# Towards Automatic Proctoring of Online Exams Using Video Anomaly Detection*

Jef Plochaet[0000−0002−2160−8416] and Toon Goedemé[0000−0002−7477−8961]

EAVISE-PSI-ESAT, KU Leuven, Sint-Katelijne-Waver, Belgium
{jef.plochaet,toon.goedeme}@kuleuven.be

**Abstract.** In this paper we present exploratory research for webcam-based automatic proctoring of online exams. We investigate the use of a video anomaly detection network to detect when a person is cheating during an online exam. We train and test this model on a dataset comprised of actors that imitate students filling out an online exam. The anomaly detection model obtains a 0.72 true positive rate and a 0.31 false positive rate. We also add a voice activity detection model and a person detection model to improve our anomaly detection rate, resulting in a true positive rate of 0.88 and a false positive rate of 0.44. After optimizing the thresholds of the entire pipeline, we decrease the false positive rate to 0.19 while still reaching a true positive rate of 0.78. Our experiments indicate that the proposed framework could be a possible way to detect students cheating during online exams. However, we conclude that increasing the training dataset will be necessary to achieve commercializable accuracy.

**Keywords:** Automatic proctoring · Video anomaly detection · Voice activity detection · Person detection.

## 1 Introduction

Due to the Covid-19 pandemic, a lot of activities were to be replaced by an online variant, including teaching. In-person classes were replaced by video recordings and live online lessons. Likewise, the evaluation of students found an online replacement. However, ensuring academic integrity during online exams remains a difficult problem to solve. The students are in an uncontrolled environment and can only be observed using their webcam and microphone. One solution is to have a live proctor that inspects the video footage of the student. However, this becomes unfeasible when multiple students are taking the exam at the same time. It is possible to use multiple proctors during the exam to watch different students. Again, this is not an ideal solution as it is an ineffective use of their time.

In order to render online proctoring scalable, in this paper we investigate the use of video anomaly detection to detect a student cheating. Anomaly detection

* This work is financially supported by VLAIO and Sensotec nv.

is used in cases where it is impossible to collect enough data of the anomalous events. Students can cheat in an infinite number of ways (using their phone, talking to someone, using a cheatsheet, ...). This makes it impossible to collect sufficient data on all ways of cheating, thus supporting our choice for video anomaly detection.

We also propose using a voice activity detection model and a person detection model to assist the anomaly detection model. Using the extra models enables us to detect more subtle anomalies, like speech, that are almost undetectable using a webcam.

## 2 Related Work

### 2.1 Online Proctoring

Since the switch to online teaching and online examinations, there has been a lot of attempts to try and ensure the academic integrity of online exams. Bilen and Matros [3] talk about evidence of cheating during online exams. They suggest using a camera set-up similar to online chess tournaments to prevent cheating. The camera is pointed at the screen of the student, to monitor their actions. Though, they add that this solution is unfeasible on a large scale.

Another option is the use of a proctoring tool during the online exam. Hussein et al. [7] compare and evaluate some of the different available proctoring tools. There are tools like ProctorU [10], that offer live and recorded proctoring. However, solutions like this still need people to review the footage of the exam. Proctorio [12] offers an automated proctoring system but seems to only use face detection and gaze detection. AIProctor [2] and Talview [14] both mention the use of AI-powered proctoring. Although it seems that they only use artificial intelligence for face detection/identification and speech detection/identification.

### 2.2 Video Anomaly Detection

There are many different approaches to video anomaly detection. Nayak et al. [11] discuss various types of models including reconstruction models, prediction models and generative models.

Reconstruction models [6, 16] learn to reconstruct the normal situation during training. This will result in a low reconstruction error for frames that do not contain an anomaly. Frames that contain anomalies will be poorly reconstructed because the model has never seen the anomaly during training.

Prediction models [4] will try to predict the current frame given a set of past frames. If the frames are normal, the model will be able to predict the current frame well. On the other hand, if the frame contains an anomaly, the prediction will be worse.

Generative models model the distribution of the normal frames. Dong et al. [5] employ a generative adversarial network (GAN) with two discriminators. The generator is used to generate future frames. A bad generation of a frame would suggest that the frame contains an anomaly.

