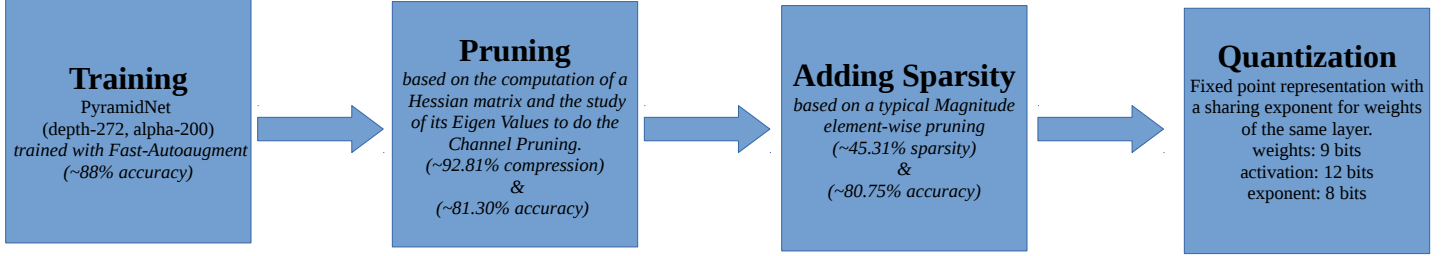


# Micronet Challenge Report:

*first submission & second submission*

Here is a schema describing our first submission:



This **first** submission gives us a final micronet score of : **0.0467**. For a corresponding accuracy of **80.15%**.  
This **second** submission gives us a final micronet score of : **0.0464**. For a corresponding accuracy of 80.47%.  
The **third submission** gives us a final micronet score of : **0.0456**. For a corresponding accuracy of 80.003%

The only difference between the first submission and the two other ones is the fact that in the first one we don't merge the BatchNorm layers with the Convolutional layers.

Also, in the case of the second submission:  $MultiBits=10$

And in the third submission:  $MultiBits=9$

As the third submission is borderline we wanted to get another submission working for sure.

To compute the micronet score we followed the following steps.

Storage:

$$Storage = \frac{TotalNonZeroParameters * ParamBits}{32} + \frac{TotalParameters}{32}$$

Math Ops:

$$MathOps = \frac{TotalFlopMults * MultiBits}{32} + \frac{TotalFlopAdds * AddBits}{32} + \frac{TotalFlopExponent * ExponentBits}{32}$$

With:

- $ParamBits=9$
- $MultiBits=12$
- $AddBits=32$
- $ExponentBits=8$

The values  $TotalFlopMults$  and  $TotalFlopAdds$  were computed following the code given by the micronet challenge. Please see the file *Micronet/Test/compute\_flops.py* for implementation details.

Score:

$$score = \frac{MathOps}{10.49 B} + \frac{Storage}{36.5 M}$$

Please see the file *Micronet/Test/main.py* for overall implementation details

If one wants to reproduce all the results, please follow the indications given in the different README files.  
The order of execution should be :

- 1/ Training
- 2/ Pruning
- 3/ Sparsity
- 4/ Quantization
- 5/ Test

**To only obtain the final micronet score run the file *MicroNet/Test/main.py* without changing anything.**