ESIGELEC Rouen
INGÉNIEUR.E.S GÉNÉRALISTES
SYSTÈMES INTELLIGENTS ET CONNECTÉS

DEEP LEARNING SUR RASPBERRY PI4, APPLIQUE A LA DETECTION ET A LA RECONNAISSANCE DE VISAGES EMBARQUES SUR UN PETIT ROBOT MOBILE

5 auteurs et une tutrice :

- **Ouerghi Safa(tutrice)**
- **HUANG Xingjian**
- **LUO Wenqi**
- **OWONO Emmanuel**
- **TEKOUO MATCHIOH Valex Wilfred**

Mai 2022

Deep Learning sur Raspberry PI4, appliqué à la détection et à la reconnaissance de visages embarqués sur un petit robot mobile

Prof .Ouerghi Safa,HUANG Xingjian,LUO Wenqi,OWONO Emmanuel and TEKOUO MATCHIOH Valex Wilfred (par ordre alphabétique)
ESIGELEC, 76800 saint éteint de Rouvray France 20/05/2022

Résumé

Dans cet article, nous décrivons une application de détection de visage en temps réel, rapide et précise, basée sur les réseaux de neurones convolutifs (CNN). Après avoir entraîné le CNN sur le PC, nous avons utilisé un Raspberry Pi 4 Modèle B(8Go) pour le programme de classification. Et nous l'appliquons à un petit robot équipé d'une caméra Raspberry Pi, afin qu'il puisse effectuer différentes actions de commande lorsqu'il détecte différentes personnes. Nous obtenons ici une précision de reconnaissance de plus de 90%.

1 Introduction

La technologie de reconnaissance automatique des visages est une technologie qui a connu un grand développement et une grande application dans le monde entier ces dernières années. Par conséquent, notre groupe a essayé de former le modèle d'apprentissage programmé sur PC, et a appliqué la fonction de reconnaissance automatique des visages sur Raspberry Pi4. Par la suite, on l'a appliquée à un petit robot équipé d'une caméra Raspberry Pi et d'un capteur à ultrasons, pour que lors du déplacement, le petit robot puisse identifier automatiquement les informations du visage capturé. Le réseau neuronal convolutif (CNN) utilise beaucoup de ressources lors de l'apprentissage et ce temps d'apprentissage est également très long. Cependant la classification d'image prend moins de ressource, est rapide et efficace. De cette façon, nous pouvons réaliser la fonction du robot pour détecter l'identité de la personne en temps réel.

2 Méthodes

2.1 DataSets

Pour entrainer notre model et effectuer des tests de la fonction de reconnaissance des visages sur le Raspberry Pi4 nous avons utilisé un package d'OpenCV pour avoir des images constituées des personnes de notre groupe. Ainsi notre dataset est constitué de 1300 images en couleur au format RGB. Ces images sont prises sous différents angles, avec des luminosités différentes. Chaque membre de l'équipe a fourni 300 images et nous avons utilisé 100 images pour une personne extérieur. 75 % de l'ensemble de notre dataset constitue notre training set et 25 % notre test set. De cette façon, nous évitons les résultats d'entraînement irréels pendant la phase de validation de l'entraînement du modèle. Toutes les images que nous avons capturées

sur openCV sont de 300*300 pixels.

Figure1 : Un ensemble de données de quatre membres et un ensemble de données de membres inconnus.