Our situation differs a lot from the popular datasets used to evaluate video anomaly detection networks. UCSD Ped1 & Ped2 [9], Subway [1] and Avenue [8] are among the most used datasets for video anomaly detection. These datasets are surveillance datasets, where most of the anomalies are visually easily noticeable (a car on the footpath, running, riding a bike, going in the opposite direction, ...). However, the anomalies found in our dataset are more subtle and harder to detect. This is because students do not want to get caught cheating and try their best to hide their actions.

## 3 Approach

Our goal is to detect students cheating during online exams. It is possible to set up a system where we use additional cameras or microphones but this is expensive and not feasible on a large scale. We operate under the assumption that only limited hardware is available, i.e. we only use a microphone and a webcam, which are almost always incoporated into a laptop. We propose to add a clip-on fisheye lens (advertised field-of-view = 238°) to the webcam, to enlarge the field-of-view, as shown in Fig. 1. The lens provides a bigger field-of-view than the standard webcam. This ensures that we can see more of the desk and surroundings. A better view of the surroundings will make it easier to detect anomalies in the videos. In Fig. 1, we can see that adding the fisheye lens enables us to see a part of the desk and laptop. It also enables us to see the smartphone lying on the desk.

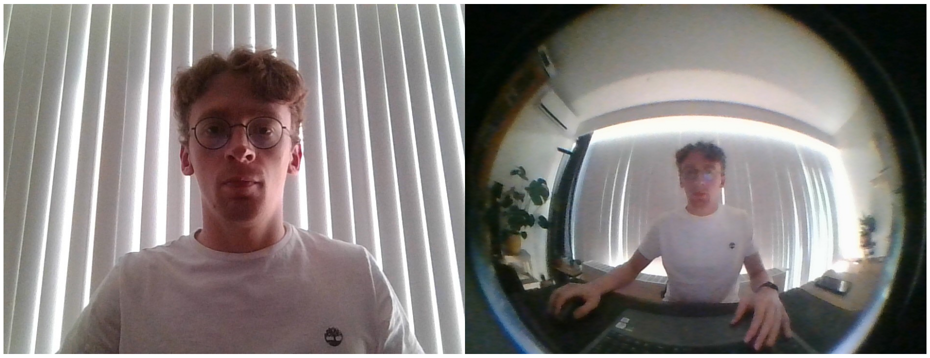


Fig. 1: Comparison of the field-of-view of a webcam without a fisheye lens (left) and a webcam with a fisheye lens (right)

### 3.1 Data Collection

To collect the training data we recorded people that were acting like students during online exams. During the recording session the actors had to behave like "good" and "bad" students. The system will be trained on footage from "good" students, that make their exam without cheating. On the other hand, a "bad" student is a student that does cheat during the exam. The student can cheat in an infinite number of ways during the online exam and it is impossible to collect data from every way. To be able to test our system, we collected eight different anomalies that a student could possibly use to cheat on an online exam. The eight different anomalies are the following: 1. student turning their back to the camera; 2. using a book or notes; 3. using a cheatsheet; 4. student leaving the frame; 5. more than one person visible in the frame; 6. student using their phone; 7. student looking for something under their desk/acting suspicious; 8. student talking to someone who is not visible for the webcam.

During the dataset recording the actors had to follow a video that notified them when to act like a "good" or "bad" student. Following this video made sure that the right data was captured. The video also informed the actor which anomaly to perform when playing the "bad" student. This ensured that all eight different anomalies mentioned before were captured.

Furthermore, a fake exam was created. The exam consisted of a few questions for which the actor had to use both their mouse and keyboard. The actors had to make this fake exam during the moments they had to act like "good" students.

In total we recorded 13 different people (9 men and 4 women), each at a different location. The locations ranged from places in the office to workspaces at home. We captured ±70 minutes of video. Table 1 shows the distribution of the data over the training and test set. Fig. 2 is an example of the captured data. Note again that for the training set we only use clips of the actors acting like "good" students.

