driving systems. By identifying unusual situations on the road, we can increase safety measures and improve the overall driving experience and quality. This task becomes more challenging due to the rarity and diversity of anomalous events, and dynamic camera motion in ego-centric videos. In our research, we explore two different methods for detecting traffic anomalies using the DoTA dataset. The first method uses an unsupervised Convolutional Autoencoder trained on normal traffic videos to reconstruct normal driving patterns. The second is a supervised approach that utilizes a VideoMAE transformer to identify and classify anomalies in our traffic data. Our Conv-AE used 10 stacked greyscale images to achieve an AUC-ROC of 0.556. On the other hand, the VideoMAE-based classifier outperformed the baseline by 49%, achieving an AUC-ROC of 0.828. Further evaluation of VideoMAE yielded 80.9% precision and 73.5% F1 score for the anomaly class. Our results show that transformer-based approaches far outperform traditional anomaly detection, although further improvement on it can be achieved through domain-adaptive pretraining.**

***Index Terms*—Anomaly Detection, Convolutional Autoencoder, Video Vision Transformer, VideoMAE, MVM.**

I. Introduction

Anomalies in videos are events that deviate from what is considered as standard, normal, or expected. They signify irregularities in the videos and are most commonly used in areas such as health monitoring, surveillance systems, intrusion detections systems, and autonomous driving. However, unusual events in long videos occur with very low probability, leading to significantly more normal data than anomalies. In this paper, we focus on traffic videos from the driver's point of view (ego-centric). This is essential in autonomous vehicles as failure to do so can lead to life-threatening injuries. This creates a pressing need for automated systems to detect anomalies in ego centric moving videos. Formally, we define this problem as a Traffic Anomaly Detection task (TAD), which involves detecting hazardous or abnormal events from ego-centric dashcam footage. Additionally, this task faces another critical challenge which is detecting anomalies in a dynamic moving videos.

In this work, we explore two key questions: first, whether an anomaly is present in a video, and second, when does that anomaly occur. We explore two different approaches to determine whether an anomaly occurs in a certain clip or frame. The first method employs unsupervised learning using a Convolutional Autoencoder, where anomalies are detected using reconstruction loss. The second method uses a supervised learning approach using a pretrained Video Vision Transformer (ViViT) followed by a linear classifier. The performance of Convolutional Autoencoder is evaluated by the AUC-ROC metrics [1] while the ViViT-based classifier is also assessed on AUC-ROC, and F1 scores. Both models are trained on DoTA, a dataset containing dynamic ego centric driving videos.

II. Related Work

The detection of anomalies has evolved significantly over the past decade. This has progressed from models detecting anomalies from fixed camera surveillance systems to ego centric moving camera approaches and more recently using foundation models-based approaches. We organize related work into three main categories, traditional anomaly detection systems, anomaly detection in dynamic moving videos, and modern foundation models.

A. *Traditional Anomaly Detection Systems*

Early video anomaly detection research focused on fixed surveillance camera scenarios, where the camera remains stationary and monitors a specific area. These foundational approaches established key principles that later influenced other methods. They provide insights on how anomalies can be detected in an unsupervised manner, learning the patterns of normality rather than explicitly modeling each possible situation, which can be very impractical.

Hasan et al. [2] approached this problem by learning a generative model for regular motion patterns. They proposed a fully convolutional autoencoder trained on stacked frames to capture temporal patterns and reconstruct normal frames with little to no supervision. The key intuition is that an encoder trained on normal videos captures regularities and motion patterns, producing low reconstruction error on normal frames while generating high error for irregular events or anomalies. Building on this reconstruction paradigm, Chong and Tay [4] proposed a spatiotemporal autoencoder that explicitly modeled both spatial and temporal dimensions. Their encoder-decoder architecture first captured spatial features using convolution

layers, then used a Conv-LSTM to learn the temporal patterns of the spatial features. Like Hasan's approach, they also trained their autoencoder on normal frames. Both methods used reconstruction error, the per-pixel intensity difference between the input and reconstructed frame, to score and evaluate their anomaly detection model.

