Machine Learning for IoT

Projet de la VAP SEM

Nimesh TAHALOOA

https://github.com/nimesh-t/pfe-ml-for-iot

Encadrants : Cedric ADJIH & Anis LAOUITI

SOMMAIRE

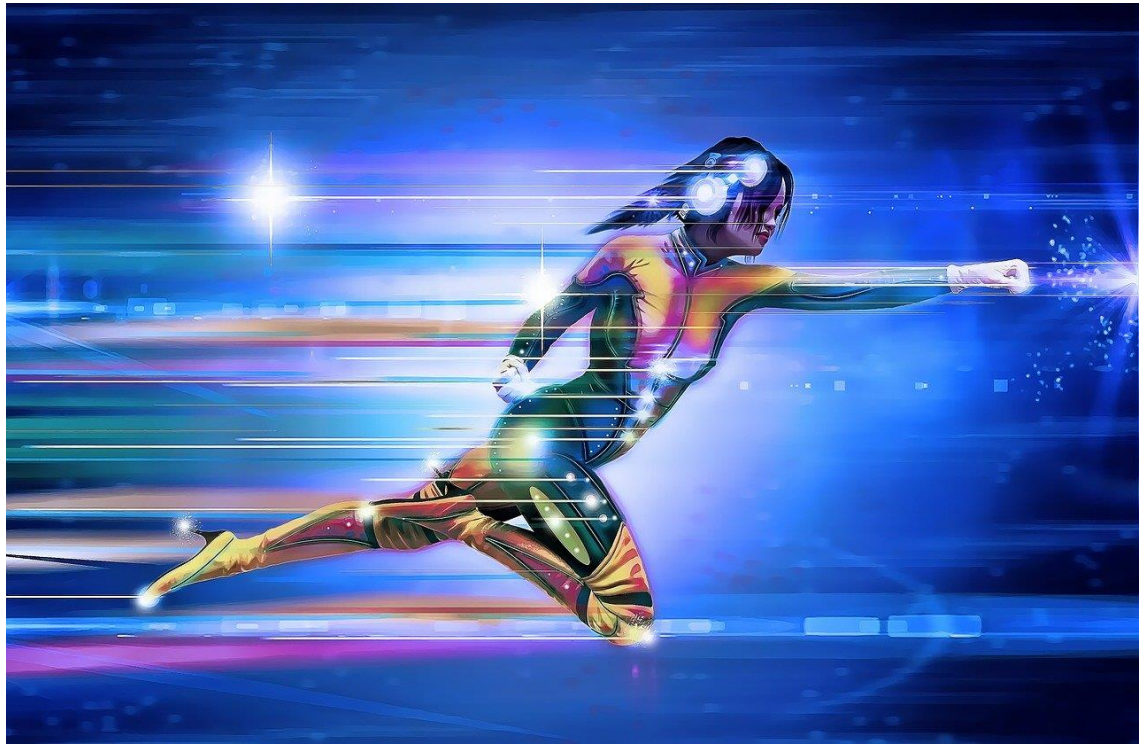
- Objectifs
- Le machine learning
  - Pourquoi l'embarqué ?
  - Un neurone
  - Exemple d'un CNN
- Solutions pour l'embarqué
  - Nvidia Jetson Nano
  - TensorRT
- Développement
- Résultats
- Conclusion

OBJECTIFS

- Développer une application ML sur un système embarqué, la Nvidia Jetson Nano
  - Détection de feu
- Comprendre et implémenter les méthodes d'optimisation utilisées pour le ML pour l'embarqué
- Etudier les algorithmes d'apprentissage distribué ou fédéré

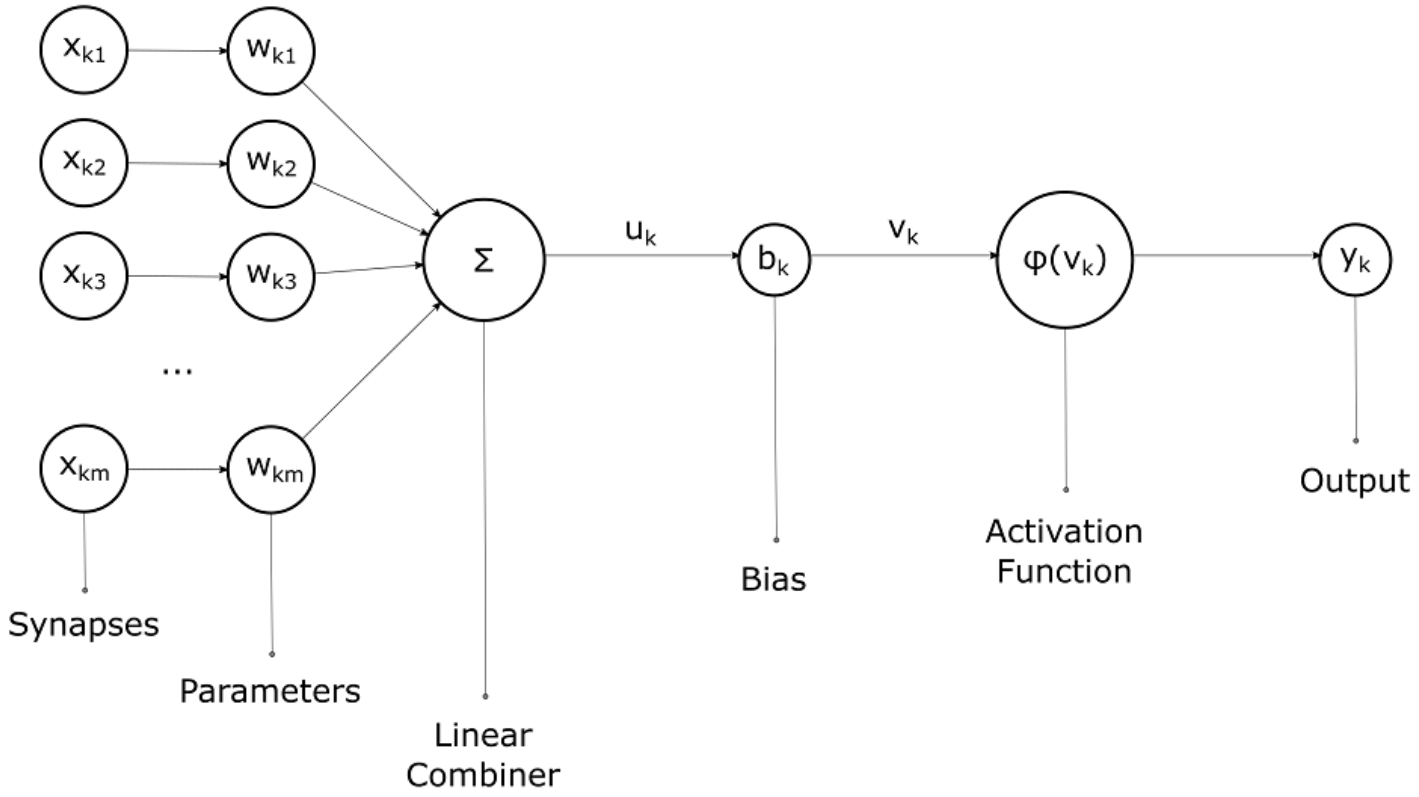
LE MACHINE LEARNING

POURQUOI L'EMBARQUÉ ?



