ThermalSynth: A Novel Approach for Generating Synthetic Thermal Human Scenarios

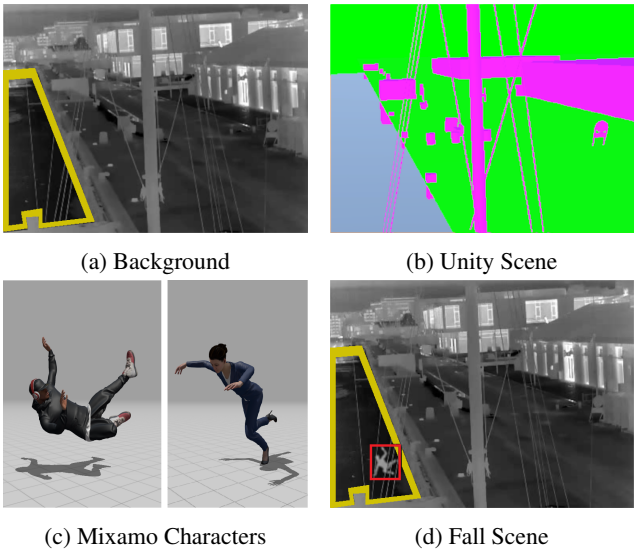
Neelu Madan*[1] Mia Sandra Nicole Siemon*[1,2] Magnus Kaufmann Gjerde[1]
Bastian Starup Petersson[1] Arijus Grotuzas[1] Malthe Aaholm Esbensen[1]
Ivan Adriyanov Nikolov[1] Mark Philip Philipsen[1] Kamal Nasrollahi[1,2]
Thomas B. Moeslund[1]
[1]Visual Analysis and Perception Lab, Aalborg University, Denmark, [2]Milestone Systems, Denmark
{nema,msns,mpph,iani,kn,tbm}@create.aau.dk, {bpeter18,mgjerd18,agroutu18,mesben18}@student.aau.dk

Abstract

In this paper, we propose ThermalSynth, a novel approach for creating synthetic thermal images by mixing 3D characters generated using the Unity game engine with real thermal backgrounds. We use a shader based on the Stefan-Boltzmann law [18] to approximate the appearance in the thermal domain of the synthetic characters. Additionally, we provide a post-processing pipeline to better blend the high-fidelity synthetic data with the lower-resolution real thermal surveillance one. The proposed approach is used to create a dataset for people falling into water near a harbor front. Diverse scenarios of such falls are generated with an ample amount of data to enable the use of deep learning algorithms. To demonstrate the effectiveness of the generated data, we train two standard deep neural networks (AlexNet and ResNet-18) on our synthetic thermal dataset using a supervised learning approach. We test our system on small datasets containing real video footage of actual falls. We observe that training these simple classification networks yields an accuracy of 98.70% *at a sensitivity of* 100% *on the real-world voluntary fall dataset. The code for ThermalSynth and the dataset is publically available at* https://github.com/NeeluMadan/Thermal-Synth.

1. Introduction

Video surveillance systems mostly employ stationary RGB cameras as they are cost-effective whilst yielding good discrimination among objects. They are, however, not immune to various quality-decreasing conditions such as occlusions, changing weather, and low illumination. Thermal cameras, on the other hand, measure the difference in heat signatures returning high-contrast images. Consequently, they are a reliable choice under diverse weather conditions while preserving privacy [20] at the same time, which also helps comply with the General Data Protection Regulation (GDPR) [32, 2]. As a result, intelligent video surveillance systems have started using a combination of RGB and thermal cameras, recently.



(a) Background (b) Unity Scene

(c) Mixamo Characters (d) Fall Scene

Figure 1: ***ThermalSynth* - Synthetic data generation proposal** 1a) An example background image from the Long-term Thermal Drift (LTD) dataset [30], 1b) the same scene synthesized in Unity, 1c) example fall animations of Mixamo characters, and 1d) a synthetically generated falling person merged with 1a, with a yellow enclosure highlighting the Region of Interest (RoI).

In the RGB domain, there exist multiple instances of synthetic datasets [12, 13, 35, 3, 6] which were generated using a virtual environment. However, only a very limited amount of research focuses on generating synthetic datasets in the thermal domain. In this paper, we propose a pipeline to generate synthetic thermal datasets. Generating such data gives the possibility to address different scenarios with very

*Equal contribution.

few instances in real life that are hard to replicate through physical testing setups. One such scenario is the depiction of humans falling into bodies of water in an outdoor environment. As these instances occur rarely and often under life-threatening conditions, it is very difficult to obtain data of such scenarios. Only a few datasets for this problem currently exist containing voluntary falls [7] and the use of dummies [29]. The foremost drawback of generating such data via intentional jumps [7] is that it might get intermittently dangerous due to unpredictable circumstances. The use of a dummy [29] on the other hand, requires a time-consuming preparation process and also has only a limited degree of freedom to add variations. As a consequence, it is difficult to obtain a large and varied dataset using either method, rendering deep neural networks and/or machine learning models inapplicable to this problem domain. We therefore propose an approach to generate a synthetic thermal datasets, and apply it to the concrete use-case of human fall detection at harbor fronts. It is important to mention, that given the modular nature of our proposed data generation approach, it could be easily modified to create synthetic thermal datasets for almost any application domain, with the only requirement of providing initial real-life images of the environment that can be used as background.

