Balancing Quality and Quantity: The Impact of Synthetic Data on Smoke Detection Accuracy in Computer Vision

Ethan Seefried* Changsoo Jung* Jack Fitzgerald
Mariah Bradford Trevor Chartier Nathaniel Blanchard

Computer Science, Colorado State University, Fort Collins, Colorado
{eseefrie, Changsoo.Jung, Nathaniel.Blanchard}@colostate.edu

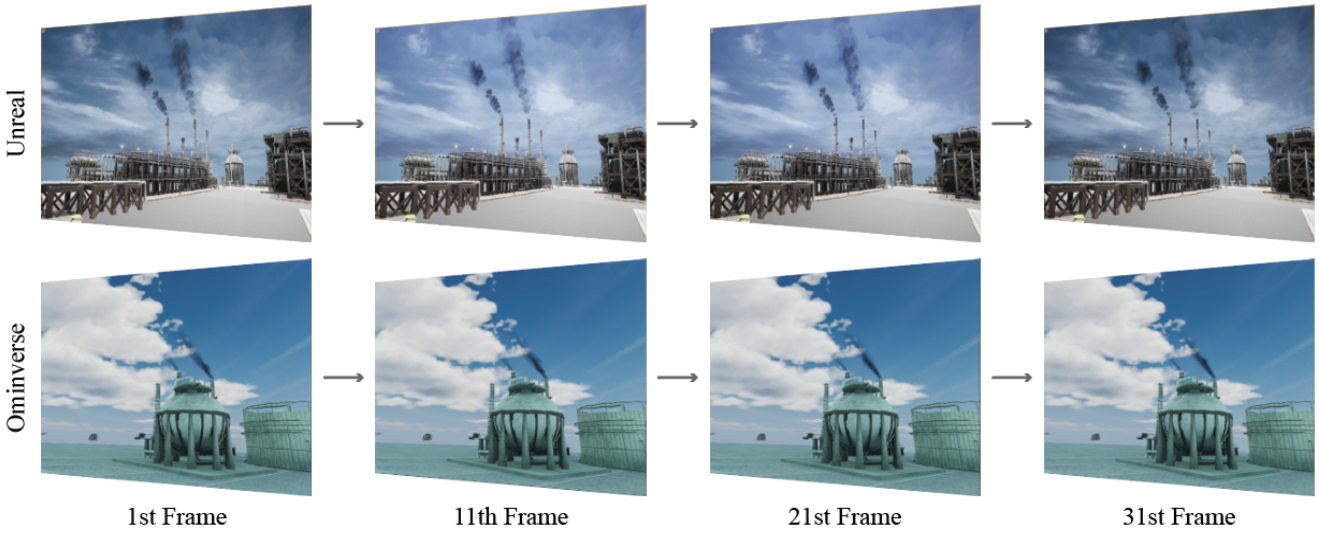


Figure 1. An overview of two distinct methods for generating synthetic smoke, as detailed in this paper. The top sequence showcases lower quality data produced using Unreal Engine 5, while the bottom sequence features higher quality data from NVIDIA Omniverse. Each set displays four frames from different intervals of the synthetic clips (frames 1, 11, 21, and 31), illustrating how the data was incorporated into a model.

Abstract

Synthetic data plays a crucial role in augmenting limited or challenging datasets. One domain with a scarcity of publicly available datasets is environmental monitoring of smoke opacity. Smoke presents a novel challenge for computer vision because its shape is amorphous and the texture is inconsistent. The dearth of public smoke datasets necessitates the generation of synthetic data to augment existing datasets. However, the generation of synthetic smoke, and explorations of how quantity and synthetic quality affects downstream model performance, remains largely unexplored. Here, we present SemiS, a novel, state-of-the-art deep learning model tailored to extract features from smoke, and use it to investigate the impact of synthetic smoke data. We used two synthetic smoke pipelines: 1) lower quality but quick to produce smoke generated with Unreal Engine, and 2) higher quality but slow to produce smoke from NVIDIA Omniverse. Across both pipelines, we found SemiS's performance peaked when synthetic data constituted approximately 30% of the initial training data. Further, higher quality data enhanced training accuracy by approximately 5%, compared to a 2.5% increase achieved with lower quality data. However, Omniverse was ∼12% slower to generate than Unreal. Finally, we dissect the quality of the generated smoke features in comparison with non-synthetic smoke. These results demonstrate the usefulness of developing methodologies that determine the value of synthetic data by analyzing their ability to improve model performance in smoke detection and similar applications.

* denotes equal contribution.

1. Introduction

In the realm of computer vision, amorphous objects like smoke plumes, which adhere to the dynamics of fluids and gases, pose substantial challenges. These objects' inherent variability and complexity defy the rigid, geometric assumptions prevalent in traditional computer vision algorithms [13, 18, 49]. In this work, we explore how synthetic data influences deep learning models for smoke in the context of industrial smoke plumes — a critical task for environmental monitoring is the identification and estimation of smoke opacity. However, the scarcity, limited diversity, and niche nature of smoke datasets significantly hinder this effort. One simple and low cost solution is to simply generate synthetic smoke to augment training data, but little is known about how variables such as the quantity and quality of synthetic smoke influence the performance of computer vision models. Here, we meticulously evaluate the influence of these variables across the training and evaluation of SemiS, a novel state-of-the-art deep learning model specifically designed to extract distinctive features from smoke.

For this work, we employ two methods to generate smoke: game engines (like Unity and Unreal Engine) and physics simulation platforms, e.g., NVIDIA Omniverse. Given the distinct characteristics of these methods, this paper delves into a critical examination of how these differences in quality influence the accuracy of computer vision models. By contrasting the more accessible yet potentially less detailed data from Unreal Engine with the highly realistic simulations provided by NVIDIA Omniverse, we aim to uncover the extent to which the fidelity of synthetic smoke affects model performance. Across generation methods, we also investigate the thresholds at which the quantity of synthetic data becomes either insufficient or excessive. This investigation not only highlights the importance of selecting appropriate synthetic data generation methods for specific applications but also sheds light on the broader implications for the development and training of robust computer vision systems capable of interpreting complex, dynamic phenomena. By examining these techniques, our goal is to determine the optimal use of synthetic data for modeling smoke and other amorphous phenomena, enhancing the quality of complex datasets.

While our analysis focuses on smoke plumes, we believe this work establishes a foundation to understand how synthetic data quality impacts the accurate modeling of similar amorphous phenomena. Further, our findings not only contribute to the discourse around strategically selecting synthetic data generation methods tailored to specific computer vision tasks, but they also contribute to the broader discourse on leveraging synthetic data to enhance the robustness and accuracy of computer vision systems. The contributions of this work can be summarized below:

- Both high and low quality synthetic data have been generated for the task of smoke detection, exploring the difficulty in creating accurate smoke.
- Through rigorous experimentation, we have identified key thresholds for the amount of synthetic data required for varying levels of quality, when training on amorphous objects.
- To our knowledge, we are the first to explore the generation of smoke data through game engines and physics simulators for computer vision.