LE MACHINE LEARNING

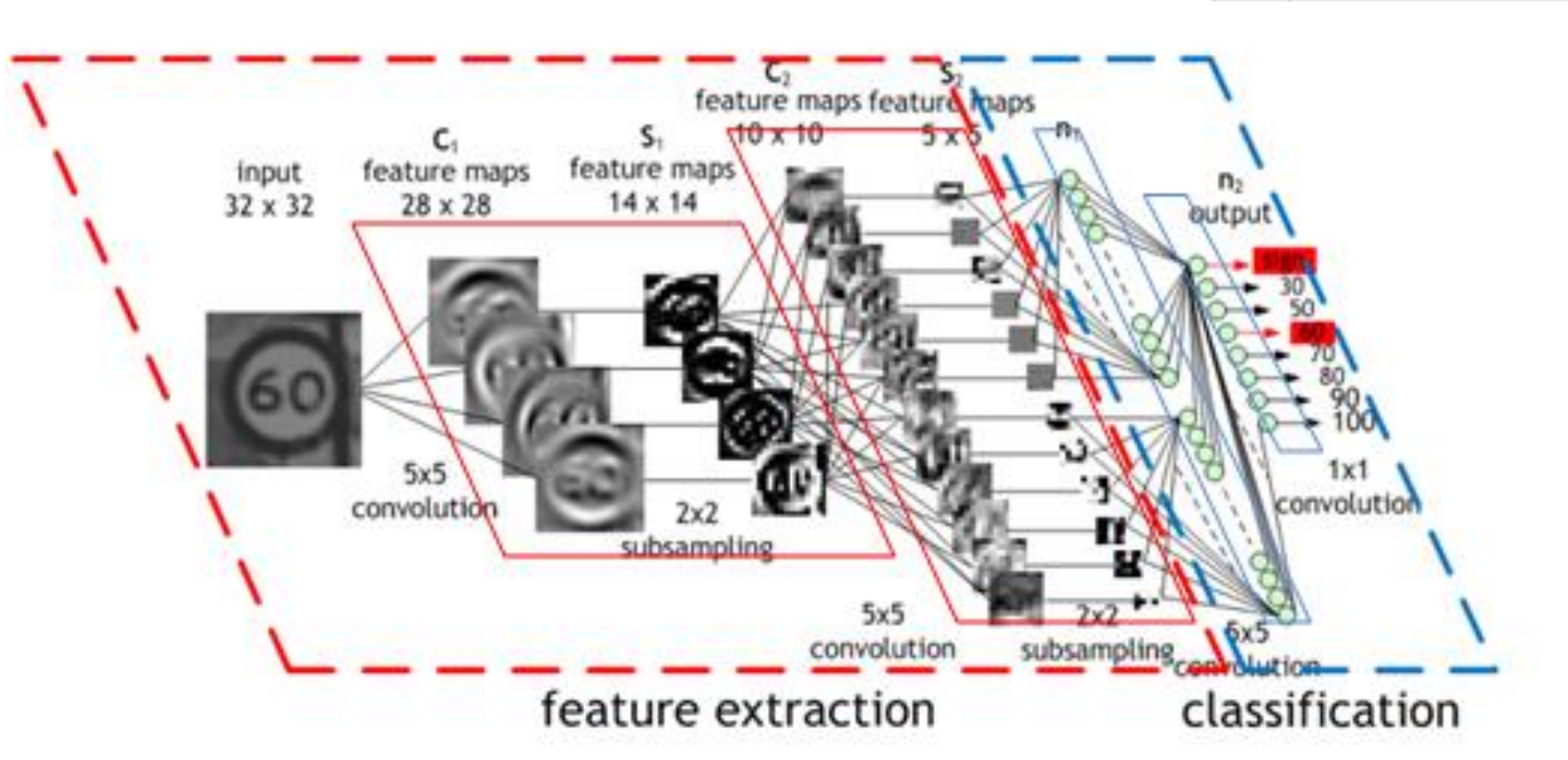
UN NEURONE



LE MACHINE LEARNING

EXEMPLE D'UN CNN

https://developer.nvidia.com/discover/convolutional-neural-network



SOLUTIONS POUR L'EMBARQUÉ

HARDWARE – NVIDIA JETSON NANO

Hello AI World – Jetson Nano – Webinar de dusty_nv

JETSON NANO DEVKIT SPECS

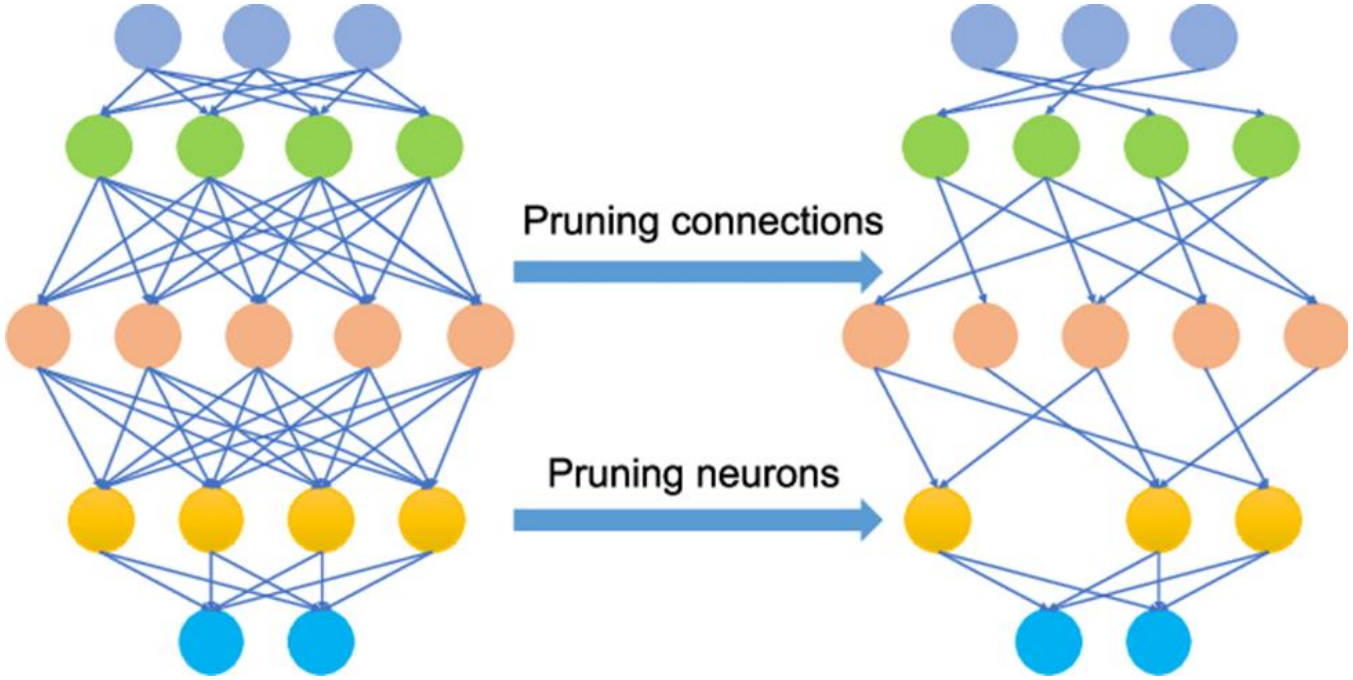
	PROCESSOR
CPU	64-bit Quad-core ARM A57 @ 1.43GHz
GPU	128-core NVIDIA Maxwell @ 921MHz
Memory	4GB 64-bit LPDDR4 @ 1600MHz | 25.6GB/s
Video Encoder	4Kp30 | (4x) 1080p30 | (2x) 1080p60
Video Decoder	4Kp60 | (2x) 4Kp30 | (8x) 1080p30 | (4x) 1080p60

	INTERFACES
USB	(4x) USB 3.0 A (Host) | USB 2.0 Micro B (Device)
