Artificial Intelligence on RISC-V processor

Made by:

Mohamed Belouarga
Abderrazak El Asfar
Safae Ouajih
Ghita El Moussi

supervised by :

M.Patrice Kadionik

Bordeaux INP
ENSEIRB
MATMECA

What is artificial intelligence ?

Bordeaux INP
ENSEIRB
MATMECA

convolutional neural network

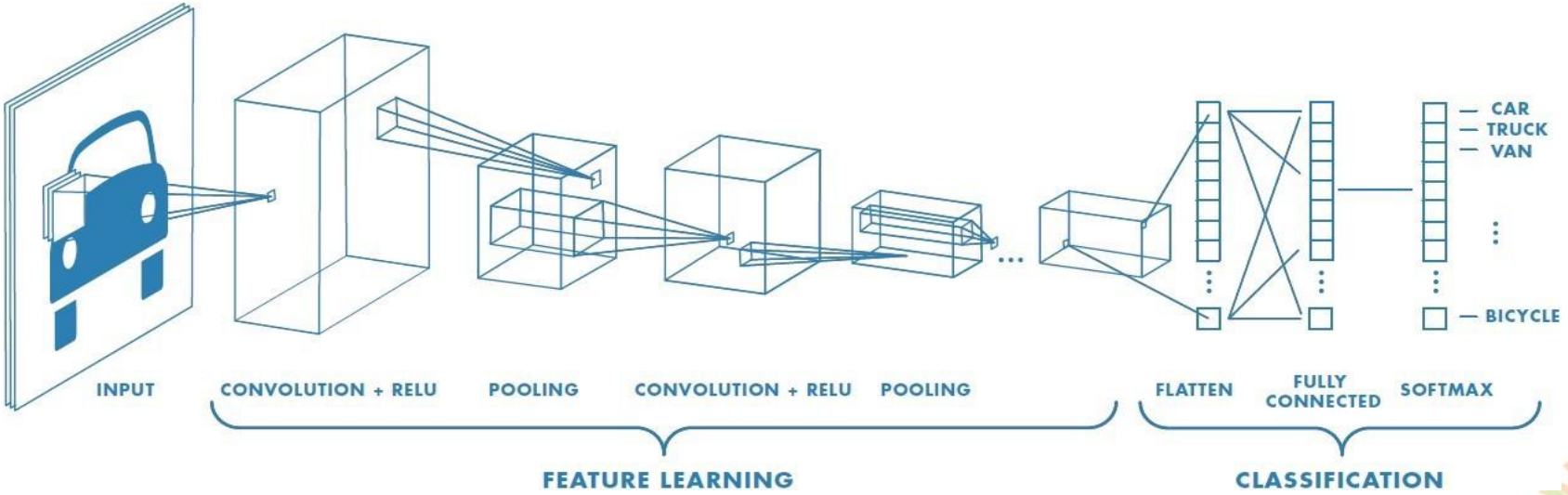


figure 1: example of convolutional neural network

Bordeaux INP
ENSEIRB
MATMECA

YOLO

YOLO is a network for object detection

