Research Project Proposal:

3D TinyML for Video Streaming Analysis

Hazem Shalby
Hazemhesham.shalby@mail.polimi.it
CCS Track

POLITECNICO
MILANO 1863

HONOURS PROGRAMME
HP-SR
in Information Technology

3D TinyML for Video Streaming Analysis

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?

3D TinyML for Video Streaming Analysis

Tiny machine learning is the field of **machine learning** technologies including hardware, algorithms and software capable of performing on-device analytics at extremely low power (≈ mW), enabling a variety of always-on use-cases and targeting battery operated devices.

- ✓**Increase autonomy**
- ✓**Reduce decision-making latency**
- ✓**Reduce transmission bandwidth**
- ✓**Increase energy-efficiency**
- ✓**Security and Privacy**
- ✓**Incremental/Adaptive Learning**
- ✓**Ecosystem of units**

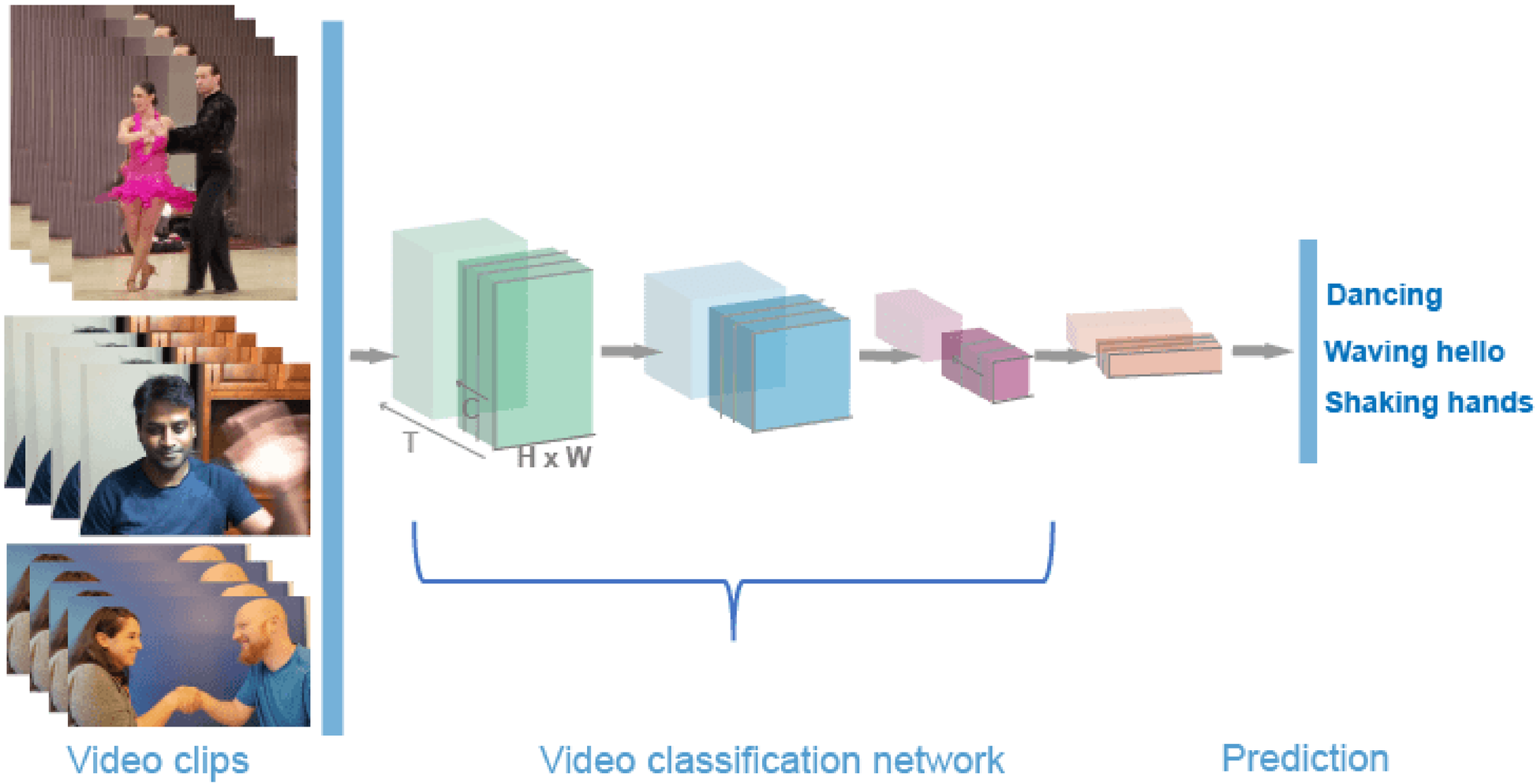
- ➢**Low computing ability**
- ➢**Constraints on energy**
- ➢**Constraints on memory (RAM/FLASH)**
- ➢**Complexity in design and development**
- ➢**Strong connection between HW, SW and ML**

