Rapport E3

INTÉGRATION DE SOLUTIONS IA POUR OPTIMISER LA RÉPONSE AUX BESOINS DU CLIENT

Développement d'un modèle (custom vision) pour détecter les personnes portant un masque et celles n'en portent pas via une caméra (Vision Ai Dev Kit)

DAO Ngoc Mai | Développeuse Data IA | Novembre 2020

1. Compréhension du besoin client

Dans le contexte de l'épidémie de COVID 19, tout le monde tente de contribuer à la lutte contre l'épidémie, notamment les entreprises technologiques.

Chez Exakis, où je fais mon alternance, nous souhaitons développer une application qui permet de contrôler le port de masques par les personnes en entreprise où tout le monde vient travailler tous les jours et est obligé de porter des masques. Le but de l'application pour détecter les personnes qui portent des masques et celles qui n'en portent pas via une caméra (Vision AI Dev Kit).

En fait, une telle application déjà existe. Je l'ai créé en appliquant un algorithme d'apprentissage en profondeur CNN avec différentes librairies Python (Tensorflow, Keras…) (cf. le projet E2). C'est pour ça, l'objectif de ce projet est d'utiliser une autre technologie pour améliorer la précision de modèle développé dans le projet E2. J'ai donc choisi un outil du Microsoft qui s'appelle « Azure Custom Vision » pour créer un modèle et puis l'intégrer dans une caméra (Vision AI Dev Kit). J'ai ensuite comparé les résultats avec le modèle que j'ai réalisé en utilisant le CNN.

2. Traduction technique et choix technique du projet

Afin de reconnaître si une personne porte un masque ou non, nous avons beaucoup de techniques différentes. Par exemple, nous pouvons programmer et utiliser des bibliothèques / paquets python (keras, tensorflow...) et appliquer des algorithmes d'apprentissage en profondeur (Réseau neuronal convolutif, Réseau neuronal récurrent...), ou nous pouvons utiliser des outils et des services disponibles à partir d'Azure Cloud Service comme Custom Vision.

Dans ce contexte, nous voulions faire une application avec une caméra dont nous disposions (caméra Vision-AI-DevKit - un kit de démarrage Azure IoT). De plus, Exakis est un Golden Partner de Microsoft, nous avons donc choisi un service Microsoft - Azure Custom Vision, qui est un service

d'intelligence artificielle et une plateforme de bout en bout pour appliquer la vision par ordinateur à notre scénario spécifique.

***Figure 1** : Caméra Vision AI Devkit*

3. Réponse technique mise en œuvre dans projet

Avec la documentation disponible de la caméra Vision AI Dev Kit sur internet (https://azure.github.io/Vision-AI-DevKit-Pages/docs), nous avons commencé à suivre les étapes pour télécharger le driver de la caméra, installer la caméra, puis entrainer le modèle et l'intégrer dans la caméra.

3.1 Ce dont nous aurons besoin

- Un compte Azure actif
- Une caméra Vision AI Dev Kit
- Des photos de personnes portant un masque ou non

3.2 Configurer la caméra Vision AI DevKit

Pour faire cette étape, nous devons réaliser deux sous étapes :

- Configurer la caméra avec le réseau Wi-Fi via le site : http://setupaicamera.ms.

