



POLYTECH[®]
DIJON



**UNIVERSITÉ
BOURGOGNE
EUROPE**



forthem.

Fostering Outreach
within European Regions,
Transnational Higher Education
and Mobility

Projet GPU

DETECTION DE PERSONNE AVEC NVIDIA

Elsa Allard – Hugo Roche
5A SE | POLYTECH DIJON

Table des matières

Introduction	2
Méthode 1 : Traitement d'image via OpenCV.....	3
Principe.....	3
Fonctionnement du code	3
Résultats	4
Méthode 2 : Intelligence artificielle avec YOLO.....	5
Principe.....	5
Fonctionnement du code	5
Évolution et optimisation de l'approche YOLO.....	6
Résultats	6
Comparaison des deux approches	7
Conclusion	7

Introduction

Le but de ce projet est de concevoir et de développer une application capable de compter automatiquement le nombre de personnes entrant dans une salle à l'aide d'une caméra fixe. Ce travail vise à exploiter les technologies proposées par NVIDIA pour le traitement vidéo en temps réel.

Pour atteindre cet objectif, deux approches différentes sont étudiées et comparées. La première repose sur des méthodes classiques de traitement d'image utilisant la bibliothèque OpenCV, tandis que la seconde utilise une approche basée sur l'intelligence artificielle à travers l'algorithme YOLO. Ces deux méthodes permettent de détecter les personnes à partir d'un flux vidéo, mais diffèrent fortement en termes de précision, de complexité et de performances.

L'objectif du projet est également d'analyser les avantages et les limites de chaque approche, notamment en ce qui concerne la robustesse face aux conditions réelles telles que les variations d'éclairage et la présence d'objets dans la scène. Enfin, ce projet vise à mettre en évidence l'apport des plateformes NVIDIA dans l'accélération des algorithmes de vision par ordinateur et des modèles d'intelligence artificielle.

Méthode 1 : Traitement d'image via OpenCV

Principe

La première méthode repose sur des techniques classiques de traitement d'image. Cette approche présente l'avantage d'être simple à mettre en œuvre et peu coûteuse en ressources de calcul. Elle peut fonctionner en temps réel sur un processeur classique. Cependant, elle est sensible aux variations d'éclairage et génère des faux positifs, notamment lorsque des objets présentent une forme similaire à celle d'une personne. De plus, elle gère difficilement les occlusions et les scènes complexes.

Fonctionnement du code

Le programme commence par l'initialisation de la caméra à l'aide de la classe « VideoCapture ». La résolution de capture est fixée à 640×480 pixels afin de garantir un bon compromis entre qualité d'image et performances en temps réel. Une vérification est effectuée pour s'assurer que la caméra est correctement ouverte avant de lancer le traitement vidéo.

Un soustracteur de fond de type MOG2 est ensuite utilisé afin de détecter les zones en mouvement dans la classe. Cette technique permet de distinguer les objets mobiles, tels que les personnes, de l'arrière-plan statique. Le traitement est limité à une région d'intérêt (ROI) correspondant à la zone de la porte, ce qui réduit le bruit et améliore la fiabilité du comptage.

À chaque image, la soustraction de fond génère un masque binaire représentant les mouvements détectés. Ce masque est ensuite nettoyé à l'aide d'opérations qui permettent d'éliminer les petits objets et de regrouper les zones correspondant à une personne. Les contours présents dans le masque sont extraits, puis filtrés selon leur surface afin d'ignorer les objets trop petits pour correspondre à une personne.

Pour chaque contour valide, un rectangle englobant est calculé et son centre est déterminé. Une ligne virtuelle verticale est définie au centre de la zone de la porte. Le comptage est réalisé en comparant la position du centre de la personne entre deux images consécutives. Lorsqu'un déplacement du centre est détecté d'un côté de la ligne vers l'autre, une entrée ou une sortie est comptabilisée. Le nombre total de personnes présentes dans la salle est mis à jour en temps réel et affiché sur l'image.

Résultats

Les résultats montrent que cette méthode permet de compter efficacement le nombre de personnes entrant et sortant de la salle dans des conditions simples. Avec une caméra fixe placée face à la porte et des personnes entrant une par une, le système détecte correctement les mouvements et met à jour le compteur de manière fiable.