Rather than reconstructing the current frames, Medel and Savakis [3] introduced a predictive approach using a Convolutional LSTM network. Their method predicted future frames based on observed sequences, following the same principle of training on normal video sequences. This causes the prediction of anomalies to diverge from the ground truth with each timestep, causing a high reconstruction error.

However, these approaches have limitations in real-world deployment. An analysis by Muhammad et al. [5] revealed that models may often flag anomalies due to superficial factors and can be misleading (e.g., lighting, different regions, background, unknown artifacts, camera shake).

B. Anomaly Detection in Dynamic Ego-Centric Videos

While providing good results on stationary videos, ego-centric dash cam footage poses unique challenges that fixed-camera methods cannot handle. These challenges are not limited to camera movement with the vehicle, dynamic background shifts, objects appearing to change positions, limited field of view, and many more. These challenges require methods specifically catered to catch anomalies from the driver's perspective.

Yao et al. [6] made the first major contribution to unsupervised traffic accident detection in first-person videos. Their main contribution to this challenge is predicting the future location of the traffic participants. This is a shift from frame-centric [2]–[4] to an object-centric approach. Their methods first detect and track individual traffic participants and then predict their future locations based on observed trajectories. Their architecture is inspired by a multi-stream RNN encoder-decoder architecture, based on GRUs, to predict the future bounding boxes of each object and the ego vehicle odometry. The output of the ego-motion cues is combined with the object-level predictions to estimate the prediction accuracy and consistency of objects' future location. In short, this allows them to flag deviations from the normal movement behaviors as an anomaly. Building on their previous work and following an unsupervised approach, Yao et al. [7] addressed the limitations of some anomalies that don't involve unusual motion. They extended their framework to predict what the object will look like and where will they be in the future, focusing on both appearance and location simultaneously.

C. Modern Approaches: Foundation Models and Transformers

Arnab et al. [9] introduced Video Vision Transformers (ViViT), adapting the transformer network (vision transformer) for video understanding. Unlike Convolutional Neural Networks, vision transformers use self-attention to model relationships across both spatial and temporal dimensions. Just like Vision Transformer, ViViT divides the video frames into spatial and temporal patches and then feeds them into the network. This allows the model to learn through attention. Building on this architecture, Orlova et al. [8] demonstrated a simplified approach where they used a pretrained ViViT to perform a Traffic Accident Detection task with high accuracy. They performed a supervised approach using a pretrained ViViT to classify video frames as either containing an anomaly or not. Their architecture consists of an ViViT encoder and a classification head to output the predictions. To achieve good performance, they used the DAPT pretraining approach, which involved further pretraining the ViViT model to adapt to the driving domain.

III. Methodology

This section presents the proposed approach for anomaly detection in traffic videos. The methodology section encompasses 3 main parts: dataset details and preprocessing, convolution autoencoder for unsupervised reconstruction-based learning and video vision transformer for supervised learning.

A. Dataset Details

1) Dataset Overview: This project utilizes the detection of traffic anomaly (DoTA) dataset which is specifically designed for traffic anomaly detection (TAD) detection in dash-cam videos. The dataset was introduced by Yao et.al [cite], where it's one of the most comprehensive traffic anomaly detection datasets. Unlike other TAD dataset, this dataset captures the unique challenge of dynamic, ego-centric driving videos where both the camera and scene elements are in constant motion.

The DoTA dataset was collected from YouTube videos, resulting in about 4,677 video clips collected from diverse real world driving scenarios. The dataset was extracted at 10 fps and represents unique driving conditions and weather patterns. Additionally, each frame is annotated to indicate whether it is a normal or contains a traffic accident. The traffic accidents are also composed into two different categories, ego-centric and non-ego-centric where which ego centric means that the driver was involved in the crash and vice versa. They also provide spatial annotations for each object in the scene. Upon further exploring the dataset, we find that there are about 66.2% of Normal frames and 33.8% percent of anomalous/accident Frames. Additionally, there was a large class difference between night and day which may cause the model to be biased on day frames. Below are the metrics that we collected on the dataset.