- Configurer la caméra pour se connecter à Azure en tant qu'appareil IoT Edge

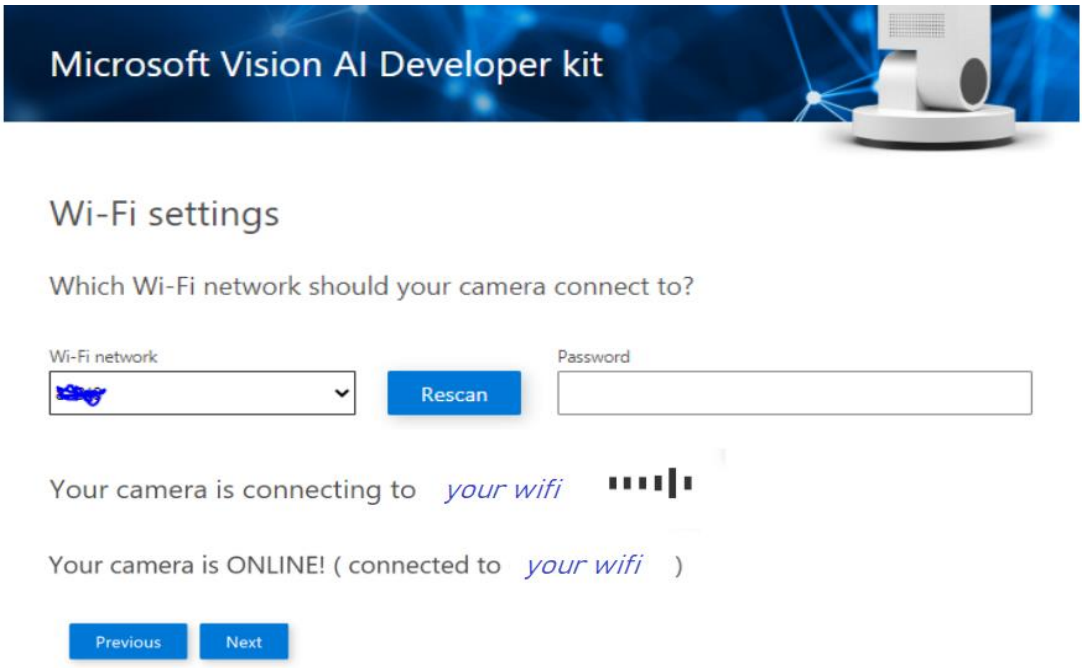

***Figure 2** : Configurer la caméra avec le réseau Wi-Fi*

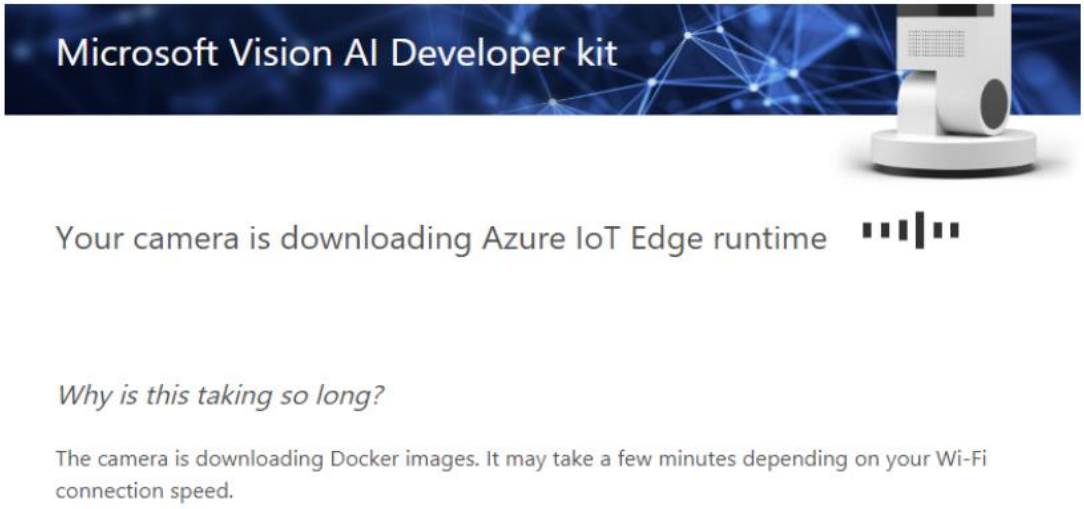

***Figure 3** : Configurer la caméra pour se connecter à Azure en tant qu'appareil IoT Edge*

Les 2 étapes ci-dessus sont détaillées dans le lien : https://azure.github.io/Vision-AI-DevKit-Pages/docs/Run_OOBE.

Pour cette raison, je n'expliquerai pas en détail cette étape dans ce rapport.

3.3 Créer un modèle d'IA de vision avec le service Azure Custom Vision

Avant de créer le modèle et de l'entraîner, nous avons besoin de données. Nos données dans ce cas seront des photos de personnes portant un masque ou non. Pour que l'application soit plus efficace, j'ai décidé d'utiliser la caméra Vision AI Dev Kit pour prendre des photos avec différents critères (voir tableau ci-dessous) comme données d'entraînement pour le modèle.

