

tinyML: Designing Efficient Neural Architectures and Scaling Strategies for Edge Computing

Francesco Paissan

Energy Efficient Embedded Digital Architectures
Fondazione Bruno Kessler
fpaissan@fbk.eu

November 28, 2023

Presentation Overview

① Introduction

The Five (-1) Ws of tinyML
Challenges of tinyML

② Neural Network design

Rise and development of CNNs
tinyML-first CNNs
Hardware-Aware Scaling

③ Some applications...

YOLO-based
Zero-shot audio classification
micromind

1 Introduction

The Five (-1) Ws of tinyML
Challenges of tinyML

2 Neural Network design

Rise and development of CNNs
tinyML-first CNNs
Hardware-Aware Scaling

3 Some applications...

YOLO-based
Zero-shot audio classification
micromind

The Five (-1) Ws of tinyML

What?

- a fast-growing subfield of machine learning targeting **on-device** and **near-sensor processing**;

The Five (-1) Ws of tinyML

What?

- a fast-growing subfield of machine learning targeting **on-device** and **near-sensor processing**;

Why?

- many practical **benefits** (e.g. bandwidth reduction, infrastructure sustainability, scalability);
- **privacy** by design: enable processing on-device, thus sensitive data is never leaked;

The Five (-1) Ws of tinyML

What?

- a fast-growing subfield of machine learning targeting **on-device** and **near-sensor processing**;

Why?

- many practical **benefits** (e.g. bandwidth reduction, infrastructure sustainability, scalability);
- **privacy** by design: enable processing on-device, thus sensitive data is never leaked;

When?

- not clear, it was a continuous process, sometimes driven by necessity...

Who?

(tiny)AI researchers:

- come up with novel ML algorithms to compress and simplify NN model;
- generally approach tinyML as a ML problem;

Who?

(tiny)AI researchers:

- come up with novel ML algorithms to compress and simplify NN model;
- generally approach tinyML as a ML problem;

(AI)Embedded engineers:

- design custom NN accelerator and neuromorphic processors to speed up NN inference;
- approach tinyML as an engineering problem;

Who?

(tiny)AI researchers:

- come up with novel ML algorithms to compress and simplify NN model;
- generally approach tinyML as a ML problem;

(AI)Embedded engineers:

- design custom NN accelerator and neuromorphic processors to speed up NN inference;
- approach tinyML as an engineering problem;

But there's stuff also in the gray area...

Challenges of tinyML?



WORKSTATION

RAM: 10-100 GB

Storage: 10s of TB

Speed: 100 Billions of ops/s



PC/SBC

RAM: 1-10 GB

Storage: 10-100 GB

Speed: 1-10 Billions of ops/s



MCU

RAM: 10s - 100s of KBs

Storage: KBs - MBs

Speed: Millions of ops/s

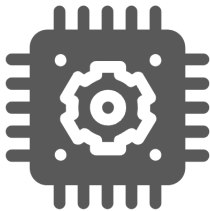


$\div 10$



$\div 10\ 000$

Target platforms

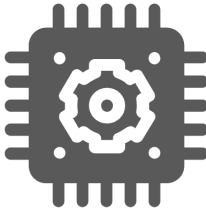


microcontrollers, SBC,
neuromorphic processors, ...

Target platforms

small parameter memory available

(kB - MB)



microcontrollers, SBC,
neuromorphic processors, ...

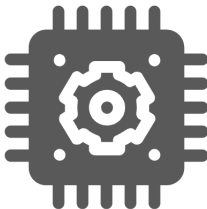
Target platforms

small parameter memory available

(kB - MB)

few operations per second

(million ops/s)



microcontrollers, SBC,
neuromorphic processors, ...

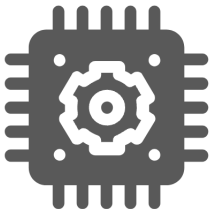
Target platforms

small parameter memory available

(kB - MB)

few operations per second

(million ops/s)



microcontrollers, SBC,

neuromorphic processors, ...

small working memory

(kB - MB)

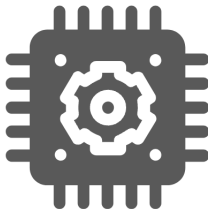
Target platforms

small parameter memory available

(kB - MB)

few operations per second

(million ops/s)



microcontrollers, SBC,

neuromorphic processors, ...

small working memory

(kB - MB)

limited operations support

(generally optimized for CNNs)

1 Introduction

The Five (-1) Ws of tinyML
Challenges of tinyML

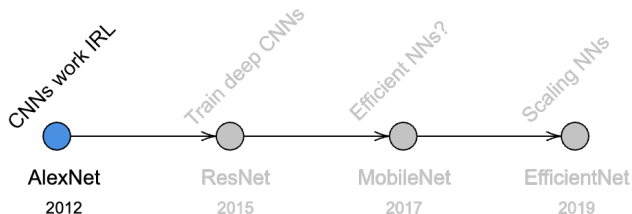
2 Neural Network design

Rise and development of CNNs
tinyML-first CNNs
Hardware-Aware Scaling

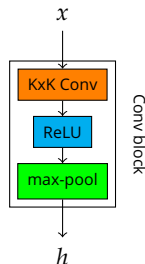
3 Some applications...

