MINISTÈRE
DES ARMÉES
Liberté
Égalité
Fraternité

TELECOM Nancy
Ingénieurs du numérique • Inspiring your digital future

Note de cadrage

Comment utiliser le GPU de ma Nintendo Switch pour faire de la reconnaissance faciale ?

How can I use my Nintendo Switch's GPU to perform facial recognition ?

Acteurs :

— **Yoann** - Encadrant industriel - MINA
— **Marie** - Encadrant industriel - MINA
— **Bombardier Vincent** - Encadrant universitaire - TELECOM Nancy
— **Becquet Albert** - Étudiant - TELECOM Nancy
— **Pasquier Pierre** - Étudiant - TELECOM Nancy
— **Simon Damien** - Étudiant - TELECOM Nancy

12 octobre 2023

1 Acteurs

Yoann et Marie, appartenant au Ministère des Armées, sont nos encadrants industriels lors de ce projet. Spécialisés dans le software et la vidéo, notre rôle est de leur apporter notre expertise en intelligence artificielle et logiciel. Les clients finaux sont les opérationnels, ce sont eux qui utiliseront le dispositif que nous réussirons à mettre en place. M. Bombardier, professeur spécialisé dans le traitement numérique d'images, est notre encadrant universitaire.

2 Finalité

Le projet consiste en la mise en place d'un dispositif permettant d'utiliser un système de reconnaissance faciale à partir d'un GPU de type Jetson Orin Nano, connu pour être utilisé sur des Nintendo Switch. Actuellement, le traitement des données est réalisé en dehors de la zone de collecte de ces dernières. Le problème est que le transport de ces données est très coûteux en temps et en argent. Ce projet permettra ainsi de réaliser une partie de l'analyse sur place en rapprochant le traitement des données des opérationnels.

3 Objectifs

3.1 Etat de l'art et appropriation du matériel

Etat de l'art L'état de l'art doit comporter les différents modèles, architectures et jeux de données que nous jugerons utiles pour ce projet. Il sera complété par Yoann et Marie sur des points intéressants qu'ils ont déjà relevés. Un PowerPoint nous est demandé afin de le présenter.

Appropriation du matériel La liste du matériel nous ayant été fourni est en annexe dans la table 1. L'objectif est d'être capable de compiler l'image yocto, de faire en quelques sortes un *hello world yocto*, établir une pipeline `GStreamer` affichant la vidéo en cours. En plus du matériel, nous devrons nous approprier de nouvelles technologies afin de mener à bien ce projet. Un aperçu de celles-ci est disponible dans la table 2 en annexe.

3.2 Détection, collecte de visages sur vidéo en temps réel et reconnaissance faciale

Pour commencer, il faut être capable de détecter qu'une personne est présente sur la vidéo. Cela permet de ne pas faire tourner l'algorithme de reconnaissance faciale s'il n'y a personne sur le flux vidéo, ce dernier étant assez gourmand en ressources. Ensuite, nous devons être en capacité de générer un dossier contenant tous les visages rencontrés durant un laps de temps. De plus, nous devons générer des sous-dossiers contenant un ensemble de visages associés chacun à une unique personne. La finalité du projet est d'être capable de gérer un fichier d'entrée contenant une liste de visages de personnes cible, et d'envoyer une alerte lorsque cette personne est rencontrée sur le flux vidéo, tout en générant un dossier contenant ses images. Une attention particulière sera apportée à l'horodatage des données.

3.3 Tests de performances

Après avoir implémenté les deux types de caméra en entrée vidéo sous la forme d'un système de plug&play, nous évaluerons et comparerons leurs performances.

Considérée comme une partie bonus. Nous devons analyser les performances de notre dispositif, en considérant un maximum de dimensions comme la mémoire et les performances de l'IA.

4 Contraintes

Nos principales contraintes sont temporelles, l'objectif étant d'avoir une solution fonctionnelle d'ici février 2024. Nous devons également être capables de nous familiariser avec l'ensemble du matériel fourni, et de

développer une solution utilisant un minimum de ressources. En ce qui concerne le développement en lui-même, nous n'avons pas de contrainte sur les langages utilisés, mais ceux-ci doivent être cohérents avec la structure globale du projet. Il nous est aussi demandé d'utiliser le formatage snake_case lorsque l'on code, et d'utiliser des noms de variables explicites en anglais. Enfin, les documentations doivent respecter le format Doxygen.

5 Organisation

5.1 Répertoires git et code

Nos répertoires doivent être bien organisés, avec des noms de dossiers et de fichiers clairs. Le contenu des répertoires est résumé en annexe dans la table 3. Chaque dépôt `git` doit disposer d'un fichier README clair et concis. Un clone des dépôts hébergés à l'école est effectué dans un répertoire public afin de le rendre accessible.

Notre code doit suivre la convention typographique snake case et être commenté en anglais. Nous utilisons `Pylint` afin de mesurer la qualité de notre code. Les documentations quant à elles, doivent respecter le format Doxygen.

5.2 Réunions

Concernant les réunions avec nos encadrants, nous nous appellerons tous les jeudis à 14h30 afin de discuter des avancées de la semaine, des difficultés rencontrées et du travail à effectuer par la suite. En amont de ces réunions, nous devons envoyer un mail pour indiquer l'ordre du jour en 4 ou 5 points et les documents pouvant être utiles, étant donné que la réunion se fera en appel téléphonique. Si nous souhaitons les contacter en dehors de ces horaires, nous pouvons leur envoyer des mails.