For demonstration purposes, we trained two classic Convolutional Neural Networks, AlexNet [24] and ResNet-18 [15], using supervised learning on our proposed synthetic fall dataset and tested the model on real [7] and semi-real (dummy) datasets [29]. The best model results in 98.70% accuracy and a sensitivity rate of 100% on real data constituting of intentional falls, indicating that our synthetic dataset contains a good approximation of the distribution of real-world human fall scenarios. The contribution of this work to video surveillance in the thermal domain is twofold:

1. A synthetic data generation pipeline in which uniquely generated foreground objects are combined with real background footage
2. The application of our proposed pipeline to generate a synthetic fall dataset for people falling into water

The rest of the paper is structured as follows: The next section provides an overview of existing research in synthetic datasets and the fall detection domain. Section 3 describes the data generation process and the standard classification models used in this research. Our experimental setup is mentioned in Section 4, which is then followed by results and discussion in Section 5. We finally conclude our research with its possible future directions in Section 6.

2. Related Work

Synthetic Datasets in Thermal Domain. The enforcement of the new GDPR [32, 2] law by the European Union has turned the acquisition of large-scale personal visual data into a challenging task. Under these circumstances, and given the fact that deep learning networks are very data-hungry, we have experienced a paradigm shift towards the generation of synthetic datasets for sensitive data. In the thermal spectrum domain, there exist two methods for generating synthetic data: (1) Mapping straight from the RGB domain, and (2) using virtual environment engines.

In the domain of deep learning, the use of approach (1) is commonly achieved by means of Generative Adversarial Networks (GANs). This takes either place in a supervised (paired data) or an unsupervised (unpaired data) setting [22, 39, 17, 40, 21]. Research [17, 22] shows, however, that the usage of supervised GANs [17, 21] delivers better results than its unsupervised counterpart [40] because of the presence of RGB-thermal image pairs. Since we only have thermal video footage at our disposal for the purpose of this project though, the usage of supervised GANs is not applicable.

With respect to approach (2), only a few instances generating thermal datasets synthetically using virtual environments [33, 6, 8] exist to date, even though there are numerous such environments for generating synthetic data in the visual spectrum: CARLA used for Advance Driver Assistance System (ADAS) [11], VIVID [25] for indoor navigation, Gazebo [23] for simulating multi-robot, and Habitat 2.0 [36] for home assistants. Apart from that, other research [33, 6, 8] uses game engines such as Unity [14] and Unreal [28] to generate photo-realistic synthetic data. Pramerdorfer *et al.* [33], for instance, generate synthetic depth and thermal images showing human behavior captured in indoor environments using Blender [10] while Blythman *et al.* [6] generate synthetic thermal human heads placed in cars using Zephyr [1]. Bongini *et al.* [8] use Unity [14] to generate synthetic thermal videos by combining 3D foreground objects with the real background images in autonomous driving scenarios. Following this trend of synthetic data generation in the thermal spectrum domain, we consequently propose to generate a synthetic thermal dataset for human fall detection using Unity [14]. Similarly to the work in [8], the proposed approach generates our dataset images by blending synthetic foreground objects with real background scenes. These are obtained from the Long-term Thermal Drift (LTD) dataset [30].

Fall Detection. Fall incidents in outdoor scenarios, like people falling into water [7, 27, 29] or on the street [31, 26], have only received little attention in recent years. There exists qualitative research [31, 26] addressing cases of falls in outdoor scenarios such as sidewalks, streets, and garages, but it still remains difficult to capture such moments using surveillance cameras. One reason for this is that there is an insufficient amount of samples available, especially when it

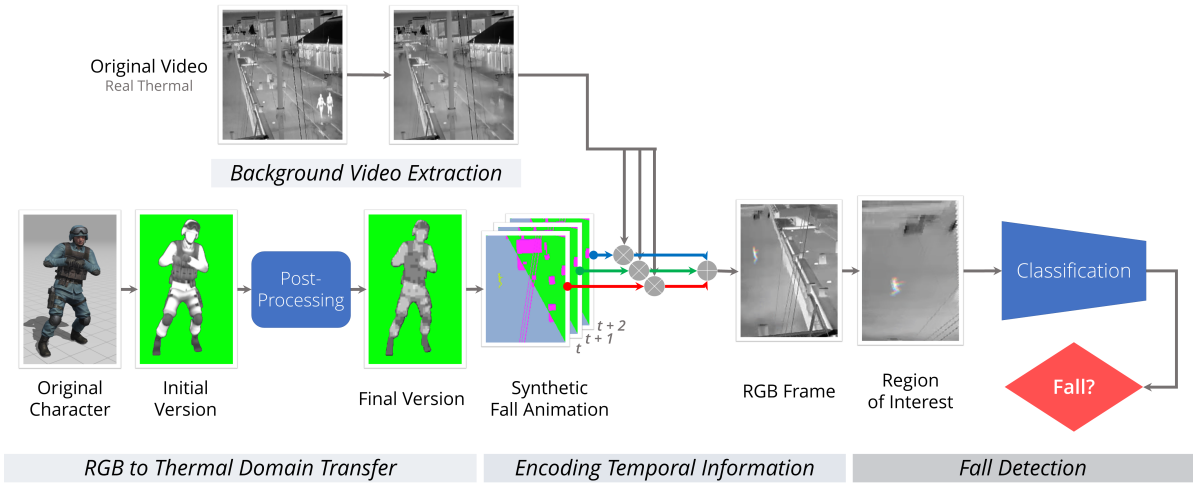


Figure 2: **Synthetic Data Generation and Fall Detection Pipeline**: Our synthetic data is limited to three frames at $t, t+1, t+2$. After merging each of them individually with the extracted background frames, they are stacked together in order to encode temporal information.

comes to people falling into water. One of the first works to address the problem was done by Bonderup *et al.* [7]. Here, a pipeline consisting of person detection, person tracking, fall prediction, and fall detection is proposed in the thermal domain. Additionally, Bonderup *et al.* [7] generated a thermal fall dataset asking people to perform intentional jumps into water. A few years later, Nikolov *et al.* [29] proposed a semi-real dataset for human fall detection simulating fall scenarios in the thermal domain using a dummy. The authors make use of calculated optical flow maps around a specific area of interest in order to detect falls.

