

Actualité – STI – septembre 2024

Mise en œuvre de l'IA dans les systèmes embarqués pour la reconnaissance d'objet - Performances et limites d'implémentation dans les systèmes embarqués

L'implémentation de l'IA sur des cibles et processeurs embarqués nécessite des précautions. L'IA dite de classification avec MATLAB présente des limites (puissance, taille des modèles, vitesse d'exécution, compatibilité des bibliothèques, consommation d'énergie). Le passage d'un environnement de simulation au prototypage constitue une solution de validation efficace et la Raspberry Pi est idéale pour expérimenter et prototyper des modèles d'IA.

L'IA et le Deep Learning dans le milieu agricole – Une application pour une plus grande productivité

<https://pedagogie.ac-toulouse.fr/sii/system/files/2021-09/ProjetTerminale-diseaseTargetingDeroulement.pdf>, réalisée par Cécile Tonnerre, enseignante STI2D du Lycée Pierre-Paul Riquet, St Orens de Gameville, académie de Toulouse.

Ce projet de smart farming propose une approche agricole utilisant des technologies avancées pour accroître la productivité face à une population mondiale croissante. L'utilisation de l'intelligence artificielle permet de surveiller les cultures et d'effectuer un épandage sélectif de produits phytosanitaires, ce qui est crucial pour l'agriculture de précision. Le projet sur la surveillance de vignes décrit le traitement des données et des calculs avec MATLAB pour détecter des feuilles de vignes malades ou saines, en utilisant une carte Raspberry Pi et une webcam pour la reconnaissance des feuilles malades. Les élèves doivent analyser les temps de calcul et utiliser des techniques de Deep Learning pour la mise en œuvre du modèle d'inférence (ou de prédiction).

Quels sont les critères à prendre en compte pour un déploiement applicatif sur cible embarquée - Quelle IA embarquer sur une cible contenant un processeur de type Raspberry Pi ?

La Raspberry Pi, avec ses ressources limitées en termes de puissance de calcul et de mémoire, nécessite l'utilisation de modèles d'IA légers et optimisés. Les modèles de machine learning classiques, tels que les arbres de décision, les forêts aléatoires et les réseaux de neurones simples, sont souvent mieux adaptés à ces contraintes. Les modèles de deep learning peuvent également être employés, mais ils doivent être soigneusement optimisés pour fonctionner efficacement sur une Raspberry Pi.

Le processus d'ingénierie adapté à l'IA peut suivre les étapes suivantes :

1. **Préparation des données** : Préparer les données pour l'entraînement, ce qui peut inclure l'annotation, le nettoyage, et l'augmentation des données afin d'améliorer la robustesse du modèle.
2. **Modélisation de l'IA ou Apprentissage** : Choisir un modèle d'IA approprié, tel qu'un réseau de neurones pré-entraîné (par exemple, SqueezeNet), et entraîner le modèle en utilisant des outils comme MATLAB, tout en configurant les options d'apprentissage pour optimiser la performance.
3. **Simulation et test** : Tester le modèle dans un environnement simulé pour évaluer ses performances et identifier les problèmes potentiels. Effectuer des ajustements pour s'assurer que le modèle répond aux exigences de précision, de vitesse et de robustesse.
4. **Déploiement** : Déployer le modèle sur une plateforme cible, comme un Raspberry Pi, en utilisant des outils comme Simulink pour configurer et tester le modèle en conditions réelles. Utiliser le mode « Monitor & Tune » pour valider le modèle et effectuer des ajustements si nécessaire. Surveiller les performances du modèle en conditions réelles et effectuer des mises à jour ou des ajustements pour maintenir l'efficacité du système. Utiliser des outils comme le Raspberry Pi Resource Monitor pour suivre les processus en cours.

Configuration et approche procédurale pour une implémentation sur une cible embarquée

Les configurations matérielles pour effectuer du deep learning avec une Raspberry Pi : Il est recommandé d'utiliser au moins un modèle Raspberry Pi 4, avec si possible 4 ou 8 Go de RAM, et une carte microSD de 32 Go ou plus configurée en 64 bits. [Lien vers le produit Raspberry Pi 4 Model B.](#)

La préparation de l'environnement de développement nécessite un processus méthodique qui assure une implémentation réussie de l'IA sur des systèmes embarqués comme la Raspberry Pi, en tenant compte des limitations de puissance de calcul et de mémoire. Voici le processus adapté à notre cas d'application en utilisant MATLAB et Simulink :

- 1- Préparation de données
- 2- Apprentissage
- 3- Validation du modèle simulé
- 4- Déploiement et validation

La préparation de données :

La fonction **imageDatastore** permet de lire facilement des données stockées dans des répertoires. Cette fonction permet également de reconnaître les différentes classes, qui sont dans notre exemple "Grape__Esca" (malade) et "Grape__Healthy" (saine).