YOLO-based
Zero-shot audio classification
micromind

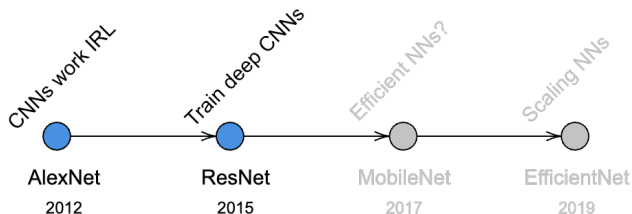
A quick peek at the literature



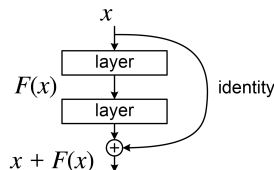
- ground-breaking CNN from 2012 was the first one to get good results on ImageNet;
- composed by a **sequence of convolutional blocks**, with varying configurations;



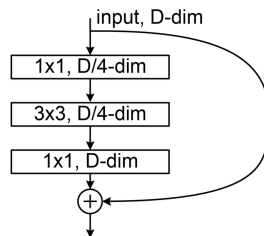
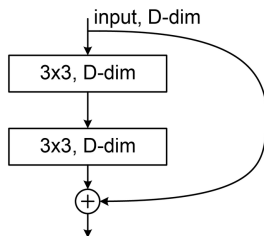
A quick peek at the literature



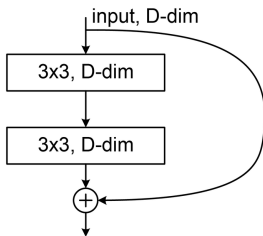
- improves the performance by enabling deeper networks via **skip connections**;
- again, is composed by a **sequence of convolutional blocks**, called residual blocks;
- residual blocks follow a wide/narrow/wide structure in the number of channels;



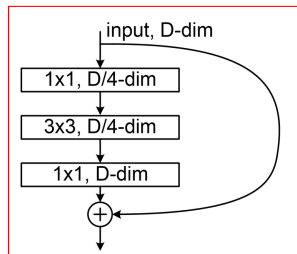
ResBlock variants



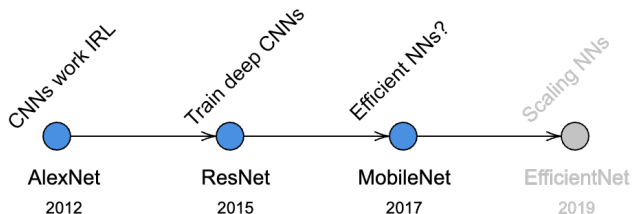
ResBlock variants



Wide-narrow-wide channel structure



A quick peek at the literature

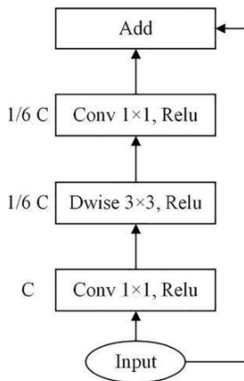


- tries to improve CNN efficiency by proposing the **inverted residual block**;

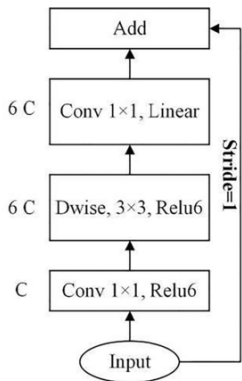
- tries to improve CNN efficiency by proposing the **inverted residual block**;
- differently from a ResBlock, this uses a narrow/wide/narrow structure in the number of channels;

- tries to improve CNN efficiency by proposing the **inverted residual block**;
- differently from a ResBlock, this uses a narrow/wide/narrow structure in the number of channels;
- additionally, groups are used inside the convolutions to reduce the computational complexity (depthwise convolutions);

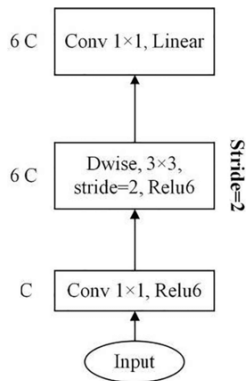
Inverted Convolutional Block



(a) Residual block

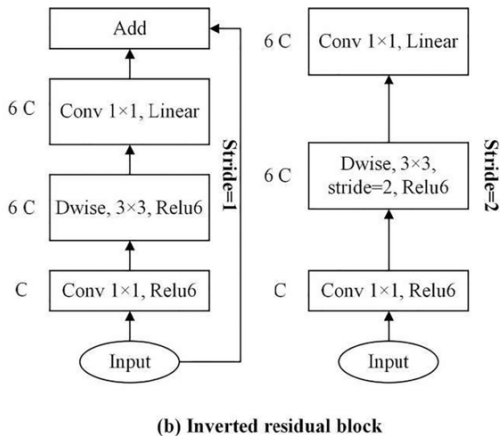
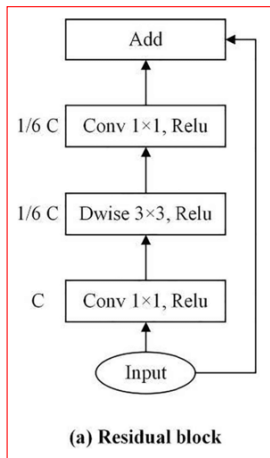