Table 1: Distribution of the training and test set

| | **Training set** | **Test set** | |
|---|---|---|---|
| | Good student | Good student | Bad student |
| **# clips** | 72 | 32 | 32 |
| **# minutes** | ±36 | ±16 | ±5.33 |
| **Persons** | 2, 3, 4, 6, 7, 8, 9, 11, 12 | 1, 5, 10, 13 | |

### 3.2 Automatic Proctoring Pipeline

Detecting cheating students during an online exam is a difficult problem to solve. This is because first of all the student may receive help from other persons (these individuals may or may not be visible to the webcam). Secondly, the way the student can cheat is only limited by their imagination. We built in extra robustness against the first problem by using a voice activity detection model

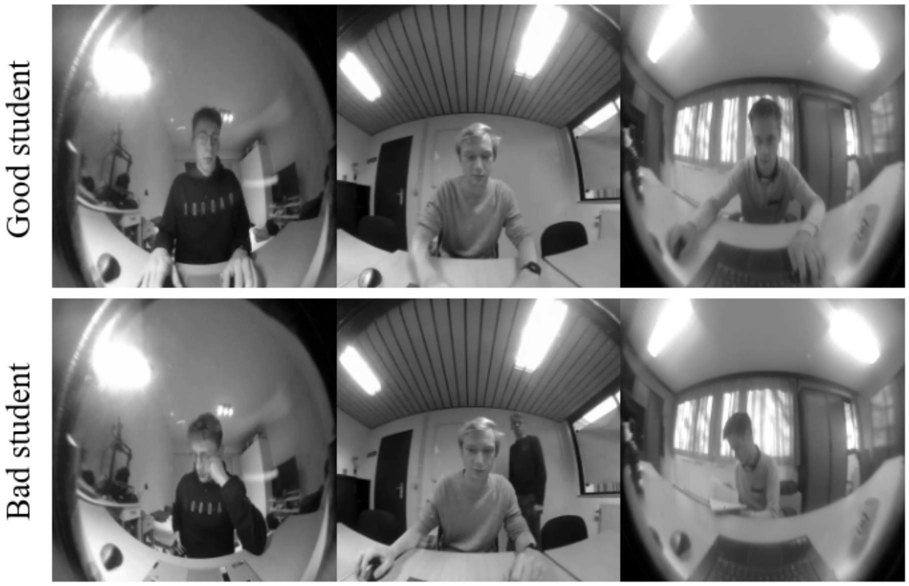


Fig. 2: Some examples of the collected data

and a person detection model. The second problem can be solved only by using anomaly detection.

Fig. 3 shows an overview of the entire pipeline. The first part of the pipeline is the voice activity detection (VAD) model that will be used to detect speech. We operate on the assumption that the student has to make the online exam individually. The goal of using a VAD network is to detect when someone is talking to the student without being in the field-of-view of the webcam. To achieve this we use a network introduced in [17]. The network works by converting the audio of the video into sequences of spectrogram images and then running these image sequences through a convolutional neural network (CNN) and bidirectional long short-term memory (BiLSTM). We chose this network because it reaches state-of-the-art performance on the AVA-Speech dataset (area under the curve = 0.95). Since the goal of this network is to detect speech in audio clips and we do not use it for any other purpose, we can use the network without retraining.

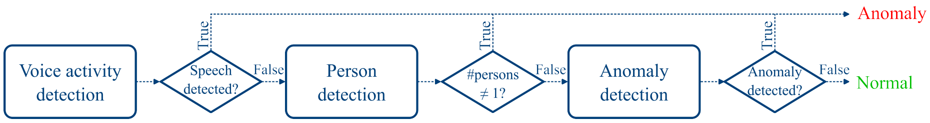


Fig. 3: Overview of the pipeline for the automatic proctoring of online exams

Secondly, we use the YOLOv5 (small) [15] network to detect if there are more or less persons than one in the room. We need to enforce that the student will be making the online exam alone. This means that the student should at all times be the only person in the frame. In addition, the student is not allowed to leave the frame. If not, this would allow the student to leave the room, look for answers and then return without triggering the automatic proctoring system. We use the YOLOv5 (small) network because it provides a great balance between accuracy and speed. As with the VAD, we use the network only for its intended purposes and do not need to retrain the network.

The goal of the last network in the pipeline is to detect every other way of cheating. As mentioned before, there are an unlimited amount of ways for a student to cheat on a online exam. This makes it impossible to find (enough) examples to train a network. That is why we use an anomaly detection network. These networks are designed to detect everything that is out of the ordinary. We use a fully convolutional autoencoder model introduced in [6] (implementation [13]). The network is only trained on video sequences that do not contain any anomalies. The idea is that after training, the network should be able to reconstruct normal video sequences with minimal error. When the network is given a video sequence that contains an anomaly, the network will not be able to reconstruct the video sequence successfully. The network is given 20 frames as input. Most of our collected data has a frame rate of 30 frames per second (FPS). If only 20 frames are given as input to the network, the network actually gets less than one second of the video as input. One second is not long enough to display the full anomaly. To solve this, we only use 1 frame every 8 frames. The skipped frames are then used as separate sequences to avoid the loss of training data.

