# State of the Art on: 3D TinyML for Video Streaming Analysis

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

## 1. Introduction to the research topic

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 sending 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. [1][8]

In order to take advantage of the Tiny ML paradigm and to cope with the obstacles discussed, standard frameworks such as TensorFlow Lite micro[3] have been developed. This framework is the state of the art for TinyML development and follows a three-step pipeline, in which the first step is training the model using the Standard TensorFlow platform, the second step is converting the model to a lighter one using TensorFlow Lite conversion, and the last step is using the output of the second step to perform on-device-inference. The Conversion part of the process is the most significant part and it uses some techniques to reduce the size of the model, such as quantization[5], compression, pruning, and usage of convolution variations (e.g. separable convolutions [2] and dilated convolutions[12]).

Nowadays there are many IoT applications that take advantage of the Tiny ML paradigm, the one which is taken into consideration in the following section is the 3D analysis of video streaming for classification purposes.

### 1.1. Preliminaries

The classification problem treated by the research, in the case of video streaming analysis using multiple frames, 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 $\bar{y}_i$.

### 1.2. Research topic

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 research purpose is to explore ways for the usage of a sequence of frames for making inferences since this makes the running ML algorithm noise-agnostic and more accurate.

# 2. Main related works

## 2.1. Classification of the main related works

The research will explore existing and efficient ways of doing the standard 2D convolutions (e.g. separable convolutions [2] and dilated convolutions[12]) and combine them with 3D convolutions [4][11], to realize a method which is suitable for the multiple frame analysis task in the TinyML field; Hence, the literature related to this research can be classified into the following category:

- Implementation of TinyML applications that perform the video streaming analysis task frame-by-frame;
- Implementation of efficient versions of 2D convolutions that satisfy the TinyML hardware constraints;
- Implementation of 3D convolutions for spatiotemporal feature learning;
- Techniques to reduce the memory and the computational demands of the network.

## 2.2. Description of the main related works

### 2.2.1 Existing TinyML video streaming analysis

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

In [9] 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 [6] 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 [10] 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

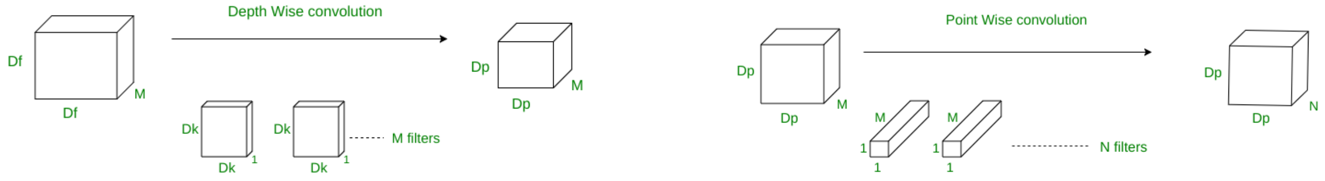
All the reported implementations present a limit to their application for the streaming video analysis task, which is the usage of frame-by-frame analysis, in which the sequence of frames preceding the one taken into consideration is not explored.

### 2.2.2 Efficient 2D convolutions for TinyML applications

In the literature, there are two common techniques that are used to approximate 2D convolutions with the goal of reducing the requirements of the Network:

- Separable convolutions [2], which are of two main type:
  - Spatial separable convolutions;
  - Depthwise separable convolutions.
- Dilated convolutions[12].

In the **spatial separable convolution** the kernel ($N \times N$) is broken into two smaller kernels ($1 \times N$ and $N \times 1$) and those kernels are multiplied sequentially with the input image to get the same effect of the full kernel, while the depthwise separable convolutions work with kernels that cannot be "factored" into two smaller kernels. Depthwise separable convolutions can be divided into two types: Depthwise convolution, and Pointwise Convolution. In Depthwise, the convolution is applied to a single channel at a time unlike standard CNN's in which it is done for all the M channels (see Figure 1a), while in pointwise, a 1×1 convolution operation is applied on the M channels (see Figure 1b).