2.2 Apprentissage et RNC

En première étape, nous définissons un modèle convolutif avec deux couches convolutives et deux couches de max-pooling. Avant d'utiliser l'algorithme de classificateur CNN, nous mettons l'image à l'échelle 100*100. Dans le CNN, d'abord, nous utilisons ces 100*100 images pour l'entrée, et 5*5 noyaux, alors nous pouvons obtenir 16 96*96 images. Ensuite, nous utilisons un max-pooling pour obtenir 16 48*48 images. Ensuite, ces 16 48*48 images sont devenues l'entrée de la deuxième couche de convolution, ce qui nous permet d'obtenir 32 44*44 images, puis nous les mettons dans un deuxième max-pooling pour obtenir 32 22*22 images. Ensuite, nous utilisons trois couches entièrement connectées, la première sortie de 120 neurones, la deuxième sortie de 84 neurones, la troisième sortie de 10 neurones. L'étape suivante consiste à définir une ***fonction forward*** qui utilise ***la fonction Relu*** comme fonction d'activation. A ce moment, nous utilisons la ***fonction transforms*** de la bibliothèque ***torchvision*** pour traiter le jeu de données en images de 100*100 pixels. Nous mettons ces images dans le ***modèle CNN*** déjà écrit pour la convolution d'image avec le CPU, en utilisant le modèle d'apprentissage déjà défini, nous effectuons ***50 epoch*** d'apprentissage, et nous obtenons une bonne valeur de perte, une bonne précision d'apprentissage, et de bonnes valeurs de poids à la fin. Enfin, nous mettons l'ensemble de test dans le modèle formé, et nous obtenons l'image de détection de visage.

2.3 Reconnaissance

Dans cette étape, nous utilisons le meilleur modèle dans l'apprentissage. Puis nous comparons l'image avec notre dataset, nous pouvons alors obtenir l'index du répertoire du dataset, nous utilisons une classe pour définir le nom des différents index du répertoire, puis nous pouvons reconnaître notre visage, et afficher le résultat.

Figure2 : image pour reconnaissance avec 4 membres et une inconnue

3 Visage Détection

Dans cette étape, nous voulons obtenir un meilleur modèle de détection en utilisant un grand ensemble de données, la FDDB, qui contient des images de visages sous différents angles de nombreuses personnes différentes, avec un nombre variable de visages dans une seule image, et nous convertissons d'abord l'ensemble de données dans format **Pascal Voc**. Nous avons ensuite entraîné l'ensemble de données 100 fois pour obtenir un meilleur modèle avec MobileNetV2-SSD, et le meilleur modèle obtenu au moment de la mesure.

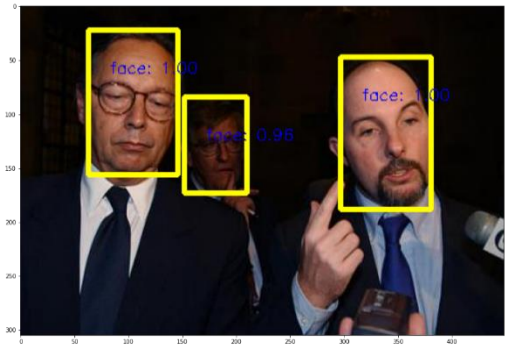

Figure3 : image où les visages sont détectés

4 Utilisations du Raspberry Pi4 et de la launchPad

Dans cette étape, nous mettons en commun le programme de détection et celui de l'identification. Nous avons également installé le système et les librairies nécessaires sur le raspberry. Le msp430 et la Raspberry communique à travers une communication SPI. Le principe est le suivant si la caméra du Raspberry Pi4 détecte quelqu'un, le Raspberry Pi4 le transmet au launchpad, qui allume la LED. Afin d'éviter les obstacles lorsque notre rebot avance nous utilisons un capteur ultrason. L'ensemble code et matériel étant reporté sur un même support (le robot final)