Camera	MIPI CSI-2 x2 (15-position Flex Connector)
Display	HDMI | DisplayPort
Networking	Gigabit Ethernet (RJ45, PoE)
Wireless	M.2 Key-E with PCIe x1
Storage	MicroSD card (16GB UHS-1 recommended minimum)
40-Pin Header	UART | SPI | I2C | I2S | Audio Clock | GPIOs
Power	5V DC (µUSB, Barrel Jack, PoE) - 5W | 10W
Size	80x100mm

Distributors Include: a, ARROW, newegg, NVIDIA, seeed, Siliconhighway, sparkfun ELECTRONICS

SOLUTIONS POUR L'EMBARQUÉ

COMPRESSION D'UN RÉSEAU



- Clustering
- Pruning
- Quantification

SOLUTIONS POUR L'EMBARQUÉ

TENSORRT

https://developer.nvidia.com/tensorrt

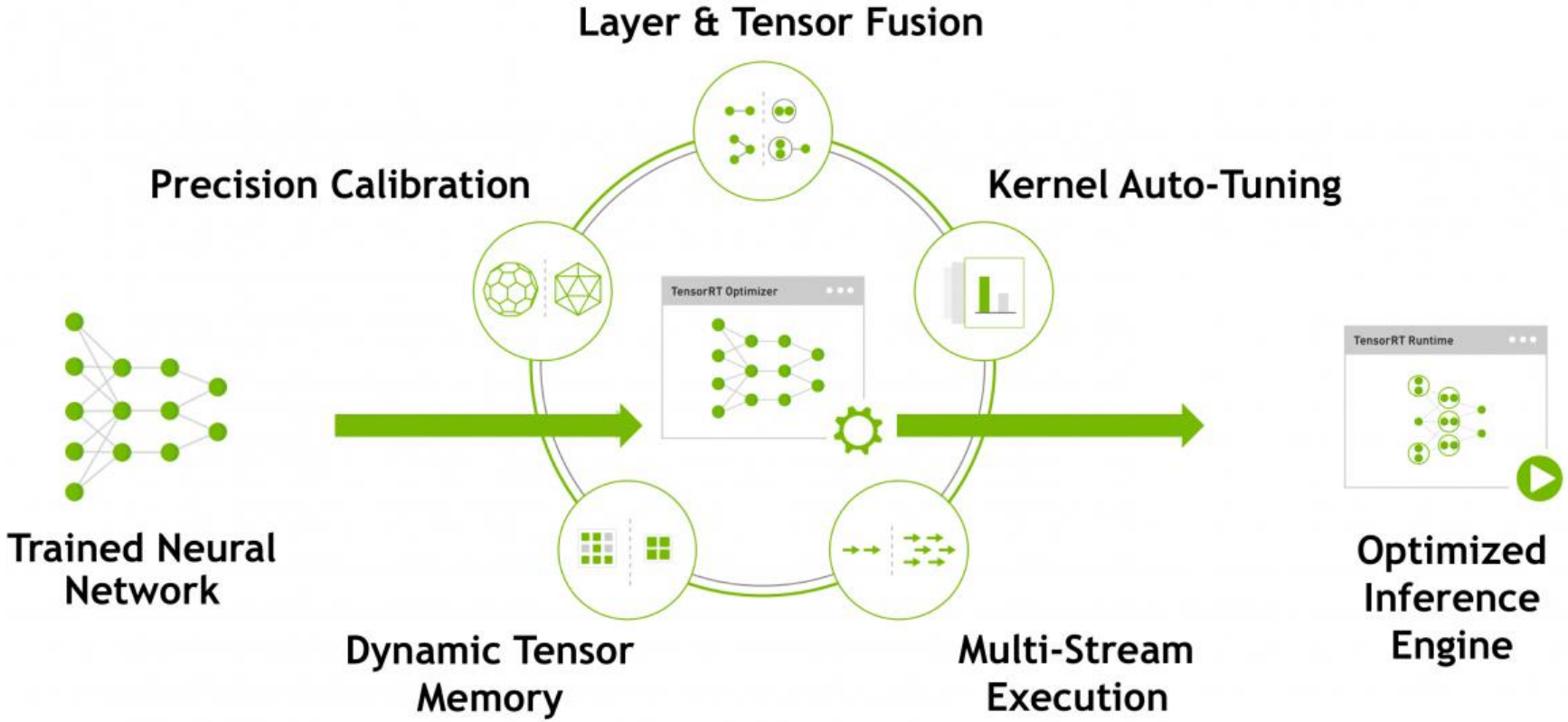
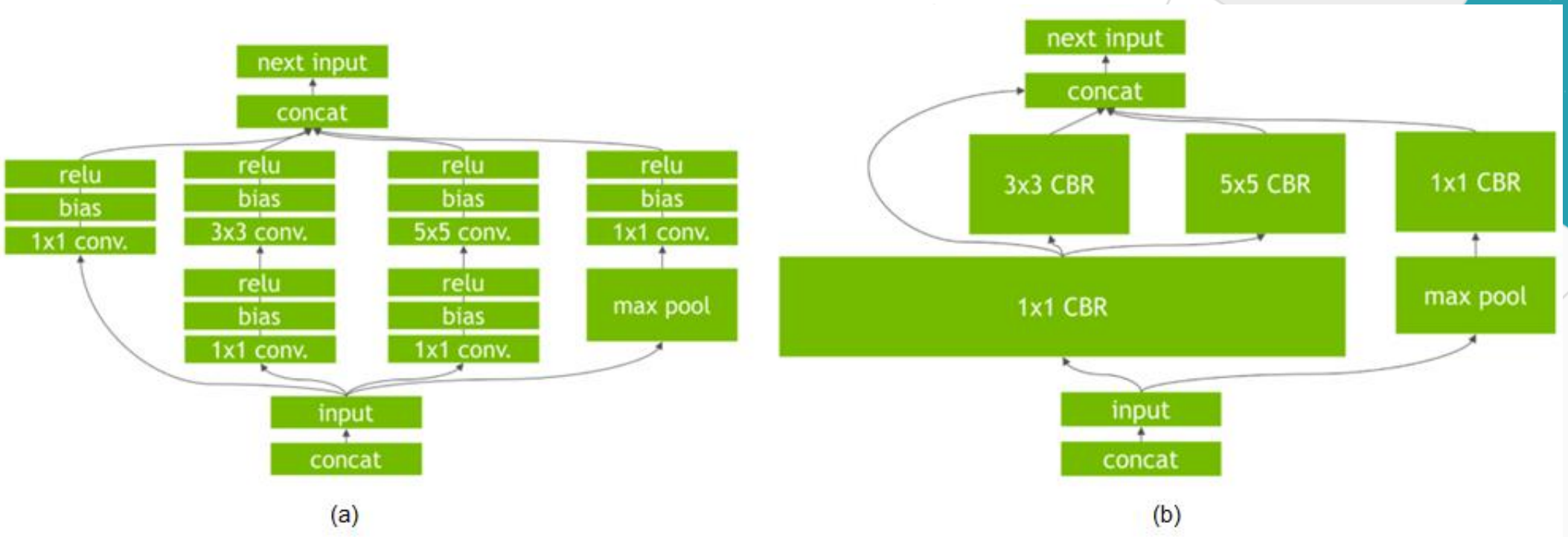