(b) Inverted residual block

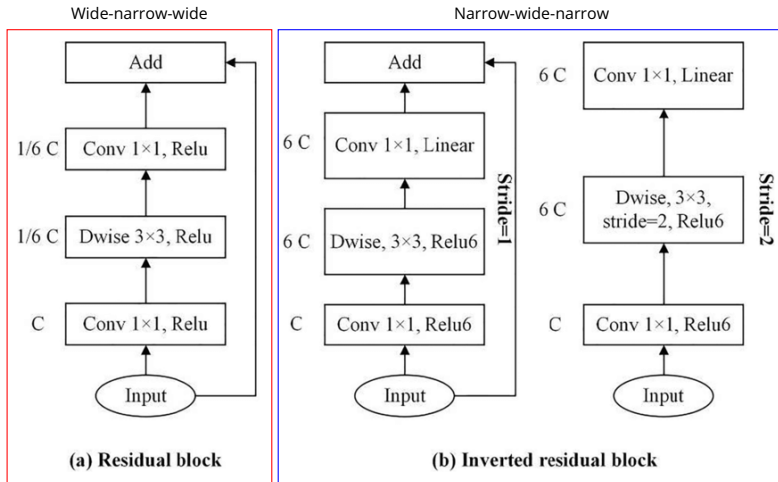


Inverted Convolutional Block

Wide-narrow-wide

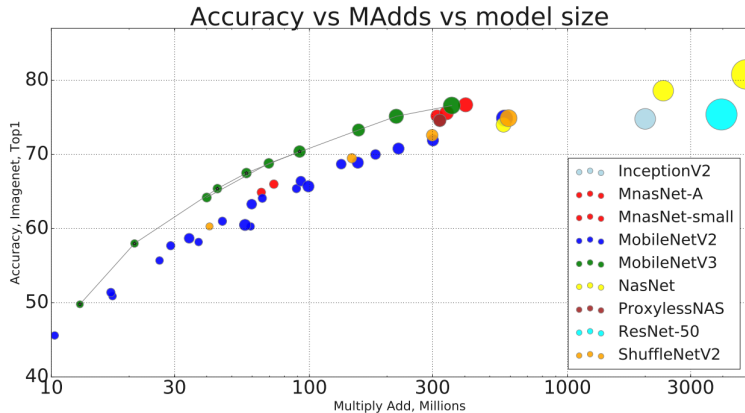


Inverted Convolutional Block

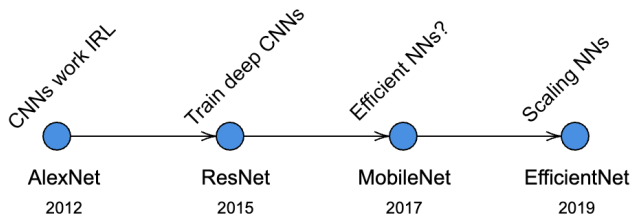


Just for comparison...

As of MobilNetv3 (Nov. 2019)...

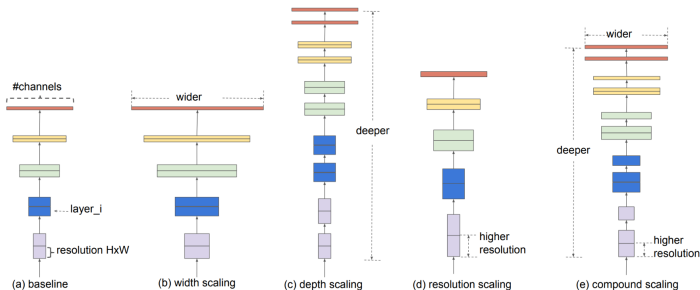


A quick peek at the literature



EfficientNet

- focuses on how we 'should' be scaling CNNs to obtain optimal performance;
- introduces the concept of compound scaling (i.e. scaling all dimensions is better than one dimension at a time);



Shortcomings of mainstream CNNs

- these neural networks are **too demanding** to run on edge devices and/or compromise performance too much trying to fit;

Shortcomings of mainstream CNNs

- these neural networks are **too demanding** to run on edge devices and/or compromise performance too much trying to fit;
- edge devices have different capabilities conf blocks **cannot exploit**;

Shortcomings of mainstream CNNs

- these neural networks are **too demanding** to run on edge devices and/or compromise performance too much trying to fit;
- edge devices have different capabilities conf blocks **cannot exploit**;
- compound scaling changes all the computational complexities in a **coupled** way;

Ideal CNN for edge processing

- a neural network that can **scale to low computational complexity** (≤ 1 MB of FLASH, ≤ 1 MB of RAM);
- a convolutional block that is designed to **exploit the available resources** maximally;
- a scaling strategy that allows fitting neural networks on **different edge platforms** based on the applications scenarios;