La fonction **imagePretrainedNetwork ("squeezeNet", NumClasses=numClasses)** adapte le réseau de neurones aux 2 classes.

La fonction **imageDataAugmenter** est utilisée pour créer un objet de configuration pour l'augmentation des images dans MATLAB, une technique importante pour améliorer la robustesse et la généralisation des modèles.

Il est important d'installer "Deep Learning Toolbox Model for SqueezeNet Network" à partir de "Get Add-Ons", une architecture conçue pour l'embarqué.

L'apprentissage :

L'apprentissage doit être effectué sur un ordinateur ou sur « MATLAB Online ». Pour plus de rapidité, vous pouvez utiliser les GPUs de votre machine. Le résultat de l'apprentissage sera notre modèle de prédiction ou d'inférence. Deux fonctions sont ici utilisées ; la fonction **trainingOptions** de MATLAB pour spécifier les options de formation pour les réseaux de deep learning. Elle permet de configurer divers paramètres tels que l'algorithme d'optimisation, le taux d'apprentissage, le nombre d'époques. Ces options sont ensuite passées à la fonction **trainNetwork** pour entraîner le réseau de neurones.

L'apprentissage :

L'apprentissage doit être effectué sur un ordinateur ou sur MATLAB Online. Pour plus de rapidité, on peut utiliser les GPUs de votre machine. **Le résultat de l'apprentissage sera notre modèle de prédiction ou d'inférence.** La fonction **trainingOptions** de MATLAB est utilisée pour spécifier les options de formation pour les réseaux de deep learning. Elle permet de configurer divers paramètres tels que l'algorithme d'optimisation, le taux d'apprentissage, le nombre d'époques, et bien plus encore. Ces options sont ensuite passées à la fonction **trainNetwork** pour entraîner le réseau de neurones.

Validation du modèle simulé :

Un même environnement de développement tel que MATLAB/Simulink permet d'assurer la simulation et la validation du modèle d'IA avant un déploiement sur la cible pour éventuellement le corriger ou l'optimiser. Dans notre cas nous pouvons spécifier le format 'single' en entrée du modèle pour réduire la taille des données dans le processeur cible et ajouter des classes prédites sur les images avec le 'Bloc Insert Text' de la bibliothèque 'Computer Vision'. Nous pourrions alors tester deux types d'images de vignes.

Déploiement et validation :

Un même IDE (environnement Intégré de Développement) permet via la fonction 'Hardware Setting' de déployer le modèle sur Raspberry Pi. Le mode « Monitor & Tune » permet de valider le prototypage en conditions réelles et de faire des ajustements si nécessaire. Cette étape consiste à tester le modèle dans des scénarios d'utilisation réels pour s'assurer qu'il répond aux attentes en termes de précision, de vitesse d'exécution et de robustesse.

Une fois validé, la fonction « Build, Deploy & Start » va rendre la carte cible autonome. Le suivi du process de la cible pourra être fait avec le fichier d'extension elf, via la fonction [Raspberry Pi Resource Monitor](#).

Conclusion :

Ce projet sur le smart farming, axé sur la surveillance et le traitement précis des cultures, démontre comment l'intelligence artificielle peut révolutionner les pratiques agricoles traditionnelles, en offrant des gains significatifs en termes de productivité et d'efficacité. Bien que l'implémentation de l'IA sur des systèmes embarqués comme le Raspberry Pi présente des défis, notamment en termes de puissance de calcul et de mémoire, les avancées en deep learning et les outils fournis par les outils de développement intégrés permettent de produire des solutions innovantes et efficaces. Cet article souligne l'importance d'une approche méthodique et rigoureuse pour garantir le succès de l'intégration de l'IA dans des environnements réels. Les configurations matérielles sont décrites pour une implantation et une exploitation optimale de l'IA sur les cibles embarquées.