Improving Data-Scarce Image Classification Through Multimodal Synthetic Data Pretraining

Carl Brander
D-MAVT, ETH Zürich
Zürich, Switzerland
cbrander@student.ethz.ch

Cristian Cioflan
D-ITET, ETH Zürich
Zürich, Switzerland
cioflanc@iis.ee.ethz.ch

Vlad Niculescu
D-ITET, ETH Zürich
Zürich, Switzerland
vladn@iis.ee.ethz.ch

Hanna Müller
D-ITET, ETH Zürich
Zürich, Switzerland
hanmuell@iis.ee.ethz.ch

Tommaso Polonelli
D-ITET, ETH Zürich
Zürich, Switzerland
tommaso.polonelli@pbl.ee.ethz.ch

Michele Magno
D-ITET, ETH Zürich
Zürich, Switzerland
michele.magno@pbl.ee.ethz.ch

Luca Benini
D-ITET, ETH Zürich
Zürich, Switzerland
lbenini@iis.ee.ethz.ch

***Abstract*—Deep Learning algorithms and models greatly benefit from the release of large-scale datasets, also including synthetically generated data, when real-life data is scarce. Multimodal datasets feature more descriptive environmental information than single-sensor ones, but they are generally small and not widely accessible. In this paper, we construct a synthetically-generated image classification dataset consisting of grayscale camera images and depth information acquired from an 8×8-pixel Time-of-Flight sensor. We propose and evaluate six Convolutional Neural Network-based feature-level fusion models to integrate the multimodal data, outperforming the accuracy of the camera-only model by up to 17% in real-world settings. By pretraining the model on synthetically-generated sample pairs, followed by fine-tuning it with only 16 real-domain samples, we outperform a non-pretrained counterpart by 35% while maintaining the storage constraints in the order of hundreds of kB. Our proposed convolutional model, pretrained on both synthetic and real-world sensor data, achieves a top-1 accuracy of 86.48%, proving the benefits of using multimodal datasets to train feature-level data fusion neural networks. Low-power emerging embedded microcontrollers, such as multi-core RISC-V systems-on-chip, are perfect candidates for running our model due to their reduced power consumption and parallel computing capabilities that speed up inference.**

***Index Terms*—sensor fusion, multimodal data fusion, time of flight sensor, synthetic data, tinyML, low-power, image classification**

I. Introduction

The development of Artificial Intelligence (AI) has been accelerated by the rapid growth of Deep Learning (DL) models, enabled by the democratization of access to computing resources [1] and the maturity of publicly available datasets [2]. This led to an accelerated development of more accurate and more complex neural networks, with ever stronger generalization. For instance, Convolutional Neural Network (CNN)-based AlexNet [2] achieved 62.5% classification accuracy on ImageNet [2] with $60\,\mathrm{Mparameters}$. The more recent work [3], relying on Visual Transformers, scores 90.71% accuracy on the same dataset, yet this is achieved by being $31\times$ more complex (i.e., $1.88\,\mathrm{Bparameters}$).

On another key trend, the cloud computing approach is challenged by the edge computing paradigm, aiming to decrease the communication energy and the system response latency and to increase the data privacy [1]. In the context of Machine Learning (ML)-at-the-edge, one must abide by the TinyML constraints [1]. As the target embedded systems are battery-powered, always-on devices, the computational effort and, thus, energy consumption should be minimized to prolong the battery lifetime. Furthermore, the memory available on edge devices is strictly limited by cost and size constraints. However, obtaining a good accuracy with TinyML-compliant models also requires large training datasets [4], which maximize the generalization capabilities.

While largely beneficial in improving the prediction performance, gathering a dataset is a non-trivial effort, often requiring volunteers to acquire data through crowd sourcing [5]. In addition, simply collecting the target number of samples is insufficient, as one must also ensure their fairness and quality – i.e., producing a demographically unbiased dataset [5]. Furthermore, the data must be correctly labeled and, albeit this step can be partially automated, tedious manual review is still required for detecting and discarding the invalid entries [5].