In this research, we address the human fall detection problem using a supervised classification approach. The related works described so far suffer from significant limitations as the datasets are captured in controlled conditions and hence contain only very limited variations. Our intention is to fill this gap and present a synthetic dataset that can serve as an extensive source of diverse human fall scenarios.

3. *ThermalSynth*: Proposed Method

The main objective of this research is to present an approach to create a pipeline for synthetically generating thermal datasets using a 3D environment. We also demonstrate its application for creating synthetic human falls into water regions at harbor fronts. This section contains the building blocks of *ThermalSynth*, our thermal synthetic data generation process followed by its application for human fall detection. The entire pipeline for our proposed method is shown in Figure 2.

Images are generated by merging real thermal backgrounds with synthetic people generated in Unity [14]. The real videos are obtained from the LTD dataset [30], which is very diverse in terms of different weather conditions as it encapsulates video data for 8 months (January-August) from a single camera view. Such single-scene recordings are most common in video surveillance setups. The main elements of our synthetic data generation pipeline are as follows: (1) Background Extraction, (2) Foreground Generation, (3) Thermal Shader and (4) Post-Processing. Each of these steps is explained in detail in the following subsections.

3.1. Background Extraction

All background scenes are extracted from the LTD dataset [30]. It contains 298 hours of single-scene videos, with a resolution of 384×288, and captures a single camera view. Each video is 2 minutes long, and all are uniformly spaced out throughout 24 hours of the day. In order to create the backgrounds, a temporal median filter is applied to the dataset. This is done as a two-step process: At first, the videos are coarsely sampled at 1 frame per second (FPS) in order to manage the computational complexity. Secondly, the median value for each matching pixel across all frames for each 2-minute video is computed. The temporal median filter is solely applied to the harbor area, in order to retain other movements caused by the water and other moving objects close to the camera due to wind like wires, ropes, and masts. This is done by manually creating a mask of the parts of the scene where the filter should be applied. To limit the complexity of our dataset, we uniformly sample 69 hours of video from the LTD dataset to extract the backgrounds. The sequence of those extracted background frames is kept as given in the original video in order to retain the fluency of non-object motions such as clouds, waves, and wires. We

stacked three consecutive frames together, which are later blended with the synthetic foreground as also shown in Figure 2.

3.2. Foreground Generation

For the generation of our synthetic thermal foreground videos the game engine Unity [14] was chosen. Depicting different scenarios of people walking and falling into water at the harbor, these synthetic videos are later merged with the created background instances. Generating synthetic videos is also a two-part process. First, the 3D models of people are selected together with a number of animations for walking, running, jumping, falling, etc. Mixamo [5] is used to select these 3D models and animations, as it is a free-to-use library of human-looking characters, together with motion-captured animations. Examples of some of the character models used are shown in Figure 1c. We choose a total set of $79,998$ unique foreground video sequences depicting jumps and falls, each comprising three consecutive frames.

As the next step, the parameters of the chosen thermal camera for recording the LTD dataset (see [16] for details) are transferred to the Unity camera. To do so, the Universal Render Pipeline in Unity is used together with the physical camera settings. The parameters are given in Table 1. A synthetic scene is then modeled with primitive objects in Unity in places where real objects can obscure the view of the camera of people walking on the street. Real-world objects deemed necessary to be modeled are selected heuristically after observing videos from the LTD dataset. The Unity camera's position and orientation are then set to best match the position of the real-world one. The resulting synthetic scene in Unity can be seen in Figure 1b, where the modeled obscuring objects are shown in pink, the background from the real images in green, and the waterfront in gray, together with a synthetic person falling. The real background is visualized on a rendered texture behind the synthetic scene. We then use the Perception package [37] provided by Unity [14], to generate a large number of combinations of 3D meshes, animations, and backgrounds. These masks of rendered people are used in the post-processing step to better blend the synthetic foreground with the background. The next step is to transform the generated synthetic pedestrian footage from RGB to thermal domain using a custom shader. An overview of the steps for creating the shader is given in the next section.

3.3. Thermal Shader

Once the synthetic pedestrians (foreground) are generated, together with their segmentation masks, their RGB representation needs to be transformed into a thermal one. The thermal shader used in our approach is inspired by Kane *et al.* [18]. It uses the Stefan-Boltzmann law to compute the black body radiation (j^*) of an object given by Equation 1, where T is the absolute temperature of the object in Kelvins, ϵ_m represents the thermal emissivity of the material, and σ is the Stefan-Boltzmann constant equalling to $5.6704 \times 10^{-8} \frac{W}{m^2 K^4}$.

Thermal Camera				
Zoom	Resolution	Frame Rate	Lens	FOV
Fixed	384×288	25/30 FPS	25mm	21.7°
Emissivity Values				
Human Skin	Cotton	Asphalt	Water	Snow
0.95	0.95	0.95	0.93	0.90

Table 1: Parameters of thermal camera used in the LTD dataset [30], Emissivity coefficients of materials found in our scenes, taken from [18]

$$j^* = \epsilon_m \, \sigma \, T^4 \tag{1}$$

To translate this in the context of a shader we first find the emissivity values of some of the materials that would be part of the generated pedestrians. In our case, we simplify this to human skin and clothes. Values for both of these are given in the article written by Kane *et al.* [18]. We have listed these together with the emissivity values of other materials for comparison in Table 1. Next, for the absolute temperature in Kelvin, we select an average value of 300.5 for simplifying the generation process.