2. Related Works

With the recent advancements in deep learning, the benefits of large quantities of data have become obvious. Deep learning techniques thrive when training over a large, diverse dataset, but collecting this data is not always straightforward. Difficulties in collecting large-scale data include ensuring quality, addressing scarcity in the dataset, and maintaining privacy and fairness [8, 27]. One potential path for addressing these challenges is generating synthetic data [6]. Data is synthetic when it was not directly collected, but rather manufactured in some way. There are various ways to create synthetic data. Popular methods identified in recent work include manual generation, variational autoencoders (VAEs), generative adversarial networks (GANs), synthetic composite imagery, and virtual synthetic data [27, 28, 33, 36]. In manual generation, the synthetic data is handcrafted to mimic real data or to add a dimension to existing data. In VAEs and GANs, artificially intelligent systems generate new samples after training over given data. Synthetic composite imagery refers to the process of combining data samples to create new samples. Virtual synthetic data has proven to be a valuable method of creating new data via virtual worlds, such as game engines [5, 7, 19–21, 37].

For the task of smoke detection, synthetic data has proven to be a useful resource to improve model performance [30, 45–47]. Various methods mentioned above are applied to generate smoke. For example, [46] used two GANs to produce synthetic smoke images, and they found that images from the higher quality GAN resulted in better smoke detection. Similarly, [44] developed a pipeline to generate synthetic smoke images that allowed for adjustable parameters to yield desired smoke components. Several previous works used Blender for manually generating synthetic smoke data [30, 45, 47]. However, generating a variety of quality smoke images can be difficult and some of the input can be automated [45].

Determining measures of quality in synthetic data is important for understanding its impact on model performance, and much work has been done to create such metrics [2, 8, 9, 40]. The Fréchet Inception Distance [14] was utilized in [38] to determine the quality of synthetic data generated by a GAN. In [39], Peak Signal-to-Noise ratio (PSNR)

and Structural Similarity Index Measure (SSIM) [43] are introduced into the loss function of a GAN with the aim of reducing noise and thus improving quality.

The impact of synthetic data quality varies by task and field. For example, in a review of synthetic data, [28] found that studies on photorealistic synthetic data presented different results, and that the impact depended on the task. Previous work has found that object detection improved with photorealistic synthetic data [31, 42]. Still, synthetic data created using domain randomization yields better results than using only real data [42]. It is important to consider that photorealistic synthetic data has a higher computational cost to produce, and unrealistic data does still show improvements in model performance [28]. The trade-off between computational cost and model improvement is an unanswered question which will likely vary by task. Here, we dive deeper into this problem for the task of smoke detection in industrial settings.

3. Dataset

In the following section we discuss the real world data used to test the model, as well as our techniques for generating synthetic data.

3.1. Real World Data

In this study, we utilize a novel real-world smoke dataset, currently undergoing peer review for potential public release. Though not yet publicly available, this dataset marks a significant contribution to smoke detection research by offering a diverse and challenging benchmark for assessing the effectiveness of synthetic data. We utilize a small portion of this dataset, comprising of 1,774 video clips, with 1,554 featuring smoke and 220 without. It is divided into training, validation, and testing sets to facilitate a thorough evaluation: 370 smoke and 190 non-smoke clips for training, 319 smoke and 6 non-smoke clips for validation, and 865 smoke and 24 non-smoke clips for testing. The lack of data size and distribution is already focused on in other studies [3, 4, 17, 32], but we propose to address this problem by incorporating synthetic data into the training set in our study. In addition, our dataset distribution, particularly the expansive unseen testing set, is proposed for enhanced generalization on evaluation, despite the constraints on the size of training set, which may help in future works, such as opacity predictions.

To address the complexity of smoke detection, we focus on the opacity of smoke, quantified by the equation:

$$Opacity = \left(1 - \frac{I}{L}\right) \times 100, \tag{1}$$

where I/L represents the transmittance of light through the smoke plume, which after being subtracted from one may be converted into a percentage [16]. Given that detecting smoke can be straightforward, our study only includes clips with opacity values between 5-30%, thus elevating the detection challenge by focusing on subtler smoke patterns.

Figure 2 underscore the dataset's diversity and the nuanced task of identifying low-opacity smoke under different conditions, pivotal for testing our model's accuracy.

Figure 2. Comparison of image quality across simulated and real-world Data: This figure illustrates a side-by-side quality comparison of images from Unreal Engine, NVIDIA Omniverse, and real-world environments, with and without the presence of smoke. These comparisons highlight the simulated datas ability to mimic real-world conditions and demonstrates the effects of environmental elements like smoke on image quality.

3.2. Unreal Engine 5

Unreal Engine 5 emerges as a promising tool in synthetic data generation, offering an accessible platform that is both easy to learn and use. Its abundant availability of free or affordable assets enables rapid scene creation, allowing researchers to swiftly commence data generation. Furthermore, its lower computational power requirements make it an attractive option for research institutions worldwide,

making the ability to perform synthetic data generation more accessible. However, while Unreal Engine facilitates quick setup and initial data generation, the fidelity of data for amorphous objects like smoke may be compromised. Achieving high-quality representations of such complex phenomena often necessitates access to advanced simulation tools specifically designed for use within these game engines. Although this lower quality might not significantly impact training for general datasets, the nuanced and fine-grained details essential for accurately detecting smoke demand a superior level of data realism. Figure 2 showcases the qualitative differences with real-world and Omniverse data, illustrating the variance in backgrounds and smoke generated using Unreal Engine. For our study, we integrated up to a total of 280 clips generated via Unreal Engine into our training set, evenly split between 140 clips depicting smoke and 140 clips without, to evaluate the engine's efficacy in supporting smoke detection research.

3.3. NVIDIA Omniverse

To achieve more realistic smoke simulations, we opted for Nvidia Omniverse, a physics simulator known for its high-fidelity outputs. While Omniverse offers unparalleled detail and realism, it introduces specific challenges, including a limited selection of publicly available assets and the need for substantial computational resources. Our experience revealed that running the engine optimally requires at least a 2080 GPU, yet we encountered notable performance issues on a single 3090 GPU. Performance markedly improved when we enabled multi-GPU mode (two 3090's), leading to more efficient data generation. The quality of smoke simulations generated by Omniverse was notably higher to that produced by Unreal Engine. Smoke visualizations in Omniverse were almost indistinguishable from real-world smoke to the human eye, in stark contrast to the more artificial appearance of smoke from Unreal Engine. This visual distinction raised an intriguing question for our research: How significant is the impact of such high-quality synthetic data on model performance? Given our focus on smoke detection, it was essential to explore whether the enhanced realism of Omniverse-generated smoke would translate into measurable improvements in model accuracy. Preliminary findings suggest that while the visual quality difference is apparent to the human eye, the incremental benefit for model training, especially in distinguishing smoke from no-smoke scenarios, might be nuanced. Our investigation aims to quantify this effect, assessing whether the superior visual fidelity of Omniverse simulations offers a tangible advantage in training accuracy compared to Unreal Engine's output.