Apart from the size of the dataset employed to train a neural network, the model's accuracy is also affected by the nature (i.e., type) of the training features. Acquiring and mixing data from multiple sensors could result in more descriptive environmental information that leads to a higher model accuracy in scenarios such as object detection and image classification [6]. This approach is becoming prevalent in the field of embedded systems; target devices such as battery-powered sensor nodes [7], smart cameras [8], smart glasses, and biomedical wearable systems need to extract environmental and foreground features using RGB images and a heterogeneous set of sensors. To this end, large Light Detection And Ranging (LiDAR) sensors with sizes of multiple centimeters [9], infeasible for power- or size-constrained systems, are replaced with smaller low-resolution Time-of-Flight (ToF)-based data acquisition modules. Such sensors, whilst being lightweight and energy efficient [10], augment the traditional camera data, monochrome or RGB format, with depth information enhancing the capabilities of the proposed system. Within this context, multimodal Deep Neural Networks (DNNs) can fuse the information from multiple low-power sensors typically onboard the power-constrained device and can enable real-time and accurate inference on computational limited Microcontroller Units (MCUs) [6], consistently increasing systems' accuracy over their single-sensor counterparts [11], [12].

To this aim, we propose six lightweight, efficient multimodal image classification DL models that integrate grayscale camera images and depth information acquired from a low power, miniaturized 8×8-pixel ToF sensor – i.e., VL53L5CX. To achieve effective multisensory integration, we perform a study and a comparison among the six different CNN-based feature-level data fusions, analyzing the impact of early-/late-fusion. We additionally verify the memory efficiency of the proposed models and demonstrate that they only require between 367 kB and 2378 kB using a `fp32` representation, or four times less with `int8` quantization, proving that our approach is suitable to operate onboard resource-constrained embedded platforms. We design our models targeting multi-core RISC-V-based platforms, whose low-power parallel computing capabilities are ideal for deep learning applications to achieve energy efficiency [13].

Training multimodal DNNs for embedded systems with human-labeled datasets is very hard because of the scarcity of such datasets and the unaffordable cost of generating them in the context of tiny and inexpensive system design. To mitigate the problem of acquiring a large real-world dataset and improve the classification accuracy of our proposed networks, we employ an *ad-hoc* synthetically-generated dataset, evaluated against a real-world dataset acquired for the scope of this paper. For this, the multi-pixel ToF sensor was fully characterized through practical experiments to accurately replicate its behavior in the simulated environment. In particular, the Gaussian sensor noise, the pixel invalidity probability, and the measured distance accuracy in the presence of different reflective or non-reflective materials were modeled.

Notably, our best-performing proposed DL-based model achieves a classification accuracy of up to 85.59% with four classes on real-world data through data fusion, an improvement of 10% over a 55% larger baseline model only using camera images, at a model cost of 126 kparameters, despite the very low depth-image resolution. Furthermore, by pretraining on synthetically-generated data, we outperform by 35% a model trained solely with real-world data, demonstrating the possibility of generating a consistent set of data pairing real-world and simulated sensors. Remarkably, only four real-world samples per class are sufficient to fine-tune the synthetically-augmented model in order to surpass a model trained from scratch with 160 real-world samples.

II. Related Work

Image classification using DL-based neural networks received significant attention from the research community [14], [15], especially when considering CNN backbones [16]. Notably, the majority of existing works target real-world systems equipped with cameras that provide the input of the neural networks. Due to the miniaturization of the cameras and therefore the reduction in their power consumption, these devices became more and more popular also in resource-constrained, low-power embedded systems [7].

In recent years, ML-capable Internet-of-Things (IoT) nodes enabled the edge computing paradigm – performing inference on the MCU, near the sensor. Challenging the traditional single-sensor approach, multi-sensor works proposed to maximize the knowledge of the environment by acquiring multimodal information. In such a context, one system property that must be accounted for is the fusion methodology. Data fusion represents the merger of information acquired from different sensors. We relax the taxonomy proposed by Cui *et al.* [17], distinguishing three fusing categories: data-, feature- and result-level fusion. In the case of data fusion, raw data acquired by the sensors are aligned and directly fed into the neural network. Conversely, the result-level method fuses the results of each pipeline, leveraging each proposal to generate the system-level prediction. Our methodology relies instead on feature-level fusion, using different DL branches to extract relevant features from the input data, followed by merging and further processing them in order to perform the classification task. As opposed to data-level fusion, our approach allows for tailoring each branch to the characteristics of its respective input data. Furthermore, compared to result-level fusion, we reduce the memory and computational costs associated with separating the processing branches by merging them at an earlier stage, thus meeting the available resources of the edge platform.