Un vendredi sur deux, nous devons envoyer un mail à nos encadrants pour expliquer notre travail des deux dernières semaines d'un point de vue plus technique que lors des réunions hebdomadaires du jeudi. Environ une fois par mois, nos encadrants viendront à l'école pour que nous puissions leur faire des présentations et des démonstrations de notre travail.

5.3 Organisation interne au groupe de travail

Nous avons commencé par élire un chef de projet : Albert Becquet. Celui-ci a pour mission d'organiser le travail au sein du groupe et communiquer avec les encadrants. De plus, nous avons commencé à nous répartir les tâches selon les conseils de nos encadrants : un élève se concentrera sur la partie Yocto (Damien Simon) et deux élèves (Albert Becquet et Pierre Pasquier) naviguent entre la partie IA et la partie Yocto. Ces rôles ne sont pas fixes et peuvent évoluer par la suite en fonction du travail à effectuer.

6 Livrables

— Note de cadrage ;
— état de l'art sous forme de PowerPoint : modèles, architectures et jeux de données que nous estimons intéressants, avec un avis critique ;
— `hello world` yocto : lancement d'une carte avec rendu vidéo en utilisant yocto, premier pipeline fonctionnel ;
— fiche de synthèse de livrable : note de livraison pour chaque rendu de livrable, explication détaillée (localisation, étape du projet...) de chaque ajout de fonctionnalité sous forme de tableau ;
— synthèse du bilan de novembre ;
— soutenances ;
— rapport final.

7 Annexes

Nombre	Matériel	Description
2	GPU Jetson Orin Nano devkit	Fourni avec 1 alimentation et 2 adaptateurs Deux modes de fonctionnement (7W et 15W) 7W à privilégier
1	Carte SD	64 GB
3	Caméras	2 caméras MIPI et 1 caméra USB

Table 1 – Matériel fourni

Technologie	Description
Git	Système de gestion de versions pour le suivi des modifications de code
Docker	Plateforme de virtualisation légère pour le déploiement d'applications dans des conteneurs
Bash	Shell Unix/Linux couramment utilisé pour l'automatisation de tâches en ligne de commande
Yocto	Outil de construction de systèmes embarqués personnalisés
Systemd	Gestionnaire de système d'initialisation pour les distributions Linux modernes
Python	Langage de programmation polyvalent et populaire
Gstreamer	Cadre de travail pour le traitement multimédia et la gestion de flux
Framework IA	Bibliothèques pour le développement de modèles d'intelligence artificielle et d'apprentissage automatique (comme TensorFlow, Keras, PyTorch...)

Table 2 – Technologies utilisées

Répertoire	Description
yocto-jetson-repo	Fichier `manifest.xml` avec les liens des images utilisées pour yocto ainsi que les instructions de compilation
meta-reco	Code applicatif spécifique à la carte, recettes yocto spécifiques à l'IA et la reconnaissance faciale
docker-repo	`dockerfile` permettant de créer une image docker avec les bonnes librairies de yocto
facial-reco-ia	Dépôt principal du projet avec tous les codes utilisés relatifs à l'IA et reconnaissance faciale respectant une structure classique de dossiers (source, tests, modèles...)

Table 3 – Répertoires `git` attendus

Signature

Étudiants

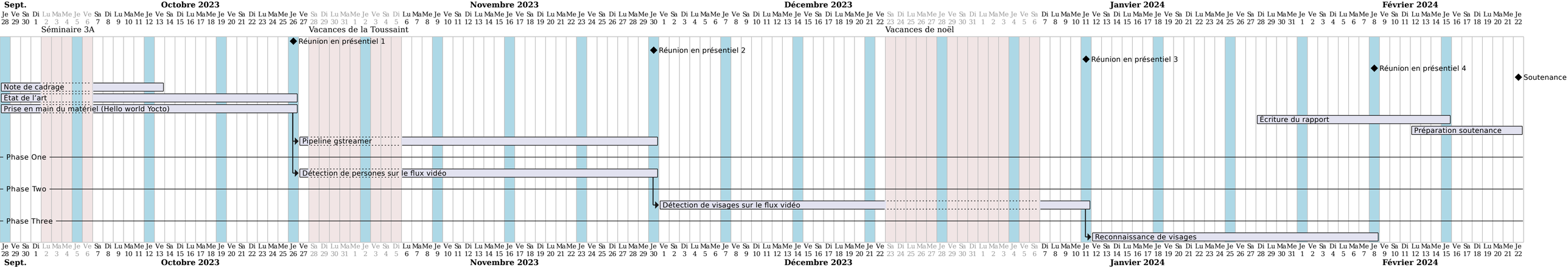
Albert Becquet **Damien Simon** **Pierre Pasquier**

Encadrants industriels

Marie **Yoann**

Encadrant universitaire

Vincent Bombardier


Sept.
Octobre 2023
Novembre 2023
Décembre 2023
Janvier 2024
Février 2024
Séminaire 3A
Vacances de la Toussaint
Vacances de noël
Réunion en présentiel 1
Réunion en présentiel 2
Réunion en présentiel 3
Réunion en présentiel 4
Soutenance
Note de cadrage
État de l'art
Prise en main du matériel (Hello world Yocto)
Pipeline gstreamer
Détection de persones sur le flux vidéo
Détection de visages sur le flux vidéo
Reconnaissance de visages
Écriture du rapport
Préparation soutenance
Phase One
Phase Two
Phase Three