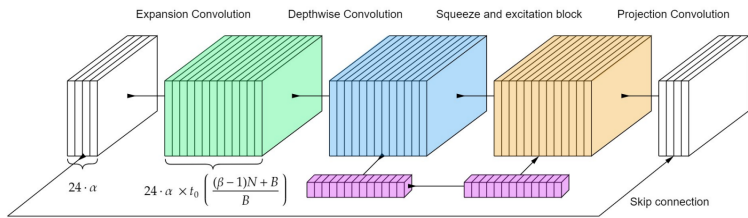
- based on **inverted residual blocks**, modified to decouple the computational resources;

- based on **inverted residual blocks**, modified to decouple the computational resources;
- designed and optimized for **multimedia analytics** at the edge (audio-video);

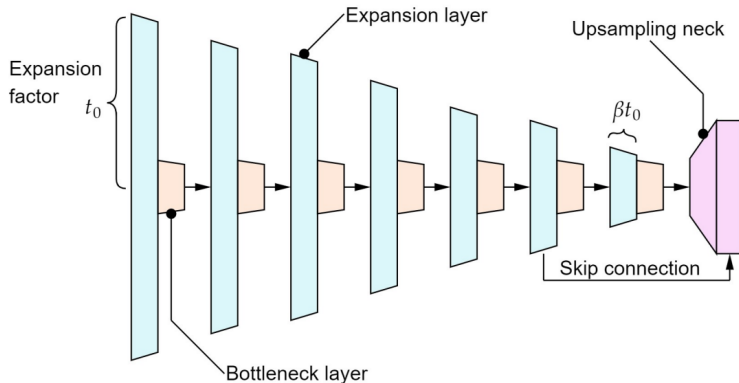
- based on **inverted residual blocks**, modified to decouple the computational resources;
- designed and optimized for **multimedia analytics** at the edge (audio-video);
- controls RAM (t_0), FLASH (β) and operations (α) using three hyperparameters;

PhiNets convolutional block

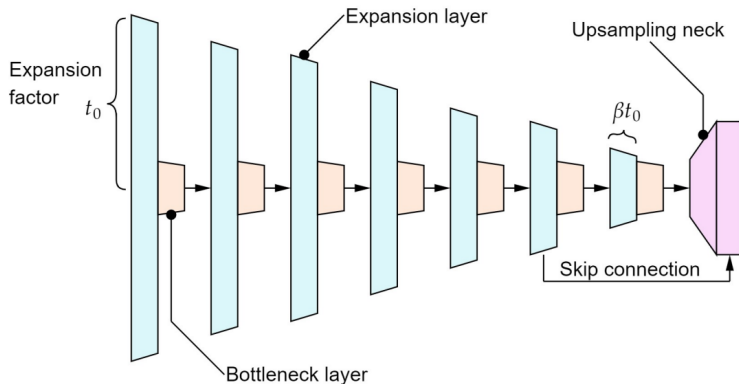
Narrow-wide-narrow structure for the number of channels...



The sequence of PhiNets conv blocks



The sequence of PhiNets conv blocks



```
from micromind.networks import PhiNet
```

Designing an optimized convolutional block

- PhiNets are designed based on **indirect efficiency metrics**, thus could be an ideal version of edge CNNs;

Designing an optimized convolutional block

- PhiNets are designed based on **indirect efficiency metrics**, thus could be an ideal version of edge CNNs;
- what happens if we try to break free of the common standards for convolutional block design and investigate from first principles?

Designing an optimized convolutional block

- PhiNets are designed based on **indirect efficiency metrics**, thus could be an ideal version of edge CNNs;
- what happens if we try to break free of the common standards for convolutional block design and investigate from first principles?

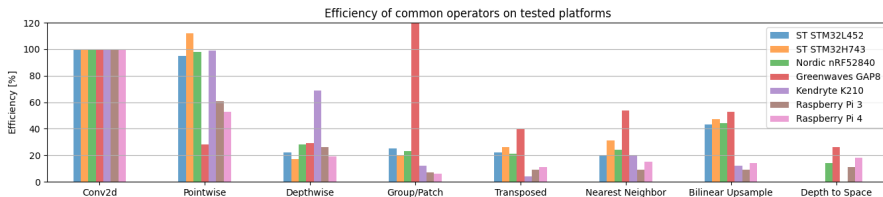
Let's see...