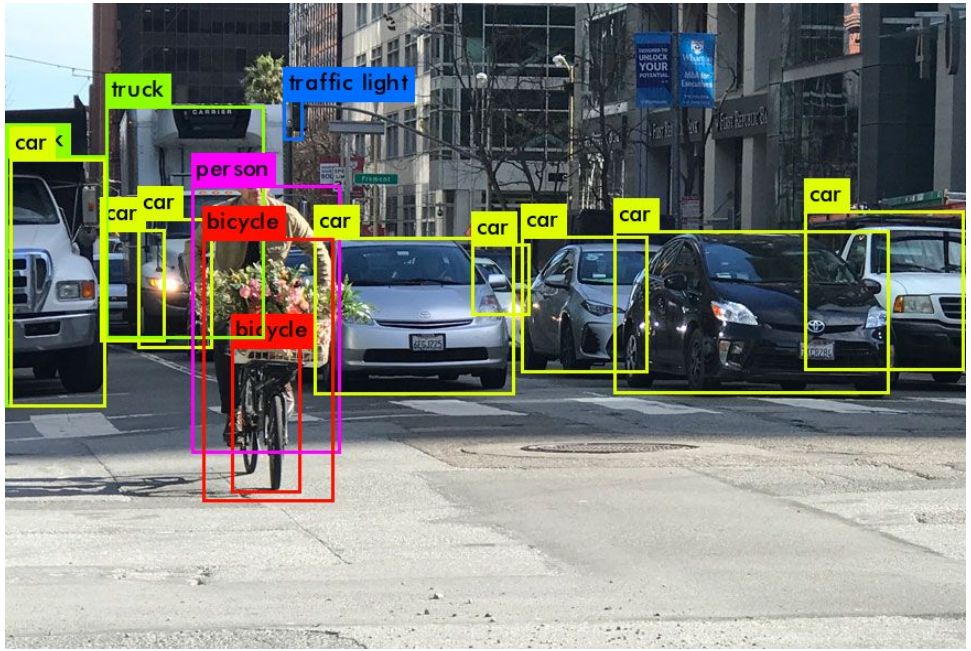


figure 2: example of object detection

The prediction vector:

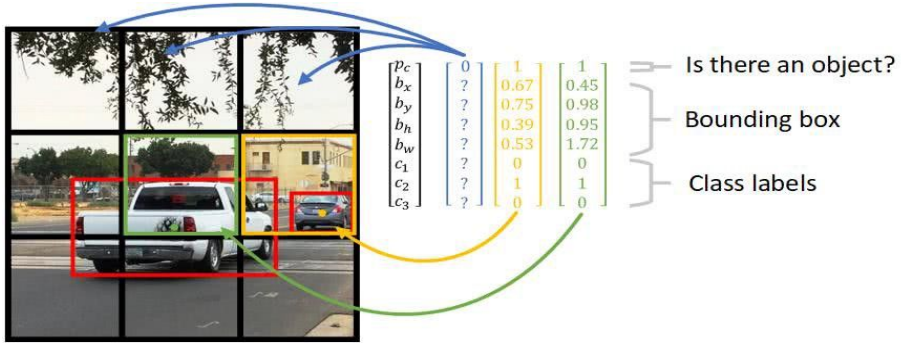


figure 3: Yolo prediction

Non-Max suppression:

figure 4: Non-Max filtration

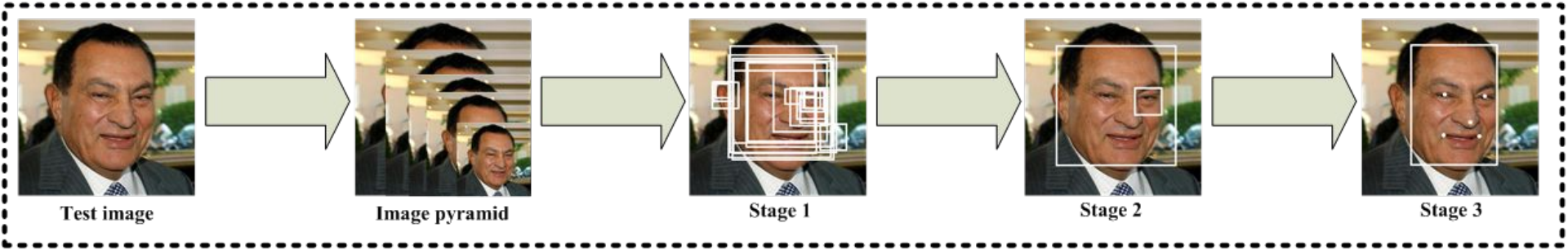
-Selecting the box

-compute its overlap

-Iteration

Bordeaux INP ENSEIRB MATMECA

MTCNN



The first step:

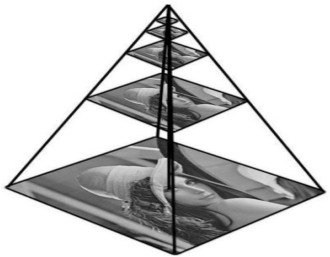


figure 5: Construction images

The three stages of MTCNN:

P-onet: To produce candidate box rapidly

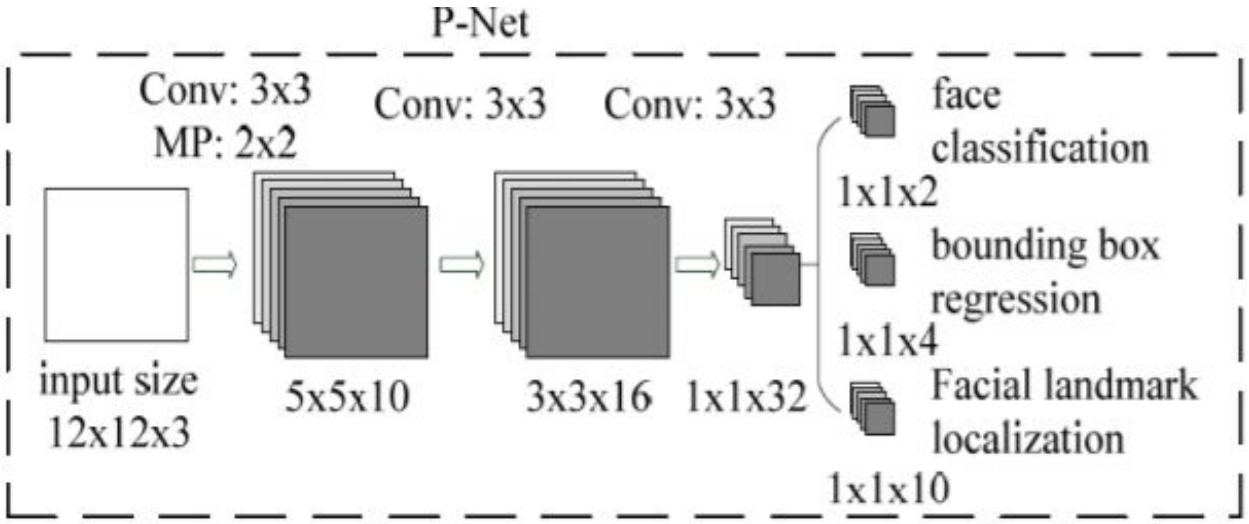


figure 6: Proposal network

R-onet: filtering for picking up the candidate box

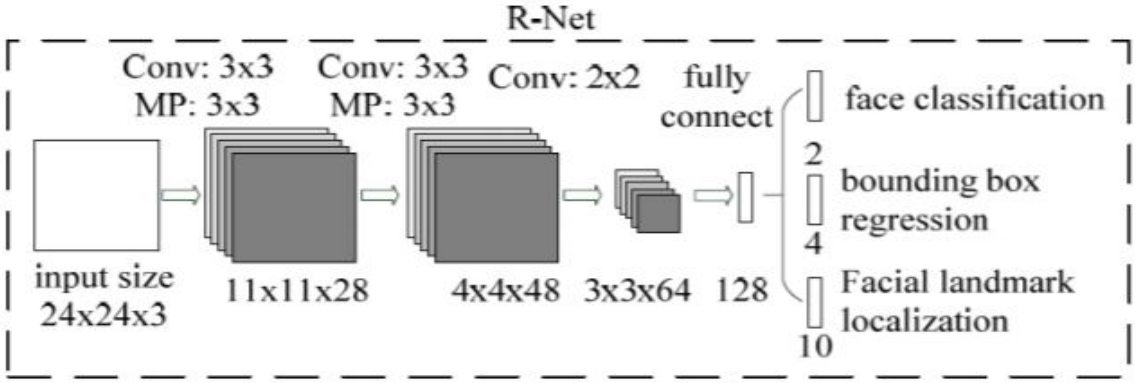


figure 7: Refine network

O-net: generating the boundary box and the five facial features

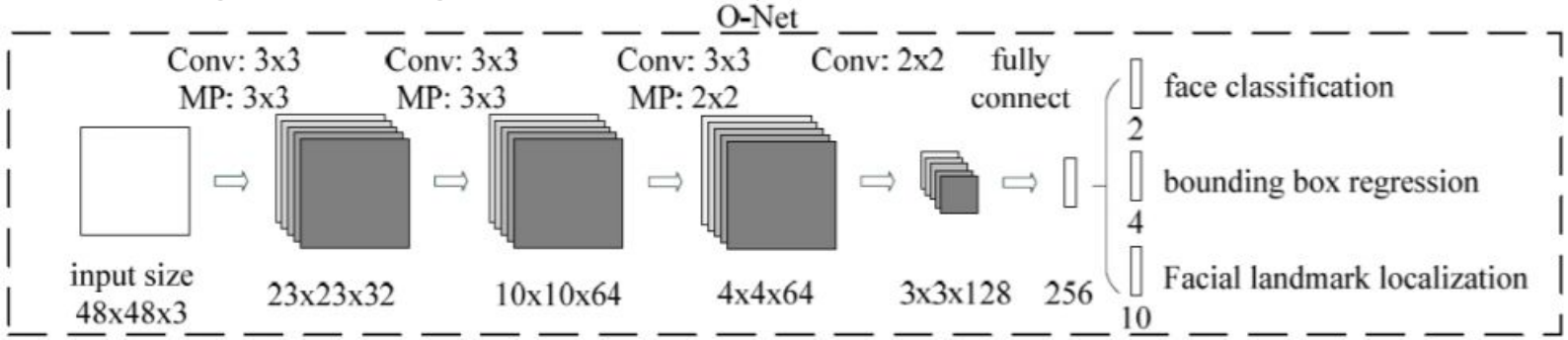


figure 8: Output network

Implementation and Test:

figure 9: Test with our CNN

Bordeaux INP
ENSEIRB
MATMECA

Sipeed Maixduino microcontroller

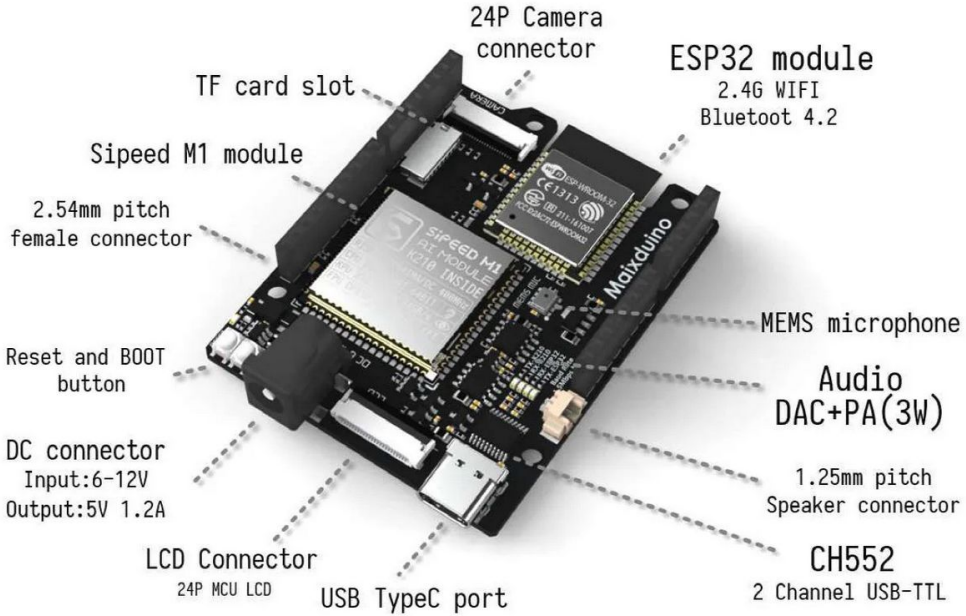


figure 10: sipeed Maixduino board

- Based on K210 RISC-V SoC:
  - -Dual core RISC-V 64 bit processor with FPU 400Mhz
  - - KPU CNN Accelerator
  - - 8 MB RAM
- 16 MB Flash-memory
- MicroSD slot