Once we have these initial values, we follow the approach described by Kane *et al.*. The albedo texture color of each pixel is transformed to luminance (L) using Equation 2, where R, G, and B, are the red, green, and blue color channels, respectively.

$$L = 0.2126 \cdot R + 0.7152 \cdot G + 0.0722 \cdot B \tag{2}$$

As presented in the article [18], the calculated luminance is then used to approximate color emissivity (ϵ_c) using Equation 3, by using the average color emissivity of a white color surface of 0.84 and the percent difference between white and black object emissivity of 0.15.

$$\epsilon_c = (1 - L) \cdot 0.15 + 0.84 \tag{3}$$

The material and color emissivities are then blended using a blend factor of 0.31. The final blended value is further used in the Stefan-Boltzmann equation shown in 1. Finally, the calculated thermal radiation value for each pixel is mapped to an intensity range between $[0, 1]$ so it can be displayed by Unity. A gain (G) and level (L) control are made available for the final pixel value p using Equation 4, so that manual adjustment can be possible. For the purpose of this paper, these values were manually set to $G = 0.05$ and $L = 20$ as these provided the blending with the extracted backgrounds (this effect is visualized in Figure 5).

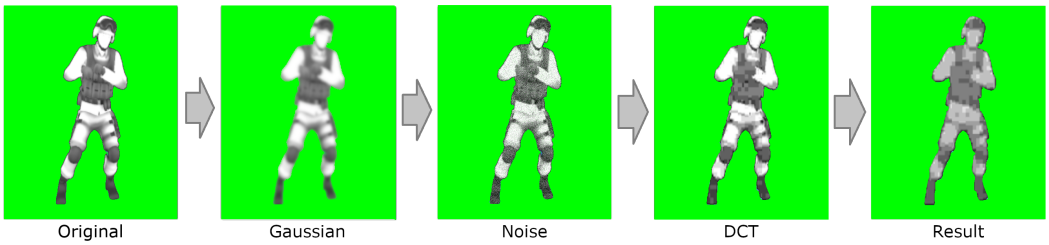


Figure 3: **Post-processing Pipeline**: Starting from left the original image generated through applying a thermal shader on a character taken from Mixamo [5], Gaussian filter with kernel size 3x3, random noise applied, and finally DCT compression artifacts added based on Kane *et. al.* [18]. The final result is the sum of all single instances.

$$p = (j^* \cdot G) + L \tag{4}$$

Once the foreground pedestrian footage is transformed into thermal, the next step is to blend it with the extracted real backgrounds. To do this, a number of post-processing steps are implemented which are discussed in the next section.

3.4. Post-Processing

In practice, thermal camera sensors are susceptible to capturing noise from the environment [18], which together with compression artifacts from storing videos may degrade the visual quality of captured footage. Bhatia *et al.* [4] propose a post-processing stack of image effects for simulating infra-red sensors and their specific characteristics. We choose three prominent effects based on Kane *et. al.* [18] from those - blurring, random noise, and compression artifacts. Together with the already implemented part of the thermal shader with gain and level processing, these effects help blending the real and synthetic parts of the image into one coherent picture.

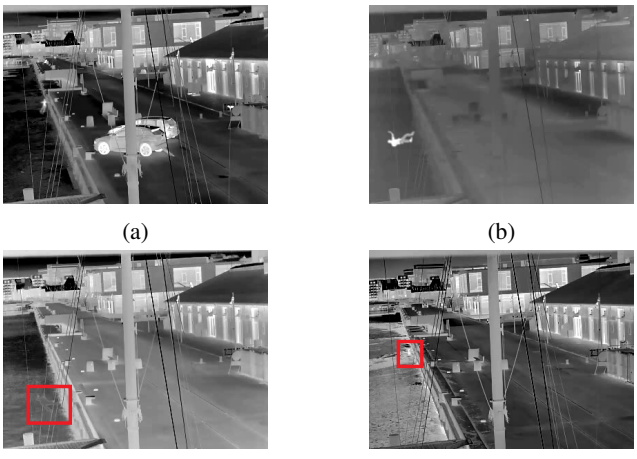


Figure 4: **Results of synthetic thermal frame generation pipeline**: Successful frames are shown in 4a and 4b where foreground objects, i.e., humans can be distinguished from background, and unsuccessful ones in 4c and 4d where foreground can't be discriminated from background.

1. **Gaussian blur:** Rendered objects in Mixamo [5] contain sharp or jagged edges in comparison to real objects. Blurring artifacts are therefore introduced to the area around the synthetic humans using a (3×3) Gaussian kernel.

2. **Random noise:** The degree of sensitivity of image sensors used to capture real-life footage by means of thermal cameras often introduces random noise. Mitigation of this effect is achieved through the application of random noise to our rendered figures by generating uniformly distributed random numbers, and through linear interpolation between this value and the rendered foreground one.

3. **Compression artifacts:** Mosquito Noise and Block Artifacts appear to be the most common side-effects caused by flawed compression algorithms that are implemented in thermal cameras which make use of block-based Discrete Cosine Transform (DCT) [19]. In order to account for this imperfection, additional encoding and decoding of the foreground character become necessary. Hence, the image is firstly converted into JPEG format, which performs DCT compression by default, setting the quality value to 5 (on a scale from 0 to 100), and secondly it is decoded to retrieve its compressed form.

