Analyzing the Robustness of Deep Learning Based Video Smoke Detection using Synthetic Smoke Videos

Philip Dietrich
Bosch Building Technologies, Grasbrunn, Germany

Abstract

While Deep Learning algorithms were able to outperform classical algorithms in many computer vision applications, they are associated with a lack of interpretability. For safety critical applications, like Video Smoke Detection, an algorithm should operate correctly regardless of potential environmental influences. Analysing the effect of external influences in reality is not feasible, as experiments are not reproducible under different settings. Therefore, we introduce the usage of synthetic data for testing safety-critical deep learning algorithms for video smoke detection. When generating synthetic smoke videos, different parameters can be changed individually. We introduce how a trained network may be analysed on synthetic smoke videos, where isolated parameters like smoke colour or smoke density are altered systematically. We further analyse, the robustness of the algorithm against the distance of smoke to the camera.

Keywords: Video Smoke Detection, Deep Learning, Smoke Simulation, Robust Deep Learning

Introduction

The detection of smoke via video cameras may provide multiple benefits, especially in large halls with high ceilings. Smoke is visible long before it reaches a ceiling-mounted detector. Therefore, detecting smoke using video cameras may lower detection times notably.

As shown in prior work (e.g. [1]), video smoke detection (VSD) is a highly non-trivial problem. In the last years, researchers have shown that deep learning (DL) can outperform classical algorithms in terms of accuracy (e.g. [2]). The downside of complex DL networks is their lack of interpretability. Usually state-of-the-art model architectures are very complex. Consequently, the exact process of a network making a certain
decision is not understood. Nevertheless, researchers are determined to make sure a network is robust against environmental influences [3], especially in safety-critical applications like fire detection. Testing algorithms on video data of all possible scenarios in reality is not feasible, as it is nearly impossible to alter isolated parameters in real settings. This issue can be overcome by synthetic data. Xu et al. have shown that smoke detection networks benefit from using synthetic smoke data in training [4].

Here, we extend the use of synthetic data to the testing of DL algorithms. We argue similar to Xu et al. that the behavior on synthetic smoke videos is reasonably similar to that on real smoke data. With that assumption, different parameters and their influence on the robustness of smoke detection networks is analyzed.

**Database**

The videos, which have been used here, can be categorized into real and synthetic videos. Real videos were obtained from an internal Bosch database. This dataset consists of videos containing smoke as well as negatives. Videos are recorded in typical surveillance scenarios and contain indoor as well as outdoor scenes.

Synthetic videos are generated by blender [7] and its python API. As backgrounds, negative sequences were used. The structure of our data generation pipeline is visualized in Fig. 1. We start by defining a number of realistic backgrounds as well as render and simulation parameters. Next, simulation and render settings have been selected from the predefined files. Via the python API, smoke is simulated and rendered as images with an alpha channel. These images are then added to a randomly selected background video. Videos do not contain flames or smoke sources.

![Data Generation workflow](image)

*Fig. 1. Data Generation workflow for synthetic smoke videos using the blender framework [7].*

The resulting videos have a duration of 45 seconds in a framerate of 3fps with a resolution of 360x640. Creating one video on an Nvidia GeForce
RTX 2080 Ti graphic card takes approximately one hour, depending on the size of the simulated smoke.

For the following evaluation, the real and synthetic videos have been divided into three sets – one set for training, one for quantitative testing on real scenarios (smoke and negatives) and one for qualitative evaluation and analyzing robustness against different environmental influences. The last set consists of only synthetic smoke sequences, where different parameters have been altered systematically. The exact numbers of videos in each set can be found in Table 1.

Table 1. Distribution of videos in the different datasets.