4. Methods

Extracting smoke features from image sequences poses notable challenges, especially in environments where smoke is subtle or when limited real-world smoke data is available for model training. To overcome these obstacles and the impracticality of collecting a vast array of real-world smoke videos, we propose a novel approach that enhances data richness without extensive real-world datasets. Our methodology employs a Residual 3D block-based architecture, enriched with Local Binary Pattern (LBP) and Normalized Absolute Difference (NAD) techniques, to effectively capture smoke dynamics and features. This paper details the SemiS architecture, including the implementation of LBP and NAD (detailed in Sections 4.2 and 4.3, respectively), our customized loss function, and the specifics of our training regimen, outlining how each component contributes to the robust detection of smoke patterns under varied conditions.

4.1. Semi-Synthetic Smoke Detector (SemiS)

In this work, we introduce Semi-Synthetic Smoke Detector (SemiS), a novel architecture designed to extract smoke features and distinguish smoke and non-smoke features. We employed two modules, Residual 3D blocks [41], to extract visual features from RGB channels and texture features from LBP frames. For efficient computation, we select only four frames from a 1.4 second video.

For the selection of frames from real-world data $\boldsymbol{I} \in \mathbb{R}^{40\times3\times224\times224}$, we defined the indices of the selected frames as:

$$i_{\text{real}} = \{i \times 10 \mid i \in \{0, 1, 2, 3\}\}. \tag{2}$$

Given the differences in frame rates between the synthetic (60 FPS) and real-world (24 FPS) data, we ensured temporal alignment of the frames to maintain consistency across the datasets. This alignment was achieved by calculating the indices for the synthetic data frames to match the temporal sequence of the real-world data, facilitating accurate comparison and integration, using the equation:

$$i_{\text{syn}} = \{i \times 10 \times \left(\frac{60}{24}\right) \mid i \in \{0, 1, 2, 3\}\}. \tag{3}$$

Following the initial selection of input frames, the textures of each frame, denoted as $\boldsymbol{T} \in \mathbb{R}^{4\times1\times224\times224}$, were derived from the selected RGB data $\hat{\boldsymbol{I}} \in \mathbb{R}^{4\times3\times224\times224}$ utilizing the Local Binary Patterns (LBP) technique (see Section 4.2). Subsequently, the changes $\boldsymbol{C} \in \mathbb{R}^{3\times1\times224\times224}$ between the selected frames—encompassing both RGB and texture information—were computed via the Normalized Absolute Difference (NAD) module (see Section 4.3). These computed changes were then input into Residual 3D Blocks to ascertain smoke movement through temporal frame differences (see Figure 3). The integrated features from both RGB $\hat{\boldsymbol{I}}$ and texture data $\boldsymbol{T}$ facilitated the prediction of probabilities for the non-smoke and smoke categories, represented as $\hat{\boldsymbol{P}} \in \mathbb{R}^{N\times2}$.

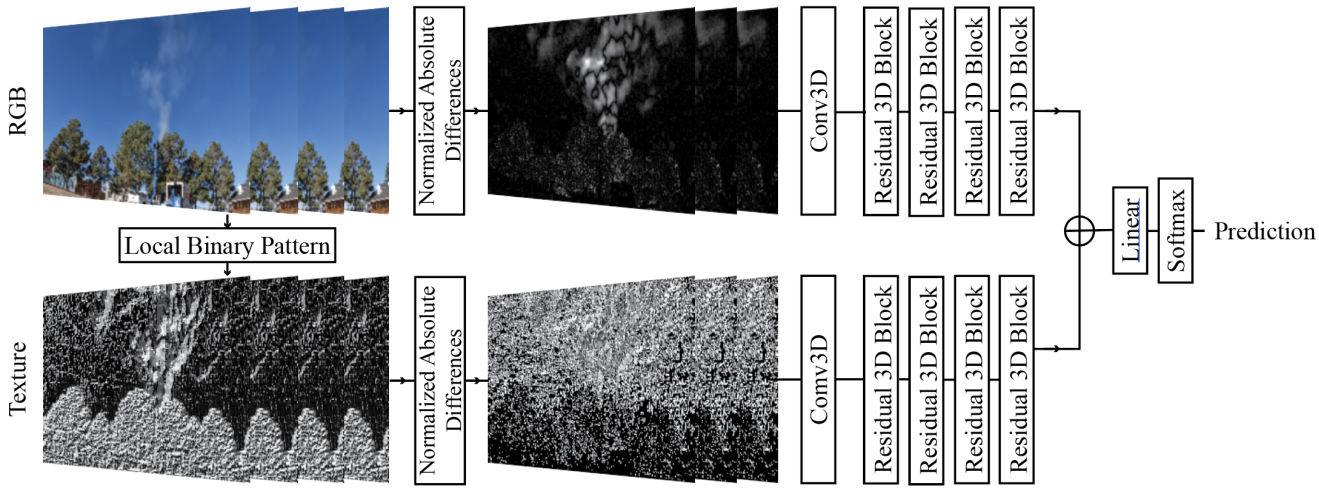


Figure 3. Our architecture (SemiS): Our approach transforms selected input frames into textures using Local Binary Patterns (LBP) and utilizes normalized absolute differences between RGB and texture frames to identify smoke features. These features, derived from RGB and texture changes, are combined to accurately detect smoke presence.

In our pursuit of achieving a balance between model accuracy and computational efficiency, a lightweight iteration of the Residual 3D Block was implemented. To preserve temporal information without incurring substantial computational costs, both the kernel and stride dimensions were meticulously reduced. This modification aimed to accelerate inference speed, making our architecture suitable for real-time deployment. Moreover, by selecting only four frames within each 1.4 second video clip, we succeed in optimizing detection accuracy while concurrently minimizing the computations.

4.2. Local Binary Pattern (LBP)

The Local Binary Patterns (LBP) method [35] is a crucial technique for texture analysis and image pattern recognition [1, 11, 15, 23]. Employed in our methodology to extract distinctive texture features of smoke, LBP used a preprocess that can see the spatial structure of an image to highlight the fine texture details (see Figure 4). This section outlines the implementation of an efficient LBP computation method tailored to our application.

Initially, RGB images $\hat{\boldsymbol{I}} \in \mathbb{R}^{4\times3\times224\times224}$ were converted to grayscale images $\boldsymbol{G} \in \mathbb{R}^{4\times224\times224}$ to simplify the texture analysis. The grayscale images were then padded to facilitate neighborhood processing, with a preference for 'reflect' padding but defaulting to zero padding if necessary.