3D TinyML for Video Streaming Analysis

3D TinyML for Video Streaming Analysis

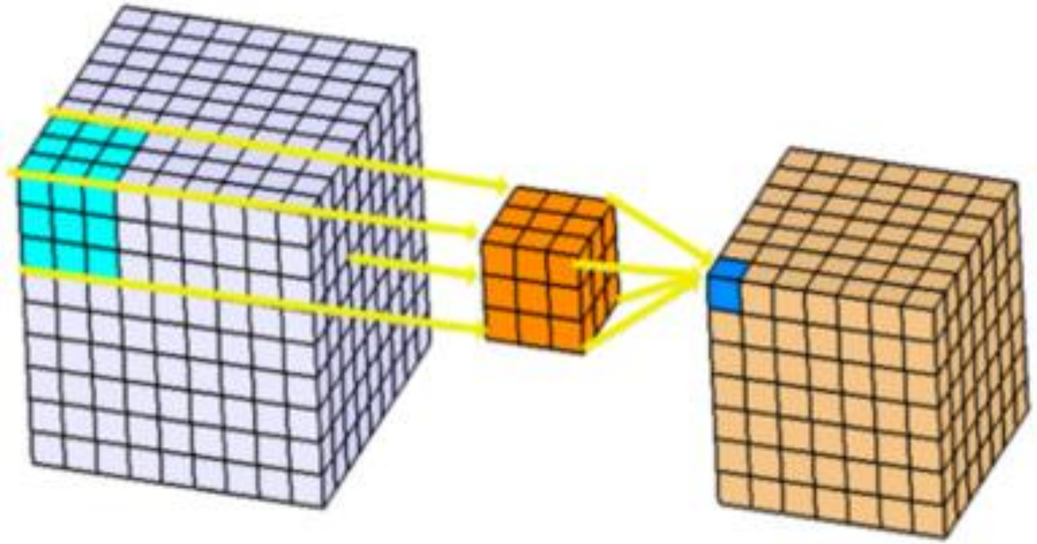
Being $x_t \in U \subset R^{M \times N \times C}$ (M, N, and C are the sizes of the input) a frame of the video streaming and $y_t \in Y = \{\Omega_1, \dots, \Omega_c\}$ a label associated to x_t, the goal is to construct a classifier $f_\theta\ (x_t, x_{t-1}, x_{t-2}, \dots)$ able to map an unseen data $\bar{x}_i$ to its label $\bar{y}_i$.