When training classification models, it is necessary that the training set and the test set belong to the same distribution [18]. Nonetheless, in order to employ data that is representative of the real-world environment in which the trained model will be deployed, said data must be acquired and labeled, an expensive and often infeasible process. Alternatively, synthetic data can be used during the pretraining stage [19], [20], acquired in virtual environments (e.g., Unity engine[1], Gazebo[2], Webots[3]) built to emulate the real ones [21]. In such scenarios, the acquisition cost is negligible, and the labeling process can be automated. Nevertheless, when models are deployed in open-world, previously unseen scenarios, the test samples will be out-of-distribution [18]. Although the real-world distribution could be captured through on-device learning [13], the energy effort scales with the number of samples required for the adaptation process. In this work, we ensure that the classification accuracy of our data-fusion, synthetically-pretrained models is sufficiently large on the on-site data in order to minimize the number of real-world samples used for fine-tuning, whilst simultaneously maximizing

[1] https://unity3d.com/

[2] https://www.gazebosim.org/

[3] https://www.cyberbotics.com/

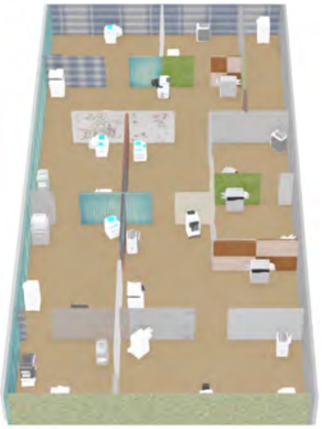

Fig. 1. **Multimodal data gathering in simulation:** static indoor environment simulated in Webots for challenging acquisition contexts (i.e., occlusion, poor illumination, multiple poses, complex background) for the printer class.

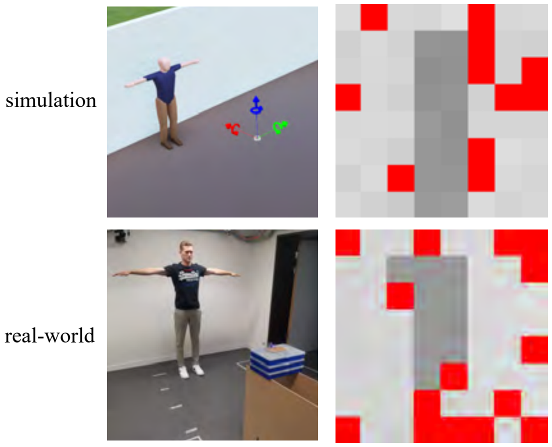

Fig. 2. **Comparison of acquisition setups**: on the left, the acquisition scenario is shown for both synthetic data (up) and real-world data (down). On the right, the ToF output for each scenario is depicted.

their contribution during the fine-tuning stage.

III. Methods

A. Data Acquisition

In this section, we present the methodologies adopted for this work and introduce the deep learning models that we evaluate. Then, in Section IV we provide a quantitative analysis of how the proposed models perform on synthetic and real-world data and show how synthetic data enables the model to better generalize in the real world. We started by collecting a real-world dataset inside office rooms, suitable for tasks such as nano-drone indoor navigation. To this aim, we used a 320×320-pixel Himax HM01B0 grayscale camera and an 8×8-pixel STM VL53L5CX ToF depth sensor. Both sensors are lightweight (i.e., below $0.5\,\mathrm{g}$) and low-power, which make them suitable candidates for limited-payload, resource-constrained embedded systems such as MCUs. The VL53L5CX and the HM01B0 grayscale camera are sampled at the same time with a synchronization error below $33\,\mathrm{ms}$. During the 41 acquisition sessions, a total of 2424 sample pairs were collected and labeled, with the samples categorized into four classes, representative for indoor environments: person, printer, door, chair/table. Note that the last class consists of sample pairs containing chairs, tables, or both, given that in real-world environments these objects are often found together.

