Research Project Proposal: 3D TinyML for Video Streaming Analysis

Shalby Hazem, hazemhesham.shalby@mail.polimi.it

1. Introduction to the problem

Tiny ML is a Field of study in Machine Learning (ML) and Embedded Systems that aims at integrating ML mechanisms within low-powered devices (typically in the order of mW), like microcontrollers.

By performing the inference on-device and near-sensor, Tiny ML paradigms enable greater responsiveness and privacy (limited outgoing data from the device) while maintaining energy consumption far lower than collecting the data on an embedded system and then send them on a centralized server where predictions are computed. The efficiency of Tiny ML enables a class of smart, battery-powered, always-on applications that can revolutionize the real-time collection and processing of data. Given the advantages just mentioned, integrating ML within cheap and low-power consumption devices presents some challenges to overcome. These are mainly related to device heterogeneity and hardware constraints. The heterogeneity obstacle is due to the existence of a large number of devices with different power consumption, processing capabilities, memory, storage capacity, communication ports, protocols, etc. Due to those aspects, the implementation of universal development and benchmarking frameworks coping with device heterogeneity is not an easy task. The hardware constraints are related mainly to the memory capacity, which may be in the order of KB, and to the computation power. [2][7]

The analysis of video streaming is the operation of scanning a video (i.e. sequence of frames) with the purpose of finding interesting patterns. Concerning the Tiny ML field, the algorithms that can be run during this type of analysis are for example object classification, object detection, image segmentation, pose estimation, anomaly detection, etc. Currently, the execution of ML models to perform video analysis on-device is limited to a frame-by-frame inspection, this is due to the presence of the hardware constraints discussed previously (memory capacity and computation power), that limit both the model size and the inference speed. This is a huge limit to all the ML algorithms that can be implemented for this task since the information regarding the evolution of the frames is not taken into consideration. The Usage of a sequence of frames for making inferences is an important support for making the running ML algorithm noise-agnostic and more accurate.

The following research aims at proposing solutions to the classification problem, which in the case of video streaming analysis can be formulated as follows: being $x_t \in U \subset \mathbb{R}^{M \times N \times C}$ (M, N, and C are the sizes of the input) a frame of the video streaming and $y_t \in Y = \{\Omega_1, ..., \Omega_c\}$ a label associated to x_t, the goal is to construct a classifier $f_\theta(x_t, x_{t-1}, x_{t-2}, ...)$ able to map an unseen data $\bar{x_i}$ to its label $\bar{y_i}$.

The solution will take into consideration the constraints in terms of memory (i.e. the usual amount of memory available is in the order of hundreds of KB), computational power (typically measured in terms of floating point operations (**FLOPs**) or multiply-and-accumulate (**MAC**) operations), and energy consumption of the current generation of Tiny devices. subsequently, those constraints are analyzed by focusing on the convolutional neural networks (CNNs), that represent the state-of-the-art solution in many recognition, classification, and detection applications; In particular, the memory demand M_{CONV} of a CONV layer can be computed as follows: Let H, W, C be the sizes of the input of the CONV layer, R and S be the sizes of the C-dimensional filter, M be the number filters in the CONV layer, and b is the number of bits used for the representation then the output feature map will have size $E \times F \times M$ and the memory demand of the CONV will be $M_{CONV} = (N^w_{CONV} + N^{fm}_{CONV}) \times b$, where $N^w_{CON} = M \times R \times S \times C$ and $N^{fm}_{CONV} = H \times W \times C + E \times F \times M$. The total number of MAC ops in a CONV filter is $MAC_{CONV} = E \times F \times R \times S \times C \times M$, while the corresponding number of FLOPs is approximately $2 \times MAC_{CONV}$. Regarding the FC layers, the number of weights is $N^w_{FC} = H \times W + W$ and the memory demand is $M_{FC} = (N^w_{FC} + H + W) \times b$, while the total number of MAC ops is $MAC_{FC} = X \times W$. [6]

2. Main related works

In literature there are plenty of Tiny ML implementations of video stream analysis; Here some of the most significant ones are reported.