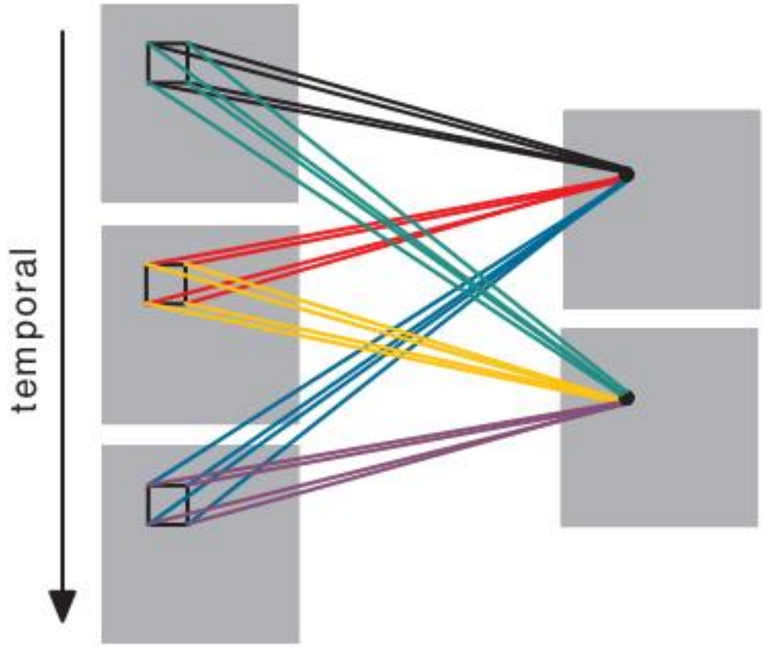
3D TinyML for Video Streaming Analysis



3D TinyML for Video Streaming Analysis

3D convolutions applies a 3-dimentional filter to the dataset and the filter moves 3-direction (x, y, z) to calculate the low-level feature representations. They are helpful in **event detection in videos**.