To mitigate the time-consuming process of real-world sample collection and labeling, rather than expanding the aforementioned dataset, we created a second, synthetic dataset using the Webots simulator. Physically-accurate models of the ToF sensor and the grayscale camera were developed, reliably resembling their real-world counterparts [10]. In this case, the two sensors are virtually sampled at the same time with perfect time synchronization. The sensor was empirically characterized measuring each pixel offset, variance, and noise. Moreover, the ToF sensor capabilities were modeled in different conditions, e.g., from total darkness to $1000\,\mathrm{lx}$ and in front of light absorbent/reflective surfaces, from which the pixel-invalid probability was extracted. Moreover, we performed domain randomization to integrate several different settings (i.e., occlusion, poor illumination, multiple poses, complex background), thus generating diverse sample pairs. Specifically, to simulate occlusion cases, we added augmentation shapes e.g., pyramids, spheres, and cubes, randomly placed in the sensors' Field of View (FoV) and partially masking the target object. Furthermore, each data acquisition session was performed in a slightly different simulated indoor environment, where the walls', ceiling's, and floor's textures were randomly selected from a set of over 50 Webots-provided textures. One instance of a so-generated office environment is depicted in Figure 1.

To minimize the acquisition time, we only capture a sample pair if the target objects' area is between 8% and 42% of the FoV area of both sensors, verified using Webots' built-in image segmentation feature. Labeling was automated during the synthetic dataset acquisition, as we generated single-class environments per acquisition session, with pre-existing knowledge of the target class. With this, 2392 data pairs were generated, balanced across the four classes of interest.

The images acquired using the simulated grayscale camera were further processed, aligning their illumination level, contrast, and sharpness with those of the Himax HM01B0 camera used for the real-world dataset acquisition. Furthermore, given the ToF sensor's FoV of 45°, the camera images, originally simulating an 87° FoV as the real sensor, were cropped to 170×170 pixels so that the fields of view of the two sensors match and are perfectly aligned. Lastly, we performed a qualitative analysis of the ToF-generated data, ensuring depth alignment (visually inspected) between the real-world and the synthetically-generated datasets, as shown in Figure 2.

B. Data Fusion Architecture

One of the main contributions of this paper is the design of a deep learning model for image classification that exploits