The essence of the LBP process involves comparing the intensity of each pixel to its eight surrounding neighbors. This comparison yields a binary value for each pixel, encapsulating local texture information. These binary values were then weighted by their spatial positions and summed to produce the center pixel of the texture images.

Our LBP computation was vectorized for efficiency, ensuring fast processing suitable for large-scale or real-time applications. This method effectively captures essential texture features critical for our subsequent analysis as smoke detection.

4.3. Normalized Absolute Difference (NAD)

The unpredictability of smoke movement, coupled with challenges posed by low smoke opacity and color similarity to the background, necessitates a robust approach to smoke detection. To address these challenges, we select four frames that allow us to see temporal changes in pixel values, thereby enhancing our model's ability to detect subtle smoke movements. Specifically, in real-world datasets, we took the 1st, 11th, 21st, and 31st frames, while for synthetic datasets, the 1st, 26th, 51st, and 76th frames were chosen. Maintaining a constant time gap facilitates the observation of pixel value changes over time, as illustrated in Figure 4. Despite weather-related visibility issues, the effectiveness of this approach was evidenced by the discernible smoke movement in the third column of Figure 4.

To capitalize on these observations, we proposed an innovative method that combines the temporal changes in both RGB and texture data, which are then fed into our neural network. This approach allowed us to extract features associated with the dynamics of smoke movement and significantly enhanced our model's detection capabilities in diverse and challenging conditions.

4.4. Loss Function

Our model employs a confidence loss to measure the discrepancy between the predicted confidences and the ground

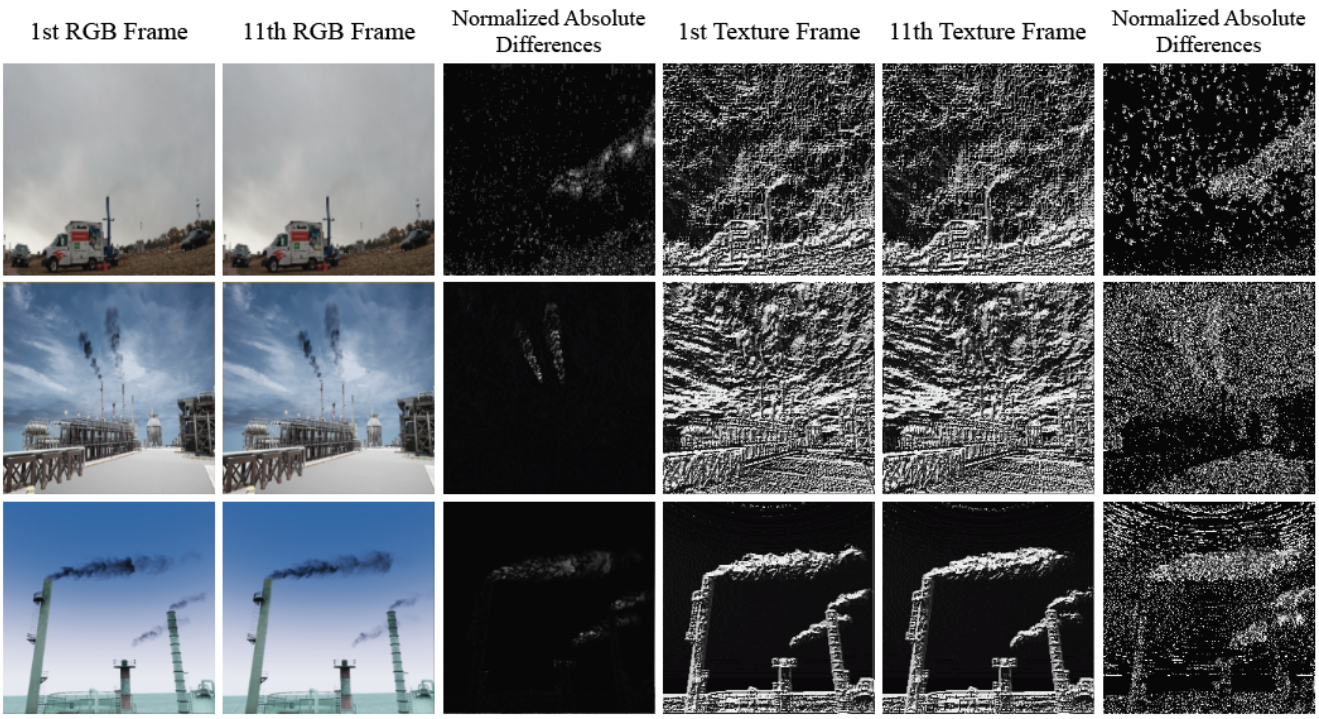


Figure 4. Comparison of RGB and texture Data in Smoke Feature Extraction: This figure illustrates the normalized absolute differences between the 1st and 11th frames for both RGB and texture data, with the 3rd and 6th columns specifically highlight these variations. The comparison underscores the qualitative disparity in smoke feature extraction between Unreal Engine and NVIDIA Omniverse-generated data. Notably, Omniverse data demonstrates a superior ability to delineate smoke features across frames, significantly improving the model's capacity to recognize authentic smoke patterns during training with real-world datasets. This visualization emphasizes the pivotal role of high-quality synthetic data in refining computer vision models for more accurate smoke detection.

truth labels for the two classes: "no smoke" and "smoke". For a given batch of size N, the predicted confidence matrix $\hat{P} \in \mathbb{R}^{N\times 2}$ and the ground truth label matrix $P \in \mathbb{R}^{N\times 2}$. The element-wise multiplication of $1-\hat{P}$ and P results in a matrix M_{conf} that represents the correct predictions' missing confidences. The confidence loss L_{conf} is then computed as the mean over all elements of M_{conf}, formally defined as:

$$L_{conf} = \frac{1}{N}\sum_{i=1}^{N}\left[w \cdot (1-\hat{P}_{i,0}) \cdot P_{i,0} + (1-\hat{P}_{i,1}) \cdot P_{i,1}\right] \tag{4}$$

where N is the number of instances, $\hat{P}_{i,j}$ represents the predicted probability of instance i for class j, and $P_{i,j}$ is the actual label of instance i for class j, with $j = 0$ for the non-smoke class and $j = 1$ for the smoke class. For both the training and validation phases, we assign a weight $w = 2$ to the non-smoke class to emphasize its importance in the model's learning process. However, during the testing phase, we adjust this weight back to $w = 1$ to evaluate the model's performance under standard class weighting conditions.

The overall loss function, which combines the cross-entropy loss (L_{CE}) and the modified confidence loss (L_{conf}), is given by:

$$L = L_{CE} + L_{conf} \tag{5}$$

where L_{CE} is the cross-entropy loss which is commonly used for classification task [22, 29, 48] calculated for the class predictions and the true labels.

This loss function encourages the model to increase the confidence for correct predictions, effectively minimizing the difference between the predicted confidences and the actual labels.