4. **Compositing:** To blend the post-processed synthetic person into the real background image, screen compositing mode is used. Being a compositing technique that preserves the edges of the foreground mask, for images that come in 8-bit integer precision the composited image can be calculated using Equation 5:

$$C = 255 - \frac{(255 - F) \times (255 - B)}{255}, \tag{5}$$

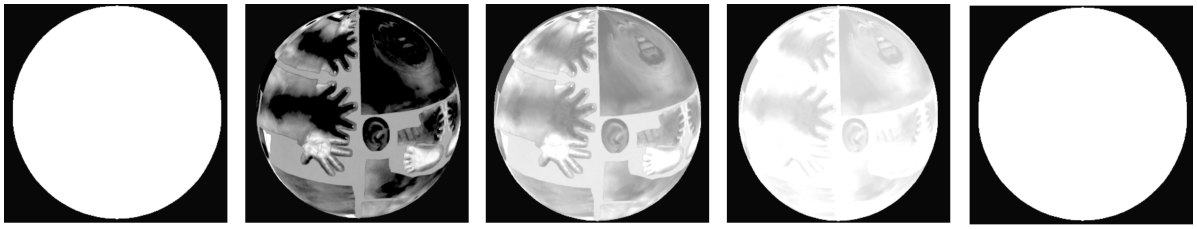


Figure 5: Illustrating the effect of different Gain (G) and Level (L) ranges on the appearance of the foreground. G=0.05 and L=20 generate the most realistic thermal appearance.

with C equalling the composited image, B to the real background, and F to the synthetic foreground. Example final results from the synthetic thermal image generation including post-processing that is visualized in Figure 3 can be seen in Figure 4. Success and failure are determined based on how well the synthetic foreground elements blend visually into the real background scenery. In comparison, decisive attributes encompass the level of visibility of the foreground compared to the background, and differentiation between different body parts, such as extremities, torso, and head, for example.

3.5. ThermalSynth for Human Fall Detection

We use the pipeline explained earlier in this section to generate a synthetic thermal dataset of humans falling into water. This dataset is kept very simplistic by restricting it to a limited number of human fall animations, as it is primarily serving demonstration purposes. We further use this dataset to train two classic Convolutional Neural Networks, AlexNet [24] and ResNet-18 [15], using supervised learning in order to perform human fall detection. The fall detection problem is modeled here as a binary classification one with our two classes being defined as *fall* and *no fall*, respectively. As mentioned earlier in this paper, *ThermalSynth* is not limited to this particular application area. Thanks to the generic design, it is applicable to a wide range of surveillance scenarios that take place in the thermal domain.
Besides privacy, another major advantage of using synthetic datasets for training machine learning models is that annotations can be generated automatically as part of the data creation process. These automatic labels are highly accurate in comparison to manually annotated ones. The absence of such noise during the training process of machine learning models results in more robust prediction performances. Based on the achieved results described in the upcoming section it can be observed that models trained on synthetic data perform almost perfectly when tested on real-world data.

4. Experiments

4.1. Datasets

We evaluate the performance of our models on two datasets with real thermal surveillance footage: *Intentional Fall Dataset* (real) [7] and *Dummy Dataset* (semi-real) [29]. Both of these datasets contain only a very limited number of samples and thus would result in severe overfitting when used to train a neural network. We therefore use our proposed synthetic dataset for training the models instead, and use the other two datasets for test purposes only.

Intentional Fall Dataset The Intentional Fall dataset was collected by Bonderup *et al.* [7]. It was recorded in the thermal domain and visualizes scenarios of a variety of jumps into water performed by volunteers. Out of the manually annotated thermal video footage (captured during Spring 2016) a subset was chosen that depicts the same harbor scene as the LTD dataset [30]. On concatenation of three consecutive frames into a single batch as an RGB image, the test set ends up consisting of 77 samples in total, out of which 18 are denoted as *fall*, and 59 as *no fall*.

Dummy Dataset The Dummy dataset, interchangeably also called mannequin or rubber doll dataset by its authors [29], was introduced in order to show that an air-filled rubber doll presents a sufficient representation of humans when generating thermal video footage that targets the detection of human falls into water. The authors of [29] generated a thermal video dataset (captured during the months of September - October 2021) that depicts artificially arranged emergencies at a harbor front. For the sake of this work, the videos are also parsed in a way that allows for the compression of three consecutive frames into a single batch in form of an RGB image. This leads to a Dummy test set which consists of $1,626$ frames out of which 580 were categorized as *fall*, and $1,046$ as *no fall*.

4.2. Evaluation Metrics

All our models are evaluated in terms of sensitivity, specificity, and accuracy. *Accuracy* describes correctly classified *fall*s and *no fall*s over all cases. *Sensitivity* describes

correctly detected *fall*s over all *fall* cases. *Specificity* on the other hand concerns correctly classified *no fall*s over all *no fall*. For the purpose of solving fall detection tasks, those systems with high sensitivity are preferable as it is crucial to detect as many *fall* cases as possible even at the expense of falsely classifying few *no fall* cases. Not achieving this, i.e., missing *fall*s, may possibly result in a person drowning.

4.3. Implementation Details

Since fall detection in water regions can be considered as a special scenario of binary classification, we employ two standard classification networks, i.e., AlexNet [24] and ResNet-18 [15], which are trained solely on the proposed synthetic thermal data. Due to the simplicity of the problem, we refrain from proceeding with more complex architectures at this leads to overfitting, and a significant decrease of the system's performance.

Training. Before launching the training of the networks, the generated synthetic images are pre-processed by cropping only the water area. Since this results in a trapezoidal image, it is further warped using *warpPerspective* function from OpenCV to convert it into a rectangular shape. Afterwards, three consecutive frames are concatenated to generate a single tensor of size $185 \times 115 \times 3$. An equal number of *fall* and *no fall* images (69,000) are used for training the baseline models. For the validation of our classification networks, 34,596 images (with 17,298 *fall* and 17,298 *no fall*) are sampled, and the model with the best validation accuracy is saved for further evaluation. The used architectures converged at epoch 20 using a batch size of 32. Stochastic Gradient Descent (SGD) [34] with a learning rate of 10^{-3} is used for optimizing the neural network. PyTorch implementations of our baseline models were chosen and trained using a single NVIDIA RTX 2080 Ti series GPU.