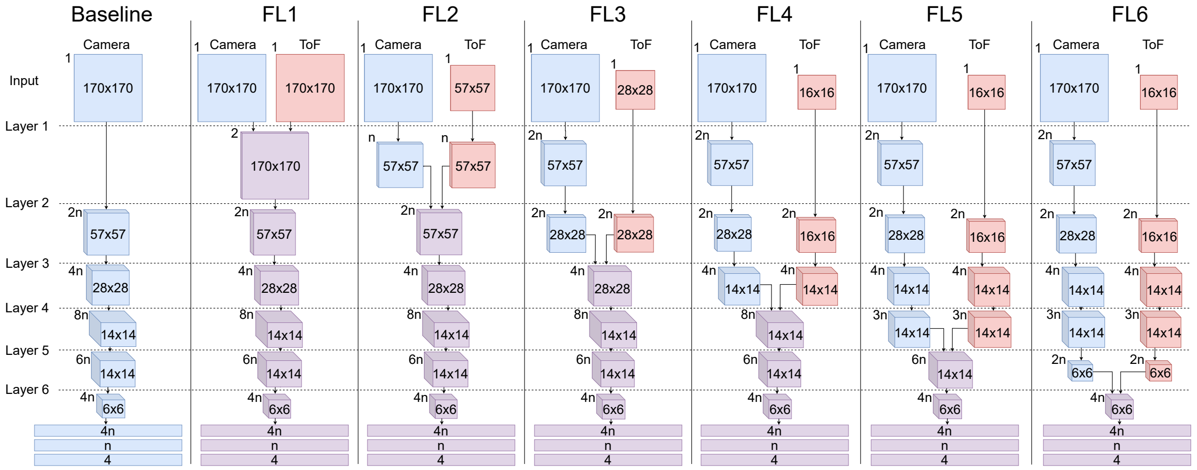

Fig. 3. **Feature-level sensor fusion in CNNs:** we propose six convolutional topologies, varying the fusion depth and the network's width (i.e., *n* indicates the feature channels). Each sensor fusion architecture concatenates the two data strands on another layer. With blue we represent grayscale camera data, red indicates ToF data, and purple, fused data.

the sensor fusion camera-ToF and meets the requirements of running in resource-constrained embedded platforms. Given the efficacy of CNNs in image classification problems, as well as their accuracy-parameters efficiency [6], we consider a CNN-based, camera-only baseline. Motivated by the benefits of sensor fusion models discussed in Section II, we introduce six feature-level fusion networks, depicted in Figure 3. We denominate the networks Fusion-on-Layer (FL)1-6, based on the layer at which the feature fusion takes place. Each proposed network consists of five convolutional stages; one stage represents one depthwise convolution, a batch normalization layer, and a ReLU activation, followed by a pointwise convolution, a batch normalization layer, and a ReLU activation. The advantage of using depthwise separable CNNs over standard 2D convolution resides in the reduced computational load and storage requirements of the former compared to the latter. This allows the increase of the first layers' width, thus accumulating knowledge over the input features, while the activations' size decreases throughout the network, increasingly focusing on coarse-grained features (i.e., low-level information). The networks conclude with two fully connected layers, aggregating the information and distilling it for the four-class softmax layer.

We propose six networks in order to investigate the impact of the fusion stage, with FL1 directly aggregating the sensor data through concatenation along the width dimension, whereas FL6 sensor fusion precedes the first fully connected layer. To enable concatenation along the channel dimension, the ToF data is upsampled through bilinear interpolation according to the dimension of the camera feature maps, with the aim of minimizing the resizing artifacts. Notably, we parametrize the number of convolutional filters (i.e., hereinafter denoted *n*) to ablate over the network width, determining the optimal number of feature channels.

TABLE I

ACCURACY[%] FOR THE PROPOSED NETWORKS, CONSIDERING THE NETWORK WIDTH (N) AND THE FUSION LEVEL, IN DIFFERENT EXPERIMENTAL SETTINGS. THE SOURCE OF THE TRAINING AND TESTING SETS IS MENTIONED IN EACH SECTION.

N	**10**	**20**	**30**	**40**	**50**	**60**
Training: real-world; Testing: real-world						
Baseline	67.48	76.44	75.69	75.62	82.73	77.89
FL1	79.74	82.95	83.46	83.84	82.67	83.12
FL2	82.00	80.87	77.87	79.42	81.63	79.46
FL3	84.70	78.47	79.66	79.41	80.75	75.67
FL4	78.21	79.68	81.04	83.50	83.66	83.09
FL5	76.75	80.05	80.68	**85.59**	82.93	81.70
FL6	78.44	83.85	78.54	81.04	82.76	80.04
Training: synthetic; Testing: synthetic						
Baseline	49.01	75.18	78.91	81.05	78.83	80.54
FL1	78.93	81.50	81.68	82.71	82.65	83.87
FL2	76.46	79.99	80.80	80.67	82.46	82.56
FL3	80.37	81.88	83.86	84.33	84.45	84.43
FL4	81.54	83.21	82.04	81.13	81.35	81.40
FL5	80.77	82.77	83.18	**83.38**	82.55	80.55
FL6	78.46	82.60	83.37	82.35	80.20	81.99
Training: synthetic; Testing: real-world						
Baseline	52.07	58.62	58.23	52.98	58.83	57.33
FL1	50.40	48.55	47.91	52.78	43.48	50.32
FL2	46.45	46.89	50.73	45.69	48.15	49.88
FL3	48.87	47.95	53.13	50.27	52.15	51.02
FL4	51.13	46.61	53.30	45.36	48.35	51.66
FL5	46.55	50.10	53.55	**53.72**	51.88	51.07
FL6	53.31	50.52	55.14	52.04	52.31	54.76