Schéma de fonctionnement de TensorRT

SOLUTIONS POUR L'EMBARQUÉ

TENSORRT

https://developer.nvidia.com/tensorrt

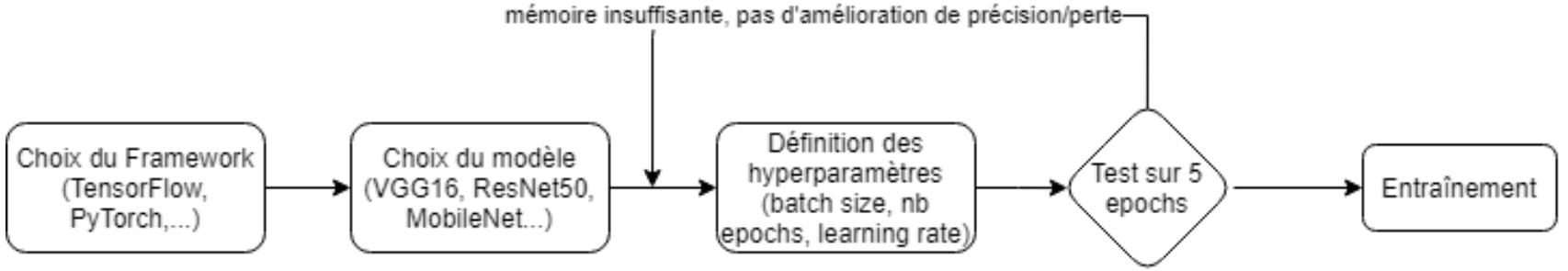


Exemple de compression avec TensorRT

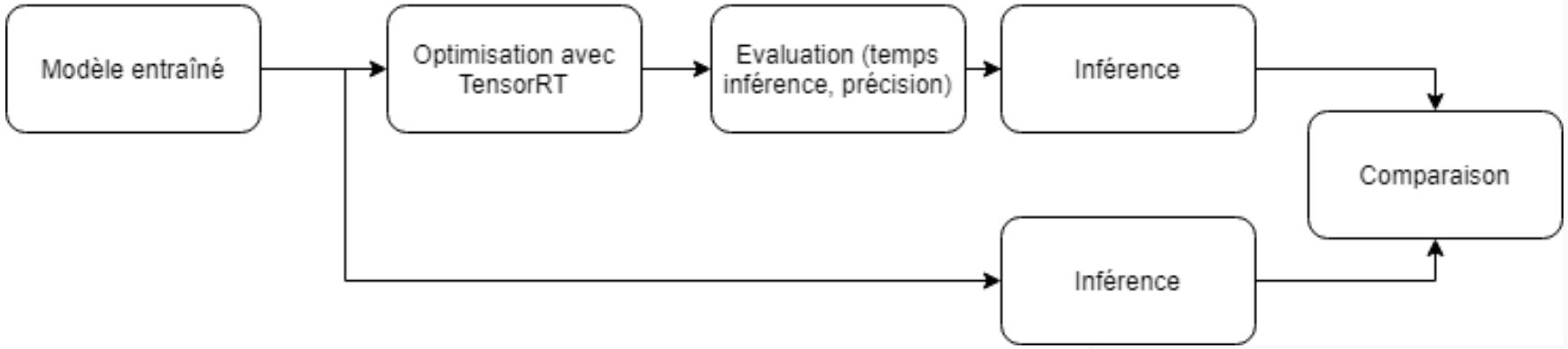
DÉVELOPPEMENT

MÉTHODOLOGIE

Le training



L'inférence



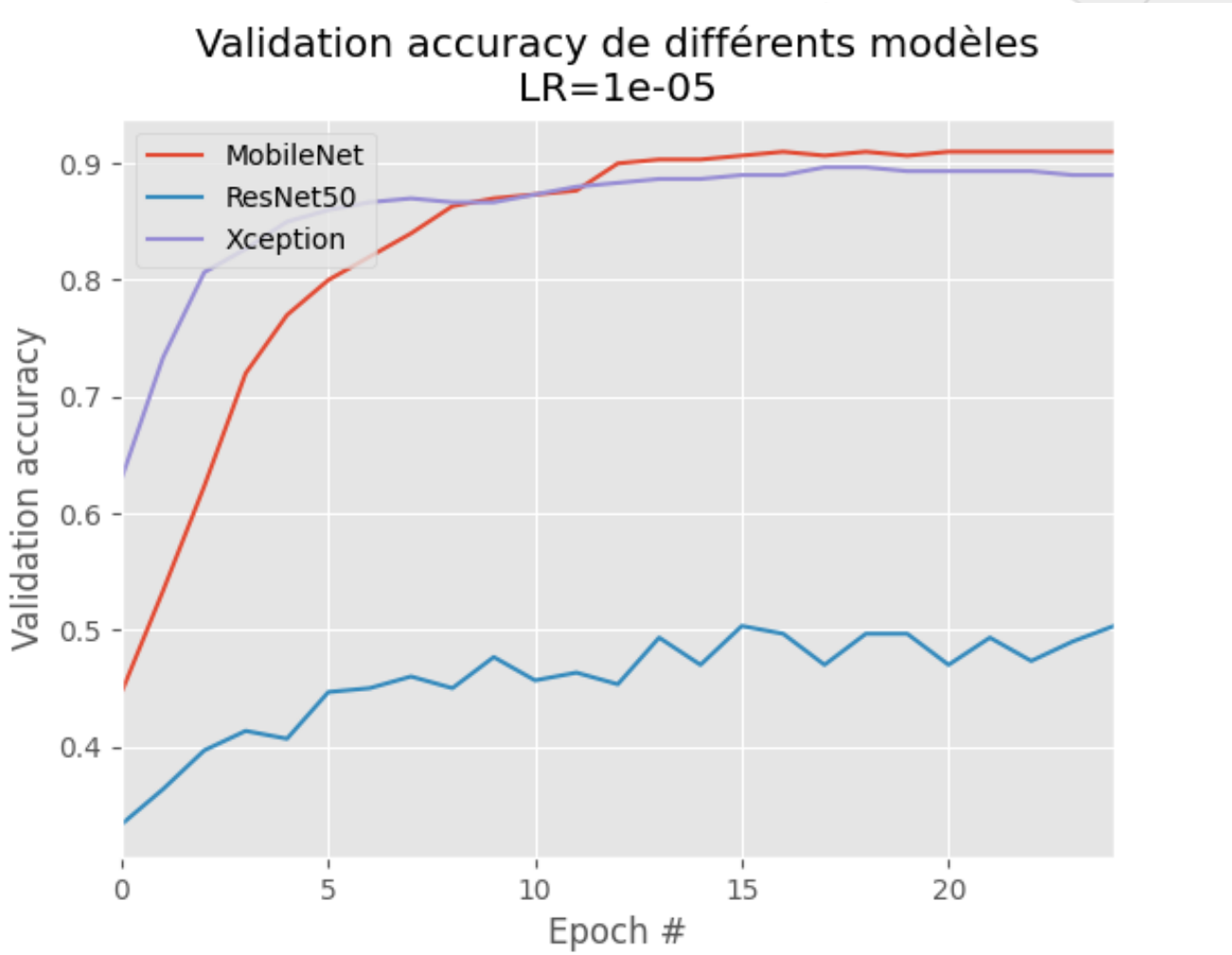
DÉVELOPPEMENT

PROBLÈMES RENCONTRÉS

- Mémoire de 4GB partagée entre le CPU et le GPU
- Absence d'un module WiFi
- Inférence avec TensorRT