Testing. The performance of our models is evaluated on three test sets coming from three different sources [7, 29] including ours, described in Subsection 4.1. Both datasets for people fall detection, i.e., Intentional Fall [7] and Dummy [29], do not contain any annotations for fall classification. These were created by means of manual frame-level annotations categorizing them as *fall* or *no fall*, respectively.

5. Results and Discussion

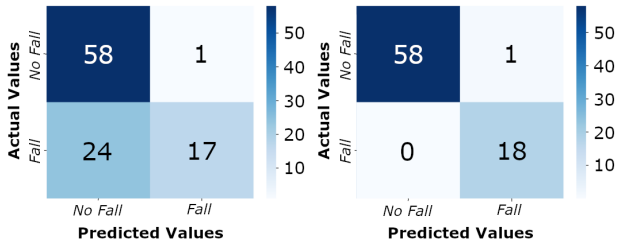
The evaluation of our fall detection approach leads to the results recorded in Tables 2 and 3. These numbers prove that our synthetically generated data provide a good approximation of the distribution of fall scenarios in the existing datasets and hence constitute a justified choice for training deep neural networks. Significant growth in classification accuracy is observed when we use the synthetic thermal dataset for training and testing. We are, however, unable to reach a similar performance when testing on the Dummy dataset, where both network models reached a 75% classification accuracy. The assumption is that the dummy constitutes a very limited representation of a human when falling/thrown into water. This is additionally supported by the results which are given in Table 3. It can be seen that both networks achieve high specificity on both the Intentional Fall and Dummy datasets. This, however, comes

Network	Accuracy (↑)		
	Synthetic Dataset	Intentional Fall Dataset	Dummy Dataset
AlexNet	**99.52**	**98.70**	75.00
ResNet-18	98.75	89.61	75.00

Table 2: Comparing accuracy in % of the two baseline classification networks. The networks are trained on our proposed Synthetic dataset and tested on a different subset of the Synthetic dataset, together with the full Intentional Fall, and Dummy datasets.

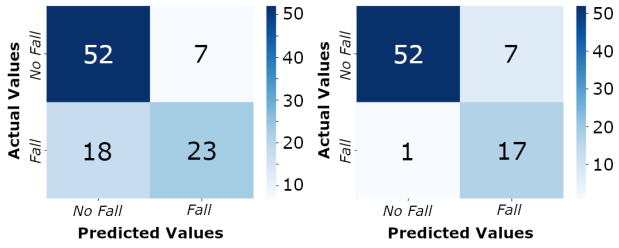
Network	Intentional Fall		Dummy	
	Sens.(↑)	Spec.(↑)	Sens.(↑)	Spec.(↑)
AlexNet	**100.00**	98.00	41.00	**98.00**
ResNet-18	94.00	88.00	**57.00**	88.00

Table 3: Comparing Sensitivity (Sens.) and Specificity (Spec.) in % of the two baseline classification networks trained on our proposed synthetic dataset and tested on Intentional Fall and Dummy dataset, respectively. Higher values indicate better systems.



(a) AlexNet (D)

(b) AlexNet (IF)



(c) ResNet-18 (D)

(d) ResNet-18 (IF)

Figure 6: **Confusion Matrices** for AlexNet and ResNet-18 during tests on the Dummy (D) and the Intentional Fall (IF) Datasets.

with the caveat that both datasets can be described in a binary way and consist of only frames of either people and dummies falling or not falling into water. The datasets do not contain the third category of objects being in the water without being classified as a person falling into water. In real-world use cases, this category has a very strong possibility of emerging - for example birds landing in the water, boats passing in front of the camera, people throwing objects into water, etc. From this perspective, the high specificity and accuracy of our results should be viewed as *ideal* cases. Automated emergency detection systems need to be robust against false positive detections, as a high number of these can result in diverting resources and drowning out real ones in noise.

Figure 6 illustrates the confusion matrices determining correct and incorrect classifications in case of fall and non-fall events for both Dummy and Intentional Fall datasets. Looking at these values proves that both our models have great difficulties when having to correctly classify actual *fall* images as *fall* when tested on the Dummy dataset. In other words, out of 41 falls given in this dataset, AlexNet is capable of correctly classifying only 17 of them as *fall* whilst ResNet-18 is performing slightly better with a total of 23 *fall*s. Comparing qualitative classification performance on the Intentional Fall dataset, however, indicates that AlexNet would be the ideal candidate leaving no falls undetected.

Last but not least, we would like to address the choices made with respect to emissivity values during this research. In contrast to the source of values reported in Table 1, i.e., [18], previous works [38, 9] have shown that these values can be taken from a great range of possibilities: Cotton, wool, and PET, for instance, lay between 0.7 and 0.83, as shown by [38]. In this work, we choose a consistent value of 0.972 to make the proposed synthetic dataset appear visually closest to the thermal domain. Additionally, we apply this emissivity value uniformly to the entire foreground image for simplification purposes.

In addition to the implicit judging of the quality of our dataset by training deep neural networks on it and testing on real-world surveillance footage, we also verify the quality-level of our data implicitly via visual inspection. Some examples of synthetic humans and real humans as foreground objects are shown in Figure 7. The synthetic humans in Figure 7 (left) contain the same overall texture from head to toe, whereas real humans shown in Figure 7 (right) show variations in appearance based on the types of clothes and additional accessories, e.g., bags. We plan to extend this work by applying a part-based thermal shader, where different emissivity values to different parts of the foreground are applied. The example of different parts when we consider humans as our foreground objects are head, torso, and legs *etc*.