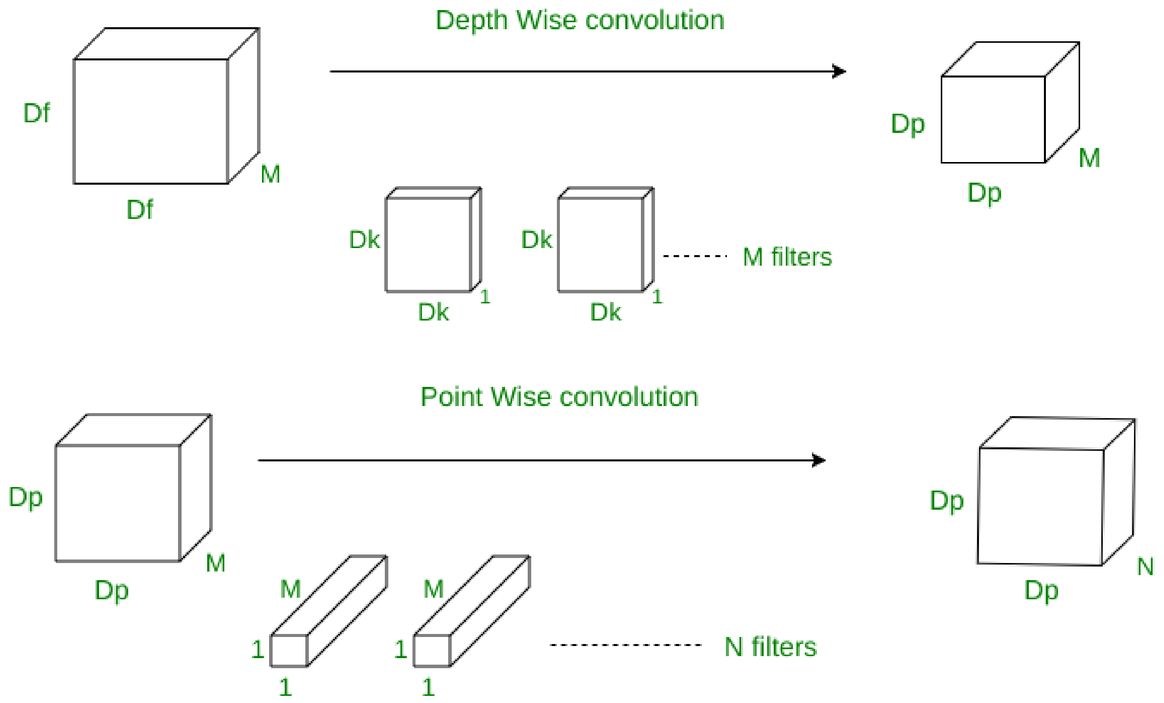


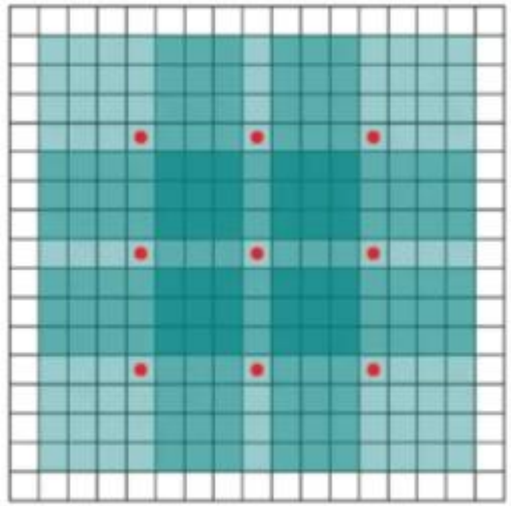
Research Steps

- Redesigning the CNN architecture
- Introducing approximate computing mechanisms
- Exploiting embedded-system code optimization

Research Steps: CNN architecture

Explore existing and efficient ways of doing the standard 2D convolutions (e.g. **separable convolutions** and **dilated convolutions**) and combine them with **3D convolutions**, which are suitable for the multiple frame analysis task but implementing them in the TinyML field is challenging.

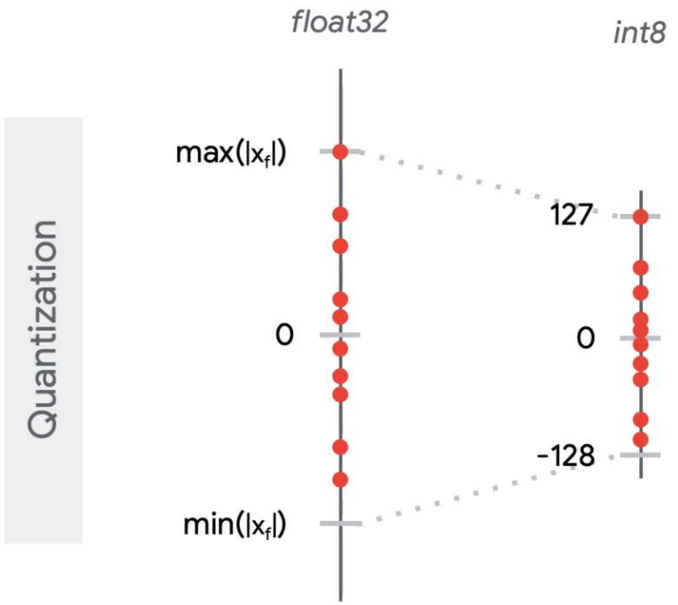


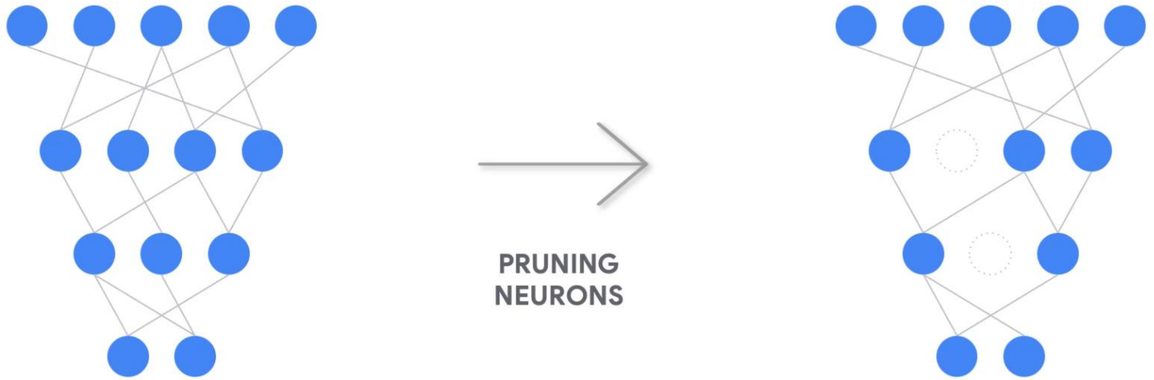


Research Steps: approximation

- **Precision scaling**: reduce the memory occupation of a CNN by changing the precision

- **Task dropping:** reduce the computational load and memory occupation by skipping the execution of certain tasks associated with the processing pipeline.