<table>
<thead>
<tr>
<th>Video Class</th>
<th>Training Set</th>
<th>Test Set</th>
<th>Structured Test Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real Smoke</td>
<td>704</td>
<td>312</td>
<td>-</td>
</tr>
<tr>
<td>Synthetic Smoke</td>
<td>508</td>
<td>-</td>
<td>562</td>
</tr>
<tr>
<td>Negatives</td>
<td>879</td>
<td>362</td>
<td>-</td>
</tr>
</tbody>
</table>

All videos have been rescaled to a spatial dimension of 256x256 and a framerate of 3 fps. For DL purposes, pixel values were normalized in the range [0, 1]. As inputs for the networks image differences between consecutive frames have been used. This operation works as a very basic background subtraction algorithm, which has shown to improve detection accuracy in prior work.

**Convolutional Neural Networks for Video Smoke Detection**

Visual features characterizing smoke are highly diverse. One way to extract a large number of features from visual data is to use Convolutional Neural Networks (CNNs). Different CNN architectures were applied to the problem of Video Smoke Detection [1, 3, 5]. A comprehensive analysis of different approaches can be found in [5]. The actual choice of a network is of minor importance here. Therefore, we decided to use the Inception-V1 architecture [8] as our baseline.

As a baseline, we use a network trained on only real data. As our evaluation metric, we choose the receiver operating characteristic (ROC) curve and its respective area under curve (AUC) evaluated on the unseen test set. The baseline network achieved an AUC of 0.942 on the test set. The ROC curve is shown in Figure 4 as the baseline curve.

**Video Smoke Detection and Synthetic Data**

Even though synthetic smoke generated by blender looks reasonably realistic, it has to be noted, that some features may differ from real smoke. Some examples may be the absence of a smoke source, the absence of occlusions or an unrealistic spreading within the room.
Generally, the degree of realism could be increased further, leading to a significantly higher effort per sample. Here, we analyze, whether network predictions made on synthetic data can be used to draw conclusions for the behavior in real scenarios.

To analyze how artefacts in synthetic data may affect the behavior of CNNs qualitatively; we investigated their “feature space”. A CNN designed for classification can be divided into two subparts. The first part, the feature extractor, consists of convolutional and pooling layers. These layers extract visual features from the input image. A typical output of a feature extractor has the shape $1 \times n_{\text{features}}$, where $n_{\text{features}}$ is the number of features extracted, in this case 1024. These features are then processed by the classification part of a network. It typically consists of multiple fully connected layers, transforming the input layer to a binary classification, which could later be translated to an alarm. This split into two subnetworks is also visualized in figure 2.

![Fig. 2. The typical structure of a CNN for image classification.](image)

Here, the feature space will be investigated in detail. Similar to [4], we analyze the features extracted from input images via t-distributed stochastic neighbor embedding (t-SNE) [9]. The t-SNE algorithm transforms high dimensional feature vectors into a lower dimensional space using the similarity of points in feature space, i.e. points close in high dimensional space will be close in low dimensional space. Here, we are using t-SNE to show how synthetic data influences the feature space.

In general, a network is trained, to extract features, which are characteristic for the desired class (e.g. smoke). Potentially optical features of real and synthetic smoke differ in some cases. This difference in features is called domain difference. To be able to draw conclusions from one domain to the other the effect of this difference should be minimized. In the literature, a wide variety of methods to cope with this problem can be found. Here, the method of Ganin et al [6] is used, where an additional network is trained to distinguish real from synthetic data. Via a combined loss, the feature extractor is trained not to extract features, by which real data may be distinguished from synthetic data and being able to detect smoke at the same time.
In Figure 3, it is visualized how samples from the different classes are placed in the feature space. Here, the distribution of synthetic smoke frames is very similar to the distribution of real smoke frames. Therefore, it is plausible, that networks extract the same features from synthetic smoke as it does from real smoke. Consequently, it is reasonable to expect a comparable behavior of networks on real and synthetic data.

![Feature Space and ROC Curves](image)

**Fig. 3.** The effects of synthetic data and Domain Adaptation on the feature space (left) and the receiver operating characteristic curves (right).