In [8] the video stream is analyzed frame-by-frame to come up with a recognition system that uses a RandomForest (RF) Classifier. The inference time of the proposed implementation is small ($\approx$ 12 ms), therefore the application runs almost in real-time.

In [5] a face recognition application is implemented using the Tiny ML paradigm; In particular the paper compares the performance of different versions of YOLO (You only look once). The most important parameter for the presented models in the paper was the value of frames per second (FPS), which exceeds 30 FPS for some implementations, therefore some of the presented models run almost in real-time.

In [9] a Real-Time Quality Assurance of Fruits and Vegetables is presented. the process is implemented via a pipeline that is composed mainly of three parts: feature extraction, dimensionality reduction via PCA, and classification part. Also in this case the analysis is done frame-by-frame

All the reported implementations use a frame-by-frame analysis, in which the sequence of frames preceding the one taken into consideration is not explored.

In literature, an important reference for the spatiotemporal feature learning is the 3-dimensional convolution networks (3D ConvNets) trained on a large-scale supervised video dataset, which is discussed in [4] and [10]. The 3D convolutions capture the motion information encoded in multiple contiguous frames, which is a desirable behavior when talking about the video analysis problem. 3D ConvNets seems the perfect fit for the problem of video analysis but for the Tiny ML field using these types of networks is challenging due to the memory and computation constraints.

3. Research plan

The main objective of the research is to answer the following question: *How is it possible to design 3D Tiny Machine Learning solutions able to support a video streaming analysis on tiny devices technologically constrained on memory, computation, and energy?* The answer will be obtained through the development of a TinyML solution that performs the classification task described before, taking into consideration the constraints introduced by the hardware and by the application itself (e.g inference time which should be real-time for the video streaming analysis task). For the task of classification the best approach is to use deep learning (DL) solutions; In particular, the development of any DL solution for a TinyML application is done following three main steps:

1. Redesigning the CNN architecture;
2. Introducing approximate computing mechanisms (e.g. pruning, quantization, etc.); and
3. Exploiting embedded-system code optimization.

The first step is the most critical, since the objective of this research is to design solutions capable of extracting features from multiple (contiguous) frames (e.g Figure 1); The idea to deliver such a feature extraction method is to explore existing and efficient ways of doing the standard 2D convolutions (e.g. separable convolutions [3] and dilated convolutions[11]) and combine them with 3D convolutions [4][10], which are suitable for the multiple frame analysis task but implementing them in the TinyML field is challenging.

The research plan is divided into five parts, which are described in the Gantt Chart reported in Figure 2; In the implementation part, the proposed solutions will be implemented both for RGB inputs(with three channel $C = 3$) and for inputs generated from the ToF sensor[1] (with one channel $C = 1$), which outputs 8×8 zones, each with a depth distribution, and its power consumption is about 200 mW. In the experimental part, the assessment of the output of the research will be done taking into consideration the accuracy of the classifier, the number of multiple contiguous frames used by the classifier (the more frame analyzed the better), the time needed for the inference (should be real-time), memory occupation, and the number FLOPs and MACs operations needed.