IV. RESULTS

In the following, we provide an experimental evaluation of the models introduced in Section III-B performed on both the real-world dataset and the synthetic dataset. Unless stated otherwise, we split each dataset in a 70:15:15 ratio for the train:validation:test sets, empirically found to be the best-performing configuration. For each experiment, the reported

TABLE II
MODEL SIZE [KPARAMETERS] FOR THE PROPOSED TOPOLOGIES, CONSIDERING THE NETWORK WIDTH (N) AND THE FUSION LEVEL.

N	10	20	30	40	50	60
Baseline	14.4	51.9	112.7	196.6	303.8	434.1
FL1	14.4	52.0	112.7	196.7	303.9	434.2
FL2	14.4	51.9	112.7	196.6	303.8	434.1
FL3	14.0	50.4	109.1	190.3	293.9	419.9
FL4	12.4	44.0	94.7	164.7	253.9	362.2
FL5	10.0	34.4	73.1	**126.3**	193.9	275.8
FL6	8.8	29.6	62.3	107.1	163.9	232.6

results are the mean of five runs. Moreover, we employed a learning rate of 10^{-3} and we used Adam [22] optimizer, considering a batch size of 20. The weight update relies on the computation of the cross-entropy loss.

A. Network Architecture Selection

With the aim of investigating the optimal network topology, we compare our FL topologies against the Baseline, performing the training and testing on the real-world dataset. We provide the results of these experiments in Table I – upper part. First of all, we note that the camera-only baseline is consistently outperformed by the multimodal networks (i.e., by up to 17% given similar width constraints), proving that image classification networks can benefit from having a more diverse environment representation. Secondly, when analyzing the networks' width, one can observe that the accuracy increases with the width, as more features from the data pairs can be extracted when more channels are available. For most networks, the accuracy peaks for $n \in \{40, 50\}$, followed by decreasing as we continue to increase the networks' width; we argue that this is due to an overfitting effect, obtained by training complex networks on few-class datasets.

In Table II, we represent the number of network parameters for each topology. The storage requirements increase with the network width while decreasing with fusion depth, with early fusion requiring the downstream processing of the concatenated features. Targeting an edge computing setting, we considered ultra-low-power RISC-V multi-core platforms as a potential deployment target. Since deep learning models are highly parallelizable, employing such embedded platforms would maximize energy efficiency while reducing the model inference time. To minimize the data movement cost and maximize the core utilization, we would aim to store our entire network in the RAM memory (typically around hundreds of kB). Given its accuracy of 85.59% and its 126 $\mathrm{kparameters}$, which can be quantized from `fp32` to `int8` with negligible accuracy losses [23], FL5 with $n = 40$ represents the optimal network candidate. The FL5 model outperforms the baseline by 10%, while being 35% smaller.

We furthermore analyze the proposed topologies on our synthetically generated dataset. In the middle part of Table I, we report the accuracy results for each model when both training and testing are performed using the synthetic dataset.

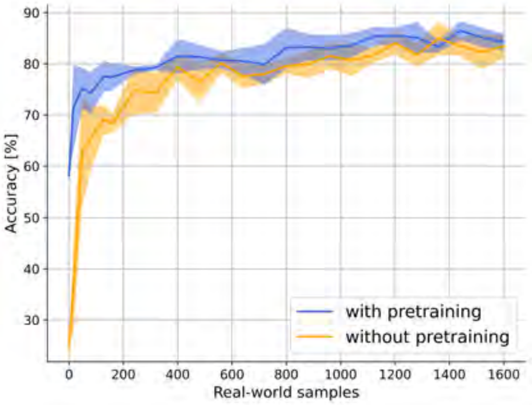

Fig. 4. **The impact of synthetic data:** a fusion model pretrained on synthetically-generated data pairs, further fine-tuned on real samples, outperforms its counterpart trained only with real-world data. Both models were tested on real scenarios.