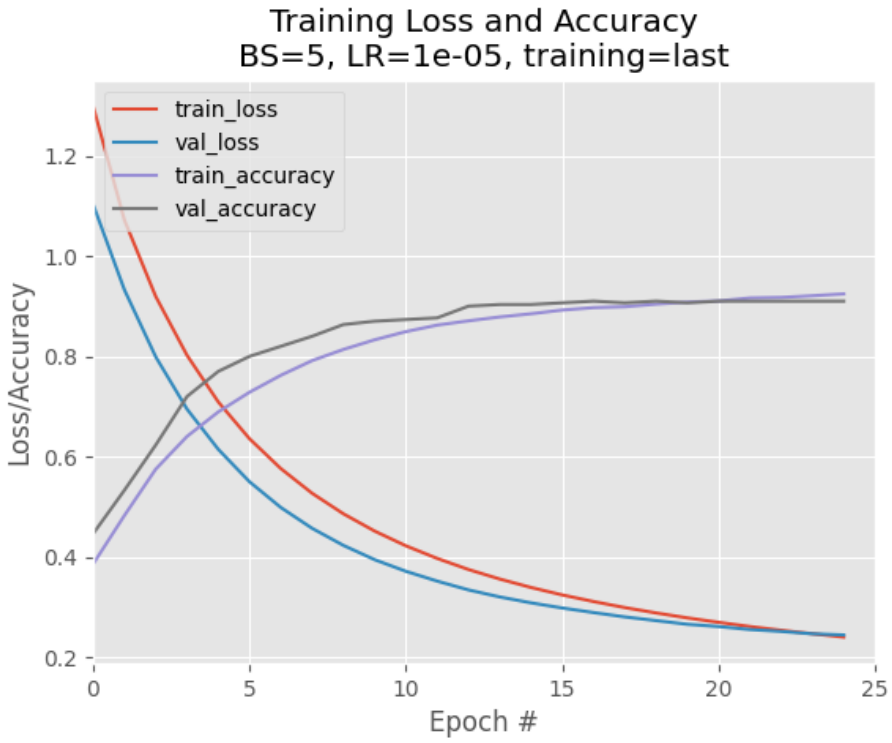
RÉSULTATS

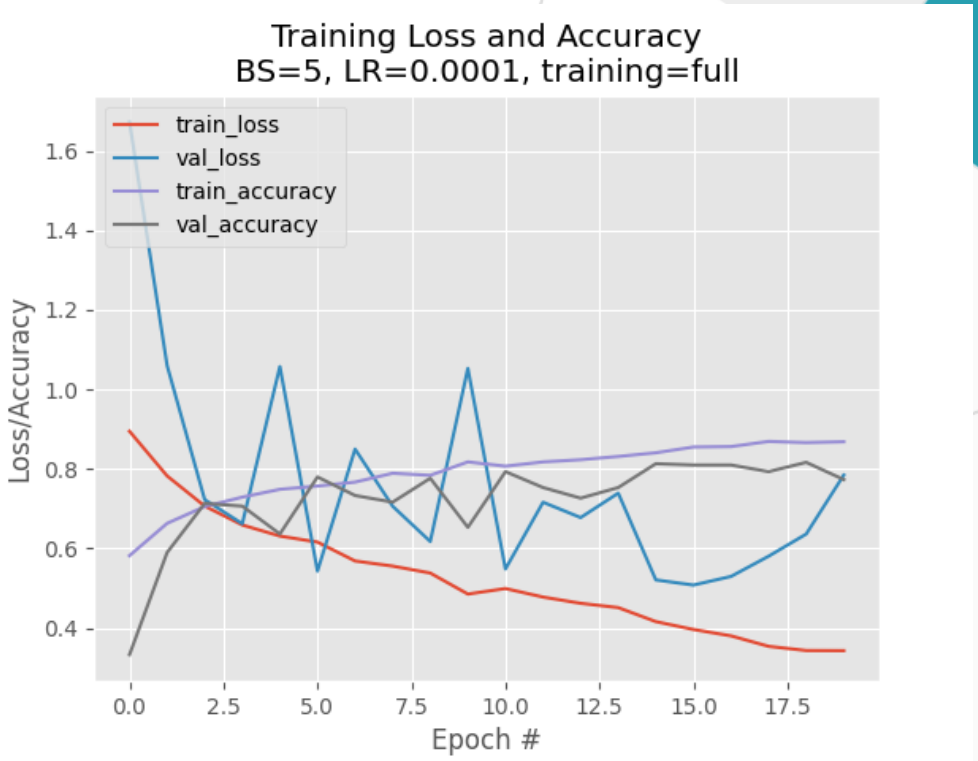
COMPARAISON DE DIFFÉRENTS MODÈLES SUR UNE BASE DE DONNÉES DE ~3000 IMAGES



RÉSULTATS

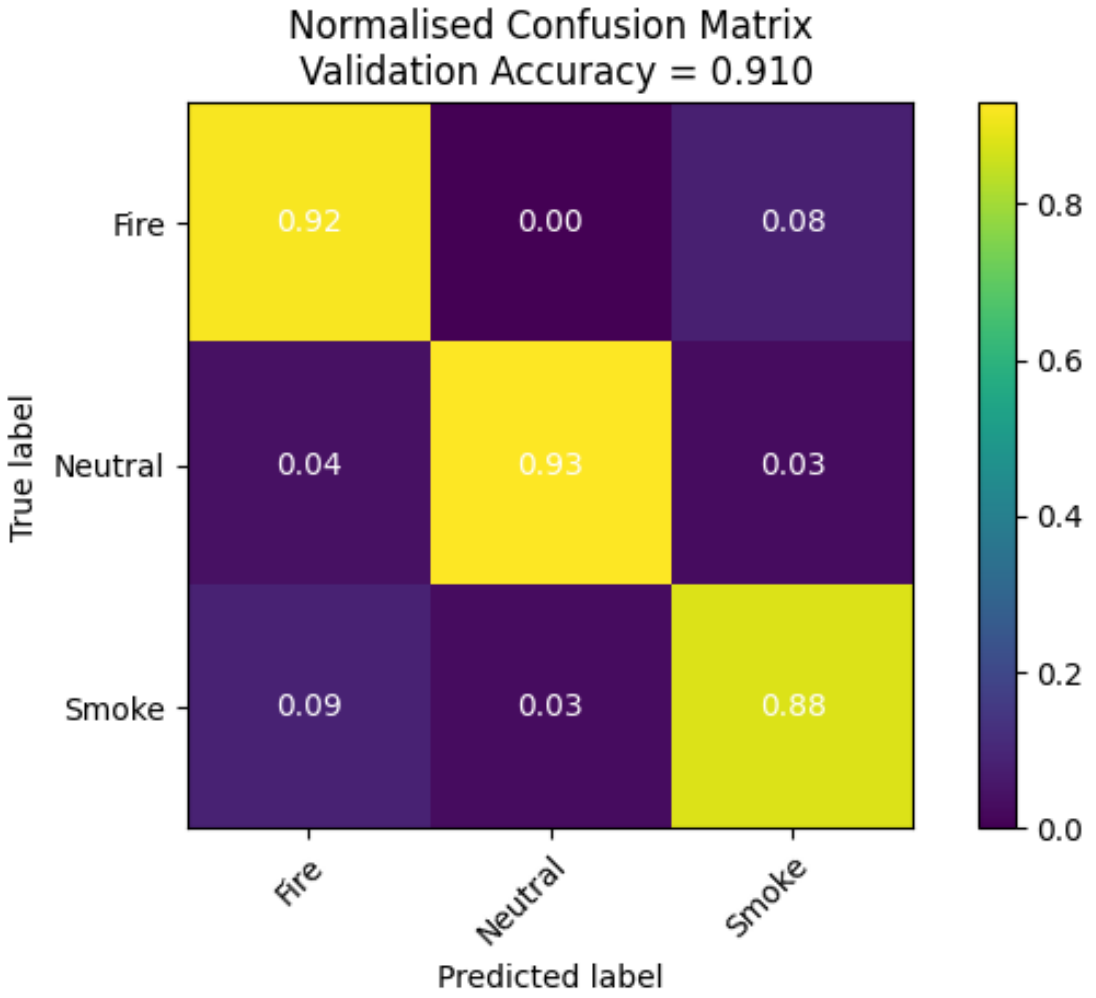
TRANSFER LEARNING OU PAS ?





RÉSULTATS

MATRICE DE CONFUSION DU MOBILENET



RÉSULTATS

EXEMPLES D'INFÉRENCE

Smoke: 74,72%
Fire: 23,39%

DEMO

Vidéo de test : Extrait du film Eragon (2006) – Bande sonore Keep Holding On
https://www.youtube.com/watch?v=UgD3JDgVxYk

CONCLUSION

- Développement d'une application de détection de feu basée sur de ML avec TensorFlow sur un mini-ordinateur, la Jetson Nano
- Comparaison de différents modèles
- Compréhension et manipulation des méthodes d'optimisation avec TensorRT

- Amélioration de la base d'images (plus de classes)
- Utilisation avec d'autres capteurs
- Apprentissage distribué ou fédéré ?
- ML sur µC ?

BIBLIOGRAPHIE

https://blog.tensorflow.org/2019/06/high-performance-inference-with-TensorRT.html

Sarkar, D. (DJ) (2018) A Comprehensive Hands-on Guide to Transfer Learning with Real-World Applications in Deep Learning, Medium.