The full pipeline works as follows: the video first gets presented to the VAD network. If the network detects speech in the video, the video is immediately classified as an anomaly and will skip the person and anomaly detection. If no speech is detected, the video will be passed on to the person detection. Similarly, if no persons or more than one person is detected, the video is classified as an anomaly and will skip the anomaly detection. Lastly, if only 1 person is detected, the video will be passed on to the anomaly detection. Naturally, if the anomaly detection detects an anomaly, the video will be classified as anomalous. If the video passes all three stages of the pipeline without being classified as an anomaly, the video is classified as a normal video. The VAD, person detection and anomaly detection are in this specific order for two reasons. The first reason is that the VAD and person detection are more accurate than the anomaly detection. If the VAD or the person detection classify the video as an anomaly, we can trust them and do not need the anomaly detection to process the video. Moreover, speech is very difficult to detect using a webcam. To detect speech we have to rely more on the VAD than on the anomaly detection. The second reason is that the models are in order of increasing inference time. If a video is classified as an anomaly earlier in the pipeline, it will bypass the remaining models and saves the most time possible.

## 4 Results

We used the dataset described in section 3.1 for our experiments. We first test the VAD, person detection and anomaly detection separately and then test the entire combined pipeline. By doing this, we can see what effect the addition of the VAD and person detection has on the anomaly detection. To evaluate the models, we use a threshold to decide if the video contains an anomaly. Using this threshold we can turn the evaluation into a binary classification problem, i.e. the video contains or does not contain an anomaly. The threshold we use depends on the model and will be explained for each model below.

### 4.1 Voice Activity Detection

Because we do not need to retrain the VAD network, as mentioned in section 3.2, we use all clips in the dataset to test it. As ground truth labels, we used the information whether the clips contained speech or not. We use a threshold to decide whether the video actually contains speech or the VAD detected noise. As threshold we use the total number of seconds of speech detected by the VAD. To empirically find the best threshold value, we test a threshold ranging from 0 seconds to 13 seconds and use a 0.1 second step increase. We calculate the false positive rate (FPR) and the true positive rate (TPR) for all the different thresholds and plot the receiver operating curve (ROC).

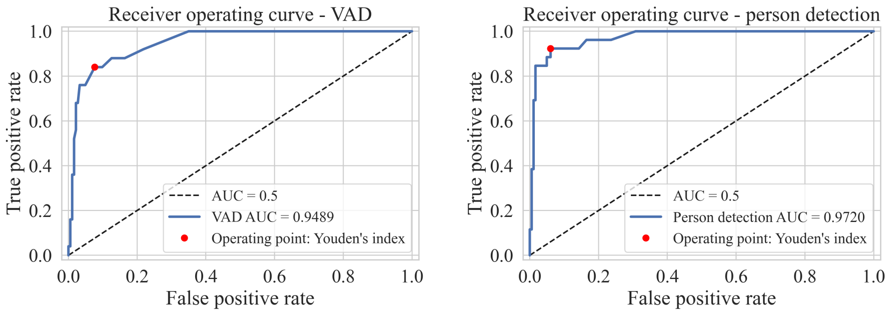
Fig. 4a show the ROC for the VAD. The voice activity model obtains an area under the curve (AUC) $\approx 0.95$. Youden's index is used to find a possible operating point for the VAD model. The Youden's index makes a trade-off between TPR and FPR and is defined as follows:

$$\text{Youden's index} = \text{sensitivity} + \text{specificity} - 1 = \text{TPR} - \text{FPR} \tag{1}$$

We reach a best trade-off for the VAD model in a operating point with TPR $= 0.84$ and FPR $= 0.08$. The threshold used to obtain these results is equal to 1.6 seconds.