Le comptage est précis lorsque les personnes traversent clairement la zone de la porte et que l'éclairage reste relativement stable. Les faux positifs sont limités grâce à l'utilisation d'une région d'intérêt et au filtrage par taille des objets détectés. Le système fonctionne en temps réel, avec un affichage fluide des résultats, ce qui le rend adapté à une utilisation continue.

Cependant, certaines limites ont été observées. Lorsque plusieurs personnes passent simultanément par la porte, le système peut ne détecter qu'un seul individu, ce qui entraîne une erreur de comptage. De plus, des variations importantes de luminosité ou des mouvements parasites dans la zone de la porte peuvent perturber la détection. Malgré ces limites, cette approche basée sur le traitement d'image classique constitue une solution simple et efficace pour des environnements contrôlés.

Méthode 2 : Intelligence artificielle avec YOLO

Principe

La seconde méthode est basée sur l'intelligence artificielle et utilise l'algorithme YOLO (You Only Look Once), qui repose sur un réseau de neurones convolutif. Ce modèle est entraîné à détecter différents objets, dont les personnes, directement à partir des images. Contrairement à la méthode classique, YOLO permet une détection plus précise et plus robuste, même dans des environnements complexes.

Grâce à l'utilisation d'un GPU NVIDIA, le modèle YOLO peut fonctionner en temps réel tout en conservant une grande précision. Cette méthode réduit considérablement les faux positifs et gère mieux les cas d'occlusion ou de variations de posture. En revanche, elle nécessite davantage de ressources de calcul et une mise en œuvre plus complexe.

Fonctionnement du code

Dans cette seconde approche, le comptage de personnes est réalisé à l'aide d'une méthode basée sur l'intelligence artificielle. Pour cela, la bibliothèque Ultralytics est d'abord installée, puis un modèle YOLOv8 pré-entraîné est utilisé. Le modèle est chargé au format ONNX, ce qui permet une meilleure portabilité et une accélération possible sur les plateformes NVIDIA.

Le traitement est implémenté sous la forme d'un script Python utilisant OpenCV pour l'acquisition vidéo et l'affichage, et YOLOv8 pour la détection des personnes. Le flux vidéo est capturé à partir d'une caméra fixe avec une résolution de 640×480 pixels. Une région d'intérêt (ROI) correspondant à la zone de la porte est définie afin de limiter la détection aux zones pertinentes et d'améliorer la fiabilité du comptage.

À chaque image, le modèle YOLOv8 effectue une inférence complète sur l'image et retourne les objets détectés sous forme de boîtes. Seules les détections appartenant à la classe « person » sont conservées. Parmi celles-ci, le programme sélectionne la personne la plus proche de la ligne virtuelle placée au centre de la porte, afin d'éviter les ambiguïtés lorsque plusieurs personnes sont visibles dans la scène.

Au centre de la boîte, la personne détectée est ensuite calculée et comparée à sa position dans l'image précédente. Le comptage est réalisé lorsque le centre traverse la ligne virtuelle dans un sens donné, ce qui correspond à une entrée ou une sortie de la salle. Le nombre total de personnes présentes est mis à jour dynamiquement et affiché en temps réel sur le flux vidéo.

Évolution et optimisation de l'approche YOLO

Dans une seconde version de l'approche basée sur YOLO, plusieurs optimisations ont été introduites afin d'améliorer les performances et la fluidité du système en temps réel. La latence d'acquisition vidéo a été réduite grâce à la limitation du tampon de la caméra, et l'inférence du modèle YOLO n'est plus réalisée à chaque image, mais toutes les quelques images. Les résultats intermédiaires sont mis en cache, ce qui permet de diminuer la charge de calcul tout en conservant une détection fiable pour le comptage.

Cette version intègre également des fonctionnalités de zoom et de déplacement de la vue, permettant d'ajuster la zone observée sans modifier la position physique de la caméra. Ces ajustements facilitent le centrage précis de la zone de la porte et améliorent la précision de la détection dans des environnements variés.

Une troisième version a ensuite été développée afin de renforcer l'ergonomie et la flexibilité du système. Dans cette version le zoom et le déplacement de la vue sont contrôlés directement via le clavier, ce qui simplifie l'interaction avec l'application. De plus, la possibilité d'exécuter le modèle YOLO sur le GPU via CUDA a été intégrée, permettant une accélération significative du traitement sur les plateformes NVIDIA. Ces améliorations rendent l'application plus réactive, plus adaptable et mieux adaptée à une utilisation en conditions réelles.