Nous avons un total de 255 images pour les données d'entrée.

	1 personne		2 personnes			Groupe			Negatif	
	Mask	No Mask	Mask	No Mask	Mask/No Mask	Mask	No Mask	Mask/No Mask		
Taille de l'objet										
Gros Plan	2	2	2	2	2	2	2	2	1	
Eloigne	2	2	2	2	2	2	2	2	1	
Mi-distance	2	2	2	2	2	2	2	2	1	
Eclairage										
Lumière jour	2	2	2	2	2	2	2	2	1	
Contre jour	2	2	2	2	2	2	2	2	1	
Lampe	2	2	2	2	2	2	2	2	1	
Contexte										
Arrière plan neutre	2	2	2	2	2	2	2	2	1	
Arrière plan objet	2	2	2	2	2	2	2	2	1	
Angle de vue										
Face	2	2	2	2	2	2	2	2	1	
Profil	2	2	2	2	2	2	2	2	1	
Tête haut	2	2	2	2	2	2	2	2	1	
Tête bas	2	2	2	2	2	2	2	2	1	
Style										
Blanc/Chir	2	2	2	2	2	2	2	2	1	
Motif	2	2	2	2	2	2	2	2	1	
Foulard	2	2	2	2	2	2	2	2	1	
	30	30	30	30	30	30	30	30	15	
TOTAL	60		90			90			15	255

Figure 4 *: Détail des données d'entrée pour le modèle*

Pour créer le modèle, nous devons d'abord nous connecter à l'Azure Custom Vision Service via https://www.customvision.ai.

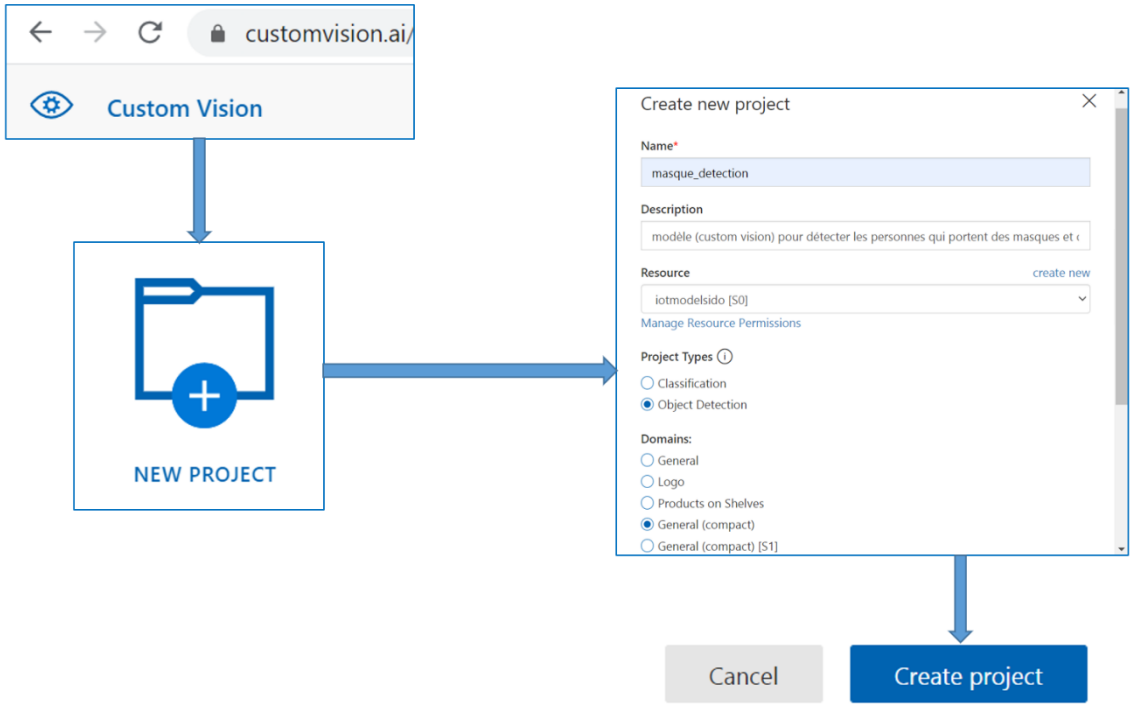

***Figure 5** : Etapes pour créer le modèle*

Comme indiqué dans la figure 5, nous pouvons voir que j'ai créé le modèle avec les paramètres suivants :

- Types de projets : Détection d'objet
- Domain : General (compact) - les modèles générés par les domaines compacts sont exportables pour s'exécuter localement.