2) Data Partitioning: The dataset was partitioned into training, validation, and testing sets with an 80/10/10 split. This random partitioning of videos ensures that there is no data leakage and maintains a balanced ratio of anomalous and normal videos across the training, validation, and testing sets.

3) Data Pre-Processing Pipeline: We implemented a preprocessing pipeline that ensures the data is ready for deep learning model training. First, each image was resized to 224×224, and then we applied model-specific preprocessing steps.

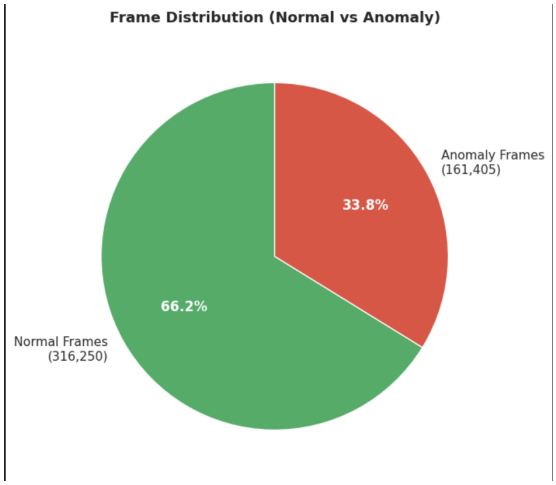

Fig. 1: Frame Ratio (Normal vs Anomaly)

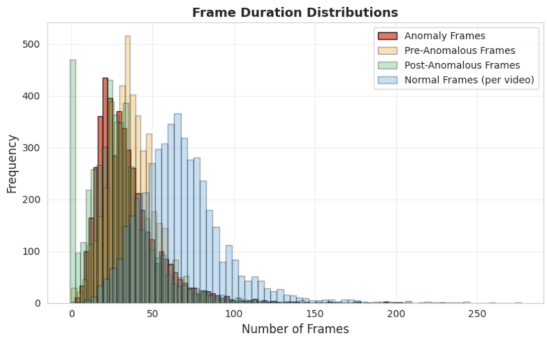

Fig. 2: Frame Distribution

For the convolutional autoencoder, we first converted the image to grayscale and computed the mean and standard deviation. We then normalized the frames using the calculated values and ensured it was a z-score-normalized tensor. To capture temporal regularities, we stacked 10 consecutive frames along the channel dimension, resulting in a tensor of shape (10, 224, 224). We implemented a dataset class for the convolutional autoencoder that reads the data on the fly. To further augment our data, we intentionally did three frames of overlap to increase the number of our training samples. We design our autoencoder to learn normal driving patterns, so we only extract normal stacked frames for training and omit anomalous ones.

For the video vision transformer, we optimize training efficiency by constructing 16-frame sequences from the original video data and labeled the sequence as an "anomaly" if it contained any anomalous frames and "normal" otherwise. Then, the sequences were processed using the pretrained model's preprocessing pipeline, which transformed the raw frames into standardized 4D tensors with dimensions (frames, channels, height, width). Note the height and width are defaulted to 224×224 shape. Finally, we implemented a custom dataset class to load the preprocessed sequences and their corresponding labels during training.

B. Conv-AE: Unsupervised Anomaly Detection

The convolutional autoencoder serves as a significant component in our proposed anomaly detection task. The convolution autoencoder is trained unsupervised to extract meaningful features and learns to reconstruct the normal frames. Unlike supervised approaches that require a vast amount of labeled data, this approach requires no labels at all and trains the model completely unsupervised on normal frames.