Sipeed Maixduino microcontroller

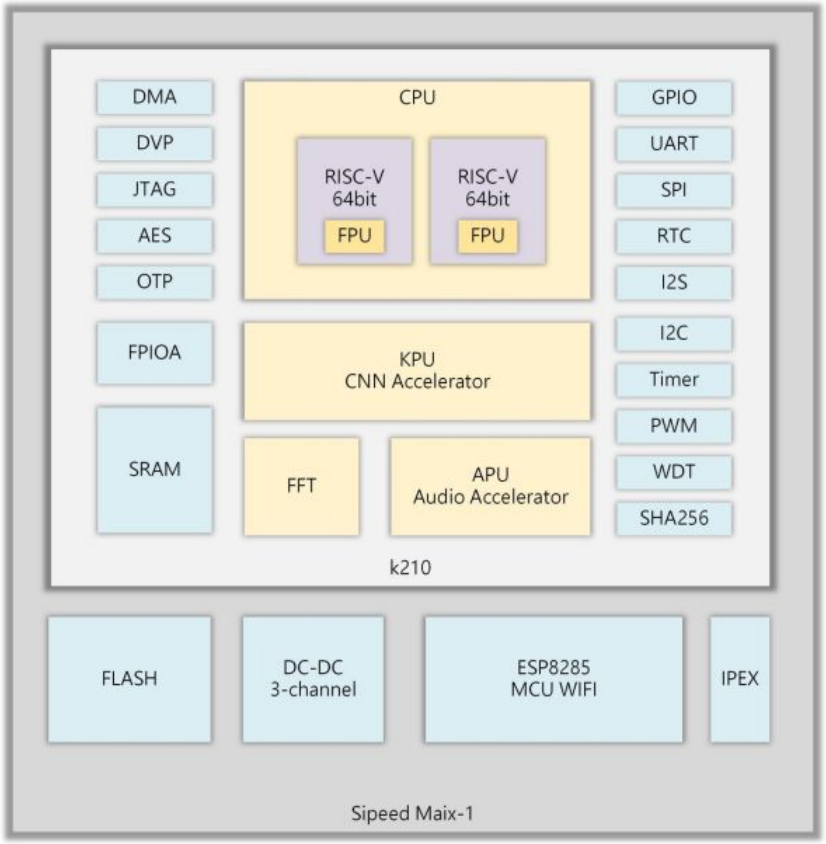


figure 11: sipeed Maix-1 K210

kernel processing unit

Speeds up the processing of the individual kernel functions underlying CNNs :

- hardware implementations of convolution and pooling kernel functions that comprise the individual layers of CNN models

Sipeed Maixduino Supports :
Tiny-Yolo, Mobilenet and TensorFlow-Lite

YOLO's little Brother :

- **442% faster**
- **Small model size**
- ***Less accurate***

MaixPy ported MicroPython to K210 :

- Small subset of the Python standard library
- Optimised to run on microcontrollers in constrained environments

Bordeaux INP
ENSEIRB
MATMECA

Implementation

Supports fixed-point models
No direct limit on the number of network layers
Good FPS : Model size : up to 5.9MB
Slow : Flash capacity - software volume

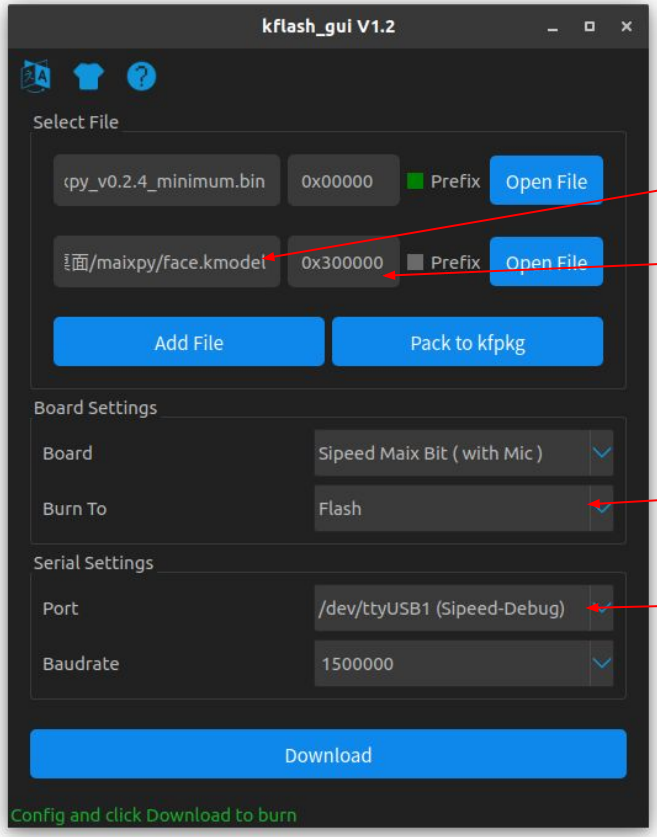
```
import KPU as kpu
task = kpu.load(offset or file_path)
```