Après avoir créé le modèle, on arrive sur une interface (figure 6), avec le nom de modèle, et 3 onglets différents : training images, performance, prédictions

Dans l'onglet « training images », j'ai ajouté des photos que j'ai prises auparavant avec la caméra.

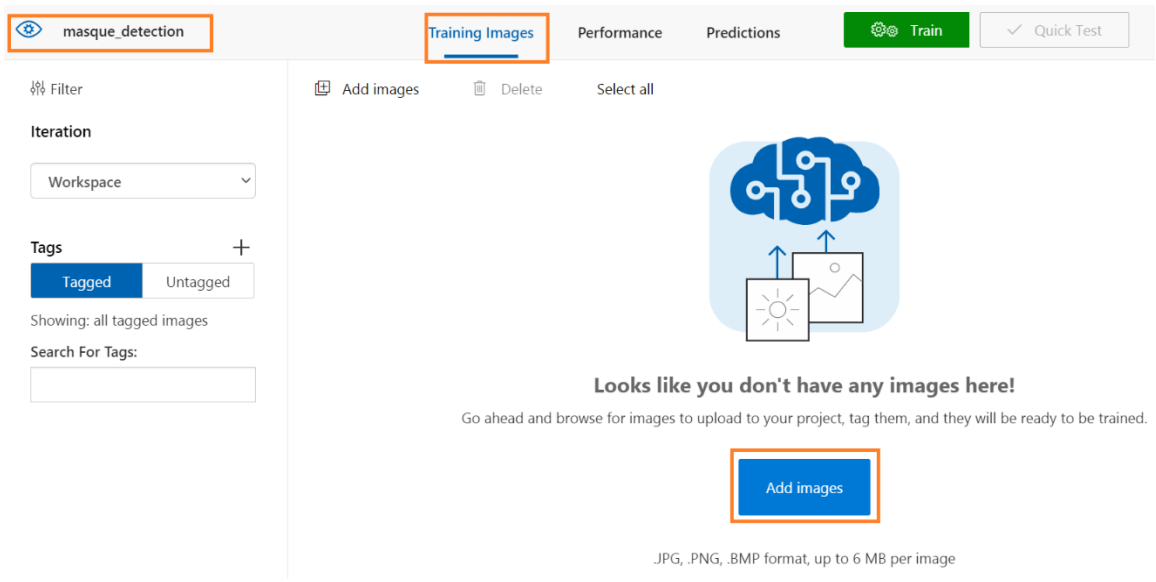

***Figure 6** : Ajout des images dans le modèle*

Une fois toutes les photos chargées, j'ai mis des tags sur chaque photo qui serviront pour la prédiction. Pour ce modèle, j'ai mis 2 tags : masque et no masque, que j'ai placé moi-même sur les zones concernées.