1) Auto-encoder Overview:

a) Convolution Layers: A 2D convolution layer applies a learnable filter across the spatial dimension of an input feature map. For an input feature map $X \in \mathbb{R}^{H \times W \times C_{\text{in}}}$ and a convolutional kernel $H \in \mathbb{R}^{K \times K \times C_{\text{in}}}$, when applied on the feature map, the filter slides over the input and creates an activation map where we can abstract high-level features. Given an input feature map X and a kernel H, the convolution layer would compute the inner product between the $k \times k$ patch of the images and the kernel, and output one activation value. Mathematically, this operation corresponds to cross-correlation.

$$G[i,j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} H[u,v]X[i+u, j+v] \tag{1}$$

Here X is the image where we are applying our learnable filter H.

b) Max Pooling: Max Pooling is a down-sampling operation used in convolution layers to reduce the spatial dimension. It works by sliding a fixed-size window and taking the maximum value of that window. This preserves the most important values. Given an activation map X, a filter of size k, and stride s, we define the output Y as taking maximum value from the $k \times k$ patch of the input activation map.

$$Y(i,j) = \max_{0 \le m,n \le k-1} X[i \cdot s + m,\ j \cdot s + n] \tag{2}$$

c) Transpose Convolution: An essential component of an encoder-decoder framework, a transpose convolutional operation is a learnable up-sampling operation that increases the spatial dimension of the image. Unlike a standard convolution, which reduces the spatial dimension of an image, a transpose convolution does the opposite. Essentially, each input to the feature map would assign a weight to the learnable filter, increasing the spatial dimension.

2) ConvAE Architecture Details: The proposed convolutional autoencoder (ConvAE) implements a symmetric encoder-decoder architecture explicitly designed to learn normal-related driving patterns. The networks comprises of two main components: an encoder to reduce the dimensionality of the input image into a compact latent representation, and a decoder that uses that latent representation to reconstruct the input back into the original spatial representation.

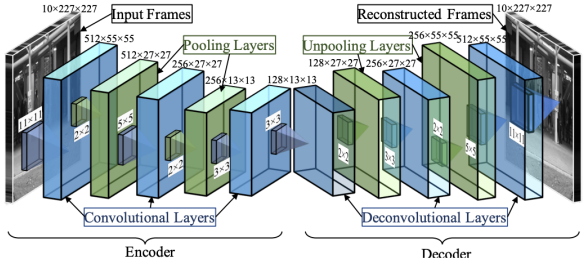

Fig. 3: ConvAE Architecture details [2]

Convolutional Encoder: The network first transforms the input frames through three convolutional blocks, each containing a convolutional layer, batch normalization, an activation function, and a max pooling operation. In the first convolutional block, we apply 512 11×11 kernels with a stride of 4. This enables us to achieve a large receptive field that captures low-level visual features quickly. The second convolutional block further refines the features by applying 256 filters of size 5×5 with a stride of 1 and a padding of 2 to preserve the spatial dimensions. Lastly, the final convolutional block gives us the bottleneck latent representation using 128 3×3 filters with no maxpooling operation. Overall, for an input of size $10 \times 224 \times 224$ (where 10 refers to 10 stacked greyscale frames), the bottleneck size becomes $128 \times 13 \times 13$.

Convolutional Decoder: Like the encoder, the decoder also has three deconvolutional blocks with the same filter and stride sizes. It uses transpose convolution operations to increase spatial resolution while learning meaningful feature reconstructions. Additionally, a ReLU activation function follows each transposed convolution layer ensuring non-linearity. In short, each input would give the learnable filter a weight, increasing its spatial dimension.