The offset of the model in flash
Example : 0x200000

Example : /sd/yolo.kmodel

Bordeaux INP
ENSEIRB
MATMECA

Implementation



.kmodel file

Offset

To Flash memory

Sipeed-Maix port

figure 12: kflash_gui tool

Test

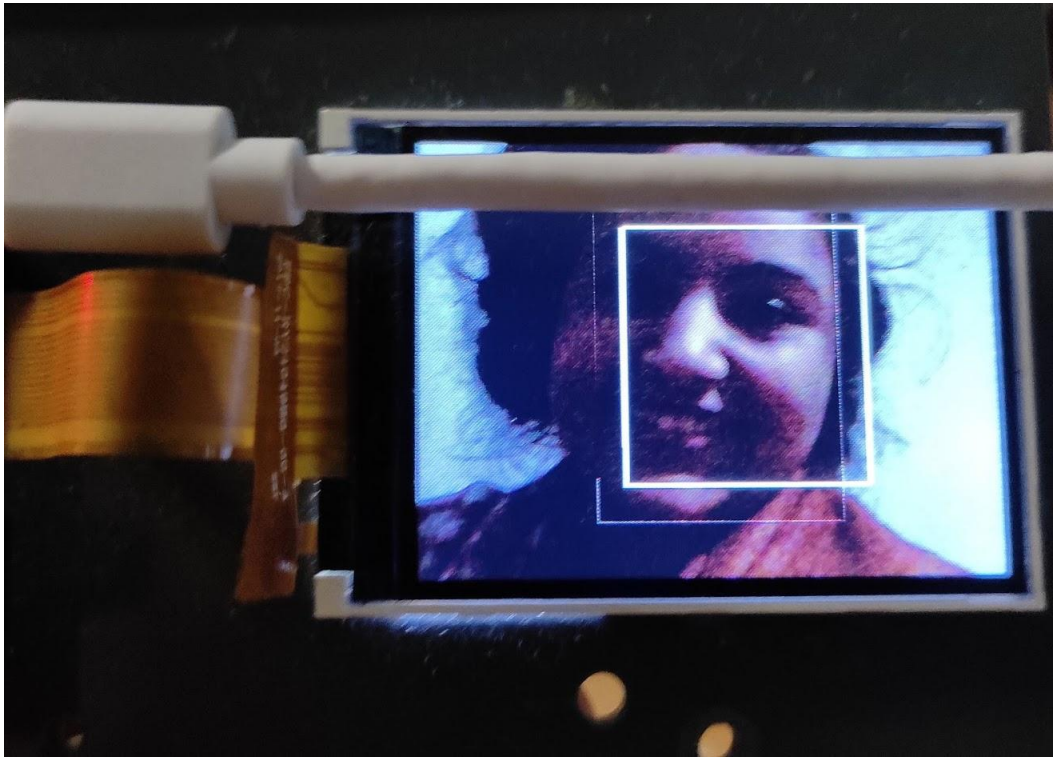


figure 13: Tiny yolo model tested

Bordeaux INP
ENSEIRB
MATMECA

aXeleRate

figure: Dmitry Maslov

figure: Logo of aXeleRate

aXeleRate logic

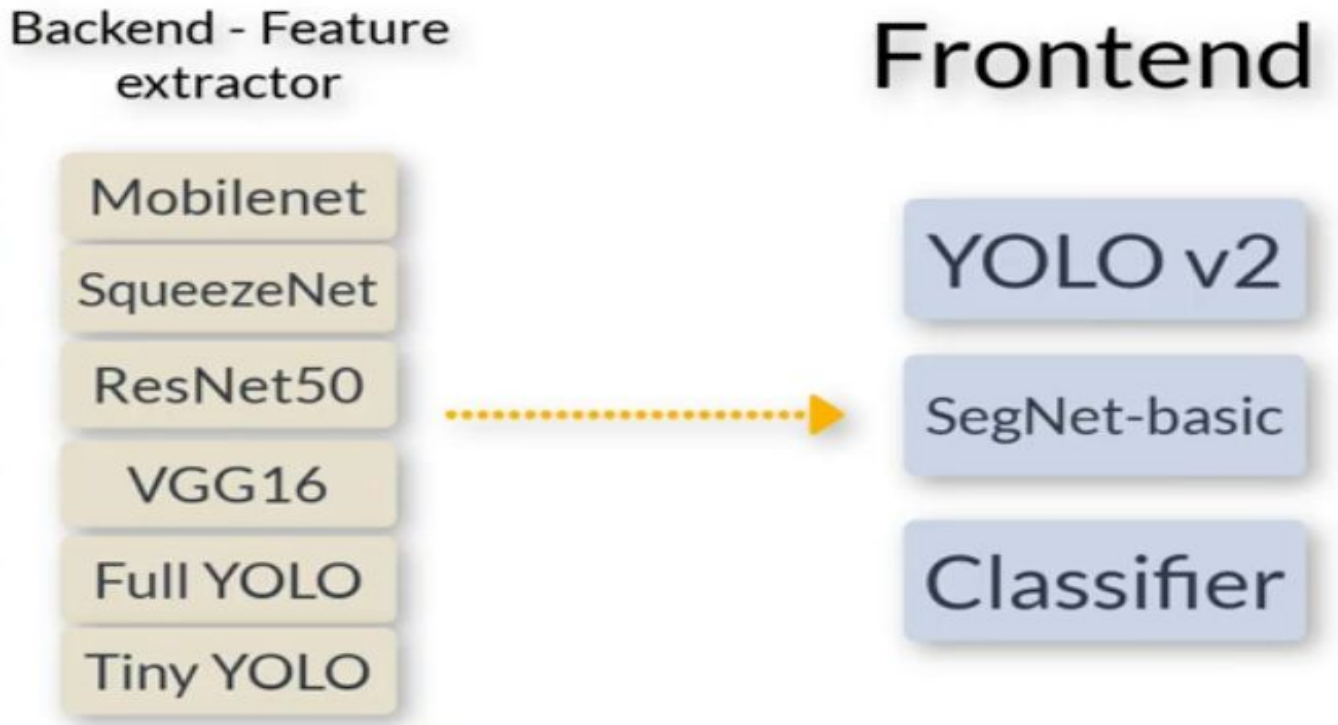


figure 14: Extraction of features

Number of parameters

```
Total params: 1,856,046
Trainable params: 1,839,630
Non-trainable params: 16,416
```