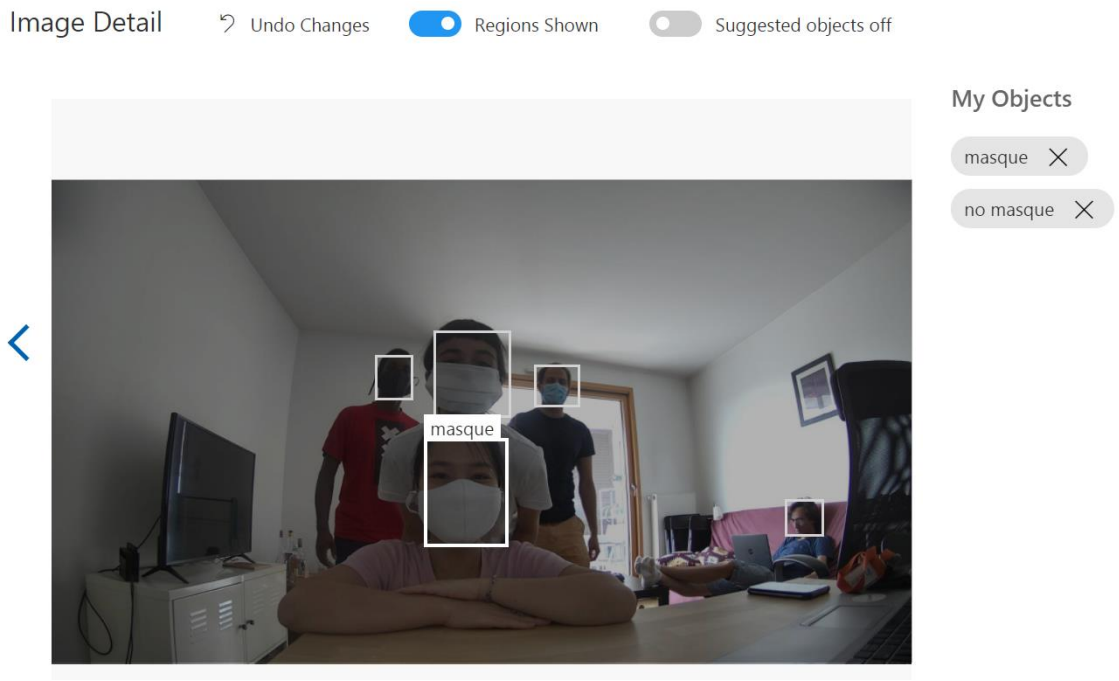

***Figure 7** : Ajout des tags pour chaque image*

Après avoir mis tous les tags pour toutes les photos, j'ai entraîné le modèle en cliquant sur le bouton « train ».

Dans l'onglet « performance », on a le résultat du modèle par tag (figure 8.1). Le service Custom Vision utilise les images que j'ai soumises pour l'entraînement afin de calculer la précision, le rappel et la précision moyenne. La précision et le rappel (recall) sont deux mesures différentes de l'efficacité d'un détecteur :

- **La précision** : indique la proportion des classifications identifiées qui étaient correctes. Par exemple, si le modèle a identifié que 100 images possédaient des masques et que 81 d'entre elles possédaient effectivement des masques, la précision est de 81 %.
- **Le rappel** : indique la proportion des classifications réelles qui ont été correctement identifiées. Par exemple, s'il y avait 100 images de masques et que le modèle en a identifié 37% comme étant des masques, le rappel est de 37 %.

Tag	Precision	Recall
masque	81.0%	37.0%
no masque	73.8%	34.1%

***Figure 8.1** : Résultat du modèle par tag*

La figure 8.2 montre le résultat moyen par tag du modèle avec les mesures « precision » et « recall ». Ce modèle a une précision de 77.4 % et un rappel de 35.5 %.

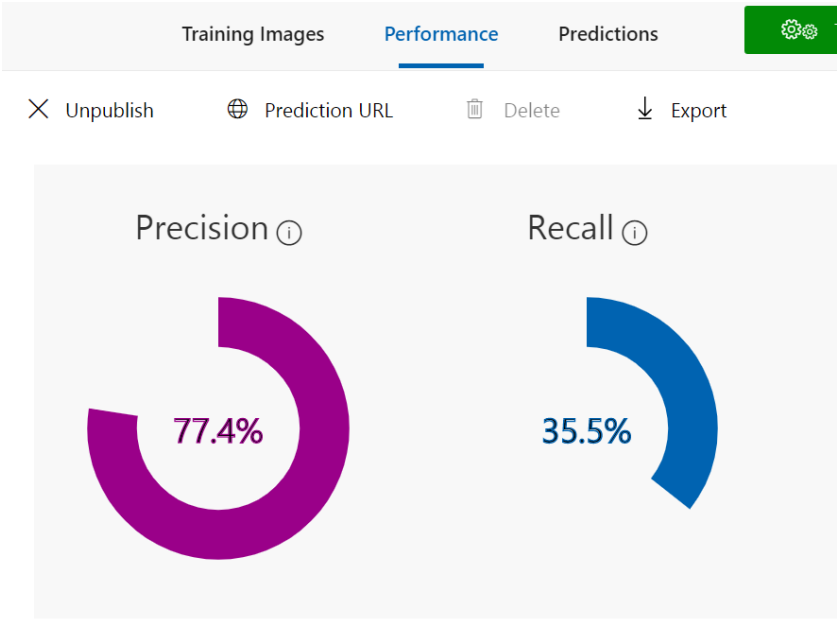

***Figure 8.2** : Résultat moyen du modèle*

En comparaison, ce modèle est plus précis que le modèle qui est appliqué l'algorithme CNN (projet E2), avec une précision moyenne de 62.5%.