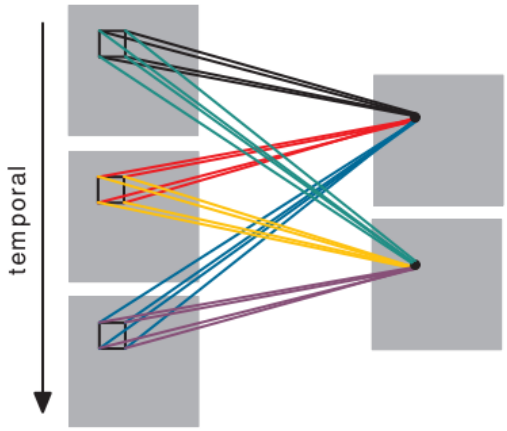

Figure 1: Example of extraction of multiple features from contiguous frames

#	Task Name	Duration
Complete thesis creation		~12 months
1	Problem formalization	~1-2 months
1.1	Identify the goals	~1-2 months
1.2	Identify the constraints	~1-2 months
1.3	Identify the innovation	~1-2 months
2	Literature analysis	-5-6 months
2.1	Identify similar works in the literature	~1-2 months
2.2	Formalize the current state of art for the problem	~1-2 month
2.3	Be updated on the works which are being released	~4 months
3	Implementation	~6-7 months
3.1	Create a very simple toy example of the proposed idea	~1-2 months
3.2	Increase the complexity to the desired one	~1-2 months
3.3	Improve, fix, test. Improve, fix, test...	~3-4 months
4	Experimental part	~3-4 months
4.1	Collect experiments results of the proposed solution	~2-3 months
4.2	Critically analyze the results, both if they are good or not	~2-3 months
4.3	Fix the obtained results	~1 month
5	Thesis writing	~3-4 months
5.1	Set up the sections	~1-2 months
4.2	Write	~2-3 months

Figure 2: Gantt chart

References

[1] VL53L5CX - Time-of-Flight 8x8 multizone ranging sensor with wide field of view - STMicroelectronics.

[2] Banbury, C. R., Reddi, V. J., Lam, M., Fu, W., Fazel, A., Holleman, J., Huang, X., Hurtado, R., Kanter, D., Lokhmotov, A., Patterson, D., Pau, D., Seo, J.-s., Sieracki, J., Thakker, U., Verhelst, M., and Yadav, P. Benchmarking TinyML Systems: Challenges and Direction, Jan. 2021. arXiv:2003.04821 [cs].

[3] Chollet, F. Xception: Deep Learning with Depthwise Separable Convolutions, Apr. 2017. arXiv:1610.02357 [cs].

[4] Ji, S., Xu, W., Yang, M., and Yu, K. 3D Convolutional Neural Networks for Human Action Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence 35*, 1 (Jan. 2013), 221–231.

[5] Probierz, E., Bartosiak, N., Wojnar, M., Skowroński, K., Gałuszka, A., Grzejszczak, T., and Kędziora, O. Application of Tiny-ML methods for face recognition in social robotics using OhBot robots. In *2022 26th International Conference on Methods and Models in Automation and Robotics (MMAR)* (Aug. 2022), pp. 146–151.

[6] Roveri, M. Is Tiny Deep Learning the New Deep Learning? In *Computational Intelligence and Data Analytics*, R. Buyya, S. M. Hernandez, R. M. R. Kovvur, and T. H. Sarma, Eds., vol. 142. Springer Nature Singapore, Singapore, 2023, pp. 23–39.

[7] Sanchez-Iborra, R., and Skarmeta, A. F. TinyML-Enabled Frugal Smart Objects: Challenges and Opportunities. *IEEE Circuits and Systems Magazine 20*, 3 (2020), 4–18.

[8] Sudharsan, B., Salerno, S., and Ranjan, R. TinyML-CAM: 80 FPS image recognition in 1 kB RAM. In *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking* (New York, NY, USA, Oct. 2022), MobiCom '22, Association for Computing Machinery, pp. 862–864.

[9] Tata, J. S., Kalidindi, N. K. V., Katherapaka, H., Julakal, S. K., and Banothu, M. Real-Time Quality Assurance of Fruits and Vegetables with Artificial Intelligence. *Journal of Physics: Conference Series 2325*, 1 (Aug. 2022), 012055.

[10] Tran, D., Bourdev, L., Fergus, R., Torresani, L., and Paluri, M. Learning Spatiotemporal Features With 3D Convolutional Networks. pp. 4489–4497.

[11] Yu, F., and Koltun, V. Multi-Scale Context Aggregation by Dilated Convolutions, Apr. 2016. arXiv:1511.07122 [cs].