We notice that all fusion models outperform the baseline by a larger margin (i.e., up to 32% improvement) compared to the previously analyzed case when training and testing are performed on the real-world dataset. This is most likely due to the synthetic dataset being generated in a fully controlled environment, thus the training samples are representative (i.e., *i.i.d.*) for the test set. We observe that FL5 with $n = 40$ achieves the highest accuracy in both the middle and upper parts of Table I, where training and testing were performed on the same dataset (i.e., synthetic or real-world). Specifically, the accuracy is 85.59% and 83.38% for training and testing on the real-world and synthetic dataset, respectively. However, our final objective is to evaluate how training on the synthetic dataset generalizes to real-world environments. We provide the results for this scenario in Table I – bottom part, and observe that the accuracy of FL5 with $n = 40$ is about 30% lower compared to the other two experiments. Although the accuracy of 53.72% exceeds that of an untrained system (i.e., 25% for a balanced, four-class dataset), the accuracy drop indicates dataset dissimilarities and emphasizes the need to fine-tune the synthetically-trained network when employed in real-world environments.

B. Fine-Tuning Results

Using our best-performing model, we test the hypothesis regarding the positive impact of pretraining a network on synthetic data. We use the entire synthetically-gathered dataset (i.e., 2392 data pairs) and we pretrain the FL5 selected in Section III-B. We fine-tune it on real-world data and we compared it against a network trained only on real-world data – testing is also performed using the real-world dataset. To ensure fairness, the training subsets were randomly selected out of the complete training set for each run. We additionally mention a fine-tuning learning rate of 5×10^{-4}, as well as a maximum number of 70 epochs for the training process. The results are shown in Figure 4.

In the absence of any information with respect to the deployment environment, the synthetically-pretrained network already achieved a classification accuracy of 58.14%. Using only four samples per class to fine-tune the model, an accuracy increase of 13.19% was measured, proving the model's ability to quickly adapt to new domains by learning the previously-unseen distributions. Therefore, the pretrained model reaches an accuracy over 70% fine-tuned on only 16 real-world samples, representing a 2× performance gain with respect to the model without pretraining. Reaching the same accuracy level by training the network from scratch (i.e., no pretraining) would require 10× more real-world data. Furthermore, the pretrained network can reach accuracy levels of more than 83% using only 800 real-world samples, whereas a similar top-1 accuracy requires training on the whole real-world dataset when no pretraining is performed. A maximum accuracy of 86.48% can be achieved when both datasets are employed during the learning process, emphasizing the benefits of large, diverse datasets in prediction performance.

V. Conclusion

The paper presented an acquisition methodology to gather multimodal synthetic data using a Webots simulator, at a negligible cost compared to gathering and sampling real-world data. We employ said methodology to simultaneously acquire grayscale camera images and ToF depth measurements.We further propose a feature-level fusion CNN and we show that multimodal data leads to accuracy gains of up to 17% over camera-only models. Lastly, by pretraining the multimodal network on synthetically-generated data and further fine-tuning and testing on real-world data, we achieve top-1 accuracies of up to 86.48%, outperforming a network trained solely on real data. For low-resource datasets, only four real-world sample pairs per class during the fine-tuning stage are sufficient to exceed 70% accuracy, our proposed method being 10× more data-efficient than the same model trained on only real-world data. Comprised of only 126 kparameters, our multimodal network is a suitable candidate for the quantization and deployment process targeting MCUs, process that will be addressed in a future work.

References

[1] T. Qiu, J. Chi, X. Zhou *et al.*, "Edge computing in industrial internet of things: Architecture, advances and challenges," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 4, pp. 2462–2488, 2020.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. Burges, L. Bottou *et al.*, Eds., vol. 25. Curran Associates, Inc., 2012.