3.4 Utiliser le modèle avec l'API de prédiction

Après avoir entraîné le modèle, on peut tester les images en les envoyant à l'API Prédiction.

Avant d'envoyer des images à l'API de prédiction, j'ai d'abord publié une nouvelle itération du modèle, ce que j'ai pu faire en sélectionnant Publier et en spécifiant un nom pour l'itération à publier. Cela rend le modèle accessible à l'API de prédiction de la ressource Azure Custom Vision.

Une fois que le modèle a été publié, je peux récupérer les informations requises en sélectionnant l'URL de prédiction.

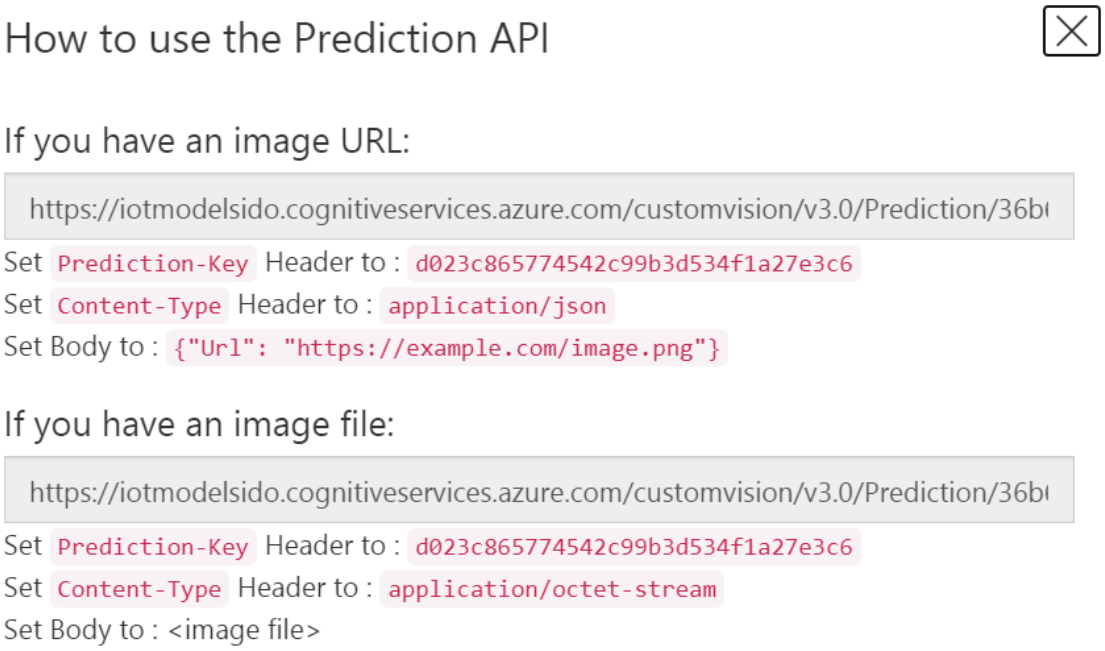

***Figure 9** : URL de prédiction*

En plus d'utiliser l'API, le service Custom Vision permet d'exporter le modèle pour l'exécuter en mode hors connexion. On peut incorporer le modèle exporté dans une application et l'exécuter localement sur un appareil pour la classification en temps réel. C'est pour cette raison que j'ai exporté le modèle et je vais maintenant le déployer sur la caméra Vision AI Dev Kit.