Résultats

Les résultats obtenus avec la méthode basée sur YOLOv8 montrent une amélioration significative par rapport à l'approche classique utilisant le traitement d'image. La détection des personnes est plus précise et beaucoup plus robuste face aux variations d'éclairage, aux changements de posture et à la présence d'objets dans la scène.

Le système parvient à détecter correctement les personnes entrant dans la salle, même lorsque plusieurs objets sont présents dans le champ de la caméra. Les faux positifs sont fortement réduits grâce à l'utilisation d'un modèle entraîné spécifiquement pour la détection d'objets, et en particulier de la classe « person ». Le comptage est stable et fiable tant que les personnes traversent la porte de manière identifiable.

L'utilisation du format ONNX permet une exécution efficace du modèle ainsi, le système fonctionne en temps réel et offre une meilleure précision globale que la méthode basée uniquement sur OpenCV.

Cependant, certaines limites subsistent, notamment lorsque plusieurs personnes passent simultanément très proches les unes des autres, ce qui peut entraîner des erreurs ponctuelles de comptage sans mécanisme de suivi multi-objets. Malgré cela, cette approche basée sur l'intelligence artificielle constitue une solution robuste et adaptée à des environnements réels.

Comparaison des deux approches

Dans ce projet, deux approches distinctes ont été mises en œuvre pour le comptage de personnes entrant dans une salle à l'aide d'une caméra fixe. La première repose sur des techniques de traitement d'image utilisant la bibliothèque « OpenCV », tandis que la seconde s'appuie sur une méthode d'intelligence artificielle basée sur l'algorithme YOLOv8. Ces deux approches poursuivent le même objectif, mais diffèrent fortement en termes de principes, de performances et de complexité.

L'approche basée sur OpenCV utilise principalement la soustraction de fond, l'extraction de contours et des règles géométriques pour détecter le passage d'une personne à travers une ligne virtuelle. Cette méthode ne nécessite aucun apprentissage préalable et peut fonctionner uniquement sur un processeur classique. Elle présente l'avantage d'être simple à implémenter, rapide à exécuter et peu de ressources. Toutefois, sa fiabilité dépend fortement des conditions de la scène. Les variations d'éclairage, les ombres ou la présence de plusieurs personnes simultanément peuvent entraîner des erreurs de détection et de comptage.

À l'inverse, l'approche basée sur YOLO repose sur un modèle de réseau de neurones convolutif pré-entraîné, capable de détecter les personnes de manière précise à partir d'images complexes. Cette méthode est beaucoup plus robuste face aux changements de luminosité, aux occlusions partielles et à la présence d'objets dans l'environnement. Le taux de faux positifs est considérablement réduit, ce qui améliore la fiabilité du comptage. En contrepartie, cette approche nécessite davantage de ressources de calcul et une mise en œuvre plus complexe, notamment l'installation de bibliothèques spécifiques et l'utilisation d'un GPU pour obtenir des performances en temps réel.

En termes de performances globales, la méthode YOLO offre de meilleurs résultats dans un environnement réel et dynamique, tandis que la méthode OpenCV reste adaptée à des scènes simples et contrôlées. Le choix entre ces deux approches dépend donc des contraintes de l'application, notamment la précision attendue, les ressources matérielles disponibles et la complexité de la scène observée. Dans le cadre de ce projet, l'utilisation de YOLO associée aux plateformes NVIDIA apparaît comme la solution la plus fiable et la plus évolutive pour le comptage automatique de personnes.

Conclusion

Ce projet a permis de concevoir une application complète de comptage de personnes à l'aide d'une caméra, en mettant en œuvre à la fois des méthodes classiques de traitement d'image et des approches basées sur l'intelligence artificielle. À travers l'utilisation d'OpenCV et de YOLO, il a été possible de comprendre les différences fondamentales entre ces deux méthodes, tant sur le plan algorithmique que sur le plan des performances et des contraintes matérielles.

Du point de vue de la filière ingénieur en systèmes embarqués, ce projet apporte une expérience concrète dans la conception d'un système temps réel intégrant capteurs, traitement logiciel et contraintes matérielles. Il met en évidence l'importance de l'optimisation des ressources, de la gestion de la latence et du choix des algorithmes en fonction du matériel disponible. L'exploitation des plateformes NVIDIA montre également le rôle central de l'accélération matérielle dans le déploiement d'applications de vision artificielle embarquées.