Definition 2.1

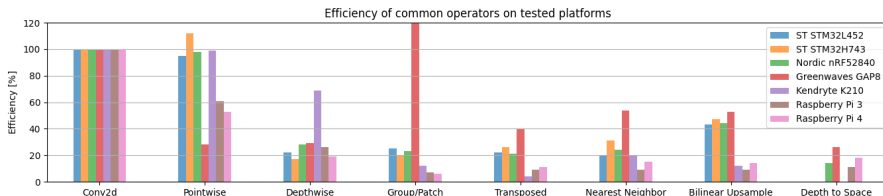
We assessed the actual efficiency of each operator (η_{op}) by calculating the ratio between the energy needed for a standard convolution (E_S) and the energy of the chosen operator (E_{op}) to perform an equivalent number of MACs.

$$\eta_{op} = \frac{E_S}{E_{op}}$$

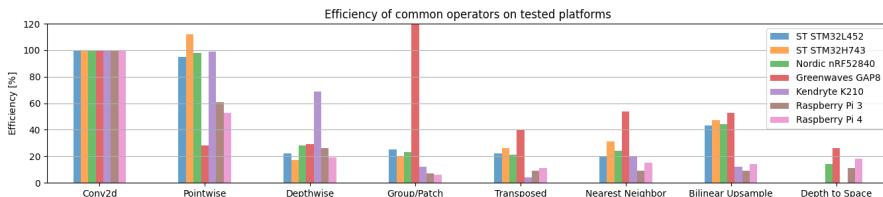
Empirical evaluation of CNN operators...



Empirical evaluation of CNN operators...

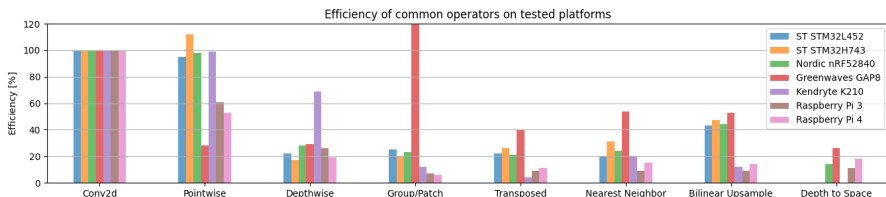


Empirical evaluation of CNN operators...



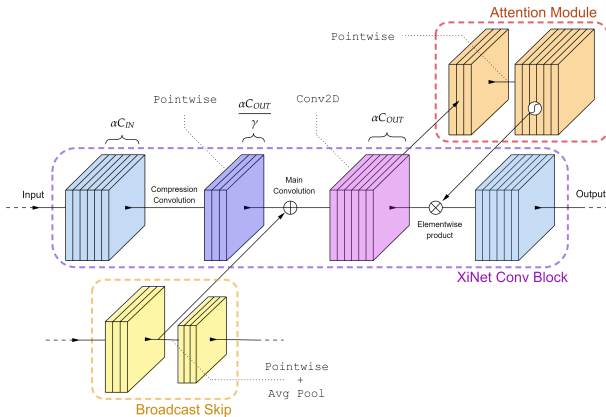
- this suggests that standard convolutions (AlexNet-style) are, on average, more efficient than other variants;

Empirical evaluation of CNN operators...

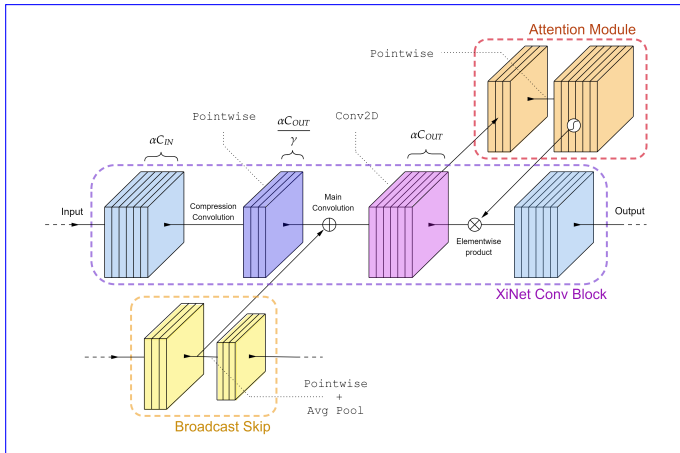


- this suggests that standard convolutions (AlexNet-style) are, on average, more efficient than other variants;
- but how do we exploit them with low parameter memory?