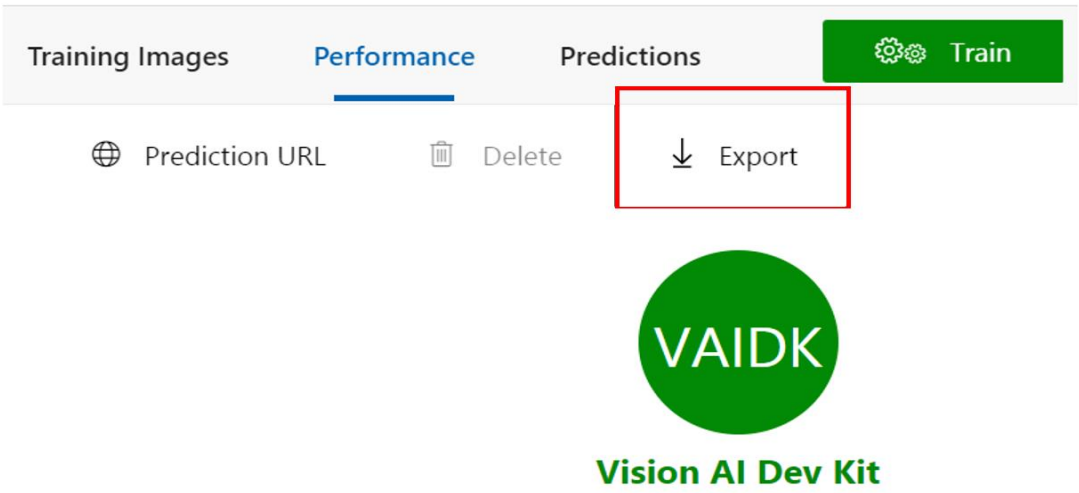

***Figure 10** : Export du modèle*

3.5 Déployer le modèle sur la caméra

Au lieu de télécharger le modèle depuis customvision.ai, j'ai appuyé sur le bouton droit de ma souris pour afficher les options. Puis, j'ai sélectionné Copier le lien.

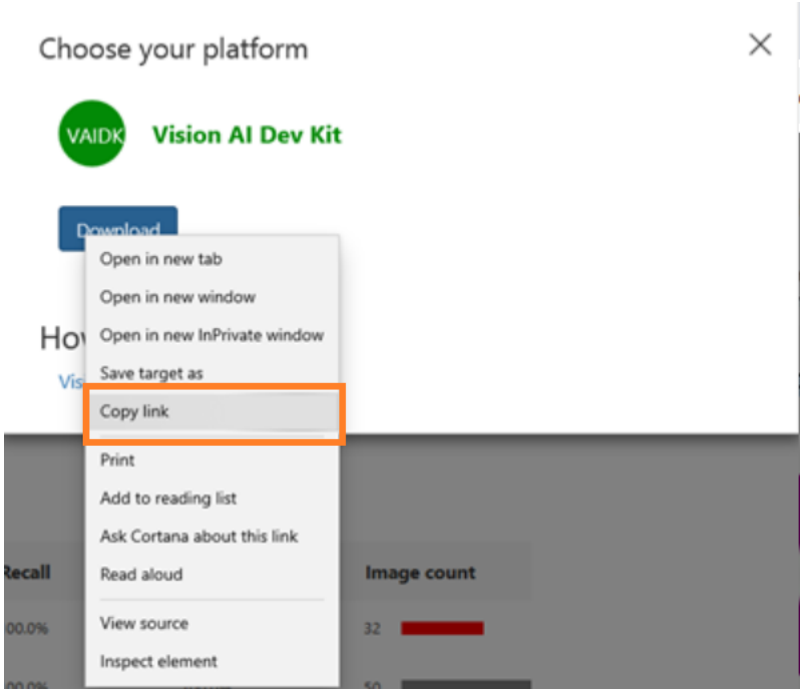

Figure 11 : *Copier le lien pour la caméra*

Pour déployer le modèle sur la caméra, j'ai suivi les étapes suivantes :

- Se connecter à http://portal.azure.com et accéder à la ressource IoT Hub que j'ai créé précédemment.
- Cliquer sur l'onglet IoT Edge, puis sur le périphérique IoT Edge nommé visionkit.
- Cliquer sur le nom du module « AI Vision Dev Kit Get Started Module », puis cliquer sur « Module Identity Twin »
- Coller le lien du modèle sur l'attribut « ModelZipUrl » (figure 12)
- Cliquer sur « Save »

Figure 12 : *Coller le lien sur « Module Identity Twin »*

Après quelques secondes, la caméra exécutera le modèle et on pourra donc l'utiliser.