### 4.2 Person Detection

As with the VAD, we can use the entire dataset to test the model because we do not need to retrain the model. We re-labeled the dataset for the number of persons visible in the frame. The person detection model shows per frame how many people were detected in that frame. In this case we use the number of frames where an anomaly was detected as threshold. Anomaly means in this case that the person detection model detected no persons or more than one person in the frame. We test a threshold ranging from 1 frame up to 300 frames with a 1 frame increase each step. Fig. 4b shows the resulting ROC curve for the person detection model. The person detection model obtains an AUC $\approx 0.97$. Again, we use Youden's index to find the best operating point for the person detection model. The operating point was placed at a TPR $= 0.92$ and a FPR $= 0.06$. The threshold used to achieve this operating point is equal to 45 frames.



(a) Receiver operating curve for the voice activity detection

(b) Receiver operating curve for the person detection

Fig. 4: Receiver operating curves for the VAD and person detection

### 4.3 Anomaly Detection

As mentioned in section 3.2, the output of the anomaly detection network is a per frame anomaly score (0-1). To decide whether the videos contain an anomaly we use two thresholds. The first threshold, score threshold, decides if a frame is anomalous. A frame is considered anomalous if the anomaly score is larger or equal than the score threshold. The second threshold, frame threshold, decides if the entire video is anomalous. A video contains an anomaly if there are equal or more consecutive anomalous frames than the frame threshold. We test with the score threshold ranging from 0.5 to 1 and with the frame threshold ranging from 2 to 10 frames. Fig. 5 shows multiple ROC curves. Each ROC curve represents a different frame threshold (ft). We use Youden's index to find the best operating point for the anomaly detection model. To produce this operating point we used a score threshold of 0.73 and a frame threshold of 4. The TPR equals 0.72 and the FPR equals 0.31 at the operation point.

### 4.4 Full Pipeline

Finally, we test the entire pipeline. The videos pass through the voice activity detection first, then the person detection and finally the anomaly detection. The thresholds of each component were chosen by optimising the Youden's index, as described above.

First, we evaluate every part of the pipeline separately for the task of cheating detection on the full dataset using previously mentioned thresholds. The results of this test are shown in table 2. The VAD and person detection are only suited to detect specific anomalies (speech and no/multiple persons). This results in a lower true positive rate because they can only detect the anomalies that they are designed for and are not able to detect any other anomalies. The VAD and person detection perform well and are able to detect these specific anomalies.

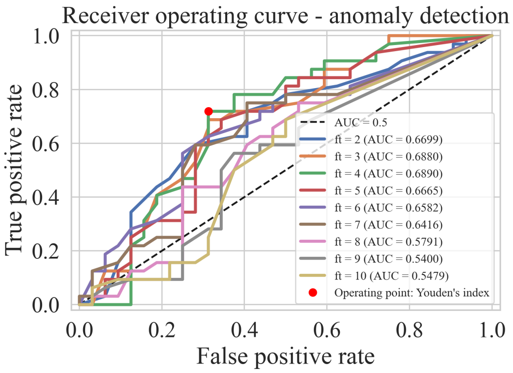


Fig. 5: Receiver operating curve for the anomaly detection (ft = frame threshold)

This results in a low false positive rate. In other words, there is a low probability of a false alarm. The anomaly detection is able to detect more anomalies (TPR = 0.72) but at the expense of a higher false positive rate.

Table 2: The true positive rate, false positive rate and resulting Youden's index scores for all subsystems and combinations of subsystems ($\star$ uses the globally optimized thresholds)

| **VAD** | **Person detection** | **Anomaly detection** | **TPR** (↑) | **FPR** (↓) | **Youden** (↑) |
|---|---|---|---|---|---|
| ✓ | | | 0.2500 | 0.0938 | 0.1562 |
| | ✓ | | 0.3438 | **0.0313** | 0.3125 |
| | | ✓ | 0.7188 | 0.3125 | 0.4063 |
| ✓ | ✓ | | 0.5000 | 0.1250 | 0.3750 |
| | ✓ | ✓ | 0.7813 | 0.3438 | 0.4375 |
| ✓ | | ✓ | 0.8125 | 0.4063 | 0.4062 |
| ✓ | ✓ | ✓ | **0.8750** | 0.4375 | 0.4375 |
| ✓($\star$) | ✓($\star$) | ✓($\star$) | 0.7813 | 0.1875 | **0.5938** |

Secondly, we test different combinations of models in the pipeline. These results are also shown in table 2. We can see that the addition of the VAD or the person detection to the anomaly detection leads to an increase in true positive rate. This proves that using the VAD and person detection to support the anomaly detection model is truly useful for detecting more anomalies. This is because some anomalies, namely speech, is very hard to detect only using a webcam. Adding the VAD to the pipeline is our solution to this problem. Unfortunately, the false positive rate also increases, indicating that the probability of a false alarm is higher due to the addition of the VAD or person detection. Adding the person detection network also increases the Youden's index, thus increasing the overall performance of the anomaly detection. When we add the VAD model to the pipeline, the Youden's index does not change. This means that the overall performance of the anomaly detection does not change. However, there is an increase in true positive rate, meaning that more anomalies are detected by adding the VAD model.