XiNet convolutional block

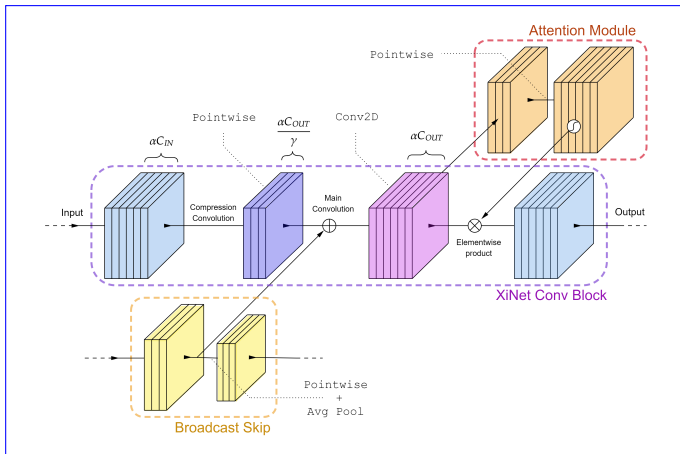


XiNet convolutional block

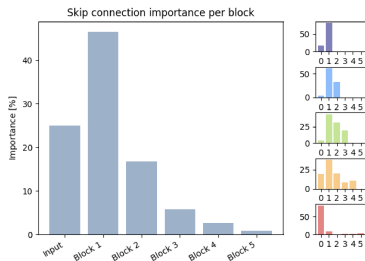


XiNet convolutional block

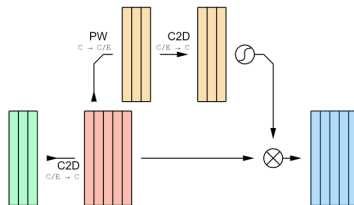
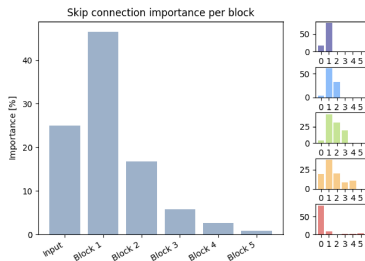
Wide-narrow-wide structure for channels, and much more...



Skip connections and attention block



Skip connections and attention block



- composed by a sequence of XiNet convolutional blocks;

- composed by a sequence of XiNet convolutional blocks;
- similarly to PhiNets, its computational complexity is controlled using **three hyperparameters** (α, γ, β) ;

- composed by a sequence of XiNet convolutional blocks;
- similarly to PhiNets, its computational complexity is controlled using **three hyperparameters** (α, γ, β);
- designed based on the **empirical benchmark** of the different operators to be very efficient;

- composed by a sequence of XiNet convolutional blocks;
- similarly to PhiNets, its computational complexity is controlled using **three hyperparameters** (α, γ, β);
- designed based on the **empirical benchmark** of the different operators to be very efficient;

```
from micromind.networks import XiNet
```

Hardware-aware scaling

- **scaling strategy** that exploits the advanced PhiNets and XiNet architectures;
- helps deploy CNNs on a wide variety of edge platforms via its one-shot network optimization procedure;
- **inverts the mapping between computational complexity and hyperparameters** so that it can be solved with a mathematical programming toolkit for specific computational requirements;

1 Introduction

The Five (-1) Ws of tinyML
Challenges of tinyML

2 Neural Network design

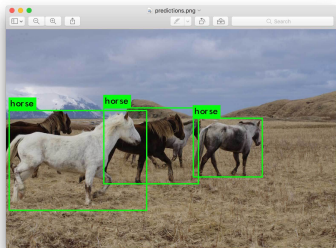
Rise and development of CNNs
tinyML-first CNNs
Hardware-Aware Scaling

3 Some applications...

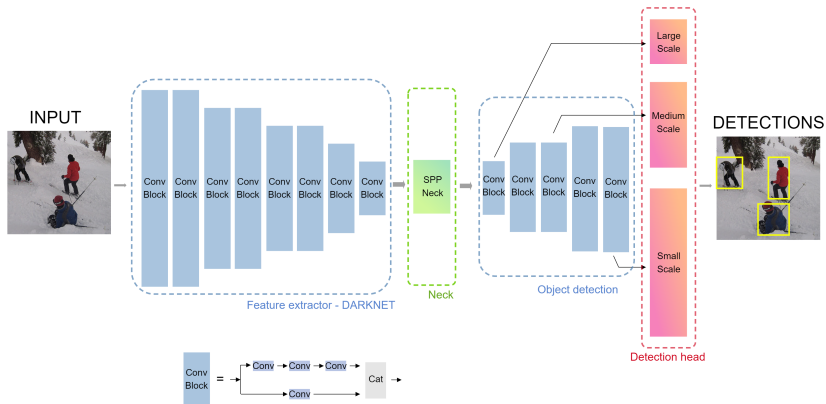
YOLO-based
Zero-shot audio classification
micromind

You Only Look Once (YOLO)

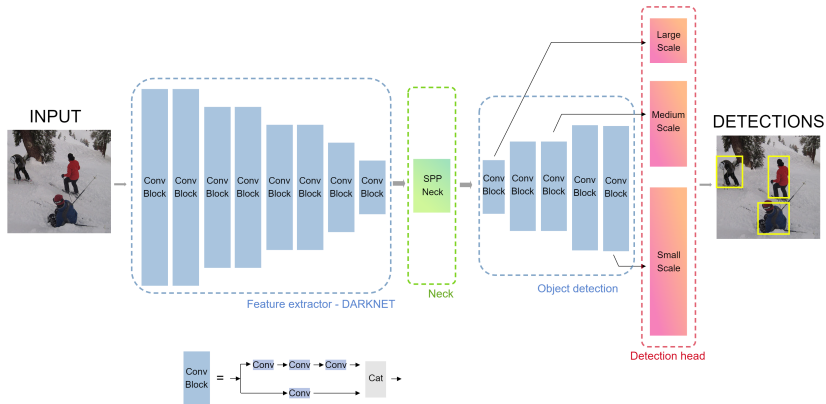
- originally proposed as an object detection pipeline;
- well known for its **good performance/complexity tradeoff**;
- mainly related to its ability to detect objects using **only one inference step** (no region proposal networks, etc...);
- recently extended to support image segmentation, keypoint detection/pose estimation;