Research Steps: Code Optimization

involves the use of **toolchains** or **code optimization** mechanisms to further reduce the computational and memory demands for a target hardware platform

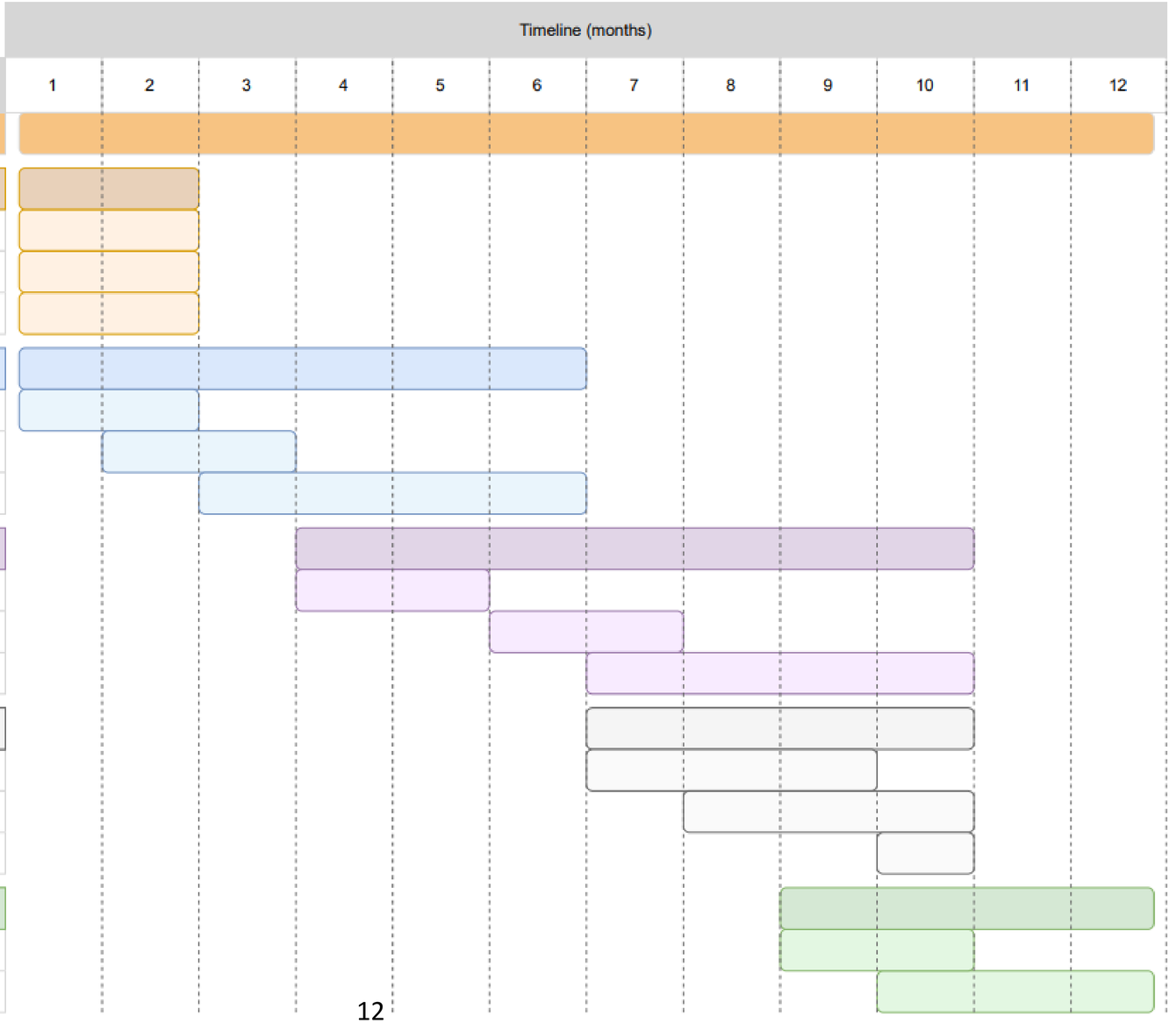
(Micro)