figure 15: Number of parameters used

figure 16: Logo of CelebA dataset

202 599 images with their features

Only 47 349 images were used

due to lack of material

Bordeaux INP
ENSEIRB
MATMECA

Training procedure

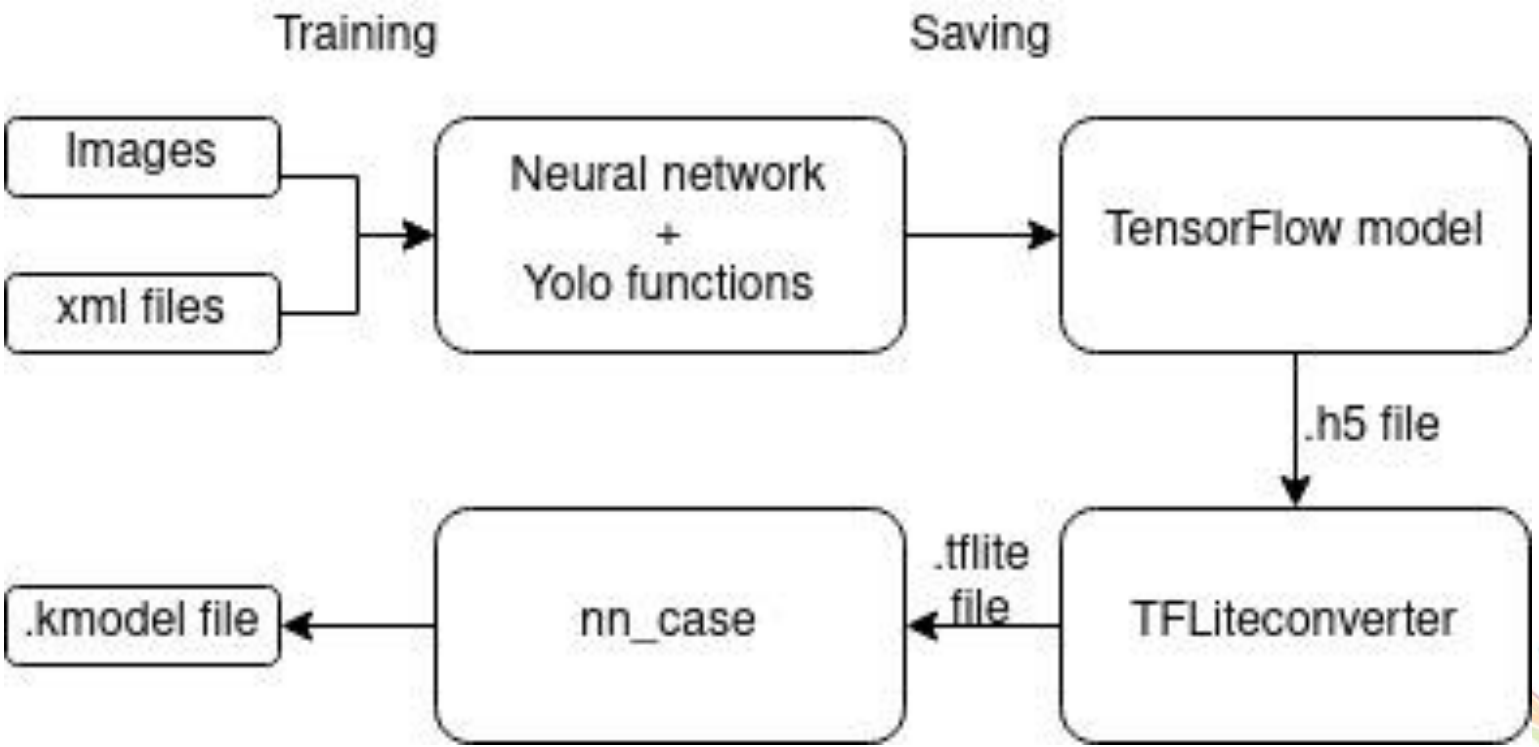


figure17: Procedure of generating the .kmodel file

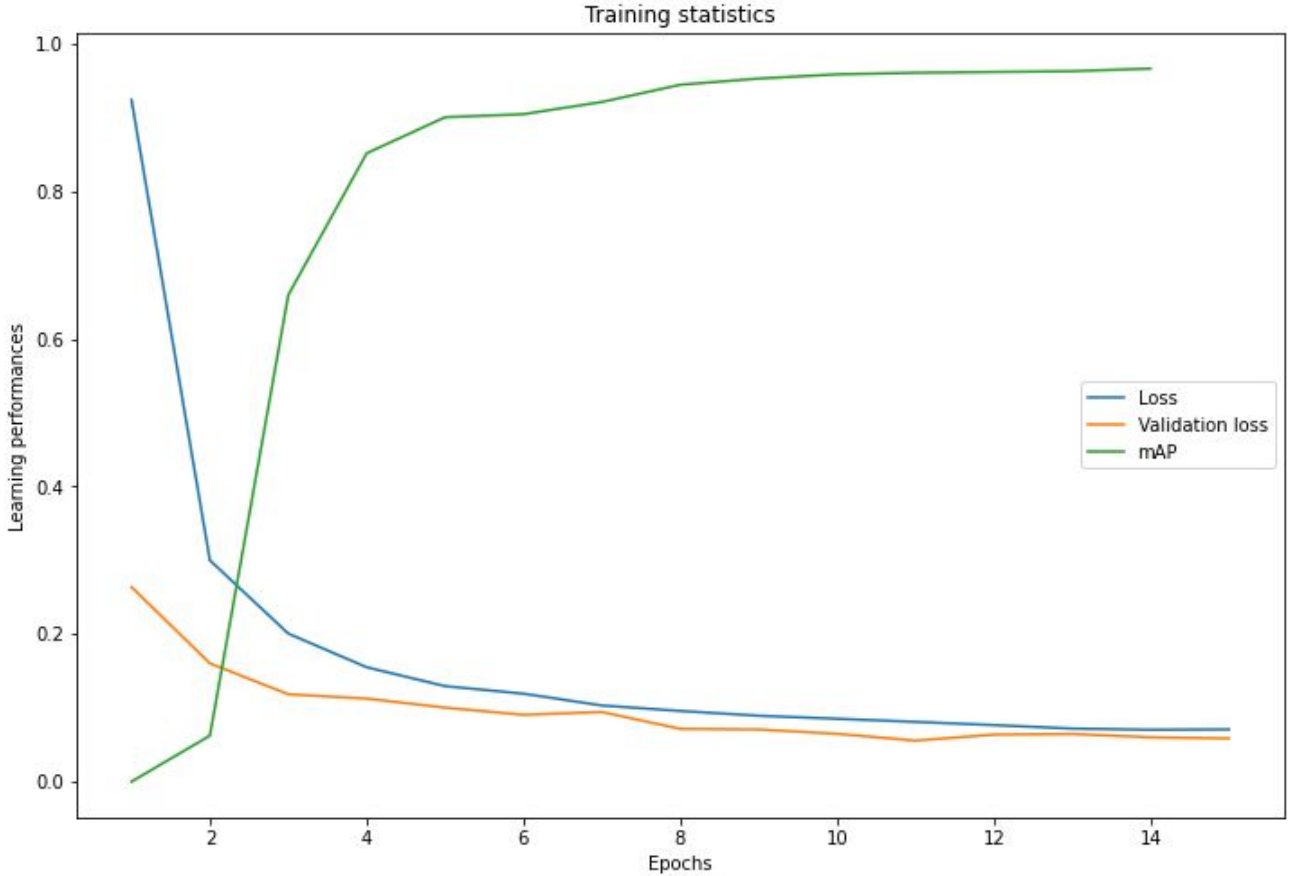


figure 18: performance curves

Conclusion