YOLO Architecture

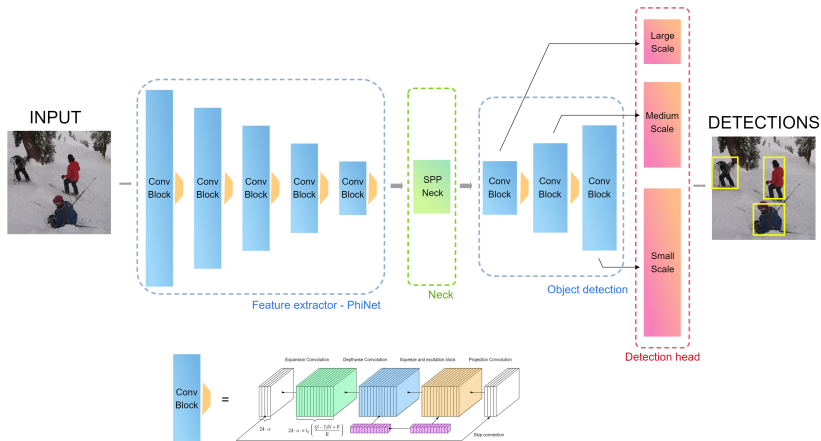


YOLO Architecture

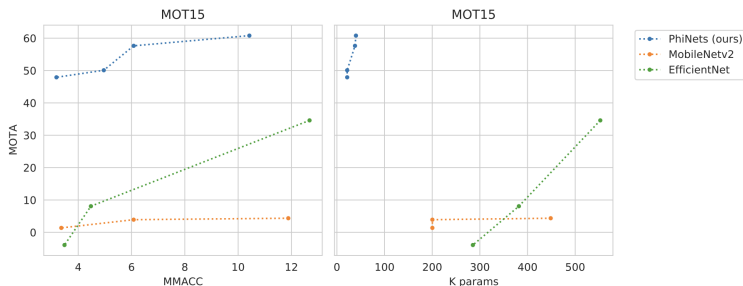


In the literature, some works propose to solve a simplified version of the object detection task; thus, reducing computational complexity... but here is what we do:

YOLOPhiNet



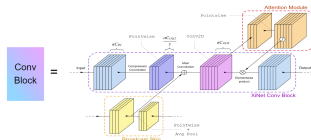
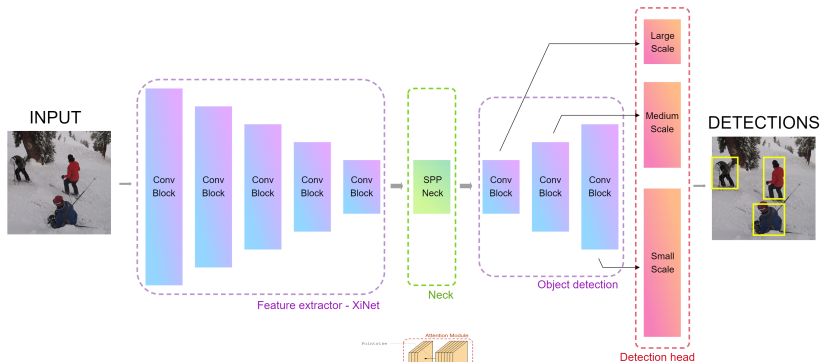
YOLOPhinet

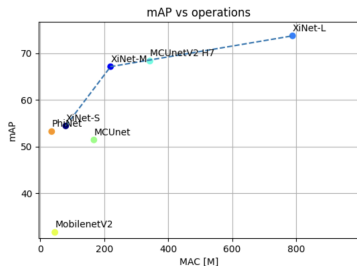
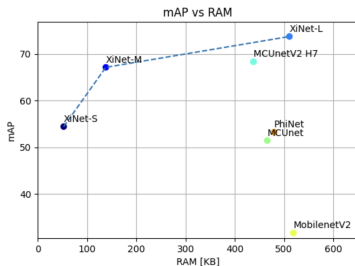


Deployed on an Arm-Cortex M7 MCU with 2 MB of internal Flash and 1 MB of RAM; achieves **power requirements in the order of 10 mW @ 52% mAP on VOC2012.**

`micromind/recipes/object_detection`

YOLOXiNet





Deployed on an Arm-Cortex M7 MCU with 2 MB of internal Flash and 1 MB of RAM; Achieves a reduction in the **number of operations of 2×** and a reduction in **RAM usage of 9×** with respect to MCUNet, with the same performance. Achieves a **power consumption of around 20 mW @ 67% mAP on VOC2012**.