3) Anomaly Detection: To perform our anomaly detection task, we rely on the reconstructed frames generated by the autoencoder and compute the reconstruction error. Frames with high reconstruction errors would be considered as an *"anomaly"* while a low error would be considered *"normal"*. Given an Input pixel, I at location (x, y) at frame t and reconstructed pixel $\hat{I}$ hat at the same location and time, the reconstructed error, $e(x, y, t)$, is given by the L2 norm of I and $\hat{I}$:

$$e(x, y, t) = \left\| I(x, y, t) - \hat{I}(x, y, t \right\|_2 \tag{3}$$

This error can be aggregated over all pixels in a frame t to obtain the frame-level reconstruction error, $e(t) = \sum_{(x,y)} e(x, y, t)$. Additionally, to quantify how closely a frame aligns with normal driving patterns we compute a regularity scoring $s(t)$. This measures the similarity between the input and the reconstructed frames. This is done by summing up all the pixel-wise errors. We define $s(t)$ as the regularity scoring of the frame t.

$$s(t) = 1 - \frac{e(t) - min_t e(t)}{max_t e(t)} \tag{4}$$

A high regularity scoring indicates a frame is following a normal driving patterns while low score indicates irregular or anomalous behaviors. This scoring ranges from 0 to 1.

4) Training Procedure: The convolutional autoencoder was trained end to end to reconstruct the normal frames. The model was trained using a temporal stack of 10 consecutive frames for 30 epochs with a batch size of 128 samples. Optimization was performed using the Adam optimizer with a learning rate of 2.4×10^{-5} and a weight decay of 3×10^{-5}. The reconstruction objective function was defined using a mean squared loss. Early stopping was applied on the validation loss with a patience of 7 epochs. Below is a figure of validation/training curve.

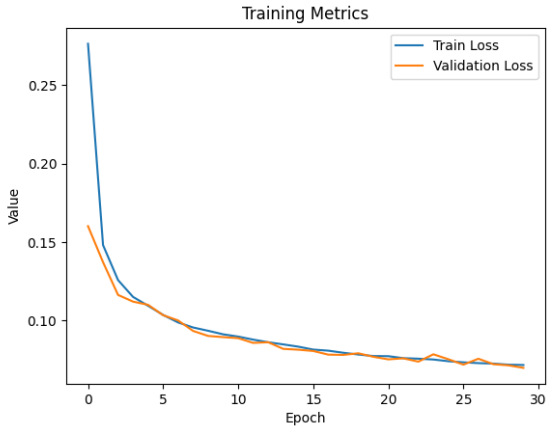

Fig. 4: ConvAE train validation curve

C. *VideoMAE Approach*

1) Video Vision Transformers: Video Vision Transformers (ViViT) take the concepts of Vision Transformers and apply them to video data by viewing a video as a sequence of individual image frames. Essentially, a video is divided into spatial and temporal patches; these are then flattened and converted into tokens. Then these tokens are passed into the transformer network which is designed to capture spatio-temporal dependencies through self-attention mechanisms. This allows it to understand relationships not just within each frame spatially, but also across the frames (time) in the video. This approach enables a more comprehensive representation of the video content.

Given a Video clip of size T and it's respective sequence V, $V \in R^{T \times H \times W \times C}$, we first obtain a sequence of tokens $z \in R^{n_t \times n_h \times n_w \times d}$ through patch embeddings. Then, learnable positional embeddings encode spatial and temporal positions. This is reshaped into a flat sequence and fed into the transformer model that uses self-attention to capture global spatial-temporal features. Below is a figure of sampling strategy taken from [9].

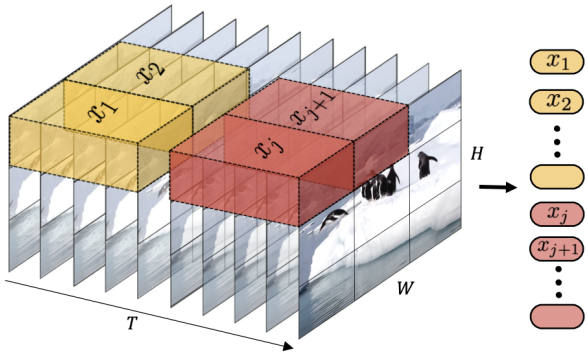

Fig. 5: Video Vision Transformer Sampling Strategy

2) Architecture Overview: VideoMAE backbone: This supervised approach is built on the VideoMAE (Video Masked Autoencoder) architecture, explicitly using the pretrained "MCG-NJU/videomae-base-finetuned-kinetics" model as the backbone. VideoMAE is pretrained using a Masked Video Modeling (MVM) approach, where a significant proportion of spatio-temporal patches are randomly masked out during pretraining. The strategy in MVM is to learn to reconstruct these masked regions. This pretraining approach is relevant to TAD because it can develop robust spatial-temporal representations by learning these movement patterns. This pretrained model can be further pretrained on domain-adaptive data to improve performance, but this is not implemented in the current pipeline due to storage constraints.