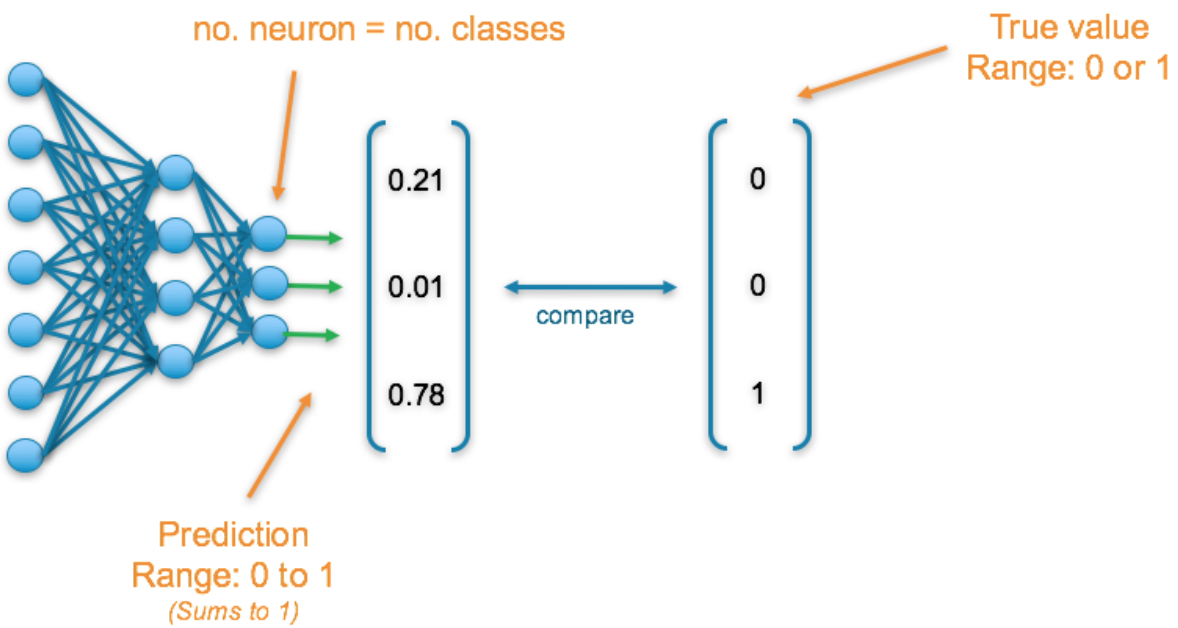
V, A. S. (2017) Understanding Activation Functions in Neural Networks

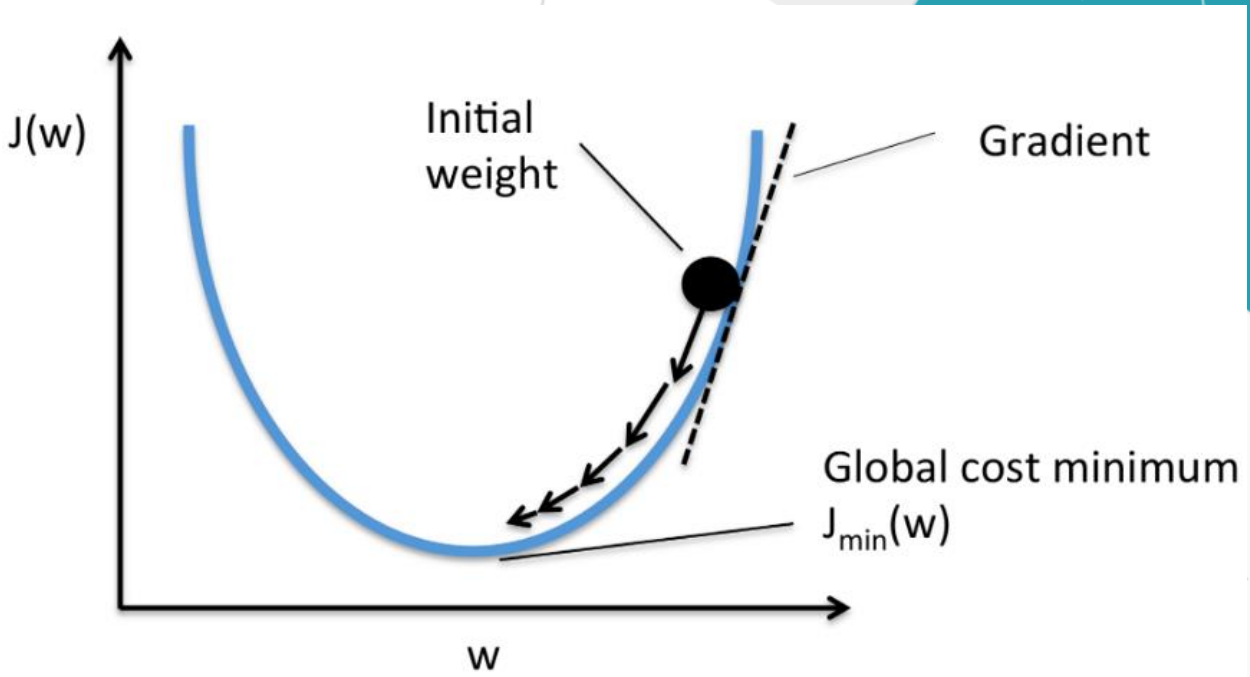
MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications

Blalock, D. et al. (2020) 'What is the State of Neural Network Pruning?'