To support the claim, that the resemblance of real and synthetic data is strong enough, we analyze how adding synthetic data in training influences the robustness on unseen real data. Figure 3 shows the ROC curves for three experiments. The baseline experiment, a network trained on only real data, leads to the lowest AUC. Adding synthetic data improves the results slightly, with domain adaptation increasing the AUC further. Therefore, we conclude, that adding synthetic data in training does not harm the generalization on real data notably. In conclusion, the degree of realism in the synthetic smoke videos, used here is high enough to expect a comparable behavior. Results presented in the following all correspond to outputs of the network trained with Domain Adaptation.

**Robustness against Environmental Influences**

The set used in the following consists of 522 videos of blender smoke simulations on selected backgrounds. For a systematic modification, multiple parameters are altered in the simulation and render settings. We focus on the evaluation of the parameters smoke color, smoke density and illumination strength, as these are some of the most important alterations in real smoke too. Figure 4 visualizes the effect of such alterations on the generated videos. We choose different backgrounds for each setting.

We evaluate the robustness of the given DL network by analyzing the number of frames in a video leading to a classification score higher than a certain threshold. We have chosen an alarm threshold as the score
leading to a 2% false positive rate in the real test set. The number of frames leading to a classification score above this threshold has been used as the measure. The total number of frames containing smoke is 115, which is therefore the maximum number of possible alarms. As the alarm threshold is purposely set very high, a correct detection of all smoke frames is not expected. The chosen metric does not refer to a realistic surveillance system but estimates how reliably alarms would be raised by a network processing single frames.

In the example in Figure 4, density and brightness are altered for a single background. Some settings lead to missed detections, as seen in the figure: very low contrast of smoke to background is critical. To support this assumption, we add the same smoke simulations to background videos of different brightness. Figure 5 shows the evaluations for two more backgrounds. The robustness against changes in density or brightness increases notably for brighter backgrounds.

Fig. 4. Alteration of single simulation and render parameters in a smoke simulation (left) and its effect on the number of alarms raised for a certain setting (right).

Fig. 5. Robustness evaluation of different background brightness levels of the same scene.

The second scenario is a typical test case for VSD algorithms. We simulated two smoke scenarios, which resemble the EN54 test fires TF2 and TF5. The first corresponds to a scenario of smoldering wood, meaning slowly expanding bright smoke. The second smoke behavior
successively. Figure 6 shows the number of alarms in a sequence for videos with a high distance between smoke and camera. As the figure shows, after a distance of about 80 m (in units of blender), no alarm is produced by the network. Probably this is caused by the low resolution of 256×256 pixels used as inputs to the network. Generally, the bright, slow moving smoke of a simulated wood fire is detected more reliably. This could be caused by a bias towards bright smoke in the training data, especially in the case of videos with a high distance between smoke and camera.

![Number of Alarms vs Distance to Camera](image)

Fig. 6. The number frames leading to scores above the alarm threshold for smoke at different distances to the camera.

**Conclusion and Outlook**

We introduced a concept for testing the robustness of DL Algorithms for VSD by using synthetic data. We analyzed Domain Adaptation to reduce the effects of potential biases and distortions in synthetic smoke data. The difference between real and synthetic data has been analyzed by a t-SNE dimensionality reduction of the extracted feature space. After the differences in feature space were reduced, the trained networks were tested on a set of synthetic smoke videos, were multiple parameters were altered systematically.

We have shown, that, although network’s predictions were influenced by different parameters, they reliably produced alarms for the majority of settings. Most critical settings were identified as such, where the resulting smoke color was very similar to the background. A second analysis showed how the distance of smoke to the camera influenced the reliability of predictions. Smoke closer to the camera has been detected more reliably, with a distance of 80m being the maximum value leading to an alarm.
These results can give rough estimations of limitations and problems of the analyzed algorithm in real applications. The analysis done here can be extended in numerous ways, investigating a large number of further parameters.

References