Next, we test the full pipeline (as shown in Fig. 3). We manage to obtain a TPR equal to 0.875 and a FPR equal to 0.4375. Adding both the VAD and person detection results in an increase in TPR (0.1562 increase versus the anomaly detection model alone). Also, the addition of both the VAD and the person detection results in a higher TPR than if we add only one or the other. Unfortunately, adding both models to the pipeline causes a further increase in FPR. However the increase in TPR (0.1562 increase) is greater than the increase in FPR (0.125 increase). This indicates that adding the VAD and person detection has more of a positive influence than a negative one. This is also reflected in the Youden's index, where the full pipeline achieves a better result (0.4375) than the anomaly detection model alone.

Lastly, we finetune the selected thresholds by performing a grid search over different combinations of thresholds for the entire pipeline. Before, we optimized the thresholds for each subsystem separately. Now, we optimize the thresholds for every subsystem when they are part of the full pipeline. The globally optimal thresholds are: 1.7 seconds for the VAD, 4 frames for the person detection model, 0.95 for the score threshold of the anomaly detection model and 2 for the frame threshold of the anomaly detection model. This results in our best trade-off point with a Youden's index equal to 0.5938. Changing the thresholds leads to a decrease of 0.0937 in TPR, indicating that less anomalies are detected. However, the change in thresholds also leads to a major decrease in FPR (-0.25). This reduction in FPR means that the full pipeline with optimized thresholds is less prone to false positives than the original pipeline.

### 4.5 Discussion

Although our explorative experiments demonstrate the potential for video anomaly detection for this activity by reaching a high true positive rate, the total FPR our system achieves is too high for any real world use. It would generate too much false alarms during a real online exam. Before deploying this concept in the real world we need to overcome this issue.

By analyzing the results of the pipeline based on separately optimized components, we noticed that 6 out of 14 false positives happened to the same person (while other people only had 2-3 false positives). The cause of this is probably the difficult background. The background is widely different from most backgrounds in the training set. This indicates that the anomaly detection model does not yet generalize well enough. To solve this problem we have to collect more data and more varied data. The latter of the two is very important if we want to implement the pipeline in real life situations. In real life we will encounter many different and very different people and backgrounds and if our dataset is not varied and large enough the model will fail in difficult situations.

After tweaking the thresholds, the false positive rate decreases to a better, but still unneglectable number. The true positive rate only drops a bit, resulting in our best Youden's index. However, optimizing the thresholds on the test set leads to a bias in our results, there is a serious risk of overfitting. Unfortunately, we did not have enough data to optimize the thresholds on a separate validation set and verify the results on a test set. Additional data needs to be collected to further verify the thresholds and results.

Until we train and test the models on a larger dataset, our framework remains an unproven concept. One option that could be used at the moment, however, is the combination of the VAD and the person detection. This combination is more robust and does not result in many false positives. The downside of only using the VAD and the person detection is that they do not detect all kinds of anomalies. The combination of VAD and person detection could be used as a first layer of defense against cheating on online exams.

## 5 Conclusion

In this paper, we investigated the use of video anomaly detection for automatic proctoring for online exams, and extended it with voice activity detection and person detection.

First, we collected a dataset using actors that imitated the behaviour of students filling out an online exam. The dataset was collected using a limited amount of hardware, i.e. a webcam, a microphone and a fisheye lens. This makes our solution inexpensive, easy to implement in real life and scalable.