3) Transfer Learning: VideoMAE processes video clips by dividing them into spatio-temporal patches, which are embedded into tokens and processed through a transformer encoder. To adapt this pretrained model for binary anomaly detection, we selectively freeze 9 of 12 transformer encoder layers and fine-tune the remaining 3; this preserves the spatiotemporal representations learned during MVM pretraining.

Classification Head: A custom network processes the [CLS] token from VideoMAE's last hidden state: Layer Normalization $\rightarrow$ Dropout $\rightarrow$ Linear $\rightarrow$ GELU $\rightarrow$ Binary logits. The end-to-end pipeline is trained with binary cross-entropy loss.

4) Training Procedure: We train the model for up to 6 epochs using Adam's optimizer with a learning rate of 6.2×10^{-4} and weight decay of 1×10^{-2} for L2 regularization, with a batch of 64 samples. Additionally, a learning rate scheduler is used to monitor the validation loss and reduce the learning rate by a factor of 0.5. The objective function was defined using a binary cross entropy loss. To prevent overfitting, we use early stopping with a patience of 2 epochs. The training loop also tracks F1-score and AUC-ROC metrics. The predictions are generated with a threshold of 0.5 on the sigmoid outputs. The final model checkpoint for the lowest validation loss is saved for inference.

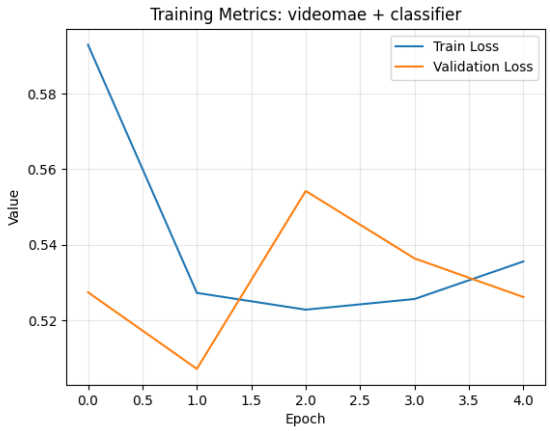

Fig. 6: VideoMAE+classifier train validation curve

IV. Evaluation and Visualization

A. *Evaluation Metrics*

Since we are performing an unsupervised approach for our baseline, we evaluate both of our models using the AUC-ROC metric. It offers a strong way to assess performance in unsupervised TAD tasks. Additionally, the AUC-ROC metric is threshold-independent, making it perfect for a reconstruction error-based model.

$$\text{TPR} = \frac{TP}{TP + FN}, \qquad \text{FPR} = \frac{FP}{FP + TN}$$

$$\text{AUC} = \int_0^1 \text{TPR}(\text{FPR})\, d(\text{FPR}) \tag{5}$$

For our supervised VideoMAE-based classifier, we report F1 scores, precision, recall, and accuracy scores in addition to AUC-ROC. This allows us to evaluate model performance as well as compare it with our baseline.

B. *Quantitative Results*

We evaluate both models on the test set and present comprehensive performance metrics. Table I summarizes the overall AUC-ROC performance comparison between our reconstruction-based ConvAE and supervised VideoMAE classifier.