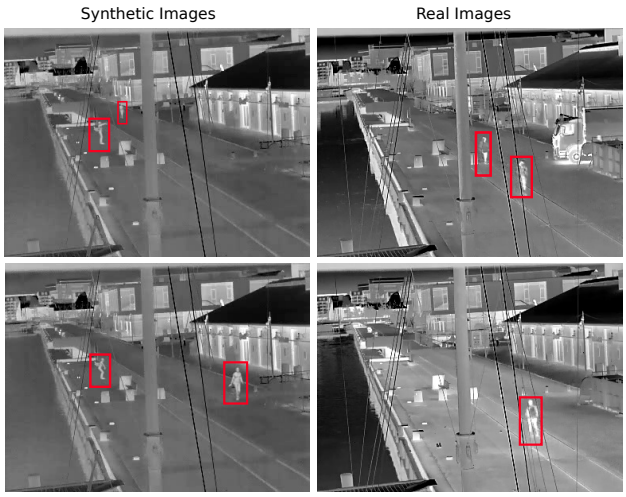


Figure 7: **Real vs. Synthetic** Left: Final synthetic images; Right: Real frames taken from the LTD dataset [30]

.

6. Conclusion and Future Work

In this paper we introduced *ThermalSynth*, a pipeline for creating synthetic thermal images showcasing one possible application of people falling into water. For generating the foregrounds we use Unity together with rigged, animated 3D models and a custom thermal shader based on the black body radiation equations along with the Stefan-Boltzmann law. To mimic CCTV camera footage, we implemented a four-stage post-processing pipeline which introduces additional image distortion and finally blends the foreground and background parts. We use this pipeline to create a synthetic thermal dataset of people falling into water and further train two standard classification models, AlexNet and ResNet-18, to detect fall cases. We test the models on a combination of synthetic 3D model falls, real-person falls, and simulated falls using a dummy. We show that the standard models achieve very good results in the given context proving the usability and potential of *ThermalSynth* for creating rarely observed emergency scenarios and enriching existing real thermal datasets with synthetic data.

A possible negative societal impact of this dataset is that it reveals different jumping and falling behavior patterns of humans. At the same time, however, this study could save many human lives. We plan to extend this research by introducing additional 3D models to the generation pipeline like vehicles, boats, birds, moving parts of the foreground, etc. in order to possibly use the dataset for multi-class classification tasks and more robust emergency detection in real-life production scenarios.

7. Acknowledgements

This work was supported by the Milestone Research Programme at Aalborg University (MRPA).

References

[1] 3DF ZEPHYR - photogrammetry software - 3D models from photos, https://www.3dflow.net/3df-zephyr-photogrammetry-software, (Accessed on 03/01/2022)

[2] General Data Protection Regulation (GDPR) – Official Legal Text, (Accessed on 03/02/2022)

[3] Acsintoae, A., Florescu, A., Georgescu, M.I., Mare, T., Sumedrea, P., Ionescu, R.T., Khan, F.S., Shah, M.: Ubnormal: New benchmark for supervised open-set video anomaly detection (2021)

[4] Bhatia, S.K., Lacy, G.M.: Infra-red sensor simulation. In: I/ITSEC. pp. 1–8 (1999)

[5] Blackman, S.: Rigging with Mixamo, pp. 565–573. Apress, Berkeley, CA (2014)

[6] Blythman, R., Elrasad, A., O'Connell, E., Kielty, P., O'Byrne, M., Moustafa, M., Ryan, C., Lemley, J.: Synthetic thermal image generation for human-machine interaction in vehicles. In: 2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX). pp. 1–6 (2020)

[7] Bonderup, S., Olsson, J., Bonderup, M., Moeslund, T.B.: Preventing drowning accidents using thermal cameras. In: Advances in Visual Computing. pp. 111–122. Springer International Publishing (2016)

[8] Bongini, F., Berlincioni, L., Bertini, M., Del Bimbo, A.: Partially Fake It Till You Make It: Mixing Real and Fake Thermal Images for Improved Object Detection, p. 5482–5490 (2021)

[9] Charlton, M., Stanley, S.A., Whitman, Z., Wenn, V., Coats, T.J., Sims, M., Thompson, J.P.: The effect of constitutive pigmentation on the measured emissivity of human skin. PLoS ONE **15** (2020)

[10] Community, B.O.: Blender - a 3D modelling and rendering package. Blender Foundation, Stichting Blender Foundation, Amsterdam (2018), http://www.blender.org

[11] Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V.: CARLA: An open urban driving simulator. In: Levine, S., Vanhoucke, V., Goldberg, K. (eds.) Proceedings of the 1st Annual Conference on Robot Learning. Proceedings of Machine Learning Research, vol. 78, pp. 1–16. PMLR (2017)

[12] Fabbri, M., Brasó, G., Maugeri, G., Cetintas, O., Gasparini, R., Ošep, A., Calderara, S., Leal-Taixé, L., Cucchiara, R.: Motsynth: How can synthetic data help pedestrian detection and tracking? In: Proceedings of ICCV. pp. 10849–10859 (2021)

[13] Fabbri, M., Lanzi, F., Calderara, S., Palazzi, A., Vezzani, R., Cucchiara, R.: Learning to detect and track visible and occluded body joints in a virtual world. In: Proceedings of ECCV (2018)

[14] Haas, J.K.: A history of the unity game engine (2014)