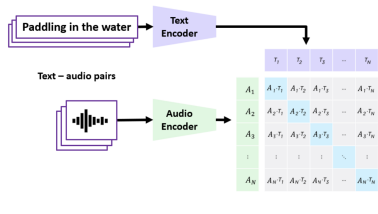
`micromind/recipes/object_detection`

Contrastive Language-Audio pretraining

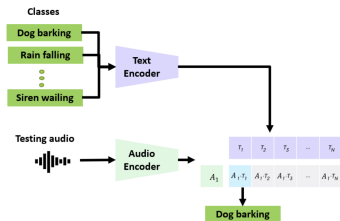
- learns a **similarity score** between two modalities (audio and text);
- can be exploited for **zero-shot** classification;
- makes the network very **flexible** wrt the applications scenario they can be deployed to;

Zero-shot classification

1. Contrastive Pretraining



2. Use pretrained encoders for zero-shot prediction in a new dataset or task



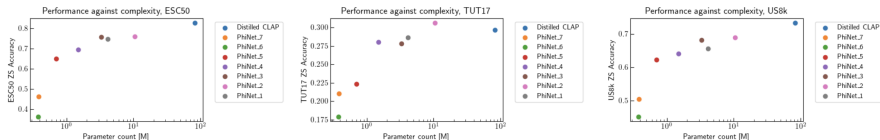
- exploits the learned similarity score to learn a **more efficient audio network** (via a distillation process);

- exploits the learned similarity score to learn a **more efficient audio network** (via a distillation process);
- assumes the pre-trained **text encoder** does **not** need to be **deployed**;

- exploits the learned similarity score to learn a **more efficient audio network** (via a distillation process);
- assumes the pre-trained **text encoder** does **not** need to be **deployed**;
- achieves good performance-complexity tradeoff for ZS classification, and state-of-the-art for a benchmark;

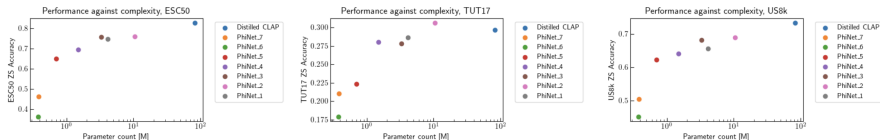
`micromind/recipes/tinyCLAP`

tinyCLAP: performance



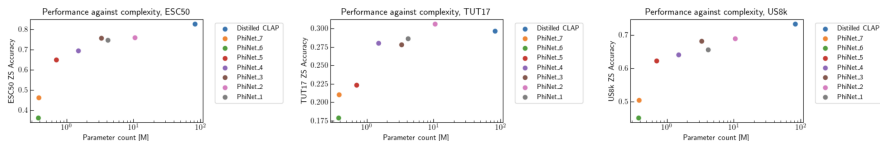
- follows a common power-law scaling behaviour;

tinyCLAP: performance



- follows a common power-law scaling behaviour;
- was not yet deployed on edge platforms (WIP);

tinyCLAP: performance



- follows a common power-law scaling behaviour;
- was not yet deployed on edge platforms (WIP);
- **94% reduction in parameter count** wrt to original CLAP (from 82M to 4M), with a minor ZS accuracy drop (4% averaged on all benchmarks);

- not a startup or a research project, just an **open-source project** for tinyML research;
- tries to provide the **full research pipeline** for model design, development, and deployment;

Checkout the project on GitHub and leave a star!

Follow me on X @fpaissan_ for updates.

Additional references to our works

Following is a list of references to works related to the topics discussed in the presentation:

- Video processing: Ancilotto, Paissan, and Farella, “On the Role of Smart Vision Sensors in Energy-Efficient Computer Vision at the Edge”; Paissan, Ancilotto, and Farella, “PhiNets: A Scalable Backbone for Low-power AI at the Edge”; Ancilotto, Paissan, and Farella, “XiNet: Efficient Neural Networks for tinyML”
- Generative modeling: Ancilotto, Paissan, and Farella, “PhiNet-GAN: Bringing real-time face swapping to embedded devices”; Ancilotto, Paissan, and Farella, “XimSwap: many-to-many face swapping for TinyML”
- Audio processing: Paissan et al., “Scalable Neural Architectures for End-to-End Environmental Sound Classification”; Brutti et al., “Optimizing PhiNet architectures for the detection of urban sounds on low-end devices”; Ali et al., “Scaling strategies for on-device low-complexity source separation with Conv-Tasnet”; Paissan et al., “Improving latency performance trade-off in keyword spotting applications at the edge”
- Multimodal processing: Paissan and Farella, “tinyCLAP: Distilling Contrastive Language-Audio Pretrained Models”

The End

Questions? Comments?



Ali, Mohamed Nabih et al. "Scaling strategies for on-device low-complexity source separation with Conv-Tasnet". In: *ArXiv abs/2303.03005* (2023). URL: <https://api.semanticscholar.org/CorpusID:257364800>.



Ancilotto, A., F. Paissan, and Elisabetta Farella. "XiNet: Efficient Neural Networks for tinyML". In: *ICCV2023* (2023). URL: https://openaccess.thecvf.com/content/ICCV2023/papers/Ancilotto_XiNet_Efficient_Neural_Networks_for_tinyML_ICCV_2023_paper.pdf.



Ancilotto, Alberto, Francesco Paissan, and Elisabetta Farella. "On the Role of Smart Vision Sensors in Energy-Efficient Computer Vision at the Edge". In: *2022 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)* (2022), pp. 497–502. URL: <https://api.semanticscholar.org/CorpusID:248546511>.



— . "PhiNet-GAN: Bringing real-time face swapping to embedded devices". In: *2023 IEEE International Conference on Pervasive Computing and Communications Workshops and other*