TABLE I: Performance Comparison on Test Set

Model	AUC-ROC
ConvAE (Unsupervised)	0.556
VideoMAE + Classifier	**0.828**

The supervised VideoMAE classifier achieves an AUC-ROC of 0.828, outperforming the unsupervised ConvAE baseline by 49%. To further evaluate the VideoMAE + classifier model we generate a confusion matrix. Table II

The confusion matrix shows that the model correctly identifies 85.7% of normal events (TN) and 67.2% of anomalous events (TP). The false positive rate of 14.3% indicates that the model occasionally misclassifies normal events as anomalies, which could lead to unnecessary alerts or interventions. The

TABLE II: Confusion Matrix - VideoMAE Classifier

	Predicted	
	Normal	Anomaly
Normal	TN: 1262	FP: 210
Anomaly	FN: 435	TP: 892

false negative rate of 32.8% suggests that approximately one-third of actual anomalies are missed by the classifier, highlighting a potential area for improvement in anomaly detection sensitivity.

TABLE III: Per-Class Performance Metrics - VideoMAE Classifier

Class	Precision	Recall	F1-Score	Support
Normal	0.7437	0.8573	0.7965	1472
Anomaly	0.8094	0.6722	0.7345	1327
Weighted Avg	**0.775**	**0.769**	**0.767**	**2799**

The VideoMAE classifier achieves better precision for anomaly detection (80.9%) but higher recall for normal events (85.7%), resulting in a balanced weighted average F1-score of 76.7%. These results can be further improved by performing DAPT (Domain adaptive pretraining) pretraining on the backbone.

C. *Qualitative Results and Analysis*

In this subsection, we will present some important plots for our traffic anomaly detection task. In these plots, the sequence index refers to the 16 frame clip index in the data frame. We will start of showing an overview of anomaly detection using the VideoMAE classifier. We plotted 200 sequences. For the following plots we use green as correct prediction and red as wrong prediction

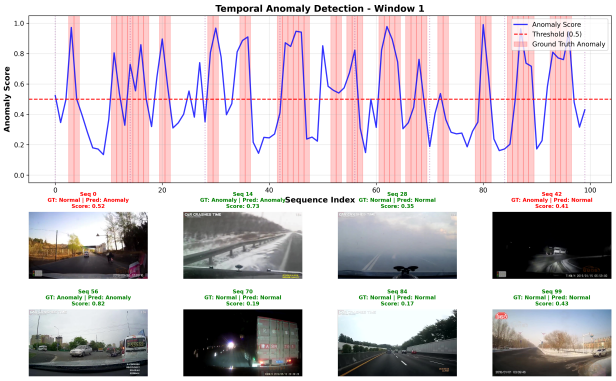

Fig. 7: Overview of prediction

- *Sequence 14:* Vehicle crashing into a barrier flagged as anomaly with high confidence
- *Sequence 42:* Night setting predicted normal when ground truth was anomaly highlighting class imbalance in the night/day settings

Here, we plot the true positive or anomalous frames that were flagged. In frames of multiple normal segments it successfully caught unusual movement patterns on left corner (car crashing into the sand) with a high 0.972 confidence.

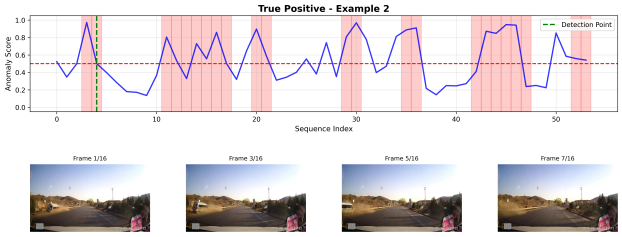

Fig. 8: True Positive GT: Anomaly Pred: Anomaly

Looking at False positive, however, we notice that model is confused in these areas and is unable to properly identify normal or anomalous segments even though the video looks normal. This was predicted with a confidence score of 0.56