In our investigation, we adapted the confidence loss, denoted as L_{conf}, to evaluate the certainty with which each model predicts its decisions. Given that P represents the probability output from a Softmax layer, we introduce a measure, $Conf$, to quantify model confidence as a percentage:

$$Conf(\%) = L_{conf} \times 100 \tag{6}$$

This formulation allows us to translate the confidence loss into a more intuitive metric, enabling a straightforward

comparison of decision confidence across different models. Through this approach, we can assess not only the accuracy but also the reliability of predictions made by our models, highlighting their effectiveness in practical scenarios.

4.5. Training Details

SemiS was trained with a batch size of 32, using an initial learning rate of $1e^{-6}$. The learning rate was halved at the 10th, 20th, 30th, and 40th epochs during 50 epochs. AdamW [26] optimization was used, and the SiLU activation function [10] was employed in constructing the residual 3D blocks. Before starting each training, the residual 3D blocks are initialized by the Kaiming initialization [12].

5. Experiments

In our experiments, we address the challenge of limited real-world data availability by starting with a modest set of 560 real-world samples. To explore the effectiveness of synthetic data in enhancing model performance, we incrementally introduced additional synthetic samples, each increment amounting to 5% of the original real-world dataset size, aiming to identify the optimal synthetic-to-real data ratio for improved model accuracy.

Moreover, to intensify the challenge and more closely mimic real-world complexities, the smoke featured in the real-world data was deliberately chosen to have an opacity of 30% or less. This choice was made to simulate the difficulty models face in detecting low-opacity smoke, which is often more subtle and harder to distinguish. Conversely, the synthetic data was generated with higher opacity levels, with the intention of facilitating the model's learning process by providing clearer examples of smoke features. This experimental setup was designed not only to test the model's ability to learn from limited data but also to evaluate the impact of synthetic data quality, in terms of opacity, on the learning outcomes.

Table 1 presents a comparison between our SemiS model and established baseline models, specifically a 3D ResNet model (R3D) [41] and a tiny Video Swin Transformer (VST) [25], with training conducted solely on real-world data. R3D extends the traditional 2D ResNet framework into three dimensions, adapting it for action recognition tasks in video sequences. Similarly, VST adapts the Swin Transformer [24] architecture for video analysis, leveraging its strengths in capturing complex spatial-temporal relationships. Our SemiS model outperformed these baselines, achieving the highest accuracy of 89.99%, demonstrating its capability in accurately detecting smoke features and indicating its potential for advancing computer vision tasks involving amorphous objects.

To investigate the impact of synthetic data on model performance, we augmented the initial set of 560 real-world training samples with synthetic data generated from both

Method	Parameters	Accuracy	Conf (↓)	VPS (↑)
R3D	33.4 M	72.55 %	36.47 %	37.1
VST (t)	28.2 M	84.70 %	**32.05** %	35.5
SemiS (ours)	**9.2 M**	**89.99 %**	37.72 %	**41.5**

Table 1. Results for the baseline tests for smoke detection over the real world dataset only. Two baseline models, one CNN (R3D) and one video transformer (VST Tiny), were compared to our efficient (9.2M parameters) model, named SemiS. We achieved the highest accuracy by 8.5% over the best baseline VST.

Unreal Engine and Omniverse. As depicted in Figure 5, the classification accuracy of SemiS consistently showed higher results with the Omniverse-generated synthetic data, indicating its superior alignment with the nuances of real-world smoke detection. The optimal integration of synthetic data, enhancing accuracy maximally, was found to be an additional 30% of the original dataset size for both sources. Further details of the experiments between Unreal Engine and Omniverse data are detailed in the appendix in Table A.1 and Table A.2.

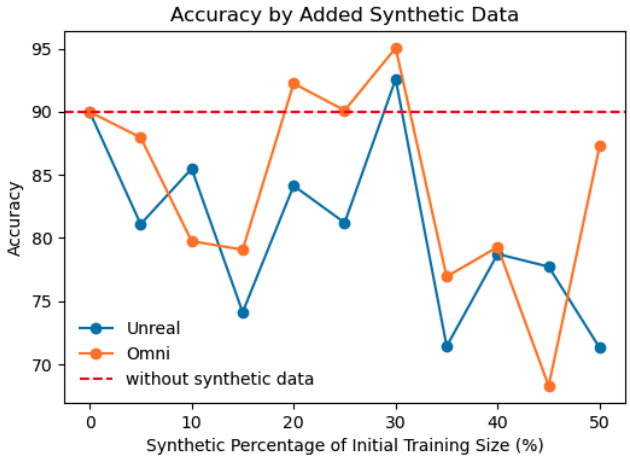


Figure 5. Graph of synthetic data integration vs. model accuracy: This graph plots the model's accuracy as a function of the synthetic data proportion added to the training set, where 50 indicates that 50% of the initial real-world data size was incorporated as synthetic data. The red line highlights the baseline accuracy achieved with no synthetic data. The trend illustrates how incorporating synthetic data impacts the model's performance, providing a visual comparison to the baseline scenario.

6. Discussion

The goal of SemiS is to achieve efficiency in training and inference times, while also achieving a high level of accuracy in detecting smoke. The inclusion of opacity labels in the real-world dataset we used enabled us to purposefully make

Module	Parameters	Accuracy	Conf (↓)
RGB	4.5 M	74.35 %	**37.56 %**
Texture	4.5 M	80.43 %	45.07 %
RGB + Texture	9.2 M	**89.99 %**	37.72 %

Table 2. This table outlines our ablation study results, comparing the accuracy of using RGB data, texture data, and their combination for smoke detection. RGB alone achieved 74.55% accuracy, texture alone reached 80.43%, and their integration significantly improved baseline accuracy to 89.99%. These results demonstrate the enhanced performance achieved through combining RGB and texture features in our SemiS model.

the smoke detection task more difficult by selecting clips of smoke releases where the opacity of the smoke was 30% or lower. In theory, low opacity smoke should be more difficult for the model to learn features from since it is more subtle. The small amount of synthetic data that we introduced contained smoke that was more clearly discernible with a higher opacity than the real-world data. As shown in Figure 3 and Figure 4, the normalized absolute difference (NAD) technique utilized by the SemiS architecture highlights the features of the smoke more clearly for both real and synthetic data, especially when compared to the RGB inputs. However, the synthetic data had no background noise which made the smoke more clearly highlighted when NAD was applied to them. We did not perform training on just the purely synthetic data, which is in contrast to other works [34, 37].

Our results show that there is likely an ideal portion of synthetic data to use where the accuracy will increase over the baseline; in our experiments this proportion seemed to be around 30%. However, the accuracy improved more with synthetic data generated by Omniverse, which could likely be due to the higher fidelity of the smoke generated by the engine. Besides the level of fidelity, there are several disparities between the layouts of the Unreal Engine and Omniverse scenes. As a result, it is possible that the difference in performance of these datasets could be partially attributed to the subtle differences in the layout of the floors, buildings, smoke stacks, sky, etc.