4. Bilan du projet et les améliorations

Comme expliqué ci-dessus, la précision donne par le modèle (à l'issue de la phase d'apprentissage) est 77,4% (masque (81%) et no masque (73,8%)).

J'ai ensuite réalisé des tests avec des images différentes.

Ci-dessous, vous trouverez une image de la caméra qui identifie les personnes portant un masque et les personnes n'en portant pas.

On voit que, sur l'image 13.1, il y a 3 personnes, et aucune d'entre elles ne portent de masques. Elles sont correctement identifiées.

Sur la figure 13.2, il y a aussi 3 personnes, mais cette fois 2 portent un masque et une seule n'en porte pas, les masques sont toujours correctement détectés par le modèle et le « no masque » aussi.

J'ai ensuite essayé avec deux personnes, puis une personne et les résultats étaient toujours corrects.

Cependant, lorsque j'utilise des photos avec les personnes plus éloignées de la caméra, le modèle ne fonctionne plus correctement. Le taux

de reconnaissants des personnes avec masque ou sans masques devient très vite mauvais avec l'éloignement des sujets. C'est un point qui serait à améliorer :

- Soit en ajoutant des images de sujets éloignes dans la phase d'apprentissage de l'outil Custom Vision.
- Soit en développent ou utilisent un autre outil basé sur d'autres technologies

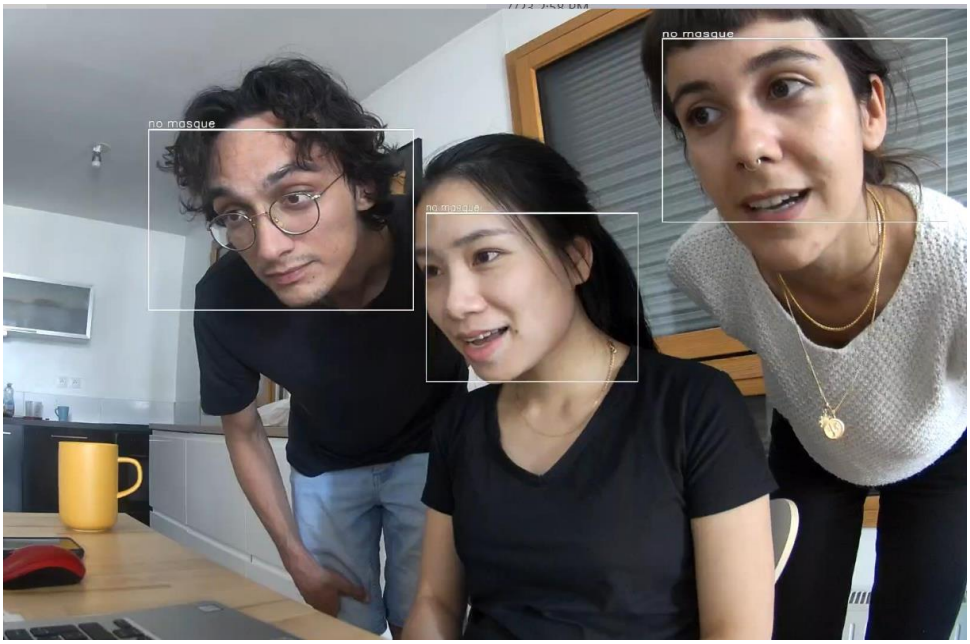

Figure 13.1 : *Identifier des personnes qui ne portant pas le masque*

Figure 13.2 : *Identification des personnes portant un masque et des personnes ne portant pas de masques*

5. Conclusion

J'ai pu réaliser et terminer ce projet en une semaine. Compte tenu de la durée assez courte du projet les résultats sont assez surprenants. Avec l'utilisation des services du Microsoft (Azure Cloud), nous n'avons pas besoin de passer du temps pour créer le modèle ou connaître le code, nous pouvons créer une application qui fonctionne efficacement et ce très rapidement. C'est l'intérêt de l'IA en général et des outils et services Azure en particulier, qui propose différents outils prêts à l'emploi.