2. Choix et description du microcontrôleur cible

Pour ce projet de déploiement d’un modèle de réseau de neurones préentraîné sur le jeu de données CIFAR-10, le choix du microcontrôleur joue un rôle essentiel. L’objectif est d’obtenir une plateforme capable d’exécuter un modèle de classification d’images tout en respectant les contraintes propres aux systèmes embarqués, notamment en termes de mémoire, de puissance de calcul et de consommation énergétique.

Le microcontrôleur STM32L4R9AII6, basé sur un cœur ARM® Cortex®-M4, a été retenu pour sa polyvalence et ses performances adaptées aux applications d’intelligence artificielle légère. Il dispose de 2 MiB de mémoire Flash et 192 KiB de RAM, ce qui permet de stocker et d’exécuter des modèles de taille modérée, surtout après quantification. La carte intègre également plusieurs périphériques utiles au projet, tels qu’un écran AMOLED tactile, un port microSD™, un connecteur USB OTG, ainsi qu’un module de débogage ST-LINK/V2-1 facilitant la programmation et l’analyse des performances.

Cette plateforme offre une bonne base pour expérimenter le déploiement de modèles de vision embarquée, grâce à la compatibilité avec les outils logiciels de la suite STM32Cube.AI. Cet environnement permet de convertir et d’optimiser le modèle afin de l’adapter aux ressources limitées du microcontrôleur, tout en conservant des performances acceptables.

3. Évaluation de l’embarquabilité du modèle initial

L’étape suivante du projet consiste à évaluer la possibilité d’exécuter le modèle de réseau de neurones initial sur le microcontrôleur choisi, en tenant compte des contraintes matérielles de la carte STM32L4R9AII6. Cette analyse vise à déterminer si le modèle peut être déployé tel quel, ou s’il nécessite une adaptation pour respecter les limites de mémoire, de puissance de calcul et de temps d’inférence du système embarqué.

Pour cela, l’outil STM32Cube.AI a été utilisé afin de convertir le modèle préentraîné (au format TensorFlow) en un format compatible avec la famille STM32. Cet outil permet également d’obtenir des estimations précises concernant la taille du modèle, la mémoire RAM requise pour l’inférence, le type d’opérations arithmétiques utilisées, ainsi que le nombre total d’opérations nécessaires à l’exécution (en millions de multiplications-accumulations, ou MACC).

Dans un premier temps, le modèle a été analysé sans compression, puis avec les trois niveaux de compression proposés par STM32Cube.AI : *Low*, *Medium* et *High*. Ces niveaux reposent sur des techniques de quantification et de réduction de poids, permettant de réduire la taille mémoire et le coût de calcul, au prix d’une éventuelle perte de précision.

Les résultats des quatre analyses sont synthétisés dans le tableau suivant :

| Niveau de compression | Taille totale (Flash) | Poids (Weights) | Mémoire RAM totale | Nombre d’opérations (MACC) | Type d’opérations principales | Observation |
| --- | --- | --- | --- | --- | --- | --- |
| Aucune (None) | 5,37 Mo | 5,12 Mo | 143,8 Ko | 32,998 M | 96,5 % smul\_f32\_f32 (float 32) | Modèle trop volumineux pour la carte |
| Faible (Low) | 3,45 Mo | 3,33 Mo | 146,2 Ko | 32,998 M | Majorité d’opérations en float32 | Encore trop lourd pour la mémoire Flash |
| Moyenne (Medium) | 1,72 Mo | 1,62 Mo | 152,1 Ko | 32,998 M | 96,5 % smul\_f32\_f32, 3,2 % smul\_f32\_f4 | Compatible avec les ressources de la carte |
| Forte (High) | 1,27 Mo | 1,20 Mo | 147,4 Ko | 32,998 M | Opérations fortement quantifiées (poids en int8) | Déployable mais risque de perte de précision |

L’analyse montre que le modèle non compressé dépasse largement les capacités mémoire de la carte STM32L4R9AII6, empêchant son exécution directe. Même avec une compression faible, la taille du modèle reste supérieure à la limite de mémoire Flash embarquée. Ce n’est qu’à partir du niveau de compression moyen que le modèle devient compatible avec les 640 Ko de RAM et la mémoire Flash disponible (2 Mo) sur la carte.

Concernant le temps d’inférence, l’analyse de la complexité du modèle indique un total d’environ 33 millions d’opérations (MACC), essentiellement des multiplications flottantes (smul\_f32\_f32). Ce volume de calcul reste conséquent pour un microcontrôleur Cortex-M4 cadencé à 120 MHz, mais il demeure gérable après quantification et optimisation logicielle via les bibliothèques CMSIS-NN intégrées à STM32Cube.AI. Le temps d’inférence estimé reste compatible avec une exécution locale sur microcontrôleur, mais il demeure trop élevé pour une utilisation en temps réel sans compression ni optimisation supplémentaire.

Ainsi, le déploiement sur STM32L4R9AII6 n’est pas possible tel quel : le modèle initial est trop volumineux pour être chargé dans la mémoire interne. Une compression intermédiaire (niveau Medium) est nécessaire pour rendre l’application réalisable, tout en maintenant un temps d’inférence acceptable. Cependant, la précision reste à vérifier.