6.1. Ablation Study

In our research, an ablation study was conducted to ascertain the contribution of each key component within our model—specifically examining the roles of RGB and texture (represented by greyscaled videos) features. Table 2 outlines the findings, clearly demonstrating the value added by each component. Utilizing only RGB video data resulted in the lowest accuracy at 74.55%, indicating the challenges of relying solely on color information for smoke detection. Incorporating texture as a standalone feature significantly improved model accuracy to 80.43%, underscoring its importance in recognizing smoke patterns regardless of color. Most notably, the integration of both RGB and texture features together enhanced the model's accuracy to 89.99%. This combination leverages the comprehensive understanding of smoke dynamics—color variations captured through RGB and structural details through texture—facilitating a more robust and accurate detection. The results from this ablation study highlight the critical balance between color and texture recognition in achieving high performance in smoke detection tasks.

7. Conclusion

In this study, we explored the potential of synthetic data for tackling challenging computer vision tasks, with a particular focus on the detection of amorphous objects such as smoke. This exploration is pivotal for advancing applications in opacity and emissions predictions, where traditional datasets may fall short. Faced with the constraints of limited and complex real-world datasets, our investigation centered on the strategic integration of synthetic data to enhance model robustness and accuracy, probing the optimal balance of quantity and quality necessary for such data. Through the generation of synthetic smoke data of varying fidelity—utilizing Unreal Engine 5 for lower fidelity and NVIDIA Omniverse for higher realism—and leveraging a novel, efficient model architecture, we established a performance baseline without synthetic data. Incrementally, we introduced synthetic data into the training regimen in 5% increments, culminating in a dataset comprised of 50% synthetic data relative to the initial real-world dataset size. Our methodology demonstrated that high-quality synthetic data from Omniverse significantly boosts model accuracy by approximately 5%, surpassing not only the baseline but also the enhancements afforded by lower fidelity synthetic data from Unreal Engine, which itself offered a 2.5% increase in accuracy.

These findings underscore the critical role of synthetic data quality and quantity in optimizing computer vision models, offering invaluable insights for those limited by resources or requiring enhanced dataset diversity. Looking forward, we see this work as foundational for the broader application of synthetic data in computer vision, especially in areas plagued by dataset scarcity or the need to model complex phenomena. It is our aspiration that the insights garnered from this study will fuel further innovations and research directions, encouraging a deeper exploration into the capabilities and applications of synthetic data for amorphous object detection, classification and beyond.

References

[1] Timo Ahonen, Abdenour Hadid, and Matti Pietikäinen. Face recognition with local binary patterns. In *Computer Vision-ECCV 2004: 8th European Conference on Computer Vision, Prague, Czech Republic, May 11-14, 2004. Proceedings, Part I 8*, pages 469–481. Springer, 2004. 5

[2] Ahmed Alaa, Boris Van Breugel, Evgeny S. Saveliev, and Mihaela van der Schaar. How Faithful is your Synthetic Data? Sample-level Metrics for Evaluating and Auditing Generative Models. In *Proceedings of the 39th International Conference on Machine Learning*, pages 290–306. PMLR, 2022. 2

[3] Guangzhou An, Masahiro Akiba, Kazuko Omodaka, Toru Nakazawa, and Hideo Yokota. Hierarchical deep learning models using transfer learning for disease detection and classification based on small number of medical images. *Scientific reports*, 11(1):4250, 2021. 3

[4] Bjorn Barz and Joachim Denzler. Deep learning on small datasets without pre-training using cosine loss. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2020. 3

[5] Steve Borkman, Adam Crespi, Saurav Dhakad, Sujoy Ganguly, Jonathan Hogins, You-Cyuan Jhang, Mohsen Kamalzadeh, Bowen Li, Steven Leal, Pete Parisi, Cesar Romero, Wesley Smith, Alex Thaman, Samuel Warren, and Nupur Yadav. Unity Perception: Generate Synthetic Data for Computer Vision, 2021. arXiv:2107.04259 [cs]. 2

[6] Tomáš Bubenıcek. Using game engine to generate synthetic datasets for machine learning. 2022. 2

[7] David Conde, Joaquín Martínez, Jesús Balado, Pedro Arias, and CINTECX GeoTECH. Generation of road zone synthetic data for training mot models with the nvidia omniverse platform. 2

[8] Fida K Dankar and Mahmoud Ibrahim. Fake it till you make it: Guidelines for effective synthetic data generation. *Applied Sciences*, 11(5):2158, 2021. Publisher: MDPI. 2

[9] Fida K Dankar, Mahmoud K Ibrahim, and Leila Ismail. A multi-dimensional evaluation of synthetic data generators. *IEEE Access*, 10:11147–11158, 2022. Publisher: IEEE. 2

[10] Stefan Elfwing, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural networks*, 107:3–11, 2018. 7

[11] Abdenour Hadid. The local binary pattern approach and its applications to face analysis. In *2008 First Workshops on Image Processing Theory, Tools and Applications*, pages 1–9, 2008. 5

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015. 7

[13] Lijun He, Xiaoli Gong, Sirou Zhang, Liejun Wang, and Fan Li. Efficient attention based deep fusion cnn for smoke detection in fog environment. *Neurocomputing*, 434:224–238, 2021. 2

[14] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 2

[15] Di Huang, Caifeng Shan, Mohsen Ardabilian, Yunhong Wang, and Liming Chen. Local binary patterns and its application to facial image analysis: A survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 41(6):765–781, 2011. 5

[16] Kirk Foster Karen Randolph. Visible emissions field manual epa methods 9 and 22. 1997. 3

[17] Muzammil Khan, Muhammad Taqi Mehran, Zeeshan Ul Haq, Zahid Ullah, Salman Raza Naqvi, Mehreen Ihsan, and Haider Abbass. Applications of artificial intelligence in covid-19 pandemic: A comprehensive review. *Expert systems with applications*, 185:115695, 2021. 3

[18] Salman Khan, Khan Muhammad, Tanveer Hussain, Javier Del Ser, Fabio Cuzzolin, Siddhartha Bhattacharyya, Zahid Akhtar, and Victor Hugo C de Albuquerque. Deepsmoke: Deep learning model for smoke detection and segmentation in outdoor environments. *Expert Systems with Applications*, 182:115125, 2021. 2

[19] Aman Kishore, Tae Eun Choe, Junghyun Kwon, Minwoo Park, Pengfei Hao, and Akshita Mittel. Synthetic data generation using imitation training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3078–3086, 2021. 2

[20] Heejae Lee, Jongmoo Jeon, Doyeop Lee, Chansik Park, Jinwoo Kim, and Dongmin Lee. Game engine-driven synthetic data generation for computer vision-based safety monitoring of construction workers. *Automation in Construction*, 155: 105060, 2023.