CMSIS-NN

CMSIS NN Software Library

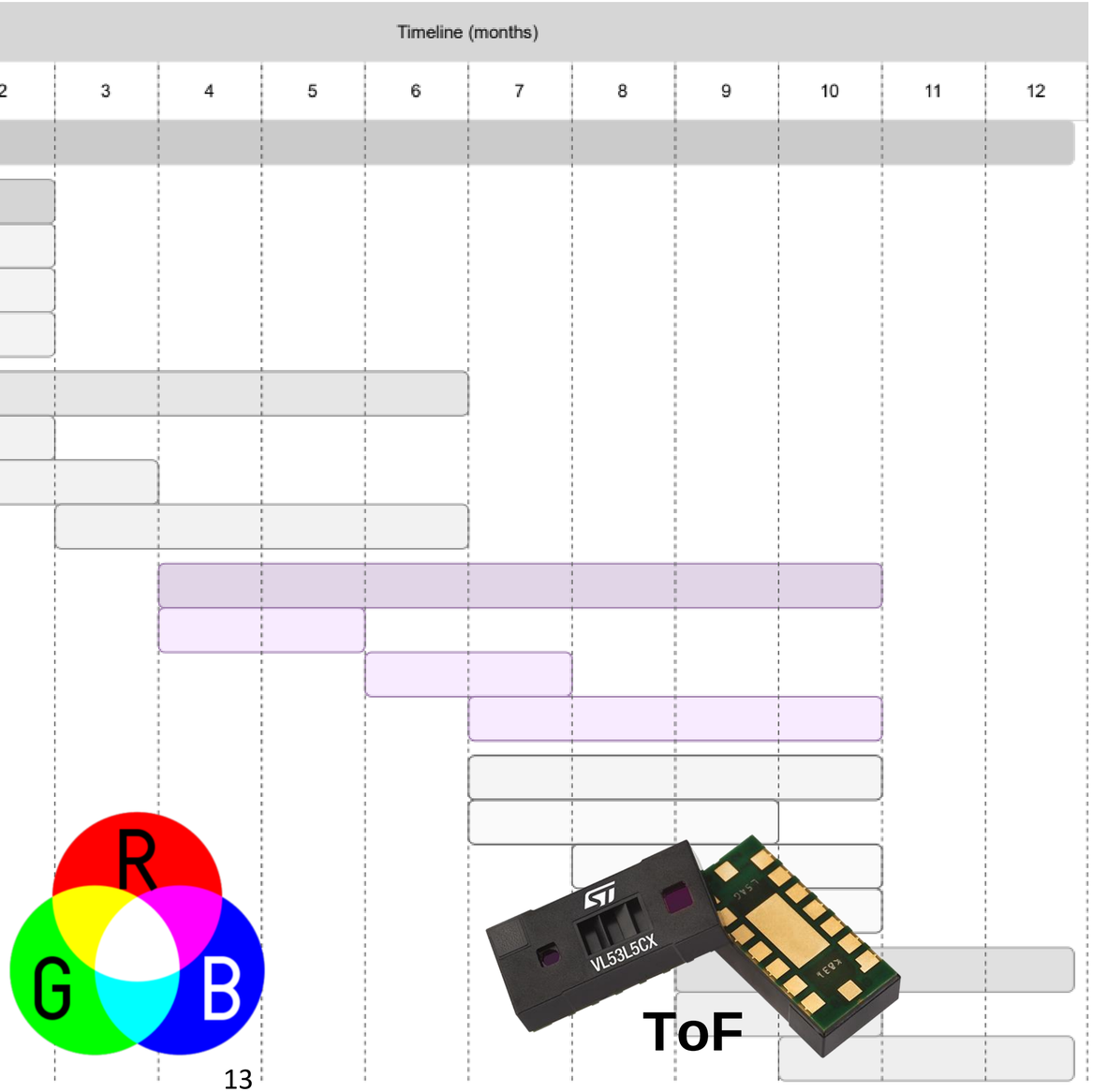
Research Plan

#	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



Research Plan

#	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



ToF

Research Assessment

The accuracy of the classifier

The number of multiple contiguous frames used by the classifier

The time needed for the inference

Memory occupation

The number FLOPs and MACs operations needed

Thanks for your attention!