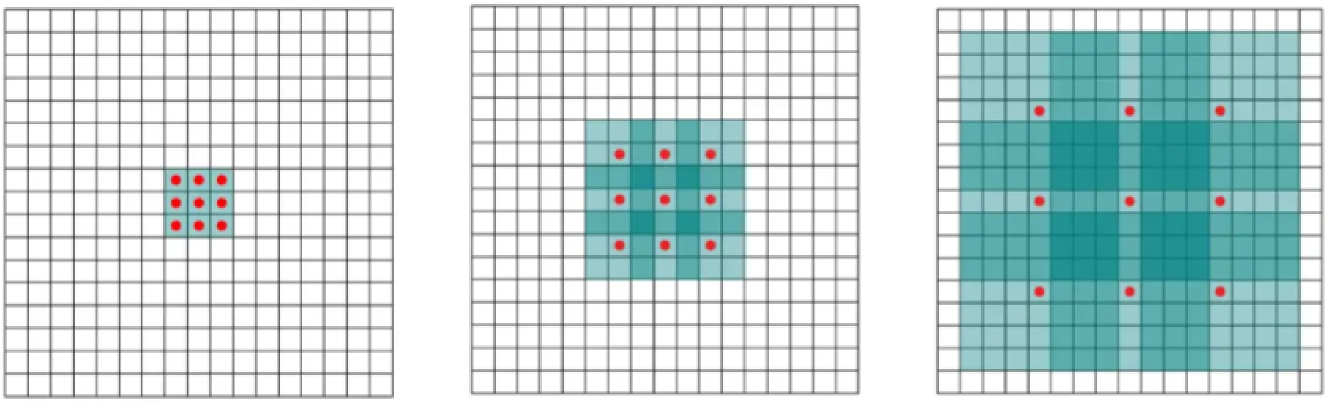


(a) Depthwise convolution

(b) Pointwise convolution

Figure 1: Depthwise separable convolutions schema

In the **dilated convolutions** the kernel is expanded by inserting empty spaces between its consecutive elements. Dilated convolution offers a wider field of view, and reduces the computational cost since fewer steps are required to cover the whole input. In Figure 2 some examples of dilated convolution are reported.



(a) 1 Dilated Convolution

(b) 2 Dilated Convolution

(c) 3 Dilated Convolution

Figure 2: Examples of application of the dilated convolutions

### 2.2.3 3D convolutions

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 [11].

**3D convolutions** applies a 3-dimensional filter to the dataset and the filter moves 3-direction (x, y, z) to calculate the low-level feature representations. Their output shape is a 3-dimensional volume space such as a cube or cuboid. 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 Convolutions seem the perfect fit for the problem of video analysis but for the TinyML field using these types of networks is challenging, due to hardware constraints (e.g. memory size and computation power).

#### 2.2.4 Technique to reduce network memory and computational demands[7]

Approximate computing mechanisms for TinyDL can be grouped into two main families:

- Precision scaling, which aims to reduce the memory occupation of a CNN by changing the precision (i.e., the number of bits used for the representation) of the weights and feature maps;
- Task dropping, which aims to reduce the computational load and memory occupation by skipping the execution of certain tasks associated with the processing pipeline.

**Precision scaling** for TinyDL relies on quantization mechanisms to reduce the memory requirements for storing weights and feature maps in CNNs. Quantization is aimed at reducing the parameter b accounts for the number of bits used for the representation of weights and feature maps.

**Task dropping mechanisms** can be classified into three main families: network pruning (a large number of weights in CNNs are often redundant and can thus be removed ), network architecture design (e.g. replacing large filters with a sequence of smaller filters), and transfer learning ( part of the pre-trained networks is used as feature extractors for application-specific and the rest is approximated).

### 2.3. Discussion

As reported in the previous section, the task of video streaming analysis using multiple (contiguous) frames is feasible with the usage of 3D convolutions, but implementing this type of solution in the TinyML field is challenging due to hardware constraints; Therefore, the main open issue is: *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?*

## References

[1] 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].

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

[3] David, R., Duke, J., Jain, A., Reddi, V. J., Jeffries, N., Li, J., Kreeger, N., Nappier, I., Natraj, M., Regev, S., Rhodes, R., Wang, T., and Warden, P. TensorFlow Lite Micro: Embedded Machine Learning on TinyML Systems, Mar. 2021. arXiv:2010.08678 [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] Nagel, M., Fournarakis, M., Amjad, R. A., Bondarenko, Y., van Baalen, M., and Blankevoort, T. A White Paper on Neural Network Quantization, June 2021. arXiv:2106.08295 [cs].

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

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

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