[21] Shenghan Li, Yaolin Zhang, and Yi Tan. Game engine-based synthetic dataset generation of entities on construction site. In *International Symposium on Advancement of Construction Management and Real Estate*, pages 1602–1614. Springer, 2022. 2

[22] Xiaoxu Li, Liyun Yu, Dongliang Chang, Zhanyu Ma, and Jie Cao. Dual cross-entropy loss for small-sample fine-grained vehicle classification. *IEEE Transactions on Vehicular Technology*, 68(5):4204–4212, 2019. 6

[23] Shengcai Liao, Xiangxin Zhu, Zhen Lei, Lun Zhang, and Stan Z Li. Learning multi-scale block local binary patterns for face recognition. In *Advances in Biometrics: International Conference, ICB 2007, Seoul, Korea, August 27-29, 2007. Proceedings*, pages 828–837. Springer, 2007. 5

[24] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. 7

[25] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3202–3211, 2022. 7

[26] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 7

[27] Yingzhou Lu, Minjie Shen, Huazheng Wang, Xiao Wang, Capucine van Rechem, and Wenqi Wei. Machine Learning for Synthetic Data Generation: A Review, 2024. arXiv:2302.04062 [cs]. 2

[28] Keith Man and Javaan Chahl. A review of synthetic image data and its use in computer vision. *Journal of Imaging*, 8 (11):310, 2022. 2, 3

[29] Anqi Mao, Mehryar Mohri, and Yutao Zhong. Cross-entropy loss functions: Theoretical analysis and applications. In *International Conference on Machine Learning*, pages 23803–23828. PMLR, 2023. 6

[30] Jun Mao, Change Zheng, Jiyan Yin, Ye Tian, and Wenbin Cui. Wildfire Smoke Classification Based on Synthetic Images and Pixel- and Feature-Level Domain Adaptation. *Sensors*, 21(23):7785, 2021. Number: 23 Publisher: Multidisciplinary Digital Publishing Institute. 2

[31] Pablo Martinez-Gonzalez, Sergiu Oprea, Alberto Garcia-Garcia, Alvaro Jover-Alvarez, Sergio Orts-Escolano, and Jose Garcia-Rodriguez. Unrealrox: an extremely photorealistic virtual reality environment for robotics simulations and synthetic data generation. *Virtual Reality*, 24:271–288, 2020. Publisher: Springer. 3

[32] Hidetoshi Matsuo, Mizuho Nishio, Tomonori Kanda, Yasuyuki Kojita, Atsushi K Kono, Masatoshi Hori, Masanori Teshima, Naoki Otsuki, Ken-ichi Nibu, and Takamichi Murakami. Diagnostic accuracy of deep-learning with anomaly detection for a small amount of imbalanced data: discriminating malignant parotid tumors in mri. *Scientific Reports*, 10(1):19388, 2020. 3

[33] Celso M. de Melo, Antonio Torralba, Leonidas Guibas, James DiCarlo, Rama Chellappa, and Jessica Hodgins. Next-generation deep learning based on simulators and synthetic data. *Trends in Cognitive Sciences*, 26(2):174–187, 2022. Publisher: Elsevier. 2

[34] Matthias Müller, Vincent Casser, Jean Lahoud, Neil Smith, and Bernard Ghanem. Sim4cv: A photo-realistic simulator for computer vision applications. *International Journal of Computer Vision*, 126:902–919, 2018. 8

[35] Timo Ojala, Matti Pietikainen, and Topi Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on pattern analysis and machine intelligence*, 24(7):971–987, 2002. 5

[36] Goran Paulin and Marina Ivasic-Kos. Review and analysis of synthetic dataset generation methods and techniques for application in computer vision. *Artificial Intelligence Review*, 56(9):9221–9265, 2023. 2

[37] Ingeborg Rasmussen, Sigurd Kvalsvik, Per-Arne Andersen, Teodor Nilsen Aune, and Daniel Hagen. Development of a novel object detection system based on synthetic data generated from unreal game engine. *Applied Sciences*, 12(17): 8534, 2022. 2, 8

[38] JoÅ¾e M RoÅ¾anec, Patrik Zajec, Spyros Theodoropoulos, Erik Koehorst, BlaÅ¾ Fortuna, and Dunja MladeniÄ‡. Synthetic data augmentation using GAN for improved automated visual inspection. *Ifac-Papersonline*, 56(2):11094–11099, 2023. Publisher: Elsevier. 2

[39] Oleksii Sidorov, Congcong Wang, and Faouzi Alaya Cheikh. Generative smoke removal. In *Machine Learning for Health Workshop*, pages 81–92. PMLR, 2020. 2

[40] Joshua Snoke, Gillian M Raab, Beata Nowok, Chris Dibben, and Aleksandra Slavkovic. General and specific utility measures for synthetic data. *Journal of the Royal Statistical Society Series A: Statistics in Society*, 181(3):663–688, 2018. Publisher: Oxford University Press. 2

[41] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018. 4, 7

[42] Jonathan Tremblay, Aayush Prakash, David Acuna, Mark Brophy, Varun Jampani, Cem Anil, Thang To, Eric Cameracci, Shaad Boochoon, and Stan Birchfield. Training deep networks with synthetic data: Bridging the reality gap by domain randomization. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 969–977, 2018. 3

[43] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 3

[44] Chao Xie and Huanjie Tao. Generating Realistic Smoke Images With Controllable Smoke Components. *IEEE Access*, 8:201418–201427, 2020. Conference Name: IEEE Access. 2

[45] Gao Xu, Yongming Zhang, Qixing Zhang, Gaohua Lin, and Jinjun Wang. Deep domain adaptation based video smoke detection using synthetic smoke images. *Fire Safety Journal*, 93:53–59, 2017. 2

[46] Hang Yin, Yurong Wei, Hedan Liu, Shuangyin Liu, Chuanyun Liu, and Yacui Gao. Deep Convolutional Generative Adversarial Network and Convolutional Neural Network for Smoke Detection. *Complexity*, 2020:e6843869, 2020. Publisher: Hindawi. 2

[47] Qi-xing Zhang, Gao-hua Lin, Yong-ming Zhang, Gao Xu, and Jin-jun Wang. Wildland Forest Fire Smoke Detection Based on Faster R-CNN using Synthetic Smoke Images. *Procedia Engineering*, 211:441–446, 2018. 2

[48] Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in neural information processing systems*, 31, 2018. 6

[49] Kailai Zhou, Yibo Wang, Tao Lv, Yunqian Li, Linsen Chen, Qiu Shen, and Xun Cao. Explore spatio-temporal aggregation for insubstantial object detection: Benchmark dataset and baseline. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3104–3115, 2022. 2

A. Additional Results

We further breakdown the performance contributions from each synthetic data source. Specifically, Table A.1 displays the analysis for data generated via Unreal Engine, and Table A.2 outlines the performance metrics for NVIDIA Omniverse-generated data.