We proposed a framework to detect cheating during online exams and tested it on our real-life dataset. Our pipeline consists out of three models: a voice activity detection model, a person detection model and an anomaly detection model. We showed that anomaly detection has potential for an automatic proctoring system, although further research is needed to confirm this. The anomaly detection model alone obtained a true positive rate of 0.72, a false positive rate of 0.31 and a Youden's index of 0.41. We then showed that adding the voice activity detection model and the person detection model increased the true positive rate to 0.875 and the Youden's index to 0.44, indicating that adding these models helps the detection of anomalies. Unfortunately, The high false positive rate of the anomaly detection model alone (0.31) and the full pipeline (0.44) prevents this concept from being used in real life. Here, the anomaly detection model lacks the ability to generalize well enough to be used in real-life situations.

After globally optimizing the thresholds for the entire pipeline we achieve a true positive rate of 0.78 and a false positive rate of 0.19, resulting in our highest Youden's index score of 0.59. These improved results indicate a better usable concept for the automatic proctoring of online exams, but indicate still that the system will generate a substantial amount of false alarms. However, the thresholds were optimized on the test set, leading to a bias in our results. To further verify these results on a separate test set, more data is needed.

The remaining challenges are that the anomaly detection model lacks the ability to generalize well and we do not have enough data to verify our results of the optimized pipeline. These problems can be alleviated by increasing the size of the dataset, whilst also making sure that there is enough variety in the data. A bigger training set would allow the model to generalize better, suppressing the amount of false positives. More data also means that we would be able to verify the results of the pipeline with the optimized thresholds. Another possibility for future work is to explore different anomaly detection models. Until further research is conducted, we can use the voice activity detection and person detection as an initial layer of defence against cheating during online exams.

## Acknowledgments

This work is supported by VLAIO and Sensotec nv.

## References

1. Adam, A., Rivlin, E., Shimshoni, I., Reinitz, D.: Robust real-time unusual event detection using multiple fixed-location monitors. IEEE transactions on pattern analysis and machine intelligence **30**(3), 555–560 (2008)
2. AI Proctor: Online Exam Proctor Monitoring Software | AI Proctor, https://ai-proctor.com/
3. Bilen, E., Matros, A.: Online cheating amid covid-19. Journal of Economic Behavior & Organization **182**, 196–211 (2021)
4. Deepak, K., Chandrakala, S., Mohan, C.K.: Residual spatiotemporal autoencoder for unsupervised video anomaly detection. Signal, Image and Video Processing **15**(1), 215–222 (2021)
5. Dong, F., Zhang, Y., Nie, X.: Dual discriminator generative adversarial network for video anomaly detection. IEEE Access **8**, 88170–88176 (2020)
6. Hasan, M., Choi, J., Neumann, J., Roy-Chowdhury, A.K., Davis, L.S.: Learning temporal regularity in video sequences. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 733–742 (2016)
7. Hussein, M.J., Yusuf, J., Deb, A.S., Fong, L., Naidu, S.: An evaluation of online proctoring tools. Open Praxis **12**(4), 509–525 (2020)
8. Lu, C., Shi, J., Jia, J.: Abnormal event detection at 150 fps in matlab. In: Proceedings of the IEEE international conference on computer vision. pp. 2720–2727 (2013)
9. Mahadevan, V., Li, W., Bhalodia, V., Vasconcelos, N.: Anomaly detection in crowded scenes. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp. 1975–1981 (Jun 2010). https://doi.org/10.1109/CVPR.2010.5539872, iSSN: 1063-6919
10. Meazure Learning: The ProctorU Proctoring Platform - Advanced Exam Technology Backed by Human Validation, https://www.proctoru.com/
11. Nayak, R., Pati, U.C., Das, S.K.: A comprehensive review on deep learning-based methods for video anomaly detection. Image and Vision Computing **106**, 104078 (2021)

12. Proctorio Inc.: Proctorio, https://proctorio.com/products/online-proctoring/
13. Ramachandra, B.: Frame-level anomaly detection in videos, https://github.com/tnybny/Frame-level-anomalies-in-videos
14. Talview Inc.: Talview Automated Proctoring, https://www.talview.com/solutions/proctoring/automated-proctoring
15. Ultralytics: ultralytics/yolov5, https://github.com/ultralytics/yolov5
16. Wang, L., Zhou, F., Li, Z., Zuo, W., Tan, H.: Abnormal event detection in videos using hybrid spatio-temporal autoencoder. In: 2018 25th IEEE International Conference on Image Processing (ICIP). pp. 2276–2280. IEEE (2018)
17. Wilkinson, N., Niesler, T.: A hybrid CNN-BiLSTM voice activity detector. In: Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP). Toronto, Canada (2021)