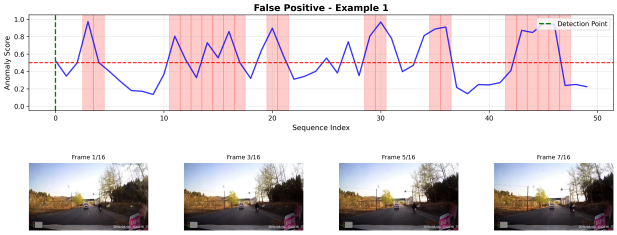

Fig. 9: False Positive - GT: Normal; Pred: Anomaly

Lastly by visualizing our Convulational Autoencoder predictions with a heatmap which tells us the regions of high error. For example, the dark areas in the beginning tell us its a normal segment while the red hot areas are clearly anomaly as the video is a head on collision. However due to a low AUC score, the model does poorly on otherwise obvious crash moving videos and guesses most of the time. It also does poorly when frames consists of both anomaly and normal images.

Fig. 10: ConvAE reconstruction error

V. Conclusion and Future Work

Anomaly detection in ego centric traffic videos are a challenging in the sense of moving videos. This study investigated traffic anomaly detection by attempting to learn temporal and spatial feature using two complementary approaches: A reconstruction based Convulational Autoencoder learning on normal frames and supervised approach on VideoMAE that was pretrained using MVM pretraining strategy. Comparing the two method VideoMAE approach achieved a strong performance with an AUC-ROC score of 0.828 significantly outperforming ConvAE (0.556). The VideoMAE model demonstrated aa

high precision of (0.8094) which is crucial for minimizing false alarms in the real world deployment. Temporal analysis however reveled that the model excels at abrupt changes to the environment but struggles with gradual shifts in the environment that would otherwise be categorized as anomaly.

This model is limited in many ways. First the class imbalance in the categorical areas as well as night and day can cause a high bias. Additionally due to the not being able to fine tune the pretrained model on domain specific dataset, accuracy has gone down. Model might also misclassify uncommon but safe maneuvers such as parallel parking or fast right turns on a light.

Future Direction would build on this solution and implement better temporal modeling to handle rapid and gradual anomalies. Additionally, by performing heavy data augmentation and class balancing techniques would bias across all areas. Also, this can extending this model into multi-class accident detection system instead of just anomalies. Lastly, results can be further improved by pretraining on domain specific data.

REFERENCES

[1] Y. Yao et al., "DoTA: Unsupervised Detection of Traffic Anomaly in Driving Videos," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 45, no. 1, pp. 444-459, 1 Jan. 2023, doi: 10.1109/TPAMI.2022.3150763

[2] Hasan, M., Choi, J., Neumann, J., Roy-Chowdhury, A.K., Davis, L.S.: Learning temporal regularity in video sequences. In: CVPR (2016)

[3] Medel, J.R., Savakis, A.: Anomaly detection in video using predictive convolutional long short-term memory networks. arXiv:1612.00390 (2016)

[4] Y. S. Chong and Y. H. Tay, "Abnormal Event Detection in Videos Using Spatiotemporal Autoencoder," in Advances in Neural Networks—ISNN (Springer International Publishing, 2017): 189–196

[5] R. Muhammad, E. Amparore, E. Ferrari, and D. Verda, "Can I Trust My Anomaly Detection System? A Case Study Based on Explainable AI," in Proc. World Conf. Explainable Artif. Intell., Valletta, Malta, 2024, pp. 243-254, doi: 10.1007/978-3-031-63803-9_13

[6] Y. Yao, M. Xu, Y. Wang, D. J. Crandall and E. M. Atkins, "Unsupervised Traffic Accident Detection in First-Person Videos," 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 2019, pp. 273-280, doi: 10.1109/IROS40897.2019.8967556.

[7] Y. Yao et al., "DoTA: Unsupervised detection of traffic anomaly in driving videos," IEEE Trans. Pattern Anal. Mach. Intell., vol. 45, no. 1, pp. 444–459, Jan. 2023.

[8] E. Orlova et al., "Simplifying Traffic Anomaly Detection with Video Foundation Models," in Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW), (2025).

[9] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Luci ˇ c, and Cordelia Schmid. Vivit: A video ´ vision transformer. In Proceedings of the IEEE/CVF international conference on computer vision, pages 6836–6846, 2021.