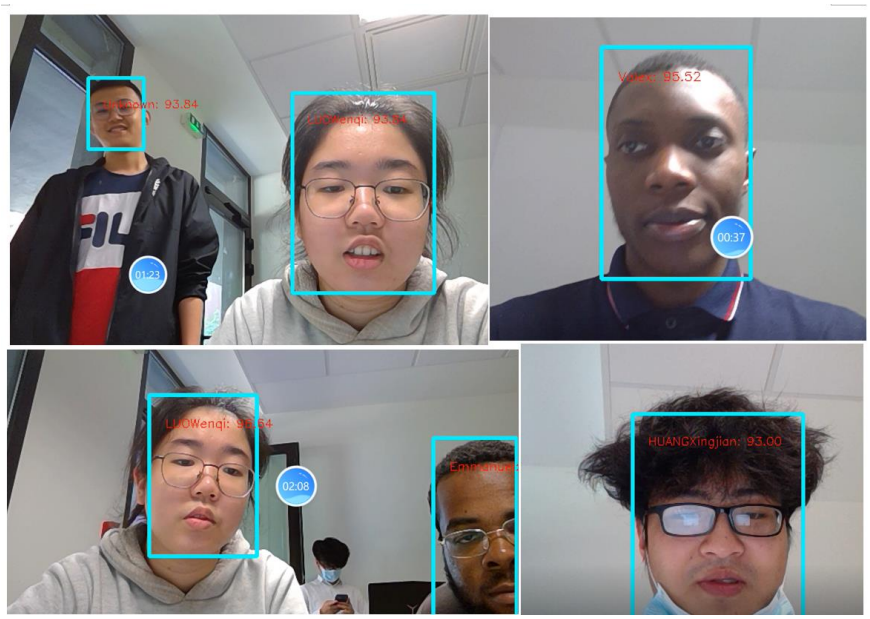

Figure4 : résultat d'implémentation

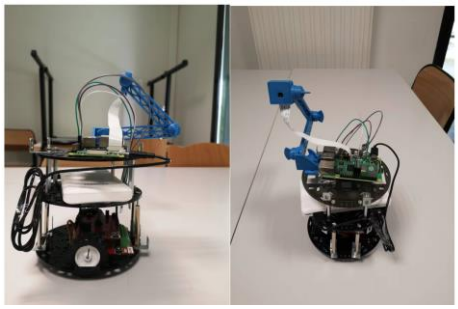

Figure4 : robot final

5 Résultat et conclusion

Nous avons réalisé un modèle d'apprentissage profond et un programme de reconnaissance de visage basé sur CNN. Après 50 epochs d'entraînement sur le PC pour le modèle d'apprentissage profond, une meilleure valeur de poids a été obtenue et la valeur de perte finale était de 0,0008. Pour les images, nous avons eu une précision numérique de plus de 99% et dans la deuxième étape de détection, nous avons fait 100 epochs d'entraînement avec une précision numérique d'environ 90%. Enfin, dans l'indice, que nous avons réalisé avec la caméra, notre précision numérique a dépassé 90%. Lors de la communication entre le MSP430 et le Raspberry nous avons noté un retard de près d'une dizaine de seconde dû au décalage lors de la transmission des données. Nous avions également des capteurs à ultrasons pour la détection des obstacles. S'il y a un obstacle à 10 centimètres du robot il s'arrête.

Références :

1. O. D¨urr, Y. Pauchard, D. Browarnik, R. Axthelm, and M. LoeserSchool of Engineering, Zurich University of Applied Sciences, Winterthur, Switzerland: Deep Learning on a Raspberry Pi for Real TimeFace Recognition.July 3, 2015.

2. Juan Du 2018 J. Phys.: Understanding of Object Detection Based on CNNFamily and YOLO.Conf. Ser.1004 012029.

3. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). Ssd: single shot multibox detector. European conference on computer vision (pp. 21–37).

4. Bay, H., Tuytelaars, T., & Van Gool, L. (2006). Surf: speeded up robust features. European conference on computer vision (pp. 404–417).

5. Mark Sandler,Andrew Howard,Menglong Zhu,Andrey Zhmoginov,Liang-Chieh Chen: MobileNetV2: Inverted Residuals and Linear Bottlenecks. 13 Jan 2018 (v1), last revised 21 Mar 2019 (this version, v4)

6. Cours d'intelligence artificielle sur l'ent (atelier1, atelier2, atelier3)