SLIDES DE SUPPORT

FONCTIONS DE COÛT





SLIDES DE SUPPORT

ARCHITECTURE DE MOBILENET
SOURCE : MOBILENETS: EFFICIENT CONVOLUTIONAL NEURAL NETWORKS FOR MOBILE VISION APPLICATIONS

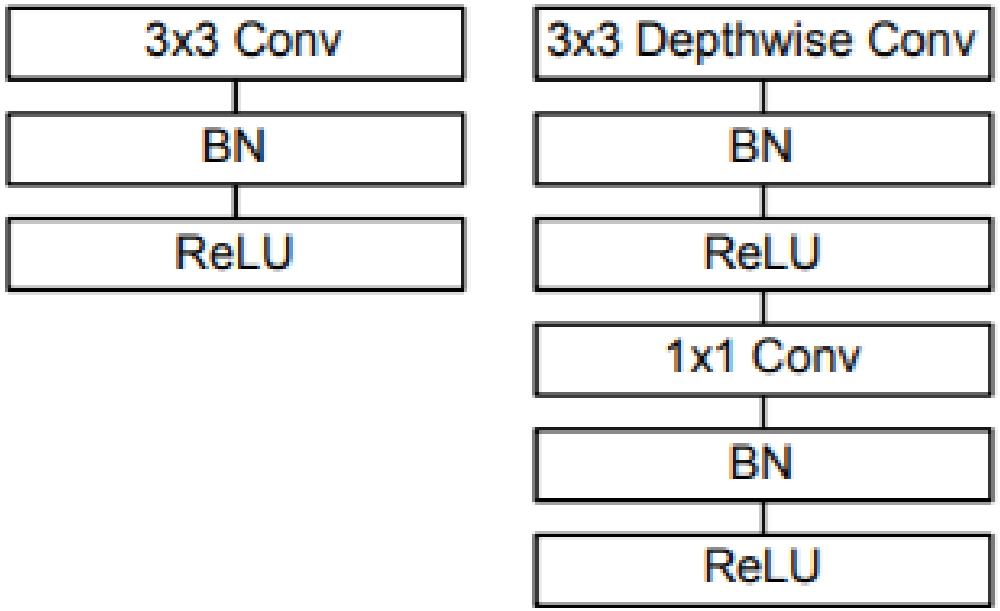
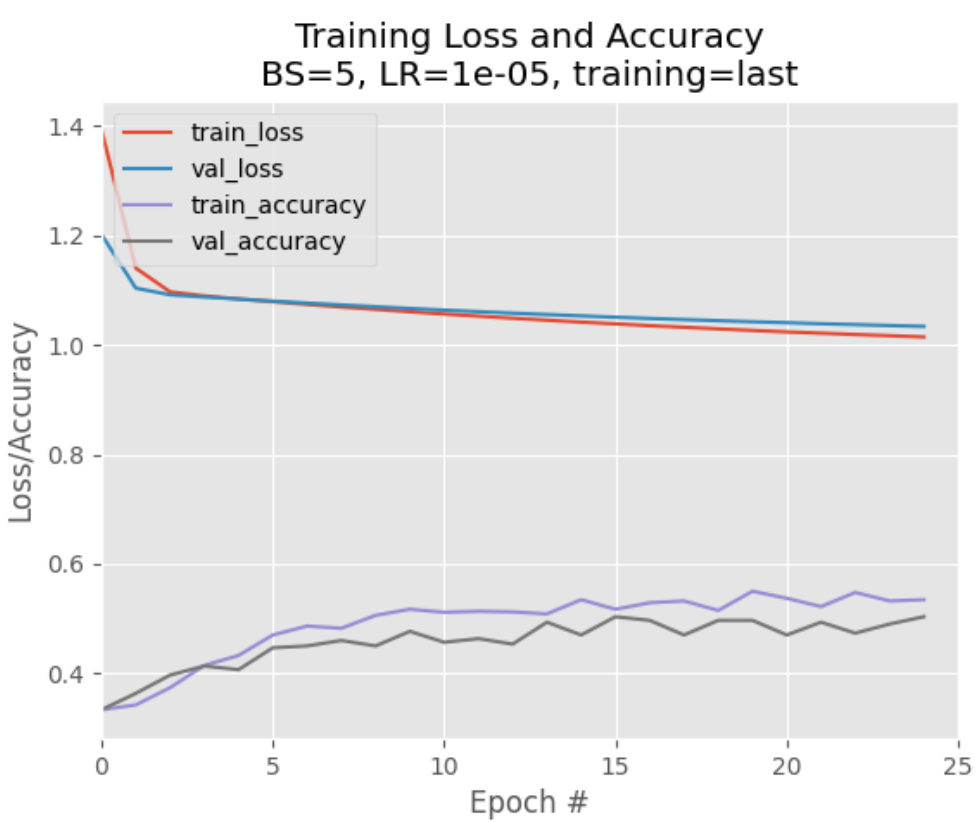


Table 1. MobileNet Body Architecture

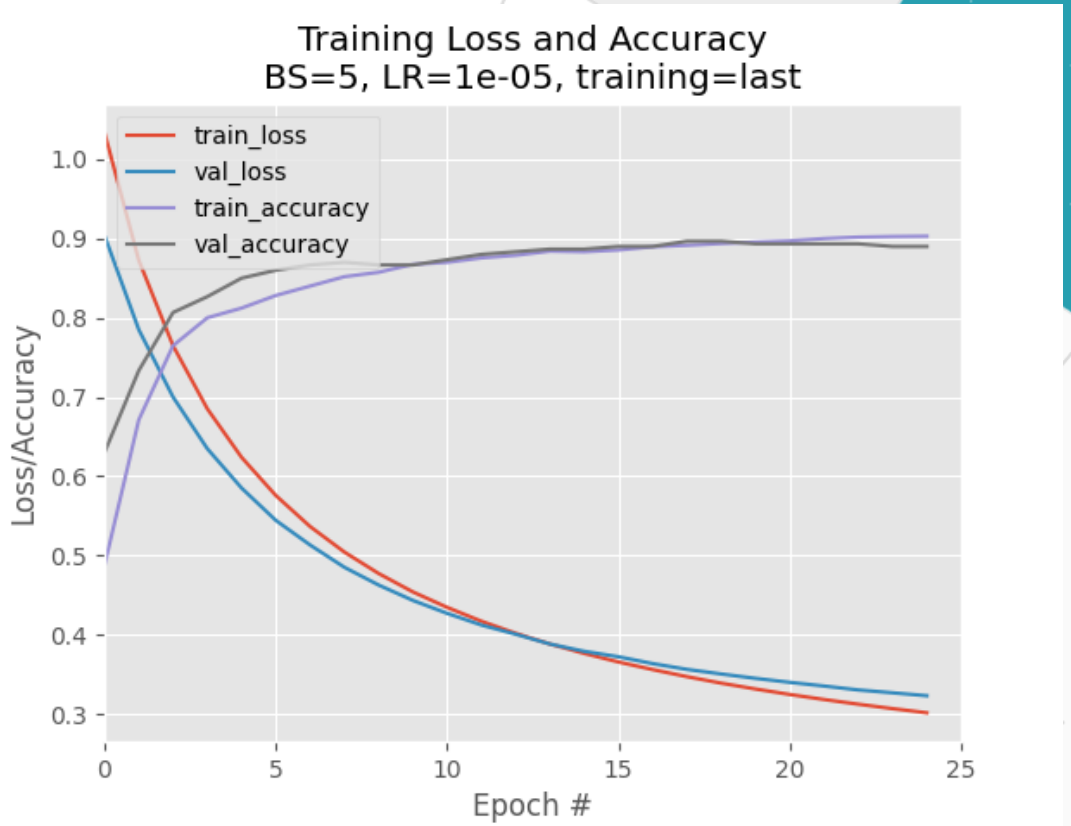
Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
$5\times$ Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
$5\times$ Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

SLIDES DE SUPPORT

COURBES D'ENTRAÎNEMENT POUR RESNET50 ET XCEPTION



ResNet50



Xception

SLIDES DE SUPPORT

COURBES DE DÉTERMINATION DES ERREURS