[15] He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. In: Proceedings of CVPR. pp. 770–778 (2016)

[16] Hikvision: Ds-2td2235d-25/50. https://us.hikvision.com/en/products/more-products/discontinued-products/thermal-camera/thermal-network-bullet-camera-ds (2015), accessed: 2021-09-27

[17] Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. Proceeding of CVPR (2017)

[18] Kane, F.: Simulation of night-vision and infrared sensors. In: Lengyel, E. (ed.) Game Engine Gems 2, pp. 45–54. A K Peters (2011)

[19] Ken, C., Gent, P.: Image compression and the discrete cosine transform. College of the Redwoods, Tech. Rep (1998)

[20] Kieu, M., Bagdanov, A.D., Bertini, M., Del Bimbo, A.: Domain Adaptation for Privacy-Preserving Pedestrian Detection in Thermal Imagery, Lecture Notes in Computer Science, vol. 11752, p. 203–213. Springer International Publishing (2019)

[21] Kieu, M., Berlincioni, L., Galteri, L., Bertini, M., Bagdanov, A.D., Bimbo, A.: Robust pedestrian detection in thermal imagery using synthesized images. 2020 25th International Conference on Pattern Recognition (ICPR) pp. 8804–8811 (2021)

[22] Kniaz, V.V., Knyaz, V.A., Hladůvka, J., Kropatsch, W.G., Mizginov, V.: ThermalGAN: Multimodal Color-to-Thermal Image Translation for Person Re-identification in Multispectral Dataset, vol. 11134, p. 606–624. Springer International Publishing (2019)

[23] Koenig, N.P., Howard, A.: Design and use paradigms for gazebo, an open-source multi-robot simulator. Proceeding of IROS **3**, 2149–2154 vol.3 (2004)

[24] Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet Classification with Deep Convolutional Neural Networks. In: Proceedings of NIPS. pp. 1106–1114 (2012)

[25] Lai, K.T., Lin, C.C., Kang, C.Y., Liao, M.E., Chen, M.S.: Vivid: Virtual environment for visual deep learning. Proceedings of the 26th ACM international conference on Multimedia (2018)

[26] Li, W., Keegan, T.H.M., Sternfeld, B., Sidney, S., Quesenberry, C.P., Kelsey, J.: Outdoor falls among middle-aged and older adults: a neglected public health problem. American journal of public health **96 7**, 1192–200 (2006)

[27] Liu, J., Philipsen, M., Moeslund, T.: Supervised versus self-supervised assistant for surveillance of harbor fronts. In: VISAPP. pp. 610–617 (2021)

[28] Martinez-Gonzalez, P., Oprea, S., Castro-Vargas, J.A., Garcia-Garcia, A., Orts-Escolano, S., García-Rodríguez, J.A., Vincze, M.: Unrealrox+: An improved tool for acquiring synthetic data from virtual 3d environments. 2021 International Joint Conference on Neural Networks (IJCNN) pp. 1–8 (2021)

[29] Nikolov, I., Liu, J., Moeslund, T.: Imitating emergencies: Generating thermal surveillance fall data using low-cost human-like dolls. Sensors **22**(3) (2022)

[30] Nikolov, I., Philipsen, M., Liu, J., Dueholm, J., Johansen, A., Nasrollahi, K., Moeslund, T.: Seasons in drift: A long-term thermal imaging dataset for studying concept drift. In: Proceedings of NeurIPS (2021)

[31] Nyman, S.R., Ballinger, C., Phillips, J.E., Newton, R.: Characteristics of outdoor falls among older people: a qualitative study. BMC Geriatrics **13**(1) (2013)

[32] Paul Voigt, A.v.d.B.: The EU General Data Protection Regulation (GDPR) - A practical guide. Cham: Springer International Publishing (2017)

[33] Pramerdorfer, C., Strohmayer, J., Kampel, M.: Sdt: A synthetic multi-modal dataset for person detection and pose classification. In: Proceedings of ICIP. pp. 1611–1615 (2020)

[34] Shamir, O., Zhang, T.: Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes. In: Proceedings of the 30th ICML. Proceedings of Machine Learning Research, vol. 28, pp. 71–79. PMLR (2013)

[35] Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., Vasudevan, V., Han, W., Ngiam, J., Zhao, H., Timofeev, A., Ettinger, S., Krivokon, M., Gao, A., Joshi, A., Zhang, Y., Shlens, J., Chen, Z., Anguelov, D.: Scalability in perception for autonomous driving: Waymo open dataset. In: Proceedings of CVPR. pp. 2443–2451 (2020)

[36] Szot, A., Clegg, A., Undersander, E., Wijmans, E., Zhao, Y., Turner, J., Maestre, N., Mukadam, M., Chaplot, D.S., Maksymets, O., Gokaslan, A., Vondrus, V., Dharur, S., Meier, F., Galuba, W., Chang, A.X., Kira, Z., Koltun, V., Malik, J., Savva, M., Batra, D.: Habitat 2.0: Training home assistants to rearrange their habitat. In: Proceedings of NeurIPS. p. 251–266 (2021)

[37] Unity Technologies: Unity Perception package. `https://github.com/Unity-Technologies/com.unity.perception` (2020)

[38] Zhang, H., Hu, T., Zhang, J.: Surface emissivity of fabric in the 8-14 m waveband. J Textile Inst **100** (2009)

[39] Zhang, L., Gonzalez-Garcia, A., van de Weijer, J., Danelljan, M., Khan, F.S.: Synthetic data generation for end-to-end thermal infrared tracking. IEEE Transactions on Image Processing **28**(4), 1837–1850 (2019)

[40] Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceeding of ICCV. p. 2242–2251 (2017)