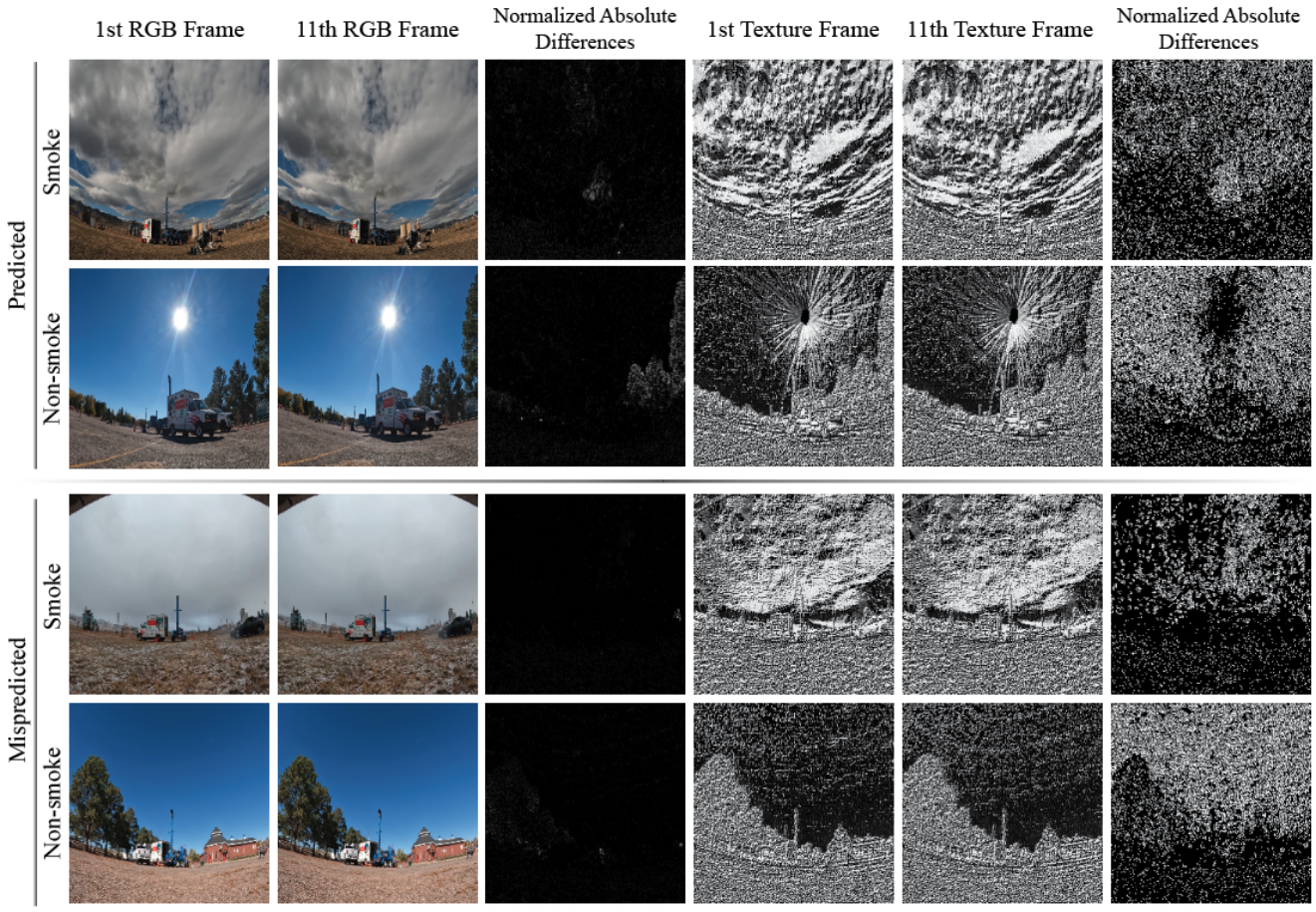
Additional	Real-World	Unreal	Accuracy	Conf (↓)
0%	560	0	89.99 %	37.72 %
5%	560	28	81.10 %	38.91 %
10%	560	56	85.49 %	**34.00 %**
15%	560	84	74.13 %	37.86 %
20%	560	112	84.14 %	34.79 %
25%	560	140	81.21 %	38.40 %
30%	560	168	**92.58 %**	36.53 %
35%	560	196	71.43 %	40.45 %
40%	560	224	78.74 %	39.59 %
45%	560	252	77.73 %	38.29 %
50%	560	280	71.31 %	40.80 %

Appendix A.1. This experiment aimed to determine the optimal amount of Unreal Engine-generated synthetic data for smoke detection training without compromising accuracy. Beginning with a dataset devoid of synthetic data, we progressively increased the synthetic portion by 5% increments, continuing until synthetic data constituted half of the original training set. This approach allowed us to identify the threshold at which additional synthetic data begins to adversely affect model performance.

Additional	Real-World	Omniverse	Accuracy	Conf (↓)
0%	560	0	89.99 %	37.72 %
5%	560	28	87.96 %	37.02 %
10%	560	56	79.75 %	39.24 %
15%	560	84	79.08 %	43.71 %
20%	560	112	92.24 %	37.16 %
25%	560	140	90.10 %	36.87 %
30%	560	168	**95.05 %**	**36.13 %**
35%	560	196	76.94 %	43.29 %
40%	560	224	79.30 %	43.10 %
45%	560	252	68.29 %	44.44 %
50%	560	280	87.29 %	41.23 %

Appendix A.2. This table displays the accuracy differences observed with varying amounts of NVIDIA Omniverse synthetic data, paralleling the experiment detailed in Table A.1. It highlights how incremental additions of Omniverse data influence model performance, mirroring the methodology applied to Unreal Engine data for comparative analysis.

Analysis of our method, SemiS: predictions and mispredictions are visualized in Figure A.I. SemiS is able to distinguish between smoke and background object movements such as trees, when the difference between frames is higher (often in higher opacity smoke). However, with limited movement as seen in lower opacity smoke, it becomes difficult to extract the smoke features from the background, which can be seen in the second and fourth rows of Figure A.I).



Appendix A.I. Visual analysis: Our examination consists of four distinct rows: the first two showcases instances of accurately predicted video clips, while the last two highlights cases of erroneous predictions. In the first row, the both RGB and texture based Normalized Absolute Difference (NAD) frames successfully identify smoke movement. Conversely, the second row induces the shape of RGB-based NAD frame contributes to recognizing non-smoke features which are movements of trees. Misinterpretations occur in the last two rows; for instance, the third row, the weather condition leads the model to falsely identify smoke. Similarly, the fourth example demonstrates how tree movements introduce confusion, being misinterpreted as smoke movement by the model. This visual analysis underscores the challenges in smoke detection, particularly in distinguishing between smoke movement and other dynamic elements within a scene.