[3] A. Dosovitskiy, L. Beyer, A. Kolesnikov *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=YicbFdNTTy

[4] Z. Dai, H. Liu, Q. V. Le *et al.*, "Coatnet: Marrying convolution and attention for all data sizes," in *Advances in Neural Information Processing Systems*, A. Beygelzimer, Y. Dauphin, P. Liang *et al.*, Eds., 2021. [Online]. Available: https://openreview.net/forum?id=dUk5Foj5CLf

[5] M. Mazumder, S. Chitlangia, C. Banbury *et al.*, "Multilingual spoken words corpus," in *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. [Online]. Available: https://openreview.net/forum?id=c20jiJ5K2H

[6] K. Bayoudh, R. Knani, F. Hamdaoui *et al.*, "A survey on deep multimodal learning for computer vision: advances, trends, applications, and datasets," *The Visual Computer*, vol. 38, no. 8, pp. 2939–2970, Aug 2022. [Online]. Available: https://doi.org/10.1007/s00371-021-02166-7

[7] D. Brunelli, T. Polonelli, and L. Benini, "Ultra-low energy pest detection for smart agriculture," in *2020 IEEE SENSORS*. IEEE, 2020, pp. 1–4.

[8] M. Giordano, P. Mayer, and M. Magno, "A battery-free long-range wireless smart camera for face detection," in *Proceedings of the 8th International Workshop on Energy Harvesting and Energy-Neutral Sensing Systems*, 2020, pp. 29–35.

[9] K. Kimoto, N. Asada, T. Mori *et al.*, "Development of small size 3d lidar," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 4620–4626.

[10] V. Niculescu, H. Müller, I. Ostovar *et al.*, "Towards a multi-pixel time-of-flight indoor navigation system for nano-drone applications," in *2022 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, 2022, pp. 1–6.

[11] C. Zhang, H. Wang, Y. Cai *et al.*, "Robust-fusionnet: Deep multimodal sensor fusion for 3-d object detection under severe weather conditions," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–13, 2022.

[12] V. Crescitelli, A. Kosuge, and T. Oshima, "Poison: Human pose estimation in insufficient lighting conditions using sensor fusion," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–8, 2020.

[13] C. Cioflan, L. Cavigelli, M. Rusci *et al.*, "Towards on-device domain adaptation for noise-robust keyword spotting," in *2022 IEEE 4th International Conference on Artificial Intelligence Circuits and Systems (AICAS)*. IEEE, 2022, pp. 82–85.

[14] L. Schmarje, M. Santarossa, S.-M. Schröder *et al.*, "A survey on semi-, self-and unsupervised learning for image classification," *IEEE Access*, vol. 9, pp. 82 146–82 168, 2021.

[15] L. Peng, B. Qiang, and J. Wu, "A survey: Image classification models based on convolutional neural networks," in *2022 14th International Conference on Computer Research and Development (ICCRD)*. IEEE, 2022, pp. 291–298.

[16] L. Chen, S. Li, Q. Bai *et al.*, "Review of image classification algorithms based on convolutional neural networks," *Remote Sensing*, vol. 13, no. 22, p. 4712, 2021.

[17] Y. Cui, R. Chen, W. Chu *et al.*, "Deep learning for image and point cloud fusion in autonomous driving: A review," *Trans. Intell. Transport. Sys.*, vol. 23, no. 2, p. 722–739, feb 2022. [Online]. Available: https://doi.org/10.1109/TITS.2020.3023541

[18] J. Yang, K. Zhou, Y. Li *et al.*, "Generalized out-of-distribution detection: A survey," *ArXiV*, vol. abs/2110.11334, 2021.

[19] Q. Wang, J. Gao, W. Lin *et al.*, "Learning from synthetic data for crowd counting in the wild," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[20] R. He, S. Sun, X. Yu *et al.*, "IS SYNTHETIC DATA FROM GENERATIVE MODELS READY FOR IMAGE RECOGNITION?" in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=nUmCcZ5RKF

[21] G. Ros, L. Sellart, J. Materzynska *et al.*, "The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 3234–3243.

[22] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *ArXiV*, vol. abs/1412.6980, 2017.

[23] Y. Li, R. Gong, X. Tan *et al.*, "{BRECQ}: Pushing the limit of post-training quantization by block reconstruction," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=POWv6hDd9XH